



University of
Strathclyde
Glasgow

A novel approximation algorithm for the solution and
control of nonlinear and coupled systems

PhD Thesis

Daniel Johnston

Mechanical and Aerospace Engineering
University of Strathclyde, Glasgow

April 29, 2026

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

The copyright of this thesis belongs to the author under the terms of the United Kingdom Copyright Acts as qualified by University of Strathclyde Regulation 3.50. Due acknowledgement must always be made of the use of any material contained in, or derived from, this thesis.

Signed: Daniel Johnston

Date: 14 February 2026

Abstract

This thesis develops and assesses the “approximation via dual auxiliary problems” (ADAP) method, a novel algorithm for solving nonlinear, coupled dynamical systems. Many existing approximation techniques use simplified initial guesses that are augmented through the use of calibrated corrections – this is the foundational principle of the ADAP method. Departing from existing methods, the ADAP method is designed to enhance the performance of linear model-predictive control (MPC) algorithms, forming the ADAP-MPC strategy - a generalised architecture for the ADAP method’s integration with MPC, introduced in the second half of this thesis. The method is assessed through solving the governing dynamics of a large-scale, high-precision industrial manipulator. This manipulator is then considered as a further case study for the ADAP-MPC strategy. The ADAP methodology demonstrates improved accuracy and computational efficiency. In the examples and case study considered in the first half of this thesis, the observed absolute error is generally lower than the Lindstedt-Poincaré technique to which it is compared. In quantifying the computational burden associated with both algorithms, the ADAP approach is shown to have a reduced cost. In the second half of this thesis, the ADAP-MPC strategy achieves similar performance improvements in the control domain compared to a commonly-used approach. The transient and steady-state errors of the studied systems are generally improved with the ADAP-MPC strategy. Furthermore, the resulting control effort is shown to be either reduced or consistent with that of the feedback linearisation strategies considered. The significance of this work is the introduction of a novel approximation algorithm which permits the use of linear MPC while simultaneously improving accuracy and reducing computational burden in controlling complex nonlinear systems.

Contents

Abstract	ii
List of Figures	vi
List of Tables	xvii
Acknowledgements	xx
1 Introduction	2
1.1 Nonlinear ordinary differential equations and approximate analytical solutions	5
1.1.1 Perturbation methods	6
1.1.2 Non-perturbation approximation strategies	15
1.1.3 Linearisation techniques	28
1.1.4 Comparison of approximate analytical solution approaches	34
2 A novel approximation algorithm for the solution of nonlinear and coupled systems	39
2.1 A system of coupled Rayleigh oscillators	41
2.1.1 Problem specification	41
2.1.2 Construction and solution of the uncoupled system	42
2.1.3 Approximation and solution of the fully-coupled system	47
2.1.4 Results and discussion	53
2.2 A system of coupled Van der Pol oscillators	58
2.2.1 Problem specification	60

Contents

2.2.2	Construction and solution of the uncoupled system	60
2.2.3	Approximation and solution of the fully-coupled system	65
2.2.4	Results and discussion	68
2.3	A note on the effect of increasing the small parameter	73
2.4	Revision and standardisation of the algorithm	78
3	Case study Part 1: dynamics of the <i>FutureForge</i> manipulator	80
3.1	Mathematical modelling of the manipulator dynamics	82
3.1.1	Revision and updates to the <i>FutureForge</i> manipulator dynamics	83
3.2	Application of the ADAP methodology for solving the manipulator dy- namics	110
3.2.1	Construction and solution of the uncoupled system	110
3.2.2	Approximation and solution of the fully-coupled system	118
3.3	Alternative methods for comparison with the ADAP method's results	119
3.4	Implementation and simulation	124
3.5	Discussion	144
4	Introduction to model-predictive control and industrial manipulators	147
4.1	Fundamentals of model-predictive control	147
4.1.1	Strengths and weaknesses of model-predictive control	150
4.1.2	Applications of model-predictive control	151
4.2	An overview of industrial manipulators	154
4.2.1	Common kinematic architectures	156
4.3	Control of industrial manipulators	160
4.3.1	Classical and hard control	160
4.3.2	Intelligent and soft control	163
4.3.3	Hybrid control	165
5	Application of the novel approximation algorithm to model predictive controllers	167
5.1	An inverted pendulum	168
5.1.1	Model definition	168

Contents

5.1.2	Control system design and parameter specifications	171
5.1.3	Simulation of the proposed control system and analysis of its performance	174
5.2	An inverted pendulum on a mobile cart	177
5.2.1	Model definition	178
5.2.2	Control system design and parameter specifications	183
5.2.3	Simulation of the proposed control system and analysis of its performance	185
5.3	Conclusion	192
6	Case study Part 2: control of the <i>FutureForge</i> manipulator	195
6.1	Development of the virtual model of the manipulator	196
6.1.1	Configuration of the simulation environment	196
6.1.2	The ADAP-MPC strategy	197
6.1.3	Test manoeuvres	201
6.1.4	Examining the computational expense of the ADAP method . .	204
6.1.5	Simulated data	205
6.2	Simulation and testing under proposed control system	205
6.2.1	Trajectory tracking control results	206
6.2.2	Control effort analysis	218
6.2.3	Trajectory tracking results for manoeuvres including a payload .	227
6.2.4	Computational expense estimation	234
6.2.5	Discussion of ADAP-MPC controller performance for the <i>Future-</i> <i>Forge</i> manipulator case study	240
6.3	Conclusion	243
7	Conclusions	245
7.1	Further work	247
A	Expanded expressions for the constant coefficients in the <i>FutureForge</i> manipulator governing differential equations	249

Contents

B Statement on the inclusion of previous work	259
--	------------

List of Figures

1.1	The structure of this thesis and how each chapter aligns with its overall contribution.	4
2.1	Numerical solutions to Eq (2.1) using <i>ode45</i> in <i>MATLAB</i>	42
2.2	Numerical solutions for Eq (2.4), “ref sol” in the legend, compared to the corresponding $O(1)$ solutions to the generating Lindstedt-Poincaré problems.	44
2.3	Numerical solutions for Eq (2.4), “ref sol” in the legend, compared to the corresponding $O(\varepsilon)$ approximations based on the Lindstedt-Poincaré approach.	46
2.4	The errors accumulated by the $O(1)$ and $O(\varepsilon)$ approximations in solving Eq (2.1).	46
2.5	The approximate analytical solutions for Eq (2.1) achieved by the in-development ADAP method compared to the reference numerical solutions.	54
2.6	The errors produced in the approximate analytical solutions for Eq (2.1) achieved by the in-development ADAP method compared to the $O(1)$ asymptotic series approximation achieved via the Lindstedt-Poincaré technique.	54
2.7	The errors produced in the approximate analytical solutions for Eq (2.1) achieved by the in-development ADAP method compared to the $O(\varepsilon)$ asymptotic series approximation achieved via the Lindstedt-Poincaré technique.	55

List of Figures

2.8	The relative error for the ADAP method compared to the Lindstedt-Poincaré technique to $O(1)$ with varying values of the small parameter, ε , in Eq (2.1).	57
2.9	The relative error for the ADAP method compared to the Lindstedt-Poincaré technique to $O(\varepsilon)$ with varying values of the small parameter, ε , in Eq (2.1).	58
2.10	Reference numerical solutions produced by <i>Mathematica</i> 's <i>NDSolve</i> function for Eq (2.42) when $\varepsilon = 0.01$, $\mathbf{y}(t = 0) = [2, -1]^T$, and $\dot{\mathbf{y}}(t = 0) = [0, -\sqrt{3}]^T$	61
2.11	Numerical solutions for Eq (2.44) compared to the corresponding $O(1)$ solutions to the generating Lindstedt-Poincaré problems with $\varepsilon = 0.01$. .	62
2.12	The $O(1)$ and $O(\varepsilon)$ approximation to Eq (2.44) shown in (a) and (b) respectively with $\varepsilon = 0.01$	64
2.13	The errors accumulated by the $O(1)$ and $O(\varepsilon)$ approximations in solving Eq (2.44) via the Lindstedt-Poincaré technique.	65
2.14	The approximate analytical solutions for Eq (2.42) achieved by the ADAP method compared to the reference numerical solutions.	68
2.15	The relative errors produced in the approximate analytical solutions for Eq (2.42), when $\varepsilon = 0.01$, achieved by the ADAP method and the $O(\varepsilon)$ asymptotic series approximation achieved by the Lindstedt-Poincaré technique.	68
2.16	The relative errors produced in the approximate analytical solutions for Eq (2.42), when $\varepsilon = 0.01$, achieved by the ADAP method and the $O(\varepsilon)$ asymptotic series approximation achieved by the Lindstedt-Poincaré technique.	69
2.17	The relative error for the ADAP method compared to the Lindstedt-Poincaré technique to $O(1)$ with varying values of the small parameter, ε , in Eq (2.42).	70

List of Figures

2.18	The relative error for the ADAP method compared to the Lindstedt-Poincaré technique to $O(\varepsilon)$ with varying values of the small parameter, ε , in Eq (2.42).	71
2.19	The relative error for the ADAP method compared to the Lindstedt-Poincaré technique to $O(1)$ with varying values of the small parameter, ε , in Eq (2.61).	72
2.20	The relative error for the ADAP method compared to the Lindstedt-Poincaré technique to $O(\varepsilon)$ with varying values of the small parameter, ε , in Eq (2.61).	73
2.21	Reference numerical solutions for Eq (2.1), when $\varepsilon = 1$, found via the <i>NDSolve</i> function in <i>Mathematica</i>	74
2.22	Relative error in the Lindstedt-Poincaré technique and ADAP methodology when applied to Eq (2.1) with $\varepsilon = 1$	75
2.23	Reference numerical solutions for Eq (2.42), when $\varepsilon = 1$, found via the <i>ode45</i> function in <i>MATLAB</i>	76
2.24	Relative error in the Lindstedt-Poincaré technique and ADAP methodology when applied to Eq (2.42) with $\varepsilon = 1$	76
2.25	Reference numerical solutions for Eq (2.61), when $\varepsilon = 1$, found via the <i>NDSolve</i> function in <i>Mathematica</i>	77
2.26	Relative error in the Lindstedt-Poincaré technique and ADAP methodology when applied to Eq (2.61) with $\varepsilon = 1$	77
3.1	A schematic diagram of the <i>FutureForge</i> manipulator first produced in [59].	82
3.2	A depiction of the updated geometry of the <i>FutureForge</i> manipulator.	85
3.3	The assembly of components - consisting of the vertical link BC, the push link ZS, and the push rocker DST - which allows the end-effector to be extended horizontally when the generalised coordinate β is actuated.	87
3.4	The base superstructure of the <i>FutureForge</i> manipulator.	88
3.5	The vertical boom, link AD, in the <i>FutureForge</i> manipulator	89

List of Figures

3.6	Detailed view of the dimensions and angles defining the location of the centre of mass (CoM) of the vertical boom.	89
3.7	The vertical link, BC, in the <i>FutureForge</i> manipulator.	91
3.8	Detailed view of the dimensions and angles defining the location of the centre of mass (CoM) of the vertical link. Note that the detailed view of the angles marked at C is required for the later determination of the location of the CoM for the push link ZS.	91
3.9	The crank components, AJK, in the <i>FutureForge</i> manipulator.	92
3.10	Detailed view of the dimensions and angles defining the location of the centre of mass (CoM) of the crank components.	93
3.11	The vertical struts, JH, in the <i>FutureForge</i> manipulator.	94
3.12	Detailed view of the dimensions and angles defining the location of the centre of mass (CoM) of the crank components.	94
3.13	The rear rocker, CDE, in the <i>FutureForge</i> manipulator.	95
3.14	Detailed view of the dimensions defining the location of the centre of mass (CoM) of the rear rocker.	95
3.15	The horizontal boom, link DG, in the <i>FutureForge</i> manipulator.	97
3.16	Detailed view of the dimensions and angles defining the location of the centre of mass (CoM) of the horizontal boom.	97
3.17	The horizontal boom, link DG, in the <i>FutureForge</i> manipulator.	98
3.18	The push link, link ZS, in the <i>FutureForge</i> manipulator.	99
3.19	The push rocker, component DST, in the <i>FutureForge</i> manipulator. . . .	101
3.20	The lower horizontal link, link TU, in the <i>FutureForge</i> manipulator. . .	102
3.21	The front rocker, component FGM, in the <i>FutureForge</i> manipulator. . .	103
3.22	The forward vertical boom, link GW, in the <i>FutureForge</i> manipulator. .	104
3.23	The pitch cylinder assembly, link MN, in the <i>FutureForge</i> manipulator. .	105
3.24	The roll rocker, link NW, in the <i>FutureForge</i> manipulator.	106
3.25	The regular trapezoid ADGW which validates Cartmell et al.'s substitution of $\gamma = \frac{\pi}{2} - \beta$	108

List of Figures

3.26	A comparison of the approximation of $\alpha(t)$ by $\alpha_0(t)$ via the numerical solution of: Eq (3.37), “Reference”; Eq (3.48), “Opt1”; and, Eq (3.49), “Opt2”.	115
3.27	A comparison of the approximation of $\beta(t)$ by $\beta_0(t)$ via the numerical solution of: Eq (3.38), “Reference”; Eq (3.48), “Opt1”; and, Eq (3.49), “Opt2”.	116
3.28	Asymptotic approximations of the solutions of the <i>FutureForge</i> manipulator system in Case 1 using 1 st -order Taylor Series approximations. . .	125
3.29	ADAP-based approximations of the solutions of the <i>FutureForge</i> manipulator system in Case 1 using 1 st -order Taylor Series approximations. . .	126
3.30	A comparison between the asymptotic and ADAP-based approximations of the solutions of the <i>FutureForge</i> manipulator system in Case 1 using 1 st -order Taylor Series approximations.	127
3.31	A comparison between the asymptotic and ADAP-based approximations of the solutions of the <i>FutureForge</i> manipulator system in Case 1 using 3 rd -order Taylor Series approximations.	128
3.32	A comparison between the asymptotic and ADAP-based approximations of the solutions of the <i>FutureForge</i> manipulator system in Case 2 using 1 st -order Taylor Series approximations.	129
3.33	A comparison between the asymptotic and ADAP-based approximations of the solutions of the <i>FutureForge</i> manipulator system in Case 2 using 3 rd -order Taylor Series approximations.	130
3.34	A comparison between the asymptotic and ADAP-based approximations of the solutions of the <i>FutureForge</i> manipulator system in Case 3 using 1 st -order Taylor Series approximations.	131
3.35	A comparison between the asymptotic and ADAP-based approximations of the solutions of the <i>FutureForge</i> manipulator system in Case 3 using 3 rd -order Taylor Series approximations.	132

List of Figures

3.36	A comparison between the asymptotic and ADAP-based approximations of the solutions of the <i>FutureForge</i> manipulator system in Case 4 using 1 st -order Taylor Series approximations.	133
3.37	A comparison between the asymptotic and ADAP-based approximations of the solutions of the <i>FutureForge</i> manipulator system in Case 4 using 3 rd -order Taylor Series approximations.	134
3.38	A comparison between the asymptotic and ADAP-based approximations of the solutions of the <i>FutureForge</i> manipulator system in Case 5 using 1 st -order Taylor Series approximations.	135
3.39	A comparison between the asymptotic and ADAP-based approximations of the solutions of the <i>FutureForge</i> manipulator system in Case 5 using 3 rd -order Taylor Series approximations.	136
3.40	A comparison between the asymptotic and ADAP-based approximations of the solutions of the <i>FutureForge</i> manipulator system in Case 6 using 1 st -order Taylor Series approximations.	137
3.41	A comparison between the asymptotic and ADAP-based approximations of the solutions of the <i>FutureForge</i> manipulator system in Case 6 using 3 rd -order Taylor Series approximations.	138
3.42	A comparison between the asymptotic and ADAP-based approximations of the solutions of the <i>FutureForge</i> manipulator system in Case 7 using 1 st -order Taylor Series approximations.	139
3.43	A comparison between the asymptotic and ADAP-based approximations of the solutions of the <i>FutureForge</i> manipulator system in Case 7 using 3 rd -order Taylor Series approximations.	140
3.44	A comparison between the asymptotic and ADAP-based approximations of the solutions of the <i>FutureForge</i> manipulator system in Case 8 using 1 st -order Taylor Series approximations.	141
3.45	A comparison between the asymptotic and ADAP-based approximations of the solutions of the <i>FutureForge</i> manipulator system in Case 8 using 3 rd -order Taylor Series approximations.	142

List of Figures

3.46	A comparison between the asymptotic and ADAP-based approximations of the solutions of the <i>FutureForge</i> manipulator system in Case 9 using 1 st -order Taylor Series approximations.	143
3.47	A comparison between the asymptotic and ADAP-based approximations of the solutions of the <i>FutureForge</i> manipulator system in Case 9 using 3 rd -order Taylor Series approximations.	144
5.1	A diagram of the simple inverted pendulum system.	169
5.2	Block diagram representation of the ADAP-based control approach . . .	172
5.3	Block diagram representation of the MPC control approach using feedback linearisation as described by the system in Eq (5.8).	173
5.4	Block diagram representation of the MPC control approach using feedback linearisation as described by the system in Eq (5.9).	173
5.5	Position and control torque graphs for the inverted pendulum using three model linearisation strategies.	175
5.6	Torque-vs-displacement plots used to indicate the work done by actuators in achieving each of the three responses.	177
5.7	A diagram of the cart-mounted pendulum system.	178
5.8	A block diagram representation of how the two vectors $\mathbf{v}$ and α contribute to the actuation of the cart-pendulum system.	182
5.9	Block diagram representation of the MPC control strategy for the cart-mounted pendulum system using feedback linearisation strategy FL01. .	184
5.10	Block diagram representation of the MPC control strategy for the cart-mounted pendulum system using feedback linearisation strategy FL02. .	184
5.11	Block diagram representation of the MPC control strategy for the cart-mounted pendulum system using feedback linearisation strategy FL03. .	184
5.12	Results of the zero disturbance test case with constant pendulum position setpoint and sinusoidal cart position setpoint.	186
5.13	Cart control effort exerted through F by each of the control strategies to achieve the trajectory tracking shown in Fig 5.12.	187

List of Figures

5.14	Pendulum control effort exerted through τ by each of the control strategies to achieve the trajectory tracking shown in Fig 5.12.	188
5.15	Results of the test case with constant pendulum position setpoint and sinusoidal cart position setpoint with an imposed disturbance on each manipulated variable as described by Eq (5.37).	189
5.16	Cart control effort exerted through F by each of the control strategies to achieve the trajectory tracking shown in Fig 5.15.	190
5.17	Pendulum control effort exerted through τ by each of the control strategies to achieve the trajectory tracking shown in Fig 5.15.	191
6.1	A simplified block diagram representation of the ADAP-MPC control strategy applied to the <i>FutureForge</i> manipulator, reproduced from [44].	199
6.2	2D workspace of the manipulator divided into nine characteristic positions. These positions are given a two-letter designation according to their proximity and elevation with respect to the manipulator base. Reading these from top left to bottom right: “near upper”, “mid upper”, “far upper”, “near mid”, “mid mid”, “far mid”, “near lower”, “mid lower”, and “far lower”.	202
6.3	End-effector Cartesian positions, (x_{ee}, y_{ee}) , for Tests 2 and 3.	203
6.4	Reference trajectories for Test 1.	207
6.5	Reference trajectories and absolute error measures for Test 1 across the interval $t \in [0, 20]$ seconds.	208
6.6	Reference trajectories and absolute error measures for Test 1 across the interval $t \in [20, 35]$ seconds.	209
6.7	Reference trajectories and absolute error measures for Test 1 across the interval $t \in [65, 100]$ seconds.	210
6.8	Reference trajectories and absolute error measures for Test 2 across the interval $t \in [10, 20]$ seconds.	211
6.9	Reference trajectories and absolute error measures for Test 2 across the interval $t \in [40, 50]$ seconds.	212

List of Figures

6.10	Absolute error in the trajectory tracking for $\beta(t)$ for Test 2 across the interval $t \in [50, 70]$ seconds.	213
6.11	Reference trajectories and absolute error measures for Test 2 across the interval $t \in [70, 80]$ seconds.	214
6.12	Reference trajectories and absolute error measures for Test 4 across the interval $t \in [10, 20]$ seconds.	215
6.13	Reference trajectories and absolute error measures for Test 4 across the interval $t \in [240, 250]$ seconds.	216
6.14	Reference trajectories and absolute error measures for Test 4 across the interval $t \in [470, 480]$ seconds.	217
6.15	Illustration of control effort for both control strategies across the $t \in [0, 20]$ s interval of Test 1.	218
6.16	Illustration of control effort for both control strategies across the $t \in [20, 35]$ s interval of Test 1.	219
6.17	Illustration of control effort for both control strategies across the $t \in [65, 100]$ s interval of Test 1.	220
6.18	Illustration of control effort for both control strategies across the $t \in [10, 20]$ s interval of Test 2.	221
6.19	Illustration of control effort for both control strategies across the $t \in [40, 50]$ s interval of Test 2.	222
6.20	Illustration of control effort for both control strategies across the $t \in [70, 80]$ s interval of Test 2.	223
6.21	Illustration of control effort for both control strategies across the $t \in [10, 20]$ s interval of Test 4.	224
6.22	Illustration of control effort for both control strategies across the $t \in [240, 250]$ s interval of Test 4.	225
6.23	Illustration of control effort for both control strategies across the $t \in [470, 480]$ s interval of Test 4.	226

List of Figures

6.24	Percentage reduction in the control effort required by the ADAP-MPC method. Positive values indicate reduced effort; negative values indicate increased effort.	227
6.25	Reference trajectories and absolute error measures for Test 3 across the interval $t \in [10, 20]$ seconds.	228
6.26	Reference trajectories and absolute error measures for Test 3 across the interval $t \in [40, 50]$ seconds.	229
6.27	Reference trajectories and absolute error measures for Test 3 across the interval $t \in [70, 80]$ seconds.	230
6.28	Reference trajectories and absolute error measures for Test 5 across the interval $t \in [10, 20]$ seconds.	231
6.29	Reference trajectories and absolute error measures for Test 5 across the interval $t \in [240, 250]$ seconds.	232
6.30	Reference trajectories and absolute error measures for Test 5 across the interval $t \in [470, 480]$ seconds.	233
6.31	Mean number of function evaluations required for each prediction horizon for both linearised prediction models. Reproduced and revised from [44].	235
6.32	Error between the standard and ADAP prediction models found via <i>ode45</i> when compared with the reference (target) solution (using the same solver). Altered since first presented in [44].	236
6.33	Error between the standard and ADAP prediction models found via <i>ode89</i> when compared with the reference (target) solution (using the same solver). Altered since first presented in [44].	237
6.34	Error between the standard and ADAP prediction models found via <i>ode113</i> when compared with the reference (target) solution (using the same solver). Altered since first presented in [44].	238
6.35	Error between the standard and ADAP prediction models found via <i>ode15s</i> when compared with the reference (target) solution (using the same solver). Altered since first presented in [44].	239

List of Figures

6.36	Error between the standard and ADAP prediction models found via <i>ode23t</i> when compared with the reference (target) solution (using the same solver). Altered since first presented in [44].	240
6.37	Trajectories obtained for β in Test 3 across the interval $t \in [10, 12]$ seconds, showing low damping properties in the transient responses. . .	242

List of Tables

1.1	Stability implications of different eigenvalues obtained for linearising the system.	30
1.2	A summary of the strengths and weaknesses of the approximate analytical techniques discussed in Chapter 1.	38
3.1	The constant values of the coefficients occurring in Eqs (3.37) and (3.38). Note the value $c_d = 10^5$ is chosen to reflect the linear viscous damping observed in the <i>FutureForge</i> manipulator.	113
3.2	The largest constant values of the coefficients occurring in Eqs (3.37) and (3.38).	113
3.3	Cases to be examined through testing the application of the ADAP methodology to the <i>FutureForge</i> manipulator dynamics.	124
5.1	MPC settings used for the control of the inverted pendulum.	174
5.2	Performance measures associated with the three responses shown in Fig 5.5.	176
5.3	MPC settings used for the control of the cart-mounted pendulum system.	185
5.4	Comparison of the control effort exertions for each control approach across the zero disturbance and disturbance-included test cases for the cart-mounted pendulum system.	191
5.5	Comparison of the control effort exertions for each control approach across the zero disturbance and white Gaussian noise (WGN) disturbance test cases for the cart-mounted pendulum system.	192

List of Tables

6.1	MPC parameter settings used in the control of the <i>FutureForge</i> manipulator, reproduced and revised from [44].	201
6.2	Parameter specification for Tests 2-5, indicating variations in payload mass, step penetration depth, and step rate.	203
6.3	The integral of squared error for each coordinate for both control strategies across the time intervals shown for Test 1 in Figs 6.5-6.7.	210
6.4	The integral of squared error for each coordinate for both control strategies across the time intervals shown for Test 2 in Figs 6.8-6.11.	214
6.5	The integral of squared error for each coordinate for both control strategies across the time intervals shown for Test 4 in Figs 6.12-6.14.	217
6.6	The integral of squared error for each coordinate for both control strategies across the time intervals shown for Test 3 in Figs 6.25-6.27.	230
6.7	The integral of squared error for each coordinate for both control strategies across the time intervals shown for Test 5 in Figs 6.28-6.30.	233
6.8	Min, max, and mean number of function evaluations required for each solver choice in MATLAB. Note that the mean values are rounded to the nearest integer value. Table first produced in [44].	235

Acknowledgements

I owe the completion of this thesis to the unwavering support and encouragement of my supervisor, Prof. Matthew Cartmell. Matthew was an inspiration during my undergraduate studies, and studying under him was a great motivating factor in my choice to undertake a PhD. If I can develop into a researcher and teacher with an ounce of his compassion, patience, and rigour, I will be truly happy. My secondary supervisor, Prof. Brad Wynne, also has my sincere gratitude. In his demanding role within the department, Brad's kindness, encouragement, and support (particularly in navigating dialogues with the project's industrial partners) were invaluable.

I would also like to extend my thanks to Dr. Vikram Pakrashi and Prof. Ivanna Kovacic for their expertise and guidance in some of the early publications and formative experiences of this project. Combined with the expertise and patience of individuals such as Chris McCrory (at Clansman Dynamics Ltd.), this work would not be a success without their input and time.

The longer-than-planned duration of this PhD was partly due to my appointment as a full-time member of the teaching team in the Department of Mechanical and Aerospace Engineering. I wouldn't exchange this path for the world; teaching is where I have found a home for my passion in engineering and mathematics. Moreover, the support of my colleagues throughout this time has been invaluable. In particular, I would like to extend my thanks to Dr. Emma Henderson, Dr. Andrew McLaren, and Dr. Gemma Houston-Scott for their continued encouragement, guidance, and kindness. Gratitude is owed to the full MAE teaching team, and many more pages could be given

Chapter 0. Acknowledgements

over to listing the instances of support and joy I have experienced as part of this team.

To Amy, my fiancée... there's not an ounce of me that hasn't been made more complete since you came into my life. Thank you for being you, and for helping me be me.

*Every wave that broke me
Every song that wrote me
Every dawn that woke me
Was to get me home to you, see
Every soul that left me
Every heart that kept me
The strangers that protected me
To bring me back to you.*

-

U2, "Landlady".

To my friends, especially those with whom I studied alongside as an undergraduate. Thank you for the hours spent on D&D, teaching me that sometimes the dice roll poorly, but the most important things are how you interpret the result and the character choices you make in the aftermath.

Lastly, thank you to my family (including our dog, Zaz). In particular - to Mum, Dad, and James - you are all sources of deep inspiration for the person I try to be, and for the good I try to see in others. Thank you for your patience and love.

Chapter 1

Introduction

A major challenge in engineering dynamics is in the analytical intractability of the mathematical models which describe systems to be analysed or controlled. Without approximations, these models are nonlinear and coupled. While approximate linear models are prevalent in many engineering problems (due to their ability to render system analytically tractable), they are not always sufficient when considering complex physical systems such as the large-scale industrial manipulator considered in this work. Existing control strategies built on these linearised tractable models, such as linear model-predictive control (MPC), can be effective and computationally efficient; however, an associated trade-off of these strategies is the inability to manage inherently nonlinear phenomena which are omitted from the linear models. This thesis is motivated by the desire for a methodology which bridges this gap: permitting the use of linear control strategies while improving accuracy and reducing the computational cost associated with the control of nonlinear, coupled systems.

The contribution of this thesis is in the development of a generalised architecture for an approximate analytical approach which enhances the effectiveness of linear model-predictive control. To develop and demonstrate this approach, the thesis is divided into two parts. Part one, Chapters 1 to 3, deals with approximate analytical methods, develops the unique approach posed in this thesis, and demonstrates its application to a particular case study. Part two, Chapters 4 to 6, further develops this approximate

Chapter 1. Introduction

analytical method into a generalised architecture which enhances the capability of linear model-predictive control. A description of each of these chapters is detailed below, while the summary of their contribution to the thesis is presented in Fig 1.1.

The present chapter, Chapter 1, offers a thorough introduction to approximate analytical methods because understanding the approaches in these methods is key to understanding the design of the later-developed generalised architecture.

Chapter 2 begins to develop the novel approach posed in this thesis. The method is explained in the context of demonstrated applications to a pair of coupled nonlinear oscillator systems. Through this chapter, a comparison between the novel approach and a well-established alternative approach is drawn. The final section of this chapter sees a standardisation of the novel approximate analytical solution algorithm.

Chapter 3 continues this demonstration in the context of the FutureForge manipulator. Although revisited later in the thesis, this case study offers a challenging and practically insightful perspective on the method's application and effectiveness.

Chapter 4, marking the beginning of the second part of this thesis, introduces several new, important concepts which were not required for a complete understanding of the preceding part. This chapter introduces concepts such as model-predictive control and industrial manipulators. While the case study in Chapter 3 ingresses on this latter topic, the understanding of the physical system is more beneficial for the latter half of this thesis than the former.

Chapter 5 follows Chapter 4 just as Chapter 2 follows Chapter 1: it expands on the former's ideas and demonstrates the method through its application to two well-studied control problems. In this chapter, linear model-predictive control is applied using the generalised architecture developed from the approximate analytical methods developed in part one of this thesis.

Chapter 1. Introduction

Chapter 6 then applies this method to the case study of the FutureForge manipulator. In this, its effectiveness is evaluated in a similar way to previous chapters of this thesis. Assessments of the trade-offs involved in selecting this method, over widely-used alternatives, are also presented.

Lastly, conclusions drawn from both parts of this thesis are presented in Chapter 7 alongside comments for the further thesis that should be undertaken by the author and/or other interested parties.

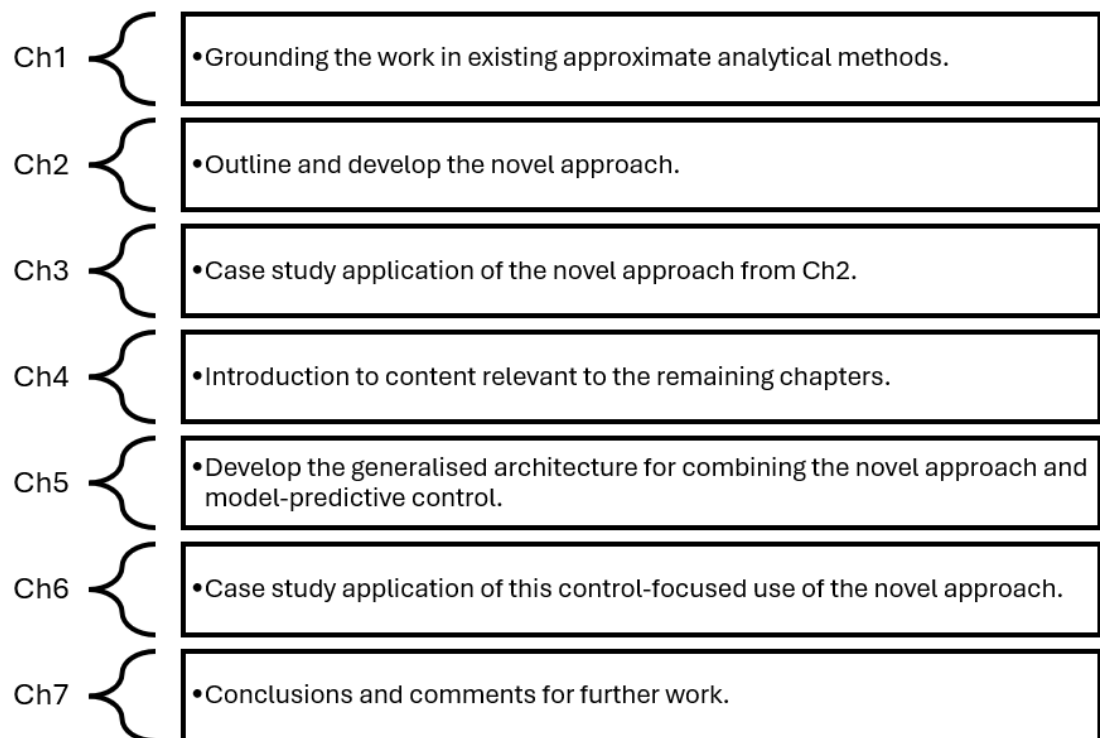


Figure 1.1: The structure of this thesis and how each chapter aligns with its overall contribution.

1.1 Nonlinear ordinary differential equations and approximate analytical solutions

Ordinary differential equations (ODEs) represent a cornerstone of mathematical modelling, providing the language to describe how systems evolve with respect to a single independent variable, typically time or a spatial coordinate in mechanical engineering contexts. In many applications, such governing equations are approximated as linear, meaning the dependent variable and its derivatives appear only to the first power and are not the input to any transcendental function. Such systems are well-understood and possess exact, closed-form solutions, forming the basis of classical engineering dynamics.

The physical reality, however, is that the majority of systems encountered in science and engineering are inherently nonlinear. Phenomena such as the dissipative effects of air resistance, the complex interactions within a biological ecosystem, or the behaviour of materials under large deformations cannot be captured accurately by linear models. Furthermore, when a system comprises several interacting components, its dynamics are described not by a single equation, but by a system of coupled ODEs, wherein the evolution of each component is dependent on the state of the others.

Such nonlinear, coupled ODEs can offer a high-fidelity representation of complex phenomena. But, they also present a profound mathematical challenge. Unlike their linear counterparts, many nonlinear systems are analytically intractable, meaning that an exact, closed-form solution cannot be found. This analytical intractability necessitates the use of alternative strategies to understand their behaviour. Consequently, a significant body of research has been devoted to the development of approximate analytical solutions: methods designed to yield solutions that, while not exact, are sufficiently accurate for the purposes of prediction, design, and analysis. The following sections of this chapter will present a detailed review of some of the most prevalent of these techniques.

Chapter 1. Introduction

Given the analytical intractability of many nonlinear systems, a diverse array of approximate analytical solution techniques has been developed and refined through established and ongoing research. These techniques can be broadly classified into distinct classes, each with its own underlying principles and domains of application. This review will consider some of the most significant approaches, but the list contained here is by-no-means exhaustive. The techniques selected for review here are chosen because of their demonstration of the core principle shared with the novel approach developed in Chapter 2: corrections being made to an initial estimate of the target solution. The methods' ubiquity and applicability to control systems is also a factor leading to the inclusion of some of the methods herein. These methods are grouped into: perturbation methods, which rely on the existence of a small parameter associated with the nonlinear terms; non-perturbation methods, which are applicable even in the absence of such a small parameter; and, linearisation techniques, which are fundamental to stability analysis and control systems design.

1.1.1 Perturbation methods

One well-studied class of approaches to solving nonlinear ordinary differential equations is a class known as “perturbation methods”. In this section, several prevalent perturbation methods are explained. However, to understand the challenge these methods were designed to answer, an explanation of the fundamental principles of perturbation methods is offered here.

Firstly, it is important to understand what a perturbation problem is. This is a problem very close to one that can be solved exactly, but with at least one slight deviation (or “perturbation”) from this exactly solvable problem. Although perhaps overly simple for a fully-detailed example, one might consider the easily solvable simple harmonic oscillator ODE. A perturbation to this exactly solvable problem is to include a small degree of air resistance or another source of nonlinear damping. A small factor, commonly denoted by ε , is taken as a coefficient of this perturbation term. The generalised strategy of all perturbation methods is then to solve the simpler, related,

Chapter 1. Introduction

exactly-solvable problem first, before then considering the perturbation dynamics and introducing “corrections” to the exactly solvable solution. The structure of the resulting solution is then a power series of the form

$$y(t) = y_0(t) + \varepsilon y_1(t) + \varepsilon^2 y_2(t) + \cdots \quad (1.1)$$

where ε is the small perturbation factor; $y_0(t)$ is the exact solution to the simpler, “unperturbed” problem; and, subsequent $y_n(t)$ ($n > 0$) terms are the corrections to these unperturbed dynamics. While relatively simple in conception, this class of methods is immensely powerful as will be detailed further in this section. However, the key reason modifications are made to this most basic description – by techniques such as Lighthill’s technique, the Lindstedt-Poincaré technique, or the method of multiple scales – is because of a common pitfall of following only this basic description.

Consider the apparently simple oscillator dynamics described by the hardening Duffing oscillator

$$\ddot{y} + y + \varepsilon y^3 = 0 \quad (1.2)$$

with initial conditions of $y(0) = 1$ units and $\dot{y} = 0$ units/s. Substituting the assumed solution form of Eq (1.1) into this ODE gives the following problem.

$$\begin{aligned} & (\ddot{y}_0(t) + \varepsilon \ddot{y}_1(t) + \varepsilon^2 \ddot{y}_2(t) + \cdots) \\ & + (y_0(t) + \varepsilon y_1(t) + \varepsilon^2 y_2(t) + \cdots) \\ & \varepsilon (y_0(t) + \varepsilon y_1(t) + \varepsilon^2 y_2(t) + \cdots)^3 = 0 \end{aligned} \quad (1.3)$$

The unperturbed problem, known as “the $O(1)$ problem” or “the generating problem”, may then be written as

$$\ddot{y}_0 + y_0 = 0 \quad (1.4)$$

which can be trivially solved to give $y_0(t) = \cos(t)$ under the present initial conditions.

Chapter 1. Introduction

The perturbation dynamics, the $O(\varepsilon)$ problem, are then written as

$$\ddot{y}_1 + y_1 = -y_0^3 \quad (1.5)$$

which is obtained by collecting the coefficients of ε^1 in Eq (1.3). Solving this problem for y_1 gives

$$y_1(t) = \frac{1}{32} \cos(3t) - \frac{1}{32} \cos(t) - \frac{3}{8} t \sin(t) \quad (1.6)$$

which results in the approximate analytical solution to the original problem, Eq (1.2), as

$$y(t) \approx \cos(t) + \varepsilon \left(\frac{1}{32} \cos(3t) - \frac{1}{32} \cos(t) - \frac{3}{8} t \sin(t) \right). \quad (1.7)$$

The $t \sin(t)$ term in this expression is known as a “secular” term and renders this approximation non-uniformly valid - meaning it is only considered a good approximation for small values of t . The pursuit of the subsequent methods outlined in this section is to eliminate secular terms such as these.

Lighthill’s technique

Lighthill’s technique belongs to a subclass of perturbation methods known as “strained coordinate methods”. These methods typically function by transforming the independent variable, time t in many cases, into a variable which can be tuned and refined in order to mitigate the appearance of secular terms. Lighthill’s technique, also known as the Poincaré-Lighthill-Kuo (PLK) technique, achieves the “straining” of the independent variable by use of a generalised expansion as follows.

$$t = \tau + \varepsilon t_1(\tau) + \varepsilon^2 t_2(\tau) + \dots \quad (1.8)$$

Using this expansion, τ is introduced as a new (strained) coordinate, and the as-yet-undefined functions $t_n(\tau)$ can be tuned such that they eliminate secular terms throughout the solution.

Chapter 1. Introduction

Lighthill's technique then poses the solution in a similar form to Eq (1.1) but in terms of the new strained coordinate.

$$y(t) = y_0(\tau) + \varepsilon y_1(\tau) + \cdots \quad (1.9)$$

The operation of differentiating with respect to time needs to be reframed in light of the new coordinate. By the chain rule,

$$\frac{d(\cdot)}{dt} = \frac{d\tau}{dt} \frac{d(\cdot)}{d\tau} \quad (1.10)$$

where

$$\frac{dt}{d\tau} = 1 + \varepsilon \frac{dt_1}{d\tau} \Rightarrow \frac{d\tau}{dt} = \frac{1}{(1 + \varepsilon t'_1(\tau))} \quad (1.11)$$

in which the prime notation is used to denote differentiation with respect to τ , $t'_1(\tau) = \frac{dt_1}{d\tau}$. Note that the present example only shows terms up to $O(\varepsilon)$ for the purposes of brevity in the demonstration. For small values of ε , the approximation

$$\frac{d\tau}{dt} \approx 1 - \varepsilon t'_1(\tau) \quad (1.12)$$

can therefore be made. The first and second derivatives of the dependent variable, $y(t)$, with respect to τ are then determined.

$$\begin{aligned} \frac{dy}{dt} &\approx \frac{d\tau}{dt} \frac{dy(t)}{d\tau} \\ &\approx (1 - \varepsilon t'_1(\tau)) \frac{d}{d\tau} (y_0(\tau) + \varepsilon y_1(\tau)) \\ &\approx y'_0(\tau) + \varepsilon (y'_1(\tau) - t'_1(\tau) y_0(\tau)) + O(\varepsilon^2) \end{aligned} \quad (1.13)$$

Chapter 1. Introduction

$$\begin{aligned}
\frac{d^2 y}{dt^2} &\approx \frac{d\tau}{dt} \frac{d}{d\tau} \left(\frac{dy}{dt} \right) \\
&\approx \frac{d\tau}{dt} \frac{d}{d\tau} (y'_0(\tau) + \varepsilon (y'_1(\tau) - t'_1(\tau)y_0(\tau))) \\
&\approx (1 - \varepsilon t'_1(\tau)) \frac{d}{d\tau} (y'_0(\tau) + \varepsilon (y'_1(\tau) - t'_1(\tau)y_0(\tau))) \\
&\approx y''_0(\tau) + \varepsilon (y''_1(\tau) - t''_1(\tau)y_0(\tau) - t'_1(\tau)y'_0(\tau) - t'_1(\tau)y''_0(\tau)) + O(\varepsilon^2)
\end{aligned} \tag{1.14}$$

Substituting these transformed derivatives into Eq (1.2) alongside the assumed solution form of Eq (1.9), the result is as follows.

$$\begin{aligned}
&(y''_0 + \varepsilon (y''_1 - 2t'_1 y''_0 - t''_1 y'_0)) + (y_0 + \varepsilon y_1) + \varepsilon (y_0 + \varepsilon y_1)^3 = 0 \\
&(y''_0 + \varepsilon (y''_1 - 2t'_1 y''_0 - t''_1 y'_0)) + (y_0 + \varepsilon y_1) + \varepsilon (y_0^3 + 3\varepsilon y_0^2 y_1 + 3\varepsilon^2 y_0 y_1^2 + \varepsilon^3 y_1^3) = 0
\end{aligned} \tag{1.15}$$

The coefficients of each order of ε are then collected and separated into the corresponding differential equation. The leading-order problem is

$$y''_0 + y_0 = 0, \tag{1.16}$$

while the $O(\varepsilon)$ problem is

$$y''_1 + y_1 = -y_0^3 + 2t'_1 y''_0 + t''_1 y'_0. \tag{1.17}$$

The solution to the leading-order problem is simply

$$y_0(\tau) = \cos(\tau) \tag{1.18}$$

given the same initial conditions as before. Substituting this result for y_0 into the $O(\varepsilon)$ problem and expanding the cubed cosine function by $\cos^3(\tau) = \frac{3}{4} \cos(\tau) + \frac{1}{4} \cos(3\tau)$, the $O(\varepsilon)$ problem can be elaborated as follows.

$$y''_1 + y_1 = -\left(\frac{3}{4} + 2t'_1\right) \cos(\tau) - \frac{1}{4} \cos(3\tau) - t''_1 \sin(\tau) \tag{1.19}$$

Chapter 1. Introduction

The benefit of the undefined function $t_1(\tau)$ can now be applied. The nonhomogeneous $\cos(\tau)$ and $\sin(\tau)$ terms will lead to resonance since they operate at the natural frequency of the homogeneous problem in Eq (1.19). Hence, the benefit afforded by Lighthill's technique is the ability to select $t_1(\tau)$ such that these terms are removed from the problem.

$$\begin{aligned}\frac{3}{4} + 2t_1' &= 0 \\ t_1' &= -\frac{3}{8} \Rightarrow t_1'' = 0\end{aligned}\tag{1.20}$$

Hence the $O(\varepsilon)$ problem can be written as

$$y_1'' + y_1 = -\frac{1}{4} \cos(3\tau)\tag{1.21}$$

to which there exists an exact, non-secular solution. As a result, the approximate analytical solution in the form of Eq (1.9) will remain uniformly valid.

One of the most prevalent areas of application of this method is in the analysis of nonlinear waves. Albalawi, El-Tantawy, and Alkhateeb used a modified version of Lighthill's technique to obtain an approximate analytical solution to the damped Korteweg-de Vries (KdV) PDE system [1]. In their work, the authors used this well-established damped KdV model to model the propagation of solitons (solitary waves) and their eventual collisions and energy dissipation. Likewise, a similar approach was adopted to examine the collision between two coupled travelling solitons by Bhatti and Lu [2], and also Narahara [3]. Narahara had also previously used this approach to outline an approach to predicting energy pulses in electrical power transmission, noting that the qualitative approximation was promising but that the signal magnitude was overestimated by Lighthill's technique [4].

The Lindstedt-Poincaré method

Similar to Lighthill's technique, the Lindstedt-Poincaré technique is another strained coordinate method. It differs from the preceding method in the precise form of coor-

Chapter 1. Introduction

dinate transformation which is introduced. Where Lighthill's technique is more highly generalised, using the functions $t_n(\tau)$ to strain the coordinate system as required to avoid secular solutions, the Lindstedt-Poincaré technique is specialised to find periodic solutions to nonlinear oscillations. In particular, it achieves this via the transformation

$$\tau = \omega t \quad (1.22)$$

where the coefficients of ε in

$$\omega = \sum_{j=0}^n \omega_j \varepsilon^j \quad (1.23)$$

are constants. To revisit the same hardening Duffing oscillator system as before, Eq (1.2), with the same initial conditions, the derivatives are firstly transformed into the new strained coordinate.

$$\frac{d(\cdot)}{dt} = \frac{d\tau}{dt} \frac{d(\cdot)}{d\tau} = \omega \frac{d(\cdot)}{d\tau} \quad (1.24)$$

$$\begin{aligned} \frac{d^2(\cdot)}{dt^2} &= \frac{d}{dt} \left(\frac{d(\cdot)}{dt} \right) \\ &= \omega \frac{d}{d\tau} \left(\omega \frac{d(\cdot)}{d\tau} \right) \\ &= \omega^2 \frac{d^2(\cdot)}{d\tau^2} \end{aligned} \quad (1.25)$$

Thus, the original ODE can be approximated by the expression

$$(\omega_0^2 y_0'' + \varepsilon (\omega_0^2 y_1'' + 2\omega_0 \omega_1 y_0'')) + (y_0 + \varepsilon y_1) + \varepsilon y_0^3 = 0 \quad (1.26)$$

where $O(\varepsilon^2)$ and above have been omitted. Collecting the $O(1)$ terms, the leading-order equation is then written as

$$\omega_0^2 y_0'' + y_0 = 0 \quad (1.27)$$

and has the particular solution

$$y_0 = \cos \left(\frac{1}{\omega_0} \tau \right). \quad (1.28)$$

Chapter 1. Introduction

Knowing that the frequency of the unperturbed problem is in fact 1 rad/s, ω_0 can be set equal to 1 rad/s. The $O(\varepsilon)$ problem is then written as

$$\begin{aligned}\omega_0^2 y_1'' + y_1 &= -y_0^3 - 2\omega_0 \omega_1 y_0'' \\ &= -y_0^3 - 2\omega_1 y_0''\end{aligned}\tag{1.29}$$

where y_0 is already known as the leading-order solution. Thus the nonhomogeneous terms in the $O(\varepsilon)$ problem can be expanded to give the following ODE.

$$\omega_0^2 y_1'' + y_1 = \left(2\omega_1 - \frac{3}{4}\right) \cos(\tau) - \frac{1}{4} \cos(3\tau)\tag{1.30}$$

As in Lighthill's technique, the goal of the final step of this strained coordinate method is to determine the value of ω_1 needed to eliminate the $\cos(\tau)$ term exciting this system at resonance.

$$2\omega_1 - \frac{3}{4} = 0 \quad \Rightarrow \quad \omega_1 = \frac{3}{8}\tag{1.31}$$

As a result, the approximate analytical solution to Eq (1.2) via the Lindstedt-Poincaré technique is

$$y(t) = \cos(\tau) + \varepsilon \frac{1}{32} \cos(3\tau)\tag{1.32}$$

$$= \cos\left(\left(1 + \varepsilon \frac{3}{8}\right)t\right) + \varepsilon \frac{1}{32} \cos\left(3\left(1 + \varepsilon \frac{3}{8}\right)t\right).\tag{1.33}$$

Regarding its use, the Lindstedt-Poincaré method has been used in structural and materials engineering research to analyse complex vibration dynamics. Avey et al. use this technique to obtain solutions describing the variation of frequency with vibration amplitude in shear-deformable multilayer composite structures [5]. The method has also been used to consider the nonlinear vibration dynamics that can occur in discontinuous structures such as a beam containing a crack. Vestroni, Luongo, and Paolone studied this particular problem, obtaining second-order corrections to the unperturbed 2-DOF problem [6]. However, they were unable to obtain a uniformly valid approxi-

mation of the solution, despite their result being suitable within their domain of interest.

Beyond direct applications to problems in engineering, the Lindstedt-Poincaré technique has been used as the foundation for more advanced analytical tools. It has been combined with methods like the Linear Delta Expansion and Padé approximants to solve more strongly nonlinear systems in vibration dynamics. In particular, the Linear Delta Expansion has been combined with the Lindstedt-Poincaré technique to solve the hardening Duffing oscillator problem by Amore and Aranda [7]. Compared to the simple approach taken in this present section, Amore and Aranda's approach results in more consistently reliable approximations in spite of large amplitudes being imposed on the system via its initial conditions. Considering a similarly widely-studied oscillator problem, Li and Tang combined the Lindstedt-Poincaré technique with the rational approximations produced by the Padé approximant to produce a more accurate result when compared to the basic method such as is described in the present work [8]. Collectively, these applications highlight this method's relevance and versatility in tackling complex engineering problems involving nonlinear dynamic systems.

The method of multiple scales

The method of multiple scales is another powerful means of obtaining approximate analytical solutions to nonlinear dynamic systems, and a detailed explanation of its deployment is given in [9].

As its name suggests, the method of multiple scales is particularly suited to approximating solutions which have differing behaviours over various timescales. Kiani-Oshtorjani et al. demonstrate its utility in modelling pressure variations in a hydraulic system, especially in the presence of small fluid volumes [10]. Pipitone, Firrone, and Zucca exploit the method in capturing the response of dynamically coupled, meshed gears where the coupling parameters are treated as time-varying quantities [11]. The method of multiple scales also sees use in comparatively simple application due to the power in accounting for differing timescales of variations. Ilyas, Alfosail, and Younis

consider this technique’s application to a simple, forced, single-DOF oscillator system, which models a micro-electromechanical oscillator driven by an electrostatic force [12].

The examples so far have considered the problems involving multiple time scales. However, in the spatial domain, the method of multiple scales is useful in the analysis of composite materials and heterogeneous structures. The multiple scales in these contexts are not temporal, but physical, spatial scales. For instance, this could consider the microscopic scale of a material’s component fibres versus the larger scale of the overall structure. This is precisely the nature of the system studied by Dong et al., wherein they use this method to predict the behaviour of a composite structure undergoing thermo-mechanical interactions [13].

1.1.2 Non-perturbation approximation strategies

Having discussed perturbation methods, which rely on the use of a small parameter to distinguish the generating dynamics and small perturbations from this, attention is now given to non-perturbation approximation strategies. These include methods such as the Adomian decomposition method, variational iteration method, homotopy perturbation method, and harmonic balance method. Similarly to the perturbation methods explored previously, the following approaches begin by seeking to capture the “most significant” aspects of the target solution before making subsequent corrections. The precise construction of these corrections and the means of obtaining the “most significant” aspects of these solutions varies across these methods without the consistency present across the perturbation methods.

Adomian decomposition method

The Adomian Decomposition Method (ADM) is a powerful technique for obtaining approximate analytical solutions to nonlinear ODEs. It has a similarity to the perturbation methods discussed previously in that it uses series expansions to approximate the solution. However, even this is a tenuous link as the series used in ADM are not

Chapter 1. Introduction

power series or asymptotic expansions as they are in perturbation methods. Nonetheless, it does utilise a recursive approach to obtain a more accurate approximation after each step.

A descriptive narration of ADM is undertaken via an example. Suppose the nonlinear ODE to be solved is

$$y'(x) - y^2(x) = 0 \quad (1.34)$$

with the initial condition $y(0) = 1$. This problem is chosen due to its relative simplicity for demonstration purposes, the presence of its nonlinearity, and the fact that the exact solution to this problem is already known to be $y(x) = \frac{1}{1-x}$. Knowing this exact solution will give a useful comparison for the result of ADM in this case.

ADM first requires that the highest-order derivative be isolated, with the remaining terms being rearranged to the right-hand side of the expression. For Eq (1.34), this results in

$$y'(x) = y^2(x), \quad (1.35)$$

but any nonlinear first-order ODE could be generalised to $y'(x) = g(x) - N(y)$ where $N(y)$ is the nonlinear part of the ODE. The independent variable is then substituted for another “dummy” variable, and the original independent variable is implemented as the upper limit to which the problem is integrated.

$$\begin{aligned} \int_0^x y'(z)dz &= \int_0^x y^2(z)dz \\ y(x) &= y(0) + \int_0^x y^2(z)dz \\ y(x) &= 1 + \int_0^x y^2(z)dz \end{aligned} \quad (1.36)$$

Notice, in this example, the x is substituted with the dummy variable z for the integration. Next, the solution, $y(x)$, and the nonlinear term, $N(y(x)) = y^2(x)$, are

Chapter 1. Introduction

decomposed into the following series

$$y(x) = y_0 + y_1 + y_2 + \cdots = \sum_{j=0}^{\infty} y_j(x) \quad (1.37)$$

$$N(y(x)) = A_0 + A_1 + A_2 + \cdots = \sum_{j=0}^{\infty} A_j \quad (1.38)$$

where the term A_n is known as a particular type of polynomial known as an Adomian polynomial and is defined by the following expression.

$$A_n = \frac{1}{n!} \left[\frac{d^n}{d\lambda^n} N \left(\sum_{j=0}^{\infty} \lambda^j y_j \right) \right]_{\lambda=0} \quad (1.39)$$

The first three Adomian polynomials for $N(y) = y^2$ are hence derived as follows. Note that, for the squared series expansion required, the coefficients of the resulting power series in λ can be determined by the Cauchy product for two infinite series.

$$\begin{aligned} A_0 &= \frac{1}{0!} \left[(y_0 + \lambda y_1 + \lambda^2 y_2 + \cdots)^2 \right]_{\lambda=0} \\ &= \left[(y_0^2 + \lambda (2y_0 y_1) + \lambda^2 (2y_0 y_2 + y_1^2) \cdots)^2 \right]_{\lambda=0} \\ &= y_0^2 \end{aligned} \quad (1.40)$$

$$\begin{aligned} A_1 &= \frac{1}{1!} \left[\frac{d}{d\lambda} (y_0 + \lambda y_1 + \lambda^2 y_2 + \cdots)^2 \right]_{\lambda=0} \\ &= \left[\frac{d}{d\lambda} (y_0^2 + \lambda (2y_0 y_1) + \lambda^2 (2y_0 y_2 + y_1^2) \cdots)^2 \right]_{\lambda=0} \\ &= 2y_0 y_1 \end{aligned} \quad (1.41)$$

$$\begin{aligned} A_2 &= \frac{1}{2!} \left[\frac{d^2}{d\lambda^2} (y_0 + \lambda y_1 + \lambda^2 y_2 + \cdots)^2 \right]_{\lambda=0} \\ &= \left[\frac{d^2}{d\lambda^2} (y_0^2 + \lambda (2y_0 y_1) + \lambda^2 (2y_0 y_2 + y_1^2) \cdots)^2 \right]_{\lambda=0} \\ &= 2y_0 y_2 + y_1^2 \end{aligned} \quad (1.42)$$

Chapter 1. Introduction

With the decomposed forms of $y(x)$ and $N(y)$, the integral equation, Eq (1.36), becomes

$$\begin{aligned}\sum_{j=0}^{\infty} y_j(x) &= 1 + \int_0^x \left(\sum_{j=0}^{\infty} A_j \right) dz \\ (y_0 + y_1 + y_2 + \cdots) &= 1 + \int_0^x (A_0 + A_1 + A_2 + \cdots) dz.\end{aligned}\tag{1.43}$$

From this, it can be derived that

$$y_0(x) = 1\tag{1.44}$$

and

$$y_{n+1}(x) = \int_0^x A_n(z) dz.\tag{1.45}$$

Hence

$$\begin{aligned}y_1 &= \int_0^x A_0(z) dz = \int_0^x y_0^2(z) dz \\ &= \int_0^x (1) dz \\ &= x\end{aligned}\tag{1.46}$$

$$\begin{aligned}y_2 &= \int_0^x A_1(z) dz = \int_0^x (2y_0(z)y_1(z)) dz \\ &= 2 \int_0^x (z) dz \\ &= x^2\end{aligned}\tag{1.47}$$

$$\begin{aligned}y_3 &= \int_0^x A_2(z) dz = \int_0^x (2y_0(z)y_2(z) + y_1^2(z)) dz \\ &= 2 \int_0^x (2z^2 + z^2) dz \\ &= x^3\end{aligned}\tag{1.48}$$

Extrapolating from this pattern in $y_{n+1}(x) = \int_0^x A_n(z) dz$, the approximate analytical

Chapter 1. Introduction

solution produced by Adomian Decomposition Method is the geometric series

$$y(x) = 1 + x + x^2 + x^3 + \dots = \sum_{j=0}^{\infty} x^j. \quad (1.49)$$

Notably, when $|x| < 1$ in Eq (1.49), the series converges to the exact solution.

$$y(x) = \frac{1}{1-x} \quad (1.50)$$

Owing to its ability to handle stronger nonlinearities than the previously described perturbation methods, ADM has been widely applied in various fields. For example, Arshad et al. consider a modified version of ADM, known as “fractional improved [ADM]” [14], in addressing wave phenomena in fluid mechanics and nonlinear optics. Using this modified technique, the authors proceed to use it in solving a fractional-order coupled KdV system. Providing another example of this technique in addressing fractional-order differential equations, Ziada uses ADM to address a broad class of fractional-order systems [15].

Turkyilmazoglu explores the introduction of an auxiliary parameter - in some of the early solution series expansion terms, e.g. y_0, y_1, y_2 , etc., in Eq (1.37) - which has the effect of improving the observed convergence of the resulting approximations [16]. While no concrete specification is given as to which terms this auxiliary parameter must be limited, the author does present an assessment of the optimal value of this parameter in their particular application context. This exemplifies a pattern of research which can be observed for several of the techniques discussed in this chapter: while all are used extensively in their appropriate use-cases, a significant body of research is devoted to developing slight modifications which improve their performance in one or more metrics such as computational burden, convergence, or precision.

Variational iteration method

Taking a different approach to ADM is the variational iteration method (VIM). In essence, this approach begins by taking an initial guess of the solution to an ODE, applying a correction to this guess, and then iteratively repeating this process. The correction made to each iterative guess is known as the “correction functional” and can be generalised to the following form.

$$y_{n+1}(x) = y_n(x) + \int_0^x \lambda(s) (L(y_n(s)) + N(\tilde{y}_n(s)) - g(s)) ds \quad (1.51)$$

In this expression, $y_n(x)$ is the current guess of the analytical solution; $y_{n+1}(x)$ is the new guess in light of the correction made by the integral; and, $\lambda(s)$ is the Lagrange multiplier. The remaining integrand is comprised of terms from the ODE to be solved: $L(y_n)$ is the linear part of the ODE; $N(\tilde{y}_n)$ is the nonlinear part of the ODE, where the tilde notation is used to indicate this term is treated as a known constant during the correction; and $g(x)$ is the nonhomogeneous term in the ODE.

To demonstrate the application of this method, consider the following example. Consider the first-order ODE, posed with the initial condition $y(0) = 1$, Eq (1.34). Variational calculus can be used to show that the Lagrange multiplier, $\lambda(s)$, for a first-order ODE is always -1. An initial guess for the solution can also be made as $y_0(x) = 1$, as this guess is the simplest function that satisfies the initial condition for the system.

Proceeding now to determine the first correction to the initial guess, $y_1(x)$, Eq (1.51) is used with $\lambda(s) = -1$ and $y_0(x) = 1$.

$$\begin{aligned} y_1(x) &= y_0(x) - \int_0^x (y_0'(s) - y_0^2(s)) ds \\ &= 1 - \int_0^x (0 - 1) ds \\ &= 1 + x \end{aligned} \quad (1.52)$$

The second correction, $y_2(x)$, can then be determined by repeating this process.

$$\begin{aligned}
 y_2(x) &= y_1(x) - \int_0^x (y_1'(s) - y_1^2(s)) ds \\
 &= 1 + x - \int_0^x (1 - (1 + s)^2) ds \\
 &= 1 + x + x^2 + \frac{1}{3}x^3
 \end{aligned} \tag{1.53}$$

In Abdou and Soliman’s work, the variational iteration method is extended to solve three partial differential equation systems designed to model wave-based phenomena (namely the Schrodinger-KdV, generalised KdV, and shallow water equations) [17]. Notably, the authors compare their approximate solutions with those obtained by ADM and show the benefit of their approach in these contexts. As far as traditional approaches of VIM are concerned, He and Wu provide a useful review of the research space and introduction to the theoretical basis of the method, covering applications of the method to various thermodynamics and fluid mechanics scenarios as well as to other more abstract mathematical phenomena such as nonlinear oscillations and relaxation characteristics [18].

As with previously discussed approaches, much of the research in this space considers improvements to the “basic” method. Turkyilmazoglu’s work seeks to enhance traditional VIM by including a parameter which allows convergence to be obtained more efficiently [19]. This is mirrored by the same author’s work in accelerating the convergence of the Adomian Decomposition Method in Section 1.1.2. The author also goes on to present a method for obtaining an optimal value of this convergence parameter depending on the contextual application. Shirazian is another author who has sought to “accelerate” VIM [20]. In this case, the author achieved this by modifying a more subtle part of the algorithm’s computation: deconstructing the integration-by-parts step offline to allow for more efficient computation online. While there may be some debate over whether this truly removes “redundant calculations”, Shirazian’s work cer-

tainly demonstrates that their algorithm is computed more efficiently. Bhargava, Jain, and Suthar consider the study of fractional-order partial differential equations in the form of heat dissipation equations [21]. In particular, the authors improve on the traditional VIM by incorporating Laplace transforms to generalise the calculation of the necessary Lagrange multipliers. Lastly, in a novel and highly relevant application to global themes, Ahmad et al. use a modified version of VIM to study the dispersion of oil spills in water [22]. Similar to Bhargava, Jain, and Suthar’s work, Ahmad et al. also consider a modified approach to computing appropriate Lagrange multipliers.

Homotopy perturbation method

Counter-intuitively for its name, the Homotopy Perturbation Method (HPM) should not be classed as a perturbation method in the same sense as Lighthill’s technique, the Lindstedt-Poincaré technique, or the method of multiple scales. The “perturbation” in HPM is provided through a slightly different mechanism to these preceding methods although it still fits the general approach of considering an unperturbed part of the target system which can be solved exactly, alongside a perturbation which complicates this solution process to the point of intractability. The term “homotopy” refers to the mathematical mapping of one function to another: HPM achieves this capacity by introducing a parameter, $\{0 \leq p \leq 1\}$, which can increase or decrease the presence of the “perturbed” part of the system. Put simply: when $p = 0$, the homotopy returns the simplified, unperturbed problem to which there is a tractable analytical solution; when $p = 1$, the homotopy returns the target system to be solved; and, for any value of p between these limits, the problem contains some, but not all of the complexity of the target problem. Formally, this homotopy is generalised in the form

$$H(v; p) = (1 - p)L(v) + pF(v) = 0 \quad (1.54)$$

where $L(v)$ is the simplified, linear, tractable problem in the generalised coordinate v , and $F(v)$ is the original, target problem.

Chapter 1. Introduction

Another important step in HPM is the expansion of the generalised coordinate as a power series in the parameter p , where v_0 should be the component which is closest to the exact solution and subsequent v_n terms function as corrections to this initial guess.

$$v = v_0 + v_1 p + v_2 p^2 + \dots \quad (1.55)$$

Note that when $p = 1$

$$v = v_0 + v_1 + v_2 + \dots \quad (1.56)$$

Similar to the previous approaches shown, including the perturbation methods outlined, the next step is to substitute this power series into the homotopy Eq (1.54) alongside the appropriate forms of $L(v)$ and $F(v)$. At this point, an example should be considered to demonstrate this method.

Consider the Riccati equation

$$y'(x) - y^2(x) = 0 = F(y) \quad (1.57)$$

with the initial condition $y(0) = 1$. The linear part can be written as $L(y) = y'(x)$. Thus, the homotopy can be written as follows, where y_0 is used as an initial guess of the solution (appropriately considering the initial condition applied to the system).

$$\begin{aligned} H(y; p) &= (1 - p) [L(y) - L(y_0)] + pF(y) = 0 \\ (1 - p) (y' - y'_0) + p (y' - y^2) &= 0 \\ (y' - y'_0) - p (y^2 - y'_0) &= 0 \end{aligned} \quad (1.58)$$

Substituting the power series approximation of $y(x)$ into this homotopy, the result is

$$(y'_1 p + y'_2 p^2 + \dots) - p (y_0^2 + 2y_0 y_1 p + (2y_0 y_2 + y_1^2) p^2 + \dots) = 0. \quad (1.59)$$

Then, collecting equations in successive powers of p results in the following problems

Chapter 1. Introduction

to be solved. Noting that no terms remain at $O(p^0)$, the $O(p^1)$ problem is

$$y_1' = y_0^2, \quad (1.60)$$

and the $O(p^2)$ is

$$y_2' = 2y_0y_1. \quad (1.61)$$

Solving these problems, recalling that $y_0 = 1$, leads to

$$\begin{aligned} y_1' &= 1 \quad \Rightarrow \quad y_1 = t + C_1 \\ y_1(0) &= 0 + C_1 = 0 \\ y_1 &= t, \end{aligned} \quad (1.62)$$

and

$$\begin{aligned} y_2' &= 2t. \quad \Rightarrow \quad y_2 = t^2 + C_2 \\ y_2(0) &= 0 + C_2 = 0 \\ y_2 &= t^2. \end{aligned} \quad (1.63)$$

Following this pattern, the approximation of $y(x)$ yielded by HPM is

$$y(x) \approx 1 + t + t^2 + \cdots = \sum_{k=0}^{\infty} t^k, \quad (1.64)$$

a geometric series. Hence, just as was observed in the Adomian Decomposition Method, this converges to the exact solution of the Riccati equation $y = \frac{1}{1-t}$ for small values of t .

Still the subject of active research across STEM fields, Jalili et al. endeavoured to ease its application in others' work by generalising the algorithm in an efficient fashion in a novel Python-based algorithm [23]. While Jalili et al.'s work doesn't explore especially novel applications beyond the development of this resource, other authors have recently sought to improve HPM in other ways with novel applications.

Chapter 1. Introduction

Guan et al. examine the development of their “improved homotopy perturbation method” in the context of inverse dynamics problems with a view to improving its utility in engineering structural applications [24]. Sharif, Molla, and Alam also modify HPM by introducing an additional frequency expansion series (alongside the standard generalised coordinate expansion) in the broadly applicable space of nonlinear vibration analysis [25]. Lastly, Alkhazaleh introduce their own “modified” HPM, which incorporates Laplace transforms and Padé approximants to efficiently improve the method’s convergence and reduce its error [26].

It should be noted that the homotopy perturbation method is an actively evolving technique, as it is still relatively recently developed (as evidenced in the preceding paragraphs). While it shows some degree of promise, this continuing research will eventually lead to greater consensus on the precise bounds of its strengths, weaknesses, and appropriate applications.

Harmonic balance method

The core idea of the harmonic balance method (HBM) is to construct the solution from sine and cosine terms at successively increasing frequencies - utilising a Fourier Series approach. That is, when y is the dependent variable (and a function of time, t) of the nonlinear system of differential equations under study,

$$\begin{aligned} y(t) &= A_1 \cos(\omega t) + B_1 \sin(\omega t) + A_2 \cos(2\omega t) + B_2 \sin(2\omega t) + \dots \\ &\approx \sum_{k=1}^{\infty} (A_k \cos(k\omega t) + B_k \sin(k\omega t)). \end{aligned} \tag{1.65}$$

As will be demonstrated via the familiar example of the hardening Duffing oscillator, the use of this approach converts the problem from one in the language of calculus to one in the language of algebra.

Consider the simple hardening Duffing oscillator which was considered previously in Eq (1.2). The problem is modified slightly here for the purposes of demonstrating

Chapter 1. Introduction

the method more fully.

$$\ddot{x} + \delta\dot{x} + \alpha x + \beta x^3 = F \cos(\omega t) \quad (1.66)$$

HBM then prescribes that a suitable order of Fourier Series is substituted into the target problem, the sine and cosine terms are then grouped (by frequency), their coefficients are determined by grouping “like-terms” and solving the resulting system of algebraic equations.

Acknowledging that the cubic nonlinearity will cause the number of terms in the resulting equation to “balloon” rapidly, this demonstration will consider only a first-order Fourier Series expansion. The necessary substitutions are

$$x \approx A_1 \cos(\omega t) + B_1 \sin(\omega t), \quad (1.67)$$

$$\dot{x} \approx -A_1 \omega \sin(\omega t) + B_1 \cos(\omega t), \quad (1.68)$$

$$\ddot{x} \approx -A_1 \omega^2 \cos(\omega t) - B_1 \omega^2 \sin(\omega t), \quad (1.69)$$

$$(1.70)$$

and

$$\begin{aligned} x^3 &\approx A_1^3 \cos^3(\omega t) + B_1^3 \sin^3(\omega t) \\ &\approx A_1^3 \cos^3(\omega t) + B_1^3 \sin^3(\omega t) \\ &\quad + 3A_1^2 B_1 \cos^2(\omega t) \sin(\omega t) + 3A_1 B_1^2 \cos(\omega t) \sin^2(\omega t). \end{aligned} \quad (1.71)$$

Noting that the $\cos^2(\omega t) \sin(\omega t)$ and $\cos(\omega t) \sin^2(\omega t)$ terms will result in only higher frequencies than ω , these are neglected from the resulting substitutions. Neglecting all higher-frequency terms from x^3 results in

$$x^3 \approx \frac{3}{4} A_1^3 \cos(\omega t) + \frac{3}{4} B_1^3 \sin(\omega t). \quad (1.72)$$

Chapter 1. Introduction

Substituting these results for x , $\ddot{x}$, and x^3 into Eq (1.66) leads to

$$\begin{aligned}
& [-A_1\omega^2 \cos(\omega t) - B_1\omega^2 \sin(\omega t)] \\
& + \delta [-A_1\omega \sin(\omega t) + B_1 \cos(\omega t)] \\
& + \alpha [A_1 \cos(\omega t) + B_1 \sin(\omega t)] \\
& + \beta \left[\frac{3}{4}A_1^3 \cos(\omega t) + \frac{3}{4}B_1^3 \sin(\omega t) \right] = F \cos(\omega t).
\end{aligned} \tag{1.73}$$

Collecting the $\cos(\omega t)$ terms leads to

$$\left(\alpha - \omega^2 + \frac{3}{4}\beta A_1 \right) A_1^2 + \delta\omega B_1 = F, \tag{1.74}$$

and collecting the $\sin(\omega t)$ terms leads to

$$\left(\alpha - \omega^2 + \frac{3}{4}\beta B_1 \right) B_1^2 - \delta\omega A_1 = 0. \tag{1.75}$$

Note that the solution of this algebraic system requires numerical methods.

Despite its apparently simple construction, the harmonic balance method has been used in the analysis of phenomena such as bifurcations in nonlinear systems. Detroux et al. study the application of HBM to understanding the bifurcation characteristics of a vibration isolation system in a small-scale satellite [27]. Their work focuses primarily on detecting Neimark-Sacker bifurcations by considering a particular variant of the Floquet theory applied to the result of HBM in their problem. Other authors, rather than appending a subsequent advancement to the output of traditional HBM, have introduced advancements to the HBM algorithm itself.

Jaumouillé, Sinou, and Petitjean, for example, present a variant of HBM which permits an adaptive approach to the number of harmonics being kept in the approximation [28]. Their work considers joints connected with bolts or rivets and the contribution of these joints to the structure's damping characteristics. Sun et al. also introduce an advancement to traditional HBM by modifying the approach into a so-called “semi-

analytical” method referred to as an “incremental harmonic balance method” [29]. Specifically, the “semi-analytical” component refers to the integration of traditional HBM with finite element method (FEM) discretisation. Lastly, Chen et al. introduce another of semi-analytical approach, comparing it to the incremental HBM [30]. Their “general and efficient harmonic balance method” is also compared to other previously developed improvements to HBM, such as the “harmonic balance-alternative frequency/time domain method”, to great success in terms of computational efficiency.

1.1.3 Linearisation techniques

While the methods discussed earlier in this chapter can be used to achieve linearised models, taking the generating problems produced by perturbation methods as an example, the remaining methods in this chapter are more commonly used in control-focused literature. These methods, Jacobian linearisation and piecewise linear approximation, are less commonly used to obtain approximate analytical solutions in the same way as the preceding methods; but they find common usage in linearised control systems, particularly when combined with methods such as gain scheduling. While these final two approaches reviewed in this chapter do not share the same characteristic of making corrections to an initial guess, their ubiquity and relevance to the control literature discussed in Chapter 4 warrants their inclusion here.

Jacobian linearisation

Jacobian linearisation fundamentally relies on the identification of equilibrium points and subsequent use in the vicinity of these points. Linearising the system about such points allows the user to assess the nature of stability at these points. As with previous methods in this chapter, the method is first explained through an example. Consider the Lotka-Volterra ODE system that describes the population dynamics of a rabbit-and-fox ecosystem. The rate of change of the rabbit population might be given by

$$\frac{dx}{dt} = 2x - xy = f(x, y) \quad (1.76)$$

Chapter 1. Introduction

while the rate of change of the fox population is given by

$$\frac{dy}{dt} = xy - 3y = g(x, y). \quad (1.77)$$

The Jacobian matrix of this system, characterising the rate of change of each parameter with respect to each variable, can be generalised to

$$J = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} \end{bmatrix} \quad (1.78)$$

where the number of rows of the Jacobian matrix, J , is determined by the number of 1st-order ODEs in the system (with higher-order problems first being decomposed into appropriate 1st-order ones); and, the number of columns is determined by the number of input variables to these functions. Computing the Jacobian for the Lotka-Volterra system described in Eqs (1.76) and (1.77) yields the following result.

$$J(x, y) = \begin{bmatrix} 2 - y & -x \\ y & x - 3 \end{bmatrix} \quad (1.79)$$

The linearised system is then written as

$$\frac{d\mathbf{u}}{dt} = J(x_0, y_0)\mathbf{u} \quad (1.80)$$

where $\mathbf{u}$ is the vector representing small deviations from the equilibrium point (x_0, y_0) , $\mathbf{u} = [x - x_0, y - y_0]^T$. $\frac{d\mathbf{u}}{dt}$ naturally therefore represents how quickly the small deviations are changing (and whether they are returning to the equilibrium point or deviating further from it).

For example, assessing a non-trivial equilibrium point of the system described by

Chapter 1. Introduction

Eqs (1.76) and (1.77) to be $(3, 2)$, the Jacobian is evaluated as

$$J(3, 2) = \begin{bmatrix} 0 & -3 \\ 2 & 0 \end{bmatrix}. \quad (1.81)$$

The resulting linear system is therefore

$$\begin{aligned} \frac{d\mathbf{u}}{dt} &= J(3, 2)\mathbf{u} \\ &= \begin{bmatrix} 0 & -3 \\ 2 & 0 \end{bmatrix} \mathbf{u}, \end{aligned} \quad (1.82)$$

hence

$$\frac{du_1}{dt} = -3u_2 \quad \text{and} \quad \frac{du_2}{dt} = 2u_1. \quad (1.83)$$

Note that, when the Jacobian matrix for this system is computed at an equilibrium point, the final resulting system of ODEs is linear in $\mathbf{u}$.

To assess this resulting system for stability about the chosen equilibrium point, the eigenvalues of the linear system may be considered. Table 1.1 provides an overview of the key observations of eigenvalue analysis of stability types.

Eigenvalue description	Stability implication
Real and negative	Stable node
Real and positive	Unstable node
Real with opposing signs	Saddle point
Complex with negative real parts	Stable spiral
Complex with positive real parts	Unstable spiral
Imaginary	Node is a centre of the system (phase trajectories form a closed loop around it)

Table 1.1: Stability implications of different eigenvalues obtained for linearising the system.

For the present example, the system linearised about $(3, 2)$ has eigenvalues of $\pm i\sqrt{6}$. Hence, referring to Table 1.1, this point is a centre of the system.

Most of the engineering-centred applications for Jacobian linearisation are in the field of control systems design, something which is considered in more detail from Ch 4 onwards. However, owing to its ubiquity in the field, omitting this method from any discussion of linearisation strategies for nonlinear ODE systems would feel incomplete. In particular, many researchers utilise Jacobian linearisation as a key component in developing gain scheduled control strategies. Kwon and Choi demonstrate this in their work where an input-output feedback linearised gain-scheduled controller is compared with another gain-scheduled controller (developed by other authors) [31]. Notably for the present discussion, both approaches rely on Jacobian linearisation to achieve the locally valid linear models. In more recent work, Liu et al. apply Jacobian linearisation in the development of the model for a state-of-the-art multibody wingtip design, which has been shown to improve flight performance of the associated aircraft [32].

There is, of course, recognition of the limitations of this method. Mobariz, Mobariz, and Ashry notably critique the use of this approach due to the resulting control systems' inability to sufficiently adapt to sudden disturbances which cause the model to depart significantly from the operating point around which it is linearised [33]. Like all of the methods discussed in this chapter, however, Jacobian linearisation certainly has appropriate use cases, and it is at the discretion of the practitioner to discern whether or not it is an appropriate tool to solve their present problem. Most recently, Luderer and Thielecke continue the earlier established practices of using Jacobian linearisation in constructing the various models required for their gain scheduling approach to approximating a nonlinear flight model [34]. The authors demonstrated the accuracy and utility of this approximation in predicting loading conditions in its deployed role as an observer in the control system adopted.

Piecewise linear approximation

Piecewise linear approximations are another exceedingly common approach to linearisation in control systems design and have a significant overlap with the concept of gain

Chapter 1. Introduction

scheduling (as described in Chapter 6). As the name suggests, approximating a function by a piecewise linear process involves treating the function as though a discrete division of the solution curve is a linear function. The “piecewise” part of the name implies that the entire solution is discretised and approximated in a linear fashion “piece-by-piece”. An example of this approach is Euler’s method. Although this is a numerical method for solving ODEs rather than an analytical one, it can be used to produce approximate analytical solutions, as will be demonstrated via an example in this section.

Consider the first-order ODE

$$\dot{y} = t - y^2 = f(t, y) \quad (1.84)$$

with the initial condition $y(0) = 1$. Euler’s method,

$$y_{k+1} = y_k + \Delta t \cdot f(t, y), \quad (1.85)$$

projects the gradient of the function, $f(t, y)$, forward over the time-step Δt to predict how the current value of the solution, y_k , will evolve to the next value, y_{k+1} . As a general rule, such approximations become more accurate with smaller step sizes. However, the computational cost of this method increases with this change. Hence, there is a trade-off to be made here - one which is heavily context dependent. Taking, for the purposes of this example, a step size of $\Delta t = 0.2$ seconds, the value of $y(t)$ at the end of this first interval, $t \in [0, 0.2]$, is

$$\begin{aligned} y_1 &= y_0 + \Delta t (t - y_0^2) \\ &= 1 + 0.2 (0 - 1^2) \\ &= 0.8. \end{aligned} \quad (1.86)$$

Projecting the solution forward across the next discrete interval, the value of y_2 is found

to be

$$\begin{aligned}
 y_2 &= y_1 + \Delta t (t - y_1^2) \\
 &= 0.8 + 0.2 (0.2 - 0.8^2) \\
 &= 0.712.
 \end{aligned} \tag{1.87}$$

In each of these intervals, a linear function approximating the solution can be constructed. In the first interval, $t \in [0, 0.2]$, the gradient of the line is found to be $m_1 = \frac{0.8-1}{0.2} = -1$, and the y -intercept of the line is found to be $c_1 = 1$. Similarly, the gradient of the line approximating the solution across the second interval, $t \in [0.2, 0.4]$, is $m_2 = \frac{0.712-0.8}{0.2} = -0.44$, and its y -intercept is $c_2 = 0.888$. Thus, a piecewise linear approximation of the solution to Eq (1.84) can be written as follows.

$$y(t) \approx \begin{cases} -t + 1 & \text{in } 0 \leq t < 0.2, \\ -0.44t + 0.888 & \text{in } 0.2 \leq t \leq 0.4 \end{cases} \tag{1.88}$$

Storace and De Feo, similar to cited works earlier in this chapter, provide a useful contribution to research into this method of approximation by developing a MATLAB toolbox which eases the application of piecewise-linear approximations in a variety of contexts, most novelly to the analysis of bifurcations in nonlinear systems [35]. The authors also provide an early confirmation of the intuitive idea that the quantitative accuracy of the approximation improves with the number of subdivisions in the independent variable's domain. Notably, however, the authors also demonstrate that an odd number of subdivisions yields better results in the presence of particular symmetry conditions. In an application that differs from many others in the available literature, Camponogara and Nazari consider the application of piecewise-linear approximations to functions for “black-box” functions for which only a limited number of data points are known [36]. The authors study this application in the context of data describing the behaviour of gas-lifted oil wells and also consider the implications of uncertain data

for their derived approximations.

More recently, Dong et al. present an algorithm which generalises the application of piecewise-linear approximations to all single-input, single-output functions [37]. The authors prove the robustness of their method, without application to niche engineering applications, and instead focus on notable and ubiquitous transcendental functions. Lastly Birkelbach, Huber, and Hofmann demonstrate the use of piecewise-linear approximations in modelling the behaviour of a generalised energy storage system [38]. Applying the technique within the context of mixed-integer linear programming, a particular challenge of this work is to ensure the resulting approximations are convex and hence this forms the main contribution of this recent work.

1.1.4 Comparison of approximate analytical solution approaches

Throughout this chapter, it should be apparent that no single method is universally superior to any other. The “best” choice is dictated by the specific characteristics of the problem under consideration. Where the preceding sections have outlined the general bases of the methods and have indicated the breadth of their utility and associated research, this section synthesises a comparison of these methods which is summarised in Table 1.2.

