ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΜΗΧΑΝΟΛΟΓΙΚΩΝ ΚΑΤΑΣΚΕΥΩΝ & ΑΥΤΟΜΑΤΟΥ ΕΛΕΓΧΟΥ

Σχεδιασμός και Υλοποίηση Σερβομηχανισμού Ρυμούλκησης Υποβρυχίων Ρομποτικών Συστημάτων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Τσούνη Ι. Βασίλειου

Επιβλέπων: Ευάγγελος Γ. Παπαδόπουλος
Καθηγητής Ε.Μ.Π.

ΕΡΓΑΣΤΗΡΙΟ ΑΥΤΟΜΑΤΟΥ ΕΛΕΓΧΟΥ

Αθήνα, Οκτώβριος 2014

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΜΗΧΑΝΟΛΟΓΙΚΩΝ ΚΑΤΑΣΚΕΥΩΝ & ΑΥΤΟΜΑΤΟΥ ΕΛΕΓΧΟΥ
ΕΡΓΑΣΤΗΡΙΟ ΑΥΤΟΜΑΤΟΥ ΕΛΕΓΧΟΥ

Σχεδιασμός και Υλοποίηση Σερβομηχανισμού Ρυμούλκησης Υποβρυχίων Ρομποτικών Συστημάτων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Τσούνη Ι. Βασίλειου

Επιβλέπων: Ευάγγελος Γ. Παπαδόπουλος
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή στης 24 Οκτωβρίου 2014.

.....	.....	.....
Ευάγγελος Γ. Παπαδόπουλος	Κων/νος Τζαφέστας Επικ.	Δημήτριος Σούντρης Επικ.
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνρα, Οκτώβριος 2014.

.....

Τσουνης Βασιλειος
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών
Ε.Μ.Π.

Copyright © Βασίλειος Τσούνης, 2014
Με επιφύλαξη παντός δικαιώματος. All right reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Ο υπολογισμός υδροδυναμικών δυνάμεων, ασκούμενων σε υποβρύχια ρομπότ συνδράμουν σημαντικά στο μηχανολογικό σχεδιασμό καθώς και στην ανάπτυξη ελεγκτών πραγματικού χρόνου για τέτοιου είδους συστήματα. Η έρευνα στο πεδίο αυτό ενισχύεται εάν υπάρχει πρόσβαση σε πειραματική διάταξη η οποία μπορεί να παρέχει άμεσες μετρήσεις υδροδυναμικών δυνάμεων όπως η ιξώδης τριβή. Αυτή η διπλωματική εργασία πραγματεύεται τον σχεδιασμό και την υλοποίηση ενός πρότυπου σερβομηχανισμού ρημούλκησης ο οποίος λειτουργεί ως πειραματική πλατφόρμα για την υποστήριξη έρευνας σε μικρού μεγέθους βιομιμητικών ρομποτικών ιχθύων. Ο σχεδιασμός της εν λόγω διάταξης βασίζεται σε συστήματα ρυμούλκησης τα οποία χρησημοποιούνται στον τομέα ναυπηγικής μηχανικής και έχουν δοκιμασμένη αξιοπιστία και χρησιμότητα. Τέτοιου είδους συστήματα χρησημοποιούνται για τον έλεγχο κίνησης δοκιμαστικού σκάφους το οποίο προσδένεται στον μηχανισμό καθώς μετρούνται οι ασκούμενες δυνάμεις είτε απο αισθητήρα δύναμης είτε από αδρανιακό αισθητήρα. Η πρωτοτυπία της διάταξής οφείλεται σε τρείς παράγοντες. Πρώτον, το σύστημα είναι ασύγκριτα απλούστερο των προαναφερθέντων και με σημαντικά μειωμένο κόστος και βάρος. Δεύτερον, είναι σχεδισμένο να λειτουργεί χρησημοποιώντας ανοιχτού-σχεδιασμού ηλεκτρονικά, καθώς και χαμηλού κόστους ενσωματωμένο υπολογιστικό σύστημα και ανοιχτό λογισμικό ανεπτυγμένο στο Εργαστήριο Αυτομάτου Ελέγχου. Τέλος, το σύστημα ελέγχεται σε ασκούμενες δυνάμεις με χρήση ενισχυτών για τους κινητήρες οι οποίοι παρέχουν άμεσο έλεγχο ρεύματος τυμπάνου, με αποτέλεσμα να επιτυγχάνεται μέτρηση των υδροδυναμικών δυνάμεων και χωρίς την χρήση εξωτερικού αισθητήρα. Ως μέρος των δραστηριοτήτων έρευνας για την υλοποίηση του σερβομηχανισμού, αναπτύχθηκε και επιβεβαιώθηκε πειραματικά μια πρότυπη μεθοδος για την μέτρηση δυνάμεων χωρίς τη χρήση αισθητήρα.

Λέξεις Κλειδιά

Μηχανοτρονική, Ενσωματωμένα Συστήματα, Σχεδιασμός Συστημάτων Ελέγχου, Μηχανική Τριβή, Υδροδυναμική Τριβή, Αναγνώριση Συστήματος, Υποβρύχια Ρομπότ, Μετρήσεις Δυνάμεων, Έλεγχος Ροπής, Συστήματα Ρυμούλκισης σε Δεξαμενή, Καρτεσιανοί Σερβομηχανισμοί.

Abstract

Estimation of hydrodynamic forces acting on underwater robots plays a major role in both the mechanical design as well as the development of real-time controllers for such systems. Research in this field is better enabled if experiments can be executed which can provide direct empirical measurements of the hydrodynamic forces such as viscous drag. This thesis involves the design and control of a novel tow-tank carriage servomechanism which serves as a test-bed platform for experiments involving a small-scale bio-mimetic robotic fish. Our platform is inspired in part by towing systems used in naval engineering which have a proven reliability and applicability for use in these types of experiments. These systems have traditionally been used to execute controlled motions of a an attached test vehicle while an external force or inertial measurement sensor can provide the necessary information about the forces. Our innovation in this regard is threefold. Firstly, the system is incomparably less cumbersome to traditional towing systems and involves significantly reduced costs and overall weight. Secondly it is designed to operate using an open-hardware low-cost embedded system and runs on open-source software developed at CSL. Finally it is directly controllable in force/-torque due to current control enabled motor amplifiers and this property allows us to even achieve measurements of the hydrodynamic forces without the use of an external sensor. As part of the R&D activities for developing the platform, a novel and multi-faceted method for sensor-less force estimation is proposed and verified experimentally by comparing against the measurements of a 6-DoF force/torque sensor.

Keywords:

Mechatronics, Embedded Systems, Control System Design, Mechanism Friction, Hydrodynamic Drag, System Identification, Underwater Robotics, Force Estimation, Torque Control, Tow-tank Systems, Cartesian Servomechanism.

National Technical University of Athens
School of Electrical and Computer Engineering

Diploma Thesis

A Tow-Tank Carriage Servomechanism for Experiments in Underwater Robotics

Author:
Vassilios Tsounis

Supervisor:
Prof. Evangelos Papadopoulos
Dept. of Mechanical Engineering

A thesis submitted in fulfilment of the requirements
for the degree of Diploma in Electrical and Computer Engineering

in the

Control Systems Lab - Evangelos Papadopoulos Research Group

Athens, October 2014

For my father John, for showing me the importance of persistence... For my mother Katerina, who helped me understand patience...

Acknowledgements

The author would like to express his gratitude to his supervisor, Prof. Papadopoulos for his unending support and guidance.

Also, the author would like thank his sub-supervising PhD, Michail Makrodimitris for all the extensive discussions and late night debugging and writing of last-minute paper submissions.

Many thanks are also given to all the colleagues at CSL who provided an extremely fun environment to work in and be creative. This especially regards Klaijd Lika, Kostas Koutsoukis, Kostas Machairas, and Thanasis Mastreogeorgiou.

“ I can live with doubt, and uncertainty, and not knowing. I think it’s much more interesting to live not knowing than to have answers which might be wrong. I have approximate answers, and possible beliefs, and different degrees of certainty about different things, but I’m not absolutely sure about anything, and in many things I don’t know anything about, such as whether it means anything to ask why we’re here, and what the question might mean. I might think about it a little, but if I can’t figure it out, then I go to something else. But I don’t have to know an answer. I don’t feel frightened by not knowing things, by being lost in a mysterious universe without having any purpose, which is the way it really is, as far as I can tell, possibly. It doesn’t frighten me.”

Richard P. Feynman

“I may not have gone where I intended to go, but I think I have ended up where I needed to be.”

Douglas Adams

Contents

List of Figures

List of Tables

Abbreviations

A/D	**A**nalog **D**igital Mixed Signals
ADC	**A**nalog **D**igital **C**onversion
AI	**A**rtificial **I**intelligence
API	**A**pplication **P**rogramming **I**nterface
AUV	**A**utonomous **U**nderwater **V**ehicle
BBMC	**B**eagle**B**one **M**otion **C**ontroller
BDM	**B**ackward **D**ifference **M**ethod
BEM	**B**oundary **E**element **M**ethod
BG	**B**ond **G**raphs
CAN	**C**ontroller **A**rea **N**etwork
CLI	**C**ommand **L**ine **I**nterface
CSL-BRF	**CSL B**iomimetic **R**obotic **F**ish
CSV	**C**omma **S**eparated **V**alue
DC	**D**irect **C**urrent
DoF	**D**egree **o**f **F**reedom
EEP	**E**nd **E**ffector **P**latform
EG	**E**arth **G**round
EMCP	**E**mbedded **M**otion **C**ontroller **P**latform
GenoM	**Gen**erator **o**f **M**odules
GPIO	**G**eneral **P**urpose **I**nput **O**utput
GUI	**G**raphical **U**ser **I**nterface
HLC	**H**igher **L**evel **C**omponent
I2C	**I**nter **I**ntergrated **C**ircuit
IDE	**I**nteractive **D**evelopment **E**nvironment
LU	**L**oad **U**nit

LVCMOS	**L**ow **V**oltage **CMOS**
MACS	**M**ulti **A**xis **C**arriage **S**ervomechanism
MLC	**M**id **L**evel **C**omponent
NM	**N**odes **M**ethod
OROCOS	**O**pen **RO**botics **CO**ntrol **S**oftware
OS	**O**perating **S**ystem
PMM	**P**lanar **M**otion **M**echanism
PS	**P**ower **S**upply
PWM	**P**ulse **W**idth **M**odulation
R& D	**R**esearch & **D**evelopment
ROCK	**RO**botics **C**ontruction **K**it
ROS	**R**obot **O**perating **S**ystem
ROV	**R**emotely **O**perated **V**ehicle
RTOS	**R**eal **T**ime **O**perating **S**ystem
SDK	**S**oftware **D**evelopment **K**it
SI	**S**ystem **I**dentification
SPI	**S**erial **P**eripheral **I**nterface
SS	**S**upport **S**tructure
STO	**S**ubmerged **T**est **O**bject
TC	**T**owing **C**arriage
TO	**T**est **O**bject
UART	**U**niversal **A**synchronous **R**eceiver **T**ransmitter
URETTS	**U**nderwater **R**obotics **E**xperimental **T**ow **T**ank **S**ervomechanism
USB	**U**niversal **S**erial **B**us

Part I

Motivation

Chapter 1

Introduction

This chapter introduces the content of this thesis by outlining the motivations and the goals which drove the presented work. In order to achieve this, we begin with a review of the relevant literature guiding most aspects of the design and providing necessary theoretical and technical background.

Subsequently, the motivations for developing the experimental platform are described. These are regarding research activities in underwater robotics and biomimetic systems conducted at the *Control Systems Lab* (CSL), of the Dept. of Mechanical Engineering at the National Technical University of Athens (NTUA).

The aforementioned research requires the design and construction of a towing system which can execute experiments that will enable the dynamic modeling of a robotic fish. This is our end goal and it defines the scope of this thesis. Therefore, this chapter presents an outline of the required experiments and research activities which the platform will have to support. Lastly, the structure of this thesis and an overview of the covered topics are described.

1.1 Literature Review

A major challenge in developing an experimental robotic platform is the fact that an inherently interdisciplinary approach is required. As such systems integrate several different, yet interconnected aspects, it becomes increasingly important to obtain a sufficient familiarity with the relevant disciplines. In this regard, reviewing the past and present literature in each related field enables us to extend the current state-of-the-art.

Our starting point of course is underwater robotics. However, this only aids in understanding the types of experiments which are required of the platform. Apart from this functional dependency, there is no other direct link to this field. Rather, our system is more closely related to other fields such as dynamical modelling, control, actuation and transmission design, real-time systems, as well as various other specialized disciplines.

Thus, to retain a logical sequence, we start from the material relevant to the higher-level design and work our way down to the more specialized fields regarding specific aspects.

Underwater Robotics

Research conducted at CSL in biomimetic and underwater systems initiated with the development of a small scale biomimetic robotic fish, [1]. The first area on which this work focused is with respect to the design and control of the locomotion mechanism which utilized an under-actuated fin-like propulsion system.

Further research in [2] presented control algorithms which would enable the robot to perform controlled maneuvering based on commands given by an operator via teleoperation. These controlled motions enabled the approximation of external hydrodynamic forces via estimations based on the trajectories. This however, could only provide *coarse* estimates of the external forces.

Therefore, the need was recognized for an external system to regulate the motion of the fish in order to reliably assume a known executed trajectory.

Towing Carriages

Naval engineering fields have long been using towing carriages to move scaled mock-ups of vessels over large water tanks, [3] and [4]. These are also used in the development of medium to large scale underwater vehicles, specifically for testing the performances of propulsion systems, [5] and [6], as well as modeling of underwater vehicle dynamics, [7] and [8]. More recent research, [9], uses towing-tank experiments to design and test autonomous naval vessels.

This approach has also seen adoption of so-called Planar Motion Mechanisms (PMMs) in the automation and robotics fields for designing and testing the propulsion systems of underwater vehicles, [10] and [11]. These are equivalent to classical tow-tank systems with the primary difference being that the latter considered only forward linear motion, while the PMMs pertain to full 3 Degrees of Freedom (DoFs) Cartesian planar motion and orientation.

From these we are thus motivated to design the aforementioned external mechanism as a 2D Catesian PMM/Towing system that maintains accurate motion of the robot fish.

Multi-Axis Systems

Considering that PMMs and tow-tank systems are quite large and require specialized facilities, our system requires scaled-down design characteristics. This means that everything must be adapted to dimensions which enable lighter and simpler mechanisms to be used for the motion of a towing system over a small water tank.

Thus, we must consider that in terms of actuation, our system is more similar to *multi-axis* systems like high-precision positioning CNCs, [12] and [13]. Well known in the controls literature is the work of Tomizuka et al., who has been prolific in the area of multi-axis systems and precision motion control, [14], [15], [16], and [17]. Work in this field has focused on the design of controllers for motor actuated Cartesian systems that can regulate task-space motion with increased accuracy compared to classical PID methods, [18] and [19].

System Identification

Although Least Squares (LS) techniques have existed for quite some time, within an engineering context System Identification (SI) was introduced in its current form by the work of L. Ljung. A significant portion of his early research has been organized into his well known text on the matter, [20]. Other such texts in this field which have been particularly useful with newer techniques include those of [21] and [22].

The tools prescribed by SI, have proven invaluable in the various fields of engineering and especially in dynamics, control, mechatronics and robotics. These enable the development of accurate dynamical and/or statistical models for describing different types of systems.

In the case of tow-tank experiments, SI methods can be used to identify both the intrinsic dynamics of the towing system, as well as the coupled dynamics of both carriage and test object. The latter is the body on which the hydrodynamic forces act and whose model parameters are of primary interest, such as the robotic fish.

Mechanism Friction

Friction in mechanisms has long been one of the standard phenomena included in dynamical models of actuators. Although many robotics applications do not take friction into account, when high-precision control of actuators is required, friction modelling and compensation techniques are adopted, [17], [23] and [24]. A primary resource in this thesis has been the work of Helouvry and Khatib, [25]. Tomizuka has also contributed significantly in this field with [26] and [14].

Past work at CSL in [27] and [28], also provided some of the original motivation for exploring the dynamics of friction and the use of model based control methods to compensate their effects in actuation. Since the towing system designed and constructed at CSL uses only rolling elements for momentum transfer, modeling and compensation of dynamic and kinetic friction can enable better velocity control, which in turn improves the reliability of the towing experiments.

Joint Elasticity

As is described in Chapter 3, the mechanical drive system of the towing-carriage exhibits oscillations which are attributed to joint elasticity. In this area of research, prominent work includes that of Siciliano et al, [29], De Luca [30], Khalil and Kokotovic [31].

These systems are known to exhibit different behavior at different time scales and designing specialized controller algorithms to mitigate the effects occurring oscillations can certainly improve the overall confidence in executed motions.

Velocity Estimation

Another specialized topic addressed in this thesis is accurate real-time estimation of velocity using motor rotary encoders. Due to the inevitable quantization of the measurable state, implementing accurate real-time digital control systems requires advanced algorithms and hardware to properly regulate motion. Work exists within the literature which covers this topic and proposes techniques for different ranges of velocity, [32] and [33].

Robotics Software

As robotic systems become ever more complex, the need arises for specialized software that aids in their management. Many projects both proprietary and open-source exist that can provide the required flexibility to integrate different components like sensors, actuators etc to construct complete robotic systems. Currently, some of the most well known systems which address this are ROS[1], [34], ROCK [2], [35], Orocos [3] and GenoM [4].

These systems however are designed to run in an operating system such as one of the many distributions of Linux. Currently, there exists no robotics framework which

[1]The Robotics Operating System,www.ros.org
[2]The RObotics Construction Kit, rock-robotics
[3]TheOrocos Project.
[4]Generator of Modules from, OpenRobots

is designed specifically for embedded systems and which is not run within an OS. The design of the CSL towing carriage in its first complete realization addresses the design of an embedded system to implement motion control with strict real-time requirements. It also serves the purpose of assessing the system's capabilities on a low level. This can then determine whether the transition to more complicated software frameworks is necessary and applicable.

Real-Time Digital Control

Since the system is to execute *hard real-time* motion control, it is absolutely necessary to understand the effects of discretization on its control. In this regard, the texts of Franklin et al., [36], Chen [37] and Nise [38] provided a sound theoretical foundation.

From the computational perspective however, several texts exist which address low-level device firmware, OS device drivers and real-time embedded software development. These have proven to be invaluable for the adoption of necessary techniques, standards and approaches in software developments. [39] and [40] cover real-time embedded software for micro-controllers, [41], [42] and [43] are useful for embedded Linux systems and [44] is a well known text for device drivers. Finally, [45] presents an interesting systems approach to real-time software analysis and design.

Force Estimation

Wind tunnels can be used to measure wind resistance. This could be used to compute the respective drag in water by equating the Reynolds numbers of the two media, [2]. Velocity measurements such as Particle Image Velocimetry (PIV) or Hot-Wire Anemometry, apply the conservation of momentum to calculate the effective forces acting on the moving body. This is described in [46].CFD software simulation techniques have also proven useful in providing non-experimental methods for calculating drag, [47]. Use of PMMs to rigorously collect steady-state and transient speed data of scaled models in a

tow-tank plus force sensor configuration. Use of basic physical principles such as standard weights and pulleys to conduct falling-time and speed measurements are conducted in [48] to estimate drag coefficients and added mass.

Other generally applicable methods for estimating external forces use System Identification (SI) algorithms. These approaches provide means for estimating - both online and offline - parameters of an assumed model structure by applying known excitations and collecting response data.

These usually include the use of Linear and Non-linear Observers like Kalman filters, Non-linear Disturbance Observers, [49], to measure the complete state. However, when applied to underwater systems, these approaches would require Inertial Measurement Units (IMUs) or external equipment such as cameras to recover the state of the system.

Unrelated to the previous, electrical measurements on motors can be used to estimate forces, [50]. This motivates the interesting question of whether certain hydrodynamic forces can be measured – at least indirectly – by measuring the current of a motor which drives an object across a water tank. We aim to show that for certain applications in robotics and control, using a platform such as URETTS, we can identify hydrodynamic drag forces acting on the submersible without needing expensive force sensors or IMUs.

1.2 Problem Statement

This section presents the formal definition of the problem driving the design of the experimental tow-tank platform. Having reviewed the existing literature on the topics relevant to our purpose, it is now necessary to specify the intended uses of the system.

Firstly, we describe the motivations driving the intended applications of the platform, i.e. the experiments involving underwater robotics, and how these lead to operational parameters. Then, based on the previous parameters, the individual sub-systems are determined and outlines for their operation are formed.

1.2.1 Motivation

Although Remotely Operated underwater Vehicles (ROVs) are probably the most prevalent form of underwater robots, there is growing interest in the development of Autonomous Underwater Vehicles (AUVs) for applications where teleoperation should be replaced by autonomy. In order to develop robust autonomous systems with a level of reliability suitable for industrial applications such as maintenance of underwater installations or pipelines, it is not enough to develop the components concerning autonomy, i.e. the Artificial Intelligence (AI) and system management layers in the on-board software, and just "plug" them into existing ROV configurations. Although this approach would make sense, in reality, the demanding requirements placed on such systems create the need for considering different alternatives, especially when it comes to mechanical design.

First of all, for autonomy to be even possible, the on-board computer must be able to accurately control the state of the system. Modelling of system dynamics provides the means to design controllers that can accurately regulate the motion of the robot in the presence of disturbances like those found in water due to turbulent flows etc, without humans providing corrections via teleoperation. Although mathematical models for the various hydrodynamic forces exist, it is still a significant challenge to experimentally measure them, especially in the case of estimating the actuation forces.

The design of the actuation system and its respective controller, affect almost all other aspects of the system. For example, in the case when the system must be capable of maintaining its operation for long periods of time without recharging batteries, actuation must be optimized in terms of power consumption. In this regard, progress in biomimetics research has shown (see [51]) that effective and efficient actuation can be implemented based on that of marine animals. Therefore in this case, dynamical modeling would involve not only the inertial and external hydrodynamic forces, but also the hydrodynamic thrust generated by actuated fins. This is the primary motivation driving research at CSL concerning biomimetic robotics using the CSL Biomimetic Robotic Fish (CSL-BRF), pictured in Figure 1.1.

The CSL-BRF is a low-cost, small form factor submersible robot controlled by an embedded microcontroller, custom electronics developed in-house and powered by a Li-ion battery. It is an under-actuated system since there is only a single motor driving

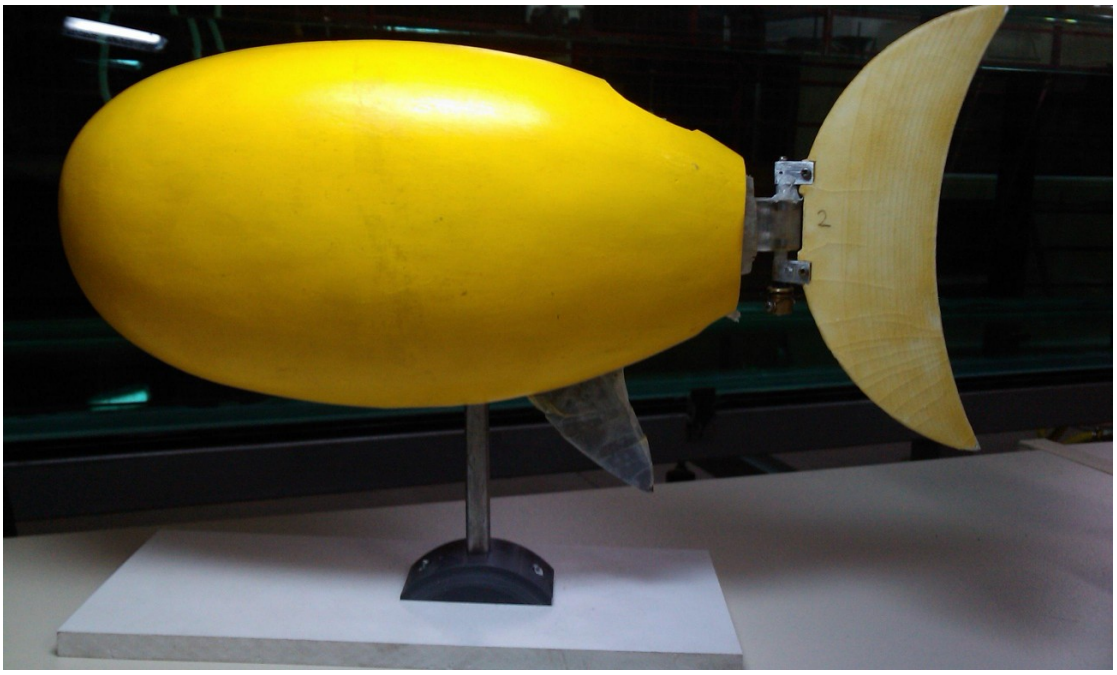


FIGURE 1.1: The CSL Robotic Fish.

a caudal fin for controlling horizontal movements, while vertical motion is executed via depth control using a low cost pressure sensor and DC pump and motor. An aspect of the research regarding this system has to do with developing a model based controller with feed-forward prediction of the hydrodynamic forces.

Specifically to the level of accuracy that underwater robotic systems require, three types of hydrodynamic forces become most significant; the external drag from the motion, added mass forces resulting from accelerations and decelerations and the forces produced by the fish itself via the actuated fin. To estimate the first two external forces, as mentioned in the previous section, there are various computational and experimental methods that can be applied which make use of computer models, simulation and even inertial sensor measurements. The acting forces however are another issue altogether. Although computational methods like Boundary Elements Method (BEM) using CAD models of the body plus fin can be used for estimation, the result is still a "ball park" estimation of the fin forces and so are not useful for feed-forward prediction.

For this purpose, the idea was proposed to design and implement an experimental platform, based on a 2-axis carriage, that could move on top of the tank and execute controlled motions while physically attached to the robotic fish. The initial concept involved the integration of a high fidelity multi-axis force/torque sensor into the set-up.

The carriage plus robot fish system could then execute specific motions in the task space, first with and then without the actuation of the caudal fin. This is the basic principle of operation on which all experimental protocols would be based on. This configuration is visualized in Figure 1.2.

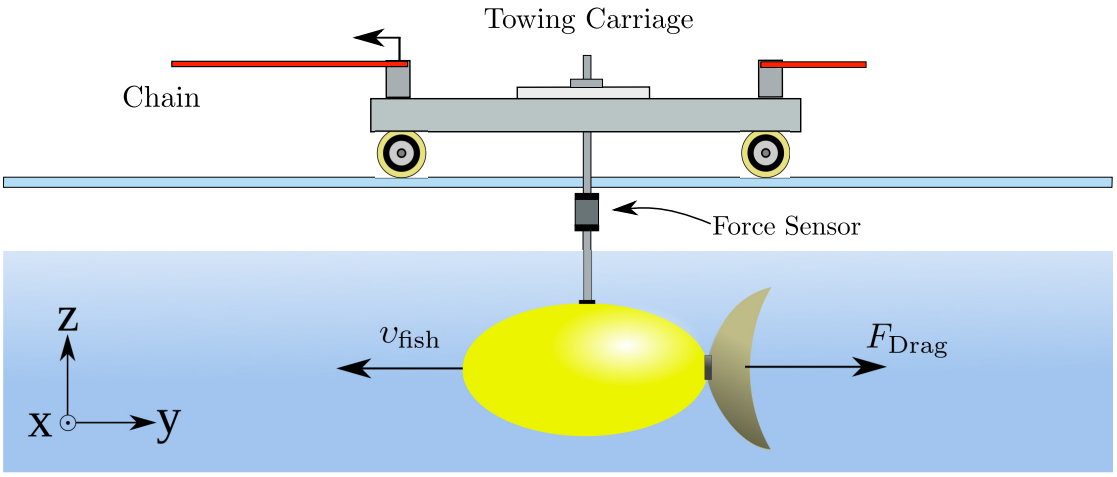


FIGURE 1.2: The towing-carriage plus robotic fish configuration.

1.2.2 Platform Functionality

Before we can specify values for the systems's design parameters, a concise description of its applications must first be considered. We therefore provide the following definition:

> The system developed within the scope of this thesis is to be known as the *Underwater Robotics Experimental Tow-Tank System* (URETTS). This platform is a tow-tank carriage servomechanism with multi-axis Cartesian manipulation whose primary application is the support of research activities in underwater robotics and biomimetic systems.

With this definition in mind we set the following goals to be achieved by the this thesis in the development of URETTS:

I. *Primary Goals*:

PG-1. Construction of a multi-axis linearly actuated servomechanism capable of 2-Dimensional planar end-effector trajectories.

PG-2. Development of an mechanism controller based on an embedded computer for rapid prototyping and testing of mechatronic and robotic systems.

PG-3. Establish a dedicated set-up for conducting experiments involving the CSL-BRF.

II. *Secondary Goals*:

SG-1. Introduction of the BeagleBone platform and assessing of capabilities for wider use at CSL.

SG-2. Increase CSL technical expertise with ARM microprocessors.

SG-3. Support R&D activities in the development of in-house embedded software.

SG-4. Development of re-usable electronics modules for interfacing of mixed-mode A/D signal devices.

SG-5. Assessment of open-source software tools as alternatives to well-known proprietary competitors.

Planned Experiments

In accordance with the previous goals, experiments which are of immediate interest for current research regarding CSL-BRF and define the core applications of URETTS are:

EXP-1. Experimental identification of the hydrodynamic drag via the two-step motor current comparison method described in [52], using linear motions in the task-space.

EXP-2. Experimental identification of the hydrodynamic drag via the two-step motor current comparison method using curved path motions in the task-space.

EXP-3. Experimental identification of the hydrodynamic drag with non nominal velocity relative to the body-fixed frame.

EXP-4. Verification and evaluation of desired path following of a robot executing maneuvers in the configuration-space.

EXP-5. Experimental identification and verification of thrust forces produced by CSL-BRF's actuators.

1.3 Thesis Structure

This thesis is divided into five primary parts:

1. Motivation
2. Analysis
3. Implementation
4. Application Layer
5. Post-Mortem

The first part involves the *motivation* for the work in this thesis by covering a preliminary aspects of this thesis and consists of the current introductory chapter, as well as Chapter 2. In the latter, the system requirements for the design of the URETTS platform are specified and an overview of the subsystems is described.

The second part consists of an analysis of the system and its dynamics. First, Chapter 3 presents dynamical models used to describe the behavior of the system and its various components. Chapter 4 presents the techniques used throughout the various experiments conducted for controlling the platform's motion.

Following the previous analysis, is the third part which deals with the *implementation* of the platform. Chapter 5 presents most of the work regarding the development of the hardware components, including both the mechanical and electronic components. Chapter 6 describes the embedded computer selected for the control of the system and presents the design of the software developed.

The fourth part involving the *application* of the implemented system is comprised of two chapters. The first, Chapter 7, presents the results of the various experiments involving the development of the system's dynamical model and control system. The

second chapter in this part, Chapter 8 demonstrates a direct application of the system for experiments with underwater vehicles.

Lastly, the final part of this thesis consists of Chapter 9, which presents the conclusions of this thesis and reviews the outcome of the implementation. Moreover, the last section of this chapter proposes the future directions and applications for the URETTS platform.

Chapter 2

Platform Design

In this chapter presents a preliminary review the design of the URETTS platform. We describe a set of goals for the project and determine the desired capabilities for the system. Furthermore, the set of requirements are presented in a listed form and are described. These requirements span the functional and operational aspects of the intended platform. The former pertain to the general operation and usage of the system while the latter pertain to the specific operations of each underlying components and how these work together in an integrated system.

Moreover, the structure of the total system and the segmentation into subsystems is described. We therefore elaborate on these by specifying a set of goals for the overall system, outline the desired capabilities and a describe the high-level organizational structure of the project. The individual parts comprising the whole are defined and specifications for their interactions are outlined.

Desired System Capabilities

This section covers the preliminary analysis for the design of our experimental tow-tank platform. Having reviewed the existing literature on the topics relevant to our purpose, it is now necessary to explore the requirements for the intended system. We outline the required end-user functionality as well as the operational requirements for the individual components of the complete servomechanism.

Firstly, we describe the motivations driving the intended applications of the platform, i.e. the experiments involving underwater robotics, and how these lead to operational parameters. Then, based on the previous parameters, the individual sub-systems are determined and outlines for their operation are formed.

With respect to the aforementioned experiments, several desired features can be determined for our desired system. Because each may increase the overall complexity or impose constraints upon the design, there must be clear reasoning for its inclusion. Following each feature presented below, the desired outcome is also described. The desired features are:

Feature 1 Multi-Axis manipulation

Outcome Provide the capability to execute trajectories within the system's configuration space.

Feature 2 Planar (Cartesian) trajectory tracking with sub-millimeter tracking error.

Outcome Planar because -at least initially- the experiments involve horizontal manoeuvring of the UR, i.e. no changes in depth, and also sub-millimetre because several experiments involve the comparison of certain data sets while others are assumed identical between iteration of the manoeuvres.

Feature 3 Low to medium loading capabilities.

Outcome The actuators must provide enough torque so that the combined carriage-load configuration must have almost the same bandwidth as the carriage by itself.

Feature 4 Wide velocity bandwidth that includes accurate high and low velocity tracking, even under loads.

Outcome Hydrodynamic forces acting on a given configuration, are dependent velocity, acceleration and also geometric properties. These must span a wide enough operational range in order to provide sufficiently accurate measurements both in low and high velocity regimes.

Feature 5 Configurable end-effector with capability to mount several types of sensors and/or additional actuators, e.g rotational θ_{xy}.

Outcome This is necessary due to some experiments involving mechanical coupling of carriage to UR while others do not. Additionally, the capability to control orientation is required for experiments involving the UR executing curved paths. Finally, this feature provides enough flexibility enabling future directions of research.

Feature 6 The initial configuration is to support the mounting of a multi-DoF force sensor.

Outcome Initial experiments involve direct measurement of hydrodynamic forces using a 6- ATI force sensor.

Feature 7 Orthogonal actuator axes of motion.

Outcome Decoupling of axes has the advantage of simplified forward and inverse kinematics which greatly decreases the computational requirements of implementing controller algorithms. Equally important is also the fact that these configurations have negligible cross-axis coupling forces.

Feature 8 Current control for motor actuator drives.

Outcome In conjunction with the previous feature, current control enables the direct control of the applied actuator (motor) torques.

User Interface & Mechanism Control

In terms of usability, the servomechanism's on-board computer must provide an adequate interface to the user. The intent is to eventually have a complete cross-platform Graphical User Interface (GUI) which can be run on a PC or laptop under Windows,

Mac or Linux. Through this GUI, the user is be able to control all the functionalities of the URETTS platform. The details of constructing and setting up such an infrastructure are discussed in the chapter 5, thus here we only provide an overview of the functionalities that are necessary for carrying out the planned experiments.

The user must be able to – at least – select and configure several predefined types of trajectories, exactly those that are required for the majority of the experiments. The controller system then commences operation of the motion drives and return the resulting execution data in an appropriate format. The available data-log can then be used in conjunction with other mathematical software for the processing of the experiment data. Additionally, debugging of hardware, software and controller algorithms must be possible, through the existence of appropriate facilities within the software, to test system components such as the sensors, actuators and other input-output devices.

