Modelling and control of a robotic manipulator subject to base disturbances

Cyprian M. Wronka

Submitted for the degree of Doctor of Philosophy

Heriot-Watt University

Department of Electrical, Electronic
and Computer Engineering
School of Engineering
and Physical Sciences

May 2010

Abstract

This thesis presents the modelling and control of a high gear ratio robotic manipulator mounted on a heavier moving base which is subject to base disturbances. The manipulator motion is assumed not to affect the base motion. The problem of a robotic manipulator on a non-inertial base can be applied to operation on sea vessels or all-terrain vehicles, where the base motion is unknown and cannot be used as a feed-forward signal to the model.
A dynamic model is derived for the PA10-6CE manipulator with the assumption of a fixed base and the model terms are analysed numerically when comparing the simulation and experimental results. Based on the obtained results a set of model based controllers is compared to a basic proportional and derivative type controller to evaluate the trajectory tracking gains and trade-offs.
The dynamic model is extended to the case of a manipulator on a moving base and numerical comparisons of simulation and experimental results are used to verify the model validity and the significance of the various model terms. From the results of this study a set of model based controllers is obtained. A novel adaptive scheme is then proposed for compensation of an unknown and varying gravity acceleration vector acting on the manipulator base. Controllers based on using an additional sensor output are compared with static and adaptive gravity controllers and the latter proved to be superior in terms of trajectory tracking performance.

Acknowledgements

I am grateful to my wife Agata and my parents for their continuous support and encouragement. I would like to thank my supervisor Dr Matthew Dunnigan for his support and cooperation. I would like to show my gratitude to Prof. Krzysztof Tchoń, Dr Krzysztof Arent and Dr Robert Muszyński (from the Wrocław University of Technology) for their invaluable comments.

Contents

List of notations

${}^{i}\mathbf{A}_{j}$	the homogenous coordinate transformation matrix from i-th joint frame to j-th joint frame
$\mathbf{A}_{p}$	the homogenous coordinate transformation matrix from the fixed platform centre to the manipulator base
$\dot{\mathbf{A}}_{p}$	the time differential of the $\mathbf{A}_{p}$
$\ddot{\mathbf{A}}_{p}$	the second time differential of the $\mathbf{A}_{p}$
$\mathbf{C}$	the Coriolis and centrifugal term
$\mathbf{e}$	vector of joint angle errors
φ	sliding controller boundary layer
g	the gravity acceleration constant
$\mathbf{g}$	the gravity acceleration vector
$\mathbf{J}_{i}$	link inertia matrix
$\mathbf{K}$	the kinetic energy
$\mathcal{K}$	the kinematics
$\mathbf{K}$	sliding controller gain
$\mathbf{K}_{P}, \mathbf{K}_{D}, \mathbf{K}_{I}$	proportional, derivative abd integral feedback control gains.
$\mathbf{\Lambda}$	sliding controller bandwidth
m_{i}	the mass of the i-th link
$\mathbf{PF}$	the manipulator dynamics equation torque term originating from the platform motion, not related to joint angle velocities
$\mathbf{PV}$	the manipulator dynamics equation torque term originating from the platform motion and related to the joint angle velocities

Q_d the matrix used to replace the symbolic differentation of ${}^{i}\mathbf{A}_j$ with algebraic matrix multiplication

$\mathbf{Q}$ the inertia Matrix

$\mathbf{g}$ the gravity torque vector

$\mathbf{q}$ the state (joint angles)

$\mathbf{q}_i$ i-th component of the state vector

$\mathbf{q}_d$ desired joint configuration

$\dot{q}_i$ time derivative of the i-th component of the state vector (joint velocity)

$\ddot{q}_i$ second time derivative of the i-th component of the state vector (joint acceleration)

$\mathbb{R}$ the real numbers

$\mathbf{R}_p$ the rotation component of the $\mathbf{A}_p$

${}^{i}\mathbf{r}_i$ the centre of mass of the *i*-th joint in its coordinate frame

${}^{0}\mathbf{r}_i$ the centre of mass of the *i*-th joint in the base coordinate frame

R_i location of *i*-th link mass point expressed in the *i*-th coordinate system (equevalent to ${}^{i}\mathbf{r}_i$)

$\mathbb{SO}$ the special rotations algebraic group

$\mathbb{SE}$ the special Euclidean algebraic group

τ the torque vector

$\mathbf{u}$ the input/control (torque)

$\mathbf{U}_{ij}$ derivative of ${}^{0}\mathbf{A}_i$ over j

$\mathbf{U}_{ijk}$ second derivative of ${}^{0}\mathbf{A}_i$ over j and k

Chapter 1

Introduction

1.1 Motivation

The subject of manipulator control has been widely studied over the past three decades and there is a significant volume of theoretical and experimental results available. An introduction of a new product to the robotic manipulators market always triggers a number of researchers to evaluate the available models/controllers with the new device. The Intelligent Robotic Systems Laboratory at Heriot-Watt university obtained one of the first PA10-6CE robotic manipulators and it has become an important area of study due to the lack of models/experimental results for this particular device. It allows easy torque control implementation due to the open controller architecture and presents new challenges in the modelling field due to its high ratio harmonic drive transmission and the significant friction contribution in the dynamic model.
With the increasing computational capacity of computers, it has also become a simple task to derive and evaluate complex non-linear manipulator models and model-based controllers. The models become even more complex when the manipulator is mounted on a moving base and the dynamics of the manipulator can be influenced by this motion. There are multiple applications of this scenario:

- manipulator mounted on a vessel at sea (disturbed by the waves, [1.1], [1.2], performing tasks on this vessel, such as operation of high loads, or high speed tasks such as automated ammunition loading of a cannon; the manipulator could also be used to launch AUVs (autonomous underwater vehicles) or move loads to another vessel or an oil platform.

- manipulator mounted on a moving vehicle (such as a Mars lander, an all-terrain car or a tank, [1.3]) performing automated or remote-controlled tasks.

In both these cases two issues arise, the dynamics of the manipulator (and the load) are disturbed by the base motion and the position of the end-effector in the global (terrain/sea attached) coordinate system is disturbed. The dynamics disturbance problem can be applied directly to the task of a manipulator operating objects on a common (moving) base.

Figure 1.1 presents the lab set-up of the PA10-6CE manipulator mounted on the CueSim motion platform.

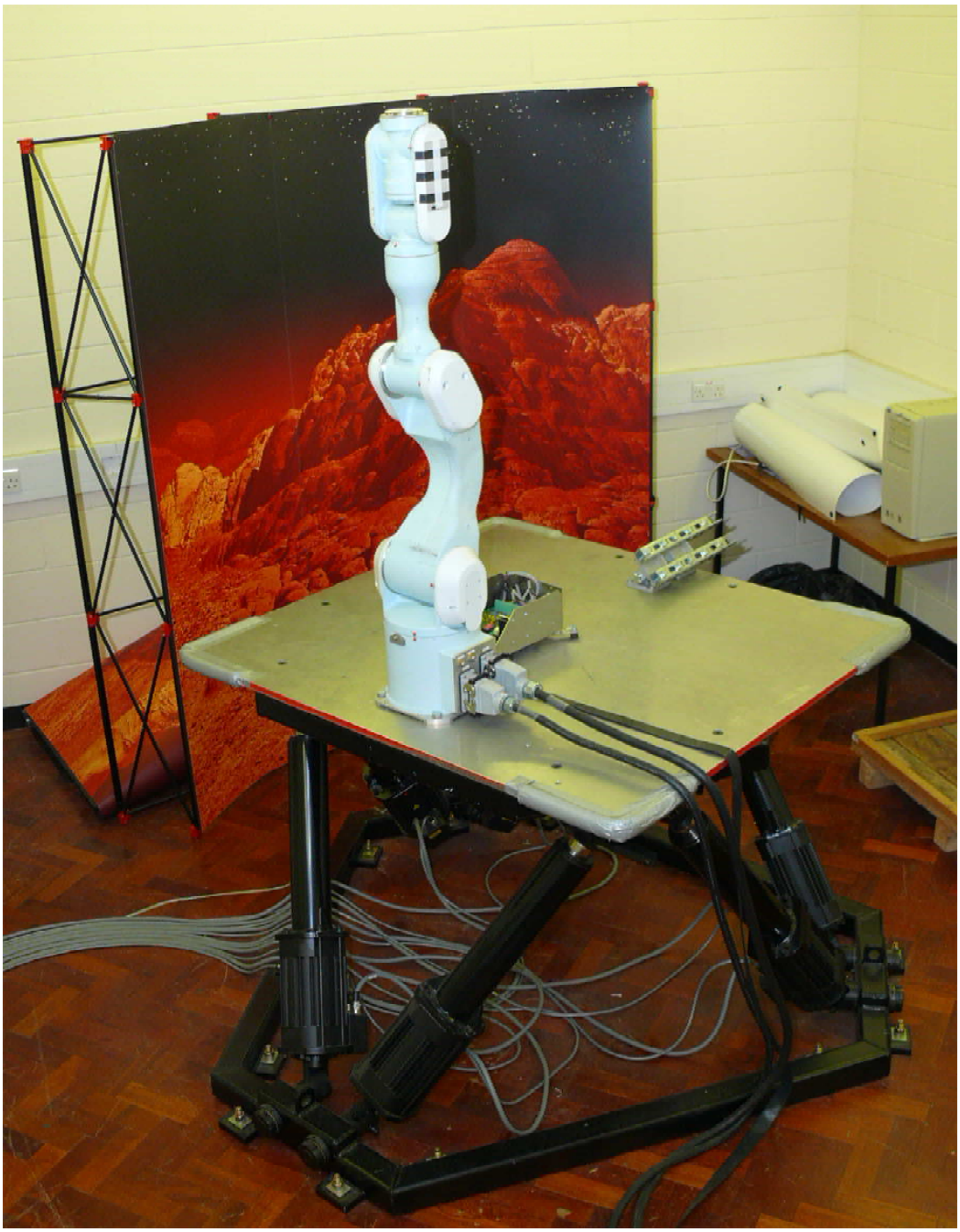


Figure 1.1: Intelligent Robotic Systems Laboratory PA10 and motion platform set-up.

1.2 Aims and objectives of thesis

This thesis addresses the manipulator dynamics when it is attached to a non-fixed base by analysing the dynamics of a manipulator on a fixed base and later extending the model to a moving base. Several control schemes employing the model and external measurements to improve the joint angle tracking precision are thoroughly assessed. The following can

be listed as research objectives:

- Derivation of the dynamic model for the PA10-6CE robotic manipulator (with fixed and moving base). This is representative of a broad range of geared manipulators with high transmission ratio and significant joint friction dynamics.
- Analysis of the terms (inertia, Coriolis and centrifugal, gravity, friction) within the dynamic model based on quantitative comparisons of model simulation and actual manipulator experiments.
- Experimental comparison of non-model based and model based control in terms of angle tracking errors and torque energy dissipation due to high frequency control signals for the fixed and moving base cases.
- Based on knowledge obtained from analysis of the manipulator model and the controllers' performance a controller is designed which compensates for base motion without any additional sensor input.

Possibilities for extending the presented solution to the global position disturbance issue are mentioned in the *future work* suggestions.

1.3 Thesis organisation

The Thesis is organised in nine chapters:

1. **Introduction** - briefly explains the motivation and aims of the research presented and outlines the thesis structure.
2. **Background and literature survey** - presents a literature survey of the topics presented in this research. Modelling techniques for friction, harmonic drives and robotic manipulators are introduced with examples of implementation and discussion of obtained results. Control strategies for robotic manipulators are reviewed featuring non-model based, model based methods and adaptive schemes. Finally, examples of research with manipulators on a moving base are presented and a relevant classification of this research is performed.
3. **Kinematics and dynamics modelling** - presents details about the kinematics and dynamic modelling of a robotic manipulator. The Denavit-Hartenberg method for forward kinematics and the Euler-Lagrange approach to determine the explicit manipulator dynamics are presented. Finally, a review of friction models and their shortfalls and advantages is presented.
4. **Model derivation and estimation of the PA10-6CE manipulator parameters** - presents details about the dynamic model derivation and initial model evaluation results. The derivation of the gravity term from kinetic energy is presented.

The calculation of geometric link models with a density distribution matching the manufacturer's data on mass centres is used to derive the inertial and Coriolis and Centrifugal terms. Experimental estimation of motor inertias is performed. A non-parametric friction model is estimated based on constant velocity movements of each joint.

5. **Joint angle control of the PA10-6CE manipulator** - presents details on joint torque control algorithms and compares their performance based on angle tracking errors and torque energy dissipation due to high frequency signals. A comparison of the PA10-6CE dynamic model (with friction) is shown by means of joint torque differences between simulation and experiment for the manipulator commanded to follow a given trajectory. Furthermore, a comparison of various manipulator controllers is shown based on experimental results obtained with the PA10-6CE manipulator.

6. **Description of base disturbance** - the experimental set-up used to model and measure the base disturbance is presented along with details on trajectories used in the simulations and experiments. Details on available sensor measurements (and their processing) and platform motion commands are presented.

7. **Modelling a robotic manipulator on a moving platform** - presents a derivation of a mathematical model of a robotic manipulator on a moving base for a general case, with the assumption that the base motion is not affected by the manipulator dynamics and is considered as a parameter (positions, velocities and accelerations) of the model. Similarities and differences from the the fixed base model are highlighted. Simulation and experimental results for the PA10-6CE on a platform moving in 2-DOF (roll and pitch) are compared quantitatively by means of joint torque differences to evaluate the significance of model terms.

8. **Control of the PA10-6CE robotic manipulator on a moving base** - presents details on extension of the fixed base controllers to compensate for base motion and presents adaptive model based control schemes for a robotic manipulator with a comparison of fixed-base, sensor based and adaptive schemes. The fixed-base controllers are extended to read the gravity vector from either accelerometer sensors mounted on the moving base or from roll+pitch inclinometers. Adaptive model based control is employed to compensate for gravity changes induced by base motion. Fixed gravity model based controllers are compared quantitatively with sensor based and adaptive model based controllers by means of trajectory tracking errors and torque energy dissipation due to high frequency signals.

9. **Conclusions** - presents summary remarks on the research, lists the author's contributions and suggestions for future work.

Chapter 2

Background and literature survey

2.1 Introduction

The field of robot control can be considered to be an amalgamation of multiple engineering and mathematical disciplines, such as mechanics, electronics and cybernetics[1] alowing to design, model and implement a particular system. It requires a combination of knowledge of the mechanical and electrical construction of a robot with the ability to model its behaviour. It further utilises a number of mathematical tools to design a device able to command a robot to perform desired tasks. Computer programming skills are also required to accomplish these tasks. Therefore, this inter-disciplinary problem has a broad knowledge requirement.
This chapter presents the background knowledge for this thesis together with recent and past attempts at similar problems.
Considering the construction of the robot, two classes can be distinguished:

- Serial manipulators - consisting of a single chain of links interconnected with joints, to achieve higher dexterity at the cost of lower accepted loads.
- Parallel robots - consisting of multiple chains of links interconnected with joints, to handle high loads compromising the dexterity.

This classification is slightly limited, since in practice stacked parallel robots forming a serial manipulator, or multiple serial manipulators operating a single object forming a parallel robot can be found. For a general introduction to manipulator modelling and control, the classical references are Craig [2.15] and Spong et al. [2.63]. Parallel robots are described in [2.48] and detailed work on two of the most popular types of platform configurations are presented in [2.24] and [2.64].
The following areas of modelling and control of robotic manipulators and mechanical systems are considered in more detail (and constitute the structure of this chapter):

- Modelling of manipulators and their mechanical elements.

[1] Cybernetics is considered by the author to the discipline researching the notion of feedback in systems.

- Position control of a robotic manipulators with fixed base.
- Modelling and control of a robotic manipulators with non-fixed base.

2.2 Modelling of manipulators

The following areas of modelling of mechanical systems are considered as an overview of the background material used in this research:

- Friction and harmonic drive models.
- Manipulator models (based on the mechanical construction).

2.2.1 Friction and harmonic drive modelling

The two topics are combined, since the PA10 manipulator used in this research has harmonic drive transmission on all joints and the friction present in the dynamic model can be attributed to this feature.

2.2.1.1 Friction models

The friction models are presented in detail in section 3.4.2. A very broad and detailed overview of friction effects and models can be found in [2.4] and [2.5]. One of the most popular recent friction models (the LuGre model, not employed in this research) is presented in [2.10]. Interesting remarks about simulation of systems with friction (both classical 'static' and the dynamic LuGre model) have been presented in [2.17].

2.2.1.2 Modelling harmonic drives

The harmonic drive is a high ratio transmission type and consists of three elements:

- an elliptical wave generator,
- a flex spline,
- a circular spline.

A schematic of a harmonic drive is shown in figure 2.1. The wave generator rotates inside the flex spline and causes its deformation. The difference of one tooth between the flex spline and the circular spline leads to the flex spline rotating by one tooth with respect to the circular spline every half rotation of the wave generator. The harmonic drives are characterised by[2]:

- high speed reduction ratio,

[2]source: www.hds.co.jp

Figure 2.1: Schematic of a harmonic drive (source: Harmonic Drive Systems website).

- no backlash,
- high precision,
- simple assembly,
- light weight,
- high torque capacity,
- high efficiency,
- quiet operation.

However, there is a tendency for a 'wind-up' effect (torsional spring rate) and a complex friction behaviour. This originates from the flexibility of the cup-shaped flexspline element - when torque is applied to the wave generator and the circular spline is fixed (e.g. due to stiction), the energy is temporarily stored in the form of material deformation, and later released creating a torsional spring effect.

Luh, Fisher and Paul present a controller for harmonic drives with strain gauges mounted on the flex spline in [2.44]. The frequency response of the controller depending on gains is considered and it is concluded that turning the gains up above a certain limit causes disturbances in the transmission and poorer trajectory tracking performance. Tuttle and Seering look at the dynamics of the harmonic drive transmission in [2.72], [2.73] and make the following observations:

- the kinematic uncertainties in the transmission cause resonant behaviour and lead to loss of energy and vibrations,
- the non-uniform friction leads to velocity fluctuation at constant torques,
- the friction dominates the drive dynamics and depends both on velocity and position,
- there are hysteresis and compliance properties in the transmission dynamics.

Another approach is taken by Taghirad and Belanger in [2.65] and [2.66]. The authors install strain gauges on the flex spline and discover that the gear dynamics are dominated by the tooth friction and the wave generator bearing friction (due to it acting on the high velocity/low torque port of the transmission). A Kalman filter is suggested to cancel the torque disturbance in the gear. Kircanski [2.34] looks at the modelling of harmonic drives in robotic manipulators for both position and force control setups.

2.2.2 Manipulator models

In order to model a manipulator, its mechanical properties require analysis. In robotics, two types of models are distinguished:

- kinematic models - analysing the dimensions of a manipulator and the instantaneous relation between the position of any point associated with the manipulator and the joint configuration. The following coordinate sets are typically used in this framework:
  - joint configuration (also called joint coordinates or joint space),
  - base coordinates (also called global or task space),
  - end effector coordinates.

  The models can also be classified as:
  - forward kinematics - takes the joint configuration as input and provides the position of a given point (rigidly associated with the manipulator) in the global coordinates,
  - inverse kinematics - transforms the position of a point to the joint configuration.

The latter is usually a more complex task due to solution non-uniqueness.

- dynamic models - considering the masses and inertias of rigid bodies constituting a manipulator and inferring the system response to a set of input excitations. The dynamic models require the forward kinematics to relate motion of all joints. There are two typical approaches to manipulator dynamic modelling:
  - explicit - based on Euler-Lagrange or Hamilton equations,
  - implicit - based on recursive Newton-Euler equations.

Details of Euler-Lagrange manipulator modelling are presented in chapter 3. The foundations of forward kinematic models have been presented by Devavit and Hartenberg in [2.16]. A general overview of manipulator modelling can be found in [2.33] or [2.36].
The manipulators can further be classified in two groups based on the type of transmission

- direct drive manipulators - without any gearing/transmission between the actuator and the joint
- geared manipulators - with additional mechanical elements transmitting the actuator force/torque.

It can immediately be identified that the second group may have additional elements in the dynamic model in comparison to direct drive systems. The manipulator employed in the simulation and experimental research in the this project has a power transmission on all joints and therefore the review focuses on geared manipulators.

2.2.2.1 PUMA 560 models

One of the most popular and widely used manipulators in research labs (including the Intelligent Robotic Systems Laboratory at Heriot-Watt University) is the PUMA. There have been several versions of this manipulator originating from different manufacturers. This popularity can be attributed to the ease of replacing the original controller with one delivering direct access to the motor torque command. The mechanical modelling results obtained for the PUMA manipulator can be viewed as a broad base to analyse the dynamics of robotic manipulators. A picture of the PUMA 560 manipulator is shown in figure 2.2.
In the early 80s the computing technology did not allow for automated symbolic derivations, and therefore many of the first approaches have been hand-derived. Tarn and Bejczy [2.68], [2.69] derived by hand the equations of motion for the first three joints of the manipulator and computed by hand the inertias and mass centres of all the links based on technical drawings, material density tables, and geometric models. Armstrong and Khatib [2.3] disassembled the manipulator to measure and weigh its elements. They use a mechanical setup to estimate the principal link inertias from fundamental mode oscillations. A feedback controller is employed to estimate motor inertias and the model is further

Figure 2.2: PUMA 560 manipulator (source: www.alaska.edu).

simplified by means of removing insignificant incorporating elements from the dynamic equation to reduce the computational load. (It needs to be noted that these authors were using a 100kFLOP computer and at present one can easily have access to GFLOPS of computing power). Neuman and Murray [2.53] created a computer program to automate the derivation of manipulator dynamic equations (with an assumption of diagonal link inertias) and present a full model for the PUMA manipulator. A review of multiple models and comparison of obtained results and computational requirements is presented by Corke and Armstrong-Helouvry in [2.13].

2.2.2.2 PA10 models

The PA10 general purpose manipulators from Mitsubishi Heavy Industries have recently (in the past 10 years) started gaining popularity due to an open controller architecture allowing for easy command of joint torques. This is a new manipulator design creating space for research in dynamic model derivation and identification. There are two versions of this manipulator, a 6-DOF and a 7-DOF, with a different shape of the shoulder link. The manufacturer provides a schematic with mass centres for each link, which can be used for partial model derivation.

Jamisola et al. presented the first set of dynamic parameters for the 7-DOF version in [2.29], [2.28]. An identification method called 'dominant inertia' using linear friction compensation and oscillatory control of each joint was used to obtain the lumped inertias. A model for static, Coulomb and Viscous friction was also obtained. Olsen and Petersen present a novel Maximum Likelihood mathematical method for dynamic parameter identification and present a set of obtained parameters for the first two joints of the 7-DOF PA10 in [2.54]. Another interesting approach is presented by Kennedy and Desai in [2.31], [2.32] for the 7-DOF version. The authors assume low velocity motion and ignore the inertial parameters of the joints. The focus is on the harmonic drive transmission and an in-depth analysis of velocity and position dependant friction is performed together with the model of transmission compliance. The gravity element is also considered. Hamner measured the positioning precision of the PA10 manipulator and verified the harmonic drive transmission compliance against the technical parameters of the gearing in [2.25]. The author of this work has presented the inertias and nonparametric friction curves for the 6-DOF version in [2.77]. Sekimoto and Arimoto employ a model of the first six joints of the 7-DOF PA10 and present the joint and motor inertia parameters and the identified friction parameters. Bompos, Artemiadis et al. identify the dynamics of the 7-DOF PA10 with Stribeck friction and transmission compliance model in [2.7] and employ a feedback linearisation controller based on the model.
A different approach is taken in [2.41], where the PA10 manipulator is considered to have rigid links and flexible joints. However, the full flexible joint 4-th order dynamics equations have been simplified to a typical 2-nd order system augmented by the motor angles to improve the regressor matrix robustness (the flexibility is considered small, which created matrix conditioning problems). An identification is performed for each joint in two steps, first the physical parameters and viscous friction; the flexibility is tackled in a separate step. Since the joint angle cannot be measured (only the motor angle can) an external position measurement system is used and the required quantities are derived from these measurements.

2.3 Manipulator control

The area of manipulator control has been widely studied over the past 30 years. There is ample literature and results covering various strategies and implementations of controllers. Depending on the input type, the following can be distinguished:

- position control - the desired position is used as a reference,
- force control - the desired force (typically expressed in the end-effector coordinate frame) is used as a reference and compared against a force sensor measurement.

It can be further differentiated by the coordinate system in which the desired position/trajectory is defined:

- joint control,
- resolved control (in the task or actuator space).

Depending on the type of command, the following groups can be considered:

- point-to-point control - with only discrete points given asynchronously,
- continuous path tracking - with desired position for each control cycle.

Considering the mathematical approach taken to control, it can be split into

- model/stability based (also called *hard-control*),
- heuristic method based (e.g. neural networks, genetic algorithm – also called *soft-control*).

A general overview of robot control techniques is presented in [2.59] and [2.71]. Details of the force control approach can be found in [2.9]. An overview of neural networks in robotics is presented in [2.79]. This is not discussed further in the thesis as only hard-control methods are investigated. The control algorithms used in this thesis are presented in detail in sections 5.1 (non model based and non-adaptive model based) and 8.2.1 (adaptive model based).

2.3.1 Robust control

In the case of robust control the manipulator is generally treated as a black box without any knowledge about its dynamics, only some assumptions about the order of the system can be made based on response types. A simple PD controller combined with torque sensor measurements on the drive shaft is shown in [2.26] to compensate for all unknown forces. Qu and Dorsey present the proof of ultimate uniform boundedness on the tracking error of a PD controller in [2.57]. A linear single-input-single-output (SISO) approach to joint control is presented in [2.23], where an additional filter is added to the control loop to compensate for unmodelled disturbances (actual nonlinearities). It is also shown that in the absence of friction and gravity the step input error converges to zero. An interesting adaptive gain PD controller is presented in [2.46], however the gain can only increase, which is not desirable for systems with harmonic drive transmission. An interesting study of torque disturbances in control of coupled conveyor belts is presented in [2.27]. The authors compare different tuning strategies for the PID controllers and add additional linear filters on feed-forward and feed-back signals to improve bandwidth. The author of this work presented a comparison of robust controllers for the PA10-6CE manipulator in [2.78]. An interesting implementation of adaptive gravity control without any model structure, based only on tracking errors is presented in [2.21] allowing algorithm portability to any robotic system. However, limits on the feedback gains are not considered.

2.3.2 Model based and adaptive control

Model based control of robotic manipulators utilises the knowledge about the mathematical model of the system, to reduce the control gains and the wear and tear of the actuators/transmission. The parameters of the system are not always known to the control designer, therefore different techniques are employed to work around this issue. The parameters can be calculated based on some geometric/design knowledge or can be identified using statistical methods. They can also be assumed unknown and an adaptive scheme employed to identify them on-line, based on the system and desired state.

Extending the basic controllers presented in [2.63] and [2.15], Kreutz presents an exact linearisation for a robotic manipulator performed in the task space in [2.37], where both the dynamics and kinematics are used to obtain a linear model in the Cartesian coordinates. Tarn et al. present an extension of the standard second order model adding the motor dynamics in [2.70]. The manipulator is controlled in the task space and a comparison shows the reduction in tracking error after considering the motor dynamics (which creates a hybrid slow (mechanical) and fast (electrical) model).

The foundations for multivariate adaptive control for linear systems are laid in [2.8] with stability proofs. However, more research was required to apply this kind of control to a robotic manipulator with nonlinear dynamic model. Craig et al. show the first adaptive computed torque controller for a robotic manipulator in [2.14] with some strong assumptions of a non-singular inertia matrix (with adapted parameters, which may not represent a valid physical system) and a requirement for joint acceleration measurements. The convergence of control error is shown even with a condition of boundedness for the unknown inertial parameters. This in turn requires prior knowledge of suitable bounds. Middleton and Goodwin take a slightly different approach to the dynamic equations and by adding a filter on the torques avoid the requirement of joint acceleration measurements in [2.49, 2.50]. The authors also present a general proof for the linear dependence of the manipulator dynamics equation terms on the unknown parameters. Sadegh and Horovitz go one step further and deliver an adaptation scheme not requiring the inversion of the inertia matrix or the measurements of joint accelerations in [2.58]. The adaptation rule is based on velocity tracking error and the authors mention that the model can be calculated either along the measured joint positions or along the desired trajectory (which at that point was a significant simplification allowing off-line calculation of the feed-forward terms in the model).

Slotine and Li approach the control of a robotic manipulator in [2.61, 2.60] by shaping the energy function and utilise the passivity properties of the dynamic equations to derive an adaptive controller feeding back the model calculated using the measured positions and velocities. The new adaptation rule utilises both velocity and and position tracking errors. Notes are made about the requirement of persistent excitation properties of the trajectory, to provide enough variability to the adaptation regressor matrix. The adaptation is further extended by adding an estimate of the unknown parameter prediction error and an adap-

tation algorithm in the task space (including hybrid position/force control) is presented. It is also stated that the adaptation gains need to be adjusted if the unknown parameters are not constant. Convergence is shown for all presented types of adaptation.
Wen and Bayard present a computed torque algorithm (with model evaluated along the desired trajectory) without exact linearisation in [2.75] and extend it to an adaptive scheme in [2.6]. The stability properties of various cases are analysed and conditions on feedback gains are made depending on the desired trajectory and starting conditions. The controllers are further extended with adaptive feedback gains, when the trajectory and start point conditions are unavailable. Spong and Ortega show an alternative way to avoid the condition of inertia matrix invertibility by applying a generic model in [2.62] with estimated constant inertia, Coriolis and centrifugal and gravity terms and add corrections based on an adaptive scheme.
A review of different robust and model based control schemes for robotic manipulators is presented in [2.1]. An interesting comparison of various adaptive and non-adaptive controllers is presented in [2.76]. The controllers are applied to two types of manipulators with direct-drive and geared joints, Coulomb and viscous friction is considered. The authors show that the Wen-Bayard controller performs best in both fixed-parameter and adaptive controller groups. It is also shown that for very fast motion, when the manipulator cannot follow the desired trajectory, the Wen-Bayard controller can lead to significantly worse performance or even instability (assuming fixed control gains). An interesting property of geared manipulators is stated and shows that the friction model dominates the inertial and Coriolis and centrifugal terms in the manipulator dynamics equations. Therefore, a PD controller with friction compensation is suggested for such cases. Moreover, it is stated that when the desired trajectory is changed during the operation, the adaptive controllers re-tune within seconds. Another comparison of robust and model based controllers is presented in [2.45] with less elaborate comparisons, but similar general conclusions about the advantage of adaptive versus fixed-parameter controllers in the case of incomplete parameter knowledge.
Arimoto combined an adaptive controller updating the parameters in the gravity compensation term with a PID controller and performs task space control in [2.2]. A completely different approach to adaptive gravity compensation by means of hydraulic design (not model based) is presented in [2.38].

2.4 Manipulator on a moving base

The majority of manipulator dynamic and kinematic modelling work applies to fixed base robots, where the manipulator is attached to an inertial coordinate system and no external forces influence its motion. Such assumption does not always hold, consider for example one of the following:

- manipulator attached to an underwater vehicle (e.g. a remotely operated vehicle or

a submarine),

- manipulator attached to a space vehicle (e.g. a satellite or a shuttle),
- manipulator attached to a water vessel (e.g. a ship or a floating oil platform),
- manipulator mounted on a vehicle driving on uneven surface.

It can be noticed immediately that in such cases the dynamics of the manipulator can be influenced by additional forces. In the case of underwater operation - fluid dynamics drag effects originating from the vehicle moving and buoyancy/gravity changes from varying orientation. The case of space vehicles presents different issues, due to no gravity effects and a free-floating base. Both devices (the manipulator and the base) influence the system dynamics. In the case of a manipulator mounted on a water vessel, the motion of the base comes from the sea state and is often difficult to predict with high accuracy and measurements, or special tracking systems are required. The following types of interaction can be distinguished depending on the relative mass/inertia of the system elements:

- Light manipulator on a heavy base - in this case the motion of the manipulator would hardly affect the dynamics of the base, and can be considered negligible, however, the base motion could have a significant effect on the manipulator dynamics.
- Manipulator and base with comparable weights - in this case the base motion influences the manipulator, and the manipulator motion influences the base.
- The third option of a heavy manipulator and light base is rather unlikely.

These dynamic disturbances create new problems in control. For the heavy base and light manipulator, the fixed base model does not apply and additional torques appear that can impact the system performance (e.g. the actuator torques may saturate and large tracking errors can occur). In the case of comparable weights, the motion of the manipulator can displace and change the base orientation affecting the kinematics model.
A dynamic model of a 1-DOF robotic manipulator on a 3-DOF moving platform is presented in [2.30]. The authors assume that the platform follows a random motion pattern (like a manipulator mounted on a vehicle driving on an uneven surface) and apply a feedback linearisation algorithm with PID controller to the linearised system. The integrator helps deal with the disturbance, but also introduces stability issues. Tanner measures the base motion disturbance and adds a non-linear compensator to a generic linear controller in [2.67]. A dynamic model for a 2-DOF manipulator on a 2-DOF (pitch + Z) platform is derived (for the application of tank cannon munition loading) and the model parameters are obtained based on cylindrical link shapes. A linear model is estimated along the desired trajectory and used to design SISO and MIMO controllers. These are shown not to perform well in the presence of base disturbance and readings from four sensors are added (angle, angular rate, angular acceleration and linear acceleration) and incorporated into a

nonlinear compensator combined with the SISO and MIMO controllers. This controller is shown to be superior in the presence of base disturbances, however, due to the lack of angular acceleration measurements it is removed from the compensator. It is shown that such a design meets the requirements on positioning error, based on simulations even for 10% parameter inaccuracies. Experiments are performed for a PUMA 250 manipulator on a 1-DOF (Z-axis translation) platform.

An analysis of cooperating manipulators on a non-fixed base is shown in [2.52]. It is assumed that the platform is not actuated but is affected by the manipulator dynamics. A set of dynamic equations for a 6-DOF platform and two 6-DOF manipulators is presented.

Ship heave measurements based on different sensors are presented in [2.22] with the conclusion that combining a low sampling rate global measurement (from a GPS device) with high sampling rate measurements from an inertial navigation system (based on accelerometers and angular rate sensors) delivers best results.

A manipulator model on a compliant base (e.g. a crane) is presented in [2.40]. The authors use the previous control step moments to estimate the unmodelled dynamics with a high command frequency of 500 Hz. The combined Coriolis and centrifugal term is ignored and the Inertia matrix is only approximately estimated. Second derivatives of the joint coordinates are required for this estimation which introduces noise. It is shown that such a controller is superior to a simple PD by being able to decouple the manipulator dynamics from the base and reducing base oscillations. A 2-DOF manipulator on a 3-DOF (X-Y-Z) platform is considered.

A dynamic model of a robotic manipulator on a mobile base is derived in [2.11] and used for simulation of a 2-DOF robot on a 3-DOF base. Only a PD controller is applied and no compensation for base motion in the dynamics is considered.

Duindam and Stramigioli present a new mathematical framework for analysing manipulators on mobile bases with coordinate frames not isomorphic to R^n spaces in [2.19]. It allows singularities originating from orientation representation to be removed.

2.4.1 Underwater manipulators

Computational methods to simulate a manipulator mounted on an underwater vehicle were presented in [2.47]. The additional hydro-dynamic elements are analysed and modelled along with the generic manipulator dynamics. The recursive Newton-Euler approach is used.

Lee and Yuh show an adaptive controller for a manipulator attached to an underwater platform in [2.39]. Hydrodynamic effects are considered but the dynamic model is assumed unknown. The inertia, Coriolis and centrifugal and gravity terms are estimated by constants and adapted using the error norm. Stability is shown for adaptation with increasing parameters, however for practical reasons the parameters should be allowed to reduce. In such case the stability cannot be guaranteed, but the authors present an argument that the adaptation would switch back to increasing values depending on error magnitude. Since

the tracking error is used in the denominator of the adaptation formula, another problem appears - high switching values of the parameters when tracking errors are close to zero, additional modification is applied with a limit on the denominator below a certain error level. This in turn breaks the asymptotic stability, and an integrator element is added to the controller to handle this. A planar 2-DOF manipulator on a 1-DOF platform is considered.

2.4.2 Space manipulators

An image of a manipulator in space is shown in figure 2.3. The gravity present on Earth

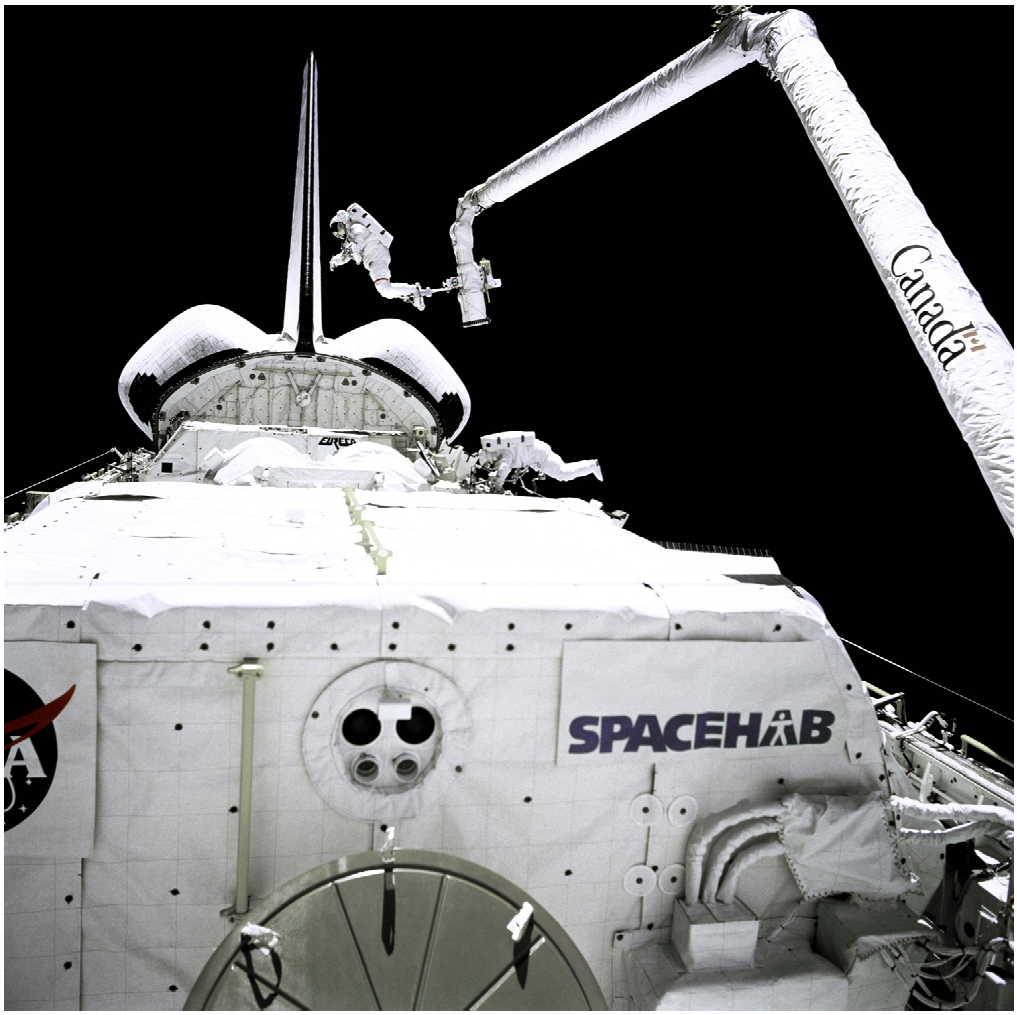


Figure 2.3: Canadarm space manipulator (source: sm.mdacorporation.com).

is not present in orbit and therefore not considered in the manipulator dynamic models. Vafa and Dubowsky present a new modelling framework for manipulators in space in [2.74]. It is based on the fact that the total momentum of the system is preserved and a *Virtual Manipulator* can be defined with the base represented as the first joint (spherical) and all manipulator joints having a corresponding virtual joint (with axes parallel to physical joint axes for rotary joints). The base of the virtual manipulator is attached

to the centre of mass of the manipulator+base system. This new method simplifies the kinematic and dynamic modelling of manipulators in space.
Papadopoulos and Dubowsky analyse the dynamics, kinematics and control of space vehicle mounted manipulators in [2.18], [2.55]. The following classification of such systems is introduced:

- **free-flying** with an actively controlled base using thrusters,
- **free-floating** with an uncontrolled base continuously affected by the manipulator motion.

A dynamic model of a robotic manipulator attached to a space vehicle is derived (friction is not considered). The motion of the base is generally performed as a separate task from the manipulator operation, therefore no dynamic influence of the base on the manipulator is considered. The kinematics is modelled and 'dynamic singularities' are pointed out for the free-floating case, where the motion of the robotic arm could change the base orientation and prevent the end-effector from reaching the desired position/orientation. It is remarked that point-to-point control is not feasible as the position depends on the path used to reach it. A linearisation controller in task space is proposed and a simulation is performed for the planar case of a 2-DOF manipulator on a 3-DOF (X-Y-orientation) space-ship. The case of multiple manipulators is presented in [2.56] and comparisons of inverse Jacobian and full linearisation in task space are made.
Chen and Canon Jr. apply adaptive control (Wen-Bayard) in the task space to compensate for load changes of the manipulated object in [2.12]. A system concatenation approach (combination of dynamic equations of all elements with constraints on forces and motion) is used to model multiple manipulators on a space ship. Different Jacobians are used (and switched between during control) to achieve different control tasks (joint, inertial space, object space). A planar case of a 2-DOF manipulator on a 3-DOF (X-Y-orientation) base is considered in simulations.
A review of present modelling, control and path planing techniques for space vehicle mounted manipulators is presented in [2.51].

2.4.3 Manipulators on water vessels

Control of a manipulator attached to a vehicle free-floating on water is presented in [2.35]. It is assumed that the water is still and the manipulator motion affects the base dynamics. Two control strategies are compared:

- passive base vibration reduction by constraining the manipulator motion,
- simultaneous position and acceleration control.

The control is performed in the task space using feedback linearisation. The first controller is able to track a given trajectory without inducing base vibration, and the second

controller can track a trajectory and simultaneously reduce the base vibration. Simulation is performed for a planar 2-DOF manipulator on a 1-DOF (angle) base.

Love, Jansen and Pin consider a robotic manipulator working on a ship in [2.42, 2.43]. The dynamic model is derived for a general 6-DOF+6-DOF case, however the actual formulae to derive the dynamics terms are not presented (only MATLAB code). A comparison of sea states and ship motion depending on wave height is presented for different ship sizes. An adaptive controller tuning to the wave period is suggested (called *'Repetitive Learning Controller'*), but it only performs well for single frequency waves, and not for sea states. In addition a beam forming technique borrowed from signal processing is used to compensate for the base motion in task space force control. Accelerometers and inclinometers are combined with a tip mounted force sensor to significantly reduce the force control error (ten fold compared to no compensation).

From, Duindam, Gravdahl and Sastry suggest in [2.20] to use the base induced torques in motion planing to reduce joint torques and control energy for performing tasks (at the cost of longer task duration). The base trajectory is assumed to be known and is used in a computed torque feedback controller. The savings in actuator energy for performing a task planned with the base motion consideration are presented. Friction is not considered in the model. Simulations are performed for a 1-DOF and 4-DOF manipulators on a 6-DOF base.

Chapter 3

Kinematics and dynamics modelling

To model the motion of a robotic manipulator (consisting of a chain of bodies), several assumptions about the nature of admissible degrees of freedom have to be made. The following assumptions apply here

- each link is a rigid body,
- the joints are fully actuated and have no flexibility,
- the mass of each link is constant.

Therefore, the notion of rigid bodies is required.

3.1 Describing the position of rigid bodies

A compact set of points, whose positions are constant with respect to each other, with a certain mass distribution, is considered as a rigid body. To define the position of such an entity in the three–dimensional Cartesian space, a particular point has to be chosen. It can be identified with a vector in the three dimensional Euclidean space ($\mathbb{R}^3$) $\mathbf{r} = (x,y,z)^T$. (A translation in the Euclidean space, can also be identified with the vector along which it is performed $\mathbf{T} = (x,y,z)^T$).
A local coordinate system can be attached to the body, and its orientation can be expressed in the (global) coordinate system, in which the position of the rigid body was defined, by means of a rotation matrix $\mathbf{R} \in M_{3\times 3}$ such that $\mathbf{R}^T\mathbf{R} = \mathbf{I}$ (identity matrix) and $\det \mathbf{R} = 1$. By extending the position vector to $(\mathbf{r}^T, 1)^T$, the position and orientation can be combined into a matrix (representing a homogeneous transformation). Each transformation matrix consists of **rotation** and **translation** elements

$$\mathbf{A} = \begin{bmatrix} \mathbf{R}_{3\times 3} & \mathbf{T}_{3\times 1} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}, \tag{3.1}$$

where

$\mathbf{R}_{3\times 3}$ is a 3-by-3 rotation matrix,

$\mathbf{T}_{3\times1}$ is a translation vector of length 3,

$\mathbf{0}_{1\times3}$ is a 1×3 zero vector,

Assuming that this transformation describes the local coordinate system in the global space, a point ($\mathbf{r}_l$) in the local coordinates, can be expressed in the global space as $((\mathbf{R}\mathbf{r}_l+\mathbf{T})^T,1)^T=\mathbf{A}(\mathbf{r}_l^T,1)^T$.
The rotation matrix (although having 9 parameters) belongs to a three dimensional space. There exist several parametrisations of it, two of them are used here:

- Euler angles (around Z-Y-Z axes),
- Roll–Pitch–Yaw angles.

Define rotations around the axes of the coordinate system

$$\begin{aligned}
\mathrm{Rot}(X,\alpha) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix}, \\
\mathrm{Rot}(Y,\beta) &= \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix}, \\
\mathrm{Rot}(Z,\gamma) &= \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}.
\end{aligned} \tag{3.2}$$

With these, the Euler angles representation can be expressed as

$$\mathbf{R}_{\mathrm{Euler}}(\phi,\theta,\psi)=\mathrm{Rot}(Z,\phi)\,\mathrm{Rot}(Y,\theta)\,\mathrm{Rot}(Z,\psi)$$

with the domain

$$\mathbf{D}_{\mathrm{Euler}}=\{(\phi,\theta,\psi)|0<\phi<2\pi,0<\theta<\pi,0<\psi<2\pi\},$$

and the Roll–Pitch–Yaw representation can be expressed as

$$\mathbf{R}_{\mathrm{RPY}}(\phi,\theta,\psi)=\mathrm{Rot}(Z,\phi)\,\mathrm{Rot}(Y,\theta)\,\mathrm{Rot}(X,\psi)$$

with the domain

$$\mathbf{D}_{\mathrm{RPY}}=\left\{(\phi,\theta,\psi)|0<\phi<2\pi,-\frac{\pi}{2}<\theta<\frac{\pi}{2},0<\psi<2\pi\right\}.$$

The rotation matrix (**R**) can be interpreted as an element of a special mathematical space denoted by the symbol $\mathbb{SO}(3)$, and the combination of rotation and translation (**A**) can be interpreted as an element of another mathematical space denoted by the symbol $\mathbb{SE}(3)$.

3.2 Forward kinematics

The forward kinematics (also shortened to kinematics) of a manipulator is a map $\mathcal{K}: \mathbb{R}^n \to \mathbb{SE}(3)$ from *joint space* (also known as *configuration space*) to *task space* (also known as *global coordinates*)

$$\begin{bmatrix} x \\ y \\ z \\ \varphi \\ \theta \\ \psi \end{bmatrix} = \mathcal{K}\left(\begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix} \right), \tag{3.3}$$

where:

$\mathbb{SE}(3)$ $=\mathbb{R}^3 \times \mathbb{SO}(3)$ is the special Euclidean group, where $\mathbb{SO}(3)$ is the special group of rotations,

n number of degrees of freedom,

x, y, z location in global coordinates,

φ, θ, ψ orientation in global coordinates,

$\mathcal{K}$ the kinematics transformation (consists of trigonometric functions and the link parameters).

q_i generalised coordinates representing the state of the robot joint configuration.

Equation (3.3) defines the position of the end-effector in the global coordinate system as a function of the manipulator joint angles. The forward kinematic transformation is easy to compute using a widely known and very popular method described by Denavit and Hartenberg [3.3].

3.2.1 Denavit – Hartenberg notation

The Denavit – Hartenberg (D-H) notation employs the introduced (equation 3.1) 4×4 homogeneous transformation matrix to describe the relationship between the various joint coordinate systems. The rotation and translation elements are calculated for each joint based on a set of Denavit – Hartenberg parameters (two angles and two translations): θ, α, a, and d (a schematic of the physical meaning of the parameters is presented in figure 3.1 and explained in table 3.1). For reference see any of [3.3], [3.6], [3.5], [3.2], [3.4].

$$^{i-1}\mathbf{A}_i = \mathrm{Trans}(z, d_i) \times \mathrm{Rot}(z, \theta_i) \times \mathrm{Trans}(x, a_i) \times \mathrm{Rot}(x, \alpha_i),$$

parameter	description
θ_i	joint angle (angle between links around joint axis)
α_i	link twist (angle between the i-th and $i+1$-th joint axes around their mutual perpendicular)
a_i	link length (along a mutual perpendicular to i-th and $i+1$-th joint axes)
d_i	link offset (distance along joint axis)

Table 3.1: Description of Denavit – Hartenberg parameters defining forward kinematics of a manipulator.

$$
{}^{i-1}\mathbf{A}_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

By superposition of these transformations for successive joints, the coordinate change between arbitrary coordinate systems in the manipulator can be obtained

$$
{}^{k}\mathbf{A}_r = {}^{k}\mathbf{A}_{k+1} \dots {}^{r-1}\mathbf{A}_r \tag{3.4}
$$

assuming $0 \leq k < r \leq n$ (n - number of degrees of freedom of the manipulator). The parameters required to fully formulate the kinematics for a given manipulator can be arranged in a table.

3.3 Inverse kinematics

To perform tasks in the Cartesian space, the manipulator controller requires the desired positions and velocities of the joints, based on the given target position and orientation in the global, or tool coordinates. There are two ways to approach this task:

- instantaneous joint velocities, derived from Cartesian velocities (*inverse Jacobian*)

$$
\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix} = \left(\frac{\partial}{\partial \mathbf{q}} \mathcal{K} \left(\left[q_1, q_2, \dots, q_n \right]^T \right) \right)^{-1} \cdot \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}, \tag{3.5}
$$

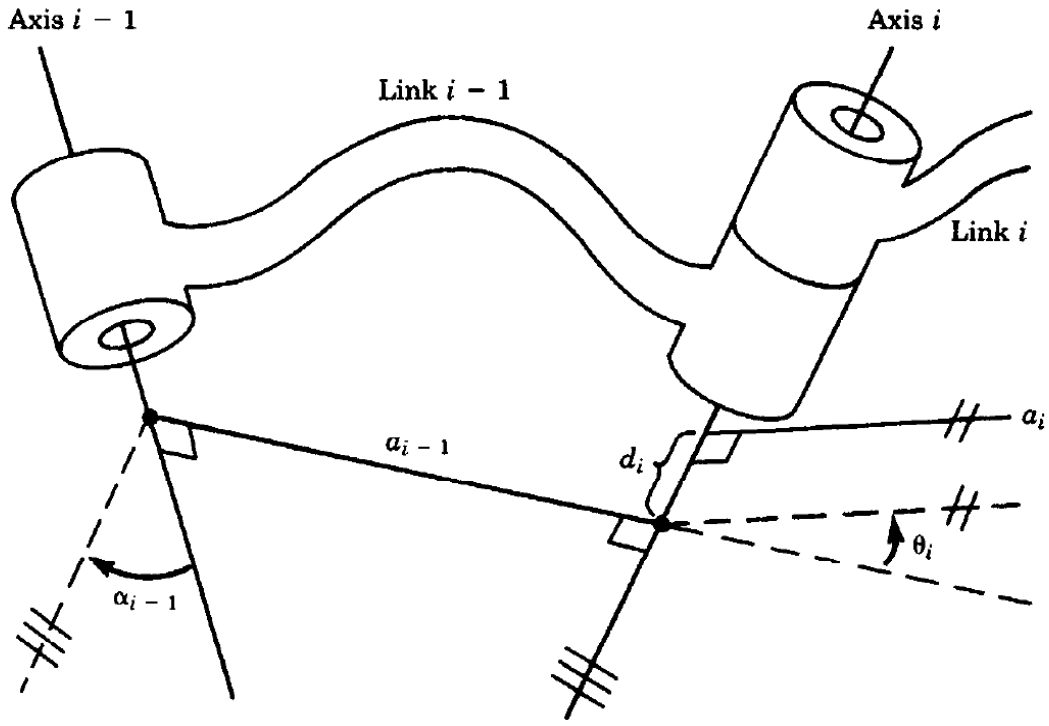


Figure 3.1: Schematic of the Denavit-Hartenberg notation parameters (source: [3.2]).

- implicit formulae to transform Cartesian positions into the joint space

$$\begin{bmatrix} x \\ y \\ z \\ \varphi \\ \theta \\ \psi \end{bmatrix} - \mathcal{K}\left(\left[q_1, q_2, \ldots, q_n\right]^T\right) = 0 \tag{3.6}$$

  this equation is solved for q_i's, with x, y, z, φ, θ and ψ given.

The first solution is the most popular, although the strict inverse exists only for non-singular square matrices. However, robust pseudo-inverses are used to deal with singularities, and extra conditions are applied to the Jacobian, for redundant manipulators (which can be easily used for optimal trajectory generation, e.g. minimising control energy). The latter solution generates precise positions, assuming that the motion is free from singularities, which implies that only one global configuration of the manipulator can be chosen. The reachable space is constrained (if no transitions between global configurations are allowed).

3.4 Dynamics

The dynamics of a robotic manipulator with non-flexible joints can be expressed by a second order differential equation (using the Euler – Lagrange formalism) [3.2], [3.5]

$$\mathbf{Q}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \tau + \mathbf{u}, \tag{3.7}$$

where:

- $\mathbf{q} \in \mathbb{R}^n$ is the state vector of generalised coordinates (n - number of degrees of freedom),
- $\mathbf{Q}$ is the inertia matrix,
- $\mathbf{C}$ is the Coriolis and centrifugal vector,
- $\mathbf{g}$ is the gravity vector,
- $\mathbf{u}$ is the vector of inputs (torques), and
- τ is a vector of frictions and disturbances.

Assume that

$$\tau = -\mathbf{F} + \mathbf{F}_e, \tag{3.8}$$

where

- $\mathbf{F}$ is the friction force,
- $\mathbf{F}_e$ are all other external or disturbance forces (un-modelled).

By substituting equation (3.8) into equation (3.7) the model of a manipulator can be written as

$$\mathbf{Q}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{F} = \mathbf{u} + \mathbf{F}_e. \tag{3.9}$$

3.4.1 Model derivation procedure

A procedure for deriving the dynamic model for a 6-DOF manipulator with revolute joints is presented in [3.4]. The dynamic equation (3.9) elements are defined as follows:

- the inertia matrix

$$\mathbf{Q}(\mathbf{q}) = [D_{ij}], \quad \text{with}$$
$$D_{ij} = \begin{cases} \sum_{k=j}^{6} Tr(\mathbf{U}_{kj}\mathbf{J}_k\mathbf{U}_{ki}^T) & \text{for } i \leq j, \text{ and} \\ D_{ji}, & \text{for } i > j \end{cases}, \tag{3.10}$$

where

$$\mathbf{U}_{ij} = \begin{cases} {}^{0}\mathbf{A}_{j-i}\ Q_d\ {}^{j-1}\mathbf{A}_i & \text{for } j \leq i \\ 0 & \text{for } j > i \end{cases} \quad \text{with } Q_d = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{J}_i = \begin{bmatrix} \frac{-\mathbf{I}_{xx}+\mathbf{I}_{yy}+\mathbf{I}_{zz}}{2} & \mathbf{I}_{xy} & \mathbf{I}_{xz} & m_i\bar{x_i} \\ \mathbf{I}_{xy} & \frac{\mathbf{I}_{xx}-\mathbf{I}_{yy}+\mathbf{I}_{zz}}{2} & \mathbf{I}_{yz} & m_i\bar{y_i} \\ \mathbf{I}_{xz} & \mathbf{I}_{yz} & \frac{\mathbf{I}_{xx}+\mathbf{I}_{yy}-\mathbf{I}_{zz}}{2} & m_i\bar{z_i} \\ m_i\bar{x_i} & m_i\bar{y_i} & m_i\bar{z_i} & m_i \end{bmatrix}$$

with

$$\mathbf{I}_{hj} = \int_{\mathcal{V}_i} \left[\delta_{hj} \left(\sum_k x_k^2 \right) - x_h x_j \right] dm. \tag{3.11}$$

This procedure ignores the motor inertias, which need to be added to the diagonal of the inertia matrix

$$\tilde{D}_{ii} = D_{ii} + k^2 I_{m_i}, \tag{3.12}$$

where

k the gear ratio

I_{m_i} the motor inertia of the i-th motor

- the Coriolis and centrifugal vector

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = [\dot{\mathbf{q}}^T \mathbf{H}_{i,\nu} \dot{\mathbf{q}}] \quad \text{with } \mathbf{H}_{i,\nu} = [h_{ijk}], \quad i, j, k \in \{1, \ldots, 6\}, \tag{3.13}$$

where

$$h_{ijk} = \sum_{m=\max(i,j,k)}^{6} Tr(\mathbf{U}_{mjk} \mathbf{J}_m \mathbf{U}_{mi}^T)$$

and

$$\mathbf{U}_i jk = \begin{cases} {}^{0}\mathbf{A}_{j-i}\ Q_d\ {}^{j-1}\mathbf{A}_{k-1}\ Q_d\ {}^{k-1}\mathbf{A}_i & i \geq k \geq j \\ {}^{0}\mathbf{A}_{k-i}\ Q_d\ {}^{k-1}\mathbf{A}_{j-1}\ Q_d\ {}^{j-1}\mathbf{A}_i & i \geq j \geq k \\ 0 & i < j \text{ or } i < k, \end{cases}$$

- the gravity vector (based on [3.6])

$$\mathbf{g}(\mathbf{q}) = \frac{\partial}{\partial \mathbf{q}} V(\mathbf{q}), \quad \text{where } V(\mathbf{q}) = -\sum_{i=1}^{6} m_i \mathbf{g}^T\, {}^{0}\mathbf{A}_i(\mathbf{q}) \mathbf{R}_i \tag{3.14}$$

with:

m_i the mass of the i-th link,

R_i location of i-th link mass point expressed in the i-th coordinate system (according to modified Denavit – Hartenberg notation) and $R_i = (\overline{x_i}, \overline{y_i}, \overline{z_i})^T$,

$\mathbf{g}^T = (\mathsf{g}_x, \mathsf{g}_y, \mathsf{g}_z, 0)$ the gravity vector expressed in the global coordinate system,

${}^0\mathbf{A}_i$ the (modified) Denavit – Hartenberg transformation from the i-th coordinate system to the global coordinate system,

$\mathcal{V}_i$ the link volume,

$V(\mathbf{q})$ the potential energy of the system,

The integral over the volume,

$$\int_{\mathcal{V}_i} () \, dm$$

can be simplified to an iterated integral

$$\int\int\int_{\mathcal{V}_i} ()\rho(x,y,x)dxdydz,$$

where $\rho(x,y,z)$ is the density at point (x,y,z).
In order to derive the dynamics of a robotic manipulator, it is required to know

- masses of links,
- centres of masses of links,
- inertias of links, which can be derived from
  - explicit formulae defining volumes of links,
  - densities of links,
- the forward kinematics transformation (parameters for the Denavit - Hartenberg transformation (table 3.1 and figure 3.1).

This information is not usually provided by the robot manufacturer (or retailer), and requires additional assumptions and analysis.

3.4.2 Modelling friction

In the classical control engineering and robotic literature, friction has often been neglected or treated as an unobservable disturbance. The main reason for that is its observable discontinuity. The importance of friction modelling has increased over the recent decades and Armstrong-Hélouvry [3.1] presents important aspects of friction modelling in the control of machines.
Friction models have usually been derived from data, based on thorough observations and simplifying assumptions. There have been attempts to understand friction since Leonardo

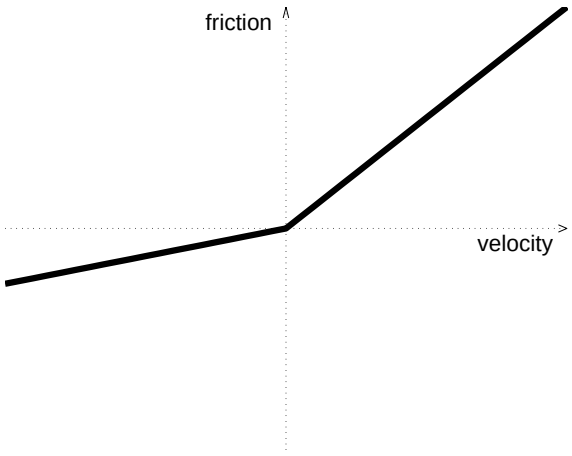


Figure 3.2: Viscous friction (linear in velocity).

da Vinci, but only in the last century with the widespread use of machines and the need for optimisation of wear and tear of mechanisms in the field of tribology[1] has developed. Due to the high complexity (especially at low velocities), the friction models are tailored to particular applications to obtain a balance between simplicity and exactness. Both parametric and non-parametric models have been derived and used, with the latter providing better estimates, at the cost of extra algorithmic complexity and with the loss of analytic properties (for stability proofs).
In order to achieve precise positioning of a robotic arm, without high gains and significant overshoot, a consistent model for friction is required. The most popular models are presented below:

- **Viscous friction** - the simplest model (Figure 3.2), describes friction force as a linear function of velocity.

$$\mathbf{F} = f_c \cdot \dot{q} \tag{3.15}$$

  This equation can be further refined, by assuming that the proportional coefficient (f_c) can be different for positive and negative velocities (model becomes non-linear (and not smooth) at zero). This model is very simple (and robust), easy to estimate and not computationally demanding, but it does not work at low velocities.

- **Viscous, kinetic and static friction** - a further extension to the previous model (Figure 3.3). Two additional features are included:

  – The amount of static friction (stiction) needed to be overcome, to initiate the motion.

  – The minimum friction level at lowest achievable velocities (kinetic friction).

[1] from Greek - the study of rubbing

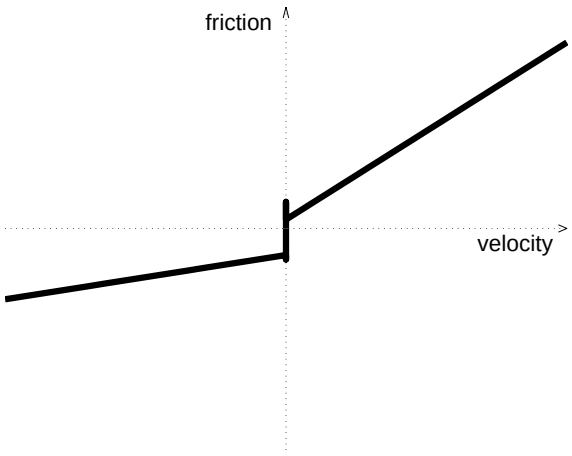


Figure 3.3: Viscous and static friction (linear in velocity except at 0).

$$\mathbf{F} = \begin{cases} f_k^+ + f_c^+ \cdot \dot{q} & \text{for} \quad \dot{q} > 0 \\ f_k^+ + f_s^+ & \text{for} \quad \dot{q} = 0^+ \\ f_k^- + f_s^- & \text{for} \quad \dot{q} = 0^- \\ f_k^- + f_c^- \cdot \dot{q} & \text{for} \quad \dot{q} < 0 \end{cases} \tag{3.16}$$

where

- $f_k^{+/-}$ kinetic friction for positive/negative velocities
- $f_s^{+/-}$ static friction for positive/negative velocities
- $f_c^{+/-}$ viscous (Coulomb) friction for positive/negative velocities

This model is not continuous at zero. Whilst it addresses the problem of providing additional force to start the motion, but the profile of friction force at low velocities is not in line with experimental results and therefore requires a further enhancement.

- **Generalised Stribeck model** - a further refinement of the friction curve at low velocities, based on analysis of uneven rubbing surfaces with lubrication (Figure 3.4). The following stages can be distinguished in the model
  - The *static friction* has to be overcome, to start the motion.
  - At very low velocities there is very little lubrication available, and the friction level depends mostly on the lubricant *boundary lubrication* properties. The level of friction at that stage is assumed to be constant [2].
  - Once the rubbing surfaces start moving fast enough to distribute some lubricant, the model passes to the *partial lubrication* stage, where friction decreases with increasing velocity (as lubrication improves).

[2] as observed in experiments

– At sufficiently high velocity, a layer of thin film of the lubricant is created between the surfaces, and the friction starts increasing proportionally with velocity. This is the stage of *full lubrication.*

In a commercial product, the composition of lubricants is a fixed parameter (often unknown), and cannot be changed, therefore the main focus (of the control engineer) is on the modelling aspects. However, any information about such properties is useful in the model development step. The following models (presented in [3.1]) include all the mentioned stages

$$\mathbf{F} = F_k + \frac{F_s - F_k}{1 + \left(\frac{\dot{x}}{\dot{x}_s}\right)^2} + F_v\dot{x} \tag{3.17}$$

$$\mathbf{F} = F_k + (F_s - F_k)e^{-\left(\frac{\dot{x}}{\dot{x}_s}\right)^\delta} + F_v\dot{x} \tag{3.18}$$

where

F_S is the level of static friction,

F_k is the minimum level of kinetic friction,

$\dot{x}_s$ is an empirical velocity scaling parameter,

δ is an empirical parameter,

F_v is the viscous friction.

Another empirical extension of this model is to use a different set of parameters for negative and positive velocities. This model, although complicated and requiring nonlinear optimisation to obtain the parameters, is widely used in control engineering to compensate friction in simple cases. However, if several elements rub at the same time, the composite effect, may not be simple enough to be described by this model. Further improvements have been attempted.

- **Non-parametric models** The parametric models are easy to formulate and have certain smoothness properties (away from zero velocity). However, the estimation of parameters and goodness of fit may not be satisfactory for a particular application, especially for highly complex mechanical systems, where the Stribeck curve may not fit the data. Due to the increasing availability of memory and processing power, nonparametric models can be created for the purpose of friction estimation/compensation. These are based on a large number of measurements, which are averaged and tabulated for precise interpolation. Such a system requires significantly more computational power for derivation, and the model may not have any smoothness properties, but for low velocities, it provides a very powerful estimation/compensation tool. Two main model groups can be distinguished

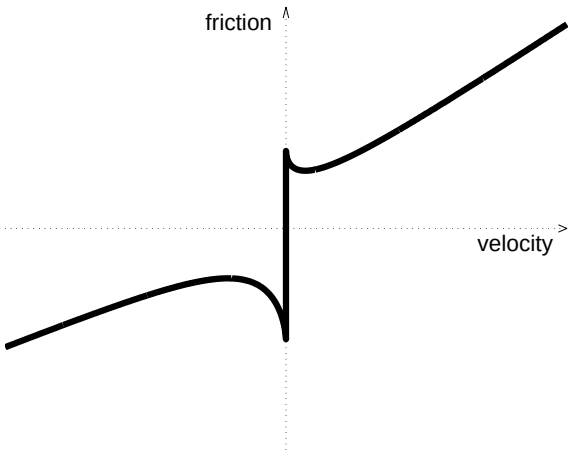


Figure 3.4: Nonlinear friction model with Stribeck effect.

– velocity dependent – the following formula presents the simplest look-up table with a linear interpolation model.

$$\mathbf{F} = F_{v_{i-1}} \frac{\dot{q} - v_{i-1}}{v_i - v_{i-1}} + F_{v_i} \frac{v_i - \dot{q}}{v_i - v_{i-1}} \qquad \text{for } \dot{q} \in (v_{i-1}, v_i) \tag{3.19}$$

where

F_{v_i} is the friction force estimate at velocity v_i.

– position dependent – the following formula presents the simplest look-up table with a linear interpolation model.

$$\mathbf{F} = F_{x_{i-1}} \frac{q - x_{i-1}}{x_i - x_{i-1}} + F_{x_i} \frac{x_i - q}{x_i - x_{i-1}} \qquad \text{for } q \in (x_{i-1}, x_i) \tag{3.20}$$

where

F_{x_i} is the friction force estimate at position x_i.

This model is highly dependent on the wear of the system, and may require a regular re-estimation after the rubbing surfaces have changed significantly.

Both methods can be combined for increased precision (and a significant increase in estimation complexity), or the position dependent model can be added to one of the parametric models (especially for low velocity feed-forward applications).

Alongside the general models, other important features of the friction phenomenon have to be considered. Depending on the lubricant composition, the **stick-slip** effect may occur. This happens mainly when motion is initiated and at very low velocities. It can be modelled using a pin-spring-damper configuration, which exhibits two main properties

- **Dwell-time -** is the amount of time a rubbing surface remains at a fixed position, although a force is applied to it. The dwell time depends on the time the machine was

left unused and on the motion velocity. It depends on the stiffness of the system and the lubricant used. By increasing the system stiffness and choosing an appropriate composition of lubricant, this effect may be eliminated.

- **Frictional lag** - is a delay in the response of a system with friction, caused by internal (surface related) dynamics. At low velocities, a change of velocity is not instantaneously followed by a change of friction force. The latter takes about 3-9 ms to settle at its new level. This effect can generate significant disturbances in the control system of a machine with friction, requiring very fast response times.

Chapter 4

Model derivation and estimation of the PA10-6CE manipulator parameters

Several attempts have been made to deliver a dynamic model of both versions of the PA10 manipulator (6-DOF and 7-DOF), these were reviewed in section 2.2.2.2. This chapter presents an approach with the following contributions:

- geometric models used to derive link inertia with the assumption of setting the link geometric elements densities to match the centre of mass,
- a non-parametric friction model based on broad velocity range experiments,
- a brief analysis of open-loop control results for the obtained model.

4.1 The mechanical design of the PA10-6CE manipulator

The PA10-6CE is a multi-purpose manipulator from Mitsubishi Heavy Industries. It has six degrees of freedom, which can be related to a human arm as shown in table 4.1 and figure 4.1. The manipulator is mainly targeted at specialised light payload tasks, with low repeatability (e.g. clean-room object handling, nuclear power plants or single welding jobs). Each joint is driven by a separate AC motor coupled via a harmonic gear. The transmission of joint W1 also includes a belt. The manipulator is relatively light (37 kg), which enables it to be mounted on mobile platforms and parallel robots (e.g. to increase the available work space/dexterity).
The manufacturer provides partial information on the construction of the manipulator. Distances between axes, masses and centres of masses of particular joint elements are available. The information from figure 4.2 is summarised in tables 4.2[1] and 4.3.

[1]The centre of mass is expressed in the coordinate system attached to the link.

Joint name	Arm motion
S1	shoulder forward/backward swing
S2	shoulder lateral raise
E1	elbow swing
E2	elbow turn
W1	wrist swing
W2	wrist rotate

Table 4.1: PA10 joints identified with human arm movements.

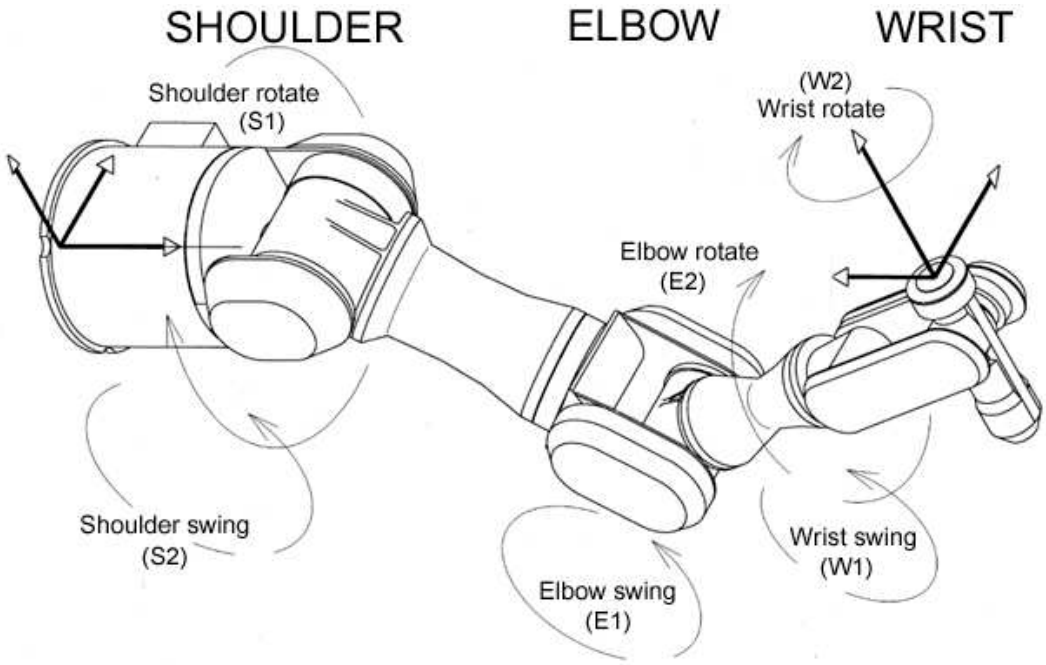


Figure 4.1: Schematic of PA10-6CE manipulator (with joint names).

Joint	centre of mass $R_i = (r_{xi}, r_{yi}, r_{zi})[m]$	mass $m_i[kg]$
S1	(0 , -0.178526, 0)	9.29
S2	(-0.321, -0.03 , 0)	12.43
E1	(0 , 0 , 0.0484568)	4.86
E2	(0 , -0.112231, 0)	3.08
W1	(0 , 0 , -0.042)	2.07
W2	(0 , 0 , -0.048)	1.05
total		32.78+

Table 4.2: Summary of PA10-6CE joints mass points (based on manufacturer data).

4.2 Kinematics

4.2.1 Forward kinematics formulation

Based on the schematic drawing provided by the manufacturer of the PA10 manipulator, the forward kinematics were obtained. The positive direction of the X axis for the base

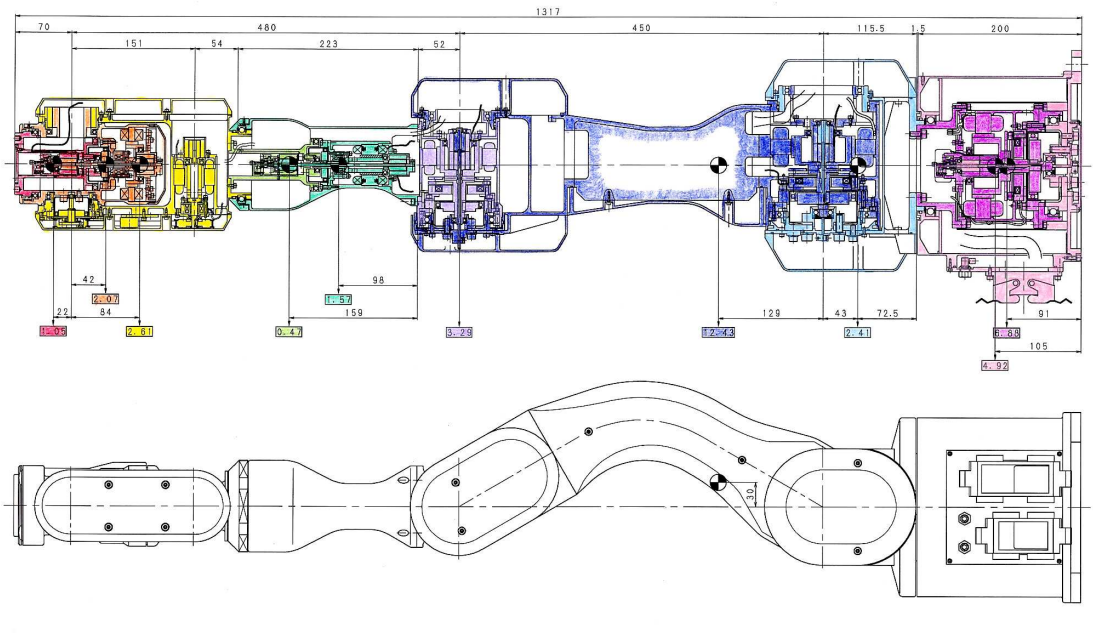


Figure 4.2: Schematic of PA10-6CE manipulator (with joint masses and centres of mass).

joint axes	distance [m]	D-H symbol
(base)↔(S2)	$\frac{317}{1000}$	d_1
(S2)↔(E1)	$\frac{9}{20}$	a_2
(E1)↔(W1)	$\frac{12}{25}$	d_4
(W1)↔(tool mount)	$\frac{7}{100}$	d_6

Table 4.3: Summary of PA10-6CE inter axes distances.

coordinate system has been chosen in the direction of positive angles of joint S2 with S1 set to zero. The complete set of Denavit–Hartenberg transformations used to define the forward kinematics is summarised in table 4.4 and the coordinate systems for each link are presented in figure 4.3.

joint	θ	α	a	d
1 (S1)	$\pi + q_1$	$\frac{\pi}{2}$	0	d_1
2 (S2)	$\frac{\pi}{2} + q_2$	0	a_2	0
3 (E1)	$\frac{\pi}{2} + q_3$	$\frac{\pi}{2}$	0	0
4 (E2)	$\pi + q_4$	$\frac{\pi}{2}$	0	d_4
5 (W1)	$\pi + q_5$	$\frac{\pi}{2}$	0	0
6 (W2)	q_6	0	0	d_6

Table 4.4: Denavit-Hartenberg transformations for PA10-6CE manipulator.

By substituting data from table 4.4 into equation 3.4, joint positions can be transformed into Cartesian positions and rotation matrices (and further angles) expressed in e.g. base, or tool coordinates. The forward kinematic transformation is also required to compute the potential energy of the manipulator.

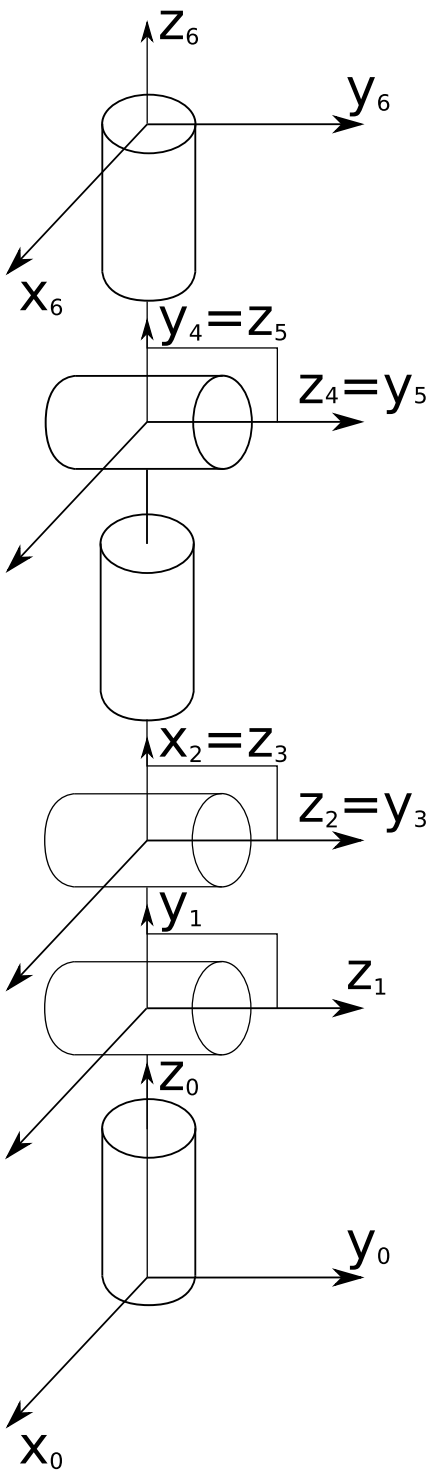


Figure 4.3: Schematic of the PA10-6CE manipulator kinematics with coordinate systems for each link. *Note: index 0 denotes the manipulator base frame; the gravity acceleration on fixed base acts along* $-z_0$

4.2.2 Inverse kinematics formulation

An inverse transformation of the forward kinematics can be obtained for the PA10-6CE, which requires additional assumptions on the posture of the manipulator. It is outside the scope of this thesis to derive this transformation. For practical applications of the manipulator [4.1], an existing solution has been adopted. The inverse kinematics algorithm,

based on decoupling the last three joints as the wrist (→ orientation) from the first three joints (→ position), has been developed by Torsten Sherer ([4.6]). It is a local solution to an implicit equation, separate for each posture (covering the entire workspace).

4.3 Dynamics

To derive the dynamics of a robotic manipulator, detailed information about its construction is required. In the case of the Mitsubishi PA10-6CE, that information was not fully provided by the manufacturer, and several assumptions have been made based on the available schematic (figure 4.2). With these assumptions a complete dynamic model has been derived, and the generalised friction forces have been estimated using elements of the dynamic model.

4.3.1 Gravity term

The gravity term of the PA10-6CE dynamics was the most straightforward to derive. All the required information has been provided by the manufacturer (tables 4.2, 4.3, 4.4). With the forward kinematic transformation and all the masses and centres of masses, the following has been obtained from equation 3.4 and equation 3.14.

$$\mathbf{g}(\mathbf{q}) = \begin{pmatrix} g_{q_1} & g_{q_2} & g_{q_3} & g_{q_4} & g_{q_5} & g_{q_6} \end{pmatrix}^T, \tag{4.1}$$

where

$$\begin{aligned}
g_{q_1} &= 0 \\
g_{q_2} &= -\mathrm{g}\Bigg(a_2 \sin(q_2) \sum_{i=2}^{6} m_i + m_2 \left(r_{y2} \cos(q_2) + r_{x2} \sin(q_2) \right) \\
&\quad + \left(d_4 \sum_{i=4}^{6} m_i + m_4 r_{y4} + m_3 r_{z3} + \cos(q_5) \left(m_5 r_{z5} + m_6 (d_6 + r_{z6}) \right) \right) \sin(q_2 - q_3) \\
&\quad + \cos(q_2 - q_3) \cos(q_4) \sin(q_5) \left(m_5 r_{z5} + m_6 (d_6 + r_{z6}) \right) \Bigg) \\
g_{q_3} &= -\mathrm{g}\Bigg(\left(d_4 \sum_{i=4}^{6} m_i + m_4 r_{y4} + m_3 r_{z3} + \left(m_5 r_{z5} + m_6 (d_6 + r_{z6}) \right) \cos(q_5) \right) \sin(q_2 - q_3) \\
&\quad + \cos(q_2 - q_3) \cos(q_4) \sin(q_5) \left(m_5 r_{z5} + m_6 (d_6 + r_{z6}) \right) \Bigg) \\
g_{q_4} &= \mathrm{g} \sin(q_2 - q_3) \sin(q_4) \sin(q_5) \left(m_5 r_{z5} + m_6 (d_6 + r_{z6}) \right) \\
g_{q_5} &= -\mathrm{g} \left(\sin(q_2 - q_3) \cos(q_4) \cos(q_5) + \cos(q_2 - q_3) \sin(q_5) \right) \left(m_5 r_{z5} + m_6 (d_6 + r_{z6}) \right) \\
g_{q_6} &= 0,
\end{aligned}$$

where $\mathsf{g} = \mathsf{g}_z$ is the vertical (along the z-axis of the base coordinate system) component of the gravitational acceleration constant[2]. The gravity vector exhibits the following properties

- There is no gravitational torque acting on joint 1 (S1) and joint 6 (W2).
- The gravity component of joint 2 (S2) depends on the following joint angle coordinates: (q_2, $q_2 - q_3$, q_4, q_5).
- The gravity component of joints 3 (E1), 4 (E2) and 5 (W1) only depend on the following coordinates: ($q_2 - q_3$, q_4, q_5).

The gravity vector is used to estimate the friction torque levels.

4.3.2 Friction estimation

The friction model is one of the elements of the dynamics model based on experimental results (identification). The non-parametric model with a velocity based look-up table has been chosen (equation 3.19), based on the following observations

- The model has to be exact for the entire range from small (0.05 rad/s) to high (≥ 1 rad/s) joint angle velocities.
- An attempt has been made to perform the parametric estimation presented in [4.2], but it does not generate a correct (stable) model for the available PA10-6CE manipulator.
- The estimation algorithm employed in [4.5] is mathematically complex and considering the equation used, it may fail at very low velocities. Moreover, it has only been presented in two dimensions, and a full 6-DOF estimation may be difficult and dependant on the goodness of fit of the dynamic model (inertia, Coriolis, centrifugal, gravity).
- The model derived in [4.3, 4.4] was mainly designed for low feed-back gain control in low velocities, and requires additional experimental identification of transmission compliance with varying load.

The approach taken was to create a novel nonparametric model, based on measurements taken from the manipulator while moving single joints at particular constant velocities. Such experiments allow for a significant simplification in the dynamics equation (3.9) leading to

$$\mathbf{g}(\mathbf{q}) + \mathbf{F} = \mathbf{F}_e + \mathbf{u}. \tag{4.2}$$

The joints of the PA10-6CE manipulator are driven by AC motors and the torque is transmitted by harmonic drive gearing. This transmission causes additional dynamics to the

[2]For experimental/simulation purposes g=9.81$\left[\frac{m}{s^2}\right]$

drive and may be position dependent. In order to measure and estimate the friction several assumptions have been made:

- the friction torque is only velocity dependent,
- the compliance of the wave generator (mentioned in [4.4]) is neglected,
- at low velocities the friction torque is more irregular and requires a more dense grid of measurements,
- the friction is not symmetric – torques for negative velocities are estimated separately,
- for velocity equal to zero the friction torque is assumed to be zero (to assure stability of the model).
- the torque required to initialise motion is estimated using the lowest possible velocity that can be commanded to the manipulator - effectively the joint starts and stops (which may be accounted for by the stick - slip effect).

The method employed to estimate the friction can be summarised in the following steps:

1. Perform a warm-up session for the selected joint, by commanding it to move along a sinusoidal trajectory (across its range) for 10 minutes.
2. Measure the torques applied to the joints by the Mitsubishi's own controller (in velocity control mode) at certain joint angle velocities[3] (over the available[4] range of joint motion angles),
3. Select only the part of motion where velocity was constant (ignore acceleration and deceleration periods).
4. Check if average velocity measured in the selected region is within 0.05 rad/s of the commanded velocity, and if not - discard the experiment. This is to eliminate measurements at higher velocities, for some joints the controller was not able to drive the arm at joint angle velocities above 1 rad/s.
5. Smooth the obtained torque measurements with a low pass filter (long moving average)

$$\tilde{\tau}_i = \sum_{j=i-n}^{i+n} \tau_j \tag{4.3}$$

where $2n+1 = \frac{1}{40}N$ is the filter length and N is the number of samples in the experiment.

[3]The torques are measured for the following velocities: $+/-$`[0.01:0.005:0.095 0.10:0.01:0.19 0.20:0.05:0.95]` rad/s and `[1:00.10:1.50]` for joints S1 and S2, `[1:00.10:2.40]` for joint E1, `[1:00.20:5.60]` for joints E2 and W2 and `[1:00.20:4.60]` for joint W1

[4]limited due to safety reasons

6. Discard the first and last 10% of smoothed samples (due to possible boundary effects of the filter).

7. Subtract the computed gravity torque from the smoothed value (at each sample point) $\hat{\tilde{\tau}}_i = \tilde{\tau}_i - \mathbf{g}_i(\mathbf{q})$,

8. Calculate the average of all residual torque values $\hat{\tilde{\tau}}_i$. This should minimise the influence of any unmodelled dynamics of the harmonic drives and possible gravity modelling errors.