Jacobian linearisation, for good reason, is the bedrock of classical control theory and is used extensively in contexts where the system operates in the vicinity of equilibrium points (regardless of the specific type of equilibria). Hence, this is often a common choice of technique for problems involving local stability analysis or problems where the system is predicted to stay in this vicinity. Tyner and Lewis share an insightful perspective on Jacobian linearisation, acknowledging its importance in the described sense and its ubiquity in control systems design [39]. However, the same authors also acknowledge the limitations of this technique in the fact that it can’t capture global behaviour.

Chapter 1. Introduction

Piecewise-linear approximations are perhaps the simplest-to-compute of the methods discussed earlier in this chapter. Additionally, given their design of discontinuous switching between linear approximations, they are ideally suited to model systems containing discontinuities [35] [36]. Given the simplicity of their construction, they are also ideally suited to applications in optimisation and hardware-embedded applications [38] [37]. Nonetheless, piecewise-linear approximations also have their trade-offs to be considered. Most notably, the computational overhead associated with these techniques needs to be calibrated carefully to the user’s needs and computational “budget” [36]. Additionally, and presenting the opposite problem to Jacobian linearisation, there are scenarios in which piecewise-linear approximations present an undesirable sacrifice in local accuracy (failing to capture subtle nonlinear dynamic characteristics) in favour of retaining more globally accurate approximations overall. This last note, of course, is part of the trade-off to be considered in determining an appropriate discretisation for the domain.

Considering the three discussed perturbation methods together initially, these are often best suited to weakly nonlinear problems (permitting the solution to be formed from an asymptotic power series in terms of the small perturbation parameter, ε). Their common premise is the removal of secular terms. Thus, they can each achieve this deliberately, while the non-perturbation techniques discussed in this chapter either do not possess this ability or may only achieve this end indirectly. Particularly comparing their performance to Jacobian linearisation, these techniques perform significantly different functions and answer substantially different questions in their usual application. Where Jacobian linearisation is constructed to support stability analysis in the vicinity of equilibria, perturbation methods are designed to produce uniformly valid approximations of the system under consideration. These methods’ key limitation, however, is their reliance on the nonlinearity of the system being “weak” (as is implicit in the construction of asymptotic series).

The Adomian Decomposition Method does not have the restriction of only being

applicable to problems involving weak nonlinearities. Thus, it tends to be more broadly applicable, including to problems classed as ODE systems, partial differential equation systems, integro-differential systems, and algebraic systems [40]. Even before considering the efforts of authors such as Turkyilmazoglu to accelerate the convergence of ADM [16], Li and Pang acknowledge the rapidly convergent series it can generate [40]. The approach has also been noted to be easier to automate than perturbation methods in many cases. However, the method's convergence rate and overall precision are noted to be sensitive to the level of truncation imposed on the solution approximation. The complexity of computing the Adomian polynomials for strongly nonlinear problems in this technique also frequently requires the use of numerical methods to resolve - often rendering this technique semi-analytical rather than strictly analytical.

The homotopy perturbation method has strengths and weaknesses similarly attributed to ADM. Its key benefit is its very broad applicability across classes of nonlinear problems - not requiring the nonlinearity to be weak. While not offering the same control over convergence speed as ADM, HPM does have the advantage of being typically simpler to compute (since complex Adomian polynomials are not required for this approach). This lack of control over convergence can be seen as a disadvantage when comparing the method with others. Similarly, the sensitivity of the overall accuracy and convergence to the choice of initial guess and construction of the homotopy equation does require problem-specific insight. Additionally, HPM is substantially less suitable for discontinuous systems than techniques such as piecewise-linear approximations [41].

Lastly, the harmonic balance method is often considered best-suited for obtaining approximate analytical solutions to nonlinear oscillators, particularly when the system exhibits strong nonlinearities or multiple harmonics [27]. However, as is demonstrated even in the simple example shown for this method earlier in this chapter, the technique can often be rendered semi-analytical - requiring the involvement of numerical methods to solve the algebraic systems - by the inclusion of relatively simple nonlinear terms like the cubic stiffness in the hardening Duffing oscillator. Such complications, alongside

Chapter 1. Introduction

the required trade-off between the higher accuracy and higher computational cost offered by considering higher harmonics, results in an often cumbersome problem which some authors have sought to overcome with modifications made to the base method [28] [42].

With the outline of the ADAP method detailed in this chapter, Chapter 3 presents a case-study of its application. After this, the ADAP method presented here is used to develop a generalised architecture which improves the effectiveness of linear model-predictive control for controlling nonlinear systems in Chapters 4-6.

Method	Notable strengths	Weaknesses
Strained-coordinate perturbation methods	Well-studied variations of these methods exist to allow them to be applied in a wide array of problems (particularly oscillators) to fruitful effect.	Reliant on the nonlinearity of the system being “weak”.
ADM	Rapidly convergent and can handle strong nonlinearities. Easier to automate than perturbation methods.	Computing Adomian polynomials for complicated problems can require numerical solvers, making this a “semi-analytical” technique.
VIM	Promising degree of control over the convergence properties via the Lagrange multipliers.	Can feature repetitive calculations (when solved analytically and unless offline computation is used to streamline them).
HPM	Simpler to compute than ADM and VIM (avoiding the computation of these methods polynomials or Lagrange multipliers respectively). Widely applicable, without reliance on “weak” nonlinearities.	Less widely-studied and accepted (than other methods in this table) by the scientific community.
HBM	Converts differential equations into algebraic problems to be solved. Highly-suited to oscillators with multiple harmonics.	Systems requiring higher harmonics will see a significantly increased computational cost. Solution of the algebraic equations often requires numerical approaches, making this a “semi-analytical” technique (like ADM).
Jacobian linearisation	Widely-used in classical and modern control systems. Can assess local stability near equilibrium points.	Cannot capture global behaviour (“far” from defined operating points).
Piecewise linear approximation	Simple to compute and suitable for systems with discontinuities or hardware-embedded applications.	High computational burden when many of these approximations are required.

Table 1.2: A summary of the strengths and weaknesses of the approximate analytical techniques discussed in Chapter 1.

Chapter 2

A novel approximation algorithm for the solution of nonlinear and coupled systems

Inclusion of previous work

This chapter includes content from two previous works by the author. These are: *A novel approximation method for the solution of weakly nonlinear coupled systems*, which is published in *Nonlinear Dynamics* [43]; and, *Application of the ADAP method to model-based control of the FutureForge manipulator*, which is published in *Journal of Sound and Vibration* [44]. These earlier works by D Johnston and MP Cartmell, feature the first published and peer-reviewed work on the ADAP methodology, which is the central focus of this thesis.

This chapter introduces the novel algorithm which is proposed in this thesis and developed to solve nonlinear, coupled systems of ordinary differential equations. Distinguishing it from other methods, it has particularly convenient integration with model-predictive control (MPC) - which is detailed in Chapters 4-6 - while also sharing a high-level characteristic of existing approaches as outlined in Chapter 1: initially rough approximations being augmented by subsequent corrections. In general terms, the theory of this novel approach can be simplified to the following explanation. A linearised

Chapter 2. A novel approximation algorithm for the solution of nonlinear and coupled systems

(approximated) system’s behaviour can be used to approximate terms in the original (unapproximated) system as necessary to render it analytically solvable.

In this chapter, the application of this methodology to two example problems is explored. The intention of the following work is to guide the reader through the natural evolution of the ADAP methodology before exploring further problems to demonstrate its application and extension to model-based control in Chapter 4. The application of the ADAP method is explored for two systems in this chapter: a system of coupled Rayleigh oscillators and a system of coupled Van der Pol oscillators. These problems are useful as they allow for appropriate benchmarking of the ADAP method against another approximation technique - the Lindstedt-Poincaré technique - which is well-suited to solving these systems. These two examples are intended to demonstrate the application of the ADAP method, while a generalisation of the method is detailed in Section 2.4. In the subsequent chapter, this is further explored for the governing dynamics of the *FutureForge* manipulator. The inclusion of this subsequent example allows examination of the ADAP methodology outside of this chapter’s oscillatory dynamic systems. For a system containing nonlinearities and coupling effects to varying extents, the solution of this system would (hypothetically) be greatly eased if these effects could be neglected or otherwise “countered”: trading-off complexity and accuracy in favour of ease of use and analytical tractability. This simple premise sets the foundation for the examples explored in the following material. Specifically, these will be used to inspect the extent to which this idea can be used to practical effect. Although the method evolves throughout this chapter before being summarised and presented in its final form at this chapter’s conclusion, the thread connecting each step of this exploration is the idea that some terms can be neglected initially to permit the solution of a relatively simple problem before being reconstituted into the original problem in some way to produce an approximate solution.

Note that throughout this chapter, the approximations resulting from the ADAP method are compared to a more widely adopted approach used in control-centred lit-

Chapter 2. A novel approximation algorithm for the solution of nonlinear and coupled systems

erature. Consistent with approaches such as [45], [46], and [47], which are discussed in more detail in Chapter 4, this standard approach involves neglecting or vastly simplifying the nonlinear and coupling effects. As will be observed in the following sections, the so-called “first auxiliary problems” are good representations of these simplifications. The method used in this work to solve the first auxiliary problems is the well-known Lindstedt-Poincaré technique, given its suitability to a wide range of problems (and not uniquely to oscillator problems) as discussed in [5], [6], [7], and [8] as cited in Chapter 1.

2.1 A system of coupled Rayleigh oscillators

Initially derived to describe the acoustic vibration of a clarinet reed, the Rayleigh oscillator has now been used in many diverse applications [48]. Similar to the later-discussed Van der Pol oscillator, some interesting applications are in the field of biomechanics. Fu et al. [49] modelled the coupled motion of two limb joints (treated as a two-degree-of-freedom manipulator) using a system of Rayleigh oscillators. This two-link manipulator comparison also occurs in de Pina Filho, Dutra, and Raptopoulos’ work [50] where robotic bipedal motion is also considered. Other, perhaps more expected, applications include modelling of oscillatory phenomena in chemical reactions [51].

2.1.1 Problem specification

In this section, the following system of coupled oscillators is considered.

$$\begin{Bmatrix} \ddot{y}_1 \\ \ddot{y}_2 \end{Bmatrix} + \varepsilon \begin{bmatrix} \dot{y}_1^2 - 1 & -1 \\ -1 & \dot{y}_2^2 - 1 \end{bmatrix} \begin{Bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{Bmatrix} + \begin{Bmatrix} y_1 \\ y_2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \quad (2.1)$$

This system of equations can be more conveniently written in the form

$$\ddot{\mathbf{Y}} + P(\dot{\mathbf{Y}}; \varepsilon) \dot{\mathbf{Y}} + \mathbf{Y} = \mathbf{0} \quad (2.2)$$

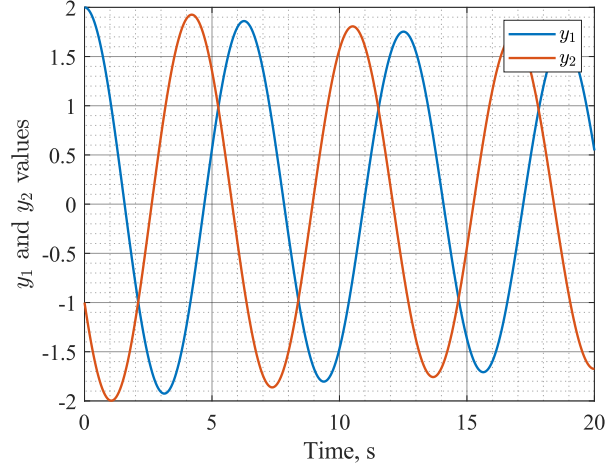


Figure 2.1: Numerical solutions to Eq (2.1) using *ode45* in *MATLAB*.

where $\mathbf{Y} = [y_1(t), y_2(t)]^T$ is the vector of displacements and $P(\dot{\mathbf{Y}}; \varepsilon)$ is the damping matrix including the small parameter ε .

$$P(\dot{\mathbf{Y}}; \varepsilon) = \varepsilon \begin{bmatrix} \dot{y}_1^2 - 1 & -1 \\ -1 & \dot{y}_2^2 - 1 \end{bmatrix} \quad (2.3)$$

Recalling the premise outlined at the start of this chapter, the first step in the algorithm being explored is to obtain a similar system of equations for which an analytical solution can be obtained with relative ease. For the purposes of visualisation, the numerical solution of Eq (2.1) is depicted in Fig 2.1 where $\mathbf{Y}(t = 0) = [2, -1]^T$ and $\dot{\mathbf{Y}}(t = 0) = [0, -\sqrt{3}]^T$.

2.1.2 Construction and solution of the uncoupled system

A “simpler” version of the target system, Eq (2.1), could be written as

$$\begin{Bmatrix} \ddot{\phi}_1 \\ \ddot{\phi}_2 \end{Bmatrix} + \varepsilon \begin{bmatrix} \dot{\phi}_1^2 - 1 & 0 \\ 0 & \dot{\phi}_2^2 - 1 \end{bmatrix} \begin{Bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \end{Bmatrix} + \begin{Bmatrix} \phi_1 \\ \phi_2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}. \quad (2.4)$$

This may be an appropriate initial problem since a solution could be obtained via a widely-used technique such as the Lindstedt-Poincaré method. Note that the $\phi_k = \phi_k(t)$

Chapter 2. A novel approximation algorithm for the solution of nonlinear and coupled systems

notation is used in place of y_k since the solution of this system will not be equivalent to the solution of the target system (2.1). By approaching Eq (2.4) via the Lindstedt-Poincaré method, n^{th} -order asymptotic approximations can be obtained in the form

$$\phi_k \approx \sum_{j=0}^n \varepsilon^j \phi_{k,j}(\tau_k) \quad (2.5)$$

where $k \in \{1, 2\}$ and $n \in \{0, 1, 2, \dots\}$ and τ_k is the stretched coordinate

$$\tau_k = t \sum_{j=0}^n \varepsilon^j \omega_{k,j} \quad (2.6)$$

and ω_j are selected parameters designed to remove secularity from the resulting solutions. As noted in Chapter 1, the removal of secular terms is crucial in order to yield uniformly valid approximations over the domain of interest. By substituting Eq (2.5) and its first two time derivatives into the uncoupled system (2.4), the following $O(\varepsilon^0)$ perturbation equations for $\phi_1(t)$ and $\phi_2(t)$ can be obtained respectively.

$$\omega_{1,0}^2 \phi_{1,0}''(\tau_1) + \phi_{1,0}(\tau_1) = 0 \quad (2.7)$$

$$\omega_{2,0}^2 \phi_{2,0}''(\tau_2) + \phi_{2,0}(\tau_2) = 0 \quad (2.8)$$

Note that the “prime”-notation, $\phi'_k(\tau)$, is used to denote differentiation with respect to τ as opposed to the “dot”-notation’s use to denote differentiation with respect to t . The solution to these generating problems is simply that of simple harmonic oscillators with arbitrary initial conditions where the frequencies are dependent on the stretched-time parameters.

$$\phi_{1,0} = y_1(0) \cos\left(\frac{\tau_1}{\omega_{1,0}}\right) + \omega_{1,0} \dot{y}_1(0) \sin\left(\frac{\tau_1}{\omega_{1,0}}\right) \quad (2.9)$$

$$\phi_{2,0} = y_2(0) \cos\left(\frac{\tau_2}{\omega_{2,0}}\right) + \omega_{2,0} \dot{y}_2(0) \sin\left(\frac{\tau_2}{\omega_{2,0}}\right) \quad (2.10)$$

These solutions to the generating problems, Eqs (2.7) and (2.8), are illustrated as shown in Fig 2.2.

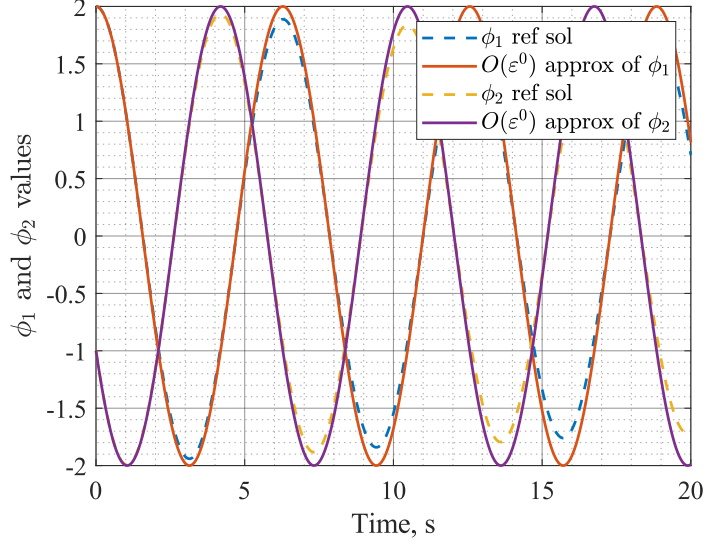


Figure 2.2: Numerical solutions for Eq (2.4), “ref sol” in the legend, compared to the corresponding $O(1)$ solutions to the generating Lindstedt-Poincaré problems.

The $O(\varepsilon)$ -problems associated with this method can then be written as

$$\begin{aligned} \omega_{1,0}^2 \phi_{1,1}''(\tau_1) + \phi_{1,1}(\tau_1) &= \omega_0^3 \left(b \cos\left(\frac{\tau_1}{\omega_0}\right) - \frac{a \sin\left(\frac{\tau_1}{\omega_0}\right)}{\omega_0} \right)^3 \\ &\quad - \omega_0 \left(b \cos\left(\frac{\tau_1}{\omega_0}\right) - \frac{a \sin\left(\frac{\tau_1}{\omega_0}\right)}{\omega_0} \right) \\ &\quad + 2\omega_1 \omega_0 \left(-\frac{a \cos\left(\frac{\tau_1}{\omega_0}\right)}{\omega_0^2} - \frac{b \sin\left(\frac{\tau_1}{\omega_0}\right)}{\omega_0} \right) \end{aligned} \quad (2.11)$$

$$\begin{aligned} \omega_{2,0}^2 \phi_{2,1}''(\tau_2) + \phi_{2,1}(\tau_2) &= \omega_0^3 \left(d \cos\left(\frac{\tau_2}{\omega_0}\right) - \frac{c \sin\left(\frac{\tau_2}{\omega_0}\right)}{\omega_0} \right)^3 \\ &\quad - \omega_0 \left(d \cos\left(\frac{\tau_2}{\omega_0}\right) - \frac{c \sin\left(\frac{\tau_2}{\omega_0}\right)}{\omega_0} \right) \\ &\quad + 2\omega_1 \omega_0 \left(-\frac{c \cos\left(\frac{\tau_2}{\omega_0}\right)}{\omega_0^2} - \frac{d \sin\left(\frac{\tau_2}{\omega_0}\right)}{\omega_0} \right) \end{aligned} \quad (2.12)$$

The process of solving Eqs (2.11) and (2.12) is as follows. Firstly, by setting the left-hand side of these equations equal to zero, the complementary functions are determined

Chapter 2. A novel approximation algorithm for the solution of nonlinear and coupled systems

similarly to Eqs (2.9) and (2.10). Then, the particular integrals are determined through the method of variation of parameters (which is more thoroughly detailed for the ADAP method shown later in this section). The complementary function and particular integral derived for each differential equation can then be summed to give the following results.

$$\begin{aligned} \phi_{1,1} = & -\frac{1}{16\omega_{1,0}^2} \left(2\tau_1\omega_{1,0} (3a^3 - 4a + 8b\omega_{1,1}) \cos\left(\frac{\tau_1}{\omega_{1,0}}\right) \right. \\ & - \omega_{1,0}^2 \left(a^3 \cos\left(\frac{2\tau_1}{\omega_{1,0}}\right) + 5a^3 - 6a^2b\tau_1 - 8a + 8b\tau_1 + 16b\omega_{1,1} \right) \sin\left(\frac{\tau_1}{\omega_{1,0}}\right) \\ & + 3b^2\omega_{1,0}^4 \left(a \cos\left(\frac{2\tau_1}{\omega_{1,0}}\right) - 3a + 2b\tau_1 \right) \sin\left(\frac{\tau_1}{\omega_{1,0}}\right) \\ & + 3ab\omega_{1,0}^3 \left(a \cos\left(\frac{2\tau_1}{\omega_0}\right) - a + 2b\tau_1 \right) \cos\left(\frac{\tau_1}{\omega_{1,0}}\right) \\ & \left. - 16a\tau_1\omega_{1,1} \sin\left(\frac{\tau_1}{\omega_{1,0}}\right) + b^3\omega_{1,0}^5 \sin\left(\frac{\tau_1}{\omega_{1,0}}\right) \sin\left(\frac{2\tau_1}{\omega_{1,0}}\right) \right) \end{aligned} \quad (2.13)$$

$$\begin{aligned} \phi_{2,1} = & -\frac{1}{16\omega_{2,0}^2} \left(2\tau_2\omega_{2,0} (3c^3 - 4c + 8d\omega_{2,1}) \cos\left(\frac{\tau_2}{\omega_{2,0}}\right) \right. \\ & - \omega_{2,0}^2 \sin\left(\frac{\tau_2}{\omega_{2,0}}\right) \left(c^3 \cos\left(\frac{2\tau_2}{\omega_{2,0}}\right) + 5c^3 - 6c^2d\tau_2 - 8c + 8d\tau_2 + 16d\omega_{2,1} \right) \\ & + 3d^2\omega_{2,0}^4 \sin\left(\frac{\tau_2}{\omega_{2,0}}\right) \left(c \cos\left(\frac{2\tau_2}{\omega_{2,0}}\right) - 3c + 2d\tau_2 \right) \\ & + 3cd\omega_{2,0}^3 \cos\left(\frac{\tau_2}{\omega_{2,0}}\right) \left(c \cos\left(\frac{2\tau_2}{\omega_{2,0}}\right) - c + 2d\tau_2 \right) \\ & \left. - 16c\tau_2\omega_{2,1} \sin\left(\frac{\tau_2}{\omega_{2,0}}\right) + d^3\omega_{2,0}^5 \sin\left(\frac{\tau_2}{\omega_{2,0}}\right) \sin\left(\frac{2\tau_2}{\omega_{2,0}}\right) \right) \end{aligned} \quad (2.14)$$

As per the examples presented in Chapter 1, the stretched coordinates may then be selected so as to exclude any secular terms from the solution. Thus, combining the solutions at the $O(1)$ - and $O(\varepsilon)$ -level as per Eq (2.5) results in the solutions as depicted in Fig 2.3. In order to eliminate secular terms in the scenario depicted in Fig 2.3,

Chapter 2. A novel approximation algorithm for the solution of nonlinear and coupled systems

the initial conditions $[a, b, c, d] = [2, 0, -1, -\sqrt{3}]$ give rise to the strained-coordinate parameters $\omega_{1,1} = \omega_{2,1} = 0$.

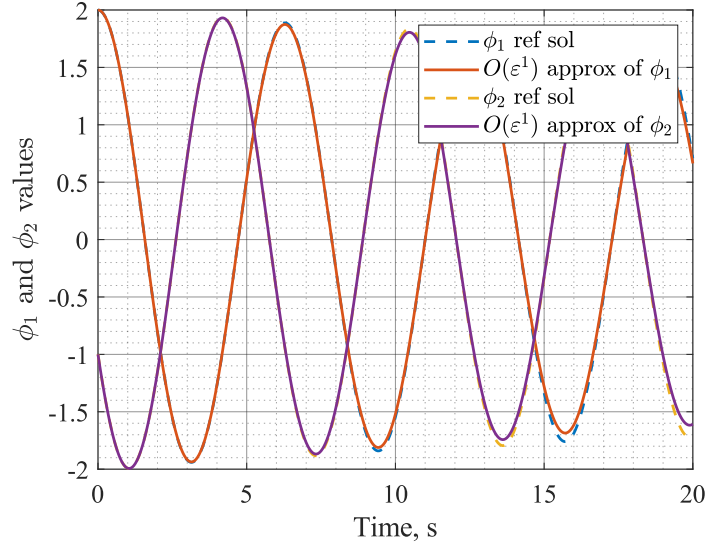


Figure 2.3: Numerical solutions for Eq (2.4), “ref sol” in the legend, compared to the corresponding $O(\varepsilon)$ approximations based on the Lindstedt-Poincaré approach.

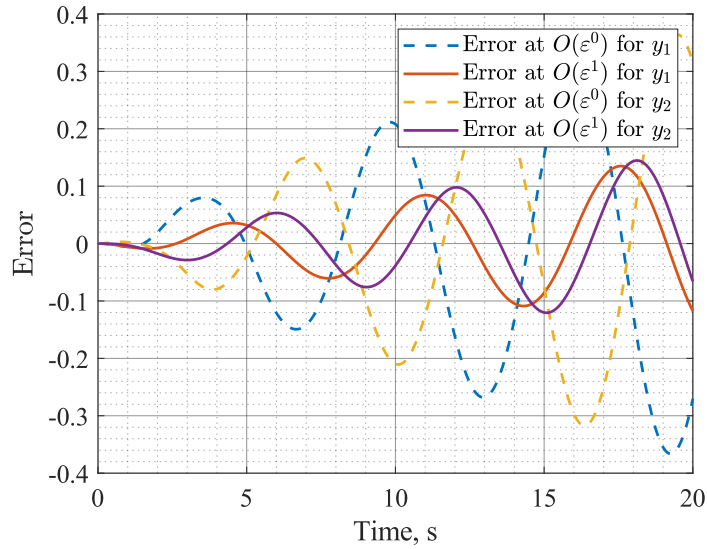


Figure 2.4: The errors accumulated by the $O(1)$ and $O(\varepsilon)$ approximations in solving Eq (2.1).

Intuitively, as illustrated in Fig 2.4, the error accumulated by the Lindstedt-Poincaré

method is reduced when the $O(\varepsilon)$ correction is included in the approximation.

2.1.3 Approximation and solution of the fully-coupled system

Recalling again the premise posed at the beginning of this chapter, the next step of the ADAP method must be to reconstitute the previously neglected terms back into the system. The first method that was explored for this step was to consider the damping terms with ϕ_1 and ϕ_2 substituted in place of y_1 and y_2 respectively.

$$\begin{aligned}\mathbf{R} &\approx -P(\dot{\mathbf{Y}}; \varepsilon)\dot{\mathbf{Y}} \\ \mathbf{R} &= -\varepsilon \begin{bmatrix} \dot{\phi}_1^2 - 1 & -1 \\ -1 & \dot{\phi}_2^2 - 1 \end{bmatrix} \begin{Bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \end{Bmatrix} = \begin{Bmatrix} r_1 \\ r_2 \end{Bmatrix}\end{aligned}\quad (2.15)$$

Using Eq (2.15) to approximate the nonlinear, coupled damping terms in the target system, Eq (2.1), the system can be simplified into two uncoupled simple harmonic oscillators with excitations applied to both.

$$\ddot{\mathbf{f}} + \mathbf{f} = \mathbf{R} \quad (2.16)$$

Note that $\mathbf{f}$, if this proposed method is effective, should approximate $\mathbf{Y}$ in a similar manner that $\phi_{1,0}(\tau_1)$ approximates $\phi_1(t)$ in Eq (2.7). In previous work [43], this additional notation was not used. On reflection, however, it's use is suitable since the assertion of this process is that the resulting solution of Eq (2.16) is an **approximation** of the solution of Eq (2.1), not that the two solutions are equivalent.

Finding the complementary functions for Eq (2.16) is relatively simple. For generality, the forms

$$f_{1,\text{cf}} = K_{1,1}\psi_1(t) + K_{1,2}\psi_2(t) \quad (2.17)$$

and

$$f_{2,\text{cf}} = K_{2,1}\psi_1(t) + K_{2,2}\psi_2(t) \quad (2.18)$$

are used where $\psi_1 = \psi_1(t)$ and $\psi_2 = \psi_2(t)$ are the bases of the solutions and the four

Chapter 2. A novel approximation algorithm for the solution of nonlinear and coupled systems

K -terms are constants to be determined by the initial conditions of the system. Using $\psi_1 = \cos(t)$ and $\psi_2 = \sin(t)$, the method of variation of parameters is used to compute the general form of the particular integral. Finding the Wronskian

$$\begin{aligned} W &= \psi_1 \dot{\psi}_2 + \psi_2 \dot{\psi}_1 \\ &= \cos^2(t) + \sin^2(t) \\ &= 1 \end{aligned} \tag{2.19}$$

the particular integral is found via

$$f_{1,p} = -\psi_1 \int \frac{\psi_2 r_1}{W} dt + \psi_2 \int \frac{\psi_1 r_1}{W} dt \tag{2.20}$$

and

$$f_{2,p} = -\psi_1 \int \frac{\psi_2 r_2}{W} dt + \psi_2 \int \frac{\psi_1 r_2}{W} dt. \tag{2.21}$$

Note that $W = 1$ implies that the solutions are linearly independent (which is the case for any non-zero constant value of the Wronskian). Computing Eq (2.20) is carried out in the following four steps. Firstly, the first integrand in Eq (2.20) is determined.

$$\begin{aligned} \frac{\psi_2 r_1}{W} &= -\varepsilon \sin(t) \left(((\dot{y}_1(0) \cos(t) - y_1(0) \sin(t)) ((\dot{y}_1(0) \cos(t) - y_1(0) \sin(t))^2 - 1) \right. \\ &\quad \left. + y_2(0) \sin(t) - \dot{y}_2(0) \cos(t) \right) \end{aligned} \tag{2.22}$$

This is followed by determining the second integrand in Eq (2.20).

$$\begin{aligned} \frac{\psi_1 r_1}{W} &= -\varepsilon \cos(t) \left(((\dot{y}_1(0) \cos(t) - y_1(0) \sin(t)) ((\dot{y}_1(0) \cos(t) - y_1(0) \sin(t))^2 - 1) \right. \\ &\quad \left. + y_2(0) \sin(t) - \dot{y}_2(0) \cos(t) \right) \end{aligned} \tag{2.23}$$

The integrals arising from the preceding two integrands are then computed in their

Chapter 2. A novel approximation algorithm for the solution of nonlinear and coupled systems

general forms here as

$$\begin{aligned} \int \frac{\psi_2 r_1}{W} dt = & -\frac{1}{32} \varepsilon \left(4t \left(-3y_1(0)^3 + y_1(0)(4 - 3\dot{y}_1(0)^2) + 4y_2(0) \right) \right. \\ & - 8 \left(-y_1(0)^3 + y_1(0) + y_2(0) \right) \sin(2t) - 4 \cos(2t) \left((3y_1(0)^2 - 2)\dot{y}_1(0) \right. \\ & + \dot{y}_1(0)^3 - 2\dot{y}_2(0) \left. \right) - y_1(0) \left(y_1(0)^2 - 3\dot{y}_1(0)^2 \right) \sin(4t) \\ & \left. - \dot{y}_1(0) \left(\dot{y}_1(0)^2 - 3y_1(0)^2 \right) \cos(4t) \right) + c_1 \end{aligned} \quad (2.24)$$

and

$$\begin{aligned} \int \frac{\psi_1 r_1}{W} dt = & -\frac{1}{32} \varepsilon \left(4 \cos(2t) \left(y_1(0)^3 + y_1(0)(3\dot{y}_1(0)^2 - 2) - 2y_2(0) \right) \right. \\ & + 4t \left((3y_1(0)^2 - 4)\dot{y}_1(0) + 3\dot{y}_1(0)^3 - 4\dot{y}_2(0) \right) \\ & + \dot{y}_1(0) \left(\dot{y}_1(0)^2 - 3y_1(0)^2 \right) \sin(4t) - y_1(0) \left(y_1(0)^2 - 3\dot{y}_1(0)^2 \right) \cos(4t) \\ & \left. - 8 \left(-\dot{y}_1(0)^3 + \dot{y}_1(0) + \dot{y}_2(0) \right) \sin(2t) \right) + c_2, \end{aligned} \quad (2.25)$$

with c_1 and c_2 being the as-yet-undetermined constants of integration, before being combined as per Eq (2.20) to give

$$\begin{aligned} f_{1,p} = & \frac{1}{32} \varepsilon \left(\left(\dot{y}_1(0)^3 - 3y_1(0)^2 \dot{y}_1(0) \right) \cos(3t) \right. \\ & - 2 \cos(t) \left(6y_1(0)^3 t + (3y_1(0)^2 - 4)\dot{y}_1(0) + 6y_1(0)\dot{y}_1(0)^2 t \right. \\ & \left. - 8y_1(0)t + 3\dot{y}_1(0)^3 - 8y_2(0)t - 4\dot{y}_2(0) \right) \\ & + \sin(t) \left(7y_1(0)^3 + 2y_1(0)(y_1(0)^2 - 3\dot{y}_1(0)^2) \cos(2t) - 12y_1(0)^2 \dot{y}_1(0)t \right. \\ & \left. + y_1(0)(3\dot{y}_1(0)^2 - 8) + 4t(-3\dot{y}_1(0)^3 + 4\dot{y}_1(0) + 4\dot{y}_2(0)) - 8y_2(0) \right) \left. \right). \end{aligned} \quad (2.26)$$

The same method is followed to compute Eq (2.21). The required integrands are first

Chapter 2. A novel approximation algorithm for the solution of nonlinear and coupled systems

computed.

$$\begin{aligned} \frac{\psi_2 r_2}{W} = & -\varepsilon \sin(t) \left(y_1(0) \sin(t) - \dot{y}_1(0) \cos(t) \right. \\ & \left. + \left(\dot{y}_2(0) \cos(t) - y_2(0) \sin(t) \right) \left(\left(\dot{y}_2(0) \cos(t) - y_2(0) \sin(t) \right)^2 - 1 \right) \right) \end{aligned} \quad (2.27)$$

$$\begin{aligned} \frac{\psi_1 r_2}{W} = & -\varepsilon \cos(t) \left(y_1(0) \sin(t) - \dot{y}_1(0) \cos(t) \right. \\ & \left. + \left(\dot{y}_2(0) \cos(t) - y_2(0) \sin(t) \right) \left(\left(\dot{y}_2(0) \cos(t) - y_2(0) \sin(t) \right)^2 - 1 \right) \right) \end{aligned} \quad (2.28)$$

Subsequently, the integrals following from these results are computed.

$$\begin{aligned} \int \frac{\psi_2 r_2}{W} dt = & -\frac{1}{32} \varepsilon \left(-8 \left(y_1(0) - y_2(0)^3 + y_2(0) \right) \sin(2t) \right. \\ & + 4t \left(4y_1(0) + y_2(0) \left(-3y_2(0)^2 - 3\dot{y}_2(0)^2 + 4 \right) \right) \\ & + 4 \cos(2t) \left(2\dot{y}_1(0) - \dot{y}_2(0) \left(3y_2(0)^2 + \dot{y}_2(0)^2 - 2 \right) \right) \\ & \left. - y_2(0) \left(y_2(0)^2 - 3\dot{y}_2(0)^2 \right) \sin(4t) - \dot{y}_2(0) \left(y_2(0)^2 - 3y_2(0)^2 \right) \cos(4t) \right) + c_3 \end{aligned} \quad (2.29)$$

$$\begin{aligned} \int \frac{\psi_1 r_2}{W} dt = & -\frac{1}{32} \varepsilon \left(-4 \cos(2t) \left(2y_1(0) - y_2(0) \left(y_2(0)^2 + 3\dot{y}_2(0)^2 - 2 \right) \right) \right. \\ & - 4t \left(4\dot{y}_1(0) + \dot{y}_2(0) \left(-3y_2(0)^2 - 3\dot{y}_2(0)^2 + 4 \right) \right) \\ & - 8 \left(b - \dot{y}_2(0)^3 + \dot{y}_2(0) \right) \sin(2t) \\ & \left. + \dot{y}_2(0) \left(\dot{y}_2(0)^2 - 3y_2(0)^2 \right) \sin(4t) - y_2(0) \left(y_2(0)^2 - 3\dot{y}_2(0)^2 \right) \cos(4t) \right) + c_4 \end{aligned} \quad (2.30)$$

where c_3 and c_4 are constants of integration. These are then compiled in simplifying

Chapter 2. A novel approximation algorithm for the solution of nonlinear and coupled systems

the resulting expression for Eq (2.21).

$$\begin{aligned}
f_{2,p} = \frac{1}{32}\varepsilon & \left(2 \cos(t) \left(8y_1(0)t + 4\dot{y}_1(0) - 6y_2(0)^3t - 3y_2(0)^2\dot{y}_2(0) \right. \right. \\
& \left. \left. - 6y_2(0)\dot{y}_2(0)^2t + 8y_2(0)t - 3\dot{y}_2(0)^3 + 4\dot{y}_2(0) \right) \right. \\
& + \sin(t) \left(-8y_1(0) + 16\dot{y}_1(0)t + 7y_2(0)^3 + 2y_2(0)(y_2(0)^2 - 3\dot{y}_2(0)^2) \cos(2t) \right. \\
& \left. - 12y_2(0)^2\dot{y}_2(0)t + 3y_2(0)\dot{y}_2(0)^2 - 8y_2(0) - 12\dot{y}_2(0)^3t + 16\dot{y}_2(0)t \right) \\
& \left. + \left(\dot{y}_2(0)^3 - 3y_2(0)^2\dot{y}_2(0) \right) \cos(3t) \right)
\end{aligned} \tag{2.31}$$

The general solutions can thus be found by summing the complementary functions, Eqs (2.17) and (2.18), and the particular integrals, Eqs (2.20) and (2.21).

$$\mathbf{f} = \begin{Bmatrix} f_{1,cf} + f_{1,p} \\ f_{2,cf} + f_{2,p} \end{Bmatrix} \tag{2.32}$$

The particular solution is then obtained by applying the initial conditions and solving for the K -terms in Eqs (2.17) and (2.18). Note, in Eq (2.33), $f_{1,p}(t) = f_{1,p}$ refers to Eq (2.26) and $f_{2,p}(t) = f_{2,p}$ refers to Eq (2.31).

$$\mathbf{f} = \begin{bmatrix} K_{1,1} & K_{1,2} \\ K_{2,1} & K_{2,2} \end{bmatrix} \begin{Bmatrix} \psi_1(t) \\ \psi_2(t) \end{Bmatrix} + \begin{Bmatrix} f_{1,p}(t) \\ f_{2,p}(t) \end{Bmatrix} \tag{2.33}$$

As described, Eq (2.33) can be solved by setting $t = 0$ and substituting-in the known conditions of the system at that time where $\mathbf{f}(t = 0) = \mathbf{y}(t = 0)$ and $\dot{\mathbf{f}}(t = 0) = \dot{\mathbf{y}}(t = 0)$. Taking Eq (2.33) in conjunction with its first derivative

$$\dot{\mathbf{f}} = \begin{bmatrix} K_{1,1} & K_{1,2} \\ K_{2,1} & K_{2,2} \end{bmatrix} \begin{Bmatrix} \dot{\psi}_1(t) \\ \dot{\psi}_2(t) \end{Bmatrix} + \begin{Bmatrix} \dot{f}_{1,p}(t) \\ \dot{f}_{2,p}(t) \end{Bmatrix}, \tag{2.34}$$

Chapter 2. A novel approximation algorithm for the solution of nonlinear and coupled systems

the system can be rewritten as

$$\begin{Bmatrix} f_1 \\ f_2 \\ \dot{f}_1 \\ \dot{f}_2 \end{Bmatrix} = \begin{bmatrix} \psi_1 & \psi_2 & 0 & 0 \\ 0 & 0 & \psi_1 & \psi_2 \\ \dot{\psi}_1 & \dot{\psi}_2 & 0 & 0 \\ 0 & 0 & \dot{\psi}_1 & \dot{\psi}_2 \end{bmatrix} \begin{Bmatrix} K_{1,1} \\ K_{1,2} \\ K_{2,1} \\ K_{2,2} \end{Bmatrix} + \begin{Bmatrix} f_{1,p} \\ f_{2,p} \\ \dot{f}_{1,p} \\ \dot{f}_{2,p} \end{Bmatrix}, \quad (2.35)$$

and thus

$$\begin{Bmatrix} K_{1,1} \\ K_{1,2} \\ K_{2,1} \\ K_{2,2} \end{Bmatrix} = \begin{bmatrix} \psi_1(0) & \psi_2(0) & 0 & 0 \\ 0 & 0 & \psi_1(0) & \psi_2(0) \\ \dot{\psi}_1(0) & \dot{\psi}_2(0) & 0 & 0 \\ 0 & 0 & \dot{\psi}_1(0) & \dot{\psi}_2(0) \end{bmatrix}^{-1} \left(\begin{Bmatrix} f_1(0) \\ f_2(0) \\ \dot{f}_1(0) \\ \dot{f}_2(0) \end{Bmatrix} - \begin{Bmatrix} f_{1,p}(0) \\ f_{2,p}(0) \\ \dot{f}_{1,p}(0) \\ \dot{f}_{2,p}(0) \end{Bmatrix} \right). \quad (2.36)$$

The K -terms are therefore evaluated as

$$\begin{Bmatrix} K_{1,1} \\ K_{1,2} \\ K_{2,1} \\ K_{2,2} \end{Bmatrix} = \frac{1}{32} \begin{Bmatrix} 32y_1(0) - 8\dot{y}_1(0)\varepsilon + 9y_1(0)^2\dot{y}_1(0)\varepsilon + 5\dot{y}_1(0)^3\varepsilon - 8\dot{y}_2(0)\varepsilon \\ 3y_1(0)^3\varepsilon + 15y_1(0)\dot{y}_1(0)^2\varepsilon - 8y_1(0)\varepsilon + 32\dot{y}_1(0) - 8y_2(0)\varepsilon \\ -8\dot{y}_1(0)\varepsilon + 9y_2(0)^2\dot{y}_2(0)\varepsilon + 32y_2(0) + 5\dot{y}_2(0)^3\varepsilon - 8\dot{y}_2(0)\varepsilon \\ -8y_1(0)\varepsilon + 3y_2(0)^3\varepsilon + 15y_2(0)\dot{y}_2(0)^2\varepsilon - 8y_2(0)\varepsilon + 32\dot{y}_2(0) \end{Bmatrix}. \quad (2.37)$$

Specifically, when $\mathbf{y}(t = 0) = [2, -1]^T$, $\dot{\mathbf{y}}(t = 0) = [0, -\sqrt{3}]^T$, and $\varepsilon = 0.01$, the K -values become $K_{1,1} = 2.0043$, $K_{1,2} = 0.0050$, $K_{2,1} = -1.0087$, and $K_{2,2} = -1.7496$.

For completeness, the ADAP approximate analytical solution of Eq (2.1) consists of the following two expressions when subject to the previously-stated conditions and

Chapter 2. A novel approximation algorithm for the solution of nonlinear and coupled systems parameters.

$$\begin{aligned}
y_1 &\approx 2.0043 \cos(t) + 0.0050 \sin(t) \\
&\quad + 0.0003 \cos(t) \left(-80t + 56 \sin(2t) - 8 \sin(4t) - 8\sqrt{3} \cos(2t) \right) \\
&\quad - 0.0003 \sin(t) \left(16\sqrt{3}t + 8\sqrt{3} \sin(2t) + 24 \cos(2t) - 8 \cos(4t) \right) \\
y_2 &\approx -1.7496 \sin(t) - 1.0087 \cos(t) \\
&\quad + 0.0003 \cos(t) \left(64t - 16 \sin(2t) - 8 \sin(4t) + 16\sqrt{3} \cos(2t) \right) \\
&\quad - 0.0003 \sin(t) \left(-32\sqrt{3}t - 16\sqrt{3} \sin(2t) - 48 \cos(2t) - 8 \cos(4t) \right)
\end{aligned} \tag{2.38}$$

$$\tag{2.39}$$

These results are specific to the system of coupled Rayleigh oscillators described by Eq (2.1). However, after undertaking a similar demonstration for a system of coupled Van der Pol oscillators, a generalisation of the ADAP methodology will be detailed in Section 2.4.

2.1.4 Results and discussion

Inspecting Fig 2.5, the approximate analytical solution produced by the ADAP methodology follows the reference numerical solutions, produced by the Runge-Kutta solver in MATLAB's *ode45*, reasonably closely. Inspecting Fig 2.6, the benefit of the proposed method can be seen in comparison to the $O(1)$ approximation via the Lindstedt-Poincaré technique. In this figure, the reduction in relative error for the ADAP methodology is significant. Although both techniques result in errors that grow over time, this growth is more gradual for the ADAP method for small values of ε (as illustrated in Fig 2.8 and onwards). The growth rates of these methods' errors are similar for $0.01 \leq \varepsilon \leq 0.05$, and the growth rate for the ADAP method's error is greater for higher values of ε . Note that, to avoid division by zero and to ensure a consistent comparative metric, the differences between the approximate analytical solutions and the numerical errors are divided by the corresponding initial displacement value (rather than any of the time-varying approximations of the solutions).

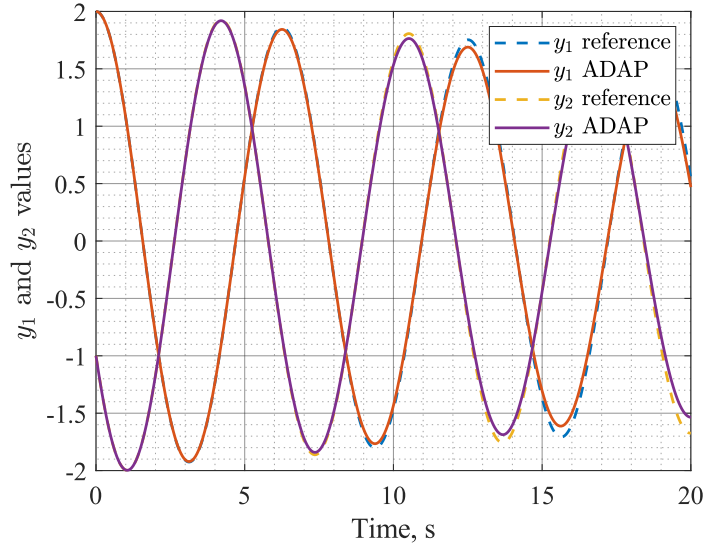


Figure 2.5: The approximate analytical solutions for Eq (2.1) achieved by the in-development ADAP method compared to the reference numerical solutions.

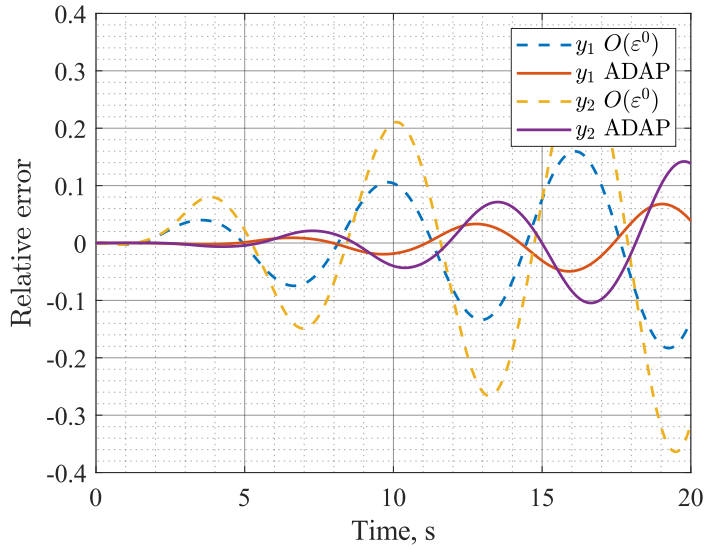


Figure 2.6: The errors produced in the approximate analytical solutions for Eq (2.1) achieved by the in-development ADAP method compared to the $O(1)$ asymptotic series approximation achieved via the Lindstedt-Poincaré technique.

The difference between the ADAP and $O(1)$ approximations can be found, through

simple subtraction of the latter from the former, to be the following expression.

$$0.0175 \sin(t) + 0.0025 \sin(3t) - 0.025t \cos(t) - 0.0087t \sin(t) \quad (2.40)$$

Note that the implied benefit of using the ADAP methodology by comparison to the $O(1)$ Lindstedt-Poincaré approximation grows with a linearly increasing amplitude across time.

Although the ADAP methodology still appears beneficial in comparison to the $O(\varepsilon)$ approximation, as shown in Fig 2.7, the benefit is less substantial than is observed in the comparison between the ADAP approximation and the $O(1)$ approximation. Intuitively, this is supported by the knowledge that the $O(\varepsilon)$ Lindstedt-Poincaré approximation is (by definition) a correction to the $O(1)$ approximation, see Fig 2.4, while the ADAP approximation is unchanged in these comparisons.

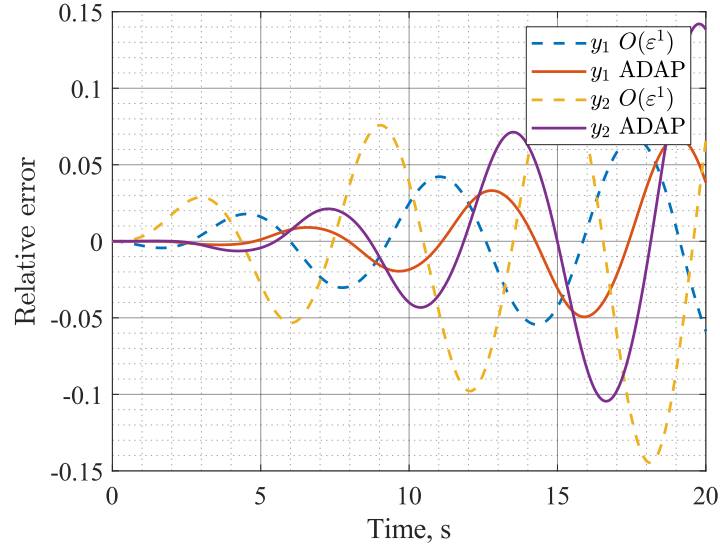


Figure 2.7: The errors produced in the approximate analytical solutions for Eq (2.1) achieved by the in-development ADAP method compared to the $O(\varepsilon)$ asymptotic series approximation achieved via the Lindstedt-Poincaré technique.

Similarly to the inspection of benefit made previously, the implied benefit of using the ADAP methodology by comparison to the $O(\varepsilon)$ Lindstedt-Poincaré approximation

Chapter 2. A novel approximation algorithm for the solution of nonlinear and coupled systems

grows with a linearly increasing amplitude across time.

$$0.005 \sin(t) - 0.005t \cos(t) - 0.0087t \sin(t) \quad (2.41)$$

Note, in particular, that the linearly growing cosine term in this expression and the previously state expression - for the $O(1)$ comparison - differ by a factor of 5. There is clearly less benefit in using the ADAP methodology if the $O(\varepsilon)$ solution is sufficiently tractable; however, there is still a significant benefit to be seen.

Varying strengths of the parameter ε can be used in the original model, Eq (2.1), to examine the ADAP method's "resilience" to stronger nonlinearities and coupling effects (although this will also be examined with other problems in the rest of this work).

Chapter 2. A novel approximation algorithm for the solution of nonlinear and coupled systems

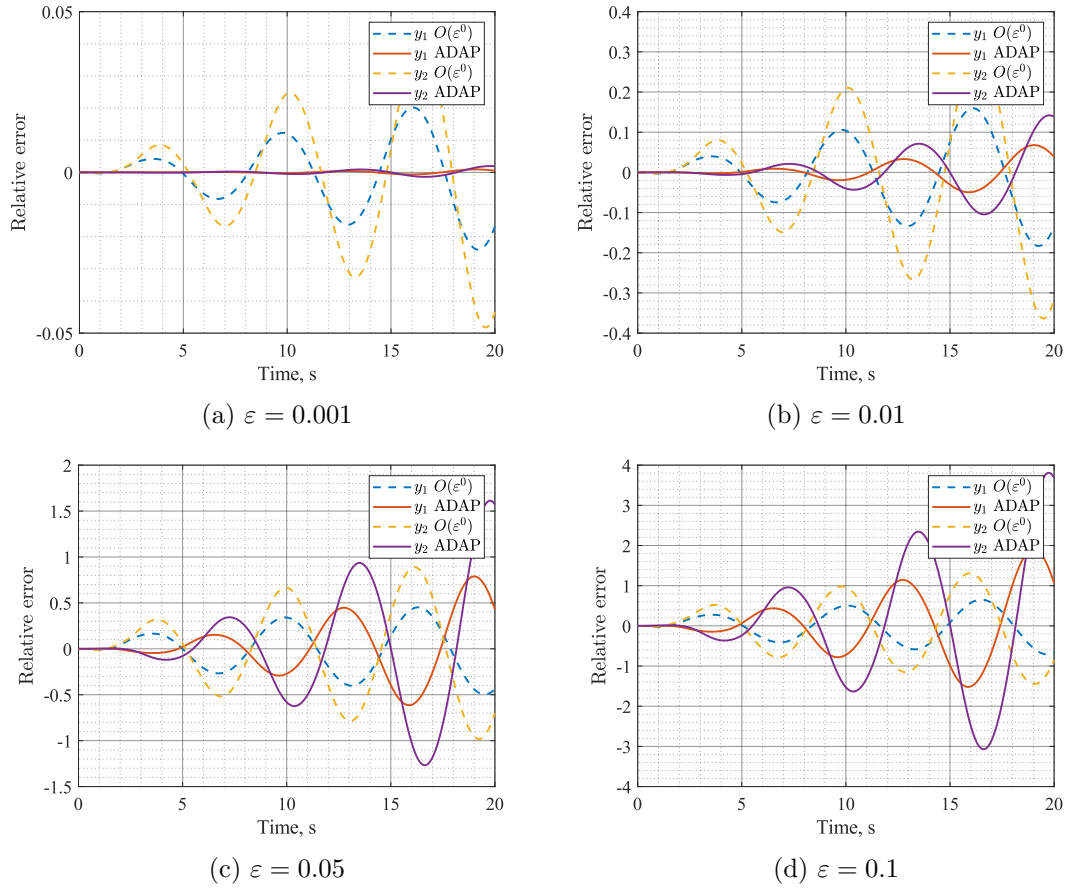


Figure 2.8: The relative error for the ADAP method compared to the Lindstedt-Poincaré technique to $O(1)$ with varying values of the small parameter, ε , in Eq (2.1).

As can be seen in Fig 2.8, based on this analysis, the ADAP method outperforms the Lindstedt-Poincaré technique's result at $O(1)$ for “very weak” nonlinearities such as in Fig 2.8a. However, although it briefly retains a lower relative error for the stronger nonlinearities in Figs 2.8c and 2.8d, this advantage is quickly lost by around 7 seconds. It is worth keeping in mind, however, that the Lindstedt-Poincaré technique is also unsuited to higher values of ε . Similar results can be seen in Fig 2.9.

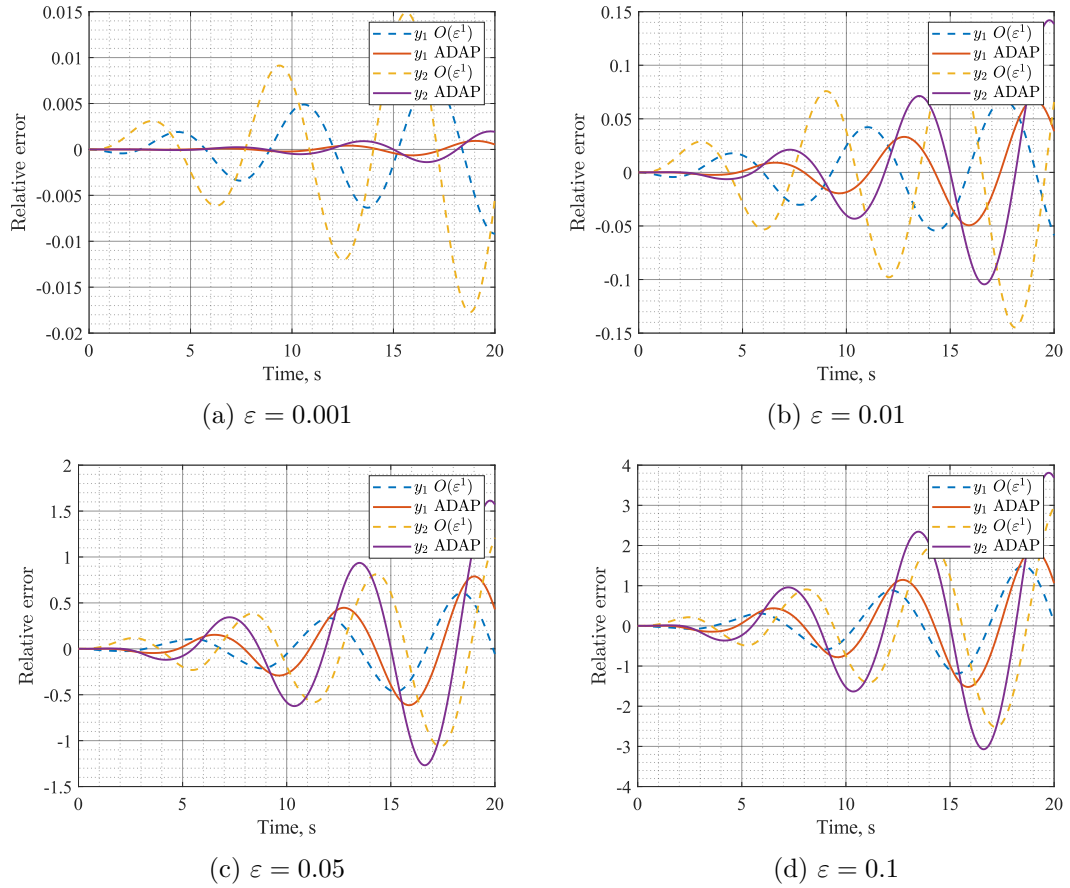


Figure 2.9: The relative error for the ADAP method compared to the Lindstedt-Poincaré technique to $O(\varepsilon)$ with varying values of the small parameter, ε , in Eq (2.1).

2.2 A system of coupled Van der Pol oscillators

Another widely studied oscillator model is the Van der Pol oscillator. The origin of this model was in observing current oscillations in vacuum tube circuits, but (like the Rayleigh oscillator) it has since been applied in a vast array of fields aside from electronics.

This oscillator model has been seen to be highly useful in modelling biological functions. Work by the original developers of the Van der Pol model (B Van der Pol and J Van der Mark) sees this model applied to create a representation of the human heart

[52]. This was achieved by analogous comparison to electrical circuits and saw a simple resistor-capacitor circuit constructed for the sinus, atrium, and ventricle individually and then connected with appropriate coupling mechanisms. The model requires some simplifications to be made, such as ignoring phenomena like flutter in the heart muscle. Building on this work, and that of others, Ryzhii and Ryzhii use differing oscillator models to describe different aspects of heart behaviour [53]. In particular, they use a variation of the classical Van der Pol model (referred to as the Banhoeffer-Van der Pol model) to model the rhythm-driving portions of the heart, “the sinoatrial and atrioventricular nodes, and the His-Purkinje system”. The authors also then used another oscillator type (namely Fitzhugh-Nagumo oscillators) to simulate more granular heart-muscle operation. This Fitzhugh-Nagumo model is reportedly a more general statement of the classical Van der Pol model and sees use across fields such as neuroscience [54].

In a mechanical engineering context, vortex-induced vibration is an area of considerable interest in which the Van der Pol model finds further use. Facchinetti, de Langre, and Biolley [55] consider the application of these oscillators to modelling the near-wake dynamics of individual fluid cells behind slender bluff bodies such as mooring lines, pipework, or other submerged cylindrical structures. Specifically, they design the coupling of these models to simulate the local coupling of adjacent fluid cells in the spanwise direction of the body. A modified Van der Pol model is used by Marra, Manini, and Bartoli after their critique of previous approaches’ inability to satisfactorily forecast the amplitude of resulting vibrations [56]. In their work, the authors focused specifically on the supporting structures of bridge decks.

In another field entirely, He and Cai have also considered the use of Van der Pol oscillators in encryption algorithms [57]. By introducing two variable parameters (to the damping and stiffness terms separately), the authors in this work were able to exploit the chaotic nature of the “Van der Pol with Variable Parameter” system to create an appropriately pseudo-random algorithm which they projected to have utility in encryption.

In essence, the Van der Pol oscillator has become a reliable model for a range of systems that exhibit self-sustaining oscillations with nonlinear damping characteristics. This is similar, in many ways to the Rayleigh oscillator considered in Section 2.1. However, as will be shown in the present section, there are notable differences in applying the ADAP methodology across these two problems.

2.2.1 Problem specification

A system of coupled Van der Pol oscillators is set up as follows.

$$\begin{Bmatrix} \ddot{y}_1(t) \\ \ddot{y}_2(t) \end{Bmatrix} + \varepsilon \begin{bmatrix} y_1(t)^2 - 1 & -1 \\ -1 & y_2(t)^2 - 1 \end{bmatrix} \begin{Bmatrix} \dot{y}_1(t) \\ \dot{y}_2(t) \end{Bmatrix} + \begin{Bmatrix} y_1(t) \\ y_2(t) \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \quad (2.42)$$

In this section, a modification of this system is later included to encompass a different set of coupled damping properties. However, in the present form of Eq (2.42), the system only differs from the Rayleigh oscillator system, Eq (2.1), in the diagonal terms in the damping matrix. Note that, as per the characterisation as a “Van der Pol-type” system, the diagonal damping terms in Eq (2.42) are functions of displacement, $P(\mathbf{Y}; \varepsilon)$; whereas, in Eq (2.1), these are functions of velocity, $P(\dot{\mathbf{Y}}; \varepsilon)$.

$$P(\mathbf{Y}; \varepsilon) = \varepsilon \begin{bmatrix} y_1^2 - 1 & -1 \\ -1 & y_2^2 - 1 \end{bmatrix} \quad (2.43)$$

2.2.2 Construction and solution of the uncoupled system

Maintaining the approach taken in the example of the coupled Rayleigh oscillators, a similar simplification of the system is adopted for the initial problem to be solved.

$$\begin{Bmatrix} \ddot{\phi}_1(t) \\ \ddot{\phi}_2(t) \end{Bmatrix} + \varepsilon \begin{bmatrix} \phi_1(t)^2 - 1 & 0 \\ 0 & \phi_2(t)^2 - 1 \end{bmatrix} \begin{Bmatrix} \dot{\phi}_1(t) \\ \dot{\phi}_2(t) \end{Bmatrix} + \begin{Bmatrix} \phi_1(t) \\ \phi_2(t) \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \quad (2.44)$$

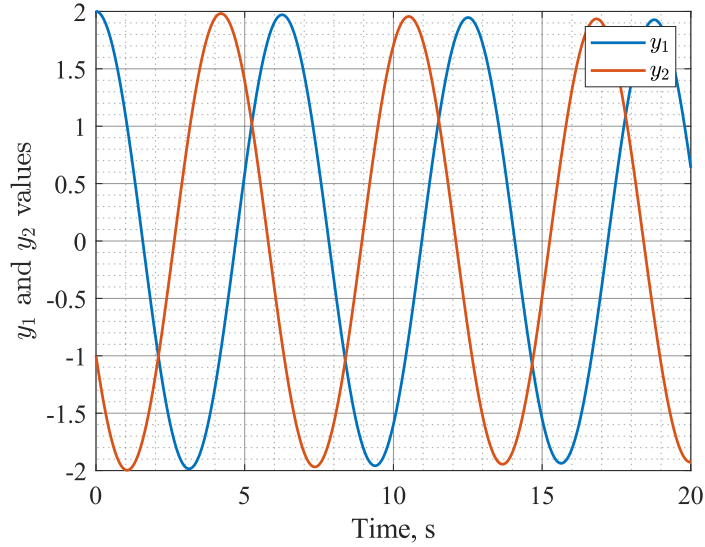


Figure 2.10: Reference numerical solutions produced by *Mathematica*'s *NDSolve* function for Eq (2.42) when $\varepsilon = 0.01$, $\mathbf{y}(t = 0) = [2, -1]^T$, and $\dot{\mathbf{y}}(t = 0) = [0, -\sqrt{3}]^T$.

Notice that, like the previous example, this first auxiliary problem could theoretically be used for variations of this system which include different forms of coupled damping terms as these are entirely left to the second auxiliary problem. One such example of this is examined later in this section. The application of the Lindstedt-Poincaré technique to obtain an asymptotic series approximation to the solution of Eq (2.42) is extremely similar to the method shown in Sec 2.1. The n^{th} -order asymptotic approximation can still be represented by Eq (2.5) with the stretched coordinate being defined by Eq (2.6). Due to the inclusion of the small parameter ε in $P(\mathbf{Y}; \varepsilon)$, the generating problem in this example is identical to that in the previous example, Eqs (2.7) and (2.8). It is in the subsequent determination of the $O(\varepsilon)$ problems where differences will be noted. Hence, these $O(\varepsilon)$ problems are shown as follows.

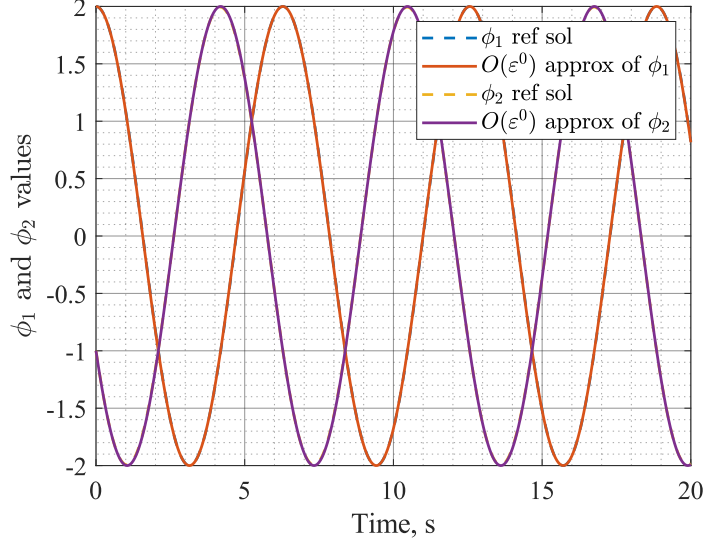


Figure 2.11: Numerical solutions for Eq (2.44) compared to the corresponding $O(1)$ solutions to the generating Lindstedt-Poincaré problems with $\varepsilon = 0.01$.

$$\begin{aligned}
 \omega_{1,0}^2 \phi_{1,1}''(\tau_1) + \phi_{1,1}(\tau_1) = & -\omega_{1,0} \left(b \cos \left(\frac{\tau_1}{\omega_{1,0}} \right) - \frac{a \sin \left(\frac{\tau_1}{\omega_{1,0}} \right)}{\omega_{1,0}} \right) \\
 & \cdot \left(a \cos \left(\frac{\tau_1}{\omega_{1,0}} \right) + b \omega_{1,0} \sin \left(\frac{\tau_1}{\omega_{1,0}} \right) \right)^2 \\
 & + \omega_{1,0} \left(b \cos \left(\frac{\tau_1}{\omega_{1,0}} \right) - \frac{a \sin \left(\frac{\tau_1}{\omega_{1,0}} \right)}{\omega_{1,0}} \right) \\
 & - 2\omega_{1,1}\omega_{1,0} \left(-\frac{a \cos \left(\frac{\tau_1}{\omega_{1,0}} \right)}{\omega_{1,0}^2} - \frac{b \sin \left(\frac{\tau_1}{\omega_{1,0}} \right)}{\omega_{1,0}} \right)
 \end{aligned} \tag{2.45}$$

Chapter 2. A novel approximation algorithm for the solution of nonlinear and coupled systems

$$\begin{aligned}
\omega_{2,0}^2 \phi_{2,1}''(\tau_2) + \phi_{2,1}(\tau_2) = & -\omega_{2,0} \left(d \cos \left(\frac{\tau_2}{\omega_{2,0}} \right) - \frac{c \sin \left(\frac{\tau_2}{\omega_{2,0}} \right)}{\omega_{2,0}} \right) \\
& \cdot \left(c \cos \left(\frac{\tau_2}{\omega_{2,0}} \right) + d \omega_{2,0} \sin \left(\frac{\tau}{\omega_0} \right) \right)^2 \\
& + \omega_{2,0} \left(d \cos \left(\frac{\tau_2}{\omega_{2,0}} \right) - \frac{c \sin \left(\frac{\tau_2}{\omega_{2,0}} \right)}{\omega_{2,0}} \right) \\
& - 2\omega_{2,1}\omega_{2,0} \left(-\frac{c \cos \left(\frac{\tau_2}{\omega_0} \right)}{\omega_0^2} - \frac{d \sin \left(\frac{\tau_2}{\omega_{2,0}} \right)}{\omega_{2,0}} \right)
\end{aligned} \tag{2.46}$$

Solving, by determining their complementary functions and particular integrals via the method of variation of parameters, these ODEs leads to the following expressions for $\phi_{1,1}(\tau_1)$

$$\begin{aligned}
\phi_{1,1}(\tau_1) = & \frac{1}{32\omega_{1,0}^2} \left(-a\omega_{1,0}^2 (a^2 - 3b^2\omega_{1,0}^2) \sin \left(\frac{3\tau_1}{\omega_{1,0}} \right) \right. \\
& + \left(\omega_{1,0}^2 (-4(a^2 - 4)b\tau_1 + a(7a^2 - 16) + 32b\omega_{1,1}) \right. \\
& \quad \left. - b^2\omega_{1,0}^4(5a + 4b\tau_1) + 32a\tau_1\omega_{1,1} \right) \sin \left(\frac{\tau_1}{\omega_{1,0}} \right) \\
& + b\omega_{1,0}^3 (3a^2 - b^2\omega_{1,0}^2) \cos \left(\frac{3\tau_1}{\omega_{1,0}} \right) \\
& + \omega_{1,0} \left(-4\tau_1 (a^3 - 4a + 8b\omega_{1,1}) \right. \\
& \quad \left. - ab\omega_{1,0}^2(3a + 4b\tau_1) + b^3\omega_{1,0}^4 \right) \cos \left(\frac{\tau_1}{\omega_{1,0}} \right) \Big)
\end{aligned} \tag{2.47}$$

Chapter 2. A novel approximation algorithm for the solution of nonlinear and coupled systems

and $\phi_{2,1}(\tau_2)$.

$$\begin{aligned}
\phi_{2,1}(\tau_2) = & \frac{1}{32\omega_{2,0}^2} \left(-c\omega_{2,0}^2 (c^2 - 3d^2\omega_{2,0}^2) \sin\left(\frac{3\tau_2}{\omega_{2,0}}\right) \right. \\
& + \left(\omega_{2,0}^2 (-4(c^2 - 4)d\tau_2 + c(7c^2 - 16) + 32d\omega_{2,1}) \right. \\
& \quad \left. \left. - d^2\omega_{2,0}^4(5c + 4d\tau_2) + 32c\tau_2\omega_{2,1} \right) \sin\left(\frac{\tau_2}{\omega_{2,0}}\right) \right. \\
& + d\omega_{2,0}^3 (3c^2 - d^2\omega_{2,0}^2) \cos\left(\frac{3\tau_2}{\omega_{2,0}}\right) \\
& + \omega_{2,0} \left(-4\tau_2 (c^3 - 4c + 8d\omega_{2,1}) \right. \\
& \quad \left. \left. - cd\omega_{2,0}^2(3c + 4d\tau_2) + d^3\omega_{2,0}^4 \right) \cos\left(\frac{\tau_2}{\omega_{2,0}}\right) \right)
\end{aligned} \tag{2.48}$$

These solutions can then be combined, with the solutions to the generating problems, into the asymptotic form of Eq (2.5). The resulting solutions are plotted in Fig 2.12 for $\phi_1(0) = 2$, $\phi_2(0) = -1$, $\dot{\phi}_1(0) = 0$, and $\dot{\phi}_2(0) = -\sqrt{3}$. These are then compared to one another in Fig 2.13 to highlight the benefit of pursuing the $O(\varepsilon)$ approximation over the $O(1)$ approximation.

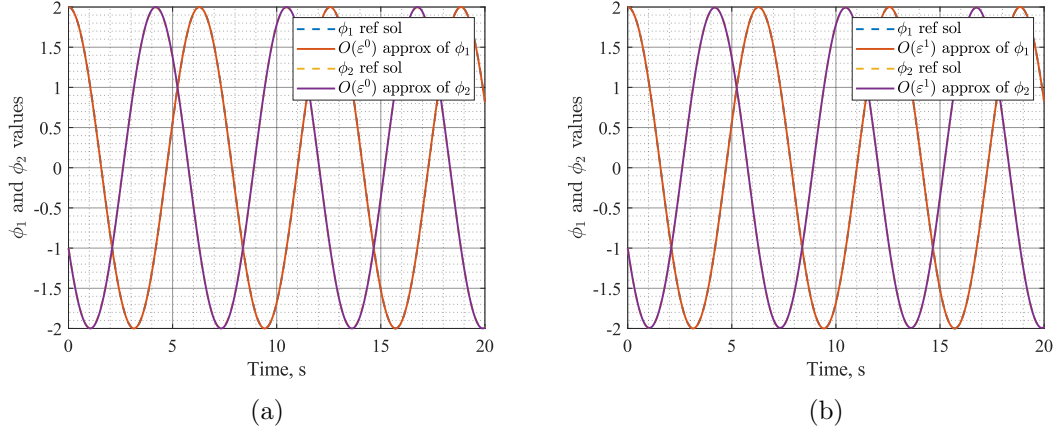


Figure 2.12: The $O(1)$ and $O(\varepsilon)$ approximation to Eq (2.44) shown in (a) and (b) respectively with $\varepsilon = 0.01$.

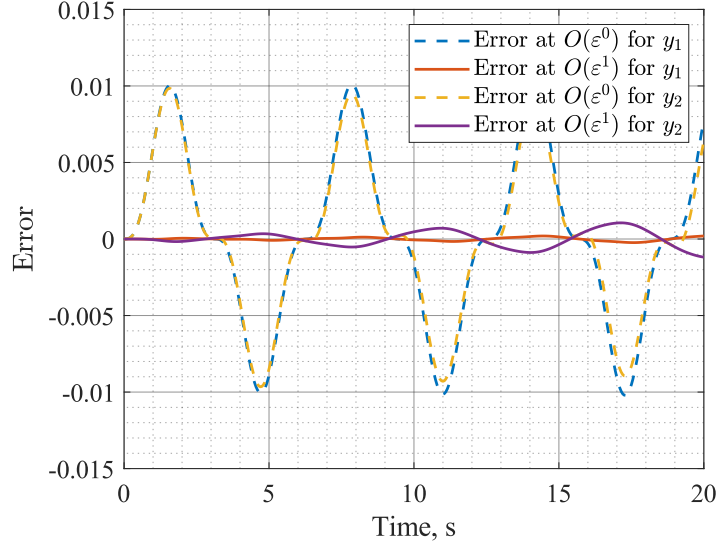


Figure 2.13: The errors accumulated by the $O(1)$ and $O(\varepsilon)$ approximations in solving Eq (2.44) via the Lindstedt-Poincaré technique.

2.2.3 Approximation and solution of the fully-coupled system

Similarly to Sec 2.1, the first explored means of reconstituting the previously neglected dynamics into the second auxiliary problem is via the damping terms.

$$\mathbf{R} \approx -P(\mathbf{Y}; \varepsilon) \dot{\mathbf{Y}}$$

$$R = -\varepsilon \begin{bmatrix} \phi_1^2(t) - 1 & -1 \\ -1 & \phi_2^2(t) - 1 \end{bmatrix} \begin{Bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \end{Bmatrix} = \begin{Bmatrix} r_1 \\ r_2 \end{Bmatrix} \quad (2.49)$$

Once again, the resulting second auxiliary problem is a system of two uncoupled simple harmonic oscillators with an excitation $\mathbf{R}$ applied to the system (as the nonhomogeneous term).

Solving this second auxiliary problem demands the same process, as was depicted in Sec 2.1, of finding the complementary function via the forms of Eqs (2.17) and (2.18) before constructing the particular integral. Since the solution bases are the same, the Wronskian is the same for this example, $W = 1$ (again indicating the linear independence of the solutions). The difference can be found when computing Eqs (2.20)

Chapter 2. A novel approximation algorithm for the solution of nonlinear and coupled systems

and (2.21) since r_1 and r_2 are different for this example.

$$\begin{aligned} \frac{\psi_2 r_1}{W} = & -\varepsilon \sin(t) \left((b \cos(t) - a \sin(t)) ((a \cos(t) + b \sin(t))^2 - 1) \right. \\ & \left. + c \sin(t) - d \cos(t) \right) \end{aligned} \quad (2.50)$$

$$\begin{aligned} \int \frac{\psi_2 r_1}{W} = & -\frac{1}{32} \varepsilon \left(-4t (a^3 + a(b^2 - 4) - 4c) + 4 \cos(2t) ((a^2 + 2)b - b^3 + 2d) \right. \\ & + a(a^2 - 3b^2) \sin(4t) + b(b^2 - 3a^2) \cos(4t) \\ & \left. - 8 \sin(2t) (-ab^2 + a + c) \right) \end{aligned} \quad (2.51)$$

$$\begin{aligned} \frac{\psi_1 r_1}{W} = & -\varepsilon \cos(t) \left((b \cos(t) - a \sin(t)) ((a \cos(t) + b \sin(t))^2 - 1) \right. \\ & \left. + c \sin(t) - d \cos(t) \right) \end{aligned} \quad (2.52)$$

$$\begin{aligned} \int \frac{\psi_1 r_1}{W} = & -\frac{1}{32} \varepsilon \left(4 \cos(2t) (a^3 - a(b^2 + 2) - 2c) + 4t ((a^2 - 4)b + b^3 - 4d) \right. \\ & - b(b^2 - 3a^2) \sin(4t) + a(a^2 - 3b^2) \cos(4t) \\ & \left. - 8 \sin(2t) (-a^2 b + b + d) \right) \end{aligned} \quad (2.53)$$

$$\begin{aligned} \frac{\psi_2 r_2}{W} = & -\varepsilon \sin(t) \left((d \cos(t) - c \sin(t)) ((c \cos(t) + d \sin(t))^2 - 1) \right. \\ & \left. + a \sin(t) - b \cos(t) \right) \end{aligned} \quad (2.54)$$

$$\begin{aligned} \int \frac{\psi_2 r_2}{W} = & -\frac{1}{32} \varepsilon \left(4t (4a - c(c^2 + d^2 - 4)) - 8 \sin(2t) (a - cd^2 + c) \right. \\ & + 4 \cos(2t) (2b + d(c^2 - d^2 + 2)) \\ & \left. + c(c^2 - 3d^2) \sin(4t) + d(d^2 - 3c^2) \cos(4t) \right) \end{aligned} \quad (2.55)$$

Chapter 2. A novel approximation algorithm for the solution of nonlinear and coupled systems

$$\begin{aligned} \frac{\psi_1 r_2}{W} = & -\varepsilon \cos(t) \left((d \cos(t) - c \sin(t)) ((c \cos(t) + d \sin(t))^2 - 1) \right. \\ & \left. + a \sin(t) - b \cos(t) \right) \end{aligned} \quad (2.56)$$

$$\begin{aligned} \int \frac{\psi_1 r_2}{W} = & -\frac{1}{32} \varepsilon \left(4t (d (c^2 + d^2 - 4) - 4b) - 8 \sin(2t) (b - c^2 d + d) \right. \\ & - 4 \cos(2t) (2a + c (-c^2 + d^2 + 2)) \\ & \left. - d (d^2 - 3c^2) \sin(4t) + c (c^2 - 3d^2) \cos(4t) \right) \end{aligned} \quad (2.57)$$

As before, these particular integrals can be added to the complementary functions to find the general solutions. The generalised expression to determine the coefficients arising from the complementary function can also be found by Eq (2.36). In this case, however, the resulting expressions for the K -terms are

$$\begin{pmatrix} K_{1,1} \\ K_{1,2} \\ K_{2,1} \\ K_{2,2} \end{pmatrix} = \frac{1}{32} \begin{pmatrix} -a^2 b \varepsilon + 32a + 3b^3 \varepsilon - 8b \varepsilon - 8d \varepsilon \\ 5a^3 \varepsilon - 7ab^2 \varepsilon - 8a \varepsilon + 32b - 8c \varepsilon \\ -8b \varepsilon - c^2 d \varepsilon + 32c + 3d^3 \varepsilon - 8d \varepsilon \\ -8a \varepsilon + 5c^3 \varepsilon - 7cd^2 \varepsilon - 8c \varepsilon + 32d \end{pmatrix}. \quad (2.58)$$

The resulting ADAP approximate analytical solution of Eq (2.42) consists of the two following expressions when subject to the previously-stated conditions and parameters.

$$\begin{aligned} y_1 \approx & (2 - 0.005t) \cos(t) + \sin(t) (-0.00866025t + 0.0025 \sin^2(t) + 0.0125) \\ & - 0.0075 \sin(t) \cos^2(t) \end{aligned} \quad (2.59)$$

$$\begin{aligned} y_2 \approx & (0.01t + 0.0025 \sin(4t) - 1) \cos(t) + (-0.0025 \cos(4t) - 1.72955) \sin(t) \\ & - 0.02 \sin(t) \cos^2(t) \end{aligned} \quad (2.60)$$

2.2.4 Results and discussion

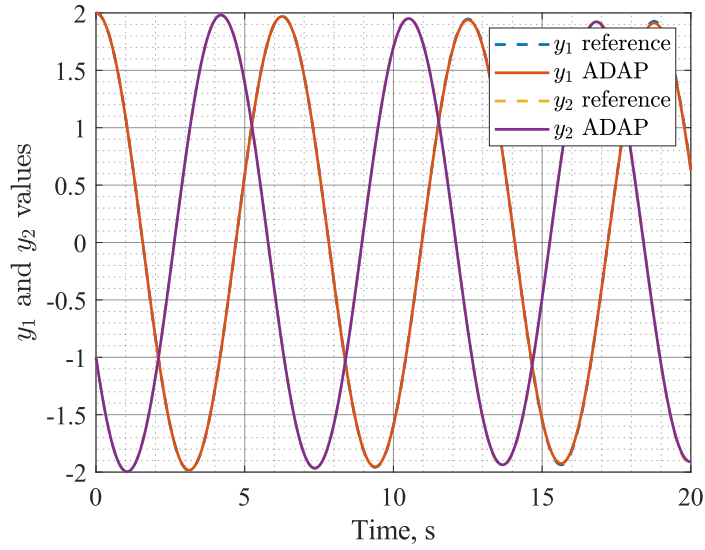


Figure 2.14: The approximate analytical solutions for Eq (2.42) achieved by the ADAP method compared to the reference numerical solutions.

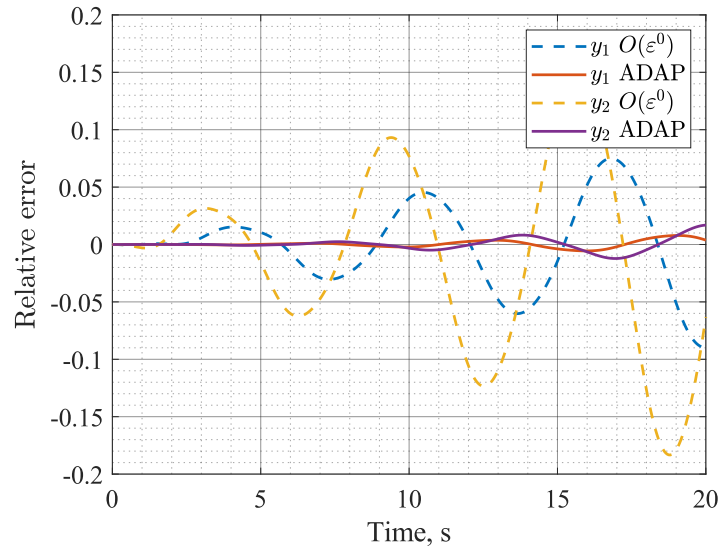


Figure 2.15: The relative errors produced in the approximate analytical solutions for Eq (2.42), when $\varepsilon = 0.01$, achieved by the ADAP method and the $O(\varepsilon)$ asymptotic series approximation achieved by the Lindstedt-Poincaré technique.