From this general picture we can identify the following desired features in the system software:

FUI-1. A graphical (GUI) or Command Line Interface (CLI) textual UI.

FUI-2. Modular on-board system with a cross-platform user front-end.

FUI-3. Selection of trajectory type and input of relevant parameters buy the user.

FUI-4. Safety and test subsystems for error handling and error aversion.

FUI-5. Structured data-logging.

FUI-6. Compatibility for interfacing with other software tools either in the form of an Application Programming Interface (API) or data I/O.

2.0.1 Outline of Sub-Systems

In order to adequately plan the overall development of URETTS, a systems engineering approach is adopted, [53]. The project is segmented in such a way so that each subsystem may be developed independently of the others while only taking into account the way they interface. In fact, this interfacing serves as to prescribe specific requirements

for each sub-system. We can therefore identify three (3) High Level Components (HLCs) for URETTS.

The first corresponds to the mechanical design and construction of the carriage, i.e. the Multi-Axis Carriage Servomechanism (MACS), and is essentially the system of actuators plus motion drives as well as the key structural elements. The second component is the Embedded Motion Control Platform (EMCP) which is comprised of an embedded computer and the interfacing electronics. The third and final component is the BeagleBone Motion Controller (BBMC) software package, the core of the URETTS software stack. This controls the MACS via the EMCP, implementing the desired motion control algorithms. Following below are break-downs for each HLC.

The tree-graph depicted in Figure 2.1 visualizes the overall organization of the URETTS platform:

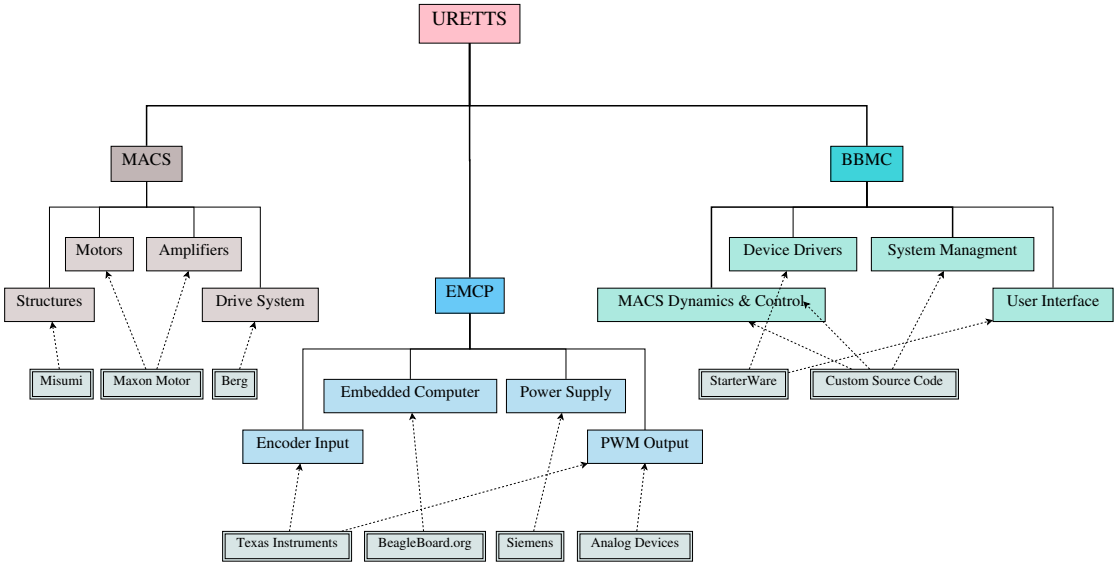


FIGURE 2.1: Project organization tree for URETTS depicting the sub-systems.

Multi-Axis Carriage Servomechanism

The MACS consists of three (3) structures, positioned around a parallelepiped water tank. The first two are support structures positioned on opposite ends across the length of the tank. They house the drive system for the long Y axis and remain fixed in their position. The third structure is a rectangular frame, whose motion is enabled by the previous drive system, and which itself houses the drive system for the smaller X axis drive system. This shorter X axis drives a smaller End Effector Platform (EEP)

onto which any experimental apparatus is mounted, e.g. the load, which can consist of a multi-axis force/torque sensor and/or the CSL-BRF. The drive system of each axis consists of a chain plus two sprockets, with one being driven by the motor with gear reduction.

The initial version of the MACS was completed in 2011 and its design and construction are described in detail in [54]. Even though certain modifications where made to the original, these did not significantly alter the design of the servomechanism. For this reason we only briefly outline the Medium Level Components (MLCs) in this section, as the reader is referred to the original documentation for further information. A concise description of this sub-system is presented in Chapter 5.

The MACS is organized into the following components:

MACS-1. Carriage Frame

MACS-2. End Effector Platform

MACS-3. Y-axis support structures.

MACS-4. X and Y axis chain drives.

MACS-5. X and Y axis motors.

Embedded Motion Control Platform

The EMCP is the platform proposed for the rapid prototyping and testing of motion controllers applicable to a wide range of mechatronics and robotics applications. With the recent popularization of low-cost, open-source and open-hardware Linux computers such as the Raspberry-Pi and the BeagleBoard, embedded computing is now more approachable than ever. Universities, research institutes, organizations and industry have long relied upon platforms such as PC104 stacks, dSPACE modules and other systems for their research. These platforms are flexible, reliable, robust and are considered as high-end products and entail high costs. Purchase costs in combination with costly license fees can usually bring the overall costs of such systems up to several thousand Euro.

While investigating candidate embedded computers for URETTS, low-cost and maintainability where prime factors. For this reason the choice was made to select one of the recent (circa 2012) open platforms, the BeagleBone from BeagleBone.org, and initiate the development of a dedicated open-source software stack for URETTS and CSL in general. With an active community and impressive array of both processing and input-output capabilities, the BeagleBone lies at the core of the EMCP, providing both low-level real-time execution as well as high-level calculations and planning.

Moreover, the EMCP must provide appropriate interfaces between the BeagleBone and the other electronics it is connected to. For instance, since the motor drive amplifiers are controlled using specific types of electronic signals, the interfacing electronics must not only create the appropriate signal channels, but also must provide sufficient protection to the hardware by way of isolation of signal and power layers. Finally, there should also be mechanisms in place which can ensure the safe operation of the MACS and can interrupt, limit and stop altogether its operation in erroneous conditions.

The EMCP therefore consists of the following MLCs:

EMCP-1. X and Y axis motor amplifiers.

EMCP-2. A single Power Supply (PS) with multiple output voltages.

EMCP-3. The BeagleBone embedded computer.

EMCP-4. Signal interfaces for outputs to motor amplifiers.

EMCP-5. Signal interfaces for inputs from encoders.

EMCP-6. Electromechanical safety end-stops for MACS operation.

EMCP-7. User-enabled kill-switches.

BeagleBone Motion Controller

As described in the beginning of this chapter, a plethora of options exist for the development of the system's software stack. The software managing and controlling the URETTS however, in this initial realization, is implemented as a custom software stack

running on a bare-metal system (i.e. no Operating System). Since the system is purposed to perform experiments which require a high degree of temporal accuracy, the added overhead of a complete OS like an embedded Linux OS, can increase the overhead at runtime, therefore reducing the capability to implement high-frequency real-time control.

Also, at the time of initial development no Real-Time Operatin System (RTOS) existed, neither open-source or proprietary, that was completely compatible with the BeagleBone and all of its functionality. Subsequently, for this initial iteration a custom software stack was developed, based on a open-source Software Development Kit (SDK) provided by the manufacturer of the BeagleBone's processor (see Chapter 6).

The BBMC software stack consists of:

BBMC-1. Base SDK software package.

BBMC-2. Customized hardware drivers for relative peripheral devices.

BBMC-3. The system's operational core and system management.

BBMC-4. Supporting functional facilities.

BBMC-5. Specialized services for experiment execution.

BBMC-6. User interface for command input and data retrieval.

2.1 System Requirements

In this section we present the requirements for the design of URETTS and specifically for the EMCP and BBMC. The reason for the exclusion of the MACS is that it already exists resulting from previous work conducted at CSL, and so now can only impose requirements onto the other HCLs of URETTS. Firstly, we list the functional requirements resulting from the discussion in the previous section. Thereafter, the operational requirements of each sub-system are listed.

2.1.1 Functional Requirements

The functional requirements describe the operation of the system from the user's perspective, not only in terms of user interaction, but also that of the capabilities it is expected to have. It is important to specify both qualitative and quantitative performances as these can drive the selection of components.

The functional requirements are therefore the following:

FR-1. Multi-Axis manipulation.

FR-2. Planar trajectory tracking with sub-millimetre tracking error.

FR-3. EEP loading capacity up to $20N$.

FR-4. Wide velocity bandwidth that includes accurate high and low velocity tracking, even under loads.

FR-5. Configurable end-effector with capability to mount several types of sensors and/or additional actuators, e.g rotational θ_{xy}.

FR-6. Support the mounting of a multi-axis force sensor.

FR-7. Negligible cross-axis force coupling.

FR-8. Current control enabled motor drive amplifiers.

2.1.2 Operational Requirements

The following operational requirements result from taking into account two aspects of each sub-system. The first, is obviously related to each sub-system's contribution the system as a whole, for meeting the previous functional requirements. The second aspect however, relates to the interfacing between each of URETTS's HLCs and their respective compatibility.

EMCP

The operational requirements for the EMCP are the following:

OR-EMCP-1 **Embedded Computing**

a. The primary processor must be 32bit with and provide hardware Floating-Point computation.

b. There must be compatibility with common communication protocols: Ethernet, Serial, CAN, I^2C, SPI, USB

c. Availability of multiple general-purpose digital timers.

d. Capable of producing PWM output signals.

e. Availability of ADC inputs.

f. Capable of decoding incremental encoder signals.

g. Availability of several GPIO channels.

OR-EMCP-2 **Signal and IO Interfaces**

a. Capability to receive 5V TTL signals and transform these to at least 3.3 LVCMOS.

b. Capable of receiving differential encoder signals and common-mode rejection.

c. Isolation of computer (signal stage) from motor power amplifiers (power stage).

d. Translation of output control signal generated from embedded computer to $-10/+10V$ DC signal.

e. Capability to be powered from $5V, 500mA$(max) supply.

OR-EMCP-3 **Hall Sensor Endstops**

a. A minimum of $3.3V$ and maximum of $5V$ supply voltage.

b. Capability of state (on-off) transition times less than $1ms$.

c. Availability of either latching or switching behaviour.

d. Capable of state transition due to presence of external magnetic field.

e. Capable of driving signal lines over cables longer than $10m$.

OR-EMCP-4 **Power Supply**

a. There must be voltage output channels at: $5V, 10V, 12V$

b. Minimum value for maximum continuous supply current of $5A$

c. Maximum permissible level of supply voltage (peak-to-peak) ripple at $150mV$.

d. Power input must be compatible with European $240V, 50$Hz mains.

e. Capability to provide true Earth Ground (EG).

BBMC

The operational requirements for the BBMC software stack are the following:

OR-BBMC-1. **User Interface**

a. Capability of online reconfiguration and re-execution.

b. Capability to receive experiment data in common file format (e.g. CSV).

c. Capability to execute standard benchmark trajectories with configurable parameters.

d. Availability of debugging and status monitoring facilities.

OR-BBMC-2. **Infrastructure**

a. Implementation hiding and level abstraction should be evident in the structuring.

b. Protection of internal state data from external accesses.

c. Existence of a uniform device driver model.

d. Capable of managing device driver states and dynamic reconfiguration of devices at runtime.

e. Capability to maintain and monitor multiple system states, i.e. global states with absolute position and local states relative to previous motion instances.

f. Capability to manage several interrupts services.

g. Capability to handle system meta-data (configurations and device states) and provide access to users.

OR-BBMC-3. **Real-Time Operation**

a. Capability to execute high frequency real-time control, with at least a 1 kHz control rate.

b. Availability of configurable control loop timer within the [1, 5k] Hz range.

c. Capability of retaining low execution jitter, less than 5% of total control period.

d. Capability to log real-time data.

e. Capability of measuring low velocity regimes, possibly below minimum quantization level of incremental encoder.

Part II

Analysis

Chapter 3

Modelling and Dynamics

This chapter introduces a formal treatment of the problem at hand by using analytical methods to obtain a better understanding of the system overall including complexities involved in its operation. Through the analysis of each aspect of its operation, we are able to design the electronics, software and algorithms necessary to obtain optimal operation of the system.

There are many techniques in physics and engineering for developing mathematical models of physical phenomena. Especially in engineering and specifically in control theory, mechatronics and robotics, having a reliable model to predict a system's behavior can prove an invaluable tool for developing advanced control algorithms. In fact, a precise model can be considered a commodity given that there exist a vast array of applications that consider a system to be a *black-box* and whose control is empirically designed. Examples of the techniques used in the previous and other engineering fields include graph-algebraic approaches like the Nodes Method (NM) for electrical networks and Bond Graphs (BG) to describe the distribution of power and energy in electromechanical systems. These however do not scale easily to systems which operate in multi DoF domains like mobile robots.

There are also techniques which are based on core principles of physics, centered on the framework of *Classical Mechanics*. Specifically, Newton-Euler and Lagrangian formulations are particularly popular in the field of robotics and these are in fact the tool of choice for describing the mechanical components used in this work. The advantage of using these methods compared to others like the BG and NM is that they are generally

applicable and simple in principle. Also, although they can become rather complicated due to the mathematics involved, they can handle the introduction of non-linear components.

The true power of using Lagrangians lies in that a system can be expressed in its underlying form, i.e the generalized coordinates and not in the emergent form that is perceived in a 3D Cartesian space. One really has to wonder how a single multi-variable *scalar* quantity can describe the complex behaviors exhibited in a vector space. This alone is enough to appreciate the elegance of the approach. This reduction to an underlying form enables us to control directly in the *joint space* of the system defined by the intrinsic state of the actuators. The Newton-Euler approach on the other hand provides a concise vectorial form which describes the internal structure of the system and its external interactions with the environment. In certain situations this enables better control of the system in the configuration *C-space* or the *task-space* where the interaction .

Let us outline the two methodologies used by specifying the steps for applying each:

Method I. **Newton-Euler**:

1. Define the masses $\mathbf{M_i}$ and moments of inertia $\mathbf{J_i}$ which contribute to the multi-body motion.
2. Define a global and inertial coordinate frame as the *G-frame* and subsequent *i-frames* for each rigid body in the system.
3. Determine kinematic and/or static relationships between reference frames in the form of rotation matrices. These can either be referenced to the Global G-frame or to any arbitrary *body-fixed* B-frame: ${}^{B}R_i$ is the rotational transformation and $\mathbf{v}_i$ and ω_i are the linear and angular velocities of each body i wrt to the B-frame.
4. Based on the *assumptions* made regarding the interactions in the *multi-body* system, determine the total set of acting forces for each body.
5. Apply the *Newton-Euler* multi-body equations of motion:

$$\sum F = {}^{B}R_i[m_i\dot{\mathbf{v}}_i + (\dot{\omega}_i \times m_i\dot{\mathbf{v}}_i)] \tag{3.1}$$

$$\sum N = {}^{B}R_i[\mathbf{M}_i\dot{\omega}_i + (\dot{\omega}_i \times \mathbf{M}_i\dot{\omega}_i)] \tag{3.2}$$

Method II. **Lagrange**:

For n state (spatial) coordinates and k degrees of freedom, use of the Lagrangian formulation consists of the following steps:

1. Define the *joint* variables as the generalized coordinates $q_k, \quad \forall k$
2. Define the *constraints* of the system in the form of *Pfaffian* equations:

$$\sum_{m=1}^{n} A_{m,k}(x_1, x_2, ..., x_n)dx_m = 0, \quad \forall k \tag{3.3}$$

3. Define the *Lagrangian* consisting of *kinetic* $\mathcal{T}$ and *potential* $\mathcal{U}$ energy components:

$$\mathcal{L} = \mathcal{T} - \mathcal{U} \tag{3.4}$$

$$\implies \mathcal{L} = \frac{1}{2}\sum_{i=0}^{n} m_i\dot{x_i}^2 + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} I_{ij}\dot{\theta}_i\dot{\theta}_j - \mathcal{U}(x_1, x_2, ..., x_n) \tag{3.5}$$

4. Introduce the constraints into the Lagrangian.
5. Insert resulting Lagrangian into the *Euler-Lagrange* equation:

$$\frac{d}{dt}(\frac{\partial\mathcal{L}}{\partial\dot{q}_i}) - \frac{\partial\mathcal{L}}{\partial q_i} = \mathcal{Q}_i, \quad \forall i = 1, 2, 3, ..., k \tag{3.6}$$

Where $\mathcal{Q}_i$ is the generalized force acting on each Degree of Freedom (DoF).

The reader is encouraged to see [55], [56], [57], [58] and [59] for further insight regarding classical mechanics and analytical dynamics. In this thesis, we express mostly scalar quantities but for multi-DoF cases, *vectorial notation* (see [58]) is used unless otherwise specified.

In this chapter we first address the mechanics of tow-tank systems and determine a set of assumptions, concerning the MACS's operation. Based on these assumptions, we develop mathematical models to describe the dynamics of the system and its components. Specifically we cover the different phenomena which effect the operation of each carriage axis such as the actuators, loads and the carriage itself. The chapter then concludes by addressing the issue of velocity estimation using incremental encoders and a hybrid algorithm is proposed for estimation across a wide range of velocities.

3.1 Tow-Tank Systems

Tow-tank mechanisms have been around for quite a while as naval engineers and researchers studying fluid dynamics have long been using tow-tank and PMM systems. Although the types of experiments used in the previous and related fields differ from what we want to pursue concerning underwater robotics, the basic idea is the same. If one attaches an object to a supporting shaft and then drags these across the length of a water tank, the forces acting on both objects load the actuator executing the motion.

In principle, neglecting the actuator's intrinsic losses (due to machine friction etc), the motion of the support and the test object is primarily determined by the exerted force of the actuator and the external hydrodynamic forces of the water. Figure 3.1 visualizes the drag force acting upon the robotic fish while being towed in the tank. Our goal is to measure the external forces produced by the object's motion within the tank.

Lets define any arbitrary object with *known* geometry as the Test Object (TO) and the supporting link to the towing carriage as the Support Shaft (SS). The part of the Towing Carriage (TC) onto which the previous are mounted, and which in fact exerts the actuation force, is defined as the End Effector Platform (EEP). We make these distinctions in order to also describe the masses contributing to the motion of the total system. When the TO is placed in the tank, it can also be referred to as a Submerged Test Object (STO).

In the next section we further analyze these masses, taking into account the MACS in detail, but for the moment it suffices to consider the effective masses as the four stated

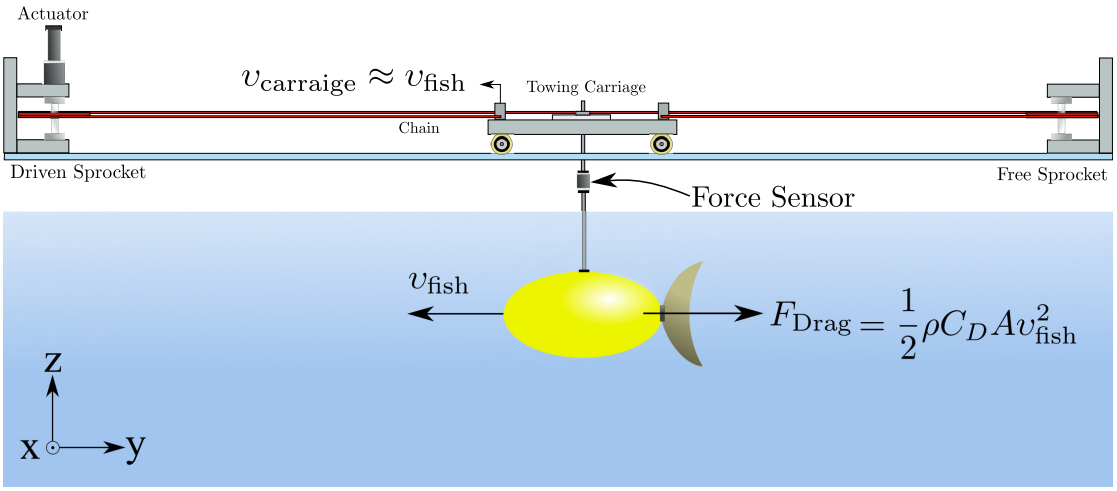


FIGURE 3.1: The Towing Carriage plus robotic fish configuration.

previously: $m_{\text{TO}}, m_{\text{SS}}, m_{\text{EEP}}, m_{\text{TC}}$. The combined TO-SS component is referred to as the Load Unit (LU) and is statically linked to the EEP. The LU is to be reconfigured depending on the type of experiment.

There are three caveats when considering this configuration. The first is concerning the assumption of rigidity. If the SS or the TO were to bend or distort due to elasticity, additional forces would occur which would induce vibrations in the mechanical structure and therefore oscillations during motion. This would also add further dissipation and internal dynamics to the system. Still, it is fair to assume that if the vibrations would be of high enough frequency, i.e. the system is rather stiff, then the vibrations could be considered negligible, possibly even lying within the range of measurement uncertainty.

The second caveat however, concerns the types of motions executed and specifically the range of velocities spanned. In fact, the motion profile itself determines the types of hydrodynamic forces that we should (or even could) take into account. Let's consider a simple case of linear motion (i.e. in a straight line) across the tank at a *constant* velocity. Since the system is not accelerating, then the only forces acting on the TO (plus SS) are a drag force due to the viscous flow of the water along the surface of the body and the forces exerted by the actuating EEP.

Also, the model adopted to describe hydrodynamic drag also depends on several parameters, amongst them, the maintained steady-state velocity. For example, if the velocity is in a lower regime, then the appropriate model would significantly differ from

that if the velocity was higher. Specifically, the two most common models used throughout the literature for describing viscous drag are:

$$F_D = \frac{1}{2}\rho C_D A \cdot \upsilon^2 \tag{3.7}$$

$$F_D = B_D \cdot \upsilon \tag{3.8}$$

Where ρ is the density of the fluid, C_D is the *drag coefficient*, A is the cross section surface area of the object, υ is the velocity of the object and B_D is the coefficient of low velocity viscous drag.

Proofs for (3.7) and (3.8) can be found in [60] and are rather simple. In fact, the proof for (3.8) is nothing more than the application of Newton's second law (force balance equation) combined with the distributed force as pressure along the surface of the moving body which is exposed to the fluid flow.

If however, the TO were to accelerate or decelerate then this would result *additionally* in another hydrodynamic phenomena known as the *added mass force*, F_{AM}. This occurs when the forces accelerating (or decelerating respectively) the TO would have to also (locally along the surface of the TO) do the same for the water being "pushed out of the way". This force is commonly described using the following model for simple geometries:

$$F_{AM} = \frac{1}{2}\rho V_s \ddot{x} \tag{3.9}$$

Where V_s is the volume of the TO and $\ddot{x}$ is its acceleration in the fluid.

The aforementioned hydrodynamic forces have long been studied and are found throughout the relevant literature. Their respective models can predict the actual forces to a good degree of accuracy, given the right conditions. Of course, fluids are notorious for their chaotic behavior, and so there are always more phenomena at play such as turbulent flows, but these tend to be minute in magnitude relative to the others. Therefore, like the possible vibrations due to elasticity, these can also be considered to lie within the range of measurement uncertainty.

The third and final caveat concerns the existence of the SS. Ideally, we would like to consider only the presence of the TO but a mechanical coupling between the TO and the EEP is necessary. The coupling provided by the SS however also contributes to the acting hydrodynamic forces therefore affecting the resulting measurements. Two alternatives are initially available. Either we neglect the contribution of the SS, or we also model the forces it produces and explicitly take it into account in the calculations. There is however another third option. Separate tests could be executed using identical motion profiles, first with *only* the SS and then without it. The resulting measurements from the two runs can be subtracted to determine the difference.

This subtractive approach is encountered again in Chapter 8. In any case, all three options can be valid, depending also on the range of values the measured forces have. Only empirical results from experiments can provide a definitive answer.

Lets summarize this section with a set of assumptions on the behavior of the load side with which we proceed in the continuation:

LU-A-1. The LU mechanism chain is considered fixed and *rigid*.

LU-A-2. Experiments are executed within an range of velocities where (2.2) can be used to model drag.

LU-A-3. Added Mass forces are described by (2.3).

LU-A-4. Other physical phenomena acting the fluid are considered to contribute to the responses as *random* disturbances.

The previous assumptions are general enough to remain valid even if the mechanical design of the LU unit changes. This *modularity* provides us with the flexibility to retain the same approach across different experimental set-ups. The design and analysis of the LU is now completely independent of whether or not we introduce additional capabilities to the actuation of the EEP. This also becomes apparent in Chapter 5 where we describe the design of the physical platform.

3.2 A Cartesian Servomechanism

In the previous section we described the physical interactions between the components of the experimental LU set-up and derived assumptions on its dynamic behaviour. Accordingly, we also do the same for the rest of the MACS's components and complete the set of assumptions concerning the mechanical sub-systems of URETTS. Firstly lets define the kinematic chain describing the mechanical relationships between the MACS's components. Figure 3.2 shows the mechanical CAD model of the MACS.

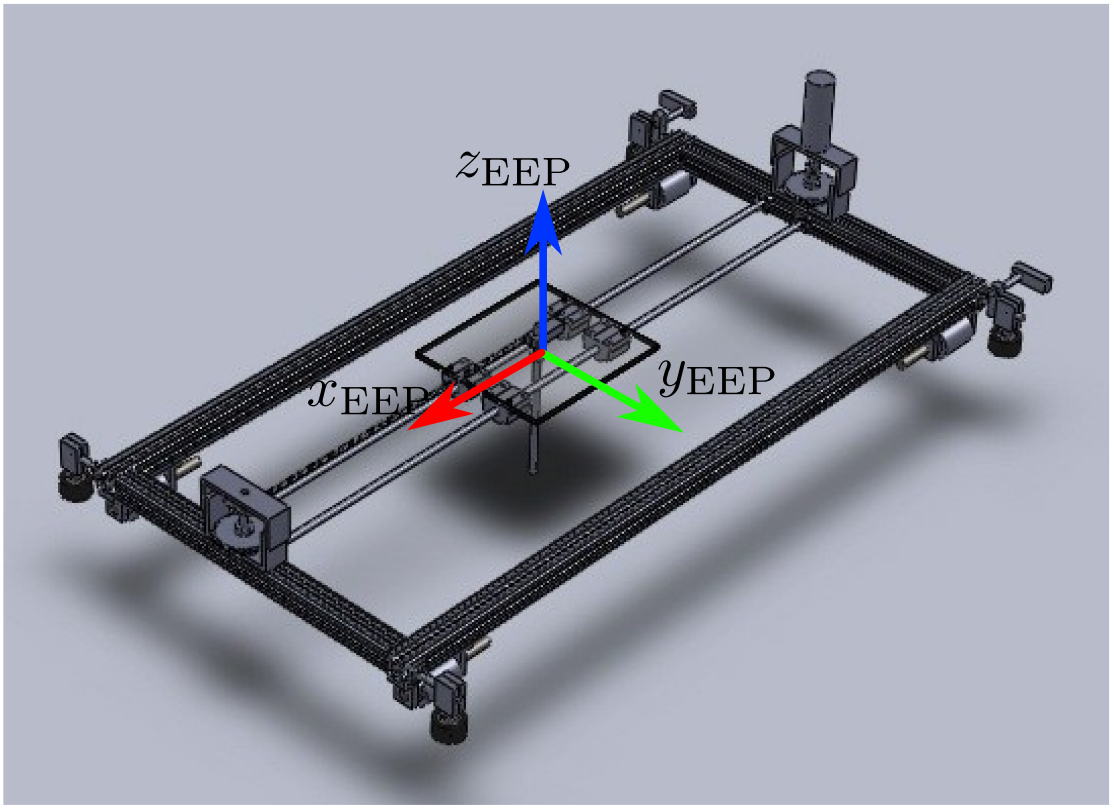


FIGURE 3.2: CAD model of the MACS

Considering the configuration outlined in Figure 3.1 and the detailed model in Figure 3.2, we can identify four types of elements comprising the kinematic chain:

1. *Primary Links*: these are the elements whose relative motions define the coordinates of the system's configuration space, e.g. the position of the carriage relative to the base structure defines the Cartesian Y axis.

2. *Actuators*: these elements initiate the motion for a particular DoF of the system.

3. *Payloads*: these are the elements whose motion defines the state of the system and therefore the total configuration space.

4. *Intermediate Links*: these are the *transmission* elements that are transferring the motion from the actuators to the primary links.

Figure 3.3 illustrates the complete kinematic chain of the MACS in the form of a tree graph with the base structure as the root and the load unit as the a leaf. To describe the kinematic relationships between the mass *nodes*, we use *edges* with unilateral arrows and with two-strikes in order to represent kinetic and static connections respectively. The aim is to show how one element affects another therefore contributing to the total dynamics. For example, the interactions between the chain and sprockets imply that multiple closures in the chain exists, such as the path 2-3-4-3-2. The significance of these closures becomes apparent when we explore more advanced models for MACS which account for oscillations of the chains (see Section 3.4).

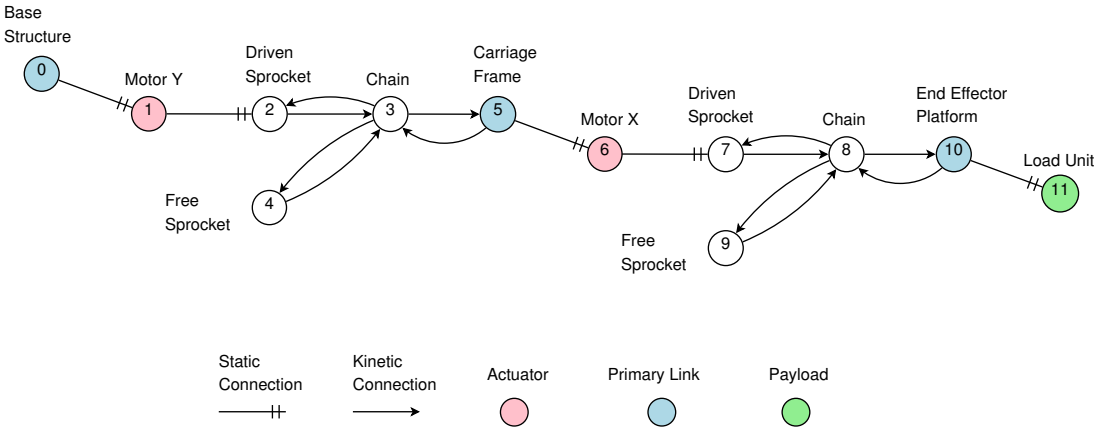


FIGURE 3.3: Kinematic Chain of the MACS.

Although we could cover this description in further detail by including more carriage parts, this would not provide any tangible advantage. In each axis drive, a motor drives an axle with a fixed sprocket. The physical connection between the motor output (planetary gear output) and axle exists via an elastic shaft coupler. Possible contributions of the shaft coupler are neglected from the kinematic chain.

It is clear from Fig. 3.3 that the kinematic configuration is identical between X and Y axes. In fact the X axis drive is nested within that of the Y axis. This linear chain-driven mechanism is commonplace in industry for systems ranging from gantries to 3D

printers. The motion of each axis is perpendicular to the others, resulting in negligible forces acting between them, i.e. the motion of one does not effect that of another.

Lets now define the masses and moments of inertia from the previous kinematic chain (node inertias) which are the effective inertias contributing to the dynamical behaviour of the MACS:

i. $J_{m,y}$: the moment of inertia of the Y axis motor's shaft.

ii. $J_{m,x}$: the moment of inertia of the X axis motor's shaft.

iii. $J_{sd,y}$: the moment of inertia of the Y axis driven sprocket.

iv. $J_{sd,x}$: the moment of inertia of the X axis driven sprocket.

v. $J_{sf,y}$: the moment of inertia of the Y axis free sprocket.

vi. $J_{sf,x}$: the moment of inertia of the X axis free sprocket.

vii. $m_{c,y}$: the mass of the Y axis chain.

viii. $m_{c,x}$: the mass of the X axis chain.

ix. m_{TC}: the total mass of the towing carriage (housings, motors, shafts etc) excluding the mass of the EEP.

x. m_{EEP}: the total mass of the EEP excluding the mass of the LU.

xi. m_{LU}: the total rest mass of the experimental payload.

With the assumption of rigidity between mass node sets $S_X := \{n_2, n_3, n_4, n_5\}$ and $S_Y := \{n_7, n_8, n_9, n_{10}\}$ representing the drive system of each Cartesian axis respectively, we can consider the following effective axis moments of inertia represented on the motor's side (due to the presence of the transmission's planetary gear-head):

$$J_Y = J_{m,y} + \frac{(J_{\mathrm{sd},y} + J_{\mathrm{sf},y})}{{n_y}^2} + \frac{{R_y}^2}{{n_y}^2} m_{\mathrm{TC}} \tag{3.10}$$

$$J_X = J_{m,x} + \frac{(J_{\mathrm{sd},x} + J_{\mathrm{sf},x})}{{n_x}^2} + \frac{{R_x}^2}{{n_x}^2} m_{\mathrm{EEP}} \tag{3.11}$$

Where, R_Y, n_Y, R_X, n_X are the sprocket radii and transmission ratios for the planetary gears for Y and X axes respectively.

Figure 3.4 illustrates the masses and inertias involved in the motion of the TC along the Y axis. The same configuration also applies to the motion of the EEP along the X axis within the TC's frame.

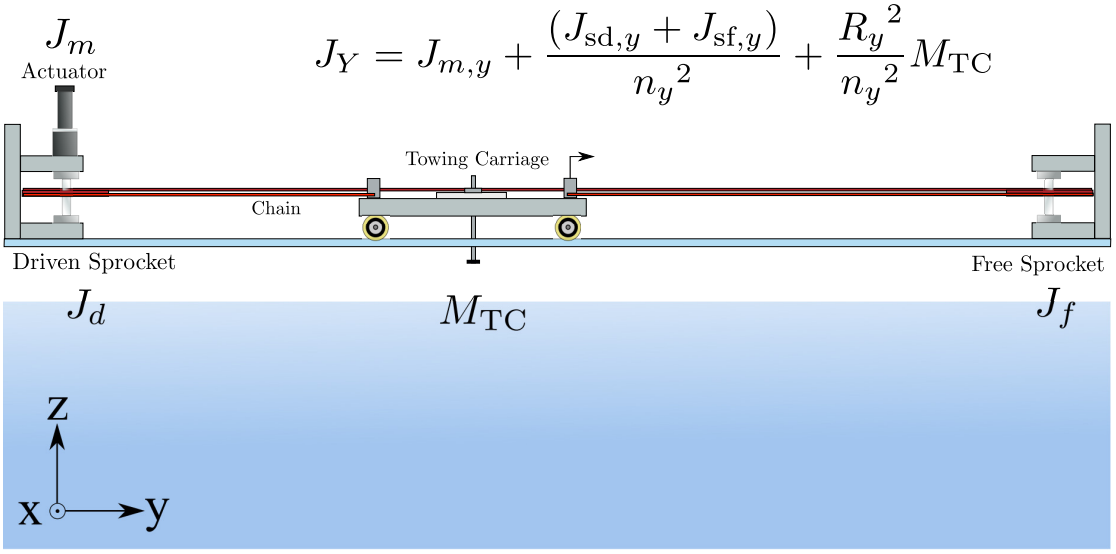


FIGURE 3.4: The Towing Carriage plus robotic fish configuration.