9. Assume this torque to be the desired estimate (for velocity equal to the mean of velocities measured over the whole selected range) $\hat{F}_v = \frac{1}{\bar{N}_v} \sum_{i=1}^{i=\bar{N}_v} \hat{\tilde{\tau}}_i$, where $\bar{N}_v$ is the final number of measurements used to estimate the friction torque at velocity v.

To obtain a continuous friction estimation function, the measurements are interpolated within the measured range and extrapolated (based on two subsequent boundary measurements) out with the measured range (Equation 3.19). It is possible to apply some smoothing to the obtained friction estimates, to improve the robustness of the model, however this has not been found necessary. It has been found that the applied torque (as returned by the servo controller) carries an additional (unmodelled) dynamics for joints affected by gravity. The selected method of estimation with averaging of measured torques is expected to be least affected by this effect. However, it is worth pointing out that, for a more in-depth analysis, the addition of harmonic gear dynamics (with flex-spline compliance) may need to be added.
The plots of friction torques for all joints are shown in fig. 4.4. The following observations can be made:

- The Stribeck effect of *boundary lubrication* can be observed for joints S1, E1, E2 and W1.

- For joints S1 and E1 and E2 the friction decreases up to 0.1 rad/s joint angle velocity, but for joint W1 this threshold is much lower.

- For joints S2 and W2 the friction is approximately linear in velocity.

- Differences between positive and negative velocity can be observed, highest for joint W1 – this could be attributed to the presence of an additional transmission element - a tooth-belt.

- It can be noticed that at low velocities the exact smooth Stribeck curve is not always followed, which shows the benefits of the non-parametric model.

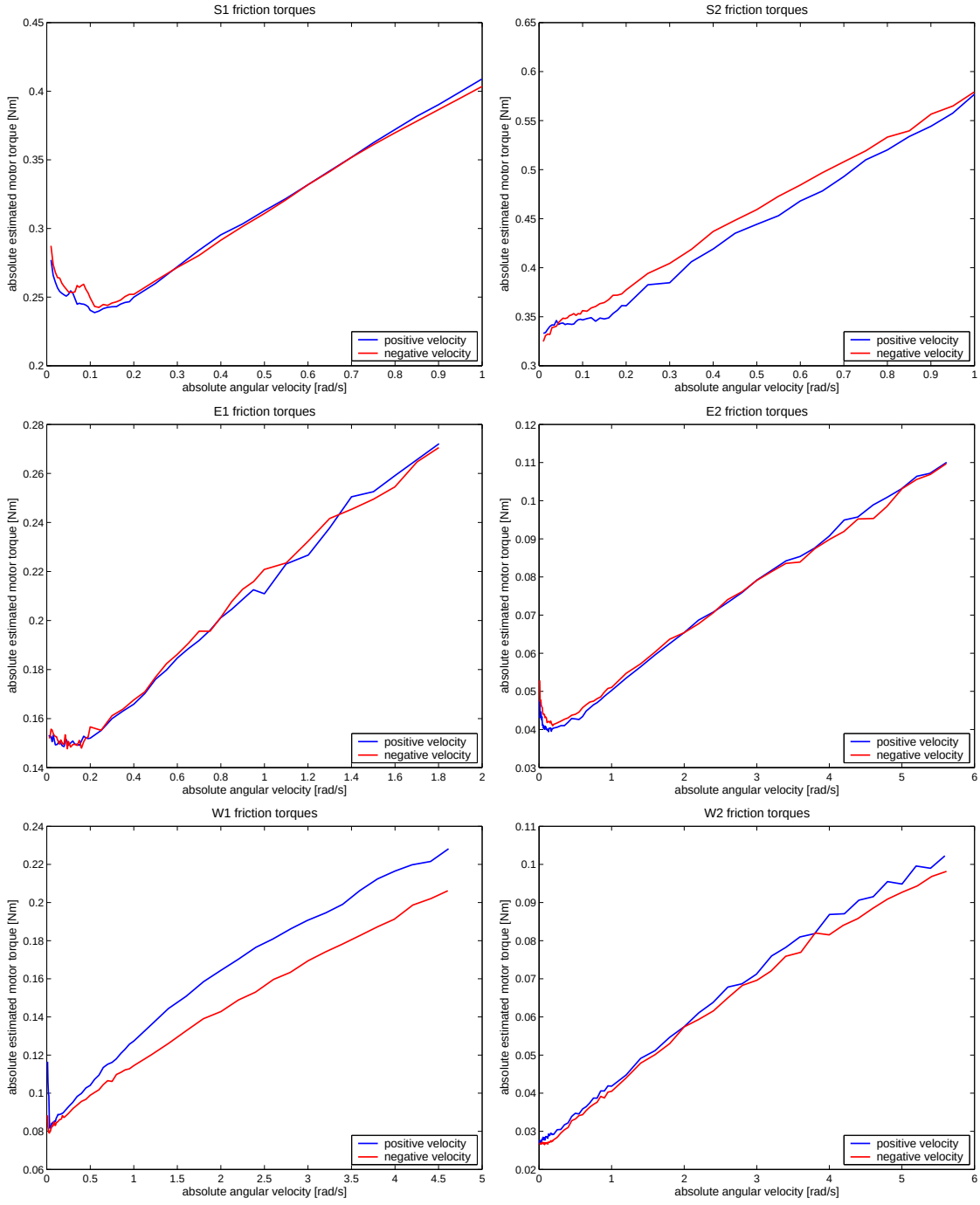


Figure 4.4: Friction estimates (negative velocity curves are plotted in absolute torque / velocity values).

4.3.3 Inertial, Coriolis and Centrifugal terms

To obtain a full dynamic model of a robotic manipulator, additional parameters are required. These were not provided by the manufacturer, therefore certain assumptions are made, to simplify the calculations and obtain reliable results.

- The motor inertias cannot be obtained analytically and need to be be identified.
- A geometric model for each link has been created, to provide analytic definition of volume.
- The density of each link is assumed to be constant over analytic regions within a link. The boundaries of these regions are based on the schematic mass distribution available from figure 4.2.
- The centre of mass of each link geometric model with density regions has to be equal to the centre of mass from table 4.2. The densities are derived from this assumption.

4.3.3.1 Link models

The assumed geometric models of links are presented in figures 4.5 to 4.11 and the parameter values are given in tables 4.5 and 4.6.

Parameter	value [m]	Parameter	value [m]	Parameter	value [m]
b_1	0.117	b_2	0.05	b_3	0.052
b_4	0.151	b_5	0.07	l_1	0.13
l_2	0.022	l_3	0.17	l_4	0.05
l_5	0.23	l_{5a}	0.1	l_6	0.21
l_7	0.18	l_8	0.11	l_9	0.1
d_1	0.04	d_2	0.123	d_3	0.15
d_4	0.03	r_1	0.05	r_2	0.1
r_3	0.05	r_{3a}	0.052	r_4	0.035
r_5	0.045	r_6	0.0125	r_7	0.015
h_1	0.11	h_2	0.095	h_3	0.085
h_4	0.09				

Table 4.5: Summary of PA10-6CE physical parameters used to compute the link inertias.

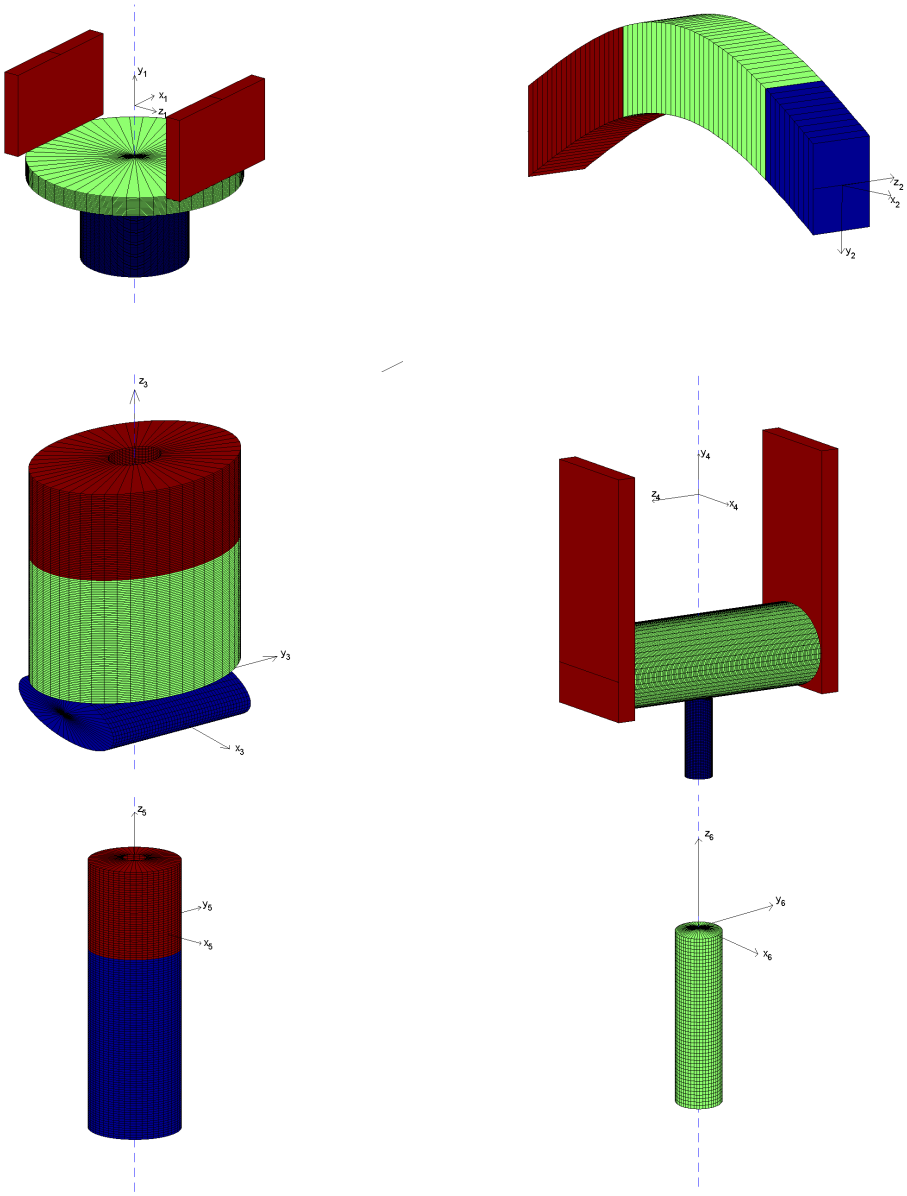


Figure 4.5: Geometric link models for PA10-6CE.

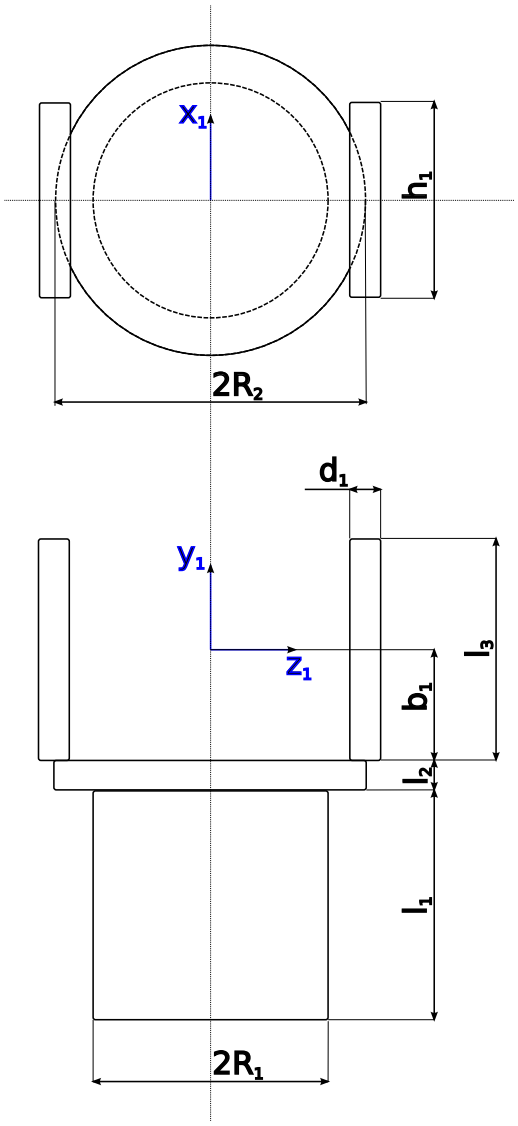


Figure 4.6: Projections with dimensions of link S1.

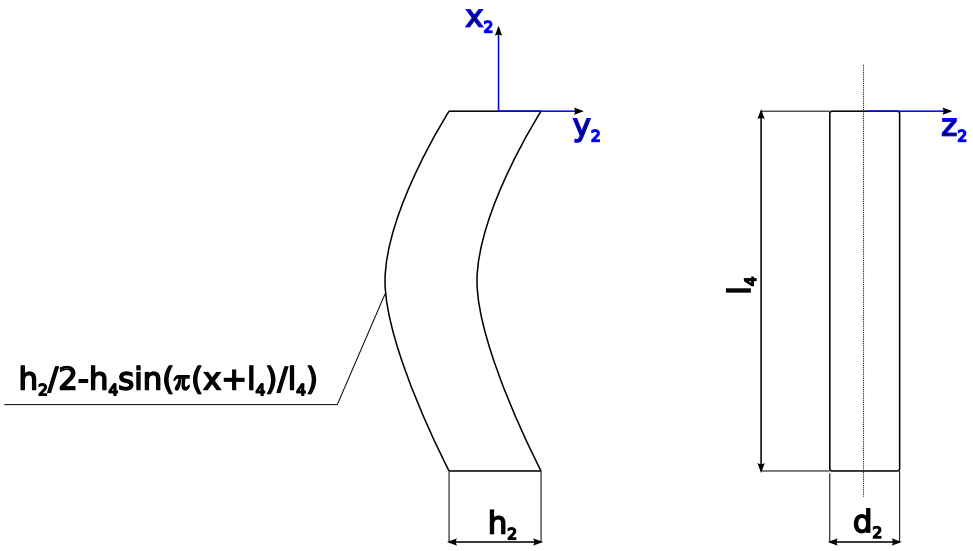


Figure 4.7: Projections with dimensions of link S2.

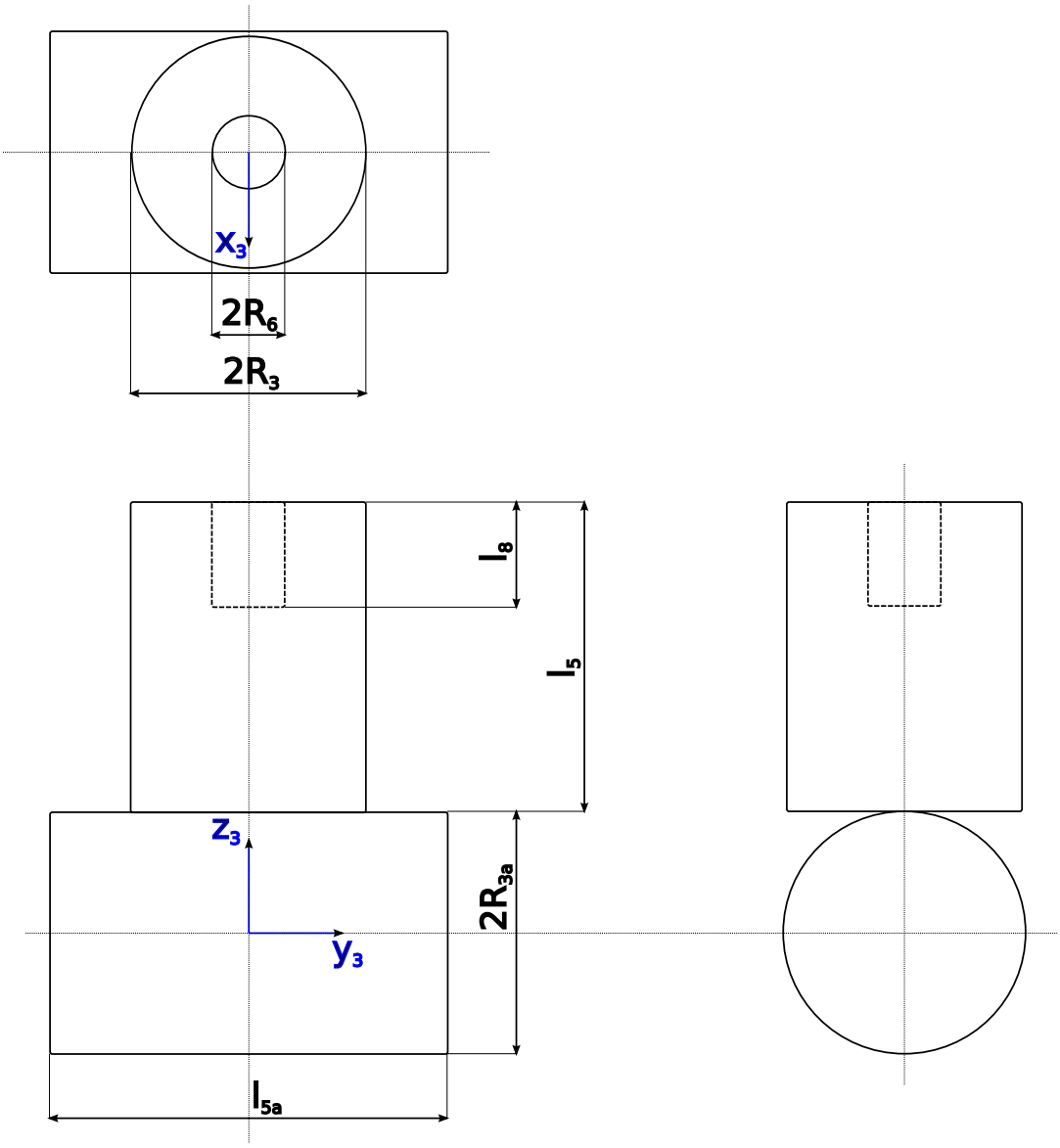


Figure 4.8: Projections with dimensions of link E1.

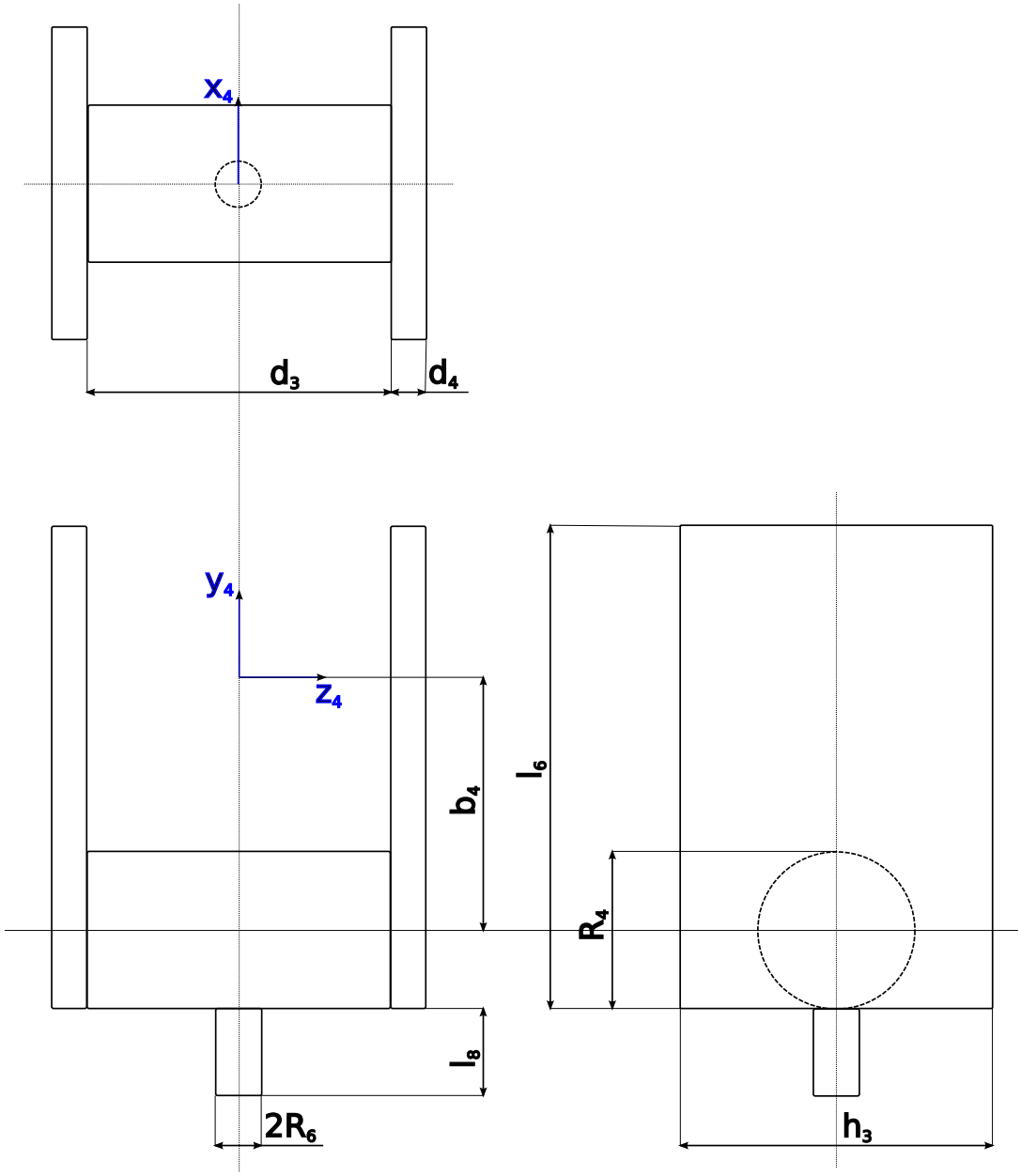


Figure 4.9: Projections with dimensions of link E2.

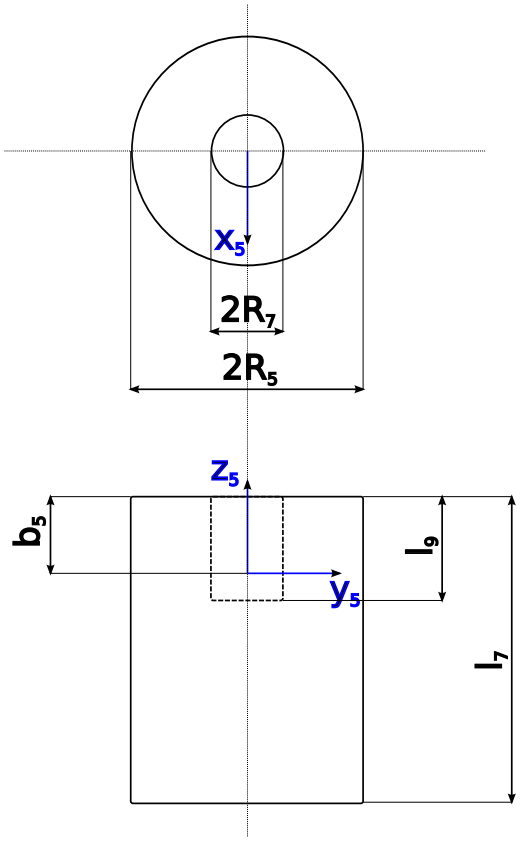


Figure 4.10: Projections with dimensions of link W1.

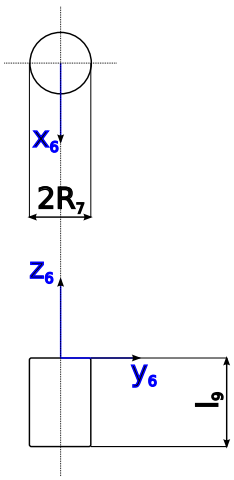


Figure 4.11: Projections with dimensions of link W2.

Link volumes (expressed in local coordinate systems - Cartesian or cylindrical for simpler representation):

$$
\begin{aligned}
S1 = &\{(r,\phi,y)|r \in [0,r_1], \phi \in [-\pi,\pi], y \in [-b_1 - l_1, -b_1], \quad \text{density: } \rho_1\} \cup \\
&\{(r,\phi,y)|r \in [0,r_2], \phi \in [-\pi,\pi], y \in [-b_1, -b_1 + l_2], \quad \text{density: } \rho_{2a}\} \cup \\
&\left\{(x,y,z)|x \in \left[-\frac{h_1}{2}, \frac{h_1}{2}\right], y \in [-b_1 + l_2, -b_1 + l_2 + l_3],\right. \\
&\left. z \in \left[-r_2 - \frac{d_1}{2}, -r_2 + \frac{d_1}{2}\right], \quad \text{density: } \rho_{2b}\right\} \cup \\
&\left\{(x,y,z)|x \in \left[-\frac{h_1}{2}, \frac{h_1}{2}\right], y \in [-b_1 + l_2, -b_1 + l_2 + l_3],\right. \\
&\left. z \in \left[r_2 - \frac{d_1}{2}, r_2 + \frac{d_1}{2}\right], \quad \text{density: } \rho_{2b}\right\} \\
S2 = &\left\{(x,y,z)|x \in \left[-l_4, -\frac{5}{6}l_4\right],\right. \\
&y \in \left[-\frac{h_2}{2} - h_4 \sin\left(\frac{x + l_4}{l_4}\pi\right), \frac{h_2}{2} - h_4 \sin\left(\frac{x + l_4}{l_4}\pi\right)\right], \\
&\left. z \in \left[-\frac{d_2}{2}, \frac{d_2}{2}\right], \quad \text{density: } \rho_{3a}\right\} \cup \\
&\left\{(x,y,z)|x \in \left[-\frac{5}{6}l_4, -\frac{1}{8}l_4\right],\right. \\
&y \in \left[-\frac{h_2}{2} - h_4 \sin\left(\frac{x + l_4}{l_4}\pi\right), \frac{h_2}{2} - h_4 \sin\left(\frac{x + l_4}{l_4}\pi\right)\right], \\
&\left. z \in \left[-\frac{d_2}{2}, \frac{d_2}{2}\right], \quad \text{density: } \rho_{3b}\right\} \cup \\
&\left\{(x,y,z)|x \in \left[-\frac{1}{8}l_4, 0\right],\right. \\
&y \in \left[-\frac{h_2}{2} - h_4 \sin\left(\frac{x + l_4}{l_4}\pi\right), \frac{h_2}{2} - h_4 \sin\left(\frac{x + l_4}{l_4}\pi\right)\right], \\
&\left. z \in \left[-\frac{d_2}{2}, \frac{d_2}{2}\right], \quad \text{density: } \rho_{3c}\right\}
\end{aligned}
$$

$$
\begin{aligned}
E1 = &\left\{(r,\phi,y)|r\in[0,r_{3a}],\phi\in[-\pi,\pi],y\in\left[-\frac{l_{5a}}{2},\frac{l_{5a}}{2}\right],\quad \text{density: } \rho_{4a}\right\}\cup\\
&\{(r,\phi,z)|r\in[0,r_3],\phi\in[-\pi,\pi],z\in[b_3,b_3+l_5-l_8],\quad \text{density: } \rho_{4b}\}\cup\\
&\{(r,\phi,z)|r\in[r_6,r_3],\phi\in[-\pi,\pi],z\in[b_3+l_5-l_8,b_3+l_5],\quad \text{density: } \rho_{4c}\}\\
E2 = &\{(r,\phi,y)|r\in[0,r_6],\phi\in[-\pi,\pi],y\in[-b_4-r_4-l_8,-b_5-r_4],\quad \text{density: } \rho_5\}\cup\\
&\left\{(x,y,z)|x\in[-\sqrt{r_4^2-(y+b_4)^2},\sqrt{r_4^2-(y+b_4)^2}],\right.\\
&\left.y\in[-b_4-r_4,-b_4+r_4],z\in\left[-\frac{d_3}{2},\frac{d_3}{2}\right],\quad \text{density: } \rho_{6a}\right\}\cup\\
&\left\{(x,y,z)|x\in\left[-\frac{h_3}{2},\frac{h_3}{2}\right],y\in[-b_4-r_4,-b_4-r_4+l_6],\right.\\
&\left.z\in\left[-\frac{d_3}{2}-d_4,-\frac{d_3}{2}\right],\quad \text{density: } \rho_{6b}\right\}\cup\\
&\left\{(x,y,z)|x\in\left[-\frac{h_3}{2},\frac{h_3}{2}\right],y\in[-b_4-r_4,-b_4-r_4+l_6],\right.\\
&\left.z\in\left[\frac{d_3}{2},\frac{d_3}{2}+d_4\right],\quad \text{density: } \rho_{6b}\right\}\\
W1 = &\{(r,\phi,z)|r\in[0,r_5],\phi\in[-\pi,\pi],z\in[b_5-l_7,b_5-l_9],\quad \text{density: } \rho_{7a}\}\cup\\
&\{(r,\phi,z)|r\in[r_7,r_5],\phi\in[-\pi,\pi],z\in[b_5-l_9,b_5],\quad \text{density: } \rho_{7b}\}\\
W2 = &\{(r,\phi,z)|r\in[0,r_7],\phi\in[-\pi,\pi],z\in[-l_9,0],\quad \text{density: } \rho_8\}
\end{aligned}
$$

4.3.3.2 Density distribution

The following equations were used for each link, to obtain a mass distribution, that satisfies two requirements[5].

- mass of link equal to the one provided on Figure 4.2
- centre of mass of link equal to the one provided in Table 4.2

The following two steps were applied

1. determine the average link density

$$\rho_i = \frac{m_i}{\mathcal{V}_i},$$

   where m_i is the mass of the link, and $\mathcal{V}_i$ is the volume of the link.

2. Equate the mass

$$\sum_j \int_{\mathcal{V}_{ij}} \rho_{ij} d\mathbf{r} = m_i$$

[5]Uniform density has been assumed for links that have a single mass element, and in such cases the centre of mass could not be exactly matched.

where $\mathcal{V}_{ij}$ are sections of volume of the j-th section of the i-th link corresponding to different densities $\rho_{ij} = c_{ij}\rho_i$, c_{ij} being weighting coefficients for the density distribution.

3. Equate the centre of mass
$$\sum_j \int_{\mathcal{V}_{ij}} \mathbf{r}\rho_{ij} d\mathbf{r} = \mathbf{R}_i$$
where $\mathbf{R}_i$ is the centre of mass of i-th link from equation 4.2.

4. The two equations from step 2 and 3 are solved simultaneously for c_{ij}, separately for each link.

The obtained density distribution (presented in table 4.6) is used to derive the link inertias. The centres of mass calculated with the obtained density distribution are presented in table 4.7.

Parameter	value $[\frac{kg}{m^3}]$
ρ_1	2138.64
ρ_2	728.913
ρ_3	1412.5
ρ_{4a}	3367.76
ρ_4	1204.21
ρ_5	935.035
ρ_6	1486.48
ρ_7	2667.29
ρ_8	2219.44

Table 4.6: Summary of PA10-6CE link densities calculated using the link model and physical parameters.

Joint	centre of mass [6] $\mathbf{R}_i = (r_{xi}, r_{yi}, r_{zi})[m]$	mass $m_i[kg]$
S1	(0 , -0.15705 , 0)	9.29
S2	(-0.321, -0.03 , 0)	12.43
E1	(0 , 0 , 0.0484568)	4.86
E2	(0 , -0.107195, 0)	3.08
W1	(0 , 0 , -0.042)	2.07
W2	(0 , 0 , -0.055)	1.05

Table 4.7: Mass points obtained from link volumes and density distribution (the link inertias are based on these values, however the gravity model is based on the values provided by the manufacturer.)

4.3.3.3 Link inertias

Using the mass distribution and volume, the inertias of links can be derived, from equation 3.11, where the integration over mass can now be computed as an iterated integral. The obtained numerical results presented in equation 4.4 have been used to derive the Inertia, Coriolis and Centrifugal terms of the manipulator dynamics.

$$
\begin{aligned}
I_1 &= \begin{bmatrix} 0.329 & 0 & 0 \\ 0 & 0.0476 & 0 \\ 0 & 0 & 0.313 \end{bmatrix} \\
I_2 &= \begin{bmatrix} 0.0569 & -0.115 & 0 \\ -0.115 & 1.734 & 0 \\ 0 & 0 & 1.738 \end{bmatrix} \\
I_3 &= \begin{bmatrix} 0.0485 & 0 & 0 \\ 0 & 0.0471 & 0 \\ 0 & 0 & 0.00817 \end{bmatrix} \\
I_4 &= \begin{bmatrix} 0.0692 & 0 & 0 \\ 0 & 0.0139 & 0 \\ 0 & 0 & 0.05882 \end{bmatrix} \\
I_5 &= \begin{bmatrix} 0.0102 & 0 & 0 \\ 0 & 0.0102 & 0 \\ 0 & 0 & 0.00281 \end{bmatrix} \\
I_6 &= \begin{bmatrix} 0.00459 & 0 & 0 \\ 0 & 0.00459 & 0 \\ 0 & 0 & 0.000719 \end{bmatrix}
\end{aligned}
\tag{4.4}
$$

All matrices are expressed in $kg \cdot m^2$.

4.3.3.4 Motor inertias

The motor masses are not provided by the manufacturer, the respective inertias have to be estimated based on experimental data. The following procedure has been used to obtain these parameters:

1. Perform an experiment with the PA10 manipulator, where a torque step input is applied to to a given joint, with all remaining joints stopped. Gather the response data.[7] The equation of motion simplifies to the following form in such a case

$$(\tilde{I}_i + k^2 I_{m_i})\ddot{q}_i + g_i(q_i) + F_i(\dot{q}_i) = \tau_i \tag{4.5}$$

[7] For joints affected by gravity, the step torque input has been combined with gravity compensation.

where $\tilde{I}_i$ is the inertia of the rigid body composed of all locked parts of the manipulator that are actuated by motor i, $k = 50$ for all joints is the transition ratio, I_{m_i} is the motor inertia for joint i. The Coriolis and Centrifugal terms vanish for a single moving joint. Therefore the following control torque is applied

$$\tau_i = g_i(q_i) + \tau_{\mathbf{1}_i} \cdot \mathbf{1}(t \geq t_0 \wedge t < t_1) \tag{4.6}$$

where $\tau_{\mathbf{1}_i}$ is the step torque level presented in table 4.8. This control cancels out the gravity element of the dynamics, leaving inertias and friction.

2. Simulate the dynamics of equation 4.5 varying the motor inertia I_{m_i}. The motor inertia can be quickly found, as it characterises the onset/offset rates of the velocity (any changes in the steady state should not be considered, as this may be due to unmodelled dynamics).

3. Tune the motor inertia to obtain closest match between simulation and experiment results (measured by MISE of angle and velocity).

$\tau_{\mathbf{1}_i}$ [Nm]	τ_{s_i} [Nm]	ω_i [rad/s]
$\frac{4.64}{10.0}$	0.55	$\frac{2\pi}{5}$
$\frac{4.64}{8.57}$	0.00	$\frac{2\pi}{4.5}$
$\frac{2.00}{8.33}$	0.00	$\frac{2\pi}{4}$
$\frac{0.29}{5.00}$	0.08	$\frac{2\pi}{4.2}$
$\frac{0.29}{2.31}$	0.00	$\frac{2\pi}{3}$
$\frac{0.29}{6.00}$	0.08	$\frac{2\pi}{4.1}$

Table 4.8: Trajectory parameters used in experiments and simulations (the step torque levels ($\tau_{\mathbf{1}_i}$) are expressed as fractions of rated motor torque, with the denominator chosen to assure that the step command does not saturate neither the control torque nor the velocity processed by the servo control unit.)

The joints have been grouped into four classes (S1,S2), (E1), (E2) and (W1,W2) – based on the manufacturer specified rated torques and available knowledge about the joint design. The motors/transmission are assumed to be identical within each class, therefore a single motor inertia has been chosen. The obtained results[8] are presented in table 4.9. From the obtained inertias, the motor masses (m_{m_i}) can be estimated assuming that the motor is a uniform cylinder of a given radius (r_{m_i}).
It can be noticed, that the motor mass for joint E2 stands out from the rest. Also, the estimated motor masses are well below the mass of the link element containing the motor, and therefore can be assumed to be acceptable. The estimated motor inertias also include the inertia of the wave generator of the harmonic drive.

[8]The motor inertias depend on the exactness of the underlying dynamic model.

Joint	$k^2 I_{m_i}$ [kg m^2]	r_{m_i} [m]	m_{m_i} [kg]
S1	2.5	0.04	0.625
S2	2.5	0.04	0.625
E1	0.8	0.03	0.3556
E2	0.06	0.025	0.0384
W1	0.16	0.025	0.1024
W2	0.16	0.025	0.1024

Table 4.9: Summary of PA10-6CE motor inertias and mass estimates.

Experimental and simulation results of the final step of choosing the motor inertia are presented in figures 4.12, 4.13, 4.14 and 4.15. Simulation 1 was performed with $I_{m_i} = 0$, and simulation 2 with the presented values of the parameter. TVA stands for (torque [Nm], velocity [rad/s], angle [rad]), the motor torque is plotted ($\frac{1}{50}$ of actual joint torque).
It can be observed that the experimental and simulation graphs do not match exactly. The motor inertias were selected to match the on-set and off-set periods in the step response, however due to modelling uncertainties, there are discrepancies. The following model deficiencies have been identified:

- lack of transmission flexibility model has a significant effect on joints with non-zero gravity torques (S2, E1, W1),
- the identified friction coefficients depend on ambient temperature and can be affected by different joint/lubricant temperature,
- there is a possibility of link inertia errors, due to assumed simplification of uniform mass distribution.

It can be further noticed on some of the step torque plots (e.g. joint S2 in figure 4.12 around 1s time) that a spike in velocity measurement (green line) is present. This was caused by a bug in the early version of the manipulator control software, it only affected the log file and not the controller. It has been fixed in a later version. Moreover, considering the plots for joints S2 and E1 in figures 4.12 and 4.14, a certain high frequency sinusoidal component can be noticed. This could be attributed to the higher speed motor and wave generator being affected by irregular friction effects. Furthermore, when analysing the experimental torque plots (e.g. joint S1 in figure 4.12), it can be seen that there is a certain level of noise in the level. This torque reading is provided by the servo control unit and is presumably based on the current measurement in the control amplifier circuit. The noise can originate form both electrical (power supply noise) and mechanical (varying friction levels on the high speed port of the harmonic drive) sources.

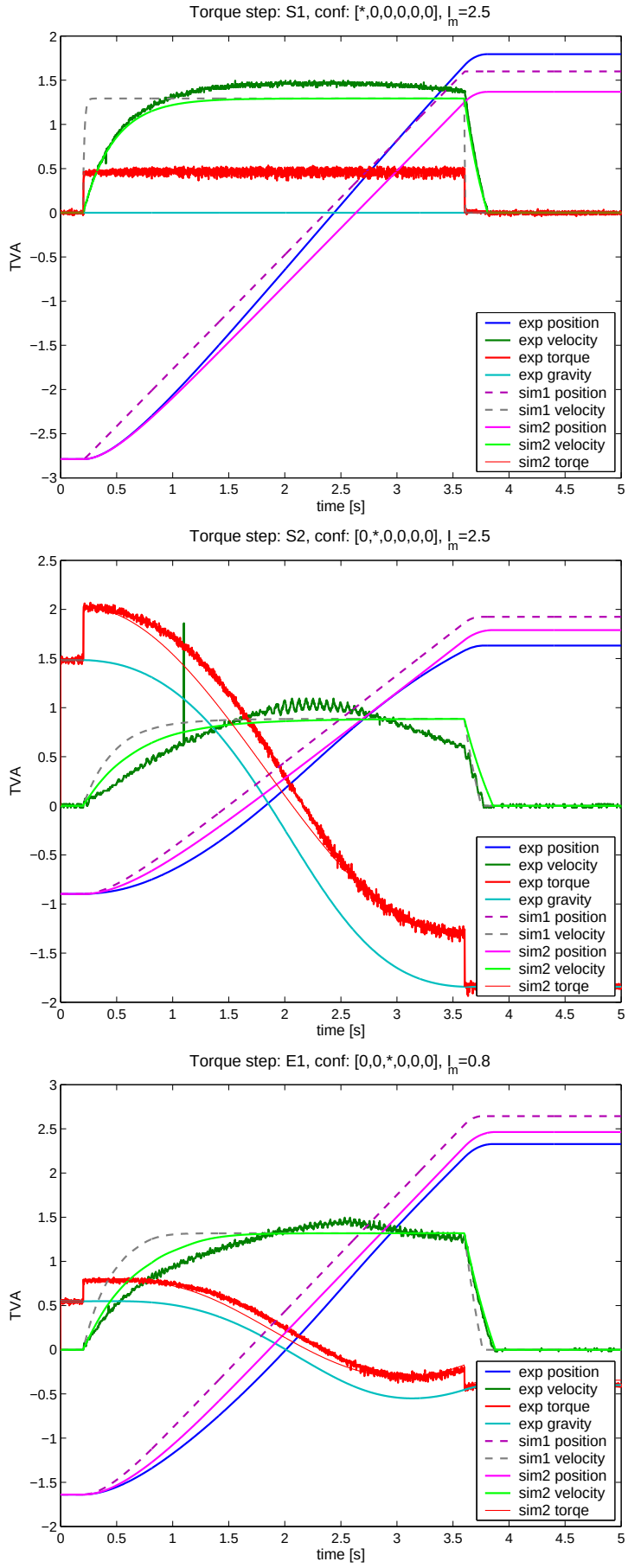


Figure 4.12: Torque step experimental and simulations comparisons (used to estimate motor inertias), joints: S1, S2, E1.

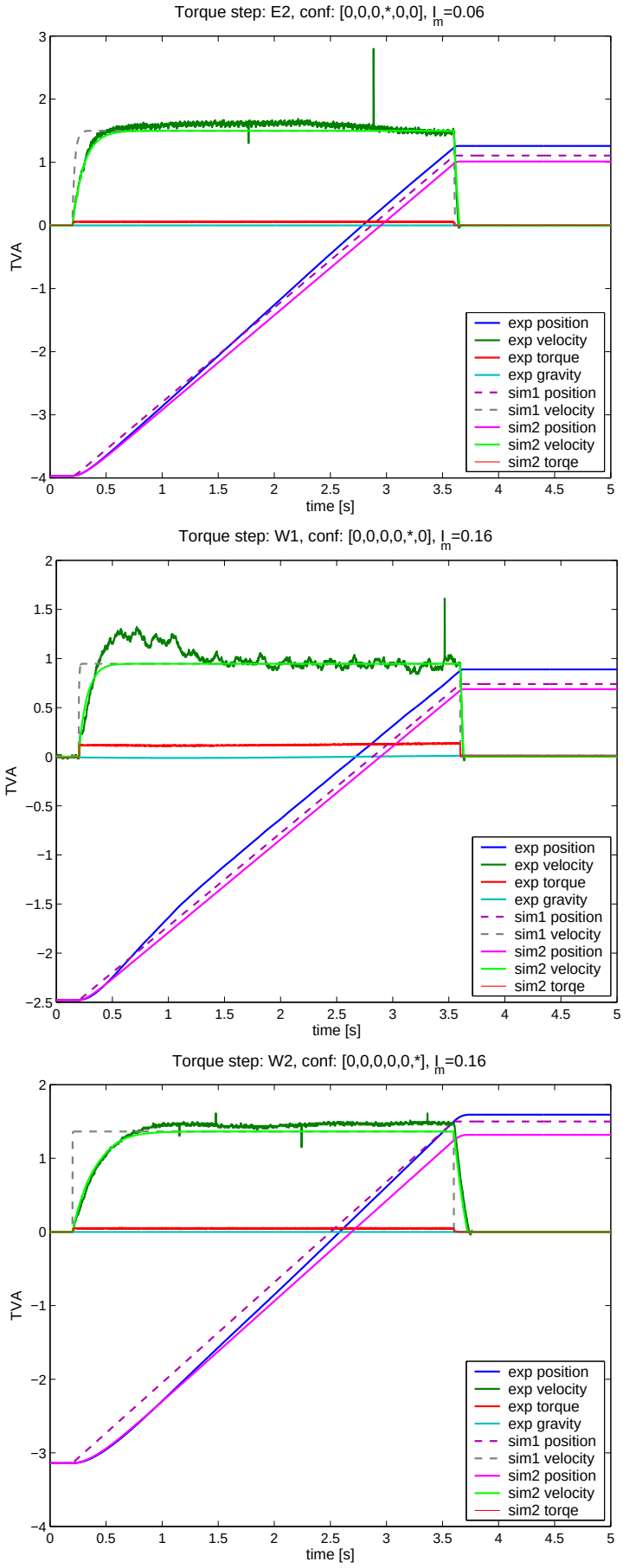


Figure 4.13: Torque step experimental and simulations comparisons (used to estimate motor inertias), joints: E2, W1, W2.

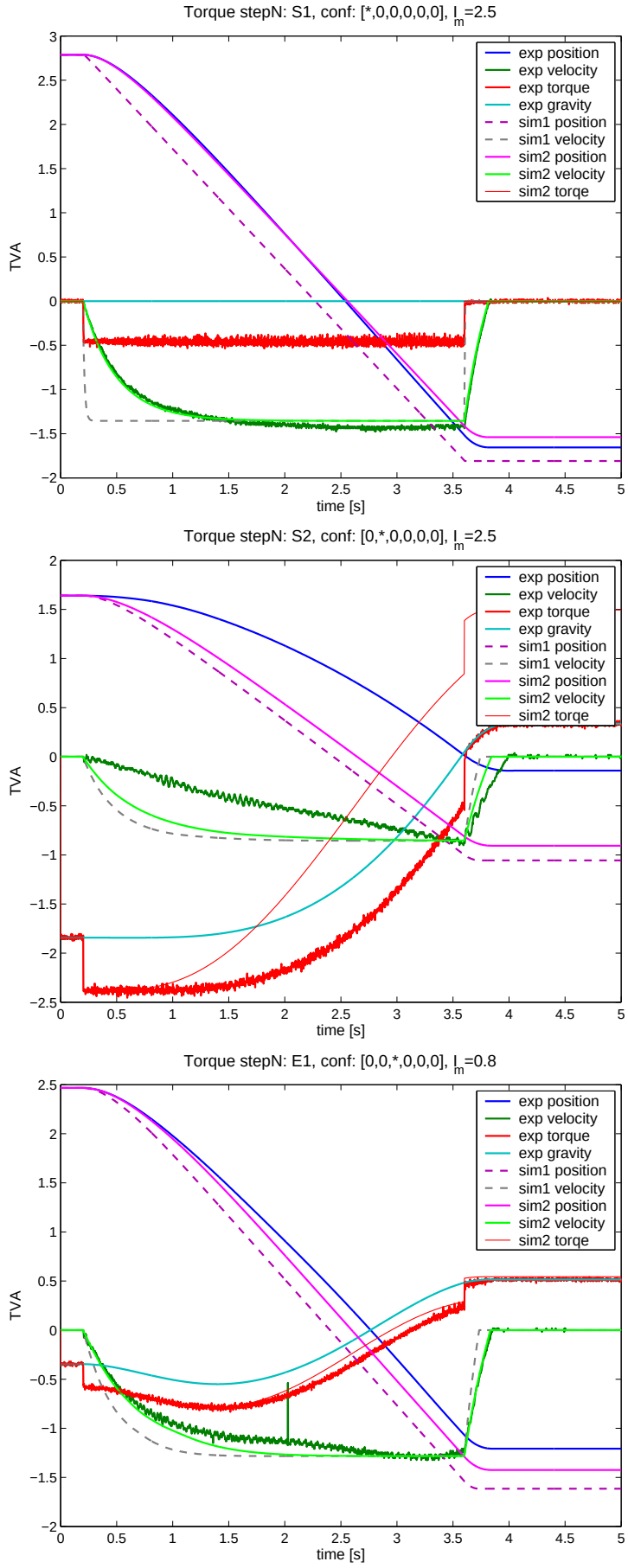


Figure 4.14: Negative torque step experimental and simulations comparisons (used to estimate motor inertias), joints: S1, S2, E1.

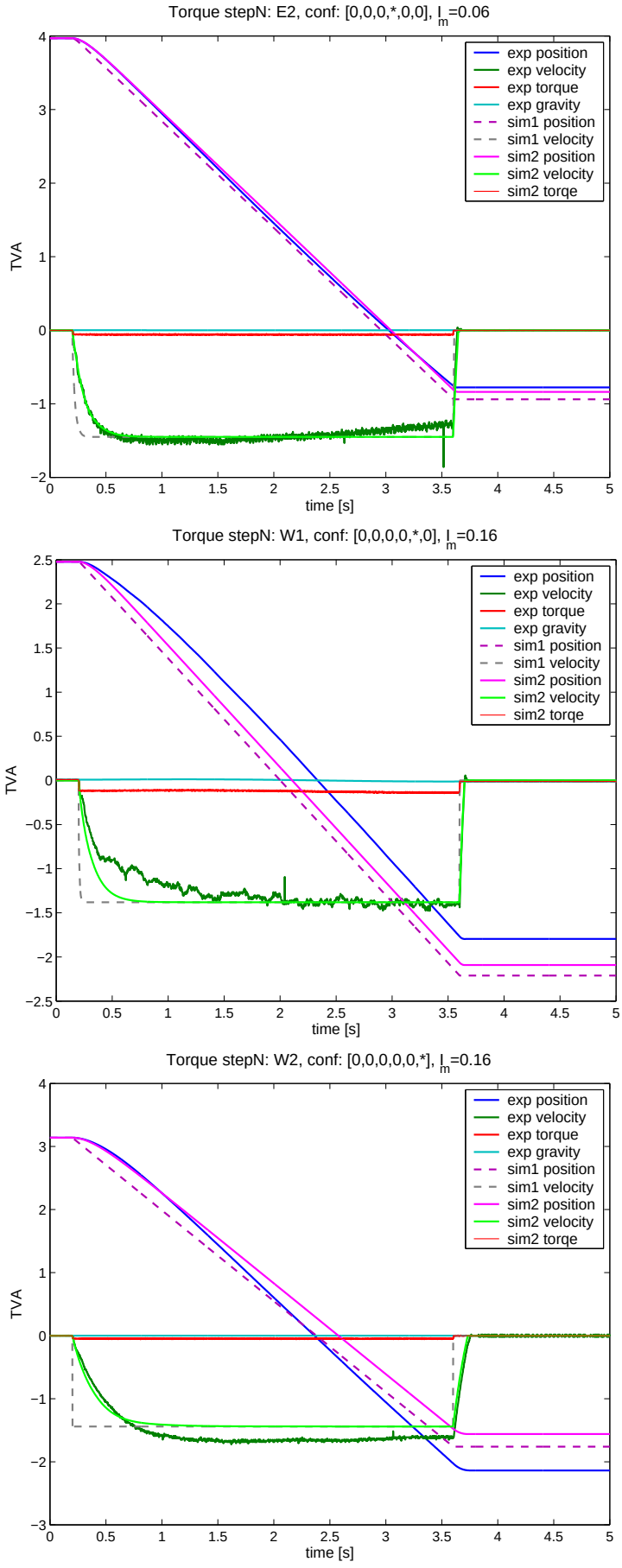


Figure 4.15: Negative torque step experimental and simulations comparisons (used to estimate motor inertias), joints: E2, W1, W2.

4.3.4 Partial verification of the model

Based on the conclusions of the *step torque* experiments it can be concluded that a quantitative evaluation of the model using open-loop control may not deliver entirely satisfactory results. However, the obtained model can be partially graphically verified by performing an open-loop control task for joints not affected by gravity. This is a simple experiment, where the motor is excited using a sinusoidal torque input and the joint position/velocity is recorded. These can be compared to the results of a Matlab simulation using the obtained model with an identical input signal. The following control signals are applied

$$\mathbf{u}(t) = [\tau_{s_i} \sin(\omega_i t)]_{i=1}^{6}, \tag{4.7}$$

where parameters τ_{s_i} and ω_i are defined in table 4.8.
The results of the comparison are presented in figure 4.16.
The following observations can be made:

- The friction model is valid and joints stop when the torque level is below the static friction.
- Due to the non-symmetric friction model the angles do not necessarily return to zero and some drift can be observed (although the control torque is symmetric).
- The experimental and simulation graphs are aligned for joints S1 and W2.
- The simulation diverges significantly from experimental results for joint E2. This can be attributed to friction and inertia modelling errors and lack of transmission flexibility error.

4.4 Summary

In this chapter a full dynamic model of the PA10 robotic manipulator has been derived. The gravity term is based on the mass distribution schematic provided by the manufacturer. The friction torque has been identified based on constant velocity experiments and is used in an interpolated look-up table model. The link inertias have been obtained analytically using geometric approximations of the link shapes and matching these mass centres to the manufacturer specification. Motor inetrias have been identified using step torque input. All inertia parameters have been used to derive the inertia and Coriolis + centrifugal terms. Additionally a simple open loop verification has been performed for the links not affected by gravity using sinusoidal torque input and comparing the simulation and experimental results.
This verification method allows a check of inertia and friction of a single joint. However, multiple joints need to be excited to verify the full dynamics of the manipulator. The main disadvantage of this comparison is that, due to the model uncertainties (leading to poor results repeatability), a quantitative measure of the model effectiveness cannot be made.

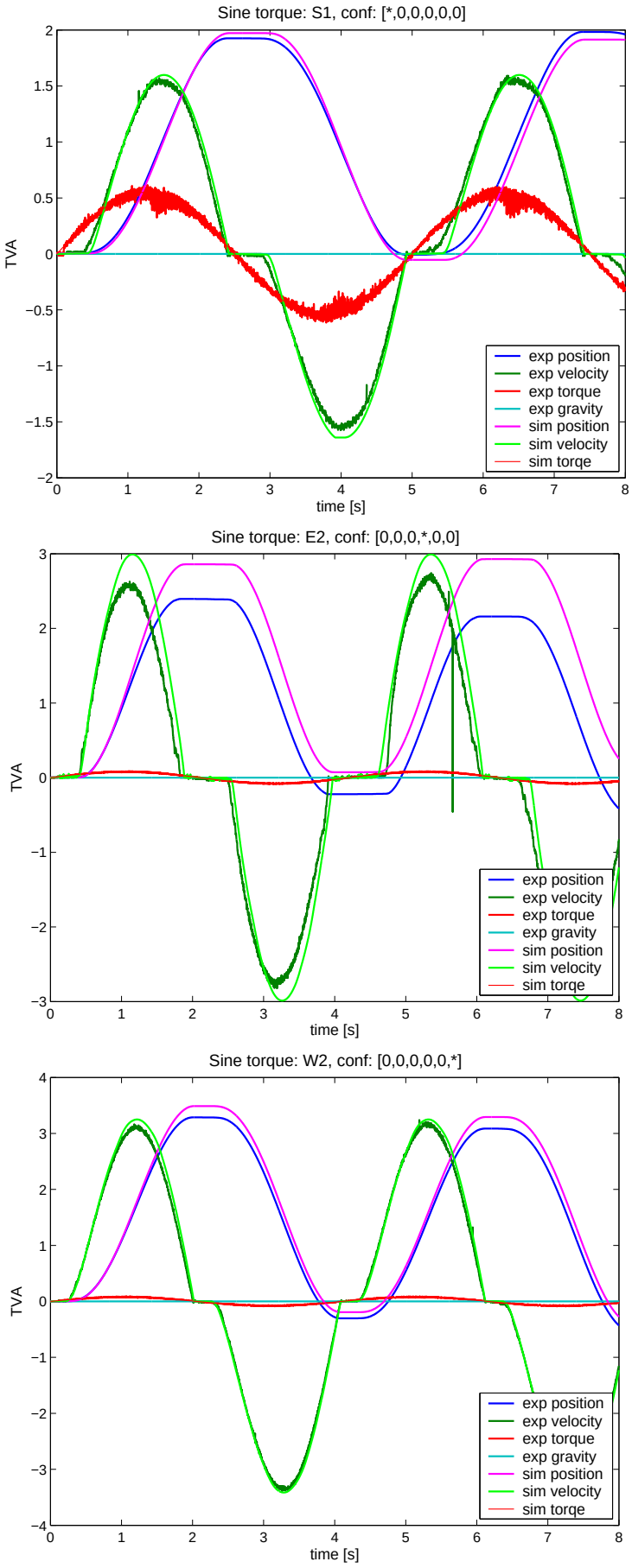


Figure 4.16: Sine torque experimental and simulations comparisons.

Chapter 5

Joint angle control of the PA10-6CE manipulator

The notion of a control system originates from the multidisciplinary field of control theory. Technological development over the past decades required some *intelligent* devices that could drive a plant. To accomplish this a device is required which provides signals (in a form understandable to the plant) that will force it to the desired states. Depending on the utilisation of data from the plant output, two controller groups can be described (figures 5.1 and 5.2)

- open loop,
- closed loop.

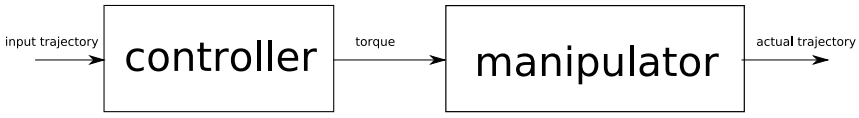


Figure 5.1: Schematic of an open-loop controller.

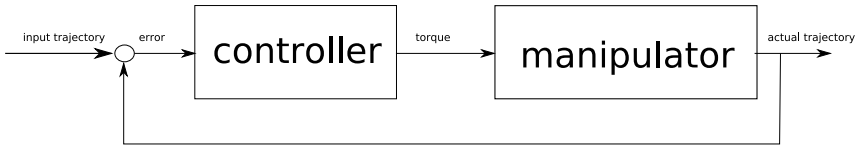


Figure 5.2: Schematic of a closed-loop controller.

An open-loop controller sends control signals 'blindly' without having any information about the state of the plant. Therefore, such a controller can only be used in ideal situations, where random disturbances are not present, and the knowledge about the dynamics of the driven plant is complete or irrelevant. However, in many practical cases the designer of the control system (even if a complete model is available) cannot always foresee the

random forces that may affect the plant, and therefore uses the output (or sometimes even the state of the plant) to improve accuracy, stability and system transient response. The feedback loop consists of three main elements: the plant, a sensor measuring the output and a controller processing the difference between the measurement and a given reference signal to compute and apply a new command to the plant. This is a universal configuration with advantages and drawbacks. The main positive outcome of feeding back the last measurement is the independence to environmental changes, as the system can simply modify the command signal to achieve the desired task. The designer has to be aware of the possible consequences of time delay between the measurement and command. This often leads to various unwanted outcomes like oscillation or instability. Therefore, the controller has to be designed with a good consideration of the available signals (measurements) and their time delays. This needs to be balanced against the achievable system behaviour. Controller design is therefore considered a difficult task especially when driving cross coupled multivariate non-linear systems.
This chapter presents results of comparisons of PA10 model simulation and manipulator experiments and the results of applying multiple robust and model based controllers to the PA10 manipulator. The contributions of this chapter constitute:

- the application of the non-parametric friction model in feedback model simulation and manipulator control of the PA10 manipulator,
- a broad comparison of robust and model based controllers applied to the PA10 manipulator with focus on trajectory tracking and high frequency torque content.

5.1 Introduction to manipulator controllers

Depending on the availability of the model and the knowledge about the structure of the manipulator, different types of controllers can be employed. The typical classification distinguishes

- robust controllers
- adaptive controllers
- model based controllers

The robust controllers such as a PD (proportional and derivative) controller provide global asymptotic stability (in the absence of gravity and friction) for step inputs at the cost of high feedback gains and possible faster actuator wear. The adaptive controllers provide the quality of a model based controller, requiring only the structure of the model, however the trade-off is the computational complexity and additional dynamics in the controller. The model based controllers provide good trajectory tracking properties, although the model and its parameters have to be known. Two groups of model based controllers can be distinguished: [5.7]

- computed torque algorithms,
- dissipative algorithms.

5.1.1 Robust control

When no information about the model of the system is available, only the tracking error can be used. The simplest approach is to feed back the negative output error, as an input to the plant

$$\mathbf{u} = -(\mathbf{q} - \mathbf{q}_d) \tag{5.1}$$

where $\mathbf{q}_d$ is the desired value, and $\mathbf{q}$ is the measured one. This scheme can be further refined by introducing a constant gain

$$\mathbf{u} = -\mathbf{K}\mathbf{e} = -\mathbf{K}(\mathbf{q} - \mathbf{q}_d) \tag{5.2}$$

where K is the feedback gain, and $\mathbf{e} = (\mathbf{q} - \mathbf{q}_d)$ is the control error. The presented setup is commonly called a **proportional controller** (**P** for short). It is the simplest continuous feedback controller. It is not considered as a valid controller, because it does not guarantee asymptotic stability of the control error. For a step input, the controller will drive the system close to the desired position with a constant error (depending on the controller gain) for a type zero system. The proportional controller can lead to oscillatory behaviour. To prevent the oscillations, a derivative of the error can also be used as part of the feedback.

5.1.1.1 Proportional and derivative (PD) controller

One of the simplest and most popular robust controllers (when both the state and its derivative are available (or the latter can be derived/estimated)) is a combination of an amplified negative control error and its time derivative [5.1].

$$\mathbf{u} = -\mathbf{K}_P\mathbf{e} - \mathbf{K}_D\dot{\mathbf{e}} \tag{5.3}$$

where K_P is the *proportional gain*, K_D is the *derivative gain*, However, this scheme does not ensure that the steady state error vanishes for a reference input of higher degree than a step for joints not affected by gravity. Moreover, it does not assure zero steady state error for step reference input when applied to joints affected by gravity. The control quality is further compromised by the presence of friction. Therefore, a term needs to be introduced that can accumulate the error measured over time and use it in the feedback loop.

5.1.1.2 The proportional, integral and derivative (PID) controller

$$\mathbf{u}(t) = -\mathbf{K}_P \cdot \mathbf{e}(t) - \mathbf{K}_D \cdot \dot{\mathbf{e}}(t) - \mathbf{K}_I \cdot \int_{t_0}^{t} \mathbf{e}(s)ds, \tag{5.4}$$

where $\mathbf{K}_I$ is the *integral gain* and t_0 is the start time of the control. The main disadvantage of this formula is that it can produce overshoots, when the system starts in a position distant from the desired one. To prevent this, the integration can be reduced only to instances when the controller output is achievable by the system.

Integrator anti-windup The integral component is modified [5.2]

$$\mathbf{u}(t) = -\mathbf{K}_P \cdot \mathbf{e}(t) - \mathbf{K}_D \cdot \dot{\mathbf{e}}(t) - \mathbf{K}_I \cdot \int_{\mathbf{I}} \mathbf{e}(s)ds, \tag{5.5}$$

with

$$\mathbf{I} = \{s \in [t_0, t] | (\mathbf{u}_i(s) < u_{i,\max}) \wedge (\dot{\mathbf{q}}_i(s) < v_{i,\max}), \text{for} i \in \{1, \ldots \text{number of joints}\}\}, \tag{5.6}$$

where $u_{i,\max}$ is the maximum torque allowed by the actuator for joint i, and $v_{i,\max}$ is the maximum generalised velocity recommended by the manufacturer for joint i. The integrator stops the integration process when the joint is driven with the maximum allowed torque, or when the joint is moving with a maximal allowed velocity. This control scheme partly prevents overshoot in tracking *step* like trajectories. Both of these methods produce persistent high feedback gains. Additionally, the integrator captures any non-zero error and may cause oscillatory behaviour in the presence of static friction.

5.1.1.3 Sliding mode controller

The problem can be approached from a slightly different point of view, by means of defining a **sliding surface** [5.5, 5.3]

$$\mathbf{s} = \dot{\mathbf{e}}(t) + \mathbf{\Lambda}\mathbf{e}(t), \tag{5.7}$$

where $\mathbf{\Lambda}$ is a parameter defining the bandwidth of the controller. The feedback loop takes the form:

$$\mathbf{u} = \mathbf{K}\,\text{sat}\left(\frac{\mathbf{s}}{\boldsymbol{\varphi}}\right), \tag{5.8}$$

where $\mathbf{K}$ is the gain parameter and $\boldsymbol{\varphi}$ is the boundary layer parameter, and

$$\text{sat}(x) = \begin{cases} 1 & \text{if } x > 1 \\ x & \text{if } -1 \leq x \leq 1 \\ -1 & \text{if } x < -1. \end{cases}$$

This controller can be viewed as a PD controller with saturation. The introduction of saturation prevents very high feedback gains, but introduces a lack of smoothness, which can be undesirable in analytical controller analysis.

5.1.2 Model based control

When the model of the system is available, it can be used to improve the controller [5.6]. There are two main groups of model based controllers:

- **computed torque** algorithms require the invertibility of the inertia matrix, and can be considered as static feedback linearisation.
- **dissipative** algorithms do not require a linearising feedback, but try to dissipate energy of the system to drive the position and velocity errors asymptotically to zero.

5.1.2.1 Gravity and friction compensation

Some of the dynamic characteristics of the controlled system can be provided by the manufacturer. Such was the case for the Mitsubishi PA10-6CE with the mass distribution diagram. Based on this data the gravity model has been derived. This can be combined in the feedback loop with a generic PD controller

$$\mathbf{u} = \mathbf{g}(\mathbf{q}) - \mathbf{K}_P\mathbf{e} - \mathbf{K}_D\dot{\mathbf{e}}. \tag{5.9}$$

After obtaining friction estimates, the model can be enriched by this component

$$\mathbf{u} = \mathbf{g}(\mathbf{q}) + \mathbf{F}(\dot{\mathbf{q}}) - \mathbf{K}_P\mathbf{e} - \mathbf{K}_D\dot{\mathbf{e}}. \tag{5.10}$$

According to the model comparison results in [5.10], the gravity and friction constitute the majority of the torque being modelled (for the PA10-6CE manipulator). Both of these controllers can be classified as dissipative, as they are unable to linearise the system, due to the lack of the inertia matrix.

5.1.2.2 Feedback linearisation.

Once the full model is available, one of the classical model based algorithms can be applied. The most popular model based control scheme is feedback linearisation (also referred to as computed torque control). The model is used to linearise the system and control the simple linear dynamics. Assuming the model is defined by equation 3.9, the controller takes the form

$$\mathbf{u} = \mathbf{Q}(\mathbf{q})\mathbf{v} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}), \tag{5.11}$$

where

$$\mathbf{v} = \ddot{\mathbf{q}}_d - \mathbf{K}_D\dot{\mathbf{e}} - \mathbf{K}_P\mathbf{e} \tag{5.12}$$

and the error is defined as

$$\mathbf{e} = \mathbf{q} - \mathbf{q}_d, \tag{5.13}$$

where $\mathbf{q}_d$ is the desired trajectory. After adding the friction element to equation 5.11

$$\mathbf{u} = \mathbf{Q}(\mathbf{q})\mathbf{v} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{F}(\dot{\mathbf{q}}), \tag{5.14}$$

a linear control system is obtained

$$\ddot{\mathbf{q}} = \ddot{\mathbf{q}}_d - \mathbf{K}_D\dot{\mathbf{e}} - \mathbf{K}_P\mathbf{e}. \tag{5.15}$$

The closed loop system poles can be shaped by means of the two feedback gains. This controller relies very much on the quality of the model and model discrepancies or environmental disturbances have detrimental effects.

5.1.2.3 Wen - Bayard's algorithm

Another representative of the computed torque group is the algorithm developed by Wen and Bayard [5.9]. The control torque is computed from the following equation

$$\mathbf{u} = \mathbf{Q}(\mathbf{q}_d)\ddot{\mathbf{q}}_d + \mathbf{C}(\mathbf{q}_d,\dot{\mathbf{q}}_d)\dot{\mathbf{q}}_d + \mathbf{g}(\mathbf{q}_d) + K_d\dot{\mathbf{e}} + K_p\mathbf{e}, \tag{5.16}$$

where K_p and K_d are the PD gains. Assuming that a model of friction is available, the following control scheme can be proposed

$$\mathbf{u} = \mathbf{Q}(\mathbf{q}_d)\ddot{\mathbf{q}}_d + \mathbf{C}(\mathbf{q}_d,\dot{\mathbf{q}}_d)\dot{\mathbf{q}}_d + \mathbf{g}(\mathbf{q}_d) + \mathbf{F}(\dot{\mathbf{q}}_d) + K_d\dot{\mathbf{e}} + K_p\mathbf{e}, \tag{5.17}$$

The main advantage of this controller is that the model-derived command signal can be pre-computed off-line. However any control errors compromise the off-line model functionality.

5.1.2.4 Slotine - Li algorithm

In the dissipative algorithms group is a scheme developed by Slotine and Li [5.4]. The torque is computed from the following equation

$$\mathbf{u} = \mathbf{Q}(\mathbf{q})\ddot{\mathbf{q}}_r + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}_r + \mathbf{g}(\mathbf{q}) - \mathbf{K}_D\mathbf{s}, \tag{5.18}$$

with

$$\dot{\mathbf{q}}_r = \dot{\mathbf{q}}_d - \mathbf{\Lambda}\mathbf{e} \tag{5.19}$$

and

$$\mathbf{s} = \dot{\mathbf{q}} - \dot{\mathbf{q}}_r = \dot{\mathbf{e}} + \mathbf{\Lambda}\mathbf{e} \tag{5.20}$$

This algorithm can also be extended by using the friction compensation element:

$$\mathbf{u} = \mathbf{Q}(\mathbf{q})\ddot{\mathbf{q}}_r + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}_r + \mathbf{g}(\mathbf{q}) + \mathbf{F}(\dot{\mathbf{q}}) - \mathbf{K}_D\mathbf{s}, \tag{5.21}$$

In this case the model is computed on-line based on measurements, which can compensate for tracking errors, but can also amplify any measurement noise.

5.2 Model evaluation

The PA10-6CE model behaviour is compared with the experimental results recorded from the robotic arm. Feedback control methods are employed to deliver repeatable results. The following two main groups of experiments/simulations were performed:

- The manipulator and the model (or parts of it) are driven by a PID controller along a given trajectory and the resulting control torques are compared.
- The manipulator is driven by a a feedback controller (various types from robust to model based) and the trajectory tracking errors are compared.

The simulation and experimental results are compared qualitatively by plotting the results on a graph and quantitatively by means of an Integral Squared Error of certain signals

$$\mathrm{ISE}(x,y) = \int_{t_0}^{t_F} (x(t) - y(t))^2 dt,$$

where $x(t)$ and $y(t)$ are the compared quantities, t_0 is the start time and t_F is the end time. All experiments are performed using the PA10 robot control application ([5.8], [5.11]) with the control sampling frequency set to 500 Hz[1]. The simulations were performed using MATLAB and a trapezoid based differential equations solver, which employs a functionality to lock joints when the applied torque does not exceed the static joint friction. The simulation was oversampled at a power of two multiple of the manipulator control frequency ($2^n \times 500$ Hz), with n selected each simulation step, to provide sufficient modelling accuracy. Results were stored only at the manipulator controller sampling instances (500 Hz). Due to the fixed sampling rate used in all experiments and simulations the ISE measure simplifies to a sum (using the rectangle integration rule)

$$\mathrm{ISE}(x,y) = \sum_{i=0}^{N-1} \left((x(i) - y(i)\right)^2 \delta_t,$$

where δ_t is the sampling period equal to 2 ms and N is the number of samples in each experiment (equal to $500\times$ experiment time in seconds).
To compare signals driving different joints simultaneously, it is desired to have no dependence on the signal magnitude or experiment time, therefore a weighted mean ISE

[1]This is a frequency at which commands are sent from the controller PC to the servo controller box (which has an internal control routine to communicate with the arm)

(WMISE) measure was applied to the data

$$\mathrm{WMISE}(x,y) = \frac{\frac{\sum_{i=0}^{N-1}(x(i)-y(i))^2\delta_t}{t(N)-t(0)}}{\frac{\sum_{i=0}^{N-1}x^2(i)\delta_t}{t(N)-t(0)}} \tag{5.22}$$

where $x(i)$ is the reference signal and $y(i)$ is the compared signal.
The measured command torque signals contain high frequency elements, which is not present in the MATLAB simulations, and can be viewed as *control noise* being above the bandwidth of the manipulator mechanical system. It introduces discrepancies between the control signals even though it is clear that the actual low frequency content is in good accordance. Therefore the torque signal has been filtered using a 9-th order Butterworth-IIR low pass filter with pass band $[0,15]$ Hz with 3 dB range and stop band from 30 Hz with 50 dB attenuation. The characteristics of the designed filter is plotted in figure 5.3.

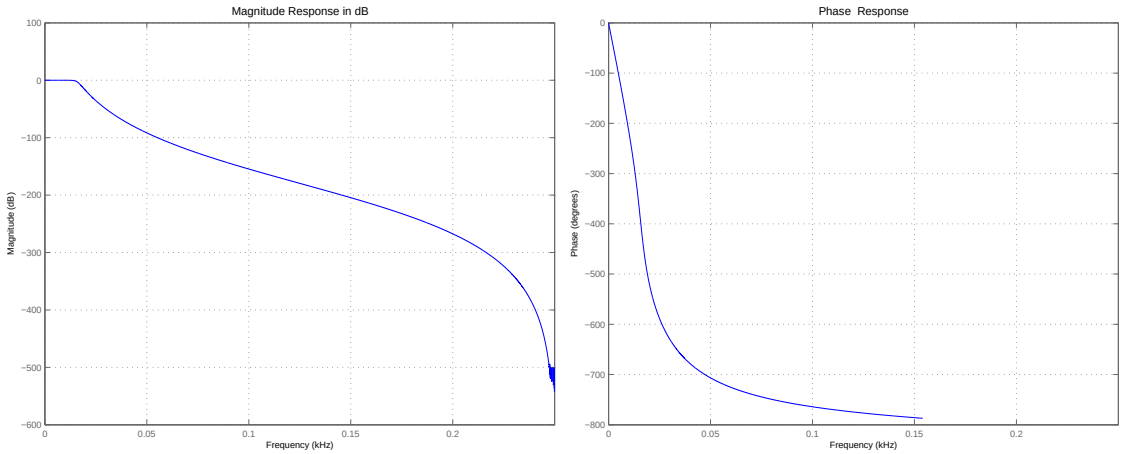


Figure 5.3: Characteristics of filter used for torque signal analysis.

In all comparisons the filter has been applied using MATLAB's zero-phase filtering command `filtfilt` (applying the same filter in both directions to the time-extended signal). For the experimental controller evaluation an additional quantity has been computed to measure the amount of *control noise* in the command torque. The following formula has been used to calculate the weighted mean control noise ratio (WMCNR)

$$\mathrm{WMCNR}(x,y) = \frac{\frac{\sum_{i=0}^{N-1}\left(u(i)-u_f(i)\right)^2\delta_t}{t(N)-t(0)}}{\frac{\sum_{i=0}^{N-1}u_f^2(i)\delta_t}{t(N)-t(0)}} \tag{5.23}$$

where $u(i)$ is the applied torque read from the controller and $u_f(i)$ is the filtered torque sample.

Trajectories The following trajectories were used

- sine trajectory tracked by all joints
  - synchronised (equal periods) (**ssine**)

$$\mathbf{q}_d(t) = [R_i(\cos(\omega_1 t) - 1)]_{i=1}^{6}, \tag{5.24}$$

  - asynchronous (different periods) (**sine**)

$$\mathbf{q}_d(t) = [R_i(\cos(\omega_i t) - 1)]_{i=1}^{6}, \tag{5.25}$$

- step trajectory tracked by all joints
  - simultaneous (all joints start moving in the same instance) (**sstep**)

$$\mathbf{q}_d(t) = [R_i \mathbf{1}(t - \Delta_1)]_{i=1}^{6}, \tag{5.26}$$

    where

$$\mathbf{1}(t) = \begin{cases} 0 & \text{for} \quad t < 0 \\ 1 & \text{for} \quad t \geq 0, \end{cases}$$

  - delayed (each joint starts moving at a different moment) (**dstep**)

$$\mathbf{q}_d(t) = [R_i \mathbf{1}(t - \Delta_i)]_{i=1}^{6}, \tag{5.27}$$

- step trajectory followed by one joint, while other joints follow an asynchronous sine trajectory (**asinestep**).

$$\mathbf{q}_d^k(t) = \left[\begin{cases} R_i \mathbf{1}(t - \Delta_i) & \text{for} \quad i = k \\ R_i(\cos(\omega_i t) - 1) & \text{for} \quad i \neq k \end{cases} \right]_{i=1}^{6}, \tag{5.28}$$

5.2.1 Comparative PID control

In this experiment the PA10 manipulator and the model in Matlab were controlled in a closed-loop configuration using a **PID** controller. The control parameters[2] are presented in table 5.1.

The **P** and **D** coefficients have been selected to minimise the trajectory tracking error. However, a simple tuning rule was not used due to control 'jitter' being caused in motors when too much derivative gain was applied (most probably caused by velocity estimation errors/noise). The **I** coefficients have been selected to deliver good tracking accuracy without any 'torque ringing' effects. The feedback gain parameters have been fixed for all experiments throughout this thesis (whether used in model based or simple robust

[2]The PA10 control application uses motor torque values, and the simulation uses joint torque

joint	K_P	K_I	K_D
S1	61.30	9.00	7.67
S2	92.00	11.0	9.20
E1	33.00	6.00	2.00
E2	7.780	3.00	0.36
W1	17.50	7.00	0.35
W2	7.875	4.00	0.35

Table 5.1: PID controller parameters used in experiment and simulation (motor torque scale).

controllers). It can be noticed that higher values have been chosen for joints strongly affected by gravity.

The sliding controllers parameters (presented in table 5.2) have been selected to match the PD gains (from table 5.1).

joint	$\mathbf{\Lambda}$	$\boldsymbol{\varphi}$	$\mathbf{K}$
S1	8.00	0.6	4.60
S2	10.0	0.5	4.60
E1	10.0	0.6	1.98
E2	50.0	1.8	0.28
W1	50.0	0.8	0.28
W2	22.5	0.8	0.28

Table 5.2: Sliding controllers parameters used in experiments and simulations (motor torque scale).

Table (5.2.1) shows the trajectory parameters used.

torque	[Nm]	range	[rad]	ω	$[s^{-1}]$	Δ	[s]
U_1	0.55	R_1	-1.0	ω_1	$\frac{2\pi}{5}$	Δ_1	2.0
U_2	0.00	R_2	-0.7	ω_2	$\frac{2\pi}{4.5}$	Δ_2	2.1
U_3	0.00	R_3	0.7	ω_3	$\frac{2\pi}{4}$	Δ_3	2.2
U_4	0.08	R_4	-1.2	ω_4	$\frac{2\pi}{4.2}$	Δ_4	2.3
U_5	0.00	R_5	-1.0	ω_5	$\frac{2\pi}{3}$	Δ_5	2.4
U_6	0.08	R_6	1.2	ω_6	$\frac{2\pi}{4.1}$	Δ_6	2.5

Table 5.3: Trajectory parameters used in experiments and simulations.

Using the **sine** trajectory the following partial models were compared with the experimental results (to evaluate the contribution of each element of the model)

- **full model**,
- full model **without friction** component,
- full model **without gravity** component,

- full model **without motor inertia** component,
- full model **without joint inertia** component,
- full model **without motor and joint inertia** components (unity inertia),
- full model **without Coriolis and Centrifugal** components.

The weighted mean integral squared errors (WMISE) of the experimental control torques and the simulation control torques are used to determine the importance of given dynamic elements in the overall model for each joint. It is expected that due to the high gear ratio the joint inertia will will make a fewer contribution to the joint torques than either gravity or friction (with the effect of the latter actually being amplified by the transmission). It was also anticipated that the S2 and E1 joints control will have significant contributions on gravity torque. The results of this comparison should justify the employment of the manipulator dynamics into a feedback controller.
The actual angle, velocity and motor torque plots were also compared for a visualisation of the contribution of each model element to the overall control torque.

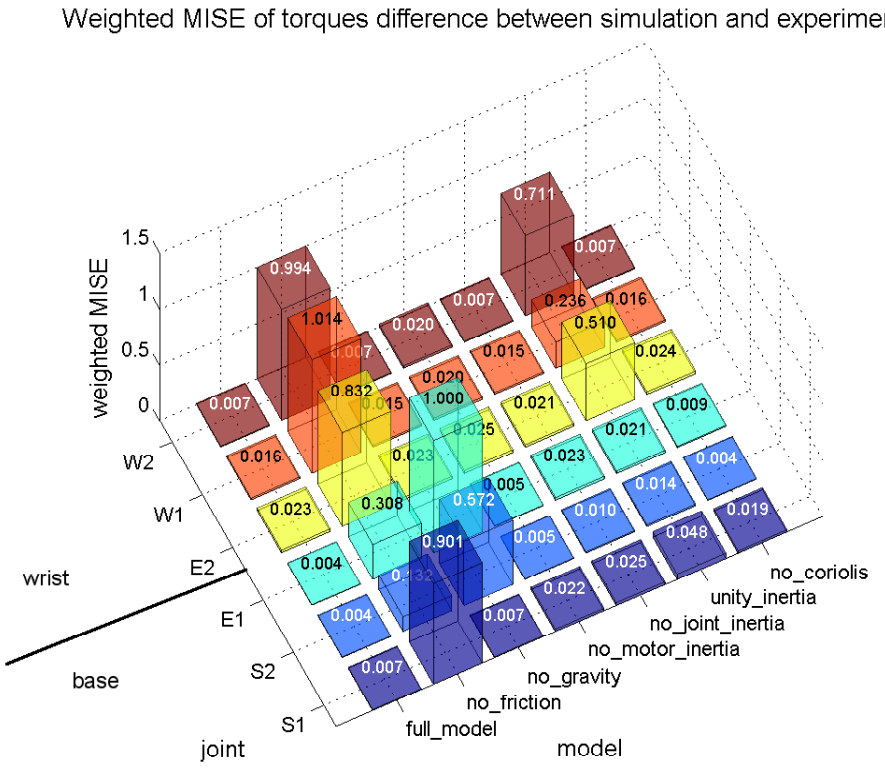


Figure 5.4: Comparison of simulation and experimental torque differences per joint for trajectory **`sine`**.

The following conclusions can be drawn from the WMISE plot (figure 5.4):

- As expected the S2 and E1 joint control torques are significantly affected by the presence of gravity terms.

- The friction is an important term for each joint.
- Motor inertias are more significant for the wrist joints (E2, W1, W2) and joint inertias for the base joints (S1, S2, E1).
- The replacement of inertia values with a unity matrix mostly affects the lighter wrist joints (with smaller motors).
- Tho Coriolis and Centrifugal term (requiring most computation) has only a minimal effect on the joint torques for S1 and E1, and could be considered negligible for all other joints (if computational efficiency were an issue).

The following points can be observed based on the *TVA* plots (figures 5.5, 5.6 and 5.7)

- For the full model there is a very good agreement between the simulation and filtered experimental results.
- The plot showing a model without gravity gives a good indication of the amount of joint torque required to counteract the static weight of the links.
- Without the friction model the command signal is continuous and *sine* based.
- The motor torque magnitudes presented on these plots also provide a good justification for using a weighted error measure, due to the torques being almost an order of magnitude greater for the S2 joint in comparison to the wrist joints.