Many of the graphs included in this section focus on the (relative) errors in the generated

solutions since, as demonstrated by Fig 2.14, the visual distinction between the numerical reference solutions and the ADAP solutions are near-imperceptible. These errors are first explored in Figs 2.15 and 2.16, where the $O(1)$ and $O(\varepsilon)$ Lindstedt-Poincaré approximations are illustrated, respectively, versus the ADAP approximations. In both cases, the ADAP solutions remain advantageous.

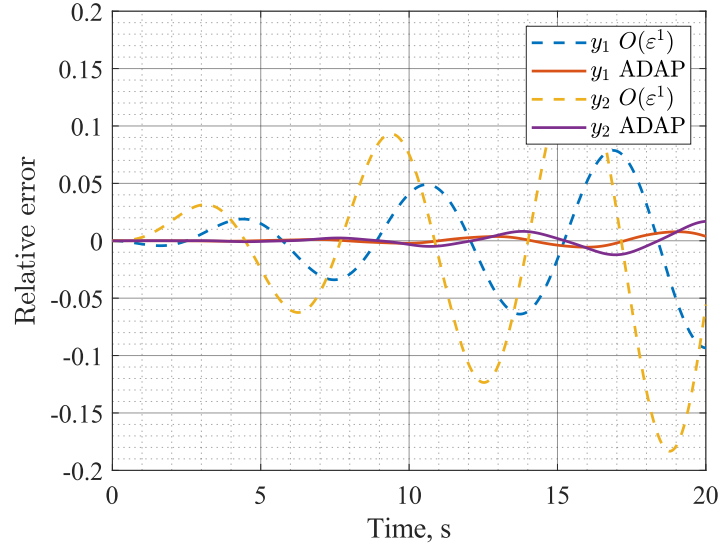


Figure 2.16: The relative errors produced in the approximate analytical solutions for Eq (2.42), when $\varepsilon = 0.01$, achieved by the ADAP method and the $O(\varepsilon)$ asymptotic series approximation achieved by the Lindstedt-Poincaré technique.

Chapter 2. A novel approximation algorithm for the solution of nonlinear and coupled systems

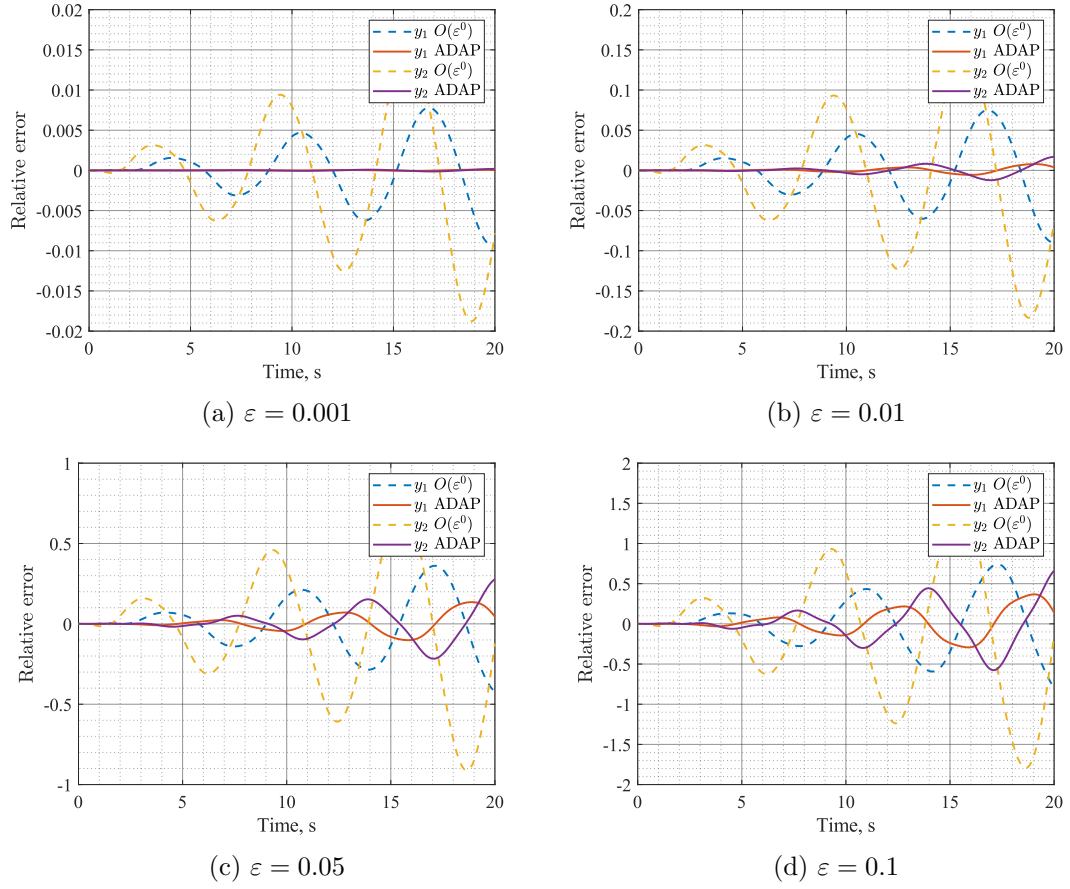


Figure 2.17: The relative error for the ADAP method compared to the Lindstedt-Poincaré technique to $O(1)$ with varying values of the small parameter, ε , in Eq (2.42).

Chapter 2. A novel approximation algorithm for the solution of nonlinear and coupled systems

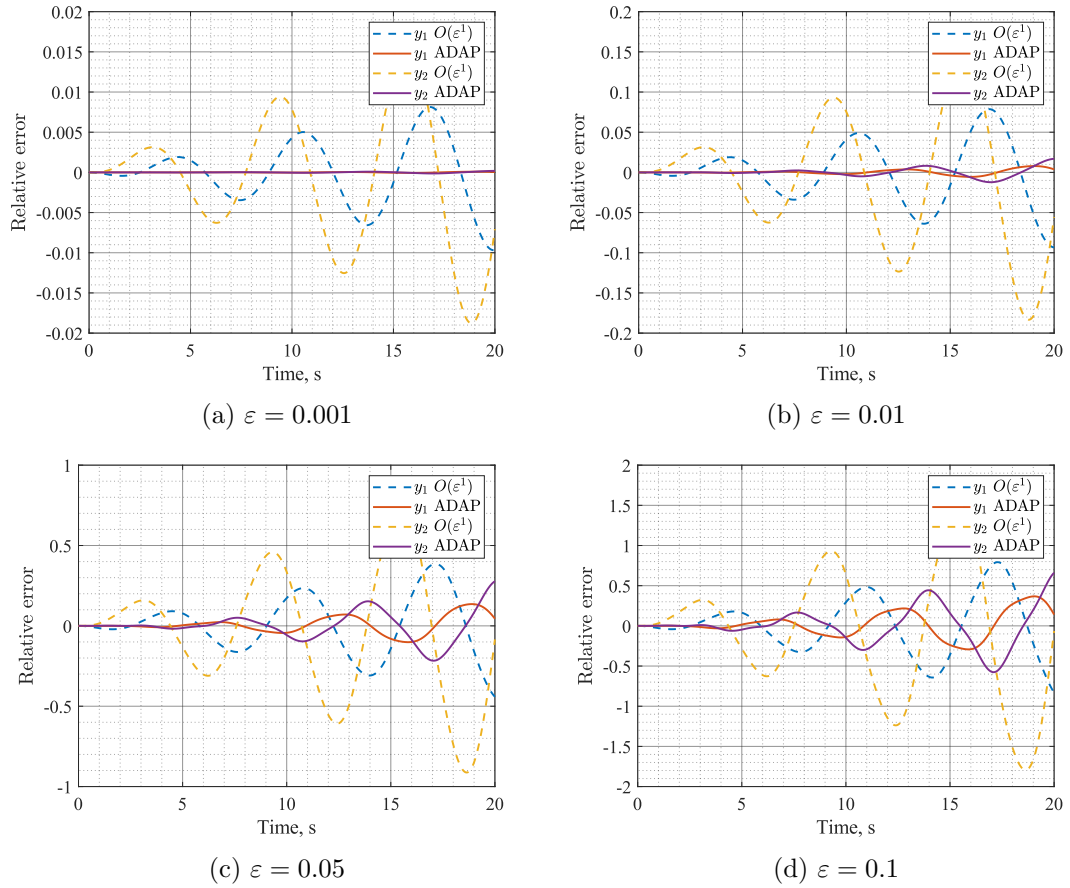


Figure 2.18: The relative error for the ADAP method compared to the Lindstedt-Poincaré technique to $O(\varepsilon)$ with varying values of the small parameter, ε , in Eq (2.42).

Suppose, instead of Eq (2.42), a slightly different coupling dynamic were to be introduced.

$$\begin{Bmatrix} \ddot{y}_1(t) \\ \ddot{y}_2(t) \end{Bmatrix} + \varepsilon \begin{bmatrix} y_1(t)^2 - 1 & y_1(t)^2 \\ y_2(t)^2 & y_2(t)^2 - 1 \end{bmatrix} \begin{Bmatrix} \dot{y}_1(t) \\ \dot{y}_2(t) \end{Bmatrix} + \begin{Bmatrix} y_1(t) \\ y_2(t) \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \quad (2.61)$$

Eq (2.61) differs from Eq (2.42) in the “off-diagonal” damping terms, which are now also functions of displacement rather than constant. This extension from the previously studied system, Eq (2.42), introduces nonlinearities across the coupling effects in the oscillator system. Following the method detailed in this section, the only substantial difference will occur in the particular integrals when solving the second auxiliary

problem due to the excitations, r_1 and r_2 , now being given as follows.

$$\mathbf{R} = -\varepsilon \begin{bmatrix} \phi_1(t)^2 - 1 & \phi_1(t)^2 \\ \phi_2(t)^2 & \phi_2(t)^2 - 1 \end{bmatrix} \begin{Bmatrix} \dot{\phi}_1(t) \\ \dot{\phi}_2(t) \end{Bmatrix} = \begin{Bmatrix} r_1 \\ r_2 \end{Bmatrix} \quad (2.62)$$

By comparing Fig 2.17 with Fig 2.19, and Fig 2.18 with Fig 2.20, there is an observable impact made by the construction of the first auxiliary problem. Studying the specific constructions for this problem is discussed as an area of interest for further work in Chapter 7.

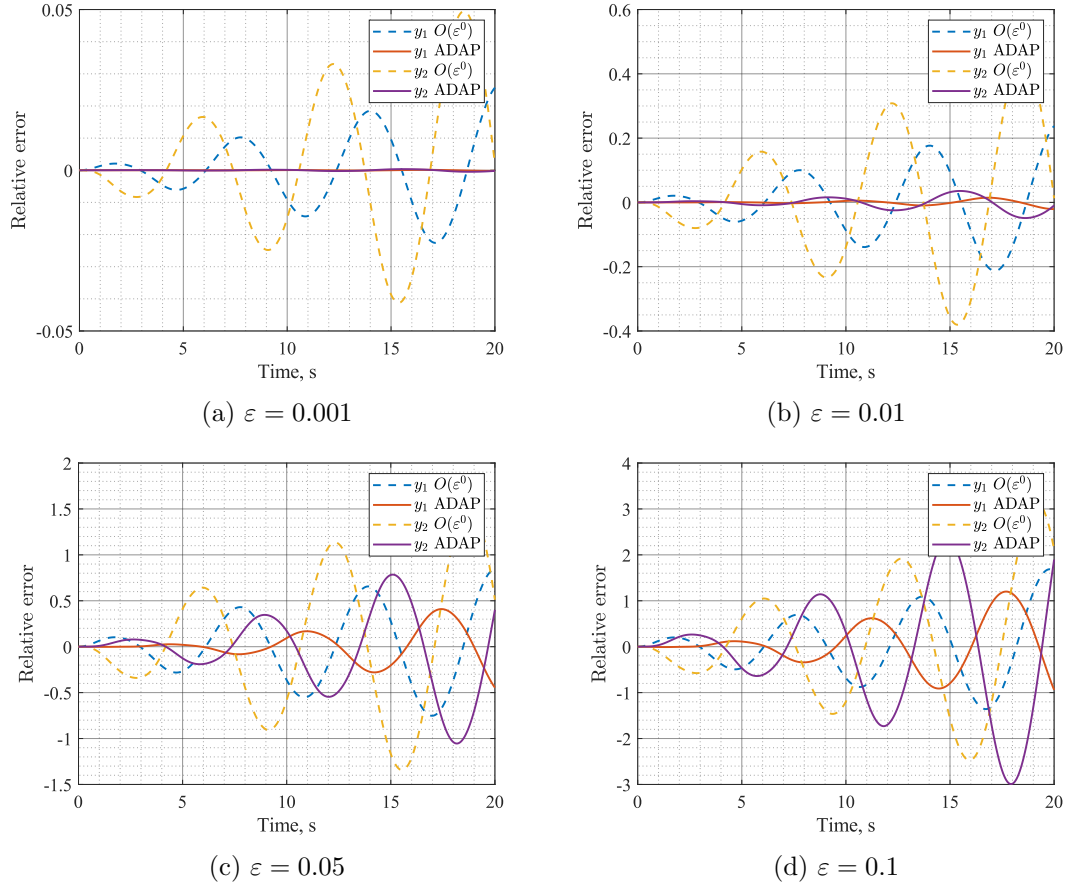


Figure 2.19: The relative error for the ADAP method compared to the Lindstedt-Poincaré technique to $O(1)$ with varying values of the small parameter, ε , in Eq (2.61).

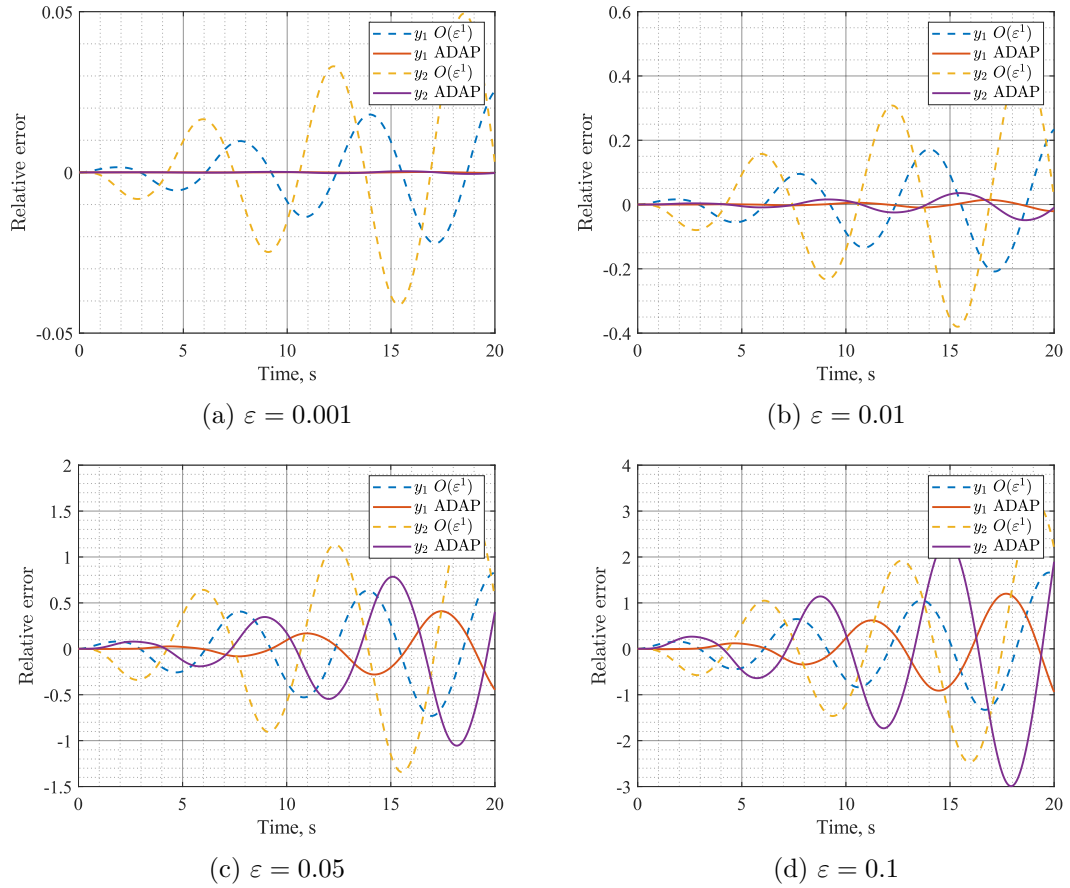


Figure 2.20: The relative error for the ADAP method compared to the Lindstedt-Poincaré technique to $O(\varepsilon)$ with varying values of the small parameter, ε , in Eq (2.61).

2.3 A note on the effect of increasing the small parameter

Importantly, in the examples of the Rayleigh and Van der Pol oscillator systems shown in the preceding sections, 2.1 and 2.2, the value of the small parameter ε was never raised beyond $\varepsilon = 0.1$. To some extent, this is deliberate to allow for a fair comparison between the ADAP methodology and the Lindstedt-Poincaré technique: while the range of capabilities of the former have not yet been fully explored, it is already known that the latter technique will not perform well when ε ceases to be considered as “small”. The implication of this limitation of the work in Sec 2.1 and 2.2 is that the conclusions they reach can only be confidently made for systems of **weakly** nonlinear,

coupled systems. This substantially shorter section examines the cases of $\varepsilon = 1$ for the same oscillator systems, Eqs (2.1), (2.42), and (2.61), as were considered in the preceding sections.

The fact that $\varepsilon = 1$ means that these systems are no longer weakly nonlinear, coupled systems, and thus the asymptotic approximations produced by the Lindstedt-Poincaré technique are not expected to be useful approximations. These asymptotic approximations are nonetheless included in the graphs for this section, if only to give a reference to the reader of the extent to which the accuracy of the ADAP method is also degraded in the context of increasing nonlinearities.

Beginning with the system of coupled Rayleigh oscillators, Eq (2.1), the reference numerical solutions are depicted in Fig 2.21.

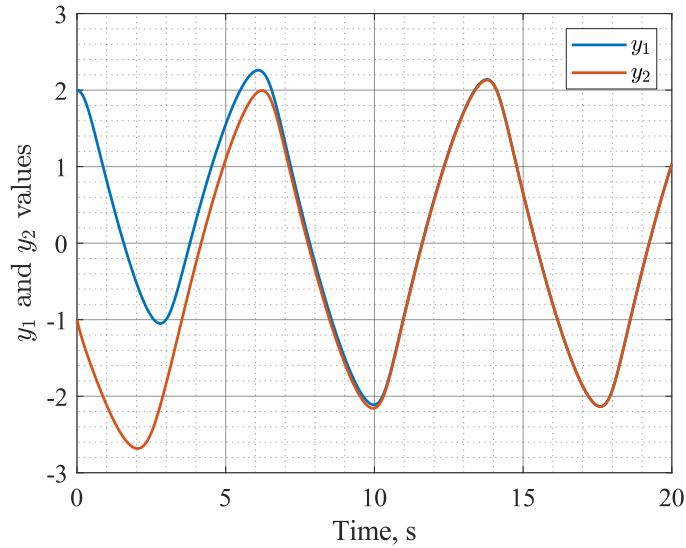


Figure 2.21: Reference numerical solutions for Eq (2.1), when $\varepsilon = 1$, found via the *NDSolve* function in *Mathematica*.

The relative errors in the approximations yielded by the Lindstedt-Poincaré and ADAP methodologies are highlighted in Fig 2.22. Note that, although the increased relative error in the $O(\varepsilon)$ approximation appears counterintuitively higher than that of the $O(1)$ approximation, this is the construction of the asymptotic approximation no

longer retains its expected validity since ε is no longer “small”.

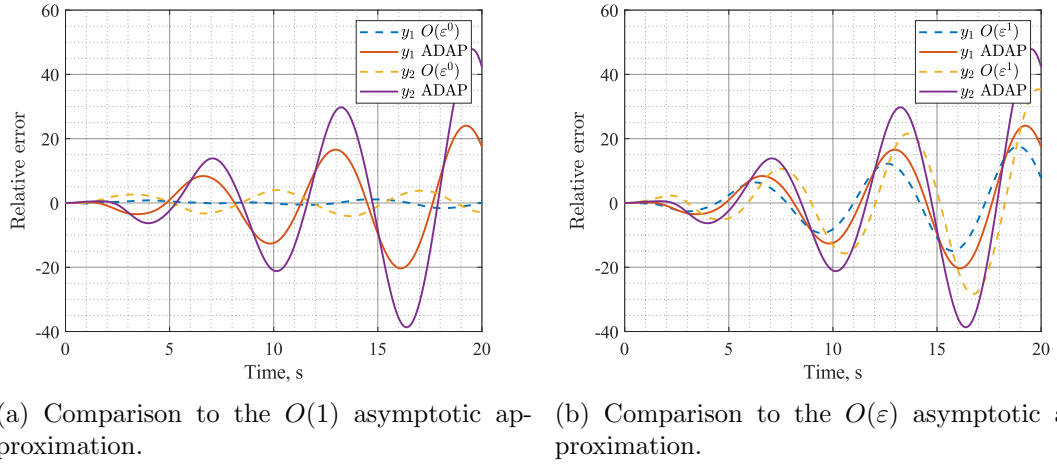


Figure 2.22: Relative error in the Lindstedt-Poincaré technique and ADAP methodology when applied to Eq (2.1) with $\varepsilon = 1$.

Leaving aside the specific relative errors observed in Fig 2.22, the most notable observation in this context is that the ADAP-derived approximation is at least as poor as the $O(\varepsilon)$ asymptotic approximation here.

Similar observations can be made for the first Van der Pol system, Eq (2.42), across Figs 2.23 and 2.24.

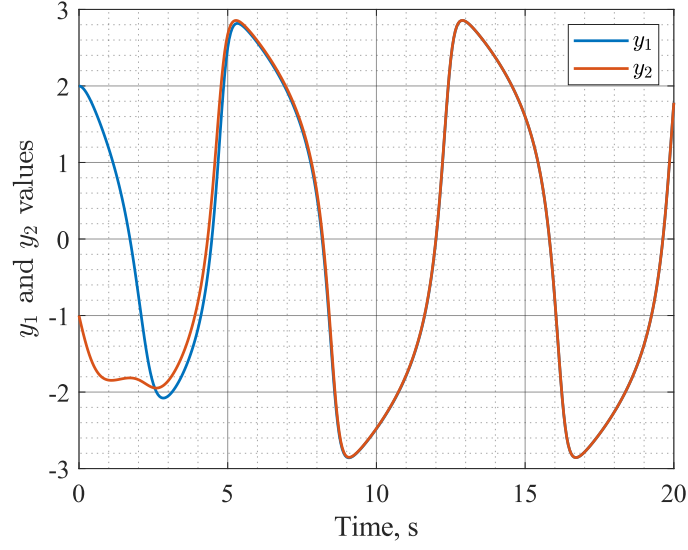
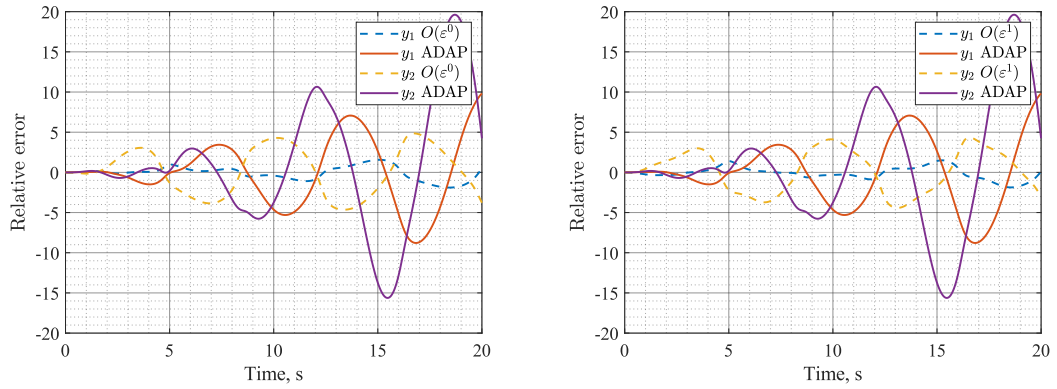


Figure 2.23: Reference numerical solutions for Eq (2.42), when $\varepsilon = 1$, found via the *ode45* function in *MATLAB*.



(a) Comparison to the $O(1)$ asymptotic approximation.

(b) Comparison to the $O(\varepsilon)$ asymptotic approximation.

Figure 2.24: Relative error in the Lindstedt-Poincaré technique and ADAP methodology when applied to Eq (2.42) with $\varepsilon = 1$.

And, similar observations can also be made for the second Van der Pol system, Eq (2.61), across Figs 2.25 and 2.26.

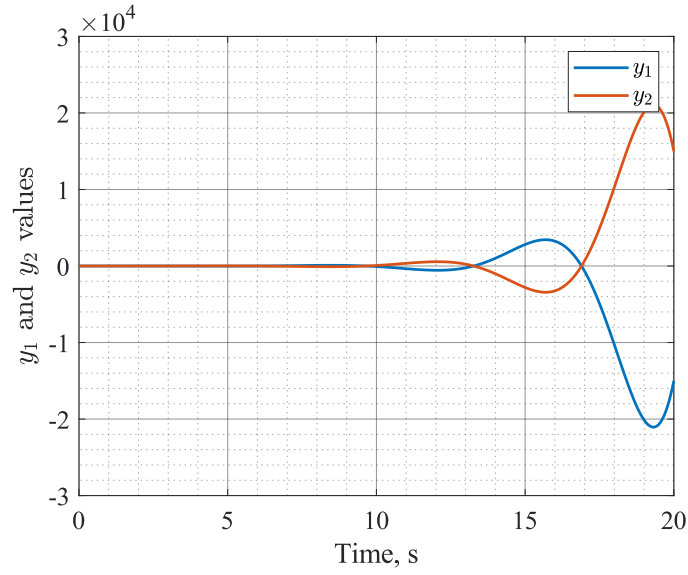
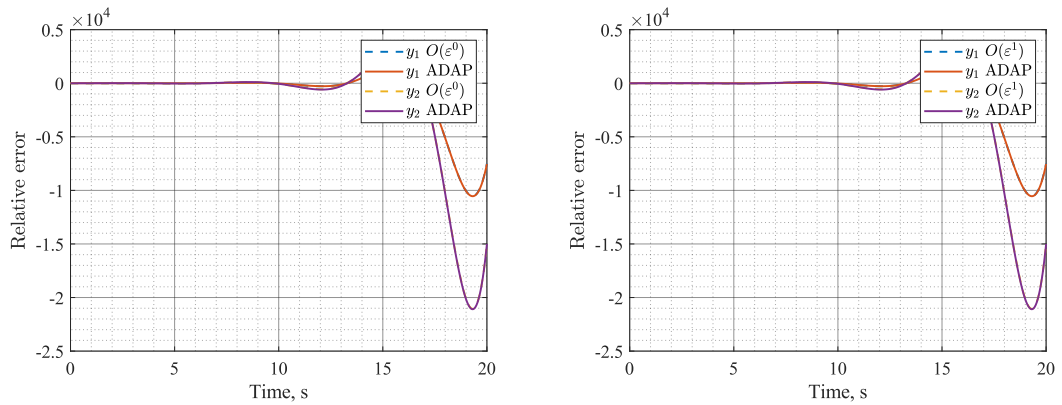


Figure 2.25: Reference numerical solutions for Eq (2.61), when $\varepsilon = 1$, found via the *NDSolve* function in *Mathematica*.



(a) Comparison to the $O(1)$ asymptotic approximation.

(b) Comparison to the $O(\varepsilon)$ asymptotic approximation.

Figure 2.26: Relative error in the Lindstedt-Poincaré technique and ADAP methodology when applied to Eq (2.61) with $\varepsilon = 1$.

Given these comparisons show the unsuitability of the ADAP methodology for approximating systems involving “strong” nonlinear or coupling effects, the subsequent sections in this chapter outline and demonstrate a standardised approach - with a more clearly defined ruleset and more robust rationale - which retains its utility in the face of stronger nonlinearities and coupling effects.

2.4 Revision and standardisation of the algorithm

In this final section of Chapter 2, the ADAP algorithm is standardised and detailed outwith the context of a particular problem.

Suppose a general n^{th} -order system of m differential equations can be written as

$$M(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{R}(t) \quad (2.63)$$

where $M = M(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ is an $m \times m$ matrix; $\mathbf{C} = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ and $\mathbf{R} = \mathbf{R}(t)$ are $m \times 1$ vectors; and $\mathbf{q}$ is a $m \times 1$ vector of generalised coordinates that vary across an independent variable t . The proposed method of Approximation via Dual Auxiliary Problems (herein referred to as the ADAP methodology) can be used to obtain approximate analytical solutions to this system or, alternatively, allow for the use of linear control strategies in required applications.

The first step of the ADAP methodology is to formulate the first of the dual auxiliary problems. This method requires that the first auxiliary problem be such that a solution is analytically tractable. This problem is denoted

$$\tilde{M}(\Phi, \dot{\Phi}, \ddot{\Phi})\ddot{\Phi} + \tilde{\mathbf{C}}(\Phi, \dot{\Phi}) = \mathbf{0} \quad (2.64)$$

where Φ is the $m \times 1$ vector of generalised coordinates which are approximations of the coordinates in $\mathbf{q}$; and, $\tilde{M}$ and $\tilde{\mathbf{C}}$ are the corresponding approximations of M and $\mathbf{C}$. Note that the simplifications and approximations used in changing M and $\mathbf{C}$ to $\tilde{M}$ and $\tilde{\mathbf{C}}$ respectively may render the extent to which Φ approximates $\mathbf{q}$ as very limited. Examples of this are contained subsequently in this chapter to illustrate the effect of this. Such necessary changes to M and $\mathbf{C}$ may remove coupling effects or nonlinearities - this is dependent on what changes the practitioner sees as necessary to render this first auxiliary problem as analytically solvable.

Chapter 2. A novel approximation algorithm for the solution of nonlinear and coupled systems

The goal of the second auxiliary problem then to use these approximations Φ to approximately solve the original (“target”) system in the generalised coordinates $\mathbf{q}$. By defining the matrices $\hat{M} = M - \tilde{M}$ and $\hat{\mathbf{C}} = \mathbf{C} - \tilde{\mathbf{C}}$ as the differences between the target problem and the first auxiliary problem, the second auxiliary problem can be written as follows.

$$\left(\hat{M}(\Phi, \dot{\Phi}, \ddot{\Phi})\ddot{\Phi} + \tilde{M}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\ddot{\mathbf{q}} \right) + \left(\hat{\mathbf{C}}(\Phi, \dot{\Phi}) + \tilde{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) \right) = \mathbf{R}(t) \quad (2.65)$$

$$\tilde{M}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\ddot{\mathbf{q}} + \tilde{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{R}(t) - \left(\hat{M}(\Phi, \dot{\Phi}, \ddot{\Phi})\ddot{\Phi} + \hat{\mathbf{C}}(\Phi, \dot{\Phi}) \right) \quad (2.66)$$

Notice that this reformulation of the target system hinges on how accurate the approximation $\Phi \approx \mathbf{q}$ is. In general terms, where the differences $\hat{M}$ and $\hat{\mathbf{C}}$ are small, then the effects of differences between Φ and $\mathbf{q}$ are also small. In the instance that $\Phi = \mathbf{q}$, then the second auxiliary problem is, through its definition, identical to the target problem since $M = \hat{M} + \tilde{M}$ and $\mathbf{C} = \hat{\mathbf{C}} + \tilde{\mathbf{C}}$. In the formulation of Eq (2.66), the terms on the right-hand side are known once the first auxiliary problem, Eq (2.64), has been solved. Thus Eq (2.66) can be interpreted as being equivalent to Eq (2.64) with a non-homogeneous “correction” derived from Eq (2.64).

As will be explored in Chapter 5, there are notable apparent similarities between the ADAP methodology and feedback linearisation. However, this generalised overview should highlight that the derivation of this approach is altogether different and has the added utility of being able to approximately solve systems of ODEs rather only permitting the use of linear control strategies. With the outline of the ADAP method detailed in this chapter, Chapter 3 presents a case-study of its application. After this, the ADAP method presented here is used to develop a generalised architecture which improves the effectiveness of linear model-predictive control for controlling nonlinear systems in Chapters 4-6.

Chapter 3

Case study Part 1: dynamics of the *FutureForge* manipulator

Inclusion of previous work

This chapter includes content from previous works by the author. Specifically, these earlier works are *The FutureForge manipulator and an approximate analytical solution algorithm for its nonlinear dynamics*, which was published in the proceedings of the *10th European Nonlinear Dynamics Conference* [58]; and, *A novel approximation method for the solution of weakly nonlinear coupled systems*, which was published and peer-reviewed in *Nonlinear Dynamics* [43]. These earlier works by D Johnston et al. build on the dynamics modelled by MP Cartmell et al. [59] to analytically solve the linearised system [58] and apply the ADAP method to this problem [43].

This chapter sees the application of the ADAP methodology, developed in Chapter 2, to a novel state-of-the-art system known as the *FutureForge* manipulator. Commissioned by the Advanced Forming Research Centre (AFRC) in Glasgow (Scotland), the *FutureForge* is a state-of-the-art hot working platform. It comprises multiple furnaces, a 2000-tonne hydraulic press, a manipulator arm, and a control room. The case study presented in this chapter focuses on the manipulator itself.

The manipulator’s intended purpose is to handle metallic workpieces in: transferring them to and from the furnaces and the hydraulic press; and, throughout manoeuvres within the press itself. The manipulator is highly versatile and flexible within the *FutureForge* platform. Hydraulic actuators are used to manoeuvre the end effector vertically and horizontally within a large 2-dimensional workspace, and electric motors are used to translate the entire manipulator along a rail on the ground and also to rotate its orientation. As a result, while the simple 2-dimensional workspace may sound modest by comparison to other state-of-the-art manipulators, this added flexibility ensures that it is more than sufficient for the application intended.

This manipulator design is unique in the field of industrial manipulators which are discussed in greater detail in Chapter 4, where this context is used to discuss control strategies in this field. At first glance, it appears to be a serial system since the end-effector’s position is governed by the positioning of three - apparently sequential - parallelogram linkages. Definitionally, serial manipulators are open kinematic chains of links, and the motion of their end-effectors is most commonly actuated by revolute or prismatic actuators located at the joints between links [60]. For these serial configurations, many powerful techniques have been developed to the end of solving the dynamic and kinematic problems posed. Such methods include Meldrum, Rodriguez, and Franklin’s recursive approach to matrix inversion [61] and Saha’s approach to decomposition of the inertia matrix [62]. Furthermore, several control approaches are also based on this type of configuration. These include Gazit and Widrow’s “back propagation through links” method [63]; Ouyang and Zhang’s virtual velocity-based method [64]; and, Talebpour and Namvar’s application of adaptive control to serial manipulators in the context of satellite testing [65]. However, this manipulator may not accurately be regarded as serial. Instead, the end-effector’s position is driven by two actuators at its base. Therefore, the *FutureForge* manipulator should be thought of as a hybrid between a serial and a parallel manipulator. To be clear what is meant by this: the end-effector’s position in the planar workspace, is the result of a series of parallelogram configurations, these individually being closed-chain mechanisms; and,

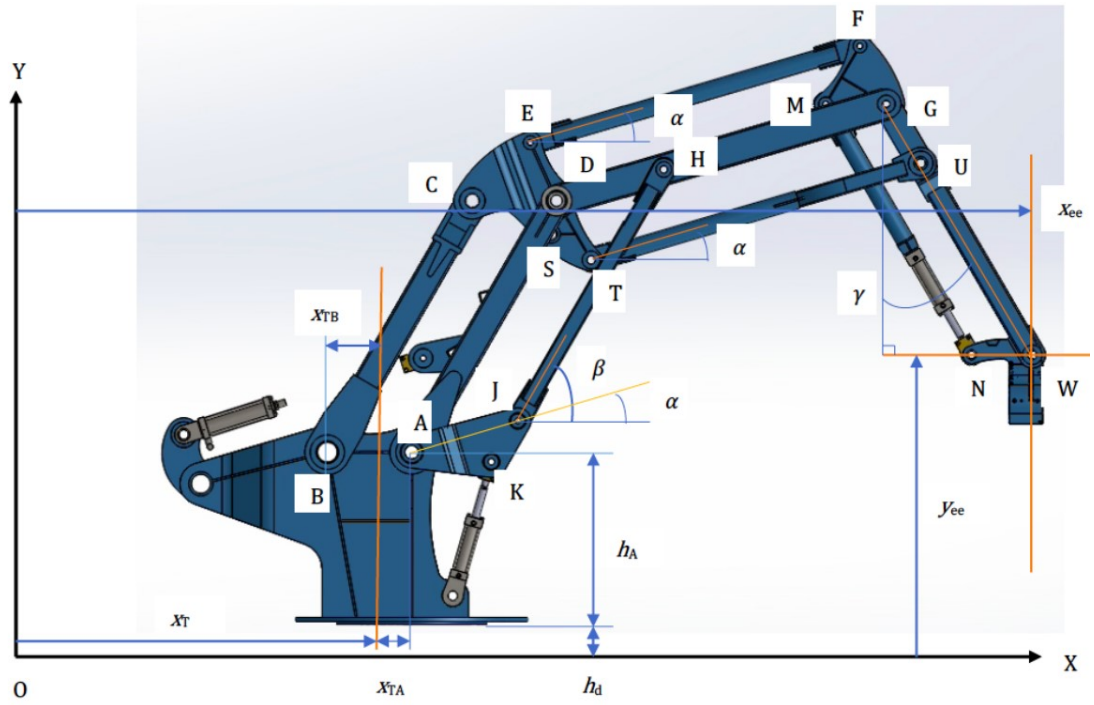


Figure 3.1: A schematic diagram of the *FutureForge* manipulator first produced in [59].

the configurations of each parallelogram are dictated by the parallelograms preceding them. In this way, the manipulator benefits from having a large workspace relative to its size and footprint [66]. Additionally, the parallelogram structure of the linkages allows it to carry heavier workpieces with greater accuracy [67]. The trade-off in realising these benefits is that the resulting system is significantly more complex to manufacture and model analytically.

3.1 Mathematical modelling of the manipulator dynamics

In 2021, Cartmell et al. [59] discussed how the *FutureForge* manipulator was intended to operate and derived the underlying dynamic equations describing this motion. The model that resulted is expressed as a system of two governing equations: one for the actuation of the hydraulic cylinders and pistons connecting the base to point K (driving the end-effector through vertical manoeuvres) as seen in Fig 3.1; and, one for the actuation of the hydraulic mechanism connecting the base to link AD (driving the

end-effector through horizontal manoeuvres). The generalised coordinates for these equations are the angles controlled by driving these two sets of actuators: $\alpha(t)$ for the vertical actuation angle, and $\beta(t)$ for the horizontal actuation angle. Thus, when the “vertical actuation problem” is referred to, the ODE being referred to is the one derived from the Euler-Lagrange equation

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\alpha}} \right) - \frac{\partial L}{\partial \alpha} = Q_v \quad (3.1)$$

where Q_v is the generalised force applied through this pair of cylinders at K. Similarly, the “horizontal actuation problem” refers to the ODE resulting from

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\beta}} \right) - \frac{\partial L}{\partial \beta} = Q_h \quad (3.2)$$

in which Q_h is the generalised force applied through the cylinder connected to AD.

Without repeating the lengthy derivation process undertaken by Cartmell et al. [59], the two equations of motion can be written as follows. Note that the constant coefficients (the a^* and b^* terms) are denoted with an “*” since the values previously derived for these coefficients are not retained in the new definitions of a and b terms included later in this chapter.

$$(a_1^* + a_2^* \cos(2\alpha)) \ddot{\alpha} + (a_3^* + a_4^* \sin(2\alpha) \dot{\alpha}) \dot{\alpha} + a_5^* \cos(\alpha) + a_6^* \sin(\alpha) = Q_v \quad (3.3)$$

$$\begin{aligned} (b_1^* + b_2^* \sin(2\beta) + b_3^* \cos(2\beta)) \ddot{\beta} + (b_4^* + b_5^* \cos(2\beta) \dot{\beta} + b_6^* \sin(2\beta) \dot{\beta}) \dot{\beta} \\ + b_7^* \cos(\beta) + b_8^* \sin(\beta) = Q_h \end{aligned} \quad (3.4)$$

3.1.1 Revision and updates to the *FutureForge* manipulator dynamics

Since the original work was published in [59] (Cartmell et al.), Eqs (3.3) and (3.4) were re-derived, resulting in Eqs (3.37) and (3.38), in light of changes to the manipulator design. These were minor adjustments made to account for the revised components’

centres of mass which had previously been taken as simply halfway along their lengths. The updated information for this was collected from the CAD files provided by the manufacturer of the manipulator, Clansman Dynamics Ltd. To ensure these were derived consistently with the new geometry, these results were assessed through comparison with a simulation in *Simscape* (which uses the information provided in the CAD files to construct a virtual model of the system). This present section therefore concerns the derivation of expressions for the Cartesian coordinates of the centres of mass for each component in the *FutureForge* manipulator. The common origin, O , used to define the position of each component is as depicted in Fig 3.1. Throughout this section, the notation used is consistent with the notation used by Cartmell et al. in the original derivation of the model [59]. The Cartesian coordinates of a centre of mass are denoted (x_C, y_C) with this being further specified to a specific link with further subscript notation. For example, the centre of mass of link AD has coordinates (x_{CAD}, y_{CAD}) .

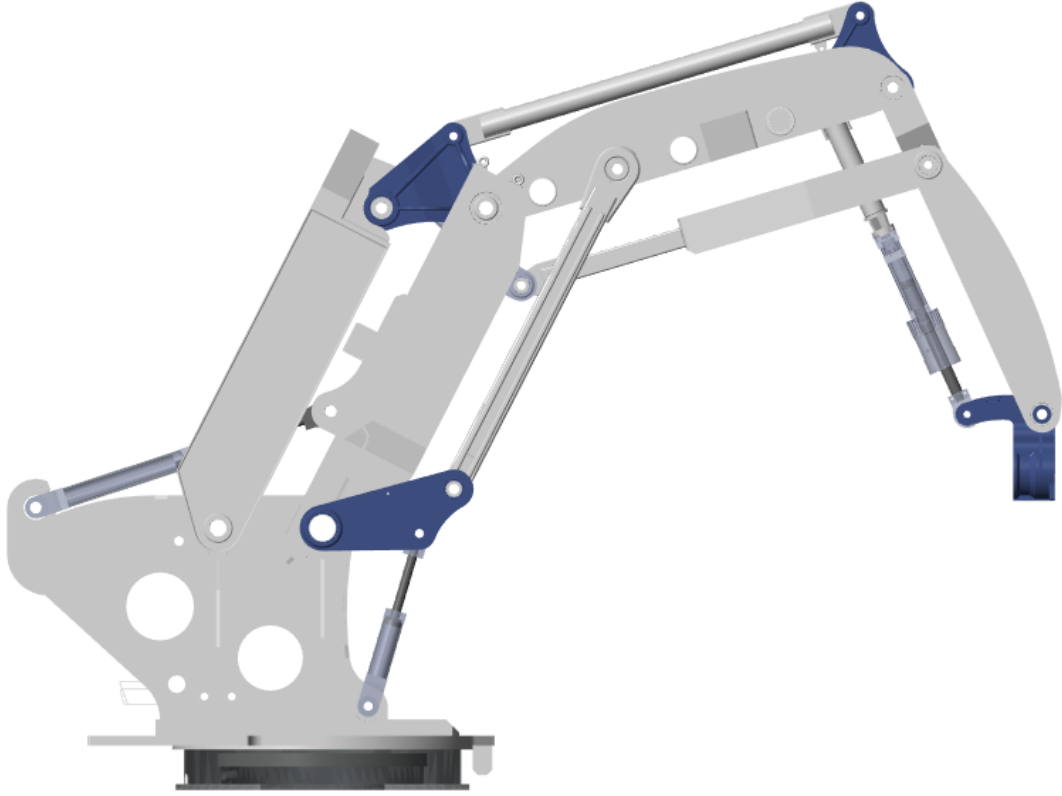


Figure 3.2: A depiction of the updated geometry of the *FutureForge* manipulator.

Before discussing the kinematics of this system in detail, it is important to ensure a solid grasp of the intended kinematic behaviour of each component and how this contributes to the wider manipulator behaviour. To convey, in writing, the intricate kinematic behaviours resulting in the manipulator's operation is a challenge, so the discussion will begin by only considering so-called “vertical actuation” - via changing values of α - before considering “horizontal actuation” - via changing values of β distinctly. Suppose α is decreasing from its value depicted in Fig 3.1: the immediately obvious consequence of this is the clockwise rotation of the crank AJK about its rotational joint at A. By consequence, this lowers the vertical strut JH which remains parallel to the vertical link, BC, and vertical boom, AD, throughout this motion. These three links, BC, AD, and JH - together with the connections across BA, CD, AJ, and DH - form the first pair of parallelograms in the manipulator, ABCD and ADHJ. Note

that, due to the rotation of the crank AJK, the pair of lines BA and AJ (and therefore CD and DH) are not necessarily parallel throughout the machine's operation. Under this scenario, with JH being lowered vertically but retaining a constant orientation of β radians to the horizontal as shown in Fig 3.1, the horizontal boom, DG, rotates clockwise about its rotational joint at D. Since DH and DG are collinear lines on the same rigid body, the angle through which the horizontal boom rotates is identical to the crank AJK. Due to the rotational joints at E and F, on the rear rocker CDE and front rocker FGM respectively, the upper horizontal link EF is constrained to mimic the horizontal boom's rotation, thus remaining parallel. The lines DT and GU remain necessarily in parallel - although the operation of the push rocker DST will be discussed more fully momentarily. As a result, the lower horizontal link TU also remains in parallel with the EF and DG, thus forming the second pair of parallelograms, DEFG and DGUT. The final part of the kinematic chain from base to end-effector is one final parallelogram, NMGW. While β remains constant, the lines DT and GU remain in their orientation shown in Fig 3.1. By consequence, when G is lowered by the clockwise rotation of the horizontal boom DG, the angle γ remains constant. Since the rear rocker CDE and front rocker FGM do not rotate, the final parallelogram NMGW remains in the configuration shown in Fig 3.1 and is lowered vertically.

Suppose, as another example, that α is held constant and β is increased towards and then past $\frac{\pi}{2}$ radians. Driven directly by the hydraulic actuator connected to the rear of the base superstructure, the vertical boom AD rotates anticlockwise about its rotational joint at A. With both AJ and AB remaining in a fixed orientation in this scenario, the vertical link BC and vertical strut JH move in parallel with AD through the same angle, β . The rear rocker CDE remains oriented with CD parallel with AB and is simply translated leftwards. Completing the parallelogram DEFG, the horizontal boom DG, the upper horizontal link EF, and the front rocker FGM also translate through the same displacement. A rather more complex mechanism is obscured by the rear rocker, vertical boom, and horizontal boom in Fig 3.1, but is isolated and made clearer in Fig 3.3. This mechanism consists of the vertical link BC, the push link ZS, and the

rear rocker DST. Pivotal to this is the offset connection between the push link and the vertical link: given that CD is a fixed length and remains horizontal, when the vertical link rotates about B , the push link ZS causes the push rocker DST to rotate about D . This is only possible because of the position of the rotational joint at Z which is further elaborated on in Fig 3.8. As will be discussed later in this section, the angle ϕ is necessary for defining the location of the centre of mass of the push link as illustrated in Fig 3.3. When β is increased, this mechanism “pulls” the lower horizontal link TU leftwards and leads to the consequent clockwise rotation of the forward vertical boom GW .

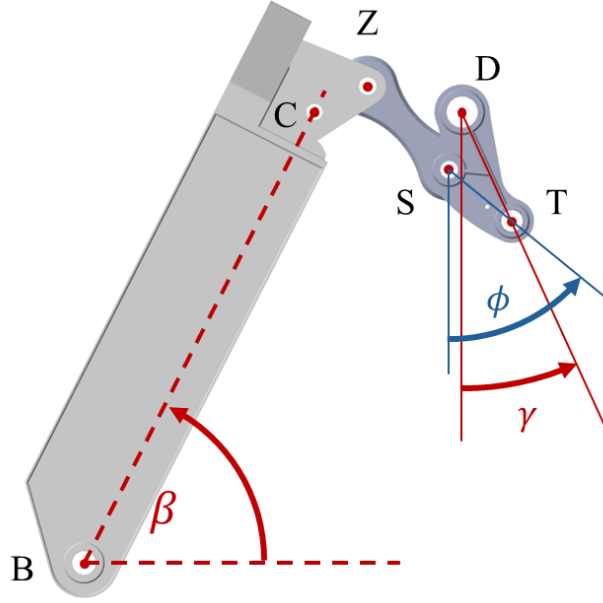


Figure 3.3: The assembly of components - consisting of the vertical link BC , the push link ZS , and the push rocker DST - which allows the end-effector to be extended horizontally when the generalised coordinate β is actuated.

The base of the manipulator is depicted in Fig 3.4. As is labelled in Fig 3.1, the horizontal distance from the arbitrary origin placement to a known point on the base is denoted x_T ; then, the horizontal distances from this (arbitrary) reference point to the joints A and B are denoted x_{TA} and x_{TB} respectively. Similarly, the vertical distance offset between the origin and the bottom of the base is h_A ; while, the distance from the bottom of the base to the joint A is h_A . The line AB is strictly horizontal;

therefore, additional notation to denote the height to joint B is not required. The use of these lengths (x_T , x_{TA} , x_{TB} , h_A , and h_d) is prevalent throughout this section. Since the hydraulic actuators are based on this component and drive the relative motion of further connected components, the behaviour of the base is not factored into this work. Neither is the behaviour of the hydraulic actuators connected to it.

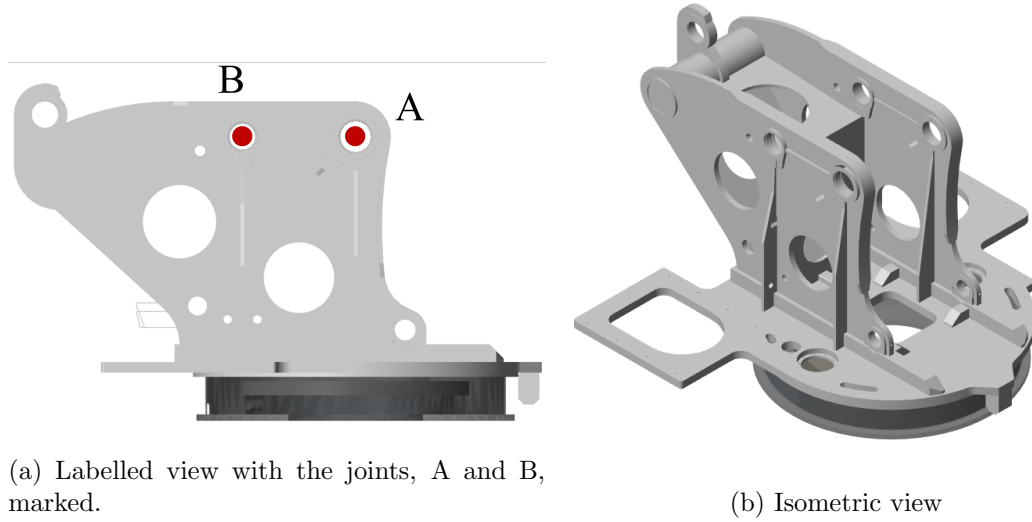


Figure 3.4: The base superstructure of the *FutureForge* manipulator.

Note, throughout this section, that the expressions achieved for describing the location of each component's centre of mass are based broadly on the preceding work by Cartmell et al. [59]. The work in this section is presented as thoroughly as necessary while avoiding duplication of this earlier work and ensuring that appropriate attribution is given.

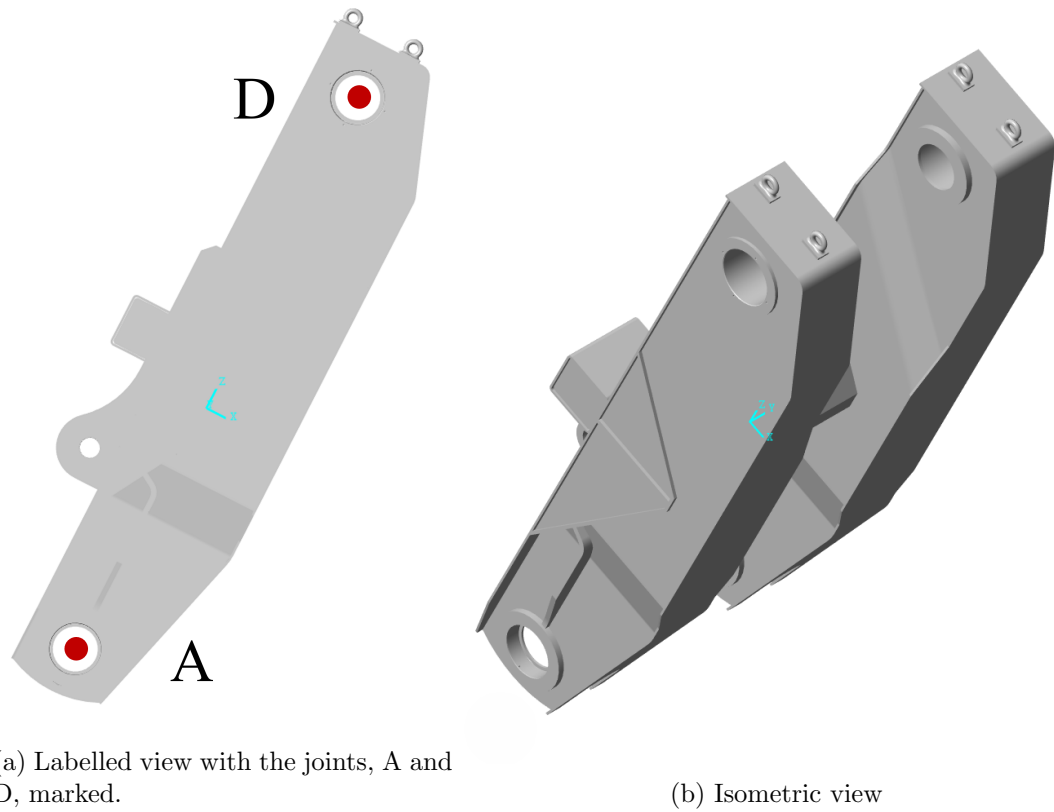
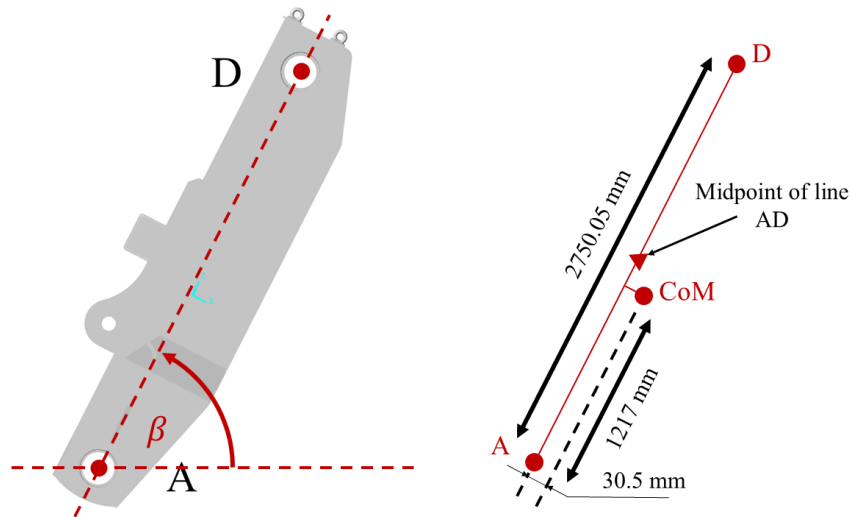


Figure 3.5: The vertical boom, link AD, in the *FutureForge* manipulator



The modified design of the vertical boom in Fig 3.5, link AD, sees the centre of mass offset from the midpoint between its two end joints at A and D as shown in Fig 3.6. This is highlighted in Fig 3.6. As a result, the Cartesian coordinates of this link's centre of mass are given as follows. The method taken to arrive at expressions such as this is, beginning at the origin of the coordinate system, to follow a kinematic chain which leads to one end of the desired link.

$$x_{CAD} = x_T + x_{TA} + 0.0305 \sin(\beta) + 1.217 \cos(\beta) \quad (3.5)$$

$$y_{CAD} = h_A + h_d - 0.0305 \cos(\beta) + 1.217 \sin(\beta) \quad (3.6)$$

As per the above description, the x_{CAD} expression begins with x_T and x_{TA} as the necessary terms to traverse from the origin to joint A. The remaining terms are based on data extracted from the Computer-Aided Design (CAD) files provided by Clansman Dynamics Ltd and describe the theoretical line between A and the vertical boom's centre of mass. The resulting equations align with a physically intuitive understanding of the component's geometry and retains the original design intention that this component is unaffected by changes in α .

Remaining parallel to the vertical boom is the vertical link, link BC, as shown in Figs 3.1, 3.7, and 3.8.

$$x_{BC} = x_T - x_{TB} - 0.0056 \sin(\beta) + 1.6419 \cos(\beta) \quad (3.7)$$

$$y_{BC} = h_A + h_d + 0.0056 \cos(\beta) + 1.6419 \sin(\beta) \quad (3.8)$$

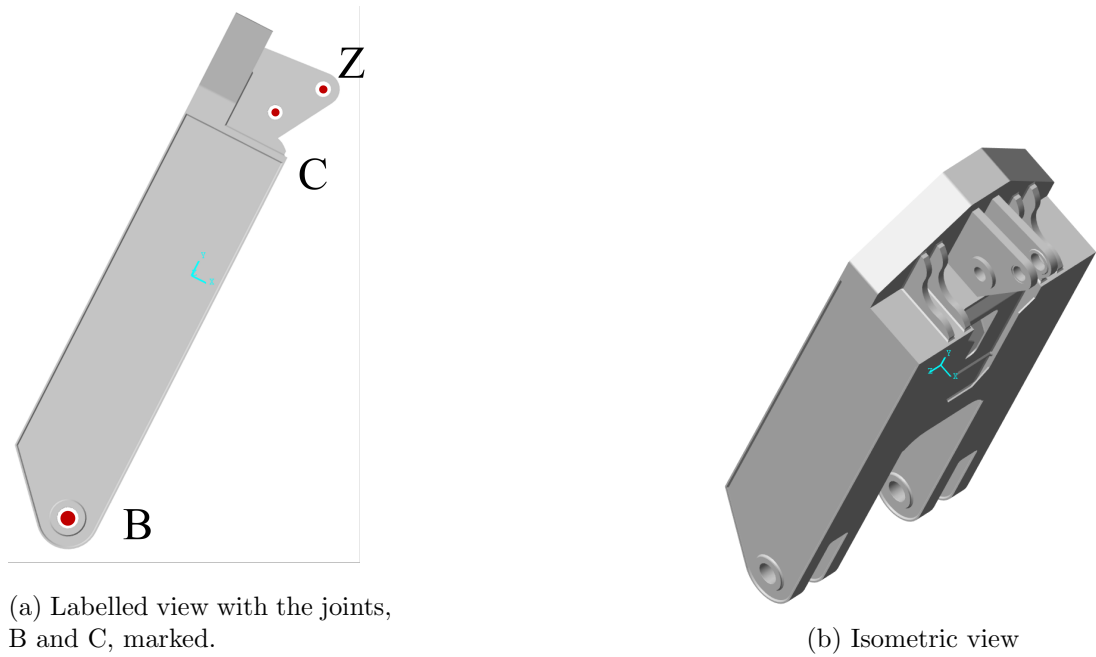


Figure 3.7: The vertical link, BC, in the *FutureForge* manipulator.

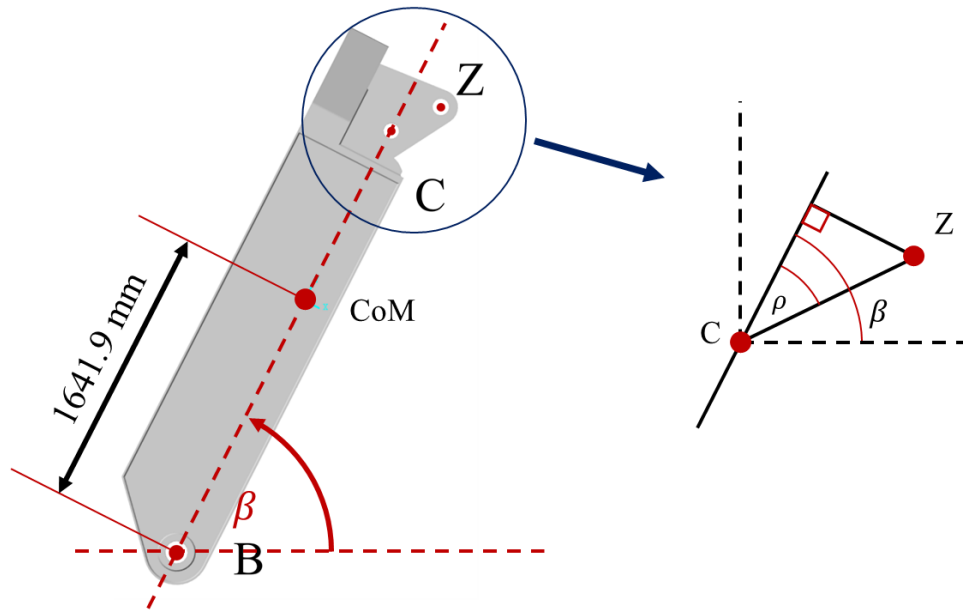
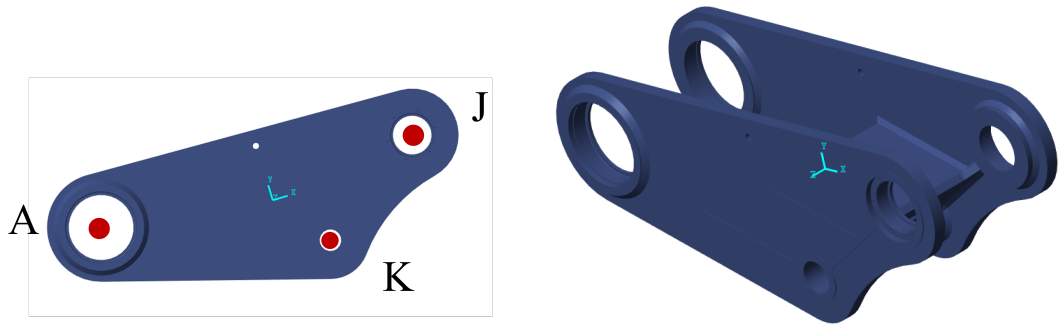


Figure 3.8: Detailed view of the dimensions and angles defining the location of the centre of mass (CoM) of the vertical link. Note that the detailed view of the angles marked at C is required for the later determination of the location of the CoM for the push link ZS.

The component designation AJK technically refers to two components, the right and left cranks. These cranks always move identically, driven by parallel hydraulic actuators, and are attached on either side of the base superstructure as shown in Fig 3.2. Their duplication has no impact on the kinematic discussion underway, although it does play a clear role when constructing the energy expressions necessary in the Euler-Lagrange formulation, where the mass is doubled necessarily to account for the component's duplication.



(a) Labelled view with the joints A, J, and K, marked.

(b) Isometric view

Figure 3.9: The crank components, AJK, in the *FutureForge* manipulator.

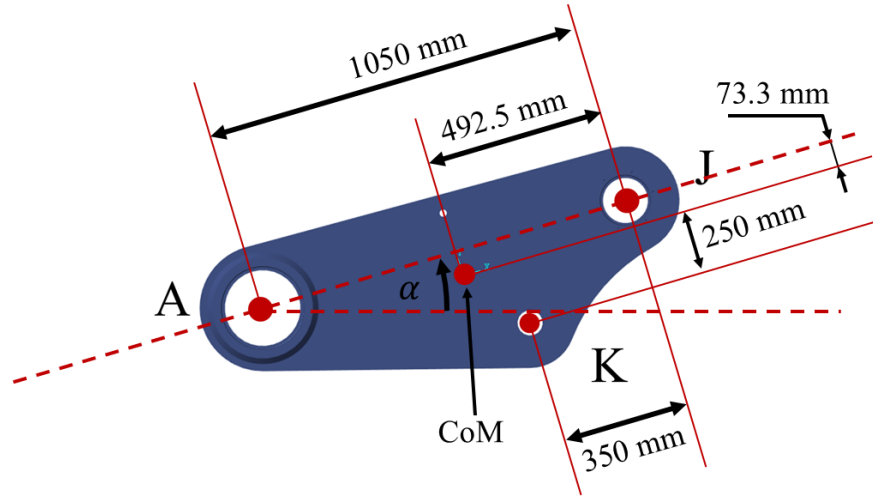


Figure 3.10: Detailed view of the dimensions and angles defining the location of the centre of mass (CoM) of the crank components.

The location of the centre of mass of this component in Figs 3.9 and 3.10, AJK, is given by the following expressions. Note that these expressions are strictly dependent on α , indicating (as can be intuited from Fig 3.1) that actuation of β through the hydraulic actuator connected to link AD has no bearing on this component's position.

$$x_{CAJK} = x_T + x_{TA} + 0.5575 \cos(\alpha) + 0.0733 \sin(\alpha) \quad (3.9)$$

$$y_{CAJK} = h_A + h_d + 0.5575 \sin(\alpha) - 0.0733 \cos(\alpha) \quad (3.10)$$

Another component that is duplicated in the full, three-dimensional model is the vertical strut, JH, as shown in Figs 3.11 and 3.12. These connect to the left and right cranks, at joint J, and to the horizontal boom, at joint H; and, they remain parallel to the vertical boom, AD, and vertical link, BC, throughout any manoeuvre.

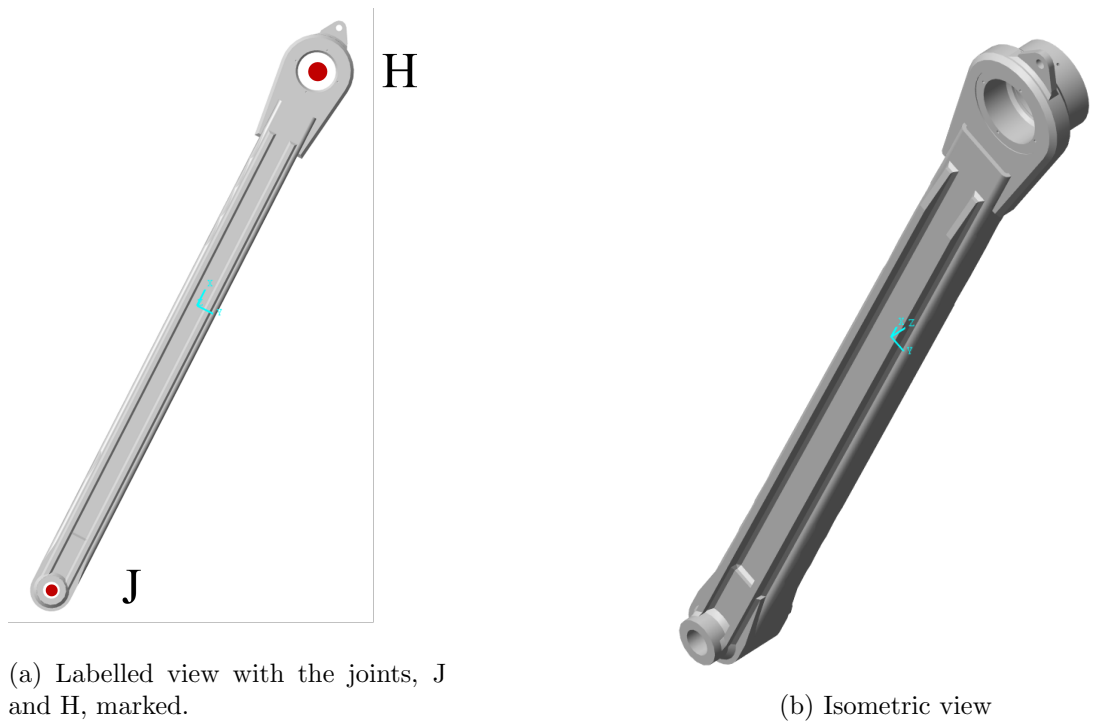


Figure 3.11: The vertical struts, JH, in the *FutureForge* manipulator.

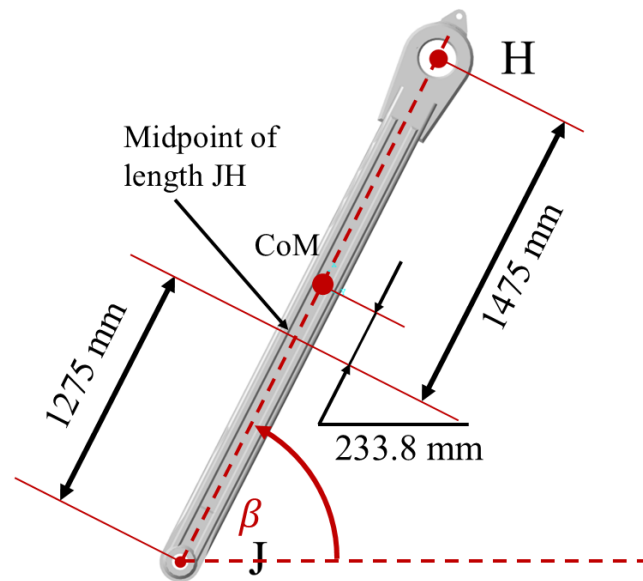


Figure 3.12: Detailed view of the dimensions and angles defining the location of the centre of mass (CoM) of the crank components.

The expressions for the location of link JH's centre of mass are given as follows.

$$x_{C_{JH}} = x_T + x_{TA} + l_{AJ} \cos(\alpha) + 1.5088 \cos(\beta) \quad (3.11)$$

$$y_{C_{JH}} = h_A + h_d + l_{AJ} \sin(\alpha) + 1.5088 \sin(\beta) \quad (3.12)$$

This is the first component considered, so far, to be actuated through both of the generalised angular coordinates.

Next, the rear rocker detailed in Figs 3.13 and 3.14, component CDE, is considered.

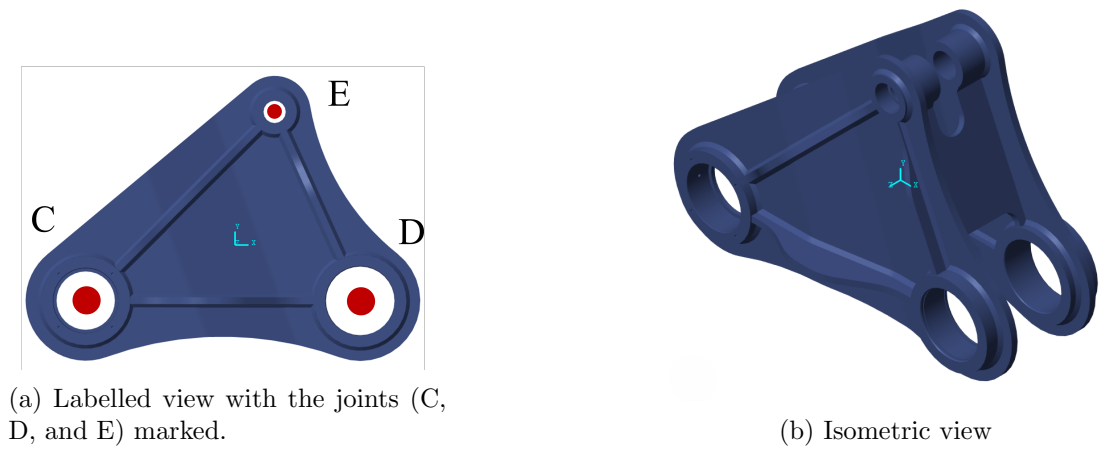


Figure 3.13: The rear rocker, CDE, in the *FutureForge* manipulator.

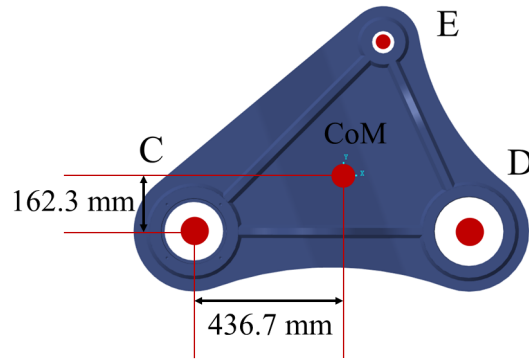


Figure 3.14: Detailed view of the dimensions defining the location of the centre of mass (CoM) of the rear rocker.

The expressions for the location of this component's centre of mass are as follows.

$$x_{C_{CDE}} = x_T - x_{TB} + l_{BC} \cos(\beta) + 0.4367 \quad (3.13)$$

$$y_{C_{CDE}} = h_A + h_d + l_{BC} \sin(\beta) + 0.1623 \quad (3.14)$$

Two notable features distinguish these two expressions from those previous. Firstly, in the expression for $x_{C_{CDE}}$, note that the kinematic chain being considered for this component moves from the origin to the arbitrary marker on the base superstructure, through a distance x_T , before then moving leftwards to point B, resulting in the subtraction of x_{TB} from the expression. The link BC is then treated as the next component of the kinematic chain. Notably, although this is its first occurrence in this work, it will become apparent that the choice of kinematic chain is often a matter of convenience based on available information. Due to the parallelogram ABCD, it would equally be possible to define the location of the centre of mass of CDE via a kinematic chain passing through A instead of B. The second and fourth terms in the present expression would necessarily change to account for this, and l_{BC} would be replaced with l_{AD} (although these two parameters are equal). Secondly, again by virtue of the parallelogram ABCD, the orientation of CDE is constant and independent of α or β . This results in the inclusion of the constant values, 0.4367 and 0.1623, in the two expressions noted and demonstrates the independence of the component's orientation from the generalised coordinates.

Having largely concluded the pair of vertical parallelograms, ABCD and ADHJ, the horizontal pair of parallelograms, DEFG and TDGU, will now be considered. The term “largely” is used here as these two pairs of important geometric structures are interrelated. To properly conclude the parallelogram ADHJ, for example, the kinematics of link DG (on which the joint H is located) must first be defined. The geometry depicted in Figs 3.15 and 3.16 shows the horizontal boom of the structure.

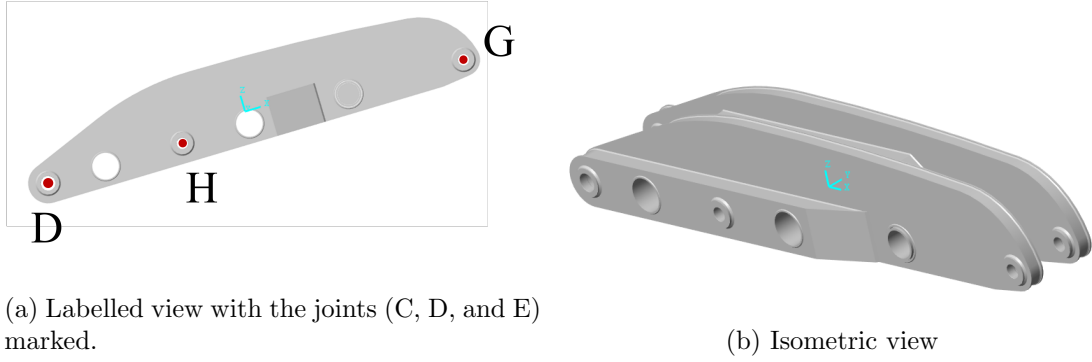


Figure 3.15: The horizontal boom, link DG, in the *FutureForge* manipulator.

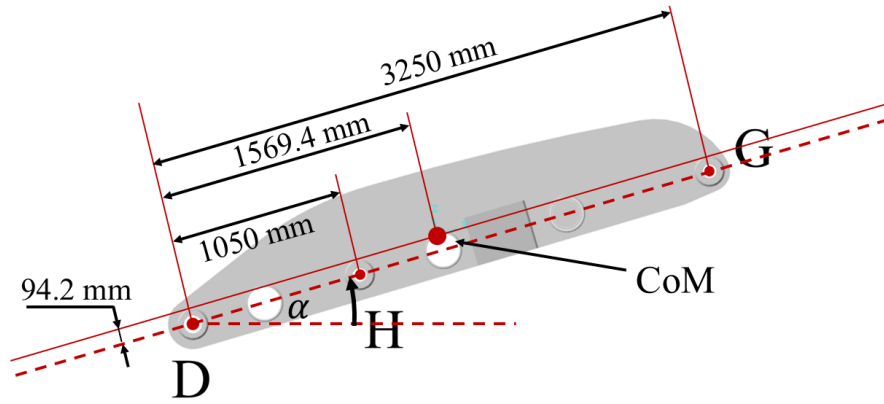


Figure 3.16: Detailed view of the dimensions and angles defining the location of the centre of mass (CoM) of the horizontal boom.

The expressions defining the location of the boom's centre of mass are given as follows. The coefficient of the final trigonometric terms in these expressions is the distance between joint D and the centre of mass, i.e. the hypotenuse of the triangle with sides 94.2 mm and 1569.4 mm.

$$x_{C_{DG}} = x_T + x_{TA} + l_{AD} \cos(\beta) + 1.5722 \cos\left(\alpha + \arctan\left(\frac{94.2}{1569.4}\right)\right) \quad (3.15)$$

$$y_{C_{DG}} = h_A + h_d + l_{AD} \sin(\beta) + 1.5722 \sin\left(\alpha + \arctan\left(\frac{94.2}{1569.4}\right)\right) \quad (3.16)$$

The upper horizontal link, link EF, remains parallel to DG and TU throughout the manipulator's operation. Despite the small portion of mass added asymmetrically near to joint F (see Fig 3.17), the centre of mass of this component remains very close to the location assumed in [59], with only a 6 mm shift along the line EF towards point F.

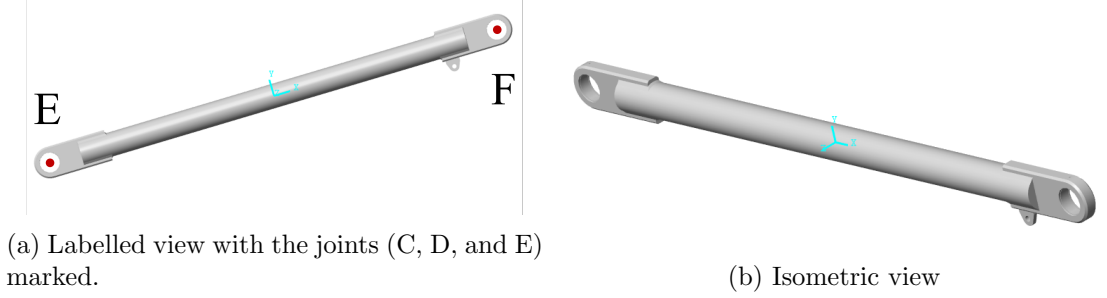


Figure 3.17: The horizontal boom, link DG, in the *FutureForge* manipulator.

Adding this slight modification to the expressions for the centre of mass leads to the following expressions.

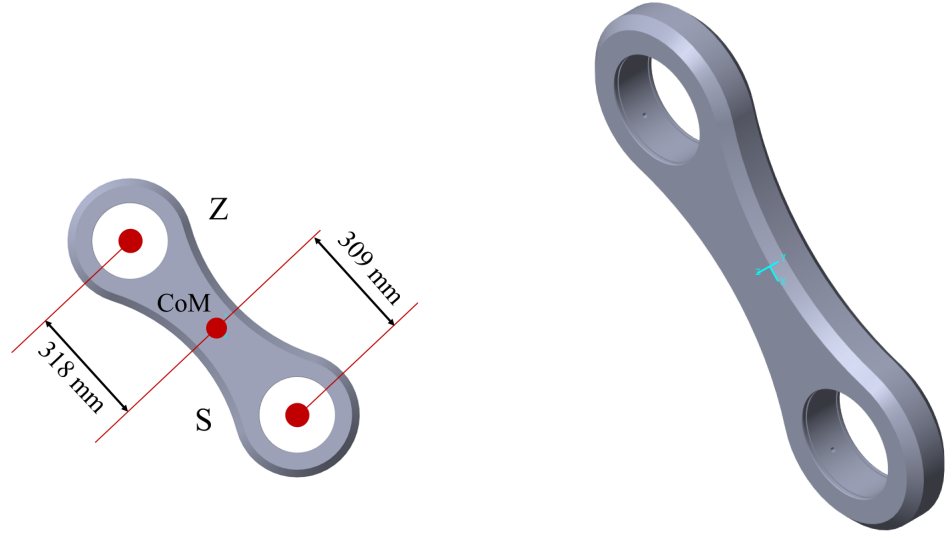
$$x_{C_{EF}} = x_T - x_{TB} + l_{BC} \cos(\beta) + l_{Cp} + \left(\frac{l_{EF}}{2} + 0.006 \right) \cos(\alpha) \quad (3.17)$$

$$y_{C_{EF}} = h_A + h_d + l_{BC} \sin(\beta) + l_{Ep} + \left(\frac{l_{EF}}{2} + 0.006 \right) \sin(\alpha) \quad (3.18)$$

Note that the terms l_{Cp} and l_{Ep} in the above expressions are taken from Fig 2 in [59] where an auxiliary point, p , is created to denote a point on the line CD which allows the right-angled triangle CpE to be created with CE as the hypotenuse. In this way, l_{Cp} denotes the horizontal distance between C and E; while, l_{Ep} denotes the vertical distance between C and E.

The push link and push rocker, shown in Figs 3.18 and 3.19 respectively, are integral to the function of the parallelogram TDGU's ability to extend the horizontal position of the end-effector. In the schematic, Fig 3.1, these comparatively small components are nearly entirely obscured by the rear rocker and vertical boom. Attached to a fixed joint Z, slightly above C (as shown in Fig 3.7), the push link, ZS, drives the push

rocker, DST, to rotate about an axis through D. This results in the parallel lines DT and GU (and GW by extension) rotating anticlockwise about their constraining joint axes, D and G respectively. By consequence, the vertical link being orientated at an angle β to the horizontal results in the lines DT and GU being orientated at an angle of $\frac{\pi}{2} - \beta$ radians. This is by virtue of the manipulator's design intent: when $\alpha = 0$, notice that the kinematic chain ADGW forms a regular trapezoid with the horizontal AW being parallel to DG and AD and GW being of equal length. In this configuration, the included angle DGW must be $\pi - \beta$ radians (this is the angle between the forward vertical boom, GW, and the horizontal boom, DG). Hence, the angle included by the forward vertical boom, GW, and a reference vertical line passing through G, is $\frac{\pi}{2} - \beta$ radians. This angle is denoted by γ for brevity.



(a) Detailed view of the dimensions defining the location of the centre of mass (CoM) of the push link.

(b) Isometric view

Figure 3.18: The push link, link ZS, in the *FutureForge* manipulator.