In terms of losses due to friction etc, and since all actuation is implemented through rolling elements, the use of ball bearings means that dissipation should be dependent on velocity. The specific model depends on the results of the identification experiments (see Section 4.5). Additionally, the carriage maintains its position and orientation relative to the lateral walls of the tank via side rollers (see Figure 3.2) meaning that misalignments can contribute to loading of the TC while traversing the tank. Similarly for the X axis, linear motion is implemented using linear bearings. The last two considerations could result in the existence of position dependent friction for each axis.

Furthermore, the misalignments of the TC mentioned previously, could also cause the carriage to rotate relative to the lateral walls therefore contradicting our previous assumption of negligible cross-axis forces. However this is a mere matter of calibration and testing in order to eliminate this rotation. Also, the LU while mounted on the EEP could cause downward forces further loading the X axis dynamics. A similar argument, however, can be made for the LU possibly pushing upwards due to buoyancy of the TO. These of course could almost cancel out (the ideal situation) but for simplicity we do

not consider these affect the dynamics of the system. Finally, we can neglect the effects of gravity since Cartesian motion is executed in the horizontal plane and the tank is positioned so to remove as much tilt as possible.

Having described the interacting masses and moments of inertia, the kinematic chain defining the motion of the system and the other possible effects, we summarize this section with the set of assumptions regarding the operation of the MACS:

MACS-A-1. The motion of the MACS is *defined* by the kinematic chain of Fig. 3.3.

MACS-A-2. The effective masses contributing to the dynamics are the ones corresponding to nodes of the kinematic chain.

MACS-A-3. The effective inertias of each motion axis (actuator DoF) are defined by equations (3.10) and (3.11).

MACS-A-4. Each axis drive is considered to be comprised of *rigidly* connected masses.

MACS-A-5. The TC does not rotate while in motion due to guiding lateral rollers.

MACS-A-6. The motion of one axis does not produce forces acting on another, i.e. the cross-axis forces are negligible.

MACS-A-7. Dissipation has a dependence on velocity.

MACS-A-8. Vertical forces produced by the EEP-LU connection are not considered.

MACS-A-9. Gravitational forces are negligible and are therefore ignored.

MACS-A-10. Operation in both directions of motion can be described by a single model and parameters.

3.3 System Dynamics

Having compiled the assumptions regarding the operation of the MACS, we develop the mathematical models for describing and predicting the behaviour of the system. To this end, we present an analysis of the system dynamics and relevant effects of these. The total system is analysed in terms of the contributing components. First we examine the

dynamics of the actuators and then an overview of mathematical models for describing the dissipative effects. Following these, a mathematical model for the motion of the TC and EEP is presented and finally we also analyse the effects of oscillations during motion and their sources.

Moreover, each system or sub-system model presented in the continuation is accompanied by an implementation in Mathworks' Matlab/Simulink. Quantitative results of these simulations are presented in Chapters 6 and 7 along with the experimental results.

3.3.1 Actuator Dynamics

Now lets consider the contributions of the motors' intrinsic dynamics. The model commonly used throughout the literature for describing the behaviour of Permanent Magnet Direct Current (PMDC) motors is a second order Linear Time-Invariant (LTI) system with the following form:

$$J_m \dot{\omega}_m = K_T i_m - \tau_d \tag{3.12}$$

$$L_m \frac{d}{dt} i_m + R_m i_m + K_T \omega_m = \nu_m \tag{3.13}$$

Where J_m $(Kg \cdot m^2)$ is the motor's intrinsic moment of inertia, ω_m $(^{\text{rad}}/_{\text{s}})$ is the rotational velocity of the motor's output shaft, K_m $(^{Nm}/_{A}$ or $^{Vs}/_{m})$ is the motor's torque constant, τ_d (Nm) is torque due to dissipative effects like friction, L_m (H) and R_m (Ω) are the armature's inductance and resistance respectively and ν_m is the applied voltage on the armature's terminals.

Motors characteristically have two time constants associated with their operation. The first corresponds to the mechanical behavior and is defined by the relationship between the moment of inertia and the dissipative elements which are dependent on velocity, e.g. viscous friction. The second corresponds to the electrical behavior in the armature and is defined by the ratio of inductance to resistance. The former in almost all cases is of an order of magnitude (at least) longer than the latter. This means that

for the purposes of control, there is reason to neglect the effects of the inductance all together and therefore consider the new system describing the actuators operation:

$$J_m\dot{\omega}_m = K_T i_m - \tau_d \tag{3.14}$$

$$R_m i_m + K_T\omega_m = \nu_m \tag{3.15}$$

$$\implies J_m\dot{\omega}_m = K_T(\frac{\nu_m - K_T\omega_m}{R_m}) - \tau_d$$

$$\implies J_m\dot{\omega}_m + \tau_d + \frac{K_T^2}{R_m}\omega_m = \frac{K_T}{R_m}\nu_m \tag{3.16}$$

Furthermore, motors are electro-mechanical by definition with ν_m as input and ω_m as output and thus, the first equation (3.12) represents the mechanical aspect of a motor while the second, equation (3.13), describing the electrical. An interesting fact about (3.12), is that the torque the motor produces is directly proportional to the current in the armature circuit. This plays a significant role in the design of the controller as one can assume – given that the current can be controlled directly – that the motor can be *controlled in torque* instead of velocity.

The previous capability is implemented in motor driver-amplifiers and is referred to as *Current Control* (CC). Thus, the two primary modes of operation for motor drives are:

i. *Voltage Control* (VC):

   Control of the voltage across the terminals of the motor's armature circuit. The input is ν_m and the output is ω_m:

$$\omega_m \propto \frac{1}{K_T}\nu_m \tag{3.17}$$

ii. *Current Control* (CC):

Control of the armature current by applying appropriate transient voltages. The input is the reference (desired) current i_r and the output is the applied torque of the motor τ_m:

$$\tau_m := K_T i_m \approx K_T i_r = K_T K_{\text{amp}} \nu_c \tag{3.18}$$

K_{amp} is the driver amplifier gain that produces an armature current proportional to the control voltage (i.e. control input) ν_c at the drivers input.

Even when CC is used, in reality, the armature voltage is still the signal that is regulated, while the motor armature current i_m is used as feedback and compared to a specified reference value i_r. This capability is provided by the electronics (i.e. the hardware) of the driver amplifier and is usually implemented with linear *PI* control using current feedback from specialized sensors. Also, since the electrical time constants are considered small (in the order of a μs), the transient behaviour of the current due to the PI control is much faster than that of the mechanical system, therefore the latter can never see variations above certain frequencies.

While only the dynamics of a PMDC motor actuator have been discussed thus far, this is still enough to commence the design of the simulation suite. This suite will consist of several models developed in Matlab/Simulink and are our primary tools for simulating and testing our assumptions against the experimental data. This also aids the design of any controller we would like to implement, as it allows us to examine its operation prior to testing on the physical system.

Taking this into consideration, two generalized simulation models have been developed to encapsulate the behaviour of a single MACS axis. The two variants differ only in whether they incorporate the PI controller for CC, or consider only Equ.(3.18) as the effective behaviour. The components incorporated include:

i. *Mechanical Plant*:

This is a single variable (i.e. DoF) Ordinary Differential Equation (ODE) plant with configurable load and friction sub-plants. The contribution of the LU to the simulation can toggled using the `NL_enable_*` variable. The equations for the implemented subsystem are:

$$J_M\ddot{\theta}_m + \tau_D = K_T i_m \tag{3.19}$$

$$\tau_D = \frac{1}{n}(\tau_d + \tau_{\mathrm{LU}}) \tag{3.20}$$

Where J_M $(Kg \cdot m^2)$ is the total moment of inertia as seen from the motor side, θ_m (rad) is the motor shaft rotation angle, τ_D (Nm) is the total load on the motor , n is the transmission gear reduction ratio and τ_{LU} (Nm) is the load torque exclusively from the motion of the Load Unit (i.e. excluding its inertia) and in the direction of the specific URETTS axis.

The models implemented in Simulink for the total Mechanical Subsystem and the Friction/Load models are shown in Fig. 3.5 and 3.6 respectively.

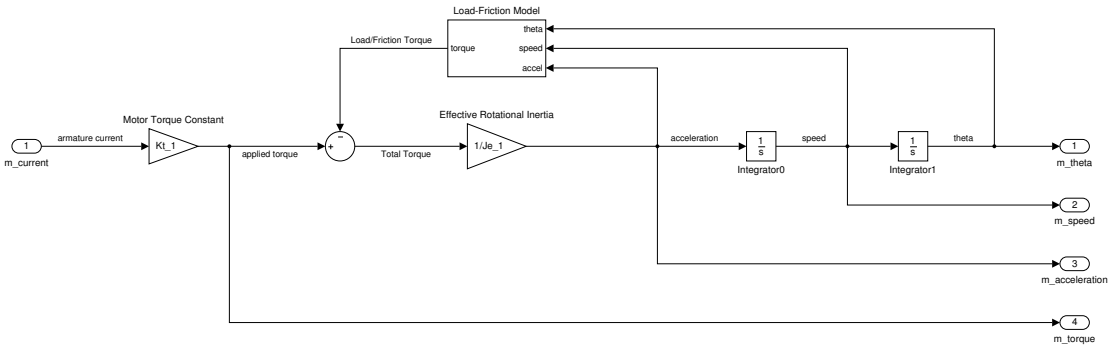


FIGURE 3.5: Mechanical plant in Simulink.

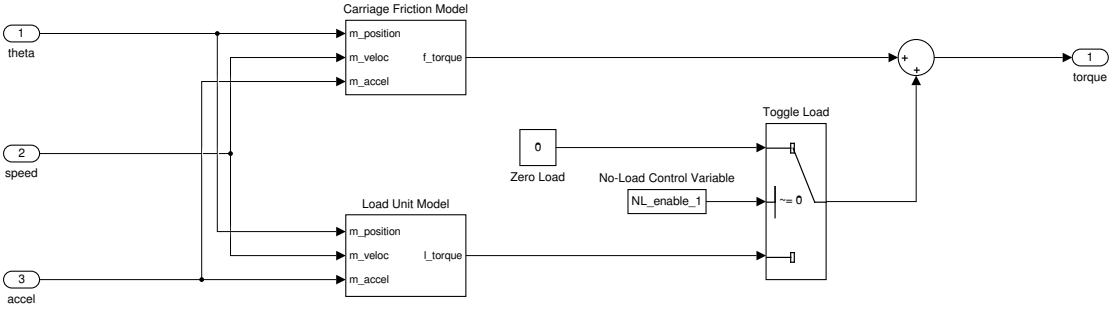


FIGURE 3.6: Configurable load plus friction model in Simulink.

The details of the friction modules for the respective models are omitted for now as these are the topics of the following subsections.

ii. *Electrical Plant*:

This model can be configured to function in either VC or CC mode via the `CC_enable_*` variable. Also, the effect of output voltage saturation has been incorporated since the amplifier is supplied with a fixed voltage. This can show whether or not the controller brings the system to saturation. The equations used in this model are (3.15) and (3.18) for each control mode respectively. Fig. 3.7 shows the Simulink model for this variant.

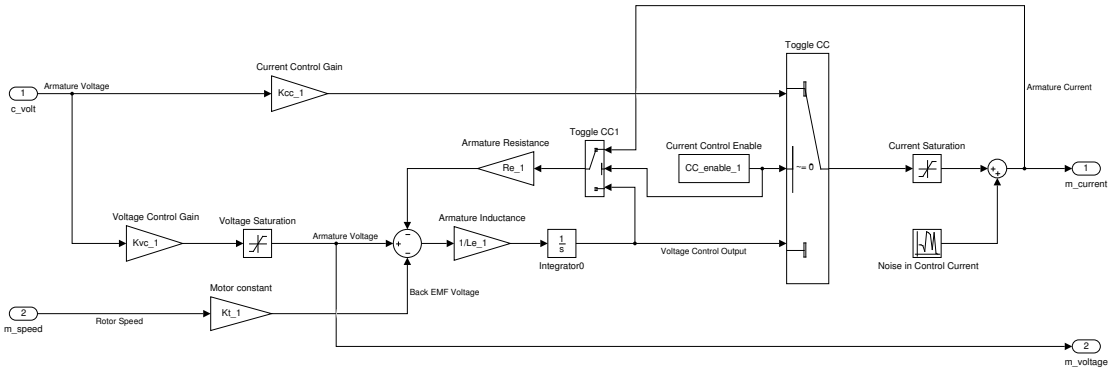


FIGURE 3.7: Electrical plant in Simulink using Eq. (3.18).

In the case of also incorporating the PI controller for CC, the following figure shows the resulting plant:

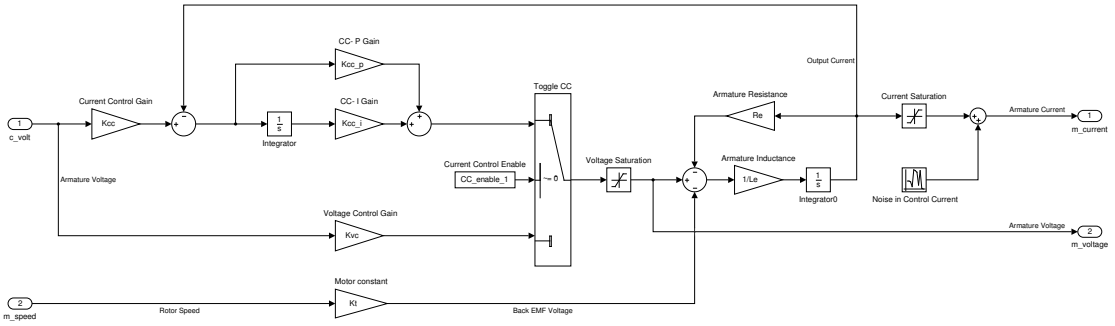


FIGURE 3.8: Electrical plant in Simulink including the PI Current Controller.

In both variants presented above, a noise generator has been included at the output of the plant in order to simulate the residual current ripple produced from the electronics but as well as to emualte the measurement uncertainty.

To summarize this section, a generalized simulation model has been developed based on an in-depth analysis of the factors contributing to the actuator dynamics as they are seen from the motor side. The model created in Simulink is configurable and can accommodate several types of plants for the load and other dissipative effects from the operation of the carriage. The aim is to have a complete simulation module that

can predict the behaviour of the relevant subsystem with high degree of accuracy and therefore serve as a test-bed for controller development prior to testing on the physical platform. Moreover, the developed model is to be re-usable across both carriage axes furthering our intentions to create a system with a high degree of modularity.

3.3.2 Dynamics of Friction

The previous section presented a skeleton of a generally applicable system plant for modelling a single carriage axis. In this section we explore physical models for describing friction that are appropriate to the type of mechanisms used in the MACS. The findings presented here are primarily based on those described in [25] and [28]. The previous sources provide excellent in-depth analyses for describing and modelling friction and the reader is advised to refer to these for further insight.

Starting with [MACS-A-7], we assume that τ_d depends solely on θ_m and $\dot{\theta}_m$. As stated previously, all motion is implemented with rotational elements. From motor shafts to final carriage motion, all momentum is transferred with gears, chains and rollers (wheels) which means that all are supported by ball bearings. These bearings are enclosed and lubricated justifying the produced friction's dependence on velocity since lubricants are known to exhibit viscous behaviour. Additionally, the geometry of the set-up (i.e. the tank and carriage) in combination with the linear bearings for the X axis drive and the rollers in the Y axis drive produce a dependence on position.

The primary issue with friction is that although it can be described to a sufficient degree of accuracy for industrial and R&D applications, it is not completely understood and not easily quantified experimentally. As is demonstrated in [25], the process of trying to explain all the effects exhibited by friction and at the same time produce quantitative results requires a series of complex and highly arduous experiments to be executed. Still, due to aspects concerning the physical materials, such as material ageing, thermal phenomena and transient behaviour at the nano-scale, the measurements are not always reproducible under all conditions of the mechanism's operation.

Models of Friction

In the general case, when two surfaces are in contact, the frictional forces produced at the interface depend on the applied forces F_{ext} and the relative velocity $\dot{x}$. As described in [25], all the phenomena arising from this interaction are separated into four velocity regimes. Thus the most general model of friction takes into account the phenomena which occur at each regime. Although in the previous reference, the phenomena are ordered by regime, we shall describe them in an order which is more descriptive of how increased detail is attained in the model.

We present the phenomena which are considered for the operation of the MACS.

i. ***Coulomb Friction***:

This is the classic model of kinetic friction that coarsely describes a constant force produced at the interface of the two surfaces. It describes the dependence on the material properties and the forces perpendicular to the interface, e.g. weight.

$$F_C = -\operatorname{sgn}(\dot{x})\mu N, \quad \dot{x} \neq 0 \tag{3.21}$$

$\dot{x}$ is the relative velocity between the two surfaces, μ is the coefficient of kinetic (Coulomb) friction and N is the effective normal force exerted at the interface.

ii. ***Stiction***:

This is the static friction force acting before any motion occurs. This extends the description of the Coulomb force by adding an *increased* friction force *prior* to any motion. Just as the two surface begin to move relative to each other, the small asperities (i.e. the small imperfections or bumps on the surface) at the *nm* scale on either sides of the interface are still in contact but the applied force causes small displacements between the contact points. This means that the asperity junctions behave as springs acting in the opposite direction of the applied force. Thus when effective motion begins due to the complete separation of the asperities, the instantaneous friction force lessens in magnitude to the level of Coulomb friction.

$$F_S = -k_S \cdot \delta x \tag{3.22}$$

Where, k_S is the equivalent spring stiffness coefficient and δx is the deviation in position. When this force exceeds a certain threshold, plastic deformations occur in the asperity junctions resulting in an instantaneous *brake-away* between the two surfaces. For this reason, stiction is also referred to as the break-away force; it increases according to the exerted external force until finally all the junctions separate and motion occurs.

There is certainly no inexpensive and direct way to experimentally measure the displacements δx to extrapolate values for k_S. This means that (3.22) does not provide a viable way to model the stiction. A much more direct but approximate approach is to use the following relationship:

$$\implies F_S \approx -F_{\text{ext}}, \quad \text{while} \quad \dot{x} = 0 \tag{3.23}$$

Where, F_{ext} is the total external forces exerted on either surface.

iii. ***Viscous Damping***:

When the interface contains lubricant, post-separation of the surfaces results in the influx of fluid. As relative velocity between the surfaces increases, so does the effect of friction due to the viscous flow of the lubricant. The result is force $F_{\dot{x}}$ that is a function of the relative velocity between the two surfaces. The specific function defining the relationship to velocity can vary depending on the type of lubrication and the material of the surfaces. It is common however to consider the force linearly proportional to the velocity:

$$F_v(\dot{x}) = -b \cdot \dot{x} \tag{3.24}$$

Where b is the damping coefficient.

iv. ***Stribeck Effect***:

If only the previous phenomena where taken into account, then the resulting model would exhibit hard non-linearities, i.e. discontinuities. The Stribeck effect describes a continuous transition from the static friction regime to all other values for increasing or decreasing values of velocity. Specifically it designates decreasing friction for increasing velocity, something which can be interpreted as a regime with negative viscous friction.

The Stribeck effect does not prescribe a specific function for this transition but several heuristic approaches exist throughout the literature. All of these attempt to follow the same requirements of a *continuous* and *smooth* transition from static to the positive viscous regime and identify a minimum velocity above which the latter regime commences. This velocity is named the characteristic velocity of Stribeck friction (see [25]) and plays a significant role in the stability of steady state motion between the surfaces.

Two approaches one can consider for describing Stribeck effect are the so called Gaussian and Lorentz models:

(a) *Gaussian*:

$$f_s(\dot{x}, \dot{x}_s) = e^{-(\frac{\dot{x}}{\dot{x}_s})^2} \tag{3.25}$$

(b) *Lorentz*:

$$f_s(\dot{x}, \dot{x}_s) = \frac{1}{1 + (\frac{\dot{x}}{\dot{x}_s})^2} \tag{3.26}$$

For each function used for $f_s(\cdot)$, we are only interested in the mapping $[0, \dot{x}_s] \mapsto [0, 1]$, where $\dot{x}_s$ is the Stribeck velocity. For this reason, we use the parameter λ to provide the appropriate amplitude for the transition from F_S to F_C:

$$F_s(\dot{x}, \dot{x}_s) = \lambda \cdot f_s(\dot{x}, \dot{x}_s) \tag{3.27}$$

The final choice always depends on the experimental results. However we should compare the characteristics of each model. The Lorentz model is exhibits a greater (faster) drop-off rate for velocities shorter than $\dot{x}_s$ and is almost linear in this regime, while the Gaussian has faster drop-off for values larger than $\dot{x}_s$ and is considerably non-linear at low velocities. This means that the Lorentz model is much closer to the approximate model:

$$F_d = F_C \cdot \text{sgn}(\dot{x}) + B \cdot \dot{x} \tag{3.28}$$

However, the larger drop-off at low velocities also means that it results in a greater instability close to the Stribeck velocity.

v. ***Position Dependent Friction***:

Position dependent friction is a complex matter. One could argue that it can be included directly as part of either static or kinetic friction. However the latter two are considered to describe the forces at the interface between the rolling elements at a small scale, i.e. the ball bearings, while the position dependence we are referring to happens on the scale of the dimensions of the tank. Specifically we seek to map the lateral loading of the rollers and surface loading of the linear bearings for the Y and X axis respectively. Although these are not intrinsic frictional loads, their effects on the surface pressure exerted between the shafts and the bearings does in fact lead to increased friction at these interfaces. We therefore consider these as additional forces $F_{P,Y}(y)$ and $F_{P,X}(x)$ which are empirically mapped to the position of the TC and EEP respectively for each axis.

There are other phenomena which can also be incorporated in the model such as *dwell time* and *frictional lag*, but there is no experimental evidence from tests done on the MACS to support their use. Moreover, these effects are highly sensitive to variations of temperature, humidity and other ambient factors and therefore would only overcomplicate matters considering the conditions under which URETTS operates.

Fig. 3.9 presents a visualization of the friction force when plotted against velocity.

We can now combine the relevant components into an integrated model of friction:

$$F_d(x, \dot{x}) = F_C + F_S + F_v(\dot{x}) + F_s + F_P(x) \tag{3.29}$$

$$\implies F_d(x, \dot{x}) = F_C + F_S + b\dot{x} + \lambda \cdot f_s(\dot{x}, \dot{x}_s) + F_P(x) \tag{3.30}$$

Equation (3.30) presents the general structure which can be used with all variations of the components. In order to produce the final model we can execute a re-factoring of the components and find an appropriate value for λ in order to produce the fitted curve:

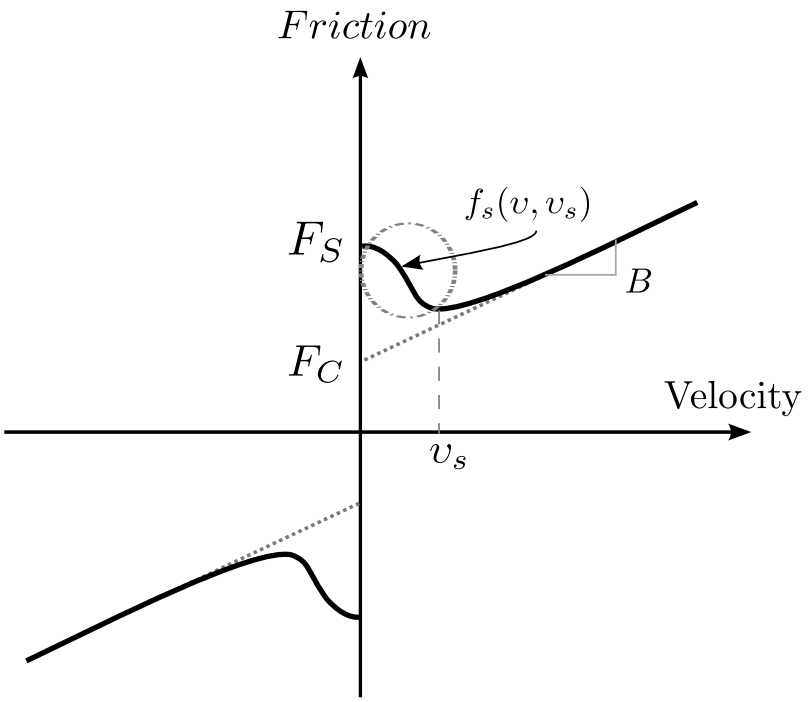


FIGURE 3.9: Stribeck friction curve when mapping torque against velocity.

$$F_d(x, \dot{x}) = \begin{cases} -F_{\text{ext}} & , \quad \dot{x} = 0,\ F_{\text{ext}} < F_S \\ F_S & , \quad \dot{x} = 0,\ \ddot{x} \neq 0 \\ b\dot{x} + F_C + (F_S - F_C) f_s(\dot{x}, \dot{x}_s) + F_P(x) & , \quad \dot{x} \neq 0 \end{cases} \tag{3.31}$$

Based on the findings in [25] and [27], we can expect to see a general trend over the entire range of positions, with smaller features appearing at larger frequencies. From (3.31) we can see that F_P can be inferred from measuring the static plus kinetic friction force along the full length of the tank and when velocity is zero. This means that the collection of data over the entire tank can be used for to measure both static and the position dependent friction. Furthermore, without empirical evidence we cannot decide *a priori* which form to use for the Stribeck curve so we leave this to be determined from the results of the experiments in Chapter 7.

Effects of Stick-Slip

Before we complete our discussion regarding the dynamics of friction we comment on a particular phenomena important to the developing models for its behaviour. This is known as *stick-slip* and is direct consequence of the negative viscous friction exhibited at low velocities in combination with the discontinuity at zero velocity when a change of direction occurs.

Lets imagine a situation where a single axis is set in motion and from a moderate initial velocity and we would like to come to rest at a specific position. If we gradually reduce the velocity as we approach the target there would be a velocity at which the axis would experience a sudden stop well before the goal is reached. Then, if we were to maintain the last value used for the applied force, the system would after some time begin to move again for a specific interval and then the whole process would repeat. This inability to maintain a linear relationship between excitation and steady-state velocity for low values of the latter, is the effect of stick-slip.

In Fig. 3.9 one can notice the characteristic velocity $\dot{x}_s$ at the tail of the negative friction regime. From [25], [27] and [28] it is clear that this velocity is practically impossible to reach experimentally due to the inherent instability at these velocities. However we only need to find an *approximation* of the real velocity $\dot{x}_s$ within some reasonable region of deviation. By estimating the approximation $\tilde{\dot{x}}_s$ we can determine a *lower bound* for the feasible bandwidth (in velocity) of our system. This plays a major role in both designing model based controllers and in selecting *feasible trajectories* for each axis. The reader is referred to [25] (Chapter 7) for an extensive analysis of this phenomena.

3.3.3 Towing Carriage Dynamics

With the assumption of rigidity (MACS-A-4), there is no relative motion between parts of the same MLCs. Rigidity is also relevant for the interactions between the chain, sprockets and primary links as this would imply that the chain is always fully extended and moves simultaneously with all the connected links. If this is in fact the case, then all components of an axis drive move synchronously, therefore contributing equally to the effective motion of the respective axis.

We shall begin by defining the following Lagrangian, neglecting of course any loading due to gravity as related to [MACS-A-9] and also consider the motion as confined to the horizontal plane (i.e. $z_i = 0, \quad \forall i$):

$$\mathcal{L}_{\text{MACS}} = \frac{1}{2}(m_X \dot{x}^2 + m_Y \dot{y}^2) = \frac{1}{2}((J_X \cdot {\theta_{m,X}}^2) + (J_Y \cdot {\theta_{m,Y}}^2)) \tag{3.32}$$

Where m_X and m_Y are the total masses of each axis, while J_X and J_Y are the total moments inertia as seen from the motor side of each carriage axis. $\theta_{m,X}$ and $\theta_{m,Y}$ are the rotations of each axis' motor shaft. Using MACS-A-5 and MACS-A-6 we can neglect any kinetic energy contributions between axes. This means that we could consider separate Lagrangians for each axis and therefore justify the modelling of each *independently*.

Furthermore, since both axes are implemented with the same two-sprocket plus chain configuration, then we can go as far as to consider using the same model (structurally) for both. The prospect of creating a modular system plant, with only a difference in model parameters can be an attractive quality as this would greatly simplify the entire modeling process. Experimental evidence is presented in Chapter 7 to support this hypothesis.

Subsequently, in order to encapsulate the total behaviour of each axis' dynamics, we present in Fig. 3.10 the overall plant implemented in Simulink. Each sub-plant shown in this figure has been describe previously and so this is the integration of these into a single model.

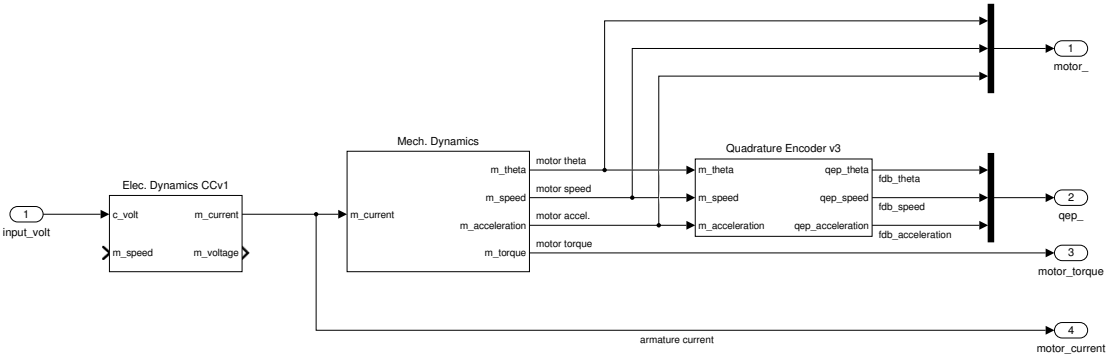


FIGURE 3.10: Overview of the system plant for a single URETTS axis of motion.

The blocks in Figure 3.10 serve as place holders for the individual dynamics for the electrical, mechanical, sensors and actuators. For each we integrate the plants described previously. Specifically the Mechanical Dynamics block integrates the dynamics of the

an individual carriage axis, the assumed friction model and a model for the LU. The Electrical Dynamics block contains either of the systems shown in Fig. 3.7 and Fig. 3.8.

To summarize, lets now present the total system dynamics for each axis. The equations describing the mechanical dynamics result from applying a Newton-Euler formulation by equating the sum of the external forces to the effective inertial force. Since control is implemented on the motor armature circuit all torques are described on the motor side. The resulting system of equations is:

I. Mechanical:

$$J\ddot{\theta} + \tau_d(\theta, \dot{\theta}) = \tau_m - \tau_{LU}(\dot{\theta}) \tag{3.33}$$

II. Dissipation:

$$\tau_d(\theta, \dot{\theta}) = B\dot{\theta} + (\tau_C + (\tau_S - \tau_C) f_s(\dot{\theta}, \dot{\theta}_s) + \tau_P(\theta)) \cdot \operatorname{sgn}(\dot{\theta}) \tag{3.34}$$

III. Actuator:

$$\tau_m = K_T K_{\text{amp}} \nu_C \tag{3.35}$$

3.3.4 Vibration from Elasticity

When initial experiments were executed with the purpose of identifying the physical parameters, noticeable oscillations were occurring in the responses. This lead to the realization that the open-loop system could not be assumed to be of first order. In order for oscillations to occur, an ODE (or system of ODEs) must be *at least of second order*. Various causes were explored in an attempt to explain this behaviour. Some of these included the contribution of the motor's intrinsic inductance, elastic deformation of the attachment points of the chains on the TC and EEP respectively, the marginal stability of the PI controller for CC in the motor amplifiers and even the wobble of the drive sprockets due to motor coupler misalignments. Testing for each of these failed to provide any evidence for further consideration.

In the end, the only possible cause left was to consider the oscillations of the chains during motion. Indeed visual inspection of the chains during motion clearly showed that they oscillated considerably when either axis was set in motion. Here in lies the *smoking*

gun. Oscillations occurred *only* when large external forces were applied. This implies that when high mechanical *jerk* (i.e. the derivative of *acceleration*) was occurring due to non-smooth actuator forces, modes of vibration were being exited. This effect is well studied in the robotics and mechatronics literature and is known equivalently as *joint elasticity*.

Clearly, the model in (3.33) would not suffice to predict this behaviour. This would require to increase the order of the system with the inclusion of spring-damper pairs. The question then arises as to where exactly should these be placed in the kinematic chain in order to define the relative motions. Since the increase in order means that an additional DoF is needed per carriage axis, the question is which variable should be used. In fact we already have the answer. Looking back to Fig. 3.3 one can see the closures in the graph occurring between sprockets, chain and primary link subsets of S_X and S_Y. Whilst the chain oscillates, deviations occur between the position of the primary link and the angular position of the motor actuators.

The question remains however regarding the placement of the equivalent spring-damper pairs. The possible positions for these are between each link, either intermediate or primary. There are three possible positions for placing a single pair, another three for placing two pairs and then there is the possibility to place all three at every available position. Figure 3.11 visualizes the equivalent systems produced by these placements via free-body diagrams of the axis chain drive system.

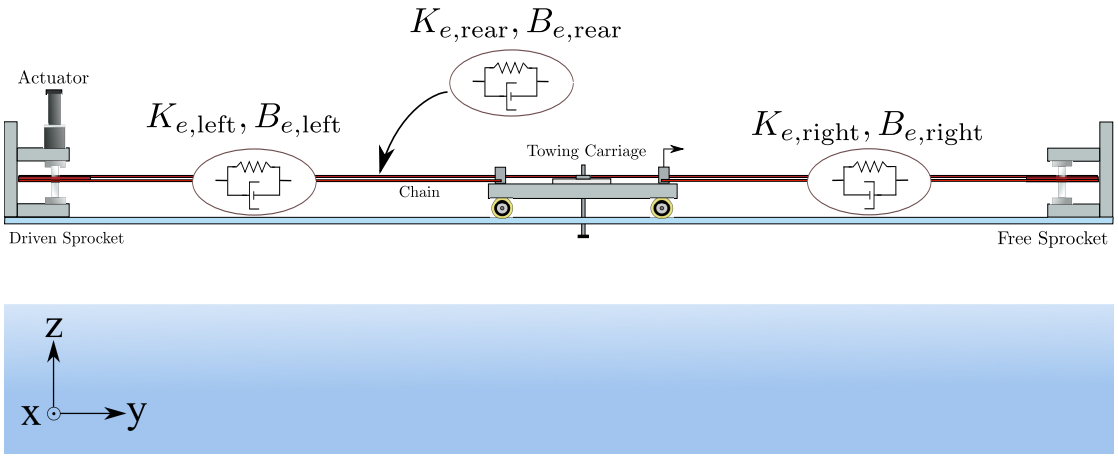


FIGURE 3.11: The possible configurations for including a spring-damper pair K_s, B_s in a MACS axis.