5.2.2 Model based control of the PA10 manipulator

In the second part of the model evaluation, a set of model based controllers (with partial and full model) was applied to the PA10 manipulator and compared with standard PD and PID controllers. The following controller types were used

- PD controller (equation 5.3),
- PID controller (equations 5.5, 5.6),
- PD controller with gravity compensation (equation 5.9),
- PD controller with gravity and friction compensation (equation 5.10),
- Slotine-Li controller (equation 5.21),
- Slotine-Li controller without the Coriolis and centrifugal term,
- Wen-Bayard controller (equation 5.17),
- Wen-Bayard controller without the Coriolis and centrifugal term,
- feedback linearisation controller (equation 5.14, 5.12),

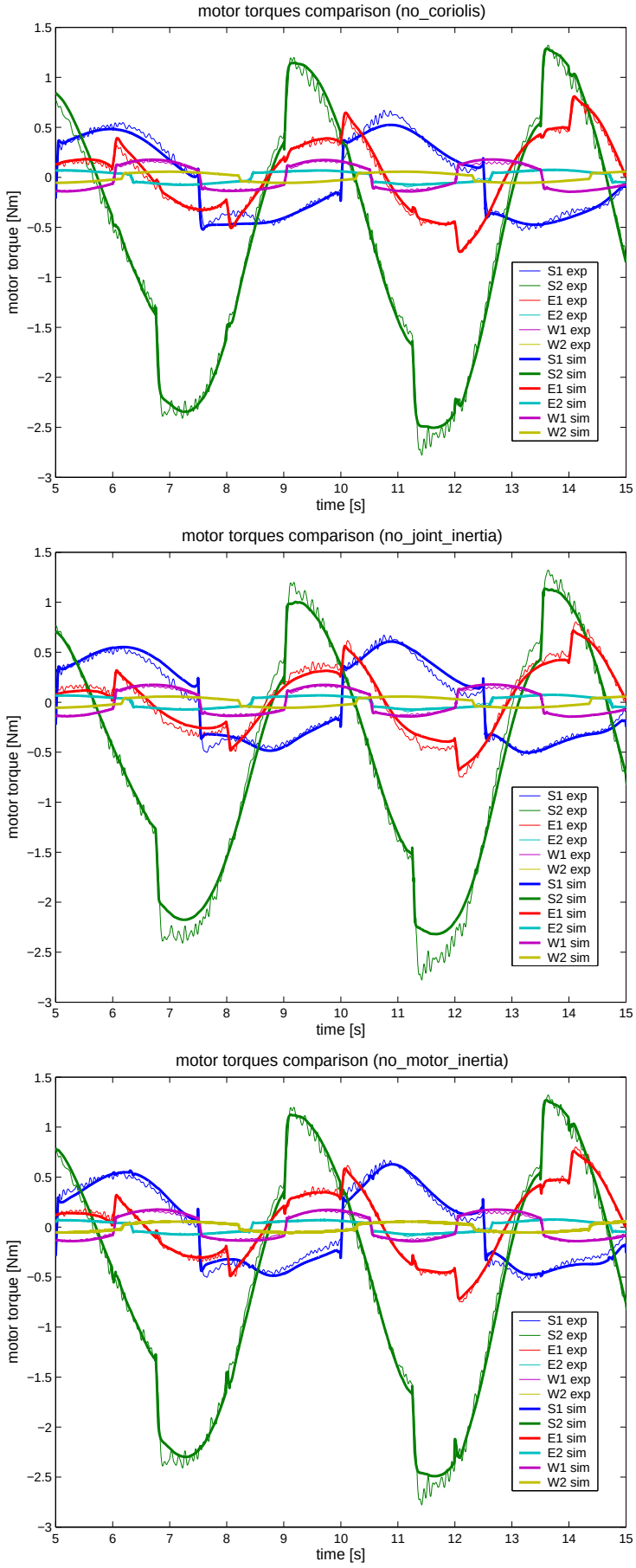


Figure 5.5: Comparison of simulation and experimental torques for different model set-ups, trajectory **`sine`.**

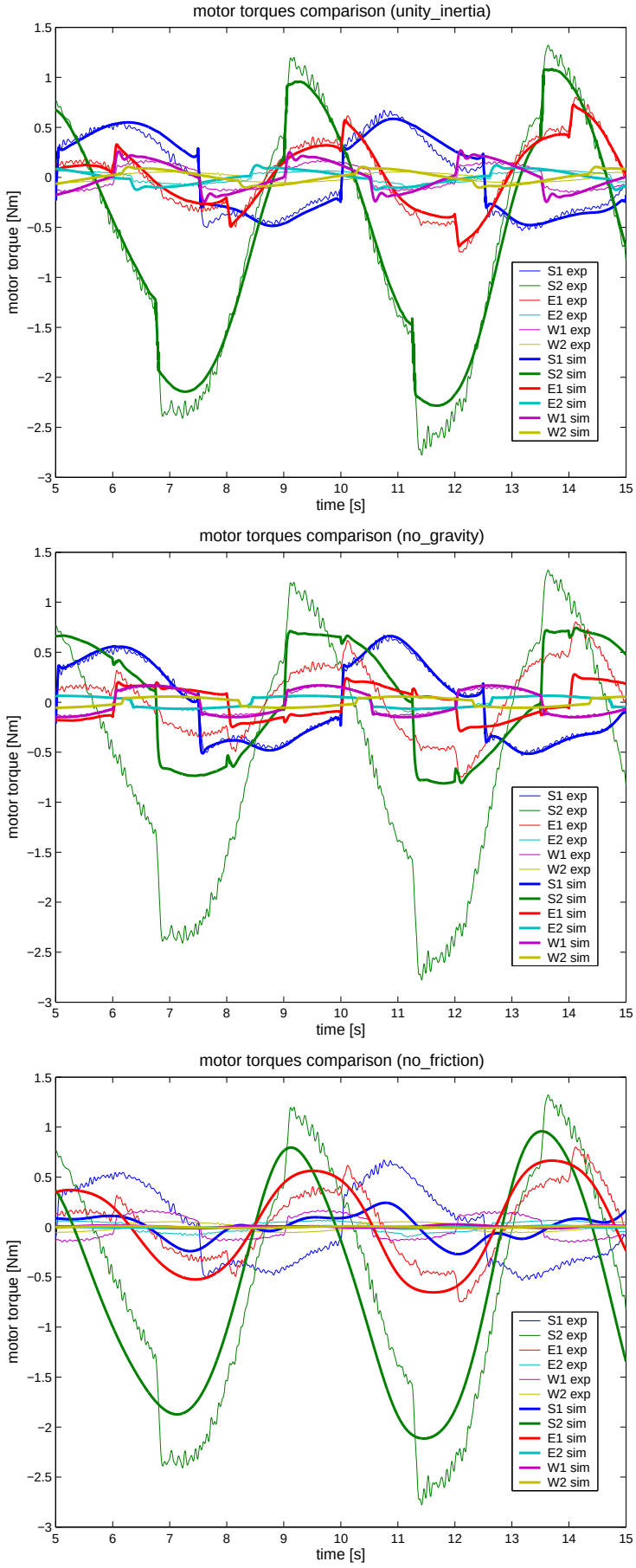


Figure 5.6: Comparison of simulation and experimental torques for different model set-ups, trajectory **`sine`.**

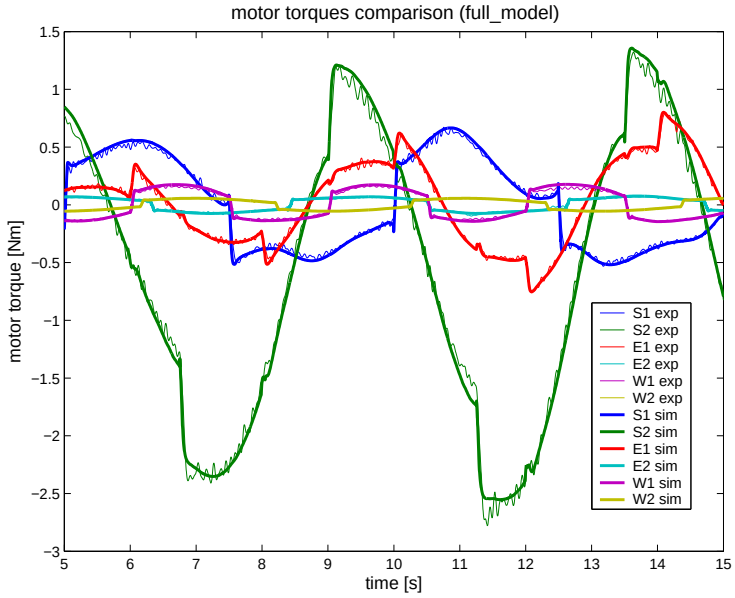


Figure 5.7: Comparison of simulation and experimental torques for different model set-ups, trajectory **`sine`.**

- feedback linearisation controller without the Coriolis and centrifugal term,
- the native PA10-6CE velocity controller [3].

The feedback linearisation controller PD parameters are shown in table 5.4. The parameters were selected manually to minimise the trajectory tracking errors while preventing the 'ringing' effect in the joint motors.

joint	K_P	K_D	poles
S1	`500.00`	`40.0`	$-20 \pm 10i$
S2	`1600.0`	`40.0`	$-20 \pm 34.6i$
E1	`1500.0`	`35.0`	$-17.5 \pm 34.6i$
E2	`2000.0`	`45.0`	$-22.5 \pm 38.6i$
W1	`3000.0`	`75.0`	$-37.5 \pm 39.9i$
W2	`3000.0`	`75.0`	$-37.5 \pm 39.9i$

Table 5.4: PD controller parameters (and poles) used in the feedback linearisation controller.

All controllers were tested with both sine trajectories (**`sine`**, **`ssine`**, equations 5.25, 5.24) to evaluate the relative per joint control quality. The PID and full linearisation controllers were applied to track the **`dstep`** trajectory (equation 5.27) for a graphical comparison of control signals and controller '*behaviour*'. The **`asinestep`** trajectory (equation 5.28) was used for graphical comparison of control signals during the step event

[3]The PA10 servo controller allows two control modes, torque command and velocity command. In the velocity command an internal controller is used.

and during the stationary period[4] (for each joint), and the stationary period was also used for quantitative control evaluation.
The following conclusions can be drawn for the **`dstep`** trajectory based experiments presented in figures 5.8 to 5.11 comparing the PID and full linearisation controllers for all joints:

- It can be noticed on all figures that the velocity saturates for joints S1, S2, E1 and E2 and the servo controller applies additional torque limiting.
- The PID controller (figures 5.8 and 5.9) provides a faster response when the motion of other joints influences the controller (this may be attributed to a high gain).
- The linearisation controller (figures 5.10 and 5.11) produces more oscillations in the control signal, which can be attributed to the friction feedback (due to a discontinuous friction model around zero velocity, an observable *oscillatory boundary stability effect* may occur).
- The response time is comparable for both controllers, although the full linearisation produces a greater overshoot in some cases (E1, W2), which can again be attributed to the discontinuous friction model.

Conclusions for the **`asinestep`** control experiments (figures 5.12 to 5.19) trajectory were made based on the comparison of control signals for joint E1[5]:

- There is no noticeable difference in the response time for all controllers (marked with black dashed lines of figures 5.12 to 5.15). This can be partly attributed to the fact that the velocity saturates during the motion, and the servo controller applies torque limiting.
- Slotine-Li (figure 5.14) and PD+grav+frict (figure 5.13) controllers show a significant amount of high frequency content in the command torque.
- It can be noticed that the control torques generated for the PD+grav+frict (figure 5.13) and linearisation controllers (figures 5.14 and 5.15) have a noticeable amount of velocity oscillation (due to straight feedback of the discontinuous (around zero) friction model), which can influence the squared error results.
- From the *stationary* period plots (figures 5.16 to 5.19) it can be seen that the velocity has smallest variation when the PD+gravity and Wen-Bayard controllers are applied.

[4]The term *stationary period* is used to denote the time instances after one joint has performed its step motion, and its trajectory is constant over that period of time - while other joints follow sinusoidal trajectories.

[5]this joint was selected as it is exhibits both signifficant gravity and frictional effects, and is well inside the manipulator kinematic chain to have possibly noticeable Coriolis and centrifugal torques

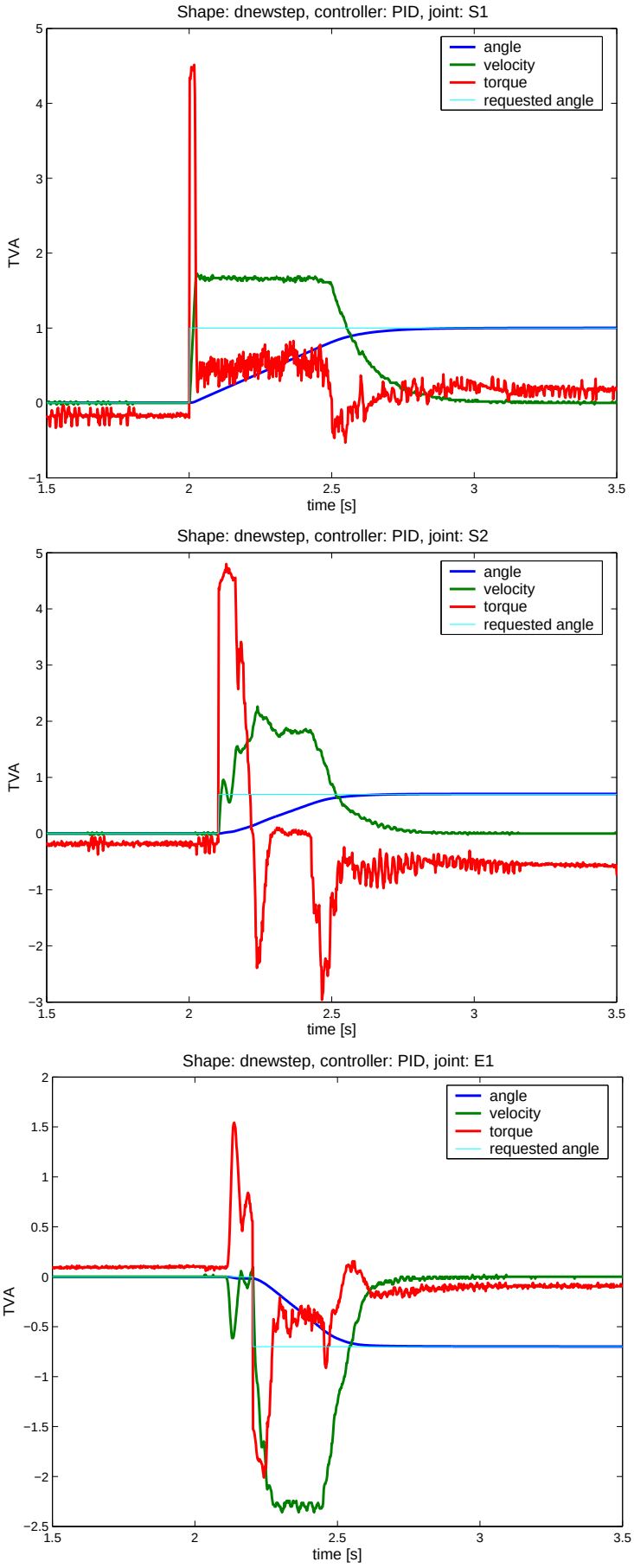


Figure 5.8: **`dstep`** trajectory followed using a PID controller, joints S1, S2, E1.

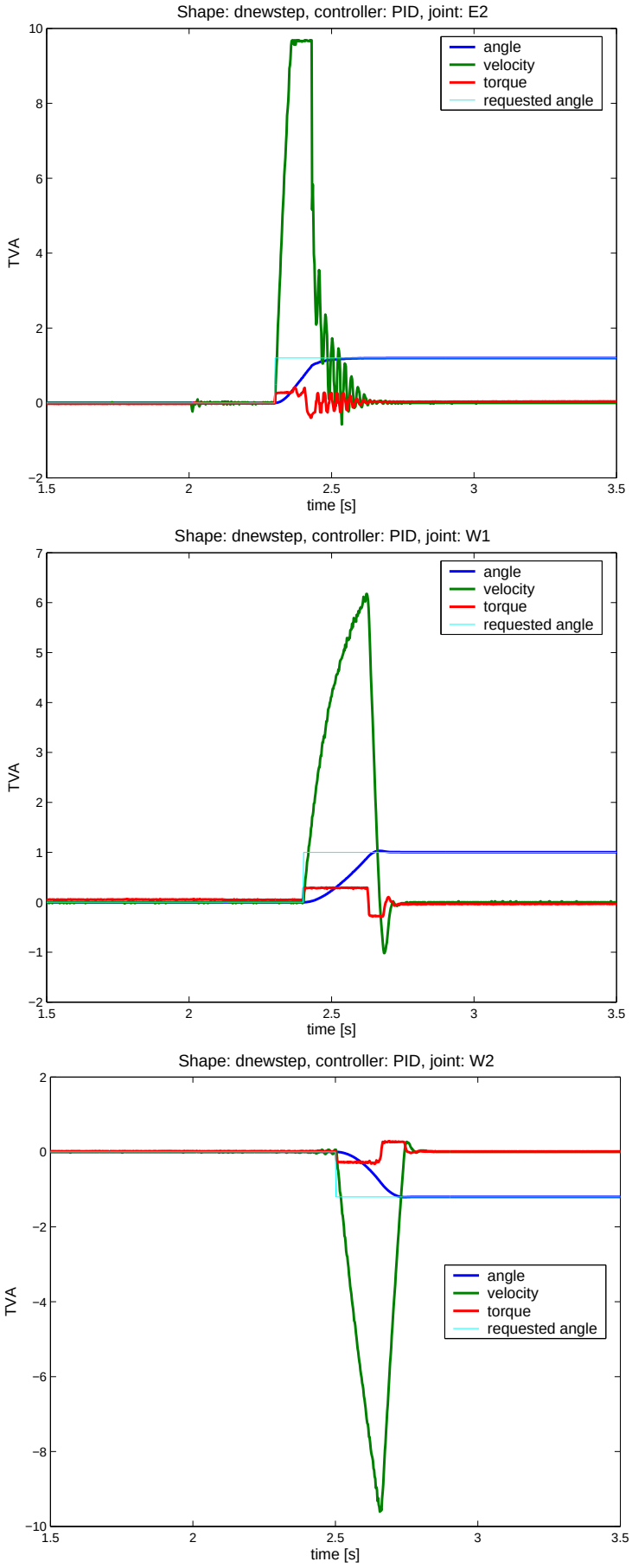


Figure 5.9: **`dstep`** trajectory followed using a PID controller, joints E2, W1, W2.

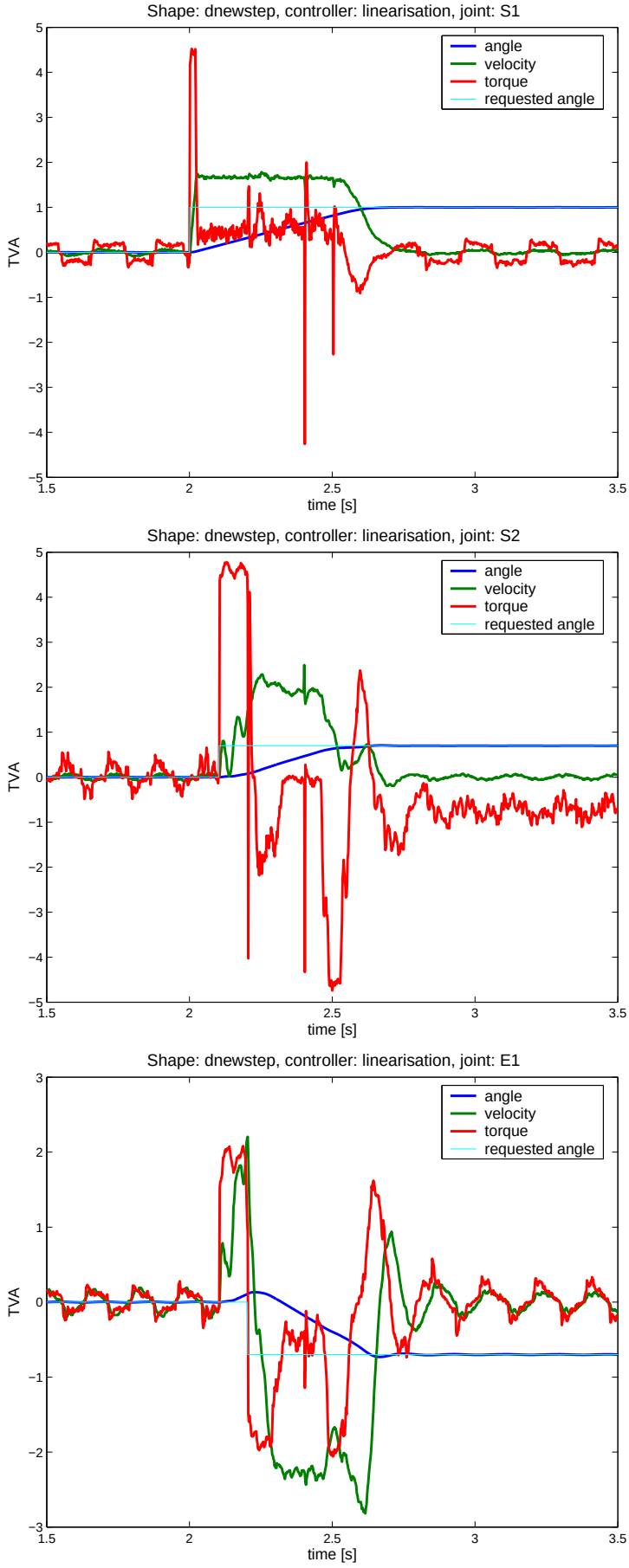


Figure 5.10: **`dstep`** trajectory followed using a full linearisation controller, joints S1, S2, E1.

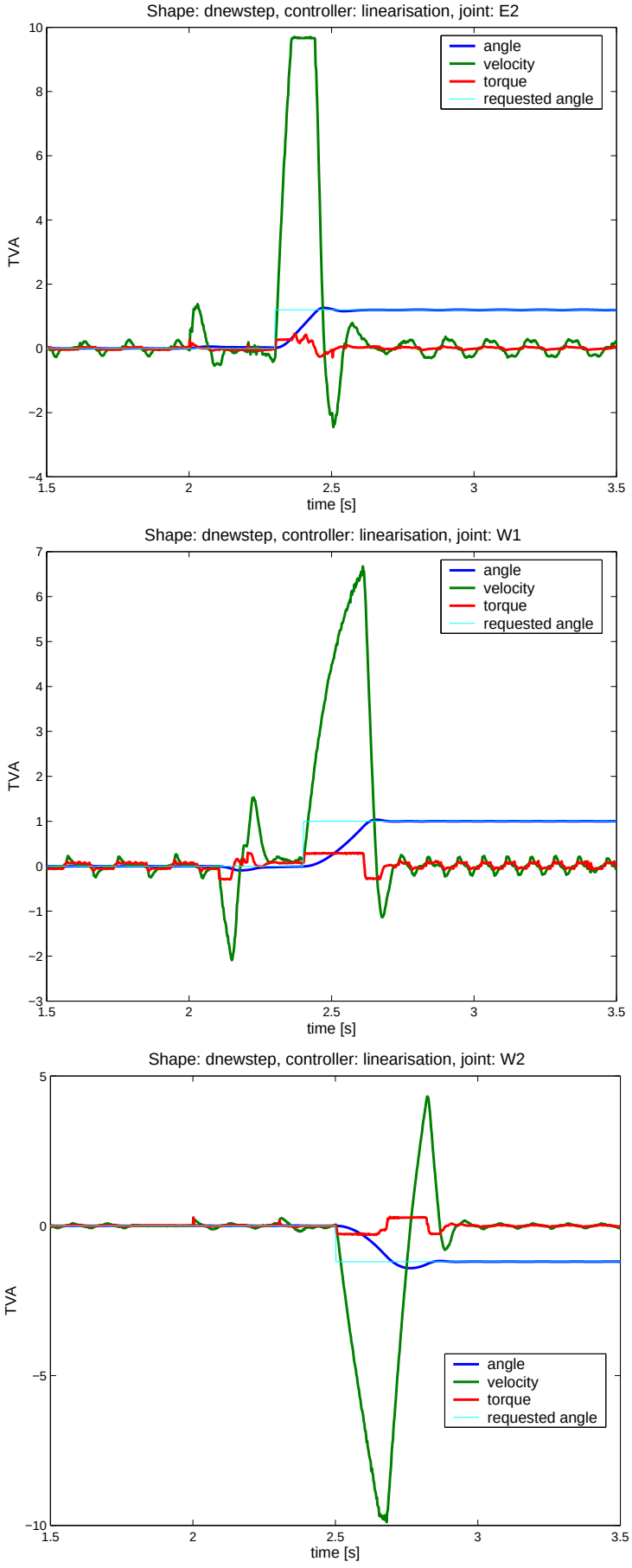


Figure 5.11: **`dstep`** trajectory followed using a full linearisation controller, joints E2, W1, W2.

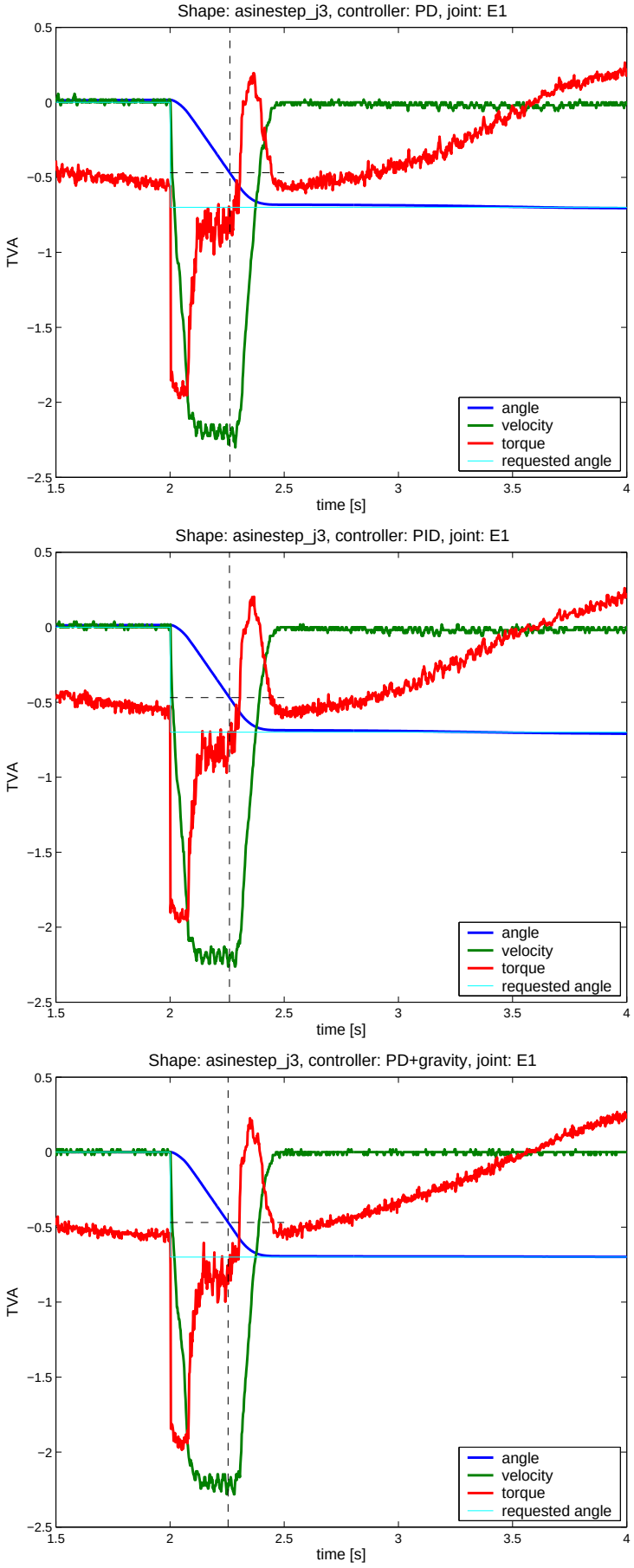


Figure 5.12: The step event of **`asinestep`** trajectory followed by joint E1 for various controllers.

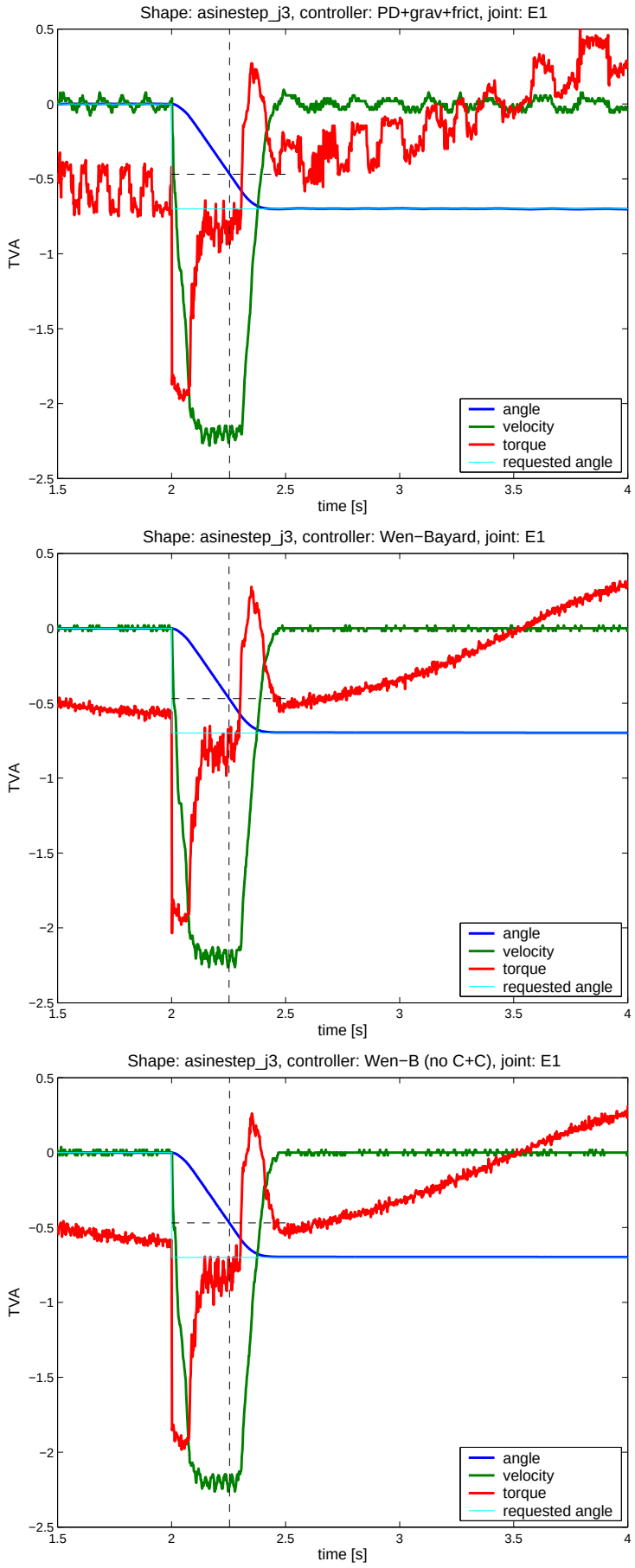


Figure 5.13: The step event of **`asinestep`** trajectory followed by joint E1 for various controllers.

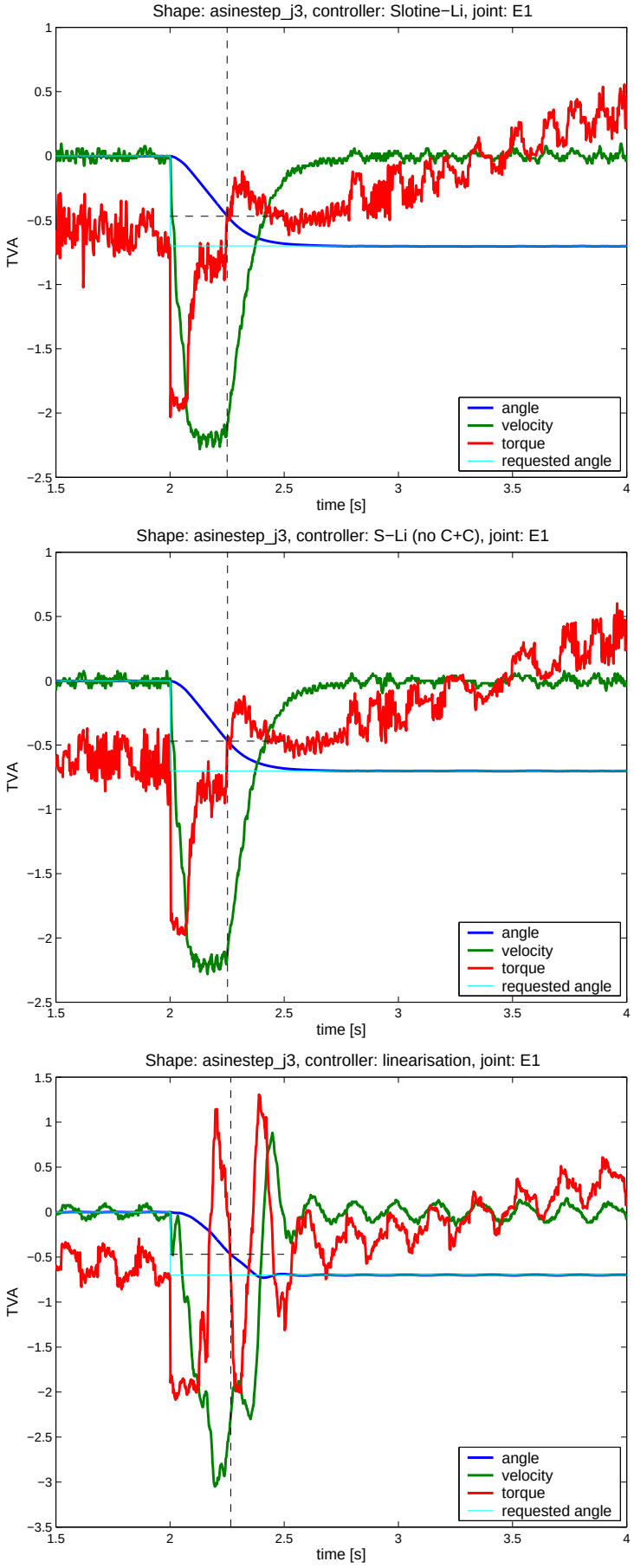


Figure 5.14: The step event of **`asinestep`** trajectory followed by joint E1 for various controllers.

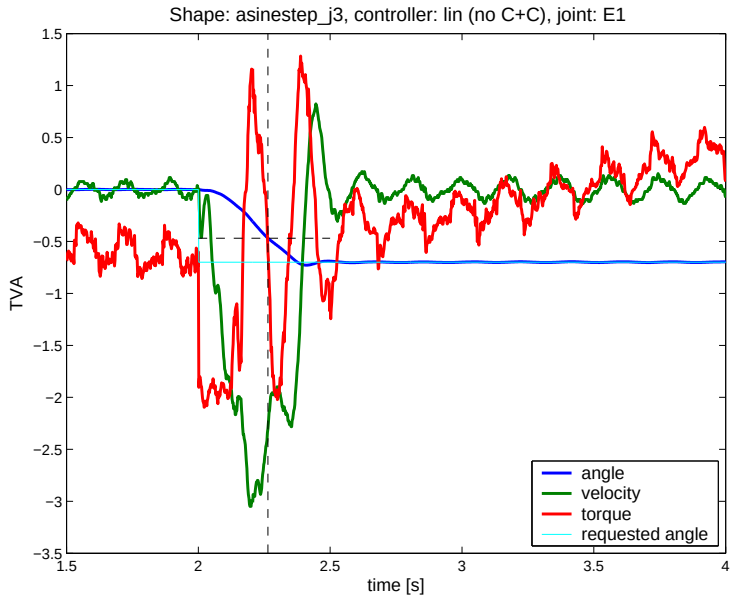


Figure 5.15: The step event of **`asinestep`** trajectory followed by joint E1 for different controllers.

- The controllers without Coriolis and centrifugal term (denoted with no C+C in figures 5.13 to 5.15 and 5.17 to 5.19) do not differ noticeably from their full model versions.
- Various levels of torque noise/oscillation can be noticed on the stationary period experiments (figures 5.16 to 5.19). It can be observed that the high frequency oscillations amplify with the increase of controller terms depending on measured velocity - such as friction and Coriolis and centrifugal.
- Based on the graphical observations it shows that the linearisation controller is not best suited for the task of holding a single joint still while other ones are moving (figures 5.18 and 5.19).

controller	S1	S2	E1	E2	W1	W2
PD	5.13e-6	1.58e-4	1.44e-4	6.42e-6	6.80e-7	5.54e-9
PID	4.85e-6	1.09e-4	1.53e-4	1.21e-6	6.02e-7	4.18e-8
PD+gravity	4.26e-6	2.51e-6	3.39e-6	2.57e-6	2.49e-6	3.04e-9
PD+grav+frict	4.73e-7	1.23e-6	7.72e-6	1.88e-6	1.11e-6	3.91e-8
Wen-Bayard	6.90e-6	5.05e-6	1.98e-6	1.07e-5	1.02e-6	3.52e-8
Wen-B (no C+C)	4.13e-6	3.20e-6	4.21e-6	7.34e-6	1.88e-6	3.88e-8
Slotine-Li	3.53e-7	2.63e-6	1.51e-5	1.27e-7	6.91e-7	7.84e-8
S-Li (no C+C)	7.01e-7	2.07e-6	8.82e-6	1.02e-7	6.67e-7	6.36e-8
linearisation	1.42e-6	6.34e-7	8.70e-6	1.06e-5	1.88e-6	4.97e-7
lin (no C+C)	1.18e-6	3.75e-7	9.42e-6	1.11e-5	1.98e-6	3.03e-7

Table 5.5: Weighted MISE of angle tracking errors, trajectory: asinestep.

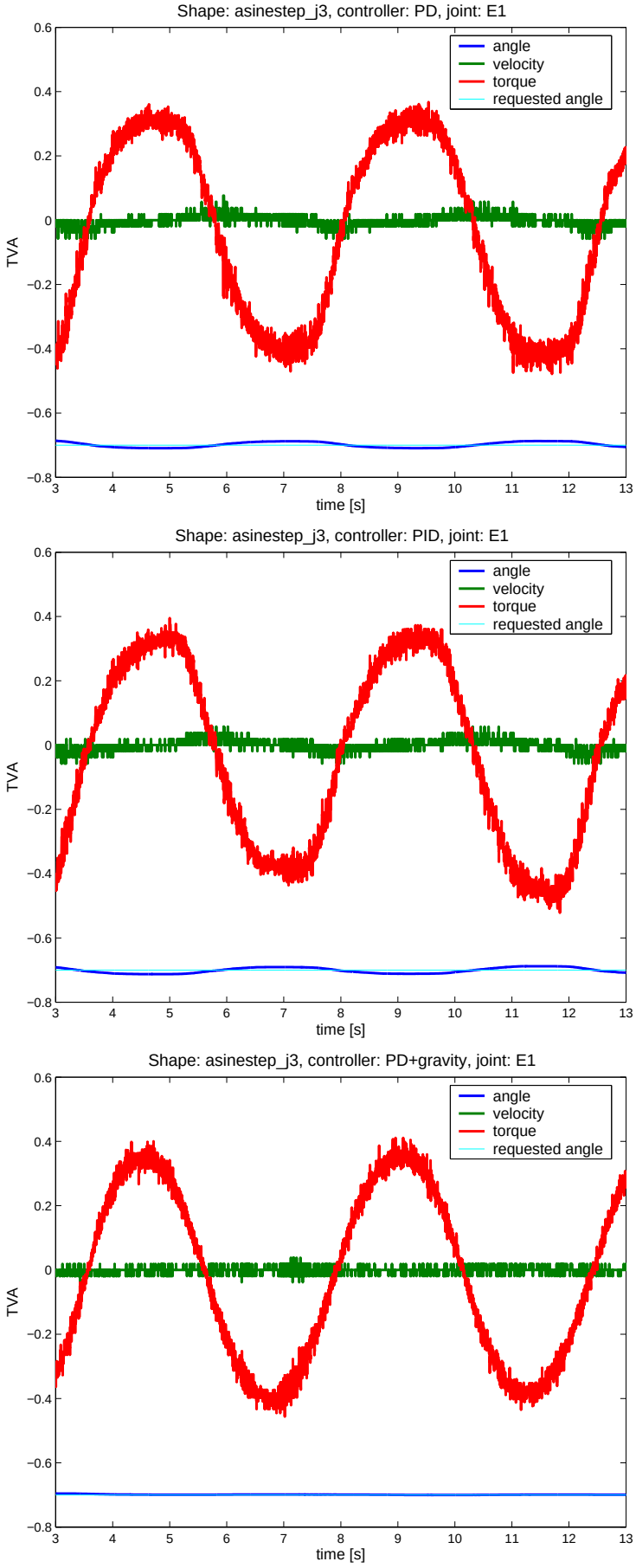


Figure 5.16: The stationary period of **`asinestep`** trajectory followed by joint E1 for various controllers.

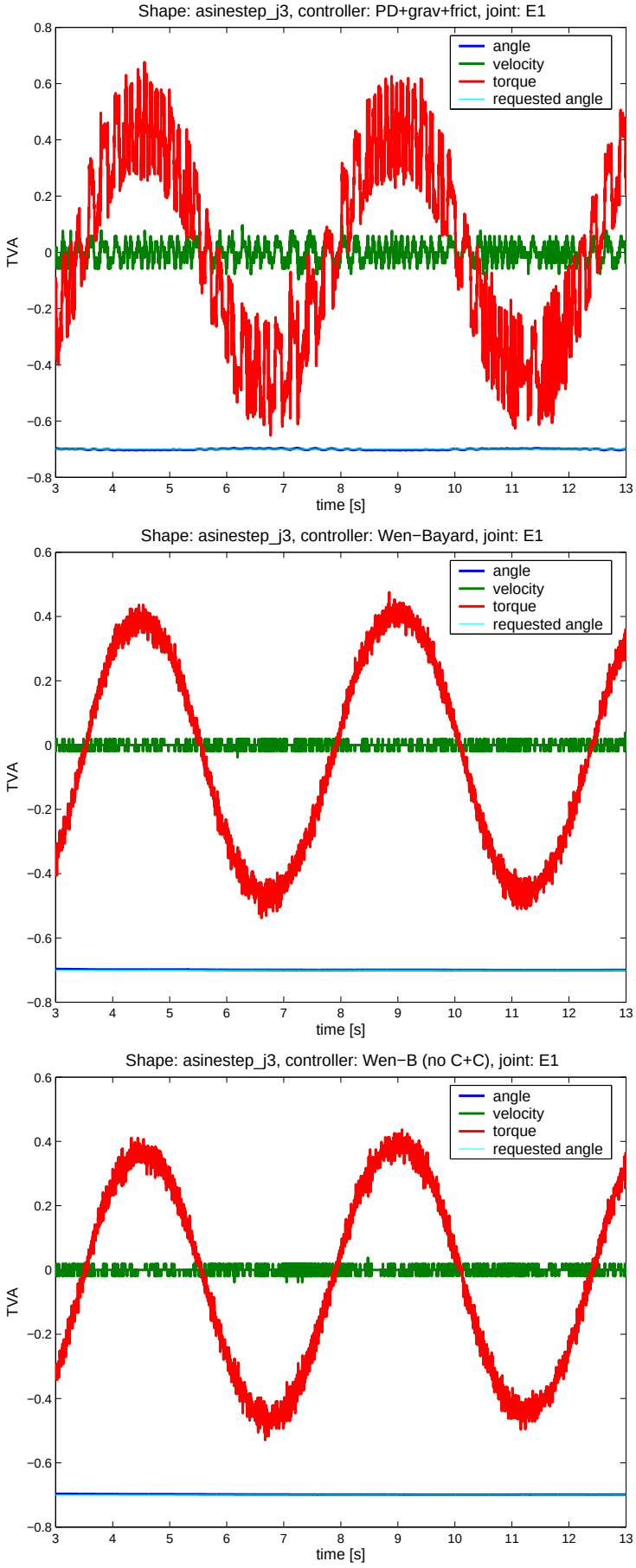


Figure 5.17: The stationary period of **`asinestep`** trajectory followed by joint E1 for various controllers.

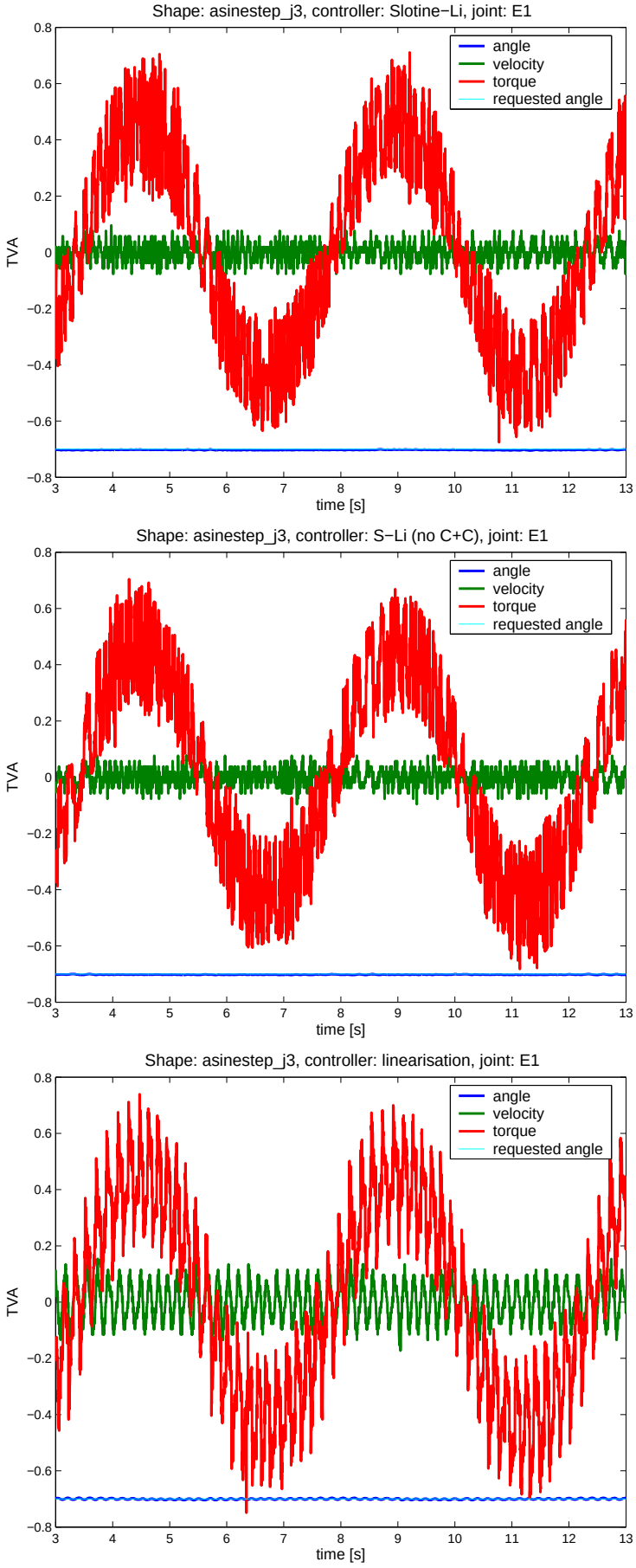


Figure 5.18: The stationary period of **`asinestep`** trajectory followed by joint E1 for various controllers.

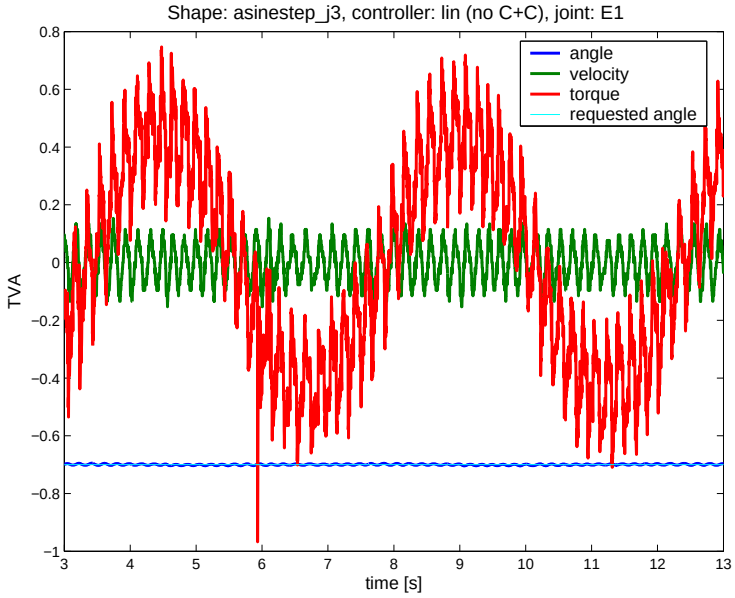


Figure 5.19: The stationary period of **`asinestep`** trajectory followed by joint E1 for different controllers.

controller	S1	S2	E1	E2	W1	W2
PD	0.1425	1.01e-2	5.32e-3	6.59e-2	2.43e-2	2.748
PID	0.3863	9.65e-3	4.70e-3	2.01e-2	1.16e-2	0.1027
PD+gravity	0.2199	7.93e-3	4.45e-3	0.3646	1.53e-2	7.291
PD+grav+frict	0.3287	6.08e-3	2.92e-2	0.1598	0.5078	0.7056
Wen-Bayard	9.12e-2	9.07e-3	3.31e-3	7.21e-3	3.73e-2	0.2496
Wen-B (no C+C)	0.2755	1.08e-2	3.93e-3	9.12e-2	1.94e-2	0.6414
Slotine-Li	0.8195	9.10e-3	4.31e-2	6.391	1.986	2.641
S-Li (no C+C)	0.5528	9.88e-3	4.88e-2	0.9706	1.717	1.531
linearisation	0.2329	1.28e-2	2.38e-2	0.1719	0.1914	1.48e-2
lin (no C+C)	0.2435	9.85e-3	2.51e-2	0.1884	0.1667	2.67e-2

Table 5.6: High to low frequency control power ratio, trajectory: asinestep.

The observations made based on the TVA plots in figures 5.12 to 5.19 can now be confirmed with the control quality indices from figure 5.20 and tables 5.5, 5.6.

- It can be noticed that the lack of gravity feedback in the controller strongly affects the joints with significant mass (S2, E1), completely disqualifying the PD and PID controllers.
- The PD+grav+frict method can be considered to be a well balanced controller delivering very good results across most joints (except tracking for E1 and controller 'noise' for E1 and E2), which could indicate some issues with the friction model for these joints.
- The PA10 velocity controller has been excluded from this analysis due to it not being able to send a step change in position using a velocity command (insufficient scale).

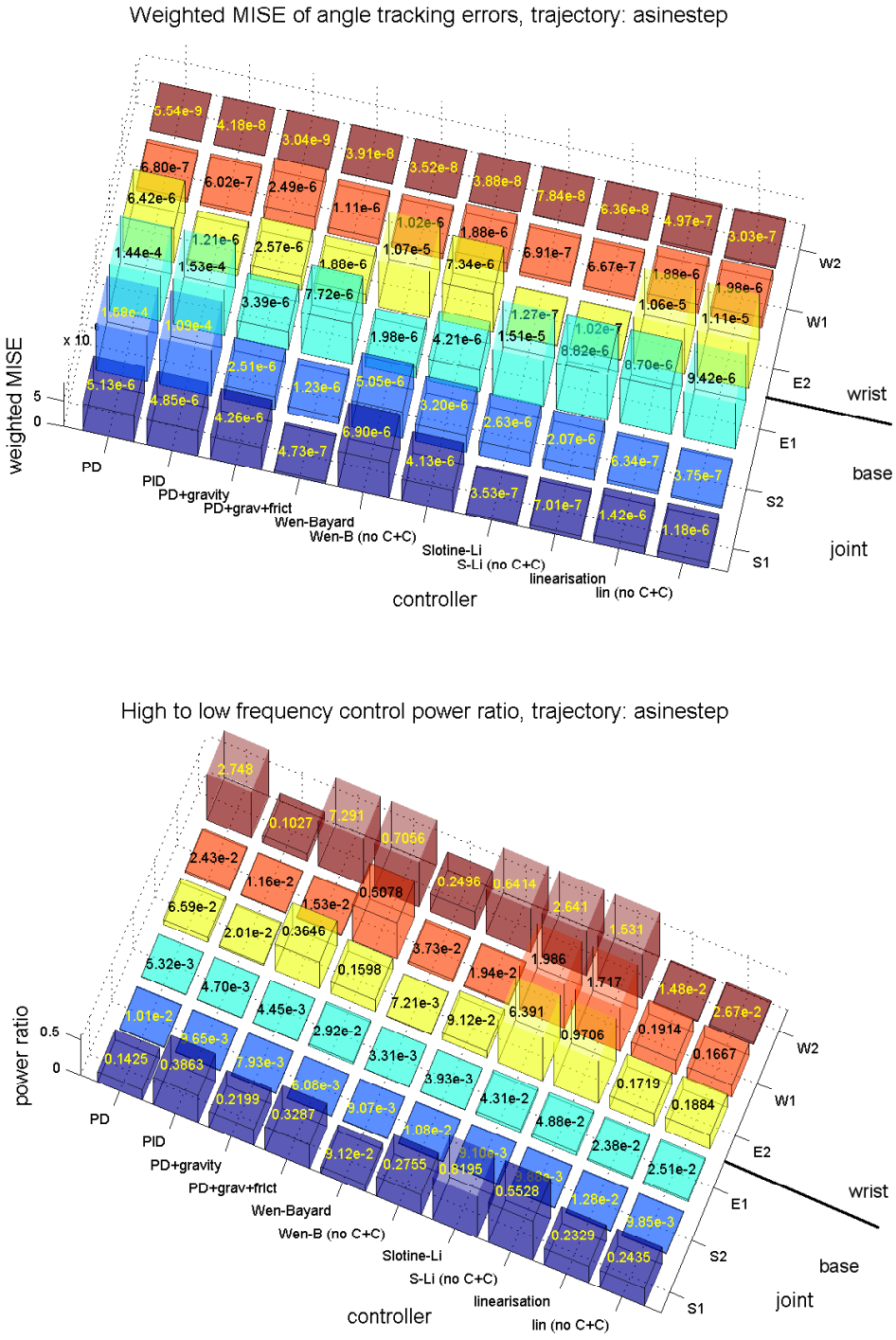


Figure 5.20: Control quality measures for the still period of **`asinestep`** trajectory.

- The controllers without Coriolis and centrifugal term show similar measures to the full model based controller, however, it can be noticed that in some cases the WMISE is smaller for the incomplete controller (this may indicate modelling errors or amplified velocity measurement noise).

- The Wen-Bayard controller shows very good WMCNR for all joints. The linearisation controller has better results only for W2, however it fails to deliver the same angle tracking quality.

controller	S1	S2	E1	E2	W1	W2
PD	1.08e-5	1.46e-4	2.23e-5	1.22e-5	1.61e-5	5.70e-6
PID	1.06e-5	7.23e-5	1.96e-5	9.54e-6	1.22e-5	4.52e-6
PD+gravity	1.05e-5	1.49e-5	2.09e-5	1.12e-5	1.27e-5	4.51e-6
PD+grav+frict	8.64e-6	1.11e-5	1.34e-5	5.02e-6	6.20e-6	5.55e-6
Wen-Bayard	7.87e-6	1.01e-5	1.28e-5	4.91e-6	5.46e-6	5.53e-6
Wen-B (no C+C)	8.04e-6	1.07e-5	1.41e-5	4.88e-6	5.56e-6	5.66e-6
Slotine-Li	8.91e-6	1.28e-5	2.70e-5	3.58e-6	6.14e-6	5.45e-6
S-Li (no C+C)	8.28e-6	1.11e-5	2.69e-5	3.21e-6	6.78e-6	5.50e-6
linearisation	1.46e-5	7.13e-6	1.63e-5	1.09e-5	1.08e-5	4.79e-6
lin (no C+C)	1.45e-5	6.97e-6	1.65e-5	1.04e-5	9.69e-6	4.47e-6
PA10 velocity	1.11e-2	2.40e-5	2.62e-5	1.28e-5	1.29e-5	1.33e-5

Table 5.7: Weighted MISE of angle tracking errors, trajectory: ssine.

controller	S1	S2	E1	E2	W1	W2
PD	0.1043	1.83e-2	4.08e-2	3.051	1.82e-2	6.47e-2
PID	7.77e-2	2.24e-2	2.57e-2	2.993	1.08e-2	3.40e-2
PD+gravity	9.59e-2	2.92e-2	3.02e-2	3.184	1.76e-2	4.25e-2
PD+grav+frict	0.1167	2.18e-2	2.26e-2	1.532	7.80e-3	2.36e-2
Wen-Bayard	9.33e-2	2.31e-2	2.53e-2	2.037	8.70e-3	2.59e-2
Wen-B (no C+C)	0.1000	2.38e-2	4.51e-2	3.159	1.00e-2	3.73e-2
Slotine-Li	0.1353	3.43e-2	6.19e-2	10.060	1.85e-2	6.01e-2
S-Li (no C+C)	0.1257	3.59e-2	5.26e-2	10.100	2.43e-2	5.18e-2
linearisation	1.60e-2	1.06e-2	3.14e-2	9.89e-3	5.65e-3	2.31e-2
lin (no C+C)	4.00e-2	8.19e-3	2.96e-2	9.47e-3	4.89e-3	1.41e-2
PA10 velocity	4.13e-2	3.40e-3	2.05e-2	0.1025	4.07e-3	1.96e-2

Table 5.8: High to low frequency control power ratio, trajectory: ssine.

After analysing the **ssine** and **sine** trajectories (figures 5.21, 5.22 and tables 5.7, 5.8, 5.9 and 5.10) the following can be observed:

- Similar to previous plots, the joints affected by gravity (S2, E1) have very poor tracking errors for controllers without the gravity feedback.

- The full linearisation controller shows very good WMCNR for all joints for both trajectories, however it is not best in terms of angle tracking errors.

- The last joint (W2) shows very similar performance for all controllers, with the PD and PID controllers having a slight advantage.
- The WMCNR for joint E2 is high for all controllers except the feedback linearisation controller. This can be attributed to a different velocity feedback level.
- The PA10 velocity controller is unable to track the trajectory for S1 due to internal velocity limits, which is reflected in the WMISE.
- The PA10 velocity controller shows mixed results for other joints. This can be partly attributed to the fact that velocity is the input parameter and any initial position error (before the control starts) will be preserved throughout the duration of the trajectory.
- It is evident, however, that the velocity controller produces very little high frequency torque element, but still not enough to overtake the linearisation controller measure for joint E2.
- The Wen-Bayard controller shows a very balanced performance for all joints for both WMISE and WMCNR.

controller	S1	S2	E1	E2	W1	W2
PD	1.14e-5	2.45e-4	1.37e-4	9.61e-6	1.49e-5	3.87e-6
PID	1.27e-5	2.02e-4	1.43e-4	1.29e-5	1.25e-5	4.13e-6
PD+gravity	1.22e-5	2.51e-5	1.36e-5	9.34e-6	1.29e-5	3.55e-6
PD+grav+frict	8.04e-6	1.21e-5	2.45e-5	5.88e-6	1.58e-5	6.27e-6
Wen-Bayard	7.86e-6	1.32e-5	1.11e-5	5.72e-6	1.39e-5	6.22e-6
Wen-B (no C+C)	5.90e-6	1.25e-5	1.51e-5	6.18e-6	1.48e-5	6.33e-6
Slotine-Li	9.17e-6	1.35e-5	1.85e-5	3.69e-6	1.53e-5	6.20e-6
S-Li (no C+C)	8.07e-6	1.28e-5	2.19e-5	3.21e-6	1.49e-5	6.23e-6
linearisation	1.14e-5	6.24e-6	9.01e-6	1.39e-5	2.05e-5	5.27e-6
lin (no C+C)	1.02e-5	6.01e-6	1.01e-5	1.41e-5	2.17e-5	5.19e-6
PA10 velocity	1.11e-2	3.97e-5	5.32e-5	2.83e-5	8.39e-5	3.19e-5

Table 5.9: Weighted MISE of angle tracking errors, trajectory: sine.

controller	S1	S2	E1	E2	W1	W2
PD	0.1181	1.53e-2	9.25e-3	0.8756	6.43e-3	2.59e-2
PID	7.41e-2	1.82e-2	8.05e-3	3.118	8.10e-3	2.22e-2
PD+gravity	7.14e-2	1.18e-2	9.05e-3	1.720	7.83e-3	3.15e-2
PD+grav+frict	7.31e-2	1.02e-2	9.12e-3	2.640	1.42e-2	2.38e-2
Wen-Bayard	8.94e-2	1.21e-2	1.64e-2	2.036	1.26e-2	2.57e-2
Wen-B (no C+C)	0.1303	1.16e-2	1.61e-2	2.616	7.97e-3	2.50e-2
Slotine-Li	0.1262	3.63e-2	2.40e-2	7.245	1.88e-2	4.67e-2
S-Li (no C+C)	0.1554	2.47e-2	2.55e-2	8.162	1.41e-2	4.29e-2
linearisation	1.99e-2	8.77e-3	1.30e-2	6.68e-3	5.59e-3	1.48e-2
lin (no C+C)	2.38e-2	7.71e-3	1.27e-2	7.25e-3	5.18e-3	1.47e-2
PA10 velocity	3.59e-2	3.29e-3	7.63e-3	0.1058	3.19e-3	2.42e-2

Table 5.10: High to low frequency control power ratio, trajectory: sine.

5.3 Summary

In this chapter various manipulator torque control techniques were presented and utilised to verify the validity of the dynamic model derived in Chapter 4. The following observations can be made based on the obtained results:

- The control experimental results confirm the conclusion from the PID controlled simulation vs. experiment comparisons about the model terms significance. The gravity and friction terms play the most important role in decreasing the angle tracking errors.
- On the other hand, the experimental results show that the inclusion of additional terms in the feedback loop can cause control *noise* issues or even poorer angle tracking results.
- The results confirm the good immunity of the Wen-Bayard to measurement noise.

- The results show the difficulty of obtaining a good model (and therefore a good model based controller). The choice is between computational complexity (and work required in model derivation), angle tracking performance and generation of unwanted high frequency control torques depending on the actual task to be performed.
  - The gravity term is a minimum requirement for any further control tasks.
  - The friction term improves the tracking quality at the cost of additional complexity and possible instability issued due to the discontinuous friction nature around zero velocity. The requirement of a dead zone is a compromise between the model exactness and the application feasibility.
  - Adding the remaining model terms to the controller can further improve the angle tracking, however caution is required with feedback gains.
  - The PID control comparison task concluded the insignificance of the Coriolis and Centrifugal term and was fully confirmed by the model based controller experiments.
  - The Coriolis and centrifugal model term in feedback control of the PA10-6CE can be omitted without significant sacrifice of tracking quality (this can be attributed to the high gear ratio). This leads to a major reduction in the numerical complexity of the model.
- The internal PA10 velocity control mode delivers good results, however, it contains tighter velocity limits, leading to increased task execution time.

In the analysis above the manipulator was mounted on a fixed base, and there was no disturbance affecting the dynamic model. This is not necessarily the case in real-life robotic implementations.

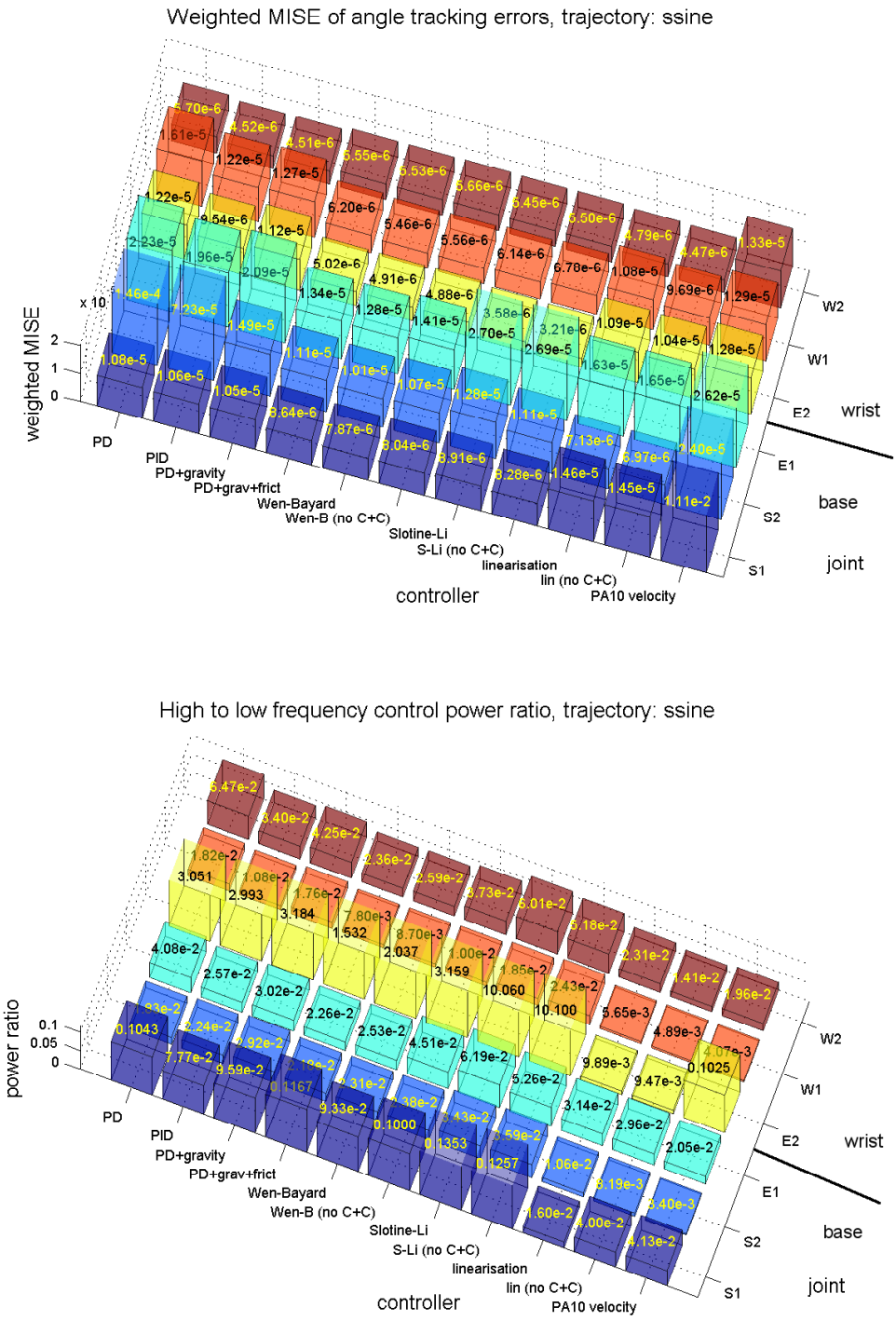


Figure 5.21: Control quality measures for **`ssine`** trajectory.

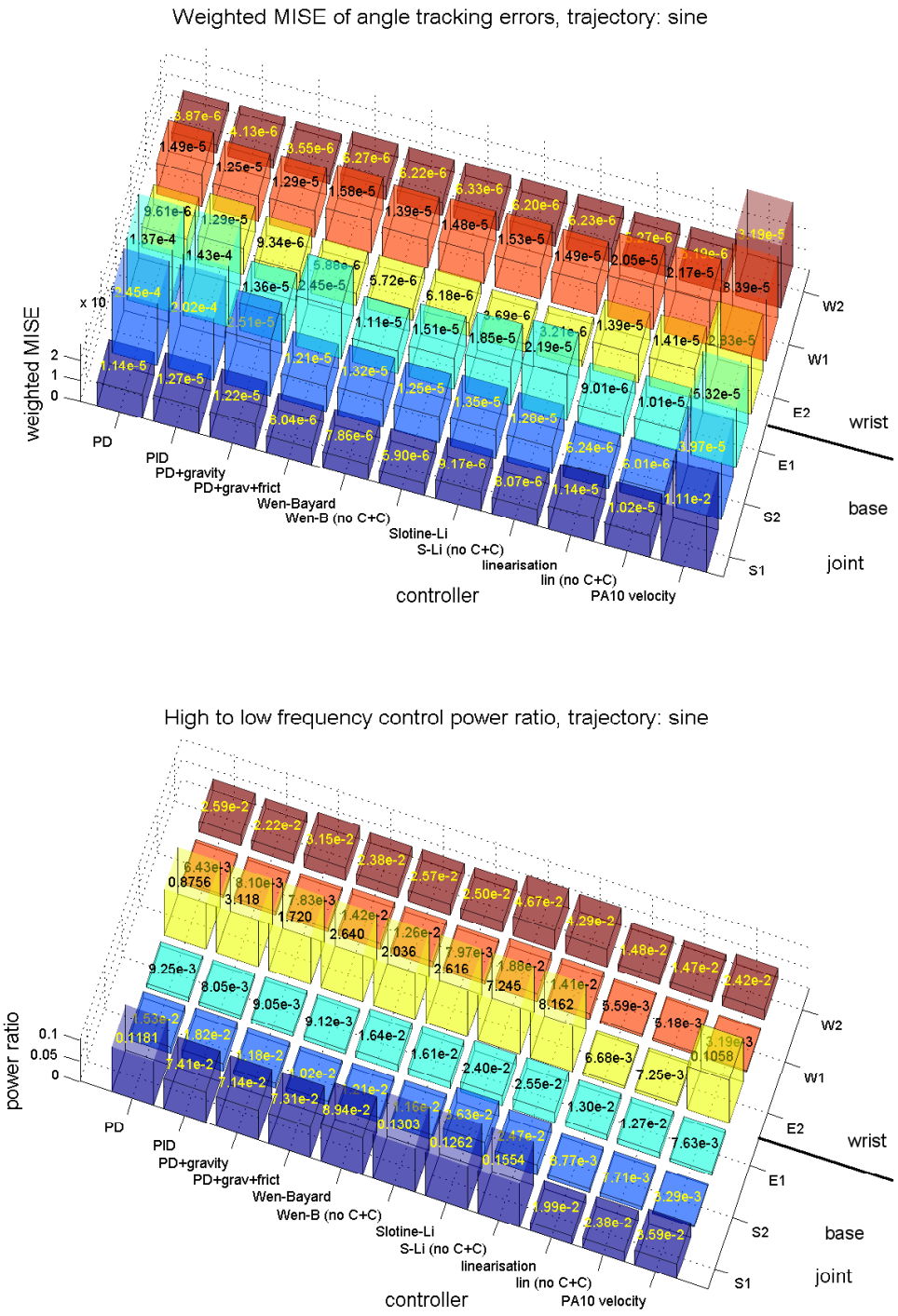


Figure 5.22: Control quality measures for **`sine`** trajectory.

Chapter 6

Description of base disturbance

6.1 Motion platform

To examine the control of a robotic manipulator with its base attached to a non-inertial coordinate frame, a motion platform can be used.
The Intelligent Robotic Systems Laboratory at Heriot-Watt university has been equipped with a 6-DOF CueSim platform [6.2] (figure 6.1). The characteristics of the motion platform are summarised in table 6.1 and figure 6.2.

Parameter	Value	Parameter	Value
Surge range	0.570m	Pitch range	$\pm 21^\circ$
Sway range	0.520m	Roll range	$\pm 21^\circ$
Heave range	0.350m	Yaw range	$\pm 30^\circ$
Peak surge/sway velocity	0.7m/s	Peak pitch/roll rate	40°/s
Peak heave velocity	0.6m/s	Peak yaw rate	60°/s

Table 6.1: MotionBase CueSim 6-DOF motion platform parameters.

A mounting plate has been installed on the frame of the CueSim motion platform, this allows other devices to be mounted on it. The PA10-6CE manipulator has been mounted on the CueSim platform and this allows the base of the manipulator to be moved (disturbed) along any achievable 6-DOF trajectory.
The platform can be controlled by sending $(x, y, z, \text{roll}, \text{pitch}, \text{yaw})$ commands at a rate of 33 Hz (T=0.03 s). The actual platform trajectory can be recorded by receiving the last measured $(x, y, z, \text{roll}, \text{pitch}, \text{yaw})$ position (derived from the platform actuator positions by the CueSim control software). This control frequency is significantly lower than the one used for the PA10 manipulator. A single platform control cycle lasts 15 (=0.03s/0.002s) PA10 control cycles. The control application is synchronised with the PA10 resource manager (manipulator control commands are computed at 500 Hz) and polls the platform resource manager (every cycle) to send/receive data when it is ready for a new control cycle. This has been done due to a requirement for jitter-less control of the manipulator and synchronised data logging. In this way, the platform command trajectory is delivered

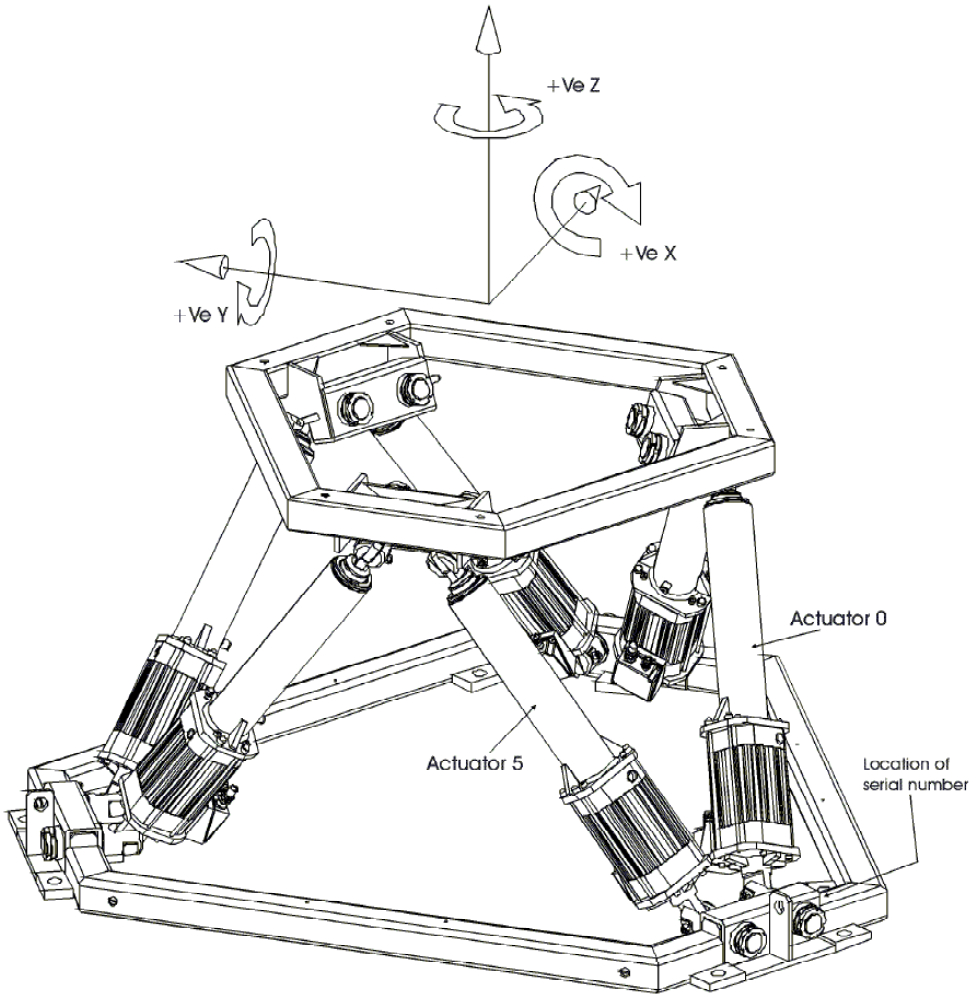


Figure 6.1: Schematic of MotionBase CueSim 6-DOF motion platform.

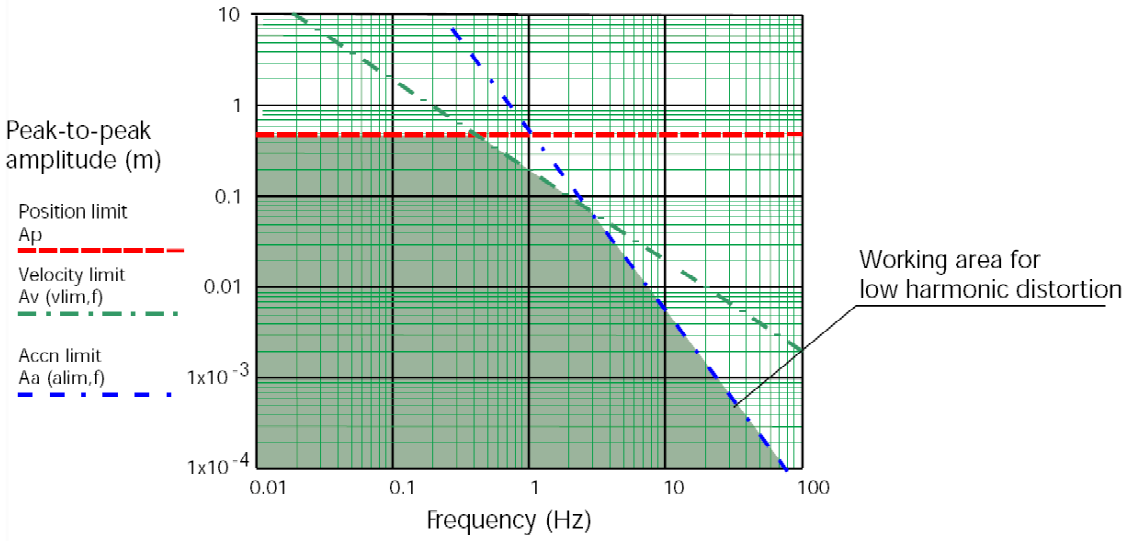


Figure 6.2: MotionBase CueSim 6-DOF motion platform bandwidth.

to its control board at the required rate. Output queueing (of length 1) is applied between the resource manager and the client application. The availability of a new read-out packet

triggers a notify event being passed to the control application, which reads the current data and writes a new desired position in the next PA10 control cycle. The new position is passed to the platform controller motion queueing algorithm at the next internal control cycle.

6.2 Disturbance trajectories

It has been decided that for simplification of the simulation model, only roll and pitch (see section 3.1) motion will be considered. It has been verified in the next chapters, that the control results hold for all 6-DOF base motion. Furthermore, the available sensors allow for additional comparisons for such type of trajectories. It needs to be noted that due to the non-zero distance of the manipulator base from the motion platform reference origin, the manipulator is also affected by linear accelerations along its Z-axis in roll and pitch trajectories.
The following types of disturbance are used to evaluate the robustness of the control algorithms presented in Chapter 5:

- sine trajectory followed by 1-DOF (roll or pitch)

$$q(t) = A\sin\left(\frac{2\pi t}{T} + \varphi\right) \tag{6.1}$$

  where T is the period ($T = 2.25$ s), A is the amplitude[1] ($A_{\text{roll}} = 0.291$ rad, $A_{\text{pitch}} = 0.274$ rad) and φ is the phase ($\varphi = 0$). These trajectories are referred to as **roll sin** and **pitch sin**.

- sine trajectory followed by 2-DOF (roll and pitch)
  - same periods

$$q_i(t) = A_i\sin\left(\frac{2\pi t}{T} + \varphi_i\right) \qquad \text{for } i \in \{\text{roll}, \text{pitch}\} \tag{6.2}$$

    with the following parameters $T = 4.5$ s, $A_{\text{roll}} = 0.274$ rad, $A_{\text{pitch}} = 0.274$ rad and $\varphi_{\text{roll}} = \frac{\pi}{2}$ rad. This trajectory is referred to as **roll + pitch sin equal periods**.
  - different periods

$$q_i(t) = A_i\sin\left(\frac{2\pi t}{T_i} + \varphi_i\right) \qquad \text{for } i \in \{\text{roll}, \text{pitch}\} \tag{6.3}$$

    with the following parameters $T_{\text{roll}} = 4.5$ s, $T_{\text{pitch}} = 2.25$ s, $A_{\text{roll}} = 0.219$ rad, $A_{\text{pitch}} = 0.219$ rad, $\varphi_{\text{roll}} = \frac{\pi}{2}$ rad and $\varphi_{\text{pitch}} = 0$ (the amplitudes are reduced in

[1] The amplitudes have been selected to cover the platform range of motion

comparison to the previous trajectory due to the range envelope shape). This trajectory is referred to as **roll + pitch sin different periods**.

- sum of several sine trajectories followed by 2-DOF (roll and pitch)

$$q_i(t) = \sum_{j=1}^{M} A_{i,j} \sin\left(\frac{2\pi t}{T_{i,j}} + \varphi_{i,j}\right) \qquad \text{for } i \in \{\text{roll}, \text{pitch}\} \tag{6.4}$$

where M is the number of combined sine functions. Two trajectories with $M = 20$ were used, the parameters are presented in tables 6.4, 6.5, 6.6 and 6.7. The amplitudes presented in these tables are only for magnitude reference, the actual trajectory has been re-scaled by the platform software not to exceed the physical limits. The angle ranges for these trajectories are presented in tables 6.2 and 6.3. These trajectories are referred to as **sin sum (1)** and **sin sum (2)**.

Coordinate	Minimum	Maximum
roll	-0.28831	0.29088
pitch	-0.19258	0.24271

Table 6.2: Range of first sum of sines trajectory.

Coordinate	Minimum	Maximum
roll	-0.16596	0.26146
pitch	-0.24968	0.29088

Table 6.3: Range of second sum of sines trajectory.

- sea state vessel motion followed by all 6-DOF[2]. The range for all coordinates is presented in table 6.8.

A ramp is applied to all trajectories (in a post processing step), to ensure they start and end at zero. The ramp is multiplied by the first 1.35 s of the trajectory (it is also applied to the last 1.35 s, but these are not captured in the performed experiments).
All trajectories are presented in figures 6.3 and 6.4. For comparison purposes, the two manipulator trajectories used in the experiments are presented in figure 6.5.

6.3 Sensors

To provide real actual measurements of the platform motion, a set of sensors is located at the origin of the mounting plate. This allows the information about the motion of the platform to be captured. This can be considered to be similar to applications such as a vessel at sea or a vehicle driving on uneven surface. The sensors provide analogue

[2]only in controller comparison experiments

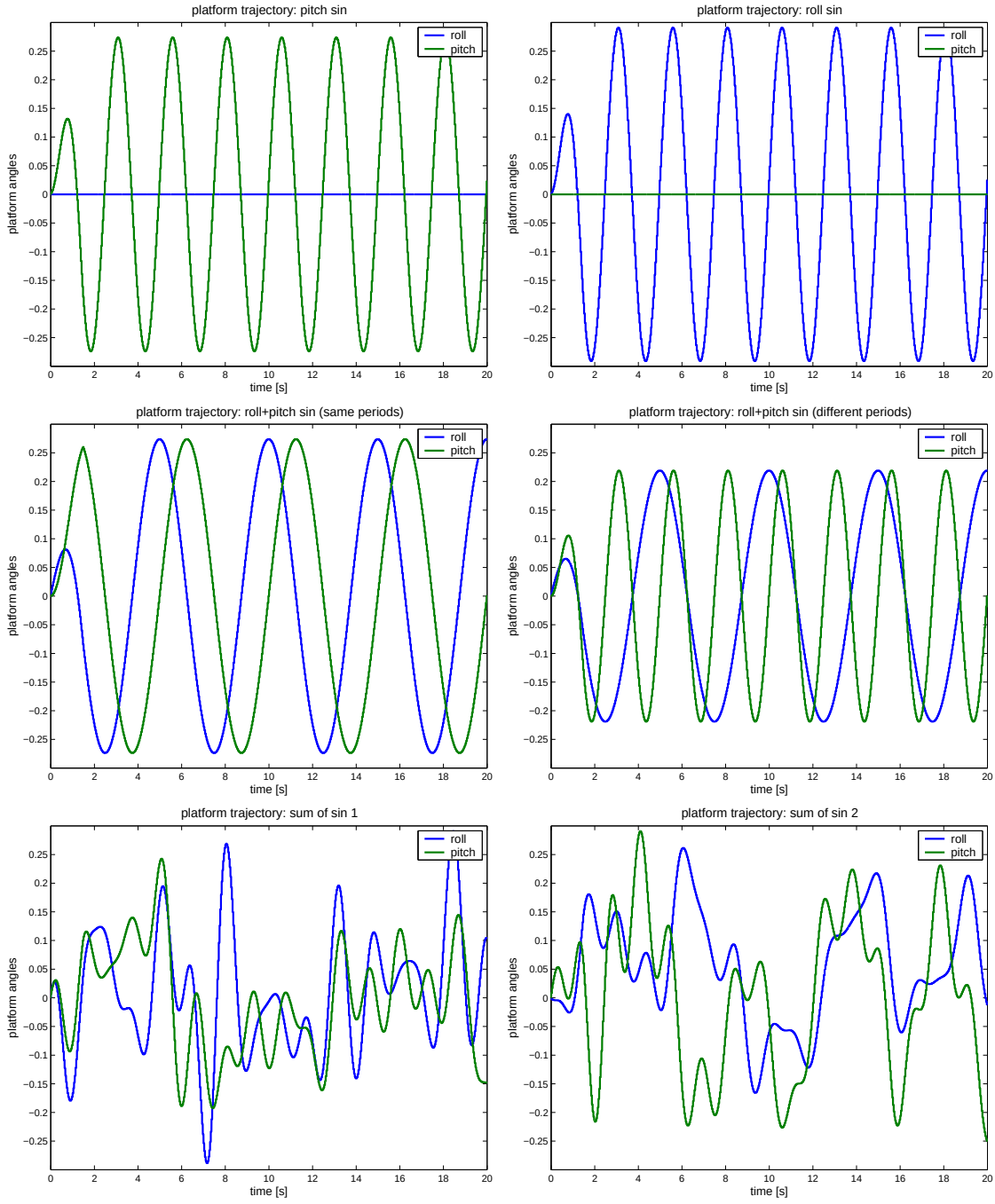


Figure 6.3: Sinusoidal platform trajectories.

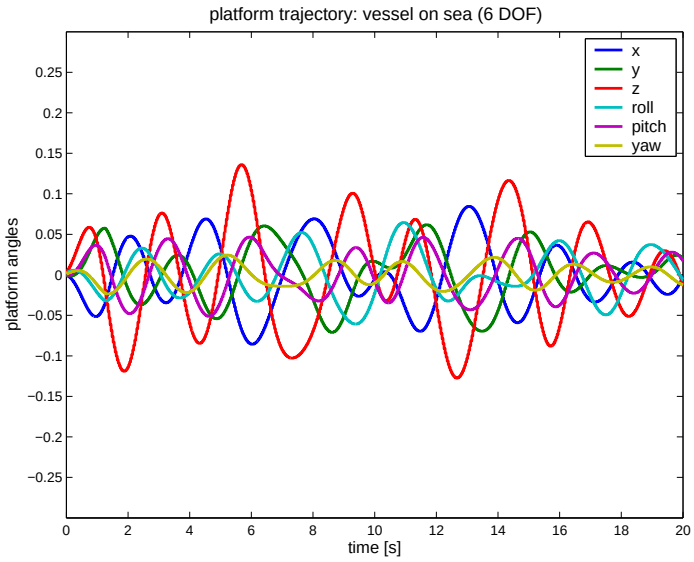


Figure 6.4: Vessel on sea platform trajectory.

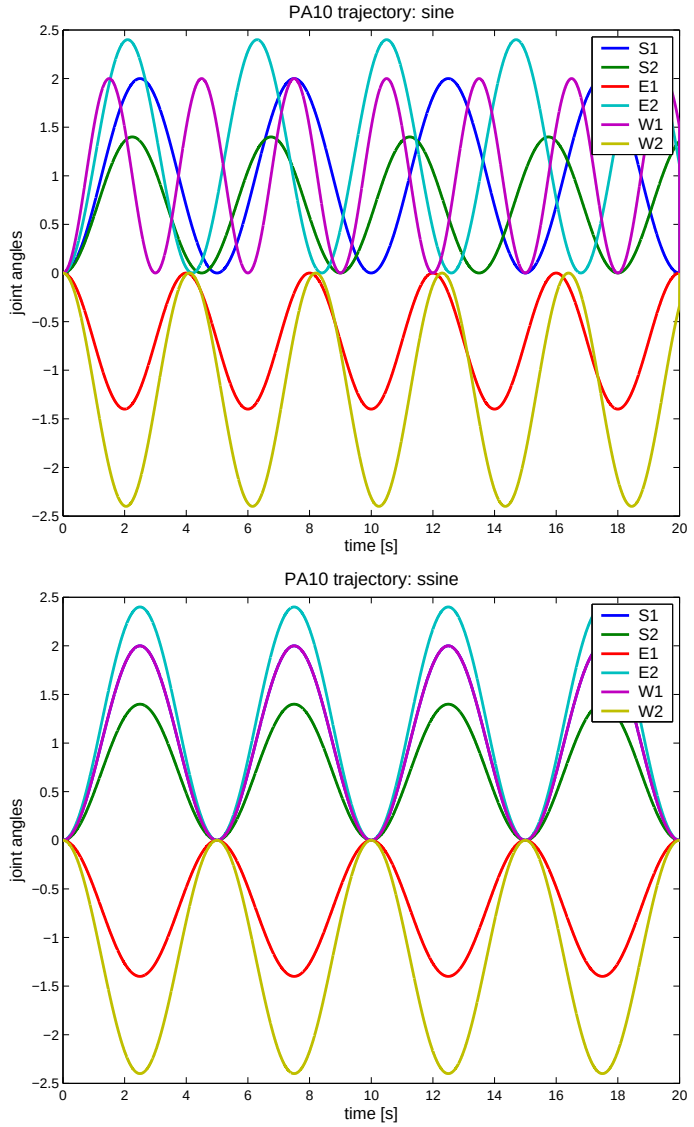


Figure 6.5: Manipulator trajectories used for experiments.