With only slight modification to the expressions derived by Cartmell et al., the

locations of the centre of mass of the push link can be written as follows.

$$x_{C_{ZS}} = x_T + x_{TA} + l_{AD} \cos(\beta) + l_{DG} \cos(\alpha) + l_{GU} \sin(\gamma) - l_{TU} \cos(\alpha) - l_{ST} \sin(\phi) - 0.309 \cos\left(\arcsin\left(\frac{y_Z - y_S}{l_{ZS}}\right)\right) \quad (3.19)$$

$$y_{C_{ZS}} = h_A + h_d + l_{AD} \sin(\beta) + l_{DG} \sin(\alpha) - l_{GU} \cos(\gamma) - l_{TI} \sin(\alpha) + l_{ST} \cos(\phi) + 0.309 \sin\left(\arcsin\left(\frac{y_Z - y_S}{l_{ZS}}\right)\right) \quad (3.20)$$

Note that ϕ retains its definition, as shown in Fig 4 of [59], as the angle between ST and an imaginary vertical reference line passing through S. The only element of these expressions which has been modified from their original derivation is the coefficient of the final trigonometric terms. As can be identified in Fig 3.18a, this is required to reflect the asymmetric mass distribution. However, as is noted in [59], this component can be neglected from the governing dynamics which will result from the present derivation. This step can be taken since the mass and mass moment of inertia of this particular link contribute very little to the overall system's mass and mass moment of inertia. Where the preceding work noted that the components comprising the “rear rocker assembly” accounted for 1.136% of the system's total mass, this push link has remained similarly insignificant throughout the redesign process. The redesigned part has a mass of 111.2 kg versus a total machine mass of 9941.1 kg (excluding any attachments to the end effector, such as gripper arms), bringing its contribution to just 1.119% of the total mass. It should be noted here that other components in the “rear rocker assembly”, such as the rear rocker (Fig 3.19), now account for a larger proportion of the machine's mass (with the rear rocker accounting for 3.073% of the mass).

In [59], the coordinates of the push rocker's centre of mass is denoted $(x_{C_{ST}}, y_{C_{ST}})$. However, the inclusion of the joint D is preferred here and so $(x_{C_{DST}}, y_{C_{DST}})$ is used instead.

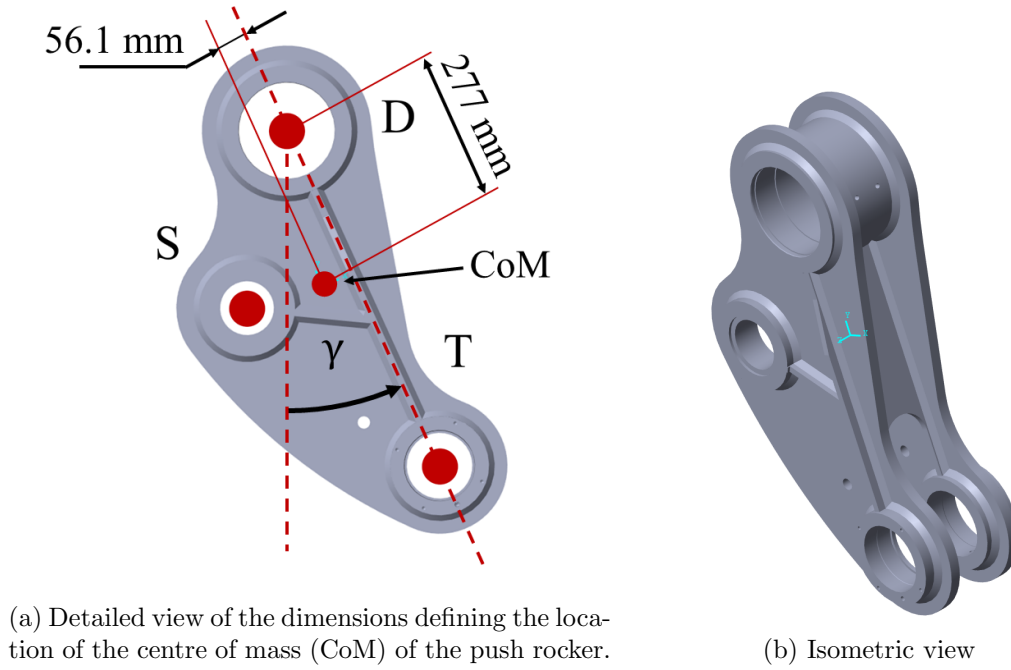
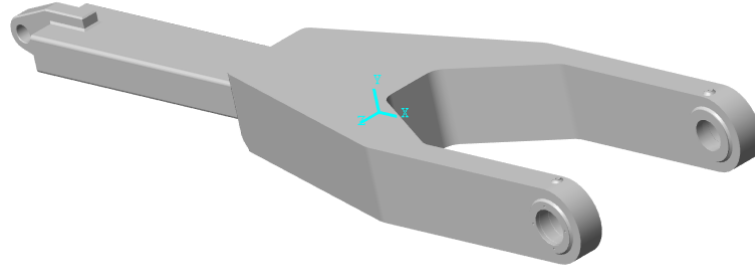


Figure 3.19: The push rocker, component DST, in the *FutureForge* manipulator.

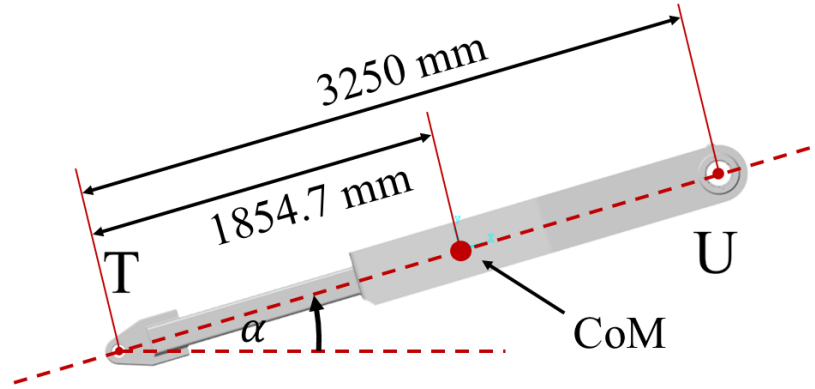
Aside from this simple change in notation, other elements of these expressions have also changed significantly. The reason for this is simply the availability of more information that can be used to specify the location of the centre of mass, which was previously unavailable to Cartmell et al. at the time of writing of [59].

$$x_{C_{DST}} = x_T + x_{TA} + l_{AD} \cos(\beta) + 0.2826 \sin \left(\gamma - \arctan \left(\frac{56.1}{277} \right) \right) \quad (3.21)$$

$$y_{C_{DST}} = h_A + h_d + l_{AD} \sin(\beta) - 0.2826 \cos \left(\gamma - \arctan \left(\frac{56.1}{277} \right) \right) \quad (3.22)$$



(a) Isometric view



(b) Detailed view of the dimensions defining the location of the centre of mass (CoM) of the lower horizontal link.

Figure 3.20: The lower horizontal link, link TU, in the *FutureForge* manipulator.

The derivation of the coordinates for the centre of mass of the lower horizontal link shown in Fig 3.20, TU, is also different to the original result. The difference here is an instance of the choice to follow an alternative kinematic chain. This choice is made because of the perceived simplicity in light of the newly available geometric parameters.

$$x_{CTU} = x_T + x_{TA} + l_{AD} \cos(\beta) + l_{DG} \cos(\alpha) + l_{GU} \sin(\gamma) - 1.3953 \cos(\alpha) \quad (3.23)$$

$$y_{CTU} = h_A + h_d + l_{AD} \sin(\beta) + l_{DG} \sin(\alpha) - l_{GU} \cos(\gamma) - 1.3953 \sin(\alpha) \quad (3.24)$$

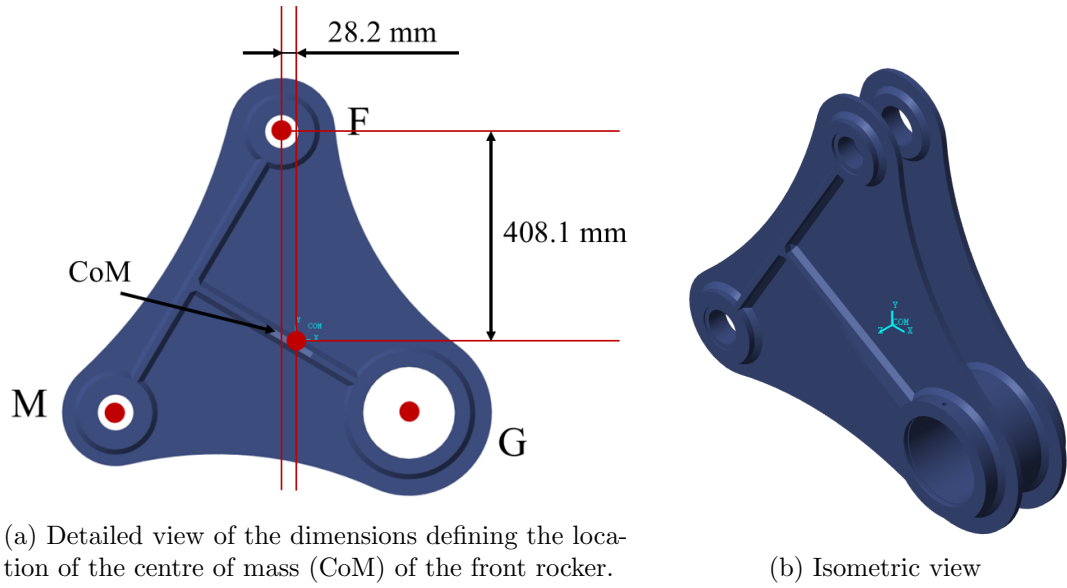


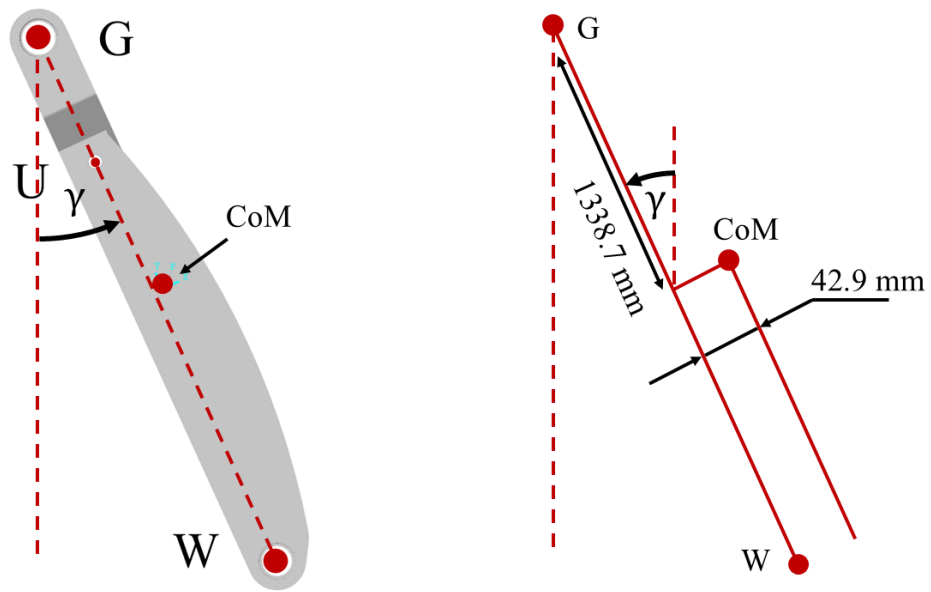
Figure 3.21: The front rocker, component FGM, in the *FutureForge* manipulator.

$$x_{C_{FGM}} = x_T - x_{TB} + l_{BC} \cos(\beta) + l_{Cp} + l_{EF} \cos(\alpha) + 0.0282 \quad (3.25)$$

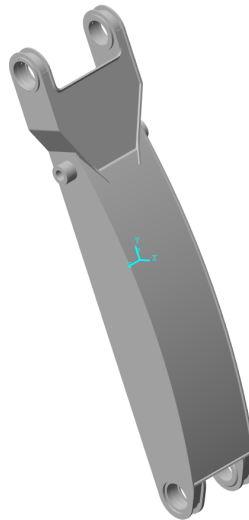
$$y_{C_{FGM}} = h_a + h_d + l_{BC} \sin(\beta) + l_{Ep} + l_{EF} \sin(\alpha) - 0.4081 \quad (3.26)$$

In these expressions for $x_{C_{FGM}}$ and $y_{C_{FGM}}$, describing the position of the front rocker's centre of mass as noted in Fig 3.21, note the constant values (0.0282 and -0.4081) added to them. These are present for the same reason similar terms are found in the expressions for $x_{C_{CDE}}$ and $y_{C_{CDE}}$: because this particular component's orientation is independent of the generalised coordinates.

The derivation of the centre of mass coordinates for the forward vertical boom shown in Fig 3.22, GW, follows a similar path to that of previous components.



(a) Detailed view of the dimensions defining the location of the centre of mass (CoM) of the forward vertical boom.



(b) Isometric view

Figure 3.22: The forward vertical boom, link GW, in the *FutureForge* manipulator.

$$x_{C_{GW}} = x_T + x_{TA} + l_{AJ} \cos(\alpha) + l_{JH} \cos(\beta) + l_{HG} \cos(\alpha) + 1.3387 \sin(\gamma) + 0.0429 \cos(\gamma) \quad (3.27)$$

$$y_{C_{GW}} = h_A + h_d + l_{BC} \sin(\beta) + l_{DG} \sin(\alpha) - 1.3387 \cos(\gamma) + 0.0429 \sin(\gamma) \quad (3.28)$$

Penultimately, the pitch cylinder and its extension link are considered as a single rigid component. This choice is made so as to limit the scope of the present problem to only consider the Cartesian position of the end-effector (moving as a rigid body with the roll rocker in Fig 3.24) rather than also considering its pitch within this space.

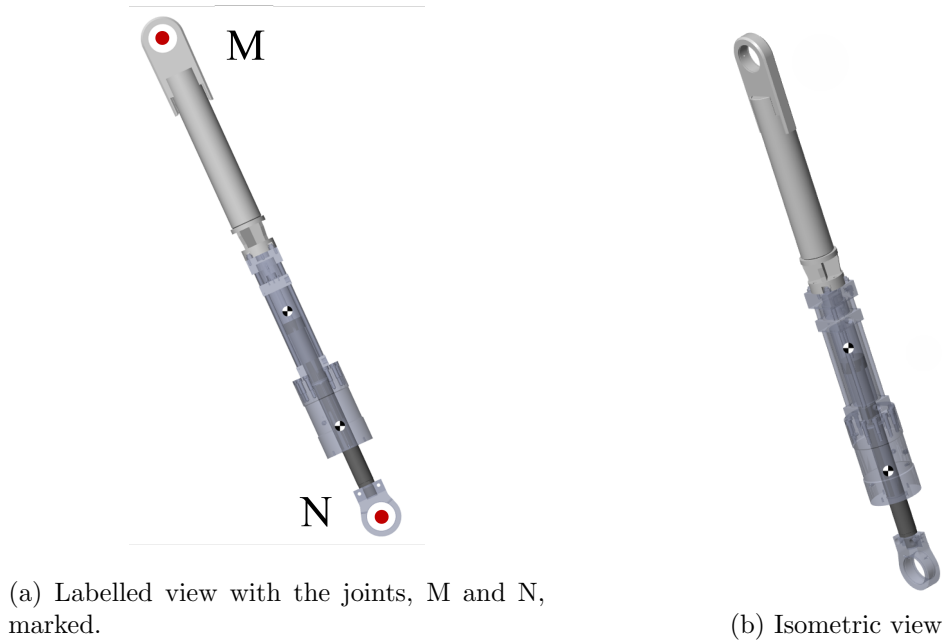


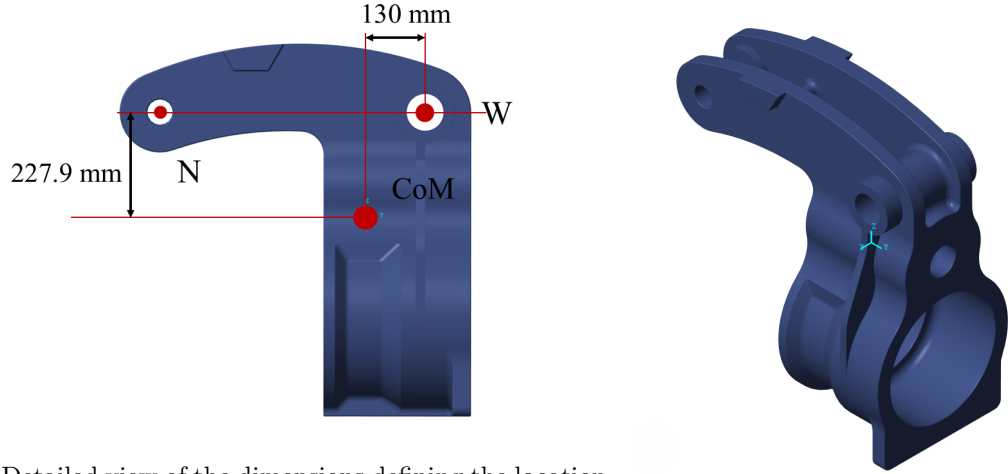
Figure 3.23: The pitch cylinder assembly, link MN, in the *FutureForge* manipulator.

To arrive at the constant coefficients of the $\sin(\gamma)$ and $\cos(\gamma)$ terms, the weighted-average of the locations of each sub-component's centre of mass is determined. This includes treating the hydraulic cylinder's body, connecting shaft, and connector socket

as separate bodies initially to determine this accurately.

$$x_{C_{MN}} = x_T + x_{TA} + l_{AJ} \cos(\alpha) + l_{HG} \cos(\alpha) + l_{JH} \cos(\beta) - l_{GM} + 1.6485 \sin(\gamma) \quad (3.29)$$

$$y_{C_{MN}} = h_A + h_d + l_{AD} \sin(\beta) + l_{DG} \sin(\alpha) - 1.6485 \cos(\gamma) \quad (3.30)$$



(a) Detailed view of the dimensions defining the location of the centre of mass (CoM) of the roll rocker.

(b) Isometric view

Figure 3.24: The roll rocker, link NW, in the *FutureForge* manipulator.

Expressions for the kinetic and potential energy of the system can then be constructed.

$$U_v = \left(2m_{AJK}y_{C_{AJK}} + 2m_{JHY}y_{C_{JH}} + m_{EFY}y_{C_{EF}} + m_{DGY}y_{C_{DG}} + m_{TUY}y_{C_{TU}} + m_{FGM}y_{C_{FGM}} + m_{MNY}y_{C_{MN}} + m_{GWY}y_{C_{GW}} + m_{NWy}y_{C_{NW}} \right) g \quad (3.31)$$

$$\begin{aligned}
 K_v = & m_{AJK} (\dot{x}_{CAJK}^2 + \dot{y}_{CAJK}^2) + \frac{1}{2} m_{DG} (\dot{x}_{CDG}^2 + \dot{y}_{CDG}^2) + \frac{1}{2} m_{EF} (\dot{x}_{CEF}^2 + \dot{y}_{CEF}^2) \\
 & + \frac{1}{2} m_{FGM} (\dot{x}_{CFGM}^2 + \dot{y}_{CFGM}^2) + \frac{1}{2} m_{GW} (\dot{x}_{CGW}^2 + \dot{y}_{CGW}^2) + m_{JH} (\dot{x}_{CJH}^2 + \dot{y}_{CJH}^2) \\
 & + \frac{1}{2} m_{MN} (\dot{x}_{CMN}^2 + \dot{y}_{CMN}^2) + \frac{1}{2} m_{NW} (\dot{x}_{CNW}^2 + \dot{y}_{CNW}^2) + \frac{1}{2} m_{TU} (\dot{x}_{CTU}^2 + \dot{y}_{CTU}^2) \\
 & + J_{AJK} \dot{\alpha}^2 + \frac{1}{2} J_{DG} \dot{\alpha}^2 + \frac{1}{2} J_{EF} \dot{\alpha}^2 + \frac{1}{2} J_{TU} \dot{\alpha}^2
 \end{aligned} \tag{3.32}$$

$$\begin{aligned}
 U_h = & \left(m_{AD} y_{CAD} + m_{BC} y_{CBC} + m_{CDE} y_{CDE} + m_{DG} y_{CDG} + m_{DST} y_{CST} \right. \\
 & + m_{EF} y_{CEF} + m_{FGM} y_{CFGM} + m_{GW} y_{CGW} + 2m_{JH} y_{CJH} + m_{MN} y_{CMN} \\
 & \left. + m_{NW} y_{CNW} + m_{TU} y_{CTU} + m_{ZS} y_{CZS} \right) g
 \end{aligned} \tag{3.33}$$

$$\begin{aligned}
 K_h = & \frac{1}{2} m_{BC} (\dot{x}_{CBC}^2 + \dot{y}_{CBC}^2) + \frac{1}{2} m_{AD} (\dot{x}_{CAD}^2 + \dot{y}_{CAD}^2) + \frac{1}{2} m_{CDE} (\dot{x}_{CDE}^2 + \dot{y}_{CDE}^2) \\
 & + \frac{1}{2} m_{EF} (\dot{x}_{CEF}^2 + \dot{y}_{CEF}^2) + \frac{1}{2} m_{DG} (\dot{x}_{CDG}^2 + \dot{y}_{CDG}^2) + \frac{1}{2} m_{FGM} (\dot{x}_{CFGM}^2 + \dot{y}_{CFGM}^2) \\
 & + \frac{1}{2} m_{TU} (\dot{x}_{CTU}^2 + \dot{y}_{CTU}^2) + m_{JH} (\dot{x}_{CJH}^2 + \dot{y}_{CJH}^2) + \frac{1}{2} m_{MN} (\dot{x}_{CMN}^2 + \dot{y}_{CMN}^2) \\
 & + \frac{1}{2} m_{GW} (\dot{x}_{CGW}^2 + \dot{y}_{CGW}^2) + \frac{1}{2} m_{NW} (\dot{x}_{CNW}^2 + \dot{y}_{CNW}^2) + \frac{1}{2} m_{DST} (\dot{x}_{CDST}^2 + \dot{y}_{CDST}^2) \\
 & + \frac{1}{2} m_{ZS} (\dot{x}_{CZS}^2 + \dot{y}_{CZS}^2) + \frac{1}{2} J_{BC} \dot{\beta}^2 + \frac{1}{2} J_{AD} \dot{\beta}^2 \\
 & + J_{JH} \dot{\beta}^2 + \frac{1}{2} J_{MN} \dot{\gamma}^2 + \frac{1}{2} J_{GW} \dot{\gamma}^2 + \frac{1}{2} J_{ST} \dot{\gamma}^2
 \end{aligned} \tag{3.34}$$

In Cartmell et al.'s earlier work, γ is identified as being $\gamma = \frac{\pi}{2} - \beta$ [59]. This can be verified by consulting the regular trapezoid shown in Fig 3.25: due to the symmetry of this shape formed by the joints A, D, G, and W, the included angle in DGW must be $\pi - \beta$, and hence (using a vertical line as the reference at G), $\gamma = \frac{\pi}{2} - \beta$ can be concluded. However, when inspecting the shape formed using these joints in the revised design, slight asymmetry has since led to $\gamma = 1.498 - \beta$ being a more suitable substitution.



Figure 3.25: The regular trapezoid ADGW which validates Cartmell et al.'s substitution of $\gamma = \frac{\pi}{2} - \beta$.

Using this substitution for γ in the energy expressions stated, the Euler-Lagrange equations

$$\frac{d}{dt} \left(\frac{\partial K_v}{\partial \dot{\alpha}} \right) - \frac{\partial K_v}{\partial \alpha} + \frac{\partial U_v}{\partial \alpha} = Q_v \quad (3.35)$$

and

$$\frac{d}{dt} \left(\frac{\partial K_h}{\partial \dot{\beta}} \right) - \frac{\partial K_h}{\partial \beta} + \frac{\partial U_h}{\partial \beta} = Q_h \quad (3.36)$$

can be computed for the vertical and horizontal actuation problems respectively. Note that the generalised torques, Q_v and Q_h , are unchanged from Cartmell et al.'s work [59]. Both of these are derived by considering the virtual work done by forces applied through the hydraulic actuators: since the work required to change α by a small amount $\delta\alpha$ is $\delta W = F_v l_{AK} \delta\alpha$, then the generalised torque in this scenario is $Q_v = F_v l_{AK}$. The same process being considered for small changes, $\delta\beta$, in β , results in $Q_h = F_h l_{PA}$ where l_{PA} is the distance from joint A to the point where the hydraulic actuator connects to the vertical boom. Due to delayed access to information on the hydraulic actuators, and the physical manipulator in general, the pragmatic decision was made to focus the present model to consider the torques Q_v and Q_h as the inputs to the dynamic model without including any dynamics from the hydraulic components themselves.

Both this revised set of equations and the original set can be written in the following

form.

$$\begin{aligned}
 & \left(a_1 + a_2 \cos(2\alpha) + a_3 \sin(2\alpha) \right) \ddot{\alpha} \\
 & + \left(a_4 \cos(\alpha - \beta) + a_5 \cos(\alpha + \beta) + a_6 \sin(\alpha - \beta) + a_7 \sin(\alpha + \beta) \right) \ddot{\beta} \\
 & + a_8 \dot{\alpha} + \left(a_9 + a_{10} \cos(2\alpha) + a_{11} \sin(2\alpha) \right) \dot{\alpha}^2 \\
 & + \left(a_{12} \cos(\alpha - \beta) + a_{13} \cos(\alpha + \beta) + a_{14} \sin(\alpha - \beta) + a_{15} \sin(\alpha + \beta) \right) \dot{\beta}^2 \\
 & + a_{16} \cos(\alpha) + a_{17} \sin(\alpha) \\
 & = Q_v
 \end{aligned} \tag{3.37}$$

$$\begin{aligned}
 & \left(b_1 + b_2 \cos(2\beta) + b_3 \sin(2\beta) \right) \ddot{\beta} \\
 & + \left(b_4 \cos(\alpha - \beta) + b_5 \cos(\alpha + \beta) + b_6 \sin(\alpha - \beta) + b_7 \sin(\alpha + \beta) \right) \ddot{\alpha} \\
 & + b_8 \dot{\beta} + \left(b_9 + b_{10} \cos(2\beta) + b_{11} \sin(2\beta) \right) \dot{\beta}^2 \\
 & + \left(b_{12} \cos(\alpha - \beta) + b_{13} \cos(\alpha + \beta) + b_{14} \sin(\alpha - \beta) + b_{15} \sin(\alpha + \beta) \right) \dot{\alpha}^2 \\
 & + b_{16} \cos(\beta) + b_{17} \sin(\beta) \\
 & = Q_h
 \end{aligned} \tag{3.38}$$

The constant a and b coefficient terms in Eqs (3.37) and (3.38) are contained in Appendix A.

The system of equations comprising Eqs (3.37) and (3.38) can be written in matrix form as follows.

$$M(\Theta) \ddot{\Theta} + P(\Theta, \dot{\Theta}) \dot{\Theta} + G(\Theta) = Q(t) \tag{3.39}$$

where $\Theta = [\alpha, \beta]^T$, $\mathbf{Q}(t) = [Q_v(t), Q_h(t)]^T$,

$$M(\Theta) = \begin{bmatrix} a_1 + a_2 \cos(2\alpha) + a_3 \sin(2\alpha) & a_4 \cos(\alpha - \beta) + a_5 \cos(\alpha + \beta) \\ & + a_6 \sin(\alpha - \beta) + a_7 \sin(\alpha + \beta) \\ b_4 \cos(\alpha - \beta) + b_5 \cos(\alpha + \beta) & b_1 + b_2 \cos(2\beta) + b_3 \sin(2\beta) \\ + b_6 \sin(\alpha - \beta) + b_7 \sin(\alpha + \beta) & \end{bmatrix}, \quad (3.40)$$

$$P(\Theta, \dot{\Theta}) = \begin{bmatrix} a_8 + (a_9 + a_{10} \cos(2\alpha) & a_{12} \cos(\alpha - \beta) + a_{13} \cos(\alpha + \beta) \\ + a_{11} \sin(2\alpha))\dot{\alpha} & + a_{14} \sin(\alpha - \beta) + a_{15} \sin(\alpha + \beta) \\ b_{12} \cos(\alpha - \beta) + b_{13} \cos(\alpha + \beta) & b_8 + (b_9 + b_{10} \cos(2\beta) \\ + b_{14} \sin(\alpha - \beta) + b_{15} \sin(\alpha + \beta) & + b_{11} \sin(2\beta))\dot{\beta} \end{bmatrix}, \quad (3.41)$$

and

$$\mathbf{G}(\Theta) = \begin{bmatrix} a_{16} \cos(\alpha) + a_{17} \sin(\alpha) \\ b_{16} \cos(\beta) + b_{17} \sin(\beta) \end{bmatrix}. \quad (3.42)$$

3.2 Application of the ADAP methodology for solving the manipulator dynamics

In this subsection, the application of the ADAP methodology to the preceding dynamic system is described. This follows the same algorithm as is followed in the examples in Chapter 2, however, some important steps - which are required to ensure the problem remains analytically tractable - are additionally highlighted and emphasised for completeness.

3.2.1 Construction and solution of the uncoupled system

Two notable simplifications can be made to the dynamic model of the manipulator. The first is described in [58] and is used to remove the gravitational restoring torques. Meanwhile, the second involves approximating the remaining diagonal inertial and damping

terms via simple Taylor Series expansions.

Exploitation of the excitation vector

The excitation torques, $\mathbf{Q} = [Q_v, Q_h]^T$, may abstractly be considered as nonlinear functions of the generalised coordinates and time. Logically, for the manipulator to resist “falling” under its own weight, i.e. when trying to execute a manoeuvre, part of its total excitation vector must contribute to balancing the gravitational restoring torques. This component of the excitation is denoted $R_v(\alpha)$ and $R_h(\beta)$ for the two excitations in Eqs (3.37) and (3.38) respectively. However, for there then to be motion (beyond the manipulator simply being held in place), there must also be an excitation beyond this gravity-balancing component. This can be represented with a time-dependent term in both cases.

$$Q_v = R_v(\alpha) + F_v(t) \quad (3.43)$$

$$Q_h = R_h(\beta) + F_h(t) \quad (3.44)$$

This approach is similar to Hsu’s work in 1960 wherein the excitation functions applied to forced oscillators are tuned to be proportional to the restoring forces [68]. Similarly, Rakaric, Kovacic, and Cartmell tune the nonhomogeneous terms in their oscillator system to permit the derivation of closed-form solutions [69].

So that these R -components can carry out the role of balancing restoring torques in $\mathbf{G}(\Theta)$, $R_v(\alpha)$ must equal the restoring torque in Eq (3.37)

$$R_v(\alpha) = G_1 = a_{16} \cos(\alpha) + a_{17} \sin(\alpha) \quad (3.45)$$

and $R_h(\beta)$ must equal the restoring torque in Eq (3.38).

$$R_h(\beta) = G_2 = b_{16} \cos(\beta) + b_{17} \sin(\beta) \quad (3.46)$$

Other sources have previously adopted methods that involve assuming the absence of gravitational restoring effects or that they are constant [47]. While the impact on the resulting governing differential equations is the same, an operational justification has been provided here, rather than simply assuming this simplification is justified as an *a priori* assumption. The resulting system can hence be written as

$$M(\Theta)\ddot{\Theta} + P(\Theta, \dot{\Theta})\dot{\Theta} = \mathbf{F}(t), \quad (3.47)$$

where $\mathbf{F}(t) = [F_v(t), F_h(t)]^T$.

Relative magnitudes of coefficients

Acknowledging the rationale of methods such as the Lindstedt-Poincaré technique, the more general method of multiple scales, and asymptotic approaches in the broader sense, a useful starting point for this problem is to assess the relative magnitudes of the a and b constants in Eqs (3.37) and (3.38) and investigate the possibility of introducing a small parameter, ε , through this. Before revising the manipulator's governing ODEs to account for the more complex component geometries, this was the approach taken and is therefore demonstrated in [59]. Accounting for the revised equations of motion, the constants have the values shown in Table 3.1. As can be observed, these coefficients span many orders of magnitude, with the smallest being $b_9 = -6.39488 \times 10^{-14}$ and the largest being $b_1 = 229479$. The greatest orders of magnitude are noted in Table 3.2. As discussed in [59], the original interpretation of the system permitted only the linear inertia, a_1 and b_1 , and linear viscous damping, a_8 and b_8 , to contribute at the highest orders of magnitude. Hence, these could be considered at $O(1)$ and the remaining terms could be relegated to $O(\varepsilon)$, $O(\varepsilon^2)$, and so on based on their magnitudes. Based on these revised coefficient values, however, it may now be more appropriate to consider another ordering of terms.

Suppose that the largest magnitude coefficients, those shown in Table 3.2, were

Coefficient in Eq (3.37)	Value	Coefficient in Eq (3.38)	Value
a_1	79447.5	b_1	229479
a_2	9.09495×10^{-13}	b_2	-14970.4
a_3	0	b_3	978.883
a_4	21688.8	b_4	21688.8
a_5	-8309.24	b_5	-8309.24
a_6	-256.2	b_6	-256.2
a_7	516.554	b_7	516.554
a_8	c_d	b_8	c_d
a_9	0	b_9	-6.39488×10^{-14}
a_{10}	0	b_{10}	978.883
a_{11}	9.09495×10^{-13}	b_{11}	14970.4
a_{12}	256.2	b_{12}	-256.2
a_{13}	516.554	b_{13}	516.554
a_{14}	21688.8	b_{14}	-21688.8
a_{15}	8309.24	b_{15}	8309.24
a_{16}	8607.97	b_{16}	15097
a_{17}	-40.0946	b_{17}	190.19

Table 3.1: The constant values of the coefficients occurring in Eqs (3.37) and (3.38). Note the value $c_d = 10^5$ is chosen to reflect the linear viscous damping observed in the *FutureForge* manipulator.

Coefficient in Eq (3.37)	Rounded magnitude	Coefficient in Eq (3.38)	Rounded magnitude
a_8	1×10^5	b_1	2×10^5
a_1	8×10^4	b_8	1×10^5
a_{14}	2×10^4	b_{14}	2×10^4
a_4	2×10^4	b_4	2×10^4
a_{16}	9×10^3	b_{16}	2×10^4
a_{15}	8×10^3	b_{11}	1×10^4
a_5	8×10^3	b_2	1×10^4

Table 3.2: The largest constant values of the coefficients occurring in Eqs (3.37) and (3.38).

included at the $O(1)$ level. The resulting simplified system would be written as

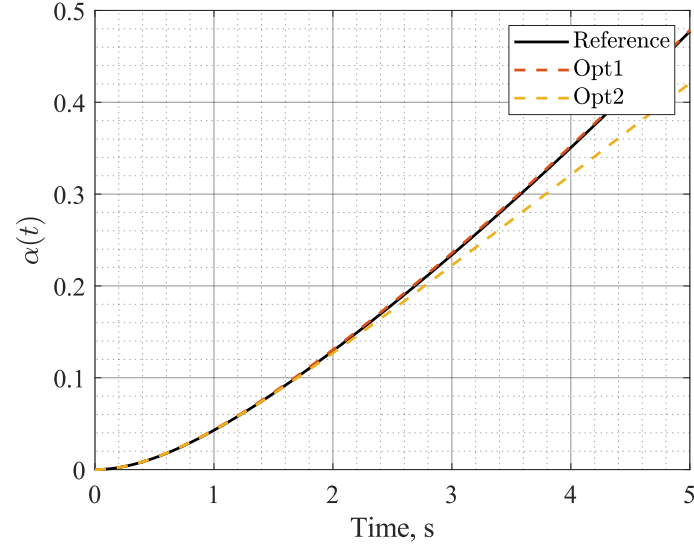
$$\begin{aligned}
 & \begin{bmatrix} a_1 & a_4 \cos(\alpha_0 - \beta_0) + a_5 \cos(\alpha_0 + \beta_0) \\ b_4 \cos(\alpha_0 - \beta_0) & b_1 + b_2 \cos(2\beta_0) \end{bmatrix} \begin{Bmatrix} \ddot{\alpha}_0 \\ \ddot{\beta}_0 \end{Bmatrix} \\
 & + \begin{bmatrix} a_8 & (a_{14} \sin(\alpha_0 - \beta_0) + a_{15} \sin(\alpha_0 + \beta_0)) \dot{\beta}_0 \\ b_8 + b_{11} \sin(2\beta_0) \dot{\beta}_0 & b_{14} \sin(\alpha_0 - \beta_0) \dot{\alpha}_0 \end{bmatrix} \begin{Bmatrix} \dot{\alpha}_0 \\ \dot{\beta}_0 \end{Bmatrix} \\
 & = \begin{Bmatrix} F_v \\ F_h \end{Bmatrix} \quad (3.48)
 \end{aligned}$$

using the very simple notation of Lighthill's technique where α_0 and β_0 are the $O(1)$ approximations of α and β . By contrast, Eq (3.49) shows the linearised, uncoupled equation used in [43][44].

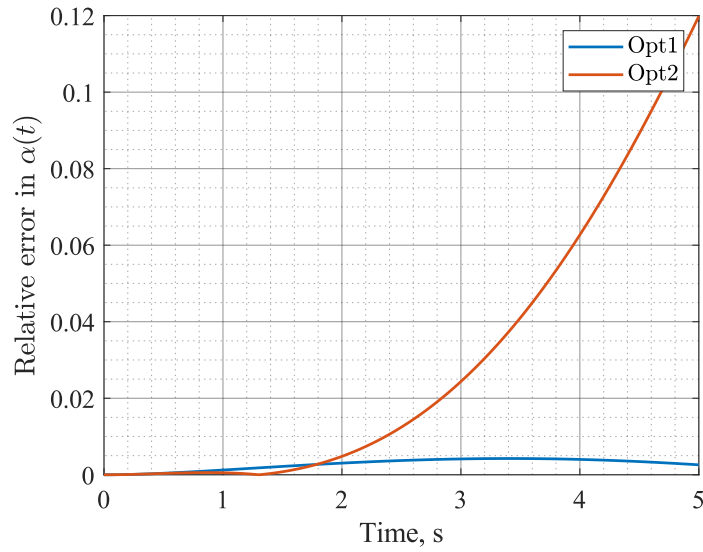
$$\begin{bmatrix} a_1 & 0 \\ 0 & b_1 \end{bmatrix} \begin{Bmatrix} \ddot{\alpha}_0 \\ \ddot{\beta}_0 \end{Bmatrix} + \begin{bmatrix} a_8 & 0 \\ 0 & b_8 \end{bmatrix} \begin{Bmatrix} \dot{\alpha}_0 \\ \dot{\beta}_0 \end{Bmatrix} = \begin{Bmatrix} F_v \\ F_h \end{Bmatrix} \quad (3.49)$$

Comparing Eqs (3.48) and (3.49), the latter is clearly far simpler and more conducive to analytical solutions. However, to examine the trade-off in this choice, consider Figs 3.26 and 3.27. From these graphs, it is evident that there is a significant sacrifice in accuracy by pursuing the vastly simplified form of Eq (3.49), labelled as “Opt2” in Figs 3.26 and 3.27, at least at the $O(1)$ level. Given the errors shown in Figs 3.26b and 3.27b, alternate methods would be advisable if modelling behaviours spanning any time frame in the order of 1 second or higher. One such method for consideration may be to use gain scheduling in order to achieve constant approximations for the off-diagonal terms in the inertia and damping matrices of Eq (3.48). However, crucially for the later application of this method to model-predictive control, this is unnecessary with the ADAP methodology as the second auxiliary problem is constructed in such a way that the “corrections” to the first auxiliary problem function in a similar way, and provide a similar benefit, to gain scheduling.

With the knowledge that later in this work, model-predictive control is to be used,

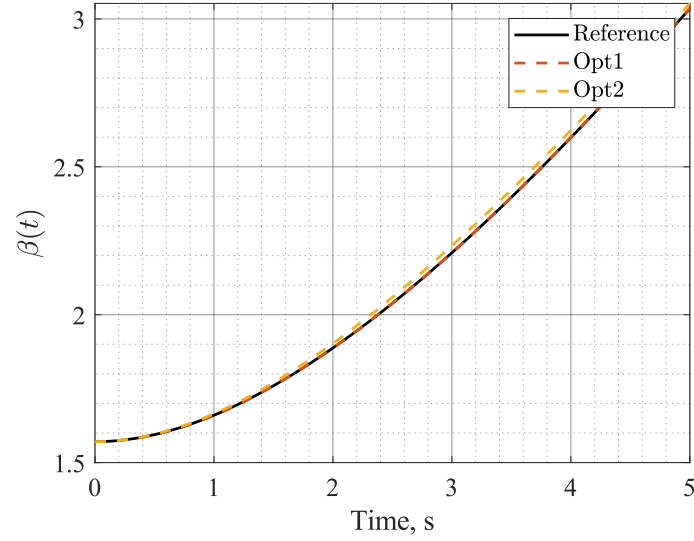


(a) Comparison of model solutions.

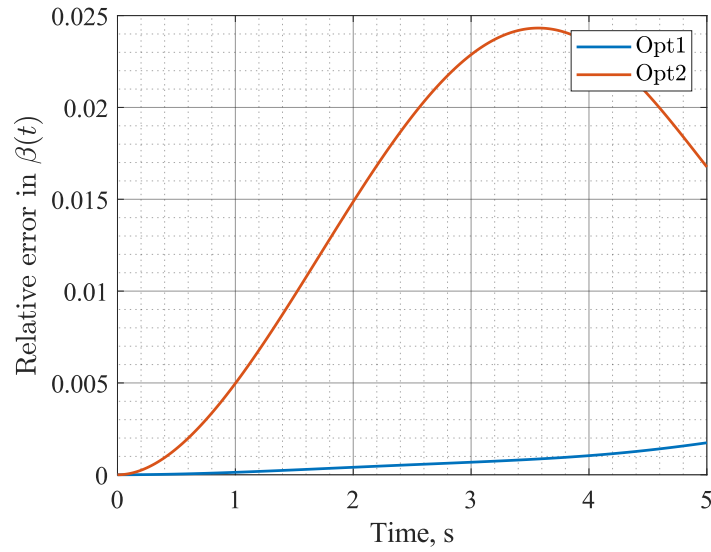


(b) Relative error in solving Eqs (3.48) and (3.49) for $\alpha_1(t)$.

Figure 3.26: A comparison of the approximation of $\alpha(t)$ by $\alpha_0(t)$ via the numerical solution of: Eq (3.37), “Reference”; Eq (3.48), “Opt1”; and, Eq (3.49), “Opt2”.



(a) Comparison of model solutions.



(b) Relative error in solving Eqs (3.48) and (3.49) for $\beta_1(t)$.

Figure 3.27: A comparison of the approximation of $\beta(t)$ by $\beta_0(t)$ via the numerical solution of: Eq (3.38), “Reference”; Eq (3.48), “Opt1”; and, Eq (3.49), “Opt2”.

the required time frame for accurate predictions is likely to be relatively short. Hence, using Eq (3.49), the first auxiliary problem is written as follows.

$$\begin{bmatrix} a_1 & 0 \\ 0 & b_1 \end{bmatrix} \begin{Bmatrix} \ddot{\phi}_1 \\ \ddot{\phi}_2 \end{Bmatrix} + \begin{bmatrix} a_8 & 0 \\ 0 & b_8 \end{bmatrix} \begin{Bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \end{Bmatrix} = \begin{Bmatrix} F_v \\ F_h \end{Bmatrix} \quad (3.50)$$

The complementary functions of this system are

$$\phi_{cf,1} = K_{v,1} + K_{v,2}e^{-\frac{a_8}{a_1}t} \quad (3.51)$$

and

$$\phi_{cf,2} = K_{h,1} + K_{h,2}e^{-\frac{b_8}{b_1}t} \quad (3.52)$$

where: the K terms are constant coefficients to be later determined by application of the initial system conditions; the characteristics of the system are $\lambda = 0$ and $\lambda = -\frac{a_8}{a_1}$ for ϕ_1 ; and, $\lambda = 0$ and $\lambda = -\frac{b_8}{b_1}$ for ϕ_2 . The particular integrals necessarily depend on the specific forms of the excitation functions, F_v and F_h . Thus, these are generalised to

$$\phi_{p,1} = -e^{-\frac{a_8}{a_1}t} \int \frac{F_v}{W_1} + \int \frac{e^{-\frac{a_8}{a_1}t} F_v}{W_1} \quad (3.53)$$

$$\phi_{p,2} = -e^{-\frac{b_8}{b_1}t} \int \frac{F_h}{W_2} + \int \frac{e^{-\frac{b_8}{b_1}t} F_h}{W_2} \quad (3.54)$$

where W_1 and W_2 are the Wronskians corresponding to the two independent ODEs.

$$\begin{aligned} W_1 &= e^{-\frac{a_8}{a_1}t} \frac{d}{dt} (e^0) + e^0 \frac{d}{dt} \left(e^{-\frac{a_8}{a_1}t} \right) \\ &= -\frac{a_1}{a_8} e^{\frac{a_8}{a_1}t} \end{aligned} \quad (3.55)$$

$$\begin{aligned} W_2 &= e^{-\frac{b_8}{b_1}t} \frac{d}{dt} (e^0) + e^0 \frac{d}{dt} \left(e^{-\frac{b_8}{b_1}t} \right) \\ &= -\frac{b_1}{b_8} e^{-\frac{b_8}{b_1}t} \end{aligned} \quad (3.56)$$

These general solutions are the solutions to the first auxiliary problems required by Eq 3.60, giving the necessary $\hat{M}$ and $\hat{P}$ terms.

3.2.2 Approximation and solution of the fully-coupled system

Recalling the notation used to describe the ADAP methodology in Sec 2.4,

$$\tilde{M} = \begin{bmatrix} a_1 & 0 \\ 0 & b_1 \end{bmatrix} \quad \text{and} \quad \tilde{P} = \begin{bmatrix} a_8 & 0 \\ 0 & b_8 \end{bmatrix}. \quad (3.57)$$

Thus,

$$\begin{aligned} \hat{M}(\Theta) &= M(\Theta) - \tilde{M}(\Theta) \\ &= \begin{bmatrix} a_2 \cos(2\alpha) + a_3 \sin(2\alpha) & M_{1,2}(\Theta) \\ M_{2,1}(\Theta) & b_2 \cos(2\beta) + b_3 \sin(2\beta) \end{bmatrix} \end{aligned} \quad (3.58)$$

$$\begin{aligned} \hat{P}(\Theta, \dot{\Theta}) &= P(\Theta, \dot{\Theta}) - \tilde{P}(\Theta, \dot{\Theta}) \\ &= \begin{bmatrix} (a_9 + a_{10} \cos(2\alpha) + a_{11} \sin(2\alpha)) \dot{\alpha} & P_{1,2}(\Theta, \dot{\Theta}) \\ P_{2,1}(\Theta, \dot{\Theta}) & (b_9 + b_{10} \cos(2\beta) + b_{11} \sin(2\beta)) \dot{\beta} \end{bmatrix} \end{aligned} \quad (3.59)$$

The second auxiliary problem is then written as

$$\tilde{M}(\Theta)\ddot{\Theta} + \tilde{P}(\Theta, \dot{\Theta})\dot{\Theta} = \mathbf{F}(t) - \left(\hat{M}(\Phi)\ddot{\Phi} + \hat{P}(\Phi, \dot{\Phi})\dot{\Phi} \right) \quad (3.60)$$

where $\Phi = [\phi_1, \phi_2]^T$ are the dependent variables in the first auxiliary problem.

To permit an analytically tractable solution, Taylor Series approximations are substituted for the bracketed terms on the right-hand side of Eq (3.60). To aid legibility, the notation $h_n(\cdot)$ is therefore introduced to indicate that an n -th order Taylor Series approximation has been taken for the $(\cdot)$ terms.

$$\tilde{M}(\Theta)\ddot{\Theta} + \tilde{P}(\Theta, \dot{\Theta})\dot{\Theta} = \mathbf{F}(t) - h_1 \left(\left(\hat{M}(\Phi)\ddot{\Phi} + \hat{P}(\Phi, \dot{\Phi})\dot{\Phi} \right) \right) \quad (3.61)$$

Given the adoption of the ADAP methodology circumvents the requirement for sub-

sequent linearisation techniques such as gain scheduling, and since the application of model-predictive control restricts the domain of interest to the sub-second interval (see Chapter 6), a first-order Taylor Series approximation may be sufficient although this assumption is considered when higher-order approximations are examined as part of the results and discussion of the present chapter.

Taylor Series approximations of the transcendental terms

In order to apply the Taylor Series approximations to the governing (approximated) system, addition formulae are applied so as to reduce terms such as $\cos(\alpha - \beta)$ to their simpler transcendental components. Taking this step, Taylor Series approximations for each of the four transcendental terms - $\cos(\alpha)$, $\cos(\beta)$, $\sin(\alpha)$, and $\sin(\beta)$ - and their double-angle counterparts are computed and substituted as required.

3.3 Alternative methods for comparison with the ADAP method's results

In this section, one of the solution approaches, to which the ADAP method is compared, is presented and detailed. The method chosen here as a comparison is a very simple perturbation method as was alluded to in the construction of Eqs (3.48) and (3.49). To permit a fair comparison with the ADAP method - particularly regarding the analytical tractability of the resulting differential equations - the introduction of a small parameter leading to the generating problem Eq (3.49) is adopted.

$$\begin{aligned}
 & \left(a_1 + \varepsilon \tilde{a}_2 \cos(2\alpha) + \varepsilon \tilde{a}_3 \sin(2\alpha) \right) \ddot{\alpha} \\
 & + \varepsilon \left(\tilde{a}_4 \cos(\alpha - \beta) + \tilde{a}_5 \cos(\alpha + \beta) + \tilde{a}_6 \sin(\alpha - \beta) + \tilde{a}_7 \sin(\alpha + \beta) \right) \ddot{\beta} \\
 & + a_8 \dot{\alpha} + \varepsilon \left(\tilde{a}_9 + \tilde{a}_{10} \cos(2\alpha) + \tilde{a}_{11} \sin(2\alpha) \right) \dot{\alpha}^2 \\
 & + \varepsilon \left(\tilde{a}_{12} \cos(\alpha - \beta) + \tilde{a}_{13} \cos(\alpha + \beta) + \tilde{a}_{14} \sin(\alpha - \beta) + \tilde{a}_{15} \sin(\alpha + \beta) \right) \dot{\beta}^2 \\
 & = F_v
 \end{aligned} \tag{3.62}$$

$$\begin{aligned}
& \left(b_1 + \varepsilon \tilde{b}_2 \cos(2\beta) + \varepsilon \tilde{b}_3 \sin(2\beta) \right) \ddot{\beta} \\
& + \varepsilon \left(\tilde{b}_4 \cos(\alpha - \beta) + \tilde{b}_5 \cos(\alpha + \beta) + \tilde{b}_6 \sin(\alpha - \beta) + \tilde{b}_7 \sin(\alpha + \beta) \right) \ddot{\alpha} \\
& + b_8 \dot{\beta} + \varepsilon \left(\tilde{b}_9 + \tilde{b}_{10} \cos(2\beta) + \tilde{b}_{11} \sin(2\beta) \right) \dot{\beta}^2 \\
& + \varepsilon \left(\tilde{b}_{12} \cos(\alpha - \beta) + \tilde{b}_{13} \cos(\alpha + \beta) + \tilde{b}_{14} \sin(\alpha - \beta) + \tilde{b}_{15} \sin(\alpha + \beta) \right) \dot{\alpha}^2 \\
& = F_h
\end{aligned} \tag{3.63}$$

With a view to substituting asymptotic series for $\alpha(t)$ and $\beta(t)$ into Eqs (3.62) and (3.63) and then extracting the $O(\varepsilon^n)$ problems which result, it is convenient to replace the trigonometric expressions with algebraic ones. Hence, Taylor Series approximations are used. An appropriate centre for the Taylor Series approximations of the trigonometric functions of the sums $\alpha \pm \beta$ is challenging to define usefully since the most appropriate value is dependent on the machine's operating conditions and the manoeuvre being undertaken. Hence, addition formulae are used to expand these terms as follows.

$$\cos(\alpha \pm \beta) = \cos(\alpha) \cos(\beta) \mp \sin(\alpha) \sin(\beta) \tag{3.64}$$

$$\sin(\alpha \pm \beta) = \sin(\alpha) \cos(\beta) \pm \cos(\alpha) \sin(\beta) \tag{3.65}$$

Taylor Series expansions for $\sin(\alpha)$, $\cos(\alpha)$, $\sin(\beta)$, and $\cos(\beta)$ are therefore required alongside expansions for $\sin(2\alpha)$, $\cos(2\alpha)$, $\sin(2\beta)$, and $\cos(2\beta)$. The expansions of functions of α are centred at 0 radians, while the expansions of functions of β are centred at $\frac{\pi}{2}$ radians since these are the respective midpoints of the generalised coordinates' domains. When implementing this step of the solution algorithm in *Mathematica*, the order of expansions is a generalised natural number n which is also used in determining the order of expansions used in the ADAP methodology described earlier in this chapter. This ensures consistency in this step across both methods.

From here, the simple asymptotic expansions

$$\alpha(t) \approx \sum_{j=0}^n \varepsilon^j \alpha_j(t) \quad (3.66)$$

$$\beta(t) \approx \sum_{j=0}^n \varepsilon^j \beta_j(t) \quad (3.67)$$

are substituted into Eqs (3.62) and (3.63) with $n = 1$. The choice of $n = 1$ is made as this is seen as the most direct comparison to the ADAP methodology: in this scenario, two systems of differential equations must be solved in order to obtain the subsequent approximation of the manipulator's response. Hence, with $n = 1$, the perturbation method examined here is another "approximation via dual auxiliary problems" (although one which has been more extensively studied historically). This should be swiftly followed with the recognition that choices of n greater than 1 will lead to more accurate approximations. So, the present discussion only considers how the ADAP methodology compares with another approximation technique which also consists of a similar two-stage solution process.

Collecting the $O(\varepsilon^0)$ terms in the differential equations resulting from the aforementioned steps, the generating problems can be written as the system depicted in Eq (3.49) and repeated here for clarity.

$$\begin{bmatrix} a_1 & 0 \\ 0 & b_1 \end{bmatrix} \begin{Bmatrix} \ddot{\alpha}_0 \\ \ddot{\beta}_0 \end{Bmatrix} + \begin{bmatrix} a_8 & 0 \\ 0 & b_8 \end{bmatrix} \begin{Bmatrix} \dot{\alpha}_0 \\ \dot{\beta}_0 \end{Bmatrix} = \begin{Bmatrix} F_v \\ F_h \end{Bmatrix}$$

Then, collecting the $O(\varepsilon^1)$ terms, the next required order of the system differential equations is determined. Note that the definitions of f_α and f_β in this system are shown here for simple linear Taylor Series approximations.

$$\begin{bmatrix} a_1 & 0 \\ 0 & b_1 \end{bmatrix} \begin{Bmatrix} \ddot{\alpha}_1 \\ \ddot{\beta}_1 \end{Bmatrix} + \begin{bmatrix} a_8 & 0 \\ 0 & b_8 \end{bmatrix} \begin{Bmatrix} \dot{\alpha}_1 \\ \dot{\beta}_1 \end{Bmatrix} = \begin{Bmatrix} f_\alpha \\ f_\beta \end{Bmatrix} \quad (3.68)$$

where

$$\begin{aligned}
 f_\alpha = & -\ddot{\beta}_0 \left((\tilde{a}_6 + \tilde{a}_7) \alpha_0 \left(\frac{\pi}{2} - \beta_0 \right) + \tilde{a}_4 \left(\alpha_0 - \beta_0 + \frac{\pi}{2} \right) + \tilde{a}_5 \left(-\alpha_0 - \beta_0 + \frac{\pi}{2} \right) - \tilde{a}_6 + \tilde{a}_7 \right) \\
 & - \tilde{a}_{10} (1 - \alpha_0^2) \dot{\alpha}_0^2 - \tilde{a}_{12} \alpha_0 \dot{\beta}_0^2 + \frac{\pi}{2} \tilde{a}_{14} \alpha_0 \dot{\beta}_0^2 - \frac{\pi}{2} \tilde{a}_{15} \alpha_0 \dot{\beta}_0^2 + \tilde{a}_9 \dot{\alpha}_0^2 - 2\tilde{a}_{11} \alpha_0 \dot{\alpha}_0^2 \\
 & - \tilde{a}_2 \ddot{\alpha}_0 - 2\tilde{a}_3 \alpha_0 \ddot{\alpha}_0 - \left(\frac{\pi}{2} \tilde{a}_{12} + \frac{\pi}{2} \tilde{a}_{13} - \tilde{a}_{14} + \tilde{a}_{15} \right) \dot{\beta}_0^2 + \tilde{a}_{13} \alpha_0 \dot{\beta}_0^2 \\
 & + \tilde{a}_{14} \alpha_0 \beta_0 \dot{\beta}_0^2 + \tilde{a}_{15} \alpha_0 \beta_0 \dot{\beta}_0^2 + \tilde{a}_2 \alpha_0^2 \ddot{\alpha}_0 + \tilde{a}_{12} \beta_0 \dot{\beta}_0^2 + \tilde{a}_{13} \beta_0 \dot{\beta}_0^2
 \end{aligned} \tag{3.69}$$

$$\begin{aligned}
 f_\beta = & \beta_0 \dot{\beta}_0^2 \left(\tilde{b}_{10} (\pi - \beta_0) + 2\tilde{b}_{11} \right) + \beta_0 \ddot{\beta}_0 \left(\tilde{b}_2 (\pi - \beta_0) + 2\tilde{b}_3 \right) \\
 & + \dot{\alpha}_0^2 \left((\tilde{b}_{14} + \tilde{b}_{15}) \alpha_0 \left(\beta_0 - \frac{\pi}{2} \right) + \tilde{b}_{13} \left(\alpha_0 + \beta_0 - \frac{\pi}{2} \right) \right. \\
 & \quad \left. + \tilde{b}_{12} \left(-\alpha_0 + \beta_0 - \frac{\pi}{2} \right) + \tilde{b}_{14} - \tilde{b}_{15} \right) \\
 & + \ddot{\alpha}_0 \left((\tilde{b}_6 + \tilde{b}_7) \alpha_0 \left(\beta_0 - \frac{\pi}{2} \right) + \tilde{b}_4 (\beta_0 - \alpha_0) + \tilde{b}_5 (\alpha_0 + \beta_0) \right) \\
 & + \left(-\frac{\pi}{2} \tilde{b}_4 - \frac{\pi}{2} \tilde{b}_5 + \tilde{b}_6 - \tilde{b}_7 \right) \ddot{\alpha}_0 + \left(-\tilde{b}_9 - \left(\frac{\pi^2}{4} - 1 \right) \tilde{b}_{10} - \pi \tilde{b}_{11} \right) \dot{\beta}_0^2
 \end{aligned} \tag{3.70}$$

Note that these terms f_α and f_β , acting as nonhomogeneous excitations in Eq (3.68), become considerably more unwieldy if higher orders of Taylor Series approximations are used for the underpinning trigonometric terms.

Crucially, note that the initial conditions of the system only pertain to the generating problem, Eq (3.49), and hence $\alpha_1(0) = \beta_1(0) = 0$ radians and $\dot{\alpha}_1(0) = \dot{\beta}_1(0) = 0$ rad/s.

The main challenge posed in solving the system of equations in this approach, Eqs (3.49) and (3.68), is determining the particular integrals associated with α_1 and β_1 . The complementary functions follow the same form as in the ADAP approach. Using *Mathematica*, the analytical solutions to the $O(\varepsilon^0)$ and $O(\varepsilon^1)$ perturbation problems are as follows. Note, to keep these expressions readable, the conditions for Case 8 in

Table 3.3 are applied.

$$\alpha_0(t) = 0.25e^{-1.25869t} (e^{1.25869t}t - 2.01621e^{1.25869t} + 0.794475) \quad (3.71)$$

$$\beta_0(t) = 0.2e^{-0.435257t} (e^{0.435257t}t + 5.55649e^{0.435257t} + 2.29749) \quad (3.72)$$

$$\begin{aligned} \alpha_1(t) = & 0.0199987t^3 - 0.172675t^2 + 1.06802t - 4.01361 \\ & + e^{-1.25869t} (-0.0238327t^2 + 0.0695735t - 46.3343) \\ & + e^{-0.870514t} (-0.223478t^2 + 2.58645t - 11.9093) \\ & + e^{-0.435257t} (0.417422t^2 + 0.155841t + 6.47338) \\ & + 0.0411636e^{-2.56446t} + e^{-1.69395t} (-0.161208t - 0.800723) \\ & + e^{-2.12921t} (0.032369t - 0.167907) \\ & + e^{-1.30577t} (2.82233t + 56.7113) \end{aligned} \quad (3.73)$$

$$\begin{aligned} \beta_1(t) = & e^{-3.77608t} (0.00231756 - 0.0011461t) + e^{-1.30577t} (0.32347 - 0.00275574t) \\ & + e^{-2.95264t} (0.0170154 - 0.0056251t) - 0.00208887e^{-4.21133t} \\ & + e^{-0.870514t} (-0.00179918t^2 + 0.562711t - 0.501977) \\ & + e^{-2.51738t} (-0.003472t^2 + 0.0159437t - 0.329679) \\ & + e^{-1.25869t} (0.0368372t^2 - 0.675147t + 1.46435) \\ & + e^{-0.435257t} (-0.01842t^3 + 0.274112t^2 + 0.559048t - 12.0377) \\ & + e^{-1.69395t} (0.0411406t + 0.350313) - 0.00158282e^{-1.74103t} \\ & + 10.7156 - 0.0144591t^3 + 0.320865t^2 - 4.09417t \end{aligned} \quad (3.74)$$

These expressions, Eqs (3.73) and (3.74), are the results against which the ADAP results are compared in the subsequent section.

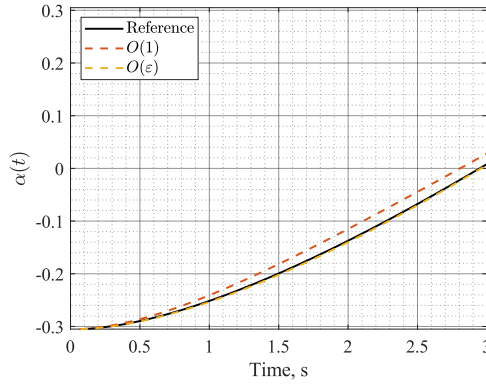
3.4 Implementation and simulation

To assess the quality of the ADAP methodology, a range of different test scenarios is examined. Nine of these scenarios are described in Table 3.3. In each of these cases, a constant actuation torque is applied. This contrasts the more realistic, time-varying actions which are provided via the control strategy in Chapter 6. For now, the constant actions are sufficient to permit inspection of manoeuvres spanning the end-effector's workspace.

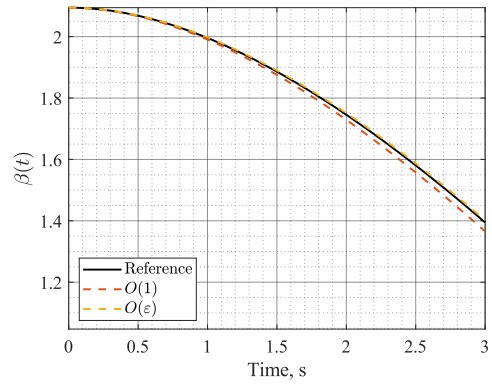
Case Number	Vertical actuation		Horizontal actuation	
	$\alpha(0)$	F_v	$\beta(0)$	F_h
1	-17.5°	1.50×10^4	120°	-5.50×10^4
2	-17.5°	1.50×10^4	120°	-2.75×10^4
3	-17.5°	2.50×10^4	60°	4.00×10^4
4	-17.5°	2.50×10^4	60°	2.00×10^4
5	-17.5°	1.25×10^4	60°	4.00×10^4
6	-17.5°	1.25×10^4	60°	2.00×10^4
7	0°	1.25×10^4	60°	4.00×10^4
8	-17.5°	2.50×10^4	90°	2.00×10^4
9	0°	1.25×10^4	90°	2.00×10^4

Table 3.3: Cases to be examined through testing the application of the ADAP methodology to the *FutureForge* manipulator dynamics.

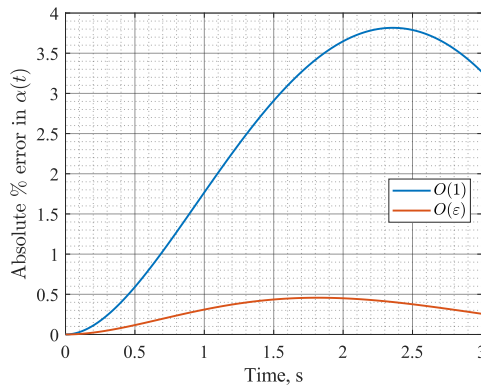
In Fig 3.28, the approximate solutions for the *FutureForge* manipulator dynamics are illustrated as obtained from the perturbation approach described in the preceding section. Over the limited manoeuvre duration depicted in these simulations, the absolute percentage error in this approximation method appears reasonably small. For a sense of scale, a 1% error in the approximation of $\alpha(t)$ would translate to a 6.1×10^{-3} radian (0.35°) mismatch between the calculated position and the actual position. In turn, this would result in an error in the end-effector position prediction of 3 mm. The errors observed for Case 1 in both $\alpha(t)$ and $\beta(t)$ are of a similar scale, while the two generalised coordinates sweep across roughly half of their respective operational ranges.



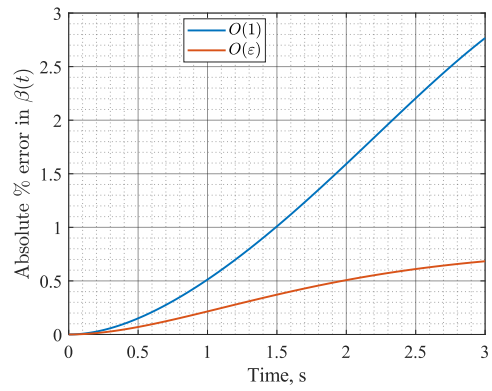
(a) Asymptotic approximations of the solution for $\alpha(t)$ in Case 1.



(b) Asymptotic approximations of the solution for $\beta(t)$ in Case 1.



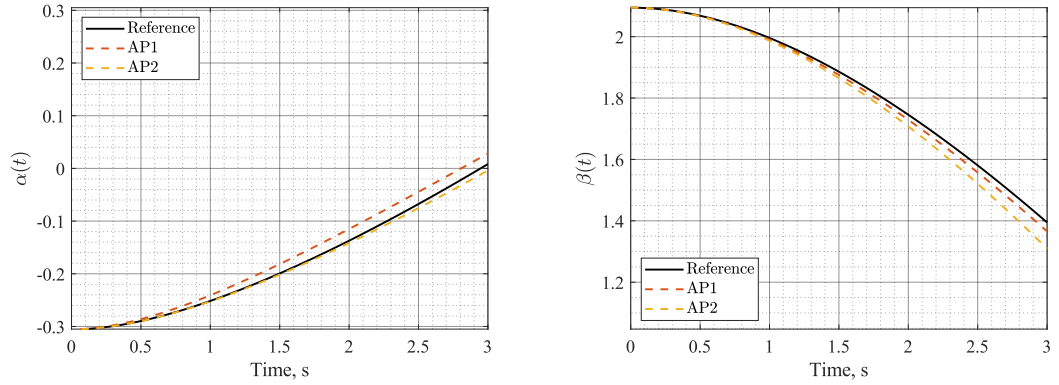
(c) Absolute percentage error in the asymptotic approximations of the solution for $\alpha(t)$ in Case 1.



(d) Absolute percentage error in the asymptotic approximations of the solution for $\beta(t)$ in Case 1.

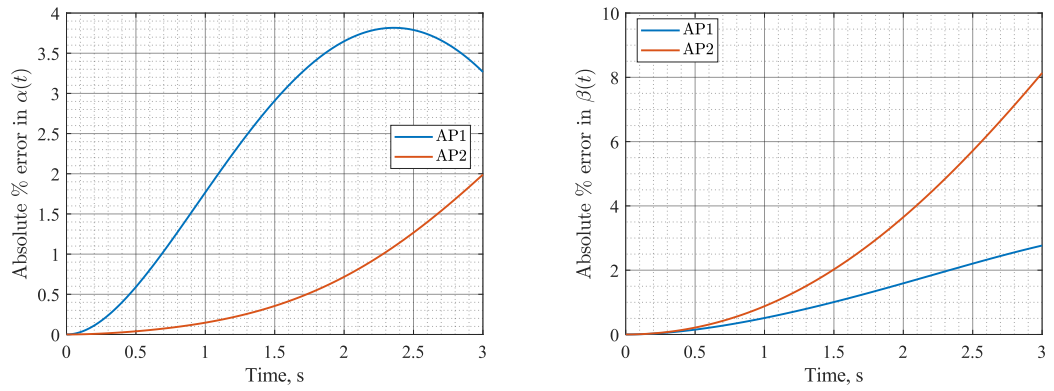
Figure 3.28: Asymptotic approximations of the solutions of the *FutureForge* manipulator system in Case 1 using 1st-order Taylor Series approximations.

In Fig 3.29, the solutions to the first and second auxiliary problems (“AP1” and “AP2” respectively) are illustrated and compared to the same numerical solution used in Fig 3.28. Note that, although the first auxiliary problem is simply an intermediate step on the way to obtaining the second auxiliary problem’s solution, AP1 represents the approach adopted by others previously, such as [70], and is hence a useful comparison.



(a) ADAP-based approximations of the solution for $\alpha(t)$ in Case 1.

(b) ADAP-based approximations of the solution for $\beta(t)$ in Case 1.

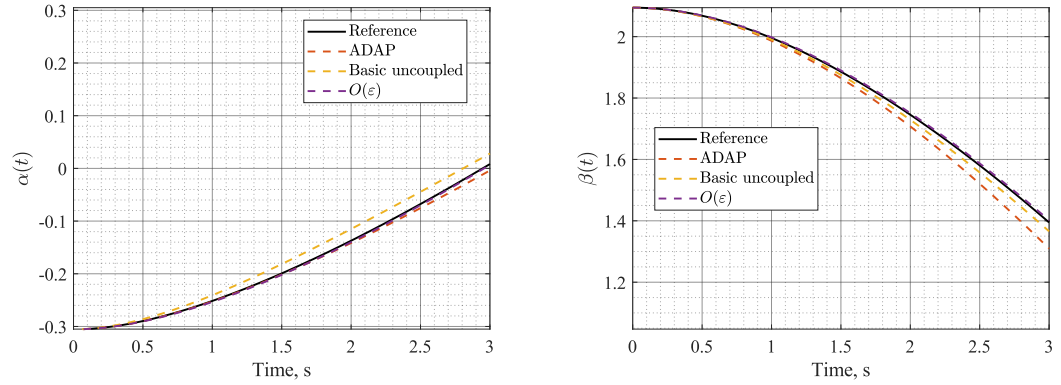


(c) Absolute percentage error in the ADAP-based approximations of the solution for $\alpha(t)$ in Case 1.

(d) Absolute percentage error in the ADAP-based approximations of the solution for $\beta(t)$ in Case 1.

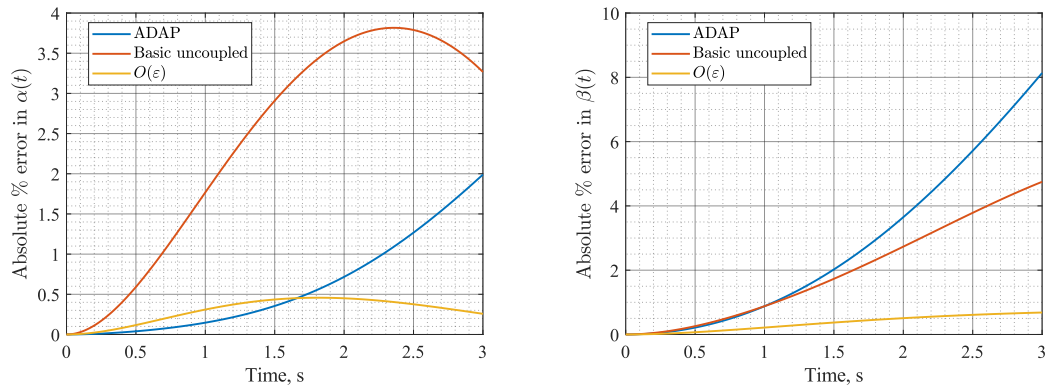
Figure 3.29: ADAP-based approximations of the solutions of the *FutureForge* manipulator system in Case 1 using 1st-order Taylor Series approximations.

Fig 3.30 compares the approximations achieved through the ADAP method, the perturbation-based approach, and the basic uncoupled approximation (which is simply “AP1” from Fig 3.29). Interestingly, the success of the ADAP method is significantly different across the $\alpha(t)$ and $\beta(t)$ approximations. As can be seen from Figs 3.29c and 3.29d, the ADAP method is the most accurate (over the manoeuvre duration) for $\alpha(t)$ but the least accurate for $\beta(t)$. This is discussed after the present cases’ results have been shown in full.



(a) Approximations of the solution for $\alpha(t)$ in Case 1.

(b) Approximations of the solution for $\beta(t)$ in Case 1.

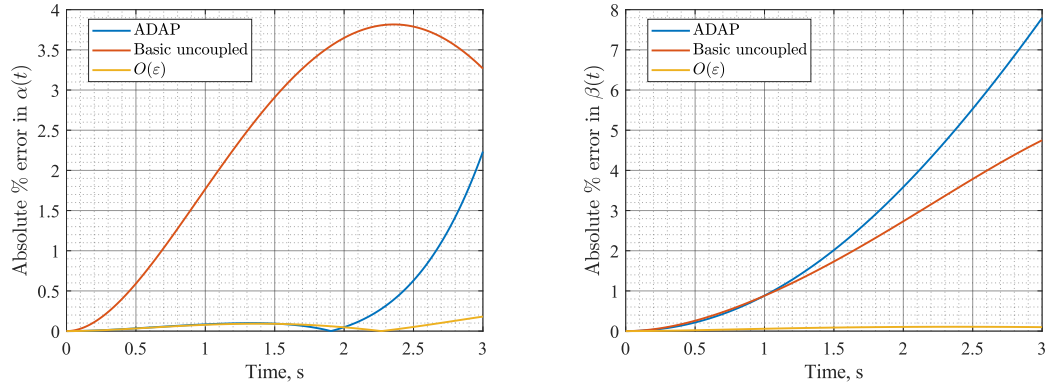


(c) Absolute percentage error in the approximations of the solution for $\alpha(t)$ in Case 1.

(d) Absolute percentage error in the approximations of the solution for $\beta(t)$ in Case 1.

Figure 3.30: A comparison between the asymptotic and ADAP-based approximations of the solutions of the *FutureForge* manipulator system in Case 1 using 1st-order Taylor Series approximations.

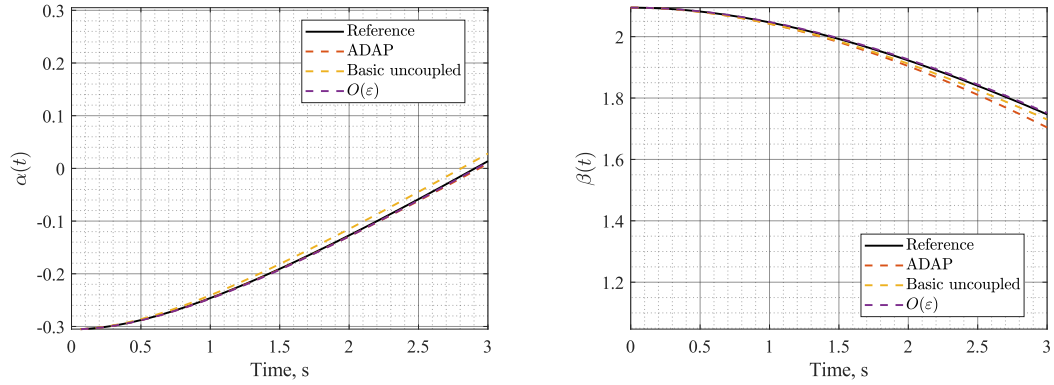
By increasing the order of Taylor Series approximation used in obtaining the $O(\varepsilon)$ and ADAP-based approximations, the accuracy of both of these is improved. The basic, uncoupled approximation is a useful reference for qualifying this improvement since it remains unaffected by changes in the order of Taylor Series approximations. In fact, in Fig 3.31, in the $t \in [0, 1.5]$ s interval for $\alpha(t)$, the ADAP and $O(\varepsilon)$ approximations are near-indistinguishable with the former performing better in the subsequent $t \in [1.5, 2]$ s interval before the $O(\varepsilon)$ approximation outperforms it later. While both approximations also improve for $\beta(t)$, the improvement for the ADAP method is significantly less than that of the $O(\varepsilon)$ approximation.



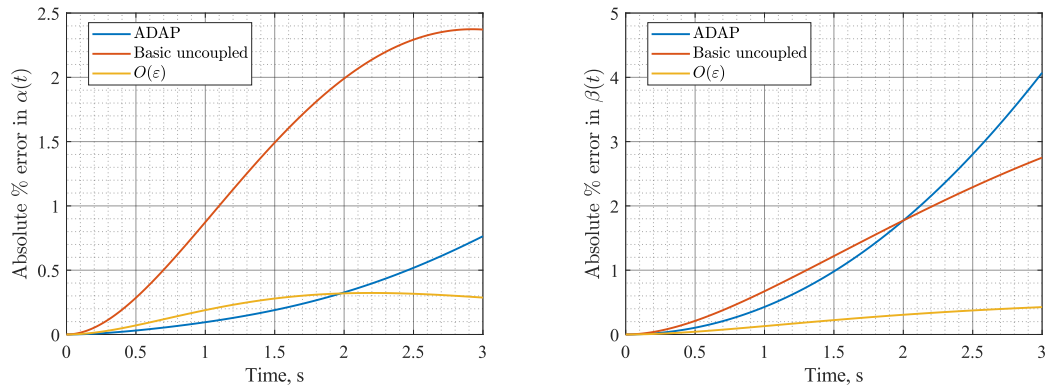
(a) Absolute percentage error in the approximations of the solution for $\alpha(t)$ in Case 1. (b) Absolute percentage error in the approximations of the solution for $\beta(t)$ in Case 1.

Figure 3.31: A comparison between the asymptotic and ADAP-based approximations of the solutions of the *FutureForge* manipulator system in Case 1 using 3rd-order Taylor Series approximations.

The difference between Case 1 and Case 2 can be observed by comparing Figs 3.30b and 3.32b. Case 2 features a reduced range of motion in the horizontal actuation coordinate, $\beta(t)$.



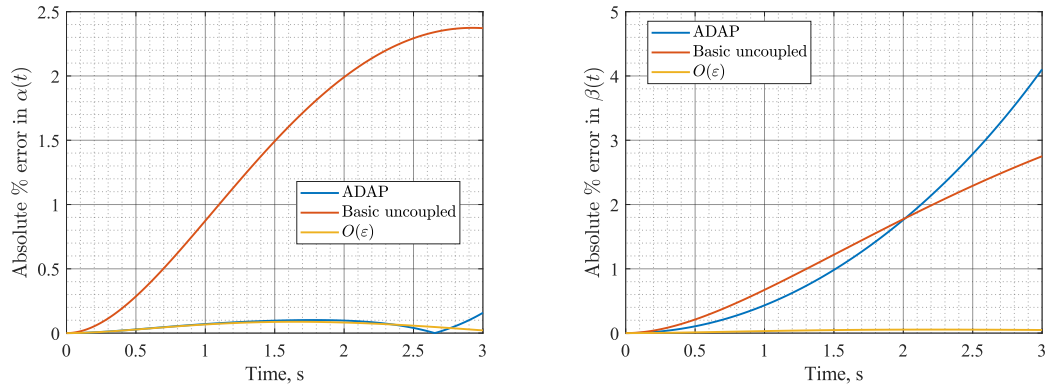
(a) Approximations of the solution for $\alpha(t)$ in Case 2. (b) Approximations of the solution for $\beta(t)$ in Case 2.



(c) Absolute percentage error in the approximations of the solution for $\alpha(t)$ in Case 2. (d) Absolute percentage error in the approximations of the solution for $\beta(t)$ in Case 2.

Figure 3.32: A comparison between the asymptotic and ADAP-based approximations of the solutions of the *FutureForge* manipulator system in Case 2 using 1st-order Taylor Series approximations.

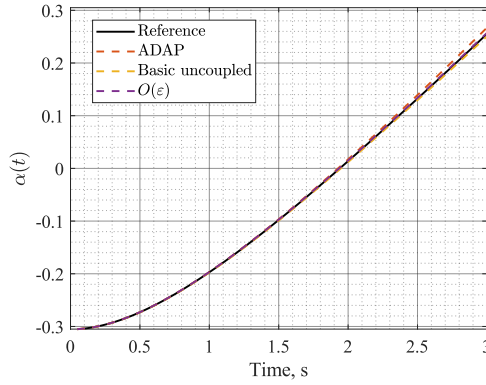
Interestingly, while the approximations of $\alpha(t)$ are substantially improved by increasing the order of Taylor Series approximation in use, the same cannot be said for the ADAP-based approximation of $\beta(t)$. While the $O(\varepsilon)$ approximation in Fig 3.33b sees a marked improvement compared to Fig 3.32d, the ADAP-based approximation remains imperceptibly different from the approximation yielded using the 1st-order Taylor Series approximations.



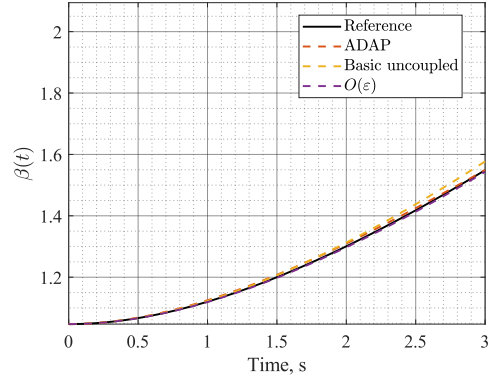
(a) Absolute percentage error in the approximations of the solution for $\alpha(t)$ in Case 2. (b) Absolute percentage error in the approximations of the solution for $\beta(t)$ in Case 2.

Figure 3.33: A comparison between the asymptotic and ADAP-based approximations of the solutions of the *FutureForge* manipulator system in Case 2 using 3rd-order Taylor Series approximations.

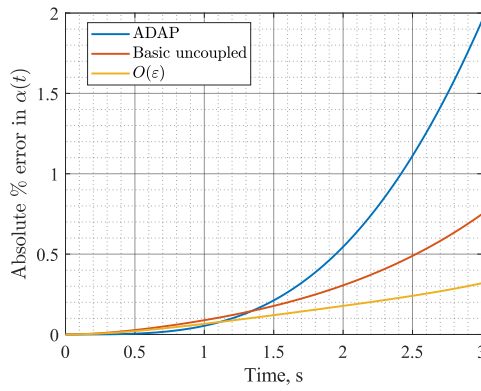
In Case 3, the manoeuvre spans the full operational range of $\alpha(t)$ and half the operational range of $\beta(t)$.



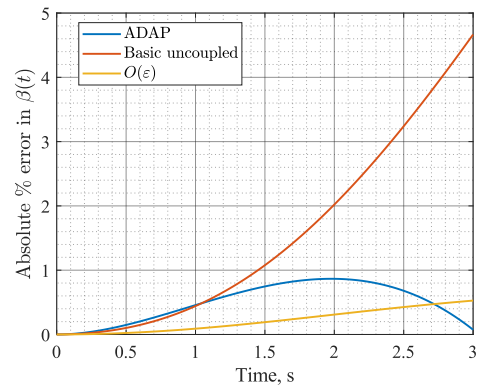
(a) Approximations of the solution for $\alpha(t)$ in Case 3.



(b) Approximations of the solution for $\beta(t)$ in Case 3.