We have identified however that the large *jerk* is due to large starting torque and results in oscillations between the position of the primary link and the motor angle. Thus we could reasonably assume that in the case of a single spring-damper pair, the most appropriate position to place it would be between the primary link and the sprocket which is *pulling* at that particular moment. This means that if we would suddenly command a change in direction, then the effective spring-damper pair would be placed between the other sprocket and the primary link. Since however the sprockets in an axis drive are identical then so are their total inertias, thus the final equations of motion are invariant to a change in direction of the external force/torque.

Let x_i be a the Cartesian coordinate of the primary link and θ_i the angle of rotation of the respective motor's shaft. Starting from the following Lagrangian we seek the equations of motion for the 2-DoF variation of the model with elasticity:

$$\mathcal{L}_{2,i}(\theta_i, x_i, \dot{\theta}_i, \dot{x}_i) = \frac{1}{2}(J_{m,i} + \frac{J_{sd,i} + J_{sf,i}}{{n_i}^2})\theta_i{}^2 + \frac{1}{2}m_{L,i}{x_i}^2 - \frac{1}{2}(x_i - \frac{R_i}{n_i}\theta_i)^2 K_{s,i} \quad (3.36)$$

Where $K_{s,i}$ is the stiffness coefficient of each equivalent spring. Applying the *Euler-Lagrange* equation of (3.6) and applying the external force/torques to the generalized forces, we arrive at the following system:

$$J_i\ddot{\theta}_i + B_i\dot{\theta}_i + \tau_{d,i}(\theta_i, \dot{\theta}_i) - K_{s,i}\frac{n_i}{R_i}(x_i - \frac{R_i}{n_i}\theta_i) - B_{s,i}\frac{n_i}{R_i}(\dot{x}_i - \frac{R}{n}\dot{\theta}_i) = \tau_{m,i} \quad (3.37)$$

$$M_i\ddot{x}_i + B_i\dot{x}_i + F_{d,i}(x_i, \dot{x}_i) + K_{s,i}(x_i - \frac{R_i}{n_i}\theta_i) + B_{s,i}(\dot{x}_i - \frac{R}{n}\dot{\theta}_i) = 0 \quad (3.38)$$

$B_{s,i}$ is now the damping coefficient of each equivalent spring. This system is shown in below in Fig. 3.12.

The last form in (3.37) and (3.38) is known in the literature as a *singularly perturbed system*. These types of systems exhibit behaviours on two different time scales, not unlike the dynamics of a PMDC motor. In fact this is exactly how PMDC motors

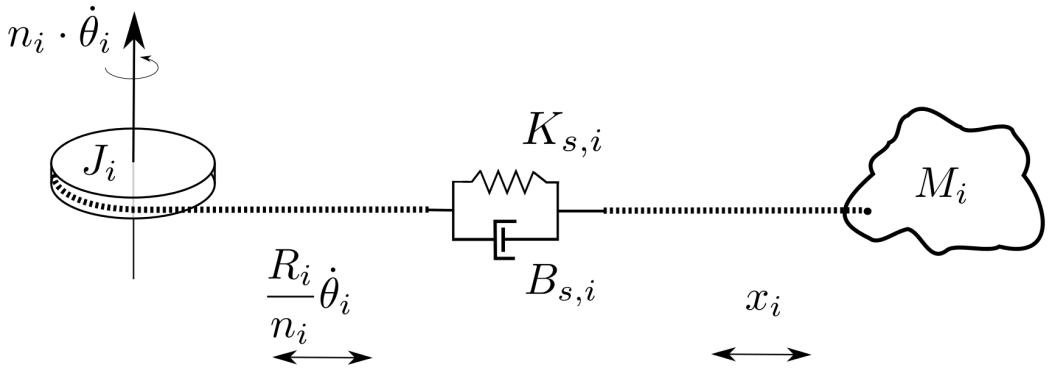


FIGURE 3.12: Free body diagram of equivalent system including a spring-damper pair.

behave. Khalil et atl. have an excellent examples on this; see [61], Chapter 11 (pg. 425-430).

The behavior exhibited on the larger time scale is that of the *nominal* system as the perturbation tends to zero. When the perturbation is considerable however, the system exhibits additional behavior on a shorter time scale. In the example of a PMDC motor, if the inductance where large enough, a step response in VC mode would result in overshoot with decaying oscillations in velocity. In the case of the carriage axis drive configuration, the oscillations occur in acceleration which then propagate throughout the other responses like position and velocity albeit occurring to a less degree due to the averaging effect of integration.

Moreover, in the case of the system in (3.37) and (3.38), the perturbation dissipates due to the dampening and so as it approaches zero, the system collapses into the *nominal* system of (3.33) of the slow time-scale. When this occurs the components of the 2-DoF system reduce to the following *effective* parameters (as seen from the motor side):

$$J_{\text{eff},i} = J_{m,i} + \frac{(J_{sd,i} + J_{sf,i})}{{n_i}^2} + \frac{{R_i}^2}{{n_i}^2} M_i \tag{3.39}$$

$$B_{\text{eff},i} = \frac{{R_i}^2}{{n_i}^2} B_{m,i} + B_{M,i} \tag{3.40}$$

$$\tau_{\text{eff},D,i}(\theta_i, \dot{\theta}_i) = \frac{R_i}{n_i} \tau_{C,m,i} + \tau_{C,M,i} \tag{3.41}$$

The system in (3.37) and (3.38) has been implemented in Simulink as the model shown below in Figure 3.13:

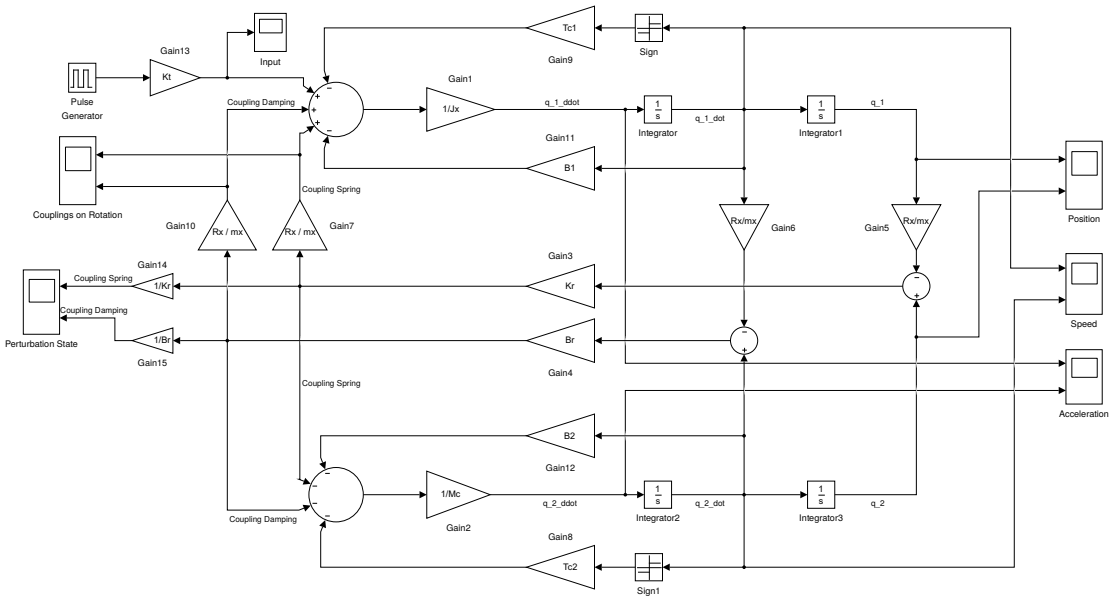


FIGURE 3.13: 2-DoF model with joint elasticity implemented in Simulink

For the purpose of simplification, we can assume that the 2-DoF model of (3.37) and (3.38) is sufficient to describe the observed oscillations in the actual system. As is shown in Chapter 4, the simplification of the model plays a major role in the identification of physical parameters for the total system.

Lastly, this advanced model of the carriage axis dynamics is implemented as the Simulink model shown in Figure 3.10. This is done for the purpose of predicting the behavior of the overall system and to support the design of advanced controllers in future research. Moreover, there is also the possibility to use two or more spring-damper pairs in the model, but this is outside the scope of this current work and is also left for future research to explore in detail.

3.4 Velocity Estimation

The final section of this chapter addresses the behavior of the incremental encoders and how velocity estimates can be attained by using appropriate methods. Motor encoders are sensors which provide measurements of the rotation of a motor's shaft. There are many varieties which are common to robotic and mechatronics applications and these

are chosen based on the functional requirements and cost of the total system. Examples of the technologies used for these sensors include optical encoders, electronic encoders which can resistive or capacitative and grey type encoders. Regardless of the type, in the end the data is provided as digital information encoding the relative motion in integer increments.

Two distinct categories exist which differ in the type of measurement they produce. These are the absolute and incremental encoders. The former retain at all times a common reference from which the measurement is made, while the latter output a measurement with no fixed reference position and which can increase until the digital value is forced to reset due to data overflow. Also the measurement can differ depending on where the encoder is mounted, i.e. directly on the motor, after a transmission and even directly on the load.

3.4.1 Incremental Encoder Operation

In MACS, the actuators are fitted with Magnetic Rotary (MR) incremental encoders mounted on the motor side shaft. The signals output to and received by the interfacing electronics are quadrature differential signals. Details and specifications for these are provided in Chapter 5. For the purposes of this chapter, it suffices to know that the encoders operate with a fixed number of total increments N_{enc} per rotation of the motor shaft. Thus for e.g. $m \times 2\pi$ rotations of the motor, the sensor yields measurement of $m \times N_{enc}$ *integer* counts.

Mounted after the motor is a planetary gear transmission with a reduction ration of n_g and also a sprocket with radius R_s, thus the *total resolution in position* of the respective primary link is:

$$N_L = \frac{N_{enc} n_g}{2\pi R_s}, \quad (\text{counts}/m) \tag{3.42}$$

Where N_L is the counts corresponding to the displacement of the load. This value can become quite large if the reduction is high and the sprocket is small, thus providing possible resolutions in position in the tens of μm scale. We therefore define the quantization

errors in position of the motor and load respectively as the displacements corresponding to a single encoder count (reciprocal of the resolution):

$$\Delta\theta_Q := \frac{2\pi}{N_{\text{enc}}}, \quad (^{\text{rad}}/_{\text{count}}) \tag{3.43}$$

$$\Delta x_Q := \frac{2\pi}{N_{\text{enc}}} \cdot \frac{R_s}{n_g}, \quad (^{m}/_{\text{count}}) \tag{3.44}$$

The previous values define the absolute smallest displacements for which we can discern that motion has in fact occurred. These encoders provide reliable measurements of position and experiments have shown them to output reproducible values with minimal introduction of noise.

3.4.2 Velocity Estimation Algorithms

The challenge now is to accurately and reliably estimate the velocity from incremental encoder measurements. Although it would seem that applying the Backward Difference Method (BDM) (see [36]) would suffice to estimate velocity in general, this does not provide good results in all cases. Since the smallest non-zero displacement measured in counts is 1, this produces significant amount of measurement uncertainty at low velocities due to the inherent quantization error.

As described in [33], the two primary types of velocity estimation methods for incremental encoders are the frequency and period based estimation methods. Both require not only a different approach in the estimation technique, but also different hardware in the system. In any case, the basic idea is to create an approximation of the true velocity:

$$\omega := \dot{\theta} = \frac{d\theta}{dt} \leftarrow \left.\frac{\Delta\theta}{\Delta t}\right|_{\Delta\theta,\Delta t\to 0} \tag{3.45}$$

Frequency Based Estimation

This is the most common type of estimation method and probably the most recognizable. This method assumes a fixed sampling period T_s and uses the difference in encoder increments (pulses) between sampling instances. The simplest form for this type uses an BDM for the calculation:

$$\tilde{\omega}_{m,f} = \frac{2\pi}{N_{enc}} \cdot \frac{n_k - n_{k-1}}{T_s} \tag{3.46}$$

Where $\tilde{\omega}_m$ ($^{\text{rad}}/_s$) is the estimate of the motor's angular velocity and n_k and n_{k-1} (counts) are the new and previous encoder measurement respectively. Since unit displacement is a single count, the quantization error in rotations is:

$$\Delta\omega_{Q,f} = \frac{2\pi}{N_{\text{enc}}} \cdot \frac{1}{T_s} \quad (^{rad}/_s) \tag{3.47}$$

The quantization error in encoder counts is:

$$\Delta n_{Q,f} = \frac{1}{T_s} \quad (^{counts}/_s) \tag{3.48}$$

It is obvious that this quantization error is sensitive to variations in the sampling period since these are inversely proportional. If one assumes that the sampling period T_s and control period T_c coincide (i.e. we sample the same as we control), then the faster we make the loop to increase the performance of the controller, the noisier the velocity estimate becomes. We can quantify this effect if we calculate the Signal to Noise Ration, considering the error's standard deviation equal to $\Delta\omega_{Q,f}$.

Define the SNR as:

$$SNR_\omega := \frac{\mathcal{E}(X)}{\sqrt{\mathcal{V}(X)}} = \frac{\mu_X}{\sigma_X} \tag{3.49}$$

Where $\mathcal{E}(X)$ and $\mathcal{V}(X)$ are the expectation and variance operators respectively acting on any random variable X. Now we define $w_{\omega,f}$ as additive Gaussian white noise in

the measurement of the signal ω and these then define the stochastic process Ω_f whose sample space generates the estimation samples $\tilde{\omega}_{m,f}$:

$$\Omega_f := \omega + w_{\omega,f} \tag{3.50}$$

Since $w_{\omega,f}$ has zero mean then the true signal should be indistinguishable from the mean (for small averaging windows). This results in the following SNR for the measurement of ω

$$\text{SNR}_\omega = \frac{\mathcal{E}(\Omega)}{\sqrt{\mathcal{V}(\Omega)}} = \frac{\mu_\omega}{\sigma_\omega} = \frac{\omega}{\Delta\omega_{Q,f}} \tag{3.51}$$

$$\implies \text{SNR}_\omega = \frac{\omega N_{\text{enc}} T_s}{2\pi} = \frac{\omega N_{\text{enc}}}{2\pi f_s} \tag{3.52}$$

From (3.49) we see that the SNR increases linearly with increasing velocity and is also proportional to the sampling period and encoder resolution. This means that increasing the resolution of the encoder, *reduces* the quantization noise, while *increasing* the sampling frequency, increases it. In an engineering context, it would seem reasonable to determine a sufficient lower bound for the velocity estimation and not bother with velocities below this. However as is the case with the MACS, since low velocity regimes are particularly of interest (consider the discussion on friction in Section 3.3.2) we require solutions to estimating velocities well within the ($^{mm}/_{s}$) range. Figure 3.14 shows an example of measured velocity using this method and the resulting quantization noise.

Period Based Estimation

The electronics in the BeagleBone contain a peripheral module that can measure and time digital pulse signals. Although this module is described in Chapter 6, we note here that the same peripheral devices can be configured to read the incoming quadrature pulses from the incremental encoder and produce different types of velocity estimates.

When using *period based* velocity estimation, instead of a fixed period, we now consider a fixed *unit count* N_s and measure the amount of time that passes between

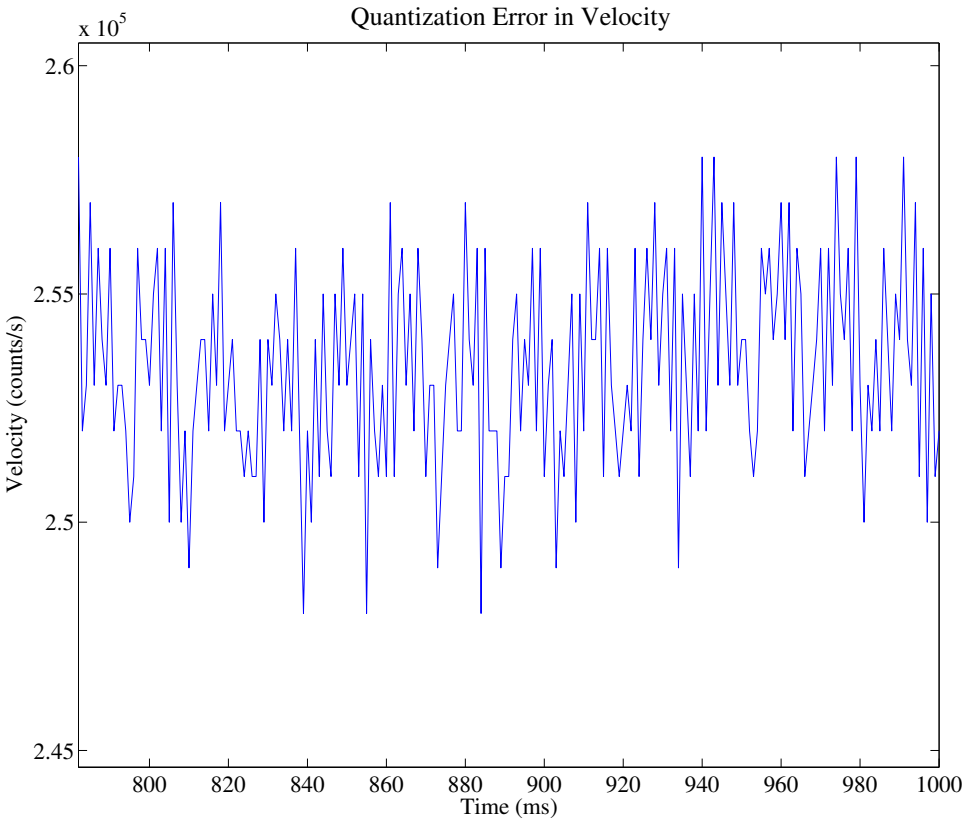


FIGURE 3.14: A frequency based velocity measurement exhibiting significant quantization error.

unit count events. In the general case, the estimate is based on the following calculation using once again an BDM approach:

$$\tilde{\omega}_{m,p} = \frac{2\pi}{N_{\text{enc}}} \cdot \frac{N_s}{t_k - t_{k-1}} \tag{3.53}$$

This method now exhibits a linear relationship to the sampling displacement N_s. The digital pulse timer uses an internal clock with period T_f to synchronize the sampling of time units, as well as configuration parameter m_f to scale the resolution of the timer. Equ. (3.53) now becomes:

$$\tilde{\omega}_{m,p} = \frac{2\pi}{N_{\text{enc}}} \cdot \frac{N_s}{T_f m_f} \cdot \frac{1}{m_k} \tag{3.54}$$

Where now m_k is defined as the number of timer *ticks* between unit events. The quantization error of period based estimation cannot be defined in the same way as the

frequency based method. We must first identify lower and upper bounds for m_k. Since the timer uses an 16-bit integer to hold the values of m_k, then it would always have to be in the range $m_k \in [1, (2^{16} - 1)]$. This then means that largest measurable value of velocity would be that when we the peripheral continuously sees a unit event:

$$\tilde{\omega}_{m,p}|_{max} = \frac{2\pi}{N_{\text{enc}}} \cdot \frac{N_s}{T_f m_f} \tag{3.55}$$

As the motor's velocity reduces, the timer would now be able to produce larger values for m_k. In fact when the largest value occurs, we get:

$$\tilde{\omega}_{m,p}|_{min} = \frac{2\pi}{N_{\text{enc}}} \cdot \frac{N_s}{T_f m_f} \cdot \frac{1}{(2^b - 1)} \tag{3.56}$$

Where b is the number of bits used to hold the integer values of the timer ticks m_k. Below this velocity the counter overflows and the motion is now undetectable. Considering that if m_f is configured appropriately, then the total time ΔT could be in the order of seconds. If single digit encoder counts are not occurring in a few seconds then we can almost be certain that no there is motion.

We can assume that *jitter* exists in the occurring values m_k, and this can partly count for the resulting noise. The approximation described in [33] was tested in on our system but was unsuccessful. It is difficult to model the stochasticity of the jitter and is beyond the scope of this work but since the velocity measurement is inversely proportional to the timer ticks, we can expect the noise in the measurement to be approximately proportional to the velocity. For these reasons we empirically measure this proportionality:

$$\text{SNR}_{\omega,p} \approx \frac{\mu_{\omega,p}}{\sigma_{\omega,p}} := \gamma_p \tag{3.57}$$

Where $\mu_{\omega,p}$ and $\sigma_{\omega,p}$ are the *statistical* mean and standard deviation respectively and γ_p is the approximated value of the SNR.

The primary challenges in using period-based velocity estimation are:

i. Sampling cannot be truly uniform as the estimations are available only when the unit distance has been traversed, therefore no new velocity data is available at regular sampling intervals.

ii. If the encoder uses a N-bit timer, then the total waiting time cannot exceed $2^N - 1$ timer counts.

iii. The proportion of the longest timer count duration relative to the regular sampling period.

In order to handle these caveats, the values used for the velocity is held constant until the estimator produces new values. Also, the maximum waiting time is set as a certain multiple of the regular sampling period. If lets say m multiples of the sampling period T_s are set for the maximum waiting time for new estimates, then $m \cdot T_s$ is the largest duration between new values and if no new values have occurred within this time, the estimator produces zero velocity.

Hybrid Estimation Methods

A combined approach could use both algorithms to perform each calculation when most appropriate. Specifically, we take advantage of the properties of each algorithm by using the period based calculation at low velocities and the frequency based velocities at large velocities. This then requires that a threshold velocity be specified to define the bounds of the two velocity regimes. It is reasonable to select this velocity as to optimally reduce the effects of the noise. For this reason we choose to define the *switching velocity* $\nu_{sw,i}$ for each axis i. Therefore we determine the appropriate value for this velocity when:

$$\mathrm{SNR}_{f,i} = \mathrm{SNR}_{p,i} \implies \frac{N_{\mathrm{enc},i} T_{s,i}}{2\pi} = \gamma_{p,i}, \quad \forall i. \tag{3.58}$$

$$\omega_{\mathrm{sw},i} = \omega|_{\mathrm{SNR}_{f,i}=\mathrm{SNR}_{p,i}} \tag{3.59}$$

Chapter 4

System Identification & Controller Design

The towing experiments assume that the trajectory of the TO is known while in motion. This assumption is critical for the experiments to be successful, as large deviations from the reference trajectory degrade the efficacy of the measurements. This means that in order for a specified path of the TO-plus-EEP to followed accurately, a *controller* must appropriately regulate the motion while also rejecting possible disturbances.

The goal therefore is to execute trajectory tracking in a Cartesian task-space with minimal error. For this to become attainable, a feedback controller must be designed and implemented which can regulate motion with the following performance requirements:

1. TO velocities in the range of $\dot{v}_{TO} \in [1, 50]^{\text{cm}}/_{s}$ must be maintainable at steady-state.
2. The *tracking error* in velocity must not exceed in peak-to-peak value, that of the measurement noise of the force sensor attached to the LU.
3. The control rate frequency must be at a rate of at least 100Hz.

The first requirement pertains to the hydrodynamic experiments, which are especially challenging in the lower velocity regimes. The second concerns the error/deviation assumption directly. Lastly, preliminary testing has shown that the system's shortest mechanical time constant is in the order of $100ms$ (i.e. $10Hz$). Therefore, for accurate

control and sampling to be practical, a control rate of at least an order of magnitude faster than the system's behavior is necessary.

In this chapter we address these requirements with the design of a simple yet effective controller that can produce the desired behavior and describe test in order to quantify its performance. Firstly using the methodologies described in [20] and [27], a set of experiments are designed with the purpose of determining the set of physical parameters which complement the models developed in Chapter 3. Then we describe the design of a linear, model-agnostic PID controller to demonstrate the capabilities of URETTS. This chapter concludes with an outline of the trajectories used to benchmark the performance of the system.

4.1 Parameter Identification

This section presents the methodologies used for the identification of the system's physical parameters. The methodologies are those described in [20] and used in [25] and [27]. The former is considered a seminal work that was one of the first to formally address the use of System Identification (SI) methods, especially in the context of control. Many methods exist within SI, but not all are suitable to address the peculiarities of every system. The applicability of each method is determined based on relative assumptions and sometimes is even *ad hoc*.

Before we describe the relevant SI methods, lets consider the structure of the dynamical model we are using to describe the system. Our goal in this section concerns only the MACS itself without the contribution of the load. Thus, we consider that the methods and experiments described, are only regarding the identification of the model parameters of the MACS. The equations are those outlined in Section 3.3.3, but are repeated here for the sake of clarity. The dynamical model assumed for any MACS axis i, is:

i. Mechanical:

$$J_i\ddot{\theta}_i + \tau_{d,i}(\theta_i, \dot{\theta}_i) = \tau_{m,i} \tag{4.1}$$

ii. Dissipation:

$$\tau_{d,i}(\theta_i, \dot{\theta}_i) = B_i\dot{\theta}_i + \tau_{C,i} \cdot \text{sgn}(\dot{\theta}) + (\tau_{S,i} - \tau_{C,i}) f_{s,i}(\dot{\theta}_i, \dot{\theta}_{s,i}) \cdot \text{sgn}(\dot{\theta}) + \tau_{P,i}(\theta_i) \quad (4.2)$$

iii. Actuator:

$$\tau_{m,i} = K_{T,i} K_{\text{amp},i} \nu_{C,i} \quad (4.3)$$

J_i $(Kg \cdot m^2)$ is the axis' total (effective) moment inertia as seen on the motor side, B_i $(^{Nms}/_{\text{rad}})$ is the total viscous damping coefficient, $\tau_{S,i}$ and $\tau_{C,i}$ (Nm) are the static and Coulomb frictional torques respectively, $\theta_{s,i}$ is the Stribeck characteristic velocity, $K_{T,i}$ is the motor's torque constant, $K_{\text{amp},i}$ is the voltage-to-current gain for the CC and $\nu_{C,i}$ is the control voltage at the driver amplifier's input. The set of parameters that are to be identified for each axis are therefore:

$$\Pi_i := \{J_i, B_i, T_{S,i}, T_{C,i}, \theta_{s,i}, K_{T,i}, K_{\text{amp},i}\} \quad (4.4)$$

Not all of these however can be determined using a single experiment. For this reason we can identify parameter subsets using specialized experiments and estimation techniques. For instance, $K_{T,i}$ and $K_{\text{amp},i}$ are found via independent calibration using experimental data from the oscilloscope and an upper bound for $\theta_{s,i}$ can be found using stiff PI velocity control at low velocities. On the other hand, the following parameter sub-set $\Pi_{SI,i}$ is found by applying regression methods which can produce parameter values by *fitting* experimental data to assumed equation structures.

$$\Pi_{SI,i} := \{J_i, B_i, T_{C,i}\} \subseteq \Pi_i \quad (4.5)$$

Finally, since $\tau_{P,i}(\theta)$ is a function and not a parameter, its values are inferred from other indirect measurements of the static friction forces that are mapped across the breadth of each axis.

4.1.1 Linear Least Squares

It is evident from (4.1) and (4.2) that – excluding the exponential component – the equations are linear in the parameters. This fact enables us to consider linear regression frameworks such as Linear Least Squares (LLS) to identify the parameters. These methods can produce algebraic estimates of the parameters based on accumulated data of the responses to known inputs.

LLS methods consider that a model's equation defines the relationship between the measured values. Each sample datum is used as a component in a vector spanning the entire duration of an experiment. Effectively, the following *overdetermined* linear system needs to be solved:

$$\underbrace{\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_N \end{bmatrix}}_{\text{output}} = \underbrace{\begin{bmatrix} x_{1,0} & x_{2,0} & x_{3,0} \\ x_{1,1} & x_{2,1} & x_{3,1} \\ \vdots & \vdots & \vdots \\ x_{1,N} & x_{2,N} & x_{3,N} \end{bmatrix}}_{\text{inputs}} \cdot \underbrace{\begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_N \end{bmatrix}}_{\text{parameters}} \tag{4.6}$$

Where N is the total length of each array in the data set. We should note however that depending on the approach taken for using LLS, the arrays represent either *time-series* data or *frequency response* data. The previous equation describes what is used in practice, however, the LLS considers that the true system is subject to uncertainty either from noise or from unmodeled dynamics, thus causing the regression to deviate from the theoretical values by an amount ϵ_k for each sample instance k. The array ϵ of ϵ_k values is termed the *estimation error*. This fact is expressed in the following equation:

$$\mathbf{y} = \mathbf{\Phi} \cdot \theta + \epsilon \tag{4.7}$$

Equation (4.7), seen in the context of SI describes the relationship between $\mathbf{y}$ the output vector, $\mathbf{\Phi}$ the *regressor* matrix, θ the parameter vector and the estimation error $\mathbf{e}$. The first two quantities when used to express the system model, are respectively the excitation signal and the response signals. Thus we finally have the *regression equation*:

$$\tau_{m,i} = \begin{bmatrix} \ddot{\theta}_{i,0} & \dot{\theta}_{i,0} & \text{sgn}(\dot{\theta}_{i,0}) \\ \ddot{\theta}_{i,1} & \dot{\theta}_{i,1} & \text{sgn}(\dot{\theta}_{i,1}) \\ \vdots & \vdots & \vdots \\ \ddot{\theta}_{i,N} & \dot{\theta}_{i,N} & \text{sgn}(\dot{\theta}_{i,N}) \end{bmatrix} \cdot \pi_i + \epsilon \tag{4.8}$$

Where for each sampling instance k, $\tau_{m,i}$ is the array of motor actuation torques, the vector datums $[\theta_{i,k} \quad \dot{\theta}_{i,k} \quad \ddot{\theta}_{i,k}]$ comprise the regressor matrix Φ_θ and contains motor angular position, velocity and acceleration, and π_i is the parameter array generated from the elements of $\Pi_{SI,i}$.

For the parameter estimation, the Ordinary Least Squares (OLS) algorithm is used. Different variations of LLS exist within the literature (see [20]) but OLS constitutes the most basic of these. The algorithm requires a data-set in either the time domain using time-series data of system responses to known excitations or the frequency domain using sinusoidal responses to find the zeros and poles of the transfer function. Our application uses the former approach using system responses to selected input signals. OLS parameter estimation is implemented by the following equation:

$$\tilde{\pi}_i = (\Phi_\theta{}^T \Phi_\theta)^{-1} \Phi_\theta{}^T \cdot \tau_{m,i} \tag{4.9}$$

This estimation produces a residual signal which approximates the estimation error and whose values can be processed in order to determine the success of the estimation. This estimation error is found using:

$$\tilde{\epsilon}_i = \tau_{m,i} - \Phi_\theta \cdot \tilde{\pi}_i \tag{4.10}$$

In order for the estimation to converge successfully, producing meaningful parameters, certain factors regarding the execution of the experiments must be taken into account. Several of these are outlined in detail in [25], while others are heuristic, found ad hoc through tests and simulation. Specifically, successful OLS estimation is attainable with the satisfaction of the following criteria:

OLS-C-1. The input signals must be capable of exciting as many behaviours of the system

OLS-C-2. The data sets must contain a minimum of approximately $N = 7,000$ samples.

OLS-C-3. The relative noise and/or uncertainty in the measurements must not exceed (approximately) 5% of the mean.

OLS-C-4. Proper pre-filtering must be able to reduce the uncertainty in the data.

OLS-C-5. The system must be excited by input signals using the maximum possible values for the input.

OLS-C-6. The system must not enter into saturation in either input or responses as it then becomes non-linear.

4.1.2 Design of Identification Experiments

Since the system's response is affected by the various effects of friction, we must consider that it behaves differently at high and low velocities. For this reason we have to regulate the motion at low velocities using the technique described in [25]. Thus we can recognize the need for using both open-loop and closed-loop experiments for the parameter identification. Also, calibration tests must also be executed in order to measure some of the parameters regarding the actuation hardware and operation of each axis. These are described below.

Preliminary Experiments

Before any of the identification calculations can be carried out, certain configuration parameters must be measured first. This ensures that hardware behaves as assumed with respect to the underlying voltages, currents and other relative signals. These signals are the ones which are truly measured and applied in the system, and are subsequently interpreted as torques, velocities etc. These preliminary tests are for every i:

EXP-P-1. Measuring of $K_{T,i}$ and $K_{\text{amp},i}$.

EXP-P-2. Coarse estimation of each axis' mechanical time constant $\hat{\tau}_{M,i}$.

EXP-P-3. Determination of the saturation velocity $\upsilon_{sat,i}$.

EXP-P-4. Determination of the smallest input voltage $\nu_{C,i}$ that can accelerate each axis to saturation velocity in durations longer than $\hat{\tau}_{M,i}$, i.e: $t_{\text{sat},i} \leq \hat{\tau}_{M,i}$.

EXP-P-5. Determination of safe ranges $[\theta_{min,i}, \theta_{max,i}]$ for the motion of each axis.

Open Loop Experiments

EXP-OL-1. ***Break-Away***

In reality the Stiction force cannot be measured because by definition it is the force which initiates motion but is not the force measured when in fact motion has happened. This quirk of friction does not permit us to measure Stiction directly, however we can approximate it by measuring the force/torque applied and taking the last value sampled exactly before motion (i.e. non zero velocity) was initiated. Therefore what is actually measured is the so-called *Break-away torque*.

Our inspiration for the design of the relevant experiment is derived from [25] and requires that the system (per axis) initially be brought to a complete stop. Then the control torque is periodically increased in *fixed* increments $\Delta\tau_i$ every $\Delta t_{\tau,i}$. We then set a velocity threshold $\omega_{\text{TH},i}$, above which we can assume that in fact motion has occurred and measure the last actuation torque prior to exceeding this threshold.

As mentioned in [25], the selection of $\Delta\tau_i$ and $\Delta t_{\tau,i}$ must be done carefully as to not be affected by other phenomena such as dwell-time and actuator transient currents.