	[s]		[rad]		[rad]
$T_{\text{roll},1}$	5.10960	$A_{\text{roll},1}$	0.14992	$\varphi_{\text{roll},1}$	0.04750
$T_{\text{roll},2}$	12.2966	$A_{\text{roll},2}$	0.43909	$\varphi_{\text{roll},2}$	0.11265
$T_{\text{roll},3}$	2.48570	$A_{\text{roll},3}$	0.85825	$\varphi_{\text{roll},3}$	2.23936
$T_{\text{roll},4}$	1.46829	$A_{\text{roll},4}$	0.20574	$\varphi_{\text{roll},4}$	0.29782
$T_{\text{roll},5}$	4.27670	$A_{\text{roll},5}$	0.64040	$\varphi_{\text{roll},5}$	2.01292
$T_{\text{roll},6}$	1.19266	$A_{\text{roll},6}$	0.81493	$\varphi_{\text{roll},6}$	1.92990
$T_{\text{roll},7}$	2.42551	$A_{\text{roll},7}$	0.02528	$\varphi_{\text{roll},7}$	2.62159
$T_{\text{roll},8}$	2.29850	$A_{\text{roll},8}$	0.82052	$\varphi_{\text{roll},8}$	1.73207
$T_{\text{roll},9}$	1.54599	$A_{\text{roll},9}$	0.99542	$\varphi_{\text{roll},9}$	2.97713
$T_{\text{roll},10}$	1.23211	$A_{\text{roll},10}$	0.34967	$\varphi_{\text{roll},10}$	2.57449
$T_{\text{roll},11}$	2.03151	$A_{\text{roll},11}$	0.71707	$\varphi_{\text{roll},11}$	2.75603
$T_{\text{roll},12}$	2.51779	$A_{\text{roll},12}$	0.66863	$\varphi_{\text{roll},12}$	2.07108
$T_{\text{roll},13}$	1.48456	$A_{\text{roll},13}$	0.73346	$\varphi_{\text{roll},13}$	1.62976
$T_{\text{roll},14}$	1.62695	$A_{\text{roll},14}$	0.45230	$\varphi_{\text{roll},14}$	1.24746
$T_{\text{roll},15}$	1.86862	$A_{\text{roll},15}$	0.86812	$\varphi_{\text{roll},15}$	2.30703
$T_{\text{roll},16}$	9.19682	$A_{\text{roll},16}$	0.51038	$\varphi_{\text{roll},16}$	2.87362
$T_{\text{roll},17}$	2.87840	$A_{\text{roll},17}$	0.92072	$\varphi_{\text{roll},17}$	2.77878
$T_{\text{roll},18}$	1.66364	$A_{\text{roll},18}$	0.38214	$\varphi_{\text{roll},18}$	0.42488
$T_{\text{roll},19}$	1.39600	$A_{\text{roll},19}$	0.38479	$\varphi_{\text{roll},19}$	0.85540
$T_{\text{roll},20}$	11.7693	$A_{\text{roll},20}$	0.50200	$\varphi_{\text{roll},20}$	0.51906

Table 6.4: Roll sin sum (1) trajectory parameters.

signals, processed in a signal conditioning board and measured using a multi-channel (multiplexed) 14-bit ADC board (PC ADDA-14 ISA CARD, FPC-011).
The following sensors were used

- 3-axis accelerometer (Crossbow CXL01LF3) [6.3] with an input range of ($\pm$1 g)
- Gravity referenced servo accelerometer (A223-3001) [6.5] with an input range of (0 to $+$2 g)
- AccuStar electronic Clinometer [6.4] with a linear range of $\pm 45^{\circ}$ and a time constant of 0.3 s.

Considering the control cycle of the PA10 manipulator and the ADC settling times the sensor read-out frequency has been fixed at 1000 Hz. It has further been identified that measurement noise is present in the sensor output. To tackle this issue, a moving average filter is applied to a number of subsequent readings, when data is requested from the sensors' resource-manager. In comparative experiments, similar to those presented in the following chapters, it has been established that a filter length of $n_{avg} = 16$ samples provides a significant reduction of measurement noise without reducing the angle tracking error.
The sensors (ADC) reading is synchronous (at 1 kHz) and the resource manager can handle asynchronous read requests, delivering the results of averaging the last n_{avg} samples. Considering that the manipulator is controlled at 500 Hz the sensor read-out is requested every 2 ADC read cycles.

	[s]		[rad]		[rad]
$T_{\text{pitch},1}$	4.06324	$A_{\text{pitch},1}$	0.05143	$\varphi_{\text{pitch},1}$	1.20501
$T_{\text{pitch},2}$	31.9297	$A_{\text{pitch},2}$	0.83521	$\varphi_{\text{pitch},2}$	2.45745
$T_{\text{pitch},3}$	2.47580	$A_{\text{pitch},3}$	0.59460	$\varphi_{\text{pitch},3}$	2.35738
$T_{\text{pitch},4}$	2.11221	$A_{\text{pitch},4}$	0.22074	$\varphi_{\text{pitch},4}$	0.50270
$T_{\text{pitch},5}$	4.23019	$A_{\text{pitch},5}$	0.24897	$\varphi_{\text{pitch},5}$	0.39753
$T_{\text{pitch},6}$	10.9562	$A_{\text{pitch},6}$	0.51418	$\varphi_{\text{pitch},6}$	0.02032
$T_{\text{pitch},7}$	2.04871	$A_{\text{pitch},7}$	0.15186	$\varphi_{\text{pitch},7}$	2.82389
$T_{\text{pitch},8}$	1.20223	$A_{\text{pitch},8}$	0.98722	$\varphi_{\text{pitch},8}$	1.76859
$T_{\text{pitch},9}$	7.83144	$A_{\text{pitch},9}$	0.17220	$\varphi_{\text{pitch},9}$	1.17212
$T_{\text{pitch},10}$	2.08393	$A_{\text{pitch},10}$	0.76602	$\varphi_{\text{pitch},10}$	2.00820
$T_{\text{pitch},11}$	5.48586	$A_{\text{pitch},11}$	0.89256	$\varphi_{\text{pitch},11}$	2.86282
$T_{\text{pitch},12}$	1.95369	$A_{\text{pitch},12}$	0.06998	$\varphi_{\text{pitch},12}$	1.87358
$T_{\text{pitch},13}$	4.44325	$A_{\text{pitch},13}$	0.89580	$\varphi_{\text{pitch},13}$	2.81447
$T_{\text{pitch},14}$	1.73124	$A_{\text{pitch},14}$	0.60024	$\varphi_{\text{pitch},14}$	3.03547
$T_{\text{pitch},15}$	168.0892	$A_{\text{pitch},15}$	0.37096	$\varphi_{\text{pitch},15}$	2.62288
$T_{\text{pitch},16}$	101.9356	$A_{\text{pitch},16}$	0.13585	$\varphi_{\text{pitch},16}$	1.32684
$T_{\text{pitch},17}$	1.53199	$A_{\text{pitch},17}$	0.76568	$\varphi_{\text{pitch},17}$	0.31636
$T_{\text{pitch},18}$	11.1925	$A_{\text{pitch},18}$	0.84744	$\varphi_{\text{pitch},18}$	0.17649
$T_{\text{pitch},19}$	1.45150	$A_{\text{pitch},19}$	0.72274	$\varphi_{\text{pitch},19}$	1.61947
$T_{\text{pitch},20}$	1.50850	$A_{\text{pitch},20}$	0.68556	$\varphi_{\text{pitch},20}$	1.85122

Table 6.5: Pitch sin sum (1) trajectory parameters.

6.4 Experimental remarks

Sensor calibration The sensor readings were calibrated using the following procedures:

- The angle sensor readings have been adjusted using the platform position read-outs as reference – platform was moved to extreme angle positions and the values from ADC board were read. Appropriate constants were chosen to provide matching measurements.
- The accelerometer readings were adjusted in a similar way. The platform was moved to its extreme angles (roll and pitch) and the rotated gravity value has been used as a reference for choosing the calibration constants.

The calibration has been confirmed to match the sensor specification sheets (considering the applied signal conditioning).

Platform control and read-outs The platform control board applies its own *motion queueing* ([6.1]) algorithm for position control. This results in the actual trajectory lagging the desired one.
New platform position readings from the controller are only available every 15 cycles of the manipulator control, therefore they are not used for any control applications.
For the purpose of using the platform position reference in MATLAB simulations, it is required to have not only the positions but also the velocities and accelerations. Due to

	[s]		[rad]		[rad]
$T_{\text{roll},1}$	2.46405	$A_{\text{roll},1}$	0.15861	$\varphi_{\text{roll},1}$	5.36716
$T_{\text{roll},2}$	3.61870	$A_{\text{roll},2}$	0.65991	$\varphi_{\text{roll},2}$	4.11499
$T_{\text{roll},3}$	1.20887	$A_{\text{roll},3}$	0.18404	$\varphi_{\text{roll},3}$	5.01785
$T_{\text{roll},4}$	2.31423	$A_{\text{roll},4}$	0.40777	$\varphi_{\text{roll},4}$	6.15066
$T_{\text{roll},5}$	4.98739	$A_{\text{roll},5}$	0.84419	$\varphi_{\text{roll},5}$	5.29788
$T_{\text{roll},6}$	2.09296	$A_{\text{roll},6}$	0.98519	$\varphi_{\text{roll},6}$	6.01578
$T_{\text{roll},7}$	10.4126	$A_{\text{roll},7}$	0.33450	$\varphi_{\text{roll},7}$	5.01947
$T_{\text{roll},8}$	9.48570	$A_{\text{roll},8}$	0.88544	$\varphi_{\text{roll},8}$	4.89229
$T_{\text{roll},9}$	7.67286	$A_{\text{roll},9}$	0.08412	$\varphi_{\text{roll},9}$	0.96777
$T_{\text{roll},10}$	7.89435	$A_{\text{roll},10}$	0.61968	$\varphi_{\text{roll},10}$	5.78139
$T_{\text{roll},11}$	2.09514	$A_{\text{roll},11}$	0.80266	$\varphi_{\text{roll},11}$	3.06933
$T_{\text{roll},12}$	4.12560	$A_{\text{roll},12}$	0.39535	$\varphi_{\text{roll},12}$	6.00942
$T_{\text{roll},13}$	1.26907	$A_{\text{roll},13}$	0.57505	$\varphi_{\text{roll},13}$	5.93053
$T_{\text{roll},14}$	6.03252	$A_{\text{roll},14}$	0.94817	$\varphi_{\text{roll},14}$	1.11766
$T_{\text{roll},15}$	78.6510	$A_{\text{roll},15}$	0.77227	$\varphi_{\text{roll},15}$	1.07431
$T_{\text{roll},16}$	10.2546	$A_{\text{roll},16}$	0.76961	$\varphi_{\text{roll},16}$	4.91236
$T_{\text{roll},17}$	1.22322	$A_{\text{roll},17}$	0.09608	$\varphi_{\text{roll},17}$	4.16012
$T_{\text{roll},18}$	1.56233	$A_{\text{roll},18}$	0.22103	$\varphi_{\text{roll},18}$	3.30442
$T_{\text{roll},19}$	1.94079	$A_{\text{roll},19}$	0.88070	$\varphi_{\text{roll},19}$	2.04050
$T_{\text{roll},20}$	1.23465	$A_{\text{roll},20}$	0.78193	$\varphi_{\text{roll},20}$	1.14348

Table 6.6: Roll sin sum (2) trajectory parameters.

the low communication rate with the platform controller, the data needs to be interpolated and smoothed for a robust measurement. Therefore zero-phase filtering (MATLAB `filtfilt`) is applied in post-processing to obtain smooth values for position, velocity and acceleration at the PA10 control sampling rate. The following steps are performed to obtain the values:

- Apply moving average filtering of length 20 to the position measurements (denoted as $\hat{q}_i$).
- Compute $\tilde{\hat{q}}_i = \frac{\hat{q}_{i+1} - \hat{q}_{i-1}}{2}$ and apply moving average filtering of length 80 to obtain the velocity measurements (denoted as $\hat{\hat{q}}_i$).
- Compute $\tilde{\hat{\hat{q}}}_i = \frac{\hat{\hat{q}}_{i+1} - \hat{\hat{q}}_{i-1}}{2}$ and apply moving average filtering of length 120 to obtain the acceleration measurements (denoted as $\hat{\hat{\hat{q}}}_i$).

A sample of these estimates is presented in figure 6.6.

Angle sensor lag The angle sensors have a time constant of 0.3 s, therefore for a typical sinusoidal trajectory the angles measured with these sensors are delayed by approximately 0.3 s. This is presented in figure 6.7 with the lags emphasised using dashed black lines. It can be noticed that the controller read-out lags the desired trajectory by approximately 0.4 s and the sensor read out further lags the actual platform orientation.

	[s]		[rad]		[rad]
$T_{\text{pitch},1}$	13.2285	$A_{\text{pitch},1}$	0.00503	$\varphi_{\text{pitch},1}$	1.41904
$T_{\text{pitch},2}$	4.27251	$A_{\text{pitch},2}$	0.99661	$\varphi_{\text{pitch},2}$	2.98648
$T_{\text{pitch},3}$	8.99840	$A_{\text{pitch},3}$	0.00814	$\varphi_{\text{pitch},3}$	0.32582
$T_{\text{pitch},4}$	1.52019	$A_{\text{pitch},4}$	0.17345	$\varphi_{\text{pitch},4}$	0.74921
$T_{\text{pitch},5}$	1.16714	$A_{\text{pitch},5}$	0.16741	$\varphi_{\text{pitch},5}$	0.55601
$T_{\text{pitch},6}$	2.17026	$A_{\text{pitch},6}$	0.40802	$\varphi_{\text{pitch},6}$	0.78636
$T_{\text{pitch},7}$	6.15938	$A_{\text{pitch},7}$	0.42564	$\varphi_{\text{pitch},7}$	2.85366
$T_{\text{pitch},8}$	1.25600	$A_{\text{pitch},8}$	0.87621	$\varphi_{\text{pitch},8}$	1.57081
$T_{\text{pitch},9}$	2.60176	$A_{\text{pitch},9}$	0.61433	$\varphi_{\text{pitch},9}$	0.52856
$T_{\text{pitch},10}$	1.24069	$A_{\text{pitch},10}$	0.17995	$\varphi_{\text{pitch},10}$	1.50542
$T_{\text{pitch},11}$	11.7316	$A_{\text{pitch},11}$	0.81523	$\varphi_{\text{pitch},11}$	0.17741
$T_{\text{pitch},12}$	4.10167	$A_{\text{pitch},12}$	0.48060	$\varphi_{\text{pitch},12}$	1.16573
$T_{\text{pitch},13}$	4.20518	$A_{\text{pitch},13}$	0.98952	$\varphi_{\text{pitch},13}$	1.86828
$T_{\text{pitch},14}$	3.94561	$A_{\text{pitch},14}$	0.52832	$\varphi_{\text{pitch},14}$	1.40296
$T_{\text{pitch},15}$	11.5416	$A_{\text{pitch},15}$	0.93542	$\varphi_{\text{pitch},15}$	0.98221
$T_{\text{pitch},16}$	1.20726	$A_{\text{pitch},16}$	0.67569	$\varphi_{\text{pitch},16}$	0.94910
$T_{\text{pitch},17}$	1.36003	$A_{\text{pitch},17}$	0.34298	$\varphi_{\text{pitch},17}$	2.87449
$T_{\text{pitch},18}$	102.2762	$A_{\text{pitch},18}$	0.99253	$\varphi_{\text{pitch},18}$	2.66406
$T_{\text{pitch},19}$	1.45198	$A_{\text{pitch},19}$	0.17403	$\varphi_{\text{pitch},19}$	1.94557
$T_{\text{pitch},20}$	1.14841	$A_{\text{pitch},20}$	0.48256	$\varphi_{\text{pitch},20}$	1.85039

Table 6.7: Pitch sin sum (2) trajectory parameters.

Coordinate	Minimum	Maximum
x [m]	-0.0856	0.0844
y [m]	-0.0710	0.0618
z [m]	-0.1274	0.1360
roll [rad]	-0.0607	0.0646
pitch [rad]	-0.0511	0.0466
yaw [rad]	-0.0236	0.0244

Table 6.8: Range of vessel on sea (6DOF) trajectory.

Sensor read-out noise To address the sensor noise, the following steps have been applied

- The data cable is shielded and grounded.
- The ADC board measurement times (programmed in the resource-manager) have been chosen to minimise the measurement noise (resulting from ADC settling times).
- Moving average filtering is applied to the measured values (in post-processing). This introduces an additional phase lag of 0.016 s (8 samples at 500 Hz).

An example of accelerometer sensor noise (with moving average filtering of lengths 2 and 16) is presented in figure 6.8. It can be noticed that even with a moving average filter of length 16, there is a noticeable amount of noise (however it is significantly smaller than for the 2-samples long filter).

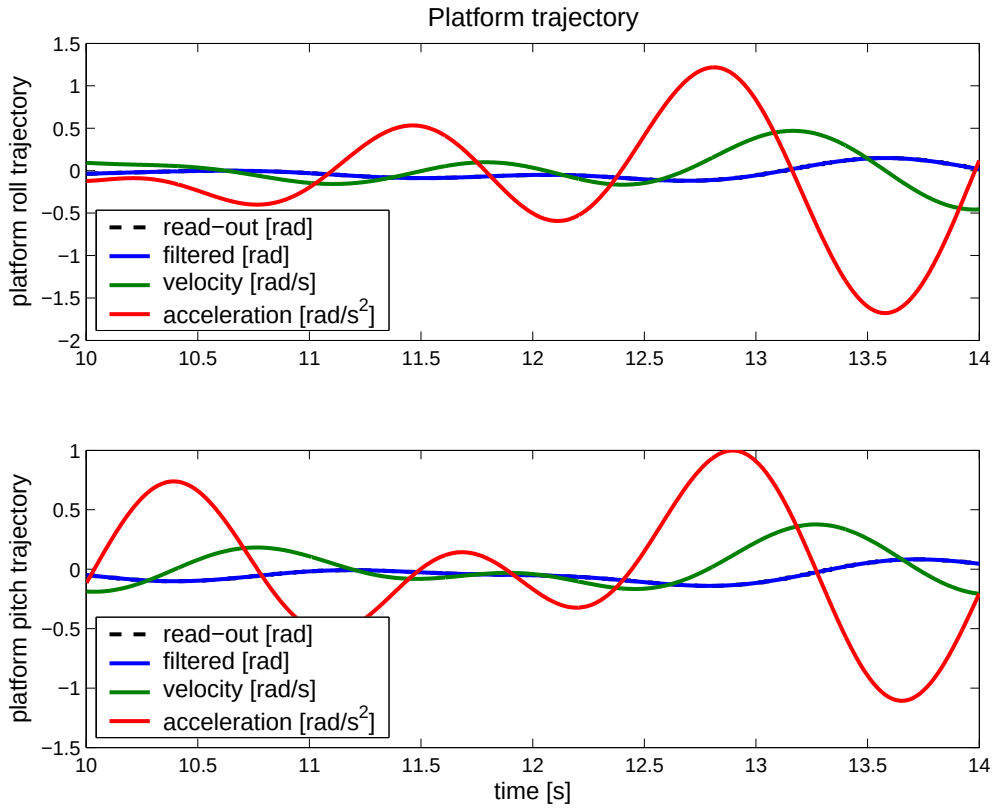


Figure 6.6: Platform trajectory (position, velocity, acceleration). Data recorded for sum of sines (1) trajectory.

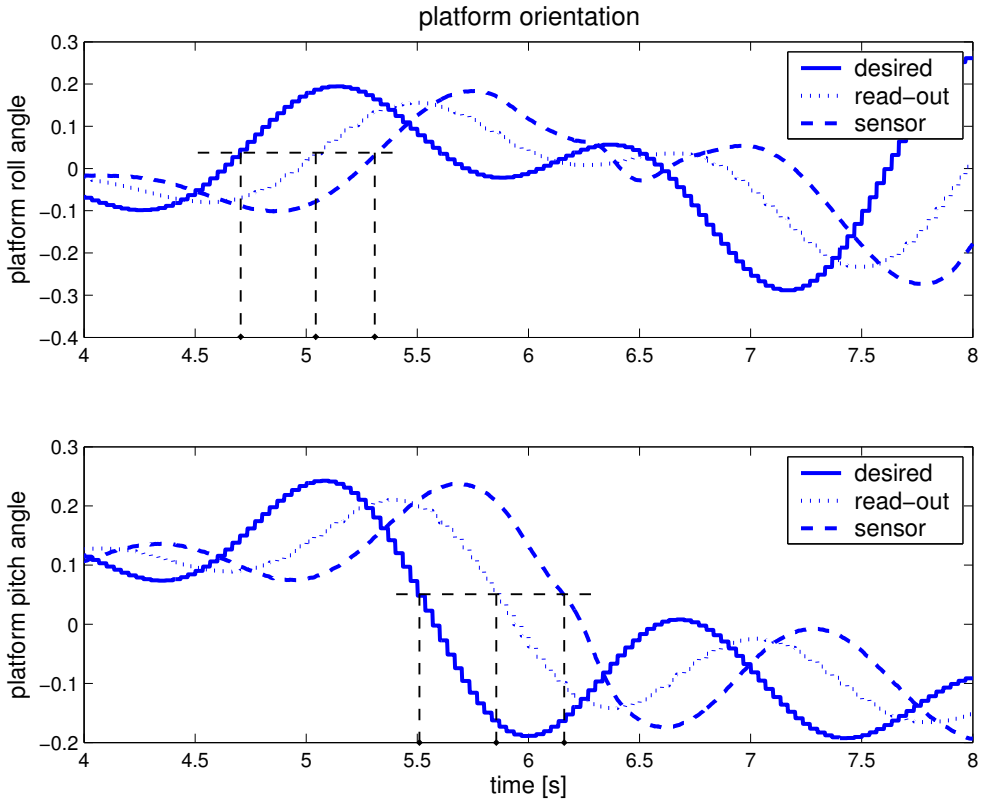


Figure 6.7: Platform trajectory sample (desired, controller read-out and sensor read-out). Data recorded for sum of sines (1) trajectory.

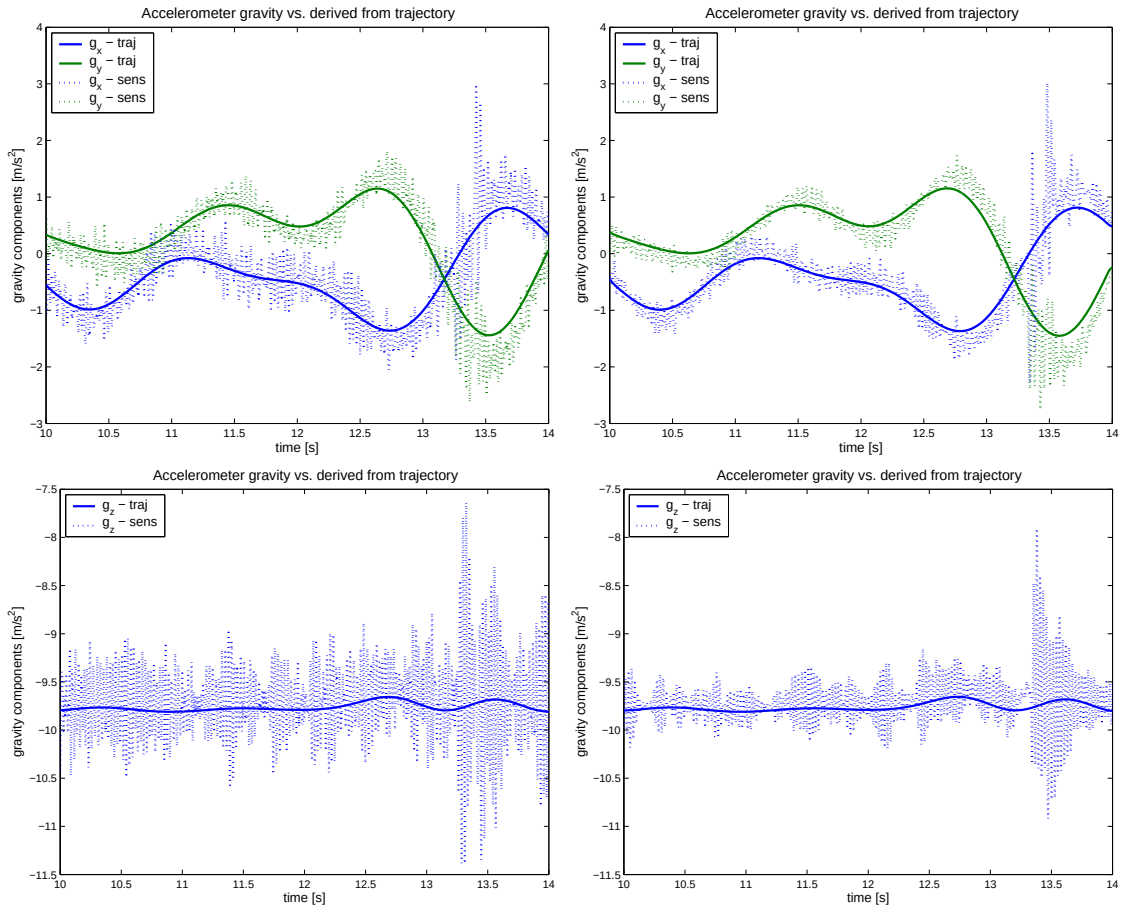


Figure 6.8: Gravity accelerations computed from platform trajectory read-outs and measured with the Crossbow CXL01LF3 accelerometer. Left - filtered with moving average of length 2, right - filtered with moving average of length 16. Data recorded for sum of sines (1) trajectory.

Sensor zero reference Before each experiment (with the platform levelled) the sensor values are read over a period of 1 s (1000 sensor samples at 1 kHz) and used as a reference zero level (or -g for the *z*-axis accelerometers).

***Z*-axis accelerometer** The A223-3001 gravity referenced servo accelerometer has been found to provide measurements comparable with the 3 axis Crossbow CXL01LF3 device. Therefore, its readings will only be recorded for reference/verification.

6.5 Summary

In this chapter the experimental setup for the next two chapters has been presented. An overview of hardware and implementation has been given with solutions used for measurement and control issues. Experimental trajectories used to command the base have been presented with comparison to manipulator trajectories, to visualise the range of motion and frequency.

Chapter 7

Modelling a robotic manipulator on a moving platform

7.1 Introduction

There are numerous situations, where the base of a robotic manipulator is attached to a non-inertial coordinate system, such as a manipulator mounted on a ship or a floating oil rig, or a manipulator mounted on a vehicle driving on uneven terrain. Consider for example a vessel on the sea with a manipulator attached to it. The vessel would be stabilised, however it would not be possible to completely dampen the motion induced by the sea waves. Although this motion would mostly occupy the lower part of the frequency range of the manipulator motion, it could affect the control quality due to unmodelled dynamics. Multiple examples of derivation of models of robotic manipulators on non-inertial bases are presented in section 2.4. In this chapter a full derivation of a generic model of a robotic manipulator on a moving platform is presented (assuming that the platform is not affected by the manipulator motion and its trajectory is considered to be a time varying parameter of the obtained model). An analysis of the model terms is performed by means of similarities to the static base model. An actual symbolic derivation is performed for the PA10-6CE manipulator on a 2-DOF base with roll and pitch angles as time varying parameters of the model and simulation results are presented. The contributions of this chapter are:

- the consideration of the base motion as a time varying parameter of the manipulator on a non-inertial base and resulting simplifications in the dynamic equations,
- a broad analysis (using a comparison of model simulations and manipulator experiments) of the obtained model for the PA10-6CE manipulator with various types of base trajectory by assessing the relative significance of the dynamic model terms per joint.

In order to describe the dynamics of a robotic manipulator on a moving platform it is required to derive the equations of motion.

7.2 Model derivation

The dynamic model of a manipulator on a mobile platform is derived using the Euler-Lagrange approach. In order to perform this, the system energies need to be defined.

7.2.1 Assumptions

The following assumptions are made in the derivation of the model of a robotic manipulator on a moving platform:

- the manipulator motion has no influence on the platform (the mass/inertia of the platform is significantly higher),
- the platform coordinates (and their respective derivatives along a given trajectory) are considered as parameters of the model (**not** as generalised coordinates of the system),
- the harmonic drive flexibility (in the manipulator joints) is not considered,
- motor inertias[1] and friction[2] are not considered in the model derivation, both terms are incorporated in the final equation.

7.2.2 Coordinates transformation

The velocity of a point in a rigid body that is an element of a kinematic chain mounted on a moving base, can be expressed as

$$ {}^0\mathbf{v}_i = \frac{d}{dt}({}^0\mathbf{r}_i) = \frac{d}{dt}(\mathbf{A}_p\,{}^0\mathbf{A}_i\,{}^i\mathbf{r}_i) \tag{7.1} $$

where ${}^0\mathbf{A}_i$ is the Denavit-Hartenberg (see section 3.2.1) transformation of the coordinates from base to *i*-th link, $\mathbf{A}_p$ is a transformation matrix expressing the position of the base (e.g. platform) in the universal coordinates and ${}^i\mathbf{r}_i$ is a point within the *i*-th link expressed in the *i*-th link coordinates. Considering that ${}^i\dot{\mathbf{r}}_i = 0$ and $\frac{d}{dt}(\mathbf{A}_p) = \dot{\mathbf{A}}_p \neq 0$ the following can be derived

$$ \frac{d}{dt}(\mathbf{A}_p\,{}^0\mathbf{A}_i\,{}^i\mathbf{r}_i) = (\dot{\mathbf{A}}_p\,{}^0\mathbf{A}_i + \mathbf{A}_p \sum_{j=1}^{i} \frac{\partial\,{}^0\mathbf{A}_i}{\partial q_j}\dot{q}_j)\,{}^i\mathbf{r}_i \tag{7.2} $$

let $\mathbf{U}_{ij} = \frac{\partial\,{}^0\mathbf{A}_i}{\partial q_j}$ which can also be expressed as

$$ \frac{\partial\,{}^0\mathbf{A}_i}{\partial q_j} = {}^0\mathbf{A}_{j-i}\,\frac{\partial\,{}^{j-1}\mathbf{A}_j}{\partial q_j}\,{}^j\mathbf{A}_i \tag{7.3} $$

[1] to simplify the dynamics model derivation process

[2] it is not affected by the non-inertial base

and the differentiation can be expressed with an algebraic operation of multiplication by the matrix Q_d

$$\frac{\partial {}^0\mathbf{A}_i}{\partial q_j} = {}^0\mathbf{A}_{j-i}\, Q_d\, {}^{j-1}\mathbf{A}_j\, {}^j\mathbf{A}_i \tag{7.4}$$

the Q_d has a different form for prismatic and revolute joints ([7.1]). Considering that the PA10 manipulator has only revolute joints, the following is used

$$Q_d = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{7.5}$$

The velocity vector (calculated along the trajectory of the base) can be expressed as

$${}^0\mathbf{v}_i = (\dot{\mathbf{A}}_p\, {}^0\mathbf{A}_i + \mathbf{A}_p \sum_{j=1}^{i} \mathbf{U}_{ij}\dot{q}_j)\, {}^i\mathbf{r}_i \tag{7.6}$$

7.2.3 Kinetic energy

The kinetic energy for a single point of mass, *m*, can be defined as

$$dK_i = \frac{1}{2}\,\mathrm{Tr}({}^0\mathbf{v}_i({}^0\mathbf{v}_i)^T)dm \tag{7.7}$$

after substituting 7.6

$$dK_i = \frac{1}{2}\,\mathrm{Tr}((\dot{\mathbf{A}}_p\, {}^0\mathbf{A}_i + \mathbf{A}_p \sum_{j=1}^{i} \mathbf{U}_{ij}\dot{q}_j)\, {}^i\mathbf{r}_i((\dot{\mathbf{A}}_p\, {}^0\mathbf{A}_i + \mathbf{A}_p \sum_{j=1}^{i} \mathbf{U}_{ij}\dot{q}_j)\, {}^i\mathbf{r}_i)^T)dm \tag{7.8}$$

after integrating over the link volume

$$K_i = \frac{1}{2}\,\mathrm{Tr}((\dot{\mathbf{A}}_p\, {}^0\mathbf{A}_i + \mathbf{A}_p \sum_{j=1}^{i} \mathbf{U}_{ij}\dot{q}_j)\mathbf{J}_i((\dot{\mathbf{A}}_p\, {}^0\mathbf{A}_i + \mathbf{A}_p \sum_{j=1}^{i} \mathbf{U}_{ij}\dot{q}_j))^T) \tag{7.9}$$

where $\mathbf{J}_i$ has been defined in section 3.4.1. Expanding the product of sums and simplifying

$$\begin{aligned} K_i = {} & \frac{1}{2}\,\mathrm{Tr}((\dot{\mathbf{A}}_p\, {}^0\mathbf{A}_i\, \mathbf{J}_i\, {}^0\mathbf{A}_i^T\, \dot{\mathbf{A}}_p^T) + \\ & + \sum_{r=1}^{i} \mathrm{Tr}(\dot{\mathbf{A}}_p\, {}^0\mathbf{A}_i\, \mathbf{J}_i \mathbf{U}_{ir}^T \mathbf{A}_p^T)\dot{q}_r + \\ & + \frac{1}{2}\sum_{r=1}^{i}\sum_{s=1}^{i} \mathrm{Tr}(\mathbf{A}_p \mathbf{U}_{ir} \mathbf{J}_i \mathbf{U}_{is}^T \mathbf{A}_p^T)\dot{q}_r\dot{q}_s. \end{aligned} \tag{7.10}$$

The total kinetic energy of the system is the sum over all joints

$$\begin{aligned} K = &\sum_{i=1}^{n} \frac{1}{2} \mathrm{Tr}((\dot{\mathbf{A}}_p {}^0\mathbf{A}_i \mathbf{J}_i {}^0\mathbf{A}_i^T \dot{\mathbf{A}}_p^T) + \\ &+ \sum_{i=1}^{n} \sum_{r=1}^{i} \mathrm{Tr}(\dot{\mathbf{A}}_p {}^0\mathbf{A}_i \mathbf{J}_i \mathbf{U}_{ir}^T \mathbf{A}_p^T) \dot{q}_r + \\ &+ \frac{1}{2} \sum_{i=1}^{n} \sum_{r=1}^{i} \sum_{s=1}^{i} \mathrm{Tr}(\mathbf{A}_p \mathbf{U}_{ir} \mathbf{J}_i \mathbf{U}_{is}^T \mathbf{A}_p^T) \dot{q}_r \dot{q}_s. \end{aligned} \tag{7.11}$$

Since only the kinetic energy depends on the generalised velocities

$$\begin{aligned} \frac{\partial K}{\partial \dot{q}_i} = &\sum_{j=i}^{n} \mathrm{Tr}(\dot{\mathbf{A}}_p {}^0\mathbf{A}_j \mathbf{J}_j \mathbf{U}_{ji}^T \mathbf{A}_p^T) + \\ &+ \frac{1}{2} \sum_{j=i}^{n} \sum_{r=1}^{j} \mathrm{Tr}(\mathbf{A}_p \mathbf{U}_{jr} \mathbf{J}_j \mathbf{U}_{ji}^T \mathbf{A}_p^T) \dot{q}_r + \\ &+ \frac{1}{2} \sum_{j=i}^{n} \sum_{s=1}^{j} \mathrm{Tr}(\mathbf{A}_p \mathbf{U}_{ji} \mathbf{J}_j \mathbf{U}_{js}^T \mathbf{A}_p^T) \dot{q}_s \end{aligned} \tag{7.12}$$

and simplifying

$$\begin{aligned} \frac{\partial K}{\partial \dot{q}_i} = &\sum_{j=i}^{n} \mathrm{Tr}(\dot{\mathbf{A}}_p {}^0\mathbf{A}_j \mathbf{J}_j \mathbf{U}_{ji}^T \mathbf{A}_p^T) + \\ &+ \sum_{j=i}^{n} \sum_{r=1}^{j} \mathrm{Tr}(\mathbf{A}_p \mathbf{U}_{jr} \mathbf{J}_j \mathbf{U}_{ji}^T \mathbf{A}_p^T) \dot{q}_r \end{aligned} \tag{7.13}$$

differentiating over time (along the trajectory of the base)

$$\begin{aligned} \frac{d}{dt}(\frac{\partial K}{\partial \dot{q}_i}) = &\sum_{j=i}^{n} \frac{d}{dt}(\mathrm{Tr}(\dot{\mathbf{A}}_p {}^0\mathbf{A}_j \mathbf{J}_j \mathbf{U}_{ji}^T \mathbf{A}_p^T)) + \\ &+ \sum_{j=i}^{n} \sum_{r=1}^{j} \frac{d}{dt}(\mathrm{Tr}(\mathbf{A}_p \mathbf{U}_{jr} \mathbf{J}_j \mathbf{U}_{ji}^T \mathbf{A}_p^T) \dot{q}_r) \end{aligned} \tag{7.14}$$

deriving element by element

$$\begin{aligned}\frac{d}{dt}(\frac{\partial K}{\partial \dot{q}_i}) = &\sum_{j=i}^{n} \mathrm{Tr}(\ddot{\mathbf{A}}_p {}^0\mathbf{A}_j \mathbf{J}_j \mathbf{U}_{ji}^T \mathbf{A}_p^T) + \\ &+ \sum_{j=i}^{n}\sum_{r=1}^{j} \mathrm{Tr}(\dot{\mathbf{A}}_p \mathbf{U}_{jr} \mathbf{J}_j \mathbf{U}_{ji}^T \mathbf{A}_p^T)\dot{q}_r + \\ &+ \sum_{j=i}^{n}\sum_{r=1}^{j} \mathrm{Tr}(\dot{\mathbf{A}}_p {}^0\mathbf{A}_j \mathbf{J}_j \mathbf{U}_{jir}^T \mathbf{A}_p^T)\dot{q}_r + \\ &+ \sum_{j=i}^{n} \mathrm{Tr}(\dot{\mathbf{A}}_p {}^0\mathbf{A}_j \mathbf{J}_j \mathbf{U}_{ji}^T \dot{\mathbf{A}}_p^T) + \\ &+ \sum_{j=i}^{n}\sum_{r=1}^{j} \mathrm{Tr}(\dot{\mathbf{A}}_p \mathbf{U}_{jr} \mathbf{J}_j \mathbf{U}_{ji}^T \mathbf{A}_p^T)\dot{q}_r + \\ &+ \sum_{j=i}^{n}\sum_{r=1}^{j}\sum_{s=1}^{j} \mathrm{Tr}(\mathbf{A}_p \mathbf{U}_{jrs} \mathbf{J}_j \mathbf{U}_{ji}^T \mathbf{A}_p^T)\dot{q}_r\dot{q}_s + \\ &+ \sum_{j=i}^{n}\sum_{r=1}^{j}\sum_{s=1}^{j} \mathrm{Tr}(\mathbf{A}_p \mathbf{U}_{jr} \mathbf{J}_j \mathbf{U}_{jis}^T \mathbf{A}_p^T)\dot{q}_r\dot{q}_s + \\ &+ \sum_{j=i}^{n}\sum_{r=1}^{j} \mathrm{Tr}(\mathbf{A}_p \mathbf{U}_{jr} \mathbf{J}_j \mathbf{U}_{ji}^T \dot{\mathbf{A}}_p^T)\dot{q}_r + \\ &+ \sum_{j=i}^{n}\sum_{r=1}^{j} \mathrm{Tr}(\mathbf{A}_p \mathbf{U}_{jr} \mathbf{J}_j \mathbf{U}_{ji}^T \mathbf{A}_p^T)\ddot{q}_r\end{aligned} \tag{7.15}$$

and the derivative of the kinetic energy over q_i is

$$\begin{aligned}\frac{\partial K}{\partial q_i} = &\frac{\partial}{\partial q_i}(\sum_{j=1}^{n} \frac{1}{2} \mathrm{Tr}(\dot{\mathbf{A}}_p {}^0\mathbf{A}_j \mathbf{J}_j {}^0\mathbf{A}_j{}^T \dot{\mathbf{A}}_p^T)) + \\ &+ \frac{\partial}{\partial q_i}(\sum_{j=1}^{n}\sum_{r=1}^{j} \mathrm{Tr}(\dot{\mathbf{A}}_p {}^0\mathbf{A}_j \mathbf{J}_j \mathbf{U}_{jr}^T \mathbf{A}_p^T)\dot{q}_r) + \\ &+ \frac{\partial}{\partial q_i}(\frac{1}{2}\sum_{j=1}^{n}\sum_{r=1}^{j}\sum_{s=1}^{j} \mathrm{Tr}(\mathbf{A}_p \mathbf{U}_{jr} \mathbf{J}_j \mathbf{U}_{js}^T \mathbf{A}_p^T)\dot{q}_r\dot{q}_s)\end{aligned} \tag{7.16}$$

expanding element by element

$$\begin{aligned}\frac{\partial K}{\partial q_i} &= \sum_{j=i}^{n} \mathrm{Tr}(\dot{\mathbf{A}}_p \mathbf{U}_{ji} \mathbf{J}_j {}^0\mathbf{A}_j{}^T \dot{\mathbf{A}}_p^T) + \\ &+ \sum_{j=i}^{n}\sum_{r=1}^{j} \mathrm{Tr}(\dot{\mathbf{A}}_p \mathbf{U}_{ji} \mathbf{J}_j \mathbf{U}_{jr}^T \mathbf{A}_p^T)\dot{q}_r + \\ &+ \sum_{j=i}^{n}\sum_{r=1}^{j} \mathrm{Tr}(\dot{\mathbf{A}}_p {}^0\mathbf{A}_j \mathbf{J}_j \mathbf{U}_{jri}^T \mathbf{A}_p^T)\dot{q}_r + \\ &+ \sum_{j=i}^{n}\sum_{r=1}^{j}\sum_{s=1}^{j} \mathrm{Tr}(\mathbf{A}_p \mathbf{U}_{jri} \mathbf{J}_j \mathbf{U}_{js}^T \mathbf{A}_p^T)\dot{q}_r \dot{q}_s\end{aligned} \tag{7.17}$$

and subtracting the differentials $\frac{d}{dt}(\frac{\partial K}{\partial \dot{q}_i}) - \frac{\partial K}{\partial q_i}$, using the property $\mathbf{U}_{ijk} = \mathbf{U}_{ikj}$

$$\begin{aligned}\frac{d}{dt}(\frac{\partial K}{\partial \dot{q}_i}) - \frac{\partial K}{\partial q_i} &= \sum_{j=i}^{n} \mathrm{Tr}(\ddot{\mathbf{A}}_p {}^0\mathbf{A}_j \mathbf{J}_j \mathbf{U}_{ji}^T \mathbf{A}_p^T) + \\ &+ 2\sum_{j=i}^{n}\sum_{r=1}^{j} \mathrm{Tr}(\dot{\mathbf{A}}_p \mathbf{U}_{jr} \mathbf{J}_j \mathbf{U}_{ji}^T \mathbf{A}_p^T)\dot{q}_r + \\ &+ \sum_{j=i}^{n}\sum_{r=1}^{j}\sum_{s=1}^{j} \mathrm{Tr}(\mathbf{A}_p \mathbf{U}_{jrs} \mathbf{J}_j \mathbf{U}_{ji}^T \mathbf{A}_p^T)\dot{q}_r \dot{q}_s + \\ &+ \sum_{j=i}^{n}\sum_{r=1}^{j} \mathrm{Tr}(\mathbf{A}_p \mathbf{U}_{jr} \mathbf{J}_j \mathbf{U}_{ji}^T \mathbf{A}_p^T)\ddot{q}_r.\end{aligned} \tag{7.18}$$

7.2.4 Potential energy

Based on the added transformation to the D-H chain

$$P_j = -m_j \mathbf{g}(\mathbf{A}_p {}^0\mathbf{A}_j R_j) \tag{7.19}$$

the total potential energy is

$$P = -\sum_{j=1}^{n} m_j \mathbf{g}(\mathbf{A}_p {}^0\mathbf{A}_j R_j) \tag{7.20}$$

and the differential over q_i is

$$\frac{\partial P}{\partial q_i} = -\sum_{j=i}^{n} m_j \mathbf{g}(\mathbf{A}_p \mathbf{U}_{ji} R_j). \tag{7.21}$$

7.2.5 Lagrange-Euler formulation

The full equations of motion for a manipulator mounted on a mobile platform are

$$\mathbf{u}_i = \frac{d}{dt}(\frac{\partial L}{\partial \dot{q}_i}) - \frac{\partial L}{\partial q_i} = \frac{d}{dt}(\frac{\partial K}{\partial \dot{q}_i}) - \frac{\partial K}{\partial q_i} + \frac{\partial P}{\partial q_i} \tag{7.22}$$

substituting equations 7.18 and 7.21 into 7.22

$$\begin{aligned}\mathbf{u}_i = &\sum_{j=i}^{n} \mathrm{Tr}(\ddot{\mathbf{A}}_p{}^0\mathbf{A}_j \mathbf{J}_j \mathbf{U}_{ji}^T \mathbf{A}_p^T) + \\ &+ 2\sum_{j=i}^{n}\sum_{r=1}^{j} \mathrm{Tr}(\dot{\mathbf{A}}_p \mathbf{U}_{jr} \mathbf{J}_j \mathbf{U}_{ji}^T \mathbf{A}_p^T)\dot{q}_r + \\ &+ \sum_{j=i}^{n}\sum_{r=1}^{j}\sum_{s=1}^{j} \mathrm{Tr}(\mathbf{A}_p \mathbf{U}_{jrs} \mathbf{J}_j \mathbf{U}_{ji}^T \mathbf{A}_p^T)\dot{q}_r\dot{q}_s + \\ &+ \sum_{j=i}^{n}\sum_{r=1}^{j} \mathrm{Tr}(\mathbf{A}_p \mathbf{U}_{jr} \mathbf{J}_j \mathbf{U}_{ji}^T \mathbf{A}_p^T)\ddot{q}_r + \sum_{j=i}^{n} m_j \mathbf{g}(\mathbf{A}_p \mathbf{U}_{ji} R_j).\end{aligned} \tag{7.23}$$

It can be shown that for any coordinate transformation matrix $\mathbf{A}_p$

$$\mathbf{A}_p = \begin{bmatrix} \mathbf{R}_{3\times3} & \mathbf{T}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix},$$

where $\mathbf{R}(\alpha_p)$ is an arbitrary rotation matrix parametrised by the vector $\alpha_p = [\alpha_{p1}, \alpha_{p2}, \alpha_{p3}]$ and $\mathbf{T} = [x_p, y_p, z_p]^T$ is an arbitrary 3×1 translation vector. For any 4×4 matrix $M = [m_{i,j}]$ the following equality holds

$$\begin{aligned}\mathrm{Tr}(\mathbf{A}_p M \mathbf{A}_p^T) = &m_{11} + m_{22} + m_{33} + \\ &+ \mathbf{f}(\alpha_p, x_p, y_p, z_p) \cdot [m_{14}, m_{24}, m_{34}, m_{41}, m_{42}, m_{43}, m_{44}]^T.\end{aligned}$$

Moreover, it can be shown that $\mathbf{U}_{jr}\mathbf{J}_j\mathbf{U}_{ji}^T$ and $\mathbf{U}_{jrs}\mathbf{J}_j\mathbf{U}_{ji}^T$ have zeros in the 4-th row and 4-th column, due to internal multiplication by the *differentiating* matrix Q_d (equation 7.5). Therefore equation 7.23 can be simplified to

$$\begin{aligned}\mathbf{u}_i = &\sum_{j=i}^{n} \mathrm{Tr}(\ddot{\mathbf{A}}_p{}^0\mathbf{A}_j \mathbf{J}_j \mathbf{U}_{ji}^T \mathbf{A}_p^T) + \\ &+ 2\sum_{j=i}^{n}\sum_{r=1}^{j} \mathrm{Tr}(\dot{\mathbf{A}}_p \mathbf{U}_{jr} \mathbf{J}_j \mathbf{U}_{ji}^T \mathbf{A}_p^T)\dot{q}_r + \\ &+ \sum_{j=i}^{n}\sum_{r=1}^{j}\sum_{s=1}^{j} \mathrm{Tr}(\mathbf{U}_{jrs} \mathbf{J}_j \mathbf{U}_{ji}^T)\dot{q}_r\dot{q}_s + \\ &+ \sum_{j=i}^{n}\sum_{r=1}^{j} \mathrm{Tr}(\mathbf{U}_{jr} \mathbf{J}_j \mathbf{U}_{ji}^T)\ddot{q}_r + \sum_{j=i}^{n} m_j \mathbf{g}(\mathbf{A}_p \mathbf{U}_{ji} R_j).\end{aligned} \tag{7.24}$$

It can be noticed that the inertia and Coriolis and centrifugal terms do not depend on the platform coordinates, and simplify to the case of a manipulator on a fixed (displaced) base

$$\mathbf{Q}_{i,r}(\mathbf{q}) = \sum_{j=i}^{n} \mathrm{Tr}(\mathbf{U}_{jr} \mathbf{J}_j \mathbf{U}_{ji}^T) \tag{7.25}$$

and

$$\mathbf{C}_{i,s}(\mathbf{q},\dot{\mathbf{q}}) = \sum_{j=i}^{n}\sum_{r=1}^{j} \mathrm{Tr}(\mathbf{U}_{jrs}\mathbf{J}_j\mathbf{U}_{ji}^T)\dot{q}_r. \tag{7.26}$$

Defining the item independent of the manipulator velocities as

$$\mathbf{PF}_i(\mathbf{q}_p,\dot{\mathbf{q}}_p,\ddot{\mathbf{q}}_p) = \sum_{j=i}^{n} \mathrm{Tr}(\ddot{\mathbf{A}}_p\,{}^0\mathbf{A}_j\,\mathbf{J}_j\mathbf{U}_{ji}^T\mathbf{A}_p^T) \tag{7.27}$$

the item depending on the manipulator velocities as

$$\mathbf{PV}_{i,r}(\mathbf{q}_p,\dot{\mathbf{q}}_p) = \sum_{j=i}^{n}\sum_{r=1}^{j} \mathrm{Tr}(\dot{\mathbf{A}}_p\mathbf{U}_{jr}\mathbf{J}_j\mathbf{U}_{ji}^T\mathbf{A}_p^T) \tag{7.28}$$

and the new gravity vector as

$$\mathbf{g}_{pi}(\mathbf{q},\mathbf{q}_p) = \sum_{j=i}^{n} m_j\mathbf{g}(\mathbf{A}_p\mathbf{U}_{ji}R_j) \tag{7.29}$$

the dynamic equation of a robotic manipulator on a moving platform can be expressed as

$$\mathbf{PF}(\mathbf{q}_p,\dot{\mathbf{q}}_p,\ddot{\mathbf{q}}_p)+\mathbf{PV}(\mathbf{q}_p,\dot{\mathbf{q}}_p)\dot{\mathbf{q}}+\mathbf{Q}(\mathbf{q})\ddot{\mathbf{q}}+\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}+\mathbf{g}_p(\mathbf{q},\mathbf{q}_p)=\mathbf{u}. \tag{7.30}$$

The following can be noticed from the above equation:

- For $\mathbf{A}_p = I_{4\times4}$ and $\ddot{\mathbf{A}}_p = \dot{\mathbf{A}}_p = \mathbf{0}$ the equations are equivalent to a manipulator on a fixed base.
- For a constant $\mathbf{A}_p$ the terms **PF** and **PV** vanish and only the gravity term is affected (this can again be regarded as a model of a manipulator on a fixed base).
- Using the same reasoning as for showing that the inertia and Coriolis and centrifugal terms do not depend on the platform variables, the gravity term can be expressed as

  $$\mathbf{g}_{pi}(\mathbf{q},\mathbf{q}_p) = \mathbf{g}\mathbf{R}_p\sum_{j=i}^{n} m_j(\mathbf{U}_{ji}R_j), \tag{7.31}$$

  where $\mathbf{R}_p$ is the rotation element of the $\mathbf{A}_p$ transformation matrix. This can be interpreted as a rotation of the gravity vector using the platform angular coordinates.

Incorporating friction and motor inertias To complete the dynamic model for the PA10 manipulator, friction and motor inertias need to be added to equation 7.30. These two terms have already been derived in sections 4.3.2 and 4.3.3.4, and can be added directly to the new model:

$$\mathbf{PF}(\mathbf{q}_p,\dot{\mathbf{q}}_p,\ddot{\mathbf{q}}_p)+\mathbf{PV}(\mathbf{q}_p,\dot{\mathbf{q}}_p)\dot{\mathbf{q}}+\mathbf{Q}(\mathbf{q})+\mathbf{Q}_M\ddot{\mathbf{q}}+\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}+\mathbf{g}_p(\mathbf{q},\mathbf{q}_p)+\mathbf{F}(\dot{\mathbf{q}})=\mathbf{u}. \tag{7.32}$$

The obtained model is used for simulating the dynamics of the PA10 manipulator along a given platform/base trajectory.

7.3 Implementation

In order to use the analytical formulae defining the dynamic model in numerical simulations the explicit equations for the PA10-6CE manipulator mounted on a platform need to be derived. The schematic of the PA10-6CE kinematics with the coordinate systems for each link and base coordinate system bound to the centre of the platform mounting plate is presented in figure 7.1.

7.3.1 Platform motion

The above derivation is valid for any coordinate transformation $\mathbf{A}_p$, however only rotation around the X and Y axes are considered. Additionally, since the manipulator is not mounted at the centre of the base, but is shifted along the *x*-axis, a constant translation needs to be applied. The orientation parameters are the roll (α) and pitch (β) angles. The orientation matrix takes the following form

$$\mathbf{R}_p = \mathrm{Rot}(Y,\beta)\cdot\mathrm{Rot}(X,\alpha)$$

using explicit matrix notation

$$\mathbf{R}_p = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix}.$$

Finally, the complete coordinate transformation matrix is

$$\mathbf{A}_p = \begin{bmatrix} \mathbf{R}_{p3\times3} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{1}_{3\times3} & \mathbf{rX}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix},$$

where $\mathbf{rX} = [rX, 0, 0]^T$. Therefore

$$\mathbf{A}_p = \begin{bmatrix} \cos(\beta) & \sin(\alpha)\sin(\beta) & \cos(\alpha)\sin(\beta) & rX\cos(\beta) \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ -\sin(\beta) & \sin(\alpha)\cos(\beta) & \cos(\alpha)\cos(\beta) & -rX\sin(\beta) \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{7.33}$$

It needs to be noted that rX is a constant parameter, and therefore $\frac{d}{dt}(rX) = 0$.

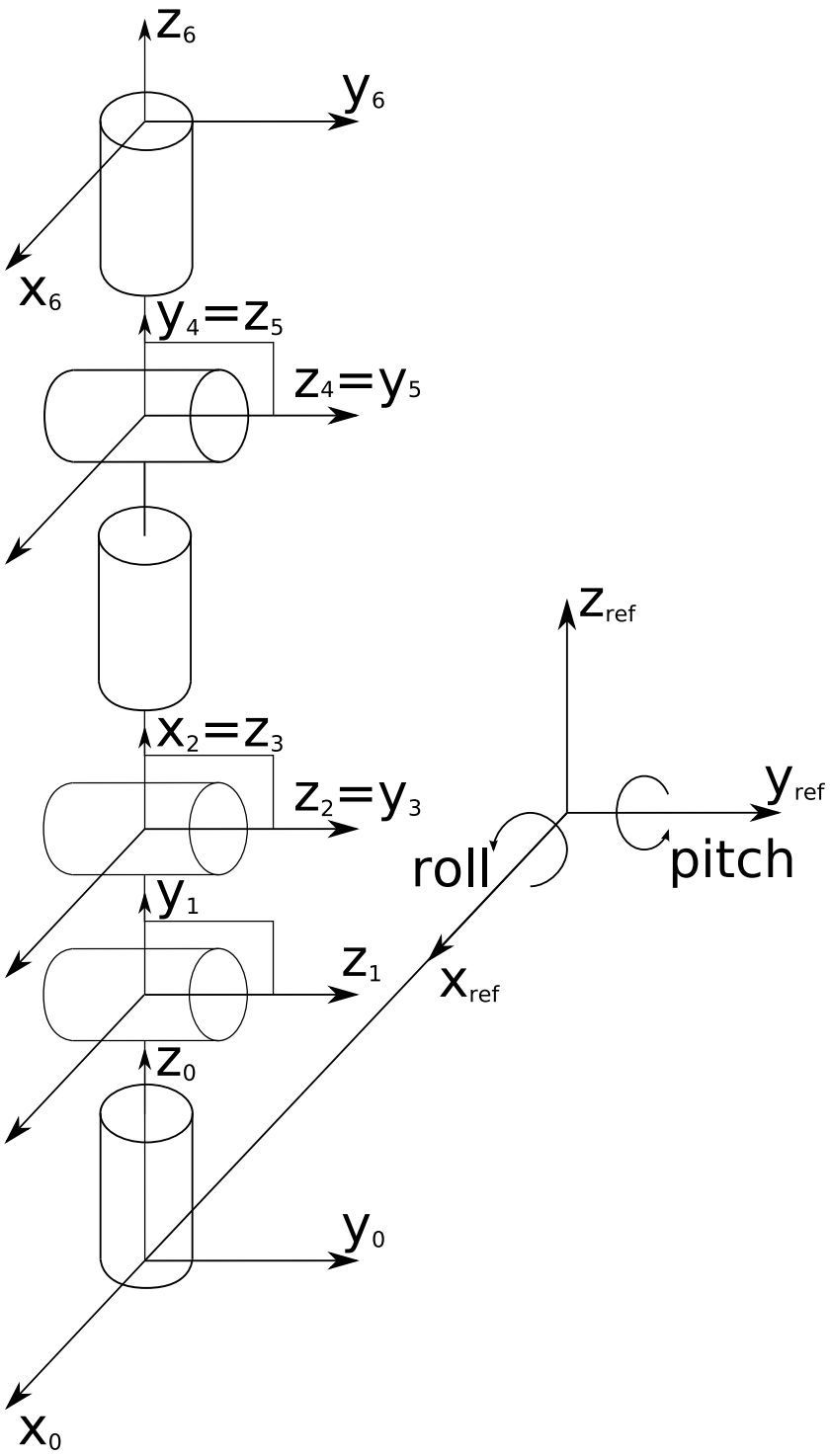


Figure 7.1: Schematic of the PA10-6CE manipulator kinematics on the motion platform with coordinate systems for each link. *Note: index 'ref' denotes the global/reference frame index 0 denotes the manipulator base frame.*

The differential of the $\mathbf{A}_p$ matrix can be obtained as

$$\dot{\mathbf{A}}_p = \frac{d}{dt}(\mathbf{A}_p) = \frac{\partial \mathbf{A}_p}{\partial \alpha}\dot{\alpha} + \frac{\partial \mathbf{A}_p}{\partial \beta}\dot{\beta} \tag{7.34}$$

and similarly the second differential

$$\begin{aligned}
\ddot{\mathbf{A}}_p =& \frac{d}{dt}(\dot{\mathbf{A}}_p) = \frac{d}{dt}(\frac{\partial \mathbf{A}_p}{\partial \alpha}\dot{\alpha}) + \frac{d}{dt}(\frac{\partial \mathbf{A}_p}{\partial \beta}\dot{\beta}) \\
&= \frac{\partial^2 \mathbf{A}_p}{\partial \alpha^2}\dot{\alpha}^2 + \frac{\partial^2 \mathbf{A}_p}{\partial \alpha \partial \beta}\dot{\alpha}\dot{\beta} + \frac{\partial \mathbf{A}_p}{\partial \alpha}\ddot{\alpha} + \frac{\partial^2 \mathbf{A}_p}{\partial \alpha \partial \beta}\dot{\beta}\dot{\alpha} + \frac{\partial^2 \mathbf{A}_p}{\partial \beta^2}\dot{\beta}^2 + \frac{\partial \mathbf{A}_p}{\partial \beta}\ddot{\beta} \\
&= \frac{\partial^2 \mathbf{A}_p}{\partial \alpha^2}\dot{\alpha}^2 + 2\frac{\partial^2 \mathbf{A}_p}{\partial \alpha \partial \beta}\dot{\alpha}\dot{\beta} + \frac{\partial^2 \mathbf{A}_p}{\partial \beta^2}\dot{\beta}^2 + \frac{\partial \mathbf{A}_p}{\partial \alpha}\ddot{\alpha} + \frac{\partial \mathbf{A}_p}{\partial \beta}\ddot{\beta}.
\end{aligned} \tag{7.35}$$

Remark It can be noticed that there are velocity and acceleration dependent terms in the second differential, this indicates that the **PF** element of equation 7.32 includes both inertial-like and Coriolis and centrifugal-like (only those originating purely from the platform velocities) terms. The **PV** term includes Coriolis and centrifugal-like terms originating from the platform-manipulator cross-coupling. The **PF** term should also include acceleration-like terms that have a similar form to the gravity model.

7.3.2 Explicit formulae derivation

The derived analytic formulae describing the platform-induced dynamics terms (equations 7.27, 7.28 and 7.29) and the matrices describing the platform motion ($\mathbf{A}_p$, $\dot{\mathbf{A}}_p$ and $\ddot{\mathbf{A}}_p$) have been used to create Mathematica code to generate the Matlab form of the dynamics equation terms.

7.4 Model evaluation

Due to the high complexity of the platform related dynamics elements, the model cannot be verified by hand. Therefore to quantitatively assess the model, a similar procedure to the one described in section 5.2 has been employed. The PA10-6CE manipulator installed on a moving platform, is controlled with a PID controller, and all the manipulator control variables and the platform trajectory are recorded. The derived equations of the PA10-6CE dynamics on a moving platform (using only roll and pitch angles) have been used in a Matlab simulation (using the same solver as described in section 5.2). The simulation results have been compared to filtered (figure 5.3) experimental torques using the WMISE (equation 5.22) measure.
The following models (based on equation 7.32) are used in this evaluation:

- *full model* - all elements present in the model,
- *no mPF* - the **PF** element not present in the model,

- *no mPV* - the **PV** element not present in the model,
- *no mPV no mPF* - the **PV** and **PF** elements are not present in the model (which effectively leaves only the platform rotated gravity term),
- *no platform* - model of PA10-6CE with fixed horizontal base (equation 3.9).

For the comparison tests the PA10-6CE is commanded along the following trajectories:

- **`sine`** - equation 5.25,
- **`ssine`** - equation 5.24.

7.4.1 Single frequency 1-DOF motion

In this analysis single frequency 1-DOF trajectories are used for the platform (equation 6.1). Comparison of simulation and experimental results are presented in tables 7.1 to 7.4. An example comparison of motor torques for the PA10 trajectory **`sine`** and platform trajectory roll sin is presented in figures 7.2 and 7.3. The corresponding bar plot of WMISE differences is presented in figure 7.4.

model \ *joint*	S1	S2	E1	E2	W1	W2
full model	0.01181	0.00545	0.00913	0.04601	0.00893	0.00510
no mPF	0.02834	0.02290	0.03400	0.04498	0.00904	0.00510
no mPV	0.01240	0.00618	0.01157	0.04629	0.00906	0.00510
no mPV no mPF	0.02927	0.02397	0.03699	0.04523	0.00919	0.00510
no platform	0.12358	0.07982	0.11483	0.04498	0.00948	0.00510

Table 7.1: WMISE experimental and simulation torques comparison, experiment: PA10: **`sine`**, PLAT: roll sin.

model \ *joint*	S1	S2	E1	E2	W1	W2
full model	0.01446	0.01133	0.01778	0.05115	0.01042	0.00477
no mPF	0.02349	0.03625	0.07179	0.05180	0.01096	0.00477
no mPV	0.01489	0.01076	0.01984	0.05102	0.01030	0.00477
no mPV no mPF	0.02407	0.03545	0.07414	0.05170	0.01092	0.00477
no platform	0.06953	0.13419	0.31167	0.05391	0.01149	0.00477

Table 7.2: WMISE experimental and simulation torques comparison, experiment: PA10: **`ssine`**, PLAT: roll sin.

Discussion of results Based on the results the following can be observed:

- The presence of just the platform-rotated gravity element significantly reduces the difference between the simulated model and experiment (the poorest torque match is in the top plot of figure 7.2 and the best match in the bottom plot of figure 7.3).

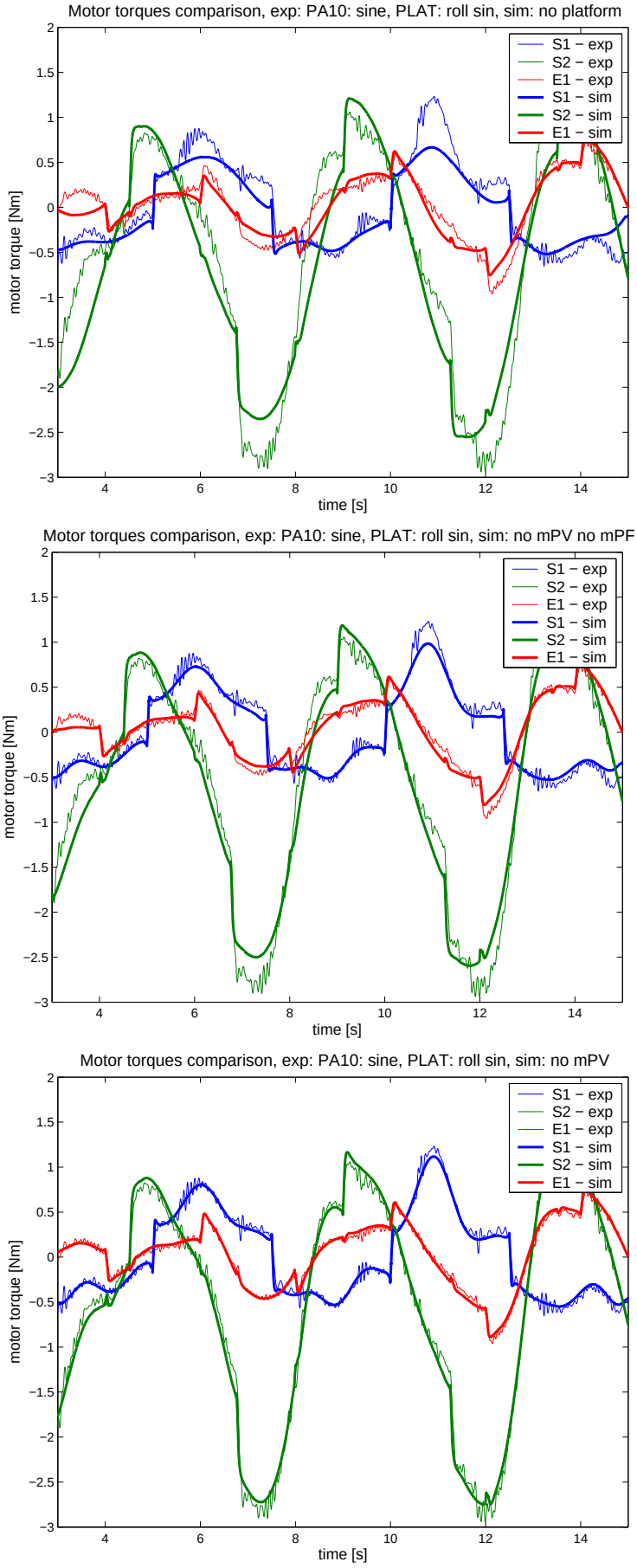


Figure 7.2: Comparison of simulation and experiment torques for different manipulator on platform model set-ups, PA10 trajectory **`sine`**, platform trajectory roll sin.

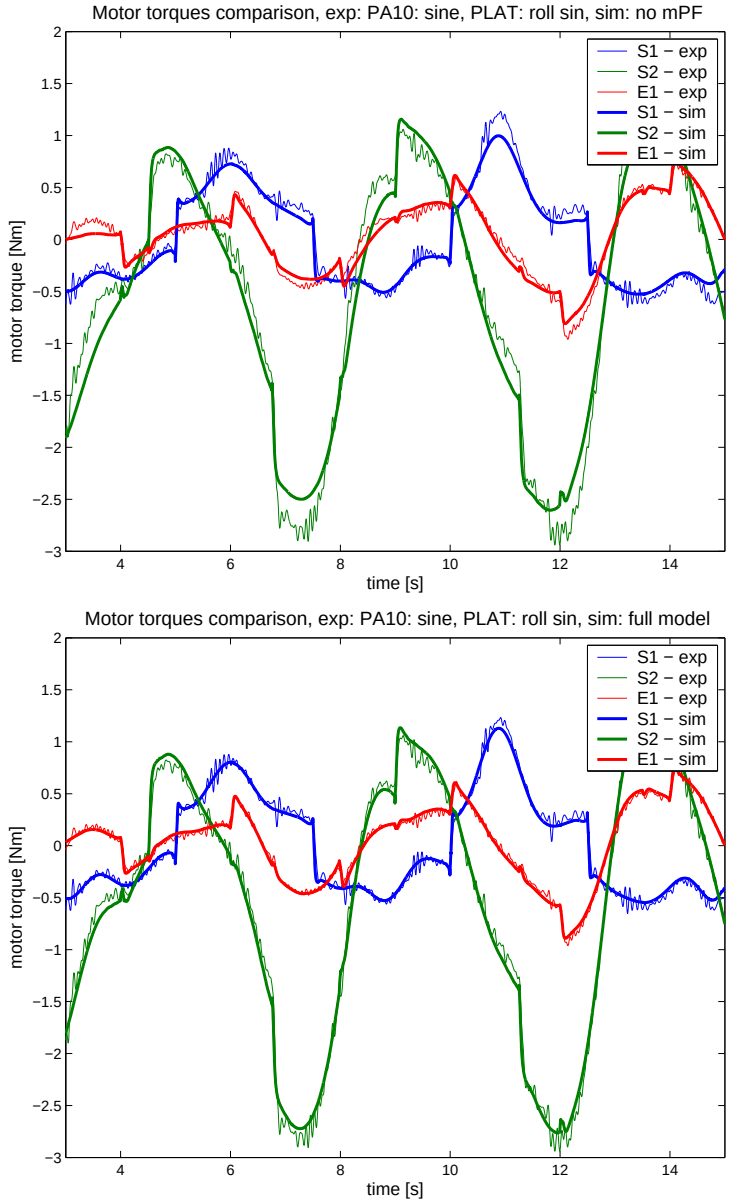


Figure 7.3: Comparison of simulation and experiment torques for different manipulator on platform model set-ups, PA10 trajectory **`sine`**, platform trajectory roll sin.

model \ *joint*	S1	S2	E1	E2	W1	W2
full model	`0.01075`	`0.00395`	`0.00764`	`0.07289`	`0.00926`	`0.00497`
no mPF	`0.03758`	`0.01108`	`0.02309`	`0.07251`	`0.00935`	`0.00497`
no mPV	`0.01253`	`0.00575`	`0.01135`	`0.07359`	`0.00929`	`0.00497`
no mPV no mPF	`0.03739`	`0.01303`	`0.02795`	`0.07320`	`0.00942`	`0.00497`
no platform	`0.21750`	`0.04983`	`0.08751`	`0.07343`	`0.00964`	`0.00497`

Table 7.3: WMISE experimental and simulation torques comparison, experiment: PA10: **`sine`**, PLAT: pitch sin.

- Based on tables 7.1 to 7.4 and figure 7.4 the following can be stated about the platform related model terms:
  - The **PF** element (depending on platform velocities and accelerations) further

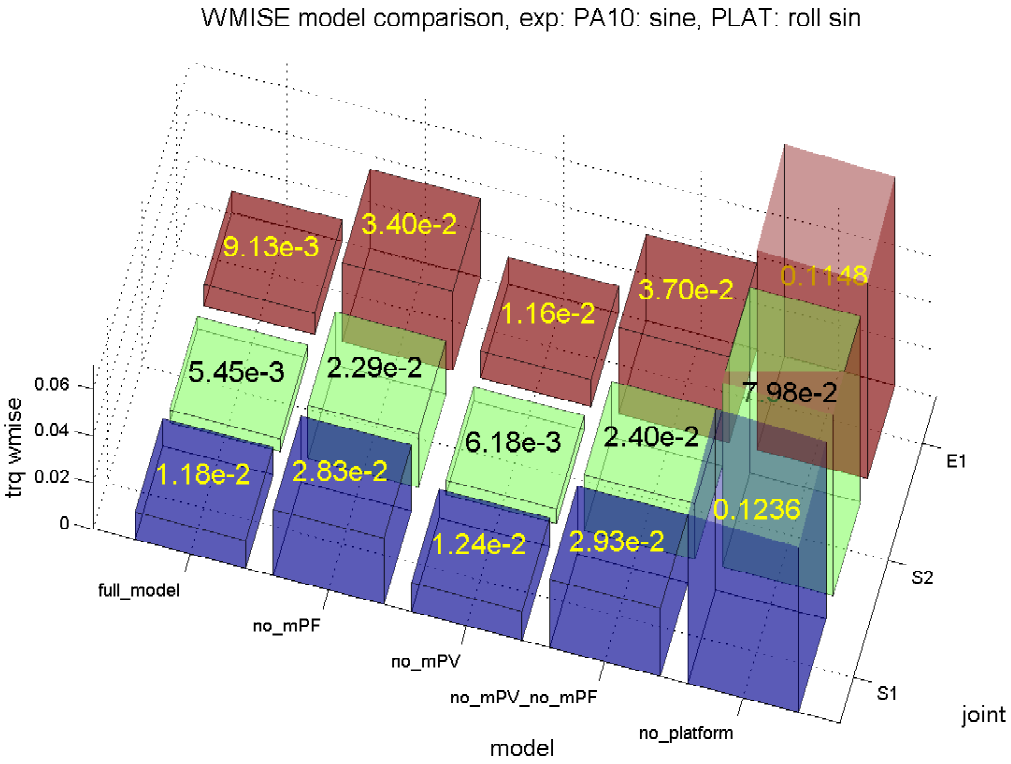


Figure 7.4: WMISE experiment and simulation torques comparison, experiment: PA10: **`sine`**, PLAT: roll sin.

model \ *joint*	S1	S2	E1	E2	W1	W2
full model	0.01724	0.00564	0.01662	0.06947	0.00912	0.00486
no mPF	0.06498	0.03117	0.06672	0.06930	0.00896	0.00486
no mPV	0.01974	0.00680	0.01812	0.06934	0.00931	0.00485
no mPV no mPF	0.06564	0.03272	0.06874	0.06913	0.00898	0.00486
no platform	0.35802	0.12698	0.25139	0.07000	0.00947	0.00485

Table 7.4: WMISE experimental and simulation torques comparison, experiment: PA10: **`ssine`**, PLAT: pitch sin.

reduces the WMISE difference.

- The **PV** element has very little effect on the model.

- Based on tables 7.1 to 7.4 it can be deduced that Joints E2, W1 and W2 can be considered to be unaffected by the platform model which can be attributed to low or zero gravity contribution to the joint model and significantly lower link inertias (and therefore less torque resulting from the platform terms).

7.4.2 Single frequency per channel 2-DOF motion

In this analysis single frequency 2-DOF trajectories are used for the platform (equations 6.2 and 6.2). Comparison of simulation and experimental results are presented in tables

7.5 to 7.8. An example comparison of motor torques for the PA10 trajectory **ssine** and platform trajectory roll+pitch with equal periods is presented in figures 7.5 and 7.6. The corresponding bar plot of WMISE differences is presented in figure 7.7.

model \ *joint*	S1	S2	E1	E2	W1	W2
full model	0.00504	0.00274	0.00566	0.04647	0.00891	0.00523
no mPF	0.00916	0.00294	0.00765	0.04636	0.00893	0.00523
no mPV	0.00660	0.00313	0.00700	0.04679	0.00884	0.00523
no mPV no mPF	0.01095	0.00339	0.00930	0.04666	0.00886	0.00523
no platform	0.23414	0.04218	0.06369	0.04785	0.00916	0.00523

Table 7.5: WMISE experimental and simulation torques comparison, experiment: PA10: **sine**, PLAT: roll + pitch sin equal periods.

model \ *joint*	S1	S2	E1	E2	W1	W2
full model	0.00461	0.00400	0.01321	0.04741	0.00757	0.00510
no mPF	0.00794	0.00565	0.01789	0.04720	0.00771	0.00511
no mPV	0.00472	0.00400	0.01526	0.04728	0.00759	0.00511
no mPV no mPF	0.00784	0.00595	0.02119	0.04707	0.00774	0.00511
no platform	0.15405	0.07155	0.24631	0.04656	0.00906	0.00511

Table 7.6: WMISE experimental and simulation torques comparison, experiment: PA10: **ssine**, PLAT: roll + pitch sin equal periods.

model \ *joint*	S1	S2	E1	E2	W1	W2
full model	0.00652	0.00326	0.00541	0.08066	0.01024	0.00524
no mPF	0.02078	0.00753	0.01496	0.08047	0.01024	0.00524
no mPV	0.00791	0.00467	0.00785	0.08144	0.01034	0.00524
no mPV no mPF	0.01990	0.00848	0.01745	0.08122	0.01035	0.00524
no platform	0.15141	0.04839	0.07551	0.08176	0.01055	0.00524

Table 7.7: WMISE experimental and simulation torques comparison, experiment: PA10: **sine**, PLAT: roll + pitch sin different periods.

model \ *joint*	S1	S2	E1	E2	W1	W2
full model	0.00894	0.00380	0.02090	0.07318	0.00866	0.00447
no mPF	0.03236	0.01718	0.06373	0.07255	0.00873	0.00447
no mPV	0.00872	0.00451	0.02431	0.07303	0.00871	0.00447
no mPV no mPF	0.03042	0.01792	0.07002	0.07248	0.00890	0.00447
no platform	0.17030	0.07369	0.35562	0.07029	0.00972	0.00447

Table 7.8: WMISE experimental and simulation torques comparison, experiment: PA10: **ssine**, PLAT: roll + pitch sin different periods.

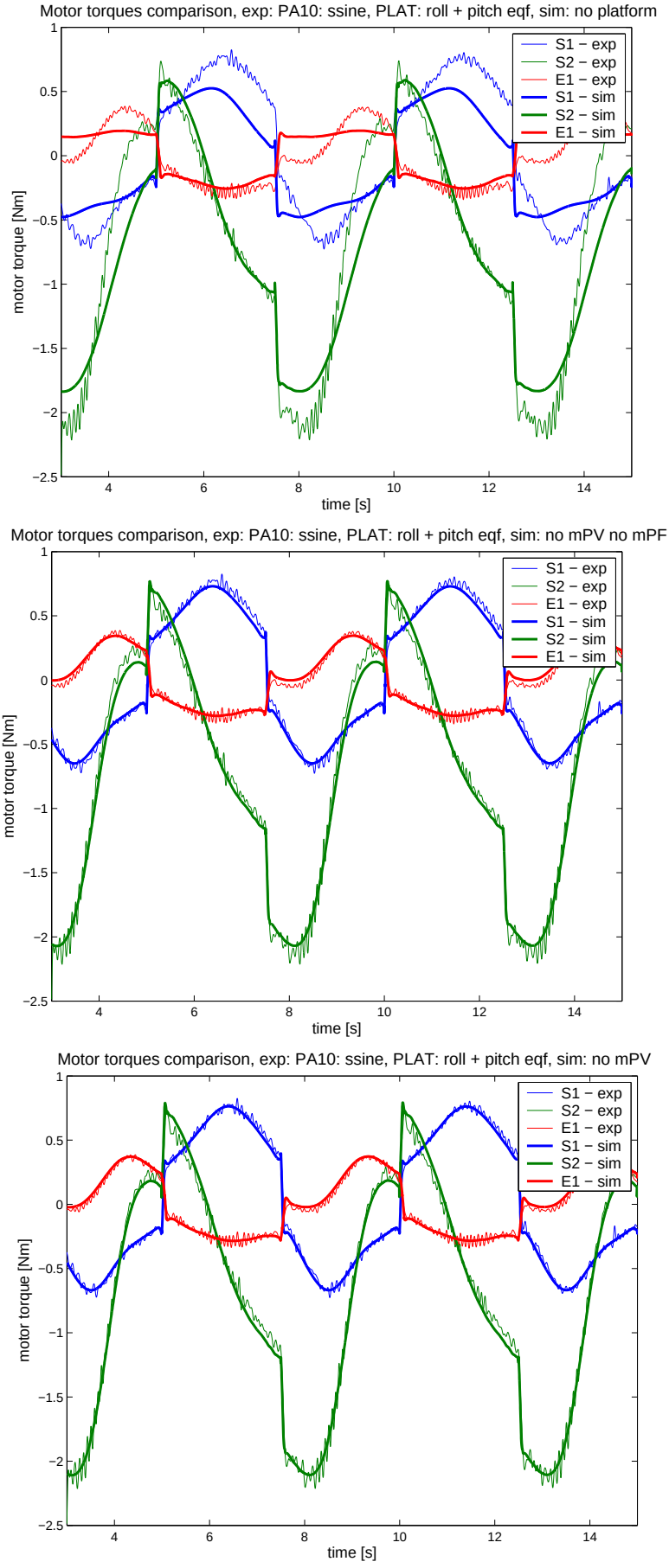


Figure 7.5: Comparison of simulation and experiment torques for different manipulator on platform model set-ups, PA10 trajectory **`ssine`**, platform trajectory roll+pitch sin with equal periods.

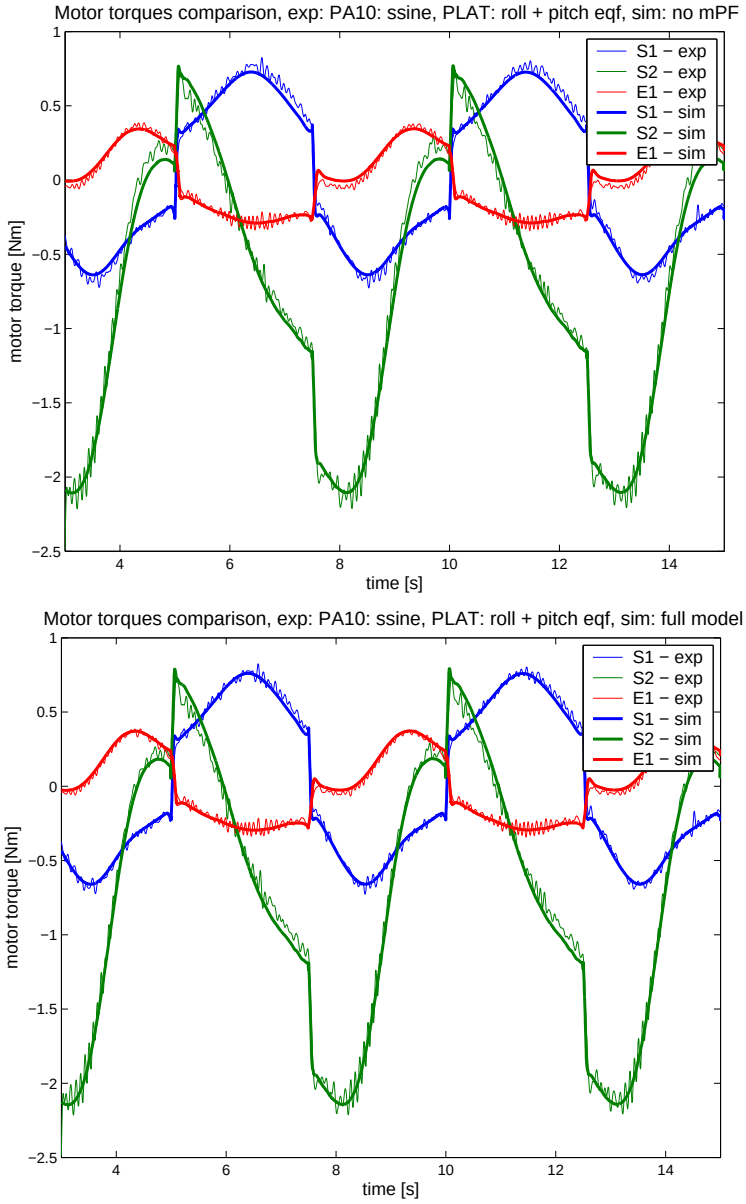


Figure 7.6: Comparison of simulation and experiment torques for different manipulator on platform model set-ups, PA10 trajectory **`ssine`**, platform trajectory roll+pitch sin with equal periods.

Discussion of results Based on the results the following can be observed:

- The general properties of the model are similar to the case of the 1-DOF platform motion in terms of torque plots in figures 7.5 and 7.6.
- For the slower trajectory with roll and pitch angles following a sine with the same period, the significance of the **PF** term is greatly reduced, and although it still improves the model, the difference is not as noticeable as in the case of faster trajectories (figure 7.7 and tables 7.5 to 7.8. This may be attributed to the fact that accelerations in a sine type trajectory increase with the square of the frequency (e.g. for a $2\times$ shorter trajectory period the accelerations are $4\times$ greater).

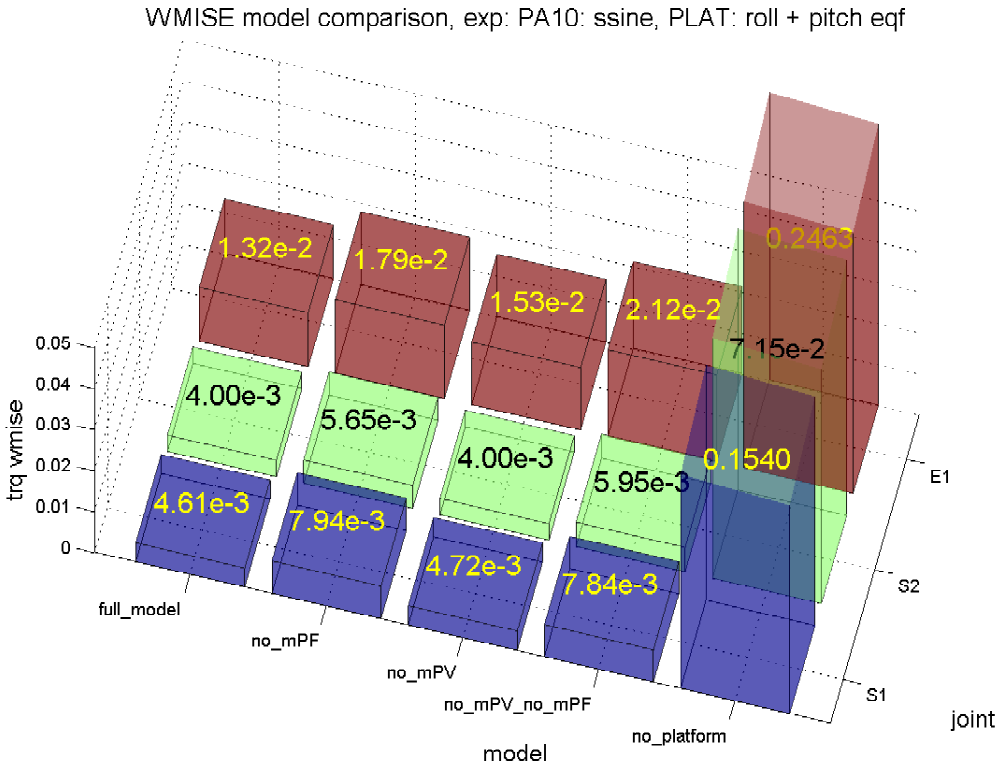


Figure 7.7: WMISE experiment and simulation torques comparison, experiment: PA10: **`ssine`**, PLAT: roll + pitch sin equal periods.

- It can be noticed in figures 7.5 and 7.6 that once the platform orientation dependant gravity term has been introduced to the model, there is not much noticeable difference when the other two terms are introduced.

7.4.3 Sum of multiple frequencies 2-DOF motion

In this analysis, trajectories comprised of sums of multiple frequency sines on 2-DOF are used for the platform (equation 6.4 and tables 6.4 to 6.7). Comparison of simulation and experimental results are presented in tables 7.9 to 7.12. An example comparison of motor torques for PA10 the trajectory **`sine`** and platform trajectory sin sum (2) is presented in figures 7.8 and 7.9. The corresponding bar plot of WMISE differences is presented in figure 7.10.

Discussion of results Based on the results the following can be observed:

- The torque plot results in figures 7.8 and 7.9 of the sum of multiple frequencies platform trajectories follow the same pattern of model elements incorporating to match the control torques as in the previous experiments.