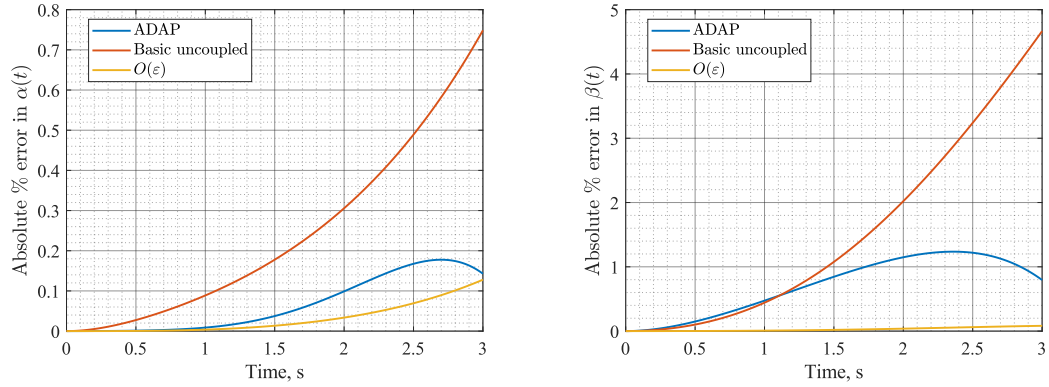
(c) Absolute percentage error in the approximations of the solution for $\alpha(t)$ in Case 3.



(d) Absolute percentage error in the approximations of the solution for $\beta(t)$ in Case 3.

Figure 3.34: A comparison between the asymptotic and ADAP-based approximations of the solutions of the *FutureForge* manipulator system in Case 3 using 1st-order Taylor Series approximations.

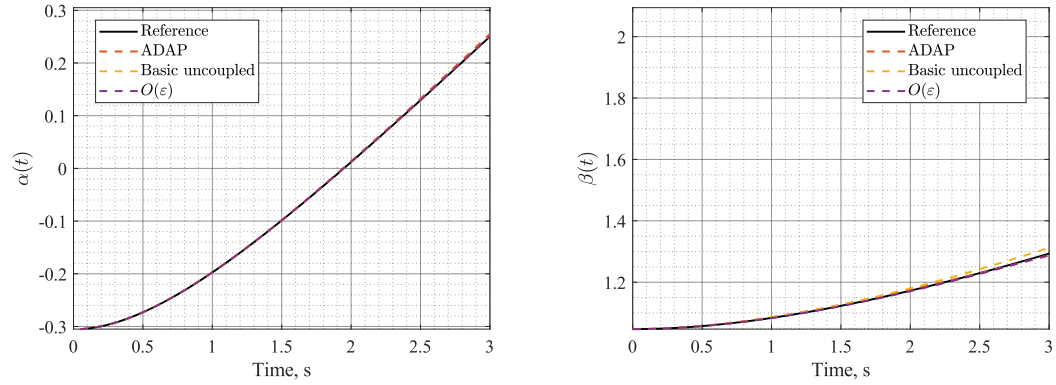
In this manoeuvre, the absolute percentage error observed in the ADAP-based approximation of $\alpha(t)$ exceeds that of either of the other approximations illustrated beyond 1.4 seconds in the time domain. Additionally, the rate of change of this error is substantially more positive than that of the other approximations. However, notably, the ADAP-based approximation has the lowest error of the three until this point.



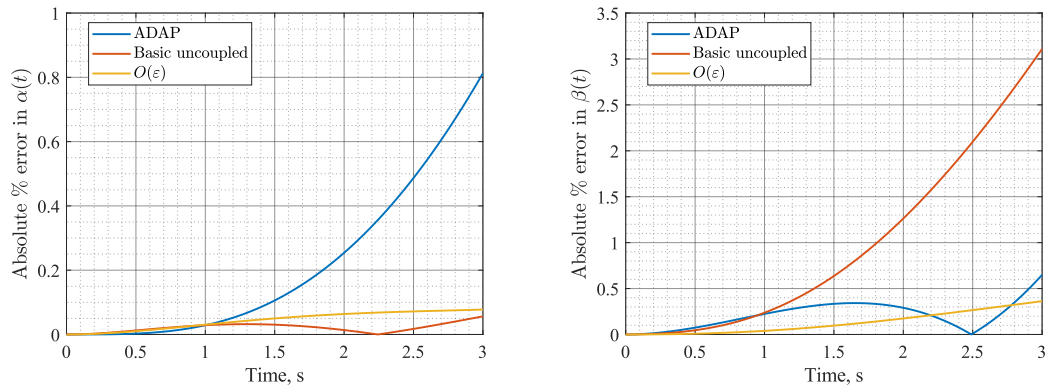
(a) Absolute percentage error in the approximations of the solution for $\alpha(t)$ in Case 3. (b) Absolute percentage error in the approximations of the solution for $\beta(t)$ in Case 3.

Figure 3.35: A comparison between the asymptotic and ADAP-based approximations of the solutions of the *FutureForge* manipulator system in Case 3 using 3rd-order Taylor Series approximations.

In Case 4, the manoeuvre's displacement through $\beta(t)$ is reduced by comparison to Case 3.



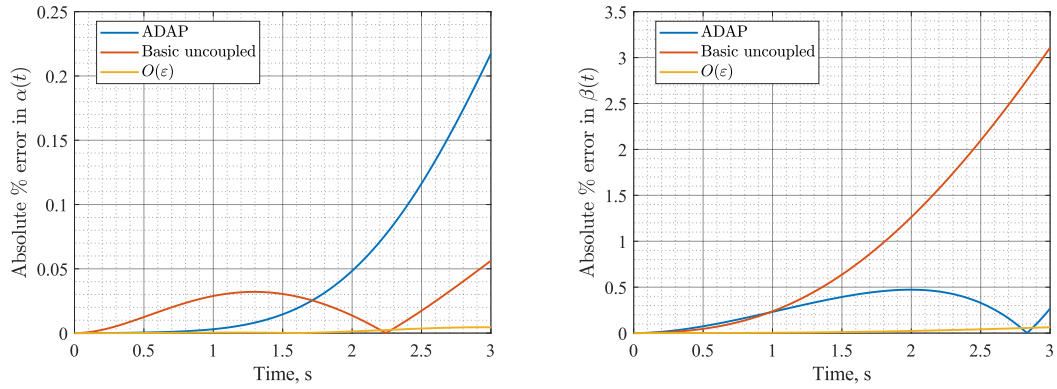
(a) Approximations of the solution for $\alpha(t)$ in Case 4. (b) Approximations of the solution for $\beta(t)$ in Case 4.



(c) Absolute percentage error in the approximations of the solution for $\alpha(t)$ in Case 4. (d) Absolute percentage error in the approximations of the solution for $\beta(t)$ in Case 4.

Figure 3.36: A comparison between the asymptotic and ADAP-based approximations of the solutions of the *FutureForge* manipulator system in Case 4 using 1st-order Taylor Series approximations.

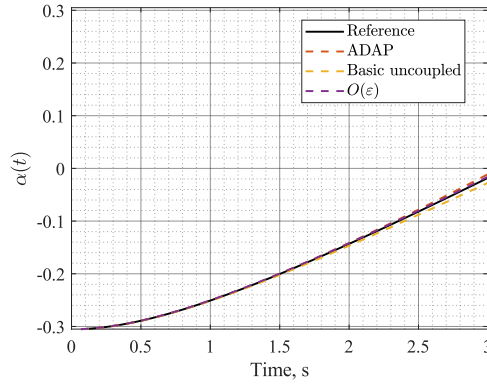
As depicted in Fig 3.36c and 3.37a, the error in the ADAP-based approximation of $\alpha(t)$ is the largest (and fastest growing) of the three methods shown. Noting the scale of the vertical axes in these two graphs, it should be noted that all three perform “well” in this manoeuvre, however. Additionally, the performance of the ADAP-based method can be notably improved by increasing the order of Taylor Series approximations used.



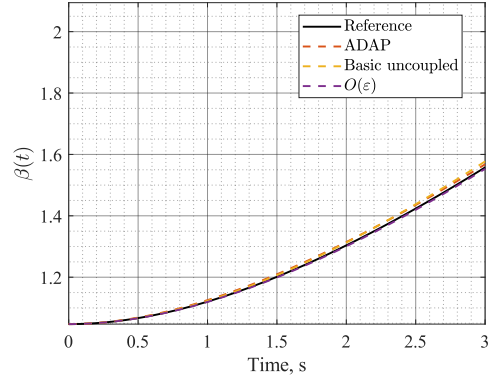
(a) Absolute percentage error in the approximations of the solution for $\alpha(t)$ in Case 4. (b) Absolute percentage error in the approximations of the solution for $\beta(t)$ in Case 4.

Figure 3.37: A comparison between the asymptotic and ADAP-based approximations of the solutions of the *FutureForge* manipulator system in Case 4 using 3rd-order Taylor Series approximations.

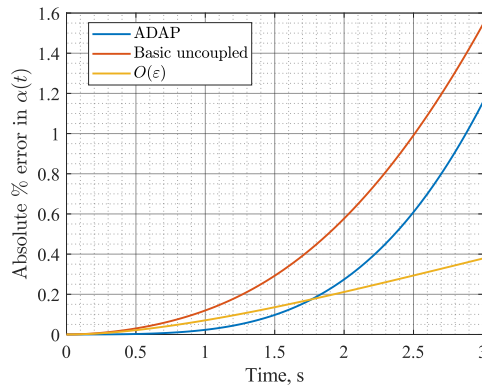
In Case 5 the manoeuvre is comparable to Case 3, but the $\alpha(t)$ range is reduced to half the value of Case 3. Observing the errors associated with the approximations of the two generalised coordinates, the ADAP-based approximation is apparently benefitted by the reduced range of $\alpha(t)$. However, the error observed for $\beta(t)$ is increased for the ADAP-based approximation (more so than the two other approximations).



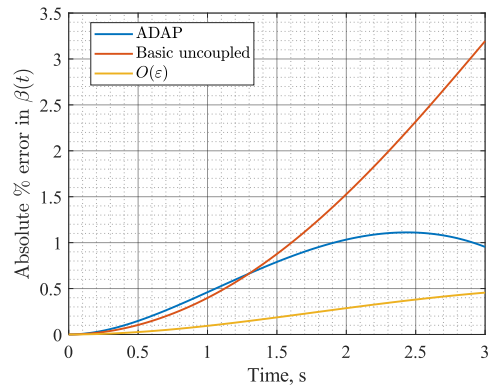
(a) Approximations of the solution for $\alpha(t)$ in Case 5.



(b) Approximations of the solution for $\beta(t)$ in Case 5.



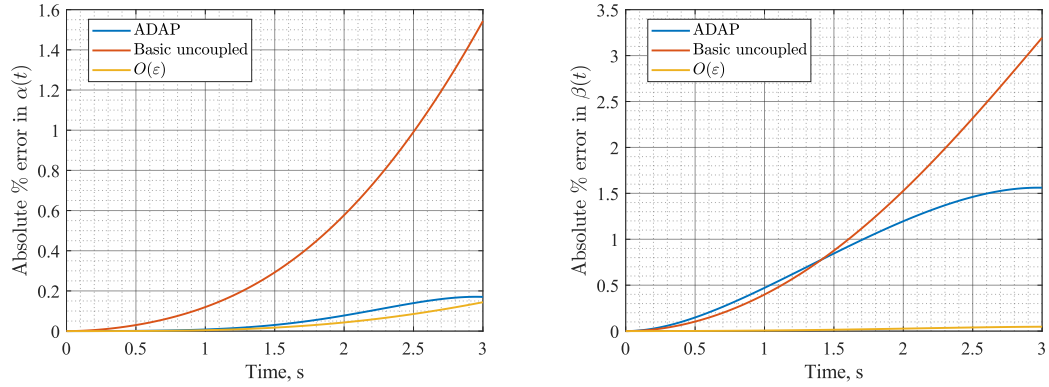
(c) Absolute percentage error in the approximations of the solution for $\alpha(t)$ in Case 5.



(d) Absolute percentage error in the approximations of the solution for $\beta(t)$ in Case 5.

Figure 3.38: A comparison between the asymptotic and ADAP-based approximations of the solutions of the *FutureForge* manipulator system in Case 5 using 1st-order Taylor Series approximations.

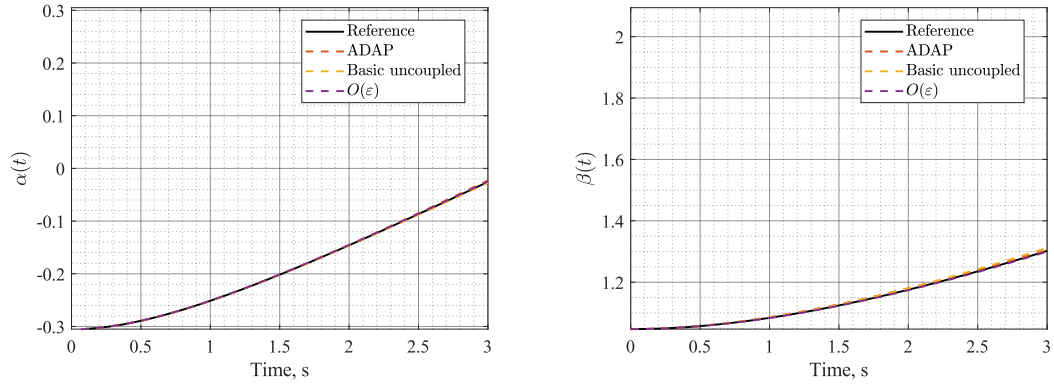
By increasing the order of Taylor Series approximation used to obtain the ADAP-based and $O(\varepsilon)$ approximations, there is a clear and substantial improvement in the approximations of $\alpha(t)$. However, while there is an improvement also noted for the $O(\varepsilon)$ approximation of $\beta(t)$, there is a notable reduction in accuracy for the ADAP-based approximation.



(a) Absolute percentage error in the approximations of the solution for $\alpha(t)$ in Case 5. (b) Absolute percentage error in the approximations of the solution for $\beta(t)$ in Case 5.

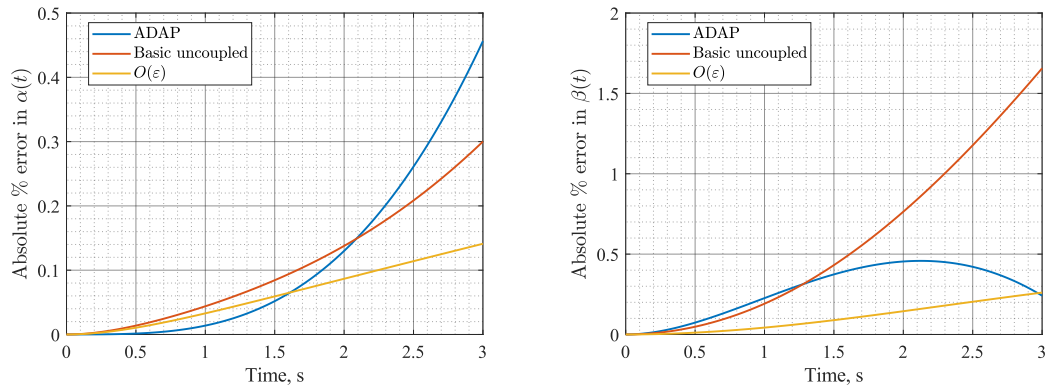
Figure 3.39: A comparison between the asymptotic and ADAP-based approximations of the solutions of the *FutureForge* manipulator system in Case 5 using 3rd-order Taylor Series approximations.

The Case 6 manoeuvre features the same range of motion in $\alpha(t)$ and a reduced range of motion in $\beta(t)$ by comparison to Case 5. The ADAP-based approximation shows the highest accuracy, over the initial part of this manoeuvre, for $\alpha(t)$; but, while it subsequently reaches a larger error than the other approximations, it should be noted that the range of percentage errors in this case are smaller than in Case 5 (for each approximation relative to its counterpart in Fig 3.38).



(a) Approximations of the solution for $\alpha(t)$ in Case 6.

(b) Approximations of the solution for $\beta(t)$ in Case 6.

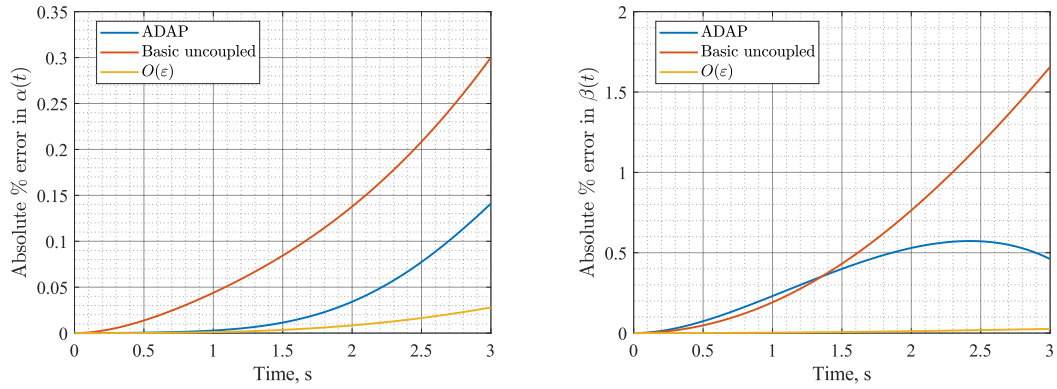


(c) Absolute percentage error in the approximations of the solution for $\alpha(t)$ in Case 6.

(d) Absolute percentage error in the approximations of the solution for $\beta(t)$ in Case 6.

Figure 3.40: A comparison between the asymptotic and ADAP-based approximations of the solutions of the *FutureForge* manipulator system in Case 6 using 1st-order Taylor Series approximations.

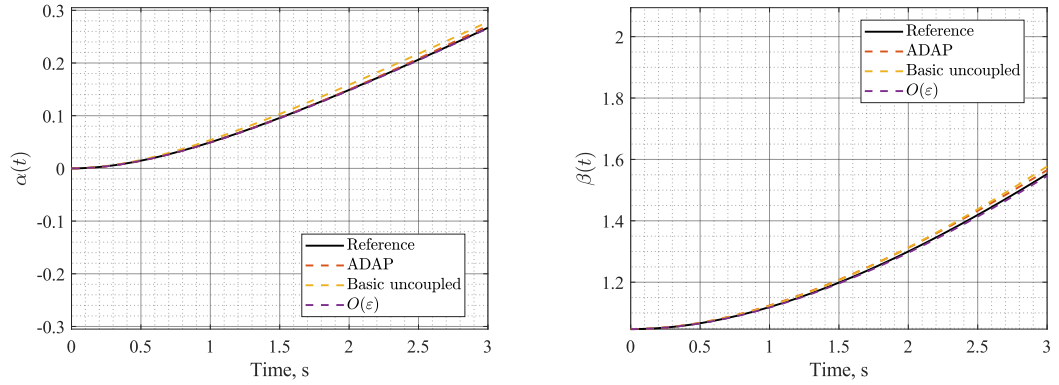
As in many of the previous figures shown, an increase in the orders of Taylor Series approximations results in improved approximations for $\alpha(t)$. However, in this case, for $\beta(t)$, this is not observed - the error for the ADAP-based approximation is actually increased in Fig 3.41b relative to 3.40d.



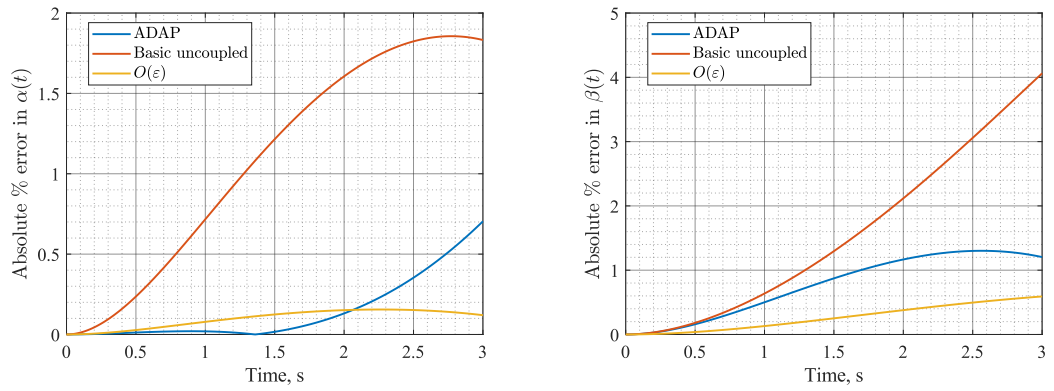
(a) Absolute percentage error in the approximations of the solution for $\alpha(t)$ in Case 6. (b) Absolute percentage error in the approximations of the solution for $\beta(t)$ in Case 6.

Figure 3.41: A comparison between the asymptotic and ADAP-based approximations of the solutions of the *FutureForge* manipulator system in Case 6 using 3rd-order Taylor Series approximations.

Case 7 sees a similar range of motion to Case 5 except that $\alpha(t)$ is varied between its midpoint and upper limit, instead of between the lower limit and midpoint. In keeping with this similarity, the error observed for all three approximations of $\beta(t)$ is very close to those observed in Case 5. The errors observed in the approximations of $\alpha(t)$ are also similar to Case 5, although the initial accuracy of the ADAP-based approximation is particularly improved. Since the Taylor Series approximations are centred at $\alpha(t) = 0$ radians, this may contribute to this observation.



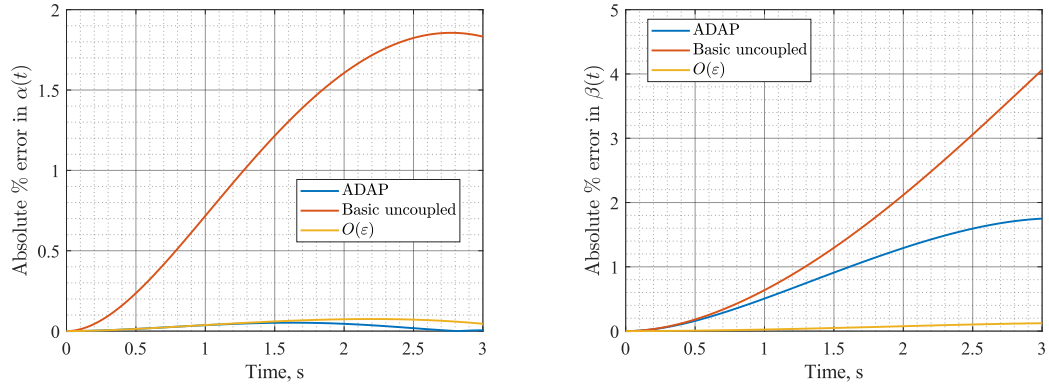
(a) Approximations of the solution for $\alpha(t)$ in Case 7. (b) Approximations of the solution for $\beta(t)$ in Case 7.



(c) Absolute percentage error in the approximations of the solution for $\alpha(t)$ in Case 7. (d) Absolute percentage error in the approximations of the solution for $\beta(t)$ in Case 7.

Figure 3.42: A comparison between the asymptotic and ADAP-based approximations of the solutions of the *FutureForge* manipulator system in Case 7 using 1st-order Taylor Series approximations.

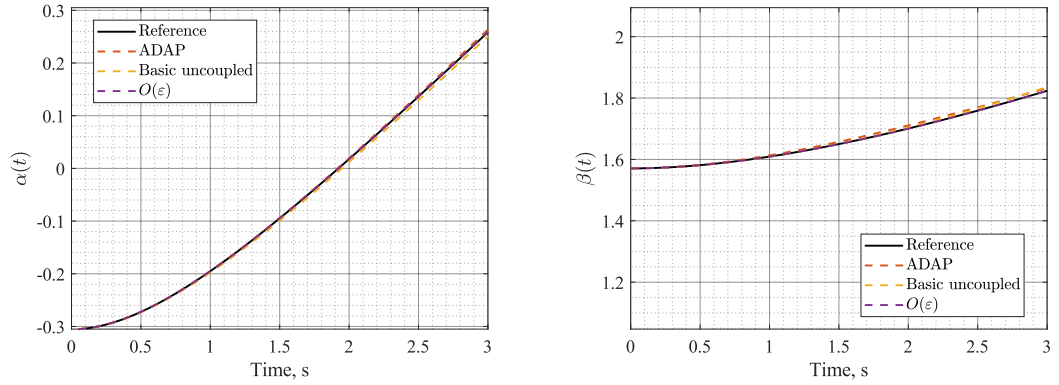
The process of increasing the order of Taylor Series approximations in use has a similar effect to previous cases. However, a notable difference here is that the ADAP-based approximation of $\alpha(t)$ is more accurate than either alternative over the entire time domain shown.



(a) Absolute percentage error in the approximations of the solution for $\alpha(t)$ in Case 7. (b) Absolute percentage error in the approximations of the solution for $\beta(t)$ in Case 7.

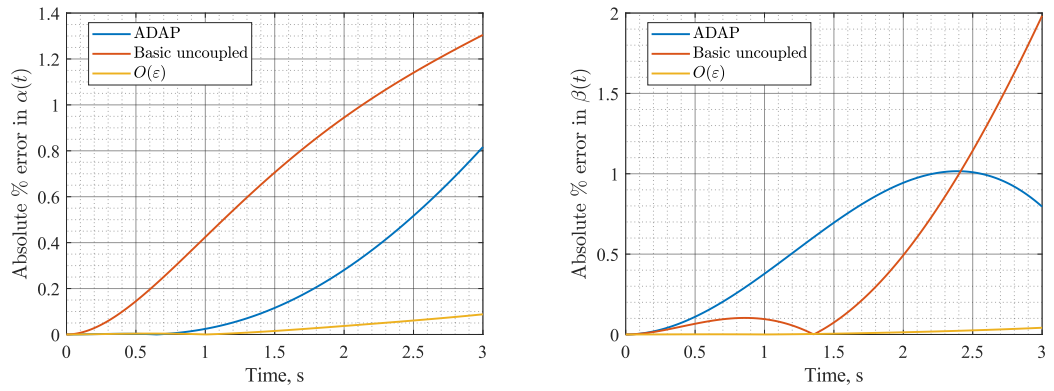
Figure 3.43: A comparison between the asymptotic and ADAP-based approximations of the solutions of the *FutureForge* manipulator system in Case 7 using 3rd-order Taylor Series approximations.

Case 8 (Figs 3.44 and 3.45) is similar to Case 4 (Figs 3.36 and 3.37) in that it covers a majority of the operational range of $\alpha(t)$ and only a small interval in $\beta(t)$. The difference is that the interval in $\beta(t)$ begins near the operational range's midpoint rather than at its lower limit (as in Case 4).



(a) Approximations of the solution for $\alpha(t)$ in Case 8.

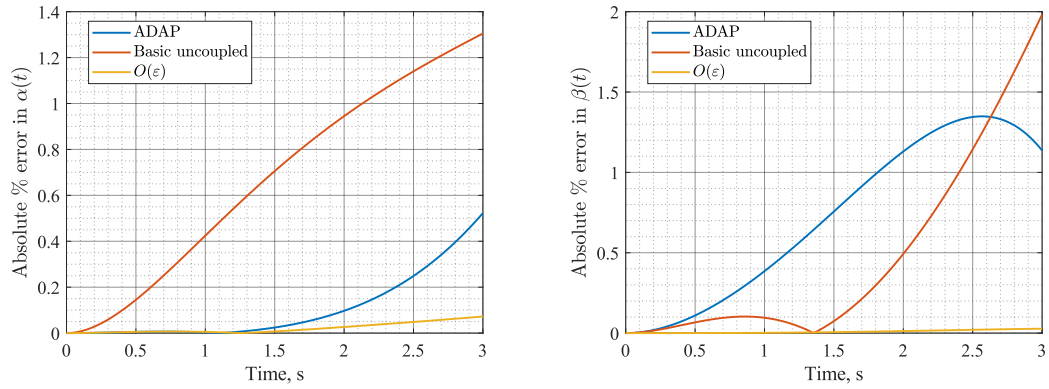
(b) Approximations of the solution for $\beta(t)$ in Case 8.



(c) Absolute percentage error in the approximations of the solution for $\alpha(t)$ in Case 8.

(d) Absolute percentage error in the approximations of the solution for $\beta(t)$ in Case 8.

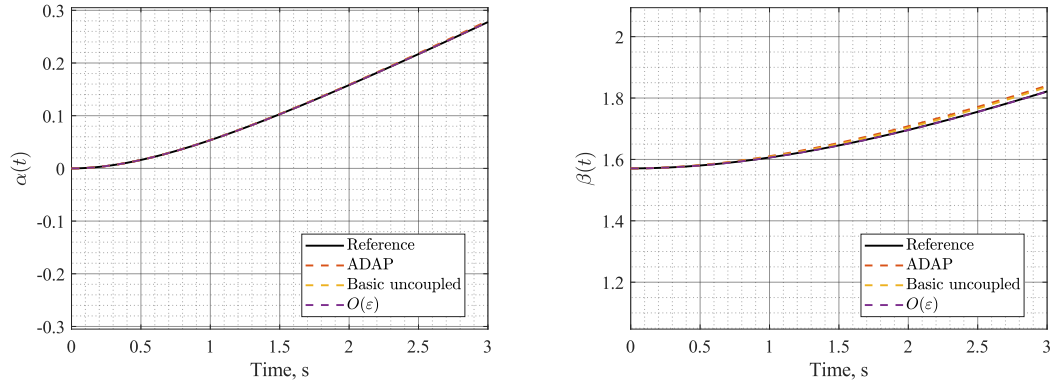
Figure 3.44: A comparison between the asymptotic and ADAP-based approximations of the solutions of the *FutureForge* manipulator system in Case 8 using 1st-order Taylor Series approximations.



(a) Absolute percentage error in the approximations of the solution for $\alpha(t)$ in Case 8. (b) Absolute percentage error in the approximations of the solution for $\beta(t)$ in Case 8.

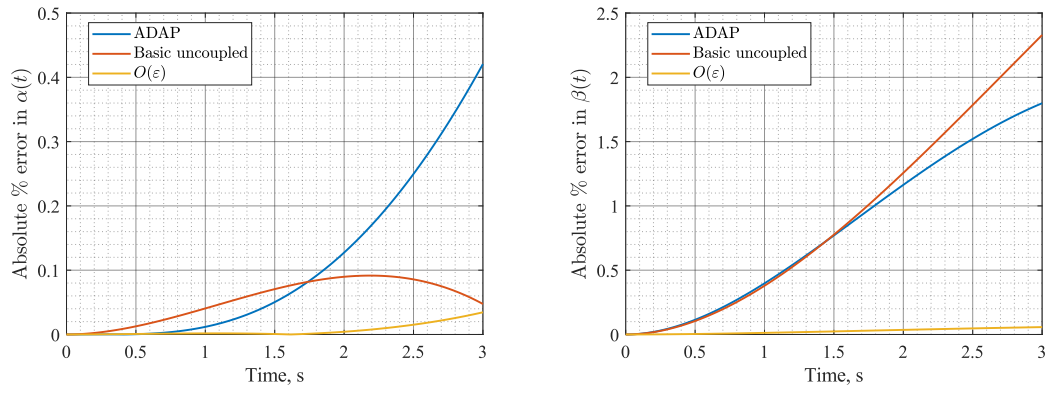
Figure 3.45: A comparison between the asymptotic and ADAP-based approximations of the solutions of the *FutureForge* manipulator system in Case 8 using 3rd-order Taylor Series approximations.

Lastly, Case 9 (Figs 3.46 and 3.47) differs from Case 8 in that the interval in $\alpha(t)$ is halved, but the interval in $\beta(t)$ remains the same.



(a) Approximations of the solution for $\alpha(t)$ in Case 9.

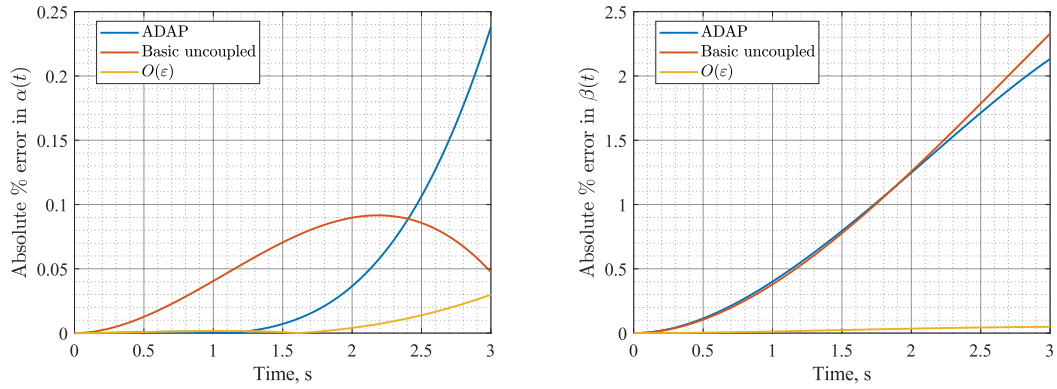
(b) Approximations of the solution for $\beta(t)$ in Case 9.



(c) Absolute percentage error in the approximations of the solution for $\alpha(t)$ in Case 9.

(d) Absolute percentage error in the approximations of the solution for $\beta(t)$ in Case 9.

Figure 3.46: A comparison between the asymptotic and ADAP-based approximations of the solutions of the *FutureForge* manipulator system in Case 9 using 1st-order Taylor Series approximations.



(a) Absolute percentage error in the approximations of the solution for $\alpha(t)$ in Case 9. (b) Absolute percentage error in the approximations of the solution for $\beta(t)$ in Case 9.

Figure 3.47: A comparison between the asymptotic and ADAP-based approximations of the solutions of the *FutureForge* manipulator system in Case 9 using 3rd-order Taylor Series approximations.

3.5 Discussion

Given the subject of this thesis is the study of the novel ADAP methodology, the present discussion of the illustrated results centres on the performance of the ADAP-based approximations. From an objective perspective, the favoured solution approach - if accuracy is the sole consideration and only the most accurate of the three presented approaches may be chosen - is the asymptotic approximation approach. It is important to acknowledge this in this particular case study. While asymptotic approximation does not perform as well as the ADAP approach in the examples demonstrated in Chapter 2, there are clearly features of the *FutureForge* manipulator system which are better suited to this approximation technique. This does not render the ADAP methodology useless; rather, it gives a boundary within which to discuss the merits of ADAP-based approximations more generally.

There are circumstances under which the ADAP-based approximation of $\alpha(t)$ appears to be more accurate, at least under a very limited time domain. Examples of this are illustrated by Cases 1 through 7. When seeking to understand why Cases 8

and 9 do not continue this trend, the differences between these cases are considered. The manoeuvre through $\alpha(t)$ for Case 8 (Fig 3.44a) is replicated in Cases 3 and 4 (Figs 3.34a and 3.36a, respectively); and the manoeuvre through $\alpha(t)$ for Case 9 (Fig 3.46a) is replicated in Case 7 (Fig 3.42a). The manoeuvres through $\beta(t)$ for Cases 8 and 9 (Figs 3.44b and 3.46b) are the only common aspects of both cases which are not replicated by other preceding cases.

In Cases 8 and 9, the initial value for $\beta(t)$ is at the midpoint of its operational range: $\frac{\pi}{2}$ radians. Considering why an approximation generally may find reduced accuracy in the vicinity of this operating point, it seems intuitive to pose that this is likely due to the poor suitability of a Taylor Series approximation for $\sin(\beta(t))$. With such a low order of approximation, this term is comparably unsuitably approximated by comparison to $\cos(\beta(t))$ which is more suitably approximated as linear around this operating point. To examine this hypothesised reason for the reduced accuracy in these cases, the coefficients of $\sin(\beta(t))$ and $\cos(\beta(t))$ are compared. Once the addition formulae have been applied to arrive at expressions containing $\sin(\beta(t))$ and $\cos(\beta(t))$, their coefficients are extracted (before any case-specific initial conditions are applied).

$$\text{Coefficients of } \sin(\beta(t)) \text{ in } M = \begin{bmatrix} 0 & 3083.9 \sin(\alpha(t) + 0.2366) \\ 3083.9 \sin(\alpha(t) + 0.2366) & 1957.77 \cos(\beta(t)) \end{bmatrix} \quad (3.75)$$

$$\text{Coefficients of } \sin(\beta(t)) \text{ in } P = \begin{bmatrix} 0 & -2335.6 \sin(\alpha(t) + 1.5513) \\ -3051.8 \sin(\alpha(t) - 1.545) & 5225.65 \cos(\beta(t)) \end{bmatrix} \quad (3.76)$$

For comparison, the matrices (3.77) and (3.78) contain the coefficients of $\cos(\beta(t))$.

$$\text{Coefficients of } \cos(\beta(t)) \text{ in } M = \begin{bmatrix} 0 & 1338.2 \sin(\alpha(t) + 1.5513) \\ 1338.2 \sin(\alpha(t) + 1.5513) & 1957.77 \sin(\beta(t)) \end{bmatrix} \quad (3.77)$$

$$\text{Coefficients of } \cos(\beta(t)) \text{ in } P = \begin{bmatrix} 0 & 5237.4 \sin(\alpha(t) + 0.0258) \\ -1361 \sin(\alpha(t) - 0.0195) & 5225.65 \sin(\beta(t)) \end{bmatrix} \quad (3.78)$$

The coefficients in Eqs (3.75)-(3.78) are obtained by substituting the coefficients listed in Appendix A. Notably, comparing matrices (3.75) and (3.77), the coefficients of $\sin(\beta(t))$ vary more greatly (with $\alpha(t)$) than the coefficients of $\cos(\beta(t))$. The same is true for the lower left elements of (3.76) and (3.78), while the upper right elements of these matrices don't follow this pattern. Nonetheless, this greater magnitude of $\sin(\beta(t))$ terms, by comparison to $\cos(\beta(t))$ terms, is a potential factor in reducing the accuracy of the demonstrated approximations in the vicinity of $\beta(t) = \frac{\pi}{2}$ radians.

In this Chapter, the ADAP method has been applied to the *FutureForge* manipulator system and assessed against the Lindstedt-Poincaré method. This system will also be the subject of Chapter 6. At present, before proceeding to develop a generalised architecture for embedding the ADAP method in control strategies, the contribution of this Chapter is the demonstration and assessment of the ADAP method as a means of solving a system with a high-degree of coupling and high-strength nonlinearities. The conclusion drawn from the results and discussion of this Chapter are that the ADAP method presents comparable performance to the Lindstedt-Poincaré method. As such, it is further developed in this thesis, exploiting a novel means of integrating it with model-predictive control.

Chapter 4

Introduction to model-predictive control and industrial manipulators

As outlined in the introduction of this thesis in Chapter 1, see Fig 1.1, the second half of this thesis, spanning Chapters 4-6, focuses on the integration of the ADAP method and model-predictive control, culminating in a case study application to the *FutureForge* manipulator. As such, this chapter introduces two key concepts foundational to the second half of this thesis: model-predictive control and industrial manipulators. An introduction to model-predictive control is given in Section 4.1, wherein a description of its function and key terms is given alongside an overview of its applications, benefits, and limitations. A general overview of industrial manipulators is then undertaken in Section 4.2, which outlines the most common architectures – a pertinent topic for understanding the present work’s focus on the *FutureForge* manipulator. Lastly, a review of control approaches for industrial manipulators is undertaken in Section 4.3.

4.1 Fundamentals of model-predictive control

Model-predictive control (MPC) is particularly valued for its ability to handle system constraints and multivariable interactions within a unified framework [71][72]. The im-

plementation of MPC is based on three key elements: a process prediction model, a cost function to be optimised, and an optimisation algorithm to solve for the control action [73]. The strategy is distinguished by its emphasis on process modelling, which "shifts the effort for the design of a controller towards modelling of the to-be-controlled process" [71]. While this approach carries a significant computational burden - a critical drawback in its early development — its implementation in complex applications has been facilitated by advances in computational power afforded by modern microprocessors [74][73].

Although early concepts emerged in the 1960s, MPC saw a resurgence in the 2000s [75]. It was initially developed for and almost exclusively employed within the process industry, particularly the petrochemical sector, where methods such as Dynamic Matrix Control (DMC) were pioneered. Unlike DMC, which was restricted to linear process models, MPC's flexibility in the choice of model formulation was a key reason for its subsequent success [71]. Today, the strategy is employed across a diverse range of fields. Prominent applications include building energy management to reduce consumption while maintaining occupant comfort [70]; water resource management [76]; power systems for the control of power converters [77][74][73][75]; and, in robotics and autonomous systems for tasks such as improving tracking robustness and managing vehicle power systems [78][72]. Despite its wide applicability, the implementation of MPC presents significant challenges. A primary obstacle is the requirement for an accurate system model; indeed, the lack of a suitable model is often cited as a major impediment to industrial application. Furthermore, MPC is not inherently more robust than classical feedback controllers, and guaranteeing stability remains an open research topic [71][73].

MPC is a well-established field of research in control systems. To understand how

the strategy works, consider the following generalised system

$$\begin{aligned}\dot{\mathbf{x}} &= f(\mathbf{x}) + g(\mathbf{x})\mathbf{u} \\ \mathbf{y} &= h(\mathbf{x})\end{aligned}\tag{4.1}$$

in which a controller is a mechanism for determining the vector of controllable actions, $\mathbf{u}$, such that the output variables, $\mathbf{y}$, track a target setpoint. In the terminology typically used in MPC: the control actions, $\mathbf{u}$, are referred to as “manipulated variables” while terms in the output vector, $\mathbf{y}$, are referred to as the “measured outputs”. A model-predictive controller determines the manipulated variables such that a cost function is minimised over a given prediction time frame. In operation, the MPC algorithm uses a model of the system dynamics and the associated output equation to anticipate how the system will behave over a given time frame. This time frame is constrained and is referred to as the “prediction horizon”. The MPC then evaluates an appropriate set of values for $\mathbf{u}$ over a smaller horizon known as the “control horizon” such that the system reaches its target states. Once each time step has elapsed, the controller re-evaluates this prediction and adjusts the control actions to account for any deviations from the previously predicted response. This continual shifting of the horizon is known as the “receding horizon principle”. Note, in the literature surrounding model-predictive control, there can be some discrepancy in the precise definitions of control and prediction horizons. Some authors define horizons as time intervals, while others define horizons in terms of the number of time steps contained within this horizon. This is a relatively trivial, although nuanced, point. However, for clarity throughout the present work, the latter definition is used. That is to say, horizons are defined as whole numbers (excluding zero). For example, if an MPC has a time step of 0.1 seconds, the prediction horizon may be set to 15 such that 1.5 seconds of the system behaviour is predicted at each time step. Thus, it is seldom appropriate to discuss the horizons without also discussing the time step to be used. An analogy can be drawn between horizons and time step sizes, and the concepts of sampling periods and frequencies – as this similarly describes the relationship between these parameters. Appropriate tuning of these parameters, amongst others, is crucial for effective model-predictive control. There are

numerous contributions to this field – some of which will be examined and recounted as needed in the subsequent material – whether they be individual applications of this control strategy to unique and novel problems, or whether they be efforts to review and encapsulate the sum of work across the continually-growing field.

4.1.1 Strengths and weaknesses of model-predictive control

MPC has seen growing use in the past decades. This has particularly accelerated with the growth in computational power of relatively affordable hardware. Nonetheless, the computational burden of this control strategy is still high enough that real-time deployment is challenging [79][80]. Given the controller, in this strategy, is required to solve a finite-horizon optimisation problem at every time step, the computational burden can quickly become unfeasibly high for systems with fast dynamics, long prediction horizons, or especially complex nonlinear models. Regarding the first of these three cases, this relates to the computational burden as it necessitates the use of smaller time steps, which results in a larger number of steps being considered in the control optimisation problem over the horizon. Adjacent to this issue is the increased time required for the optimisation problem to be solved in the presence of notably restrictive constraints [80].

This computational burden, however, is intrinsically linked to a key benefit of the strategy. Contrasting many, more well-established, control techniques, MPC can handle constraints within the optimisation of each control horizon. This capacity to account for constraints means MPC systems are typically able to better incorporate the realistic operating conditions and physical limitations of the system [80]. Consequently, in certain contexts, MPC offers significant advantages over simpler techniques, such as PID control, despite the disadvantages discussed in this present work [81]. Due to this ability to account for constraints, the range of applications is often considered more flexible and adaptable [82][81]. This is further aided by the feasibility of integrating MPC with other control strategies (a process which brings MPC from [83]’s “hard control” category to their “hybrid control” category).

The wide array of hybrid control developments, of which MPC is a constituent part, is proof of its flexibility [82][79]. Additionally, taking off from the assertion that simple MPC (in [83]’s “hard control” sense) is not necessarily more robust than traditional control approaches (as repeated in [71]), these hybrid techniques and extensions to MPC can provide previously lacking robustness [80]. Nonetheless, this does not detract from the fact that the formal, generalised analysis of stability and robustness are very active areas of research pertaining to MPC [80].

When considering traditional methods, such as PID or SMC, the complexity associated with tuning the controller grows considerably as the system moves from SISO to MIMO categorisation. Able to handle multivariable systems with far greater ease, MPC does not face the same growth in complexity [79].

Aside from the difficulty associated with formally evaluating the stability of simple MPC systems, one of the most pressing challenges is the system’s dependence on accurate prediction models. Some authors have sought to overcome this with data-driven, machine learning approaches and others which circumvent the need for the user to define the model [71][84]. As such, the present work focuses on this area of development.

4.1.2 Applications of model-predictive control

MPC has seen broad usage across a variety of applications. A selection of these applications is reviewed herein. The examples included here, although just a few of the various applications of MPC, are useful in their illustration of the variety of timescales and performance requirements which MPC is used to satisfy. Noting this thesis’ later case study application in Chapter 6, this is a supporting reason for the selection of MPC in this context.

Power electronics

Vazquez et al. provide a useful review of MPC’s usage in power converters and drive systems. This statement applies to two highly-cited works, the first in 2014 [74] and

the second in 2017 [73]. In the 2014 work, the authors discuss the benefits of using MPC in the context of three use cases: grid-connected converters, inverters, and high-performance drives. Also in their 2014 work, the authors present different MPC strategies which aim to address the differing requirements of each use case. For example, finite-control-set MPC is discussed as being useful for power electronics applications because of its capacity to handle the discontinuous switching behaviour of converters. However, the authors note that the already-high computational burden of MPC is raised further still by this method as the number of discontinuities (switches) increases [74]. On the other hand, generalised predictive control is observed to be able to accommodate considerably larger prediction horizons without similarly large increases in computational cost. The trade-off of this method, is that it can't account for discontinuities for converters in the same way as finite-control-set MPC [74]. In the authors' 2017 work, continue this work in a similar vein, making recommendations to address key problems in the use of MPC for power electronics based on then-recent advances of their peers in this field [73]. Departing from precisely the same field of study, Huang et al. provide a thorough review of strategies to optimally distribute power between different energy sources in hybrid electric vehicles (HEVs) [77]. While, power management in HEVs had been extensively studied prior to Huang et al.'s work and MPC had previously been applied in this way, this work is reportedly the first review of the intersection of these topics. More recently than these other works in the realm of MPC's application to power systems, Karamanakos et al. provide a summary of work completed to help tailor MPC strategies to applications in this field [75]. In particular, the authors distinguish between "direct MPC" and "indirect MPC", the key difference being that the latter contains a modulation step which clearly addresses the limitations of some of the approaches reviewed previously by Vazquez et al. [74].

Building comfort and water resource management

Considering a broader field of energy-centred applications, Drgona et al. review the subject of MPC's use in climate control in the built environment [70]. While there have previously been reviews of this field, Drgona et al.'s useful contribution with this

work is in how they combine the underlying theory, a review of the research, and practical guidelines to using MPC for building climate control [70]. Moving beyond climate control in buildings, to a similarly widespread challenge, Castelletti et al. review research on the use of MPC to manage water reservoirs and urban water networks [76]. The authors present how MPC's strengths favour applications in this space and have led to further advancements. For instance, the use of sensors to gather vast quantities of data through existing flow networks has been used to integrate machine learning methods with MPC. However, the authors raise the recurring issue of computational expense and dependence on model accuracy as a drawback in the use of MPC in this field [76].

Autonomous vehicles

Yu et al. consider the application of MPC to autonomous ground vehicles (AGVs) [78]. As part of their broad review, the authors consider the use of MPC with individual vehicles for tasks such as motion planning and control. Through modifications made to the cost function in the controller, the authors discuss how earlier work has explored scenarios where an AGV has to avoid obstacles in path planning or otherwise undergo emergency manoeuvres during planned motion [78]. Yu et al. also consider distributed MPC systems across swarms of AGVs to accommodate "platoon" manoeuvres and collaborative autonomous operation. Also considering distributed and network-based MPC, Shi and Zhang consider a broader approach to this field, reviewing applications of MPC to autonomous vehicles in a much more general sense [72]. In particular, the authors consider vehicles in aerial, subsea, and ground environments [72]. A notable advancement to MPC, which Shi and Zhang mention, is the integration of event triggers into MPC. In particular, such systems (known as event-triggered MPC or event-based MPC) demonstrate a reduced computational expense by only running the required optimisation algorithms when the measured output states deviate from the desired trajectory by more than a set threshold [72].

Lastly, considering a far broader review of MPC applications in engineering fields,

Schwenzer et al. present a review of the underlying theory, evolution of widely-accepted approaches, practical guidance, and strengths and weaknesses of MPC [71]. Overlapping with many of the points raised in previously-cited work, Schwenzer et al. discuss the advances in MPC as including its broad applicability, reductions in its computational expense, improved guidance on tuning, and potential for integration with data-driven and machine learning approaches to circumvent the reliance on potentially uncertain (or incorrect) models [71].

This chapter, so far, has demonstrated that model-predictive control is a powerful and versatile strategy, valued (among many reasons) for its capacity to handle system constraints but simultaneously challenged by its computational expense and reliance on highly accurate system models for prediction. This thesis will continue by investigating the application of MPC to the FutureForge manipulator, a member of a class of system where these challenges are particularly pronounced (industrial manipulators). To establish the remainder of the necessary context, the discussion in this chapter now transitions to an overview of these systems. Understanding their kinematic architectures alongside the nonlinear dynamics outlined in Chapter 3 is fundamental to appreciating the control problems that subsequent chapters seek to address.

4.2 An overview of industrial manipulators

This section lays out another important context for the remaining portion of this thesis, defining the subject (industrial manipulators) and their importance in modern industry.

Industrial manipulators have become a common feature of modern manufacturing and automation. The terms “industrial manipulator” and “industrial robot” are often used interchangeably in the academic literature. Formally, the Robotic Industries Association defines an industrial robot as an “automatically controlled, reprogrammable, multipurpose manipulator, programmable in three or more axes, which can be either fixed in place or mobile for use in industrial automation applications” [85]. This definition captures the essential features which distinguish such systems from machines

with only one function: autonomous operation, which facilitates flexibility of application, and spatial mobility to position and orient an end-effector. The *FutureForge* manipulator does meet the specification of “industrial robot” according to the quoted definition. However, the system studied in Chapters 3 and 6 is best referred to as an “industrial manipulator” for two key reasons. Firstly, the automatic switching between applications is not directly explored in this work. While different manoeuvre types are considered in Chapter 6, different sensing and decision-making processes are not considered. Secondly, only the 2-DOF motion of the manipulator is considered in this work. While the manipulator is capable of 6-DOF motion, the present work does not directly consider this.

At its core, an industrial manipulator is a machine comprising a collection of rigid segments (links) connected by joints, which can either be actuated or unactuated. These joints are typically either revolute (rotational) or prismatic (linear). Hedberg provides a useful overview of how different combinations of joints characterise some common machine types, such as the “Cartesian robot” (featuring only prismatic joints) [86]. Typically, manipulators are designed to perform physical “manipulation” tasks, such as lifting, moving, and placing objects – avoiding the need for direct human intervention, thus supporting safety and efficiency.

Industrial manipulators have evolved substantially in their capabilities over recent decades. Zamalloa et al. provide a useful series of “generations” of developments [87]. Starting with their “Generation 0: Pre-Robots”, which encapsulate the relatively simple, functionally repetitive machines supporting labour until 1950, the authors then describe the “first manipulators” generation. This generation features the Unimate created by George Charles Devol, for General Motors in 1954 [88], which is commonly accepted as the first machine recognisable as an “industrial robot”. The following generation (spanning 1968 to 1977) is characterised by the incorporation of sensing technologies into manipulators. This period also saw the introduction of the programmable logic controller (PLC), which accelerated the implementation of new technologies and

control strategies for these machines. “Generation 3: Industrial robots” is seen as the so-called “boom” of automation, with large companies investing huge sums of money in manipulator system design and deployment. Spanning until 1999, this generation even includes the popularisation of such technologies in the wider public, with LEGO producing its first Mindstorms kit in 1998 and Sony producing the AIBO in 1999 [87]. Lastly, at the time of publication, Zamalloa et al. indicated that their present (fourth) generation of industrial robots could be described as “intelligent robots”. This is the first generation where robots were seen as capable of making decisions based on the data they gather from their environments. Projecting forward into the present day, Zamalloa et al. described the next generation of robotics as “collaborative and personal robots”, which emphasises the capacity of robots and manipulators to be adaptive and responsive to users’ needs.

4.2.1 Common kinematic architectures

In this work, the phrase “kinematic architecture” is used to refer to the arrangement of a manipulator’s links and joints between the base and end-effector. The performance characteristics of an industrial manipulator are dictated by its kinematic architecture. There are three primary architectures used for industrial manipulators: serial, parallel, and hybrid.

Serial manipulators

Serial manipulators are the most common and perhaps the most recognisable configurations of manipulator systems. In these structures, the end-effector is connected to the base through a single series of links and joints forming an open kinematic chain. Crudely, this is similar to a human arm if one considers the shoulder-to-elbow structure as a single link, the elbow-to-wrist as another single link, the torso as the base, and the hand as the end-effector. This serial arrangement provides a large, dexterous workspace and high versatility. This reach and manoeuvrability is this architecture’s key strength, allowing manipulators of this type to navigate environments which may be cluttered with obstacles.

However, this architecture also introduces trade-offs. The cantilevered nature of these manipulators mean that each link and actuator must support the mass of all subsequent links as well as the payload. This results in a relatively low payload-to-weight ratio capacity and reduced structural stiffness compared to parallel architectures, which can lead to vibrations and deflections. A further implication of these concerns is that errors in the position of links/joints closer to the base can be amplified to cause larger errors in links/joints closer to the end-effector.

Due to this architecture’s advantages, serial manipulators are seen as beneficial in many cases, especially where flexibility and large workspaces are favoured. Welding, painting, assembly and packing tasks are a common example of this. Ropo et al. designed a low-cost pick-and-place manipulator system, citing these widely accepted strengths and further favouring the fact that the placement of actuators at each joint permitted each link to be driven directly [89]. The simplified geometry considered by Zhao, Li, and Chen – to design a bio-inspired UAV-mounted manipulator – is the serial mechanism of a bird of prey’s leg and talon [90]. Although, note that the authors then augment this simplified basis to a hybrid architecture, improving the system’s stiffness and precision. Saheb, Babu, and Raju provide a comprehensive overview of serial manipulator structures and compare them with parallel manipulators [91]. Comparatively, the authors note that serial manipulators have the advantage of being simpler to obtain the governing dynamics for. Saheb, Babu, and Raju state that forward kinematic analysis is typically more difficult for serial manipulators, and that the payload-to-weight ratio is higher for serial manipulators [91]. However, this contradicts the established consensus in the field. By contrast, Pandilov and Dukovski provide a comparison of the pros and cons of serial and parallel manipulators: identifying the lower payload capacity, simpler forward dynamics, and more complex inverse kinematics of the serial architecture [92].

Parallel manipulators

Parallel manipulators are defined as closed kinematic chains where the end-effector is connected to the base by multiple, separate kinematic chains. Returning to the analogous comparison to human arms used for serial manipulators, a parallel architecture could be mimicked by treating a tool held in both hands as the end-effector, the torso as the base, and the arms as the separate kinematic chains. This design distributes loads through the separate chains, resulting in higher stiffness, accuracy, and payload capacity compared to serial manipulators of similar size. Due to the parallel architecture, these systems are also capable of higher speeds and accelerations [92].

However, these benefits typically come with trade-offs. Notably, to achieve the same workspace dimensions, a parallel mechanism requires a significantly bigger machine footprint (of its base) compared to serial mechanisms. Additionally, tasks such as calibration, modelling the governing dynamics, and solving the forward kinematic problems are considerably more challenging for parallel mechanisms than for serial ones [92].

Illustrating the strengths and weaknesses discussed here, Pierrot et al. design and optimise an alternative parallel manipulator to the more popular Delta robot [93]. To improve on existing designs, the authors of this work attempt to ensure that the end-effector’s “travelling plate” is free of singularities (configurations in which no actuation can control the end-effector’s placement and orientation) [93]. Comparatively, the authors’ novel parallel mechanism achieves improved load distribution among each of its kinematic chains although with an increased machine mass. In a broader review of parallel manipulators, Patel and George discuss their application in a broad range of engineering industries spanning aerospace, medical robotics, and traditional manufacturing settings [67]. Settings such as surgery are eminently suited to parallel mechanisms due to the importance of accuracy and structural stiffness; the reduced flexibility compared to serial mechanisms can be comfortably accounted for by careful planning by the human surgical team in advance of the process. The well-known Gough-Stewart platform is widely used in this context as well as others, such as space-

based applications [67]. In particular, the Gough-Stewart platform has seen use as part of craft-to-craft docking mechanisms; and, Patel and George highlight the use of such architectures in satellite tracker dishes [67].

Hybrid manipulators

Hybrid manipulators are composed of both parallel and serial kinematic chains, used strategically to leverage the benefits (or minimise the weaknesses) of each separate architecture. The use of hybrid robotics is a diverse field, but is tremendously useful given the proper design approach. The particular blend of strengths and weaknesses found in any hybrid mechanism is dependent on how the serial and parallel components are combined. The FutureForge manipulator system is one such hybrid manipulator in so far as it connects multiple parallel mechanisms (parallelograms) in series.

Gao and Zhang present the design, analysis, and multiobjective optimisation of a hybrid manipulator for CNC machining [94]. The machining tool is connected to a Gough-Stewart platform (comprising the parallel mechanism), while the workpiece is mounted on a gantry system which can move the specimen horizontally below the tool (the serial mechanism). In this apparently simple approach, the authors were able to extend the usefulness of the high-stiffness parallel mechanism by including a serial mechanism to increase the utility of its workspace. Note the semantic distinction here: Gao and Zhang’s design doesn’t enable the parallel manipulator to move through a greater workspace directly; instead, the authors’ solution allows the workspace to be moved within the parallel manipulator’s reach (achieving a qualitatively similar outcome).

In earlier work, Tanev presents a kinematic analysis of a novel hybrid manipulator architecture [95]. Specifically, Tanev’s architecture consists of two parallel mechanisms connected in series: each of the two parallel structures consists of three prismatic links connecting a triangular base to a triangular end-effector (which, in the case of the intermediate parallel structure, is a triangular base for the next mechanism in the

series). Tanev then derives closed-form solutions to the forward and inverse kinematic problems for this system. The benefits which the author subsequently observed in their work are the improved rigidity and workspace reach of the structure compared to traditional serial and parallel mechanisms respectively, as well as the simplicity with which the kinematics problems could be computed by comparison to that of the traditional constituent architectures [95].

4.3 Control of industrial manipulators

Having detailed the most common kinematic architectures of industrial manipulators, the discussion now turns to the methods employed to control their motion. The control of industrial manipulators is a well-established field of research that evidences a broad range of strategies. To structure this overview, this section organises control approaches into three broad categories according to Afram and Janabi-Sharifi’s definitions of “hard”, “soft”, and “hybrid” control strategies [83]. These categories are “classical and hard control”, which includes traditional and robust model-based techniques; “intelligent and soft control”, which includes data-driven methods such as neural networks and fuzzy logic; and “hybrid control”, which combines elements from both preceding categories.

4.3.1 Classical and hard control

Ajwad et al. explain the prevalence of PID control strategies in the control of industrial manipulators [66]. However, the authors also qualify that these classical control approaches are increasingly commonly augmented by other strategies (turning them into hybrid control strategies) in approaches such as Fuzzy-PID, or NN-PID. However, simple PID approaches are still common occurrences in industrial settings due to their simplicity and well-understood procedures for design. Advancements in tuning methods (including approaches such as Ziegler-Nichols, Lambda tuning, and optimisation-based techniques like particle swarm optimisation) have been a great contributor to the reason PID retains this prevalence. Notably, [66] voices their prediction that “PID control

[will get] smarter and will continue to be the main workhorse of industry” while other, less traditional, techniques also see advances in the reduction of their computational burdens. While simpler classical approaches, such as on/off control, have not seen significant use in industrial manipulator systems, there is still great appeal for practitioners in using well-understood, classical approaches.

Gain scheduling, although not an alternative control strategy on its own, is a tool which sees frequent use in this space. Notably, it facilitates the use of simpler approaches (like PID-based control) since it permits the application of these techniques to nonlinear systems by first linearising them.

Robust and nonlinear control strategies are important and have found use in industrial manipulators as they fill something of a void left by constant gain feedback systems like classical PID which often perform poorly for nonlinear or uncertain systems (without augmentations such as gain scheduling) [66]. So-called “robust” control strategies include approaches such as computed torque control (CTC), sliding mode control (SMC), model reference adaptive control (MRAC), model-predictive control (MPC), and H_∞ control. Note that this definition may be convenient for the purposes of structuring research, but authors such as [71] report the limitations of established stability analysis to inspect model-predictive control.

Computed torque control (CTC) is another strategy which, extending the classifications of hard and soft control in [83], should be noted here. A useful overview of CTC’s deployment, and the underlying theory is presented separately by [96][97][98]. In each of these works, the authors use PID-CTC combinations to achieve improved control of their nonlinear systems by comparison to the results observed by PID methods on their own. In comparison to other methods discussed here, which have high computational costs associated with them, CTC and its variants are much less computationally expensive to implement. Despite its advantages, however, [96] highlights the method’s model dependency – given that precise knowledge of the manipulator dynamics is required,

this can be seen as a disadvantage to its use in industrial settings.

While SMC can be considered a mature field, it remains an active area of research, with continual development of new approaches and hybrid strategies to mitigate established limitations and enhance performance. Recent advances focus on mitigating the chattering phenomena observed in conventional SMC. Using their respective backstepping approaches, [99] and [100] demonstrate how their system retains a high level of robustness, favourable convergence characteristics by comparison to other control approaches, and reduced chattering through their use of an online adaptive element to the strategy. As [99] shows, the use of an adaptive control law in this way reduces the model-dependency of this control strategy.

Although very different to classical approaches like PID, adaptive and predictive approaches such as MRAC and MPC should also be considered here as examples of hard control. Zhang and Wei present a thorough review of how MRAC has been used to control manipulator systems [101]. In this work, a number of MRAC's strengths are identified, including: its ability to handle unknown or changing parameters, which could include payload parameters; its ability to mitigate the unmodelled (or incorrectly modelled) portions of the dynamic system; and, connecting to both of these points, its lower sensitivity to uncertainties or changes in parameter values. Note that [101]'s statements on the relative performance of MRAC are typically given in comparison to characteristics and observations of PID systems. The limitations of this method, however, are arguably as numerous as the benefits (although the weight of each is always determined contextually by the practitioner). In particular, [101] highlights the flaw that MRAC systems are not amenable to traditional, established stability analysis techniques. Nonetheless, some authors cited by [101] were able to assess their MRAC systems for at least local stability, and global stability in some cases, by introducing modifications and/or assumptions to the basic MRAC algorithm. This concern about the capacity to guarantee stability is echoed for MPC in [71]. Both of these methods are also critiqued for their increased computational requirements compared to other

methods, such as CTC and PID.

Fractional-order PID (FOPID) control is another noteworthy, state-of-the-art hard control approach. In this approach, the standard integral and derivative are replaced with a fractional-order integral and fractional-order derivative. Bingi, Prusty, and Singh present a review of application of FOPID control alongside fractional-order SMC and fractional-order fuzzy control approaches, noting that (while all three approaches have seen use in research), the SMC and fuzzy control approaches are more challenging to tune due to their increased number of tunable parameters by comparison to FOPID [102]. Even in the case of FOPID, tuning is a challenging, computationally intensive task by comparison to traditional PID approaches: the existing tuning approaches rely on sophisticated optimisation algorithms rather than heuristics and graphical methods which can be used for PID. However, this acknowledgment of the computational intensity of the process is not the same as in the case of MRAC or MPC given the tuning process for FOPID can be completed offline and is not required as frequently as the optimisation problems in MRAC or MPC. Compared to PID, [102] notes improved reference tracking and stability across all timeframes (steady-state and transient) as well as improved flexibility in its application due to the fractional-order calculus underpinning the method.

4.3.2 Intelligent and soft control

Following [83]’s definition of “soft control” strategies, this section considers techniques such as neural networks and fuzzy logic as the basis of control systems for industrial manipulators. However, note that – even with today’s relatively accessible computational power – some approaches under this category are seldom implemented in industry due to their computational requirements.

Fuzzy logic approaches are often especially favourable concepts because they can be developed based solely on operator-informed heuristics (from human experts), displacing the immediate need for complex tuning algorithms. However, authors such as

Chotikunnan et al. have developed data-driven approaches to the design of fuzzy rule-sets which remove this dependence on human experience [103]. Instead, their proposed approach relies on basing the fuzzy ruleset on the observed output of a PID-controlled system. While an important step, which provides useful insights and frameworks, the approach described in [103] lacks robust comparison as the authors compare the resulting fuzzy-controlled output to a simple P control system (which is intuitively less effective than the PID system on which the fuzzy system is tuned). Urrea, Kern, and Alvarado extend existing research into the use of fuzzy logic to control a 2-link manipulator, which consists of one controller for each of the system’s degrees of freedom [104]. In particular, the novelty of this work is derived from their decision to consider the manipulator’s acceleration as one of the variables fuzzified and used to determine the appropriate control actions. The authors demonstrate the benefit of their fuzzy logic controller by comparison to traditional PID control. It should be noted that the authors of this work don’t explicitly state the method by which they tuned the fuzzy rulesets of their controllers (although the absence of a specifically defined method suggests that the tuning process was manual and based on their intuition – an advantage offered by fuzzy logic). The PID control approach against which this was compared was tuned using the “auto-tune” functionality provided by MathWorks in the PID Designer app.

Li, Zhang, and Jin consider the positioning control of a redundant 6-DOF serial manipulator using a recurrent neural network (RNN) [105]. Differing from prior work, the authors purport to address two known shortcomings of RNNs: error accumulation in their reference tracking and their dependence on convex projection sets. Addressing a simpler mechanical system but a more general problem with neural network control systems, Gao et al. develop a neural network controller which is more robust in application to uncertain dynamic models [106]. They also developed theoretical stability guarantees for their particular system architecture, which was a gap left unaddressed by previous sources. Jin et al. provide a thorough review of the use of neural network control of manipulator systems [107]. The authors review the use of different training approaches (both on-line and off-line methods); network architectures, such as

feedforward neural networks (using backpropagation to adjust their internal weights), recurrent neural networks (with differing specialised applications) such as dual networks.

4.3.3 Hybrid control

Combining elements of “soft” and “hard” (or “intelligent” and “classical”) control approaches, hybrid control strategies have seen widespread use and research for industrial manipulators. Such strategies, therefore, combine elements such as fuzzy logic, neural networks, and robust/adaptive methods with methods such as PID, SMC, or CTC to enhance adaptability, robustness, and precision in their respective applications. As with the constituent strategies, no single hybrid control approach is universally optimal, and hence their strengths and limitations should be properly reviewed when considering their implementation.

Liu et al. address a perceived gap in achieving high robustness and high accuracy in manipulator positioning control using a Fuzzy-PID approach tuned using particle swarm optimisation [108]. Comparing their result to traditional PID controllers, the authors note the Fuzzy-PID strategy’s reduced vulnerability to uncertainties in the model. However, as appears unavoidable with hybrid control methods, the complexity of implementing such methods invariably increases in contrast to the simple hard control components. Nguyen and Bui present two distinct control strategies and compare their results: the first consists of two PID controllers for position and vibration control in a manipulator system, the second also consists of two PID controllers with an additional fuzzy logic component on the vibration mitigation controller [109]. In their implementation, the authors found that their second approach reduced the control action required to control the system’s unwanted oscillations. Paul et al. look to refine the use of Fuzzy-PID systems further by considering the fine details of these systems [110]. Specifically, the authors present the benefits and limitations of using type-2 fuzzy logic compared to type-1 fuzzy logic. As with many of the approaches presented in this section, the principal disadvantage in upgrading the algorithm from type-1 to type-2 is

the added complexity and computational cost.

Building on this further, Barhaghtalab et al. also include a neural network approach in a similar framework. Specifically, they construct what is referred to as an “adaptive fuzzy-neural inference system-based control approach” [111]. In essence, their control strategy consists of a PID controller in parallel with an adaptive fuzzy-neural inference system (ANFIS), which they explain as a fuzzy system which can be improved via neural-network-based learning approaches.

This chapter has provided a review of the two foundational areas relating to the remainder of this thesis: model-predictive control and industrial manipulators. The review has established that MPC is a powerful and versatile control strategy, capable of handling complex systems with constraints and multivariable interactions. However, this strategy’s effectiveness is often limited by a high computational burden and a dependence on accurate, predefined system models to facilitate high-fidelity predictions. Additionally, industrial manipulators – particularly hybrid architectures such as the FutureForge manipulator – are complex nonlinear systems for which a diverse array of control methods have been developed to enhance performance.

Referring to the work presented in the previous half of this thesis, the ADAP method is identified as a possible means of enhancing the accuracy of linearised models – addressing one of the identified vulnerabilities of MPC. Subsequent chapters of this thesis, therefore, present the development of an MPC control strategy relying on the ADAP method – contributing an improved approach to manipulator control using linearised prediction models.

Chapter 5

Application of the novel approximation algorithm to model predictive controllers

Chapter Preamble

This chapter extends the ADAP method developed in Chapter 2, applying it to the context of model-predictive control - creating a novel “ADAP-MPC” strategy. As this chapter will demonstrate, parallels can be drawn between the ADAP method and feedback linearisation. Nonetheless, it is proposed that the ADAP method offers advantages over the more traditional approaches to feedback linearisation discussed in this work. Two examples of this technique’s application are examined. These examples are a simple inverted pendulum system and a cart-mounted inverted pendulum system. These systems are selected due to their prevalence in control-focused literature. These examples also present increasing complexity, serving as a foundation for the more complex case study examined in Chapter 6.

In the context of linear model predictive control (MPC), much as with other control strategies, the model of the system must be linear and time-invariant (LTI). This does not mean that non-LTI systems cannot be controlled via linear MPC; it means that

Chapter 5. Application of the novel approximation algorithm to model predictive controllers

the model utilised by the control to make predictions must be LTI. Therefore, the algorithm presented in Chapter 2 initially appears inappropriate as it effectively introduces a time-varying component to the dynamic system. While this is perfectly reasonable for the solution of the system in question, it is not amenable for the construction of transfer functions or state-space models commonly used by model-based controllers.

This apparent limitation can be addressed by exploiting a core feature of the MPC framework. By treating the newly-introduced time-varying terms as measured disturbances, an LTI representation of the system is retained. In a conventional approach, where nonlinearities and coupling terms are simply ignored to create a linear model, the controller would experience these omitted dynamics as unmodelled disturbances.

5.1 An inverted pendulum

In this section, the application of the ADAP methodology to the control of a simple inverted pendulum is explored. The model is first specified, with its key features highlighted. Then, a model-based controller for this system is designed and the performance of this system is analysed and reviewed.

5.1.1 Model definition

Consider an inverted pendulum with a controllable torque applied at its base. The mass, m kg, is located at a fixed length, L m, from the revolute joint at its base where a torque, τ , is applied. All of the pendulum's mass is assumed to be lumped at this single point.

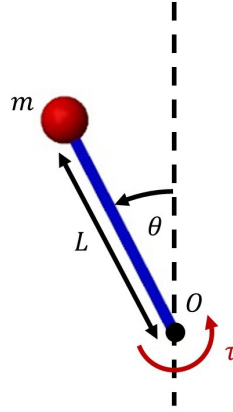


Figure 5.1: A diagram of the simple inverted pendulum system.

The pendulum is free to rotate about an axis through the point O which is perpendicular to the vertical dashed line. The generalised coordinate $\theta(t) = \theta$ describes the pendulum's angular displacement with respect to its unstable equilibrium position (balanced vertically). A torque, τ , can be applied about the axis through O - its positive direction being consistent with the positive direction of the generalised coordinate θ .

Through the application of Newton's 2nd Law, summing the moments acting about the axis through O , the ordinary differential equation describing the pendulum's motion is

$$\ddot{\theta} - \frac{g}{L} \sin(\theta) = \frac{1}{mL^2} \tau. \quad (5.1)$$

As described in Chapter 2, the ADAP method is applied by decomposing Eq (5.1) into two simpler problems. The first auxiliary problem for Eq (5.1) can be written as

$$\ddot{\phi} - \frac{g}{L} \phi = 0 \quad (5.2)$$

where $\phi(t) = \phi$ is the generalised coordinate for this auxiliary problem. In this context, ϕ will approximate θ for small values of θ (such that the small angle approximation is applicable) and under its unforced motion. The second auxiliary problem is similar to

Chapter 5. Application of the novel approximation algorithm to model predictive controllers

Eq (5.2) but with a nonhomogeneous term included

$$\ddot{\theta} - \frac{g}{L}\theta = \frac{1}{mL^2}(\tau + f) \quad (5.3)$$

where

$$f = \sin(\phi) - \phi. \quad (5.4)$$

By design, the ADAP method ensures that Eq (5.3) provides an adequate approximation of Eq (5.1), hence the generalised coordinate θ is used. The analytical solution to Eq (5.2), and therefore the ϕ -terms in Eq (5.4), is

$$\phi = \frac{1}{2}\phi_0(e^{\lambda t} + e^{-\lambda t}) + \frac{1}{2\lambda}\dot{\phi}_0(e^{\lambda t} - e^{-\lambda t}) \quad (5.5)$$

where ϕ_0 and $\dot{\phi}_0$ are the initial displacement and velocity conditions for ϕ and $\lambda = \sqrt{\frac{g}{L}}$. As will be observed in the later stages of this section, Eq (5.5) can be used to form a measured disturbance for the MPC controller to account for in its algorithm. In this application, the initial conditions are the instantaneous pendulum states (displacement and velocity) and the time, t , can be replaced with a one-dimensional row array of values spanning the prediction horizon. This formulation allows Eq (5.3) - the prediction model used in the MPC system - to account for the nonlinear dynamics of the inverted pendulum system while allowing the use of a linear MPC algorithm rather than requiring a nonlinear MPC algorithm.

For implementation in MATLAB, a state-space formulation is constructed based on the preceding equations, (5.1)-(5.4). Using the state vector $\mathbf{x} = [\theta, \dot{\theta}]^T$, the output vector $\mathbf{y} = \mathbf{x}$, and the 2×2 identity matrix $\mathbf{I}_2$,

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ \frac{g}{L} & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 & 0 \\ \frac{1}{mL^2} & \frac{1}{mL^2} \end{bmatrix} \begin{Bmatrix} \tau \\ f \end{Bmatrix} \quad (5.6)$$

and

$$\mathbf{y} = \mathbf{I}_2 \mathbf{x}. \quad (5.7)$$

For comparison, two feedback linearisation approaches are also considered (denoted “FL01” and “FL02”). In these two approaches, the feedback linearisation loop does not include the controller. This distinction is more clearly illustrated across Figs 5.2-5.4. In the first of these approaches, FL01, feedback linearisation is used to remove the restoring torque $mgL \sin(\theta)$ from the system dynamics. This can be summarised by the ODE and state-space equations as follows.

$$\begin{aligned}\ddot{\theta} &= \frac{1}{mL^2}\tau \\ \dot{\mathbf{x}} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x} + \left\{ \begin{array}{c} 0 \\ \frac{1}{mL^2} \end{array} \right\} \tau \\ \mathbf{y} &= \mathbf{I}_2 \mathbf{x}\end{aligned}\tag{5.8}$$

The second feedback linearisation approach, FL02 is similar to this but can be expressed as follows.

$$\begin{aligned}\ddot{\theta} - \frac{g}{L}\theta &= \frac{1}{mL^2}\tau \\ \dot{\mathbf{x}} &= \begin{bmatrix} 0 & 1 \\ \frac{g}{L} & 0 \end{bmatrix} \mathbf{x} + \left\{ \begin{array}{c} 0 \\ \frac{1}{mL^2} \end{array} \right\} \tau \\ \mathbf{y} &= \mathbf{I}_2 \mathbf{x}\end{aligned}\tag{5.9}$$

As can be seen in Fig 5.3, Eq (5.8) is achieved by feedback of the nonlinear restoring force to account for this effect in the actuating torque for the system. Similarly, as can be seen in Fig 5.4, Eq (5.9) is achieved by feedback of the nonlinear restoring force minus the linear approximation of this effect (thus allowing the linear effect to remain in the system response to the control action).

5.1.2 Control system design and parameter specifications

Firstly, this section begins by describing how the ADAP approach integrates with MPC control. In this novel system design, as illustrated in Fig 5.2, the MPC controller’s pre-

diction is based on the plant behaving like the first auxiliary problem (5.2). Clearly, however, this is not the case unless another intervention is applied. The plant does indeed behave like this simplified dynamic system when the additional term in the second auxiliary problem (5.3), the $\frac{f}{mL^2}$ term, is “fed back” to the controller as a measured disturbance. Rather than controlling a feedback linearised plant, the proposed ADAP method considers the effects omitted from the first auxiliary problem as predictable (measured) disturbances from the expected (first auxiliary problem) behaviour. Moreover, using the analytical solution in Eq (5.5), the evolution of this measured disturbance can be predicted and thus incorporated into the controller to enhance the resulting behaviour. This predictive capacity is notable when compared with the more traditional feedback linearisation approaches in Eqs (5.8), FL01, and (5.9), FL02, which can only yield the instantaneous torque required to linearise the system.

In each of the cases compared in this section, the MPC controller is specified identically. The only difference in the three cases is the linearisation approach. The first of these is illustrated in Fig 5.2 and is described using Eqs (5.1)-(5.4); the second is illustrated in Fig 5.3 and is described using Eqs (5.8); and, the third is illustrated in Fig 5.4 and is described using Eqs (5.9).

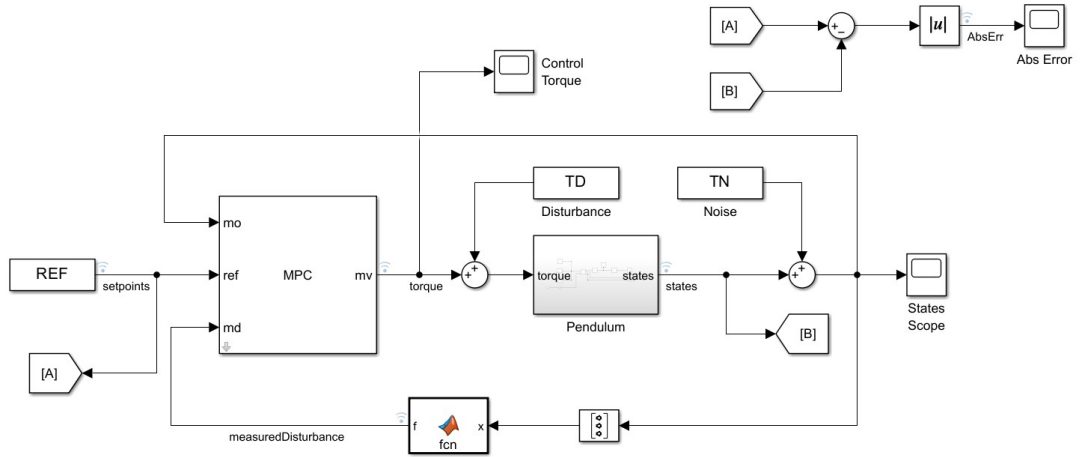


Figure 5.2: Block diagram representation of the ADAP-based control approach

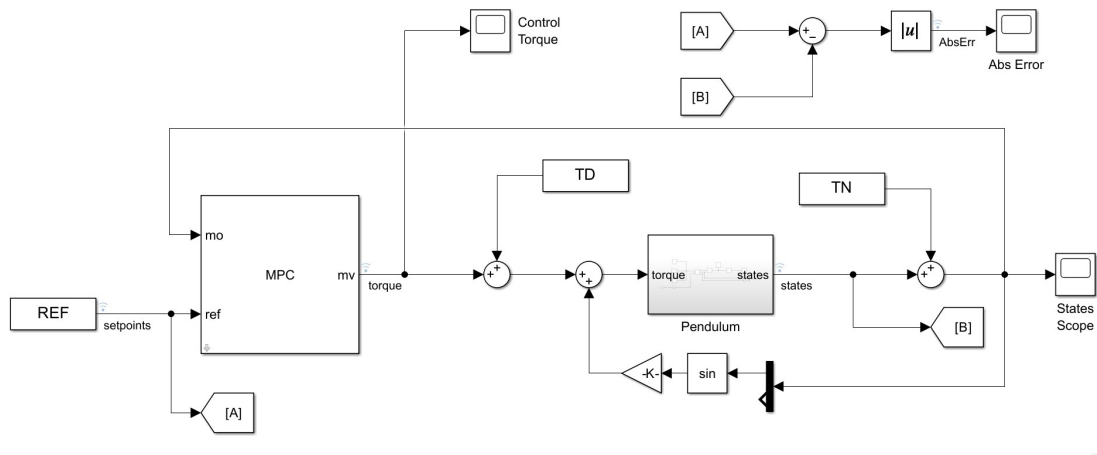


Figure 5.3: Block diagram representation of the MPC control approach using feedback linearisation as described by the system in Eq (5.8).

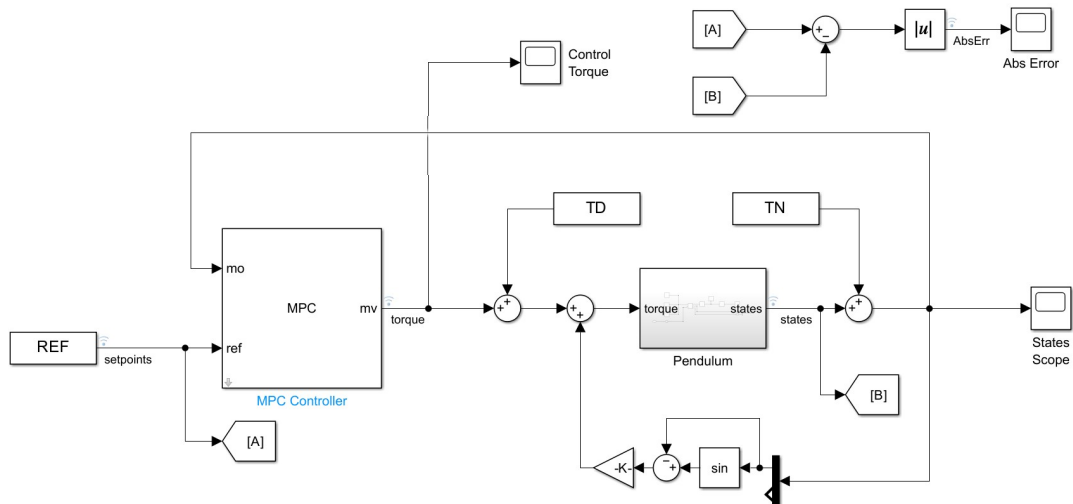


Figure 5.4: Block diagram representation of the MPC control approach using feedback linearisation as described by the system in Eq (5.9).