EXP-OL-2. ***Square Wave Pulse Responses***

The second type of open loop experiment considers Square Wave Pulses (SWP) to replace the classic *step* excitation. The reason for repeating the step input, and thus producing square wave pulses is in order to accumulate a sufficiently large data set. If only a single step input was considered

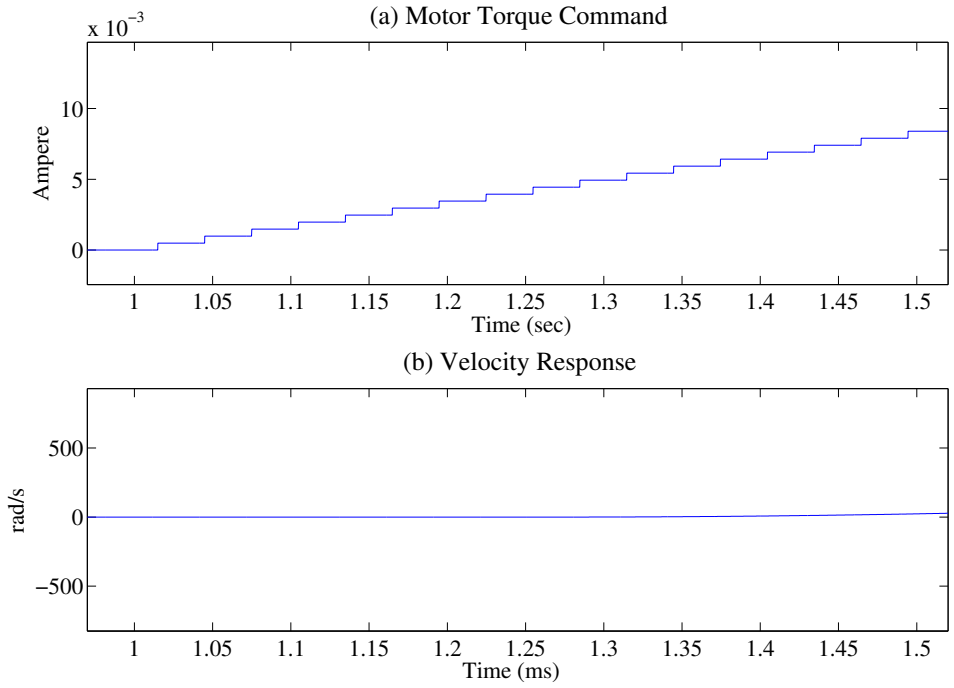


FIGURE 4.1: Design of the Break-away experiment using the Matlab/Simulink Model. The graphic depicts: (a) the command torque and (b) the motor velocity.

the experiment would have an interesting data set no larger than 300-500 samples even when using $1Khz$ sampling.

As stated previously when discussing the use of OLS, the data set must be at least seven thousand samples long in order to guarantee convergence. For this reason we repeat the input constantly in a single instance in order to enlarge the duration of the experiment. We must be careful though in selecting the duration of the step's *on and off and total time durations* T_{on}, T_{off} and T_{total} respectively, so as not to saturate the response in velocity, avoid zero velocities and prevent reaching the mechanical limits of each axis. Also a proper selection of amplitude for the step input involves selecting an excitation which would bring the velocity response to saturation in time durations much larger than the axis' mechanical time constant.

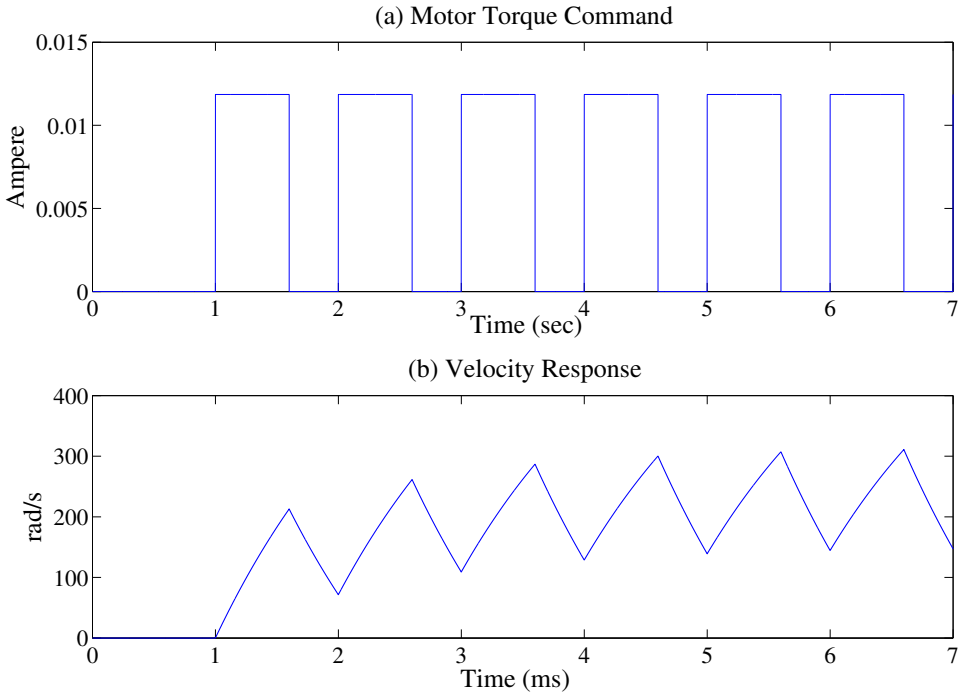


FIGURE 4.2: Design of Square Wave Pulse Response experiment using the Matlab/Simulink Model. The graphic depicts: (a) the command torque and (b) the motor velocity.

Closed Loop Experiments

The reason for considering *closed-loop* experiments is three-fold. Firstly, due to the systems's instabilities at low velocities the experiments considering fixed, known inputs cannot maintain a linear relationship between input and responses in this regime. Second, the position dependent loads that vary across the breadth of each axis undoubtedly effect these responses and so greatly interfere with the convergence to proper parameter values since we do not include position dependent friction when using OLS. Finally, given that the values of the linear controller as selected appropriately, by reversing what is assumed and what is measured (i.e. the opposite assumption from the open-loop experiments), we can now measure *directly* the effect of all the dynamically varying phenomena on the torque. This means that now we could use the resulting torque values to infer the position dependent loading from the correlation with the break-away experiments.

The procedures designed for the closed-loop experiments were inspired by the work in [27], [28] and [25]. These experiments are designed as:

EXP-CL-1. ***Frictional Torque-Velocity Mapping***

This experiment attempts to map the relationship between the actuator force/torque to velocity. A linear PID controller is used to regulate the velocity to a given *trapezoidal* reference signal. The experiment is repeated for a selected step Δv_d in *desired* steady-state velocity that samples the entire range of feasible velocities for each axis.

The experiment collects the necessary data, but the post-processing that is carried out offline selects to handle the estimates as necessary. What is meant by this is that although ideally, we would like to use the data to identify even the appropriate model for the Stribeck curve, the instability at low velocities might prevent this from being possible. Also, the same data set is also used to infer the position dependent friction or load by a different estimation procedure.

4.2 Axis Controller Scheme

Multi-axis systems, like those in Cartesian configurations have used various advanced control techniques to produce high accuracy trajectory tracking. Especially in those which are used for manufacturing tasks, the motion can be regulated to a minimal tracking error using such techniques as Cross-Coupling Control (CCC), [18], and Non-linear Disturbance Observers (NDOs), [49]. These techniques are reported to greatly improve the performance of these systems in the relative tasks.

This work however is centered on the development of the core infrastructure for the URETTS platform and not in the use of such advanced control methods. In this scope, the requirements specified in Chapter 1 regarding the performance of the control system can be met by the design of an empirically tuned Proportional-Integral-Derivative (PID) controller as is shown in Chapter 7 in the experimental results. Still we present a simple alternative control method, based on the developed system model.

This section describes a standard PID controller, tuned using the Ziegler-Nichols method and emp irical adjustment. This controller is to serve as a proof-of-concept implementation for demonstrating the capabilities of URETTS. Additionally though, it

provides a reference for all subsequent controllers when testing them against it using generally applicable performance benchmarks.

The identification experiments described in the previous section provide us with the capacity to go even further and design more specialized controllers that can improve the performance of the experiments with the LU. Both CCC and NDO though expand on the basic model based control scheme in their own way. For this purpose we only use consider a simple model based tuning method for the linear gains in order to compare against the performance of the empirically tuned version.

4.2.1 Linear PID Control

Arguably the most straight forward approach for controlling any mechanism is to make the *black-box* assumption and use empirically tuned PID control for regulating motion and not consider the system's dynamics at all.

Regarding URETTS specifically, there is an emphasis on the overall design of the system and the reduced cost this brings without the compromise of performance compared to other industrial solutions. Within this frame, a *black-box* approach demonstrates that the system behaves as required without necessitating specialized procedures to improve the performance.

We describe the design of the controller and the procedure used for its tuning. The PID controller that is used for controlling each of the MACS axes follows the standard form:

$$u(t) = K_P(x_d - x) + K_I \int (x_d - x)dt + K_D(\dot{x}_d - \dot{x}) \tag{4.11}$$

Where $u(t)$ is the controller action, x and $\dot{x}$ are the components of the system state. The reference trajectory is given as the desired values for the state x_d and $\dot{x}_d$. Finally, K_P, K_I and K_D are the proportional, integral and derivative actions respectively. When this equation is adapted for the case of controlling a PMDC motor fitted with encoders the state variables are now θ_i and $\dot{\theta}_i$ and the control value is the motor torque $\tau_{m,i}$. Equation (4.11) then becomes:

$$\tau_{m,i}(t) = K_P(\theta_{d,i} - \theta_i) + K_I \int (\theta_{d,i} - \theta_i)dt + K_D(\dot{\theta}_{d,i} - \dot{\theta}_i) \tag{4.12}$$

The reference trajectory is given as the desired values for the angular position and velocity θ_d and $\dot{\theta}_d$. We reformulate (4.12) in preparation for the use of the Ziegler-Nichols tuning method:

$$\tau_{m,i}(t) = K_P \left((\theta_{d,i} - \theta_i) + \frac{1}{T_I} \int (\theta_{d,i} - \theta_i)dt + T_D(\dot{\theta}_{d,i} - \dot{\theta}_i) \right) \tag{4.13}$$

The parameters used for Ziegler-Nichols tuning this method are:

a. K_P is the primary gain.

b. The integral time

$$T_I := \frac{K_P}{K_I} \tag{4.14}$$

c. The derivative time

$$T_D := \frac{K_D}{K_P} \tag{4.15}$$

The following procedure is followed in order to determine the appropriate tuning parameters:

1. Set K_P, K_I and K_D
2. Increase K_P until stable (non-growing) oscillations occur.
3. Record the value of $K_P = K_C$ for which the stable oscillations occur, as well as T_C, the period of oscillation.
4. Estimate the other parameters based on the values shown in Table 4.1

TABLE 4.1: Ziegler-Nichols tuning parameters based on type of desired state control.

Control Type	$\mathbf{K_P}$	$\mathbf{T_I}$	$\mathbf{T_D}$
PD	$K_P = 0.45K_C$	$T_I = {}^{T_C}/_{1.2}$	$T_D = 0$
PID	$K_P = 0.6K_C$	$T_I = {}^{T_C}/_{2}$	$T_I = {}^{T_C}/_{8}$

4.3 Trajectory Benchmarks

The towing experiments are executed on the assumption that each axis' controller, can regulate the motion following a given reference trajectory. Since we would like to create a test-bed for developing and testing various carriage control algorithms, the question remains as to how to quantify and compare the different controller performances. One way to approach this is to decide to use specific trajectory profiles as benchmark tests.

In this framework, we could test every developed controller algorithm against a reference implementation such as the empirically tuned PID controller in Section 4.2.1. The specific trajectories chosen though would have to be selected based on the following criteria:

i. The trajectories must be simple yet capable of testing the bandwidths of the actuators.

ii. The responses to these trajectories must be rich in transient features.

iii. The trajectories must be general enough to be indicative of those actually used in the towing experiments.

To meet these criteria we select two fundamental trajectory profiles; a trapezoidal velocity profile, and a sinusoidal velocity profile. The former is probably the simplest complete motion profile, other than a fixed set-point value. The latter is capable of showing the various harmonics of the system within the responses. Although these are simple motion profiles, they in fact are core to all others that could be used as they can generate almost all other types.

For instance, if we apply a sinusoidal on the smaller X axis and a trapezoidal in the larger Y axis, the EEP in effect executes a *spatial sinusoid* while moving along the length of the tank. On the other hand if we apply sinusoids on *both* axes, then the resulting path in the task-space is an ellipse or circle (depending on the amplitude and frequency selected for each axis).

Clearly these are the most appropriate tests for benchmarking the performances of any controller algorithm. Lets elaborate on the specific characteristics that each trajectory attempts to test and detail how they are implemented.

4.3.1 Trapezoidal Velocity Profiles

It is common in robotics applications to use polynomials to generate paths and trajectories. The primary advantage of polynomials is their simplicity and computational efficiency especially when considering polynomials of fifth order and lower (see [62]). One can select either a single segment directly between starting and final positions or use multiple segments and only requires that the desired values be specified for the boundary conditions of position, velocity and acceleration and total duration.

The aim is to generate a mathematically *continuous* and *smooth* path from start to finish which ensures that there are no sudden jumps (discontinuities) in the reference signals. This is especially important in the case of force/torque control where smooth profiles in acceleration mean that we can avoid jerking motions which can excite modes of vibration.

Trapezoidal velocity profiles are among the simplest and most common forms of polynomial trajectories and are ubiquitous in industrial applications. These consider three primary segments; an initial constant acceleration phase, a steady-state velocity phase and finally a constant deceleration. Integration of the velocity profile results in a smooth polynomial profile in position.

In the general case of an asymmetrical trapezoid, a total of five (5) parameters completely describe the path; three primary parameters plus a constraint and two auxiliary. The primary parameters are the steady state velocity $\dot{\theta}_{ss}$, the acceleration time t_{acc} and the deceleration time t_{dec}. The constraint can either be the total time t_{f} or the total displacement θ_f, while the auxiliary parameters are the initial time t_{init} and

the initial position θ_0. The last two are completely optional for a single segment motion but become absolutely necessary if multiple trapezoids are to be chained together consecutively. The equations which generate the trajectories are thus:

1. *Total time*:

$$t_f = t_{\text{init}} + t_{\text{acc}} + t_{\text{dec}} + t_{ss} \tag{4.16}$$

2. *Position Reference Signal*:

$$\theta_d(t) = \begin{cases} \theta_0 & , 0 \leq t < t_{\text{init}} \\ \theta_0 + \dfrac{\dot{\theta}_{ss}}{2t_{\text{acc}}} t^2 & , t_{\text{init}} \leq t < t_{\text{acc}} \\ \theta_0 + \dot{\theta}_{ss}\left(t - \dfrac{t_{\text{acc}}}{2}\right) & , t_{\text{acc}} \leq t < t_{\text{acc}} + t_{\text{ss}} \\ \theta_0 + \dot{\theta}_{ss} \cdot \Pi_{\text{pos}}(t) & , t_{\text{acc}} + t_{\text{ss}} \leq t < t_f \end{cases} \tag{4.17}$$

Where the polynomial $\Pi_{\text{pos}}(t)$ is defined as:

$$\Pi_{\text{pos}}(t) := \left(t\left(1 + \frac{t_{\text{acc}} + t_{ss}}{t_{\text{dec}}}\right) - \frac{t^2}{2t_{\text{dec}}} - \frac{t_{\text{acc}} t_{\text{dec}} - (t_{\text{acc}} + t_{\text{ss}})^2}{2t_{\text{dec}}} \right) \tag{4.18}$$

3. *Velocity Reference Signal*:

$$\dot{\theta}_d(t) = \begin{cases} 0 & , 0 \leq t < t_{\text{init}} \\ \dfrac{\theta_{ss}}{t_{\text{acc}}} & , t_{\text{init}} \leq t < t_{\text{acc}} \\ \dot{\theta}_{ss} & , t_{\text{acc}} \leq t < t_{\text{acc}} + t_{\text{ss}} \\ \dot{\theta}_{ss}\left(1 - \dfrac{t_{\text{ss}} + t_{\text{acc}}}{t_{\text{dec}}}\right) & , t_{\text{acc}} + t_{\text{ss}} \leq t < t_f \end{cases} \tag{4.19}$$

4. *Acceleration Reference Signal*:

$$\ddot{\theta}_d(t) = \begin{cases} 0 & , 0 \leq t < t_{\text{init}} \\ \dfrac{\dot{\theta}_{ss}}{t_{\text{acc}}} & , t_{\text{init}} \leq t < t_{\text{acc}} \\ 0 & , t_{\text{acc}} \leq t < t_{\text{acc}} + t_{\text{ss}} \\ -\dfrac{\dot{\theta}_{ss}}{t_{\text{dec}}} & , t_{\text{acc}} + t_{\text{ss}} \leq t < t_f \end{cases} \tag{4.20}$$

As it turns out, the final position can be calculated via the following simple relationship:

$$\theta_f = \theta_0 + \dot{\theta}_{ss}\left(t_{\text{acc}} + \frac{3}{2}t_{ss} + \frac{t_{\text{dec}}}{2}\right) \tag{4.21}$$

Figure 4.3 visualizes the trajectories as plots against time:

It is clear to see why this trajectory is so attractive. The core equations are those concerning the velocity profile; everything else is derived form those. One can adapt or optimize the final reference signal for their specific task by using (4.16) and/or (4.21) to constrain the selection of the parameters.

A noticeable disadvantage of this type of trajectory is the non-smooth acceleration. The transitions between acceleration and steady state can cause large jerks if not selected carefully, as is exactly the case with the MACS. Since the system is controlled in CC mode, then discontinuous acceleration directly implies discontinuous (on-off) torque and forcing accelerations close or even beyond the mechanical time constant will definitely cause large tracking errors at the segment changes.

These behaviors, even the undesirable, can provide measures of the effectiveness of the controller. For this reason trapezoidal profiles can provide a large amount of information about the transient behavior of the axis drive.

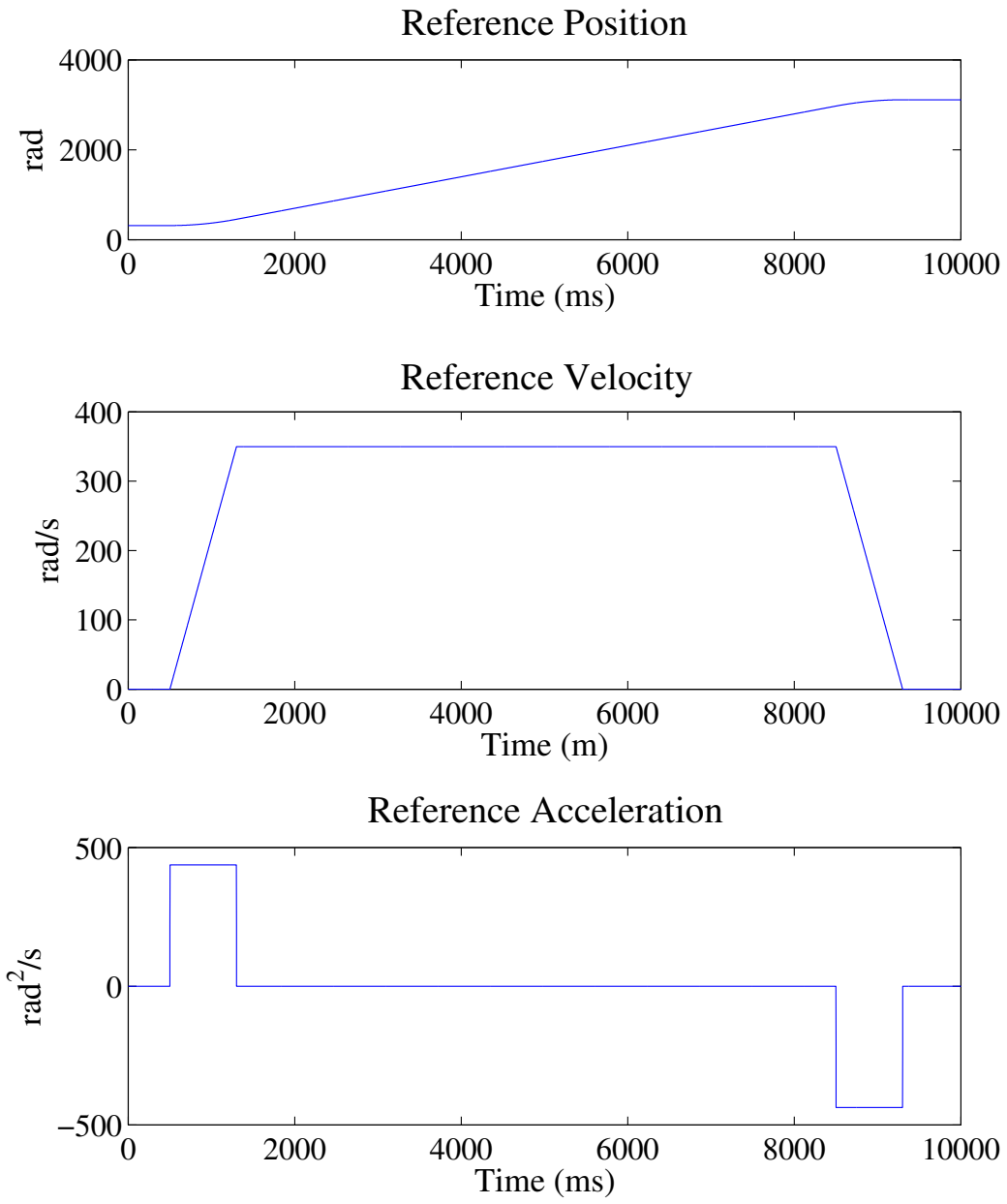


FIGURE 4.3: Trapezoidal trajectory profiles in: position (top), velocity (middle) and acceleration (bottom).

4.3.2 Sinusoidal Profiles

Sinusoidal functions are some of the most fundamental in system dynamics and are generally ubiquitous in the engineering fields. In the context of dynamics and control, they are used to analyse the harmonics of electrical and mechanical systems in order to create models directly in the frequency domain. In this application however we take advantage of their geometric properties in order to create paths smooth and continuous

paths for the EEP and thus for the TO. This serves two purposes simultaneously, i.e. by using sinusoidal reference signals we can test both the *frequency response of the closed-loop system* and the tracking performance for each axis.

Using a sinusoid in position and velocity is simple to implement and the equations are not computationally demanding. The equations describing the reference state trajectory are:

1. *Position Reference Signal*:

$$\theta_d(t) = \theta_0 - A \cdot \cos\left(\frac{2\pi}{T_s}t\right) \tag{4.22}$$

2. *Velocity Reference Signal*:

$$\dot{\theta}_d(t) = A\frac{2\pi}{T_s} \cdot \sin\left(\frac{2\pi}{T_s}t\right) \tag{4.23}$$

3. *Acceleration Reference Signal*:

$$\ddot{\theta}_d(t) = A\left(\frac{2\pi}{T_s}\right)^2 \cdot \cos\left(\frac{2\pi}{T_s}t\right) \tag{4.24}$$

A is the oscillation amplitude, θ_0 is bias value of the oscillation and T_s is the period of oscillation. These three parameters alone can determine the entire motion of the system.

In the case of sinusoidal trajectories, we see that the initial reference acceleration could be instantaneously large and could therefore excite jerking motions. This is in fact on other property which we can test using this type of reference signal. Another is the large accelerations which occur at the peaks of the sinusoid. What is interesting here

is that we bring the system to limits of possible acceleration in a continuous manner without having to instantaneously excite it. This means we can avoid the oscillations due to the chain elasticity and push the system to its limits. We can expect the maximum tracking error to occur at these limit accelerations.

Part III

Implementation

Chapter 5

Underwater Robotics Experimental Tow-Tank Servomechanism

This chapter presents the design and implementation of the URETTS hardware components. So far we have seen the system's design requirements, an analysis of the kinematic properties has been described and the dynamics of its operation have been explored. To complement these analyses, the design of each major component and specifications of the system's operation are presented.

By describing how the MACS and EMCP sub-systems integrate in terms of their inter-operation, the individual design requirements are determined. This permits us to adopt a top-down approach for the design of each sub-system individually. However, the fact that the MACS existed prior to the current work, means that a bottom-up approach is equally justified. For this reason, both approaches are adopted and applied appropriately.

Regardless of the approach taken, it is clear that specific dependencies exist between the sub-systems. The components selected for the MACS constrain the design of the EMCP and this in turn constrains certain aspects of the software. This dependency is therefore used to define how the sub-systems interface, while the internal design of each can be done independently. Hence, the internally top-down design is checked against an external bottom-up in order to fit the existing with the newly developed components.

The BBMC software stack is described exclusively in Chapter 6 and so any references to it in this chapter are only with respect to the co-design of the EMCP, to which it is tied.

5.1 Multi-Axis Carriage Servomechanism

The components of the MACS –for the most part– have already been selected in the work of Ntouskas et al., as described in [54]. The primary components that comprise this sub-system are the motor mounts and TC structures, the chain drive actuation system, the motor actuator modules and the payload mount for the EEP. These were selected based on the following requirements:

ME-NT-1. The linear velocity of the EEP frame would not exceed $60^{cm}/_{s}$.

ME-NT-2. The motor's RPM does not exceed 4000rpm.

ME-NT-3. The maximum payload mounted on the EEP would not exceed $10kg$.

ME-NT-4. Current Control capability must be available.

ME-NT-5. The motors must be controllable using signals from micro-controllers.

ME-NT-6. Power supply capable of providing maximum 5Amp of total current.

ME-NT-7. Light in weight so it can be supported directly on the glass water tank.

ME-NT-8. Total travel distance of $5m$ along the length of the tank.

ME-NT-9. Cartesian actuation.

Although still applicable, these requirements were not sufficient for the total design of the system (see Chapter 2). In fact, they are met only partially by the original selection of components. Several mismatches occurred which necessitated adaptation of certain parts in order to function properly. The design of each mechanical component is detailed below in the respective sections, and the advantages and drawbacks are discussed. Amendments to these faults were partially addressed in this work and Chapter 9 describes the required improvements for future iterations.

5.1.1 Mechanical Design

Carriage Assembly

The most prominent feature of the MACS is the rectangular frame, i.e. the base of the TC. This structure executes the motions along the Cartesian Y axis, which is aligned along the *length* of the tank. Within this frame, a rectangular platform –the EEP– moves perpendicular to the motion of the carriage and implements the Cartesian X axis and is aligned to the *width* of the tank. In Figure 5.1 the TC and EEP are pictured moving along the tank.

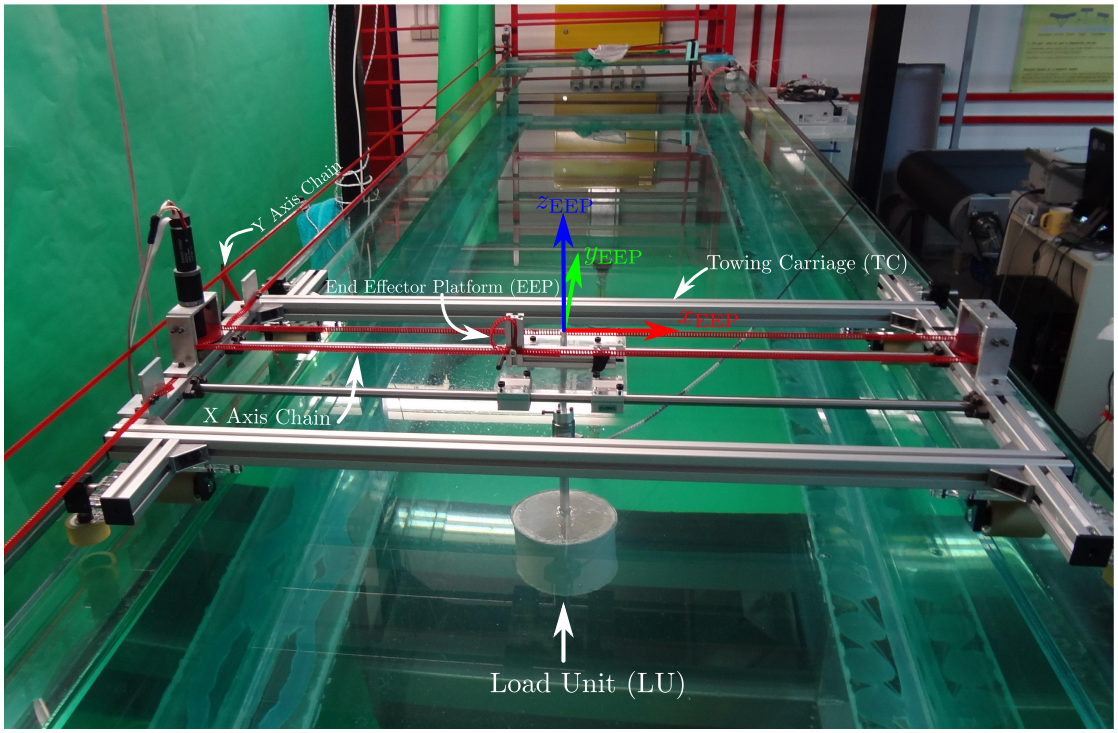


FIGURE 5.1: The Towing Carriage in motion executing a linear manoeuvre while towing a cylindrical Test Object.

The global inertial frame is positioned exactly on the shaft of the Y axis motor. The position and orientation of this frame were selected for ease of the user so that all manoeuvres could be executed and monitored in an understandable way. From the perspective of the control station (i.e. desktop with primary host computer) this global frame is oriented in a way commonly depicted in physics and mathematics literature. Figure 5.2 depicts the position and orientation of the primary frames of reference used for the motion of the MACS as seen from this perspective.

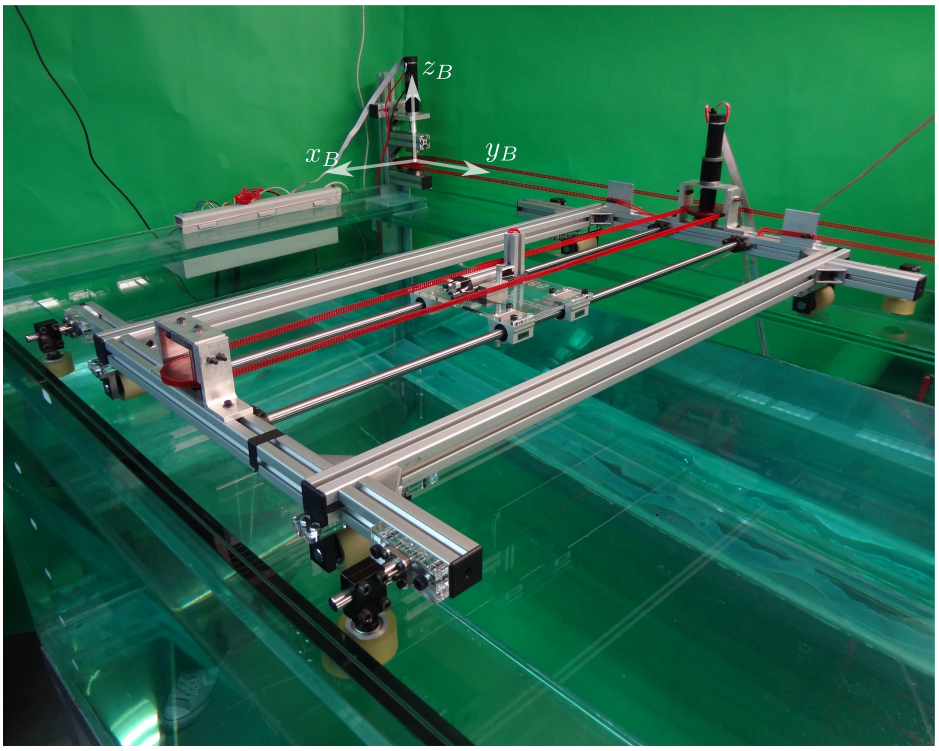


FIGURE 5.2: The orientation and position of the inertial frame on the base structure as viewed from the control station.

The TC frame is constructed using aluminium modular struts from Misumi Corp. with linear guides along their lengths. These guides allow a multitude of configurations for the structure. Figure 5.3 shows the TC and EEP completely disassembled in the lab and depicts the components used. Attached on the underside are small rollers and housings. These drive on the horizontal guides on the top of the tank and provide the linear motion of the TC. Additionally, lateral support is provided with smaller rollers that guide the motion of the TC and prevent it from rotating in the horizontal plane. Both primary and lateral rollers were also manufactured by Misumi.

One problem with this design is that it was not anticipated that any deformation of the tank would occur. Specifically, due to the load of the water, the tank has deflected outwards in the middle section by up to 2cm from the nominal position. This results in variable contact of the rollers with the lateral walls and is the primary contributor of position dependent load (which in the model presented in Chapter 3 was considered as friction). This however, has not produced any significant issues with the MACS operation apart from some strain and wear on the lateral guides.

Moreover, the EEP's motion is implemented using linear bearings on steel shafts.

The EEP itself is not one single component but is in fact any plate mounted on the linear bearings. The first and current versions are constructed from $6mm$ plexi-glass and appropriate holes positioned in the geometric center of the plate serve as the mount point for the TO-SS. Reconfigurability of the EEP allows quick and efficient adaptation for varying payloads. However, due to prolonged loading, the steel shafts have been measured to deflect downward by approximately $1mm$, but this has not affected motion in any noticeable way.

On the top side of the TC, customized aluminium mounts, host the chain drive components for the X axis, the attachment point for the Y axis chain and the X axis actuator module. These were designed and manufacturesịn-house at CSL.

Despite the mechanical *wear-and-tear*, the TC-EEP structure has faired well, given the extensive use and increased humidity in the lab. Furthermore, the significantly low overall cost of the system negates any further costs associated with physical damage. Quite simply, the components are readily available, low cost and therefore easily replaceable.

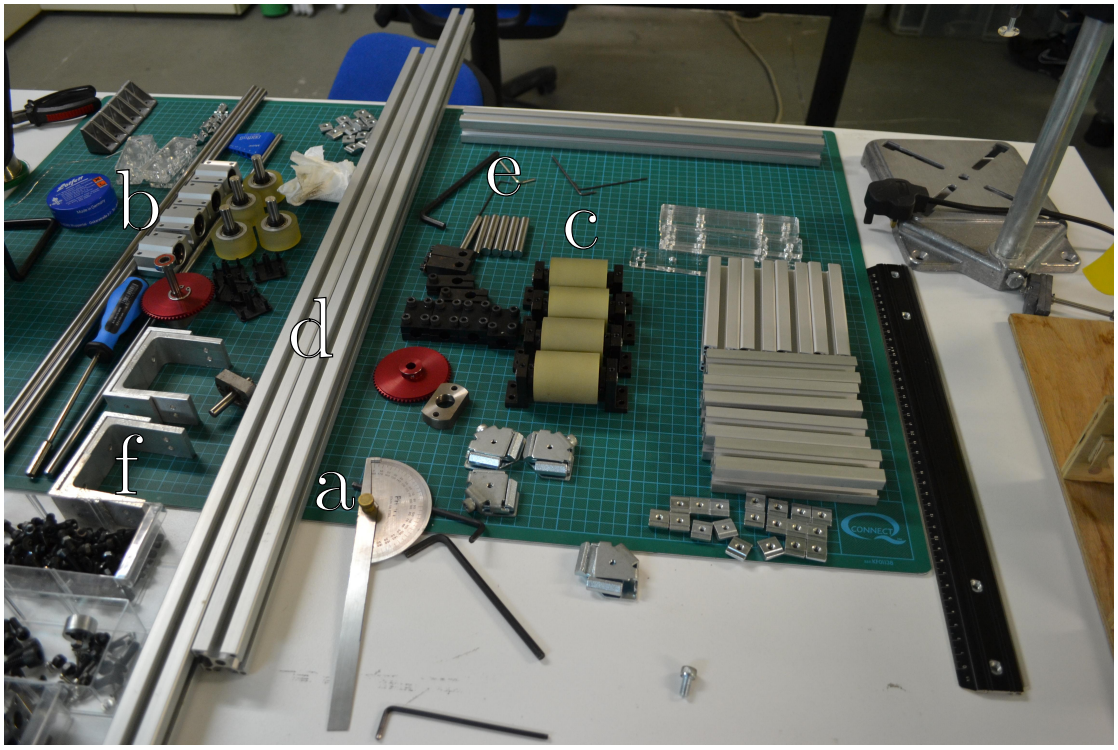


FIGURE 5.3: The TC and EEP disassembled on the workbench. In this image, one can see: (a) the primary struts , (b) linear bearings, (c) large and small rollers, (d) chain drive sprockets, (e) steel axles and shafts and (f) aluminium mounts

Chain Drive

The chain drive provides the linear actuation for each axis. Both Y and X axes use exactly the same configuration consisting of a pair of sprockets which circulate a reinforced *plastic chain* (with internal steel threading) which is attached onto a moving mass and guided along a *linear support*. Each sprocket pair consists of one passive and one driven sprocket which are otherwise identical in geometry and mass. The driven sprocket of each axis is actuated via a motor plus planetary-gear transmission in order to provide both increased resolution in position and increased output torque. The linear support mentioned previously is nothing more than the rollers and linear bearings for the Y and X axis respectively.