- These trajectories are characterised by lower average absolute platform angles than the simple sin trajectories and higher accelerations (due to the richer frequency con-

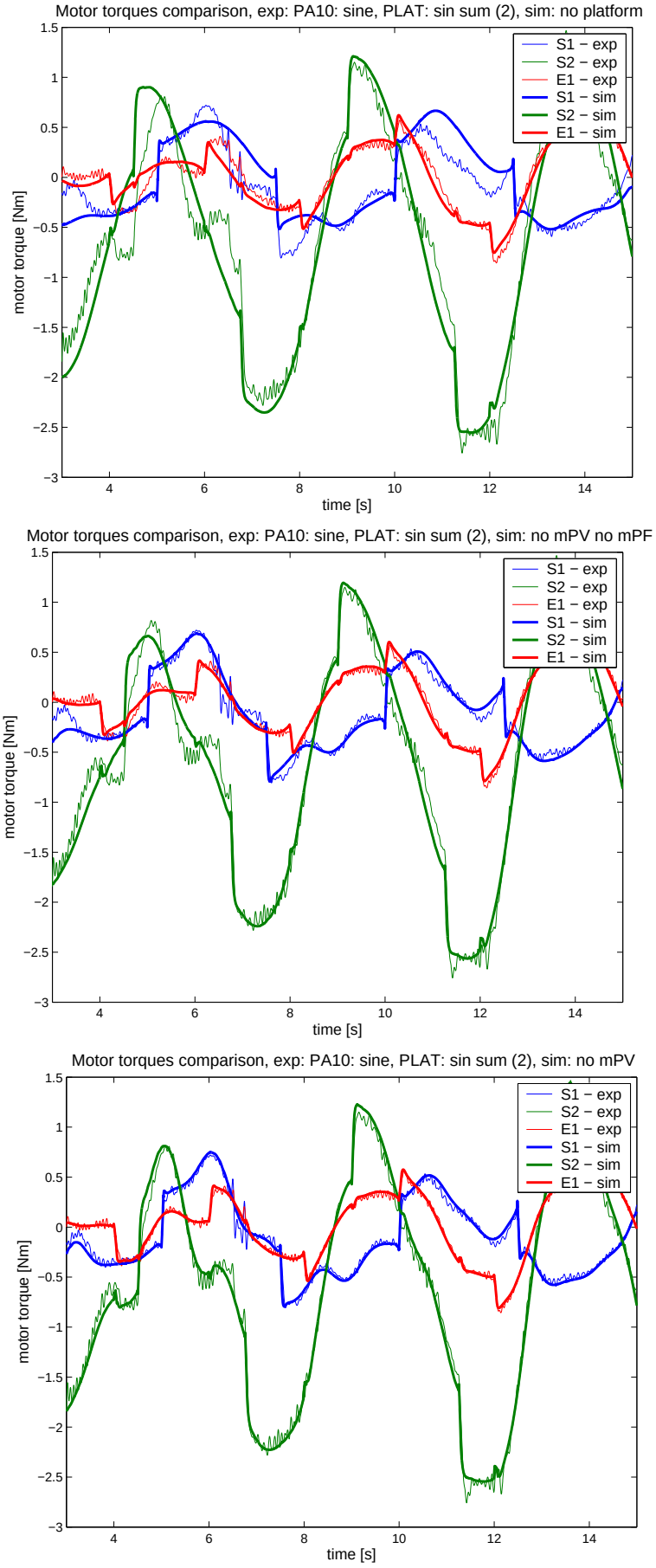


Figure 7.8: Comparison of simulation and experiment torques for different manipulator on platform model set-ups, PA10 trajectory **`sine`**, platform trajectory sin sum (2).

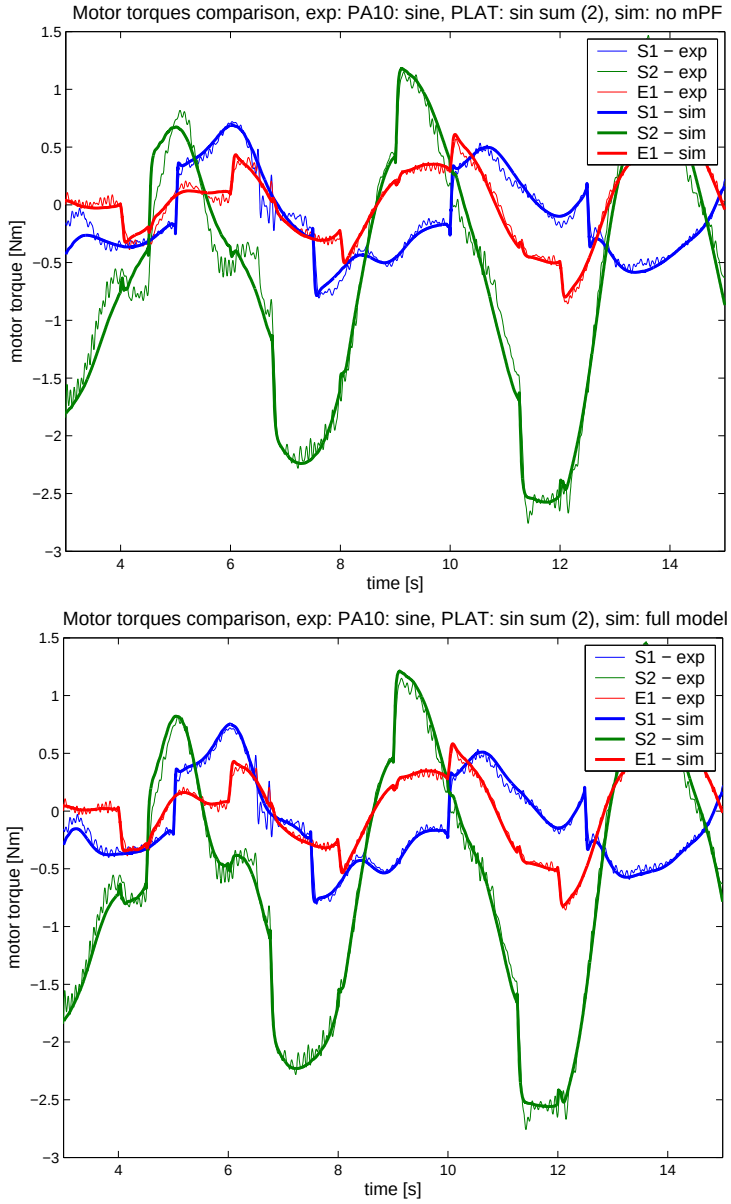


Figure 7.9: Comparison of simulation and experiment torques for different manipulator on platform model set-ups, PA10 trajectory **`sine`**, platform trajectory sin sum (2).

model \ *joint*	S1	S2	E1	E2	W1	W2
full model	`0.01829`	`0.01034`	`0.01789`	`0.06609`	`0.00984`	`0.00598`
no mPF	`0.04412`	`0.02594`	`0.04292`	`0.06504`	`0.00975`	`0.00598`
no mPV	`0.01936`	`0.01044`	`0.01876`	`0.06660`	`0.00988`	`0.00598`
no mPV no mPF	`0.04665`	`0.02585`	`0.04403`	`0.06556`	`0.00976`	`0.00598`
no platform	`0.09599`	`0.06281`	`0.09608`	`0.06535`	`0.00975`	`0.00598`

Table 7.9: WMISE experimental and simulation torques comparison, experiment: PA10: **`sine`**, PLAT: sin sum (1).

tent). This leads to a lower contribution of the gravity term and a higher contribution of the **PF** term to the model (tables 7.9 to 7.12 and figure 7.10).

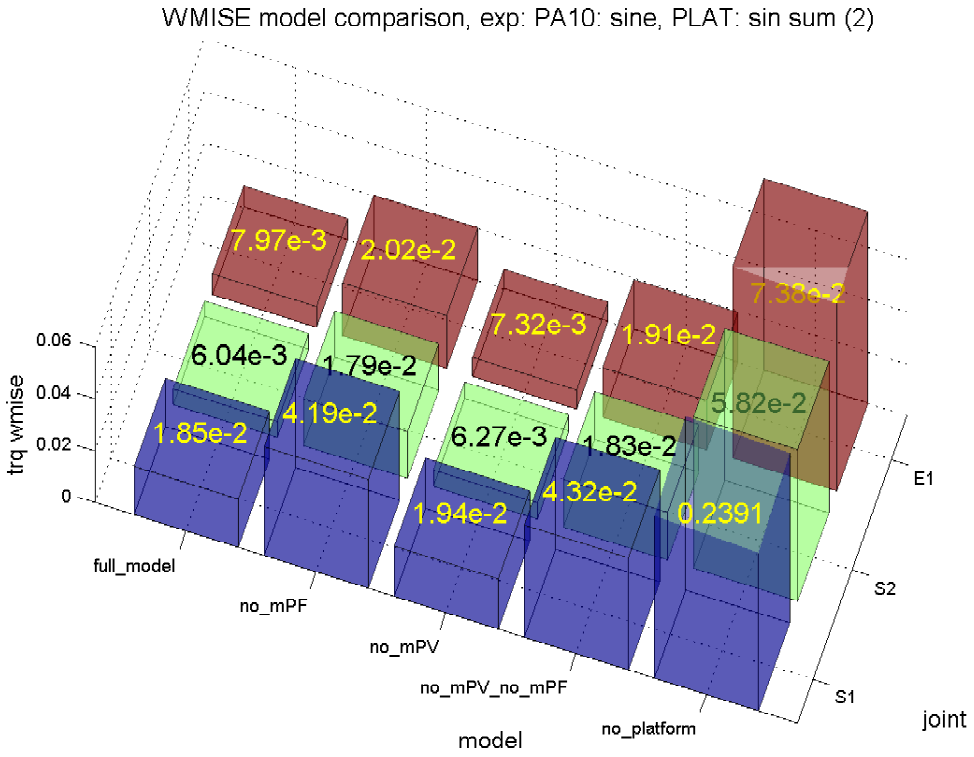


Figure 7.10: WMISE experiment and simulation torques comparison, experiment: PA10: **`sine`**, PLAT: sin sum (2).

model \ *joint*	S1	S2	E1	E2	W1	W2
full model	0.02553	0.01374	0.04574	0.06139	0.00806	0.00565
no mPF	0.05448	0.03755	0.12644	0.06210	0.00784	0.00565
no mPV	0.02595	0.01388	0.04923	0.06133	0.00806	0.00565
no mPV no mPF	0.05517	0.03759	0.12941	0.06173	0.00783	0.00565
no platform	0.10521	0.08838	0.33269	0.06214	0.00806	0.00565

Table 7.10: WMISE experimental and simulation torques comparison, experiment: PA10: **`ssine`**, PLAT: sin sum (1).

model \ *joint*	S1	S2	E1	E2	W1	W2
full model	0.01854	0.00604	0.00797	0.05646	0.01045	0.00552
no mPF	0.04190	0.01787	0.02021	0.05750	0.01058	0.00552
no mPV	0.01935	0.00627	0.00732	0.05638	0.01043	0.00552
no mPV no mPF	0.04321	0.01833	0.01912	0.05742	0.01058	0.00552
no platform	0.23907	0.05825	0.07381	0.06082	0.01088	0.00552

Table 7.11: WMISE experimental and simulation torques comparison, experiment: PA10: **`sine`**, PLAT: sin sum (2).

7.5 Summary

In this chapter a generic model of a robotic manipulator on a moving platform (not dynamically influenced by the manipulator) has been derived.

model \ *joint*	S1	S2	E1	E2	W1	W2
full model	0.02279	0.00866	0.03335	0.09333	0.00771	0.00517
no mPF	0.05086	0.02585	0.11062	0.09358	0.00766	0.00517
no mPV	0.02287	0.00868	0.03359	0.09353	0.00770	0.00517
no mPV no mPF	0.05083	0.02593	0.11064	0.09355	0.00772	0.00517
no platform	0.21988	0.08207	0.37961	0.09422	0.00792	0.00517

Table 7.12: WMISE experimental and simulation torques comparison, experiment: PA10: **`ssine`**, PLAT: sin sum (2).

The derived equations have been used to obtain a model of the PA10-6CE manipulator on a 2-DOF (roll and pitch) platform, with the configuration parameters presented in this chapter. However, the model derivation procedure is general and applies to any rigid manipulator on a 6-DOF moving base.

The obtained model has been compared with the experimental results from the actual PA10 manipulator mounted on a moving platform by means of the weighted squares of differences of control torques for the manipulator and the platform moving along a given trajectory.

It has been confirmed (graphically and numerically) that the obtained model is well suited for simulating the system.

The following properties of the model on platform (equation 7.32) have been revealed by simulation and experimental comparisons:

- The gravity term significantly improves the model for any tested trajectory.
- The **PF** term (independent of manipulator velocities) improves the model mostly for fast changing (containing high angular accelerations) trajectories.
- The **PV** term has the least contribution when compared to the experimental control torques in any case.

Based on the presented derivations and observations made, the problem of controlling a robotic manipulator on a moving platform can be approached with verified prior knowledge about the dynamic model.

Chapter 8

Control of the PA10-6CE robotic manipulator on a moving base

8.1 Introduction

In Chapter 7 a model of a robotic manipulator mounted on a moving base was presented. Based on the results of the experiments and simulations presented, it was concluded that certain parts of this model make a significant contribution to the manipulator dynamic model. Using these observations, a controller can be designed for a manipulator mounted on a moving base with the following assumptions:

- The frequencies present in the base motion can be considered to be of the same order (or less) than the manipulator trajectories.
- The manipulator is mounted on a base that is significantly heavier (with greater inertia), therefore it is not influenced by the manipulator dynamics.
- The base motion is constrained to angles of $\leq 20^{\circ}$.

Based on the results from Chapter 7 it can be anticipated that only two of the additional elements of the model will have a significant influence on the manipulator dynamics (from equation 7.32), namely:

- the gravity element $\mathbf{g}_p(\mathbf{q}, \mathbf{q}_p)$ (depending only on the manipulator angles and base orientation),
- the free item $\mathbf{PF}(\mathbf{q}_p, \dot{\mathbf{q}}_p, \ddot{\mathbf{q}}_p)$ (depending only on the manipulator angles and base trajectory (angles, velocities and accelerations)).

Considering that the latter term depends on angular acceleration of the base, it may be difficult to obtain these in a noise free form at each control cycle (without a phase delay of several samples). However, the gravity term, which has been determined to make a significant contribution to the PA10 dynamic model, can be considered to be available in several forms. In Chapter 6 it was presented that base orientation can be measured using

a clinometer sensor. Moreover, the actual gravity vector (acting on the centre of the platform mounting plate) can be measured with the accelerometer. These two measurement methods are candidates for improving the trajectory tracking precision of the model based controllers. The sensors, available in the lab set-up, may not always be installed (or measurements may not be available) in real-life systems (e.g. off-shore). Therefore, it would be desirable to have a control algorithm that performs similarly without these sensors.
This chapter presents results of experiments performed with the PA10-6CE manipulator mounted on a CueSim motion platform for a number of model based controllers with various base trajectories including a typical vessel on sea 6-DOF motion. The main contributions of this chapter are:

- a broad comparison of the model based manipulator controllers in the presence of base disturbances,
- a comparison of compensation methods for the base motion using sensors and adaptive control,
- the application of a model based adaptive controller to compensate for the varying gravity vector due to base motion.

8.2 Control algorithms

The control schemes described in section 5.1, can all be easily modified to utilise the sensor (clinometers or accelerometers) measurements in the gravity term. Consider any of the controllers:

- gravity and friction compensation (equation 5.10),
- feedback linearisation (equation 5.14),
- Wen-Bayard algorithm (equation 5.17),
- Slotine-Li algorithm (equation 5.21)

and replace the fixed-base gravity term with the new one from equation 7.31.
The symbol for the generic manipulator gravity term (independent of the gravity acceleration vector) is

$$\mathbf{G}_i(\mathbf{q}) = \sum_{j=i}^{n} m_j(\mathbf{U}_{ji}R_j) \tag{8.1}$$

The symbols in this equation have been explained in section 3.4.1. The following replacement gravity acceleration vector forms are then used

- using angular orientation sensors

$$\mathbf{g}_{p_i}(\mathbf{q},\mathbf{q}_p) = \mathbf{g}\cdot \mathrm{Rot}(Y,\hat{\beta})\cdot \mathrm{Rot}(X,\hat{\alpha})\cdot \mathbf{G}_i(\mathbf{q}) \tag{8.2}$$

where $\hat{\alpha}$ is the roll angle measurement from the sensor and $\hat{\beta}$ is the pitch angle measurement from the sensor;

- using accelerometer sensors

$$\mathbf{g}_{p_i}(\mathbf{q}, \mathbf{q}_p) = \hat{\mathbf{g}} \cdot \mathbf{G}_i(\mathbf{q}), \tag{8.3}$$

where $\hat{\mathbf{g}} = (\hat{g}_x, \hat{g}_y, \hat{g}_z)$ is the vector of measured accelerations.

It can be noticed that the accelerometer measurement could account for translation accelerations (automatically incorporating their effect into the model), whereas the angular measurements would not deliver such extra functionality.
Both of these methods for replacing the static gravity vector require additional hardware consisting of sensors with the respective signal conditioning, data acquisition circuitry and control system components. This may not be easily available in some applications, therefore it would be desirable to have a gravity estimate without the need for measurements.

8.2.1 Adaptive control

If the structure of the model is known, but the parameters are either impossible to identify [8.2] or their values change during the operation of the system, an adaptive model-based algorithm can be applied. Both **computed torque** and **dissipative** algorithms form a base for adaptive algorithms. The main steps of an adaptive algorithm are:

- use the present estimates (or assumed values) of the unknown parameters to produce a control signal,
- use the control error to update the parameter estimates.

Consider the manipulator dynamic model (3.7) and assume that some of its parameters are unknown, the new equation is

$$\mathbf{Q}(\mathbf{q}, \hat{\vartheta})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}, \hat{\vartheta})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}, \hat{\vartheta}) + \mathbf{F}(\dot{\mathbf{q}})^1 = \mathbf{u}, \tag{8.4}$$

where $\hat{\vartheta}$ is the vector of the unknown parameters that require to be adapted. However, the dynamics equation is linear in these parameters and can be expressed as

$$\mathbf{Q}(\mathbf{q}, \hat{\vartheta})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}, \hat{\vartheta})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}, \hat{\vartheta}) + \mathbf{F}(\dot{\mathbf{q}}) = Y(\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q})\hat{\vartheta} + \mathbf{F}(\dot{\mathbf{q}}) = \mathbf{u}. \tag{8.5}$$

In a generic controller formulation, the friction model is not present, since it is not continuous and it does not satisfy properties allowing the stability/convergence of the controllers to be proved. However, due to the significant friction torque contribution in the PA10 manipulator, the friction compensation is used in the adaptive controllers.

[1] there is no adaptation of the friction parameters assumed, since the model cannot be expressed as linear in the parameters

8.2.1.1 Adaptive feedback linearisation

The adaptive routine can be integrated into the feedback linearisation algorithm [8.4]. The 'static' element of the controller is

$$\begin{cases} \mathbf{u} = \mathbf{Q}(\mathbf{q}, \hat{\vartheta})\mathbf{v} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}, \hat{\vartheta})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}, \hat{\vartheta}) + \hat{\mathbf{F}}(\dot{\mathbf{q}}) \\ \mathbf{v} = \ddot{\mathbf{q}}_d - K_d\dot{\mathbf{e}} - K_p\mathbf{e} \end{cases} \tag{8.6}$$

and the parameters are updated according to

$$\dot{\hat{\vartheta}} = -\Gamma Y^T(\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q}) \left(\left((\mathbf{Q}(\mathbf{q}, \hat{\vartheta}) \right)^T \right)^{-1} B^T P z, \tag{8.7}$$

where $\Gamma = \Gamma^T$ is the gain matrix for the estimation algorithm and $P = P^T > 0$ is the solution of a Lyapunov equation

$$A^T P + PA = -M,$$

where

$$A = \begin{bmatrix} 0 & \boldsymbol{I}_n \\ -K_p & -K_d \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \boldsymbol{I}_n \end{bmatrix}, \quad z = (\mathbf{e}^T, \dot{\mathbf{e}}^T)^T$$

and $M = M^T$ is an arbitrary matrix responsible for the speed of convergence of the variable z to zero.
The adaptive scheme can also be applied to the Wen-Bayard and Slotine-Li algorithms.

8.2.1.2 Adaptive Wen-Bayard controller

Since the Wen-Bayard controller does not perform a full feedback linearisation, the adaptation scheme is simpler in this case [8.1].

$$\mathbf{u} = \mathbf{Q}(\mathbf{q}_d, \hat{\vartheta})\ddot{\mathbf{q}}_d + \mathbf{C}(\mathbf{q}_d, \dot{\mathbf{q}}_d, \hat{\vartheta})\dot{\mathbf{q}}_d + \mathbf{g}(\mathbf{q}_d, \hat{\vartheta}) + \hat{\mathbf{F}}(\dot{\mathbf{q}}) - K_d\dot{\mathbf{e}} - K_p\mathbf{e} \tag{8.8}$$

and the parameters are updated according to

$$\dot{\hat{\vartheta}} = -\Gamma Y^T(\ddot{\mathbf{q}}_d, \dot{\mathbf{q}}_d, \dot{\mathbf{q}}_d, \mathbf{q}_d)(\dot{\mathbf{e}} + \varepsilon\mathbf{e}), \tag{8.9}$$

where $K_d = K_d^T > 0$ and $K_p = K_p^T > 0$ are the PD controller constants, $\Gamma = \Gamma^T > 0$ is the parameter adaptation gain (matrix) and ε is the adaptation bandwidth parameter (similar to $\boldsymbol{\Lambda}$ in the sliding controller – here it has been chosen as $\varepsilon = \frac{K_p}{K_d}$).

8.2.1.3 Adaptive Slotine-Li controller

The adaptive Slotine-Li [8.3] algorithm utilises the $\mathbf{q}_r$ variable defined in equation 5.19 and the sliding surface defined in equation 5.20.

$$\mathbf{u} = \mathbf{Q}(\mathbf{q}, \hat{\vartheta})\ddot{\mathbf{q}}_r + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}, \hat{\vartheta})\dot{\mathbf{q}}_r + \mathbf{g}(\mathbf{q}, \hat{\vartheta}) + \hat{\mathbf{F}}(\dot{\mathbf{q}}) - K_d\mathbf{s}, \tag{8.10}$$

where

$$\mathbf{s} = \dot{\mathbf{e}} + \Lambda\mathbf{e}$$

and the parameters are updated according to

$$\dot{\hat{\vartheta}} = -\Gamma Y^T(\ddot{\mathbf{q}}_r, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \mathbf{q})\mathbf{s}, \tag{8.11}$$

where $\Gamma = \Gamma^T > 0$ is the parameter adaptation gain (matrix). The controller can be presented in a form equivalent to a PD controller by letting $K_d\Lambda = K_p$.

8.2.2 Adaptive gravity

The adaptive controllers listed above have been designed with the assumption that the unknown parameters are constant over sufficiently long periods of time so that the adaptation scheme can tune to a value which guarantees minimisation of the tracking error. With this in mind it could be argued that the gravity vector in the case of a manipulator mounted on a moving base does not fulfil such conditions. However, considering the PA10 manipulator size and the control command sample rate, it can be observed that the base orientation does not change significantly during a series of control samples. Therefore, if the adaptation were fast enough to adjust the unknown parameters, it could be worth to evaluate such a scheme.

Considering the adaptive schemes listed above, it can be observed that the adaptive feedback linearisation method requires the solution of the Lyapunov equation and the inverse of the inertia matrix for each control cycle, which is computationally demanding and may cause numerical instabilities. Additionally, the feedback linearisation controller is considered to be highly sensitive to model inaccuracies and by assumption a part of the dynamics originating from the platform motion (the **PF** and **PV** terms) is intentionally omitted in the controller. Therefore, it has been decided to evaluate the two remaining adaptive schemes presented above – the Wen-Bayard (equation 8.8) and Slotine-Li (equation 8.10). The *regressor* matrix $Y(\ldots)$ used in the parameter update scheme (depending on various angle / trajectory parameters for different control schemes) simplifies to the generic gravity element

$$Y(\ldots) = \mathbf{G}(\mathbf{q}), \tag{8.12}$$

where $\mathbf{G}(\mathbf{q}) = (\mathbf{G}_1(\mathbf{q}), \ldots, \mathbf{G}_3(\mathbf{q}))$ is the full 6×3 matrix of gravity terms from equation 8.1. It can be noticed that the generic gravity term used is exactly the same as for a

manipulator with a static base (which simplifies the controller design effort).

8.2.3 Extension to simpler controllers

The manufacturer of the PA10 manipulator provides information on centres of mass, but not the link inertias, and an adaptive controller based on a simple model can be attempted. Assuming that the friction identification is an automated task, the gravity and friction could be used in an adaptive controller. This is a natural extension to the controller presented in equation 5.10. Considering that such a controller can use different error based feedback loop elements, the following controllers are evaluated in this analysis

- gravity + friction with a PD

$$\mathbf{u} = \hat{\mathbf{g}}\mathbf{G}(\mathbf{q}) + \mathbf{F}(\dot{\mathbf{q}}) - \mathbf{K}_P\mathbf{e} - \mathbf{K}_D\dot{\mathbf{e}} \tag{8.13}$$

  with adaptation

$$\dot{\hat{\mathbf{g}}}^T = -\Gamma\mathbf{G}^T(\mathbf{q})(\mathbf{K}_A\mathbf{e} + \dot{\mathbf{e}}).$$

  with $\mathbf{K}_A = \text{diag}(\frac{K_{p_1}}{K_{d_1}}, \ldots, \frac{K_{p_6}}{K_{d_6}})$

- gravity + friction with Slotine-Li sliding scheme

$$\mathbf{u} = \hat{\mathbf{g}}\mathbf{G}(\mathbf{q}) + \mathbf{F}(\dot{\mathbf{q}}) - \mathbf{K}_D\mathbf{s}_{SL} \tag{8.14}$$

  where $\mathbf{s}_{SL}$ is defined as the Slotine-Li sliding element (equation 5.20), with adaptation

$$\dot{\hat{\mathbf{g}}}^T = -\Gamma\mathbf{G}^T(\mathbf{q})(\mathbf{s}_{SL})$$

- gravity + friction with saturated sliding term

$$\mathbf{u} = \hat{\mathbf{g}}\mathbf{G}(\mathbf{q}) + \mathbf{F}(\dot{\mathbf{q}}) - \mathbf{K}\,\text{sat}\left(\frac{\mathbf{s}_{\text{sat}}}{\varphi}\right) \tag{8.15}$$

  where $\mathbf{s}_{\text{sat}}$ is defined as the sliding element (equation 5.8), with adaptation

$$\dot{\hat{\mathbf{g}}}^T = -\Gamma\mathbf{G}^T(\mathbf{q})\,\text{sat}\left(\frac{\mathbf{s}_{\text{sat}}}{\varphi}\right).$$

It can be noticed that the amount of modelling required to derive these controllers is significantly reduced (as is their computational load) in comparison to the controllers employing inertia and Coriolis and Centrifugal model terms.

8.2.4 Algorithm tuning

It has been suggested in [8.1] not to perform adaptation when the control is saturated, therefore an additional condition has been added to the algorithm to prevent the adapted

parameter update when the motor torques saturate at the manufacturer's rated levels. Additionally, for the algorithm using the saturated sliding, the adaptation is not performed when the saturation occurs.

8.3 Experiment set-up

To evaluate the performance of different controllers presented here, experiments using the PA10-6CE manipulator mounted on a moving platform have been carried out. The complete list of controllers to be evaluated in experiments, together with their abbreviated names used in the tables and graphs is presented in table 8.1.
The PA10-6CE manipulator is commanded along two sinusoidal trajectories (see section 5.2)

- synchronised (**`ssine`**),
- asynchronous (**`sine`**).

The platform is commanded along the following trajectories (with the abbreviated names used in tables and figures – see section 6.2):

- no platform motion (for reference):
  - zero[2];
- 1-DOF:
  - roll sine (sin_r2p0),
  - pitch sine (sin_r0p2);
- 2-DOF:
  - roll + pitch sine with different frequencies (sin_r1p2_pi2),
  - sum of sin (1) (sin2_s20),
  - sum of sin (2) (sin2_s20_2);
- 6-DOF:
  - vessel on sea motion (sea_all_2).

[2] this is not equivalent to platform being off due to additional noise induced in the system by the platform control / communication signals

Abbreviation	Type	Error	Model	Gravity
PD sat_sl slid PID	robust	PD saturated sliding Slotine-Li sliding PID	none	not present
grv_frc_PD grv_frc_PD_rp grv_frc_PD_acc grv_frc_PD_adap	partial	PD	gravity, friction	constant roll + pitch accelerometer adaptive
grv_frc_sat_sl grv_frc_sat_sl_s_rp grv_frc_sat_sl_s_acc grv_frc_sat_sl_adap	partial	saturated sliding	gravity, friction	constant roll + pitch accelerometer adaptive
grv_frc_slid grv_frc_slid_s_rp grv_frc_slid_s_acc grv_frc_slid_adap	partial	Slotine-Li sliding	gravity, friction	constant roll + pitch accelerometer adaptive
lin_FULL lin_FULL_s_rp lin_FULL_s_acc	linearisation	linearised PD	full	constant roll + pitch accelerometer
lin_noCC lin_noCC_s_rp lin_noCC_s_acc	linearisation	linearised PD	no Coriolis	constant roll + pitch accelerometer
W-B_FULL W-B_FULL_s_rp W-B_FULL_s_acc W-B_FULL_adap	Wen-Bayard	PD	full	constant roll + pitch accelerometer adaptive
W-B_noCC W-B_noCC_s_rp W-B_noCC_s_acc W-B_noCC_adap	Wen-Bayard	PD	no Coriolis	constant roll + pitch accelerometer adaptive
S-L_FULL S-L_FULL_s_rp S-L_FULL_s_acc S-L_FULL_adap	Slotine-Li	Slotine-Li sliding	full	constant roll + pitch accelerometer adaptive
S-L_noCC S-L_noCC_s_rp S-L_noCC_s_acc S-L_noCC_adap	Slotine-Li	Slotine-Li sliding	no Coriolis	constant roll + pitch accelerometer adaptive

Table 8.1: List of evaluated controllers with abbreviations used in tables and graphs.

8.3.1 Control parameters

For all non-adaptive parts of the controllers, the parameters presented in Chapter 5 have been used. Unless otherwise stated the gravity vector has been initialised to $(0, 0, -\mathsf{g})^T$ for all adaptive controllers. The adaptation gain has been set to $(6400.0, 6400.0, 6400.0)$. These values were chosen by trial and error in an attempt to maximise the adaptation

speed without causing controller instability.

8.4 Experimental results

To compare the properties of the different controllers for various platform and manipulator trajectories, multiple experiments have been performed and the results are presented through a series of graphs and tables.

8.4.1 Angle tracking

The weighted mean squared angle tracking error (equation 5.22 - abbreviated as angle WMISE) is used as the primary performance measure for all controllers. Based on all the results (available in appendix E), it has been concluded that only joints S1, S2 and E1 need to be analysed in closer detail, as there is no significant amount of variation in the results for the remaining three joints.
For reference, the results of the stationary platform are presented in figure 8.1[3] The following properties of the controllers can be observed:

- The adaptive controllers (Wen-Bayard and gravity+friction with either PD or saturated sliding error feedback) perform best, even though the gravity vector is constant.
- The full linearisation controller comes close for the sine trajectory (exhibiting more joint inertia and Coriolis and centrifugal interactions). It is interesting to notice that the version based on the roll/pitch sensor is marginally better than the static gravity controller, and the one based on a noisy accelerometer reading is the worst from this set. Marginally worse are the linearisation controllers with the Coriolis and centrifugal term disabled.
- The non-saturated sliding based controller is the poorest scheme.
- The Slotine-Li controllers, on average, perform worse than the Wen-Bayard. This is due to more measurement noise (especially from the velocity signal) being passed through.
- The angle WMISE is twice as small for the adaptive version of the Wen-Bayard controller compared to the constant (actual) gravity based Wen-Bayard controller. The ratio for the Slotine-Li controllers is 1.5. This needs to be considered when comparing the controllers with a moving platform.

As an example of 1-DOF base motion, the pitch sine results are presented in figure 8.2. The following observations can be made based on these results.

[3]The robust controller results and the PA10 velocity controller results have been excluded from these plots due to significantly higher errors (see appendix E for all results).

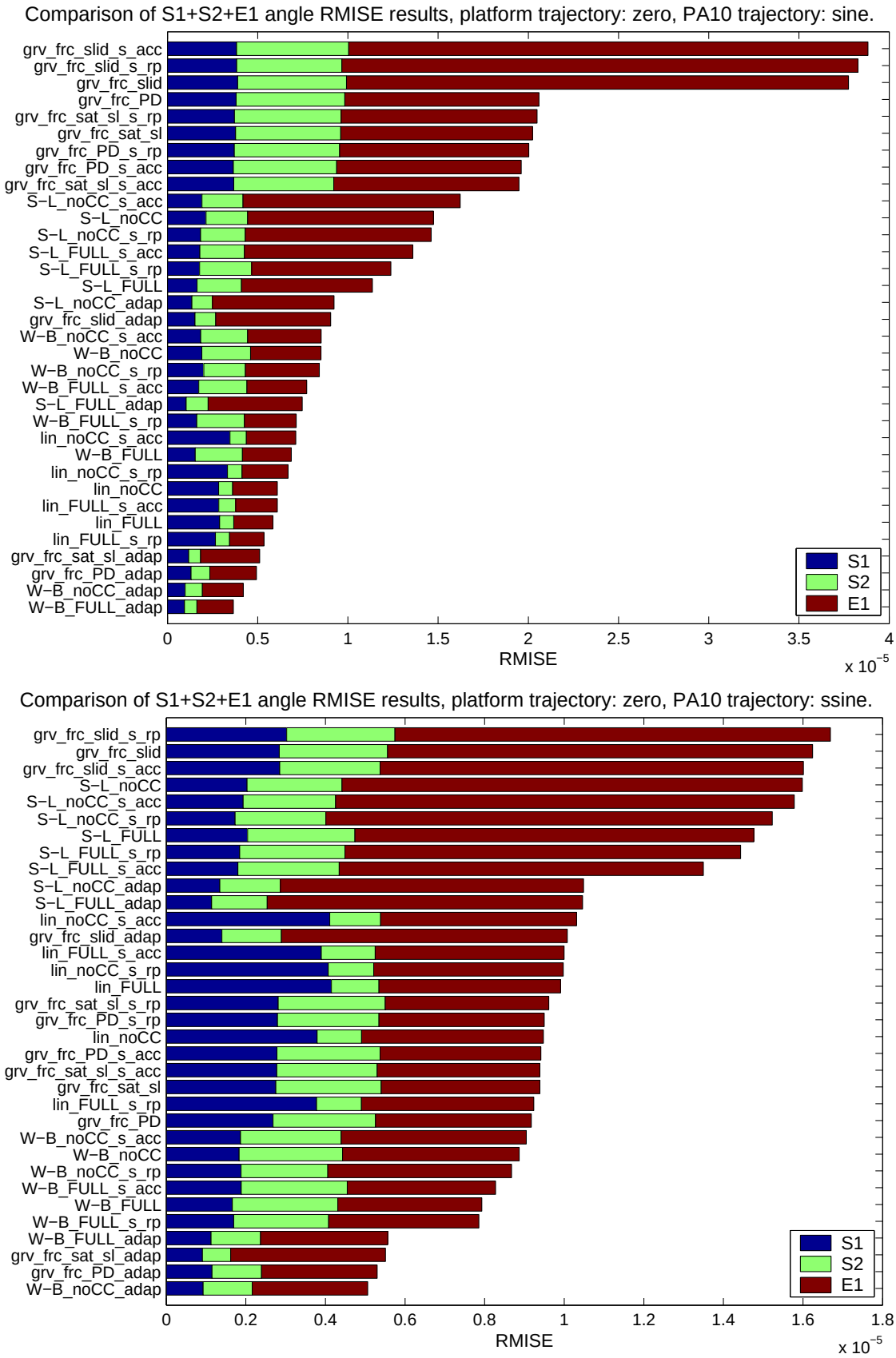


Figure 8.1: Angle WMISE ordered by sum for S1, S2, E1, platform trajectory: zero, for both PA10 trajectories.

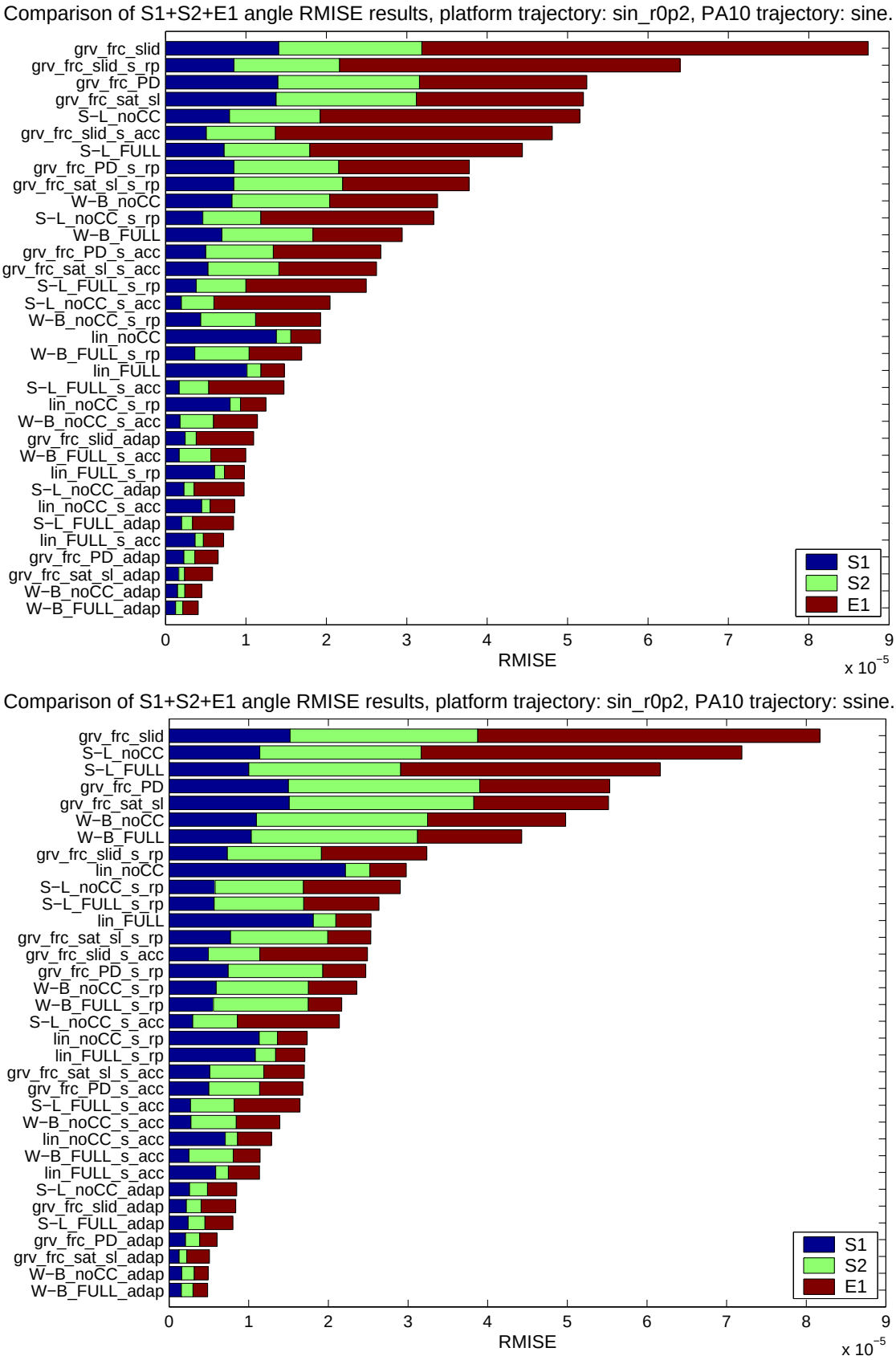


Figure 8.2: Angle WMISE ordered by sum for S1, S2, E1, platform trajectory: pitch sine, for both PA10 trajectories.

- For the **ssine** trajectory the adaptive controllers outperform the accelerometer based ones. It is even more interesting to notice that the order among both groups is very similar with the feedback linearisation outperforming the Wen-Bayard controller in the accelerometer 'group'.

- The adaptive gravity+friction with either PD or saturated sliding error feedback perform similarly to the adaptive Wen-Bayard controllers. This could indicate that both performance improvements came from adaptation being more significant than the lack of a complete model in the controller.

- The angle WMISE with the adaptive Slotine-Li controller for joints S2 and E1 (mostly affected by gravity) is lower than with the accelerometer measurement, which can be attributed to additional noise being transmitted from the accelerometer sensor gravity measurement (which is also induced by the manipulator joint torque chattering).

Results for the platform following a sum of multiple sine trajectories in 2-DOF are presented in 8.3. The angle tracking results are very similar to the 1-DOF platform trajectory results, which would indicate that the relative performance of the controllers is not affected by the complexity of the base trajectory. It should still be noticed that the 1-DOF sine trajectory utilises the entire pitch range of the platform with high speed, and the 2-DOF trajectory has similar maximum angular velocities, however not lasting for such long periods of time (therefore the absolute error levels are higher for the 1-DOF trajectory).
To evaluate the real-life performance of the analysed controllers, the motion platform is commanded to follow a representative 6-DOF trajectory of a vessel on the sea. It needs to be noted that the X, Y, Z and yaw motion of the base has not been considered in the model simulations. The experimental results are presented in figure 8.4, the following can be remarked upon:

- The relative performance of the controllers is very similar to the case of no base motion (which can be attributed to lower angular velocities and accelerations than in the artificially generated paths).

- The adaptive controllers still perform best of all (even in conditions not considered in the design step).

As can be noticed in figure 8.5, the PD controller angle WMISE are two orders of magnitude higher than the adaptive controller using only gravity and friction models with PD error feedback. It is interesting to notice that with the introduction of gravity and friction models (based on constant gravity vector) the error reduces ten-fold, it further reduces by approximately 30% when deriving the gravity values from the roll+pitch clinometers. Accelerometer sensors reduce the figure by another 30% and the adaptive gravity algorithm gives a further four-fold tracking error reduction.

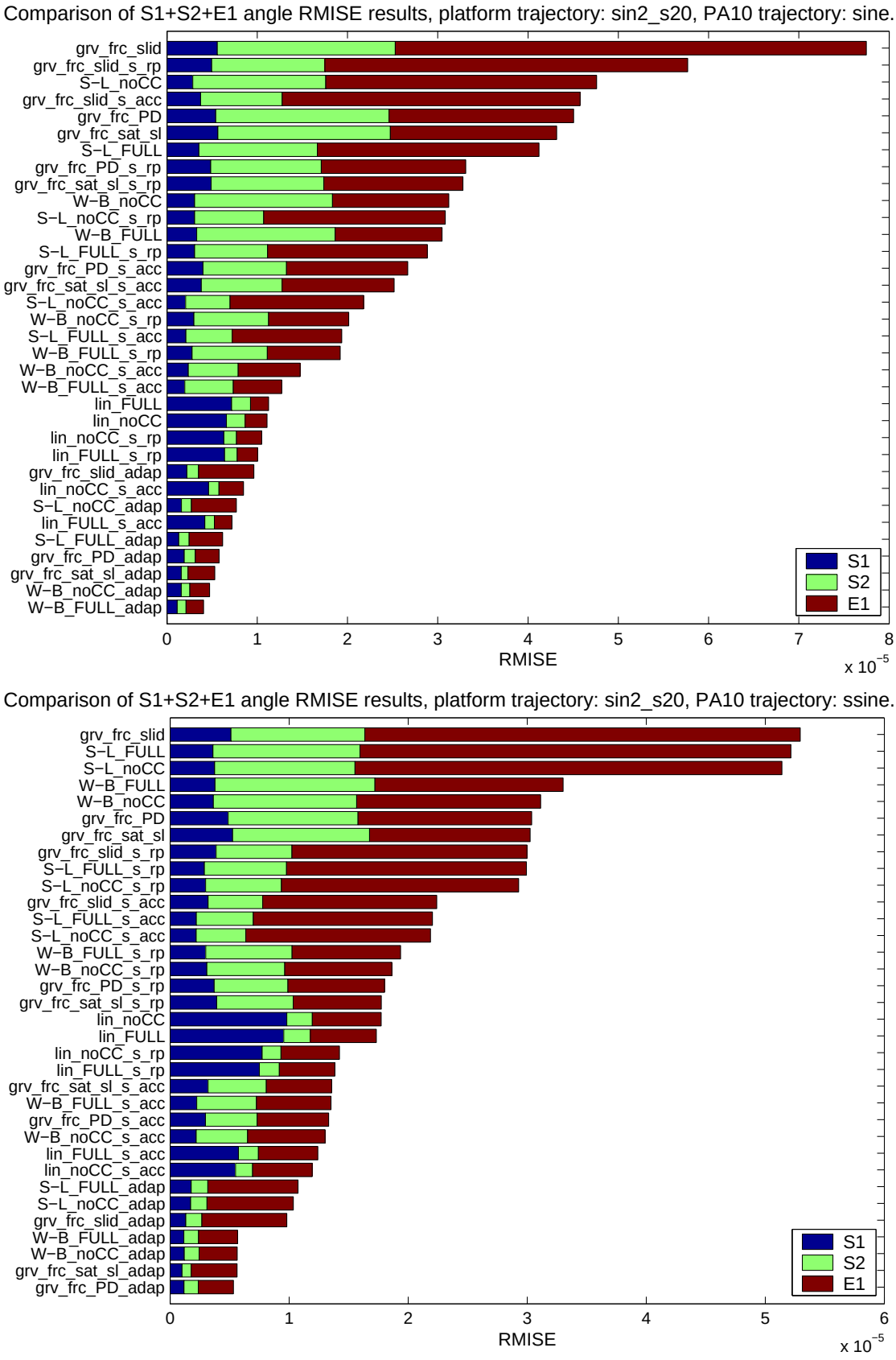


Figure 8.3: Angle WMISE ordered by sum for S1, S2, E1, platform trajectory: sum of sin (1), for both PA10 trajectories.

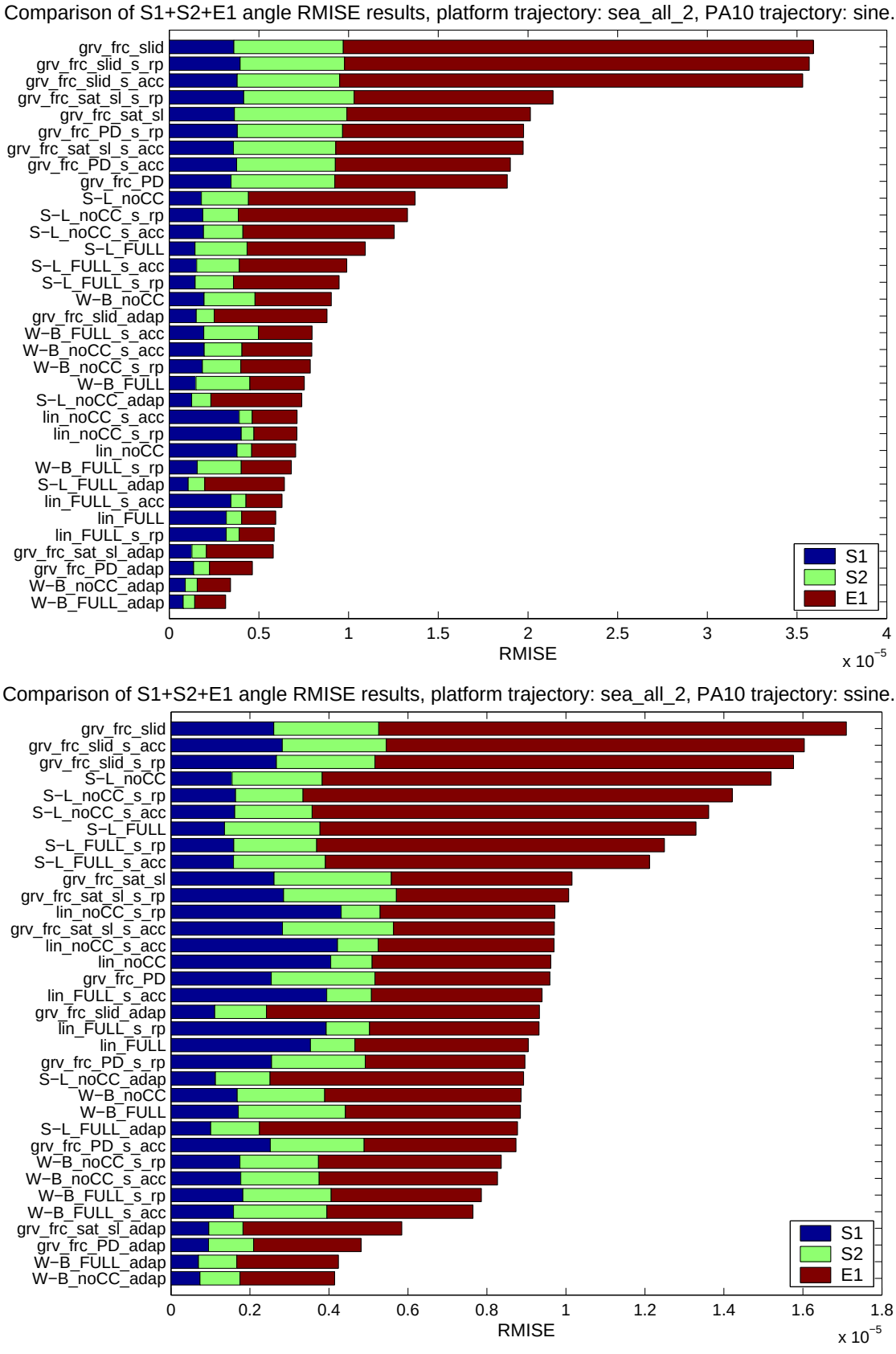


Figure 8.4: Angle WMISE ordered by sum for S1, S2, E1, platform trajectory: vessel on sea (6-DOF), for both PA10 trajectories.

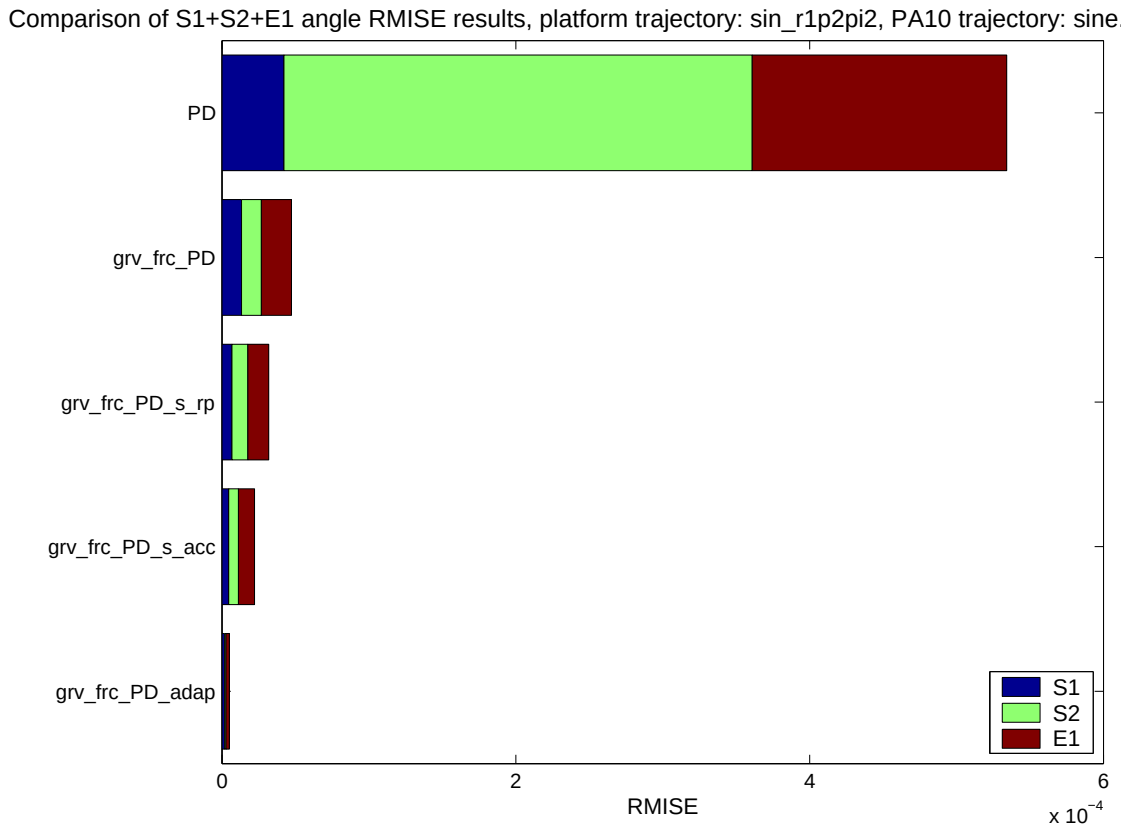


Figure 8.5: Angle WMISE ordered by sum for S1, S2, E1, platform trajectory: roll+pitch sine with different periods, PA10 trajectory **`ssine`**.

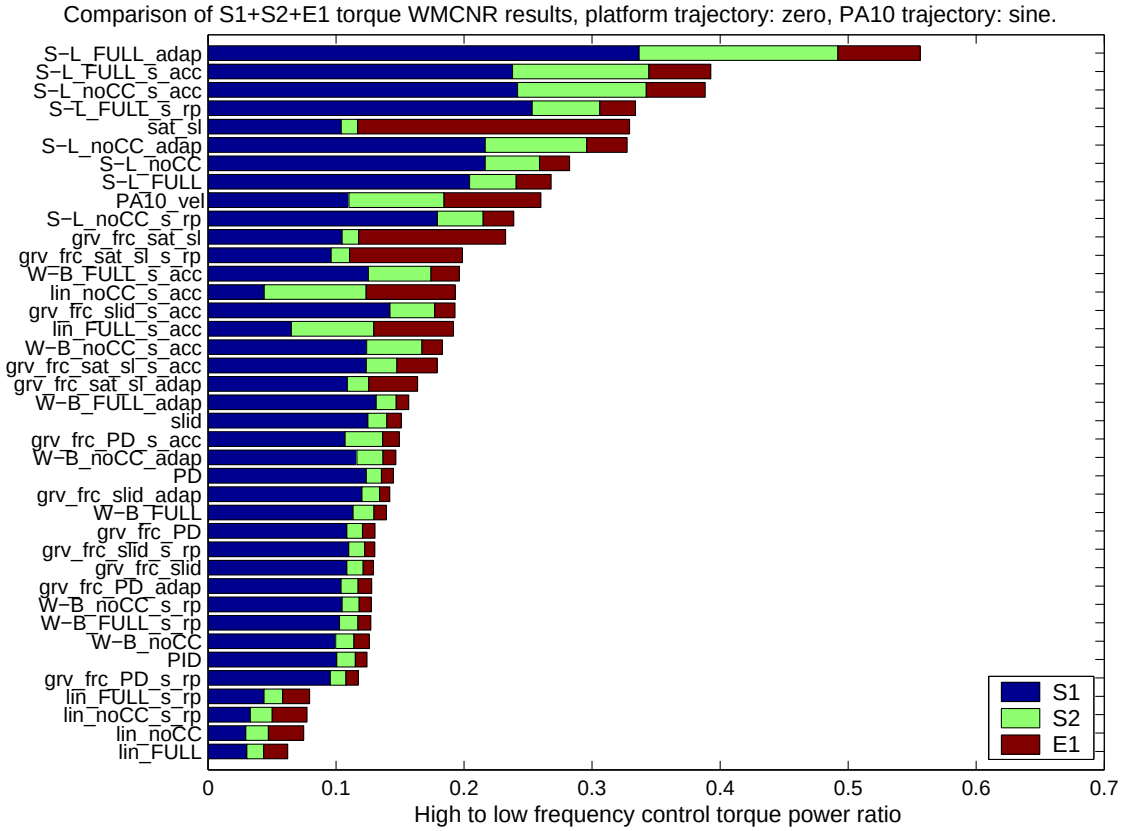
8.4.2 Weighted mean control noise ratio

The high to low frequency control torque energy ratio results (equation 5.23) for the cases presented above are shown in figures 8.6 to 8.9. The complete tabulated results are in appendix F. The following observations can be made regarding the tested controllers based on the obtained results

- The lowest noise energy occurs for the feedback linearisation algorithm.
- Wen-Bayard controllers generate less torque noise energy than the Slotine-Li.
- Accelerometer sensor measurements generate additional noise energy in comparison to adaptation or roll+pitch sensors.
- More dynamic base trajectories result in less relative torque noise.
- Adaptive gravity based controllers are generally scattered in the middle of the relative torque noise ranking.

8.4.3 Adaptive gravity estimation results

Gravity estimates initialisation To analyse the *static* performance of the adaptive algorithms the initial value of the gravity estimates has been set to $\hat{\mathbf{g}}_0 = (5.66, -5.66, 5.66)^T$



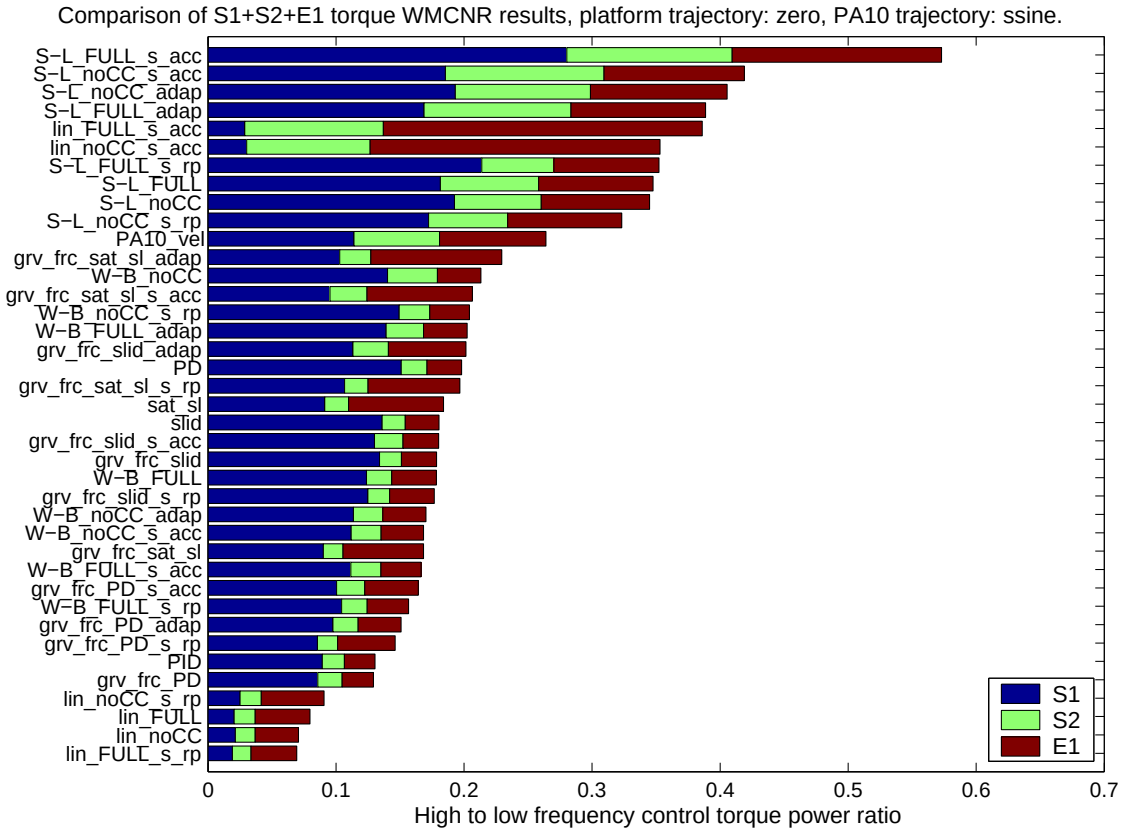
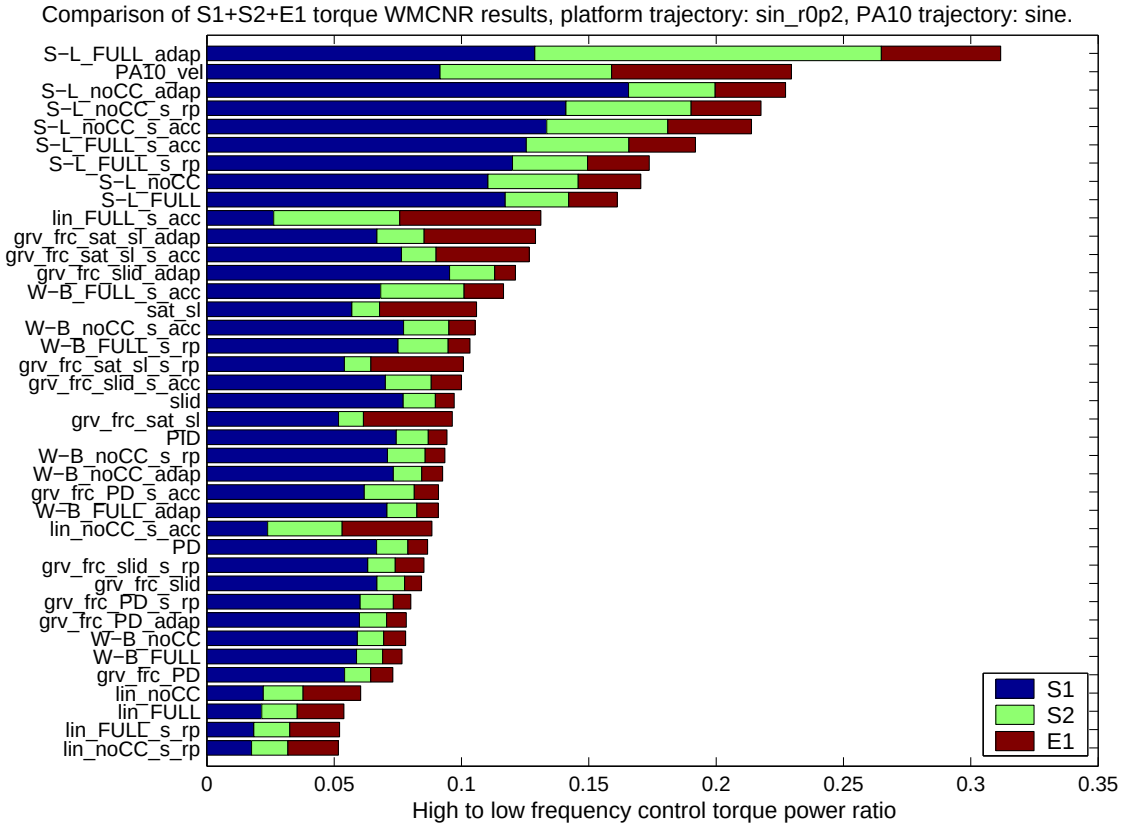


Figure 8.6: High to low frequency control torque power ratio ordered by sum for S1, S2, E1, platform trajectory: zero, for both PA10 trajectories.



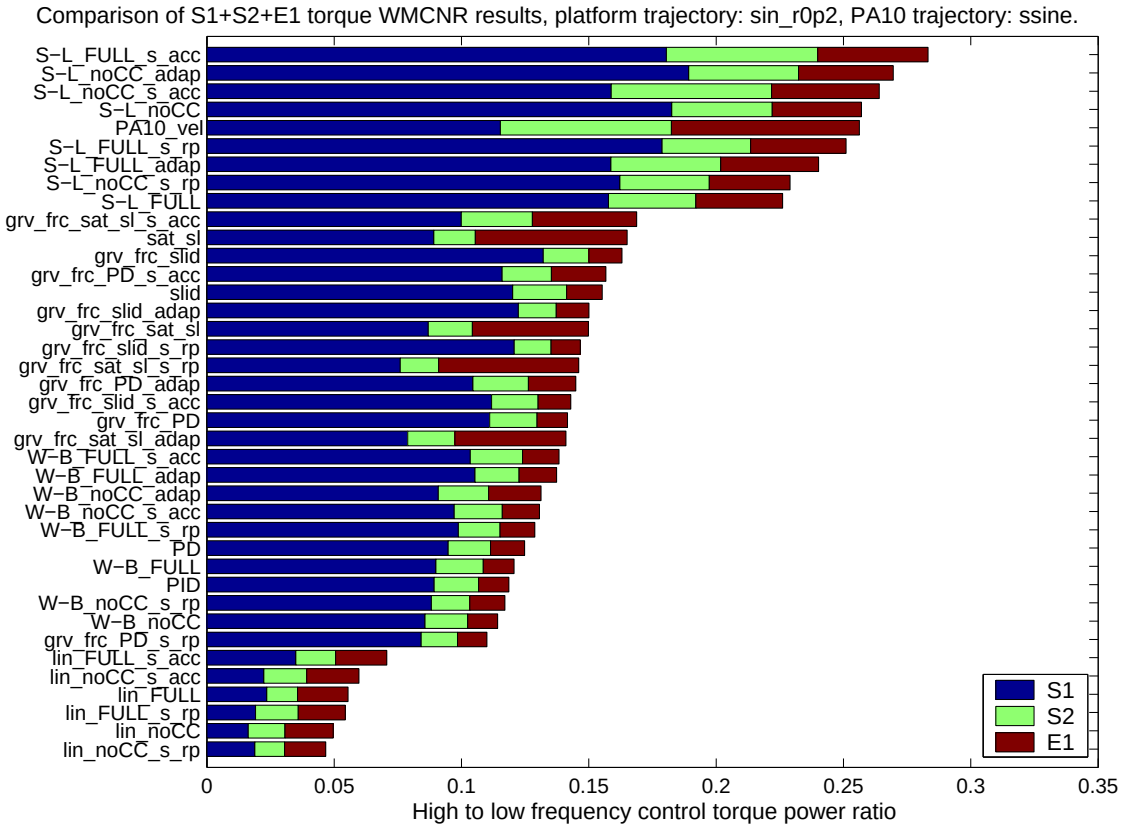
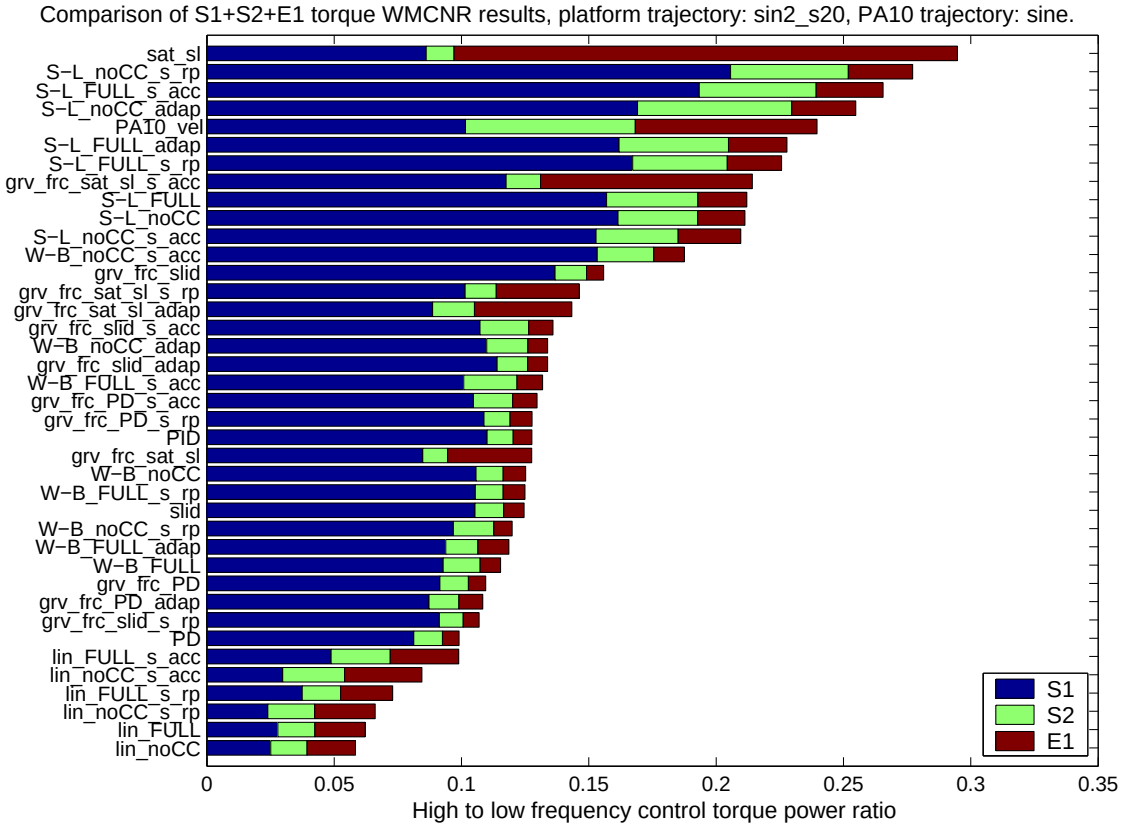


Figure 8.7: High to low frequency control torque power ratio ordered by sum for S1, S2, E1, platform trajectory: pitch sine, for both PA10 trajectories.



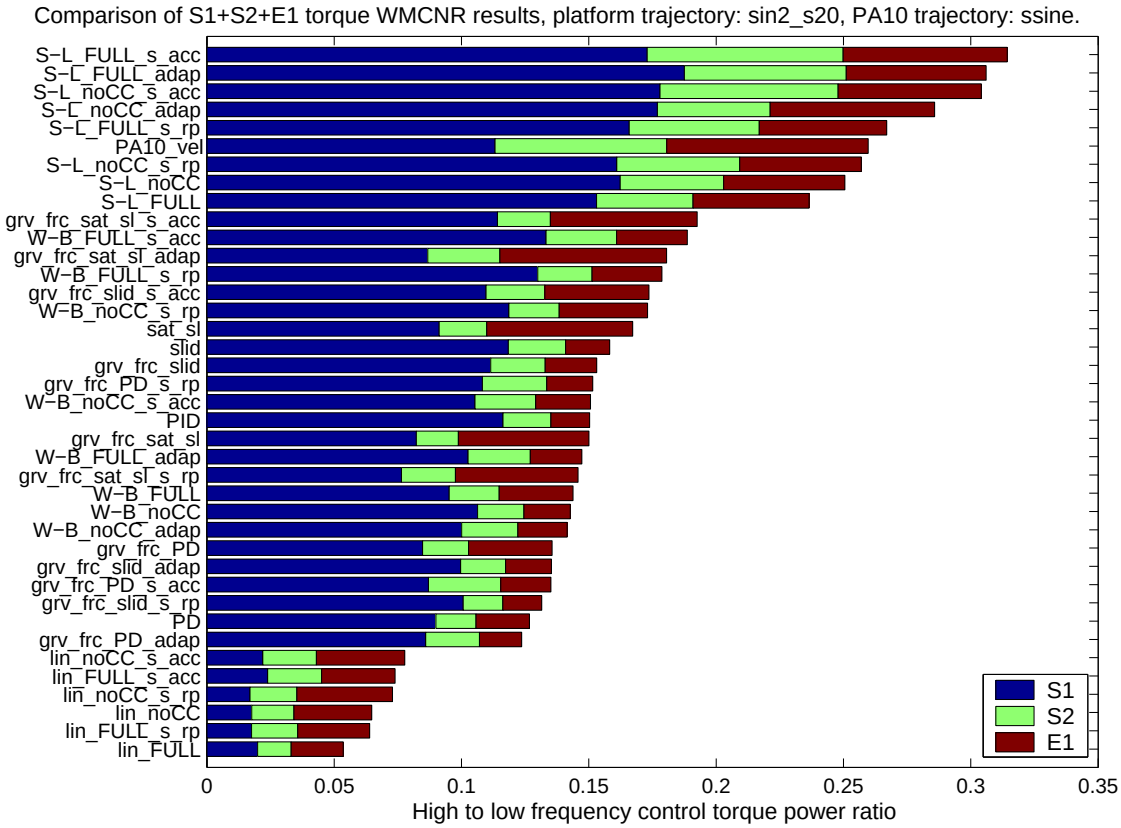


Figure 8.8: High to low frequency control torque power ratio ordered by sum for S1, S2, E1, platform trajectory: sum of sin (1), for both PA10 trajectories.

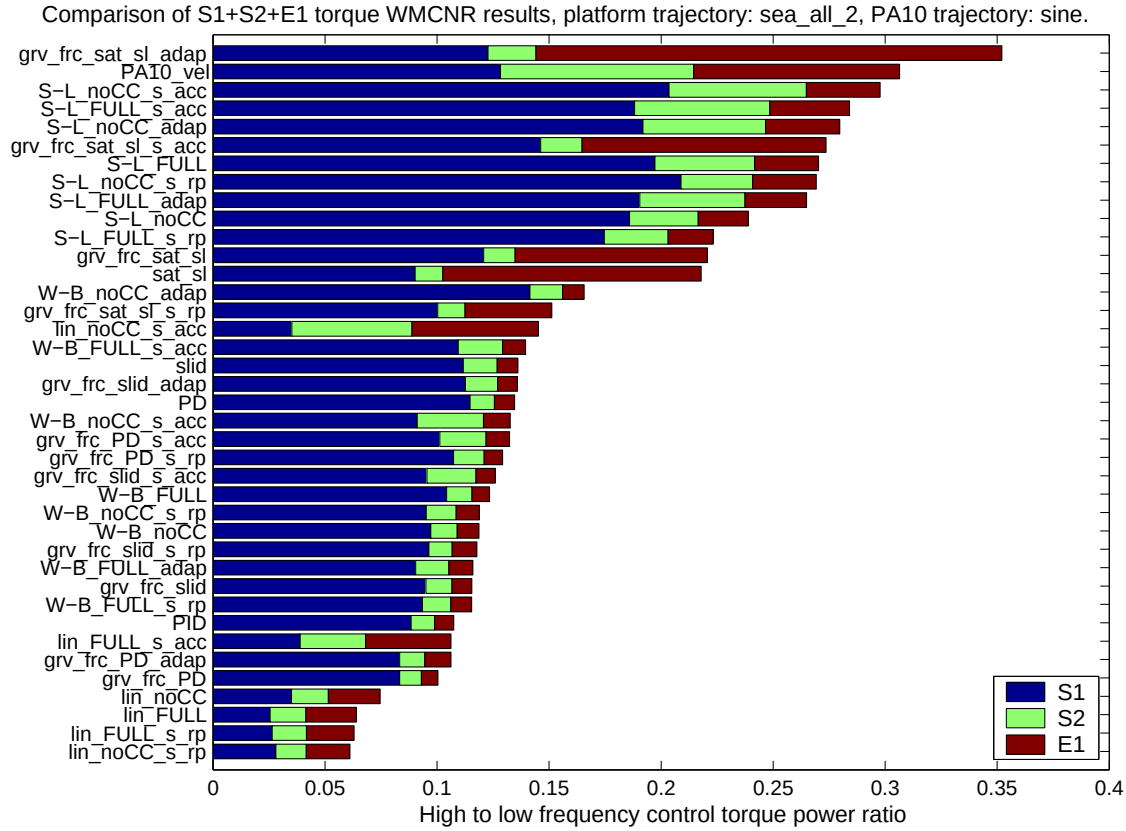


Figure 8.9: High to low frequency control torque power ratio ordered by sum for S1, S2, E1, platform trajectory: vessel on sea (6-DOF), for both PA10 trajectories.

(which is a vector of length |g| pointing in a wrong direction (to maximise the estimate to actual value differences)). Experiments have been performed with the platform switched off. The transient behaviour of adapted gravity estimates, along with the sum of absolute errors for the first three joints is presented in figures 8.10 to 8.13. It can be noticed that

- The gravity estimates reach the actual values within 1.5 s for the **`sine`** trajectory and within 1.8 s for the **`ssine`**. The cause of this can be the **`sine`** trajectory *exciting* the generic gravity term better and leading to a faster estimation. It could also be attributed to higher tracking errors at the start of the **`sine`** trajectory feeding to the adaptation step of the control algorithm.

- The estimates move close to the reference value but vary with time (trajectory). This could be explained with the adaptation compensating for unmodelled dynamics.

- The gravity estimates for the simpler controllers (gravity+friction compensation based) vary more (and therefore are more distant from the reference accelerometer reference measurements) than for the full model Wen-Bayard and Slotine-Li controllers. This can be attributed to the adaptive algorithm trying to compensate for the missing dynamics terms.

- The noise in gravity estimates present on the Slotine-Li plots originates from the physical transmission of the PA10 control torque noise (causing joint vibrations) into the platform mounting plate and to the sensors.

Gravity estimates and joint tracking errors The gravity estimates for selected experiments discussed in sections 8.4.1 and 8.4.2 are analysed here. Angle tracking errors for PA10 trajectory **`sine`** and various platform trajectories with the gravity + friction + PD controller using constant, accelerometer and adaptive gravity, along with the gravity estimates are presented in figures 8.14 to 8.17.

- The adaptive gravity estimates can be considered to be correlated with the reference accelerometer readings, however there are significant discrepancies.

- It can be observed that the joint angle errors are significantly reduced by using the gravity measurements from the accelerometer and further reduced by applying adaptive control.

- It is interesting to notice that even though the adapted estimates differ significantly from the reference sensor readings, the algorithm performs noticeably better at keeping the angle tracking errors low.

- It can also be observed that there is less *error noise* (resulting from joint vibration caused by control torque noise) present in the adaptive controller results. This translates to less physical wear of the actuator/transmission.

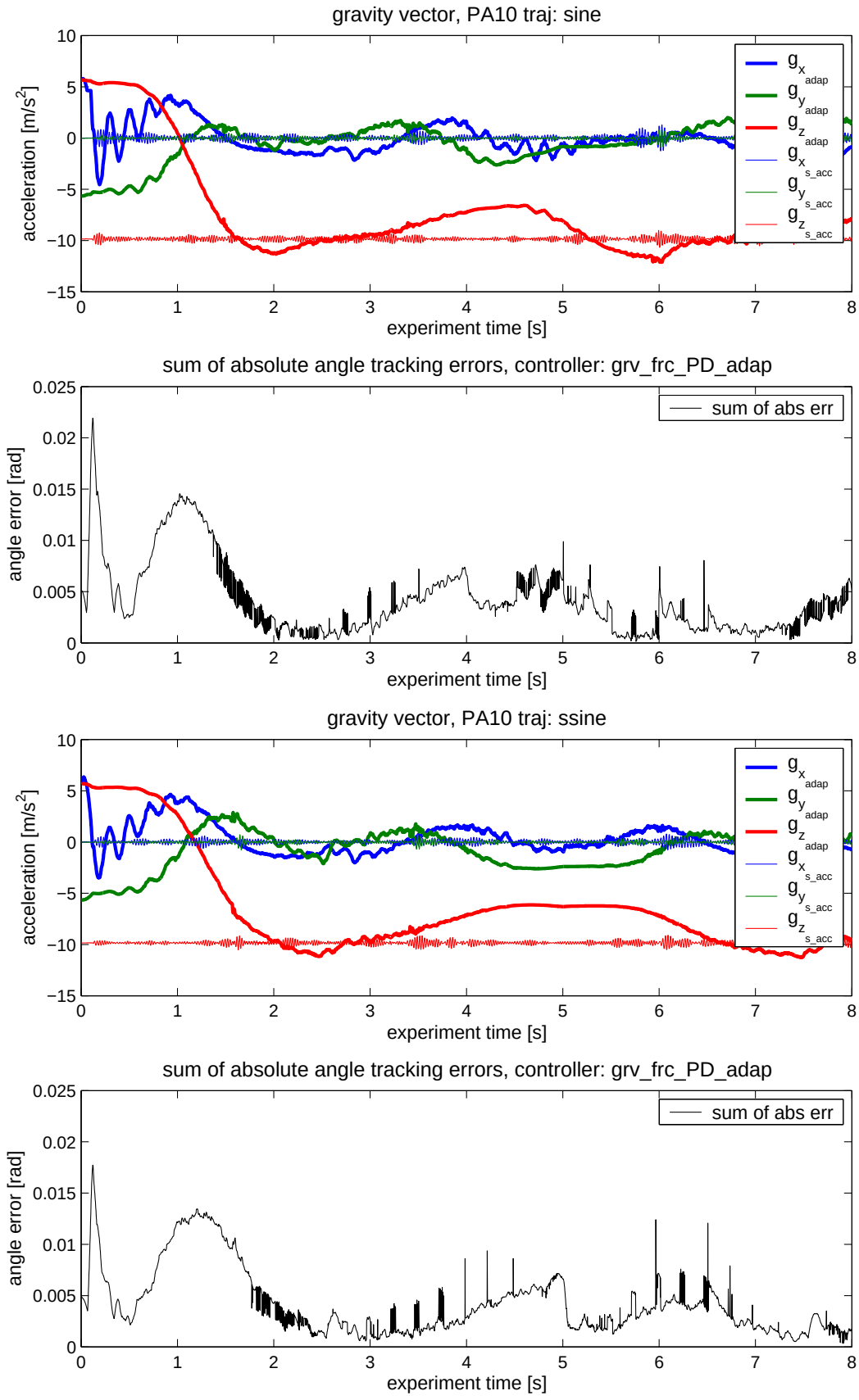


Figure 8.10: Gravity estimates with reference accelerometer measurements and sum of absolute joint tracking errors for S1, S2 and S3 with incorrectly initialised gravity vector (to $\hat{g}_0 = (5.66, -5.66, -5.66)^T$). Platform stopped, both PA10 trajectories. Controller used: gravity+friction+PD with adaptive gravity.

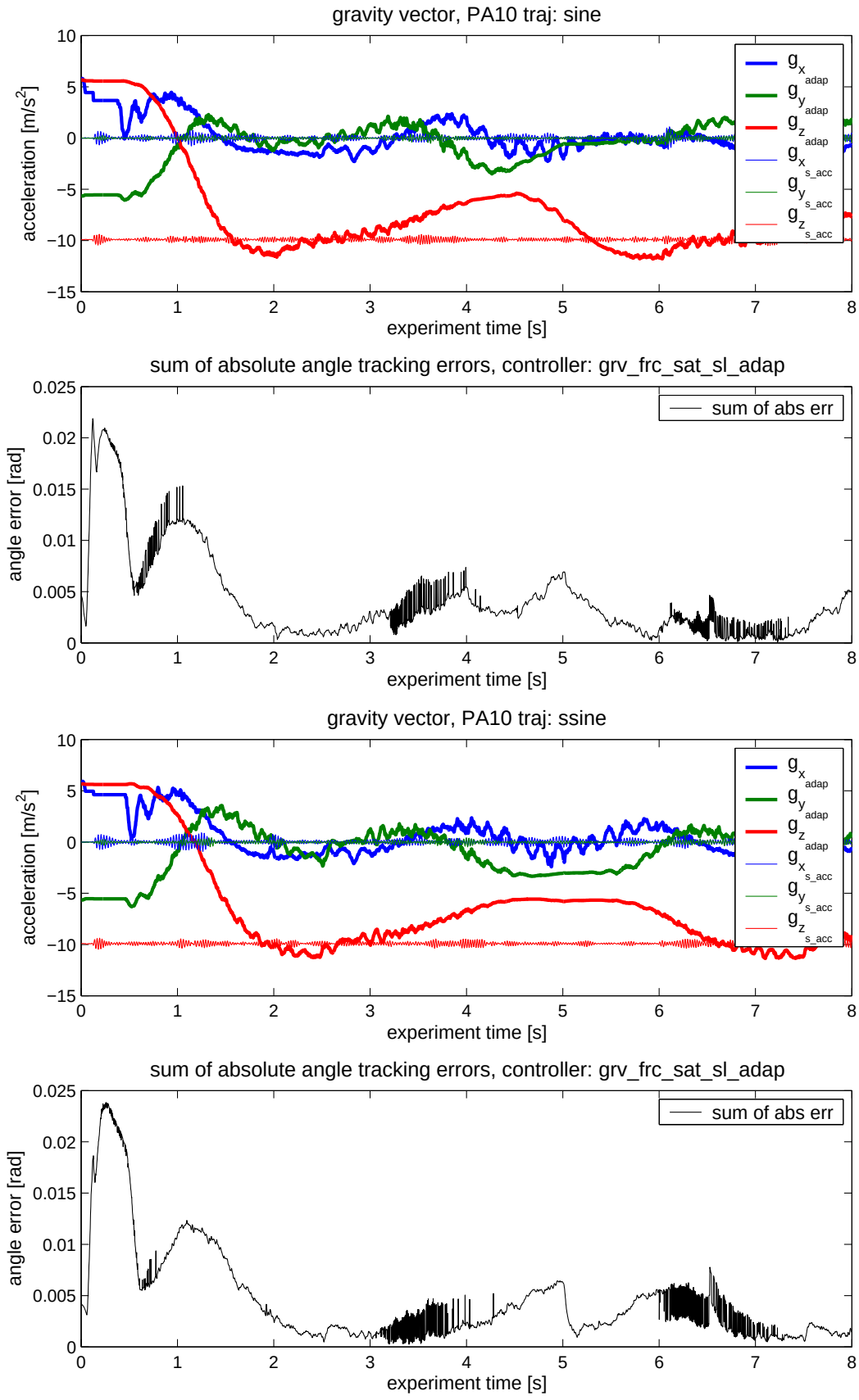


Figure 8.11: Gravity estimates with reference accelerometer measurements and sum of absolute joint tracking errors for S1, S2 and S3 with incorrectly initialised gravity vector (to $\hat{g}_0 = (5.66, -5.66, -5.66)^T$). Platform stopped, both PA10 trajectories. Controller used: gravity+friction+saturated sliding with adaptive gravity.