For the MPC controller in these approaches, the settings are as summarised in Table 5.1. The sample time and prediction horizon are selected so as to allow for a significant range of motion to be undertaken within the horizon interval. The control horizon was initially varied from half to a third of the prediction horizon although marginally improved performance was noted when the latter of these was selected. The selection of the limits on the manipulated variable (the control torque, τ) is arbitrary. If this

Chapter 5. Application of the novel approximation algorithm to model predictive controllers

scenario were intended to mimic a real physical system, then these limits would be informed by the actuator parameters. The output state weights are set such that there is no target for the pendulum’s velocity, only for the displacement state. The output variable scale factor is set to span the operational range of the generalised coordinate. Similarly, the scale factor for the manipulated variable is the lower τ limit subtracted from the upper limit.

Setting	Value
Sample time	0.05 s
Prediction horizon	30 (steps)
Control horizon	20 (steps)
Maximum manipulated variable, τ	± 20 Nm
Output state weights	[100, 0]
Manipulated variable rate weight	0.1
Output variable scale factor	2π
Manipulated variable scale factor	40

Table 5.1: MPC settings used for the control of the inverted pendulum.

A key feature to note when adapting the ADAP method for use with MPC is the role of the measured disturbance. The measured disturbance, the $\frac{1}{mL^2}f$ term in Eq (5.6), informs the MPC algorithm of the dynamics which are otherwise lost in the simplified state matrix. By contrast, for feedback linearisation approaches, as illustrated by Figs 5.3 and 5.4, this signal is typically added to the plant input to effectively “neutralise” the nonlinearities.

5.1.3 Simulation of the proposed control system and analysis of its performance

In this section the controller described by the settings in Table 5.1 is applied to the inverted pendulum system depicted by Fig 5.1 and Eq (5.1). A range of scenarios are explored in order to compare the three linearised systems: the ADAP approach described by Eqs (5.2) and (5.3); the first feedback linearisation approach, FL01, is described by (5.8); and, the second feedback linearisation approach, FL02, is described by (5.9).

In this illustrative scenario, the pendulum is initially balanced at its unstable equilibrium. The setpoint of the controller is then stepped from 0 radians to $\frac{\pi}{2}$ radians at $t = 5$ seconds.

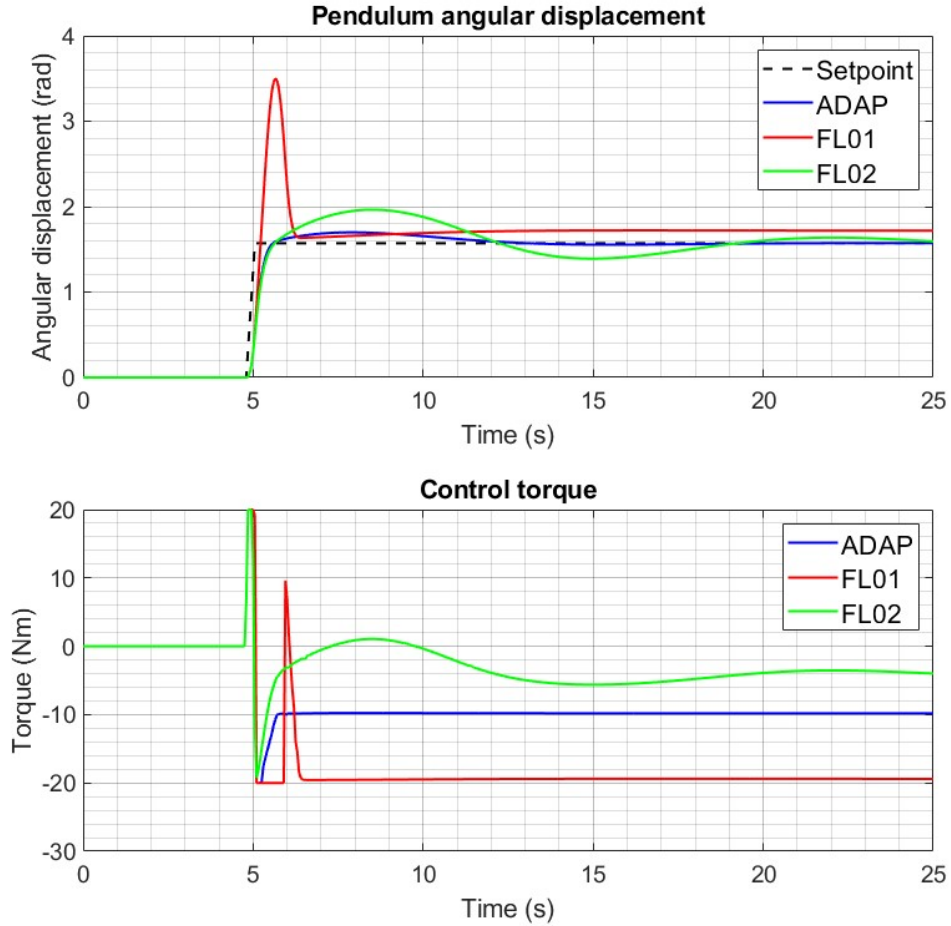


Figure 5.5: Position and control torque graphs for the inverted pendulum using three model linearisation strategies.

In this simple step response, the three responses are notably different. The FL01 response overshoots the target position most significantly, overshooting by 123%, and results in a steady-state error of 0.15 radians without any oscillation after its drop from its peak value. The FL02 response overshoots substantially less than the FL01 response, overshooting by 25%, and it is eventually able to reach a steady-state error of 0 radians although it does so with a settling time of 24.6 seconds (beyond the range

Chapter 5. Application of the novel approximation algorithm to model predictive controllers

shown in Fig 5.5).

The ADAP strategy yields the most desirable response of the three methods in Fig 5.5, exhibiting an overshoot of only 8% and achieving a steady-state error of 0 radians with a settling time of 11.9 seconds. These performance characteristics are summarised in Table 5.2.

Response	% overshoot	Steady-state error (radians)	Settling time
ADAP	8%	0	11.9
FL01	123%	0.15	N/A
FL02	25%	0	24.6

Table 5.2: Performance measures associated with the three responses shown in Fig 5.5.

Depending on the specific context of a controller's implementation, different objectives may be set for the desired performance. While Table 5.2 indicates typical response characteristics in relation to positional accuracy in the transient and steady-state regions, another measure may be the work required by the actuator. Figure 5.6 illustrates the work done by an actuator by considering the area enclosed in the torque-displacement graphs for the three responses depicted in Fig 5.5. As can be seen from the areas labelled here (computed for the $t \in [0, 25]$ second interval to correspond to Fig 5.5), the ADAP controller is able to achieve the response characteristics in Table 5.2 while demanding less energy from the actuator.

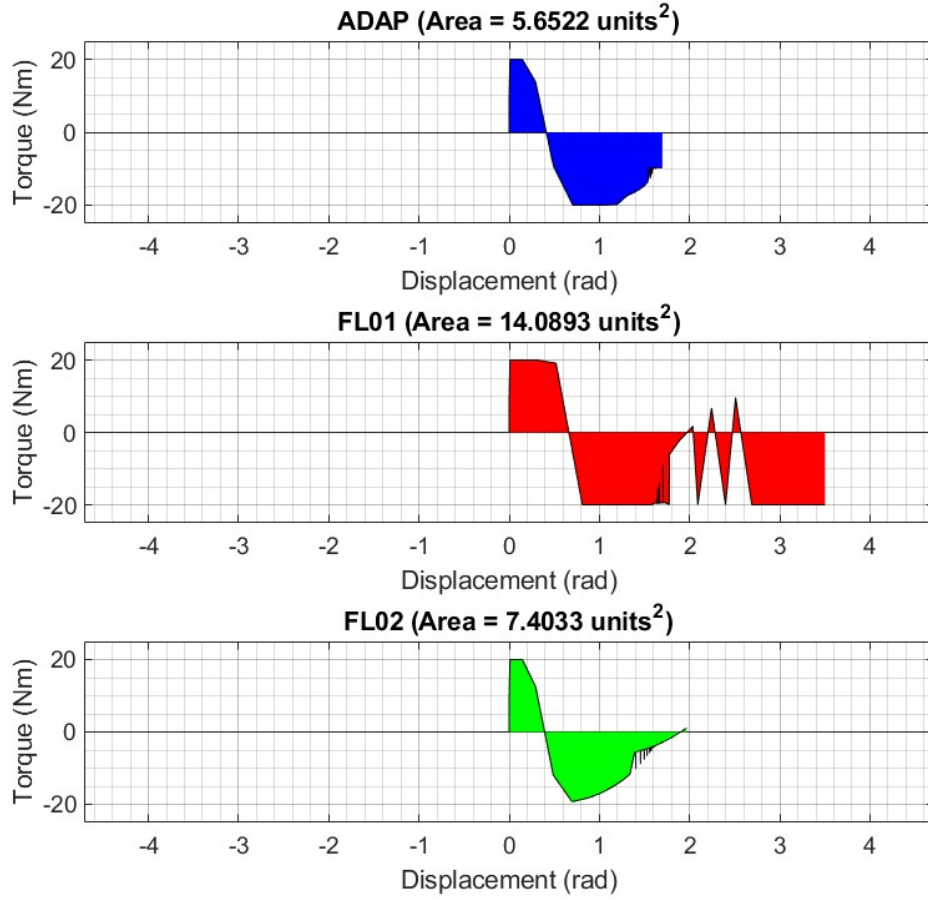


Figure 5.6: Torque-vs-displacement plots used to indicate the work done by actuators in achieving each of the three responses.

5.2 An inverted pendulum on a mobile cart

In this example, the ADAP methodology is now applied to the classical control problem of a pendulum mounted to a cart. This is a common problem to consider in control systems development as it considers a highly unstable system (when the pendulum is to be balanced in its inverted pose) and also facilitates the study of coupled motion and/or multiobjective control.

5.2.1 Model definition

In this example, the cart-pendulum system is actuated by both a force applied to the cart and a torque applied directly to the pendulum about its axis connecting it to the cart. In this way, the system has greater similarity to the industrial manipulator to be considered later in this work: the number of degrees of freedom is equal to the number of control inputs, meaning the system is neither redundantly actuated nor underactuated.

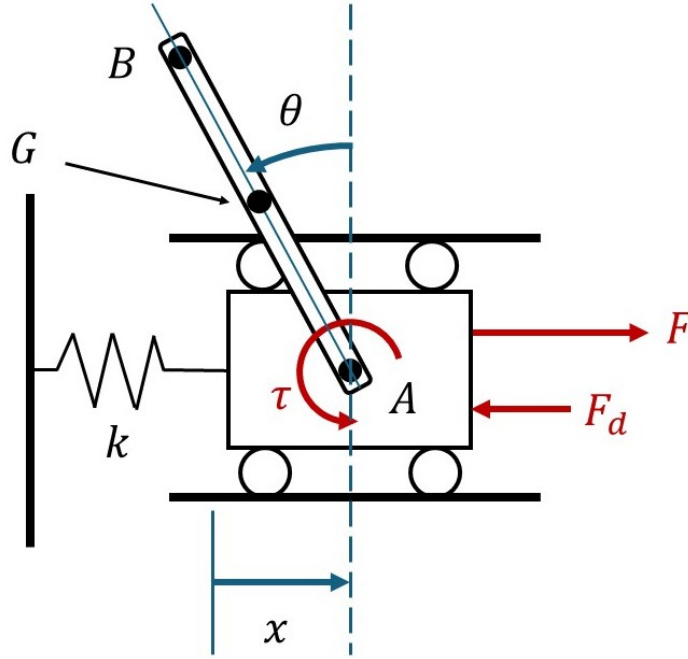


Figure 5.7: A diagram of the cart-mounted pendulum system.

As can be seen in Figure 5.7, a simple linear, spring of stiffness k N/m, connects the cart to an arbitrary wall while the cart is also subject to a frictional force, F_d , as it is driven by the applied (control) force, F . The pendulum is driven by the torque τ acting about its axis through the point A and may, without loss of generality, be considered as being subjected to a linear viscous frictional force about the same axis. The pendulum has its mass distributed uniformly along its length, $L = |AB|$, with its centre of mass thus being located halfway along this length at G . The cart translates through the generalised coordinate $x = x(t)$, and the pendulum's generalised rotational coordinate about A is denoted $\theta = \theta(t)$. The mass of the cart is denoted by m_1 kg

Chapter 5. Application of the novel approximation algorithm to model predictive controllers

while the mass of the pendulum is denoted by m_2 .

The two ordinary differential equations describing the motion of these masses through the two described generalised coordinates are as follows.

$$(m_1 + m_2)\ddot{x} - \frac{1}{2}m_2L\cos(\theta)\ddot{\theta} + c_1\dot{x} + \frac{1}{2}m_2L\sin(\theta)\dot{\theta}^2 + kx = F \quad (5.10)$$

$$-\frac{1}{2}m_2L\cos(\theta)\ddot{x} + I_A\ddot{\theta} + c_2\dot{\theta} - \frac{1}{2}m_2gL\sin(\theta) = \tau \quad (5.11)$$

In applying the ADAP methodology for this dynamic system, the desired set of equations for the first auxiliary problem is

$$(m_1 + m_2)\ddot{\tilde{x}} - \frac{1}{2}m_2L\ddot{\tilde{\theta}} + c_1\dot{\tilde{x}} + k\tilde{x} = 0 \quad (5.12)$$

$$-\frac{1}{2}m_2L\ddot{\tilde{x}} + I_A\ddot{\tilde{\theta}} + c_2\dot{\tilde{\theta}} - \frac{1}{2}m_2gL\tilde{\theta} = 0. \quad (5.13)$$

Solving these two ODEs then permits the computation of the subsequent two ODEs forming the second auxiliary problem

$$(m_1 + m_2)\ddot{x} - \frac{1}{2}m_2L\ddot{\theta} + c_1\dot{x} + kx = F + w_1 \quad (5.14)$$

$$-\frac{1}{2}m_2L\ddot{x} + I_A\ddot{\theta} + c_2\dot{\theta} - \frac{1}{2}m_2gL\theta = \tau + w_2. \quad (5.15)$$

where

$$w_1 = \frac{1}{2}m_2L(\cos(\tilde{\theta}) - 1)\ddot{\tilde{\theta}} + \frac{1}{2}m_2L\sin(\tilde{\theta})\dot{\tilde{\theta}}^2 \quad (5.16)$$

$$w_2 = \frac{1}{2}m_2L(\cos(\tilde{\theta}) - 1)\ddot{\tilde{x}} + \frac{1}{2}m_2gL(\sin(\tilde{\theta}) - \tilde{\theta}). \quad (5.17)$$

As in previous examples, using Eqs (5.16) and (5.17) requires that the first auxiliary problem, Eqs (5.12) and (5.13), be solved first. For compactness (and alignment with the notation used in the MATLAB-based simulations described later in this section),

these problems can be written in the following first-order representation

$$\dot{\mathbf{p}} = \frac{1}{\Delta} \begin{bmatrix} 0 & \Delta & 0 & 0 \\ -kI_A & -c_1I_A & \left(\frac{1}{2}m_2L\right)^2g & -\frac{1}{2}m_2Lc_2 \\ 0 & 0 & 0 & \Delta \\ -\frac{1}{2}m_2Lk & -\frac{1}{2}m_2Lc_1 & \frac{1}{2}(m_1+m_2)m_2gL & -(m_1+m_2)c_2 \end{bmatrix} \mathbf{p} \quad (5.18)$$

where $\mathbf{p} = [\tilde{x}, \dot{\tilde{x}}, \tilde{\theta}, \dot{\tilde{\theta}}]^T$ and $\dot{\mathbf{p}}$ is its derivative. Note the determinant of the inertia matrix from Eqs (5.12) and (5.13) is denoted by Δ and is a constant in this problem.

$$\Delta = (m_1 + m_2)I_A - \left(\frac{1}{2}m_2L\right)^2 \quad (5.19)$$

The solution to this system of first-order differential equations can be written generally as

$$\mathbf{p} = \sum_{j=1}^4 c_j \mathbf{v}_j e^{\lambda_j t} \quad (5.20)$$

where $\mathbf{u}_j$ is the j -th eigenvector, corresponding to the eigenvalue λ_j , of the system and c_j is a constant to be determined via the application of the system's initial conditions. These latter constants can be found by solving the system

$$\mathbf{c} = \mathbf{V}^{-1} \mathbf{p}(@t = 0) \quad (5.21)$$

where $\mathbf{c} = [c_1, c_2, c_3, c_4]^T$; $\mathbf{U}$ is the 4×4 square matrix of eigenvectors, $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4]$; and, $\mathbf{p}(@t = 0)$ is the vector of the system's initial conditions.

For comparison, a more traditional feedback linearisation approach is also explored. The original, nonlinear system can be written as

$$M \begin{Bmatrix} \ddot{x} \\ \ddot{\theta} \end{Bmatrix} = \mathbf{G} \quad (5.22)$$

Chapter 5. Application of the novel approximation algorithm to model predictive controllers

where M is the inertia matrix

$$M = \begin{bmatrix} (m_1 + m_2) & -\frac{1}{2}m_2L \cos(\theta) \\ -\frac{1}{2}m_2L \cos(\theta) & I_A \end{bmatrix} \quad (5.23)$$

and $\mathbf{G}$ is the vector of remaining, non-inertial terms

$$\mathbf{G} = \begin{Bmatrix} F - kx - c_1\dot{x} - \frac{1}{2}m_2L \sin(\theta)\dot{\theta}^2 \\ \tau - c_2\dot{\theta} + \frac{1}{2}m_2gL \sin(\theta) \end{Bmatrix}. \quad (5.24)$$

The intended state-space representation of the feedback linearised model in this case, should permit $\dot{\mathbf{q}} = [\dot{x}, \ddot{x}, \dot{\theta}, \ddot{\theta}]^T$. Hence expressions for $\ddot{x}$ and $\ddot{\theta}$ are required. These can be found by arranging Eq (5.22) to obtain

$$\begin{Bmatrix} \ddot{x} \\ \ddot{\theta} \end{Bmatrix} = M^{-1}\mathbf{G}. \quad (5.25)$$

This can be written as

$$\begin{Bmatrix} \ddot{x} \\ \ddot{\theta} \end{Bmatrix} = \alpha + \beta \begin{Bmatrix} F \\ \tau \end{Bmatrix} \quad (5.26)$$

where $\beta = M^{-1}$ and $\alpha = [\alpha_1, \alpha_2]^T$ with

$$\begin{aligned} \alpha_1 = -\frac{1}{\det(M)} & \left(M_{22} \left(kx + c_1\dot{x} + \frac{1}{2}m_2L \sin(\theta)\dot{\theta}^2 \right) \right. \\ & \left. + M_{12} \left(-c_2\dot{\theta} + \frac{1}{2}m_2gL \sin(\theta) \right) \right) \end{aligned} \quad (5.27)$$

and

$$\begin{aligned} \alpha_2 = \frac{1}{\det(M)} & \left(M_{21} \left(kx + c_1\dot{x} + \frac{1}{2}m_2L \sin(\theta)\dot{\theta}^2 \right) \right. \\ & \left. + M_{11} \left(-c_2\dot{\theta} + \frac{1}{2}m_2gL \sin(\theta) \right) \right) \end{aligned} \quad (5.28)$$

This can be used to obtain the required actions, F and τ , to achieve the linearised

behaviour sought in the system.

$$\begin{Bmatrix} F \\ \tau \end{Bmatrix} = \beta^{-1} \left(\begin{Bmatrix} \ddot{x} \\ \ddot{\theta} \end{Bmatrix} - \begin{Bmatrix} \alpha_1 \\ \alpha_2 \end{Bmatrix} \right) \quad (5.29)$$

The accelerations yielded by these actions can be denoted by $\ddot{x} = v_1$ and $\ddot{\theta} = v_2$ in $\mathbf{v} = [v_1, v_2]^T$. These are the output of the MPC controller and feed into the system model as illustrated in Fig 5.8.

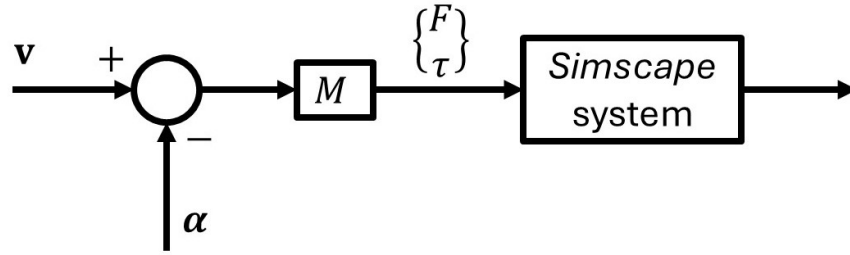


Figure 5.8: A block diagram representation of how the two vectors $\mathbf{v}$ and α contribute to the actuation of the cart-pendulum system.

In addition to this feedback linearisation approach, two simpler feedback strategies are also employed for comparison. The simplest of these (FL01) presents the highest levels of assumptions and simplifications, while the second (FL02) is more similar to the third method presented previously (FL03).

FL01 only feeds back the signal

$$\mathbf{u}_{\text{FL}} = \begin{Bmatrix} \frac{1}{2}m_2L\sin(\theta)\dot{\theta}^2 \\ -\frac{1}{2}m_2gL\sin(\theta) \end{Bmatrix} \quad (5.30)$$

and the prediction model assumes that the remaining portion is the following linear expression.

$$(m_1 + m_2)\ddot{x} + c_1\dot{x} + kx = F \quad (5.31)$$

$$I_A\ddot{\theta} + c_2\dot{\theta} = \tau \quad (5.32)$$

The state-space representation for this model is $\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$ and $\mathbf{y} = C\mathbf{x} + D\mathbf{u}$ where

$$A = \frac{1}{\det(M)} \begin{bmatrix} 0 & \det(M) & 0 & 0 \\ -kI_A & -c_1I_A & 0 & -\frac{1}{2}m_2Lc_2 \\ 0 & 0 & 0 & \det(M) \\ \frac{1}{2}m_2Lk & -\frac{1}{2}m_2Lc_1 & 0 & (m_1 + m_2)c_2 \end{bmatrix} \quad (5.33)$$

$$B = \frac{1}{\det(M)} \begin{bmatrix} 0 & 0 \\ I_A & \frac{1}{2}m_2L \end{bmatrix} \quad (5.34)$$

$$C = I_4 \quad (5.35)$$

$$D = 0 \quad (5.36)$$

By contrast, FL02 is far more similar to FL03, except that the M matrix is linearised about $\theta = 0$ radians. The subsequent tests are then constructed with this operating point in mind.

5.2.2 Control system design and parameter specifications

The block diagram representation for the ADAP implementation is identical to that of Fig 5.2 in Section 5.1. Again, the ADAP methodology allows the controller to control the linearised system subject to measured disturbances which are the system's predicted deviations from this simplified model.

Figs 5.9 and 5.10 show the diagrams used to implement FL01 and FL02. While the block diagram for the feedback linearisation approach described by Eqs (5.22)-(5.29), FL03, is illustrated in Fig 5.11.

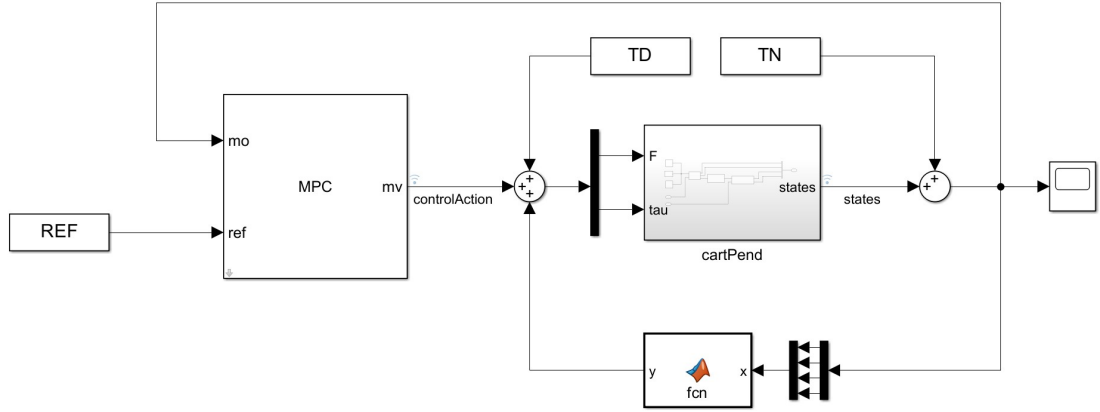


Figure 5.9: Block diagram representation of the MPC control strategy for the cart-mounted pendulum system using feedback linearisation strategy FL01.

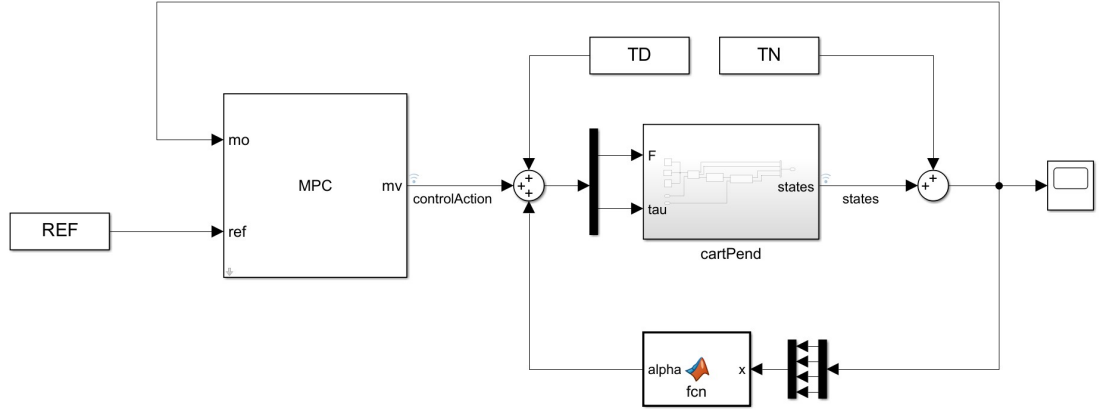


Figure 5.10: Block diagram representation of the MPC control strategy for the cart-mounted pendulum system using feedback linearisation strategy FL02.

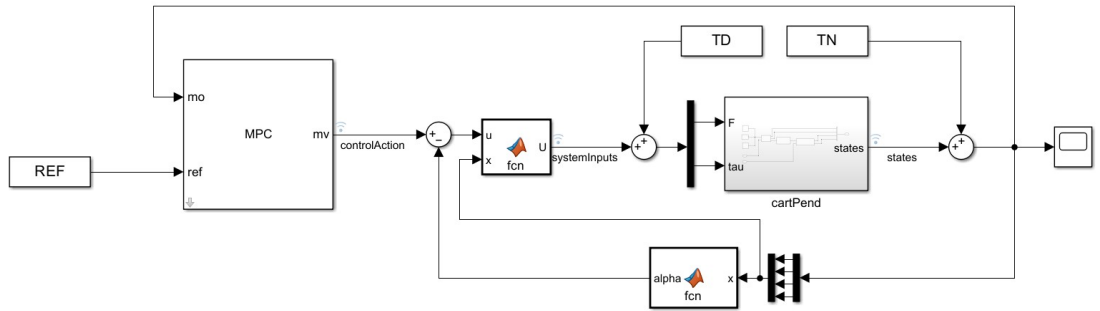


Figure 5.11: Block diagram representation of the MPC control strategy for the cart-mounted pendulum system using feedback linearisation strategy FL03.

The MPC settings used across each of these strategies are detailed in Table 5.3.

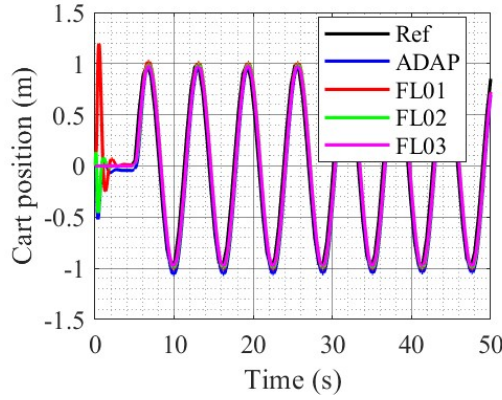
Setting	Value
Sample time	0.01 s
Prediction horizon	30 (steps)
Control horizon	3 (steps)
Maximum cart force (manipulated variable), F	± 50 N
Maximum pendulum torque (manipulated variable), τ	± 50 Nm
Measured output (state) weights	[5, 0, 5, 0]
Manipulated variable rate weight	[1, 1]
x scale factor (measured output)	10
θ scale factor (measured output)	0.2618 radians (15°)
Manipulated variable scale factors, $[F, \tau]$	[100, 100]

Table 5.3: MPC settings used for the control of the cart-mounted pendulum system.

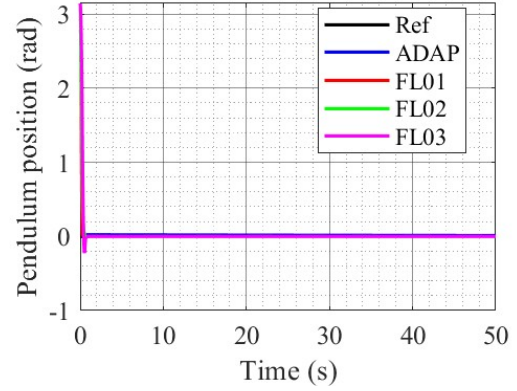
5.2.3 Simulation of the proposed control system and analysis of its performance

In the first test conducted on this system with the various proposed control strategies, the initial state of the system sees the pendulum at $\theta = \pi$ radians (hanging vertically downwards) with the state setpoints set to $[x_d, \dot{x}_d, \theta_d, \dot{\theta}_d] = [0, 0, 0, 0]$ where the subscript d denotes the “desired” value of these state variables. After 5 seconds have elapsed, the cart displacement setpoint varies according to $x_d = \sin(t - t_{\text{switch}})$, where t_{switch} is the time at which this motion is triggered. There is no disturbance present in this first test.

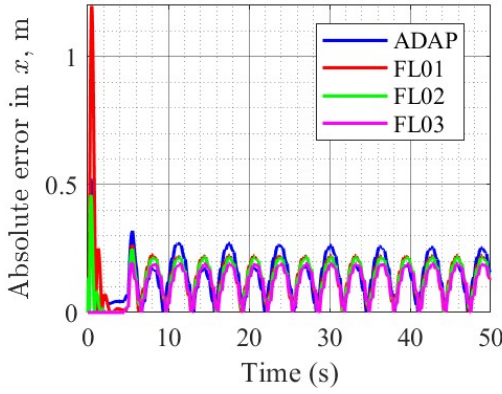
The trajectory tracking results in this simulation are shown in Fig 5.12. Additionally, the efforts exerted by both control actions in each strategy are illustrated in Figs 5.13 and 5.14.



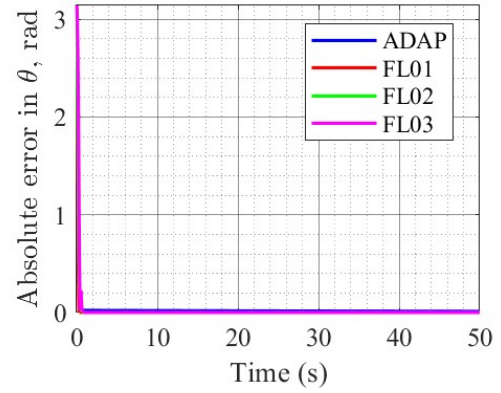
(a) Cart position tracking.



(b) Pendulum position tracking.



(c) Absolute error in cart position tracking.



(d) Absolute error in pendulum position tracking.

Figure 5.12: Results of the zero disturbance test case with constant pendulum position setpoint and sinusoidal cart position setpoint.

The preliminary results presented in Fig 5.12 suggest that the ADAP prediction model is less effective for trajectory tracking in both x and θ (though more noticeably in the latter case). Additionally, the control effort expended when using the ADAP prediction model is neither higher nor lower than the effort expended when using the other three strategies. In the case of both control actions, F shown in Fig 5.13 and τ shown in 5.14, the FL02 strategy exhibits the lowest expended effort.

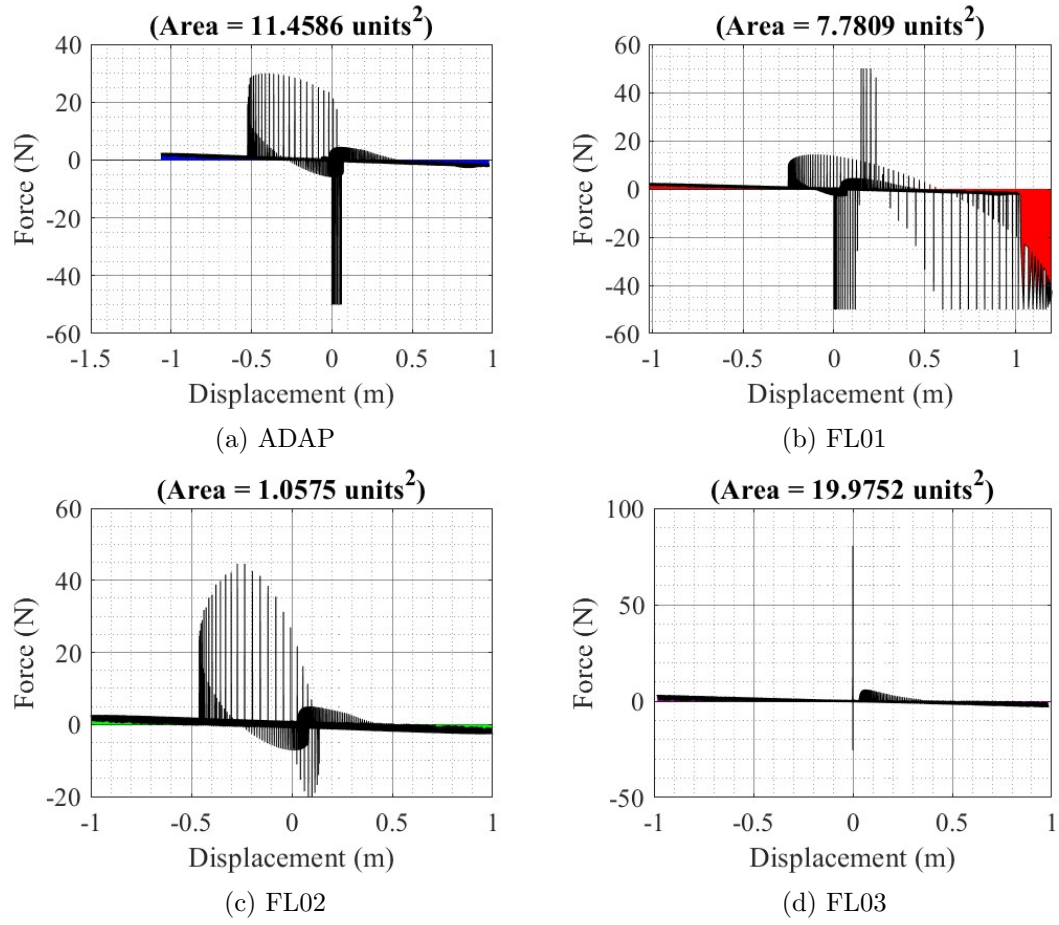


Figure 5.13: Cart control effort exerted through F by each of the control strategies to achieve the trajectory tracking shown in Fig 5.12.

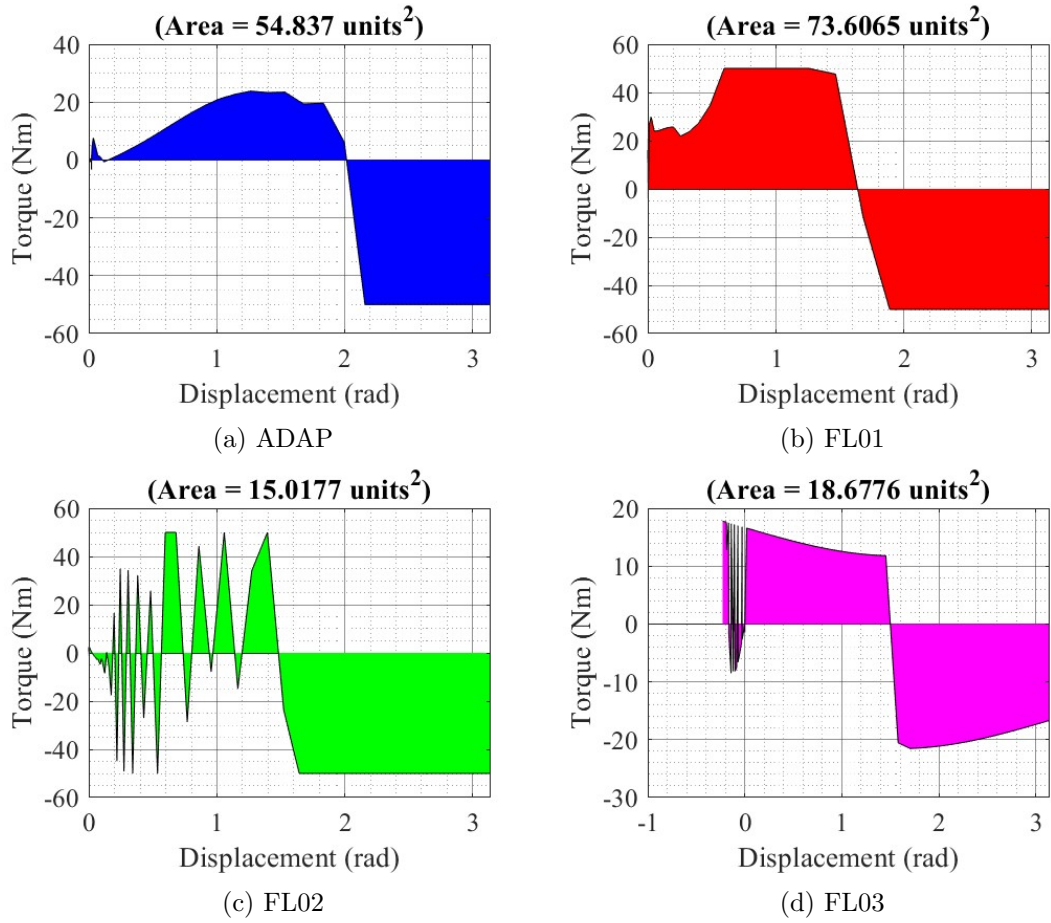
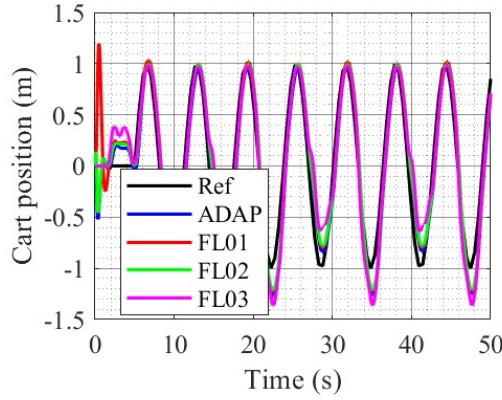


Figure 5.14: Pendulum control effort exerted through τ by each of the control strategies to achieve the trajectory tracking shown in Fig 5.12.

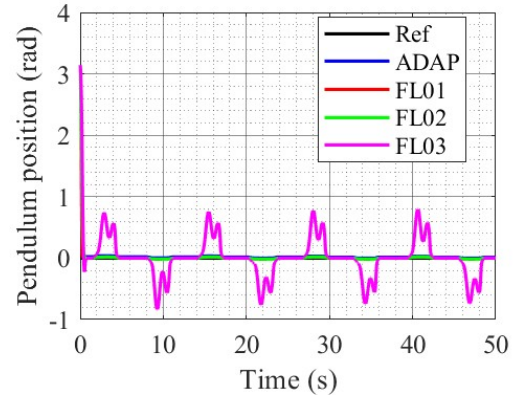
These results appear to conflict with the observations made in the preceding example (the inverted pendulum), suggesting a potential limitation in the ADAP strategy's ability to model strong coupling interactions. A range of other tests was then conducted to assess the effectiveness of the four proposed linearised control approaches, each using the same setpoints as the preceding test which led to Figs 5.12 - 5.14.

A second test saw the introduction of a harmonic disturbance to both manipulated variables. In both cases, the disturbance (T_d) was modelled by

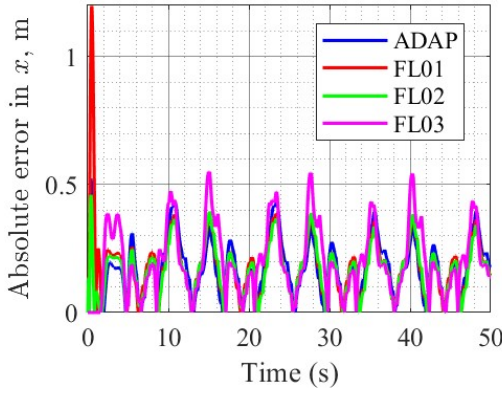
$$T_d(t) = \sin(0.5t). \quad (5.37)$$



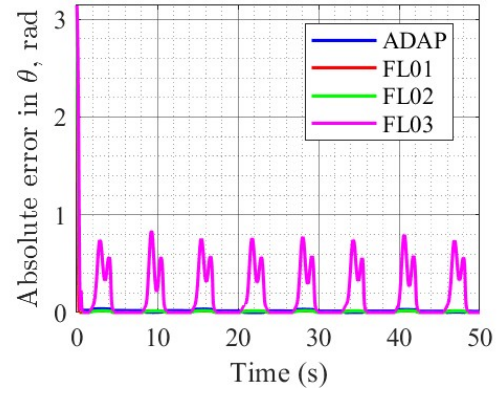
(a) Cart position tracking.



(b) Pendulum position tracking.



(c) Absolute error in cart position tracking.



(d) Absolute error in pendulum position tracking.

Figure 5.15: Results of the test case with constant pendulum position setpoint and sinusoidal cart position setpoint with an imposed disturbance on each manipulated variable as described by Eq (5.37).

In this set of results, which illustrates the capacity of the various control strategies to account for disturbances described by Eq (5.37), the performance of the ADAP method is improved relative to the other three approaches. The errors illustrated in Fig 5.15 for the ADAP method are substantially closer to the highest performing of the four strategies, with FL03 being the worst performing approach (in terms of absolute error) for both generalised coordinates.

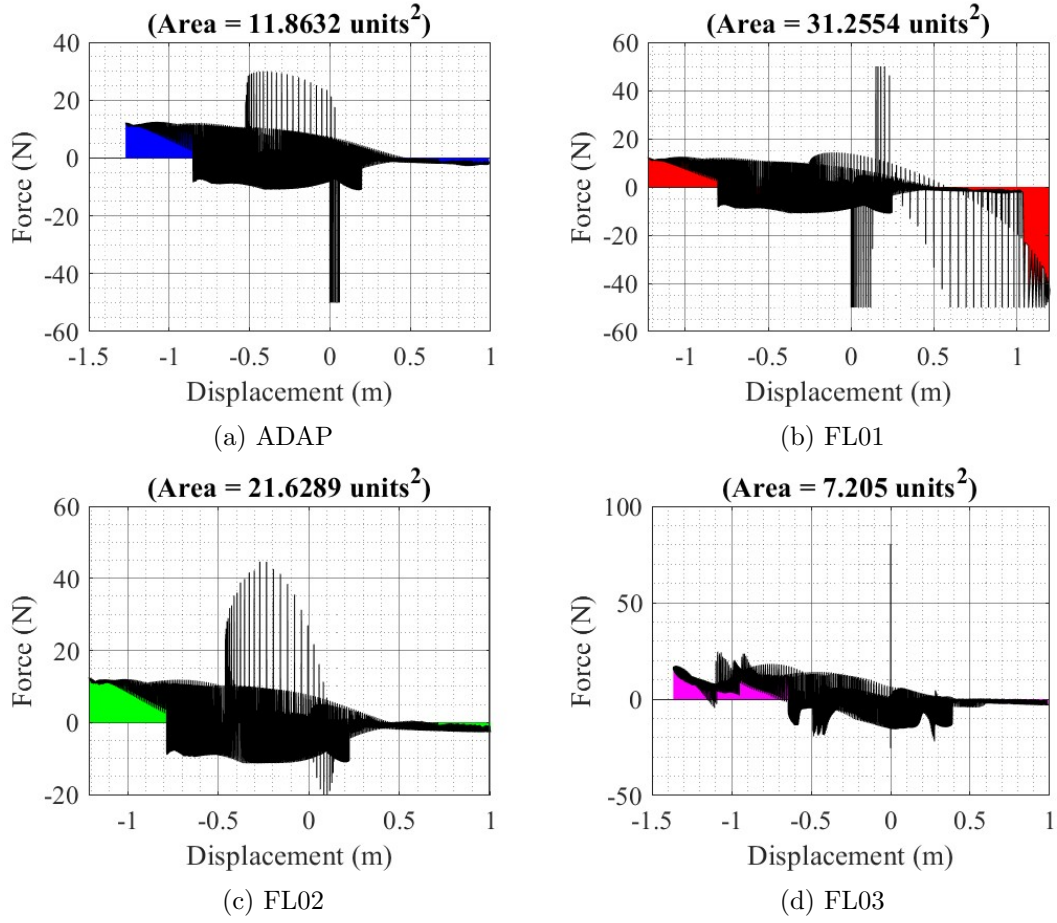


Figure 5.16: Cart control effort exerted through F by each of the control strategies to achieve the trajectory tracking shown in Fig 5.15.

The ADAP methodology remains neither the highest nor lowest in terms of control effort exertion. However, an interesting observation arises from a comparison of the control efforts in Figs 5.16 and 5.17 with comparison to Figs 5.13 and 5.14. Notably, the ADAP method and FL03 lead to similar energy exertion across both of these experiments, while FL01 and FL02 both exhibit substantial increases in control effort in the test involving the disturbance, Eq (5.37). Note, in Table 5.4, that while the ADAP method shows only a small variation in the work done by F and a negligible variation in the work done by τ , FL03 shows a significant decrease and increase across these two values respectively.

Method	Work done by F (units ²)			Work done by τ (units ²)		
	$T_d = 0$	$T_d = \sin(0.5t)$	% increase	$T_d = 0$	$T_d = \sin(0.5t)$	% increase
ADAP	11.4586	11.8632	4%	54.837	55.0437	0%
FL01	7.7809	31.2554	302%	73.6065	43.8502	0%
FL02	1.0575	21.6289	1945%	15.0177	15.346	2%
FL03	19.975	7.205	-64%	18.0776	29.3049	62%

Table 5.4: Comparison of the control effort exertions for each control approach across the zero disturbance and disturbance-included test cases for the cart-mounted pendulum system.

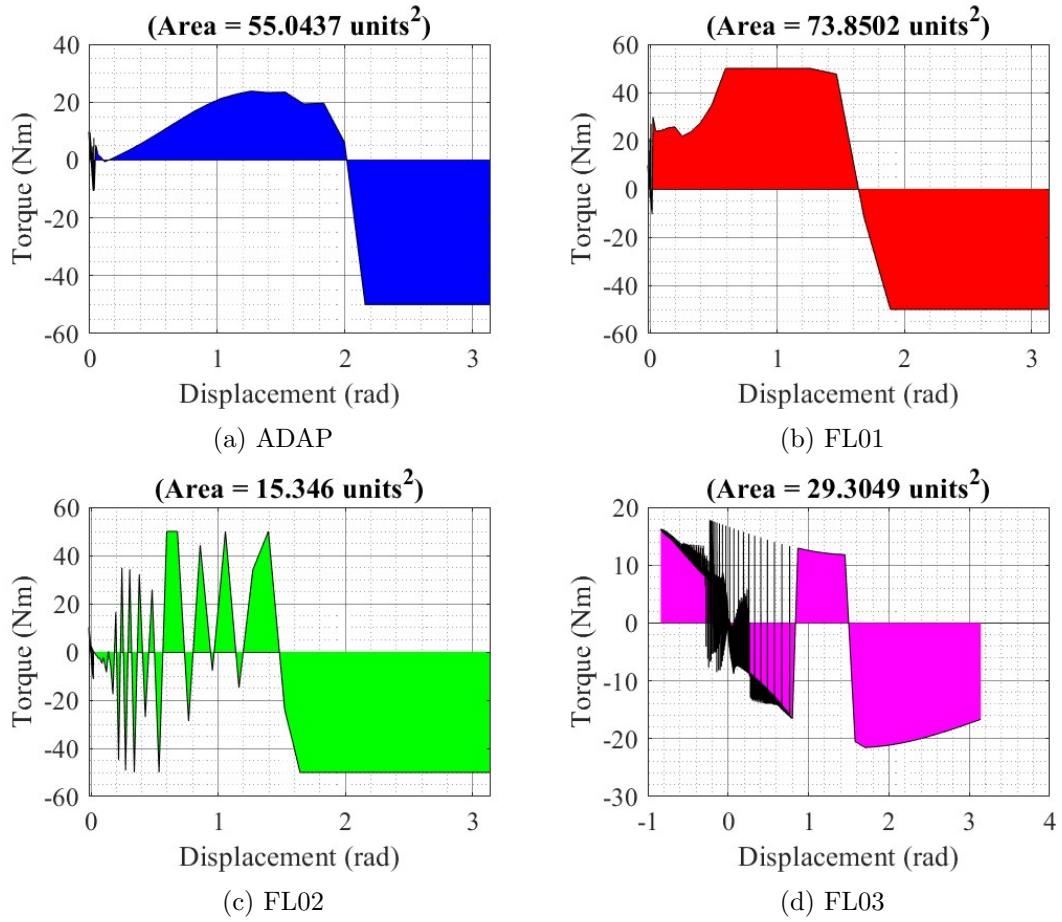


Figure 5.17: Pendulum control effort exerted through τ by each of the control strategies to achieve the trajectory tracking shown in Fig 5.15.

When tested with disturbances proportionally larger than Eq (5.37), results illustrate that the FL03 approach remains the least effective approach for trajectory tracking

Chapter 5. Application of the novel approximation algorithm to model predictive controllers

accuracy, while the ADAP approach remains closer to FL01 and FL02 in its accuracy. Disturbances proportionally smaller than Eq (5.37) resulted in the ADAP and FL03 approaches swapping roles as the least effective for trajectory tracking.

Conducting one final test on this example, to further explore the change in effectiveness of the ADAP method in the presence of disturbances to the manipulated variables, a white noise disturbance signal is added in place of Eq (5.37). The percentage increases in exerted control efforts, shown in Table 5.5, are based on a white Gaussian noise (WGN) signal generated by the MATLAB function `0.9*wgn(n,1,0)`, which produces 1 row of n Gaussian noise samples (chosen to align with the number of samples in the reference trajectories) with a power of 0 dBW. FL03 resulted in the pendulum undergoing several complete rotations about its axis, whereas the other three approaches only produced minor deviations from the reference trajectory, θ_d .

Method	Work done by F (units ²)			Work done by τ (units ²)		
	zero	dis-	WGN dis-	zero	dis-	WGN dis-
	turbance	turbance	turbance	turbance	turbance	turbance
ADAP	11.4586		78.92	54.837	57.34	5%
FL01	7.7809		98.18	73.6065	73.8502	4%
FL02	1.0575		87.87	15.0177	18.33	22%
FL03	19.975		143.49	18.0776	1486.96	8125%

Table 5.5: Comparison of the control effort exertions for each control approach across the zero disturbance and white Gaussian noise (WGN) disturbance test cases for the cart-mounted pendulum system.

5.3 Conclusion

This chapter has demonstrated the application of the ADAP methodology to the model-predictive control of two well-understood, unstable dynamical systems: a simple inverted pendulum and a cart-mounted inverted pendulum. Exploration of these two examples provides useful insight into the effectiveness, strengths, and limitations of the ADAP method's role as a prediction model in MPC.

In the first example of the simple inverted pendulum, the ADAP-MPC method is

Chapter 5. Application of the novel approximation algorithm to model predictive controllers

superior to the two traditional feedback linearisation approaches (FL01 and FL02) in its ability to track the reference trajectory. While the absolute control torque value in the steady-state remains higher for the ADAP-MPC method than for FL02, the energy expended (work done) by the torques is lower when the ADAP-MPC method is used. This is attributed to the ADAP-MPC strategy's reduced settling time with comparison to FL02 (meaning that the integral of the torque signal over the resulting displacement is lower in the former case).

In the second example of the cart-mounted inverted pendulum, the performance of the ADAP-MPC method is not distinctly superior to the three traditional feedback linearisation approaches (FL01, FL02, and FL03). The ADAP-MPC strategy in fact leads to greater trajectory tracking errors in the case of no external disturbances affecting the manipulated variables (the cart force and pendulum torque). However, in the presence of disturbances acting on the manipulated variables, the ADAP-MPC strategy presents a clear advantage. Depending on the application, this may not be viewed as substantial enough to warrant the choice of the ADAP-MPC strategy by comparison to the other, simpler approaches. However, it should be noted that the ADAP-MPC strategy's control effort experiences smaller increases in the presence of disturbances than the other three approaches.

Considering these examples, the means by which the ADAP method can be applied to construct a prediction model for an MPC system (referred to as the ADAP-MPC strategy) is demonstrated. Across these examples, this chapter demonstrates a simple, single-input single-output case where the ADAP-MPC strategy results in a clear benefit to both trajectory tracking accuracy and energy expenditure. An important assumption which is made in both of these examples (as well as in the subsequent case study in Chapter 6) is that all states are observable. In practice, if this assumption was not possible, an appropriate means of state observation/estimation would be required.

Furthermore, this chapter presents a case in which the ADAP-MPC strategy offers

Chapter 5. Application of the novel approximation algorithm to model predictive controllers

a more nuanced trade-off between trajectory tracking accuracy and energy efficiency in the presence of disturbances. In the case of the cart-mounted pendulum, the trajectory tracking accuracy of the ADAP-MPC strategy is lower than at least two of the standard feedback linearisation methods (depending on the imposed disturbance signal). However, when subjected to disturbances acting on the manipulated variables, the ADAP-MPC strategy demonstrates a reduced increase in energy expenditure with comparison to the feedback linearisation techniques used.

Chapter 6

Case study Part 2: control of the *FutureForge* manipulator

Inclusion of previous work

This chapter includes content from previous work by the author. This is *Application of the ADAP method to model-based control of the FutureForge manipulator*, which is published in *Journal of Sound and Vibration* [44]. This earlier work, by D Johnston and MP Cartmell, features the first use of the ADAP-MPC method, which is the central focus of this chapter of this thesis.

This chapter integrates the concepts and methods of the preceding chapters to demonstrate the application of the ADAP methodology within a model-predictive control (MPC) strategy applied to the FutureForge manipulator. While the control of industrial manipulators has been extensively researched (as cited in Chapter 4), the novel ADAP methodology has not previously been applied in this specific context. The primary objective of this case study is to evaluate the effectiveness of a strategy which integrates the ADAP methodology and an MPC framework for manipulator control. Section 6.1 details the construction of the virtual manipulator model and the configuration of the MATLAB environment to suit this purpose. Section 6.1.2 outlines the ADAP-MPC application, concluding with a presentation and discussion of the results in Section 6.2.

Before undertaking the described case study, the *FutureForge* manipulator system previously described in Chapter 3 should be recalled. This is important because it demonstrates the relevance of developing an effective control strategy for this system. In particular, as noted in Chapter 3, the function of the manipulator in the *FutureForge* system is to manoeuvre material specimens through forging operations. These manoeuvres may consist of transferring the material to and from the furnace, and carefully handling the specimen throughout its treatment in the hydraulic press. As described in Chapter 3, the 2-dimensional workspace of the manipulator is modelled in Eqs (3.37) and (3.38). Only the behaviour modelled in these equations is controlled in this case study chapter.

6.1 Development of the virtual model of the manipulator

This section outlines how the MATLAB environment was configured to facilitate the simulations relied upon in this case study.

6.1.1 Configuration of the simulation environment

The simulation environment for the *FutureForge* manipulator was configured within *Simulink*, particularly using the *Simscape* toolbox, to create a high-fidelity digital representation of the physical system. The manipulator's components (as detailed in Chapter 3) were represented in *Simscape* by importing the manufacturer-provided STEP files using *File Solid* blocks. This approach ensured a high degree of fidelity in the model representation. Careful inspection of the STEP files and associated engineering drawings was undertaken to establish accurate rigid coordinate transforms between components, thereby preserving the previously derived kinematic and dynamic properties of the manipulator.

The *Simscape* model served as the plant model within the broader simulation, generating the necessary measured output states and receiving the control actions from the MPC block. The four system states, $\mathbf{x} = [\alpha, \dot{\alpha}, \beta, \dot{\beta}]^T$, were taken as outputs from

the *Simscape* model and passed along two paths. The first path directed the signal to the “measured outputs” port of the MPC block. The second path, meanwhile, directed the output states into a custom MATLAB function block. This block contained the solution to the ADAP methodology’s first auxiliary problem, providing data that was then interpreted as a measured disturbance to the linearised plant model used by the MPC prediction algorithm. This novel configuration enabled the ADAP approach to provide dynamic corrections to the prediction generated by the simple, linearised plant model.

Solid body properties (mass, inertia, centre of gravity, etc.) and geometries were also extracted from manufacturer-provided information. This data was collated within a centralised MATLAB script, ensuring that the environment could be initialised and updated with revised physical parameters without requiring modifications to the dependent *Simscape* model.

All simulations were executed using the variable step-size ode45 (Dormand Prince) solver within *Simulink*. A list and description of the various MATLAB-based numerical solvers is detailed in [44].

6.1.2 The ADAP-MPC strategy

Using the same ADAP procedure as described and assessed in Chapters 2 and 3, the ADAP methodology and MPC approach can be combined as in the examples in Chapter 5. Combining these approaches, as the ADAP-MPC strategy, required the ADAP model to be considered in the form

$$\begin{bmatrix} a_1 & 0 \\ 0 & b_1 \end{bmatrix} \begin{Bmatrix} \ddot{\alpha}_0 \\ \ddot{\beta}_0 \end{Bmatrix} + \begin{bmatrix} a_8 & 0 \\ 0 & b_8 \end{bmatrix} \begin{Bmatrix} \dot{\alpha}_0 \\ \dot{\beta}_0 \end{Bmatrix} = \begin{Bmatrix} Q_v \\ Q_h \end{Bmatrix} + \begin{Bmatrix} w_v \\ w_h \end{Bmatrix} \quad (6.1)$$

where the vector $\mathbf{w} = [w_v, w_h]^T$ is treated as a measured disturbance by the controller. Note that this form reflects the standard ADAP forms in Eqs (3.60) and (3.61) in Chapter 2, where $\mathbf{w} = \left(\hat{M}(\Phi)\ddot{\Phi} + \hat{P}(\Phi, \dot{\Phi})\dot{\Phi} \right)$.

This representation of the system is then converted into a state-space representation as described in [44]. The state vector consists of the four measured outputs, $\mathbf{x} = [\alpha, \dot{\alpha}, \beta, \dot{\beta}]^T$, and evolve according to the state differential equation

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{a_8}{a_1} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{b_8}{b_1} \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 & 0 \\ \frac{1}{a_1} & 0 \\ 0 & 0 \\ 0 & \frac{1}{b_1} \end{bmatrix} \begin{Bmatrix} Q_v \\ Q_h \end{Bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{1}{a_1} & 0 \\ 0 & 0 \\ 0 & -\frac{1}{b_1} \end{bmatrix} \begin{Bmatrix} w_v \\ w_h \end{Bmatrix} \quad (6.2)$$

For the purposes of this work, it is assumed that all of the states in $\mathbf{x}$ can be obtained through appropriate sensing and estimation approaches. There is a substantial body of literature devoted to state observation and estimation, although this falls outside the scope of the present work. As noted briefly at the end of the preceding chapter, if it were not possible to observe all of the required states for this approach, then an appropriate alternative approach would need to be adopted from the existing literature. For the current work, it is assumed that full-state feedback (of the displacement and velocity states, α , β , $\dot{\alpha}$, and $\dot{\beta}$) is possible. In reality, it is likely that direct sensing of the displacement states would be possible, but the velocity states would need to be estimated through the use of accelerometers or similar equipment. Thus, for the present work, the state output equation for this implementation simply consists of the 4×4 identity matrix, $\mathbf{I}$,

$$\mathbf{y} = \mathbf{I}\mathbf{x} \quad (6.3)$$

where $\mathbf{y}$ is the output signal fed back to the controller as the measured outputs. In Fig 6.1, the “estimation of omitted effects” block computes the terms w_v and w_h in Eq (6.2) as the measured disturbances to the linear prediction model used by the MPC algorithm.

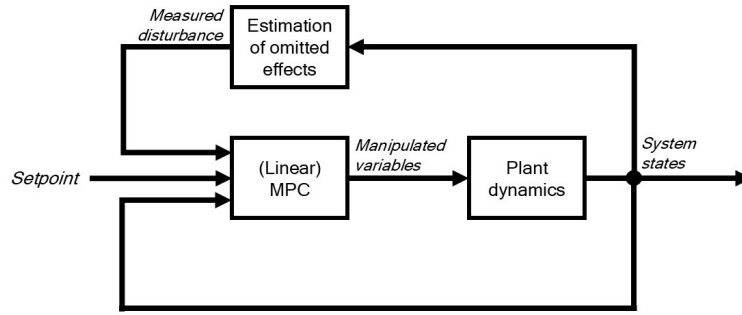


Figure 6.1: A simplified block diagram representation of the ADAP-MPC control strategy applied to the *FutureForge* manipulator, reproduced from [44].

Using this state-space model, the command-line interface was used to design the model-predictive controller and assign values to its various parameters (instead of the built-in “MPC Designer” app). As described in [44], the input signals were specified as either manipulated variables, Q_v and Q_h , or measured disturbances, w_v and w_h .

The horizon parameters (including the prediction and control horizon lengths and the sampling time) were then tuned according to guidance in [112] - setting the control horizon to be half of the prediction horizon and manually observing the impact of small changes to this. Note that, following from the Chapter 4 discussion of MPC, the definition of these horizon lengths in MATLAB is an integer value indicating the number of time steps it contains.

As also noted in Chapter 4, an advantage of MPC is the capacity to account for constraints within the optimisation of the manipulated variables. For the present case study, this included the physical limits on the generalised coordinates, $\alpha \in [-17.5^\circ, 17.5^\circ]$ and $\beta \in [60^\circ, 120^\circ]$. The physical constraints arising from the maximum pressures offered by the hydraulic actuators result in necessary constraints on the actuation torques, $Q_v \leq 1$ MNm and $Q_h \leq 1$ MNm (which were arrived upon following consultation with industrial partners on this project). Note that there are two actuators which contribute to this maximum value of Q_v , while only one (larger) actuator provides Q_h .

To form an appropriate cost function for its optimisation computations, the MPC design must specify relative weightings signifying the relative “importance” of tracking each measured output’s reference signal. Note, these weightings are relative: if the weight assigned to the first measured output $\mathbf{y}$, y_1 , is 1, while the weight assigned to y_2 is 0.5, this implies that accurately tracking y_1 is “twice as important” as accurately tracking y_2 . While the controller may still provide sufficient tracking of y_2 , it will more harshly penalise errors in y_1 and manipulate the control actions accordingly to address this. In this case study, the weights assigned to the displacement states are several orders of magnitude larger than the weights assigned to the velocity states (see Table 6.1). While relatively small, the velocity states are not simply zero because this would hinder the controller’s ability to reach a displacement setpoint **and** bring the end-effector to rest at this position. As described in [44], through the manual tuning process, the least desirable performance was obtained when all weights were equal (in the earliest stages of tuning). Performance was observed to improve where the displacement weights were at least 10-times larger than the velocity weights. However, further improved performance is found with the values stated in Table 6.1.

Lastly, the Scale Factor was also a key parameter in the tuning of the controller. This parameter is a measure of each variable’s domain of possible values. As such, these are often obtained by simply finding the range of the relevant domain. For the measured disturbance terms, however, these values were obtained empirically by inspecting the range of values obtained in early experiments.

Variable	Min value	Max value	Weighting	Scale Factor	Nominal value
α	-0.305 rad	0.305 rad	50	0.611	0 rad
$\dot{\alpha}$			0.01	1	0 rad/s
β	$\frac{\pi}{3}$ rad	$\frac{2\pi}{3}$ rad	100	$\frac{\pi}{3}$	$\frac{\pi}{2}$ rad
$\dot{\beta}$			0.01	1	0 rad/s
Q_v	-1 MNm	1 MNm		2×10^6	-8.61×10^3 Nm
Q_h	-1 MNm	1 MNm		2×10^6	-190.2 Nm
w_v				0.5×10^5	0 Nm
w_h				1.5×10^5	0 Nm

Table 6.1: MPC parameter settings used in the control of the *FutureForge* manipulator, reproduced and revised from [44].

6.1.3 Test manoeuvres

This section defines the test manoeuvre inputs used to evaluate the ADAP-MPC strategy on the *FutureForge* manipulator model (introduced in Chapter 3, with the controller described in Section 6.1.2). These manoeuvres are broadly categorised into two types of manoeuvre, designed to mimic the full dynamic range and the high-precision requirements of the manipulator’s real-world operation. Test 1 is an exploration of the entire workspace, and Tests 2-5 are replications of the manoeuvres requiring the highest accuracy.

Test 1 features the broadest displacement range of manoeuvres. This test systematically examines the motion between every possible pair of defined points in the workspace (as illustrated in Fig 6.2). In other words, the manipulator begins at the Near-Lower (NL) position before moving to the Mid-Lower (ML) point and back; then, the end-effector is moved to the Far-Lower (FL) position before returning to NL and being directed to arrive at the NM, MM, FM, NU, MU, and FU points sequentially (returning to NL) each time. Then, the process is repeated with ML being the point to which each manoeuvre returns; then, this repeats for FL as the point of return, and so on. This comprehensive suite of test manoeuvres ensures that the motion between every pair of identified points in the workspace is examined. Through this set of manoeuvres, the full 2-dimensional workspace of the manipulator is explored.

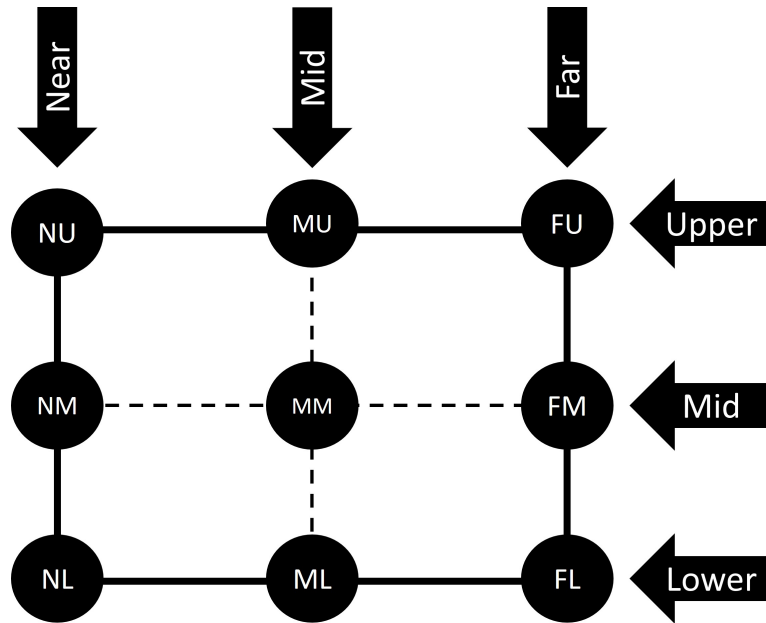


Figure 6.2: 2D workspace of the manipulator divided into nine characteristic positions. These positions are given a two-letter designation according to their proximity and elevation with respect to the manipulator base. Reading these from top left to bottom right: “near upper”, “mid upper”, “far upper”, “near mid”, “mid mid”, “far mid”, “near lower”, “mid lower”, and “far lower”.

Tests 2-5 are, by contrast, designed to emulate the scenarios in which the manipulator requires the greatest accuracy: feeding specimens into the press. All four tests simulate the feeding of a 4-metre-long specimen into the press. Tests 2 and 3 are conducted at 60 steps per minute with a step penetration of 50 mm. Test 2 is conducted with no payload attached to the end-effector, while Test 3 is conducted with the 4-metre-long specimen with a mass of 1 tonne. Tests 4 and 5 are conducted at 100 steps per minute with a step penetration length of 5 mm. Similarly to Tests 2 and 3, Tests 4 and 5 explore the manoeuvre without and with a payload of 1 tonne respectively. These specific parameters and operational rates were defined in consultation with the AFRC and Clansman Dynamics, reflecting the acceptance testing criteria for the delivered physical manipulator.

Test	Step penetration (mm)	Step rate (steps per minute)	Payload mass (kg)
Test 2	50	60	0
Test 3	50	60	1000
Test 4	5	100	0
Test 5	5	100	1000

Table 6.2: Parameter specification for Tests 2-5, indicating variations in payload mass, step penetration depth, and step rate.

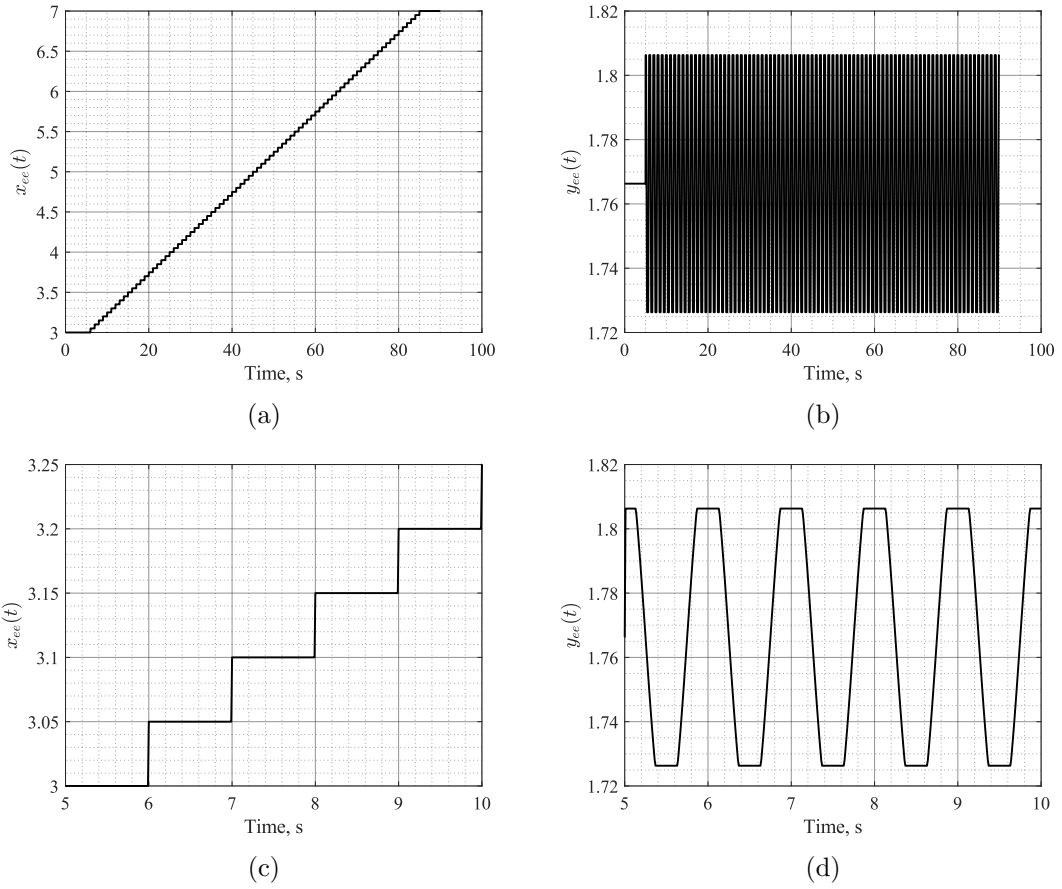


Figure 6.3: End-effector Cartesian positions, (x_{ee}, y_{ee}) , for Tests 2 and 3.

The required Cartesian trajectories for the end-effector, (x_{ee}, y_{ee}) , in Tests 2-5 are illustrated in Table 6.2. The generalised coordinates $(\alpha(t), \beta(t))$ are subsequently determined by solving the inverse kinematic problem using MATLAB's *fmincon* function.

6.1.4 Examining the computational expense of the ADAP method

The computational cost associated with an approximation algorithm is an important consideration. Of course, the precise measure and implication of this cost can often be subject to context: the computation of the approximation may be executed offline; portions of the algorithm may be simplified or substituted to account for known results; or the computation may be suitable for parallel processing on appropriate hardware. Given the breadth of factors which can change how the expense is measured, a simple, transparent method is detailed here to examine the computational cost of the ADAP method by comparison to the standard approach used earlier in this section.

Seeking to achieve a metric which is independent of hardware, the measured variable is the number of function evaluations computed by MATLAB's ordinary differential equation solvers. A description of these solvers is given in [44]. To give a useful frame of reference for this expense, the computational cost of a "reference" method, in addition to the standard prediction model, is also found. While this algorithm uses the receding horizon principle to correct its predictions over time, the control actions at each stage are prescribed, so no optimisation occurs. This algorithm is not designed to achieve trajectory tracking, simply to achieve a measure of the number of function calls required by each solver for each prediction model.

The reference model is simply the fully nonlinear, coupled model. The standard model, as described previously, consists only of the ADAP method's first auxiliary problem (with no corrections). And, the ADAP model is as explained previously. The scenario modelled through this test of the computational expense is arbitrary, and so the excitations are designed only to ensure the manipulator remains within its constrained physical limits and explores a reasonable range of the workspace. In the definitions of Q_v and Q_h in this test, the added constants are provided to ensure that the restoring forces for the system are sufficiently counteracted, while the amplitudes and frequencies of the trigonometric terms are chosen to allow a substantial range of α and β to be

spanned in the simulation.

$$Q_v = 20 \cos(0.75t) + 8.45, \text{ in kNm} \quad (6.4)$$

$$Q_h = 0.7 \sin(0.75t) + 195, \text{ in kNm} \quad (6.5)$$

6.1.5 Simulated data

The *Simulink* environment was set up to log a variety of signals throughout the simulation of each test scenario. As noted previously, the four system states were logged. These were essential for verifying the effectiveness of the controller as they correspond to the manipulator's actual path when required to follow the prescribed reference trajectories. The two control actions (output by the MPC block and taken as inputs to the manipulator's *Simscape* representation) were also recorded. Although not essential, as the MPC algorithm ensured constraints such as limitations on the manipulated variables were enforced, this gave valuable feedback when testing and tuning early versions of the controller. The two measured disturbance signals (resulting from the ADAP formulation as described previously) were also recorded to allow inspection and aid in understanding through the development of the algorithm.

The reference trajectory signals used to define the manoeuvres were formulated over a discretised time domain using a time step equal to the MPC sampling time, ensuring consistency and ease of comparison across collected data.

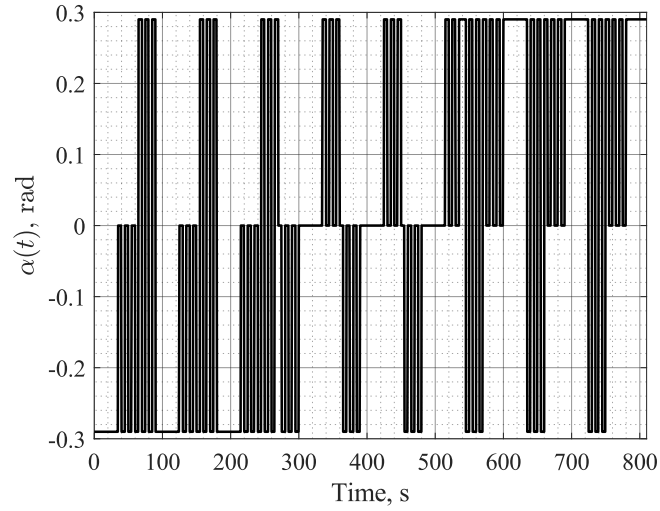
6.2 Simulation and testing under proposed control system

This section presents the results and accompanying analysis of the performance of the proposed ADAP-MPC control strategy in conducting Tests 1-5, detailed in the preceding section and partially illustrated in Table 6.2. Tests 1, 2, and 4 are first considered to assess the trajectory tracking accuracy for the proposed control system. The control effort of this strategy will then be analysed and compared against that of a standard MPC approach. Lastly, Tests 3 and 5 are considered to demonstrate how

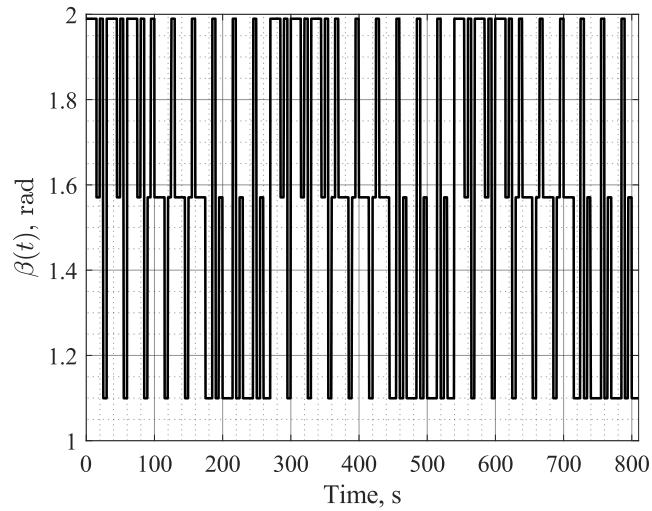
the inclusion of a payload alters the success of this novel control approach. For clarity, the only difference between the ADAP-MPC approach and the standard approach to which it is compared is the use of measured disturbance terms as corrections to the simplified plant model. In the standard approach, no measured disturbance terms are used in this way. All MPC parameters, such as horizon parameters and constraints, are identical in both cases. This is to ensure that the following comparison accurately shows the impact of using the ADAP-MPC method in place of an alternative.

6.2.1 Trajectory tracking control results

As previously noted, Test 1 demonstrates the manipulator moving between every pair of marked points in the workspace (as highlighted in Fig 6.2). The reference trajectories and absolute errors for the full 810-second duration are illustrated in Fig 6.4, but smaller intervals in the time domain are illustrated through Figs 6.5-6.7.



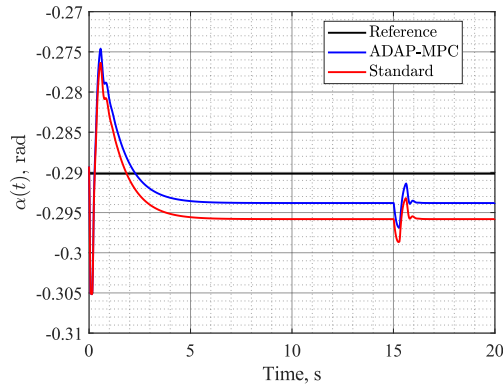
(a) Reference trajectory for the vertical actuation angle, $\alpha(t)$ radians.



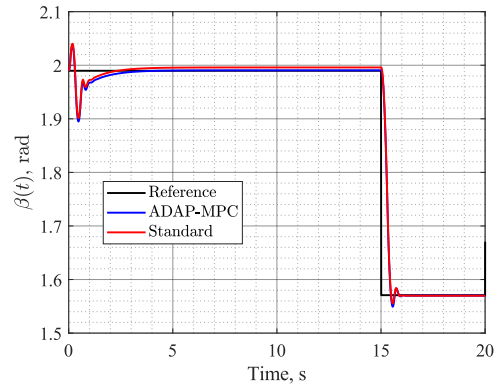
(b) Reference trajectory for the horizontal actuation angle, $\beta(t)$ radians.

Figure 6.4: Reference trajectories for Test 1.

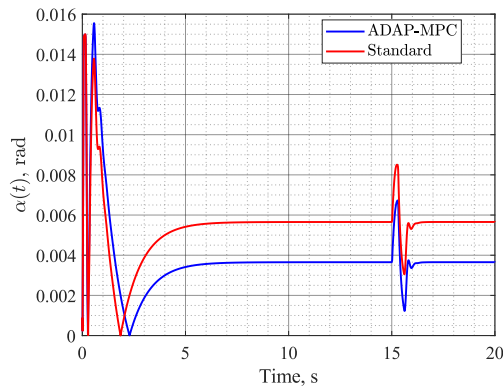
Fig 6.5 shows the trajectory tracking of the generalised coordinate setpoints when the manipulator steps from its NL configuration to the ML configuration. The setpoint for α is constant, while the setpoint for β steps from its maximum to the midpoint of its range.



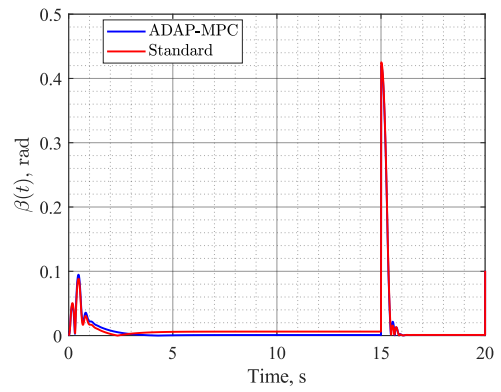
(a) Vertical actuation angle, $\alpha(t)$ radians.



(b) Horizontal actuation angle, $\beta(t)$ radians.



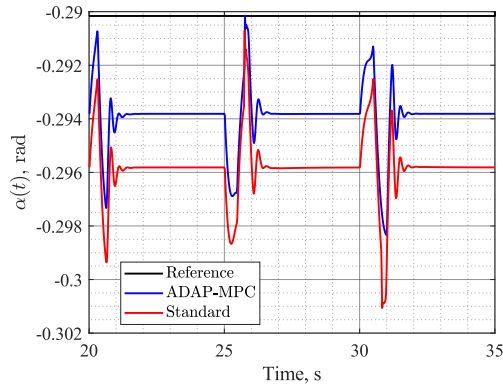
(c) Absolute error in the trajectory tracking for $\alpha(t)$.



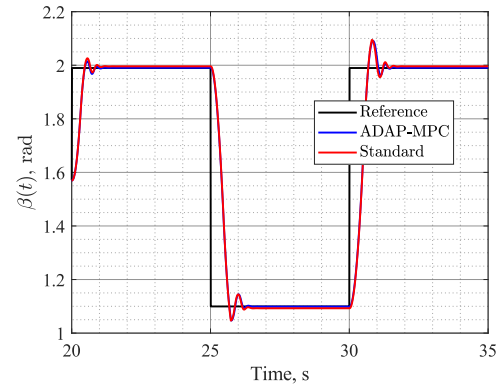
(d) Absolute error in the trajectory tracking for $\beta(t)$.

Figure 6.5: Reference trajectories and absolute error measures for Test 1 across the interval $t \in [0, 20]$ seconds.

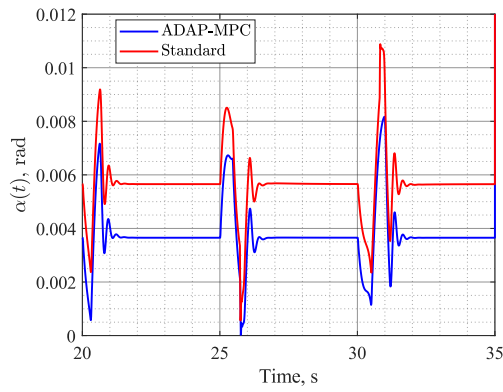
Illustrating the 15 seconds following Fig 6.5, Fig 6.6 then shows the end-effector moving from the NL configuration to and from the FL configuration.