As was discussed in Chapter 3, this drive configuration has an undesirable oscillatory behavior. This is due to strong vibrations in the chain which occur with large, sharp accelerations (i.e. large jerks) of the actuator. Although this is the primary drawback of using plastic chains, it does not pose a significant problem for the MACS's operation, as the majority of the underwater experiments involve relatively smooth paths of the TO and/or constant velocity segments. Moreover, the significantly reduced costs and weight compared to using other linear mechanisms such as ball-screws, greatly outweigh the issues due to the vibrations, which do not necessarily occur and are naturally dampened anyway.

Furthermore, because both primary axes use an identical and perpendicular configuration, we can adopt independent and similar controllers to regulate the motion of each. This greatly reduces the computational requirements for the design of the EMCP and BBMC which are used to control the MACS. This therefore gives us the capacity to use low cost embedded electronics for the control of the system. This is detailed in Section 5.2.

Mechanical Loads

The LU mount is situated at the geometric center of the EEP. This ensures that it is also close to the EEP's center of gravity so as not to exert any additional moments (torques) during motion. As described in Chapter 1, the LU consists of the mount base, the force sensor adapter, the SS (support shaft) and the TO (test object).

The mount base and SS are manufactured in-house at CSL. The TO can be any object which is used in the experiments, like the CSL robotic fish. Finally, there is the possibility to attach a multi-DoF force/torque sensor between the mount and SS in order to directly measure the forces acting on the TO. Figure 5.4 shows a complete LU used in the case-study of Chapter 8.

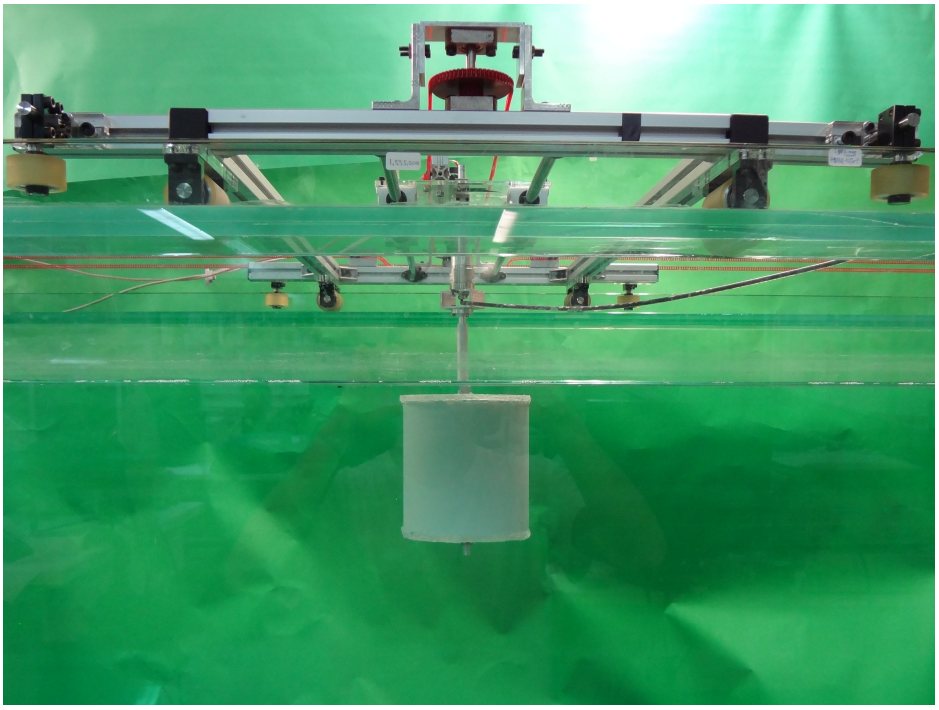


FIGURE 5.4: A sample LU used in Tsounis, Papadopoulos et al. ([52]) consisting of a simple hollow plastic cylinder.

5.1.2 Actuation

Motor Actuators

The actuators used for the motion of the TC and EEP are both Rare-Earth Brushed PMDC motors from *Maxon Motor*. They are both fitted with planetary-gear transmissions with a $35.97 : 1$ reduction ratio and Magnetic Resonance (MR series) type incremental encoders with 1000 counts per rotation. The data-sheets for these can be found in Appendix A. Figure 5.5 shows an axis motor with the attached encoder on the motor head and the transmission on the output.

FIGURE 5.5: Maxon RE series motors fitted the MR type incremental encoders and planetary-gear transmissions.

The motors used for the X and Y axis are 25W and 30W Brushed PMDC from Maxon's RE series. These are medium range motors that provide substantial mechanical efficiency (approximately 90%) that remains relatively high, even when fitted with the planetary gears. The primary advantage of these motors is the ease with which they can be controlled; unlike brush-less DC (BLDC) motors which require specialized electronics and three armature terminals, Brushed motors only require two connections on the armature and a power source. The difference lies in the commutation of the armature current which is done mechanically in Brushed DC and electronically in BLDC.

The disadvantages include the inevitable wear which the brushes experience over time, as well as the increased weight and form factor compared to BLDC. Also, other disadvantages include the increased rotor inertia and the increased friction at low velocities. For the operation of the URETTS, the latter factors do not pose an issue since the motions are rated well below the maximum motor velocities.

Finally, it is important to mention that although the motors were selected for $24V$ nominal operation, this was incompatible with the selected transmissions. Both RE-25 and RE-30 motors were fitted with the GP-32 planetary-gear transmission which is rated for 4000 rpm maximum operation. The motors, however, operate at a maximum

of 8810 rpm and 9560 rpm respectively when powered at $24V$. These velocities are over double the limit of the transmissions and for this reason the power supply was reduced to $12V$ in order to reduce the motors' output or shaft velocity.

Armature and Encoder Signals

This section serves a precursor for the design of the EMCP described in Section 5.2. The operation of each actuator involves the transmission of two signal channels; the sensor input signal from the encoder and the output control signal from the BeagleBone.

Regarding the inputs from the encoder, the signals generated from the sensors internal electronics are 0-5V square wave quadrature A-B signals and an auxiliary index I signal. The primary A-B pair consists of identical square wave signals with a 90^o offset between them. The on-time of the pulse is proportional to the velocity of the motor shaft and so measurements of the period and the number of High-Low transitions can provide measurements of the position and velocity of the motor. Whether the 90^o phase difference is positive or negative determines whether the rotation is counter-clockwise or clockwise respectively. All three channels are output in differential, complementary pairs, e.g. A, $\bar{A}$ for use with specialized electronics that remove noise or interference from the environment. Figure 5.6 depicts the waveforms of the quadrature signals.

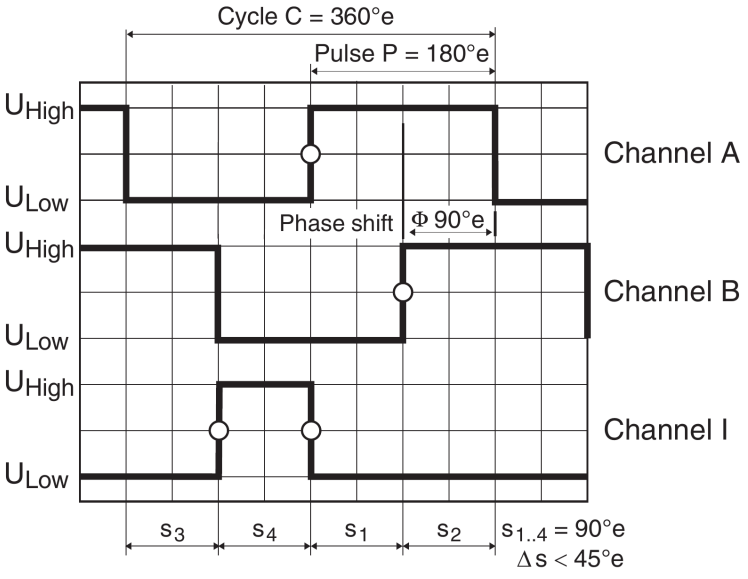


FIGURE 5.6: Quadrature signals output from the MR encoder

Each motor needs a controllable voltage applied on the terminals of the armature circuit. Since we require both forward and reverse rotations, the total applied signal ν_C must be a differential voltage, i.e. neither terminals are connected to ground, so that we can have both a negative and positive voltage applied to the armature. This poses certain safety concerns since the applied voltage does not have a direct path to ground and is therefore floating. This risk is mitigated by the fact that the amplifier has a direct connection to true Earth-Ground that is shorted in case of a fault.

Finally, we see from Figure 5.7 an overview of a single MACS motor signal paths and connections:

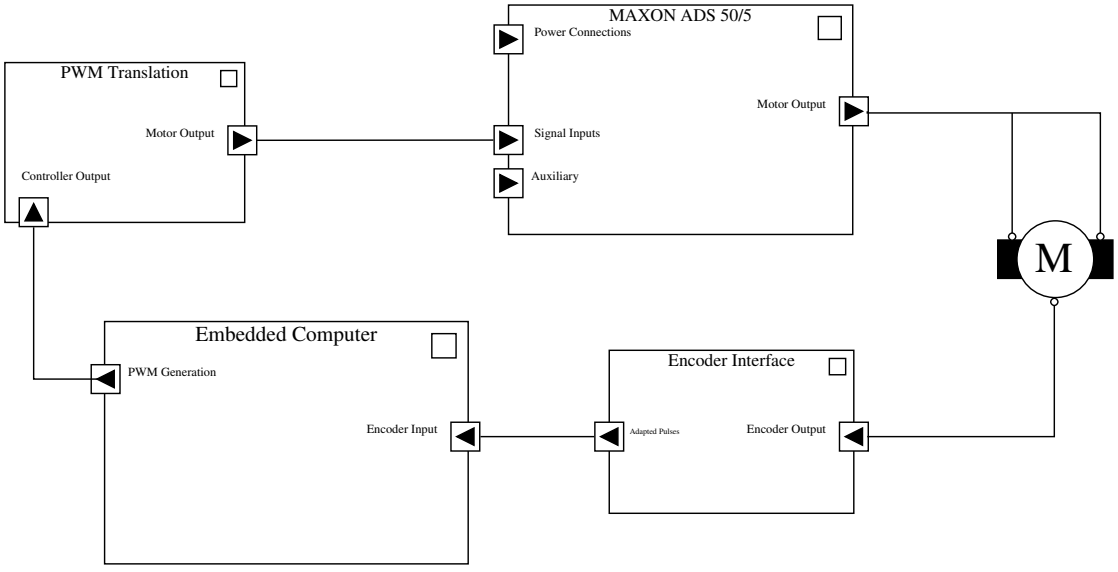


FIGURE 5.7: Connection overview for each axis motor module.

Driver Amplifiers

The driver amplifier provides controlled, regulated electrical power to the armature circuit of the motor. The selection of the amplifier is equally important to the selection of the motors themselves and the two must be compatible. Therefore, for both the RE-25 and RE-30 motors, a ADS 50/5 servo-amplifier from Maxon was selected. These motor amplifiers rated for 12-50V DC supply voltage and a 5A maximum continuous output current (hence the 50/5 designation).

This servo-amplifier is CC (current control) enabled and therefore provides direct control of the motor's output torque. The input signal is a DC voltage in the -10/+10V

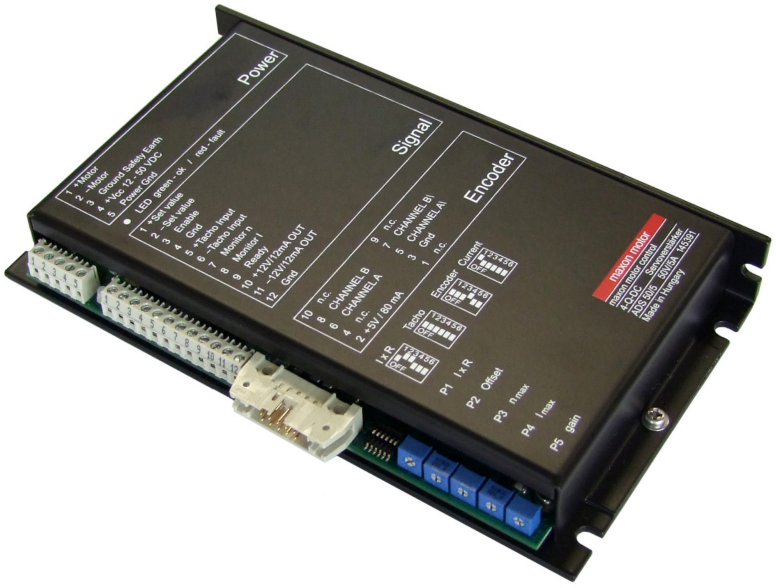


FIGURE 5.8: The Maxon ADS 50/5 servo-amplifier.

range which maps to the output torque via the product of the motor's torque constant K_T and the armature current $i_m(t)$.

Moreover, exploration of the amplifiers documentation has revealed that in fact, PI control is applied to regulate the armature current in CC mode. We can identify the *outline* of the circuit used to implement this capability. Figure 5.9 shows the circuit found in the motor driver's manual:

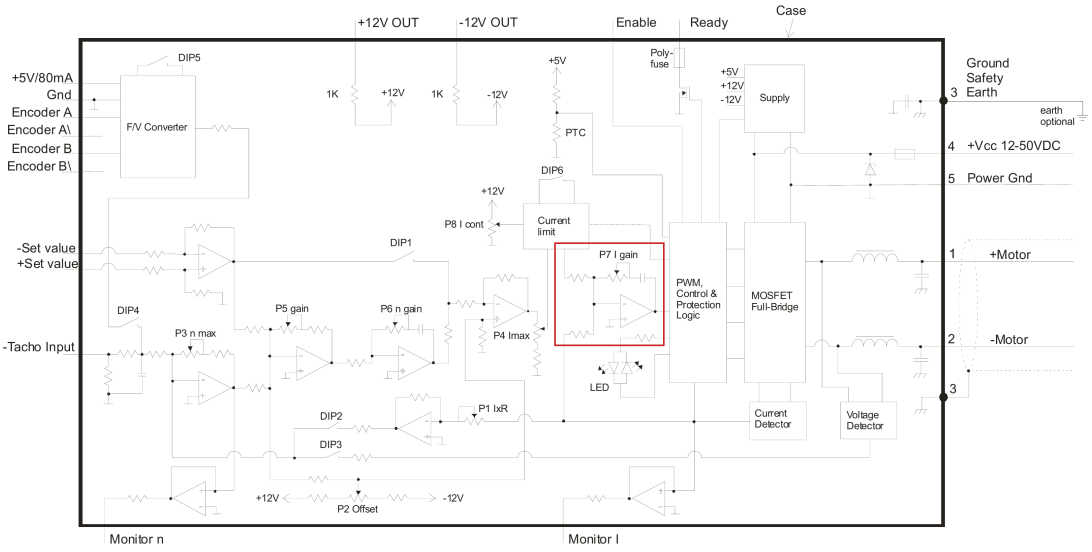


FIGURE 5.9: Outline of the ADS's internal circuitry. The topology used to implement Current Control is shown in the red frame.

This circuit uses an Operation Amplifier (Op-Amp) topology with *inverting* feedback (see [63]). The general case of this configuration is show in Figure 5.10.

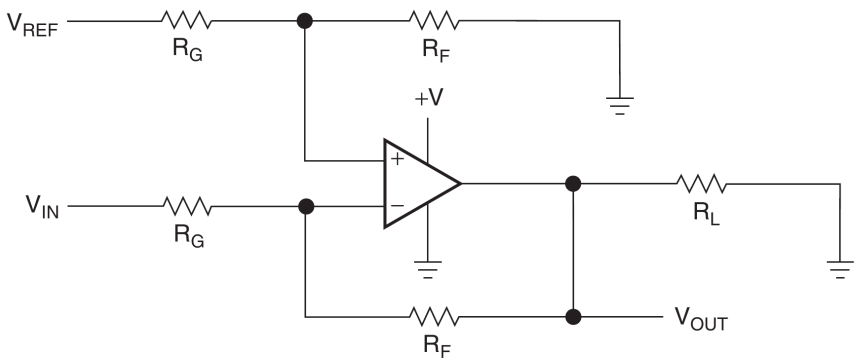


FIGURE 5.10: Generalized Op-Amp topology using inverting feedback.

To implement the PI linear controller, the circuit places a capacitor in series with the feedback resistor and also grounds the non-inverting input. This op-amp topology is named the *Augmented Integrator* and is described in [63]. The final circuit is shown in Figure 5.11 and the transfer function can be found by applying Kirchoff's current law at the potentials at the op-amp's input nodes and the output. This results in:

$$G_{\mathrm{AI}}(s) := \frac{\nu_o(s)}{\nu_i(s)} = -\frac{j\omega R_o C_o + 1}{j\omega R_i C_o} = -\frac{R_o}{R_i} - \frac{1}{j\omega R_i C_o} \tag{5.1}$$

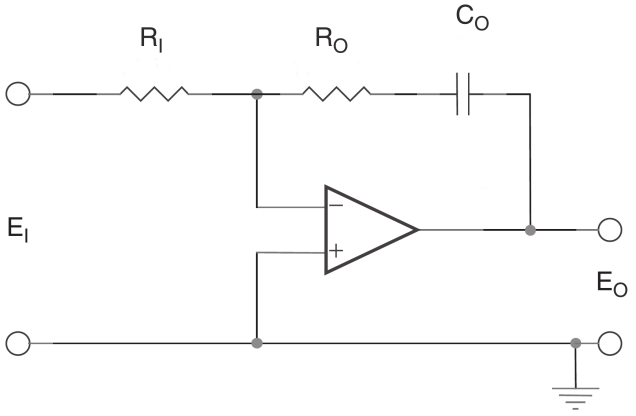


FIGURE 5.11: Augmented Integrator topology used to implement the PI Current Controller.

5.2 Embedded Motion Control Platform

The Embedded Motion Control Platform (EMCP) comprises the electronic portion of the URETTS hardware. It consists of two primary components, the BeagleBone embedded computer and the specialized Control Signal Interface (CSI). At the time this document was written, the EMCP used the BeagleBone White Rev. A6 and the alpha (testing) version of the CSI. The total system is depicted in Figure 5.12.[1]

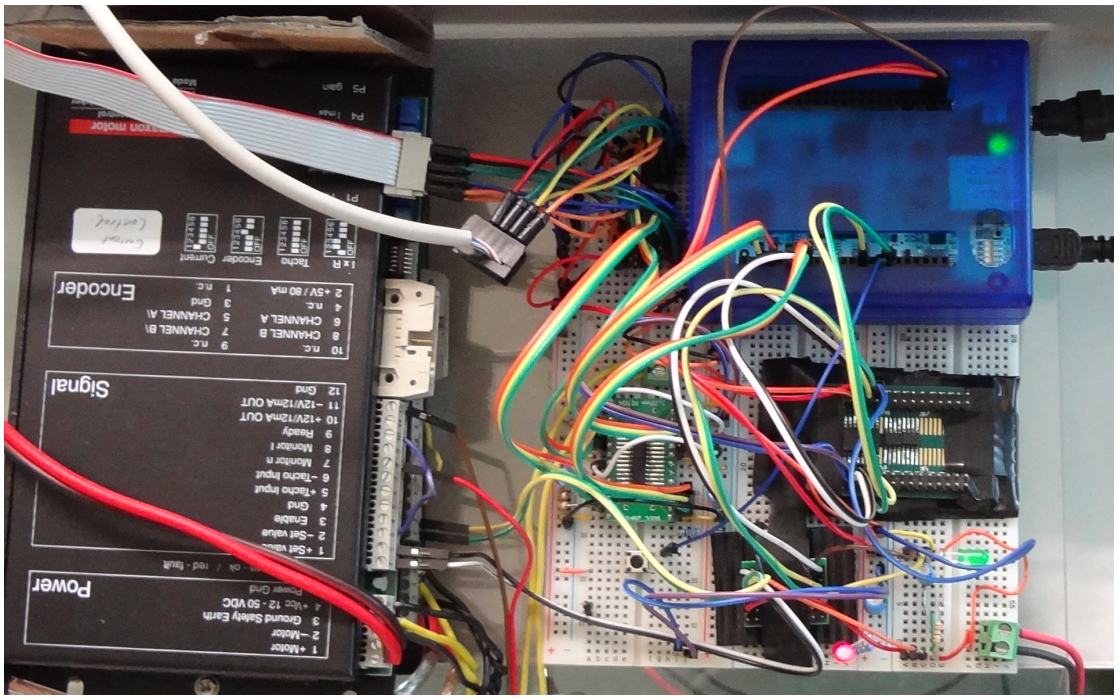


FIGURE 5.12: The first complete version (v0.2) of the EMCP.

5.2.1 Signal Specifications

The signals that are transmitted over the CSI to and from the BeagleBone and ADS are the incoming quadrature signals from the encoder and the output PWM . Due to incompatibilities between the raw signals and the end-receivers, specialized signal paths had to be designed and implemented.

[1] Any similarities with the *flying spaghetti monster* were not intentional.

Encoder Input

As mentioned in the previous section, the encoder signals are generated as 0-5V square wave quadrature signals. These adhere to TTL signal specifications, which have moderate state transition thresholds and are not as strict as CMOS specifications. Figure 5.13 shows the different signal categories and the respective margins.

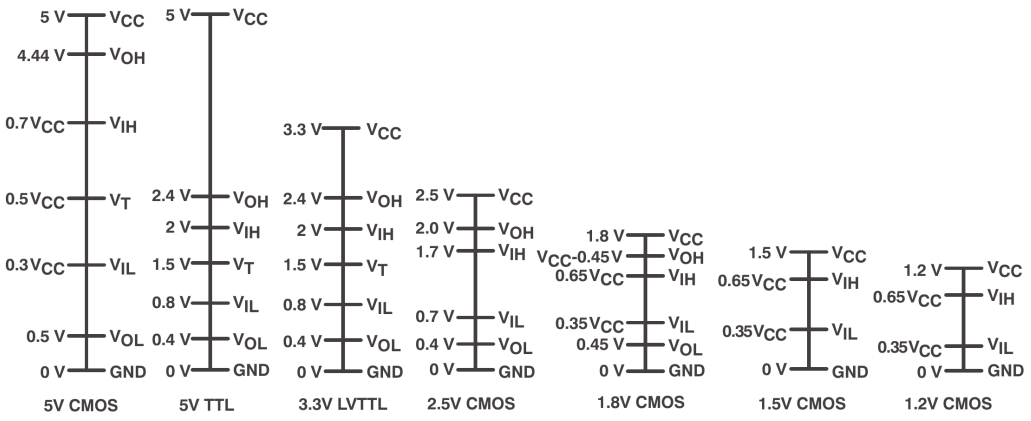


FIGURE 5.13: State transition margins for the various digital signal specifications.

Regarding the input from the encoder the main design requirements are:

ENC-R-1. The differential pair of each channel of the encoder must be used to remove *common-mode noise or interference.*

ENC-R-2. The *TTL* signal must be translated to adhere to *LVCMOS* specifications.

ENC-R-3. The *0-5V* quadrature signals from the encoder must be level-translated down to *3.3V.*

ENC-R-4. Total signal path propagation delay must not exceed $100\mu s$.

Furthermore, another feature which is not mandatory, but is desirable is the input signal path to be powered exclusively by the BeagleBone so that the circuit remains isolated from the ADS and motor, i.e. the power stage.

Controller Output

On the other hand, the PWM output must also be adapted for the ADS's expected input. Though using a specialized Integrated Circuit (IC) chip such as 12-bit DAC would be ideal for generating the appropriate signal, this would not only add complexity to the system but also delay. The latter is due to the fact that most DACs are controlled via common communication protocols such as SPI or I^2C, and this would add considerable communication delay between initial output of the controller and final output in torque.

To address this we took advantage of the intrinsic properties of Pulse Width Modulation (PWM) signals. PWM waveforms output a controllable mean value via control of the waveforms *Duty Cycle*. This means that if the waveform's period is short enough, the electronics using it as input would only "see" the resulting mean value. The reasoning is that, as the input impedance of the IC's circuit acts as a Low-Pass-Filter (LPF) the IC's input would be dominated by the slower components, i.e. DC-like values of the PWM. The design requirements for translating the PWM to the appropriate input signals for the ADS are:

PWM-R-1. Translation of 0-5V *LVCMOS* PWM signals to -10/+10V range.

PWM-R-2. Electrical isolation of the BeagleBone from the power stage of the ADS servo-amplifiers.

PWM-R-3. The propagation delay must not exceed $100\mu s$.

PWM-R-4. The final input the ADS must behave like DC signal.

PWM-R-5. The resolution of the control torque must be at approximately 10-bit or more.

The circuit is to be powered primarily by an external power supply, the same one in fact, which provides power to the ADS servo-amplifiers. Initially, a Siemens SITOP power supply was selected which provides 24V nominal DC voltage and 5A maximum continuous current. This matches the selection of the motors and amplifiers. However, since the necessary reduction of the supply voltage to 12V, the SITOP could not be reconfigured to meet this value. For this reason, development and testing of the EMCP's

power stage was powered by external DC power supplies available in the lab[2] set at 12V and 4.5A Over Current Protection (OCP) set-up. The EMCP's PWM channel was powered by a Hameg HM7042-5 Triple Power Supply to provide the 5V and 10V power buses. These two supplies were connected with a common Power Ground and provide a true Earth-Ground connection for safety.

The final signal is a high-frequency 20V peak-to-peak differential PWM signal running at 100kHz. Due to selected configuration, a maximum of 10-bit resolution was achieved for the actuator control. Furthermore, the switching in the input of the amplifier, although it occurs at a significantly high rate, there is a residual oscillation which propagates to the output PWM on the motor armature.

5.2.2 Control Signal Interface Design

Using the previous requirements to constrain the selection of solution ICs, we arrive at the first complete version of the *CSI, v0.2*. There is a clear independence between the input and output channels which is maintained throughout the design of the system. The details of each signal path's design are presented below.

Encoder Input Signal Path

The first stage of the signal path from the encoders consists of a *differential line-receiver* IC which takes the difference between the differential (complementary) signal pairs and removes common-mode noise. This is termed Common-Mode Rejection (CMR) and is based on the idea that since all channels travel along the same bundle of wires, they experience the same interference. This results in the injection of a *common* noise signal in all the wires. By taking the difference between each quadrature signal and its complements, this common-mode noise is removed. The IC used is the SN75175N from Texas Instruments (TI).

The second stage consits of the level translation which brings the 0-5V TTL signal from the encoder and outputs a 0-3.3V LVCMOS digital signal. This is implemented using the TXB0104 *level-translator* also from TI.

[2]The ADS was powered by an *Agilent U8002A* 5Amp PS.

Figure 5.14 shows the schematic of the total circuit for the encoder signal path. The total propagation delay is measured on the oscilloscope[3] to be within $20\,ns$ and the total current draw from the power supply is $30\,mA$. These values are well above the required performances and the latter value translates to the power consumption being within the BeagleBone's output capacity.

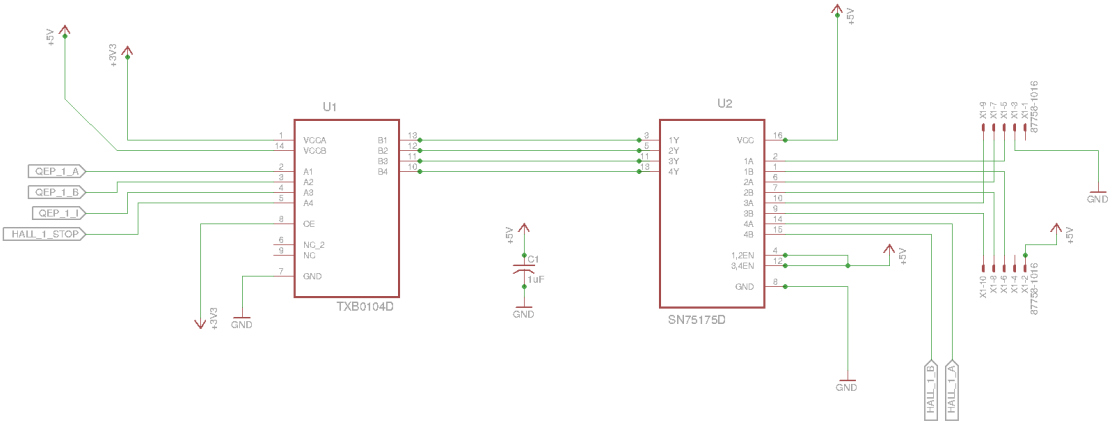


FIGURE 5.14: The sub-circuit responsible for interfacing a single input signal channel between the BeagleBone and an MR encoder.

PWM Output Signal Path

We require both directions of motion to be controllable, the voltage across the V_s^+ and V_s^- input terminals of the ADS must also have negative values. The BeagleBone has three (3) PWM generator peripherals devices, each capable of outputting two (2) independent PWM channels. This means six (6) independent PWM outputs in total. By using the *difference* in PWM channels from each peripheral we can have three (3) differential PWM channels which can output in both negative and positive voltage.

Therefore, to control the two axes motors we output a total of four (4) PWM channels at a base frequency of 100KHz. The software takes care of the appropriate modes of operation so that only one channel is active per motor, i.e. the other in each differential pair is grounded (set to zero). Using these high frequency PWM signals we can create DC-like voltages on the across the input terminals of the servo-amplifiers.

The PWM translation consists of two stages. The first stage of the PWM output consists of the ISO7640M digital isolator. This chip effectively separates electrically the input from the output circuit using galvanic isolation and requires two separate

[3]The oscilloscope used is a MSOX3014A from Agilent Technologies.

ground references to be connected. Additionally, it translates each PWM output from the BeagleBone from a 0-3.3V to a 0-5V waveform. The second stage translates the low-voltage 0-5V isolated PWM signals to the 0-10V medium voltages. The final circuit is shown in Figure 5.15.

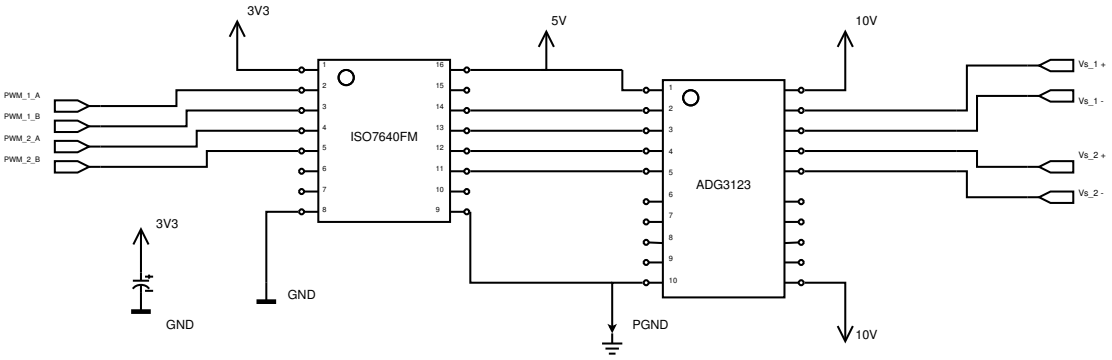


FIGURE 5.15: The sub-circuit responsible for interfacing the BeagleBone's PWM channels to the ADS 50/5 amplifiers.

BeagleBone Connections

Lastly, before concluding this chapter we document the configuration and pin layout of the BeagleBone with respect to the electrical connections in the EMCP. The BeagleBone has two pin headers P8 on the right and P9 on the left, as can be seen in Appendix B. The *pinout* is highly configurable, as the BeagleBone actually has more signals than there are pins to populate. For this reason the signals are multiplexed with a dedicated hardware register handling this configuration.

Specifically for the EMCP connection, the pins were configured so that the signals could be grouped according to the input output channels. Apart from the power connection, which can only be made on the P9 header (left), the data channels are all configured on the P8, therefore enabling the construction of two connectors; a 32-pin data connector and a 3-pin power connector.

Chapter 6

BeagleBone Motion Controller

Computer control of mechanisms is at the core of automation and robotics. With the availability of low-cost, high-performance integrated electronics, one can control from the smallest motor or MEMS (Micro Electro-Mechanical System) actuator to the most complex mult-DoF robotic system using just a few of micro-controllers, a debugger and a serial communication cable.

In academia and industry, there are *de-facto* embedded platforms which are used for robotics, mechatronics and controls research. These include the several platforms conforming to the PC-104 standard, FPGAs and even ordinary desktops fitted with specialized input-output interfaces like dSPACE systems. These are usually proprietary and licenses can cost several thousands of Euros or Dollars.

The work in this thesis focuses on creating a low-cost platform for conducting experiments in underwater robotics. Because the low-cost aspect must be retained throughout all aspects of the design, emphasis is placed on the overall cost of the control electronics, which includes the selection of then computer for controlling the system.

Recently, with the increased popularity of the *maker* and open-source communities, an ever growing population of professionals and enthusiasts propagate a new model for software and hardware development. Examples of these platforms resulting from these trends include the Arduino, BeagleBoard.org and Raspberry-Pi. One of the most interesting aspects of these is how quickly the shift came from embedded firmware on micro-controllers to embedded versions of complete Operating Systems (OSs) like Linux.

In line with this rapidly growing market, the choice was made to select an open-hardware and open-source platform that can meet the computational and functional requirements for the implementation of motion control in URETTS. Table 6.1 summarizes a comparison of the considered solutions in terms of which platform has the best in each feature category.