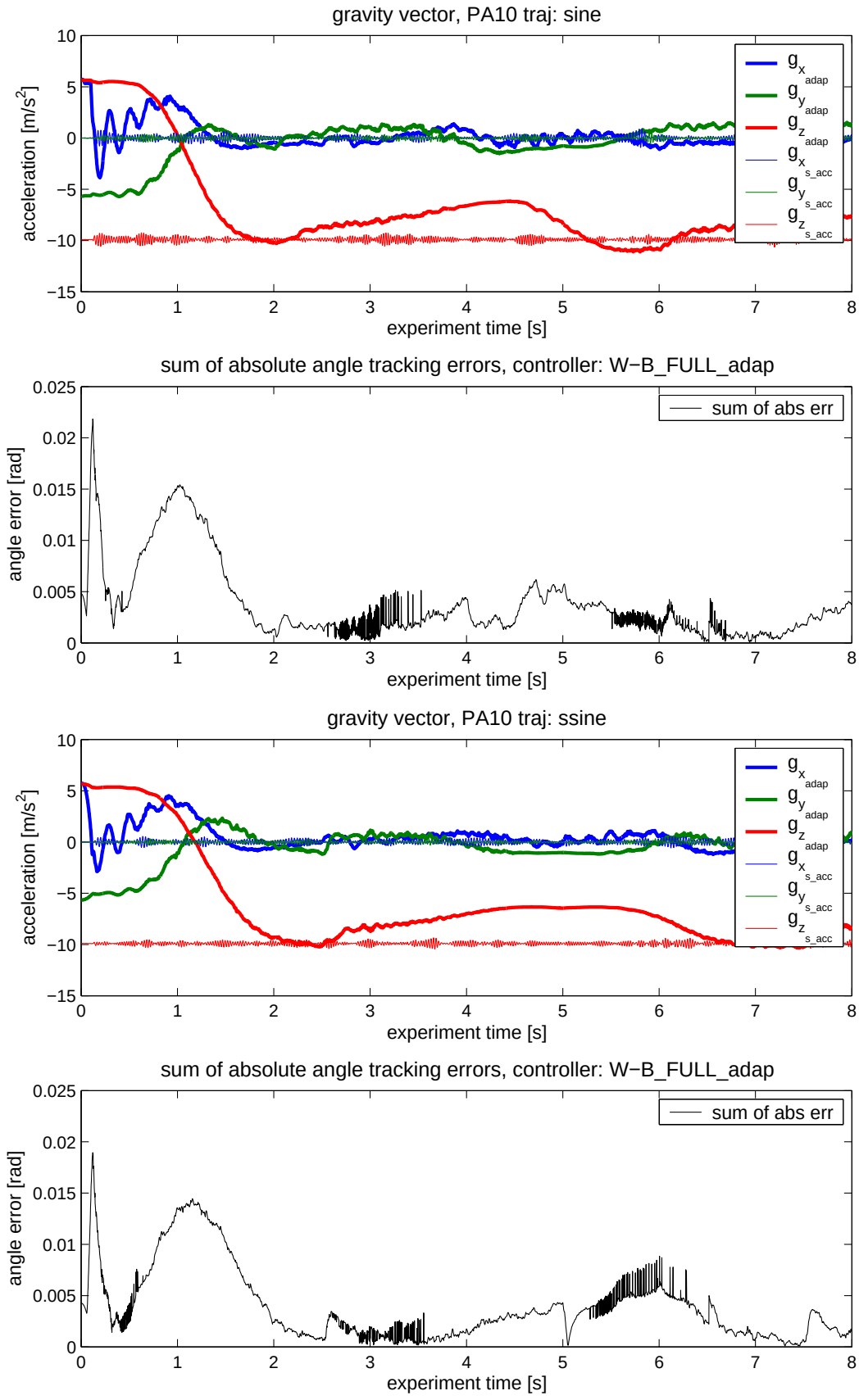


Figure 8.12: Gravity estimates with reference accelerometer measurements and sum of absolute joint tracking errors for S1, S2 and S3 with incorrectly initialised gravity vector (to $\hat{g}_0 = (5.66, -5.66, -5.66)^T$). Platform stopped, both PA10 trajectories. Controller used: Wen-Bayard with full model and adaptive gravity.

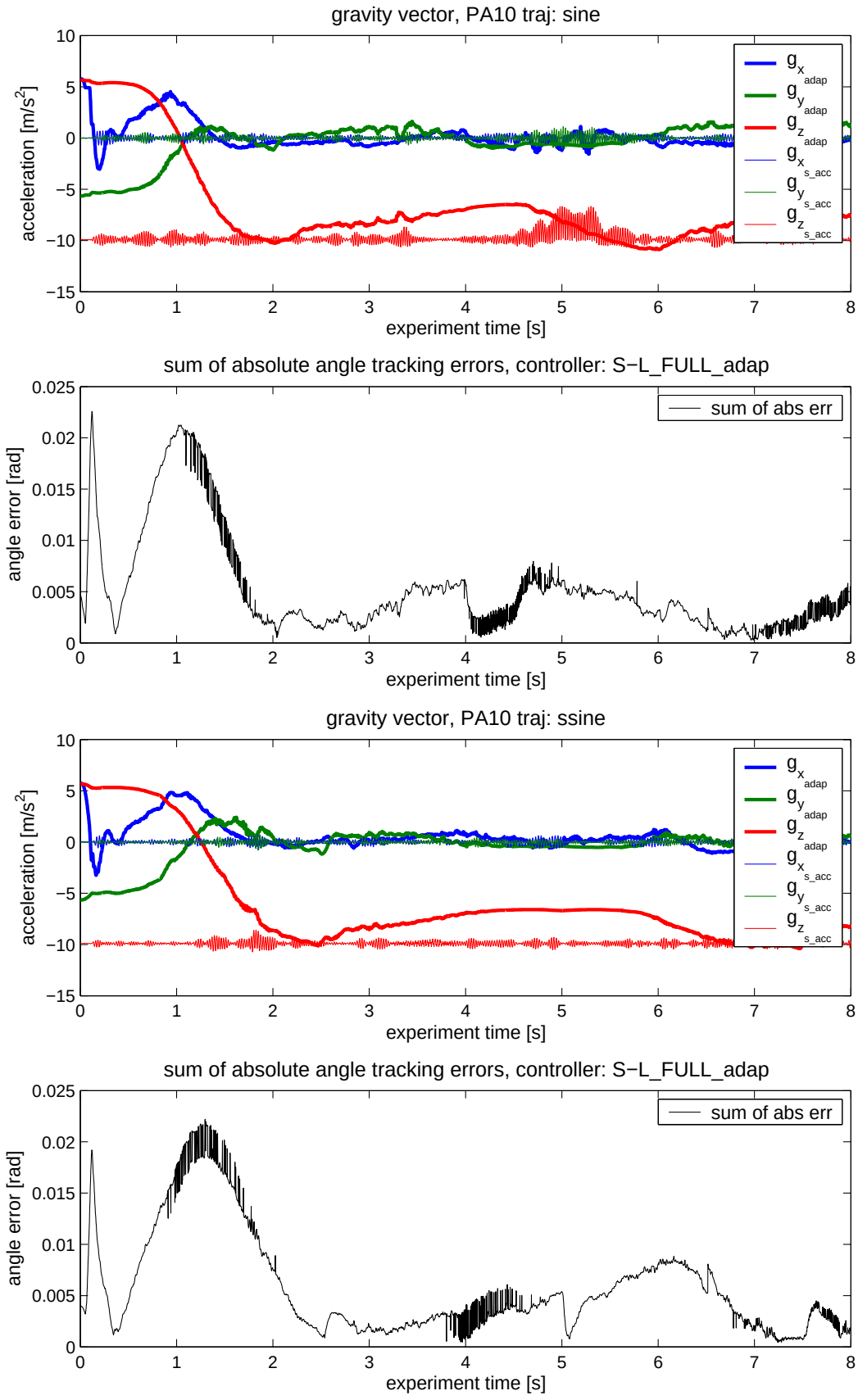


Figure 8.13: Gravity estimates with reference accelerometer measurements and sum of absolute joint tracking errors for S1, S2 and S3 with incorrectly initialised gravity vector (to $\hat{g}_0 = (5.66, -5.66, -5.66)^T$). Platform stopped, both PA10 trajectories. Controller used: Slotine-Li with full model and adaptive gravity.

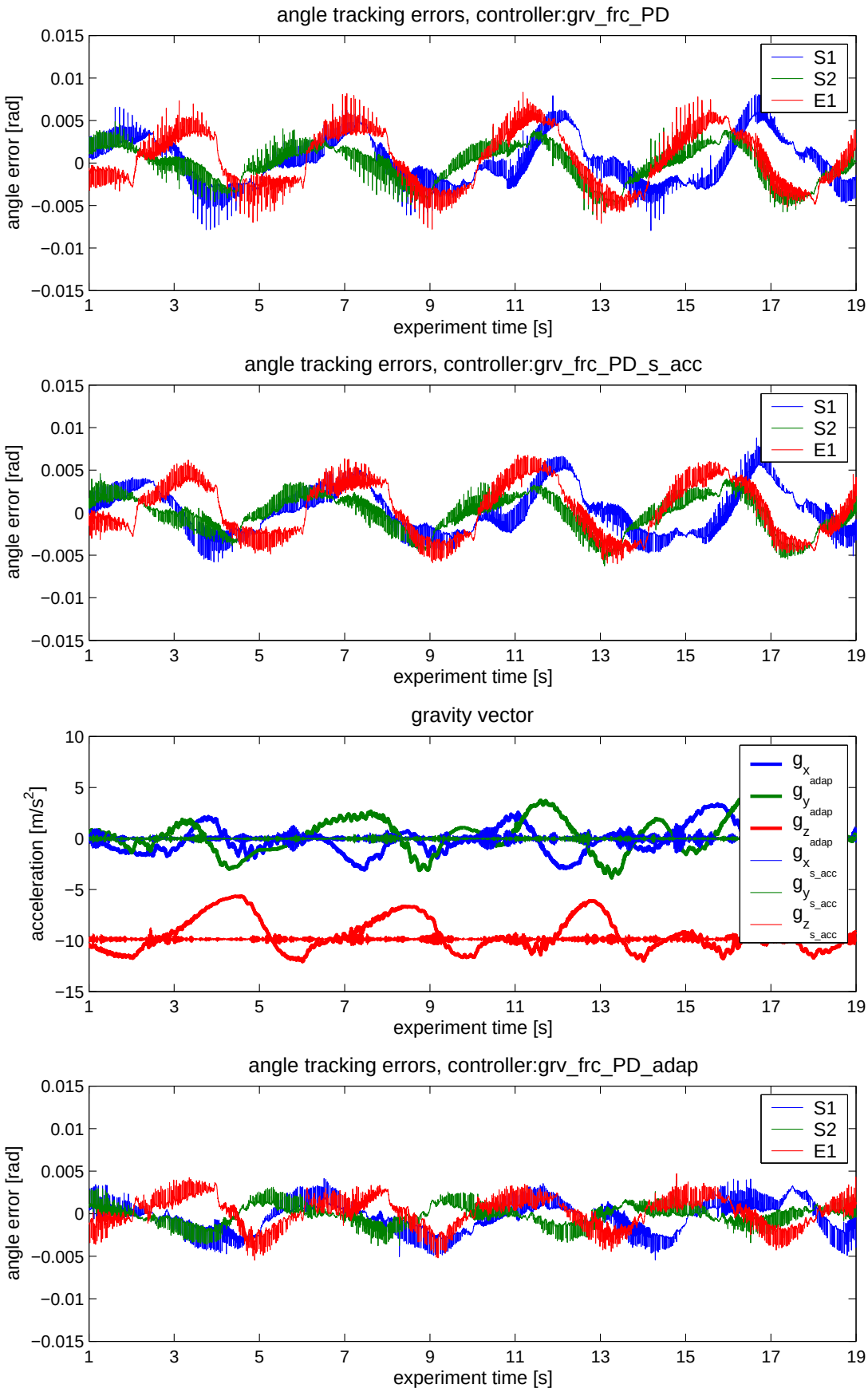


Figure 8.14: Angle tracking errors for gravity+friction+PD controller with constant gravity, accelerometer values and adaptive gravity (with gravity vector estimates). Platform trajectory: zero, PA10 trajectory **`sine`**.

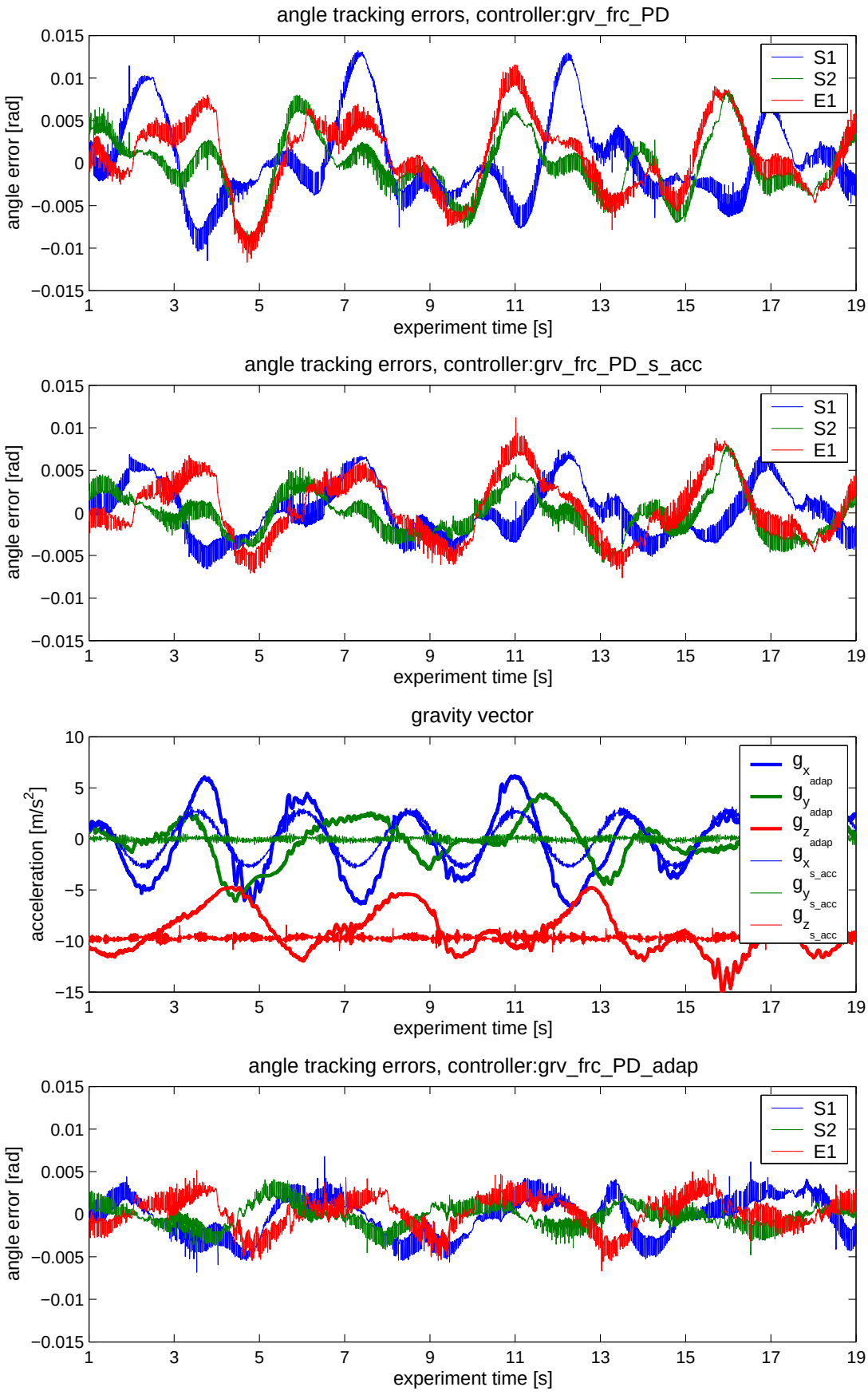


Figure 8.15: Angle tracking errors for gravity+friction+PD controller with constant gravity, accelerometer values and adaptive gravity (with gravity vector estimates). Platform trajectory: pitch sine, PA10 trajectory **`sine`**.

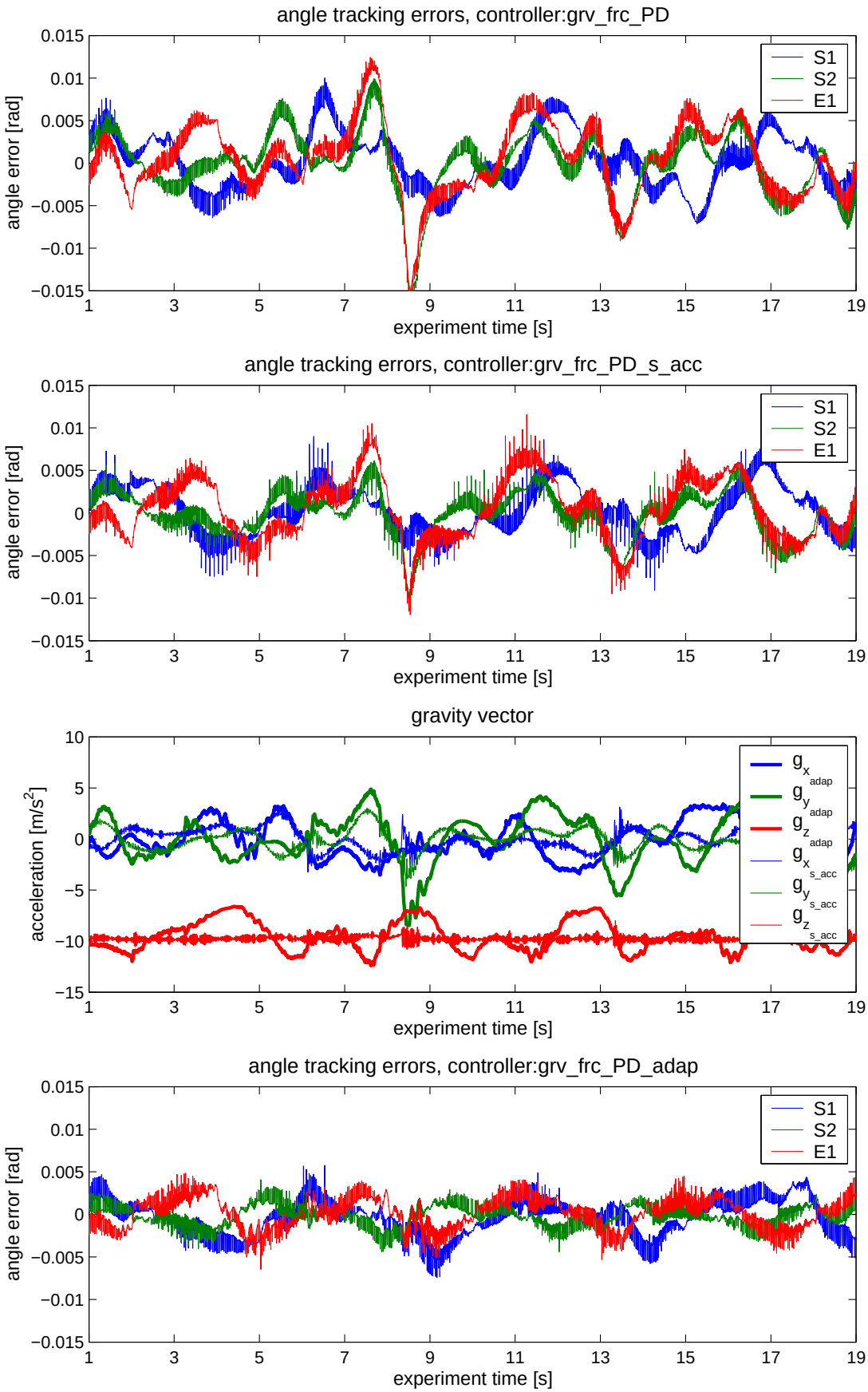


Figure 8.16: Angle tracking errors for gravity+friction+PD controller with constant gravity, accelerometer values and adaptive gravity (with gravity vector estimates). Platform trajectory: sum of sin (1), PA10 trajectory **`sine`**.

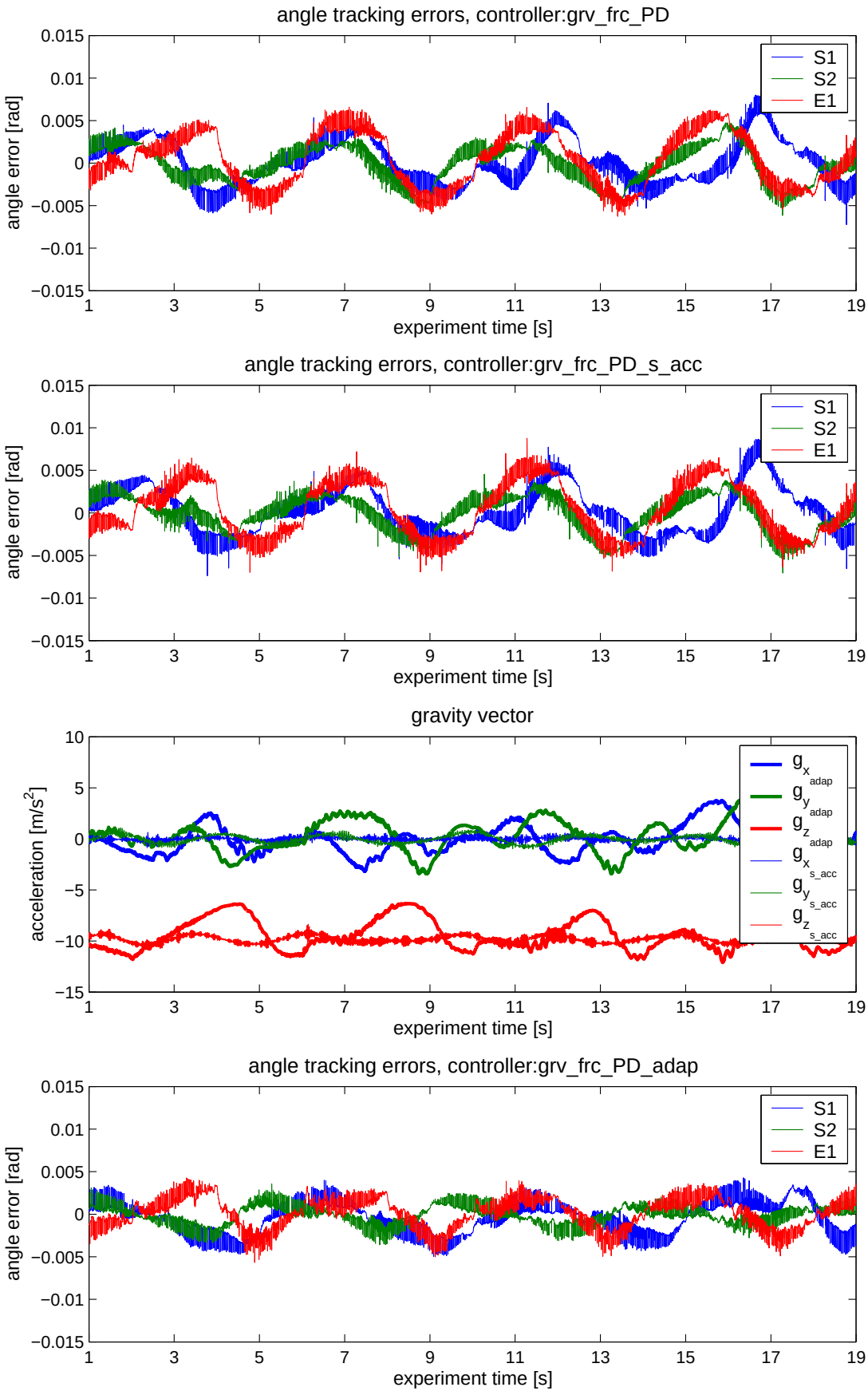


Figure 8.17: Angle tracking errors for gravity+friction+PD controller with constant gravity, accelerometer values and adaptive gravity (with gravity vector estimates). Platform trajectory: vessel on sea (6DOF), PA10 trajectory **`sine`**.

For comparison, the results of the same sets of PA10 and platform trajectories with the full model Wen-Bayard controller (using constant, accelerometer and adaptive gravity) are presented in figures 8.18 to 8.21:

- The adaptive gravity estimates are much closer to the reference sensor measurement than in the case of the reduced model (gravity+friction).
- The angle tracking errors and the *error noise* are smallest for the adaptive algorithm.

8.5 Summary

Based on the results presented in this chapter, the following conclusions can be drawn:

1. Adaptive controllers deliver superior joint angle tracking properties with average torque noise generation.
2. The Wen-Bayard controllers outperformed the Slotine-Li ones in the above tests, however it needs to be considered that these controllers may lose their performance benefits when the manipulator cannot be driven very close to the desired trajectory, which could happen:
   - for lower torque motors,
   - when a load is mounted on the tip.

   In this case, the **`sine`** trajectory velocities are within 55-126% of the PA10-6CE rated velocities (for reference see table 8.2[4]), and the torques do not saturate, which indicates that the manipulator has enough torque to remain on the desired trajectory.
3. The acceleration at the base of the manipulator is not equal to the gravitational acceleration measured at the centre of the mounting plate. There is also an element originating from the angular acceleration around the *y*-axis (pitch) due to the manipulator being mounted at a distance rX (see section 7.3.1) from the centre. The adaptive scheme attempts to compensate for this additional acceleration, as it is additive to the *z*-axis gravitational acceleration acting on the manipulator base.
4. The adaptive gravity controller attempts to compensate for unmodelled dynamics (e.g. the harmonic drive compliance/flexibility), resulting in significantly reduced angle tracking errors.

[4] it has been experimentally verified, that joints S1 and S2 are actually able to operate at 1.8 times rated velocities

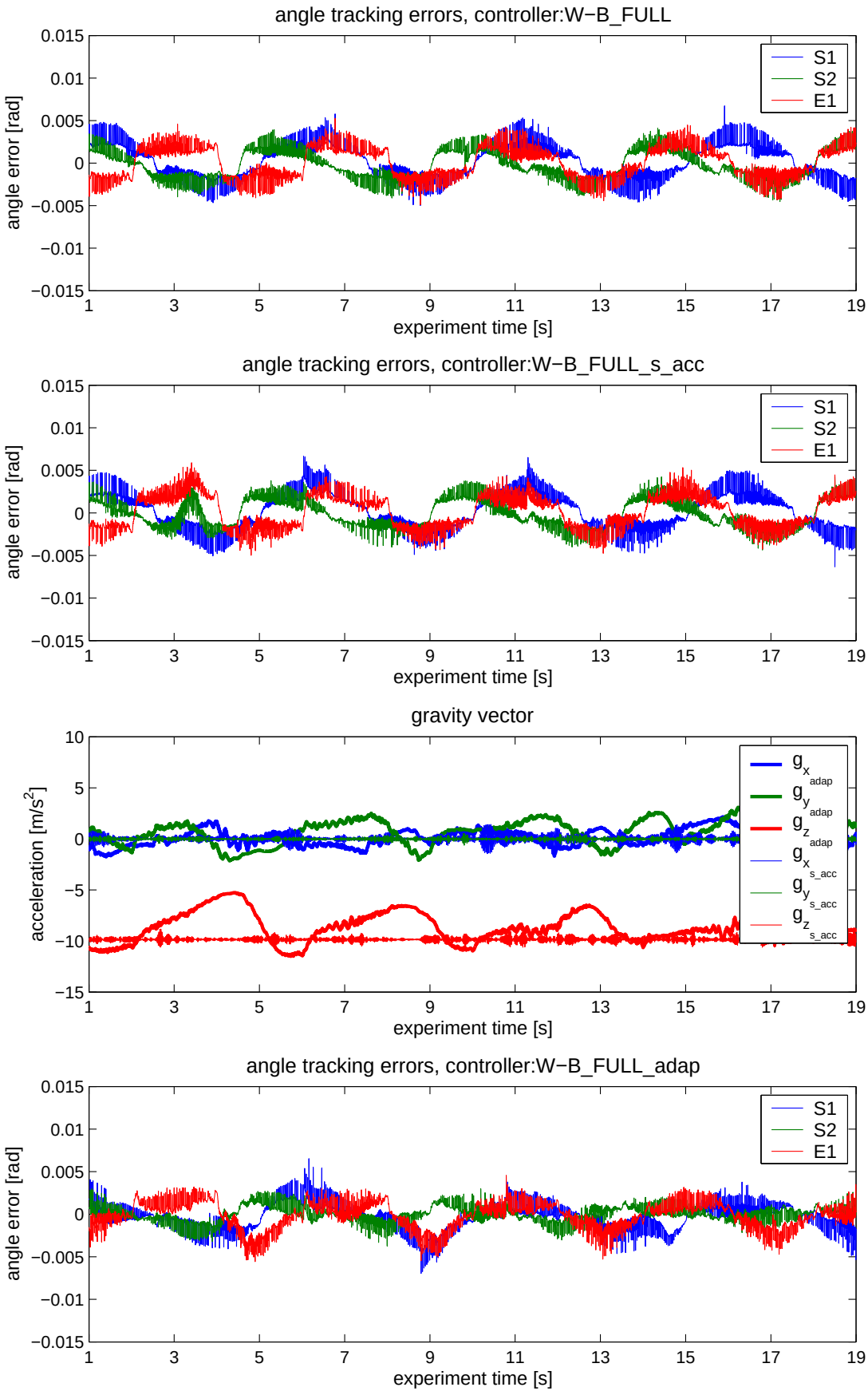


Figure 8.18: Angle tracking errors for full model based Wen-Bayard controller with constant gravity, accelerometer values and adaptive gravity (with gravity vector estimates). Platform trajectory: zero, PA10 trajectory **`sine`**.

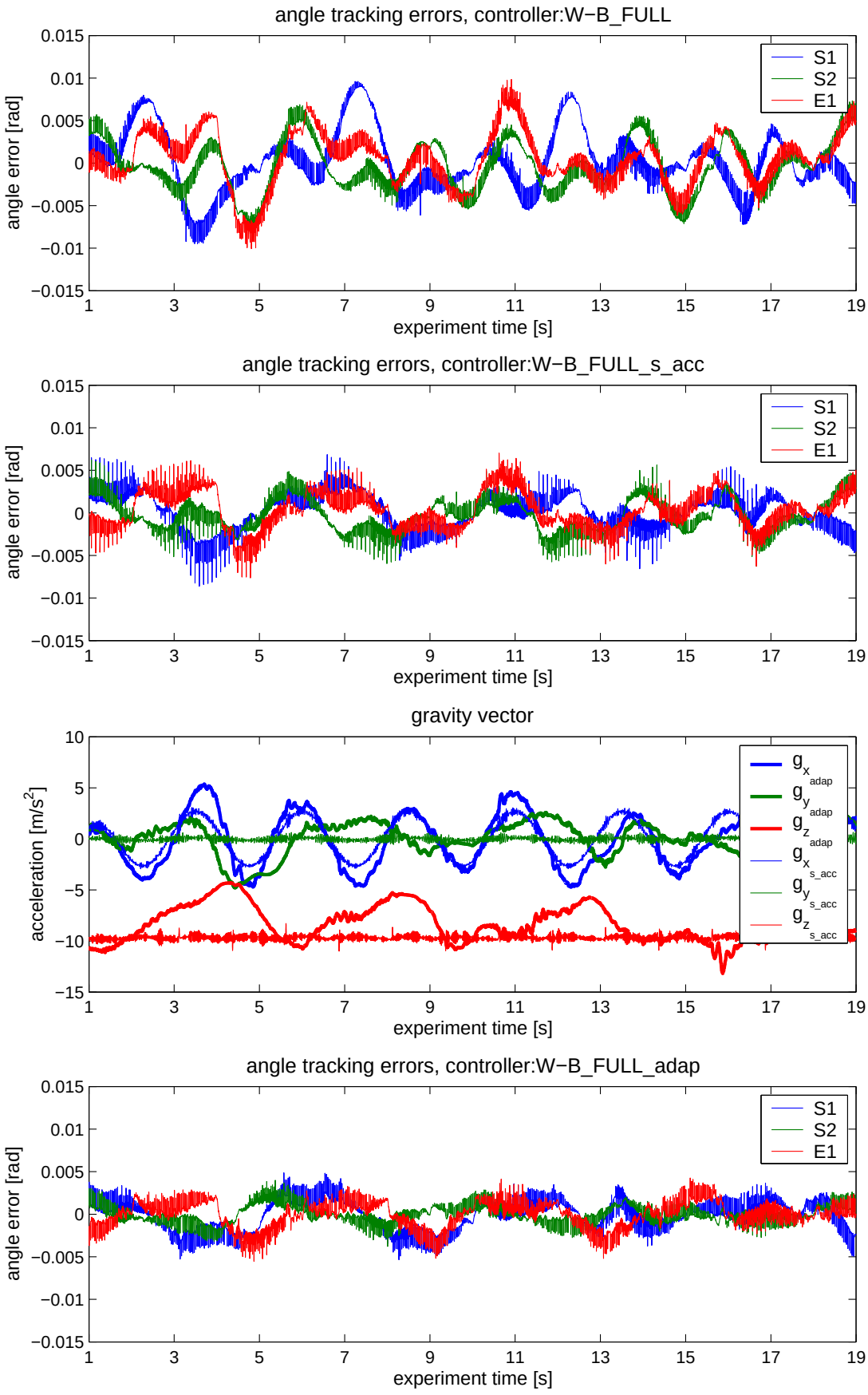


Figure 8.19: Angle tracking errors for full model based Wen-Bayard controller with constant gravity, accelerometer values and adaptive gravity (with gravity vector estimates). Platform trajectory: pitch sine, PA10 trajectory **`sine`**.

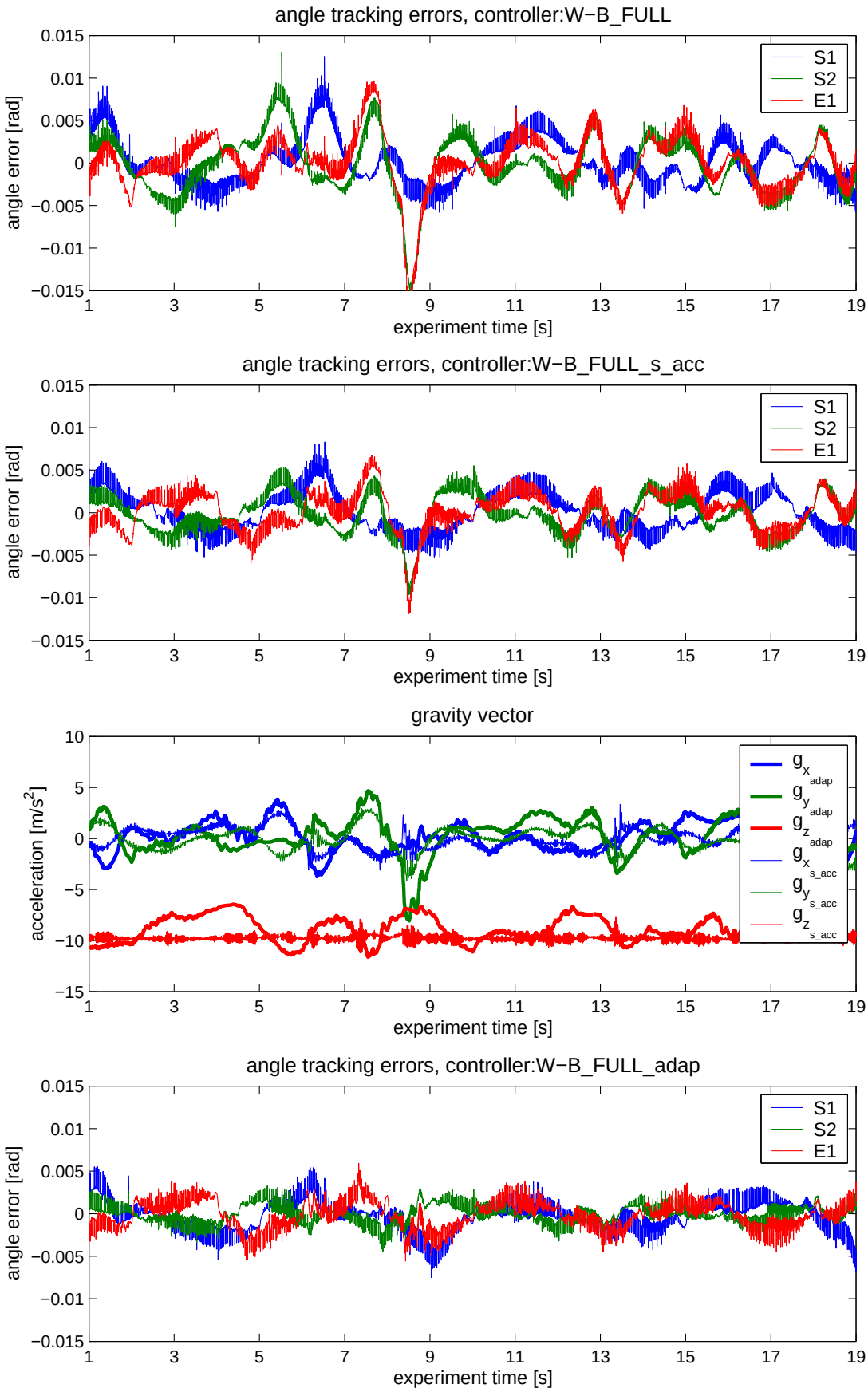


Figure 8.20: Angle tracking errors for full model based Wen-Bayard controller with constant gravity, accelerometer values and adaptive gravity (with gravity vector estimates). Platform trajectory: sum of sin (1), PA10 trajectory **`sine`**.

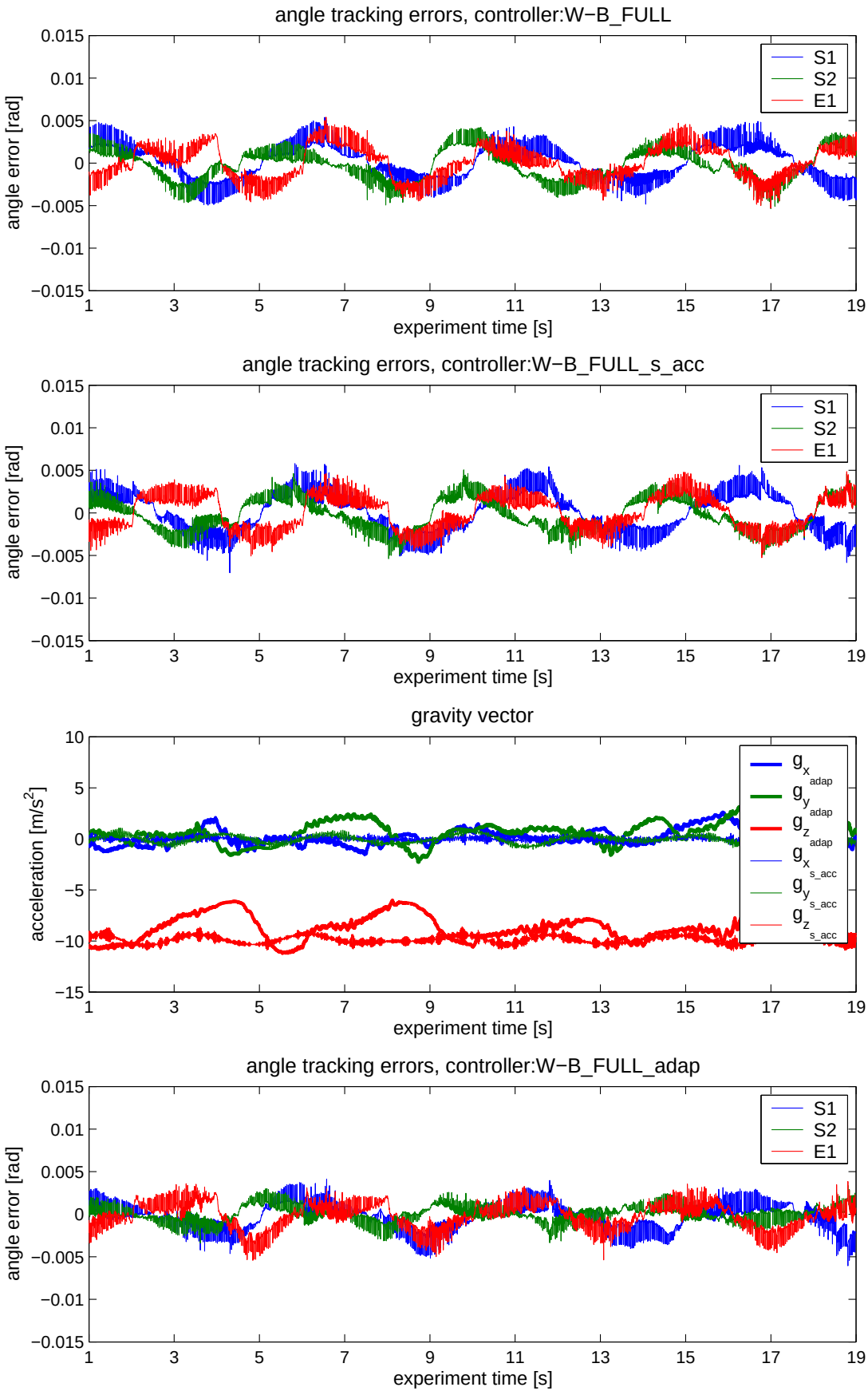


Figure 8.21: Angle tracking errors for full model based Wen-Bayard controller with constant gravity, accelerometer values and adaptive gravity (with gravity vector estimates). Platform trajectory: vessel on sea (6DOF), PA10 trajectory **`sine`**.

joint	max(abs($\dot{q}_d$))	rated $\dot{q}$	% of rated
S1	1.257	1.00	126%
S2	0.977	1.00	98%
E1	1.010	2.00	55%
E2	1.795	6.28	29%
W1	2.094	6.28	33%
W2	1.839	6.28	29%

Table 8.2: Maximum joint velocities for the **`sine`** trajectory, with reference rated velocities and fraction of rated (in rad/s).

Chapter 9

Conclusions

9.1 Summary

This thesis addresses several aspects of manipulator modelling and control on a fixed and non-fixed base with simulations and experiments employing a Mitsubishi PA10-6CE manipulator mounted on a CueSim 6-DOF motion platform. A dynamic model of the PA10-6CE manipulator with fixed base has been derived using geometric models of the links and available information on link mass and mass centres. Motor inertias have been estimated experimentally and a non-parametric friction model for each joint has been obtained. The model has been evaluated using a comparison of simulation and experimental results and the following key conclusions were reached:

- gravity and friction terms are the most important parts of the manipulator dynamic equation,
- link and motor inertias are important in modelling transients (e.g. response to a torque step change),
- Coriolis and centrifugal term has the least contribution to the dynamic model of this manipulator (due to the high transmission ratio).

Various controllers have been implemented using the derived model and their performance evaluated quantitatively by means of *the weighted mean squared joint angle tracking error* and *the weighted mean control torque high to low frequency component ratio*. The following key conclusions can be stated from these experiments:

- non-model based controllers such as PD and PID have significantly higher tracking errors in comparison to model based controllers,
- model based controllers evaluated along a desired trajectory (such as the evaluated Wen-Bayard controller [9.3]) deliver very good joint angle tracking performance without generating unnecessary high frequency torques (resulting from amplification of measurement noise) in comparison to the other controllers,

- the last three joints of the PA10-6CE manipulator are mostly affected by friction and other model based control schemes do not introduce significant improvements to their tracking error performance figures.

A model of a robotic manipulator on a moving base has been derived (with an assumption of the base not being influenced by the manipulator motion) and evaluated based on simulation and experimental closed-loop control torque comparisons for the PA10-6CE robotic arm. The following key points were observed:

- the extension of the gravity model to include base motion is the most significant element for modelling a high gear ratio manipulator on a moving base,
- the inertial and Coriolis and centrifugal torques arising from the base motion contribute little to the dynamic model accuracy (especially for slower base motion trajectories), which can again be attributed to the high gear ratio.

Based on the the modelling experiences, an adaptive model based controller (using schemes presented in [9.1] and [9.2]) has been suggested to compensate for the varying gravity vector due to base motion and compared experimentally against model based controllers employing sensor measurements (roll+pitch angles or 3-axis accelerometer) by means of *the weighted mean squared joint angle tracking error* and *the weighted mean control torque high to low frequency component ratio*. The following key conclusions were obtained:

- suggested adaptive controllers outperform sensor based controllers, due to their ability to compensate for modelled dynamics (such as harmonic drive flexibility/compliance),
- Wen-Bayard type controllers are less prone to measurement noise and deliver superior joint angle tracking performance,
- there is no improvement possible beyond the friction + PD control for the last three joints of the PA10-6CE manipulator.

In conclusion, the importance of model based control has been presented together with the importance of analysing the dynamic model terms contribution to the model to remove insignificant terms without reducing the joint angle tracking performance. It has been shown that an adaptive controller can be successfully applied to compensate for the varying gravity vector and its performance is superior in comparison to direct accelerometer measurement based algorithms.

9.2 Author's contributions

The following can be stated as novel contributions presented in this work:

- The author was the first to present a comparison of robust controllers for the PA10-6CE manipulator in [9.5].
- The author was the first to present the full dynamic model of the PA10-6CE with a nonparametric friction model and inertias of links calculated with the link element densities chosen in such way to match the link centre of mass in [9.4]. This presentation was accompanied by a brief analysis of the incorporated dynamic model terms in a comparison of simulation and experimental results using PID controllers.
- This thesis presents a novel approach to model the PA10-6CE robotic manipulator on a moving platform.
- This thesis presents an extensive study of the model of a robotic manipulator with harmonic drive transmission on a moving platform with simulation and experimental results compared to evaluate the relevance of various terms in the dynamic model.
- To the author's best knowledge this thesis is novel in presenting a successful application of various model based adaptive controllers to compensate for an unknown varying gravity vector in a robotic manipulator's gravity model.
- To the author's best knowledge this thesis is novel in presenting an extensive comparison of various manipulator controllers (mostly model based) with constant, sensor derived and an adaptive gravity vector based on angle tracking error (sum of squares) and generated torque noise (above anticipated bandwidth of the mechanical model).

9.3 Suggestions for Future Work

- Identify the transmission compliance and add it to the simulations and evaluate it in the controller.
- Add adaptation to the feedback linearisation controller and compare against existing experimental results.
- Evaluate the adaptive gravity profile against the accelerometer reading after adding the transmission compliance element to the controller. If it is well correlated, design an inertial navigation system for task space control, using the adaptive control gravity estimates to obtain the manipulator base position and compensate for base motion in task space operations.
- Evaluate a double adaptation scheme – adapt the gravity vector to compensate for base motion and adapt some of the model parameters to compensate as well e.g. a change in load carried by the manipulator. Consider adaptation of each element at each control step, or interleave both every odd/even step.

Bibliography

Chapter 1

[1] P. J. From, V. Duindam, J. T. Gravdahl, and Sastry S. Modeling and motion planning for mechanisms on a non-inertial base. In *Proc. of the IEEE Int. Conf. Robotics and Automation*, pages 3320–3326, May 2009.

[2] L.J. Love, J.F. Jansen, and F.G. Pin. Compensation of wave-induced motion and force phenomena for ship-based high performance robotic and human amplifying systems. Technical Report ORNL/TM-2003/233, Oak Ridge National Laboratory, Oak Ridge, Tenn., October 2003.

[3] A.B. Tanner. Study of robotic manipulators subjected to base disturbances. Master's thesis, Army Military Personnel Center Alexandria VA, JAN 1987.

Chapter 2

[1] C. Abdallah, D.M. Dawson, P. Dorato, and M. Jamshidi. Survey of robust control for rigid robots. *Control Systems Magazine, IEEE*, 11(2):24–30, Feb 1991.

[2] S. Arimoto. Passivity-based control. In *ICRA*, pages 227–232, 2000.

[3] B. Armstrong, O. Khatib, and J. Burdick. The explicit dynamic model and inertial parameters of the PUMA 560 arm. In *Robotics and Automation. Proc. 1986 IEEE Int. Conf.*, volume 3, pages 510–518, Apr 1986.

[4] B. Armstrong-Hélouvry. *Control of Machines with Friction*. Kluwer Academic Publishers, 1991.

[5] B. Armstrong-Hélouvry, P. Dupont, and C. Canudas de Wit. A survey of models, analysis tools and compensation methods for the control of machines with friction. *Automatica*, 30(7):1083–1138, 1994.

[6] D.S. Bayard and J.T. Wen. New class of control laws for robotic manipulators: Part 2 — adaptive case. *Int. J. of Control*, 47(5):1387–1406, 1988.

[7] N.A. Bompos, P.K. Artemiadis, A.S. Oikonomopoulos, and K.J. Kyriakopoulos. Modeling, full identification and control of the Mitsubishi PA-10 robot arm. In *Advanced intelligent mechatronics, 2007 IEEE/ASME Int. Conf.*, pages 1–6, Sept. 2007.

[8] C.I. Byrnes and J.C. Willems. Adaptive stabilization of multivariable linear systems. In *Decision and Control, 1984. The 23rd IEEE Conf.*, volume 23, pages 1574–1577, Dec. 1984.

[9] C. Canudas de Wit, G. Bastin, and B. Siciliano, editors. *Theory of Robot Control*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1996.

[10] C. Canudas de Wit, H. Olsson, K.J. Astrom, and P. Lischinsky. A new model for control of systems with friction. *Automatic Control, IEEE Trans.*, 40(3):419–425, Mar 1995.

[11] F. M. Carter and D. B. Cherchas. Motion control of non-fixed base robotic manipulators. *Robotica*, 17(2):143–157, 1999.

[12] V.W. Chen and Jr. Cannon, R.H. Experiments in nonlinear adaptive control of multi-manipulator free-flying robots. In *Robotics and Automation, 1994. Proc., 1994 IEEE Int. Conf.*, pages 2213–2220 vol.3, May 1994.

[13] P.I. Corke and B. Armstrong-Helouvry. A search for consensus among model parameters reported for the PUMA 560 robot. In *Robotics and Automation, 1994. Proc., 1994 IEEE Int. Conf.*, pages 1608–1613 vol.2, May 1994.

[14] J. Craig, P. Hsu, and S. Sastry. Adaptive control of mechanical manipulators. In *Robotics and Automation. Proc. 1986 IEEE Int. Conf.*, volume 3, pages 190–195, Apr 1986.

[15] J.J. Craig. *Introduction to robotics*. Addison - Wesley, 1986.

[16] J. Denavit and R.S. Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. *ASME Trans. J. Applied Mechanics*, 77(2):215–221, 1955.

[17] N.B. Do, A.A. Ferri, and O.A. Bauchau. Efficient simulation of a dynamic system with LuGre friction. *J. of Computational and Nonlinear Dynamics*, 2(4):281–289, 2007.

[18] S. Dubowsky and E. Papadopoulos. The kinematics, dynamics, and control of free-flying and free-floating space robotic systems. *Robotics and Automation, IEEE Trans.*, 9(5):531–543, Oct 1993.

[19] V. Duindam and S. Stramigioli. Singularity-free dynamic equations of open-chain mechanisms with general holonomic and nonholonomic joints. *Robotics, IEEE Trans.*, 24(3):517–526, June 2008.

[20] P. J. From, V. Duindam, J. T. Gravdahl, and Sastry S. Modeling and motion planning for mechanisms on a non-inertial base. In *Proc. of the IEEE Int. Conf. Robotics and Automation*, pages 3320–3326, May 2009.

[21] K. Glass and R. Colbaugh. Development and implementation of real-time control modules for robotic waste management. In *Robotics and Automation, 1997. Proc., 1997 IEEE Int. Conf.*, volume 1, pages 650–655 vol.1, Apr 1997.

[22] J.-M. Godhavn. High quality heave measurements based on gps rtk and accelerometer technology. In *OCEANS 2000 MTS/IEEE Conference and Exhibition*, volume 1, pages 309–314 vol.1, 2000.

[23] R. Gorez and D. Galardini. Robot control with disturbance observers. In *Advanced Robotics, 1991. 'Robots in Unstructured Environments', 91 ICAR., Fifth Int. Conf.*, pages 417–422 vol.1, Jun 1991.

[24] V.E. Gough. Contribution to discussion of papers on research in automobile stability, control and tyre performance. Proc. Auto Div. Inst. Mech. Eng., 1956-1957.

[25] S. R. Hamner. Static positional accuracy and joint stiffness characterization of a Mitsubishi PA10-6CE robot arm. Master's thesis, University of Florida Orthopaedic Biomechanics Laboratory, Departament of Mechanical Aerospace Engineering, 2006.

[26] M. Hashimoto. Robot motion control based on joint torque sensing. In *Robotics and Automation, 1989. Proc., 1989 IEEE Int. Conf.*, pages 256–261 vol.1, May 1989.

[27] S. Ho, A. Jones, and C.S. Cox. Torque disturbances and their effect on process control. In *Power Electronics and Variable-Speed Drives, 1994. Fifth Int. Conf.*, pages 602–607, Oct 1994.

[28] R. Jamisola. Full dynamics identification and control of PUMA 560 and MITSUBISHI PA-10 robots. Master's thesis, National University of Singapore, Department of Mechanical and Production Engineering, 2001.

[29] R. Jamisola, M. Ang, T.M. Lim, O. Khatib, and S.Y. Lim. Dynamics identification and control of an industrial robot. In *The 9th Int. Conf. Advanced Robotics*, pages 323–328, 1999.

[30] J. Joshi and A. Desrochers. Modeling and control of a mobile robot subject to disturbances. In *Robotics and Automation. Proc. 1986 IEEE Int. Conf.*, volume 3, pages 1508–1513, Apr 1986.

[31] C.W. Kennedy and J.P. Desai. Estimation and modeling of the harmonic drive transmission in the Mitsubishi PA-10 robot arm. In *Intelligent Robots and Systems, 2003.*

(IROS 2003). Proc. 2003 IEEE/RSJ Int. Conf., volume 4, pages 3331–3336 vol.3, Oct. 2003.

[32] C.W. Kennedy and J.P. Desai. Modeling and control of the Mitsubishi PA-10 robot arm harmonic drive system. *Mechatronics, IEEE/ASME Trans.*, 10(3):263–274, June 2005.

[33] W. Khalil and E. Dombre. *Modeling, Identification & Control of Robots*. Hermes Penton Science, 2002.

[34] N.M. Kircanski and A.A. Goldenberg. An Experimental Study of Nonlinear Stiffness, Hysteresis, and Friction Effects in Robot Joints with Harmonic Drives and Torque Sensors. *The Int. J. of Robotics Research*, 16(2):214–239, 1997.

[35] K. Kosuge, H. Kajita, and T. Fukuda. Force control of manipulator/vehicle system floating on the water utilizing vehicle restoring force. In *Advanced Motion Control, 1996. AMC '96-MIE. Proc., 1996 4th Int. Workshop*, volume 2, pages 506–511 vol.2, Mar 1996.

[36] K. Kozłowski. *Modelling and identification in robotics*. Springer-Verlag, 1998.

[37] K. Kreutz. On manipulator control by exact linearization. *Automatic Control, IEEE Trans.*, 34(7):763–767, Jul 1989.

[38] N. Lauzier, C. Gosselin, T. Laliberté, and Tremblay P. Adaptive gravity compensation of decoupled parallel and serial manipulators using a passive hydraulic transmission. *Proc. of the Institution of Mechanical Engineers, Part C: J. of Mechanical Engineering Science*, 225, 2009.

[39] P.-M. Lee and J. Yuh. Application of non-regressor based adaptive control to an underwater mobile platform-mounted manipulator. In *Control Applications, 1999. Proc. of the 1999 IEEE Int. Conf.*, volume 2, pages 1135–1140 vol. 2, 1999.

[40] J.Y. Lew and S.-M. Moon. A simple active damping control for compliant base manipulators. *Mechatronics, IEEE/ASME Trans.*, 6(3):305–310, Sep 2001.

[41] C. Lightcap and S. Banks. Dynamic identification of a Mitsubishi PA10-6CE robot using motion capture. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ Int. Conf.*, pages 3860–3865, 29 2007-Nov. 2 2007.

[42] L.J. Love, J.F. Jansen, and F.G. Pin. Compensation of wave-induced motion and force phenomena for ship-based high performance robotic and human amplifying systems. Technical Report ORNL/TM-2003/233, Oak Ridge National Laboratory, Oak Ridge, Tenn., October 2003.

[43] L.J. Love, J.F. Jansen, and F.G. Pin. On the modeling of robots operating on ships. In *Robotics and Automation, 2004. Proc. ICRA '04. 2004 IEEE Int. Conf.*, volume 3, pages 2436–2443, April-1 May 2004.

[44] J. Y. S. Luh, W. D. Fisher, and R. P. C. Paul. Joint torque control by a direct feedback for industrial robots. In *Decision and Control including the Symposium on Adaptive Processes, 1981 20th IEEE Conf.*, volume 20, pages 265–270, Dec. 1981.

[45] Q.-H. Max Meng. Comparison study of model-based and non-model-based robot controllers. In *Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century., IEEE Int. Conf.*, volume 1, pages 61–66 vol.1, Oct 1995.

[46] A. Mazur. Uniwersalny adaptacyjny λ-śledzący układ sterowania robota. In *Materiały 5 Krajowej Konferencji Robotyki*, pages 124–131, 1996.

[47] S. McMillan, D.E. Orin, and R.B. McGhee. Efficient dynamic simulation of an underwater vehicle with a robotic manipulator. *Systems, Man and Cybernetics, IEEE Trans.*, 25(8):1194–1206, Aug 1995.

[48] J.P. Merlet. *Parallel Robots*. Kluwer Academic Publishers, 2000.

[49] R. H. Middleton and G. C. Goodwin. Adaptive computed torque control for rigid link manipulators. *Syst. Control Lett.*, 10(1):9–16, 1988.

[50] R. H. Middletone and G. C. Goodwin. Adaptive computed torque control for rigid link manipulators. In *Decision and Control, 1986 25th IEEE Conf.*, volume 25, pages 68–73, Dec. 1986.

[51] S.A.A. Moosavian and E. Papadopoulos. Free-flying robots in space: an overview of dynamics modeling, planning and control. *Robotica*, 25(5):537–547, 2007.

[52] S.H. Murphy, J.T.-Y. Wen, and G.N. Saridis. Simulation of cooperating robot manipulators on a mobile platform. *Robotics and Automation, IEEE Trans.*, 7(4):468–478, Aug 1991.

[53] C.P. Neuman and J.J. Murray. The complete dynamic model and customized algorithms of the PUMA robot. *Systems, Man and Cybernetics, IEEE Trans.*, 17(4):635–644, July 1987.

[54] M.M. Olsen and H.G. Petersen. A new method for estimating parameters of a dynamic robot model. *Robotics and Automation, IEEE Trans.*, 17(1):95–100, Feb 2001.

[55] E. Papadopoulos and S. Dubowsky. On the nature of control algorithms for space manipulators. In *Robotics and Automation, 1990. Proc., 1990 IEEE Int. Conf.*, pages 1102–1108 vol.2, May 1990.

[56] E. Papadopoulos and S.A.A. Moosavian. Dynamics and control of space free-flyers with multiple arms. *J. Adv. Robot*, 9(6):603–624, 1995.

[57] Z. Qu and J. Dorsey. Robust tracking control of robots by a linear feedback law. *Automatic Control, IEEE Trans.*, 36(9):1081–1084, Sep 1991.

[58] N. Sadegh and R. Horowitz. Stability analysis of an adaptive controller for robotic manipulators. In *Robotics and Automation. Proc. 1987 IEEE Int. Conf.*, volume 4, pages 1223–1229, Mar 1987.

[59] B. Siciliano and L. Villani. *Robot Force Control*. Springer, 2000.

[60] J.-J. E. Slotine and W. Li. Composite adaptive control of robot manipulators. *Automatica*, 25(4):509–519, 1989.

[61] J.-J.E. Slotine and Weiping Li. Adaptive strategies in constrained manipulation. In *Robotics and Automation. Proc. 1987 IEEE Int. Conf.*, volume 4, pages 595–601, Mar 1987.

[62] M. Spong and R. Ortega. On adaptive inverse dynamics control of rigid robots. *IEEE Trans. Autom. Contr.*, 35(1):92–95, 1990.

[63] M.W. Spong and M. Vidyasagar. *Robot dynamics and control*. Wiley, 1989.

[64] D. Stewart. A platform with 6 degrees of freedom. *Proc. of the Institution of mechanical engineers*, 180(1,15):371–386, 1965.

[65] H.D. Taghirad. *Robust torque control of harmonic drive systems*. PhD thesis, McGill University, Montreal, Department of Electrical Engineering, 1997.

[66] H.D. Taghirad and P.R. Bélanger. A nonlinear model for harmonic drive friction and compliance, 1998.

[67] A.B. Tanner. Study of robotic manipulators subjected to base disturbances. Master's thesis, Army Military Personnel Center Alexandria VA, JAN 1987.

[68] T. J. Tarn, A. K. Bejczy, Sh. Han, and X. Yun. Inertia parameters of PUMA 560 robot arm. Technical Report SSM-RL-85-01, Departament of System Science and Mathematics, Washington University, St. Louis, Missouri 63130, September 1985.

[69] T. J. Tarn, A. K. Bejczy, and X. Yun. Dynamic equations for PUMA 560 robot arm. Technical Report SSM-RL-85-02, Departament of System Science and Mathematics, Washington University, St. Louis, Missouri 63130, July 1985.

[70] T.-J. Tarn, A.K. Bejczy, X. Yun, and Z. Li. Effect of motor dynamics on nonlinear feedback robot arm control. *Robotics and Automation, IEEE Trans.*, 7(1):114–122, Feb 1991.

[71] K. Tchoń, A. Mazur, I. Dulęba, R. Hossa, and R. Muszyński. *Manipulators and Mobile Robots: Modelling, Motion Planning, and Control. (in Polish)*. Academic Publishing House, 2000.

[72] T.D. Tuttle and W.P. Seering. A nonlinear model of a harmonic drive gear transmission. *Robotics and Automation, IEEE Trans.*, 12(3):368–374, Jun 1996.

[73] W. Tuttle, T.D. Seering. Modeling a harmonic drive gear transmission. In *Robotics and Automation, 1993. Proc., 1993 IEEE Int. Conf.*, volume 2, pages 624–629, 1993.

[74] Z. Vafa and S. Dubowsky. On the dynamics of manipulators in space using the virtual manipulator approach. In *Robotics and Automation. Proc. 1987 IEEE Int. Conf.*, volume 4, pages 579–585, Mar 1987.

[75] J.T. Wen and D.S. Bayard. New class of control laws for robotic manipulators: Part 1 — non-adaptive case. *Int. J. of Control*, 47(5):1361–1386, 1988.

[76] L.L. Whitcomb, A.A. Rizzi, and D.E. Koditschek. Comparative experiments with a new adaptive controller for robot arms. *Robotics and Automation, IEEE Trans.*, 9(1):59–70, Feb 1993.

[77] C.M. Wronka and M.W. Dunnigan. Derivation and analysis of the full dynamic model for a Mitsubishi PA10-6CE manipulator. In *Methods and Models in Automation and Robotics*. IEEE, SEP 2005.

[78] C.M. Wronka, A. Uhlmann, M.W. Dunnigan, and A.M.S. Zalzala. Fast robust controllers for a Mitsubishi PA-10 manipulator developed under QNX Neutrino. In *ISR2004*, MAR 2004.

[79] A.M.S. Zalzala and A.S. Morris, editors. *Neural networks for robotic control: theory and application*. Ellis Horwood, 1996.

Chapter 3

[1] B. Armstrong-Hélouvry. *Control of Machines with Friction*. Kluwer Academic Publishers, 1991.

[2] J.J. Craig. *Introduction to robotics*. Addison - Wesley, 1986.

[3] J. Denavit and R.S. Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. *ASME Trans. J. Applied Mechanics*, 77(2):215–221, 1955.

[4] K.S. Fu, R.C. Gonzales, and C.S.G. Lee. *ROBOTICS, Control, Sensing, Vision and Intelligence*. McGraw-Hill, 1987.

[5] M.W. Spong and M. Vidyasagar. *Robot dynamics and control*. Wiley, 1989.

[6] K. Tchoń, A. Mazur, I. Dulęba, R. Hossa, and R. Muszyński. *Manipulators and Mobile Robots: Modelling, Motion Planning, and Control. (in Polish)*. Academic Publishing House, 2000.

Chapter 4

[1] M.W. Dunnigan C.M. Wronka. Internet remote control interface for a multipurpose robotic arm. *The Int. J. Of Advanced Robotic Systems*, 3:179–182, June 2006.

[2] R. Jamisola, M. Ang, T.M. Lim, O. Khatib, and S.Y. Lim. Dynamics identification and control of an industrial robot. In *The 9th Int. Conf. Advanced Robotics*, pages 323–328, 1999.

[3] C.W. Kennedy and J.P. Desai. Estimation and modeling of the harmonic drive transmission in the Mitsubishi PA-10 robot arm. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proc. 2003 IEEE/RSJ Int. Conf.*, volume 4, pages 3331–3336 vol.3, Oct. 2003.

[4] C.W. Kennedy and J.P. Desai. Modeling and control of the Mitsubishi PA-10 robot arm harmonic drive system. *Mechatronics, IEEE/ASME Trans.*, 10(3):263–274, June 2005.

[5] M.M. Olsen and H.G. Petersen. A new method for estimating parameters of a dynamic robot model. *Robotics and Automation, IEEE Trans.*, 17(1):95–100, Feb 2001.

[6] T. Scherer. pa10kinematic. RCCL/RCI C functions, 1999.

Chapter 5

[1] Z. Qu and J. Dorsey. Robust tracking control of robots by a linear feedback law. *Automatic Control, IEEE Trans.*, 36(9):1081–1084, Sep 1991.

[2] J.T. Shearer and B.T. Kulakowski. *Dynamic Modeling and Control of Engineering Systems*. Maccillan-Collier, 1990.

[3] J.-J.E. Slotine. Sliding controller design for non-linear systems. *Int. J. of Control*, 40(2):421–434, 1984.

[4] J.-J.E. Slotine and W. Li. Adaptive manipulator control: A case study. *IEEE Trans. Autom. Contr*, 33(11):995–1003, 1988.

[5] J.-J.E. Slotine and S. S. Sastry. Tracking control of non-linear systems using sliding surfaces, with application to robot manipulators. *Int. J. of Control*, 38(2):465–492, 1983.

[6] M.W. Spong and M. Vidyasagar. *Robot dynamics and control*. Wiley, 1989.

[7] K. Tchoń, A. Mazur, I. Dulęba, R. Hossa, and R. Muszyński. *Manipulators and Mobile Robots: Modelling, Motion Planning, and Control. (in Polish)*. Academic Publishing House, 2000.

[8] A. Uhlmann. Development of driver software to support the Mitsubishi PA-10 manipulator under the QNX Neutrino Real - Time Operating System. Master's thesis, Heriot - Watt University, 2003.

[9] J.T. Wen and D.S. Bayard. New class of control laws for robotic manipulators: Part 1 — non-adaptive case. *Int. J. of Control*, 47(5):1361–1386, 1988.

[10] C.M. Wronka and M.W. Dunnigan. Derivation and analysis of the full dynamic model for a Mitsubishi PA10-6CE manipulator. In *Methods and Models in Automation and Robotics*. IEEE, SEP 2005.

[11] C.M. Wronka, A. Uhlmann, M.W. Dunnigan, and A.M.S. Zalzala. Fast robust controllers for a Mitsubishi PA-10 manipulator developed under QNX Neutrino. In *ISR2004*, MAR 2004.

Chapter 6

[1] *Guide to the Motionbase motion queing algorithms*, 1999. Motionbase.

[2] *Maxcue 600 - series operator's manual*, 2001. Motionbase.

[3] Crossbow, http://www.xbow.com/. *3-axis Accelerometers*.

[4] Measurement Specialties, Inc, http://www.meas-spec.com/. *AccuStar Electronic Clinometer*.

[5] OmniInstruments, http://www.omniinstruments.co.uk/. *Gravity referenced servo accelerometer A215/A220*.

Chapter 7

[1] K.S. Fu, R.C. Gonzales, and C.S.G. Lee. *ROBOTICS, Control, Sensing, Vision and Intelligence*. McGraw-Hill, 1987.

Chapter 8

[1] D.S. Bayard and J.T. Wen. New class of control laws for robotic manipulators: Part 2 — adaptive case. *Int. J. of Control*, 47(5):1387–1406, 1988.

[2] K. Kozłowski. *Modelling and identification in robotics*. Springer-Verlag, 1998.

[3] J.-J.E. Slotine and W. Li. On the Adaptive Control of Robot Manipulators. *The Int. J. of Robotics Research*, 6(3):49–59, 1987.

[4] M. Spong and R. Ortega. On adaptive inverse dynamics control of rigid robots. *IEEE Trans. Autom. Contr.*, 35(1):92–95, 1990.

Chapter 9

[1] D.S. Bayard and J.T. Wen. New class of control laws for robotic manipulators: Part 2 — adaptive case. *Int. J. of Control*, 47(5):1387–1406, 1988.

[2] J.-J. E. Slotine and W. Li. Composite adaptive control of robot manipulators. *Automatica*, 25(4):509–519, 1989.

[3] J.T. Wen and D.S. Bayard. New class of control laws for robotic manipulators: Part 1 — non-adaptive case. *Int. J. of Control*, 47(5):1361–1386, 1988.

[4] C.M. Wronka and M.W. Dunnigan. Derivation and analysis of the full dynamic model for a Mitsubishi PA10-6CE manipulator. In *Methods and Models in Automation and Robotics*. IEEE, SEP 2005.

[5] C.M. Wronka, A. Uhlmann, M.W. Dunnigan, and A.M.S. Zalzala. Fast robust controllers for a Mitsubishi PA-10 manipulator developed under QNX Neutrino. In *ISR2004*, MAR 2004.

Appendix A

Relevant work published by the author

- C.M. Wronka, A. Uhlmann, M.W. Dunnigan, and A.M.S. Zalzala. Fast robust controllers for a Mitsubishi PA-10 manipulator developed under QNX neutrino. In *ISR2004*, Mar 2004.

  Abstract: The new robotics facility at Heriot–Watt University in Edinburgh has been established to test control algorithms for cooperating manipulators in the presence of environmental disturbances (e.g. the manipulator base not fixed). There are real-world applications where the manipulator base is not fixed and is effectively "floating". These applications include sub-sea (the vehicle cannot attach itself to a suitable structure to stabilise), land (mobile robots for mining, repair, etc.), and space (satellite repair and creation of new space structures). The system comprises two manipulator arms i.e. a PUMA 560 and a MITSUBISHI PA–10 and a CueSim 6-DOF motion platform, together with their controllers and two PCs to provide communication with the user and to house all interface boards for controlling the robots.

  The control system is running on QNX Neutrino, a real–time OS, that assures all the commands are issued on time. Due to its POSIX compliance, it allows fast development of new software using existing templates and standard programming languages (ANSI C, C++). Each robot is controlled in a different way, but the software developed has an open structure to enable the user to communicate to all the machines in a similar manner. The hardware realisations vary significantly. The PA-10 is controlled with torque commands via an ARCnet interface, the PUMA is fed with torque values via a MultiQ I/O board, and the motion platform has its own control system running on a separate DSP board and can only be controlled by giving the desired positions. However, the software separates the operator from the hardware with an abstract structure to represent any 6-DOF robot.

  A control system has already been developed for the MITSUBISHI arm and in this article a comparison of several robust control algorithms is presented (experimental results together with a brief presentation of the controller structure)

– standard PD and PID controllers,

– λ-tracking (adaptive PD) controller,

– sliding control algorithm.

This choice of algorithms, which do not require any knowledge about the model of the arm, has been made as it is often difficult/impractical to obtain the inertial parameters accurately and the payload is variable.
By combining the hard real–time operating system (producing little overhead) with fast robust control methods one can obtain good control properties without excessive computational demand.
We are currently at the stage of expanding the system by adding a PUMA560 dynamic model based controller and an interface (driver) for the motion platform.

- C.M. Wronka and M.W. Dunnigan. Derivation and analysis of the full dynamic model for a Mitsubishi PA10-6CE manipulator. In *Methods and Models in Automation and Robotics*. IEEE, Sep 2005.

  Abstract: The open controller architecture of the PA10-6CE manipulator from Mitsubishi Heavy Industries allows the user to perform torque control and derive and test different control algorithms and manipulator models. Due to the inherent difficulty of the task there have been very few attempts to model the dynamics of the arm. Neither of these attempts seem to accurately model the behaviour of the 6CE version of the arm, therefore there is a need to develop an accurate and consistent model. This paper presents a full set of the dynamic parameters based on certain assumption regarding the construction of the manipulator and identification of the friction torques for each joint.

- M.W. Dunnigan and C.M. Wronka. Internet remote control interface for a multipurpose robotic arm. *The International Journal Of Advanced Robotic Systems*, 3:179–182, Jun 2006.

  Abstract: This paper presents an Internet remote control interface for a MITSUBISHI PA10-6CE manipulator established for the purpose of the ROBOT museum exhibition during spring and summer 2004. The robotic manipulator is a part of the Intelligent Robotic Systems Laboratory at Heriot – Watt University, which has been established to work on dynamic and kinematic aspects of manipulator control in the presence of environmental disturbances. The laboratory has been enriched by a simple vision system consisting of three web-cameras to broadcast the live images of the robots over the Internet.

  The Interface comprises of the TCP/IP server providing command parsing and execution using the open controller architecture of the manipulator and a client Java applet web-site providing a simple robot control interface.

- M.W. Dunnigan and C.M. Wronka. Derivation and verification of the dynamic model for the PA10-6CE robotic manipulator. submitted to International Journal on Modelling, Identification and Control, Mar 2010.

  Abstract: This paper presents a novel dynamic model derivation for the PA10-6CE manipulator. The model consists of: gravity term based on manufacturer specifications, non-parametric friction term estimated experimentally, inertia, Coriolis and centrifugal terms computed based on simplified geometric link models and motor inertia estimated experimentally. The incorporation of the model terms is analysed with a comparison of simulated and experimental closed-loop control tasks. Model based controller performance is evaluated and analysed in terms of complexity versus tracking error performance.

- M.W. Dunnigan and C.M. Wronka. Derivation and analysis of a dynamic model of a robotic manipulator on a moving base. submitted to Control Engineering Practice, Apr 2010.

  Abstract: A dynamic model of a robotic manipulator mounted on a moving base is derived using the Euler-Lagrange approach. It is assumed that the base inertia is large enough not to be influenced by the manipulator motion and therefore can be treated as a time-varying parameter in the dynamic equations. The presented derivation is applied to a Mitsubishi PA10-6CE robotic manipulator mounted on a 2-DOF platform. The model is analysed by comparing simple closed-loop control results of the simulated model with experimental data from the manipulator mounted on the platform.

Appendix B

Mathematica code

Mathematica code used to generate the dynamic equations for the PA10-6CE on a moving platform.

B.1 Tools module

```
BeginPackage["Tools‘"]
MatA::usage = "MatA[theta_,d_,a_,alpha_] returns the Dennavit Hartenberg
matrix for a joint from parameters theta_, d_, a_, and alpha_ (see Spong)"

RotM::usage = "RotM[phi_,chi_,psi_] returns a Denavit Hartenberg rotation
matrix, where phi_ is around x, chi_ around y, and psi_ around z"

TransM::usage = "TransM[x_,y_,z_] returns a Denavit Hartenberg rotation matrix,
where phi_ is around x, chi_ around y, and psi_ around z"

DenHart::usage = "DenHart[matK_,n_] returns the DH kinematics matrix
up to joint n"

DenHartmn::usage = "DenHart[matK_,m_,n_] returns the DH kinematics matrix
from joint m up to joint n"

PotentialEn::usage = "PotentialEn[mVect_,gVect_,matK_,rVect_] "

MatUij::usage = "MatUij[matK_,i_,j_] returns the U_ij matrix
for dynamics model (see Fu, Lee, Gonzales)"

GravityVector::usage = "GravityVector[potential_,vars_] "

Begin["‘Private‘"]

RotM[phi_,chi_,psi_]:=Block[{matX,matY,matZ,matrix},
 matX={{1, 0, 0, 0}, {0, Cos[phi], -Sin[phi],0},
       { 0, Sin[phi], Cos[phi],0}, {0,0,0,1}};
 matY={{Cos[chi], 0, Sin[chi],0}, {0, 1, 0,0},
       { -Sin[chi], 0, Cos[chi],0}, {0,0,0,1}};
 matZ={{Cos[psi], -Sin[psi], 0,0}, {Sin[psi], Cos[psi], 0,0},
```

```
      {0, 0, 1,0}, {0,0,0,1}};
 matrix=matX.matY.matZ;
matrix]

TransM[x_,y_,z_]:=Block[{matrix},
 matrix={{1, 0, 0, x},{0, 1, 0, y},{0, 0, 1, z},{0, 0, 0, 1}};
matrix]

DenHart[matK_,n_]:=Block[{i,matA},
matA=TransM[0,0,0];
For[i=1,i<=n,i++,matA=matA.RotM[0,0,matK[[i,1]]].
                     TransM[0,0,matK[[i,2]]].
                     TransM[matK[[i,3]],0,0].
                     RotM[matK[[i,4]],0,0]];
matA]

DenHartmn[matK_,m_,n_]:=Block[{i,matA},
matA=TransM[0,0,0];
For[i=m+1,i<=n,i++,matA=FullSimplify[TrigReduce[matA.
                       RotM[0,0,matK[[i,1]]].
                       TransM[0,0,matK[[i,2]]].
                       TransM[matK[[i,3]],0,0].
                       RotM[matK[[i,4]],0,0]]]];
matA]

MatUij[matK_, ui_, uj_] := Block[{matA1, matA2, matQ, matU},
  matQ = {{0, -1, 0, 0}, {1, 0, 0, 0},
          {0, 0, 0, 0}, {0, 0, 0, 0}};
  If[uj <= ui, matA1 = DenHartmn[matK, 0, uj - 1];
               matA2 = DenHartmn[matK, uj - 1, ui];
     matU = matA1 . matQ . matA2;,
     matU = {{0, 0, 0, 0}, {0, 0, 0, 0},
             {0, 0, 0, 0}, {0, 0, 0, 0}};];
  matU]


End[]
Protect[RotM, TransM, DenHart, DenHartmn, PotentialEn,
        MatUij, GravityVector]
EndPackage[]
```

B.2 Main script

```
--- load tools ---

«Tools.m

--- define the kinematics D-H transformations ---

matK={{Pi+q1,d_1,0,Pi/2},{q2+Pi/2,0,a_2,0},
    {q3+Pi/2,0,0,Pi/2},{Pi+q4,d_4,0,Pi/2},
    {Pi+q5,0,0,Pi/2},{q6,d_6,0,0}};
```

```
--- define inertias ---
```

$$\mathrm{mJ}_1 = \{\{(-\mathrm{mI}_{xx1} + mI_{yy1} + mI_{zz1})/2, 0, 0, \mathrm{mm}_1\mathrm{mX}_1\},$$
$$\{0, (\mathrm{mI}_{xx1} - mI_{yy1} + mI_{zz1})/2, 0, \mathrm{mm}_1\mathrm{mY}_1\},$$
$$\{0, 0, (\mathrm{mI}_{xx1} + mI_{yy1} - mI_{zz1})/2, \mathrm{mm}_1\mathrm{mZ}_1\},$$
$$\{\mathrm{mm}_1\mathrm{mX}_1, \mathrm{mm}_1\mathrm{mY}_1, \mathrm{mm}_1\mathrm{mZ}_1, \mathrm{mm}_1\}\};$$

$$\mathrm{mJ}_2 = \{\{(-\mathrm{mI}_{xx2} + mI_{yy2} + mI_{zz2})/2, \mathrm{mI}_{xy2}, 0, \mathrm{mm}_2\mathrm{mX}_2\},$$
$$\{\mathrm{mI}_{xy2}, (\mathrm{mI}_{xx2} - mI_{yy2} + mI_{zz2})/2, 0, \mathrm{mm}_2\mathrm{mY}_2\},$$
$$\{0, 0, (\mathrm{mI}_{xx2} + mI_{yy2} - mI_{zz2})/2, \mathrm{mm}_2\mathrm{mZ}_2\},$$
$$\{\mathrm{mm}_2\mathrm{mX}_2, \mathrm{mm}_2\mathrm{mY}_2, \mathrm{mm}_2\mathrm{mZ}_2, \mathrm{mm}_2\}\};$$

$$\mathrm{mJ}_3 = \{\{(-\mathrm{mI}_{xx3} + mI_{yy3} + mI_{zz3})/2, 0, 0, \mathrm{mm}_3\mathrm{mX}_3\},$$
$$\{0, (\mathrm{mI}_{xx3} - mI_{yy3} + mI_{zz3})/2, 0, \mathrm{mm}_3\mathrm{mY}_3\},$$
$$\{0, 0, (\mathrm{mI}_{xx3} + mI_{yy3} - mI_{zz3})/2, \mathrm{mm}_3\mathrm{mZ}_3\},$$
$$\{\mathrm{mm}_3\mathrm{mX}_3, \mathrm{mm}_3\mathrm{mY}_3, \mathrm{mm}_3\mathrm{mZ}_3, \mathrm{mm}_3\}\};$$

$$\mathrm{mJ}_4 = \{\{(-\mathrm{mI}_{xx4} + mI_{yy4} + mI_{zz4})/2, 0, 0, \mathrm{mm}_4\mathrm{mX}_4\},$$
$$\{0, (\mathrm{mI}_{xx4} - mI_{yy4} + mI_{zz4})/2, 0, \mathrm{mm}_4\mathrm{mY}_4\},$$
$$\{0, 0, (\mathrm{mI}_{xx4} + mI_{yy4} - mI_{zz4})/2, \mathrm{mm}_4\mathrm{mZ}_4\},$$
$$\{\mathrm{mm}_4\mathrm{mX}_4, \mathrm{mm}_4\mathrm{mY}_4, \mathrm{mm}_4\mathrm{mZ}_4, \mathrm{mm}_4\}\};$$

$$\mathrm{mJ}_5 = \{\{(-\mathrm{mI}_{xx5} + mI_{yy5} + mI_{zz5})/2, 0, 0, \mathrm{mm}_5\mathrm{mX}_5\},$$
$$\{0, (\mathrm{mI}_{xx5} - mI_{yy5} + mI_{zz5})/2, 0, \mathrm{mm}_5\mathrm{mY}_5\},$$
$$\{0, 0, (\mathrm{mI}_{xx5} + mI_{yy5} - mI_{zz5})/2, \mathrm{mm}_5\mathrm{mZ}_5\},$$
$$\{\mathrm{mm}_5\mathrm{mX}_5, \mathrm{mm}_5\mathrm{mY}_5, \mathrm{mm}_5\mathrm{mZ}_5, \mathrm{mm}_5\}\};$$

$$\mathrm{mJ}_6 = \{\{(-\mathrm{mI}_{xx6} + mI_{yy6} + mI_{zz6})/2, 0, 0, \mathrm{mm}_6\mathrm{mX}_6\},$$
$$\{0, (\mathrm{mI}_{xx6} - mI_{yy6} + mI_{zz6})/2, 0, \mathrm{mm}_6\mathrm{mY}_6\},$$
$$\{0, 0, (\mathrm{mI}_{xx6} + mI_{yy6} - mI_{zz6})/2, \mathrm{mm}_6\mathrm{mZ}_6\},$$
$$\{\mathrm{mm}_6\mathrm{mX}_6, \mathrm{mm}_6\mathrm{mY}_6, \mathrm{mm}_6\mathrm{mZ}_6, \mathrm{mm}_6\}\};$$

```
--- define platform related transformations ---
```

mAf=RotM[0,pPitch[t],0].RotM[pRoll[t],0,0].TransM[rX,0,0];

mAp=mAf/.{pPitch[t]$\to$ pitch, pRoll[t] $\to$ roll,
D[pPitch[t],t] $\to$ dPitch, D[pRoll[t],t] $\to$ dRoll,
D[pPitch[t],{t,2}] $\to$ ddPitch, D[pRoll[t],{t,2}] $\to$ ddRoll};

dAp=FullSimplify[D[mAf,t]]/.{pPitch[t]$\to$ pitch, pRoll[t] $\to$ roll,
D[pPitch[t],t] $\to$ dPitch, D[pRoll[t],t] $\to$ dRoll,
D[pPitch[t],{t,2}] $\to$ ddPitch, D[pRoll[t],{t,2}] $\to$ ddRoll};
ddAp=FullSimplify[D[mAf,{t,2}]]/.{pPitch[t]$\to$ pitch, pRoll[t] $\to$ roll,
D[pPitch[t],t] $\to$ dPitch, D[pRoll[t],t] $\to$ dRoll,
D[pPitch[t],{t,2}] $\to$ ddPitch, D[pRoll[t],{t,2}] $\to$ ddRoll};
J={$\mathrm{mJ}_1, \mathrm{mJ}_2, \mathrm{mJ}_3, \mathrm{mJ}_4, \mathrm{mJ}_5, \mathrm{mJ}_6$}/.{$\mathrm{mX}_6 \to 0, \mathrm{mY}_6 \to 0,$
$\mathrm{mX}_5 \to 0, \mathrm{mY}_5 \to 0, \mathrm{mX}_4 \to 0, \mathrm{mZ}_4 \to 0,$
$\mathrm{mX}_3 \to 0, \mathrm{mY}_3 \to 0, \mathrm{mZ}_2 \to 0, \mathrm{mX}_1 \to 0, \mathrm{mZ}_1 \to 0$};

```
--- function to calculate second derivatives of the D-H transformation
over time ---
```

```
MatUijk[matK_,ui_,uj_,uk_]:=Block[{matA1,matA2,matA3,matQ,matU},
    matQ = {{0,-1,0,0},{1,0,0,0},{0,0,0,0},{0,0,0,0}};
    If[And[uj ≤ ui, uk ≤ uj],
    matA1 = DenHartmn[matK, 0, uk - 1];
    matA2 = DenHartmn[matK, uk - 1, uj - 1];
    matA3 = DenHartmn[matK, uj - 1, ui];
    matU = matA1.matQ.matA2.matQ.matA3;,
    If[And[uk ≤ ui, uj ≤ uk],
    matA1 = DenHartmn[matK, 0, uj - 1];
    matA2 = DenHartmn[matK, uj - 1, uk - 1];
    matA3 = DenHartmn[matK, uk - 1, ui];
    matU = matA1.matQ.matA2.matQ.matA3;,
    matU = {{0,0,0,0},{0,0,0,0},{0,0,0,0},{0,0,0,0}};]];matU]


--- function to compute model items ---

VelocityItemPlat[i_,j_]:=Simplify[
    Sum[Tr[Simplify[dAp.MatUij[matK, k, j]].J[[k]].
    Simplify[Transpose[MatUij[matK, k, i]].Transpose[mAp]]], {k, i, 6}]]

FreeItemPlat[i_]:=Simplify[
    Sum[Tr[Simplify[ddAp.DenHartmn[matK, 0, j]].J[[j]].
    Simplify[Transpose[MatUij[matK, j, i]].Transpose[mAp]]], {j, i, 6}]]

InertiaPlat[i_,j_]:=Simplify[
    Sum[Tr[mAp.MatUij[matK, k, j].J[[k]].
    Transpose[MatUij[matK, k, i]].Transpose[mAp]], {k, i, 6}]]

CoriolisCentPlat[i_,j_,k_]:=Simplify[
    Sum[Tr[mAp.MatUijk[matK, m, j, k].J[[m]].
    Transpose[MatUij[matK, m, i]].Transpose[mAp]], {m, i, 6}]]


--- function preparing precalculated constants for output form ---

cStyleReplacedElem[e_]:=Block[{newE},
    newE = Simplify[e]/.{Cos[q1] → c1, Sin[q1] → s1,
    Cos[q2] → c2, Sin[q2] → s2, Cos[2q2] → c22,
    Sin[2q2] → s22, Cos[q2 + q3] → c23, Sin[q2 + q3] → s23,
    Cos[2(q2 + q3)] → c2233, Sin[2(q2 + q3)] → s2233,
    Cos[2q2 + q3] → c223, Sin[2q2 + q3] → s223,
    Cos[q3] → c3, Sin[q3] → s3, Cos[q4] → c4, Sin[q4] → s4,
    Cos[2q4] → c44, Sin[2q4] → s44, Cos[q5] → c5,
    Sin[q5] → s5, Cos[2q5] → c55, Sin[2q5] → s55,
    Cos[q6] → c6, Cos[2q6] → c66, Sin[q6] → s6,
    Sin[2q6] → s66, mI_xx1 → mIxx1, mI_yy1 → mIyy1,
    mI_zz1 → mIzz1, mI_xx2 → mIxx2, mI_yy2 → mIyy2,
    mI_zz2 → mIzz2, mI_xx3 → mIxx3, mI_yy3 → mIyy3, mI_zz3 → mIzz3,
    mI_xx4 → mIxx4, mI_yy4 → mIyy4, mI_zz4 → mIzz4,
    mI_xx5 → mIxx5, mI_yy5 → mIyy5, mI_zz5 → mIzz5,
    mI_xx6 → mIxx6, mI_yy6 → mIyy6, mI_zz6 → mIzz6, mI_xy2 → mIxy2,
    mm_1 → mm1, mm_2 → mm2, mm_3 → mm3, mm_4 → mm4, mm_5 → mm5,
    mm_6 → mm6, mX_1 → mX1, mY_1 → mY1, mZ_1 → mZ1, mX_2 → mX2,
```

```
    mY_2 → mY2, mZ_2 → mZ2, mX_3 → mX3,
    mY_3 → mY3, mZ_3 → mZ3, mX_4 → mX4, mY_4 → mY4,
    mZ_4 → mZ4, mX_5 → mX5, mY_5 → mY5, mZ_5 → mZ5,
    mX_6 → mX6, mY_6 → mY6, mZ_6 → mZ6, d_1->d1,
    a_2 → a2, d_4 → d4, d_6 → d6, Sin[roll] → sroll,
    Cos[roll] → croll, Sin[pitch] → spitch, Cos[pitch] → cpitch}; newE]

--- functions printing results to files ---

PrintSubList[list_,strm_,sName_]:=Block[
    {dim, i, newSName}, dim = Dimensions[list];
    If[Length[dim] > 1, For[i = 1, i<= dim[[1]], i++,
    newSName = sName<>ToString[i]<>",";
    PrintSubList[list[[i]], strm, newSName]],
    For[i = 1, i<= dim[[1]], i++,
    newSName = sName<>ToString[i]<>")=";
    WriteString[strm, newSName, CForm[
    cStyleReplacedElem[list[[i]]]], ";\n\n"]]]]

PrintList[list_,fName_,sName_]:=Block[
    {strm}, strm = OpenWrite[fName];
    WriteString[strm, "% ***** Matlab:\n\n"];
    PrintSubList[list, strm, sName<>"("];
    Close[strm];]

PrintFunc3D[Func_,range_,fName_,sName_]:=Block[{i,j,strm},
    strm = OpenWrite[fName];
    WriteString[strm, "% ***** Matlab:\n\n"];
    For[i = range[[1,1]], i ≤ range[[1,2]], i++,
    For[j = range[[2,1]], j ≤ range[[2,2]], j++, For[k = range[[3,1]], k ≤ range[[3,2]], k++,
    WriteString[strm, sName, "(", i, ",", j, ",", k, ")=",
    CForm[cStyleReplacedElem[Func[i, j, k]]], ";\n\n"]]];
    Close[strm];]

PrintFunc2D[Func_,range_,fName_,sName_]:=Block[{i,j,strm},
    strm = OpenWrite[fName];
    WriteString[strm, "% ***** Matlab:\n\n"];
    For[i = range[[1,1]], i ≤ range[[1,2]], i++,
    For[j = range[[2,1]], j ≤ range[[2,2]], j++, WriteString[strm, sName, "(", i, ",", j, ")=",
    CForm[cStyleReplacedElem[Func[i, j]]], ";\n\n"]];
    Close[strm];]
PrintFunc1D[Func_,range_,fName_,sName_]:=Block[{i,j,strm},
    strm = OpenWrite[fName];
    WriteString[strm, "% ***** Matlab:\n\n"];
    For[i = range[[1,1]], i ≤ range[[1,2]], i++,
    WriteString[strm, sName, "(", i, ")=",      CForm[cStyleReplacedElem[Func[i]]], ";\n\n"];
    Close[strm];]

--- calculate and save terms ---

PrintFunc2D[VelocityItemPlat,{{1,6},{1,6}},"plat_velocity_item.m","mPV"]
PrintFunc2D[InertiaPlat,{{1,6},{1,6}},"plat_inertia_item.m","mQ"]
PrintFunc3D[CoriolisCentPlat,{{1,6},{1,6},{1,6}},"plat_coriolis_item.m","mC"]
PrintFunc1D[FreeItemPlat,{{1,6}},"plat_free_item.m","mPF"]
```