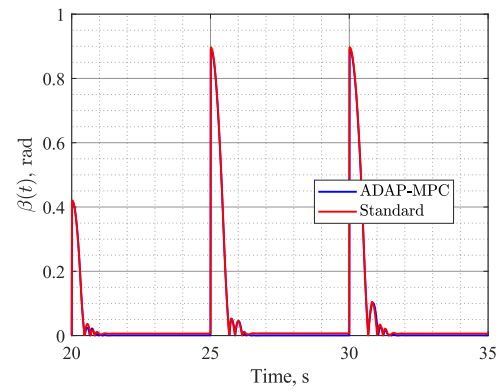
(a) Vertical actuation angle, $\alpha(t)$ radians.



(b) Horizontal actuation angle, $\beta(t)$ radians.



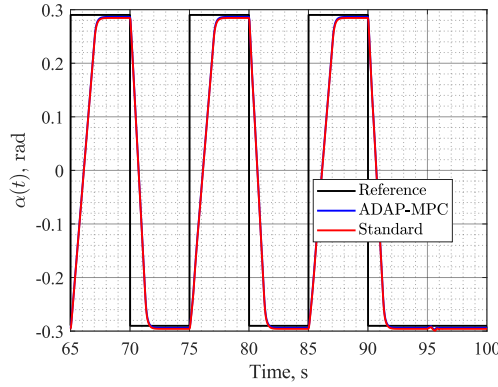
(c) Absolute error in the trajectory tracking for $\alpha(t)$.



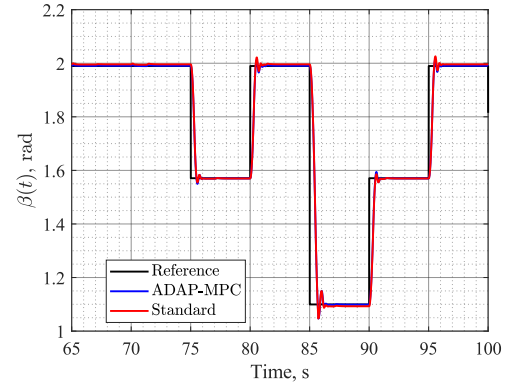
(d) Absolute error in the trajectory tracking for $\beta(t)$.

Figure 6.6: Reference trajectories and absolute error measures for Test 1 across the interval $t \in [20, 35]$ seconds.

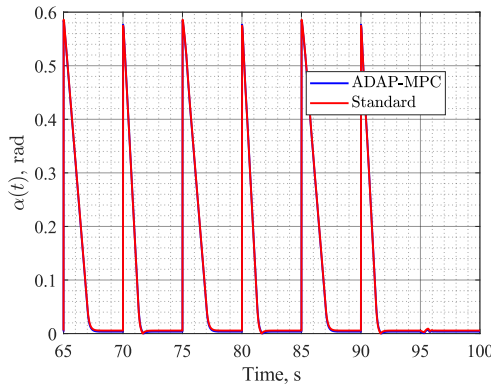
Fig 6.7 then depicts the manipulator's manoeuvres between the NL configuration and the NU, MU, FU, and ML configurations.



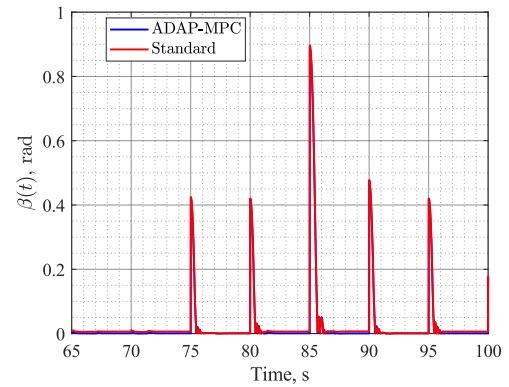
(a) Vertical actuation angle, $\alpha(t)$ radians.



(b) Horizontal actuation angle, $\beta(t)$ radians.



(c) Absolute error in the trajectory tracking for $\alpha(t)$.



(d) Absolute error in the trajectory tracking for $\beta(t)$.

Figure 6.7: Reference trajectories and absolute error measures for Test 1 across the interval $t \in [65, 100]$ seconds.

Time interval	Integral of squared error in α (rad^2s^2)		Integral of squared error in β (rad^2s^2)	
	ADAP-MPC	Standard	ADAP-MPC	Standard
$0 \leq t \leq 20$	2.37×10^{-4}	3.25×10^{-4}	3.91×10^{-2}	4.021×10^{-2}
$20 \leq t \leq 35$	2.44×10^{-5}	1.95×10^{-5}	0.55	0.55
$65 \leq t \leq 100$	1.26	1.26	0.41	0.42

Table 6.3: The integral of squared error for each coordinate for both control strategies across the time intervals shown for Test 1 in Figs 6.5-6.7.

Hydraulic press penetration manoeuvres

For Test 2, three intervals of the time domain are illustrated across Figs 6.8-6.11. These are selected so as to capture performance variations which may occur as a result of improved/reduced accuracy in the vicinity of specific operating points. Note that the reference trajectory for α is identical in all three intervals (and for the rest of the simulation), and the observed absolute errors are also indistinguishably different. The performance in β more significantly varies across the duration of the test.

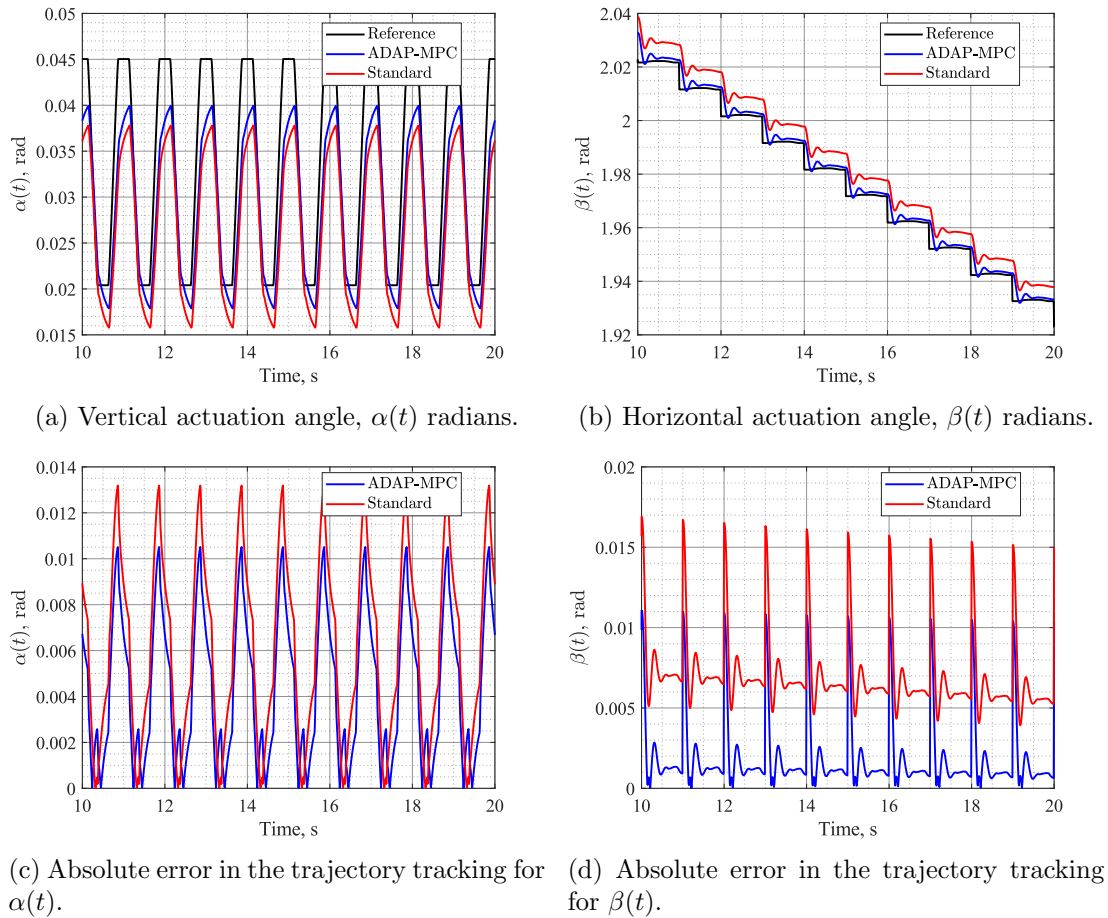
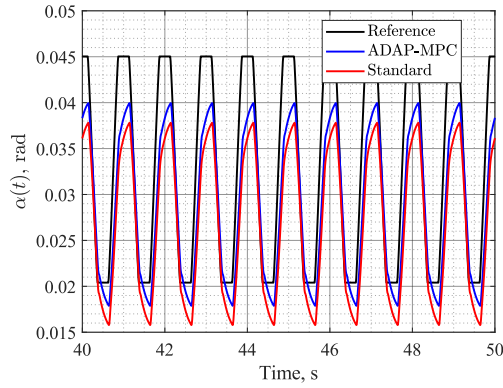


Figure 6.8: Reference trajectories and absolute error measures for Test 2 across the interval $t \in [10, 20]$ seconds.

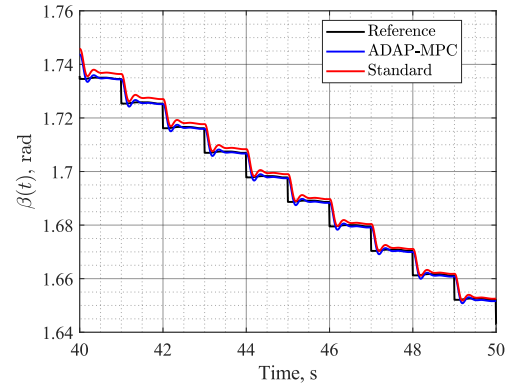
In Fig 6.8, the error observed to result from the ADAP-MPC approach is significantly lower than that resulting from the standard approach. The ADAP-MPC approach yields an error reduction of 80-83% in Fig 6.8d (from inspecting the portions

of these signals where the transient response characteristics have largely diminished, for example in $t \in [10.5, 11]$, $t \in [11.5, 12]$ and so on).

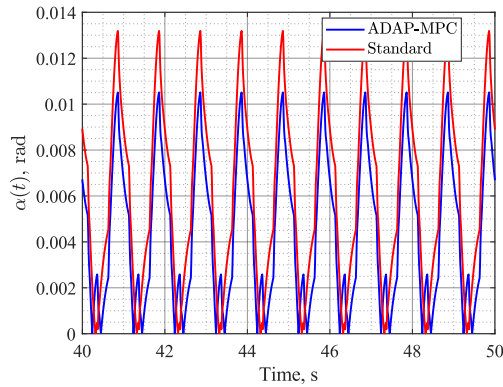
The error reduction found in Fig 6.8 is significantly reduced in the interval shown in Fig 6.9. This performance advantage of the ADAP-MPC approach is temporarily lost between $t = 48$ seconds and $t = 58$ seconds (as illustrated in Fig 6.10).



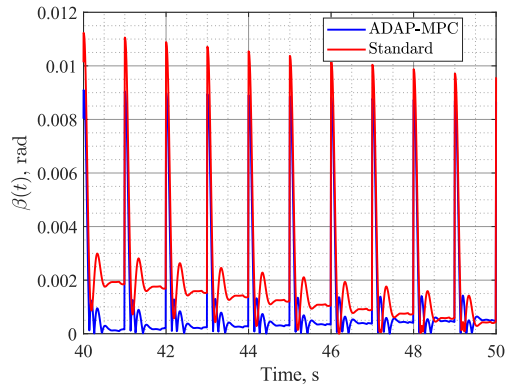
(a) Vertical actuation angle, $\alpha(t)$ radians.



(b) Horizontal actuation angle, $\beta(t)$ radians.



(c) Absolute error in the trajectory tracking for $\alpha(t)$.



(d) Absolute error in the trajectory tracking for $\beta(t)$.

Figure 6.9: Reference trajectories and absolute error measures for Test 2 across the interval $t \in [40, 50]$ seconds.

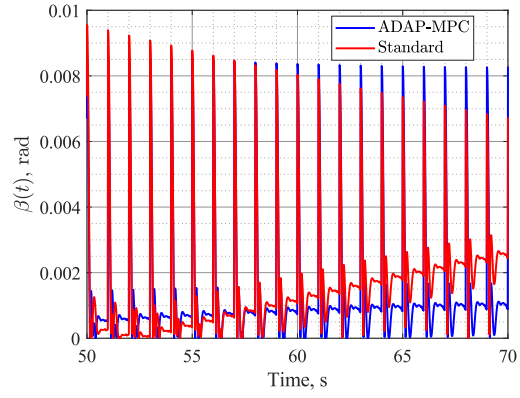


Figure 6.10: Absolute error in the trajectory tracking for $\beta(t)$ for Test 2 across the interval $t \in [50, 70]$ seconds.

Considering the last of the three intervals in the time domain for Test 2, Fig 6.11, the beneficial increased accuracy offered by the ADAP-MPC approach is regained and sustained.

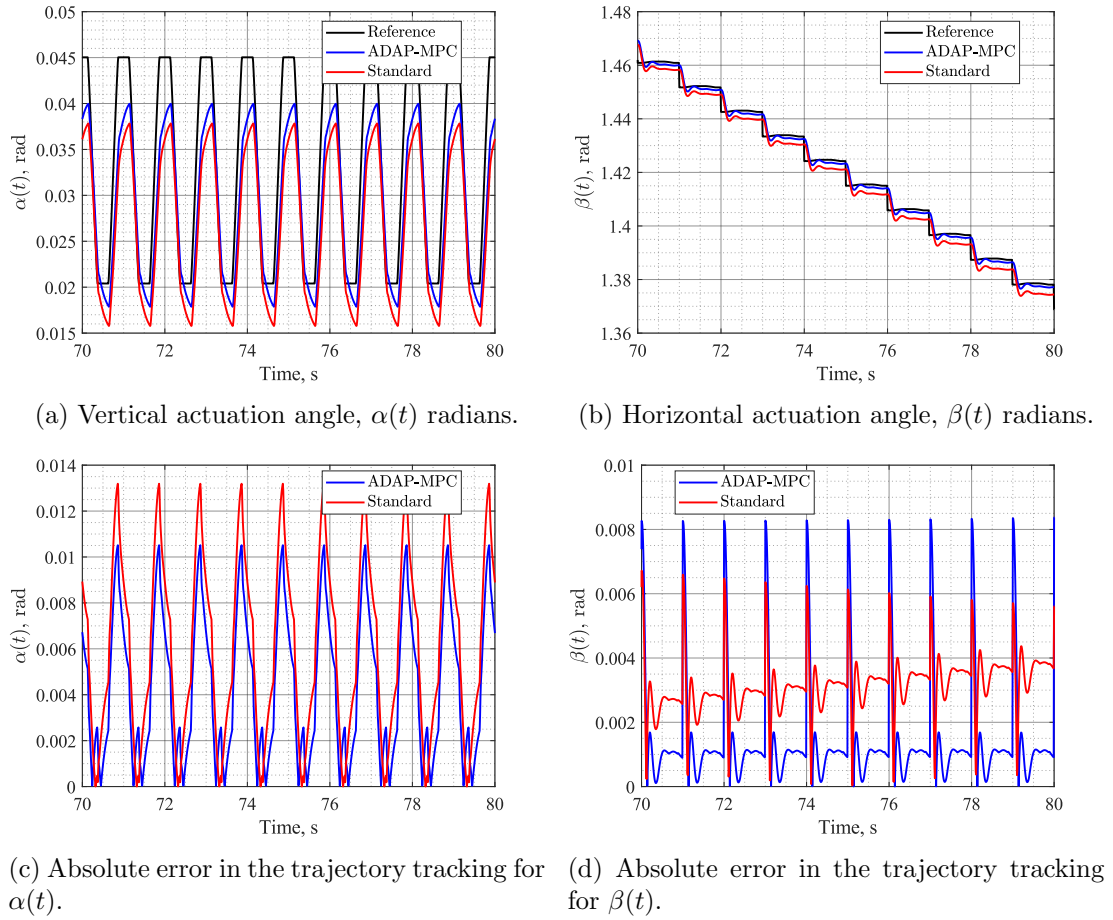
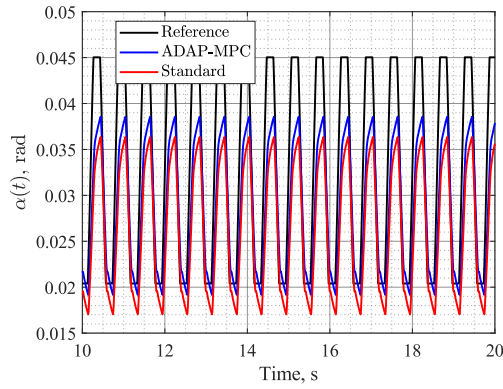


Figure 6.11: Reference trajectories and absolute error measures for Test 2 across the interval $t \in [70, 80]$ seconds.

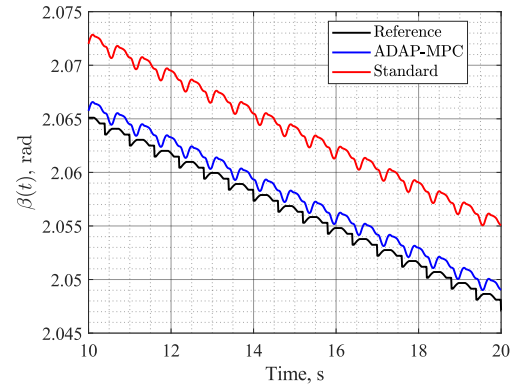
Time interval	Integral of squared error in α (rad^2s^2)		Integral of squared error in β (rad^2s^2)	
	ADAP-MPC	Standard	ADAP-MPC	Standard
$10 \leq t \leq 20$	1.51×10^{-4}	2.63×10^{-4}	1.055×10^{-3}	1.98×10^{-3}
$20 \leq t \leq 35$	2.27×10^{-4}	3.96×10^{-4}	1.35×10^{-3}	2.31×10^{-3}
$65 \leq t \leq 100$	3.80×10^{-4}	6.62×10^{-4}	7.46×10^{-4}	5.87×10^{-4}

Table 6.4: The integral of squared error for each coordinate for both control strategies across the time intervals shown for Test 2 in Figs 6.8-6.11.

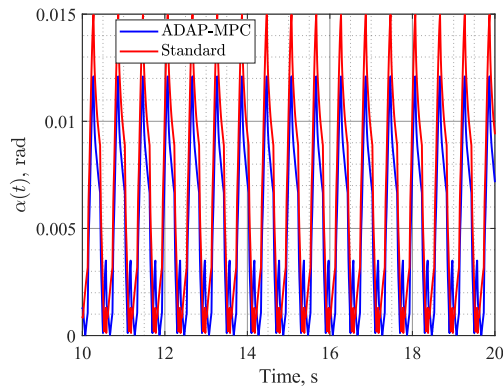
Similarly to Test 2, the results for Test 4 (shown over Figs 6.12-6.14) consider three intervals spanning the test's duration. As in the previous test, the accuracy in both controllers tracking the setpoint for α is consistent across each of the three intervals.



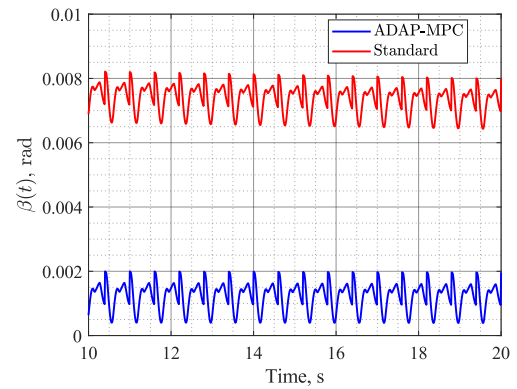
(a) Vertical actuation angle, $\alpha(t)$ radians.



(b) Horizontal actuation angle, $\beta(t)$ radians.



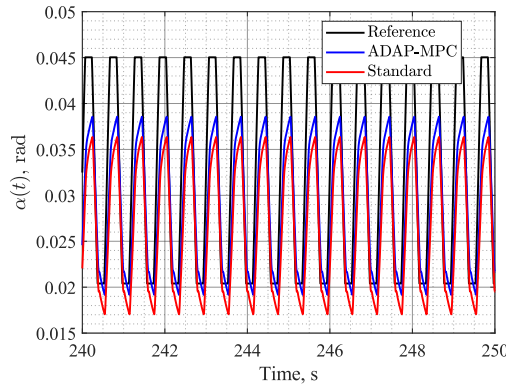
(c) Absolute error in the trajectory tracking for $\alpha(t)$.



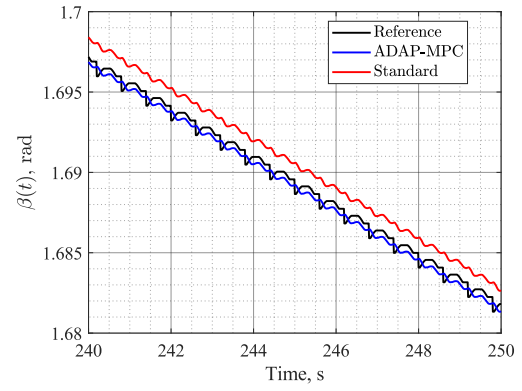
(d) Absolute error in the trajectory tracking for $\beta(t)$.

Figure 6.12: Reference trajectories and absolute error measures for Test 4 across the interval $t \in [10, 20]$ seconds.

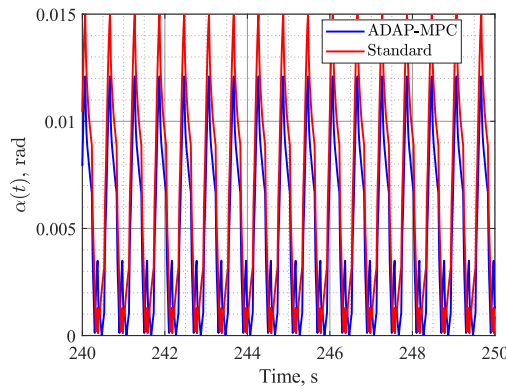
The most obvious difference regarding the accuracy of the Test 4 results relative to the Test 2 results is found in the tracking of the β reference trajectory. The benefit afforded by the ADAP-MPC method is dramatically increased by comparison with the observations made for the Test 2 results.



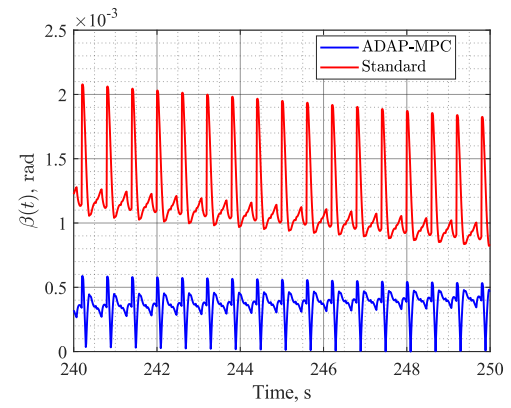
(a) Vertical actuation angle, $\alpha(t)$ radians.



(b) Horizontal actuation angle, $\beta(t)$ radians.



(c) Absolute error in the trajectory tracking for $\alpha(t)$.



(d) Absolute error in the trajectory tracking for $\beta(t)$.

Figure 6.13: Reference trajectories and absolute error measures for Test 4 across the interval $t \in [240, 250]$ seconds.

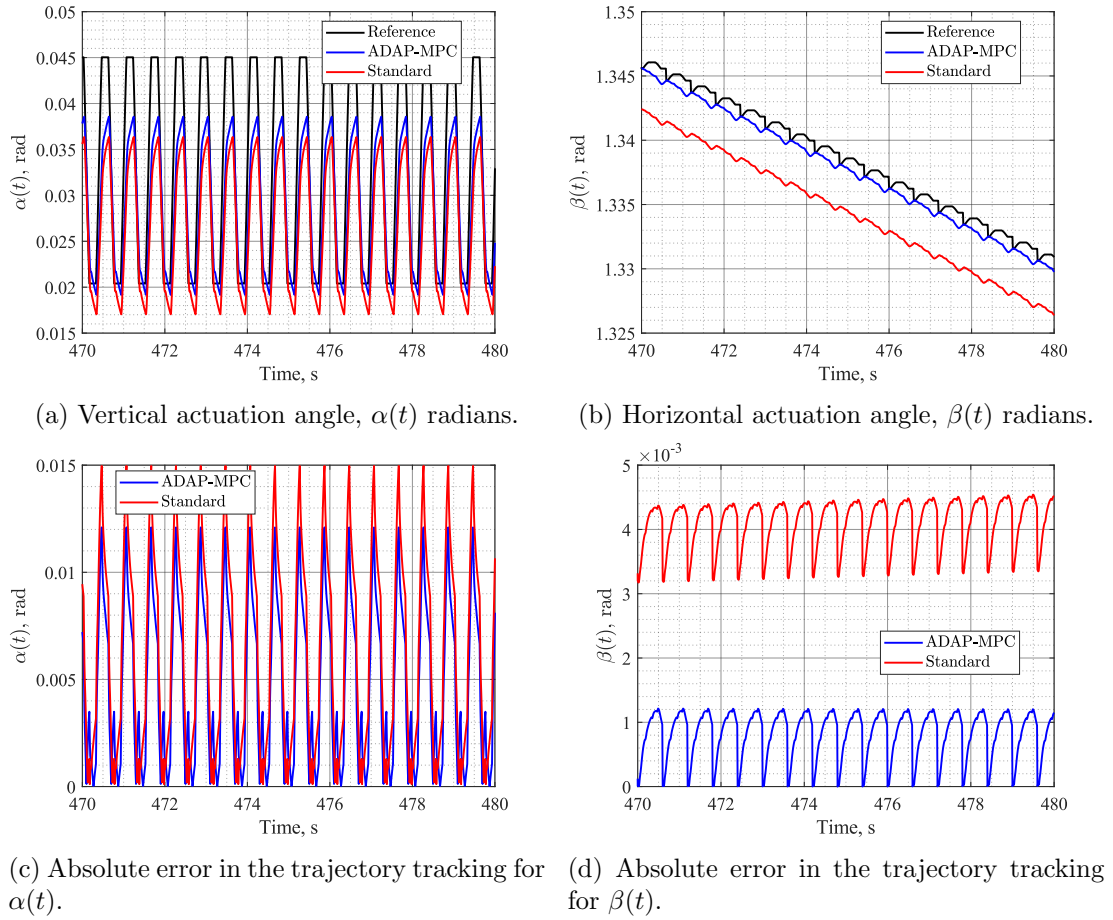


Figure 6.14: Reference trajectories and absolute error measures for Test 4 across the interval $t \in [470, 480]$ seconds.

The integral of the squared error is computed for each of the tests and is tabulated in Tables 6.3-6.5, and are also repeated alongside the results of Tests 3 and 5 in Tables 6.6 and 6.7.

Time interval	Integral of squared error in α (rad^2s^2)		Integral of squared error in β (rad^2s^2)	
	ADAP-MPC	Standard	ADAP-MPC	Standard
$10 \leq t \leq 20$	3.78×10^{-4}	5.81×10^{-4}	0.49×10^{-4}	2.56×10^{-4}
$240 \leq t \leq 250$	39.8×10^{-4}	71.9×10^{-4}	0.38×10^{-4}	5.44×10^{-4}
$470 \leq t \leq 480$	76.8×10^{-4}	139×10^{-4}	0.012×10^{-4}	49.0×10^{-4}

Table 6.5: The integral of squared error for each coordinate for both control strategies across the time intervals shown for Test 4 in Figs 6.12-6.14.

6.2.2 Control effort analysis

Similar to the graphs shown in the examples considered in Chapter 5, the areas enclosed between the exerted control torques and the associated displacement are shown as a measure of work done by the controller. While the work done by each of the control actions is not solely dependent on only one of the generalised coordinates, this measure remains a useful illustration. While there may be a variation in β which happens as a result the control action τ_v , this variation is minimal when compared with the variation in α . As a result, the work done by τ_v is illustrated by determining the area between τ_v and α ; meanwhile, the work done by τ_h is illustrated by determining the area between τ_h and β .

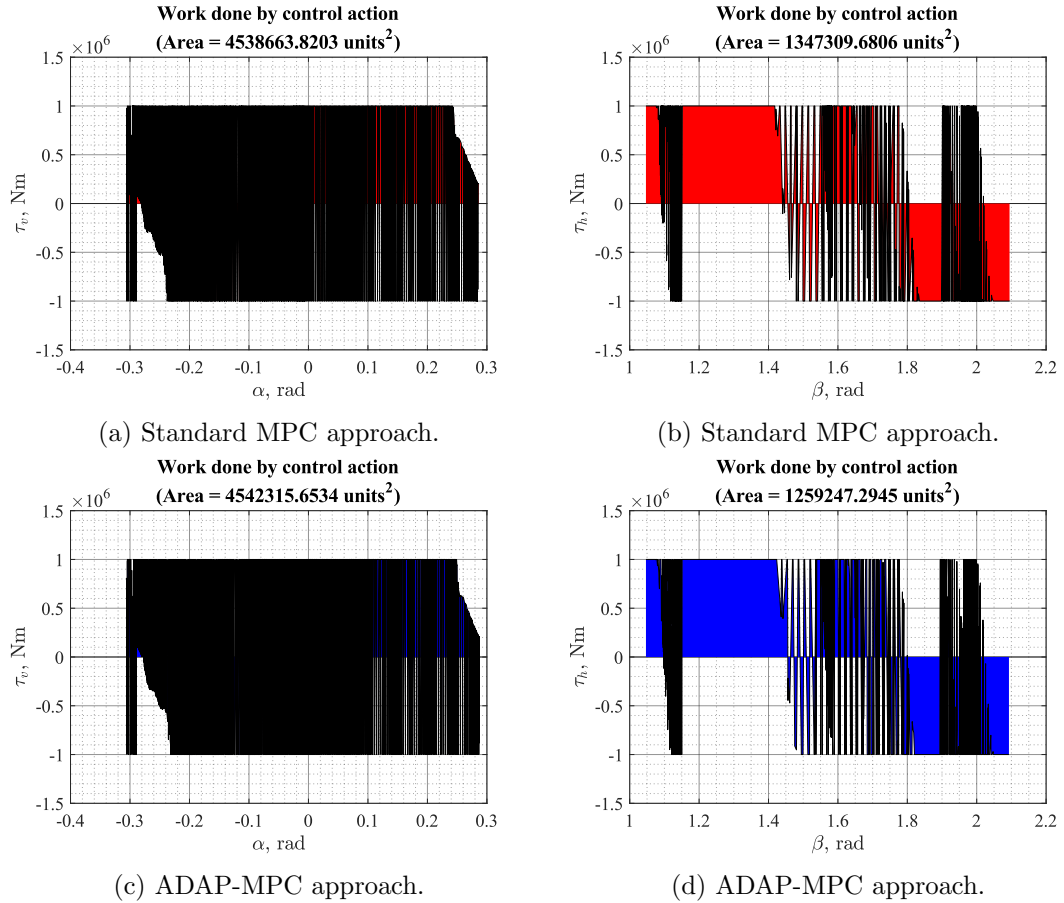


Figure 6.15: Illustration of control effort for both control strategies across the $t \in [0, 20]$ s interval of Test 1.

In this first set of results, Fig 6.15, the estimated control effort exerted by the ADAP-MPC approach is marginally greater for τ_v and marginally lower for τ_h with comparison to the standard MPC approach.

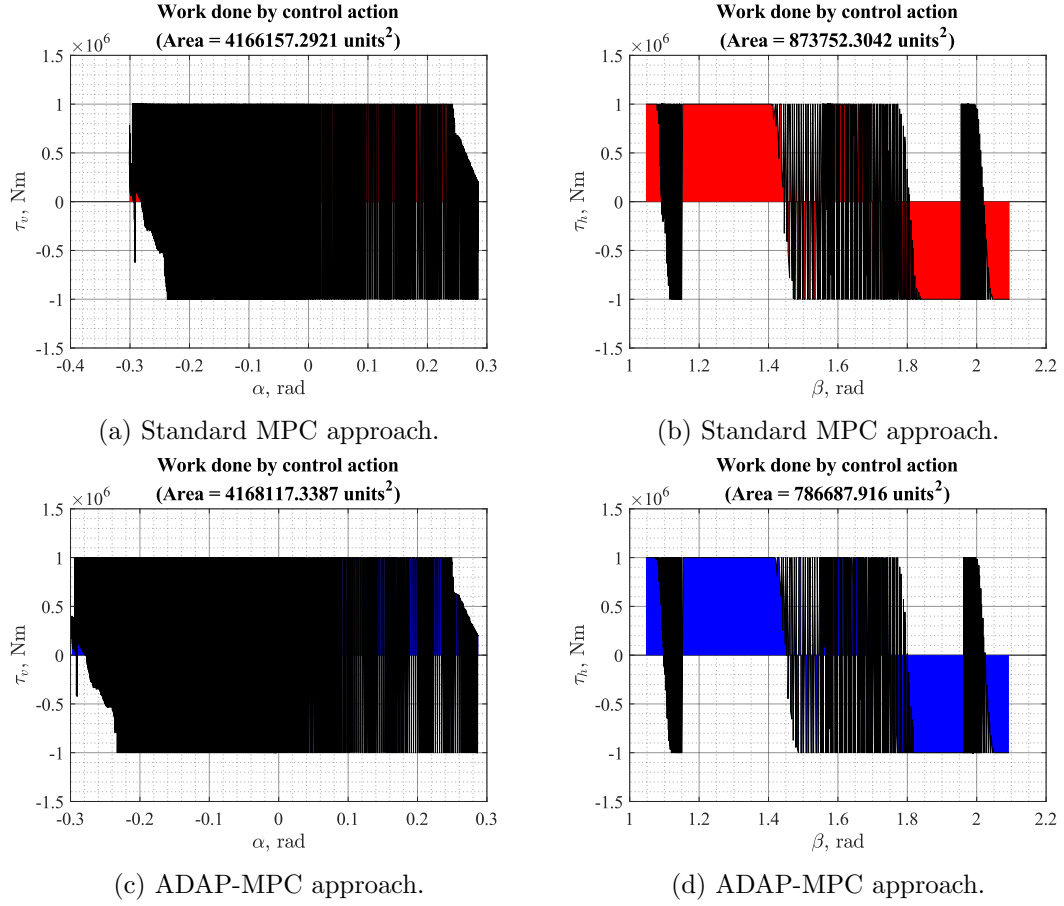


Figure 6.16: Illustration of control effort for both control strategies across the $t \in [20, 35]$ s interval of Test 1.

The same observation made for Fig 6.15 can also be made for 6.16.

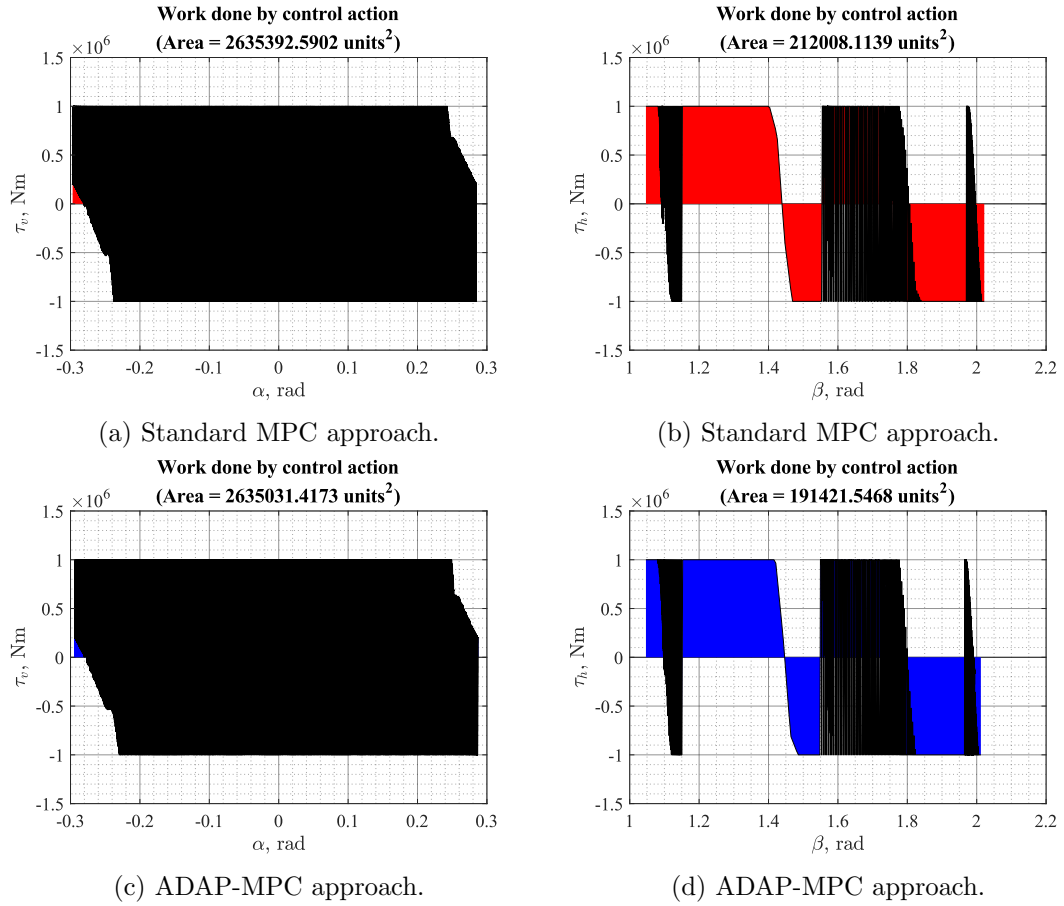


Figure 6.17: Illustration of control effort for both control strategies across the $t \in [65, 100]$ s interval of Test 1.

For Fig 6.17, the effort exerted through τ_v is considerably closer for the two control strategies while the effort exerted through τ_h remains significantly lower for the ADAP-MPC approach. This concludes the results of this metric for Test 1.

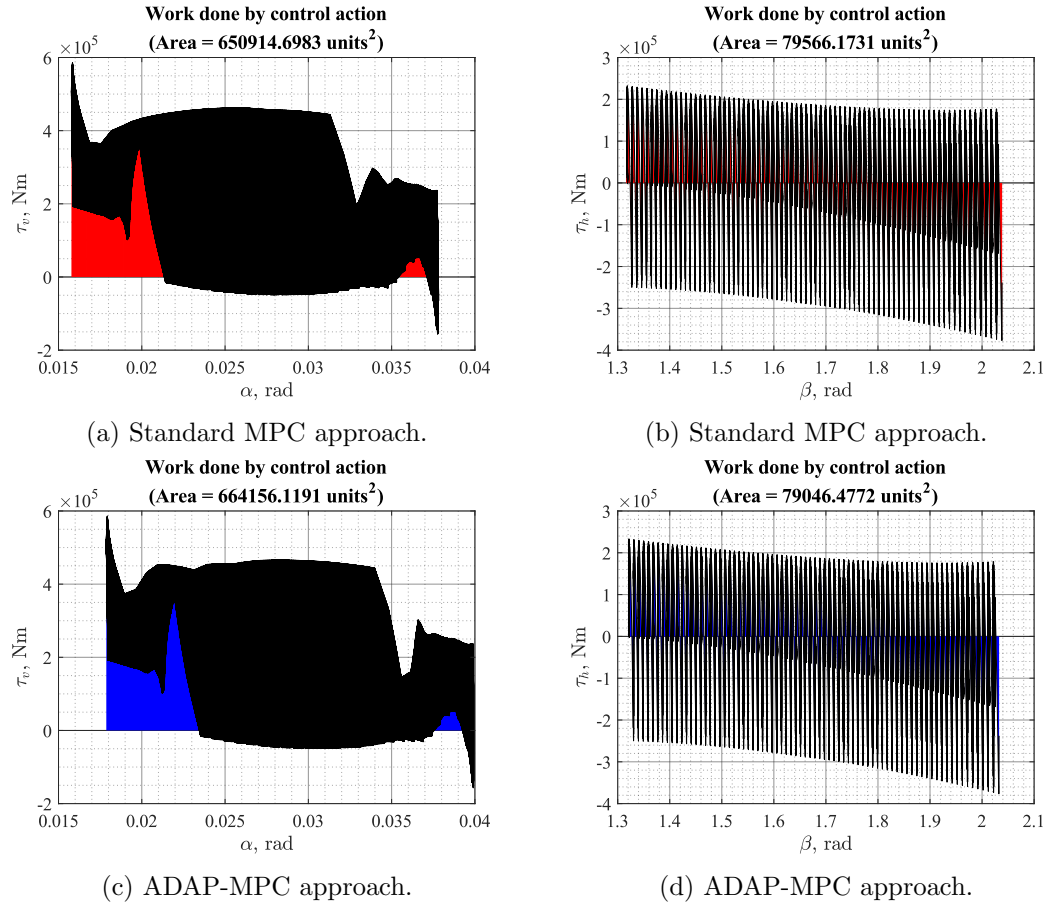


Figure 6.18: Illustration of control effort for both control strategies across the $t \in [10, 20]$ s interval of Test 2.

Similarly for Test 2 (Figs 6.18-6.20) as in the Test 1 (Figs 6.15-6.18) cases, the control effort exerted through τ_v is typically higher for the ADAP-MPC strategy. However, the effort via τ_h is consistently lower for the ADAP-MPC strategy.

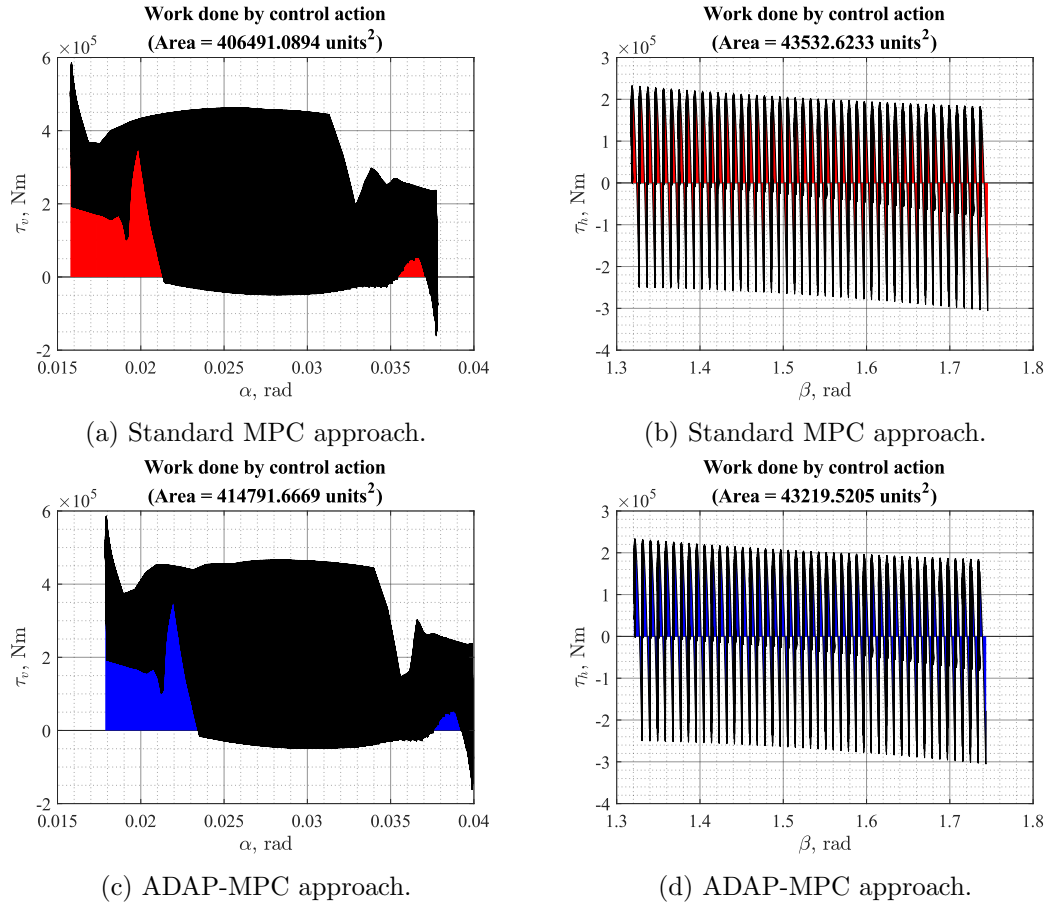


Figure 6.19: Illustration of control effort for both control strategies across the $t \in [40, 50]$ s interval of Test 2.

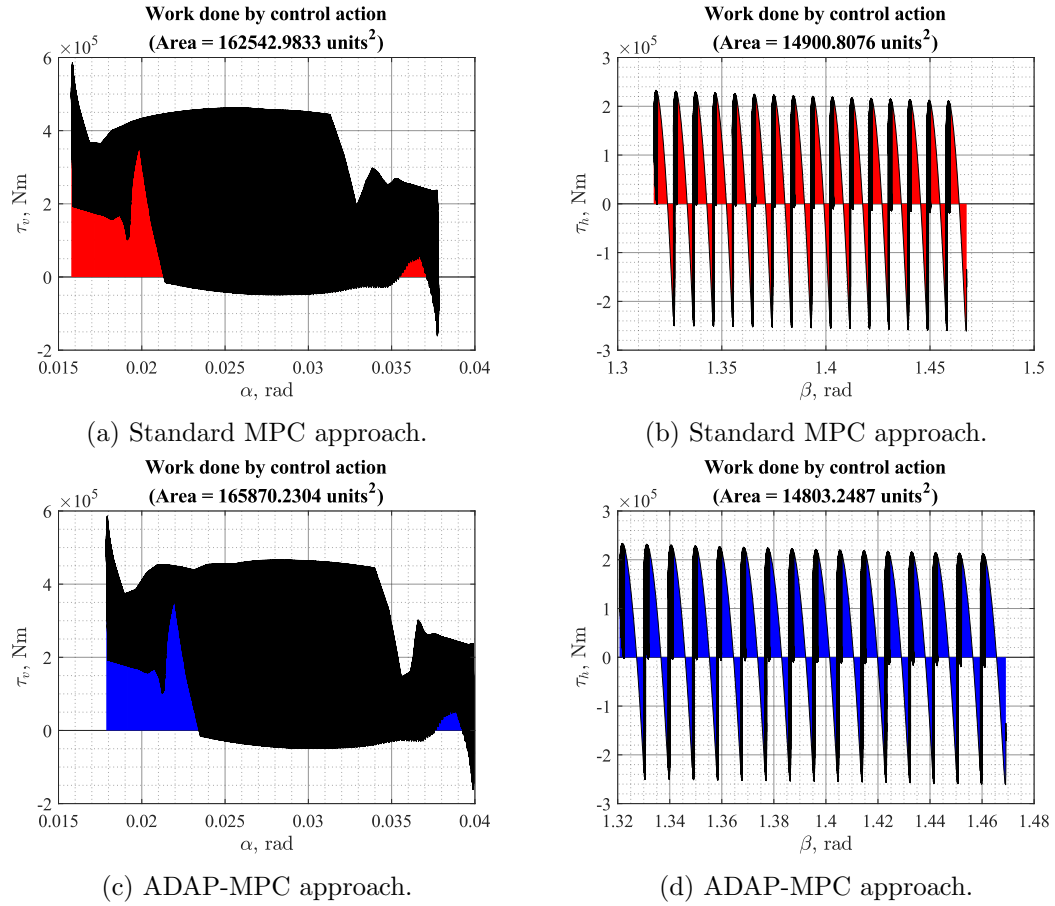
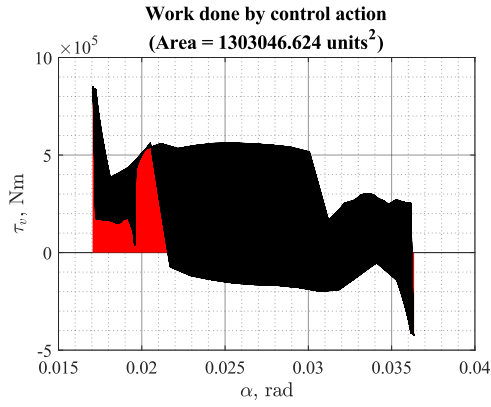
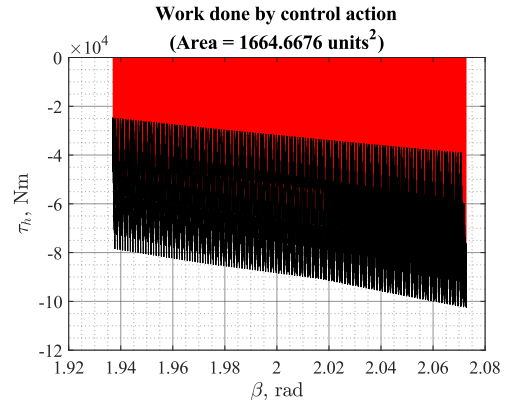


Figure 6.20: Illustration of control effort for both control strategies across the $t \in [70, 80]$ s interval of Test 2.

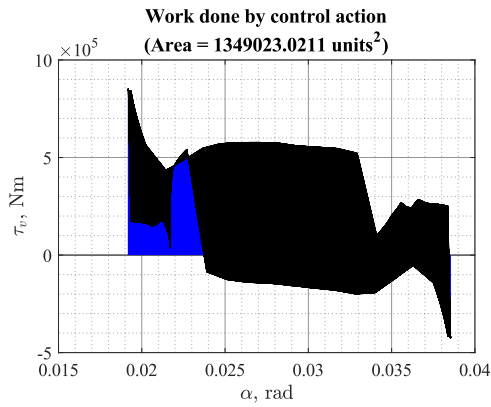
Previously made observations - that the work done by τ_v in the ADAP-MPC strategy is typically greater than or equivalent to that of the standard MPC strategy - are consistent for Test 4 (Figs 6.21-6.23).



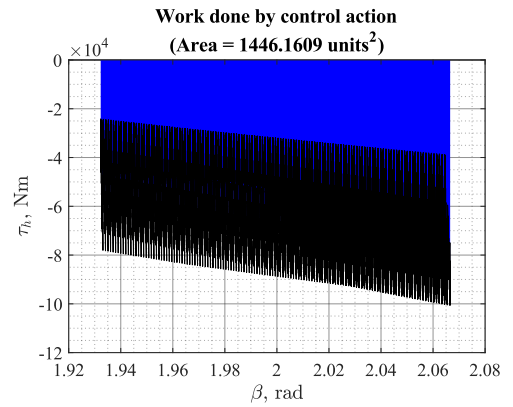
(a) Standard MPC approach.



(b) Standard MPC approach.



(c) ADAP-MPC approach.



(d) ADAP-MPC approach.

Figure 6.21: Illustration of control effort for both control strategies across the $t \in [10, 20]$ s interval of Test 4.

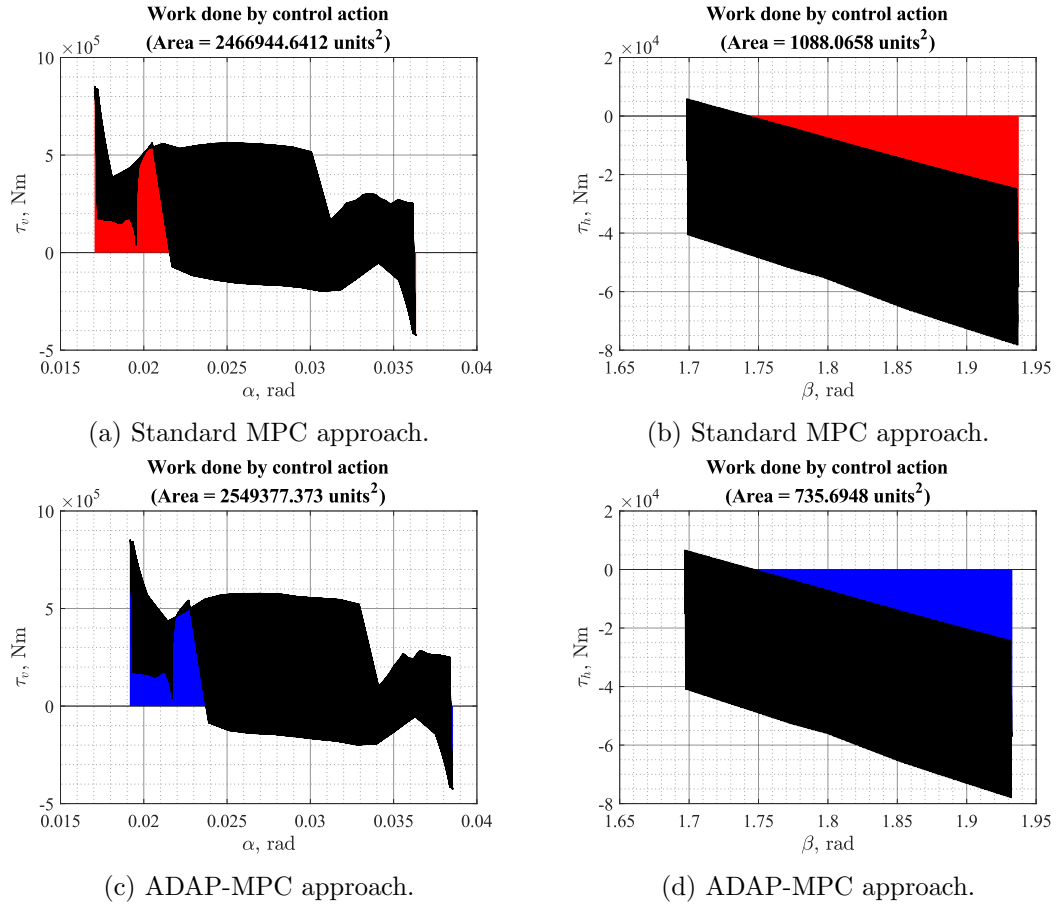


Figure 6.22: Illustration of control effort for both control strategies across the $t \in [240, 250]$ s interval of Test 4.

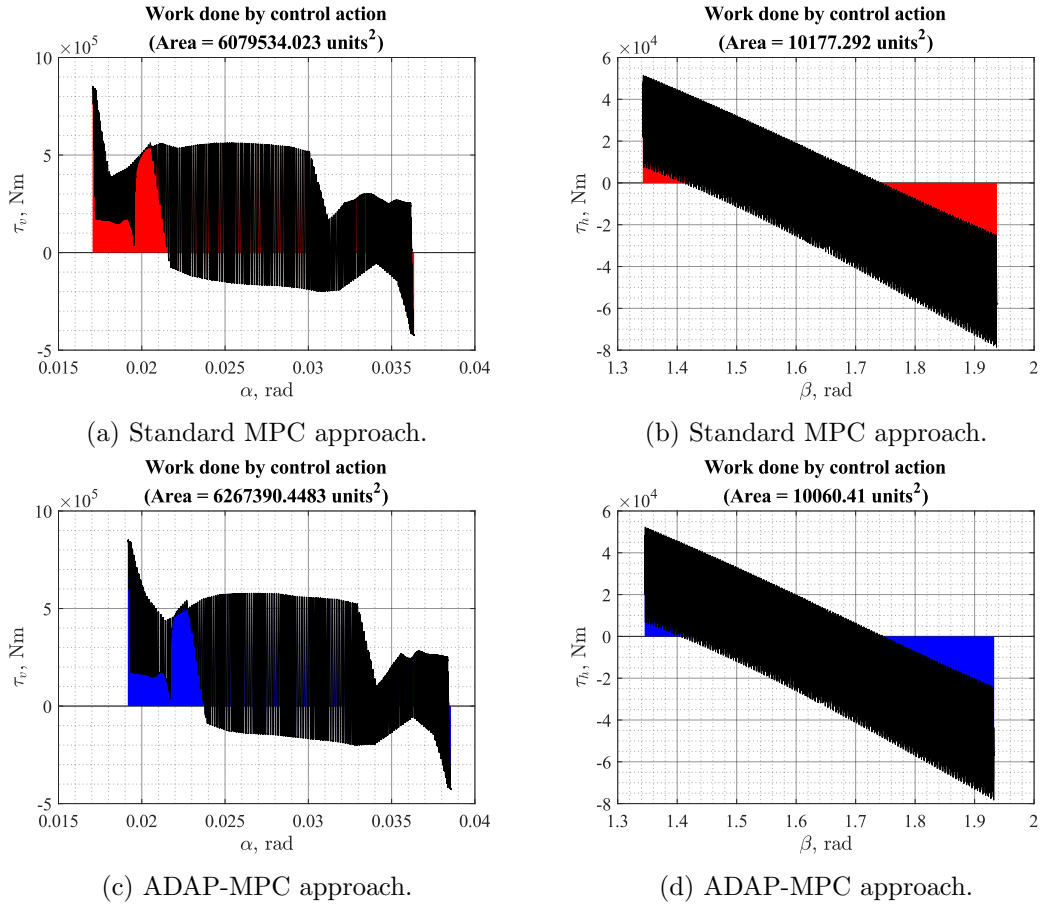


Figure 6.23: Illustration of control effort for both control strategies across the $t \in [470, 480]$ s interval of Test 4.

Fig 6.24 illustrates a summary of the results otherwise illustrated in this section, Figs 6.15-6.23. Specifically, this graph shows the percentage reduction in control effort exerted when using the ADAP-MPC strategy with comparison to the standard MPC strategy. For example, in Test 4 over the $t \in [10, 20]$ s range, the ADAP-MPC strategy requires approximately 3% **more** effort than the standard approach in the α - τ_v measure, but it requires approximately 13% **less** effort when considering the β - τ_h measure.

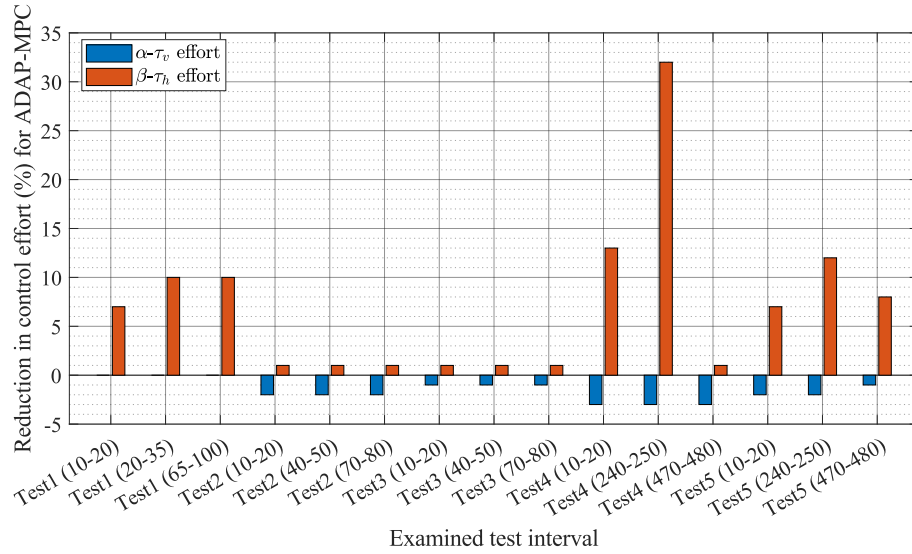
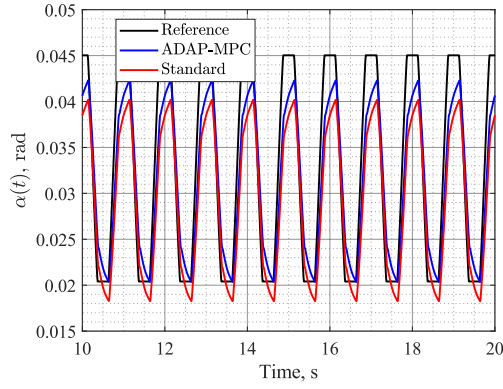


Figure 6.24: Percentage reduction in the control effort required by the ADAP-MPC method. Positive values indicate reduced effort; negative values indicate increased effort.

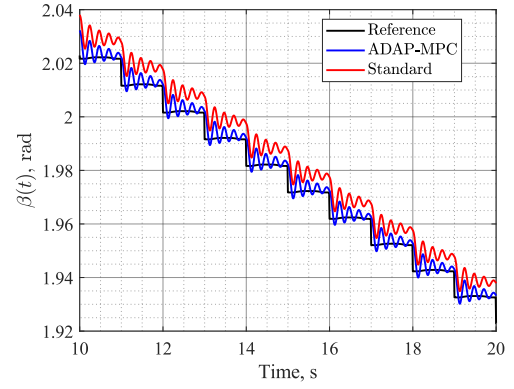
From Fig 6.24, it can be observed that the system controlled by the ADAP-MPC approach typically results in lower energy expenditure than the system controlled via the standard approach against which this thesis benchmarks the method.

6.2.3 Trajectory tracking results for manoeuvres including a payload

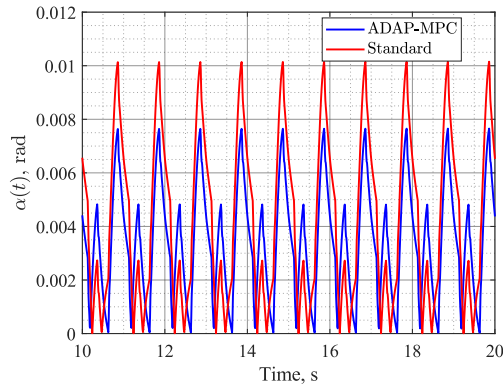
Using the same trajectories as Test 2, Test 3 includes a 1000-kg, 4-metre-long payload. The trajectory tracking results for Test 3, and their associated errors, are shown in Figs 6.25-6.27 alongside the trajectory tracking errors for the same interval in Test 2.



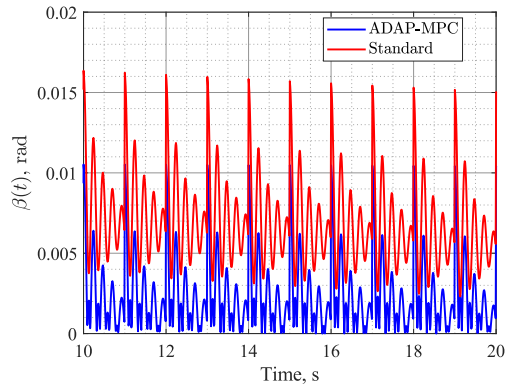
(a) Vertical actuation angle, $\alpha(t)$ radians.



(b) Horizontal actuation angle, $\beta(t)$ radians.

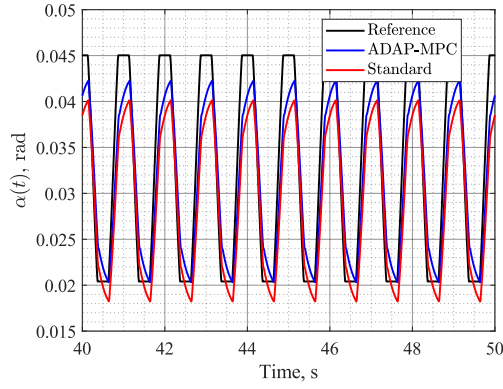


(c) Absolute error in the trajectory tracking for $\alpha(t)$.

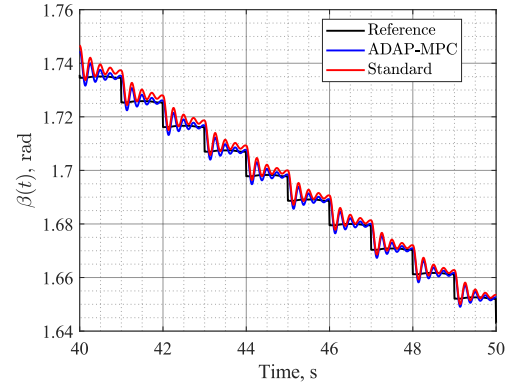


(d) Absolute error in the trajectory tracking for $\beta(t)$.

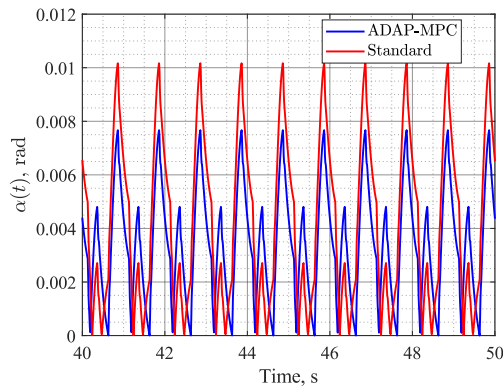
Figure 6.25: Reference trajectories and absolute error measures for Test 3 across the interval $t \in [10, 20]$ seconds.



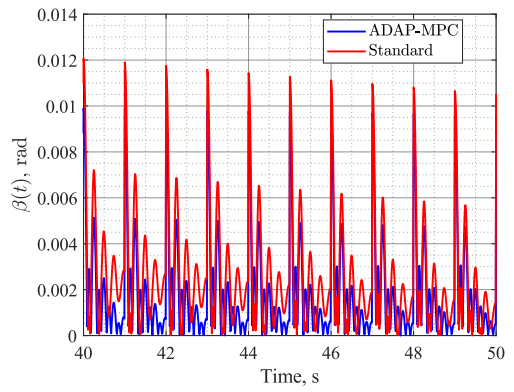
(a) Vertical actuation angle, $\alpha(t)$ radians.



(b) Horizontal actuation angle, $\beta(t)$ radians.



(c) Absolute error in the trajectory tracking for $\alpha(t)$.



(d) Absolute error in the trajectory tracking for $\beta(t)$.

Figure 6.26: Reference trajectories and absolute error measures for Test 3 across the interval $t \in [40, 50]$ seconds.

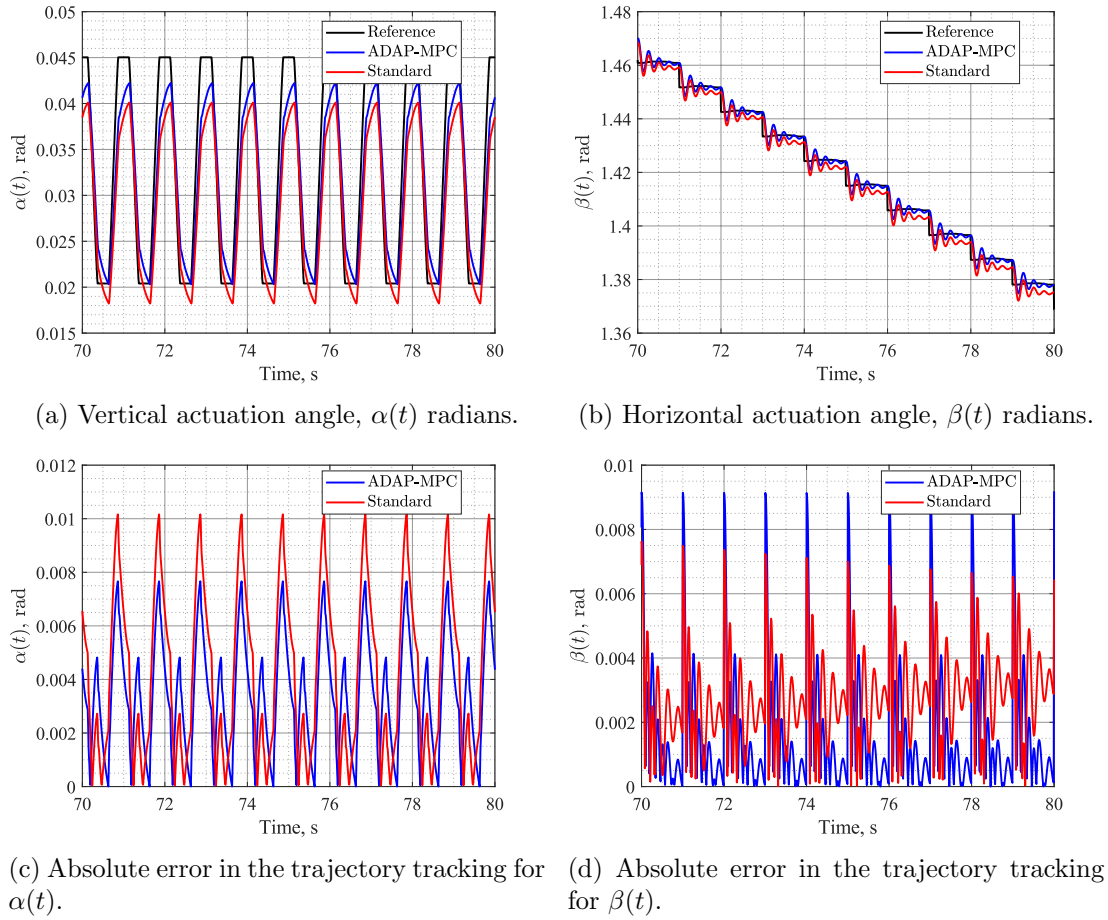


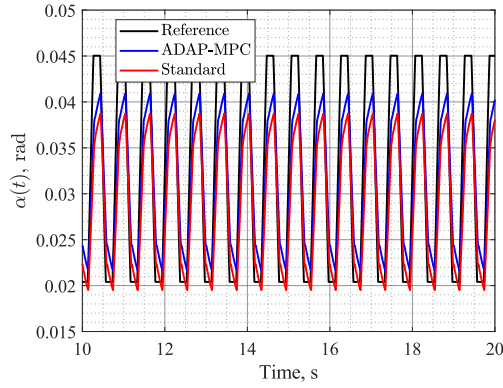
Figure 6.27: Reference trajectories and absolute error measures for Test 3 across the interval $t \in [70, 80]$ seconds.

Time interval	Integral of squared error in α (rad^2s^2)		Integral of squared error in β (rad^2s^2)	
	ADAP-MPC	Standard	ADAP-MPC	Standard
$10 \leq t \leq 20$	0.32×10^{-4}	1.53×10^{-4}	9.91×10^{-4}	19.0×10^{-4}
$40 \leq t \leq 50$	0.32×10^{-4}	1.52×10^{-4}	8.71×10^{-4}	11.5×10^{-4}
$70 \leq t \leq 80$	0.31×10^{-4}	1.53×10^{-4}	7.72×10^{-4}	3.99×10^{-4}

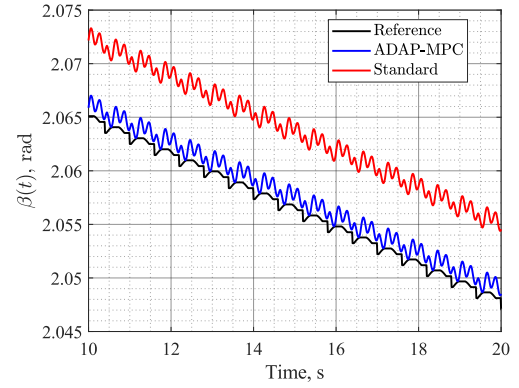
Table 6.6: The integral of squared error for each coordinate for both control strategies across the time intervals shown for Test 3 in Figs 6.25-6.27.

Lastly, the trajectories and absolute errors for Test 5 are shown in Figs 6.28-6.30 alongside illustrations of the errors over the same intervals in Test 4. Similarly to Tests 2 and 3, the trajectories for Tests 4 and 5 are indistinguishable given the constant mass

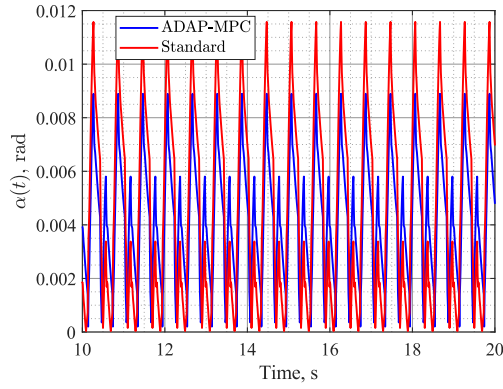
payload included in Test 5.



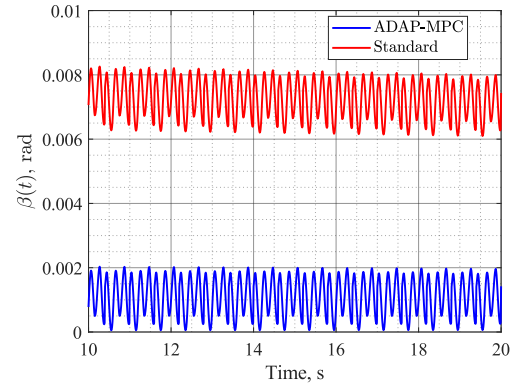
(a) Vertical actuation angle, $\alpha(t)$ radians.



(b) Horizontal actuation angle, $\beta(t)$ radians.

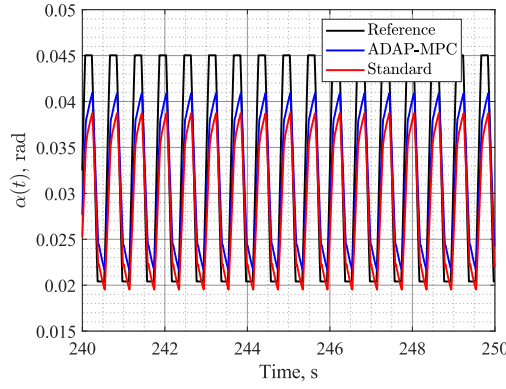


(c) Absolute error in the trajectory tracking for $\alpha(t)$.

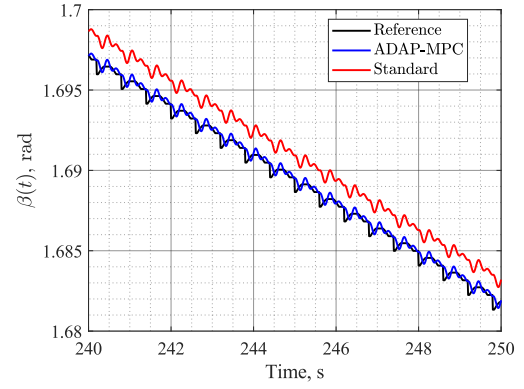


(d) Absolute error in the trajectory tracking for $\beta(t)$.

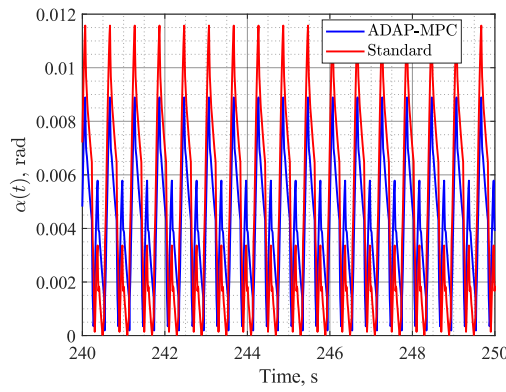
Figure 6.28: Reference trajectories and absolute error measures for Test 5 across the interval $t \in [10, 20]$ seconds.



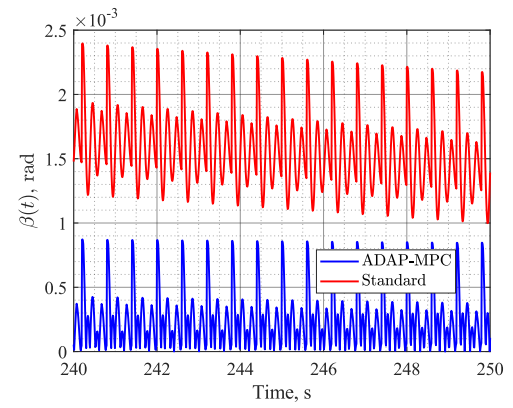
(a) Vertical actuation angle, $\alpha(t)$ radians.



(b) Horizontal actuation angle, $\beta(t)$ radians.



(c) Absolute error in the trajectory tracking for $\alpha(t)$.



(d) Absolute error in the trajectory tracking for $\beta(t)$.

Figure 6.29: Reference trajectories and absolute error measures for Test 5 across the interval $t \in [240, 250]$ seconds.

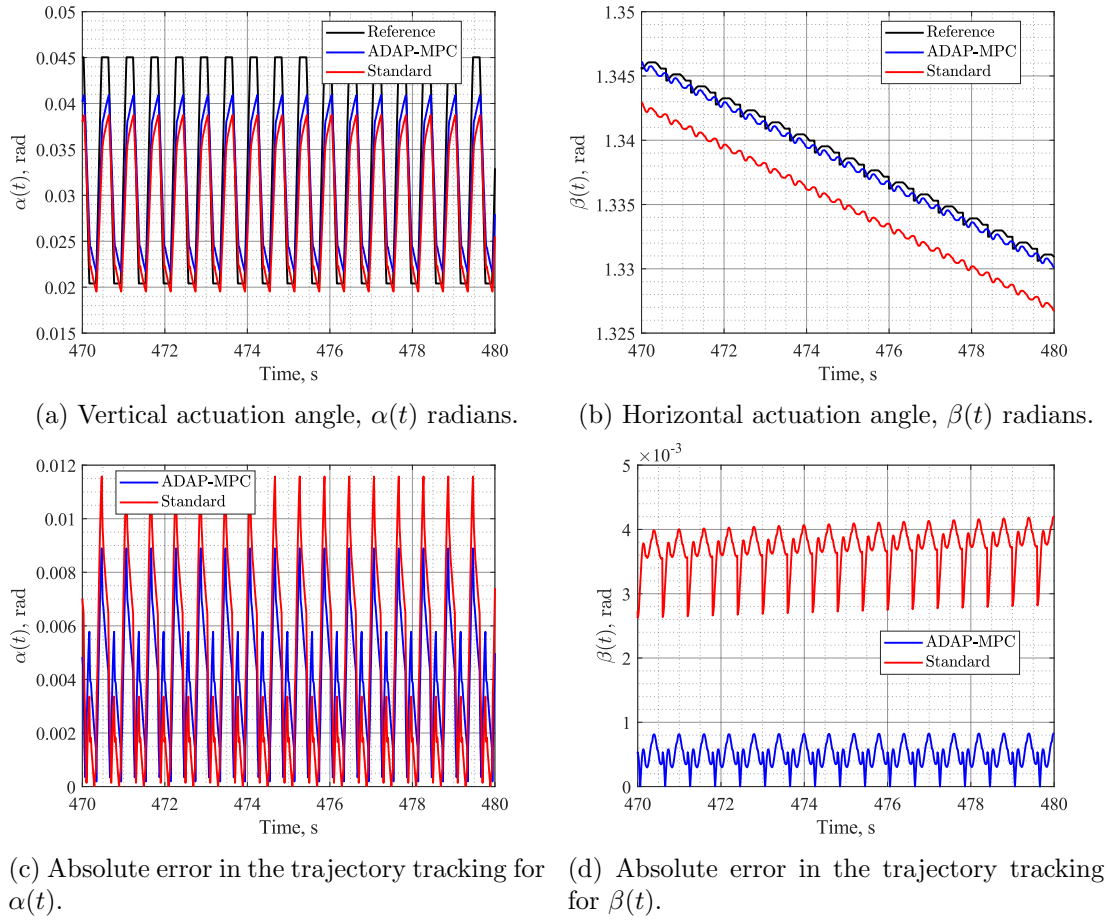


Figure 6.30: Reference trajectories and absolute error measures for Test 5 across the interval $t \in [470, 480]$ seconds.

Time interval	Integral of squared error in α (rad^2s^2)		Integral of squared error in β (rad^2s^2)	
	ADAP-MPC	Standard	ADAP-MPC	Standard
$10 \leq t \leq 20$	2.79×10^{-5}	8.77×10^{-5}	0.25×10^{-5}	0.11×10^{-5}
$240 \leq t \leq 250$	2.57×10^{-5}	6.56×10^{-5}	0.10×10^{-5}	0.61×10^{-5}
$470 \leq t \leq 480$	2.087×10^{-5}	17.7×10^{-5}	0.21×10^{-5}	1.58×10^{-5}

Table 6.7: The integral of squared error for each coordinate for both control strategies across the time intervals shown for Test 5 in Figs 6.28-6.30.

6.2.4 Computational expense estimation

The number of function evaluations required over each prediction horizon was recorded, and the mean, minimum, and maximum values recorded for each solver as shown in Table 6.8. To clarify this with an example, note the results for *ode45*. For this case, using the standard prediction model, the function was evaluated an average of 62 times across each of the prediction horizons evaluated. In the same scenario, the ADAP prediction model required 61 evaluations of the function on average. The reference result for this solver shows that the fully coupled, nonlinear prediction model was called a total of 1903 times when simply numerically solved across the entire simulation's time domain.

To offer an alternative explanation of the averaging process which is conducted here: recall the general operation of the receding horizon in MPC. If a step-size of 0.1 seconds were used to form a prediction of the system's response over an interval of 10 seconds, 100 prediction horizons would then have been required. If each prediction horizon spans 1 second, then the first horizon would span $t \in [0, 1]$, the second would span $t \in [0.1, 1.1]$, the third spanning $t \in [0.2, 1.2]$, and so on. From Table 6.8, it can be seen that the lowest number of function evaluations required over any of these 100 horizons is 61 (for both prediction models).

Note that this explanation is simplified by only considering prediction horizons with constant step-sizes. However, given that these numerical solvers use variable step-sizes to obtain solutions within prescribed error tolerances, this is not the case for these simulations. As a result, accounting for the accuracy of each prediction (to the same relative error tolerance of 1×10^{-5}) is inherent to this process. The results shown in Table 6.8 is

Solver	Number of evaluations						Ref
	Standard			ADAP			
	Min	Mean	Max	Min	Mean	Max	
ode45	61	62	67	61	61	67	1903
ode23	31	34	43	34	34	40	4054
ode113	27	35	49	31	34	49	1011
ode78	170	170	170	170	170	170	9901
ode89	210	210	210	210	210	210	19788
ode15s	32	43	48	34	41	44	744
ode23s	83	92	139	83	85	138	13349
ode23t	28	32	41	26	29	39	2517
ode23tb	33	39	54	33	34	50	5292

Table 6.8: Min, max, and mean number of function evaluations required for each solver choice in MATLAB. Note that the mean values are rounded to the nearest integer value. Table first produced in [44].

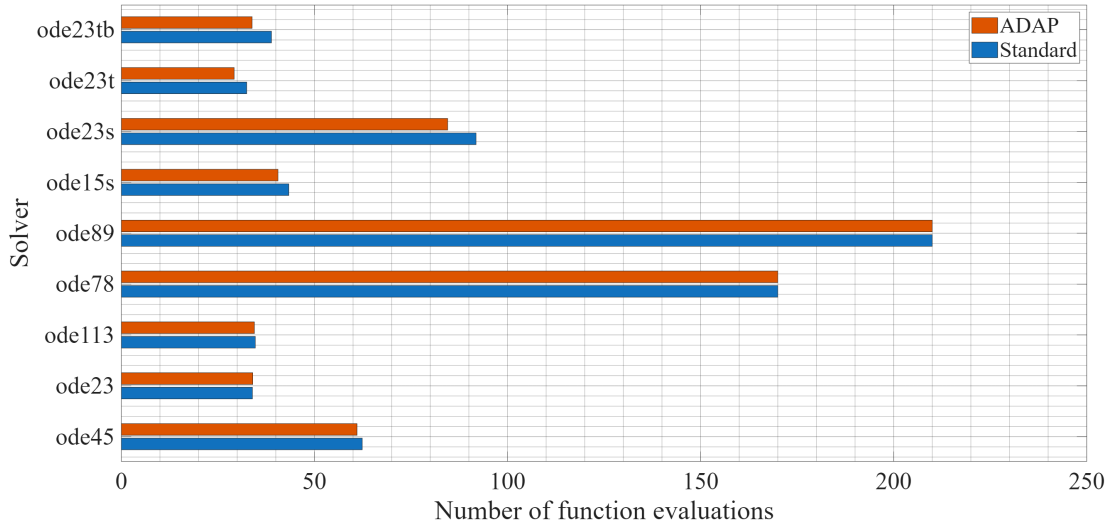
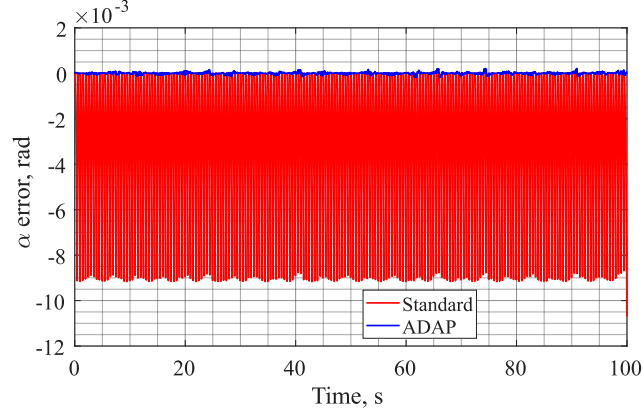


Figure 6.31: Mean number of function evaluations required for each prediction horizon for both linearised prediction models. Reproduced and revised from [44].