TABLE 6.1: Comparison of considered embedded computers for the EMCP.

Platform Capabilities	Arduino Mega	Raspberry-Pi - Rev.B	BeagleBoard-xM	BeagleBone - Rev.A6
Processor	-	-	X	X
Co-processors	-	-	-	X
Memory (RAM)	-	-	X	-
Storage (ROM)	-	X	X	X
Communication	-	-	-	X
PWM channels	X	-	-	-
GPIO signals	X	-	-	-
Encoder Inputs	-	-	-	X
USB	X	X	X	X
Ethernet	-	X	X	X
Wifi	-	X	X	X
Linux Compatible	-	X	X	X
Software Tools	X	X	X	X
Ease of Programming	X	-	-	-

The only solution which – at the time of this selection – met all the requirements and best fitted the needs of the URETTS was the BeagleBone. [1] In fact, since the time when the project was initiated, no other system was available that could integrate all the desired components for use in URETTS.

This chapter begins with a description of the BeagleBone and the relevant aspects concerning URETTS, which include the hardware, software, and available development

[1] In the opinion of the author, an article with a fair comparison between the Raspberry-Pi and the BeagleBone Black is "raspberry pi or beaglebone black?"

tools. Moreover, the BBMC software stack will be presented and the design of its primary components will be described. This chapter concludes with an overview of the commands in the BBMC that are relevant to the functionalities used in the experiments presented in Chapter 7.

6.1 The BeagleBone Embedded Computer

6.1.1 Board Description

The BeagleBone, pictured here in Figure 6.1, is a small form factor embedded computer targeted at applications which require both fast computation and intensive input-output processing and communication. At its core, it is an ARM based computer running on an Cortex-A8 micro-processor which is inegrated into the AM335X[2] line of System-on-Chips (SoCs) in the *Sitara* series from Texas Instruments.

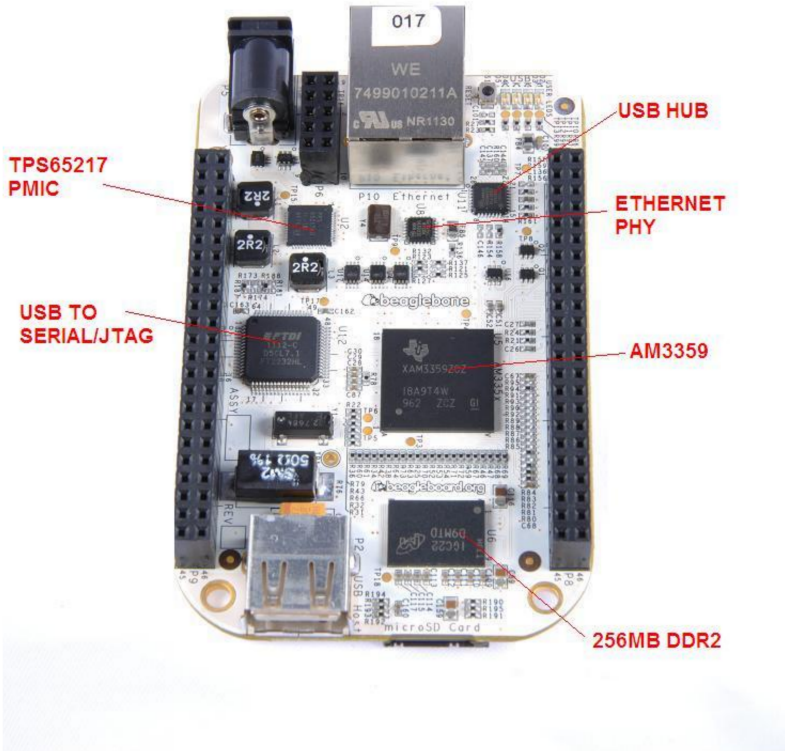


FIGURE 6.1: The BeagleBone Embedded Computer.

The platform's initial release was in Q4 of 2011 and has since experienced many major revisions and a successor platform, the *BeagleBone Black*. The version used in

[2]The first versions were shipped with an XAM3359ZCZ72 processor IC, but as of 2014, it has transitioned to the final release version which comes with a AM3358ZCZ72.

the implementation of URETTS is the BeagleBone (original) Rev.6. The BeagleBone's primary features, i.e. those which are most relevant for our implementation are presented in Table 6.2.

TABLE 6.2: BeagleBone Features

Processing	720MHz 32-bit Cortex-A8 CPU
	2x 32-bit PRU-ICSS, real-time co-processors
	Cortex-M3 Wakeup/Power-down co-processor for power management
	VFPv3 and NEON hardware Floating-Point and vectorization units
Memory	64KB internal (non-cached) RAM
	256MB DDR2 external RAM
Input - Output	6-channel PWM output
	3-channel quadrature encoder input
	12-bit ADC
	UART, SPI, I^2C, CAN, Ethernet communication
Timing	Real-Time Clock (RTC)
	8 Digital Timers
Storage	4GB microSD ROM storage
Power Supply	3.3V signals on the P8 and P9 pin-out headers, a total of 92 pins
	5V input power supply
	5V, and 3.3V output power buses
Programming	JTAG debugger
	FTDI serial-over-USB connection IC, routed to he host port

The board can be powered either by the USB connection to host computer or a 5V external power supply. When powered by the latter option, the BeagleBone can output a 5V supply voltage on the P9 header that can be used to power other electronics added externally such as the so called BeagleBone *Capes*[3]. These are expansion boards that are literally *stacked* on top of the BeagleBone, connected on the primary headers and extend the functionality of the original board in varying ways. Third-party companies and hobbyists have provided a large selection of expansion capes.

[3]BeagleBone Capes at eLinux.org

Regarding the implementation of the software stack, the primary *take-away* from Table 6.2 is the processing power, the PWM outputs, encoder inputs, the GPIO signals and the debugger interface. These are described below in the design of the BeagleBone Motion Controller (BBMC) software stack.

6.1.2 ARM System-on-Chip

The AM3359 SoC provides an immense abundance of peripherals for interfacing with other devices. In fact in its category, there is no other processor IC with similar input-output capabilities and comparable cost. This is the primary advantage the BeagleBone has over the other candidates and this is exactly why it is a SoC and not a micro-processor. This one single chip integrates all the functionalities which in conventional computers would be spread over multiple stack modules or external cards. It is by now way the most powerful system of this type, especially when one considers the capabilities of FPGAs. However, to the author's knowledge there is no FPGA at a similar cost and at the level of maturity that makes it usable directly by novices.

To better understand the level of integration, Figure 6.2 provides a visualization of the internal components of the Sitara AM335x series of SoC devices:

As is common in many microprocessors and microcontrollers, all the internal components are mapped directly to the memory space of the system. Considering that the system is 32-bit, the total addressable memory space is considerably large. Thankfully only a subset of the total memory space actually maps to something. All data transfers in the memory space are conducted over the so called L3-L4 internal connection network (interconnect). This allows the processor but as well as the other co-processors to also access all peripheral devices also connected in this interconnect.

The BeagleBone has added a 256MB external DDR2 RAM where the users primary code is stored by default. A fixed program in the ROM (pre-programmed in the actual silicon) always seeks the user's code at a particular location, address `0x8000000` external RAM (the `0x` prefix means that the values following represent a value in hexadecimal format).

For the BBMC, the most important peripherals, and in fact the ones for which software was developed are:

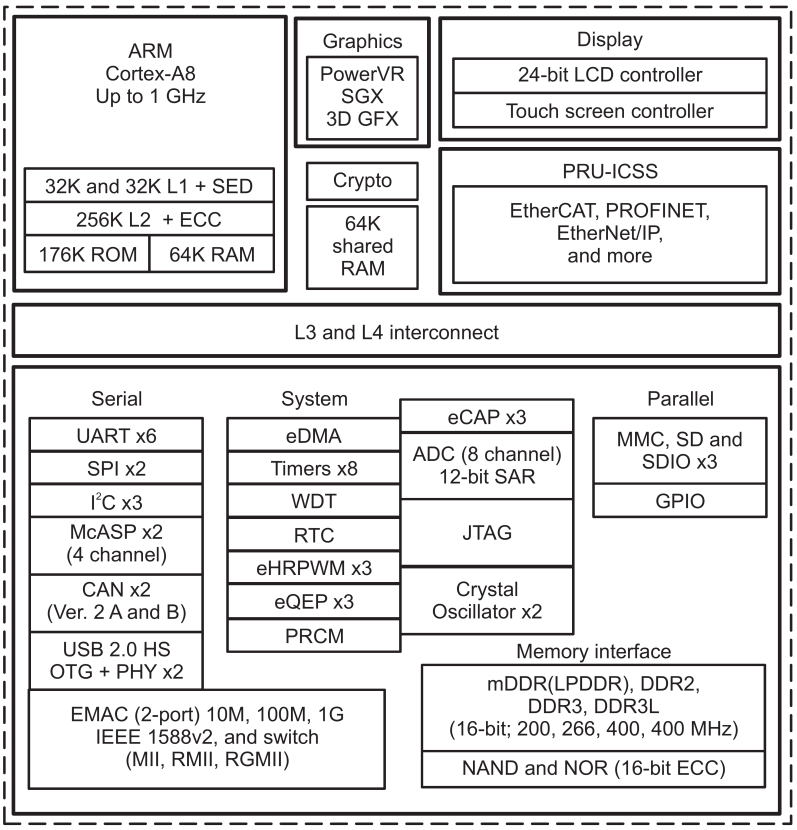


FIGURE 6.2: Internal components of the Sitara AM335x series SoCs.

i. *eHRPWM*:

This device generates a controllable 0-3.3V PWM signal. The internal configuration device registers can be programmed so that the PWM can be changed online by the software. The user can specify the frequency and the duty in a direct way. Additionally, each instance of this peripheral can output two simultaneous PWM channels which can be configures to be independent or can be tied together. This provides the user with may possible configurations which can be used to control external devices.

ii. *eQEP*:

This peripheral is responsible for receiving and decoding the quadrature signals from an incremental encoder. The peripheral can provide many configurations to change the way the input signals are interpreted. Taking advantage of the quadrature phase difference, the decoding of the position is effectively quadrupled, meaning that using an encoder with N_{enc} counts per revolution, the final resolution in position is $4 \cdot N_{\text{enc}}$. Also, the device has an internal *capture* module which can be used to

time the duration of the input waveform and thus produce *period-based* estimation of velocity (see section 3.4).

iii. *DMTimer*:

The internal digital timers provide high-resolution time keeping capability which can be configured to either measure large scale durations, in the order of minutes, to really small scale durations, in the order of micro-seconds. These can generate an interrupt to the execution of the processor when a programmed duration has lapsed. These are used in BBMC to implement high frequency Real-Time motion control.

iv. *GPIO*:

The General Purpose Input-Output device can provide multiple single bit data channels that can interface with many different devices and electronics. These provide an easy and direct way to communicate and control the operation of external components.

v. *AINTC*:

This is the peripheral responsible for managing the interrupt signals receive from all the other peripheral devices on the L3-L4 interconnect. This sends the interrupt request to the internal interrupt module of the Cortex-A8 processor, which acts according to the users programmed functionality to handle interrupt requests.

vi. *UART*:

UART devices are ubiquitous in all modern computer systems and are most commonly used to interface with external devices using an RS232 or RS485 type of connection for serial communication. The Serial-over-USB connection to the host computer from the BeagleBone essentially serves the Rx/Tx functions of the UART0 module over the USB0 port of the AM3359. Also, additional UARTS ports are mapped to the P8 and P9 headers.

6.1.3 Development Tools

Most users will program on the BeaglBone using the standard POSIX[4] API in Linux and one of the several compatible languages like Python, Java, C/C++, Ruby etc. In

[4]For more information, see POSIX API programming.

fact, apart from the absence of a Desktop Environment, the experience of using a Linux distribution on the BeagleBone is otherwise indistinguishable from using one on any normal desktop computer.

As long as there is a version of your favourite compiler or development tool which has been ported for ARM architectures and Cortex-A8 processors, you are *good-to-go*. GCC and the other compilers provided by the *GNU Project* have already been available for several years with ever increasing support and stability. Figure 6.3 shows using a host computer running Ubuntu, interfacing to the Angstrom Linux on the BeagleBone via a terminal running the `screen` program.

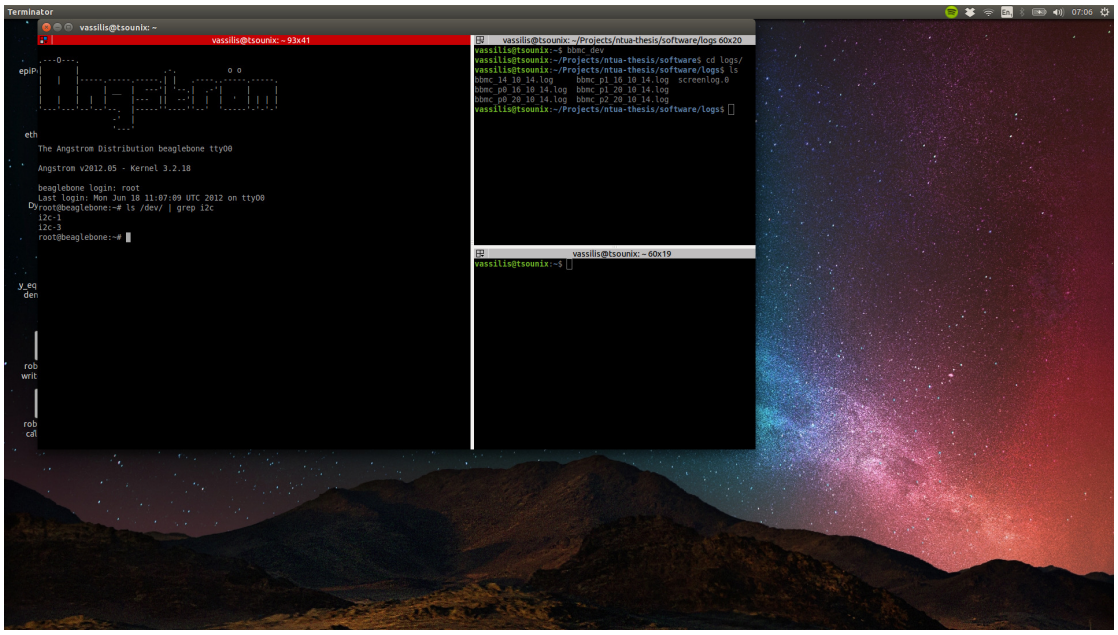


FIGURE 6.3: Using Angstrom Linux on the BeagleBone. The host computer is running Ubuntu.

Additionally for advanced users, especially ones wanting to debug the code on the lowest level (even for those crazy enough to look at the Assembly code) the tool officially provided by Texas Instruments is the *Code Composer Studio* (CCS) Integrated Development Environment (IDE). This development tool is essentially an extension plug-in (with additional features) to the common *Eclipse* IDE, which is a popular tool for development in C/C++ and Java. Figure 6.4 demonstrates the use of CCS to debug the BeagleBone online.

The primary features of interest for using CCS for the development of the BBMC is the capability to debug and monitor the processor and peripheral device registers online. Code stepping, breakpoint creation and interrupt debugging functionality make this an

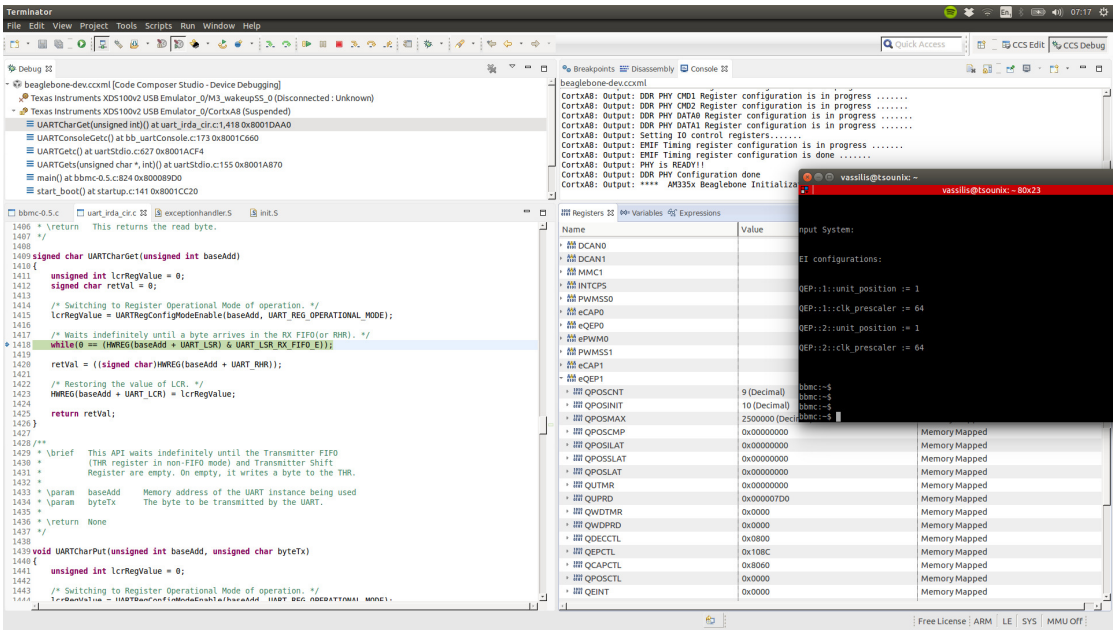


FIGURE 6.4: Debugging software on the BeagleBone using Code Composer Studio's environment.

invaluable tool for demanding software development on the BeagleBone. For further information see the relevant documentation.[5]

Finally, for the most demanding users who do not want to run an entire Linux kernel, Texas Instruments provides the *StarterWare*[6] Software Development Kit (SDK) for developing firmware for the AM335x series SoCs. This SDK is open-source, with a custom license from TI. It is free to download and is the most low-level way of programming on the BeagleBone, as the programmer is completely exposed to all the inner workings of the system and so to all the responsibilities and risks that this entails.

The StarterWare SDK serves not as base for programming the peripherals of a AM335x, as drivers for most internal devices are already provided. It's scope is not limited to only bare-metal programming (i.e. no OS). It is also a good starting point for kernel driver developers and is reported to work directly with TI-RTOS, a dedicated real-time micro-kernel OS.

[5] CCSv6 IDE documentation
[6] TI starterware open-source SDK

6.2 BeagleBone Motion Controller Design Overview

The BBMC is the dedicated software stack for the control of the URETTS platform written in the C programming language. It integrates the lower-level functionalities available on the BeagleBone's ARM SoC in order to provide a complete embedded motion control system meeting *hard real-time* execution requirements. It is based on the aforementioned StarteWare SDK and provides the firmware device drivers, *middleware* and functional layers necessary for executing required the experimental procedures.

The BBMC software uses two open-source licenses; the custom license from TI for the modified and expanded components derived from StarterWare, as well as an GPLv3 license for all custom software components developed which are independent of the SDK. The source code is hosted on public repository[7] and can be accessed either by direct download or via the use of `Git`[8].

6.2.1 System Design

Before describing the implementation, we must consider the limitations of the used SDK in order to understand how to appropriately design and adapt the software. One important point is that the software is designed so that future iterations may not only reuse several of the approaches or specialized algorithms, but may outright use the existing source code, needing only to adapt certain file and device dependencies.[9]

Since the BBMC is written as bare-metal code running on the processor without the support of an OS or RTOS, it makes less assumptions on what the underlying software libraries can provide. This is due to the fact that the StarterWare SDK does not provide the following components commonly found on most complete systems (i.e. those using a OS like Linux):

1. There is no file system meaning that data cannot be stored locally and retained after a restart or power-down.

[7] BBMC on github.org
[8] Git software version control system.
[9] This process is referred to as software *porting*. In essentially means that one takes an existing piece of software and adapts parts of it, in order to migrate the functionality to a completely different system or processor.

2. There is no support for dynamic allocation of memory as the C library used by the SDK does not provide functions such as `malloc()`.

3. There is no primary environment for online programming as there is no OS and relevant facilities.

Despite these limitations the system was designed so that the desired experiments and operations can be executed and configured online. The user accesses the system using a Command Line Interface (CLI) which communicates with the BeagleBone via the primary USB connection to the host computer.

The BBMC software has been developed by applying various software engineering principles that provide the necessary flexibility and robustness in the code. These techniques are found in the various literature and material on the internet. The primary principles and concepts adopted for BBMC include:

i. *Modularity.*

ii. *Encapsulation* and implementation hiding.

iii. Standardized interfacing of software components/modules (POSIX-like function calls).

iv. Structured access to device hardware.

Lastly, we note that the resulting system achieves the goals of real-time and deterministic execution as well as retaining a structure that isolates each functional layer. The primary motivation for this design approach is so that once the system has demonstrated proven stability and maturity, it can be migrated to the complete environment of an OS, specifically an Ubuntu distribution of Linux. This enables the use of the BBMC as either a standalone software package, or to be integrated into a dedicated robotics middleware system like ROS[10].

6.2.2 System Architecture

The resulting system consists of four (4) primary functional layers. These can be understood to separate the higher-level operations from the lower level functionalities that

[10] ROS on the BeagleBone

maintain internal data, states and interfaces to the several sub-systems. Figure 6.5 portrays an overview of the system's layered architecture. These are outlined in the continuation.

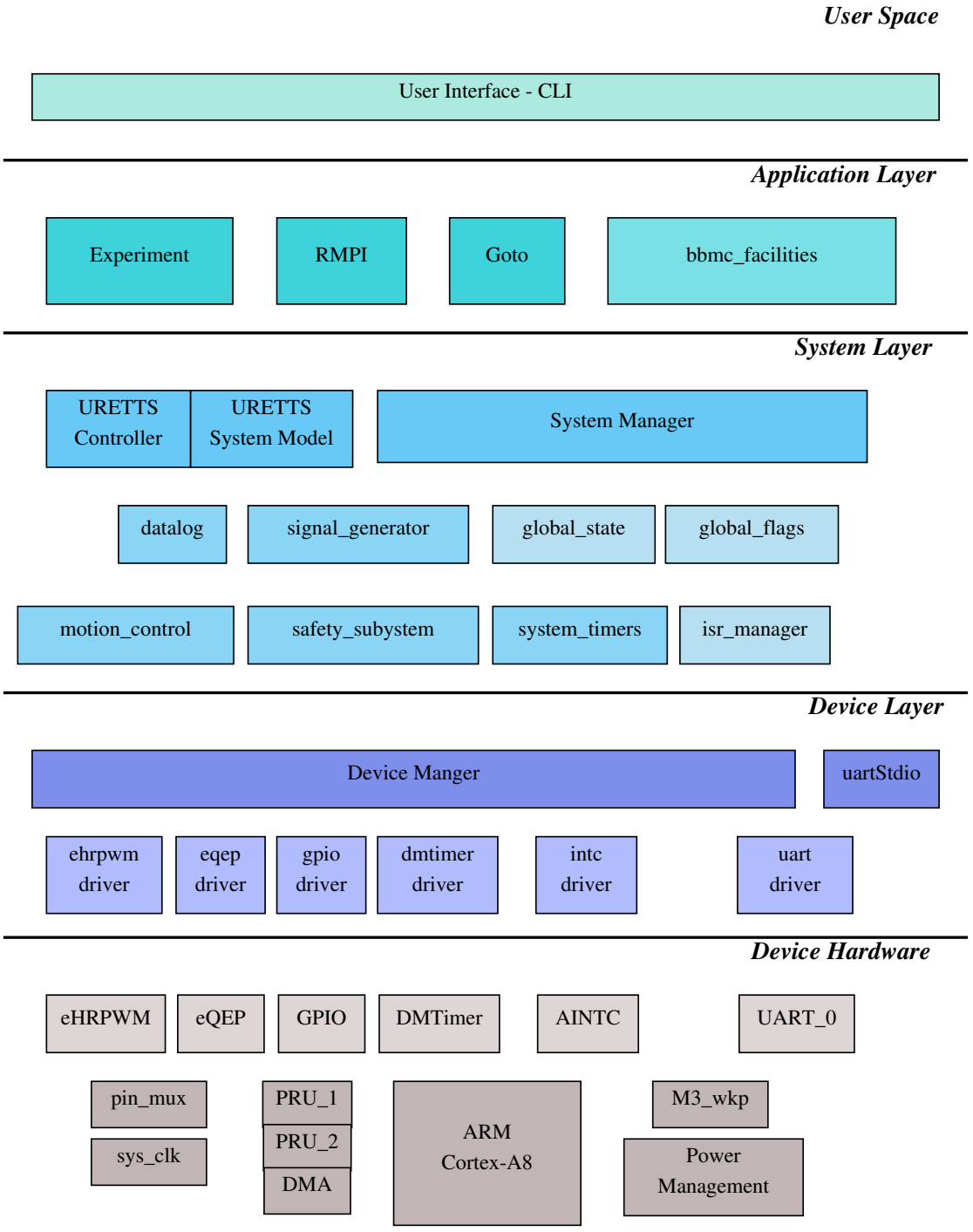


FIGURE 6.5: Overview of the BBMC functional layers.

The BBMC, over its many iterations has adopted various structures, with ever increasing complexity. There are significant challenges to developing reusable software that enables faster and more efficient execution of experiments. From the very beginning,

an effort was made to avoid task specific code that was usable only to execute single motion profiles and which could not be interrupted and monitored in a controllable way.

Indeed it is tempting to write code that *just does the job*. However, this approach fails in the case of programming the URETTS due to complexities in its operation. Despite being simple in terms of mechanical configuration and dynamical characteristics, the system must be able to monitor functional limits such as software end stops and is very cumbersome to manually reset after every experiment.

For these reasons and in order to meet the requirements stated in Chapter 2, a system was developed that could *automate* several of the procedures and experiments. Also, it can provide online reconfiguration and a change of settings such as trajectory parameters and controller gains. To this end, a properly structured approach was necessary in order to efficiently develop, organize, test and maintain the software.

Application Layer

This is where the high level operations occur. These include determining which subroutines are executed as well as handling the necessary configurations of the system prior to the execution of an experiment. The user issues commands via the CLI which are then sent to this layer where the primary and optional parameters specified are interpreted in order to begin the configurations. This layer is primarily procedural in that the types of actions are pre-determined, following a standardized execution path.

The application layer consists of three primary services:

a. *Experiment/Run*: The primary experiment execution services such as the trajectory benchmarks described in Chapter 4.

b. *Response-based Model Parameter Identification*: The *RMPI* service is used to execute single-axis experiments and tests on the system.

c. *Goto* This is a pick-and-place type service, where the user specifies a set of Cartesian X-Y coordinates over the tank, where the TC-EEP should position itself.

System Layer

The System Layer (SL) consists of and manages the internal state data of the system (not only dynamical state) using specialized data-types implemented as C `struct` types. Moreover, it contains the dedicated subsystems like `motion_control` which provide the interfaces to the input-output devices during real-time execution. There are also components which manage the system interrupts, data-logging and trajectory generation functionalities.

The core component in this layer is the *System Manager* through which all configuration requests from the application layer are processed and executed accordingly. There are of-course built in states which maintain a level of safety and request checking in order to ensure that invalid requests do not propagate not to the other system components and device layer.

Each component in this layer is essentially a static C library, which also maintains internal data structures representing the module's state and data. Encapsulation and data protection approaches used here are inspired by object-oriented practices, and implemented in C rather in C++. The primary reason for this is that the StarterWare SDK and compatible version of embedded GCC compiler have full support only for C programming.

Device Layer

This device layer is where all hardware accesses for communication, data input-output, timing and interrupt handling is managed. It is comprised of the device hardware device drivers for both core and peripheral units of the AM335x SoC, as well as the `device_manager` component. The latter is responsible for servicing all access requests to the device drivers and provides a layer of protection against erroneous operations.

While the application and system layer where mainly platform independent, this layer is in fact, directly tied to the specific devices found on the BeagleBone. An interfacing technique is applied to integrate the device layer with the rest of the BBMC stack. This entails using *type-renaming* to redefine standard C types and structures into data formats which are exported as types compatible with those used in the core of the system level.

By using this method of type definitions, the device layer can have and internal bottom-up design that is appropriate for the available hardware. This is done independently of the other layers and only requires that the final accesses are standardized. The standardization used here is that of the POSIX API. It is by no means compliant to the full extent with the POSIX standard, rather it adheres to the common API definitions. Thus for every device driver which *encapsulates* the functionality of a hardware unit in the SoC, a similar API is defined using the following function calls:

i. `<device>_open()`

ii. `<device>_close()`

iii. `<device>_read()`

iv. `<device>_write()`

v. `<device>_init()`

vi. `<device>_config()`

The `<device>_open()` and `<device>_close()` are used in Unix-like systems to handle *file descriptors*. In the case of BBMC, each driver maintains internal state metadata for each device, e.g. the eHRPWM driver maintains separate states for all three PWM units. This way the relevant operations can always check this state when they are called in order to safely manage access to the hardware.

Part IV

Application

Chapter 7

Experimental Results

This chapter presents the primary results of this thesis. These include both numerical simulations executed in Matlab/Simulink as well as experiments on the physical platform. The analyses in Chapter 3 presented various aspects of the system's dynamics and Chapter 4 described the design of control methods for the regulation of motion during tracking tasks.

Experiments were performed over a period of several months and during various stages of development. The configuration of the MACS poses several challenges with respect to executing certain types of experiments. This is primarily due to the fact that the towing carriage is guided along both axes of motion using rollers. Whenever the system requires maintenance or re-configuration, it is necessary to at least partially re-assemble it, resulting in a measurable change in behavior.

As is evident in the experiments presented here, although some *global behaviour* is retained across reconfigurations of the system, many local effects vary. Thus the system cannot have *exactly* reproducible responses over large periods of time as it is inevitable that some manual handling of the carriage and re-assembly is bound to occur.

Still, the developed identification procedures manage to be rapidly applicable, and can thus be run after each such reconfiguration of the TC-EEP. These procedures can be re-executed to estimate the new parameter set. This capability and various others which the system exhibits are presented accordingly.

Most experiments to be presented in this chapter are using the larger Y axis, i.e. the TC carriage. The reason for this, is that most experiments regarding the STOs will be executed along this axis which increases the priority in terms of project development. Furthermore, the smaller X axis is much simpler and considerably faster in terms of response.

7.1 Preliminary Tests

Before we can perform the target identification and benchmark experiments, we need to configure the system and properly characterize the operation. This entails ensuring that we properly measure and map the input-output behavior of the MACS and EMCP.

Specifically, this involves measuring and characterizing the gain parameters necessary to map the PWM outputs of the BeagleBone to the applied motor torques and determine actuator saturation values.

7.1.1 Actuator Parameters

Using the Agilent MSOX oscilloscope, a multi-meter and velocity measurements we perform *unit tests* on each stage of the actuator control channel of the EMCP. The amplifiers for each motor were identically configured and the resulting parameters are presented in Table 7.1.

Table 7.1: Actuator Configuration Parameters

Parameter	**Symbol**	**Value**	**Units**
Motor Torque Constant	K_T	26.7e-3	$N \cdot m/A$
Current Control Gain	K_{amp}	0.37	A/V
Maximum Current	$I_{\max}$	3.7	A
PWM Amplitude	V_{CC}	10	V

7.1.2 Controller Design

The controller used throughout the various experiments was tuned using the Ziegler-Nichols tuning method. The procedure described in Chapter 3 was executed and then manually adapted in order to further improve the performance of tracking tasks. The final parameters used are presented in Table 7.2:

TABLE 7.2: Parameters Values for Y axis system.

Parameter	**Value**	**Units**
$K_{P,y}$	1.32e-6	$\frac{Nm}{rad}$
$K_{I,y}$	3.29e-10	$\frac{N \cdot m}{rad \cdot s}$
$K_{D,y}$	1.32e-7	$\frac{N \cdot m \cdot s}{rad}$

7.2 System Identification

The SI experiments were executed with the primary goal of understanding both the qualitative and quantitative properties of the MACS. This endeavour serves the dual purpose of firstly enabling the design of controllers with improved performance and secondly it allows us to predict the acting forces during motion. The former increases the reliability of the assumed trajectories when the combined TC-EEP-STO configuration is in motion. The latter can be used to indirectly measure the external hydrodynamic forces acting on the STO *without* using a force sensor. This last concept is explored and *proven* experimentally in Chapter 8.

First we present the results of the open-loop responses and then expand to closed loop identification methods. The combination of the two techniques provides an improved understanding of the system as each covers different aspects of the behavior. While open loop methods manifest the intrinsic characteristics of the system, closing the loop can provide useful information about the way the system responds to external loads and disturbances.

7.2.1 Open Loop Experiments

Break-Away Experiments

The *break-away* experiments were executed in order to measure the stiction force/-torques, but also to test for positional dependence. This experiment posed quite a challenge in terms execution as the tank extends quite a measurable distance. Unlike the work of *Chasparis* and *Papadopoulos* in [27] were similar experiments were executed on a ball-screw mechanism, on the URETTS they proved impractical, especially along the longer Y axis.

Due to technical difficulties and a great degree of trial and error, it was decided that a complete map of the stiction would not be required for this current research. Instead, the decision was made to sample three regions of tank and use these to create a rough, average estimate of stiction. The three regions lied on the left side of the tank, close to the origin of the global tank frame.

The procedure described in Chapter 4 was executed over a total distance of about 0.52 meters. The experiment consisted in total the execution of the procedure a total of 628 iterations to cover the aforementioned length. A sample instance of the procedure is shown in Figure 7.1.

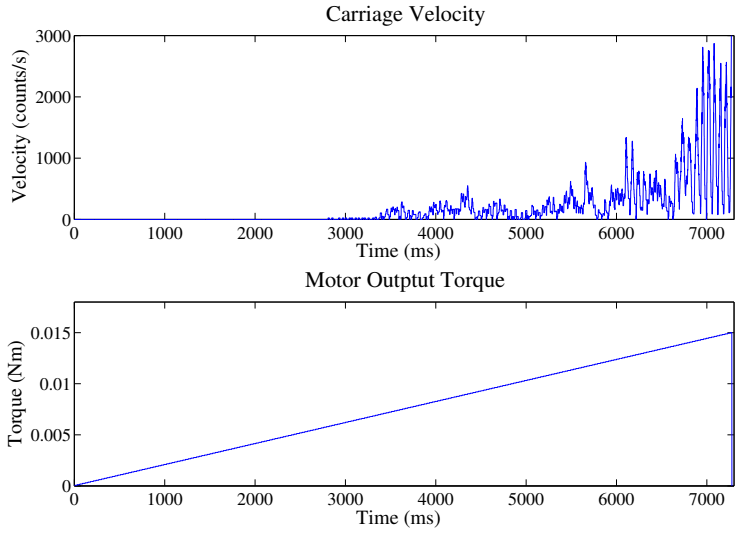


FIGURE 7.1: An example instance of the a Break-Away experiment.

The slow creep motion around the 3000ms marker is immediately noticeable as also is the fact that at these low velocities, thee state quantization is considerable. If we were to focus in on a region, as is portrayed in Figure 7.2, these become even more clear.