Appendix C

Matlab code

Elements of Matlab code used for PA10 dynamic model simulation

C.1 Solver function

```
function [t,x,u]=ode_cw(dyn_fcn,ts,x0,dyn_params)

MAX_ABS_TOL=.1;

MIN_SUB_STEPS=2;
MAX_SUB_STEPS=128;

state=x0;

t(1)=ts(1);
local_t=t(1);
x(1,:)=x0';
u(1,:)=zeros(1,6);

last_sub_steps=MIN_SUB_STEPS;

for i=2:length(ts)

    SUB_STEPS=max(last_sub_steps,MIN_SUB_STEPS);
    sol_OK=0;
    dyn_step.local_t=local_t;
    dyn_step.state=state;
    while sol_OK==0
        local_t=dyn_step.local_t;
        state=dyn_step.state;
        break_done=0;
        for j=1:SUB_STEPS
            new_t=ts(i-1)+(j/SUB_STEPS)*(ts(i)-ts(i-1));
            dt=new_t-local_t;
            [dstate temp_u stop_vect]=feval(dyn_fcn,local_t,state,dyn_params);
            old_state=state;
            state=state+dstate*dt;
```

```
            if ~isempty(stop_vect)
                stop1_ind=find(stop_vect<=6);
                state(6+stop_vect(stop1_ind))=0;
                stop2_ind=find(stop_vect>6);
                for k=stop2_ind
                    % if the two velocities have different signs, then stop.
                    if (old_state(stop_vect(k))*state(stop_vect(k)))<0
                        state(stop_vect(k))=0;
                    end
                end
            end
            if max(abs(state-old_state)) > MAX_ABS_TOL
                SUB_STEPS=SUB_STEPS*2;
                if SUB_STEPS>MAX_SUB_STEPS
                    fprintf('#');
                end
                break_done=1;
                break;
            end
            local_t=new_t;
        end
        if break_done==0
            sol_OK=1;
            SUB_STEPS=SUB_STEPS/2;
        end
    end
    t(i)=local_t;
    x(i,:)=state';
    u(i,:)=temp_u';
    last_sub_steps=SUB_STEPS;
end;
```

C.2 Main dynamic structure function

```
function [dx, out_u, out_stop_joint]=dynamics_m(t,x,varargin)

global marker_t;
global marker_last_int;

MARKER_STEP=0.02;

FRICT_VEL_LIMIT=9.5e-3;

dyn_params=varargin{1};

q=x(1:6);
dq=x(7:12);

if isfield(dyn_params,'platformModel')
    plat_coord_fcn=dyn_params.platformModel{1}{1};
    plat_coord_data=dyn_params.platformModel{1}{2};
```

```
    is_p_inert=dyn_params.platformModel{1}{3};
    is_p_corcent=dyn_params.platformModel{1}{4};
    plat_coord=feval(plat_coord_fcn,t,plat_coord_data);

    [mQ mMQ vh fg mPF mPV]=pa10_i_dyn(x,plat_coord);

    gv=[0 0 -9.81585];
    % alow variable gravity vector.
    rgv=gv*rotM(0, plat_coord.pitch, 0)*rotM(plat_coord.roll, 0, 0);
    vg=fg*rgv';
else
    is_p_inert=0;
    is_p_corcent=0;

    [mQ mMQ vh fg]=pa10_i_dyn(x);

    gv=[0 0 -9.81585]';
    vg=fg*gv;
end

if isfield(dyn_params,'motorInertia')
    mMQ=diag(dyn_params.motorInertia{1});
else
    for i=1:6
        if isfield(dyn_params,strcat('motorInertia',num2str(i)))
            mMQ(i,i)=dyn_params.(strcat('motorInertia',num2str(i))){1};
        end
    end
end

if isfield(dyn_params,'frict')
    f=dyn_params.frict{1};
    ft=friction(dq,f)*50;

    if isfield(dyn_params,'frictScale')
        FRICT_SCALE=dyn_params.frictScale{1};
    else
        FRICT_SCALE=ones(6,1);
    end
    ft=ft.*FRICT_SCALE;
else
    ft=zeros(6,1);
end

if isfield(dyn_params,'transmissionStiffness')
    transmission_stiffness=1;
else
    transmission_stiffness=0;
end

u=zeros(6,1);

if isfield(dyn_params,'inputFcn')
```

```
    if length(dyn_params.inputFcn)==1
        inp_fcn=dyn_params.inputFcn{1};
        inp_fcn_params={};
    else
        inp_fcn=dyn_params.inputFcn{1};
        inp_fcn_params=dyn_params.inputFcn(2:end);
    end;

    inp_params.t=t;
    inp_params.q=q;
    inp_params.dq=dq;
    if isfield(dyn_params,'integrateError')
        inp_params.errInt=x(13:18);
    end;
    inp_params.mQ=mQ;
    inp_params.mMQ=mMQ;
    inp_params.vh=vh;
    inp_params.vg=vg;


    if isfield(dyn_params,'trajFcn')
        if ~isempty(inp_fcn_params)
            u=feval(inp_fcn,inp_params,inp_fcn_params,dyn_params.trajFcn{1});
        else
            u=feval(inp_fcn,inp_params,dyn_params.trajFcn{1});
        end
    else
        u=feval(inp_fcn,inp_params,inp_fcn_params);
    end
else
    u=zeros(6,1);
end

if isfield(dyn_params,'integrateError')
    if length(u)>6
        err_delta=u(7:12);
        u=u(1:6);
    else
        err_delta=zeros(6,1);
    end
end

dx(1:6,1)=reshape(x(7:12),6,1);

internal_part=-vh-vg+u; %-ft;

if isfield(dyn_params,'noGravity')
    internal_part=internal_part+vg;
end
if isfield(dyn_params,'noCoriolis')
    internal_part=internal_part+vh;
end
```

```
if is_p_inert~=0
    internal_part=internal_part-mPF;
end

if is_p_corcent~=0
    plat_vel=mPV*dq;
    internal_part=internal_part-2*plat_vel;
end

stop_joint=[];
stop_joint2=[];

if isfield(dyn_params,'frict')
    for i=1:6
        if dq(i)>0
            if internal_part(i)==0 && ft(i)>0
                stop_joint2(end+1)=i;
            else
                if dq(i)<f.p{i}(1,1)
                    if (internal_part(i)>0) && internal_part(i)<ft(i)
                        stop_joint(end+1)=i;
                    end
                end
            end
        else
            if dq(i)<0
                if internal_part(i)==0 && ft(i)<0
                    stop_joint2(end+1)=i;
                else
                    if dq(i)>f.n{i}(1,1)
                        if (internal_part(i)<0) && internal_part(i)>ft(i)
                            stop_joint(end+1)=i;
                        end
                    end
                end
            end
        end
    end

    dx(stop_joint)=0;

    full_internal_part=internal_part-ft;

    full_internal_part(stop_joint)=0;

else
    full_internal_part=internal_part;
end

if nargout>2
    out_stop_joint=[stop_joint stop_joint2+6];
end
```

```
if isfield(dyn_params,'lockedJoints')
    locked_joints=dyn_params.lockedJoints{1};
    full_internal_part(locked_joints,1)=0;
end

if isfield(dyn_params,'onlyMotorInertia')
    dx(7:12,1)=(((mMQ)^-1)*(full_internal_part));
elseif isfield(dyn_params,'onlyJointInertia')
    dx(7:12,1)=(((mQ)^-1)*(full_internal_part));
elseif isfield(dyn_params,'unityInertiaMatrix')
    dx(7:12,1)=full_internal_part;
else
    dx(7:12,1)=(((mQ+mMQ)^-1)*(full_internal_part));
end
if isfield(dyn_params,'integrateError')
    dx(13:18)=err_delta;
end

PA10_RatedVelocity=[1.64 1.705 2.24 9.63 9.63 9.63]';

ind=find((x(7:12)>PA10_RatedVelocity).*(dx(7:12)>0));
if ~isempty(ind)
    dx(6+ind)=0;
    if isfield(dyn_params,'integrateError')
        dx(12+ind)=0;
    end
    u(ind)=vh(ind)+vg(ind)+ft(ind);

end

ind=find((x(7:12)<-PA10_RatedVelocity).*(dx(7:12)<0));
if ~isempty(ind)
    dx(6+ind)=0;
    if isfield(dyn_params,'integrateError')
        dx(12+ind)=0;
    end
    u(ind)=vh(ind)+vg(ind)+ft(ind);
end

if nargout>1
    out_u=u;
end;


if isfield(dyn_params,'lockedJoints')
    locked_joints=dyn_params.lockedJoints{1};
    dx(locked_joints,1)=0;
    dx(6+locked_joints,1)=0;
    if isfield(dyn_params,'integrateError')
        dx(12+locked_joints,1)=0;
    end
end
```

```
if (t==0.002)
    marker_t=0;
    marker_last_int=0;
end
if (floor(t)>marker_last_int)
    fprintf('\n');
    marker_last_int=floor(t);
end;
if (t>(marker_t+MARKER_STEP))
    fprintf('*');
    marker_t=t;
end
```

Appendix D

C code

Elements of C code used in the PA10 control application.

```
// ctrl_all_controllers.cpp
if (g.cntrl_change>0) {
    if (((controller_type&CRTL_MASK_CMD_TYPE)==CRTL_MASK_CMD_VELOCITY)
        &&((g.cntrl_change&CRTL_MASK_CMD_TYPE)==CRTL_MASK_CMD_TORQUE)) {
        pPA10->ToTorqueControl();
        vel_trq_ctrl=1;
        fprintf(stdout,"Switching to torque control\n");
    } else if (((controller_type&CRTL_MASK_CMD_TYPE)==CRTL_MASK_CMD_TORQUE)
           &&((g.cntrl_change&CRTL_MASK_CMD_TYPE)==CRTL_MASK_CMD_VELOCITY)) {
            pPA10->ToVelocityControl();
            vel_trq_ctrl=0;
            fprintf(stdout,"Switching to velocity control\n");
    }
    fprintf(stdout,"switching controller from %08x to %08x\n",
              controller_type,g.cntrl_change);
    controller_type=g.cntrl_change;
    g.cntrl_change=0;
}

g.pRobots[robotNum]->BeginCalculating();

// filter the velocity.
for (Loop = 0; Loop < NUM_JOINTS; Loop++) {
    debug_val[Loop]=0;
    if (filter_len>0) {
        double temp_buf[20];

        vel_buff[Loop][count % filter_len]=pV[Loop];

        for(Loop2=0; Loop2<filter_len; Loop2++) {
            temp_buf[Loop2]=vel_buff[Loop][Loop2];
        }

        switch (filter_type) {
        case 0:
```

```
            cur_vel[Loop]=mean(temp_buf,0,filter_len);
            break;
        case 1:
            cur_vel[Loop]=median(temp_buf,0,filter_len);
            break;
        case 2:
            cur_vel[Loop]=(mean(temp_buf,0,filter_len) +
                       median(temp_buf,0,filter_len))/2.0;
            break;
        default:
            cur_vel[Loop]=pV[Loop];
            break;
        }

    } else {
        cur_vel[Loop]=pV[Loop];
    }
}

// populate the gravity vector and set flag
if (!g.nosensors) {
if ((controller_type&CRTL_MASK_GRAVITY_TYPE)>0) {
    dynamics.type=PA10_DYN_GRAVITY_VECTOR;
    if ((controller_type&CRTL_MASK_GRAVITY_TYPE)==CRTL_MASK_GRAVITY_S_ACC) {
        // take gravity vector from acceleration sensors
        dynamics.gravity_vector[0]=sens_p.ax;
        dynamics.gravity_vector[1]=sens_p.ay;
        dynamics.gravity_vector[2]=sens_p.az;
    } else if ((controller_type&CRTL_MASK_GRAVITY_TYPE)==CRTL_MASK_GRAVITY_S_RP) {
        // take ravity vector from roll+pitch sensors
        dynamics.gravity_vector[0]=roll_pitch_grav[0];
        dynamics.gravity_vector[1]=roll_pitch_grav[1];
        dynamics.gravity_vector[2]=roll_pitch_grav[2];
    }
}
}

if ((controller_type&CRTL_MASK_DYN_TYPE)>0) {
    if ((controller_type&CRTL_MASK_DYN_TYPE)==CRTL_MASK_DYN_REFERENCE) {
        for (Loop = 1; Loop <= NUM_JOINTS; Loop++) {
            dynamics.pA[Loop-1]=pos[Loop-1][count];
            dynamics.pV[Loop-1]=vel[Loop-1][count];
            dynamics.pV2[Loop-1]=vel[Loop-1][count];
            dynamics.pX[Loop-1]=acc[Loop-1][count];
        }
    } else if ((controller_type&CRTL_MASK_DYN_TYPE)==CRTL_MASK_DYN_SLIDING) {
        for (Loop = 1; Loop <= NUM_JOINTS; Loop++) {
#ifndef CTRL_ALL_COPY_pA
            dynamics.pA[Loop-1]=pA[Loop-1];
#else //CTRL_ALL_COPY_pA
            dynamics.pA[Loop-1]=pa10_angle[Loop-1];
#endif //CTRL_ALL_COPY_pA
            dynamics.pV[Loop-1]=cur_vel[Loop-1];
```

```
            dynamics.pX[Loop-1]=acc[Loop-1][count]-lambda[Loop-1]*
                        (pV[Loop-1]-vel[Loop-1][count]);
#ifndef CTRL_ALL_COPY_pA
            dynamics.pV2[Loop-1]=vel[Loop-1][count]-lambda[Loop-1]*
                                (pA[Loop-1]-pos[Loop-1][count]);
#else //CTRL_ALL_COPY_pA
            dynamics.pV2[Loop-1]=vel[Loop-1][count]-lambda[Loop-1]*
                         (pa10_angle[Loop-1]-pos[Loop-1][count]);
#endif //CTRL_ALL_COPY_pA
        }
    }
} else {
    // direct model
    for (Loop = 1; Loop <= NUM_JOINTS; Loop++) {
#ifndef CTRL_ALL_COPY_pA
        dynamics.pA[Loop-1]=pA[Loop-1];
#else //CTRL_ALL_COPY_pA
        dynamics.pA[Loop-1]=pa10_angle[Loop-1];
#endif //CTRL_ALL_COPY_pA
        dynamics.pV[Loop-1]=cur_vel[Loop-1];
        dynamics.pV2[Loop-1]=cur_vel[Loop-1];
        dynamics.pX[Loop-1]=0; //there are no acceleration estimates
                               //for direct model.
        // for model based controllers:
        // the filtered velocities are only used in model calculation,
        // and the remainder feedback element uses the direct last measurement.
    }
}

// calculate dynamics
pa10Dynamics(&dynamics);

for (Loop = 0; Loop < NUM_JOINTS; Loop++) {
    // reset velocity to raw (unfiltered)
    cur_vel[Loop]=pV[Loop];
    // reset the temporary control values
    newTorque[Loop]=0.0;
    c_error[Loop]=0.0;
}

saturated=0;

double tmp_pos[NUM_JOINTS];
double tmp_vel[NUM_JOINTS];
double tmp_acc[NUM_JOINTS];

if ((controller_type&CRTL_MASK_SPC_TYPE)>0) {
    if ((controller_type&CRTL_MASK_SPC_TYPE)==CRTL_MASK_SPC_DTZ) {
        for (Loop = 1; Loop <= NUM_JOINTS; Loop++) {
            tmp_pos[Loop-1]=0;
            tmp_vel[Loop-1]=0;
            tmp_acc[Loop-1]=0;
        }
```

```
    }
} else {
    for (Loop = 1; Loop <= NUM_JOINTS; Loop++) {
        tmp_pos[Loop-1]=pos[Loop-1][count];
        tmp_vel[Loop-1]=vel[Loop-1][count];
        tmp_acc[Loop-1]=acc[Loop-1][count];
    }
}

if ((controller_type&CTRL_MASK_ERROR_C_TYPE)>0) {
    if ((controller_type&CTRL_MASK_ERROR_PD)==CTRL_MASK_ERROR_PD) {
        for (Loop = 1; Loop <= NUM_JOINTS; Loop++) {
            c_error[Loop-1]=-kp[Loop-1]*(pA[Loop-1]-tmp_pos[Loop-1])
                    -kd[Loop-1]*(cur_vel[Loop-1]-tmp_vel[Loop-1]);
        }
    }
    if ((controller_type&CTRL_MASK_ERROR_SAT_SLIDING)==
                        CTRL_MASK_ERROR_SAT_SLIDING) {
        for (Loop = 1; Loop <= NUM_JOINTS; Loop++) {
            c_error[Loop-1]=(cur_vel[Loop-1]-tmp_vel[Loop-1])
                    +lambda[Loop-1]*(pA[Loop-1]-tmp_pos[Loop-1]);
            c_error[Loop-1]=-K_m[Loop-1]*sat(c_error[Loop-1]/varphi[Loop-1]);
            if (abs(sat(c_error[Loop-1]/varphi[Loop-1]))==1) {
                saturated=1;
            }
        }
    }
    if ((controller_type&CTRL_MASK_ERROR_SLIDING)==CTRL_MASK_ERROR_SLIDING) {
        for (Loop = 1; Loop <= NUM_JOINTS; Loop++) {
            c_error[Loop-1]=(cur_vel[Loop-1]-tmp_vel[Loop-1])
                    +lambda[Loop-1]*(pA[Loop-1]-tmp_pos[Loop-1]);
            c_error[Loop-1]=-kd[Loop-1]*c_error[Loop-1];
        }
    }
    if ((controller_type&CTRL_MASK_ERROR_I)==CTRL_MASK_ERROR_I) {
        for (Loop = 1; Loop <= NUM_JOINTS; Loop++) {
            c_error[Loop-1]+=-ki[Loop-1]*int_err[Loop-1];
        }
    }
}

if ((controller_type&CRTL_MASK_MODEL_TYPE)>0) {
    if ((controller_type&CTRL_MASK_MODEL_GRAVITY)==CTRL_MASK_MODEL_GRAVITY) {
        for (Loop = 1; Loop <= NUM_JOINTS; Loop++) {
            newTorque[Loop-1]+=dynamics.gravity[Loop-1]/50.0;
        }
    }
    if ((controller_type&CTRL_MASK_MODEL_FRICTION)==CTRL_MASK_MODEL_FRICTION) {
        for (Loop = 1; Loop <= NUM_JOINTS; Loop++) {
            newTorque[Loop-1]+=dynamics.frict[Loop-1];
        }
    }
    if ((controller_type&CTRL_MASK_MODEL_COR_CENT)==CTRL_MASK_MODEL_COR_CENT) {
```

```
        for (Loop = 1; Loop <= NUM_JOINTS; Loop++) {
            newTorque[Loop-1]+=dynamics.cV[Loop-1]/50.0;
        }
    }

    if ((controller_type&CTRL_MASK_MODEL_FEEDBACK_LINEARISATION)==
                            CTRL_MASK_MODEL_FEEDBACK_LINEARISATION) {
        for (Loop = 1; Loop <= NUM_JOINTS; Loop++) {
            c_error[Loop-1]=(acc[Loop-1][count])
                -kdl[Loop-1]*(cur_vel[Loop-1]-tmp_vel[Loop-1])
                -kpl[Loop-1]*(pA[Loop-1]-tmp_pos[Loop-1]);
        }
        for (Loop = 1; Loop <= NUM_JOINTS; Loop++) {
            newTorque[Loop-1]+=matrix_by_vector(dynamics.mQ,c_error,
                                                Loop-1)/50.0;
        }
    } else if ((controller_type&CTRL_MASK_MODEL_INERTIA)==
                               CTRL_MASK_MODEL_INERTIA) {
        for (Loop = 1; Loop <= NUM_JOINTS; Loop++) {
            newTorque[Loop-1]+=matrix_by_vector(dynamics.mQ,dynamics.pX,
                                                Loop-1)/50.0;
        }
    }
}

for (Loop = 1; Loop <= NUM_JOINTS; Loop++) {
    if ((controller_type&CTRL_MASK_MODEL_FEEDBACK_LINEARISATION)!=
                            CTRL_MASK_MODEL_FEEDBACK_LINEARISATION) {
        newTorque[Loop-1]+=c_error[Loop-1];
    }
    if (newTorque[Loop-1] > PA10_RatedTorque[Loop-1]) {
        newTorque[Loop-1] = PA10_RatedTorque[Loop-1];
        saturated=1;
    }
    if (newTorque[Loop-1] < -PA10_RatedTorque[Loop-1]) {
        newTorque[Loop-1] = -PA10_RatedTorque[Loop-1];
        saturated=1;
    }
}

if ((controller_type&CRTL_MASK_GRAVITY_TYPE)==CRTL_MASK_GRAVITY_ADAPT) {
    if (!saturated) {
        int i,j;
        double diff_el=0;
        double dt=static_cast<double>(period)/1000.0;
        int adapt_err_computed=0;
        if ((controller_type&CTRL_MASK_ERROR_PD)==CTRL_MASK_ERROR_PD) {
            for (i = 0; i < NUM_JOINTS; i++) {
                adapt_err[i]=-c_error[i]/kd[i];
                /*
                adapt_err[i]=((cur_vel[i]-vel[i][count])+
                    wb_adapt_p_gn[i]*(pA[i]-pos[i][count]));
                */
```

```
            }
            adapt_err_computed=1;
        }
        if ((controller_type&CTRL_MASK_ERROR_SAT_SLIDING)==
                            CTRL_MASK_ERROR_SAT_SLIDING) {
            for (i = 0; i < NUM_JOINTS; i++) {
                adapt_err[i]=-c_error[i]/K_m[i];
            }
            adapt_err_computed=1;
        }
        if ((controller_type&CTRL_MASK_ERROR_SLIDING)==
                            CTRL_MASK_ERROR_SLIDING) {
            for (i = 0; i < NUM_JOINTS; i++) {
                adapt_err[i]=-c_error[i]/kd[i];
            }
            adapt_err_computed=1;
        }
        if (adapt_err_computed!=0) {
            for (j=0; j<CARTESIAN_DIM; j++) {
                dadapt_grav[j]=0;
                for (i=0; i<NUM_JOINTS; i++) {
                    dadapt_grav[j]+=(dynamics.full_gravity[i][j]/50.0)*
                            adapt_err[i];
                }
                dynamics.gravity_vector[j]+=-dadapt_grav[j]*g_adpt_gn[j]*dt;
            }
        }
    }
}


//angle limit control
for (Loop = 1; Loop <= NUM_JOINTS; Loop++) {
    double tmp_a=static_cast<double>(pA[Loop-1]);
    if ((tmp_a>=PA10_PositivePositionLimit[Loop-1])){
        newTorque[Loop-1]=-PA10_RatedTorque[Loop-1];
        fprintf(stderr,"<%i|",Loop-1);
        fprintf(stderr,
                "LIMIT angle[%i]=%2.5Lf, tmp_a=%2.5f, max: %2.5f, last=%2.5f\n",
                Loop-1,pA[Loop-1],tmp_a,PA10_PositivePositionLimit[Loop-1],
                last_pos[Loop-1]);
    }
    if ((tmp_a<=PA10_NegativePositionLimit[Loop-1])){ //
        newTorque[Loop-1]=PA10_RatedTorque[Loop-1];
        fprintf(stderr,"|%i>",Loop-1);
        fprintf(stderr,
                "LIMIT angle[%i]=%2.5Lf, tmp_a=%2.5f, min: %2.5f, last=%2.5f\n",
                Loop-1,pA[Loop-1],tmp_a,PA10_NegativePositionLimit[Loop-1],
                last_pos[Loop-1]);
    }
}

{
```

```
    int saturated_t[NUM_JOINTS]={0,0,0,0,0,0};
    //filter torques + gain adaptation:
    for (Loop = 1; Loop <= NUM_JOINTS; Loop++) {
#ifdef ADVANCED_TORQUE_FILTER
        sgn_diff= newTorque[Loop-1]-oldTorque[Loop-1]>=0 ? 1 : -1;
        if (abs(newTorque[Loop-1]-oldTorque[Loop-1])>
            alpha[Loop-1]*PA10_RatedTorque[Loop-1]) {
            newTorque[Loop-1]=oldTorque[Loop-1]+sgn_diff*
                              alpha[Loop-1]*PA10_RatedTorque[Loop-1];
        }
        if ((newTorque[Loop-1]*oldTorque[Loop-1]<0)&&(alpha[Loop-1]!=1)) {
            newTorque[Loop-1]=0;
        }
#endif
        if (abs(pV[Loop-1])>abs(PA10_RatedVelocityZone[Loop-1]*
                            PA10_RatedVelocity[Loop-1])) {
            // velocity close to saturation.
            if ((controller_type&CRTL_MASK_SPC_TYPE)==CRTL_MASK_SPC_DTZ) {
                newTorque[Loop-1]=newTorque[Loop-1]/vel_of_trq_div[Loop-1];
            }
#ifdef ADVANCED_TORQUE_FILTER
             else {
                newTorque[Loop-1]=newTorque[Loop-1]/vel_of_trq_div[Loop-1];
            }
#endif
            saturated_t[Loop-1]=1;
            //fprintf(stderr,".");
        }

        if (abs(newTorque[Loop-1]) == abs(PA10_RatedTorque[Loop-1])) {
            // joint torque saturated
            saturated_t[Loop-1]=1;
        }
    }

    for (Loop = 1; Loop <= NUM_JOINTS; Loop++) {
        if ((anti_windup==0)||(saturated_t[Loop-1]=0)) {
            int_err[Loop-1] += (pA[Loop-1]-tmp_pos[Loop-1]) *
                       static_cast<double>(period)/1000.0;
        }
    }
}

if ((controller_type&CRTL_MASK_CMD_TYPE)==CRTL_MASK_CMD_TORQUE) {
    for (Loop = 1; Loop <= NUM_JOINTS; Loop++) {
        pT[Loop-1] = newTorque[Loop-1];
        oldTorque[Loop-1]= newTorque[Loop-1];
        newTorque[Loop-1]=0.0;
    }
} else if ((controller_type&CRTL_MASK_CMD_TYPE)==CRTL_MASK_CMD_VELOCITY) {
    //fprintf(stdout,"Velocity control commands \n");
    for (Loop = 1; Loop <= NUM_JOINTS; Loop++) {
        pV[Loop-1] = vel[Loop-1][count];
```

```
        pT[Loop-1] = 0.0;
    }
}
for (Loop = 1; Loop <= NUM_JOINTS; Loop++) {
    last_c_error[Loop-1]=c_error[Loop-1];
    c_error[Loop-1]=0.0;
    last_vel[Loop-1]=pV[Loop-1];
    last_pos[Loop-1]=pA[Loop-1];
}
```

Appendix E

Tables with results of manipulator on platform control experiments (angle WMISE)

The results of angle tracking weighted mean squared error are presented in tables: E.1 to E.14.

controller	S1	S2	E1	E2	W1	W2
PD	2.20e-5	2.76e-4	1.58e-4	2.14e-5	3.21e-5	1.32e-5
sat_sl	2.21e-5	2.76e-4	1.59e-4	1.85e-5	3.18e-5	1.32e-5
slid	2.20e-5	2.76e-4	4.27e-4	6.13e-6	3.21e-5	1.35e-5
PID	2.16e-5	2.74e-4	1.58e-4	2.15e-5	3.24e-5	1.37e-5
grv_frc_PD	3.81e-6	6.02e-6	1.08e-5	1.67e-6	3.70e-6	1.38e-6
grv_frc_PD_s_rp	3.70e-6	5.84e-6	1.05e-5	1.54e-6	3.38e-6	1.22e-6
grv_frc_PD_s_acc	3.65e-6	5.73e-6	1.02e-5	1.30e-6	3.37e-6	1.12e-6
grv_frc_PD_adap	1.31e-6	1.03e-6	2.59e-6	1.48e-6	3.44e-6	1.28e-6
grv_frc_sat_sl	3.78e-6	5.82e-6	1.06e-5	1.31e-6	3.44e-6	1.21e-6
grv_frc_sat_sl_s_rp	3.71e-6	5.92e-6	1.09e-5	1.28e-6	3.37e-6	1.14e-6
grv_frc_sat_sl_s_acc	3.68e-6	5.54e-6	1.03e-5	1.23e-6	3.42e-6	1.16e-6
grv_frc_sat_sl_adap	1.17e-6	6.64e-7	3.27e-6	1.32e-6	3.45e-6	1.28e-6
grv_frc_slid	3.89e-6	6.03e-6	2.78e-5	1.52e-6	3.49e-6	1.27e-6
grv_frc_slid_s_rp	3.83e-6	5.83e-6	2.86e-5	1.68e-6	3.64e-6	1.18e-6
grv_frc_slid_s_acc	3.83e-6	6.22e-6	2.88e-5	1.87e-6	3.66e-6	1.30e-6
grv_frc_slid_adap	1.52e-6	1.15e-6	6.38e-6	1.69e-6	3.42e-6	1.20e-6
lin_FULL	2.89e-6	7.89e-7	2.17e-6	5.89e-6	7.83e-6	6.76e-7
lin_noCC	2.84e-6	7.74e-7	2.48e-6	5.28e-6	7.97e-6	7.01e-7
lin_FULL_s_rp	2.66e-6	7.84e-7	1.91e-6	5.27e-6	7.41e-6	7.77e-7
lin_noCC_s_rp	3.32e-6	8.08e-7	2.55e-6	5.44e-6	7.79e-6	8.33e-7
lin_FULL_s_acc	2.84e-6	9.43e-7	2.30e-6	5.73e-6	7.79e-6	8.09e-7
lin_noCC_s_acc	3.46e-6	9.06e-7	2.74e-6	5.68e-6	7.49e-6	7.41e-7
W-B_FULL	1.54e-6	2.61e-6	2.70e-6	2.42e-6	3.26e-6	9.46e-7
W-B_noCC	1.91e-6	2.70e-6	3.90e-6	1.75e-6	3.46e-6	1.04e-6
W-B_FULL_s_rp	1.63e-6	2.64e-6	2.87e-6	1.97e-6	3.39e-6	1.16e-6
W-B_noCC_s_rp	2.02e-6	2.30e-6	4.10e-6	2.54e-6	3.43e-6	1.05e-6
W-B_FULL_s_acc	1.74e-6	2.67e-6	3.31e-6	2.17e-6	3.28e-6	1.00e-6
W-B_noCC_s_acc	1.85e-6	2.60e-6	4.08e-6	2.02e-6	3.38e-6	8.96e-7
W-B_FULL_adap	9.39e-7	6.91e-7	2.02e-6	1.84e-6	3.38e-6	1.09e-6
W-B_noCC_adap	9.78e-7	9.57e-7	2.27e-6	2.50e-6	3.43e-6	9.40e-7
S-L_FULL	1.64e-6	2.46e-6	7.26e-6	4.17e-6	3.36e-6	9.98e-7
S-L_noCC	2.14e-6	2.31e-6	1.03e-5	4.21e-6	3.71e-6	1.07e-6
S-L_FULL_s_rp	1.78e-6	2.89e-6	7.71e-6	4.85e-6	3.79e-6	9.77e-7
S-L_noCC_s_rp	1.83e-6	2.48e-6	1.03e-5	4.27e-6	3.59e-6	9.51e-7
S-L_FULL_s_acc	1.80e-6	2.47e-6	9.34e-6	4.14e-6	3.54e-6	1.08e-6
S-L_noCC_s_acc	1.91e-6	2.28e-6	1.20e-5	4.10e-6	3.58e-6	9.91e-7
S-L_FULL_adap	1.03e-6	1.22e-6	5.22e-6	4.02e-6	3.36e-6	9.14e-7
S-L_noCC_adap	1.35e-6	1.14e-6	6.74e-6	4.62e-6	3.72e-6	1.14e-6
PA10_vel	1.12e-2	4.51e-5	4.79e-5	3.38e-5	1.59e-4	3.37e-5

Table E.1: Comparison of angle WMISE results for different controllers, platform trajectory: zero, PA10 trajectory: **`sine`**.

controller	S1	S2	E1	E2	W1	W2
PD	2.06e-5	1.57e-4	3.60e-5	1.49e-5	2.20e-5	1.12e-5
sat_sl	2.03e-5	1.57e-4	3.49e-5	1.42e-5	2.23e-5	1.16e-5
slid	2.04e-5	1.56e-4	9.71e-5	5.11e-6	2.21e-5	1.14e-5
PID	2.07e-5	1.57e-4	3.58e-5	1.50e-5	2.24e-5	1.14e-5
grv_frc_PD	2.68e-6	2.57e-6	3.91e-6	9.32e-7	1.77e-6	8.16e-7
grv_frc_PD_s_rp	2.80e-6	2.55e-6	4.15e-6	1.02e-6	1.88e-6	8.86e-7
grv_frc_PD_s_acc	2.78e-6	2.59e-6	4.04e-6	9.93e-7	1.76e-6	8.09e-7
grv_frc_PD_adap	1.15e-6	1.25e-6	2.90e-6	1.10e-6	1.87e-6	9.45e-7
grv_frc_sat_sl	2.75e-6	2.65e-6	3.99e-6	9.55e-7	1.69e-6	8.05e-7
grv_frc_sat_sl_s_rp	2.81e-6	2.69e-6	4.11e-6	9.62e-7	1.81e-6	8.99e-7
grv_frc_sat_sl_s_acc	2.78e-6	2.52e-6	4.09e-6	9.37e-7	1.75e-6	8.26e-7
grv_frc_sat_sl_adap	9.10e-7	7.09e-7	3.89e-6	1.02e-6	1.85e-6	9.54e-7
grv_frc_slid	2.84e-6	2.72e-6	1.07e-5	1.15e-6	1.96e-6	9.55e-7
grv_frc_slid_s_rp	3.03e-6	2.72e-6	1.09e-5	1.27e-6	1.87e-6	8.73e-7
grv_frc_slid_s_acc	2.85e-6	2.52e-6	1.06e-5	1.18e-6	1.86e-6	8.90e-7
grv_frc_slid_adap	1.40e-6	1.49e-6	7.18e-6	1.19e-6	1.80e-6	8.90e-7
lin_FULL	4.15e-6	1.20e-6	4.56e-6	4.22e-6	4.61e-6	5.74e-7
lin_noCC	3.79e-6	1.12e-6	4.57e-6	3.74e-6	4.37e-6	6.26e-7
lin_FULL_s_rp	3.78e-6	1.13e-6	4.33e-6	3.87e-6	4.30e-6	6.65e-7
lin_noCC_s_rp	4.07e-6	1.14e-6	4.76e-6	3.78e-6	4.37e-6	6.32e-7
lin_FULL_s_acc	3.90e-6	1.35e-6	4.75e-6	4.29e-6	4.50e-6	6.86e-7
lin_noCC_s_acc	4.11e-6	1.28e-6	4.92e-6	4.21e-6	4.39e-6	6.73e-7
W-B_FULL	1.66e-6	2.65e-6	3.62e-6	1.44e-6	1.80e-6	7.52e-7
W-B_noCC	1.83e-6	2.60e-6	4.44e-6	1.55e-6	1.91e-6	8.76e-7
W-B_FULL_s_rp	1.70e-6	2.38e-6	3.78e-6	1.83e-6	1.85e-6	1.02e-6
W-B_noCC_s_rp	1.88e-6	2.18e-6	4.62e-6	1.81e-6	1.87e-6	8.33e-7
W-B_FULL_s_acc	1.88e-6	2.66e-6	3.73e-6	1.51e-6	1.88e-6	9.16e-7
W-B_noCC_s_acc	1.87e-6	2.53e-6	4.65e-6	1.68e-6	2.01e-6	8.50e-7
W-B_FULL_adap	1.12e-6	1.25e-6	3.20e-6	1.69e-6	1.89e-6	8.41e-7
W-B_noCC_adap	9.28e-7	1.23e-6	2.90e-6	1.77e-6	1.91e-6	7.85e-7
S-L_FULL	2.04e-6	2.70e-6	1.00e-5	2.98e-6	1.94e-6	8.81e-7
S-L_noCC	2.03e-6	2.39e-6	1.16e-5	2.89e-6	1.87e-6	8.30e-7
S-L_FULL_s_rp	1.84e-6	2.64e-6	9.94e-6	3.43e-6	1.94e-6	8.57e-7
S-L_noCC_s_rp	1.73e-6	2.28e-6	1.12e-5	3.08e-6	1.94e-6	7.53e-7
S-L_FULL_s_acc	1.80e-6	2.54e-6	9.15e-6	2.94e-6	1.90e-6	8.69e-7
S-L_noCC_s_acc	1.93e-6	2.32e-6	1.15e-5	2.85e-6	1.94e-6	7.68e-7
S-L_FULL_adap	1.14e-6	1.40e-6	7.93e-6	2.97e-6	1.93e-6	7.02e-7
S-L_noCC_adap	1.35e-6	1.53e-6	7.61e-6	3.41e-6	1.90e-6	8.24e-7
PA10_vel	1.15e-2	3.57e-5	2.36e-5	1.77e-5	3.77e-5	1.75e-5

Table E.2: Comparison of angle WMISE results for different controllers, platform trajectory: zero, PA10 trajectory: **ssine**.

controller	S1	S2	E1	E2	W1	W2
PD	3.63e-5	3.02e-4	1.74e-4	2.02e-5	2.98e-5	1.40e-5
sat_sl	3.52e-5	3.02e-4	1.70e-4	1.95e-5	3.00e-5	1.34e-5
slid	3.50e-5	3.01e-4	4.61e-4	7.85e-6	2.98e-5	1.38e-5
PID	3.51e-5	3.01e-4	1.74e-4	2.08e-5	2.95e-5	1.35e-5
grv_frc_PD	5.87e-6	2.54e-5	2.17e-5	1.27e-6	4.01e-6	1.06e-6
grv_frc_PD_s_rp	5.56e-6	1.46e-5	1.46e-5	1.41e-6	3.96e-6	1.01e-6
grv_frc_PD_s_acc	3.57e-6	8.89e-6	1.16e-5	1.23e-6	4.16e-6	1.06e-6
grv_frc_PD_adap	1.99e-6	1.41e-6	3.12e-6	1.58e-6	4.59e-6	1.38e-6
grv_frc_sat_sl	6.43e-6	2.63e-5	2.22e-5	1.14e-6	4.30e-6	1.24e-6
grv_frc_sat_sl_s_rp	5.40e-6	1.46e-5	1.42e-5	1.01e-6	3.95e-6	1.25e-6
grv_frc_sat_sl_s_acc	3.60e-6	9.76e-6	1.15e-5	1.11e-6	3.80e-6	1.10e-6
grv_frc_sat_sl_adap	1.45e-6	7.35e-7	3.36e-6	1.09e-6	3.60e-6	1.16e-6
grv_frc_slid	6.12e-6	2.58e-5	5.89e-5	1.85e-6	4.15e-6	1.12e-6
grv_frc_slid_s_rp	5.38e-6	1.46e-5	3.86e-5	1.75e-6	4.12e-6	1.08e-6
grv_frc_slid_s_acc	3.81e-6	9.68e-6	3.09e-5	1.53e-6	4.28e-6	1.14e-6
grv_frc_slid_adap	1.97e-6	1.28e-6	6.96e-6	1.56e-6	4.07e-6	1.12e-6
lin_FULL	5.62e-6	2.52e-6	2.83e-6	4.39e-6	8.23e-6	6.54e-7
lin_noCC	7.74e-6	2.70e-6	3.44e-6	4.64e-6	9.78e-6	8.58e-7
lin_FULL_s_rp	3.19e-6	1.57e-6	2.29e-6	4.37e-6	9.30e-6	6.53e-7
lin_noCC_s_rp	4.64e-6	1.57e-6	2.88e-6	4.57e-6	9.92e-6	6.94e-7
lin_FULL_s_acc	2.55e-6	1.04e-6	2.16e-6	4.65e-6	9.19e-6	6.99e-7
lin_noCC_s_acc	3.28e-6	9.40e-7	2.55e-6	4.62e-6	9.46e-6	6.65e-7
W-B_FULL	3.58e-6	2.10e-5	1.72e-5	1.40e-6	3.86e-6	9.23e-7
W-B_noCC	4.52e-6	1.99e-5	1.72e-5	1.79e-6	3.89e-6	7.21e-7
W-B_FULL_s_rp	1.89e-6	9.84e-6	8.14e-6	1.72e-6	4.18e-6	9.61e-7
W-B_noCC_s_rp	2.83e-6	9.52e-6	9.09e-6	1.90e-6	4.12e-6	8.92e-7
W-B_FULL_s_acc	1.27e-6	5.63e-6	5.25e-6	1.35e-6	3.98e-6	9.65e-7
W-B_noCC_s_acc	1.87e-6	4.71e-6	5.68e-6	1.68e-6	3.98e-6	9.16e-7
W-B_FULL_adap	1.19e-6	9.71e-7	2.38e-6	1.60e-6	4.16e-6	8.84e-7
W-B_noCC_adap	1.49e-6	1.11e-6	2.51e-6	1.93e-6	4.02e-6	8.54e-7
S-L_FULL	3.43e-6	1.99e-5	4.17e-5	3.15e-6	4.14e-6	8.99e-7
S-L_noCC	4.49e-6	1.90e-5	4.19e-5	4.02e-6	4.23e-6	8.08e-7
S-L_FULL_s_rp	2.08e-6	9.41e-6	1.91e-5	3.39e-6	4.30e-6	9.09e-7
S-L_noCC_s_rp	2.74e-6	8.90e-6	2.18e-5	3.81e-6	4.33e-6	1.01e-6
S-L_FULL_s_acc	1.29e-6	5.47e-6	1.21e-5	3.02e-6	4.21e-6	8.41e-7
S-L_noCC_s_acc	1.86e-6	4.87e-6	1.39e-5	4.46e-6	4.37e-6	8.95e-7
S-L_FULL_adap	1.54e-6	1.59e-6	6.68e-6	3.51e-6	4.22e-6	8.52e-7
S-L_noCC_adap	1.99e-6	1.54e-6	7.25e-6	4.72e-6	4.68e-6	1.15e-6
PA10_vel	1.11e-2	4.27e-5	5.42e-5	4.11e-5	1.28e-4	3.21e-5

Table E.3: Comparison of angle WMISE results for different controllers, platform trajectory: sin_r2p0, PA10 trajectory: **`sine`**.

controller	S1	S2	E1	E2	W1	W2
PD	2.41e-5	1.39e-4	7.45e-5	1.53e-5	2.10e-5	1.22e-5
sat_sl	2.35e-5	1.40e-4	6.88e-5	1.46e-5	2.13e-5	1.14e-5
slid	2.31e-5	1.38e-4	1.99e-4	5.86e-6	2.10e-5	1.15e-5
PID	2.34e-5	1.38e-4	7.46e-5	1.48e-5	2.09e-5	1.14e-5
grv_frc_PD	2.82e-6	1.79e-5	1.70e-5	9.94e-7	1.96e-6	7.80e-7
grv_frc_PD_s_rp	2.12e-6	8.05e-6	6.78e-6	1.09e-6	1.88e-6	7.75e-7
grv_frc_PD_s_acc	2.31e-6	5.35e-6	4.01e-6	9.44e-7	1.94e-6	8.04e-7
grv_frc_PD_adap	6.13e-7	1.27e-6	2.54e-6	1.02e-6	2.10e-6	8.22e-7
grv_frc_sat_sl	2.77e-6	1.79e-5	1.64e-5	9.04e-7	1.90e-6	8.23e-7
grv_frc_sat_sl_s_rp	1.95e-6	7.95e-6	6.34e-6	8.49e-7	1.78e-6	9.09e-7
grv_frc_sat_sl_s_acc	2.44e-6	5.37e-6	4.01e-6	9.21e-7	1.81e-6	8.02e-7
grv_frc_sat_sl_adap	5.62e-7	6.89e-7	3.16e-6	8.45e-7	1.88e-6	8.45e-7
grv_frc_slid	2.59e-6	1.79e-5	4.36e-5	1.29e-6	2.03e-6	8.09e-7
grv_frc_slid_s_rp	2.07e-6	8.14e-6	1.67e-5	1.28e-6	1.94e-6	7.91e-7
grv_frc_slid_s_acc	2.30e-6	5.34e-6	1.03e-5	1.19e-6	2.04e-6	7.82e-7
grv_frc_slid_adap	7.45e-7	1.33e-6	6.20e-6	1.29e-6	2.06e-6	8.27e-7
lin_FULL	4.99e-6	2.89e-6	5.60e-6	3.29e-6	4.25e-6	5.90e-7
lin_noCC	6.53e-6	2.64e-6	5.46e-6	3.20e-6	4.97e-6	5.29e-7
lin_FULL_s_rp	2.65e-6	1.88e-6	3.90e-6	3.26e-6	4.66e-6	5.71e-7
lin_noCC_s_rp	2.98e-6	1.77e-6	4.29e-6	3.21e-6	5.00e-6	5.55e-7
lin_FULL_s_acc	3.27e-6	1.65e-6	4.35e-6	3.04e-6	4.82e-6	5.61e-7
lin_noCC_s_acc	3.91e-6	1.56e-6	4.80e-6	3.11e-6	4.65e-6	5.74e-7
W-B_FULL	2.51e-6	2.06e-5	2.19e-5	1.40e-6	1.87e-6	7.74e-7
W-B_noCC	2.90e-6	1.69e-5	1.86e-5	1.30e-6	2.00e-6	6.64e-7
W-B_FULL_s_rp	1.33e-6	8.89e-6	7.20e-6	1.34e-6	1.98e-6	6.82e-7
W-B_noCC_s_rp	1.51e-6	7.70e-6	7.38e-6	1.42e-6	1.96e-6	7.36e-7
W-B_FULL_s_acc	1.59e-6	6.08e-6	4.79e-6	1.50e-6	2.01e-6	8.70e-7
W-B_noCC_s_acc	1.78e-6	4.43e-6	4.62e-6	1.28e-6	1.98e-6	8.19e-7
W-B_FULL_adap	4.87e-7	1.16e-6	3.05e-6	1.45e-6	2.07e-6	7.32e-7
W-B_noCC_adap	5.25e-7	1.25e-6	2.88e-6	1.42e-6	2.05e-6	7.27e-7
S-L_FULL	2.46e-6	1.92e-5	4.96e-5	2.40e-6	2.07e-6	7.14e-7
S-L_noCC	2.99e-6	1.61e-5	4.37e-5	2.62e-6	2.16e-6	6.56e-7
S-L_FULL_s_rp	1.40e-6	8.83e-6	1.70e-5	2.34e-6	2.10e-6	8.17e-7
S-L_noCC_s_rp	1.53e-6	7.74e-6	1.71e-5	2.91e-6	2.18e-6	8.19e-7
S-L_FULL_s_acc	1.61e-6	5.93e-6	1.18e-5	2.57e-6	2.07e-6	7.72e-7
S-L_noCC_s_acc	1.95e-6	4.82e-6	1.16e-5	3.88e-6	2.26e-6	9.18e-7
S-L_FULL_adap	6.78e-7	1.30e-6	8.33e-6	2.67e-6	2.05e-6	7.54e-7
S-L_noCC_adap	7.06e-7	1.52e-6	6.92e-6	2.57e-6	2.20e-6	7.38e-7
PA10_vel	1.13e-2	3.05e-5	2.46e-5	1.72e-5	3.26e-5	1.63e-5

Table E.4: Comparison of angle WMISE results for different controllers, platform trajectory: sin_r2p0, PA10 trajectory: **ssine**.

controller	S1	S2	E1	E2	W1	W2
PD	3.67e-5	3.02e-4	1.75e-4	2.06e-5	3.05e-5	1.33e-5
sat_sl	3.64e-5	3.01e-4	1.71e-4	1.85e-5	3.15e-5	1.35e-5
slid	3.45e-5	3.00e-4	4.63e-4	8.26e-6	3.13e-5	1.37e-5
PID	3.48e-5	2.99e-4	1.75e-4	1.89e-5	3.11e-5	1.36e-5
grv_frc_PD	1.40e-5	1.76e-5	2.08e-5	1.28e-6	3.52e-6	1.19e-6
grv_frc_PD_s_rp	8.51e-6	1.30e-5	1.62e-5	1.64e-6	3.49e-6	1.15e-6
grv_frc_PD_s_acc	5.00e-6	8.40e-6	1.34e-5	1.39e-6	3.36e-6	1.21e-6
grv_frc_PD_adap	2.28e-6	1.36e-6	2.89e-6	1.52e-6	3.37e-6	1.20e-6
grv_frc_sat_sl	1.37e-5	1.75e-5	2.08e-5	1.14e-6	3.57e-6	1.19e-6
grv_frc_sat_sl_s_rp	8.49e-6	1.35e-5	1.57e-5	1.20e-6	3.42e-6	1.31e-6
grv_frc_sat_sl_s_acc	5.29e-6	8.82e-6	1.21e-5	1.21e-6	3.50e-6	1.26e-6
grv_frc_sat_sl_adap	1.65e-6	7.22e-7	3.47e-6	1.31e-6	3.34e-6	1.19e-6
grv_frc_slid	1.41e-5	1.78e-5	5.55e-5	1.89e-6	3.81e-6	1.20e-6
grv_frc_slid_s_rp	8.51e-6	1.31e-5	4.23e-5	1.84e-6	3.55e-6	1.23e-6
grv_frc_slid_s_acc	5.07e-6	8.59e-6	3.44e-5	1.57e-6	3.63e-6	1.28e-6
grv_frc_slid_adap	2.44e-6	1.38e-6	7.15e-6	2.11e-6	3.89e-6	1.34e-6
lin_FULL	1.01e-5	1.73e-6	2.91e-6	5.59e-6	7.80e-6	6.90e-7
lin_noCC	1.38e-5	1.80e-6	3.68e-6	5.68e-6	8.21e-6	7.58e-7
lin_FULL_s_rp	6.11e-6	1.21e-6	2.48e-6	5.78e-6	7.91e-6	6.83e-7
lin_noCC_s_rp	8.02e-6	1.29e-6	3.18e-6	5.78e-6	8.23e-6	7.60e-7
lin_FULL_s_acc	3.68e-6	1.03e-6	2.51e-6	5.92e-6	8.47e-6	7.50e-7
lin_noCC_s_acc	4.50e-6	1.04e-6	3.06e-6	5.82e-6	8.49e-6	7.57e-7
W-B_FULL	7.01e-6	1.13e-5	1.11e-5	1.69e-6	3.67e-6	7.96e-7
W-B_noCC	8.27e-6	1.21e-5	1.34e-5	1.42e-6	3.88e-6	8.48e-7
W-B_FULL_s_rp	3.65e-6	6.76e-6	6.53e-6	1.80e-6	3.76e-6	7.70e-7
W-B_noCC_s_rp	4.38e-6	6.82e-6	8.09e-6	1.52e-6	3.78e-6	8.23e-7
W-B_FULL_s_acc	1.70e-6	3.93e-6	4.35e-6	2.31e-6	4.11e-6	9.88e-7
W-B_noCC_s_acc	1.81e-6	4.16e-6	5.46e-6	1.43e-6	3.63e-6	7.98e-7
W-B_FULL_adap	1.24e-6	8.93e-7	1.93e-6	1.86e-6	3.79e-6	8.04e-7
W-B_noCC_adap	1.49e-6	9.15e-7	2.11e-6	1.66e-6	3.95e-6	7.91e-7
S-L_FULL	7.29e-6	1.06e-5	2.64e-5	3.96e-6	4.26e-6	8.02e-7
S-L_noCC	7.96e-6	1.13e-5	3.23e-5	3.50e-6	3.99e-6	8.19e-7
S-L_FULL_s_rp	3.82e-6	6.17e-6	1.50e-5	3.15e-6	4.12e-6	7.61e-7
S-L_noCC_s_rp	4.64e-6	7.21e-6	2.15e-5	5.71e-6	4.37e-6	9.90e-7
S-L_FULL_s_acc	1.70e-6	3.67e-6	9.35e-6	3.77e-6	4.11e-6	8.03e-7
S-L_noCC_s_acc	1.97e-6	4.04e-6	1.45e-5	3.88e-6	4.10e-6	8.43e-7
S-L_FULL_adap	1.99e-6	1.36e-6	5.10e-6	3.47e-6	4.08e-6	8.71e-7
S-L_noCC_adap	2.30e-6	1.26e-6	6.21e-6	4.09e-6	3.97e-6	7.86e-7
PA10_vel	1.11e-2	4.33e-5	1.05e-4	3.28e-5	1.48e-4	3.41e-5

Table E.5: Comparison of angle WMISE results for different controllers, platform trajectory: sin_r0p2, PA10 trajectory: **`sine`**.

controller	S1	S2	E1	E2	W1	W2
PD	2.25e-5	1.78e-4	7.26e-5	1.49e-5	2.10e-5	1.15e-5
sat_sl	2.32e-5	1.81e-4	6.82e-5	1.43e-5	2.18e-5	1.17e-5
slid	2.25e-5	1.79e-4	1.92e-4	5.12e-6	2.12e-5	1.16e-5
PID	2.02e-5	1.73e-4	7.40e-5	1.48e-5	2.14e-5	1.17e-5
grv_frc_PD	1.50e-5	2.40e-5	1.63e-5	9.47e-7	1.93e-6	9.03e-7
grv_frc_PD_s_rp	7.43e-6	1.19e-5	5.40e-6	9.73e-7	1.89e-6	9.02e-7
grv_frc_PD_s_acc	5.00e-6	6.39e-6	5.43e-6	9.55e-7	1.99e-6	9.64e-7
grv_frc_PD_adap	2.08e-6	1.78e-6	2.17e-6	1.16e-6	2.13e-6	9.24e-7
grv_frc_sat_sl	1.51e-5	2.32e-5	1.69e-5	8.71e-7	2.01e-6	9.20e-7
grv_frc_sat_sl_s_rp	7.72e-6	1.22e-5	5.39e-6	8.84e-7	1.92e-6	9.90e-7
grv_frc_sat_sl_s_acc	5.13e-6	6.78e-6	5.07e-6	8.80e-7	2.07e-6	9.35e-7
grv_frc_sat_sl_adap	1.27e-6	9.53e-7	2.85e-6	9.18e-7	1.94e-6	8.45e-7
grv_frc_slid	1.52e-5	2.35e-5	4.30e-5	1.20e-6	1.99e-6	8.15e-7
grv_frc_slid_s_rp	7.32e-6	1.18e-5	1.32e-5	1.23e-6	1.97e-6	7.81e-7
grv_frc_slid_s_acc	4.94e-6	6.47e-6	1.35e-5	1.25e-6	2.03e-6	9.00e-7
grv_frc_slid_adap	2.18e-6	1.85e-6	4.35e-6	1.16e-6	1.92e-6	9.14e-7
lin_FULL	1.81e-5	2.83e-6	4.41e-6	3.57e-6	5.18e-6	5.90e-7
lin_noCC	2.22e-5	3.05e-6	4.58e-6	3.50e-6	5.06e-6	6.67e-7
lin_FULL_s_rp	1.08e-5	2.55e-6	3.67e-6	3.79e-6	5.41e-6	7.07e-7
lin_noCC_s_rp	1.13e-5	2.30e-6	3.74e-6	3.47e-6	5.20e-6	5.90e-7
lin_FULL_s_acc	5.86e-6	1.58e-6	3.91e-6	3.63e-6	5.54e-6	5.72e-7
lin_noCC_s_acc	7.06e-6	1.54e-6	4.25e-6	3.58e-6	5.57e-6	6.04e-7
W-B_FULL	1.03e-5	2.09e-5	1.31e-5	1.14e-6	2.09e-6	6.89e-7
W-B_noCC	1.10e-5	2.15e-5	1.73e-5	1.04e-6	2.21e-6	6.52e-7
W-B_FULL_s_rp	5.59e-6	1.19e-5	4.18e-6	1.33e-6	2.18e-6	7.03e-7
W-B_noCC_s_rp	5.94e-6	1.16e-5	6.06e-6	1.50e-6	2.42e-6	7.03e-7
W-B_FULL_s_acc	2.51e-6	5.55e-6	3.37e-6	1.01e-6	2.14e-6	6.77e-7
W-B_noCC_s_acc	2.76e-6	5.68e-6	5.47e-6	1.07e-6	2.33e-6	6.78e-7
W-B_FULL_adap	1.54e-6	1.47e-6	1.82e-6	1.09e-6	2.38e-6	6.73e-7
W-B_noCC_adap	1.60e-6	1.56e-6	1.75e-6	1.37e-6	2.56e-6	6.87e-7
S-L_FULL	1.00e-5	1.91e-5	3.26e-5	2.91e-6	2.52e-6	6.78e-7
S-L_noCC	1.14e-5	2.02e-5	4.03e-5	3.67e-6	2.67e-6	7.51e-7
S-L_FULL_s_rp	5.67e-6	1.12e-5	9.44e-6	2.97e-6	2.51e-6	6.60e-7
S-L_noCC_s_rp	5.79e-6	1.11e-5	1.22e-5	2.99e-6	2.59e-6	7.22e-7
S-L_FULL_s_acc	2.69e-6	5.49e-6	8.25e-6	2.89e-6	2.57e-6	7.11e-7
S-L_noCC_s_acc	2.98e-6	5.60e-6	1.28e-5	2.66e-6	2.78e-6	7.01e-7
S-L_FULL_adap	2.40e-6	2.11e-6	3.51e-6	2.62e-6	2.50e-6	7.75e-7
S-L_noCC_adap	2.58e-6	2.24e-6	3.68e-6	2.60e-6	2.56e-6	6.43e-7
PA10_vel	1.15e-2	3.27e-5	5.52e-5	1.70e-5	2.99e-5	1.67e-5

Table E.6: Comparison of angle WMISE results for different controllers, platform trajectory: sin_r0p2, PA10 trajectory: **ssine**.

controller	S1	S2	E1	E2	W1	W2
PD	4.21e-5	3.19e-4	1.73e-4	2.14e-5	3.41e-5	1.59e-5
sat_sl	4.05e-5	3.20e-4	1.70e-4	1.99e-5	3.38e-5	1.52e-5
slid	3.99e-5	3.20e-4	4.65e-4	6.89e-6	3.15e-5	1.45e-5
PID	4.15e-5	3.20e-4	1.73e-4	2.24e-5	3.27e-5	1.49e-5
grv_frc_PD	1.34e-5	1.34e-5	2.05e-5	1.20e-6	3.71e-6	9.34e-7
grv_frc_PD_s_rp	6.87e-6	1.08e-5	1.41e-5	1.30e-6	3.95e-6	9.03e-7
grv_frc_PD_s_acc	4.58e-6	6.63e-6	1.10e-5	9.81e-7	3.94e-6	9.09e-7
grv_frc_PD_adap	1.76e-6	1.02e-6	2.29e-6	1.16e-6	3.95e-6	9.87e-7
grv_frc_sat_sl	1.36e-5	1.31e-5	2.05e-5	1.05e-6	4.27e-6	1.01e-6
grv_frc_sat_sl_s_rp	7.07e-6	1.11e-5	1.45e-5	1.08e-6	4.15e-6	9.54e-7
grv_frc_sat_sl_s_acc	4.40e-6	6.38e-6	1.12e-5	1.01e-6	4.05e-6	9.34e-7
grv_frc_sat_sl_adap	1.35e-6	8.05e-7	3.00e-6	1.21e-6	4.05e-6	1.03e-6
grv_frc_slid	1.34e-5	1.34e-5	5.39e-5	2.07e-6	4.00e-6	9.08e-7
grv_frc_slid_s_rp	7.05e-6	1.08e-5	3.88e-5	1.62e-6	3.93e-6	9.74e-7
grv_frc_slid_s_acc	4.69e-6	6.68e-6	2.91e-5	1.48e-6	3.99e-6	9.03e-7
grv_frc_slid_adap	1.90e-6	1.08e-6	5.45e-6	1.83e-6	3.99e-6	8.54e-7
lin_FULL	7.86e-6	1.55e-6	3.14e-6	4.60e-6	9.51e-6	4.96e-7
lin_noCC	1.02e-5	1.65e-6	3.90e-6	4.98e-6	1.00e-5	5.77e-7
lin_FULL_s_rp	4.74e-6	8.22e-7	2.12e-6	4.50e-6	9.48e-6	5.53e-7
lin_noCC_s_rp	6.41e-6	8.62e-7	2.72e-6	4.48e-6	9.40e-6	4.95e-7
lin_FULL_s_acc	2.87e-6	7.21e-7	1.86e-6	4.52e-6	9.21e-6	4.98e-7
lin_noCC_s_acc	3.90e-6	7.15e-7	2.53e-6	4.46e-6	9.08e-6	4.53e-7
W-B_FULL	6.16e-6	1.17e-5	1.24e-5	1.62e-6	3.95e-6	6.03e-7
W-B_noCC	7.81e-6	1.07e-5	1.47e-5	1.93e-6	3.99e-6	5.91e-7
W-B_FULL_s_rp	3.12e-6	3.99e-6	5.32e-6	1.45e-6	3.82e-6	6.97e-7
W-B_noCC_s_rp	3.79e-6	4.49e-6	7.24e-6	1.85e-6	3.89e-6	6.47e-7
W-B_FULL_s_acc	1.35e-6	2.46e-6	3.36e-6	1.42e-6	3.82e-6	6.07e-7
W-B_noCC_s_acc	1.89e-6	2.99e-6	5.19e-6	1.86e-6	3.87e-6	4.95e-7
W-B_FULL_adap	1.06e-6	7.71e-7	1.76e-6	1.69e-6	3.95e-6	6.61e-7
W-B_noCC_adap	1.19e-6	7.78e-7	1.81e-6	2.17e-6	3.67e-6	5.73e-7
S-L_FULL	6.23e-6	1.05e-5	2.96e-5	3.68e-6	3.93e-6	5.34e-7
S-L_noCC	7.29e-6	1.07e-5	3.53e-5	4.11e-6	3.84e-6	6.56e-7
S-L_FULL_s_rp	3.41e-6	3.75e-6	1.28e-5	3.40e-6	4.00e-6	6.04e-7
S-L_noCC_s_rp	4.12e-6	4.04e-6	1.83e-5	3.95e-6	4.05e-6	5.99e-7
S-L_FULL_s_acc	1.60e-6	2.32e-6	7.79e-6	3.36e-6	3.93e-6	6.65e-7
S-L_noCC_s_acc	1.98e-6	2.56e-6	1.24e-5	3.74e-6	3.92e-6	5.74e-7
S-L_FULL_adap	1.58e-6	9.00e-7	3.99e-6	3.63e-6	4.08e-6	6.12e-7
S-L_noCC_adap	1.89e-6	1.04e-6	4.75e-6	4.57e-6	4.19e-6	7.23e-7
PA10_vel	1.15e-2	4.36e-5	9.63e-5	3.27e-5	1.12e-4	2.91e-5

Table E.7: Comparison of angle WMISE results for different controllers, platform trajectory: sin_r1p2pi2, PA10 trajectory: **`sine`**.

controller	S1	S2	E1	E2	W1	W2
PD	2.14e-5	2.11e-4	6.18e-5	1.64e-5	2.30e-5	1.38e-5
sat_sl	2.02e-5	2.08e-4	6.24e-5	1.56e-5	2.24e-5	1.32e-5
slid	2.01e-5	2.08e-4	1.73e-4	5.49e-6	2.05e-5	1.26e-5
PID	2.06e-5	2.12e-4	6.06e-5	1.59e-5	2.11e-5	1.28e-5
grv_frc_PD	8.75e-6	1.89e-5	2.24e-5	8.33e-7	2.34e-6	6.47e-7
grv_frc_PD_s_rp	5.64e-6	9.84e-6	4.05e-6	8.32e-7	2.40e-6	6.14e-7
grv_frc_PD_s_acc	3.67e-6	5.20e-6	5.35e-6	7.71e-7	2.45e-6	5.84e-7
grv_frc_PD_adap	1.33e-6	1.44e-6	1.87e-6	8.83e-7	2.44e-6	6.39e-7
grv_frc_sat_sl	9.13e-6	1.85e-5	2.36e-5	8.08e-7	2.51e-6	6.45e-7
grv_frc_sat_sl_s_rp	5.74e-6	1.01e-5	3.96e-6	7.67e-7	2.41e-6	5.50e-7
grv_frc_sat_sl_s_acc	3.98e-6	5.41e-6	5.31e-6	7.98e-7	2.46e-6	7.02e-7
grv_frc_sat_sl_adap	9.63e-7	9.15e-7	2.55e-6	8.08e-7	2.46e-6	6.17e-7
grv_frc_slid	8.95e-6	1.87e-5	6.16e-5	1.19e-6	2.61e-6	6.30e-7
grv_frc_slid_s_rp	5.75e-6	9.98e-6	1.07e-5	1.23e-6	2.48e-6	6.49e-7
grv_frc_slid_s_acc	3.91e-6	5.36e-6	1.44e-5	1.20e-6	2.45e-6	6.35e-7
grv_frc_slid_adap	1.44e-6	1.67e-6	4.19e-6	1.11e-6	2.35e-6	5.56e-7
lin_FULL	1.03e-5	1.80e-6	6.96e-6	3.11e-6	6.45e-6	4.05e-7
lin_noCC	1.30e-5	1.87e-6	7.94e-6	3.10e-6	6.65e-6	3.66e-7
lin_FULL_s_rp	8.14e-6	1.77e-6	3.06e-6	2.98e-6	5.95e-6	4.28e-7
lin_noCC_s_rp	9.14e-6	1.81e-6	3.49e-6	2.90e-6	6.00e-6	3.99e-7
lin_FULL_s_acc	4.30e-6	1.09e-6	3.43e-6	3.06e-6	5.94e-6	3.83e-7
lin_noCC_s_acc	5.20e-6	1.08e-6	3.97e-6	2.96e-6	6.12e-6	3.55e-7
W-B_FULL	5.76e-6	1.23e-5	2.10e-5	1.21e-6	2.44e-6	4.75e-7
W-B_noCC	6.58e-6	1.39e-5	2.58e-5	1.21e-6	2.35e-6	4.64e-7
W-B_FULL_s_rp	4.84e-6	8.62e-6	3.07e-6	1.47e-6	2.25e-6	5.92e-7
W-B_noCC_s_rp	5.08e-6	8.99e-6	4.40e-6	1.40e-6	2.34e-6	5.28e-7
W-B_FULL_s_acc	2.39e-6	3.59e-6	3.86e-6	1.40e-6	2.35e-6	5.18e-7
W-B_noCC_s_acc	2.64e-6	4.91e-6	5.57e-6	1.22e-6	2.42e-6	4.70e-7
W-B_FULL_adap	9.65e-7	1.37e-6	2.15e-6	1.61e-6	2.34e-6	5.04e-7
W-B_noCC_adap	1.07e-6	1.52e-6	1.98e-6	1.39e-6	2.43e-6	4.62e-7
S-L_FULL	6.27e-6	1.14e-5	5.20e-5	2.84e-6	2.49e-6	4.41e-7
S-L_noCC	6.41e-6	1.25e-5	6.71e-5	2.74e-6	2.58e-6	5.69e-7
S-L_FULL_s_rp	4.99e-6	8.33e-6	6.51e-6	2.45e-6	2.45e-6	4.68e-7
S-L_noCC_s_rp	5.51e-6	8.82e-6	9.79e-6	3.07e-6	2.52e-6	4.52e-7
S-L_FULL_s_acc	2.40e-6	3.34e-6	9.87e-6	2.78e-6	2.38e-6	5.27e-7
S-L_noCC_s_acc	3.00e-6	3.92e-6	1.50e-5	3.16e-6	2.63e-6	5.25e-7
S-L_FULL_adap	1.65e-6	2.07e-6	4.77e-6	2.78e-6	2.52e-6	4.94e-7
S-L_noCC_adap	1.94e-6	2.29e-6	4.25e-6	2.80e-6	2.51e-6	5.08e-7
PA10_vel	1.14e-2	3.16e-5	4.91e-5	1.70e-5	3.19e-5	1.56e-5

Table E.8: Comparison of angle WMISE results for different controllers, platform trajectory: sin_r1p2pi2, PA10 trajectory: **`ssine`**.

controller	S1	S2	E1	E2	W1	W2
PD	2.46e-5	2.97e-4	1.80e-4	1.98e-5	3.09e-5	1.44e-5
sat_sl	2.46e-5	2.96e-4	1.75e-4	1.92e-5	3.15e-5	1.43e-5
slid	2.43e-5	2.96e-4	4.76e-4	8.22e-6	3.14e-5	1.46e-5
PID	2.44e-5	2.96e-4	1.79e-4	2.05e-5	3.07e-5	1.45e-5
grv_frc_PD	5.41e-6	1.92e-5	2.04e-5	1.18e-6	3.62e-6	9.10e-7
grv_frc_PD_s_rp	4.85e-6	1.23e-5	1.60e-5	1.14e-6	3.66e-6	9.91e-7
grv_frc_PD_s_acc	4.00e-6	9.24e-6	1.34e-5	1.67e-6	4.01e-6	1.11e-6
grv_frc_PD_adap	1.92e-6	1.23e-6	2.64e-6	1.54e-6	3.47e-6	1.04e-6
grv_frc_sat_sl	5.64e-6	1.91e-5	1.84e-5	1.08e-6	4.06e-6	9.52e-7
grv_frc_sat_sl_s_rp	4.90e-6	1.25e-5	1.54e-5	1.14e-6	3.84e-6	1.01e-6
grv_frc_sat_sl_s_acc	3.81e-6	8.95e-6	1.24e-5	1.07e-6	3.47e-6	9.26e-7
grv_frc_sat_sl_adap	1.60e-6	7.47e-7	2.96e-6	1.17e-6	3.37e-6	9.59e-7
grv_frc_slid	5.59e-6	1.97e-5	5.22e-5	1.37e-6	3.69e-6	9.41e-7
grv_frc_slid_s_rp	4.95e-6	1.25e-5	4.02e-5	1.33e-6	3.95e-6	1.03e-6
grv_frc_slid_s_acc	3.74e-6	9.03e-6	3.30e-5	1.76e-6	3.53e-6	8.83e-7
grv_frc_slid_adap	2.21e-6	1.29e-6	6.14e-6	1.95e-6	3.69e-6	8.77e-7
lin_FULL	7.16e-6	2.10e-6	2.00e-6	4.87e-6	8.84e-6	6.19e-7
lin_noCC	6.59e-6	2.09e-6	2.41e-6	4.69e-6	8.27e-6	6.11e-7
lin_FULL_s_rp	6.39e-6	1.39e-6	2.28e-6	4.85e-6	8.15e-6	5.95e-7
lin_noCC_s_rp	6.31e-6	1.40e-6	2.79e-6	4.82e-6	8.11e-6	5.14e-7
lin_FULL_s_acc	4.20e-6	1.06e-6	1.95e-6	4.96e-6	7.67e-6	5.43e-7
lin_noCC_s_acc	4.62e-6	1.14e-6	2.71e-6	4.88e-6	8.20e-6	5.40e-7
W-B_FULL	3.29e-6	1.53e-5	1.18e-5	1.67e-6	3.40e-6	6.77e-7
W-B_noCC	3.07e-6	1.53e-5	1.29e-5	1.74e-6	3.41e-6	6.84e-7
W-B_FULL_s_rp	2.79e-6	8.33e-6	8.08e-6	2.01e-6	3.29e-6	6.16e-7
W-B_noCC_s_rp	2.99e-6	8.27e-6	8.87e-6	1.49e-6	3.50e-6	6.94e-7
W-B_FULL_s_acc	1.97e-6	5.37e-6	5.37e-6	1.63e-6	3.41e-6	7.41e-7
W-B_noCC_s_acc	2.36e-6	5.54e-6	6.86e-6	2.48e-6	4.12e-6	9.67e-7
W-B_FULL_adap	1.14e-6	9.84e-7	1.94e-6	2.32e-6	3.26e-6	7.17e-7
W-B_noCC_adap	1.59e-6	9.57e-7	2.18e-6	1.68e-6	3.28e-6	8.38e-7
S-L_FULL	3.54e-6	1.31e-5	2.45e-5	4.20e-6	3.29e-6	7.41e-7
S-L_noCC	2.87e-6	1.47e-5	3.00e-5	3.67e-6	3.33e-6	6.23e-7
S-L_FULL_s_rp	3.07e-6	8.09e-6	1.77e-5	3.84e-6	3.59e-6	6.68e-7
S-L_noCC_s_rp	3.07e-6	7.65e-6	2.01e-5	4.26e-6	3.90e-6	8.18e-7
S-L_FULL_s_acc	2.11e-6	5.13e-6	1.21e-5	3.27e-6	3.79e-6	6.39e-7
S-L_noCC_s_acc	2.07e-6	4.92e-6	1.48e-5	4.02e-6	3.76e-6	6.27e-7
S-L_FULL_adap	1.32e-6	1.14e-6	3.71e-6	4.01e-6	2.97e-6	5.81e-7
S-L_noCC_adap	1.61e-6	1.09e-6	4.98e-6	3.41e-6	3.14e-6	6.25e-7
PA10_vel	1.13e-2	4.78e-5	6.29e-5	3.24e-5	1.26e-4	2.72e-5

Table E.9: Comparison of angle WMISE results for different controllers, platform trajectory: sin2_s20, PA10 trajectory: **`sine`**.

controller	S1	S2	E1	E2	W1	W2
PD	2.22e-5	1.53e-4	5.13e-5	1.55e-5	2.17e-5	1.24e-5
sat_sl	2.16e-5	1.53e-4	4.91e-5	1.48e-5	2.20e-5	1.21e-5
slid	2.16e-5	1.54e-4	1.38e-4	5.16e-6	2.17e-5	1.26e-5
PID	2.19e-5	1.55e-4	5.09e-5	1.57e-5	2.18e-5	1.29e-5
grv_frc_PD	4.88e-6	1.09e-5	1.46e-5	8.38e-7	1.71e-6	4.94e-7
grv_frc_PD_s_rp	3.71e-6	6.20e-6	8.11e-6	9.20e-7	1.75e-6	6.86e-7
grv_frc_PD_s_acc	2.98e-6	4.34e-6	6.01e-6	8.72e-7	1.81e-6	6.29e-7
grv_frc_PD_adap	1.18e-6	1.19e-6	2.96e-6	9.63e-7	1.80e-6	6.48e-7
grv_frc_sat_sl	5.26e-6	1.15e-5	1.35e-5	8.69e-7	2.01e-6	6.67e-7
grv_frc_sat_sl_s_rp	3.92e-6	6.42e-6	7.40e-6	7.96e-7	1.69e-6	6.12e-7
grv_frc_sat_sl_s_acc	3.20e-6	4.89e-6	5.49e-6	7.87e-7	1.60e-6	6.28e-7
grv_frc_sat_sl_adap	1.01e-6	7.61e-7	3.86e-6	8.56e-7	1.87e-6	5.75e-7
grv_frc_slid	5.11e-6	1.12e-5	3.66e-5	1.16e-6	1.73e-6	7.00e-7
grv_frc_slid_s_rp	3.87e-6	6.37e-6	1.97e-5	1.09e-6	1.79e-6	6.45e-7
grv_frc_slid_s_acc	3.19e-6	4.59e-6	1.46e-5	1.13e-6	1.81e-6	6.00e-7
grv_frc_slid_adap	1.32e-6	1.34e-6	7.14e-6	1.17e-6	2.07e-6	6.27e-7
lin_FULL	9.55e-6	2.24e-6	5.54e-6	3.23e-6	4.39e-6	4.42e-7
lin_noCC	9.80e-6	2.15e-6	5.79e-6	3.24e-6	4.43e-6	4.99e-7
lin_FULL_s_rp	7.50e-6	1.67e-6	4.67e-6	3.27e-6	4.40e-6	4.74e-7
lin_noCC_s_rp	7.73e-6	1.58e-6	4.91e-6	3.21e-6	4.43e-6	3.61e-7
lin_FULL_s_acc	5.74e-6	1.69e-6	4.98e-6	3.44e-6	4.41e-6	5.87e-7
lin_noCC_s_acc	5.50e-6	1.43e-6	5.01e-6	3.15e-6	4.31e-6	3.97e-7
W-B_FULL	3.78e-6	1.34e-5	1.58e-5	1.63e-6	1.70e-6	5.10e-7
W-B_noCC	3.64e-6	1.20e-5	1.55e-5	1.27e-6	1.72e-6	5.97e-7
W-B_FULL_s_rp	2.99e-6	7.25e-6	9.13e-6	1.49e-6	1.92e-6	5.80e-7
W-B_noCC_s_rp	3.09e-6	6.55e-6	8.99e-6	1.52e-6	1.80e-6	5.66e-7
W-B_FULL_s_acc	2.24e-6	5.02e-6	6.26e-6	1.39e-6	1.90e-6	6.06e-7
W-B_noCC_s_acc	2.19e-6	4.32e-6	6.50e-6	1.26e-6	1.85e-6	5.51e-7
W-B_FULL_adap	1.16e-6	1.24e-6	3.28e-6	1.63e-6	1.77e-6	5.61e-7
W-B_noCC_adap	1.20e-6	1.24e-6	3.20e-6	1.49e-6	1.53e-6	6.24e-7
S-L_FULL	3.60e-6	1.24e-5	3.62e-5	2.68e-6	1.61e-6	5.05e-7
S-L_noCC	3.74e-6	1.18e-5	3.59e-5	2.54e-6	1.85e-6	5.27e-7
S-L_FULL_s_rp	2.88e-6	6.90e-6	2.02e-5	2.90e-6	1.72e-6	5.81e-7
S-L_noCC_s_rp	2.99e-6	6.36e-6	1.99e-5	2.83e-6	1.74e-6	6.07e-7
S-L_FULL_s_acc	2.20e-6	4.79e-6	1.51e-5	2.97e-6	1.96e-6	5.63e-7
S-L_noCC_s_acc	2.20e-6	4.16e-6	1.55e-5	2.75e-6	1.96e-6	4.95e-7
S-L_FULL_adap	1.78e-6	1.40e-6	7.58e-6	3.58e-6	1.56e-6	5.25e-7
S-L_noCC_adap	1.72e-6	1.38e-6	7.25e-6	2.69e-6	1.71e-6	5.39e-7
PA10_vel	1.15e-2	3.02e-5	2.97e-5	1.68e-5	3.35e-5	1.57e-5

Table E.10: Comparison of angle WMISE results for different controllers, platform trajectory: sin2_s20, PA10 trajectory: **ssine**.

controller	S1	S2	E1	E2	W1	W2
PD	2.76e-5	2.55e-4	1.62e-4	2.14e-5	2.89e-5	1.51e-5
sat_sl	2.81e-5	2.56e-4	1.61e-4	1.82e-5	2.92e-5	1.52e-5
slid	2.80e-5	2.55e-4	4.28e-4	7.91e-6	2.92e-5	1.49e-5
PID	2.74e-5	2.55e-4	1.61e-4	1.99e-5	2.95e-5	1.49e-5
grv_frc_PD	1.41e-5	1.12e-5	1.14e-5	1.68e-6	4.17e-6	8.60e-7
grv_frc_PD_s_rp	5.38e-6	8.24e-6	1.19e-5	1.29e-6	4.20e-6	1.00e-6
grv_frc_PD_s_acc	4.98e-6	6.62e-6	1.05e-5	1.48e-6	4.07e-6	1.01e-6
grv_frc_PD_adap	1.55e-6	1.03e-6	2.84e-6	1.62e-6	4.08e-6	9.48e-7
grv_frc_sat_sl	1.46e-5	1.13e-5	1.08e-5	1.43e-6	4.55e-6	1.08e-6
grv_frc_sat_sl_s_rp	5.35e-6	8.08e-6	1.17e-5	1.16e-6	4.34e-6	8.69e-7
grv_frc_sat_sl_s_acc	4.61e-6	6.99e-6	9.88e-6	1.18e-6	4.27e-6	8.79e-7
grv_frc_sat_sl_adap	1.06e-6	5.80e-7	3.13e-6	1.23e-6	4.14e-6	8.93e-7
grv_frc_slid	1.45e-5	1.13e-5	2.83e-5	1.38e-6	4.24e-6	8.09e-7
grv_frc_slid_s_rp	5.37e-6	8.08e-6	2.95e-5	2.02e-6	4.14e-6	8.87e-7
grv_frc_slid_s_acc	4.52e-6	6.70e-6	2.49e-5	1.47e-6	4.26e-6	8.85e-7
grv_frc_slid_adap	1.43e-6	1.03e-6	6.54e-6	1.57e-6	3.97e-6	8.18e-7
lin_FULL	9.43e-6	1.94e-6	1.89e-6	5.72e-6	1.03e-5	4.89e-7
lin_noCC	1.02e-5	1.96e-6	2.04e-6	5.70e-6	1.04e-5	4.94e-7
lin_FULL_s_rp	5.06e-6	1.17e-6	2.19e-6	5.56e-6	1.01e-5	4.99e-7
lin_noCC_s_rp	5.27e-6	1.19e-6	2.42e-6	5.58e-6	1.02e-5	4.89e-7
lin_FULL_s_acc	3.62e-6	1.11e-6	2.18e-6	5.85e-6	1.04e-5	5.99e-7
lin_noCC_s_acc	3.83e-6	1.08e-6	2.57e-6	5.60e-6	1.02e-5	4.79e-7
W-B_FULL	8.15e-6	1.23e-5	6.74e-6	1.86e-6	4.25e-6	6.12e-7
W-B_noCC	8.89e-6	1.15e-5	6.40e-6	2.10e-6	4.30e-6	6.43e-7
W-B_FULL_s_rp	3.67e-6	5.59e-6	4.59e-6	2.19e-6	4.45e-6	5.70e-7
W-B_noCC_s_rp	3.68e-6	5.37e-6	4.99e-6	2.16e-6	4.48e-6	6.98e-7
W-B_FULL_s_acc	2.16e-6	4.71e-6	3.61e-6	1.97e-6	4.36e-6	6.55e-7
W-B_noCC_s_acc	2.49e-6	3.96e-6	4.14e-6	1.97e-6	4.28e-6	6.17e-7
W-B_FULL_adap	8.79e-7	7.66e-7	2.22e-6	1.88e-6	4.31e-6	6.96e-7
W-B_noCC_adap	9.53e-7	7.96e-7	2.39e-6	1.94e-6	4.18e-6	7.43e-7
S-L_FULL	8.53e-6	1.20e-5	1.49e-5	4.10e-6	4.50e-6	7.63e-7
S-L_noCC	8.97e-6	1.13e-5	1.45e-5	3.65e-6	4.67e-6	6.66e-7
S-L_FULL_s_rp	3.55e-6	5.27e-6	1.06e-5	3.92e-6	4.72e-6	7.30e-7
S-L_noCC_s_rp	3.63e-6	5.34e-6	1.17e-5	3.35e-6	4.91e-6	7.96e-7
S-L_FULL_s_acc	2.44e-6	4.39e-6	8.22e-6	5.36e-6	5.04e-6	8.21e-7
S-L_noCC_s_acc	2.57e-6	4.15e-6	9.97e-6	3.41e-6	4.76e-6	7.32e-7
S-L_FULL_adap	1.33e-6	1.01e-6	5.29e-6	3.84e-6	4.64e-6	7.63e-7
S-L_noCC_adap	1.40e-6	1.13e-6	6.33e-6	3.22e-6	4.67e-6	8.53e-7
PA10_vel	1.14e-2	4.87e-5	5.13e-5	3.15e-5	1.42e-4	3.22e-5

Table E.11: Comparison of angle WMISE results for different controllers, platform trajectory: sin2_s20_2, PA10 trajectory: **`sine`**.