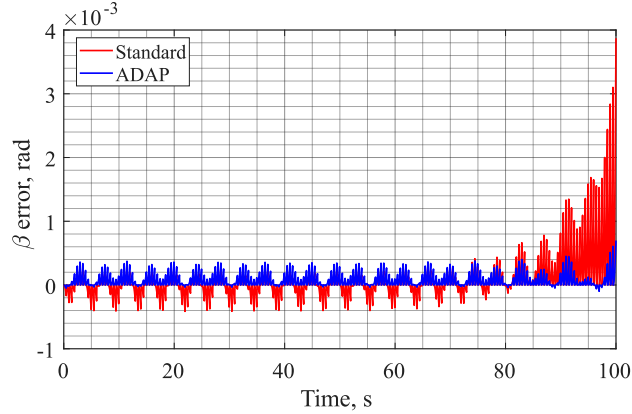
Fig 6.31 illustrates the comparison in the computational efficiency of the ADAP-MPC method. Considering this in conjunction with the improved accuracy in trajectory tracking which was previously observed, it can be stated that the ADAP-MPC algorithm achieves its improved accuracy with no increase in computational expense.

Since the error associated with these prediction models is implicitly considered in

examining the computational burden, Figs 6.32-6.36 are used to illustrate the errors yielded by the various solvers in this process. Note that these figures are each reproduced from the author's work in [44].

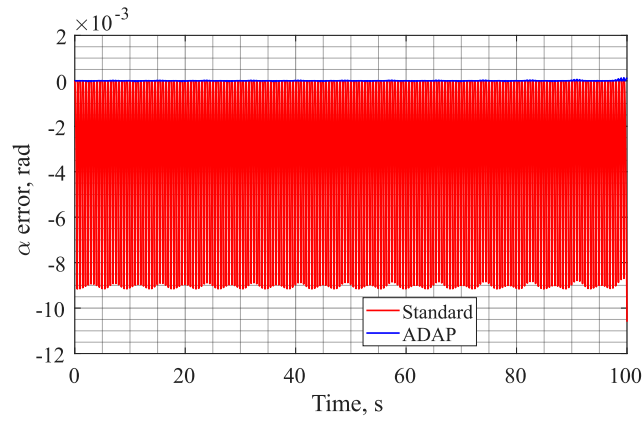


(a) Error in the prediction of $\alpha(t)$.

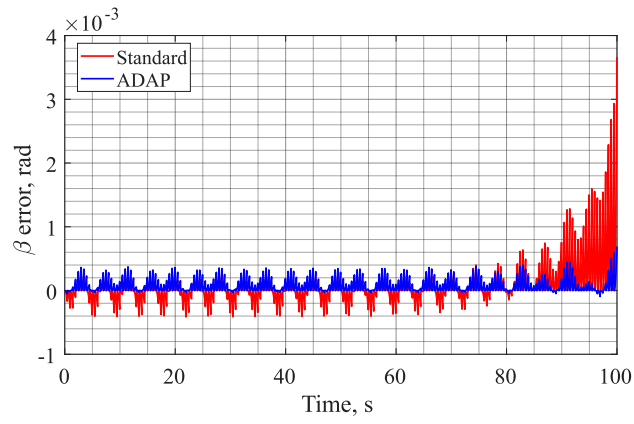


(b) Error in the prediction of $\beta(t)$.

Figure 6.32: Error between the standard and ADAP prediction models found via *ode45* when compared with the reference (target) solution (using the same solver). Altered since first presented in [44].

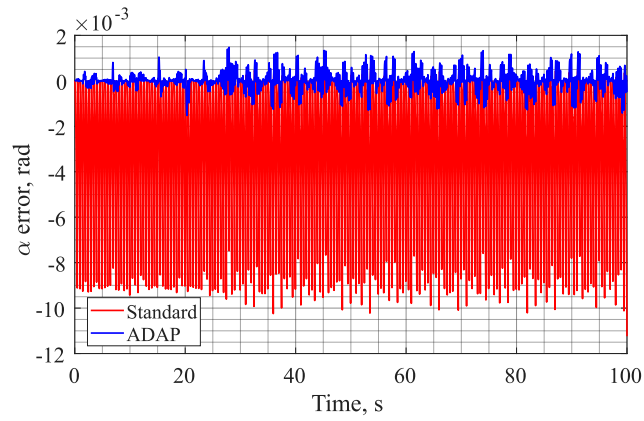


(a) Error in the prediction of $\alpha(t)$.

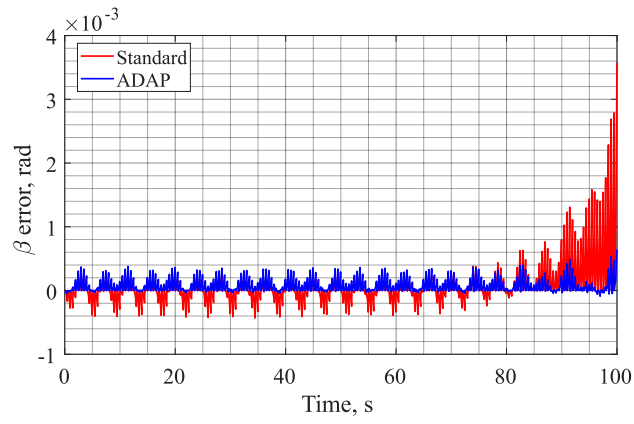


(b) Error in the prediction of $\beta(t)$.

Figure 6.33: Error between the standard and ADAP prediction models found via *ode89* when compared with the reference (target) solution (using the same solver). Altered since first presented in [44].

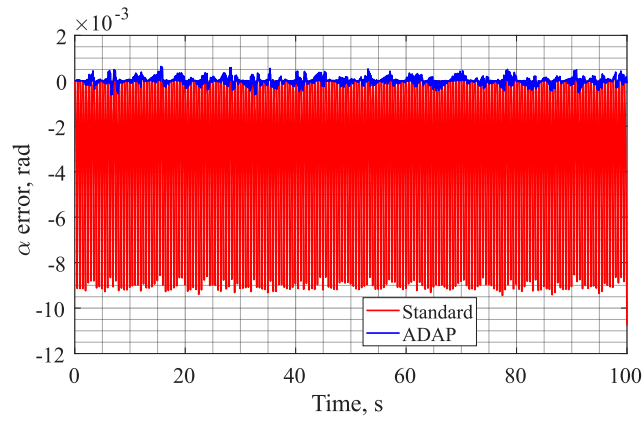


(a) Error in the prediction of $\alpha(t)$.

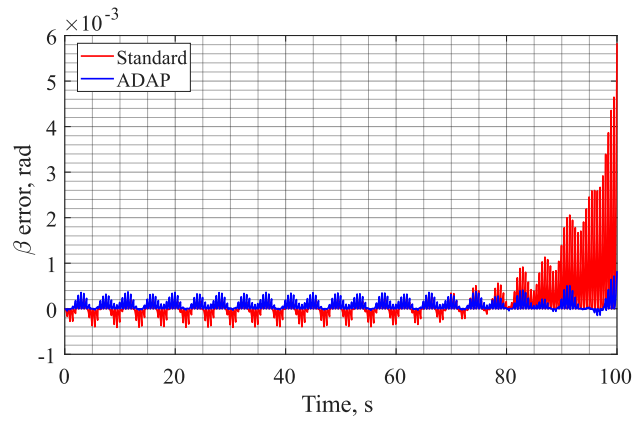


(b) Error in the prediction of $\beta(t)$.

Figure 6.34: Error between the standard and ADAP prediction models found via *ode113* when compared with the reference (target) solution (using the same solver). Altered since first presented in [44].

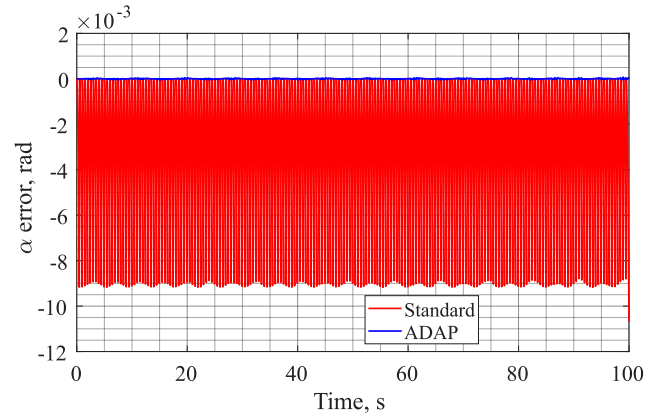


(a) Error in the prediction of $\alpha(t)$.

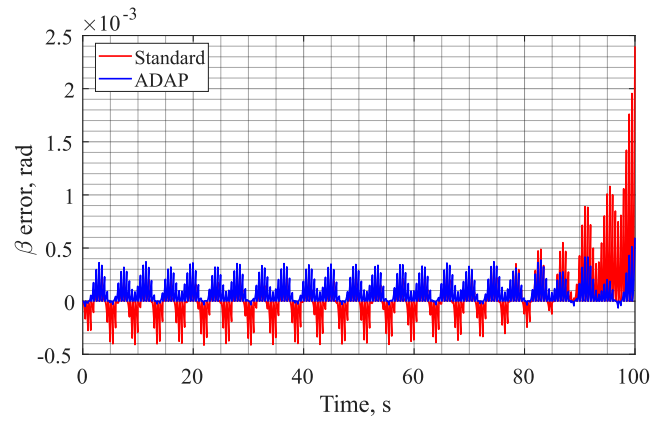


(b) Error in the prediction of $\beta(t)$.

Figure 6.35: Error between the standard and ADAP prediction models found via *ode15s* when compared with the reference (target) solution (using the same solver). Altered since first presented in [44].



(a) Error in the prediction of $\alpha(t)$.



(b) Error in the prediction of $\beta(t)$.

Figure 6.36: Error between the standard and ADAP prediction models found via *ode23t* when compared with the reference (target) solution (using the same solver). Altered since first presented in [44].

6.2.5 Discussion of ADAP-MPC controller performance for the *FutureForge* manipulator case study

The results from Tests 1 to 5 provide a range of useful insights into the effectiveness of the ADAP-MPC method by comparison to the standard MPC approach against which it is compared.

Principally, it should be noted that the accuracy of the ADAP-MPC controller was consistently equivalent or superior to the standard approach. Considering the subfigures (c) and (d) in Figs 6.5-6.7, and considering the steady-state portions of the various

step responses in Test 1, it can be seen that the absolute error in trajectory tracking yielded by the ADAP-MPC approach is consistently less than or equal to that of the standard approach. This is most dramatically illustrated in Figs 6.5c and 6.6c. Fig 6.7 illustrates the case of the two control approaches giving much closer trajectory tracking responses. Notably, although there is a clear visible difference between the two signals in Fig 6.5d, the benefit offered by the ADAP-MPC approach is less pronounced for β trajectory tracking. For transparency, these observations are made by considering the steady-state responses of both results; however, the transient error is very slightly higher for the ADAP-MPC approach in Figs 6.5c and 6.5d. The initial deviation in the measured output tracking of the setpoints is likely a result of the method used to feed the gravitational restoring forces into the MPC controller via estimated projections over the prediction horizon accompanied by small errors in the initial estimates of these values. Essentially, because the initial estimate of the restoring forces contains some inaccuracy, resulting in the measured disturbance underestimating the gravitational restoring actions, the manipulator initially falls away from its target position before a more accurate prediction is made.

The increased accuracy offered by the ADAP-MPC approach is also notable when considering Tests 2, 3, 4, and 5. Significantly, the clearest illustrations of this occur for trajectory tracking in β for Tests 2 and 4. This suggests that the more significantly limited benefit observed for β in Test 1 is not a common feature across all manoeuvres. The only instance across these manoeuvres where the error produced by the ADAP-MPC method is greater than the standard approach is specifically highlighted across Figs 6.9d and 6.10. Considering these manoeuvres through β , it should be noted that the benefit across all four tests is least significant when β is in the vicinity of the midpoint of its range. Although the error observed for the ADAP-MPC approach is less than that of the standard approach, the control parameters common to both approaches should ideally be amended to reduce the overshoot observed in α .

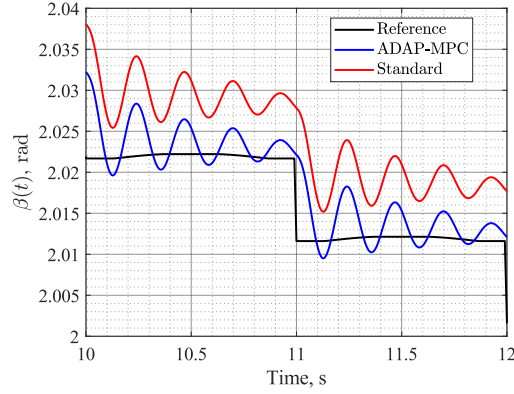


Figure 6.37: Trajectories obtained for β in Test 3 across the interval $t \in [10, 12]$ seconds, showing low damping properties in the transient responses.

When a payload is mounted to the end-effector, as in Tests 3 and 5, the transient response characteristics are less desirable than in Tests 2 and 4, suggesting that further tuning of the controller is required. The logarithmic decrement for both oscillating signals in Fig 6.37 is calculated to be approximately 0.48. This is estimated by observing two peaks, such as (10.24, 2.02837) and (10.47, 2.02646) on the ADAP-MPC signal which (at this interval) oscillates about an estimated equilibrium of 2.02335, and computing the logarithmic decrement via the following expression.

$$\delta = \ln \left(\frac{2.02837 - 2.02335}{2.02646 - 2.02335} \right) = 0.48 \quad (6.6)$$

This is estimated to be the same value for both the standard MPC and ADAP-MPC approach, indicating that the transient characteristics are due to parameter values which are common to both strategies. Note that the logarithmic decrement is chosen as a measure of the energy dissipation in the oscillatory response instead of the damping ratio, since estimating the latter would require further estimations (of frequency and period) and therefore introduces further sources of error. Nonetheless, the steady-state errors and integral of squared errors are consistently lower for the ADAP-MPC case with the exception of the $70 \leq t \leq 80$ interval in β for Test 3 (Table 6.6) and the $10 \leq t \leq 20$ interval in β for Test 5 (Table 6.7).

The work done by each control strategy to achieve its respective accuracy is also noteworthy. For Test 1 manoeuvres, the control effort required from both strategies appears to be approximately equal for variations in α , but the ADAP-MPC approach appears to offer a significant (5-10% in Fig 6.24) reduction in effort for variations in β over the same time intervals. For Test 2 and 3 manoeuvres, the control effort appears to be approximately equal for both control strategies, with reductions in effort offered by the ADAP-MPC approach for β being balanced by increased effort required for the manoeuvres in α over the same time interval. The net benefit in this metric, noted in Test 1 manoeuvres, is also noted in Tests 4 and 5, where the reduction in control effort for variations in β is substantial.

Lastly, the computational expense of the ADAP-MPC strategy was compared to the standard MPC strategy. Across the variety of MATLAB solvers shown in Table 6.8 and Fig 6.31, the computational burden of the ADAP-MPC approach is observed to be consistently less or equal to that of the standard MPC approach. Taken in the context of the subsequent error measures yielded by the same solvers, the ADAP-MPC approach is shown to be built on an effective and efficient prediction model.

6.3 Conclusion

This chapter presented a case study to evaluate the performance of the proposed novel control strategy which incorporates the ADAP methodology, proposed in previous chapters, with MPC. This case study saw the ADAP-MPC strategy applied to a high-fidelity virtual model of the *FutureForge* manipulator. The effectiveness of this control strategy was compared to that of a standard, more commonly-used MPC approach across a suite of five tests which were designed to replicate the full operational scope of the physical system. The results provide a clear assessment of the benefit of the proposed ADAP-MPC strategy.

The main finding of this case study is that the ADAP-MPC system consistently demonstrated superior or, at least, equivalent trajectory tracking accuracy compared

to the standard MPC approach. The reduction in steady-state error was particularly significant in these tests, confirming that the ADAP methodology is effective in compensating for the nonlinear dynamics in the plant that have been omitted from the prediction model through the linearisation process.

The introduction of a 1-tonne payload demonstrated that further tuning of the MPC parameters (common to both control strategies) was required. While the transient response characteristics were undesirable for both controllers, the ADAP-MPC approach generally maintained its advantage in steady-state accuracy. The analysis of work done by both controllers indicates that the ADAP-MPC strategy offers greater efficiency within the context of enhanced accuracy. While the control effort expended in manoeuvres through α is very slightly greater for the ADAP-MPC approach, the control effort for manoeuvres through β is more significantly reduced.

The contribution of this case study is the assessment of the integrated ADAP-MPC framework for controlling the *FutureForge* manipulator. The results demonstrate that the ADAP methodology, developed in previous chapters of this work, provides an effective mechanism for correcting predictions made by the linearised prediction model, enabling a simple linear MPC strategy to achieve higher performance without requiring a nonlinear control strategy to be used instead. The results presented in this chapter of the thesis highlight that the ADAP-MPC approach is useful in providing higher accuracy trajectory tracking (than the simple, commonly-used linear MPC approach against which it is compared) without an increase in the computational expense of the resulting control algorithm. Furthermore, this is achieved while providing more energy-efficient trajectory tracking (in the context of the *FutureForge* manipulator).

The most intuitive element of further work, which would augment the present case study, would be to enhance the MPC tuning. Additionally, another appropriate area for further work would be to benchmark the ADAP-MPC strategy against other control strategies, including nonlinear MPC and hybrid approaches as described in Chapter 4.

Chapter 7

Conclusions

Many existing approximation techniques start with an initial, simple guess (or prediction) of the target function and then improve this via calibrated corrections. The ability of linearising approximations to permit the use of linear control strategies is especially appealing due to their associated lower computational costs. The central aim of this thesis was to develop and assess the use of a novel approximation algorithm - the approximation via dual auxiliary problems (ADAP) method. After examining this technique and its utility in solving coupled, nonlinear dynamical systems, the integration of this approach with model-predictive control is explored - extending the ADAP method to the ADAP-MPC strategy. These aims are addressed in two distinct halves of this thesis.

The first half of this thesis dealt with the development of the ADAP methodology and the assessment of its effectiveness in approximating solutions to nonlinear, coupled systems. The method was consistently demonstrated to show comparable accuracy and a reduced computational burden when compared to the well-established Lindstedt-Poincaré technique, particularly in the analysis of the Van der Pol and Rayleigh oscillator system. Despite this advantage, it was observed that the ADAP method is similarly vulnerable to strong nonlinearities in these oscillator problems – losing accuracy, similarly to the asymptotic approximations of the Lindstedt-Poincaré technique, when the small parameter failed to satisfy the condition $\varepsilon \ll 1$.

Applying the ADAP method to the large-scale, high-precision manipulator from the *FutureForge* case study, it was concluded that the ADAP method is unlikely to supersede some traditional approximation methods for ODE solutions computed across long time intervals. While the ADAP methodology provided superior accuracy over the very short, initial time interval of the simulations, the asymptotic approximations proved more accurate over longer durations. This finding - that the ADAP method's advantage lies in the short time interval - was a key motivation for exploring its integration with model-predictive control (MPC), which involves repeatedly updating predictions over short time horizons.

In Chapters 4-6, the focus of the thesis shifted to consider the ADAP-MPC strategy in the context of three examples: a simple inverted pendulum, a cart-mounted pendulum, and the *FutureForge* manipulator. In the first of these examples, the ADAP-MPC is very clearly superior to either of the two feedback linearisation approaches to which it is compared. Not only was the observed trajectory tracking performance improved, but the work required by the controller was reduced. Presenting a more nuanced picture, the cart-mounted pendulum example did not show superior performance for the ADAP-MPC strategy. The trajectory tracking error for this strategy was, in fact, the highest or second-highest of the four methods used in this example. A nuanced benefit of the ADAP-MPC strategy in this example was only observed when comparing the work done by the controller over different disturbance scenarios. It was demonstrated that, as the disturbance to the manipulated variables increased, the percentage increase in work required from the controller was lower for the ADAP-MPC strategy than any of the other control strategies. This contrasting success between the two pendulum-based examples is likely a reflection of a limitation of the ADAP-MPC strategy to deal with strong coupling effects - potentially a repetition of the limitation identified in the Van der Pol and Rayleigh oscillator examples in the preceding half of the thesis.

In the *FutureForge* case study, presented in Chapter 6, the ADAP-MPC strategy

consistently achieved either superior or equivalent accuracy compared to the standard MPC approach presented. The accuracy measures considered here were the transient and steady-state trajectory tracking manoeuvres, which were designed to simulate the real-world operation of the manipulator system. Similar to the previous examples, the ADAP-MPC strategy was identified as resulting in a reduced control effort requirement.

This thesis makes two contributions to the field of system approximations and control. Firstly, it introduces the ADAP method as a novel, computationally efficient algorithm for the approximation of nonlinear, coupled ODE systems. Secondly, and more significantly, it establishes the ADAP-MPC strategy, which provides a novel approach for permitting linear model-predictive control in the context of nonlinear systems. In conclusion, this thesis developed and assessed a novel, accurate, and computationally efficient technique for the solution of nonlinear, coupled systems and also demonstrated its potential in control applications, thereby establishing new and versatile approximation and control strategies.

7.1 Further work

Future work based on the contributions of this thesis focus on further refinement of the ADAP and ADAP-MPC algorithms. Perhaps most intuitively, an immediate area of interest for further work is interrogating the most effective formulations of the first and second auxiliary problems required by the ADAP method. This particular work could see further similarity between this method and existing approximate analytical methods discussed in Chapter 1. In the scenario that this becomes true, then the term “ADAP” would cease to refer to the specific construction of the auxiliary problems as depicted in this thesis; instead, “ADAP” would become a description of the specific architecture of the approximate analytical solution approach.

As is alluded to in Chapter 2, there is some similarity between the ADAP method’s reduction in accuracy in the presence of strong nonlinearities and that of the Lindstedt-Poincaré technique. This discussion in Chapter 2 is limited to the archetypal oscillators

discussed here, but it would be suitable to expand this research in future to consider more generalised parametric studies: determining precise definitions of where and how accuracy degrades in the ADAP method. Importantly, this piece of future work could only pragmatically take place after the previous suggested work of exploring alternative formulations of the auxiliary problems. These constructions undoubtedly have an effect on the algorithm's accuracy and capacity to handle strong nonlinearities.

Lastly, while the ADAP-MPC approach is developed and assessed in this work, there is some curiosity in understanding how effectively the ADAP algorithm integrates with other control strategies. Intuitively, these strategies would have to be model-based, but hybrid control strategies may also present a useful route for further expansion of the technique.

Appendix A

Expanded expressions for the constant coefficients in the *FutureForge* manipulator governing differential equations

$$\begin{aligned} a_1 = & l_{AJ}m_{GW}l_{HG} + 0.5l_{AJ}^2m_{GW} + l_{AJ}l_{HG}m_{MN} + l_{AJ}l_{HG}m_{NW} \\ & + l_{AJ}^2m_{JH} + 0.5l_{AJ}^2m_{MN} + 0.5l_{AJ}^2m_{NW} + J_{AJK} + 0.632358m_{AJK} \\ & + 0.5l_{DG}^2m_{GW} + J_{DG} + 0.5l_{DG}^2m_{MN} + 0.5l_{DG}^2m_{NW} + l_{DG}^2m_{TU} \\ & - 2.7906l_{DG}m_{TU} + 2.47189m_{DG} + l_{EF}^2m_{FGM} + J_{EF} \\ & + 0.25l_{EF}^2m_{EF} + 0.006l_{EF}m_{EF} + 0.000036m_{EF} + 0.5m_{GW}l_{HG}^2 \\ & + 0.5l_{HG}^2m_{MN} + 0.5l_{HG}^2m_{NW} + J_{TU} + 1.94686m_{TU} \end{aligned} \tag{A.1}$$

$$a_2 = -\frac{1}{2} (2l_{AJ}l_{HG} + l_{AJ}^2 - l_{DG}^2 + l_{HG}^2) (m_{GW} + m_{MN} + m_{NW}) \tag{A.2}$$

$$a_3 = 0 \tag{A.3}$$

Appendix A. Expanded expressions for the constant coefficients in the *FutureForge* manipulator governing differential equations

$$\begin{aligned}
a_4 = & l_{AD} (l_{DG} (0.5m_{MN} + m_{TU}) + 1.5694m_{DG} - 1.3953m_{TU}) \\
& + 0.5l_{AJ}m_{GW}l_{JH} + 0.498676l_{AJ}l_{GW}m_{NW} + 0.669137l_{AJ}m_{GW} \\
& + 0.5l_{AJ}l_{JH}m_{MN} + 0.5l_{AJ}l_{JH}m_{NW} + 3.0176l_{AJ}m_{JH} \\
& + 0.833038l_{AJ}m_{MN} \\
& + l_{BC} (l_{DG} (0.5m_{GW} + 0.5m_{NW}) + l_{EF}m_{FGM} + (0.5l_{EF} + 0.006) m_{EF}) \\
& - 0.498676l_{DG}l_{GW}m_{NW} - 0.669137l_{DG}m_{GW} - 0.833038l_{DG}m_{MN} \\
& + 0.5m_{GW}l_{HG}l_{JH} + 0.498676l_{GW}l_{HG}m_{NW} + 0.669137m_{GW}l_{HG} \\
& + 0.5l_{HG}l_{JH}m_{MN} + 0.5l_{HG}l_{JH}m_{NW} + 0.833038l_{HG}m_{MN}
\end{aligned} \tag{A.4}$$

$$\begin{aligned}
a_5 = & 0.5 \left(l_{DG} \left((l_{AD} - 1.66608) m_{MN} + (l_{BC} - 1.33827) m_{GW} \right. \right. \\
& + l_{BC}m_{NW} - 1.9947l_{GU}m_{TU} - 0.997352l_{GW}m_{NW}) \\
& + l_{AJ} \left(m_{NW} (-0.997352l_{GW} - 1.l_{JH}) \right. \\
& + m_{GW} (-1.l_{JH} - 1.33827) + (-1.l_{JH} - 1.66608) m_{MN}) \\
& + 2.78321l_{GU}m_{TU} - 1.m_{GW}l_{HG}l_{JH} - 0.997352l_{GW}l_{HG}m_{NW} \\
& - 1.33827m_{GW}l_{HG} - 1.l_{HG}l_{JH}m_{MN} - 1.l_{HG}l_{JH}m_{NW} \\
& \left. \left. - 1.66608l_{HG}m_{MN} \right) \right)
\end{aligned} \tag{A.5}$$

$$\begin{aligned}
a_6 = & -0.0942l_{AD}m_{DG} \\
& + l_{AJ} (0.036366l_{GW}m_{NW} + 0.02729m_{GW} + 0.0607494m_{MN}) \\
& - 0.036366l_{DG}l_{GW}m_{NW} - 0.02729l_{DG}m_{GW} \\
& - 0.0607494l_{DG}m_{MN} + 0.036366l_{GW}l_{HG}m_{NW} \\
& + 0.02729m_{GW}l_{HG} + 0.0607494l_{HG}m_{MN}
\end{aligned} \tag{A.6}$$

Appendix A. Expanded expressions for the constant coefficients in the *FutureForge* manipulator governing differential equations

$$\begin{aligned}
a_7 = & l_{AJ} (0.036366l_{GW}m_{NW} + 0.02729m_{GW} + 0.0607494m_{MN}) \\
& + l_{DG} \left(0.072732l_{GU}m_{TU} + 0.036366l_{GW}m_{NW} \right. \\
& \quad \left. + 0.02729m_{GW} + 0.0607494m_{MN} \right) \\
& - 0.101483l_{GU}m_{TU} + 0.036366l_{GW}l_{HG}m_{NW} \\
& + 0.02729m_{GW}l_{HG} + 0.0607494l_{HG}m_{MN}
\end{aligned} \tag{A.7}$$

$$a_8 = c_d \tag{A.8}$$

$$a_9 = 0 \tag{A.9}$$

$$a_{10} = 0 \tag{A.10}$$

$$a_{11} = \frac{1}{2} (2l_{AJ}l_{HG} + l_{AJ}^2 - l_{DG}^2 + l_{HG}^2) (m_{GW} + m_{MN} + m_{NW}) \tag{A.11}$$

$$\begin{aligned}
a_{12} = & 0.0942l_{AD}m_{DG} \\
& + l_{AJ} (-0.036366l_{GW}m_{NW} - 0.02729m_{GW} - 0.0607494m_{MN}) \\
& + 0.036366l_{DG}l_{GW}m_{NW} + 0.02729l_{DG}m_{GW} \\
& + 0.0607494l_{DG}m_{MN} - 0.036366l_{GW}l_{HG}m_{NW} \\
& - 0.02729m_{GW}l_{HG} - 0.0607494l_{HG}m_{MN}
\end{aligned} \tag{A.12}$$

Appendix A. Expanded expressions for the constant coefficients in the *FutureForge* manipulator governing differential equations

$$\begin{aligned}
a_{13} = & l_{AJ} (0.036366l_{GW}m_{NW} + 0.02729m_{GW} + 0.0607494m_{MN}) \\
& + l_{DG} \left(0.072732l_{GU}m_{TU} + 0.036366l_{GW}m_{NW} \right. \\
& \quad \left. + 0.02729m_{GW} + 0.0607494m_{MN} \right) \\
& - 0.101483l_{GU}m_{TU} + 0.036366l_{GW}l_{HG}m_{NW} \\
& + 0.02729m_{GW}l_{HG} + 0.0607494l_{HG}m_{MN}
\end{aligned} \tag{A.13}$$

$$\begin{aligned}
a_{14} = & l_{AD} (l_{DG} (0.5m_{MN} + 1.m_{TU}) + 1.5694m_{DG} - 1.3953m_{TU}) \\
& + 0.5l_{AJ}m_{GW}l_{JH} + 0.498676l_{AJ}l_{GW}m_{NW} + 0.669137l_{AJ}m_{GW} \\
& + 0.5l_{AJ}l_{JH}m_{MN} + 0.5l_{AJ}l_{JH}m_{NW} + 3.0176l_{AJ}m_{JH} \\
& + 0.833038l_{AJ}m_{MN} + l_{BC} \left(l_{DG} (0.5m_{GW} + 0.5m_{NW}) \right. \\
& \quad \left. + 1.l_{EF}m_{FGM} + (0.5l_{EF} + 0.006) m_{EF} \right) \\
& - 0.498676l_{DG}l_{GW}m_{NW} - 0.669137l_{DG}m_{GW} \\
& - 0.833038l_{DG}m_{MN} + 0.5m_{GW}l_{HG}l_{JH} \\
& + 0.498676l_{GW}l_{HG}m_{NW} + 0.669137m_{GW}l_{HG} \\
& + 0.5l_{HG}l_{JH}m_{MN} + 0.5l_{HG}l_{JH}m_{NW} + 0.833038l_{HG}m_{MN}
\end{aligned} \tag{A.14}$$

$$\begin{aligned}
a_{15} = & 0.5 \left(l_{DG} (m_{MN} (1.66608 - 1.l_{AD}) + m_{GW} (1.33827 - 1.l_{BC}) \right. \\
& \quad \left. - 1.l_{BC}m_{NW} + 1.9947l_{GU}m_{TU} + 0.997352l_{GW}m_{NW}) \right. \\
& + l_{AJ} (m_{NW} (0.997352l_{GW} + l_{JH}) + m_{GW} (l_{JH} + 1.33827) \\
& \quad \left. + (l_{JH} + 1.66608) m_{MN}) \right. \\
& - 2.78321l_{GU}m_{TU} + m_{GW}l_{HG}l_{JH} + 0.997352l_{GW}l_{HG}m_{NW} \\
& + 1.33827m_{GW}l_{HG} + l_{HG}l_{JH}m_{MN} + l_{HG}l_{JH}m_{NW} \\
& \left. + 1.66608l_{HG}m_{MN} \right)
\end{aligned} \tag{A.15}$$

Appendix A. Expanded expressions for the constant coefficients in the *FutureForge* manipulator governing differential equations

$$\begin{aligned}
a_{16} = & \left(2l_{AJ}m_{JH} + 1.115m_{AJK} + l_{DG}m_{GW} + l_{DG}m_{MN} \right. \\
& + l_{DG}m_{NW} + l_{DG}m_{TU} + 1.5694m_{DG} + l_{EF}m_{FGM} \\
& \left. + \frac{l_{EF}m_{EF}}{2} + 0.006m_{EF} - 1.3953m_{TU} \right) g
\end{aligned} \tag{A.16}$$

$$a_{17} = \left(0.1466m_{AJK} - 0.0942m_{DG} \right) g \tag{A.17}$$

$$\begin{aligned}
b_1 = & l_{AD}^2m_{DG} + l_{AD}^2m_{DST} + J_{AD} + 0.5l_{AD}^2m_{MN} \\
& - 1.66608l_{AD}m_{MN} + l_{AD}^2m_{TU} + 1.48202m_{AD} \\
& + l_{BC}^2m_{CDE} + l_{BC}^2m_{EF} + l_{BC}^2m_{FGM} \\
& - 0.997352l_{BC}l_{GW}m_{NW} + 0.5l_{BC}^2m_{GW} \\
& - 1.33827l_{BC}m_{GW} + J_{BC} + 0.5l_{BC}^2m_{NW} \\
& + 2.69584m_{BC} + 0.0798762m_{DST} + l_{GU}^2m_{TU} \\
& + J_{GW} + 0.997352l_{GW}l_{JH}m_{NW} + 0.5m_{GW}l_{JH}^2 \\
& + 1.33827m_{GW}l_{JH} + l_{GW}^2m_{NW} + 1.79396m_{GW} \\
& + J_{JH} + J_{MN} + J_{ST} + 0.5l_{JH}^2m_{MN} + 1.66608l_{JH}m_{MN} \\
& + 0.5l_{JH}^2m_{NW} + 4.55295m_{JH} + 2.79057m_{MN}
\end{aligned} \tag{A.18}$$

$$\begin{aligned}
b_2 = & -0.544372l_{AD}m_{DST} - 1.9947l_{AD}l_{GU}m_{TU} + \frac{1}{2}l_{AD}^2m_{MN} \\
& - 1.66608l_{AD}m_{MN} - 0.997352l_{BC}l_{GW}m_{NW} + \frac{1}{2}l_{BC}^2m_{GW} \\
& - 1.33827l_{BC}m_{GW} + \frac{1}{2}l_{BC}^2m_{NW} - 0.997352l_{GW}l_{JH}m_{NW} \\
& - \frac{1}{2}m_{GW}l_{JH}^2 - 1.33827m_{GW}l_{JH} - \frac{1}{2}l_{JH}^2m_{MN} \\
& - 1.66608l_{JH}m_{MN} - \frac{1}{2}l_{JH}^2m_{NW}
\end{aligned} \tag{A.19}$$

Appendix A. Expanded expressions for the constant coefficients in the *FutureForge* manipulator governing differential equations

$$\begin{aligned}
b_3 = & 0.152196l_{AD}m_{DST} + 0.145464l_{AD}l_{GU}m_{TU} + 0.121499l_{AD}m_{MN} \\
& + 0.072732l_{BC}l_{GW}m_{NW} + 0.05458l_{BC}m_{GW} + 0.072732l_{GW}l_{JH}m_{NW} \\
& + 0.05458m_{GW}l_{JH} + 0.121499l_{JH}m_{MN}
\end{aligned} \tag{A.20}$$

$$\begin{aligned}
b_4 = & \frac{1}{2}l_{AD}l_{DG}m_{MN} + l_{AD}l_{DG}m_{TU} + 1.5694l_{AD}m_{DG} - 1.3953l_{AD}m_{TU} \\
& + \frac{1}{2}l_{AJ}m_{GW}l_{JH} + 0.498676l_{AJ}l_{GW}m_{NW} + 0.669137l_{AJ}m_{GW} \\
& + \frac{1}{2}l_{AJ}l_{JH}m_{MN} + \frac{1}{2}l_{AJ}l_{JH}m_{NW} + 3.0176l_{AJ}m_{JH} \\
& + 0.833038l_{AJ}m_{MN} + \frac{1}{2}l_{BC}l_{DG}m_{GW} + \frac{1}{2}l_{BC}l_{DG}m_{NW} \\
& + l_{BC}l_{EF}m_{FGM} + 0.006l_{BC}m_{EF} + \frac{1}{2}l_{BC}l_{EF}m_{EF} \\
& - 0.498676l_{DG}l_{GW}m_{NW} - 0.669137l_{DG}m_{GW} \\
& - 0.833038l_{DG}m_{MN} + \frac{1}{2}m_{GW}l_{HG}l_{JH} + 0.498676l_{GW}l_{HG}m_{NW} \\
& + 0.669137m_{GW}l_{HG} + \frac{1}{2}l_{HG}l_{JH}m_{MN} + \frac{1}{2}l_{HG}l_{JH}m_{NW} \\
& + 0.833038l_{HG}m_{MN}
\end{aligned} \tag{A.21}$$

$$\begin{aligned}
b_5 = & \frac{1}{2}l_{AD}l_{DG}m_{MN} - \frac{1}{2}l_{AJ}m_{GW}l_{JH} - 0.498676l_{AJ}l_{GW}m_{NW} \\
& - 0.669137l_{AJ}m_{GW} - \frac{1}{2}l_{AJ}l_{JH}m_{MN} - \frac{1}{2}l_{AJ}l_{JH}m_{NW} \\
& - 0.833038l_{AJ}m_{MN} + \frac{1}{2}l_{BC}l_{DG}m_{GW} + \frac{1}{2}l_{BC}l_{DG}m_{NW} \\
& - 0.997352l_{DG}l_{GU}m_{TU} - 0.498676l_{DG}l_{GW}m_{NW} \\
& - 0.669137l_{DG}m_{GW} - 0.833038l_{DG}m_{MN} + 1.3916l_{GU}m_{TU} \\
& - \frac{1}{2}m_{GW}l_{HG}l_{JH} - 0.498676l_{GW}l_{HG}m_{NW} \\
& - 0.669137m_{GW}l_{HG} - \frac{1}{2}l_{HG}l_{JH}m_{MN} - \frac{1}{2}l_{HG}l_{JH}m_{NW} \\
& - 0.833038l_{HG}m_{MN}
\end{aligned} \tag{A.22}$$

Appendix A. Expanded expressions for the constant coefficients in the *FutureForge* manipulator governing differential equations

$$\begin{aligned}
b_6 = & -0.0942l_{AD}m_{DG} + 0.036366l_{AJ}l_{GW}m_{NW} + 0.02729l_{AJ}m_{GW} \\
& + 0.0607494l_{AJ}m_{MN} - 0.036366l_{DG}l_{GW}m_{NW} \\
& - 0.02729l_{DG}m_{GW} - 0.0607494l_{DG}m_{MN} \\
& + 0.036366l_{GW}l_{HG}m_{NW} + 0.02729m_{GW}l_{HG} \\
& + 0.0607494l_{HG}m_{MN}
\end{aligned} \tag{A.23}$$

$$\begin{aligned}
b_7 = & 0.036366l_{AJ}l_{GW}m_{NW} + 0.02729l_{AJ}m_{GW} + 0.0607494l_{AJ}m_{MN} \\
& + 0.072732l_{DG}l_{GU}m_{TU} + 0.036366l_{DG}l_{GW}m_{NW} \\
& + 0.02729l_{DG}m_{GW} + 0.0607494l_{DG}m_{MN} - 0.101483l_{GU}m_{TU} \\
& + 0.036366l_{GW}l_{HG}m_{NW} + 0.02729m_{GW}l_{HG} + 0.0607494l_{HG}m_{MN}
\end{aligned} \tag{A.24}$$

$$b_8 = c_d \tag{A.25}$$

$$b_9 = 0 \tag{A.26}$$

$$\begin{aligned}
b_{10} = & 0.152196l_{AD}m_{DST} + 0.145464l_{AD}l_{GU}m_{TU} + 0.121499l_{AD}m_{MN} \\
& + 0.072732l_{BC}l_{GW}m_{NW} + 0.05458l_{BC}m_{GW} \\
& + 0.072732l_{GW}l_{JH}m_{NW} + 0.05458m_{GW}l_{JH} + 0.121499l_{JH}m_{MN}
\end{aligned} \tag{A.27}$$

Appendix A. Expanded expressions for the constant coefficients in the *FutureForge* manipulator governing differential equations

$$\begin{aligned}
b_{11} = & 0.544372l_{\text{AD}}m_{\text{DST}} + 1.9947l_{\text{AD}}l_{\text{GU}}m_{\text{TU}} - \frac{1}{2}l_{\text{AD}}^2m_{\text{MN}} \\
& + 1.66608l_{\text{AD}}m_{\text{MN}} + 0.997352l_{\text{BC}}l_{\text{GW}}m_{\text{NW}} - \frac{1}{2}l_{\text{BC}}^2m_{\text{GW}} \\
& + 1.33827l_{\text{BC}}m_{\text{GW}} - \frac{1}{2}l_{\text{BC}}^2m_{\text{NW}} + 0.997352l_{\text{GW}}l_{\text{JH}}m_{\text{NW}} \\
& + \frac{1}{2}m_{\text{GW}}l_{\text{JH}}^2 + 1.33827m_{\text{GW}}l_{\text{JH}} + \frac{1}{2}l_{\text{JH}}^2m_{\text{MN}} \\
& + 1.66608l_{\text{JH}}m_{\text{MN}} + \frac{1}{2}l_{\text{JH}}^2m_{\text{NW}}
\end{aligned} \tag{A.28}$$

$$\begin{aligned}
b_{12} = & -0.0942l_{\text{AD}}m_{\text{DG}} + 0.036366l_{\text{AJ}}l_{\text{GW}}m_{\text{NW}} + 0.02729l_{\text{AJ}}m_{\text{GW}} \\
& + 0.0607494l_{\text{AJ}}m_{\text{MN}} - 0.036366l_{\text{DG}}l_{\text{GW}}m_{\text{NW}} \\
& - 0.02729l_{\text{DG}}m_{\text{GW}} - 0.0607494l_{\text{DG}}m_{\text{MN}} \\
& + 0.036366l_{\text{GW}}l_{\text{HG}}m_{\text{NW}} + 0.02729m_{\text{GW}}l_{\text{HG}} + 0.0607494l_{\text{HG}}m_{\text{MN}}
\end{aligned} \tag{A.29}$$

$$\begin{aligned}
b_{13} = & 0.036366l_{\text{AJ}}l_{\text{GW}}m_{\text{NW}} + 0.02729l_{\text{AJ}}m_{\text{GW}} + 0.0607494l_{\text{AJ}}m_{\text{MN}} \\
& + 0.072732l_{\text{DG}}l_{\text{GU}}m_{\text{TU}} + 0.036366l_{\text{DG}}l_{\text{GW}}m_{\text{NW}} \\
& + 0.02729l_{\text{DG}}m_{\text{GW}} + 0.0607494l_{\text{DG}}m_{\text{MN}} - 0.101483l_{\text{GU}}m_{\text{TU}} \\
& + 0.036366l_{\text{GW}}l_{\text{HG}}m_{\text{NW}} + 0.02729m_{\text{GW}}l_{\text{HG}} + 0.0607494l_{\text{HG}}m_{\text{MN}}
\end{aligned} \tag{A.30}$$

Appendix A. Expanded expressions for the constant coefficients in the *FutureForge* manipulator governing differential equations

$$\begin{aligned}
b_{14} = & -\frac{1}{2}l_{AD}l_{DG}m_{MN} - l_{AD}l_{DG}m_{TU} - 1.5694l_{AD}m_{DG} \\
& + 1.3953l_{AD}m_{TU} - \frac{1}{2}l_{AJ}m_{GW}l_{JH} \\
& - 0.498676l_{AJ}l_{GW}m_{NW} - 0.669137l_{AJ}m_{GW} \\
& - \frac{1}{2}l_{AJ}l_{JH}m_{MN} - \frac{1}{2}l_{AJ}l_{JH}m_{NW} - 3.0176l_{AJ}m_{JH} \\
& - 0.833038l_{AJ}m_{MN} - \frac{1}{2}l_{BC}l_{DG}m_{GW} - \frac{1}{2}l_{BC}l_{DG}m_{NW} \\
& - l_{BC}l_{EF}m_{FGM} - 0.006l_{BC}m_{EF} - \frac{1}{2}l_{BC}l_{EF}m_{EF} \\
& + 0.498676l_{DG}l_{GW}m_{NW} + 0.669137l_{DG}m_{GW} \\
& + 0.833038l_{DG}m_{MN} - \frac{1}{2}m_{GW}l_{HG}l_{JH} \\
& - 0.498676l_{GW}l_{HG}m_{NW} - 0.669137m_{GW}l_{HG} - \frac{1}{2}l_{HG}l_{JH}m_{MN} \\
& - \frac{1}{2}l_{HG}l_{JH}m_{NW} - 0.833038l_{HG}m_{MN}
\end{aligned} \tag{A.31}$$

$$\begin{aligned}
b_{15} = & -\frac{1}{2}l_{AD}l_{DG}m_{MN} + \frac{1}{2}l_{AJ}m_{GW}l_{JH} + 0.498676l_{AJ}l_{GW}m_{NW} \\
& + 0.669137l_{AJ}m_{GW} + \frac{1}{2}l_{AJ}l_{JH}m_{MN} + \frac{1}{2}l_{AJ}l_{JH}m_{NW} \\
& + 0.833038l_{AJ}m_{MN} - \frac{1}{2}l_{BC}l_{DG}m_{GW} - \frac{1}{2}l_{BC}l_{DG}m_{NW} \\
& + 0.997352l_{DG}l_{GU}m_{TU} + 0.498676l_{DG}l_{GW}m_{NW} \\
& + 0.669137l_{DG}m_{GW} + 0.833038l_{DG}m_{MN} - 1.3916l_{GU}m_{TU} \\
& + \frac{1}{2}m_{GW}l_{HG}l_{JH} + 0.498676l_{GW}l_{HG}m_{NW} + 0.669137m_{GW}l_{HG} \\
& + \frac{1}{2}l_{HG}l_{JH}m_{MN} + \frac{1}{2}l_{HG}l_{JH}m_{NW} + 0.833038l_{HG}m_{MN}
\end{aligned} \tag{A.32}$$

Appendix A. Expanded expressions for the constant coefficients in the *FutureForge* manipulator governing differential equations

$$\begin{aligned}
b_{16} = & \left(l_{AD}m_{DG} + l_{AD}m_{DST} + l_{AD}m_{MN} + l_{AD}m_{TU} \right. \\
& + 1.217m_{AD} + l_{BC}m_{CDE} + l_{BC}m_{EF} + l_{BC}m_{FGM} \\
& + l_{BC}m_{GW} + l_{BC}m_{NW} + 1.6419m_{BC} \\
& - 0.272186m_{DST} - 0.997352l_{GU}m_{TU} \\
& - 0.997352l_{GW}m_{NW} - 1.33827m_{GW} \\
& \left. + 3.0176m_{JH} - 1.66608m_{MN} \right) g
\end{aligned} \tag{A.33}$$

$$\begin{aligned}
b_{17} = & 0.0305m_{AD} + 0.0760982m_{DST} + 0.072732l_{GU}m_{TU} \\
& + 0.072732l_{GW}m_{NW} + 0.05458m_{GW} + 0.121499m_{MN}
\end{aligned} \tag{A.34}$$

Appendix B

Statement on the inclusion of previous work

As indicated by the boxed statements at the beginning of Chapters 2, 3, and 6, several previously published works ([59], [58], [43], and [44]) feature prominently in this thesis and form part of its content.

The earliest of these works features the author as a minor contributor under the lead authorship of their supervisor [59]. This work, titled *Modelling the dynamics of a large-scale industrial manipulator for precision control*, saw the derivation of the equations of motion of the *FutureForge* manipulator. It was published in the *Russian Academy of Sciences Summer School on Advanced Problems in Mechanics* in 2021. In this thesis, the derivation presented in [59] is adapted in Chapter 3 to account for revisions to the manipulator design.

Subsequently, the author contributed (as lead author) to a conference paper presented at the *10th European Nonlinear Dynamics Conference* in 2022 [58]. This work, titled *The FutureForge manipulator and an approximate analytical solution algorithm for its nonlinear dynamics*, proposed a means of solving a linearised version of the equations of motion for the manipulator. As such, this approach formed the basis of applying the ADAP methodology to the manipulator system and forms part of the

Appendix B. Statement on the inclusion of previous work

work in 3.

In 2023, Johnston and Cartmell proposed the ADAP methodology (although not referring to it by this name) under the title *A novel approximation method for the solution of weakly nonlinear coupled systems*, [43]. This peer-reviewed article in *Nonlinear Dynamics* used the examples of coupled systems of Van der Pol oscillators and Rayleigh oscillators alongside the *FutureForge* manipulator as case studies in this method's application. As such, this paper forms a large portion of Chapters 2 and 3.

Lastly, in 2025, Johnston and Cartmell extended the ADAP methodology to focus specifically on the model-based control of the manipulator system [44]. This final peer-reviewed article was published in a Special Issue of the *Journal of Sound and Vibration*, and was titled *Application of the ADAP method to model-based control of the FutureForge manipulator*. Given its focus on model-based control and the ADAP method, this publication contributes a large portion of Chapter 6.

Bibliography

- [1] W. Albalawi, S. A. El-Tantawy, and S. A. Alkhateeb, “The phase shift analysis of the colliding dissipative kdv solitons,” *Journal of Ocean Engineering and Science*, vol. 7, no. 6, pp. 521–527, 2022, ISSN: 2468-0133. DOI: <https://doi.org/10.1016/j.joes.2021.09.021>
- [2] M. M. Bhatti and D. Q. Lu, *Analytical study of the head-on collision process between hydroelastic solitary waves in the presence of a uniform current*, Electronic Article, 2019. DOI: 10.3390/sym11030333
- [3] K. Narahara, “Head-on collision of solitary waves in coupled korteweg–de vries systems modeling nonlinear transmission lines,” *Wave Motion*, vol. 51, no. 6, pp. 935–946, 2014, ISSN: 0165-2125. DOI: <https://doi.org/10.1016/j.wavemoti.2014.03.006>
- [4] K. Narahara, “Characterization of collision-induced generation of pulses in coupled electrical nonlinear transmission lines,” *Japanese Journal of Applied Physics*, vol. 53, no. 6, p. 067 301, 2014, ISSN: 1347-4065 0021-4922. DOI: 10.7567/JJAP.53.067301
- [5] M. Avey, F. Tornabene, N. M. Aslanova, and A. H. Sofiyev, *The application of the modified lindstedt–poincaré method to solve the nonlinear vibration problem of exponentially graded laminated plates on elastic foundations*, Electronic Article, 2024. DOI: 10.3390/math12050749
- [6] F. Vestroni, A. Luongo, and A. Paolone, “A perturbation method for evaluating nonlinear normal modes of a piecewise linear two-degrees-of-freedom system,”

Bibliography

- Nonlinear Dynamics*, vol. 54, no. 4, pp. 379–393, 2008, ISSN: 1573-269X. DOI: 10.1007/s11071-008-9337-3
- [7] P. Amore and A. Aranda, “Improved lindstedt-poincare method for the solution of nonlinear problems,” *Journal of Sound and Vibrations*, vol. 283, pp. 1115–1136, 2005, ISSN: 0022-460X. DOI: 10.1016/j.jsv.2004.06.009
- [8] Z. Li and J. Tang, “High accurate homo-heteroclinic solutions of certain strongly nonlinear oscillators based on generalized padé–lindstedt–poincaré method,” *Journal of Vibration Engineering & Technologies*, vol. 10, no. 4, pp. 1291–1308, 2022, ISSN: 2523-3939. DOI: 10.1007/s42417-022-00446-7
- [9] M. Cartmell, S. Ziegler, R. Khanin, and D. Forehand, “Multiple scales analyses of the dynamics of weakly nonlinear mechanical systems,” *Applied Mechanics Reviews*, vol. 56, no. 5, pp. 455–492, Aug. 2003, ISSN: 0003-6900. DOI: 10.1115/1.1581884 eprint: https://asmedigitalcollection.asme.org/appliedmechanicsreviews/article-pdf/56/5/455/5440201/455_1.pdf. [Online]. Available: <https://doi.org/10.1115/1.1581884>
- [10] M. Kiani-Oshtorjani, S. Ustinov, H. Handroos, P. Jalali, and A. Mikkola, “Real-time simulation of fluid power systems containing small oil volumes, using the method of multiple scales,” *IEEE Access*, vol. 8, pp. 196 940–196 950, 2020, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3034698
- [11] E. Pipitone, C. M. Firrone, and S. Zucca, *Application of multiple-scales method for the dynamic modelling of a gear coupling*, Electronic Article, 2019. DOI: 10.3390/app9061225
- [12] S. Ilyas, F. K. Alfosail, and M. I. Younis, “On the application of the multiple scales method on electrostatically actuated resonators,” *Journal of Computational and Nonlinear Dynamics*, vol. 14, no. 4, 2019, ISSN: 1555-1415. DOI: 10.1115/1.4042694
- [13] H. Dong, X. Zheng, J. Cui, Y. Nie, Z. Yang, and Z. Yang, “High-order three-scale computational method for dynamic thermo-mechanical problems of composite structures with multiple spatial scales,” *International Journal of Solids*

Bibliography

- and Structures*, vol. 169, pp. 95–121, 2019, ISSN: 0020-7683. DOI: <https://doi.org/10.1016/j.ijsolstr.2019.04.017>
- [14] M. Arshad, S. F. Aldosary, S. Batool, I. Hussain, and N. Hussain, “Exploring fractional-order new coupled korteweg-de vries system via improved adomian decomposition method,” *PLOS ONE*, vol. 19, 2024. DOI: [10.1371/journal.pone.0303426](https://doi.org/10.1371/journal.pone.0303426)
- [15] E. A. A. Ziada, “Solution of nonlinear fractional differential equations using adomain decomposition method,” *International Journal of Systems Science and Applied Mathematics*, 2021. DOI: [10.11648/J.IJSSAM.20210604.11](https://doi.org/10.11648/J.IJSSAM.20210604.11)
- [16] M. Turkyilmazoglu, “Nonlinear problems via a convergence accelerated decomposition method of adomian,” *Computer Modeling in Engineering & Sciences*, vol. 127, no. 1, pp. 1–22, 2021, ISSN: 1526-1506. DOI: [10.32604/cmes.2021.012595](https://doi.org/10.32604/cmes.2021.012595) [Online]. Available: <http://www.techscience.com/CMES/v127n1/41942>
- [17] M. A. Abdou and A. A. Soliman, “New applications of variational iteration method,” *Physica D: Nonlinear Phenomena*, vol. 211, no. 1, pp. 1–8, 2005, ISSN: 0167-2789. DOI: <https://doi.org/10.1016/j.physd.2005.08.002>
- [18] J.-H. He and X.-H. Wu, “Variational iteration method: New development and applications,” *Computers & Mathematics with Applications*, vol. 54, no. 7, pp. 881–894, 2007, ISSN: 0898-1221. DOI: <https://doi.org/10.1016/j.camwa.2006.12.083>
- [19] M. Turkyilmazoglu, “An optimal variational iteration method,” *Applied Mathematics Letters*, vol. 24, no. 5, pp. 762–765, 2011, ISSN: 0893-9659. DOI: <https://doi.org/10.1016/j.aml.2010.12.032>
- [20] M. Shirazian, “A new acceleration of variational iteration method for initial value problems,” *Mathematics and Computers in Simulation*, vol. 214, pp. 246–259, 2023, ISSN: 0378-4754. DOI: <https://doi.org/10.1016/j.matcom.2023.07.002>

Bibliography

- [21] A. Bhargava, D. Jain, and D. L. Suthar, “Applications of the laplace variational iteration method to fractional heat like equations,” *Partial Differential Equations in Applied Mathematics*, vol. 8, p. 100 540, 2023, ISSN: 2666-8181. DOI: <https://doi.org/10.1016/j.padiff.2023.100540>
- [22] H. Ahmad, T. A. Khan, H. Durur, G. M. Ismail, and A. Yokus, “Analytic approximate solutions of diffusion equations arising in oil pollution,” *Journal of Ocean Engineering and Science*, vol. 6, no. 1, pp. 62–69, 2021, ISSN: 2468-0133. DOI: <https://doi.org/10.1016/j.joes.2020.05.002>
- [23] P. Jalili, B. Jalili, I. Ahmad, A. S. Hendy, M. R. Ali, and D. D. Ganji, “Python approach for using homotopy perturbation method to investigate heat transfer problems,” *Case Studies in Thermal Engineering*, vol. 54, p. 104 049, 2024, ISSN: 2214-157X. DOI: <https://doi.org/10.1016/j.csite.2024.104049>
- [24] H. Guan et al., “An improved homotopy perturbation method for dynamic force reconstruction,” *Structures*, vol. 54, pp. 1237–1242, 2023, ISSN: 2352-0124. DOI: <https://doi.org/10.1016/j.istruc.2023.05.075>
- [25] N. Sharif, H. U. Molla, and M. S. Alam, “Analytical solution of couple-mass-spring systems by novel homotopy perturbation method,” *International Journal of Non-Linear Mechanics*, vol. 167, p. 104 923, 2024, ISSN: 0020-7462. DOI: <https://doi.org/10.1016/j.ijnonlinmec.2024.104923>
- [26] O. Alkhazaleh, “Modification of homotopy perturbation method for addressing systems of pdes,” *Partial Differential Equations in Applied Mathematics*, vol. 13, p. 101 065, 2025, ISSN: 2666-8181. DOI: <https://doi.org/10.1016/j.padiff.2024.101065>
- [27] T. Detroux, L. Renson, L. Masset, and G. Kerschen, “The harmonic balance method for bifurcation analysis of large-scale nonlinear mechanical systems,” *Computer Methods in Applied Mechanics and Engineering*, vol. 296, pp. 18–38, 2015, ISSN: 0045-7825. DOI: <https://doi.org/10.1016/j.cma.2015.07.017>
- [28] V. Jaumouillé, J. J. Sinou, and B. Petitjean, “An adaptive harmonic balance method for predicting the nonlinear dynamic responses of mechanical systems

Bibliography

- application to bolted structures,” *Journal of Sound and Vibration*, vol. 329, no. 19, pp. 4048–4067, 2010, ISSN: 0022-460X. DOI: <https://doi.org/10.1016/j.jsv.2010.04.008>
- [29] P. Sun, X. Zhao, X. Yu, Q. Huang, Z. Feng, and J. Zhou, “Incremental harmonic balance method for multi-harmonic solution of high-dimensional delay differential equations: Application to crossflow-induced nonlinear vibration of steam generator tubes,” *Applied Mathematical Modelling*, vol. 118, pp. 818–831, 2023, ISSN: 0307-904X. DOI: <https://doi.org/10.1016/j.apm.2023.02.018>
- [30] Y. Chen, L. Hou, R. Lin, W. Toh, T. Y. Ng, and Y. Chen, “A general and efficient harmonic balance method for nonlinear dynamic simulation,” *International Journal of Mechanical Sciences*, vol. 276, p. 109388, 2024, ISSN: 0020-7403. DOI: <https://doi.org/10.1016/j.ijmecsci.2024.109388>
- [31] H.-y. Kwon and H.-l. Choi, “Gain scheduling control of nonlinear systems based on approximate input-output linearization: Ijcas,” *International Journal of Control, Automation, and Systems*, vol. 12, no. 5, pp. 1131–1137, 2014, ISSN: 15986446. DOI: <https://doi.org/10.1007/s12555-013-0462-5>
- [32] D. Liu, C. An, C. Xie, and G. Hong, *Multibody dynamic characteristics of wingtip-jointed composite aircrafts*, Conference Paper, 2019. DOI: [10.1109/ICMAE.2019.8880950](https://doi.org/10.1109/ICMAE.2019.8880950)
- [33] K. N. Mobariz, E. N. Mobarez, and M. M. Ashry, “Fuel cell propulsion control system based on pid approaches,” in *2021 17th International Computer Engineering Conference (ICENCO)*, 2021, pp. 82–87, ISBN: 2475-2320. DOI: [10.1109/ICENCO49852.2021.9698973](https://doi.org/10.1109/ICENCO49852.2021.9698973)
- [34] O. Luderer and F. Thielecke, “Aircraft load estimation using linear parameter-varying system-based hybrid observers,” *Journal of Aircraft*, vol. 62, no. 3, pp. 517–529, 2025, ISSN: 0021-8669. DOI: [10.2514/1.C037946](https://doi.org/10.2514/1.C037946)
- [35] M. Storace and O. D. Feo, “Piecewise-linear approximation of nonlinear dynamical systems,” *IEEE Transactions on Circuits and Systems I: Regular Papers*,

Bibliography

- vol. 51, no. 4, pp. 830–842, 2004, ISSN: 1558-0806. DOI: 10.1109/TCSI.2004.823664
- [36] E. Camponogara and L. F. Nazari, “Models and algorithms for optimal piecewise-linear function approximation,” *Mathematical Problems in Engineering*, vol. 2015, no. 1, p. 876 862, 2015, ISSN: 1024-123X. DOI: <https://doi.org/10.1155/2015/876862>
- [37] H. Dong et al., “Plac: Piecewise linear approximation computation for all non-linear unary functions,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 9, pp. 2014–2027, 2020, ISSN: 1557-9999. DOI: 10.1109/TVLSI.2020.3004602
- [38] F. Birkelbach, D. Huber, and R. Hofmann, “Piecewise linear approximation for milp leveraging piecewise convexity to improve performance,” *Computers & Chemical Engineering*, vol. 183, p. 108 596, 2024, ISSN: 0098-1354. DOI: <https://doi.org/10.1016/j.compchemeng.2024.108596>
- [39] D. Tyner and A. Lewis, “Jacobian linearisation in a geometric setting,” in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, vol. 6, 2003, 6084–6089 Vol.6. DOI: 10.1109/CDC.2003.1272230
- [40] W. Li and Y. Pang, “Application of adomian decomposition method to nonlinear systems,” *Advances in Difference Equations*, vol. 2020, no. 1, p. 67, 2020, ISSN: 1687-1847. DOI: 10.1186/s13662-020-2529-y
- [41] J.-H. He, “Homotopy perturbation method with an auxiliary term,” *Abstract and Applied Analysis*, vol. 2012, no. 1, p. 857 612, 2012. DOI: <https://doi.org/10.1155/2012/857612> eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1155/2012/857612>. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1155/2012/857612>
- [42] H. Dai, Z. Yan, X. Wang, X. Yue, and S. N. Atluri, “Collocation-based harmonic balance framework for highly accurate periodic solution of nonlinear dynamical system,” *International Journal for Numerical Methods in Engineering*, vol. 124, no. 2, pp. 458–481, 2023. DOI: <https://doi.org/10.1002/nme.7128> eprint:

Bibliography

- <https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.7128>. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.7128>
- [43] D. Johnston and M. Cartmell, “A novel approximation method for the solution of weakly nonlinear coupled systems,” *Nonlinear Dynamics*, vol. 111, no. 17, pp. 16 271–16 292, 2023, ISSN: 1573-269X. DOI: 10.1007/s11071-023-08723-0
- [44] D. Johnston and M. Cartmell, “Application of the ADAP method to model-based control of the *FutureForge* manipulator,” *Journal of Sound and Vibration*, vol. 617, p. 119 302, 2025, ISSN: 0022-460X. DOI: <https://doi.org/10.1016/j.jsv.2025.119302> [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0022460X25003761>
- [45] Q. Guo, W. Perruquetti, and D. Efimov, “Universal robust adaptive control of robot manipulators using real time estimation,” in *IFAC-PapersOnLine*, A. Bobtsov, S. Kolyubin, A. Pyrkin, and A. Fradkov, Eds., vol. 48, Elsevier B.V., 2015, pp. 499–504, ISBN: 24058963 (ISSN). DOI: 10.1016/j.ifacol.2015.09.235
- [46] S. S. Saab and P. Ghanem, “A multivariable stochastic tracking controller for robot manipulators without joint velocities,” *IEEE Trans. Autom. Control*, vol. 63, no. 8, pp. 2481–2495, 2018, ISSN: 1558-2523. DOI: 10.1109/TAC.2017.2771154
- [47] A. Andreev and O. Peregudova, “Trajectory tracking control for robot manipulators using only position measurements,” *Int. J. Control*, vol. 92, no. 7, pp. 1490–1496, 2019, ISSN: 00207179 (ISSN). DOI: 10.1080/00207179.2017.1397755
- [48] L. Cveticanin, “Lord rayleigh and rayleigh oscillator: An overview,” in *The 14th IFToMM World Congress*, 2015. DOI: 10.6567/IFToMM.14TH.WC.OS20.012
- [49] Q. Fu, T. Luo, C. Pan, and G. Wu, “Central pattern generator-based coupling control method for synchronously controlling the two-degrees-of-freedom robot,” *Science Progress*, vol. 103, no. 1, p. 0036 850 419 877 731, 2020, PMID: 31829884. DOI: 10.1177/0036850419877731 eprint: <https://doi.org/10.1177/0036850419877731>. [Online]. Available: <https://doi.org/10.1177/0036850419877731>

Bibliography

- [50] A. de Pina Filho, M. Dutra, and L. Raptopoulos, "Modeling of a bipedal robot using mutually coupled rayleigh oscillators," *Biol. Cybern.*, vol. 92, pp. 1–7, 2005. DOI: 10.1007/s00422-004-0531-1
- [51] S. Ghosh and D. Ray, "Chemical oscillator as a generalized rayleigh oscillator," *J. Chem. Phys.*, vol. 139, 2013. DOI: 10.1063/1.4826169
- [52] B. van der Pol and J. van der Mark, "Lxxii. the heartbeat considered as a relaxation oscillation, and an electrical model of the heart," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 6, no. 38, pp. 763–775, 1928. DOI: 10.1080/14786441108564652 eprint: <https://doi.org/10.1080/14786441108564652>. [Online]. Available: <https://doi.org/10.1080/14786441108564652>
- [53] E. Ryzhii and M. Ryzhii, "A heterogeneous coupled oscillator model for simulation of ecg signals," *Computer Methods and Programs in Biomedicine*, vol. 117, no. 1, pp. 40–49, 2014, SI: ACBIT 2013, ISSN: 0169-2607. DOI: <https://doi.org/10.1016/j.cmpb.2014.04.009> [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0169260714001485>
- [54] O. O. Onyejekwe, "Dynamical computations of the fitzhugh-nagumo equation," *Archive of Biomedical Science and Engineering*, pp. 027–036, Sep. 2021, ISSN: 2641-3027. DOI: 10.17352/abse.000026 [Online]. Available: <http://dx.doi.org/10.17352/abse.000026>
- [55] M. Facchinetti, E. de Langre, and B. F., "Vortex shedding modeling using diffusive van der pol oscillators," *C.R. Mec.*, vol. 330, pp. 451–456, 2002. DOI: 10.1016/S1631-0721(02)01492-4
- [56] A. M. Marra, C. Mannini, and G. Bartoli, "Van der pol-type equation for modeling vortex-induced oscillations of bridge decks," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 99, no. 6, pp. 776–785, 2011, The Eleventh Italian National Conference on Wind Engineering, IN-VENTO-2010, Spoleto, Italy, June 30th - July 3rd 2010, ISSN: 0167-6105. DOI: <https://doi.org/10.1016/j.jweia.2011.05.009>

Bibliography

- org/10.1016/j.jweia.2011.03.014 [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0167610511000742>
- [57] J. He and J. Cai, “Design of a new chaotic system based on van der pol oscillator and its encryption application,” *Mathematics*, 2019. [Online]. Available: <https://api.semanticscholar.org/CorpusID:201110149>
- [58] D. Johnston, M. Cartmell, B. Wynne, V. Pakrashi, and I. Kovacic, “The *FutureForge* manipulator and an approximate analytical solution algorithm for its nonlinear dynamics,” in *10th European Nonlinear Dynamics Conference*, 2022, pp. 774–783.
- [59] M. Cartmell, I. Gordon, D. Johnston, S. Liang, L. McIntosh, and B. Wynne, “Modelling the dynamics of a large-scale industrial manipulator for precision control,” in *Russian Academy of Sciences Summer School on Advanced Problems in Mechanics*, 2021, pp. 298–318. DOI: 10.1007/978-3-031-37246-9_23
- [60] Y. Chen, G. Leitmann, and J. Chen, “Robust control for rigid serial manipulators: A general setting,” in *American Control Conference*, vol. 2, IEEE, 1998, 912–916 vol.2, ISBN: 0743-1619. DOI: 10.1109/ACC.1998.703540
- [61] D. Meldrum, G. Rodriguez, and G. Franklin, “An order (n) recursive inversion of the jacobian for an n-link serial manipulator,” in *IEEE International Conference on Robotics and Automation*, vol. 2, IEEE Comput. Soc. Press, 1991, pp. 1175–1180. DOI: 10.1109/ROBOT.1991.131768
- [62] S. Saha, “A decomposition of the manipulator inertia matrix,” *IEEE Transactions on Robotics and Automation*, vol. 13, no. 2, pp. 301–304, 1997, ISSN: 1042-296X. DOI: 10.1109/70.563652
- [63] R. Gazit and B. Widrow, “Backpropagation through links: A new approach to kinematic control of serial manipulators,” in *Tenth International Symposium on Intelligent Control*, IEEE, 1995, pp. 99–104, ISBN: 2158-9860. DOI: 10.1109/ISIC.1995.525044

Bibliography

- [64] F. Ouyang and T. Zhang, "Virtual velocity vector-based offline collision-free path planning of industrial robotic manipulator," *International journal of advanced robotic systems*, vol. 12, no. 9, p. 129, 2015, ISSN: 1729-8806 1729-8814. DOI: 10.5772/60127
- [65] M. Talebpour and M. Namvar, "Zero-gravity emulation of satellites in presence of uncalibrated sensors and model uncertainties," in *2009 IEEE Control Applications, (CCA) & Intelligent Control, (ISIC)*, IEEE, 2009, pp. 1063–1068, ISBN: 1085-1992. DOI: 10.1109/CCA.2009.5280698
- [66] S. A. Ajwad, J. Iqbal, M. I. Ullah, and A. Mehmood, "A systematic review of current and emergent manipulator control approaches," *Frontiers of Mechanical Engineering*, vol. 10, no. 2, pp. 198–210, 2015, ISSN: 2095-0241. DOI: 10.1007/s11465-015-0335-0
- [67] Y. D. Patel and P. M. George, "Parallel manipulator applications - a survey," *Modern Mechanical Engineering*, vol. 2, no. 3, pp. 57–64, 2012. DOI: 10.4236/mme.2012.23008
- [68] C. S. HSU, "On the application of elliptic functions in non-linear forced oscillations," *Quarterly of Applied Mathematics*, vol. 17, no. 4, pp. 393–407, 1960, ISSN: 0033569X, 15524485. Accessed: Oct. 25, 2025. [Online]. Available: <http://www.jstor.org/stable/43634717>
- [69] Z. Rakaric, I. Kovacic, and M. Cartmell, "On the design of external excitations in order to make nonlinear oscillators respond as free oscillators of the same or different type," *International Journal of Non-Linear Mechanics*, vol. 94, pp. 323–333, 2017, A Conspectus of Nonlinear Mechanics: A Tribute to the Oeuvres of Professors G. Rega and F. Vestroni, ISSN: 0020-7462. DOI: <https://doi.org/10.1016/j.ijnonlinmec.2016.06.012> [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0020746216300671>
- [70] J. Drgoňa et al., "All you need to know about model predictive control for buildings," *Annual Reviews in Control*, vol. 50, pp. 190–232, 2020, ISSN: 1367-5788. DOI: <https://doi.org/10.1016/j.arcontrol.2020.09.001>

Bibliography

- [71] M. Schwenzer, M. Ay, T. Bergs, and D. Abel, “Review on model predictive control: An engineering perspective,” *The International Journal of Advanced Manufacturing Technology*, vol. 117, pp. 1327–1349, 2021. DOI: 10.1007/s00170-021-07682-3
- [72] Y. Shi and K. Zhang, “Advanced model predictive control framework for autonomous intelligent mechatronic systems: A tutorial overview and perspectives,” *Annual Reviews in Control*, vol. 52, pp. 170–196, 2021, ISSN: 1367-5788. DOI: <https://doi.org/10.1016/j.arcontrol.2021.10.008>
- [73] S. Vazquez, J. Rodríguez, M. Rivera, L. Franquelo, and M. Norambuena, “Model predictive control for power converters and drives: Advances and trends,” *IEEE Transactions on Industrial Electronics*, vol. 64, pp. 935–947, 2017. DOI: 10.1109/TIE.2016.2625238
- [74] S. Vazquez et al., “Model predictive control: A review of its applications in power electronics,” *IEEE Industrial Electronics Magazine*, vol. 8, pp. 16–31, 2014. DOI: 10.1109/MIE.2013.2290138
- [75] P. Karamanakos, E. Liegmann, T. Geyer, and R. Kennel, “Model predictive control of power electronic systems: Methods, results, and challenges,” *IEEE Open Journal of Industry Applications*, vol. 1, pp. 95–114, 2020. DOI: 10.1109/OJIA.2020.3020184
- [76] A. Castelletti et al., “Model predictive control of water resources systems: A review and research agenda,” *Annual Reviews in Control*, vol. 55, pp. 442–465, 2023, ISSN: 1367-5788. DOI: <https://doi.org/10.1016/j.arcontrol.2023.03.013>
- [77] Y. Huang, H. Wang, A. Khajepour, H. He, and J. Ji, “Model predictive control power management strategies for hevs: A review,” *Journal of Power Sources*, vol. 341, pp. 91–106, 2017, ISSN: 0378-7753. DOI: <https://doi.org/10.1016/j.jpowsour.2016.11.106>

Bibliography

- [78] S. Yu, M. Hirche, Y. Huang, H. Chen, and F. Allgöwer, “Model predictive control for autonomous ground vehicles: A review,” *Autonomous Intelligent Systems*, vol. 1, 2021. DOI: 10.1007/s43684-021-00005-z
- [79] A. Abdelrauf, W. Saad, A. Hebala, and M. Galea, “Model predictive control based pid controller for pmsm for propulsion systems,” *2018 IEEE International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles & International Transportation Electrification Conference (ESARS-ITEC)*, pp. 1–7, 2018. DOI: 10.1109/ESARS-ITEC.2018.8607585
- [80] M. Homaeinezhad and F. Fotoohinia, “Robust nonlinear model predictive sliding mode control algorithm for saturated uncertain multivariable mechanical systems,” *Journal of Vibration and Control*, vol. 29, pp. 1524–1538, 2022. DOI: 10.1177/10775463211065883
- [81] S. Bayat and J. T. Allison, “Ss-mpc: A user-friendly software based on single shooting optimization to solve model predictive control problems,” *Software Impacts*, vol. 17, p. 100566, 2023, ISSN: 2665-9638. DOI: <https://doi.org/10.1016/j.simpa.2023.100566>
- [82] P. Christofides and N. El-Farra, “Hybrid predictive control of constrained linear systems,” in *Control of Nonlinear and Hybrid Process Systems* (Lecture Notes in Control and Information Sciences), 1st ed., Lecture Notes in Control and Information Sciences. Springer Berlin, Heidelberg, 2005, ch. 5, pp. 127–177. DOI: 10.1007/b105110
- [83] A. Afram and F. Janabi-Sharifi, “Theory and applications of hvac control systems – a review of model predictive control (mpc),” *Building and Environment*, vol. 72, pp. 343–355, 2014. DOI: 10.1016/J.BUILDENV.2013.11.016
- [84] J. Berberich, J. Köhler, M. Müller, and F. Allgöwer, “Data-driven model predictive control with stability and robustness guarantees,” *IEEE Transactions on Automatic Control*, vol. 66, pp. 1702–1717, 2021. DOI: 10.1109/TAC.2020.3000182

Bibliography

- [85] C. Franklin, *Robotic industries association: Robot standards*, Conference Paper, 19 May 2019 2019. [Online]. Available: https://www.ieee-ras.org/images/Standards_strategy_Meeting_19_May_2019/12-Franklin--RIA-IEEE_2019-05-19_v2.pdf
- [86] E. Hedberg, *Control, Models and Industrial Manipulators*. Linköping, SWEDEN: Linkopings Universitet, 2020, ISBN: 9789179297404.
- [87] I. Zamalloa et al., “Dissecting robotics - historical overview and future perspectives,” 2017. DOI: 10.48550/arXiv.1704.08617
- [88] Robots.com, *Industrial robot history*, Web Page, n.d. [Online]. Available: <https://www.robots.com/articles/industrial-robot-history>
- [89] O. O. Ropo, O. F. Adekunle, O. A. Oluwaseun, and P. Babatope Olabanji, “Design of a pick and place serial manipulator,” *IOP Conference Series: Materials Science and Engineering*, vol. 413, no. 1, p. 012056, 2018, ISSN: 1757-899X 1757-8981. DOI: 10.1088/1757-899X/413/1/012056
- [90] R. Zhao, X. Li, and J. Chen, “Eagle-inspired manipulator with adaptive grasping and collapsible mechanism and modular dof for uav operations,” *Computers and Electronics in Agriculture*, vol. 215, p. 108344, 2023, ISSN: 0168-1699. DOI: <https://doi.org/10.1016/j.compag.2023.108344>
- [91] S. H. Saheb, G. S. Babu, and N. V. S. Raju, “Relative kinematic analysis of serial and parallel manipulators,” *IOP Conference Series: Materials Science and Engineering*, vol. 455, no. 1, p. 012040, 2018, ISSN: 1757-899X 1757-8981. DOI: 10.1088/1757-899X/455/1/012040
- [92] Z. Pandilov and V. Dukovski, “Comparison of the characteristics between serial and parallel robots,” 2014.
- [93] F. Pierrot, V. Nabat, O. Company, S. Krut, and P. Poignet, “Optimal design of a 4-dof parallel manipulator: From academia to industry,” *IEEE Transactions on Robotics*, vol. 25, no. 2, pp. 213–224, 2009, ISSN: 1941-0468. DOI: 10.1109/TR0.2008.2011412

Bibliography

- [94] Z. Gao and D. Zhang, "Performance analysis, mapping, and multiobjective optimization of a hybrid robotic machine tool," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 1, pp. 423–433, 2015, ISSN: 1557-9948. DOI: 10.1109/TIE.2014.2327008
- [95] T. K. Tanev, "Kinematics of a hybrid (parallel–serial) robot manipulator," *Mechanism and Machine Theory*, vol. 35, no. 9, pp. 1183–1196, 2000, ISSN: 0094-114X. DOI: [https://doi.org/10.1016/S0094-114X\(99\)00073-7](https://doi.org/10.1016/S0094-114X(99)00073-7)
- [96] F. Caccavale, "Computed-torque control," *Encyclopedia of Robotics*, 2020. DOI: 10.1007/978-3-642-41610-1_91-1
- [97] C. Feng, Z. Chen, W. Jin, and W. Guo, "Motion control and simulation analysis of a manipulator based on computed torque control method," pp. 10–22, 2023. DOI: 10.1007/978-981-99-6483-3_2
- [98] J. Shah and S. Rattan, "Dynamic analysis of two link robot manipulator for control design using pid computed torque control," vol. 5, pp. 277–283, 2016. DOI: 10.11591/IJRA.V5I4.PP277-283
- [99] M. Van, M. Mavrovouniotis, and S. Ge, "An adaptive backstepping nonsingular fast terminal sliding mode control for robust fault tolerant control of robot manipulators," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, pp. 1448–1458, 2019. DOI: 10.1109/TSMC.2017.2782246
- [100] N. Adhikary and C. Mahanta, "Sliding mode control of position commanded robot manipulators," *Control Engineering Practice*, 2018. DOI: 10.1016/J.CONENGPRAC.2018.09.011
- [101] D. Zhang and B. Wei, "A review on model reference adaptive control of robotic manipulators," *Annu. Rev. Control.*, vol. 43, pp. 188–198, 2017. DOI: 10.1016/j.arcontrol.2017.02.002
- [102] K. Bingi, R. Prusty, and A. Singh, "A review on fractional-order modelling and control of robotic manipulators," *Fractal and Fractional*, 2023. DOI: 10.3390/fractalfract7010077

Bibliography

- [103] P. Chotikunnan, R. Chotikunnan, A. Nirapai, A. Wongkamhang, P. Imura, and M. Sangworasil, “Optimizing membership function tuning for fuzzy control of robotic manipulators using pid-driven data techniques,” *Journal of Robotics and Control (JRC)*, 2023. DOI: 10.18196/jrc.v4i2.18108
- [104] C. Urrea, J. Kern, and J. Alvarado, “Design and evaluation of a new fuzzy control algorithm applied to a manipulator robot,” *Applied Sciences*, 2020. DOI: 10.3390/app10217482
- [105] S. Li, Y. Zhang, and L. Jin, “Kinematic control of redundant manipulators using neural networks,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, pp. 2243–2254, 2017. DOI: 10.1109/TNNLS.2016.2574363
- [106] H. Gao, W. He, C. Zhou, and C. Sun, “Neural network control of a two-link flexible robotic manipulator using assumed mode method,” *IEEE Transactions on Industrial Informatics*, vol. 15, pp. 755–765, 2019. DOI: 10.1109/TII.2018.2818120
- [107] L. Jin, S. Li, J. Yu, and J. He, “Robot manipulator control using neural networks: A survey,” *Neurocomputing*, vol. 285, pp. 23–34, 2018, ISSN: 0925-2312. DOI: <https://doi.org/10.1016/j.neucom.2018.01.002>
- [108] Y.-T. Liu et al., “Self-tuning control of manipulator positioning based on fuzzy pid and pso algorithm,” *Frontiers in Bioengineering and Biotechnology*, vol. 9, 2022. DOI: 10.3389/fbioe.2021.817723
- [109] V. Nguyen and X. C. Bui, “Hybrid vibration control algorithm of a flexible manipulator system,” *Robotics*, vol. 12, p. 73, 2023. DOI: 10.3390/robotics12030073
- [110] S. Paul, A. Arunachalam, D. Khodadad, H. Andreasson, and O. Rubanenko, “Fuzzy tuned pid controller for envisioned agricultural manipulator,” *International Journal of Automation and Computing*, vol. 18, pp. 568–580, 2021. DOI: 10.1007/s11633-021-1280-5
- [111] M. H. Barhaghtalab, M. A. Sepestanaki, S. Mobayen, A. Jalilvand, A. Fekih, and V. Meigoli, “Design of an adaptive fuzzy-neural inference system-based control

Bibliography

- approach for robotic manipulators,” *Appl. Soft Comput.*, vol. 149, p. 110970, 2023. DOI: 10.1016/j.asoc.2023.110970
- [112] M. Alhajeri and M. Soroush, “Tuning guidelines for model-predictive control,” *Industrial & Engineering Chemistry Research*, vol. 59, pp. 4177–4191, 2020. DOI: 10.1021/acs.iecr.9b05931

Bibliography