controller	S1	S2	E1	E2	W1	W2
PD	1.79e-5	1.52e-4	3.65e-5	1.55e-5	2.02e-5	1.29e-5
sat_sl	1.78e-5	1.53e-4	3.45e-5	1.43e-5	2.04e-5	1.33e-5
slid	1.74e-5	1.52e-4	9.77e-5	5.24e-6	2.00e-5	1.30e-5
PID	1.83e-5	1.51e-4	3.61e-5	1.48e-5	2.04e-5	1.28e-5
grv_frc_PD	9.52e-6	6.09e-6	1.20e-5	9.46e-7	2.19e-6	5.80e-7
grv_frc_PD_s_rp	4.07e-6	4.51e-6	7.30e-6	1.01e-6	2.25e-6	6.53e-7
grv_frc_PD_s_acc	3.42e-6	3.35e-6	5.30e-6	1.01e-6	2.27e-6	6.96e-7
grv_frc_PD_adap	1.62e-6	1.28e-6	3.08e-6	1.07e-6	2.29e-6	7.02e-7
grv_frc_sat_sl	9.23e-6	6.20e-6	1.11e-5	8.53e-7	2.25e-6	5.66e-7
grv_frc_sat_sl_s_rp	3.86e-6	4.57e-6	6.87e-6	8.79e-7	2.35e-6	5.81e-7
grv_frc_sat_sl_s_acc	3.30e-6	3.06e-6	5.32e-6	8.54e-7	2.24e-6	5.71e-7
grv_frc_sat_sl_adap	1.16e-6	6.47e-7	3.57e-6	9.49e-7	2.29e-6	5.32e-7
grv_frc_slid	1.00e-5	6.42e-6	3.10e-5	1.27e-6	2.35e-6	6.65e-7
grv_frc_slid_s_rp	3.66e-6	4.59e-6	1.76e-5	1.09e-6	2.22e-6	5.52e-7
grv_frc_slid_s_acc	3.18e-6	3.24e-6	1.27e-5	1.04e-6	2.22e-6	4.97e-7
grv_frc_slid_adap	1.39e-6	1.26e-6	6.89e-6	1.10e-6	2.15e-6	5.90e-7
lin_FULL	1.47e-5	1.63e-6	4.93e-6	3.53e-6	5.86e-6	3.47e-7
lin_noCC	1.50e-5	1.52e-6	5.23e-6	3.41e-6	6.04e-6	3.67e-7
lin_FULL_s_rp	7.46e-6	1.31e-6	4.65e-6	3.65e-6	5.87e-6	3.74e-7
lin_noCC_s_rp	7.38e-6	1.29e-6	4.99e-6	3.76e-6	6.04e-6	4.04e-7
lin_FULL_s_acc	5.59e-6	1.32e-6	4.63e-6	3.83e-6	6.01e-6	3.99e-7
lin_noCC_s_acc	5.29e-6	1.20e-6	4.91e-6	3.81e-6	6.10e-6	3.57e-7
W-B_FULL	7.21e-6	9.69e-6	1.37e-5	1.50e-6	2.40e-6	5.11e-7
W-B_noCC	7.20e-6	8.63e-6	1.29e-5	1.42e-6	2.54e-6	4.96e-7
W-B_FULL_s_rp	3.31e-6	4.76e-6	7.38e-6	1.37e-6	2.41e-6	4.28e-7
W-B_noCC_s_rp	3.39e-6	4.37e-6	7.73e-6	1.45e-6	2.48e-6	4.98e-7
W-B_FULL_s_acc	2.41e-6	3.77e-6	5.47e-6	1.77e-6	2.52e-6	5.47e-7
W-B_noCC_s_acc	2.37e-6	3.39e-6	5.78e-6	1.64e-6	2.56e-6	5.24e-7
W-B_FULL_adap	1.41e-6	1.30e-6	3.34e-6	1.88e-6	2.62e-6	6.49e-7
W-B_noCC_adap	1.35e-6	1.23e-6	3.05e-6	1.50e-6	2.51e-6	5.86e-7
S-L_FULL	7.54e-6	8.92e-6	3.26e-5	2.62e-6	2.44e-6	5.55e-7
S-L_noCC	7.52e-6	8.12e-6	3.05e-5	2.93e-6	2.68e-6	5.43e-7
S-L_FULL_s_rp	3.31e-6	4.40e-6	1.78e-5	2.90e-6	2.57e-6	6.08e-7
S-L_noCC_s_rp	3.66e-6	4.12e-6	1.89e-5	2.68e-6	2.67e-6	5.89e-7
S-L_FULL_s_acc	2.50e-6	3.64e-6	1.28e-5	2.62e-6	2.56e-6	5.90e-7
S-L_noCC_s_acc	2.42e-6	3.36e-6	1.46e-5	2.82e-6	2.70e-6	5.09e-7
S-L_FULL_adap	1.78e-6	1.35e-6	8.74e-6	2.62e-6	2.69e-6	5.84e-7
S-L_noCC_adap	1.80e-6	1.40e-6	8.55e-6	2.63e-6	2.77e-6	6.21e-7
PA10_vel	1.15e-2	3.01e-5	2.32e-5	1.68e-5	3.11e-5	1.60e-5

Table E.12: Comparison of angle WMISE results for different controllers, platform trajectory: sin2_s20_2, PA10 trajectory: **ssine**.

controller	S1	S2	E1	E2	W1	W2
PD	2.29e-5	2.75e-4	1.57e-4	2.35e-5	3.36e-5	1.29e-5
sat_sl	2.30e-5	2.75e-4	1.55e-4	1.95e-5	3.33e-5	1.39e-5
slid	2.33e-5	2.75e-4	4.19e-4	6.91e-6	3.31e-5	1.34e-5
PID	2.31e-5	2.75e-4	1.57e-4	2.27e-5	3.36e-5	1.33e-5
grv_frc_PD	3.45e-6	5.78e-6	9.62e-6	1.29e-6	2.97e-6	1.07e-6
grv_frc_PD_s_rp	3.80e-6	5.87e-6	1.01e-5	1.26e-6	3.40e-6	1.38e-6
grv_frc_PD_s_acc	3.76e-6	5.50e-6	9.76e-6	1.37e-6	2.96e-6	1.23e-6
grv_frc_PD_adap	1.37e-6	8.80e-7	2.39e-6	1.35e-6	2.81e-6	1.11e-6
grv_frc_sat_sl	3.63e-6	6.27e-6	1.02e-5	1.30e-6	3.69e-6	1.16e-6
grv_frc_sat_sl_s_rp	4.16e-6	6.16e-6	1.11e-5	1.48e-6	3.92e-6	1.61e-6
grv_frc_sat_sl_s_acc	3.58e-6	5.70e-6	1.04e-5	1.33e-6	3.61e-6	1.32e-6
grv_frc_sat_sl_adap	1.27e-6	8.08e-7	3.72e-6	1.53e-6	3.59e-6	1.46e-6
grv_frc_slid	3.61e-6	6.10e-6	2.62e-5	1.36e-6	3.14e-6	1.10e-6
grv_frc_slid_s_rp	3.96e-6	5.82e-6	2.59e-5	1.84e-6	3.05e-6	1.21e-6
grv_frc_slid_s_acc	3.79e-6	5.71e-6	2.58e-5	1.66e-6	3.05e-6	1.27e-6
grv_frc_slid_adap	1.51e-6	1.00e-6	6.29e-6	1.49e-6	3.13e-6	1.13e-6
lin_FULL	3.19e-6	8.51e-7	1.90e-6	4.86e-6	6.66e-6	7.43e-7
lin_noCC	3.79e-6	8.09e-7	2.46e-6	4.81e-6	6.40e-6	7.25e-7
lin_FULL_s_rp	3.18e-6	7.21e-7	1.95e-6	5.01e-6	6.58e-6	7.07e-7
lin_noCC_s_rp	4.02e-6	7.07e-7	2.40e-6	4.98e-6	6.66e-6	6.93e-7
lin_FULL_s_acc	3.44e-6	8.37e-7	2.01e-6	5.24e-6	7.11e-6	7.84e-7
lin_noCC_s_acc	3.90e-6	7.28e-7	2.50e-6	5.12e-6	6.60e-6	6.36e-7
W-B_FULL	1.49e-6	3.00e-6	3.04e-6	1.81e-6	2.85e-6	9.29e-7
W-B_noCC	1.94e-6	2.84e-6	4.26e-6	2.18e-6	2.95e-6	8.43e-7
W-B_FULL_s_rp	1.57e-6	2.45e-6	2.79e-6	2.33e-6	3.20e-6	1.02e-6
W-B_noCC_s_rp	1.85e-6	2.14e-6	3.87e-6	1.57e-6	3.15e-6	9.39e-7
W-B_FULL_s_acc	1.93e-6	3.04e-6	3.00e-6	1.73e-6	3.16e-6	1.04e-6
W-B_noCC_s_acc	1.95e-6	2.11e-6	3.89e-6	1.70e-6	3.07e-6	8.02e-7
W-B_FULL_adap	7.79e-7	6.42e-7	1.73e-6	1.53e-6	3.19e-6	8.54e-7
W-B_noCC_adap	8.96e-7	6.69e-7	1.85e-6	1.69e-6	3.11e-6	9.09e-7
S-L_FULL	1.44e-6	2.90e-6	6.59e-6	4.07e-6	3.46e-6	1.12e-6
S-L_noCC	1.80e-6	2.62e-6	9.30e-6	4.49e-6	3.40e-6	1.01e-6
S-L_FULL_s_rp	1.45e-6	2.14e-6	5.89e-6	3.61e-6	3.42e-6	1.14e-6
S-L_noCC_s_rp	1.87e-6	1.98e-6	9.43e-6	3.61e-6	3.29e-6	9.07e-7
S-L_FULL_s_acc	1.53e-6	2.38e-6	5.99e-6	3.32e-6	3.57e-6	8.83e-7
S-L_noCC_s_acc	1.91e-6	2.19e-6	8.44e-6	4.09e-6	3.35e-6	1.02e-6
S-L_FULL_adap	1.06e-6	9.18e-7	4.44e-6	3.64e-6	3.45e-6	1.11e-6
S-L_noCC_adap	1.26e-6	1.07e-6	5.07e-6	3.60e-6	3.42e-6	8.99e-7
PA10_vel	1.14e-2	5.24e-5	5.13e-5	3.86e-5	2.02e-4	3.19e-5

Table E.13: Comparison of angle WMISE results for different controllers, platform trajectory: sea_all_2, PA10 trajectory: **`sine`**.

controller	S1	S2	E1	E2	W1	W2
PD	2.20e-5	1.60e-4	3.55e-5	1.57e-5	2.28e-5	1.14e-5
sat_sl	2.12e-5	1.58e-4	3.45e-5	1.51e-5	2.31e-5	1.17e-5
slid	2.16e-5	1.58e-4	1.00e-4	5.12e-6	2.30e-5	1.15e-5
PID	2.15e-5	1.59e-4	3.58e-5	1.57e-5	2.30e-5	1.12e-5
grv_frc_PD	2.54e-6	2.62e-6	4.43e-6	8.64e-7	1.46e-6	7.66e-7
grv_frc_PD_s_rp	2.55e-6	2.37e-6	4.04e-6	9.17e-7	1.54e-6	9.08e-7
grv_frc_PD_s_acc	2.51e-6	2.38e-6	3.85e-6	9.22e-7	1.53e-6	8.67e-7
grv_frc_PD_adap	9.51e-7	1.14e-6	2.72e-6	9.04e-7	1.48e-6	8.35e-7
grv_frc_sat_sl	2.61e-6	2.97e-6	4.57e-6	1.03e-6	1.83e-6	8.75e-7
grv_frc_sat_sl_s_rp	2.85e-6	2.86e-6	4.36e-6	1.02e-6	1.77e-6	1.05e-6
grv_frc_sat_sl_s_acc	2.82e-6	2.81e-6	4.07e-6	1.01e-6	1.73e-6	1.02e-6
grv_frc_sat_sl_adap	9.58e-7	8.68e-7	4.02e-6	1.12e-6	1.83e-6	1.06e-6
grv_frc_slid	2.60e-6	2.66e-6	1.18e-5	1.28e-6	1.59e-6	7.53e-7
grv_frc_slid_s_rp	2.67e-6	2.49e-6	1.06e-5	1.28e-6	1.55e-6	8.78e-7
grv_frc_slid_s_acc	2.82e-6	2.63e-6	1.06e-5	1.31e-6	1.62e-6	9.59e-7
grv_frc_slid_adap	1.11e-6	1.31e-6	6.91e-6	1.12e-6	1.53e-6	7.92e-7
lin_FULL	3.53e-6	1.13e-6	4.39e-6	3.30e-6	3.60e-6	5.61e-7
lin_noCC	4.05e-6	1.05e-6	4.52e-6	3.28e-6	3.58e-6	6.12e-7
lin_FULL_s_rp	3.93e-6	1.10e-6	4.29e-6	3.45e-6	3.82e-6	6.19e-7
lin_noCC_s_rp	4.31e-6	9.86e-7	4.42e-6	3.48e-6	3.77e-6	5.84e-7
lin_FULL_s_acc	3.95e-6	1.13e-6	4.32e-6	3.46e-6	3.88e-6	6.28e-7
lin_noCC_s_acc	4.22e-6	1.03e-6	4.45e-6	3.42e-6	3.89e-6	5.39e-7
W-B_FULL	1.71e-6	2.71e-6	4.43e-6	1.47e-6	1.62e-6	9.43e-7
W-B_noCC	1.68e-6	2.22e-6	4.97e-6	1.50e-6	1.60e-6	6.42e-7
W-B_FULL_s_rp	1.82e-6	2.24e-6	3.80e-6	1.46e-6	1.61e-6	7.82e-7
W-B_noCC_s_rp	1.75e-6	1.99e-6	4.63e-6	1.45e-6	1.59e-6	7.80e-7
W-B_FULL_s_acc	1.59e-6	2.36e-6	3.70e-6	1.35e-6	1.59e-6	7.73e-7
W-B_noCC_s_acc	1.77e-6	1.98e-6	4.52e-6	1.37e-6	1.60e-6	6.70e-7
W-B_FULL_adap	6.96e-7	9.71e-7	2.57e-6	1.63e-6	1.79e-6	7.20e-7
W-B_noCC_adap	7.34e-7	1.01e-6	2.40e-6	1.46e-6	1.65e-6	7.82e-7
S-L_FULL	1.36e-6	2.42e-6	9.52e-6	2.51e-6	1.75e-6	7.68e-7
S-L_noCC	1.55e-6	2.28e-6	1.14e-5	2.83e-6	1.87e-6	7.82e-7
S-L_FULL_s_rp	1.59e-6	2.10e-6	8.81e-6	2.42e-6	1.77e-6	8.78e-7
S-L_noCC_s_rp	1.64e-6	1.71e-6	1.09e-5	2.76e-6	1.92e-6	8.40e-7
S-L_FULL_s_acc	1.58e-6	2.32e-6	8.21e-6	2.63e-6	1.84e-6	7.68e-7
S-L_noCC_s_acc	1.61e-6	1.96e-6	1.00e-5	2.78e-6	1.90e-6	9.10e-7
S-L_FULL_adap	1.01e-6	1.23e-6	6.54e-6	2.49e-6	1.84e-6	8.51e-7
S-L_noCC_adap	1.13e-6	1.38e-6	6.42e-6	3.16e-6	1.93e-6	7.16e-7
PA10_vel	1.13e-2	4.18e-5	2.61e-5	2.13e-5	3.28e-5	2.03e-5

Table E.14: Comparison of angle WMISE results for different controllers, platform trajectory: sea_all_2, PA10 trajectory: **`ssine`**.

Appendix F

Tables with results of manipulator on platform control experiments (torque noise power ratio)

The results of high to low frequency control torque power ratio are presented in tables: F.1 to F.14.

controller	S1	S2	E1	E2	W1	W2
PD	2.20e-5	2.76e-4	1.58e-4	2.14e-5	3.21e-5	1.32e-5
sat_sl	2.21e-5	2.76e-4	1.59e-4	1.85e-5	3.18e-5	1.32e-5
slid	2.20e-5	2.76e-4	4.27e-4	6.13e-6	3.21e-5	1.35e-5
PID	2.16e-5	2.74e-4	1.58e-4	2.15e-5	3.24e-5	1.37e-5
grv_frc_PD	3.81e-6	6.02e-6	1.08e-5	1.67e-6	3.70e-6	1.38e-6
grv_frc_PD_s_rp	3.70e-6	5.84e-6	1.05e-5	1.54e-6	3.38e-6	1.22e-6
grv_frc_PD_s_acc	3.65e-6	5.73e-6	1.02e-5	1.30e-6	3.37e-6	1.12e-6
grv_frc_PD_adap	1.31e-6	1.03e-6	2.59e-6	1.48e-6	3.44e-6	1.28e-6
grv_frc_sat_sl	3.78e-6	5.82e-6	1.06e-5	1.31e-6	3.44e-6	1.21e-6
grv_frc_sat_sl_s_rp	3.71e-6	5.92e-6	1.09e-5	1.28e-6	3.37e-6	1.14e-6
grv_frc_sat_sl_s_acc	3.68e-6	5.54e-6	1.03e-5	1.23e-6	3.42e-6	1.16e-6
grv_frc_sat_sl_adap	1.17e-6	6.64e-7	3.27e-6	1.32e-6	3.45e-6	1.28e-6
grv_frc_slid	3.89e-6	6.03e-6	2.78e-5	1.52e-6	3.49e-6	1.27e-6
grv_frc_slid_s_rp	3.83e-6	5.83e-6	2.86e-5	1.68e-6	3.64e-6	1.18e-6
grv_frc_slid_s_acc	3.83e-6	6.22e-6	2.88e-5	1.87e-6	3.66e-6	1.30e-6
grv_frc_slid_adap	1.52e-6	1.15e-6	6.38e-6	1.69e-6	3.42e-6	1.20e-6
lin_FULL	2.89e-6	7.89e-7	2.17e-6	5.89e-6	7.83e-6	6.76e-7
lin_noCC	2.84e-6	7.74e-7	2.48e-6	5.28e-6	7.97e-6	7.01e-7
lin_FULL_s_rp	2.66e-6	7.84e-7	1.91e-6	5.27e-6	7.41e-6	7.77e-7
lin_noCC_s_rp	3.32e-6	8.08e-7	2.55e-6	5.44e-6	7.79e-6	8.33e-7
lin_FULL_s_acc	2.84e-6	9.43e-7	2.30e-6	5.73e-6	7.79e-6	8.09e-7
lin_noCC_s_acc	3.46e-6	9.06e-7	2.74e-6	5.68e-6	7.49e-6	7.41e-7
W-B_FULL	1.54e-6	2.61e-6	2.70e-6	2.42e-6	3.26e-6	9.46e-7
W-B_noCC	1.91e-6	2.70e-6	3.90e-6	1.75e-6	3.46e-6	1.04e-6
W-B_FULL_s_rp	1.63e-6	2.64e-6	2.87e-6	1.97e-6	3.39e-6	1.16e-6
W-B_noCC_s_rp	2.02e-6	2.30e-6	4.10e-6	2.54e-6	3.43e-6	1.05e-6
W-B_FULL_s_acc	1.74e-6	2.67e-6	3.31e-6	2.17e-6	3.28e-6	1.00e-6
W-B_noCC_s_acc	1.85e-6	2.60e-6	4.08e-6	2.02e-6	3.38e-6	8.96e-7
W-B_FULL_adap	9.39e-7	6.91e-7	2.02e-6	1.84e-6	3.38e-6	1.09e-6
W-B_noCC_adap	9.78e-7	9.57e-7	2.27e-6	2.50e-6	3.43e-6	9.40e-7
S-L_FULL	1.64e-6	2.46e-6	7.26e-6	4.17e-6	3.36e-6	9.98e-7
S-L_noCC	2.14e-6	2.31e-6	1.03e-5	4.21e-6	3.71e-6	1.07e-6
S-L_FULL_s_rp	1.78e-6	2.89e-6	7.71e-6	4.85e-6	3.79e-6	9.77e-7
S-L_noCC_s_rp	1.83e-6	2.48e-6	1.03e-5	4.27e-6	3.59e-6	9.51e-7
S-L_FULL_s_acc	1.80e-6	2.47e-6	9.34e-6	4.14e-6	3.54e-6	1.08e-6
S-L_noCC_s_acc	1.91e-6	2.28e-6	1.20e-5	4.10e-6	3.58e-6	9.91e-7
S-L_FULL_adap	1.03e-6	1.22e-6	5.22e-6	4.02e-6	3.36e-6	9.14e-7
S-L_noCC_adap	1.35e-6	1.14e-6	6.74e-6	4.62e-6	3.72e-6	1.14e-6
PA10_vel	1.12e-2	4.51e-5	4.79e-5	3.38e-5	1.59e-4	3.37e-5

Table F.1: Comparison of high to low frequency control torque power ratio for different controllers, platform trajectory: zero, PA10 trajectory: **`sine`**.

controller	S1	S2	E1	E2	W1	W2
PD	2.06e-5	1.57e-4	3.60e-5	1.49e-5	2.20e-5	1.12e-5
sat_sl	2.03e-5	1.57e-4	3.49e-5	1.42e-5	2.23e-5	1.16e-5
slid	2.04e-5	1.56e-4	9.71e-5	5.11e-6	2.21e-5	1.14e-5
PID	2.07e-5	1.57e-4	3.58e-5	1.50e-5	2.24e-5	1.14e-5
grv_frc_PD	2.68e-6	2.57e-6	3.91e-6	9.32e-7	1.77e-6	8.16e-7
grv_frc_PD_s_rp	2.80e-6	2.55e-6	4.15e-6	1.02e-6	1.88e-6	8.86e-7
grv_frc_PD_s_acc	2.78e-6	2.59e-6	4.04e-6	9.93e-7	1.76e-6	8.09e-7
grv_frc_PD_adap	1.15e-6	1.25e-6	2.90e-6	1.10e-6	1.87e-6	9.45e-7
grv_frc_sat_sl	2.75e-6	2.65e-6	3.99e-6	9.55e-7	1.69e-6	8.05e-7
grv_frc_sat_sl_s_rp	2.81e-6	2.69e-6	4.11e-6	9.62e-7	1.81e-6	8.99e-7
grv_frc_sat_sl_s_acc	2.78e-6	2.52e-6	4.09e-6	9.37e-7	1.75e-6	8.26e-7
grv_frc_sat_sl_adap	9.10e-7	7.09e-7	3.89e-6	1.02e-6	1.85e-6	9.54e-7
grv_frc_slid	2.84e-6	2.72e-6	1.07e-5	1.15e-6	1.96e-6	9.55e-7
grv_frc_slid_s_rp	3.03e-6	2.72e-6	1.09e-5	1.27e-6	1.87e-6	8.73e-7
grv_frc_slid_s_acc	2.85e-6	2.52e-6	1.06e-5	1.18e-6	1.86e-6	8.90e-7
grv_frc_slid_adap	1.40e-6	1.49e-6	7.18e-6	1.19e-6	1.80e-6	8.90e-7
lin_FULL	4.15e-6	1.20e-6	4.56e-6	4.22e-6	4.61e-6	5.74e-7
lin_noCC	3.79e-6	1.12e-6	4.57e-6	3.74e-6	4.37e-6	6.26e-7
lin_FULL_s_rp	3.78e-6	1.13e-6	4.33e-6	3.87e-6	4.30e-6	6.65e-7
lin_noCC_s_rp	4.07e-6	1.14e-6	4.76e-6	3.78e-6	4.37e-6	6.32e-7
lin_FULL_s_acc	3.90e-6	1.35e-6	4.75e-6	4.29e-6	4.50e-6	6.86e-7
lin_noCC_s_acc	4.11e-6	1.28e-6	4.92e-6	4.21e-6	4.39e-6	6.73e-7
W-B_FULL	1.66e-6	2.65e-6	3.62e-6	1.44e-6	1.80e-6	7.52e-7
W-B_noCC	1.83e-6	2.60e-6	4.44e-6	1.55e-6	1.91e-6	8.76e-7
W-B_FULL_s_rp	1.70e-6	2.38e-6	3.78e-6	1.83e-6	1.85e-6	1.02e-6
W-B_noCC_s_rp	1.88e-6	2.18e-6	4.62e-6	1.81e-6	1.87e-6	8.33e-7
W-B_FULL_s_acc	1.88e-6	2.66e-6	3.73e-6	1.51e-6	1.88e-6	9.16e-7
W-B_noCC_s_acc	1.87e-6	2.53e-6	4.65e-6	1.68e-6	2.01e-6	8.50e-7
W-B_FULL_adap	1.12e-6	1.25e-6	3.20e-6	1.69e-6	1.89e-6	8.41e-7
W-B_noCC_adap	9.28e-7	1.23e-6	2.90e-6	1.77e-6	1.91e-6	7.85e-7
S-L_FULL	2.04e-6	2.70e-6	1.00e-5	2.98e-6	1.94e-6	8.81e-7
S-L_noCC	2.03e-6	2.39e-6	1.16e-5	2.89e-6	1.87e-6	8.30e-7
S-L_FULL_s_rp	1.84e-6	2.64e-6	9.94e-6	3.43e-6	1.94e-6	8.57e-7
S-L_noCC_s_rp	1.73e-6	2.28e-6	1.12e-5	3.08e-6	1.94e-6	7.53e-7
S-L_FULL_s_acc	1.80e-6	2.54e-6	9.15e-6	2.94e-6	1.90e-6	8.69e-7
S-L_noCC_s_acc	1.93e-6	2.32e-6	1.15e-5	2.85e-6	1.94e-6	7.68e-7
S-L_FULL_adap	1.14e-6	1.40e-6	7.93e-6	2.97e-6	1.93e-6	7.02e-7
S-L_noCC_adap	1.35e-6	1.53e-6	7.61e-6	3.41e-6	1.90e-6	8.24e-7
PA10_vel	1.15e-2	3.57e-5	2.36e-5	1.77e-5	3.77e-5	1.75e-5

Table F.2: Comparison of high to low frequency control torque power ratio for different controllers, platform trajectory: zero, PA10 trajectory: **ssine**.

controller	S1	S2	E1	E2	W1	W2
PD	3.63e-5	3.02e-4	1.74e-4	2.02e-5	2.98e-5	1.40e-5
sat_sl	3.52e-5	3.02e-4	1.70e-4	1.95e-5	3.00e-5	1.34e-5
slid	3.50e-5	3.01e-4	4.61e-4	7.85e-6	2.98e-5	1.38e-5
PID	3.51e-5	3.01e-4	1.74e-4	2.08e-5	2.95e-5	1.35e-5
grv_frc_PD	5.87e-6	2.54e-5	2.17e-5	1.27e-6	4.01e-6	1.06e-6
grv_frc_PD_s_rp	5.56e-6	1.46e-5	1.46e-5	1.41e-6	3.96e-6	1.01e-6
grv_frc_PD_s_acc	3.57e-6	8.89e-6	1.16e-5	1.23e-6	4.16e-6	1.06e-6
grv_frc_PD_adap	1.99e-6	1.41e-6	3.12e-6	1.58e-6	4.59e-6	1.38e-6
grv_frc_sat_sl	6.43e-6	2.63e-5	2.22e-5	1.14e-6	4.30e-6	1.24e-6
grv_frc_sat_sl_s_rp	5.40e-6	1.46e-5	1.42e-5	1.01e-6	3.95e-6	1.25e-6
grv_frc_sat_sl_s_acc	3.60e-6	9.76e-6	1.15e-5	1.11e-6	3.80e-6	1.10e-6
grv_frc_sat_sl_adap	1.45e-6	7.35e-7	3.36e-6	1.09e-6	3.60e-6	1.16e-6
grv_frc_slid	6.12e-6	2.58e-5	5.89e-5	1.85e-6	4.15e-6	1.12e-6
grv_frc_slid_s_rp	5.38e-6	1.46e-5	3.86e-5	1.75e-6	4.12e-6	1.08e-6
grv_frc_slid_s_acc	3.81e-6	9.68e-6	3.09e-5	1.53e-6	4.28e-6	1.14e-6
grv_frc_slid_adap	1.97e-6	1.28e-6	6.96e-6	1.56e-6	4.07e-6	1.12e-6
lin_FULL	5.62e-6	2.52e-6	2.83e-6	4.39e-6	8.23e-6	6.54e-7
lin_noCC	7.74e-6	2.70e-6	3.44e-6	4.64e-6	9.78e-6	8.58e-7
lin_FULL_s_rp	3.19e-6	1.57e-6	2.29e-6	4.37e-6	9.30e-6	6.53e-7
lin_noCC_s_rp	4.64e-6	1.57e-6	2.88e-6	4.57e-6	9.92e-6	6.94e-7
lin_FULL_s_acc	2.55e-6	1.04e-6	2.16e-6	4.65e-6	9.19e-6	6.99e-7
lin_noCC_s_acc	3.28e-6	9.40e-7	2.55e-6	4.62e-6	9.46e-6	6.65e-7
W-B_FULL	3.58e-6	2.10e-5	1.72e-5	1.40e-6	3.86e-6	9.23e-7
W-B_noCC	4.52e-6	1.99e-5	1.72e-5	1.79e-6	3.89e-6	7.21e-7
W-B_FULL_s_rp	1.89e-6	9.84e-6	8.14e-6	1.72e-6	4.18e-6	9.61e-7
W-B_noCC_s_rp	2.83e-6	9.52e-6	9.09e-6	1.90e-6	4.12e-6	8.92e-7
W-B_FULL_s_acc	1.27e-6	5.63e-6	5.25e-6	1.35e-6	3.98e-6	9.65e-7
W-B_noCC_s_acc	1.87e-6	4.71e-6	5.68e-6	1.68e-6	3.98e-6	9.16e-7
W-B_FULL_adap	1.19e-6	9.71e-7	2.38e-6	1.60e-6	4.16e-6	8.84e-7
W-B_noCC_adap	1.49e-6	1.11e-6	2.51e-6	1.93e-6	4.02e-6	8.54e-7
S-L_FULL	3.43e-6	1.99e-5	4.17e-5	3.15e-6	4.14e-6	8.99e-7
S-L_noCC	4.49e-6	1.90e-5	4.19e-5	4.02e-6	4.23e-6	8.08e-7
S-L_FULL_s_rp	2.08e-6	9.41e-6	1.91e-5	3.39e-6	4.30e-6	9.09e-7
S-L_noCC_s_rp	2.74e-6	8.90e-6	2.18e-5	3.81e-6	4.33e-6	1.01e-6
S-L_FULL_s_acc	1.29e-6	5.47e-6	1.21e-5	3.02e-6	4.21e-6	8.41e-7
S-L_noCC_s_acc	1.86e-6	4.87e-6	1.39e-5	4.46e-6	4.37e-6	8.95e-7
S-L_FULL_adap	1.54e-6	1.59e-6	6.68e-6	3.51e-6	4.22e-6	8.52e-7
S-L_noCC_adap	1.99e-6	1.54e-6	7.25e-6	4.72e-6	4.68e-6	1.15e-6
PA10_vel	1.11e-2	4.27e-5	5.42e-5	4.11e-5	1.28e-4	3.21e-5

Table F.3: Comparison of high to low frequency control torque power ratio for different controllers, platform trajectory: sin_r2p0, PA10 trajectory: **`sine`**.

controller	S1	S2	E1	E2	W1	W2
PD	2.41e-5	1.39e-4	7.45e-5	1.53e-5	2.10e-5	1.22e-5
sat_sl	2.35e-5	1.40e-4	6.88e-5	1.46e-5	2.13e-5	1.14e-5
slid	2.31e-5	1.38e-4	1.99e-4	5.86e-6	2.10e-5	1.15e-5
PID	2.34e-5	1.38e-4	7.46e-5	1.48e-5	2.09e-5	1.14e-5
grv_frc_PD	2.82e-6	1.79e-5	1.70e-5	9.94e-7	1.96e-6	7.80e-7
grv_frc_PD_s_rp	2.12e-6	8.05e-6	6.78e-6	1.09e-6	1.88e-6	7.75e-7
grv_frc_PD_s_acc	2.31e-6	5.35e-6	4.01e-6	9.44e-7	1.94e-6	8.04e-7
grv_frc_PD_adap	6.13e-7	1.27e-6	2.54e-6	1.02e-6	2.10e-6	8.22e-7
grv_frc_sat_sl	2.77e-6	1.79e-5	1.64e-5	9.04e-7	1.90e-6	8.23e-7
grv_frc_sat_sl_s_rp	1.95e-6	7.95e-6	6.34e-6	8.49e-7	1.78e-6	9.09e-7
grv_frc_sat_sl_s_acc	2.44e-6	5.37e-6	4.01e-6	9.21e-7	1.81e-6	8.02e-7
grv_frc_sat_sl_adap	5.62e-7	6.89e-7	3.16e-6	8.45e-7	1.88e-6	8.45e-7
grv_frc_slid	2.59e-6	1.79e-5	4.36e-5	1.29e-6	2.03e-6	8.09e-7
grv_frc_slid_s_rp	2.07e-6	8.14e-6	1.67e-5	1.28e-6	1.94e-6	7.91e-7
grv_frc_slid_s_acc	2.30e-6	5.34e-6	1.03e-5	1.19e-6	2.04e-6	7.82e-7
grv_frc_slid_adap	7.45e-7	1.33e-6	6.20e-6	1.29e-6	2.06e-6	8.27e-7
lin_FULL	4.99e-6	2.89e-6	5.60e-6	3.29e-6	4.25e-6	5.90e-7
lin_noCC	6.53e-6	2.64e-6	5.46e-6	3.20e-6	4.97e-6	5.29e-7
lin_FULL_s_rp	2.65e-6	1.88e-6	3.90e-6	3.26e-6	4.66e-6	5.71e-7
lin_noCC_s_rp	2.98e-6	1.77e-6	4.29e-6	3.21e-6	5.00e-6	5.55e-7
lin_FULL_s_acc	3.27e-6	1.65e-6	4.35e-6	3.04e-6	4.82e-6	5.61e-7
lin_noCC_s_acc	3.91e-6	1.56e-6	4.80e-6	3.11e-6	4.65e-6	5.74e-7
W-B_FULL	2.51e-6	2.06e-5	2.19e-5	1.40e-6	1.87e-6	7.74e-7
W-B_noCC	2.90e-6	1.69e-5	1.86e-5	1.30e-6	2.00e-6	6.64e-7
W-B_FULL_s_rp	1.33e-6	8.89e-6	7.20e-6	1.34e-6	1.98e-6	6.82e-7
W-B_noCC_s_rp	1.51e-6	7.70e-6	7.38e-6	1.42e-6	1.96e-6	7.36e-7
W-B_FULL_s_acc	1.59e-6	6.08e-6	4.79e-6	1.50e-6	2.01e-6	8.70e-7
W-B_noCC_s_acc	1.78e-6	4.43e-6	4.62e-6	1.28e-6	1.98e-6	8.19e-7
W-B_FULL_adap	4.87e-7	1.16e-6	3.05e-6	1.45e-6	2.07e-6	7.32e-7
W-B_noCC_adap	5.25e-7	1.25e-6	2.88e-6	1.42e-6	2.05e-6	7.27e-7
S-L_FULL	2.46e-6	1.92e-5	4.96e-5	2.40e-6	2.07e-6	7.14e-7
S-L_noCC	2.99e-6	1.61e-5	4.37e-5	2.62e-6	2.16e-6	6.56e-7
S-L_FULL_s_rp	1.40e-6	8.83e-6	1.70e-5	2.34e-6	2.10e-6	8.17e-7
S-L_noCC_s_rp	1.53e-6	7.74e-6	1.71e-5	2.91e-6	2.18e-6	8.19e-7
S-L_FULL_s_acc	1.61e-6	5.93e-6	1.18e-5	2.57e-6	2.07e-6	7.72e-7
S-L_noCC_s_acc	1.95e-6	4.82e-6	1.16e-5	3.88e-6	2.26e-6	9.18e-7
S-L_FULL_adap	6.78e-7	1.30e-6	8.33e-6	2.67e-6	2.05e-6	7.54e-7
S-L_noCC_adap	7.06e-7	1.52e-6	6.92e-6	2.57e-6	2.20e-6	7.38e-7
PA10_vel	1.13e-2	3.05e-5	2.46e-5	1.72e-5	3.26e-5	1.63e-5

Table F.4: Comparison of high to low frequency control torque power ratio for different controllers, platform trajectory: sin_r2p0, PA10 trajectory: **ssine**.

controller	S1	S2	E1	E2	W1	W2
PD	3.67e-5	3.02e-4	1.75e-4	2.06e-5	3.05e-5	1.33e-5
sat_sl	3.64e-5	3.01e-4	1.71e-4	1.85e-5	3.15e-5	1.35e-5
slid	3.45e-5	3.00e-4	4.63e-4	8.26e-6	3.13e-5	1.37e-5
PID	3.48e-5	2.99e-4	1.75e-4	1.89e-5	3.11e-5	1.36e-5
grv_frc_PD	1.40e-5	1.76e-5	2.08e-5	1.28e-6	3.52e-6	1.19e-6
grv_frc_PD_s_rp	8.51e-6	1.30e-5	1.62e-5	1.64e-6	3.49e-6	1.15e-6
grv_frc_PD_s_acc	5.00e-6	8.40e-6	1.34e-5	1.39e-6	3.36e-6	1.21e-6
grv_frc_PD_adap	2.28e-6	1.36e-6	2.89e-6	1.52e-6	3.37e-6	1.20e-6
grv_frc_sat_sl	1.37e-5	1.75e-5	2.08e-5	1.14e-6	3.57e-6	1.19e-6
grv_frc_sat_sl_s_rp	8.49e-6	1.35e-5	1.57e-5	1.20e-6	3.42e-6	1.31e-6
grv_frc_sat_sl_s_acc	5.29e-6	8.82e-6	1.21e-5	1.21e-6	3.50e-6	1.26e-6
grv_frc_sat_sl_adap	1.65e-6	7.22e-7	3.47e-6	1.31e-6	3.34e-6	1.19e-6
grv_frc_slid	1.41e-5	1.78e-5	5.55e-5	1.89e-6	3.81e-6	1.20e-6
grv_frc_slid_s_rp	8.51e-6	1.31e-5	4.23e-5	1.84e-6	3.55e-6	1.23e-6
grv_frc_slid_s_acc	5.07e-6	8.59e-6	3.44e-5	1.57e-6	3.63e-6	1.28e-6
grv_frc_slid_adap	2.44e-6	1.38e-6	7.15e-6	2.11e-6	3.89e-6	1.34e-6
lin_FULL	1.01e-5	1.73e-6	2.91e-6	5.59e-6	7.80e-6	6.90e-7
lin_noCC	1.38e-5	1.80e-6	3.68e-6	5.68e-6	8.21e-6	7.58e-7
lin_FULL_s_rp	6.11e-6	1.21e-6	2.48e-6	5.78e-6	7.91e-6	6.83e-7
lin_noCC_s_rp	8.02e-6	1.29e-6	3.18e-6	5.78e-6	8.23e-6	7.60e-7
lin_FULL_s_acc	3.68e-6	1.03e-6	2.51e-6	5.92e-6	8.47e-6	7.50e-7
lin_noCC_s_acc	4.50e-6	1.04e-6	3.06e-6	5.82e-6	8.49e-6	7.57e-7
W-B_FULL	7.01e-6	1.13e-5	1.11e-5	1.69e-6	3.67e-6	7.96e-7
W-B_noCC	8.27e-6	1.21e-5	1.34e-5	1.42e-6	3.88e-6	8.48e-7
W-B_FULL_s_rp	3.65e-6	6.76e-6	6.53e-6	1.80e-6	3.76e-6	7.70e-7
W-B_noCC_s_rp	4.38e-6	6.82e-6	8.09e-6	1.52e-6	3.78e-6	8.23e-7
W-B_FULL_s_acc	1.70e-6	3.93e-6	4.35e-6	2.31e-6	4.11e-6	9.88e-7
W-B_noCC_s_acc	1.81e-6	4.16e-6	5.46e-6	1.43e-6	3.63e-6	7.98e-7
W-B_FULL_adap	1.24e-6	8.93e-7	1.93e-6	1.86e-6	3.79e-6	8.04e-7
W-B_noCC_adap	1.49e-6	9.15e-7	2.11e-6	1.66e-6	3.95e-6	7.91e-7
S-L_FULL	7.29e-6	1.06e-5	2.64e-5	3.96e-6	4.26e-6	8.02e-7
S-L_noCC	7.96e-6	1.13e-5	3.23e-5	3.50e-6	3.99e-6	8.19e-7
S-L_FULL_s_rp	3.82e-6	6.17e-6	1.50e-5	3.15e-6	4.12e-6	7.61e-7
S-L_noCC_s_rp	4.64e-6	7.21e-6	2.15e-5	5.71e-6	4.37e-6	9.90e-7
S-L_FULL_s_acc	1.70e-6	3.67e-6	9.35e-6	3.77e-6	4.11e-6	8.03e-7
S-L_noCC_s_acc	1.97e-6	4.04e-6	1.45e-5	3.88e-6	4.10e-6	8.43e-7
S-L_FULL_adap	1.99e-6	1.36e-6	5.10e-6	3.47e-6	4.08e-6	8.71e-7
S-L_noCC_adap	2.30e-6	1.26e-6	6.21e-6	4.09e-6	3.97e-6	7.86e-7
PA10_vel	1.11e-2	4.33e-5	1.05e-4	3.28e-5	1.48e-4	3.41e-5

Table F.5: Comparison of high to low frequency control torque power ratio for different controllers, platform trajectory: sin_r0p2, PA10 trajectory: **`sine`**.

controller	S1	S2	E1	E2	W1	W2
PD	2.25e-5	1.78e-4	7.26e-5	1.49e-5	2.10e-5	1.15e-5
sat_sl	2.32e-5	1.81e-4	6.82e-5	1.43e-5	2.18e-5	1.17e-5
slid	2.25e-5	1.79e-4	1.92e-4	5.12e-6	2.12e-5	1.16e-5
PID	2.02e-5	1.73e-4	7.40e-5	1.48e-5	2.14e-5	1.17e-5
grv_frc_PD	1.50e-5	2.40e-5	1.63e-5	9.47e-7	1.93e-6	9.03e-7
grv_frc_PD_s_rp	7.43e-6	1.19e-5	5.40e-6	9.73e-7	1.89e-6	9.02e-7
grv_frc_PD_s_acc	5.00e-6	6.39e-6	5.43e-6	9.55e-7	1.99e-6	9.64e-7
grv_frc_PD_adap	2.08e-6	1.78e-6	2.17e-6	1.16e-6	2.13e-6	9.24e-7
grv_frc_sat_sl	1.51e-5	2.32e-5	1.69e-5	8.71e-7	2.01e-6	9.20e-7
grv_frc_sat_sl_s_rp	7.72e-6	1.22e-5	5.39e-6	8.84e-7	1.92e-6	9.90e-7
grv_frc_sat_sl_s_acc	5.13e-6	6.78e-6	5.07e-6	8.80e-7	2.07e-6	9.35e-7
grv_frc_sat_sl_adap	1.27e-6	9.53e-7	2.85e-6	9.18e-7	1.94e-6	8.45e-7
grv_frc_slid	1.52e-5	2.35e-5	4.30e-5	1.20e-6	1.99e-6	8.15e-7
grv_frc_slid_s_rp	7.32e-6	1.18e-5	1.32e-5	1.23e-6	1.97e-6	7.81e-7
grv_frc_slid_s_acc	4.94e-6	6.47e-6	1.35e-5	1.25e-6	2.03e-6	9.00e-7
grv_frc_slid_adap	2.18e-6	1.85e-6	4.35e-6	1.16e-6	1.92e-6	9.14e-7
lin_FULL	1.81e-5	2.83e-6	4.41e-6	3.57e-6	5.18e-6	5.90e-7
lin_noCC	2.22e-5	3.05e-6	4.58e-6	3.50e-6	5.06e-6	6.67e-7
lin_FULL_s_rp	1.08e-5	2.55e-6	3.67e-6	3.79e-6	5.41e-6	7.07e-7
lin_noCC_s_rp	1.13e-5	2.30e-6	3.74e-6	3.47e-6	5.20e-6	5.90e-7
lin_FULL_s_acc	5.86e-6	1.58e-6	3.91e-6	3.63e-6	5.54e-6	5.72e-7
lin_noCC_s_acc	7.06e-6	1.54e-6	4.25e-6	3.58e-6	5.57e-6	6.04e-7
W-B_FULL	1.03e-5	2.09e-5	1.31e-5	1.14e-6	2.09e-6	6.89e-7
W-B_noCC	1.10e-5	2.15e-5	1.73e-5	1.04e-6	2.21e-6	6.52e-7
W-B_FULL_s_rp	5.59e-6	1.19e-5	4.18e-6	1.33e-6	2.18e-6	7.03e-7
W-B_noCC_s_rp	5.94e-6	1.16e-5	6.06e-6	1.50e-6	2.42e-6	7.03e-7
W-B_FULL_s_acc	2.51e-6	5.55e-6	3.37e-6	1.01e-6	2.14e-6	6.77e-7
W-B_noCC_s_acc	2.76e-6	5.68e-6	5.47e-6	1.07e-6	2.33e-6	6.78e-7
W-B_FULL_adap	1.54e-6	1.47e-6	1.82e-6	1.09e-6	2.38e-6	6.73e-7
W-B_noCC_adap	1.60e-6	1.56e-6	1.75e-6	1.37e-6	2.56e-6	6.87e-7
S-L_FULL	1.00e-5	1.91e-5	3.26e-5	2.91e-6	2.52e-6	6.78e-7
S-L_noCC	1.14e-5	2.02e-5	4.03e-5	3.67e-6	2.67e-6	7.51e-7
S-L_FULL_s_rp	5.67e-6	1.12e-5	9.44e-6	2.97e-6	2.51e-6	6.60e-7
S-L_noCC_s_rp	5.79e-6	1.11e-5	1.22e-5	2.99e-6	2.59e-6	7.22e-7
S-L_FULL_s_acc	2.69e-6	5.49e-6	8.25e-6	2.89e-6	2.57e-6	7.11e-7
S-L_noCC_s_acc	2.98e-6	5.60e-6	1.28e-5	2.66e-6	2.78e-6	7.01e-7
S-L_FULL_adap	2.40e-6	2.11e-6	3.51e-6	2.62e-6	2.50e-6	7.75e-7
S-L_noCC_adap	2.58e-6	2.24e-6	3.68e-6	2.60e-6	2.56e-6	6.43e-7
PA10_vel	1.15e-2	3.27e-5	5.52e-5	1.70e-5	2.99e-5	1.67e-5

Table F.6: Comparison of high to low frequency control torque power ratio for different controllers, platform trajectory: sin_r0p2, PA10 trajectory: **ssine**.

controller	S1	S2	E1	E2	W1	W2
PD	4.21e-5	3.19e-4	1.73e-4	2.14e-5	3.41e-5	1.59e-5
sat_sl	4.05e-5	3.20e-4	1.70e-4	1.99e-5	3.38e-5	1.52e-5
slid	3.99e-5	3.20e-4	4.65e-4	6.89e-6	3.15e-5	1.45e-5
PID	4.15e-5	3.20e-4	1.73e-4	2.24e-5	3.27e-5	1.49e-5
grv_frc_PD	1.34e-5	1.34e-5	2.05e-5	1.20e-6	3.71e-6	9.34e-7
grv_frc_PD_s_rp	6.87e-6	1.08e-5	1.41e-5	1.30e-6	3.95e-6	9.03e-7
grv_frc_PD_s_acc	4.58e-6	6.63e-6	1.10e-5	9.81e-7	3.94e-6	9.09e-7
grv_frc_PD_adap	1.76e-6	1.02e-6	2.29e-6	1.16e-6	3.95e-6	9.87e-7
grv_frc_sat_sl	1.36e-5	1.31e-5	2.05e-5	1.05e-6	4.27e-6	1.01e-6
grv_frc_sat_sl_s_rp	7.07e-6	1.11e-5	1.45e-5	1.08e-6	4.15e-6	9.54e-7
grv_frc_sat_sl_s_acc	4.40e-6	6.38e-6	1.12e-5	1.01e-6	4.05e-6	9.34e-7
grv_frc_sat_sl_adap	1.35e-6	8.05e-7	3.00e-6	1.21e-6	4.05e-6	1.03e-6
grv_frc_slid	1.34e-5	1.34e-5	5.39e-5	2.07e-6	4.00e-6	9.08e-7
grv_frc_slid_s_rp	7.05e-6	1.08e-5	3.88e-5	1.62e-6	3.93e-6	9.74e-7
grv_frc_slid_s_acc	4.69e-6	6.68e-6	2.91e-5	1.48e-6	3.99e-6	9.03e-7
grv_frc_slid_adap	1.90e-6	1.08e-6	5.45e-6	1.83e-6	3.99e-6	8.54e-7
lin_FULL	7.86e-6	1.55e-6	3.14e-6	4.60e-6	9.51e-6	4.96e-7
lin_noCC	1.02e-5	1.65e-6	3.90e-6	4.98e-6	1.00e-5	5.77e-7
lin_FULL_s_rp	4.74e-6	8.22e-7	2.12e-6	4.50e-6	9.48e-6	5.53e-7
lin_noCC_s_rp	6.41e-6	8.62e-7	2.72e-6	4.48e-6	9.40e-6	4.95e-7
lin_FULL_s_acc	2.87e-6	7.21e-7	1.86e-6	4.52e-6	9.21e-6	4.98e-7
lin_noCC_s_acc	3.90e-6	7.15e-7	2.53e-6	4.46e-6	9.08e-6	4.53e-7
W-B_FULL	6.16e-6	1.17e-5	1.24e-5	1.62e-6	3.95e-6	6.03e-7
W-B_noCC	7.81e-6	1.07e-5	1.47e-5	1.93e-6	3.99e-6	5.91e-7
W-B_FULL_s_rp	3.12e-6	3.99e-6	5.32e-6	1.45e-6	3.82e-6	6.97e-7
W-B_noCC_s_rp	3.79e-6	4.49e-6	7.24e-6	1.85e-6	3.89e-6	6.47e-7
W-B_FULL_s_acc	1.35e-6	2.46e-6	3.36e-6	1.42e-6	3.82e-6	6.07e-7
W-B_noCC_s_acc	1.89e-6	2.99e-6	5.19e-6	1.86e-6	3.87e-6	4.95e-7
W-B_FULL_adap	1.06e-6	7.71e-7	1.76e-6	1.69e-6	3.95e-6	6.61e-7
W-B_noCC_adap	1.19e-6	7.78e-7	1.81e-6	2.17e-6	3.67e-6	5.73e-7
S-L_FULL	6.23e-6	1.05e-5	2.96e-5	3.68e-6	3.93e-6	5.34e-7
S-L_noCC	7.29e-6	1.07e-5	3.53e-5	4.11e-6	3.84e-6	6.56e-7
S-L_FULL_s_rp	3.41e-6	3.75e-6	1.28e-5	3.40e-6	4.00e-6	6.04e-7
S-L_noCC_s_rp	4.12e-6	4.04e-6	1.83e-5	3.95e-6	4.05e-6	5.99e-7
S-L_FULL_s_acc	1.60e-6	2.32e-6	7.79e-6	3.36e-6	3.93e-6	6.65e-7
S-L_noCC_s_acc	1.98e-6	2.56e-6	1.24e-5	3.74e-6	3.92e-6	5.74e-7
S-L_FULL_adap	1.58e-6	9.00e-7	3.99e-6	3.63e-6	4.08e-6	6.12e-7
S-L_noCC_adap	1.89e-6	1.04e-6	4.75e-6	4.57e-6	4.19e-6	7.23e-7
PA10_vel	1.15e-2	4.36e-5	9.63e-5	3.27e-5	1.12e-4	2.91e-5

Table F.7: Comparison of high to low frequency control torque power ratio for different controllers, platform trajectory: sin_r1p2pi2, PA10 trajectory: **`sine`**.

controller	S1	S2	E1	E2	W1	W2
PD	2.14e-5	2.11e-4	6.18e-5	1.64e-5	2.30e-5	1.38e-5
sat_sl	2.02e-5	2.08e-4	6.24e-5	1.56e-5	2.24e-5	1.32e-5
slid	2.01e-5	2.08e-4	1.73e-4	5.49e-6	2.05e-5	1.26e-5
PID	2.06e-5	2.12e-4	6.06e-5	1.59e-5	2.11e-5	1.28e-5
grv_frc_PD	8.75e-6	1.89e-5	2.24e-5	8.33e-7	2.34e-6	6.47e-7
grv_frc_PD_s_rp	5.64e-6	9.84e-6	4.05e-6	8.32e-7	2.40e-6	6.14e-7
grv_frc_PD_s_acc	3.67e-6	5.20e-6	5.35e-6	7.71e-7	2.45e-6	5.84e-7
grv_frc_PD_adap	1.33e-6	1.44e-6	1.87e-6	8.83e-7	2.44e-6	6.39e-7
grv_frc_sat_sl	9.13e-6	1.85e-5	2.36e-5	8.08e-7	2.51e-6	6.45e-7
grv_frc_sat_sl_s_rp	5.74e-6	1.01e-5	3.96e-6	7.67e-7	2.41e-6	5.50e-7
grv_frc_sat_sl_s_acc	3.98e-6	5.41e-6	5.31e-6	7.98e-7	2.46e-6	7.02e-7
grv_frc_sat_sl_adap	9.63e-7	9.15e-7	2.55e-6	8.08e-7	2.46e-6	6.17e-7
grv_frc_slid	8.95e-6	1.87e-5	6.16e-5	1.19e-6	2.61e-6	6.30e-7
grv_frc_slid_s_rp	5.75e-6	9.98e-6	1.07e-5	1.23e-6	2.48e-6	6.49e-7
grv_frc_slid_s_acc	3.91e-6	5.36e-6	1.44e-5	1.20e-6	2.45e-6	6.35e-7
grv_frc_slid_adap	1.44e-6	1.67e-6	4.19e-6	1.11e-6	2.35e-6	5.56e-7
lin_FULL	1.03e-5	1.80e-6	6.96e-6	3.11e-6	6.45e-6	4.05e-7
lin_noCC	1.30e-5	1.87e-6	7.94e-6	3.10e-6	6.65e-6	3.66e-7
lin_FULL_s_rp	8.14e-6	1.77e-6	3.06e-6	2.98e-6	5.95e-6	4.28e-7
lin_noCC_s_rp	9.14e-6	1.81e-6	3.49e-6	2.90e-6	6.00e-6	3.99e-7
lin_FULL_s_acc	4.30e-6	1.09e-6	3.43e-6	3.06e-6	5.94e-6	3.83e-7
lin_noCC_s_acc	5.20e-6	1.08e-6	3.97e-6	2.96e-6	6.12e-6	3.55e-7
W-B_FULL	5.76e-6	1.23e-5	2.10e-5	1.21e-6	2.44e-6	4.75e-7
W-B_noCC	6.58e-6	1.39e-5	2.58e-5	1.21e-6	2.35e-6	4.64e-7
W-B_FULL_s_rp	4.84e-6	8.62e-6	3.07e-6	1.47e-6	2.25e-6	5.92e-7
W-B_noCC_s_rp	5.08e-6	8.99e-6	4.40e-6	1.40e-6	2.34e-6	5.28e-7
W-B_FULL_s_acc	2.39e-6	3.59e-6	3.86e-6	1.40e-6	2.35e-6	5.18e-7
W-B_noCC_s_acc	2.64e-6	4.91e-6	5.57e-6	1.22e-6	2.42e-6	4.70e-7
W-B_FULL_adap	9.65e-7	1.37e-6	2.15e-6	1.61e-6	2.34e-6	5.04e-7
W-B_noCC_adap	1.07e-6	1.52e-6	1.98e-6	1.39e-6	2.43e-6	4.62e-7
S-L_FULL	6.27e-6	1.14e-5	5.20e-5	2.84e-6	2.49e-6	4.41e-7
S-L_noCC	6.41e-6	1.25e-5	6.71e-5	2.74e-6	2.58e-6	5.69e-7
S-L_FULL_s_rp	4.99e-6	8.33e-6	6.51e-6	2.45e-6	2.45e-6	4.68e-7
S-L_noCC_s_rp	5.51e-6	8.82e-6	9.79e-6	3.07e-6	2.52e-6	4.52e-7
S-L_FULL_s_acc	2.40e-6	3.34e-6	9.87e-6	2.78e-6	2.38e-6	5.27e-7
S-L_noCC_s_acc	3.00e-6	3.92e-6	1.50e-5	3.16e-6	2.63e-6	5.25e-7
S-L_FULL_adap	1.65e-6	2.07e-6	4.77e-6	2.78e-6	2.52e-6	4.94e-7
S-L_noCC_adap	1.94e-6	2.29e-6	4.25e-6	2.80e-6	2.51e-6	5.08e-7
PA10_vel	1.14e-2	3.16e-5	4.91e-5	1.70e-5	3.19e-5	1.56e-5

Table F.8: Comparison of high to low frequency control torque power ratio for different controllers, platform trajectory: sin_r1p2pi2, PA10 trajectory: **ssine**.

controller	S1	S2	E1	E2	W1	W2
PD	2.46e-5	2.97e-4	1.80e-4	1.98e-5	3.09e-5	1.44e-5
sat_sl	2.46e-5	2.96e-4	1.75e-4	1.92e-5	3.15e-5	1.43e-5
slid	2.43e-5	2.96e-4	4.76e-4	8.22e-6	3.14e-5	1.46e-5
PID	2.44e-5	2.96e-4	1.79e-4	2.05e-5	3.07e-5	1.45e-5
grv_frc_PD	5.41e-6	1.92e-5	2.04e-5	1.18e-6	3.62e-6	9.10e-7
grv_frc_PD_s_rp	4.85e-6	1.23e-5	1.60e-5	1.14e-6	3.66e-6	9.91e-7
grv_frc_PD_s_acc	4.00e-6	9.24e-6	1.34e-5	1.67e-6	4.01e-6	1.11e-6
grv_frc_PD_adap	1.92e-6	1.23e-6	2.64e-6	1.54e-6	3.47e-6	1.04e-6
grv_frc_sat_sl	5.64e-6	1.91e-5	1.84e-5	1.08e-6	4.06e-6	9.52e-7
grv_frc_sat_sl_s_rp	4.90e-6	1.25e-5	1.54e-5	1.14e-6	3.84e-6	1.01e-6
grv_frc_sat_sl_s_acc	3.81e-6	8.95e-6	1.24e-5	1.07e-6	3.47e-6	9.26e-7
grv_frc_sat_sl_adap	1.60e-6	7.47e-7	2.96e-6	1.17e-6	3.37e-6	9.59e-7
grv_frc_slid	5.59e-6	1.97e-5	5.22e-5	1.37e-6	3.69e-6	9.41e-7
grv_frc_slid_s_rp	4.95e-6	1.25e-5	4.02e-5	1.33e-6	3.95e-6	1.03e-6
grv_frc_slid_s_acc	3.74e-6	9.03e-6	3.30e-5	1.76e-6	3.53e-6	8.83e-7
grv_frc_slid_adap	2.21e-6	1.29e-6	6.14e-6	1.95e-6	3.69e-6	8.77e-7
lin_FULL	7.16e-6	2.10e-6	2.00e-6	4.87e-6	8.84e-6	6.19e-7
lin_noCC	6.59e-6	2.09e-6	2.41e-6	4.69e-6	8.27e-6	6.11e-7
lin_FULL_s_rp	6.39e-6	1.39e-6	2.28e-6	4.85e-6	8.15e-6	5.95e-7
lin_noCC_s_rp	6.31e-6	1.40e-6	2.79e-6	4.82e-6	8.11e-6	5.14e-7
lin_FULL_s_acc	4.20e-6	1.06e-6	1.95e-6	4.96e-6	7.67e-6	5.43e-7
lin_noCC_s_acc	4.62e-6	1.14e-6	2.71e-6	4.88e-6	8.20e-6	5.40e-7
W-B_FULL	3.29e-6	1.53e-5	1.18e-5	1.67e-6	3.40e-6	6.77e-7
W-B_noCC	3.07e-6	1.53e-5	1.29e-5	1.74e-6	3.41e-6	6.84e-7
W-B_FULL_s_rp	2.79e-6	8.33e-6	8.08e-6	2.01e-6	3.29e-6	6.16e-7
W-B_noCC_s_rp	2.99e-6	8.27e-6	8.87e-6	1.49e-6	3.50e-6	6.94e-7
W-B_FULL_s_acc	1.97e-6	5.37e-6	5.37e-6	1.63e-6	3.41e-6	7.41e-7
W-B_noCC_s_acc	2.36e-6	5.54e-6	6.86e-6	2.48e-6	4.12e-6	9.67e-7
W-B_FULL_adap	1.14e-6	9.84e-7	1.94e-6	2.32e-6	3.26e-6	7.17e-7
W-B_noCC_adap	1.59e-6	9.57e-7	2.18e-6	1.68e-6	3.28e-6	8.38e-7
S-L_FULL	3.54e-6	1.31e-5	2.45e-5	4.20e-6	3.29e-6	7.41e-7
S-L_noCC	2.87e-6	1.47e-5	3.00e-5	3.67e-6	3.33e-6	6.23e-7
S-L_FULL_s_rp	3.07e-6	8.09e-6	1.77e-5	3.84e-6	3.59e-6	6.68e-7
S-L_noCC_s_rp	3.07e-6	7.65e-6	2.01e-5	4.26e-6	3.90e-6	8.18e-7
S-L_FULL_s_acc	2.11e-6	5.13e-6	1.21e-5	3.27e-6	3.79e-6	6.39e-7
S-L_noCC_s_acc	2.07e-6	4.92e-6	1.48e-5	4.02e-6	3.76e-6	6.27e-7
S-L_FULL_adap	1.32e-6	1.14e-6	3.71e-6	4.01e-6	2.97e-6	5.81e-7
S-L_noCC_adap	1.61e-6	1.09e-6	4.98e-6	3.41e-6	3.14e-6	6.25e-7
PA10_vel	1.13e-2	4.78e-5	6.29e-5	3.24e-5	1.26e-4	2.72e-5

Table F.9: Comparison of high to low frequency control torque power ratio for different controllers, platform trajectory: sin2_s20, PA10 trajectory: **`sine`**.

controller	S1	S2	E1	E2	W1	W2
PD	2.22e-5	1.53e-4	5.13e-5	1.55e-5	2.17e-5	1.24e-5
sat_sl	2.16e-5	1.53e-4	4.91e-5	1.48e-5	2.20e-5	1.21e-5
slid	2.16e-5	1.54e-4	1.38e-4	5.16e-6	2.17e-5	1.26e-5
PID	2.19e-5	1.55e-4	5.09e-5	1.57e-5	2.18e-5	1.29e-5
grv_frc_PD	4.88e-6	1.09e-5	1.46e-5	8.38e-7	1.71e-6	4.94e-7
grv_frc_PD_s_rp	3.71e-6	6.20e-6	8.11e-6	9.20e-7	1.75e-6	6.86e-7
grv_frc_PD_s_acc	2.98e-6	4.34e-6	6.01e-6	8.72e-7	1.81e-6	6.29e-7
grv_frc_PD_adap	1.18e-6	1.19e-6	2.96e-6	9.63e-7	1.80e-6	6.48e-7
grv_frc_sat_sl	5.26e-6	1.15e-5	1.35e-5	8.69e-7	2.01e-6	6.67e-7
grv_frc_sat_sl_s_rp	3.92e-6	6.42e-6	7.40e-6	7.96e-7	1.69e-6	6.12e-7
grv_frc_sat_sl_s_acc	3.20e-6	4.89e-6	5.49e-6	7.87e-7	1.60e-6	6.28e-7
grv_frc_sat_sl_adap	1.01e-6	7.61e-7	3.86e-6	8.56e-7	1.87e-6	5.75e-7
grv_frc_slid	5.11e-6	1.12e-5	3.66e-5	1.16e-6	1.73e-6	7.00e-7
grv_frc_slid_s_rp	3.87e-6	6.37e-6	1.97e-5	1.09e-6	1.79e-6	6.45e-7
grv_frc_slid_s_acc	3.19e-6	4.59e-6	1.46e-5	1.13e-6	1.81e-6	6.00e-7
grv_frc_slid_adap	1.32e-6	1.34e-6	7.14e-6	1.17e-6	2.07e-6	6.27e-7
lin_FULL	9.55e-6	2.24e-6	5.54e-6	3.23e-6	4.39e-6	4.42e-7
lin_noCC	9.80e-6	2.15e-6	5.79e-6	3.24e-6	4.43e-6	4.99e-7
lin_FULL_s_rp	7.50e-6	1.67e-6	4.67e-6	3.27e-6	4.40e-6	4.74e-7
lin_noCC_s_rp	7.73e-6	1.58e-6	4.91e-6	3.21e-6	4.43e-6	3.61e-7
lin_FULL_s_acc	5.74e-6	1.69e-6	4.98e-6	3.44e-6	4.41e-6	5.87e-7
lin_noCC_s_acc	5.50e-6	1.43e-6	5.01e-6	3.15e-6	4.31e-6	3.97e-7
W-B_FULL	3.78e-6	1.34e-5	1.58e-5	1.63e-6	1.70e-6	5.10e-7
W-B_noCC	3.64e-6	1.20e-5	1.55e-5	1.27e-6	1.72e-6	5.97e-7
W-B_FULL_s_rp	2.99e-6	7.25e-6	9.13e-6	1.49e-6	1.92e-6	5.80e-7
W-B_noCC_s_rp	3.09e-6	6.55e-6	8.99e-6	1.52e-6	1.80e-6	5.66e-7
W-B_FULL_s_acc	2.24e-6	5.02e-6	6.26e-6	1.39e-6	1.90e-6	6.06e-7
W-B_noCC_s_acc	2.19e-6	4.32e-6	6.50e-6	1.26e-6	1.85e-6	5.51e-7
W-B_FULL_adap	1.16e-6	1.24e-6	3.28e-6	1.63e-6	1.77e-6	5.61e-7
W-B_noCC_adap	1.20e-6	1.24e-6	3.20e-6	1.49e-6	1.53e-6	6.24e-7
S-L_FULL	3.60e-6	1.24e-5	3.62e-5	2.68e-6	1.61e-6	5.05e-7
S-L_noCC	3.74e-6	1.18e-5	3.59e-5	2.54e-6	1.85e-6	5.27e-7
S-L_FULL_s_rp	2.88e-6	6.90e-6	2.02e-5	2.90e-6	1.72e-6	5.81e-7
S-L_noCC_s_rp	2.99e-6	6.36e-6	1.99e-5	2.83e-6	1.74e-6	6.07e-7
S-L_FULL_s_acc	2.20e-6	4.79e-6	1.51e-5	2.97e-6	1.96e-6	5.63e-7
S-L_noCC_s_acc	2.20e-6	4.16e-6	1.55e-5	2.75e-6	1.96e-6	4.95e-7
S-L_FULL_adap	1.78e-6	1.40e-6	7.58e-6	3.58e-6	1.56e-6	5.25e-7
S-L_noCC_adap	1.72e-6	1.38e-6	7.25e-6	2.69e-6	1.71e-6	5.39e-7
PA10_vel	1.15e-2	3.02e-5	2.97e-5	1.68e-5	3.35e-5	1.57e-5

Table F.10: Comparison of high to low frequency control torque power ratio for different controllers, platform trajectory: sin2_s20, PA10 trajectory: **ssine**.

controller	S1	S2	E1	E2	W1	W2
PD	2.76e-5	2.55e-4	1.62e-4	2.14e-5	2.89e-5	1.51e-5
sat_sl	2.81e-5	2.56e-4	1.61e-4	1.82e-5	2.92e-5	1.52e-5
slid	2.80e-5	2.55e-4	4.28e-4	7.91e-6	2.92e-5	1.49e-5
PID	2.74e-5	2.55e-4	1.61e-4	1.99e-5	2.95e-5	1.49e-5
grv_frc_PD	1.41e-5	1.12e-5	1.14e-5	1.68e-6	4.17e-6	8.60e-7
grv_frc_PD_s_rp	5.38e-6	8.24e-6	1.19e-5	1.29e-6	4.20e-6	1.00e-6
grv_frc_PD_s_acc	4.98e-6	6.62e-6	1.05e-5	1.48e-6	4.07e-6	1.01e-6
grv_frc_PD_adap	1.55e-6	1.03e-6	2.84e-6	1.62e-6	4.08e-6	9.48e-7
grv_frc_sat_sl	1.46e-5	1.13e-5	1.08e-5	1.43e-6	4.55e-6	1.08e-6
grv_frc_sat_sl_s_rp	5.35e-6	8.08e-6	1.17e-5	1.16e-6	4.34e-6	8.69e-7
grv_frc_sat_sl_s_acc	4.61e-6	6.99e-6	9.88e-6	1.18e-6	4.27e-6	8.79e-7
grv_frc_sat_sl_adap	1.06e-6	5.80e-7	3.13e-6	1.23e-6	4.14e-6	8.93e-7
grv_frc_slid	1.45e-5	1.13e-5	2.83e-5	1.38e-6	4.24e-6	8.09e-7
grv_frc_slid_s_rp	5.37e-6	8.08e-6	2.95e-5	2.02e-6	4.14e-6	8.87e-7
grv_frc_slid_s_acc	4.52e-6	6.70e-6	2.49e-5	1.47e-6	4.26e-6	8.85e-7
grv_frc_slid_adap	1.43e-6	1.03e-6	6.54e-6	1.57e-6	3.97e-6	8.18e-7
lin_FULL	9.43e-6	1.94e-6	1.89e-6	5.72e-6	1.03e-5	4.89e-7
lin_noCC	1.02e-5	1.96e-6	2.04e-6	5.70e-6	1.04e-5	4.94e-7
lin_FULL_s_rp	5.06e-6	1.17e-6	2.19e-6	5.56e-6	1.01e-5	4.99e-7
lin_noCC_s_rp	5.27e-6	1.19e-6	2.42e-6	5.58e-6	1.02e-5	4.89e-7
lin_FULL_s_acc	3.62e-6	1.11e-6	2.18e-6	5.85e-6	1.04e-5	5.99e-7
lin_noCC_s_acc	3.83e-6	1.08e-6	2.57e-6	5.60e-6	1.02e-5	4.79e-7
W-B_FULL	8.15e-6	1.23e-5	6.74e-6	1.86e-6	4.25e-6	6.12e-7
W-B_noCC	8.89e-6	1.15e-5	6.40e-6	2.10e-6	4.30e-6	6.43e-7
W-B_FULL_s_rp	3.67e-6	5.59e-6	4.59e-6	2.19e-6	4.45e-6	5.70e-7
W-B_noCC_s_rp	3.68e-6	5.37e-6	4.99e-6	2.16e-6	4.48e-6	6.98e-7
W-B_FULL_s_acc	2.16e-6	4.71e-6	3.61e-6	1.97e-6	4.36e-6	6.55e-7
W-B_noCC_s_acc	2.49e-6	3.96e-6	4.14e-6	1.97e-6	4.28e-6	6.17e-7
W-B_FULL_adap	8.79e-7	7.66e-7	2.22e-6	1.88e-6	4.31e-6	6.96e-7
W-B_noCC_adap	9.53e-7	7.96e-7	2.39e-6	1.94e-6	4.18e-6	7.43e-7
S-L_FULL	8.53e-6	1.20e-5	1.49e-5	4.10e-6	4.50e-6	7.63e-7
S-L_noCC	8.97e-6	1.13e-5	1.45e-5	3.65e-6	4.67e-6	6.66e-7
S-L_FULL_s_rp	3.55e-6	5.27e-6	1.06e-5	3.92e-6	4.72e-6	7.30e-7
S-L_noCC_s_rp	3.63e-6	5.34e-6	1.17e-5	3.35e-6	4.91e-6	7.96e-7
S-L_FULL_s_acc	2.44e-6	4.39e-6	8.22e-6	5.36e-6	5.04e-6	8.21e-7
S-L_noCC_s_acc	2.57e-6	4.15e-6	9.97e-6	3.41e-6	4.76e-6	7.32e-7
S-L_FULL_adap	1.33e-6	1.01e-6	5.29e-6	3.84e-6	4.64e-6	7.63e-7
S-L_noCC_adap	1.40e-6	1.13e-6	6.33e-6	3.22e-6	4.67e-6	8.53e-7
PA10_vel	1.14e-2	4.87e-5	5.13e-5	3.15e-5	1.42e-4	3.22e-5

Table F.11: Comparison of high to low frequency control torque power ratio for different controllers, platform trajectory: sin2_s20_2, PA10 trajectory: **`sine`**.

controller	S1	S2	E1	E2	W1	W2
PD	1.79e-5	1.52e-4	3.65e-5	1.55e-5	2.02e-5	1.29e-5
sat_sl	1.78e-5	1.53e-4	3.45e-5	1.43e-5	2.04e-5	1.33e-5
slid	1.74e-5	1.52e-4	9.77e-5	5.24e-6	2.00e-5	1.30e-5
PID	1.83e-5	1.51e-4	3.61e-5	1.48e-5	2.04e-5	1.28e-5
grv_frc_PD	9.52e-6	6.09e-6	1.20e-5	9.46e-7	2.19e-6	5.80e-7
grv_frc_PD_s_rp	4.07e-6	4.51e-6	7.30e-6	1.01e-6	2.25e-6	6.53e-7
grv_frc_PD_s_acc	3.42e-6	3.35e-6	5.30e-6	1.01e-6	2.27e-6	6.96e-7
grv_frc_PD_adap	1.62e-6	1.28e-6	3.08e-6	1.07e-6	2.29e-6	7.02e-7
grv_frc_sat_sl	9.23e-6	6.20e-6	1.11e-5	8.53e-7	2.25e-6	5.66e-7
grv_frc_sat_sl_s_rp	3.86e-6	4.57e-6	6.87e-6	8.79e-7	2.35e-6	5.81e-7
grv_frc_sat_sl_s_acc	3.30e-6	3.06e-6	5.32e-6	8.54e-7	2.24e-6	5.71e-7
grv_frc_sat_sl_adap	1.16e-6	6.47e-7	3.57e-6	9.49e-7	2.29e-6	5.32e-7
grv_frc_slid	1.00e-5	6.42e-6	3.10e-5	1.27e-6	2.35e-6	6.65e-7
grv_frc_slid_s_rp	3.66e-6	4.59e-6	1.76e-5	1.09e-6	2.22e-6	5.52e-7
grv_frc_slid_s_acc	3.18e-6	3.24e-6	1.27e-5	1.04e-6	2.22e-6	4.97e-7
grv_frc_slid_adap	1.39e-6	1.26e-6	6.89e-6	1.10e-6	2.15e-6	5.90e-7
lin_FULL	1.47e-5	1.63e-6	4.93e-6	3.53e-6	5.86e-6	3.47e-7
lin_noCC	1.50e-5	1.52e-6	5.23e-6	3.41e-6	6.04e-6	3.67e-7
lin_FULL_s_rp	7.46e-6	1.31e-6	4.65e-6	3.65e-6	5.87e-6	3.74e-7
lin_noCC_s_rp	7.38e-6	1.29e-6	4.99e-6	3.76e-6	6.04e-6	4.04e-7
lin_FULL_s_acc	5.59e-6	1.32e-6	4.63e-6	3.83e-6	6.01e-6	3.99e-7
lin_noCC_s_acc	5.29e-6	1.20e-6	4.91e-6	3.81e-6	6.10e-6	3.57e-7
W-B_FULL	7.21e-6	9.69e-6	1.37e-5	1.50e-6	2.40e-6	5.11e-7
W-B_noCC	7.20e-6	8.63e-6	1.29e-5	1.42e-6	2.54e-6	4.96e-7
W-B_FULL_s_rp	3.31e-6	4.76e-6	7.38e-6	1.37e-6	2.41e-6	4.28e-7
W-B_noCC_s_rp	3.39e-6	4.37e-6	7.73e-6	1.45e-6	2.48e-6	4.98e-7
W-B_FULL_s_acc	2.41e-6	3.77e-6	5.47e-6	1.77e-6	2.52e-6	5.47e-7
W-B_noCC_s_acc	2.37e-6	3.39e-6	5.78e-6	1.64e-6	2.56e-6	5.24e-7
W-B_FULL_adap	1.41e-6	1.30e-6	3.34e-6	1.88e-6	2.62e-6	6.49e-7
W-B_noCC_adap	1.35e-6	1.23e-6	3.05e-6	1.50e-6	2.51e-6	5.86e-7
S-L_FULL	7.54e-6	8.92e-6	3.26e-5	2.62e-6	2.44e-6	5.55e-7
S-L_noCC	7.52e-6	8.12e-6	3.05e-5	2.93e-6	2.68e-6	5.43e-7
S-L_FULL_s_rp	3.31e-6	4.40e-6	1.78e-5	2.90e-6	2.57e-6	6.08e-7
S-L_noCC_s_rp	3.66e-6	4.12e-6	1.89e-5	2.68e-6	2.67e-6	5.89e-7
S-L_FULL_s_acc	2.50e-6	3.64e-6	1.28e-5	2.62e-6	2.56e-6	5.90e-7
S-L_noCC_s_acc	2.42e-6	3.36e-6	1.46e-5	2.82e-6	2.70e-6	5.09e-7
S-L_FULL_adap	1.78e-6	1.35e-6	8.74e-6	2.62e-6	2.69e-6	5.84e-7
S-L_noCC_adap	1.80e-6	1.40e-6	8.55e-6	2.63e-6	2.77e-6	6.21e-7
PA10_vel	1.15e-2	3.01e-5	2.32e-5	1.68e-5	3.11e-5	1.60e-5

Table F.12: Comparison of high to low frequency control torque power ratio for different controllers, platform trajectory: sin2_s20_2, PA10 trajectory: **ssine**.

controller	S1	S2	E1	E2	W1	W2
PD	2.29e-5	2.75e-4	1.57e-4	2.35e-5	3.36e-5	1.29e-5
sat_sl	2.30e-5	2.75e-4	1.55e-4	1.95e-5	3.33e-5	1.39e-5
slid	2.33e-5	2.75e-4	4.19e-4	6.91e-6	3.31e-5	1.34e-5
PID	2.31e-5	2.75e-4	1.57e-4	2.27e-5	3.36e-5	1.33e-5
grv_frc_PD	3.45e-6	5.78e-6	9.62e-6	1.29e-6	2.97e-6	1.07e-6
grv_frc_PD_s_rp	3.80e-6	5.87e-6	1.01e-5	1.26e-6	3.40e-6	1.38e-6
grv_frc_PD_s_acc	3.76e-6	5.50e-6	9.76e-6	1.37e-6	2.96e-6	1.23e-6
grv_frc_PD_adap	1.37e-6	8.80e-7	2.39e-6	1.35e-6	2.81e-6	1.11e-6
grv_frc_sat_sl	3.63e-6	6.27e-6	1.02e-5	1.30e-6	3.69e-6	1.16e-6
grv_frc_sat_sl_s_rp	4.16e-6	6.16e-6	1.11e-5	1.48e-6	3.92e-6	1.61e-6
grv_frc_sat_sl_s_acc	3.58e-6	5.70e-6	1.04e-5	1.33e-6	3.61e-6	1.32e-6
grv_frc_sat_sl_adap	1.27e-6	8.08e-7	3.72e-6	1.53e-6	3.59e-6	1.46e-6
grv_frc_slid	3.61e-6	6.10e-6	2.62e-5	1.36e-6	3.14e-6	1.10e-6
grv_frc_slid_s_rp	3.96e-6	5.82e-6	2.59e-5	1.84e-6	3.05e-6	1.21e-6
grv_frc_slid_s_acc	3.79e-6	5.71e-6	2.58e-5	1.66e-6	3.05e-6	1.27e-6
grv_frc_slid_adap	1.51e-6	1.00e-6	6.29e-6	1.49e-6	3.13e-6	1.13e-6
lin_FULL	3.19e-6	8.51e-7	1.90e-6	4.86e-6	6.66e-6	7.43e-7
lin_noCC	3.79e-6	8.09e-7	2.46e-6	4.81e-6	6.40e-6	7.25e-7
lin_FULL_s_rp	3.18e-6	7.21e-7	1.95e-6	5.01e-6	6.58e-6	7.07e-7
lin_noCC_s_rp	4.02e-6	7.07e-7	2.40e-6	4.98e-6	6.66e-6	6.93e-7
lin_FULL_s_acc	3.44e-6	8.37e-7	2.01e-6	5.24e-6	7.11e-6	7.84e-7
lin_noCC_s_acc	3.90e-6	7.28e-7	2.50e-6	5.12e-6	6.60e-6	6.36e-7
W-B_FULL	1.49e-6	3.00e-6	3.04e-6	1.81e-6	2.85e-6	9.29e-7
W-B_noCC	1.94e-6	2.84e-6	4.26e-6	2.18e-6	2.95e-6	8.43e-7
W-B_FULL_s_rp	1.57e-6	2.45e-6	2.79e-6	2.33e-6	3.20e-6	1.02e-6
W-B_noCC_s_rp	1.85e-6	2.14e-6	3.87e-6	1.57e-6	3.15e-6	9.39e-7
W-B_FULL_s_acc	1.93e-6	3.04e-6	3.00e-6	1.73e-6	3.16e-6	1.04e-6
W-B_noCC_s_acc	1.95e-6	2.11e-6	3.89e-6	1.70e-6	3.07e-6	8.02e-7
W-B_FULL_adap	7.79e-7	6.42e-7	1.73e-6	1.53e-6	3.19e-6	8.54e-7
W-B_noCC_adap	8.96e-7	6.69e-7	1.85e-6	1.69e-6	3.11e-6	9.09e-7
S-L_FULL	1.44e-6	2.90e-6	6.59e-6	4.07e-6	3.46e-6	1.12e-6
S-L_noCC	1.80e-6	2.62e-6	9.30e-6	4.49e-6	3.40e-6	1.01e-6
S-L_FULL_s_rp	1.45e-6	2.14e-6	5.89e-6	3.61e-6	3.42e-6	1.14e-6
S-L_noCC_s_rp	1.87e-6	1.98e-6	9.43e-6	3.61e-6	3.29e-6	9.07e-7
S-L_FULL_s_acc	1.53e-6	2.38e-6	5.99e-6	3.32e-6	3.57e-6	8.83e-7
S-L_noCC_s_acc	1.91e-6	2.19e-6	8.44e-6	4.09e-6	3.35e-6	1.02e-6
S-L_FULL_adap	1.06e-6	9.18e-7	4.44e-6	3.64e-6	3.45e-6	1.11e-6
S-L_noCC_adap	1.26e-6	1.07e-6	5.07e-6	3.60e-6	3.42e-6	8.99e-7
PA10_vel	1.14e-2	5.24e-5	5.13e-5	3.86e-5	2.02e-4	3.19e-5

Table F.13: Comparison of high to low frequency control torque power ratio for different controllers, platform trajectory: sea_all_2, PA10 trajectory: **`sine`**.

controller	S1	S2	E1	E2	W1	W2
PD	2.20e-5	1.60e-4	3.55e-5	1.57e-5	2.28e-5	1.14e-5
sat_sl	2.12e-5	1.58e-4	3.45e-5	1.51e-5	2.31e-5	1.17e-5
slid	2.16e-5	1.58e-4	1.00e-4	5.12e-6	2.30e-5	1.15e-5
PID	2.15e-5	1.59e-4	3.58e-5	1.57e-5	2.30e-5	1.12e-5
grv_frc_PD	2.54e-6	2.62e-6	4.43e-6	8.64e-7	1.46e-6	7.66e-7
grv_frc_PD_s_rp	2.55e-6	2.37e-6	4.04e-6	9.17e-7	1.54e-6	9.08e-7
grv_frc_PD_s_acc	2.51e-6	2.38e-6	3.85e-6	9.22e-7	1.53e-6	8.67e-7
grv_frc_PD_adap	9.51e-7	1.14e-6	2.72e-6	9.04e-7	1.48e-6	8.35e-7
grv_frc_sat_sl	2.61e-6	2.97e-6	4.57e-6	1.03e-6	1.83e-6	8.75e-7
grv_frc_sat_sl_s_rp	2.85e-6	2.86e-6	4.36e-6	1.02e-6	1.77e-6	1.05e-6
grv_frc_sat_sl_s_acc	2.82e-6	2.81e-6	4.07e-6	1.01e-6	1.73e-6	1.02e-6
grv_frc_sat_sl_adap	9.58e-7	8.68e-7	4.02e-6	1.12e-6	1.83e-6	1.06e-6
grv_frc_slid	2.60e-6	2.66e-6	1.18e-5	1.28e-6	1.59e-6	7.53e-7
grv_frc_slid_s_rp	2.67e-6	2.49e-6	1.06e-5	1.28e-6	1.55e-6	8.78e-7
grv_frc_slid_s_acc	2.82e-6	2.63e-6	1.06e-5	1.31e-6	1.62e-6	9.59e-7
grv_frc_slid_adap	1.11e-6	1.31e-6	6.91e-6	1.12e-6	1.53e-6	7.92e-7
lin_FULL	3.53e-6	1.13e-6	4.39e-6	3.30e-6	3.60e-6	5.61e-7
lin_noCC	4.05e-6	1.05e-6	4.52e-6	3.28e-6	3.58e-6	6.12e-7
lin_FULL_s_rp	3.93e-6	1.10e-6	4.29e-6	3.45e-6	3.82e-6	6.19e-7
lin_noCC_s_rp	4.31e-6	9.86e-7	4.42e-6	3.48e-6	3.77e-6	5.84e-7
lin_FULL_s_acc	3.95e-6	1.13e-6	4.32e-6	3.46e-6	3.88e-6	6.28e-7
lin_noCC_s_acc	4.22e-6	1.03e-6	4.45e-6	3.42e-6	3.89e-6	5.39e-7
W-B_FULL	1.71e-6	2.71e-6	4.43e-6	1.47e-6	1.62e-6	9.43e-7
W-B_noCC	1.68e-6	2.22e-6	4.97e-6	1.50e-6	1.60e-6	6.42e-7
W-B_FULL_s_rp	1.82e-6	2.24e-6	3.80e-6	1.46e-6	1.61e-6	7.82e-7
W-B_noCC_s_rp	1.75e-6	1.99e-6	4.63e-6	1.45e-6	1.59e-6	7.80e-7
W-B_FULL_s_acc	1.59e-6	2.36e-6	3.70e-6	1.35e-6	1.59e-6	7.73e-7
W-B_noCC_s_acc	1.77e-6	1.98e-6	4.52e-6	1.37e-6	1.60e-6	6.70e-7
W-B_FULL_adap	6.96e-7	9.71e-7	2.57e-6	1.63e-6	1.79e-6	7.20e-7
W-B_noCC_adap	7.34e-7	1.01e-6	2.40e-6	1.46e-6	1.65e-6	7.82e-7
S-L_FULL	1.36e-6	2.42e-6	9.52e-6	2.51e-6	1.75e-6	7.68e-7
S-L_noCC	1.55e-6	2.28e-6	1.14e-5	2.83e-6	1.87e-6	7.82e-7
S-L_FULL_s_rp	1.59e-6	2.10e-6	8.81e-6	2.42e-6	1.77e-6	8.78e-7
S-L_noCC_s_rp	1.64e-6	1.71e-6	1.09e-5	2.76e-6	1.92e-6	8.40e-7
S-L_FULL_s_acc	1.58e-6	2.32e-6	8.21e-6	2.63e-6	1.84e-6	7.68e-7
S-L_noCC_s_acc	1.61e-6	1.96e-6	1.00e-5	2.78e-6	1.90e-6	9.10e-7
S-L_FULL_adap	1.01e-6	1.23e-6	6.54e-6	2.49e-6	1.84e-6	8.51e-7
S-L_noCC_adap	1.13e-6	1.38e-6	6.42e-6	3.16e-6	1.93e-6	7.16e-7
PA10_vel	1.13e-2	4.18e-5	2.61e-5	2.13e-5	3.28e-5	2.03e-5

Table F.14: Comparison of high to low frequency control torque power ratio for different controllers, platform trajectory: sea_all_2, PA10 trajectory: **ssine**.