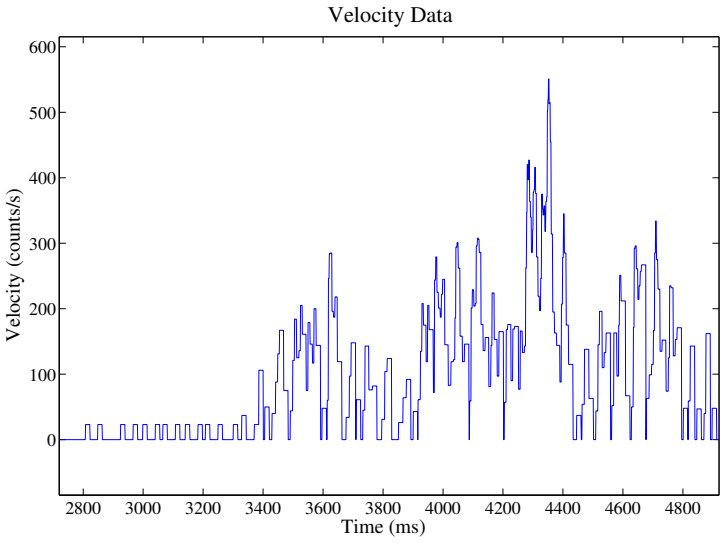


FIGURE 7.2: Scaled view of a Break-Away experiment showing stick-slip behaviour at low velocities.

In order to check for a dependence in position, we collect the torques which initiate motion, i.e. those measured exactly prior to motion being detected and plot these against position. This results in the following plots:

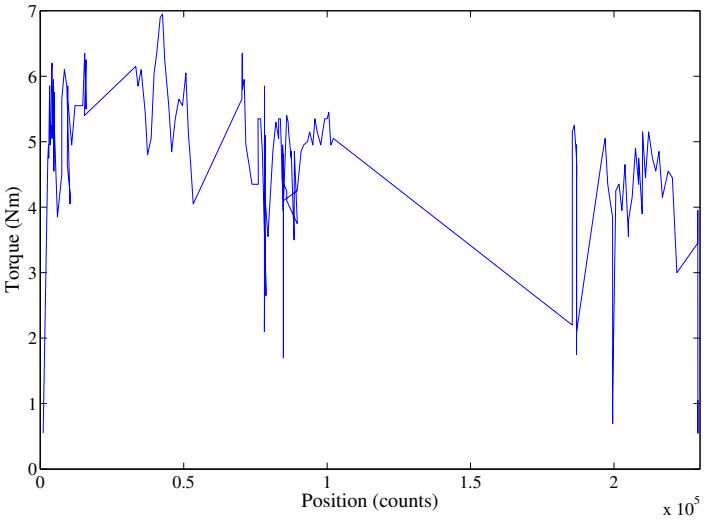


FIGURE 7.3: Stiction mapped against position for the duration of the Break-Away experiments.

There is a clear dependence on position in Figure 7.3, however we cannot directly attribute this to position dependent friction. Instead we can consider that the position dependence is included in the stiction force and equate the two for use in our mathematical model:

$$F_d(x, \dot{x}) = F_C + F_S(x) + b\dot{x} + \lambda \cdot f_s(\dot{x}, \dot{x}_s) \tag{7.1}$$

The primary challenged regarding the stiction mapping and the MACS is due to the rolling elements, which is why it is not feasible to consider a *fixed* configuration for the system and so also for stiction. Additionally, a region of about 50cm takes just over 12 hours to map, while the tank is 5m long in its entirety.

The resolution of the encoders in each axis effectively becomes 4000 counts per *motor* revolution. This means that after reduction, the smallest discernible distance from the encoder is of the order of less than ten microns (i.e. $\leq 10\,\mu m$). In fact the entire Y axis, form end-to-end travel of the carriage, the total distance equates to approximately $1,760,00.00$ counts.

Of course there is no reason to expect that we would require such a fine grain knowledge of the tank. This in all likelihood, may not be practically feasible at all. Nonetheless, testing for the position with a coarse, *cm* scale, accuracy could prove useful for attaining a similar increase of performance as reported by Helouvry in [25] and Chasparis in [28].

Finally, taking the average across the entire set of values in Figure 7.3, we produce an average estimate of the stiction torque:

$$\hat{\tau}_{S,y} = 3.7\text{e-}3 \quad N \cdot m \tag{7.2}$$

Square Wave Pulse Response

The second series of experiments involved the execution of a open-loop responses to Square Waveform Pulses (SWP). This excitation is used in place of step responses since in order to avoid response and actuator saturation and to ensure safe operation of the system. The experiments were repeated over several iterations and highly reproducible responses resulted.

The amplitude and period of the excitation were selected empirically as a result of trial and error. The final parameters used for the square wave pulse are presented in Table 7.4 and the excitation's time-series is shown in Figure 4.2. Even though the control value is the motor's output torque for an amplifier using CC, we consider the *Duty Cycle (%)* of the output PWM signal from the BeagleBone effective control value. The mapping of friction force to torque is estimated from the preliminary tests.

TABLE 7.3: Square Wave Pulse Excitation Parameters

Parameter	**Value**	**Units**
$Amplitude$	15	$(\%)Duty$
T_{On}	0.6	seconds
T_{off}	0.4	seconds

A range of amplitudes were used so to avoid the unwanted saturation and transient effects, as described in Chapter 4. The values which produced the best results were those in the $[10, 20](\%)$ range. The experiments were executed across a span of $2.68m$ in the middle of the tank where there is less lateral loading on the carriage. The resulting responses are presented in Figure 7.4.

In Figure 7.4(c), we can see the dynamical effects discussed in Chapter 3. Specifically, we observe that the dependence on position becomes evident on the pulse responses and especially during the start and end of each run. Also, the oscillations in the velocity response due to the vibrations of the chain are clearly recognizable. The interesting feature which these oscillations exhibit is that they are dissipated, converging onto the values of the nominal system (i.e. without the elasticity).

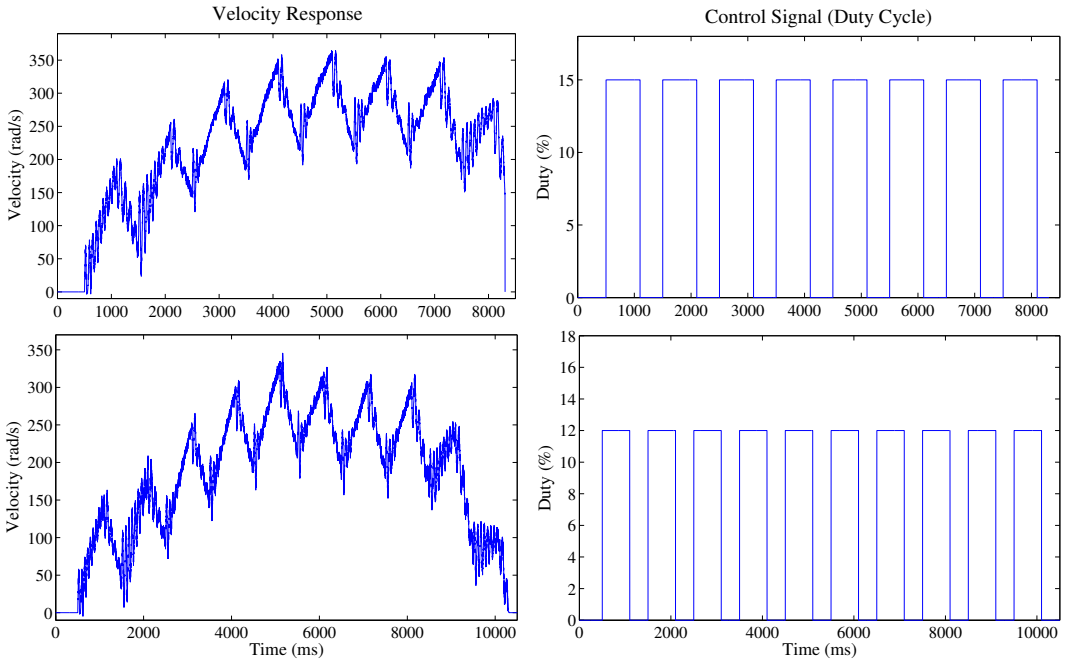


FIGURE 7.4: Three instances of Square Wave Pulse responses.

Thus we recognize the two-time scale behaviour of the singularly perturbed dynamics. The short time-scale is the faster transient oscillations which essentially creates oscillations, superimposed onto the slower long time-scale response of the nominal dynamics.

In fact, simulating the system in Matlab/Simulink using approximated parameters and arbitrarily adapted values for the spring-damper parameters results in the comparison shown in Figure 7.5. From this comparison we can see that the responses in velocity and acceleration fit closely to those of the experiments.

The problem with the higher order model which includes the spring-damper pair, is that it requires a direct measurement of the TC carriage's position in the task space in order to determine values for the K_s, B_s parameters. Thus we are currently unable to measure these parameters.

However, for the purposes of developing an initial model for the carriage dynamics we will equate this system as the nominal (effective) with additive oscillatory noise. We therefore use a Gaussian smoothing filter over the a selected instance of the SWP experiments and produce the filtered responses shown in Figures 7.6 and 7.7. Thus we consider this to be the effective system and apply the OLS procedure (see Chapter 4) to estimate the model parameters. The resulting values are presented in Table 7.4.

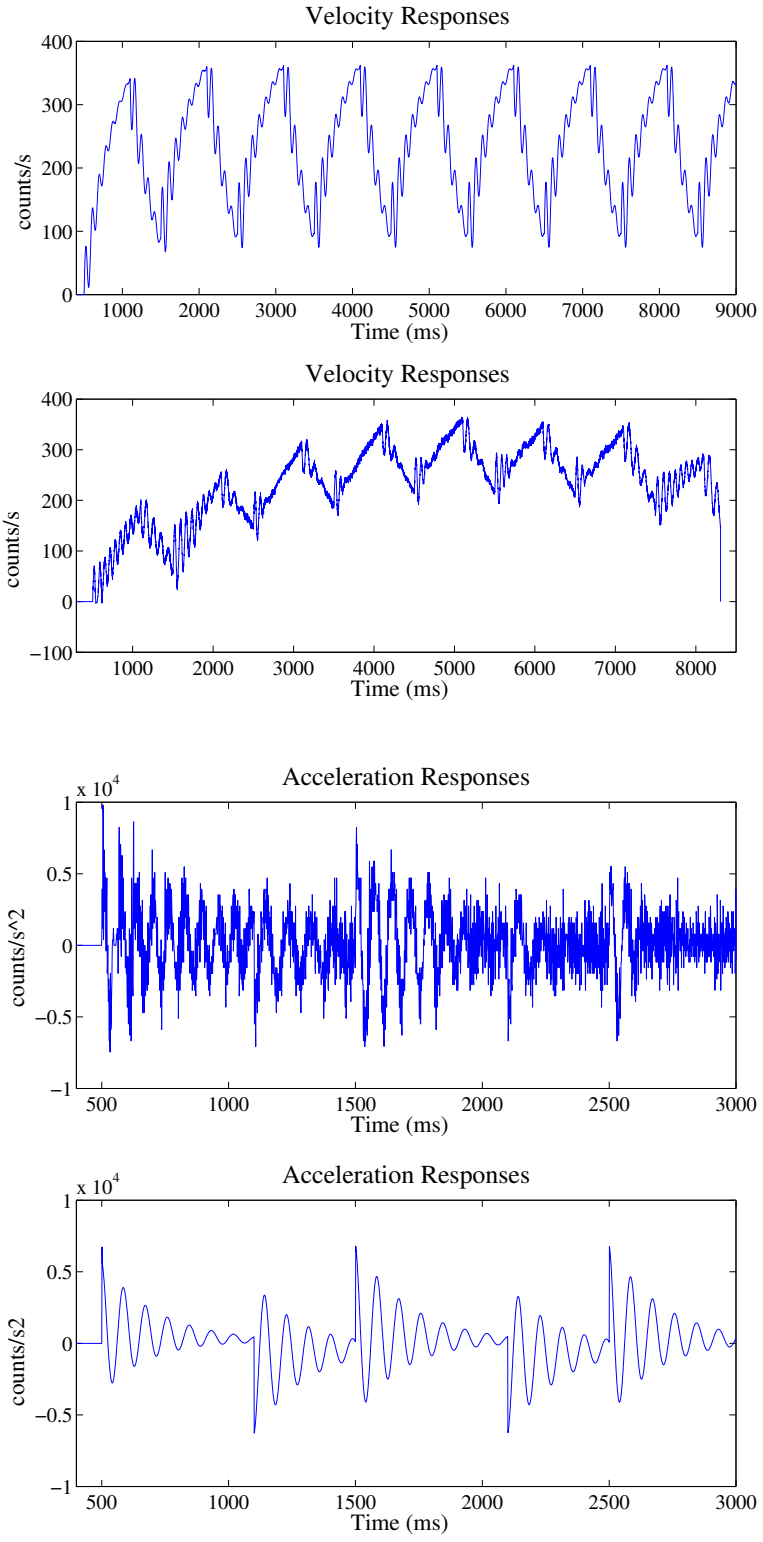


FIGURE 7.5: Comparison between the responses of the simulation (bottom) and experiment (top) showing the validity of the model with elasticity.

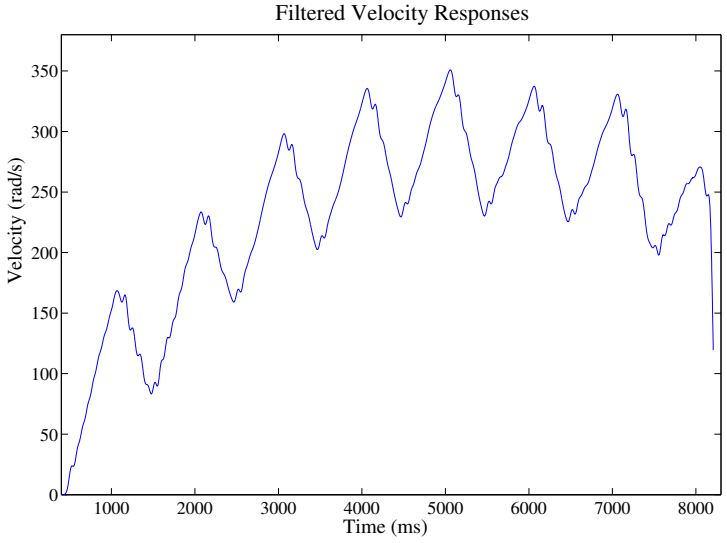


FIGURE 7.6: Filtered response in velocity.

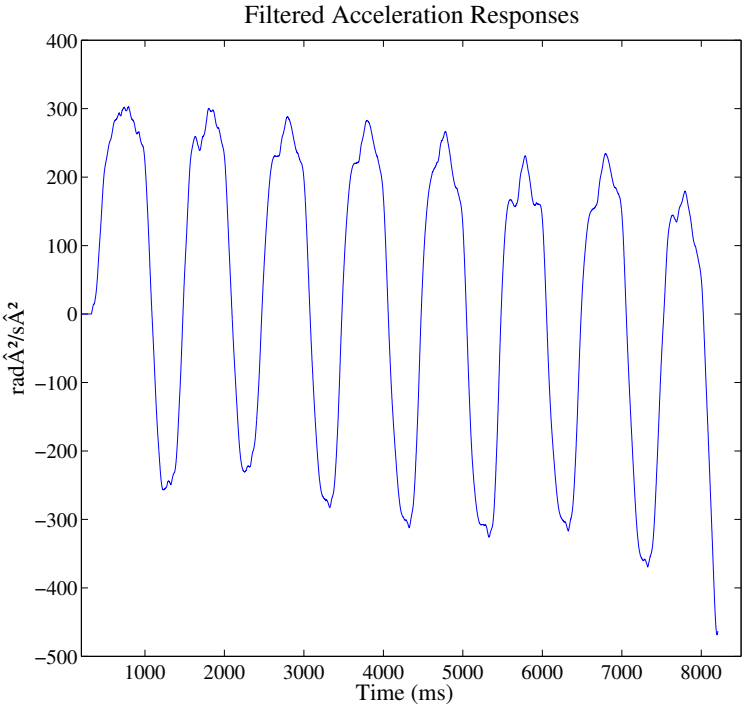


FIGURE 7.7: Filtered response in acceleration.

TABLE 7.4: Parameters Values for effective Y axis dynamics.

Parameter	**Value**	**Units**
$J_{\text{eff},y}$	1.714e-5	$Kg \cdot m^2$
$B_{\text{eff},y}$	1.106e-5	$N \cdot s$
$C_{\text{eff},y}$	2.26e-3	$N \cdot m$

Using the Simulink model shown in Figure 3.10 to simulate the dynamics using these parameters results in the comparison of velocity responses shown in Figures 7.8.

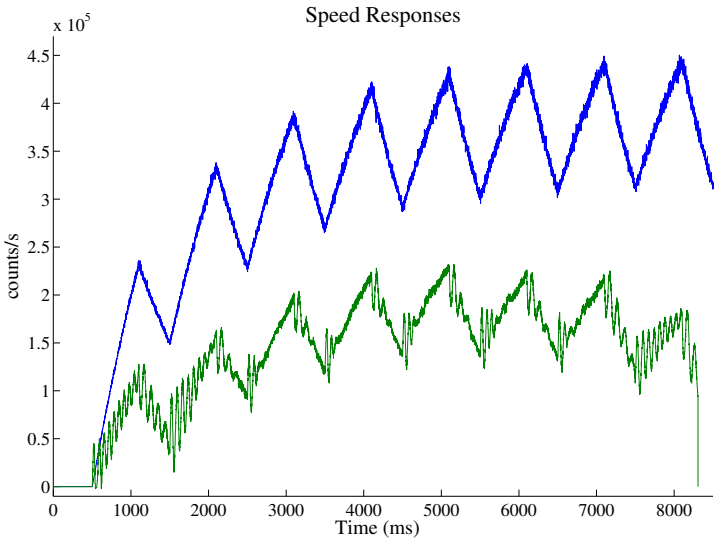


FIGURE 7.8: Comparison between the responses of the simulation (left) and experiment (right) using the identified model parameters.

There is a clear similarity in the responses even though they do deviate considerably. However we must consider that the model does not take into account the position dependent values of friction. This means that as the total loading fluctuates, and thus we cannot expect the value produced for the Coulomb component to be perfectly matched. Also, the OLS estimation algorithm is bound to result in the inaccuracies (deviations) of the estimated parameters. This is due to the fact that we filter out oscillations, which in part, removes the transient characteristics which would otherwise produce a better fit.

Thus we *manually adapt* the model parameters using the following empirical relationship:

$$\hat{J}_{\text{eff},y}\ddot{\theta} + \hat{B}_{\text{eff},y}\dot{\theta} = \alpha \cdot \frac{K_T K_{amp} V_{CC}}{100} D - \beta \cdot \hat{C}_{\text{eff},y}\,\text{sgn}(\dot{\theta}) \tag{7.3}$$

$$\Longrightarrow \frac{J_{\text{eff},y}}{\alpha} \cdot \ddot{\theta} + \frac{B_{\text{eff},y}}{\alpha} \cdot \dot{\theta} + \frac{\beta C_{\text{eff},y}}{\alpha} \cdot \text{sgn}(\dot{\theta}) = \frac{K_T K_{amp} V_{CC}}{100} D \tag{7.4}$$

Where D is the duty cycle control values used by the EMCP. The adapted parameters are define as:

$$\hat{J}_{\text{eff},y} := \frac{J_{\text{eff},y}}{\alpha}, \quad \hat{B}_{\text{eff},y} := \frac{B_{\text{eff},y}}{\alpha}, \quad \hat{C}_{\text{eff},y} := \frac{\beta C_{\text{eff},y}}{\alpha} \tag{7.5}$$

This relationship constitutes an *intuitive guess*. The reasoning is based on the aforementioned issues with the OLS deviations. We select the following values:

$$\alpha = 0.8 \tag{7.6}$$

$$\beta = 1.8 \tag{7.7}$$

The adapted parameter values are presented in Table 7.5 and the new comparison is presented in Figures 7.9 and 7.10.

TABLE 7.5: Adapted Parameters Values for effective Y axis dynamics.

Parameter	**Value**	**Units**
$\hat{J}_{\text{eff},y}$	2.1425e-5	$Kg \cdot m^2$
$\hat{B}_{\text{eff},y}$	1.3825e-5	$N \cdot s$
$\hat{C}_{\text{eff},y}$	5.085e-3	$N \cdot m$

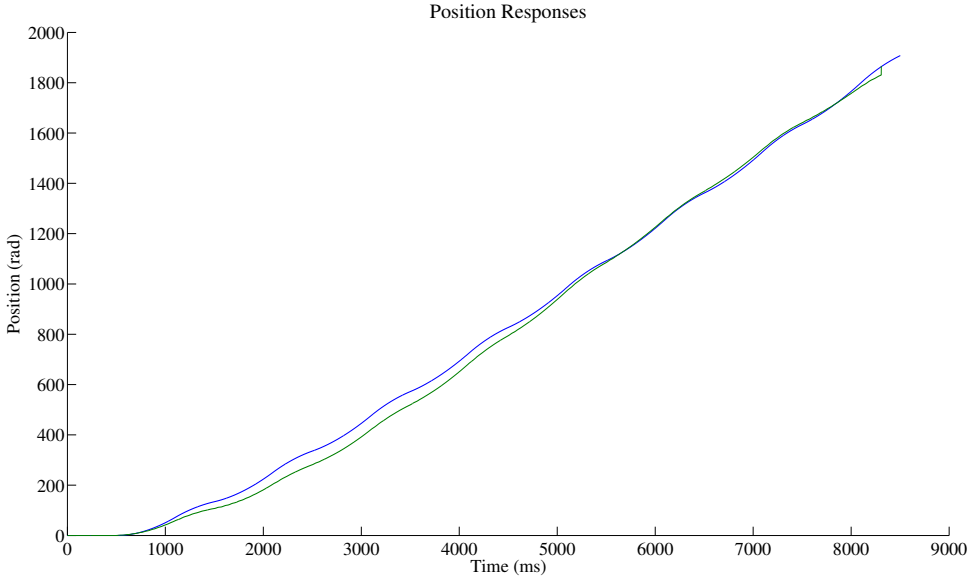


FIGURE 7.9: Comparison between the position responses of the simulation (left) and experiment (right) using the adapted model parameters.

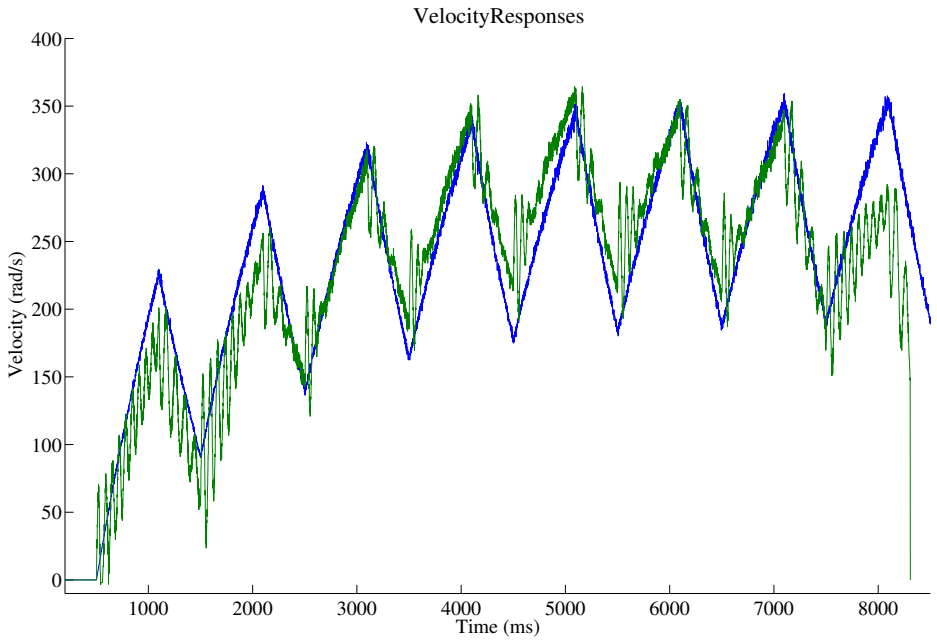


FIGURE 7.10: Comparison between the velocity responses of the simulation (blue) and experiment (green) using the adapted model parameters.

This final comparison shows that the adapted parameters produce a close fit to the experimental responses.

7.2.2 Closed Loop Experiments

Closed loop experiments were conducted in order to understand how the carriage interacts with the environment. Using a closed-loop controller to regulate motion enables us to experiment in ranges of operation that would otherwise be inaccessible by open loop commands.

What we accomplish is essentially the inversion of the open-loop assumption. Where in the case of the former, we assume a known excitation and measure the state, now we can assume a known state and measure the input.

The controller used to conduct these experiments is the one which is described later in this chapter (see section 7.3.1). We thusly present the results of the two closed-loop identification experiments.

Friction Torque-Velocity Mapping

With the execution of this experiment we aim to directly measure the intrinsic loading of the MACS due to the loading of the TC by the water tank, as well as to isolate the frictional effects. Specifically we aim to test the following dependencies:

1. Test for position dependence during motion.
2. Test for Velocity Dependence.
3. Compare results against open-loop experiments.

The first dependency will verify whether our assumption of TC lateral loading due to deformation of the water tank. In contrast to the breakaway experiments where a force is measured prior to motion, these experiments consider position dependent loading while in motion. The second goal helps with validation of the Stribeck curve described in Chapter 3. The third goal is trivial as we can then determine weather the system can in fact be described effectively using the model in Eq. (4.1)-(4.3).

In this set of experiments we execute trapezoidal velocity profiles and measure the velocities and control torques during steady-state. Under this condition we can assume that the inertial reaction force is negligible. Thus, we can assume that the dynamics of the system are described by:

$$\tau_{d,y}(\theta_y, \dot{\theta}_y) = \tau_{m,y} \tag{7.8}$$

The symmetrical trapezoids covered a range of steady-state velocities in $[10, 40]\,{}^{cm}/_{s}$ using $10\,{}^{cm}/_{s}$ steps as well as the range $[1, 10]\,{}^{cm}/_{s}$ using $1\,{}^{cm}/_{s}$ steps. The acceleration time was configured as $0.8\,s$ and the maximum traversable distance of $3.98\,m$ along the Y axis was covered. Figure 7.11 presents the velocity responses across all experiments:

Some oscillation around the steady state values are evident but there is no significant deviation. To prepare the data, we use Gaussian smoothing filters remove high frequency variations and then isolate the steady state values and average temporally. This is done for both torque and velocity data. The result is presented in Figure 7.12.

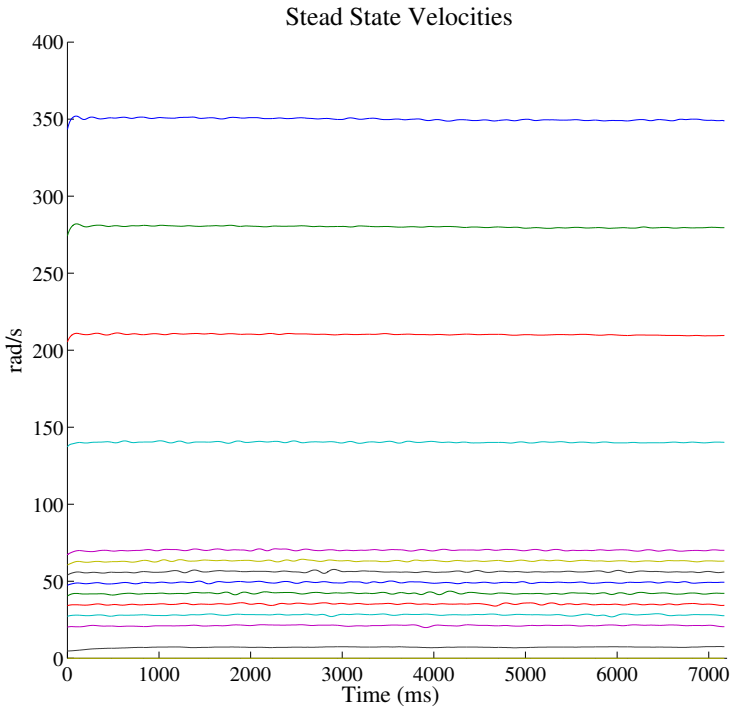


FIGURE 7.11: Velocity responses used for the Torque-Velocity mapping.

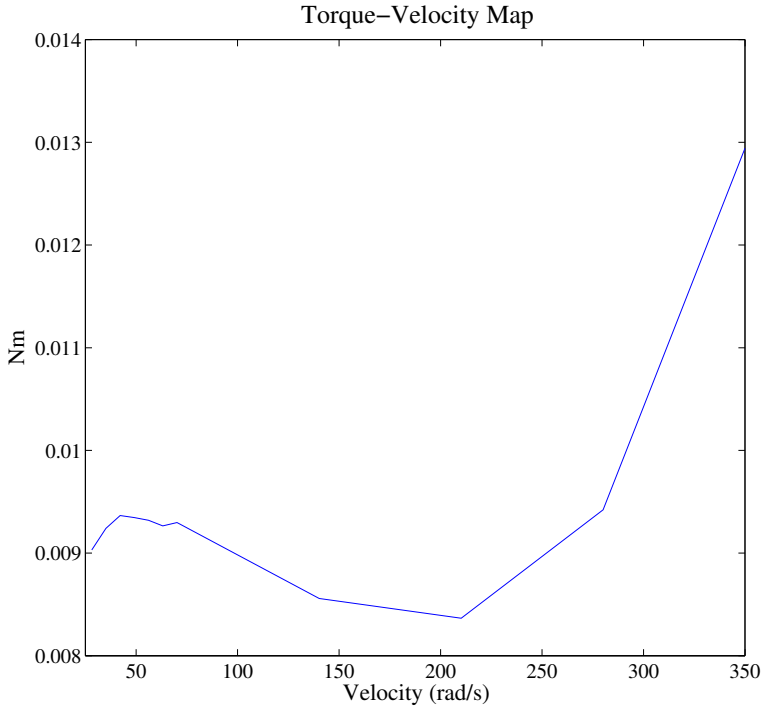


FIGURE 7.12: Torque-Velocity mapping.

From Figure 7.12 it is clear that our assumption about the torque-velocity relationship is invalidated. This is clearly non-linear behaviour which can only be handled ad

hoc, i.e. we would have to characterize this using advanced fitting but is left for future work. We must consider though that this result may be misleading to some extent.

The relationship portrayed in the previous figure is expressed only in terms of velocity at different operating points. We now map the torque signals against the measured position responses and this is shown in Figure 7.13.

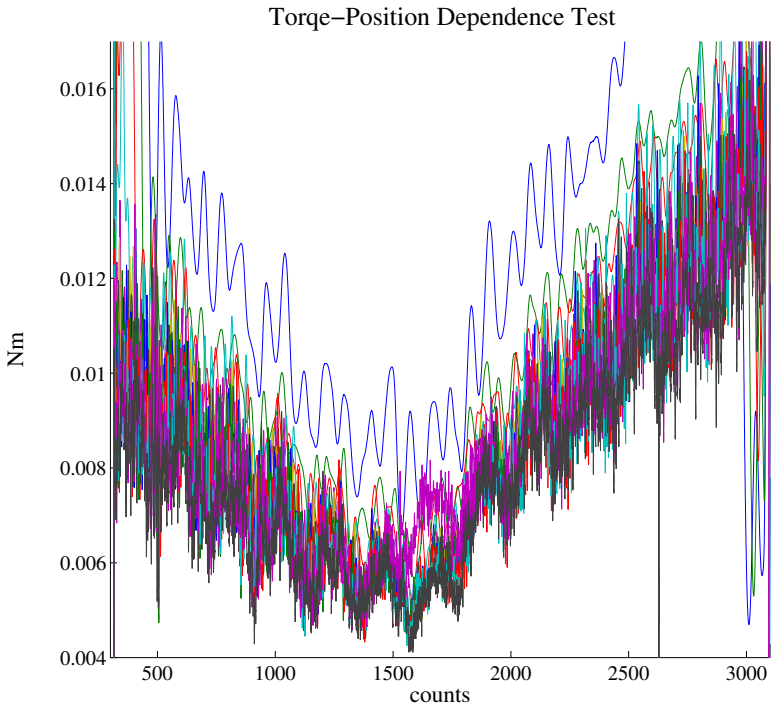


FIGURE 7.13: Torques at steady-state velocity when mapped against position.

It is clear that there is an seemingly linear and piecewise dependence of position. Also interesting in this plot is that the relationship to position is almost the same across al experiments, which implies that the dependence does not change in when different velocities are used.

7.3 Trajectory Benchmarks

7.3.1 Trapezoidal Profiles

We have already referred to the use of the trapezoidal profile extensively throughout this chapter. To determine whether our assumption of known state during tracking is valid

we must check to see if in fact the controller can maintain low tracking error. Presented below in Table 7.6 are the average tracking errors while executing the torque-velocity mapping experiments.

TABLE 7.6: Trapezoid experiment - approximated task-space trajectories and errors.

Steady-state Velocity $[mm/s]$	**Mean Velocity Error** $[mm/s]$	**Mean Position Error** $[mm]$
5	−2.873e-4	1.226e-3
10	−3.783e-4	1.391e-3
20	−4.269e-4	1.195e-3
30	−4.334e-4	1.153e-3
40	−2.669e-4	1.170e-3
50	−2.668e-4	1.63e-3
60	−2.611e-4	1.194e-3
70	$-2,516$e-4	1.195e-3
80	−2.224e-4	1.117e-3
90	−2.086e-4	1.116e-3
100	−2.456e-4	0.988e-3
200	−3.022e-4	0.919e-3
300	−6.327e-4	1.093e-3
400	−1.242e-4	1.645e-3

The controller has thus proven to be able to maintain even sub-millimeter accuracy in position tracking and substantially low-error velocity tracking within less than a mm/s. These are sufficient to verify our assumption that during steady-state, the trajectory can be assumed to be known and constant in velocity. This enables us to conduct experiments using a sample STO in the next chapter.

7.3.2 Sinusoidal Profiles

Sinusoidal profiles enable us to gauge the efficiency of the controller while remaining in a transient state at all times. Sinusoids can produce responses which are steady in the average and over large periods of time while also exciting transient the transient characteristics of the system. Therefore, tracking a relatively slow sine wave reference is indicative of the capabilities of the controller.

To test the sinusoidal responses we executed a circular trajectory in the task space with a radius of $20cm$ and a period of 7 seconds. The responses are presented in Figure 7.14 and the tracking errors in 7.15. From these plots we can observe the following characteristics exhibited by the transient responses:

1. The tracking error is largest during the sinusoid position peaks (i.e. maximum accelerations and decelerations).
2. The tracking error is large during changes in direction of motion due to stick-slip due to insufficient torque at low reference velocities.
3. The Y axis error is larger than that of the X axis due to the longer chain.

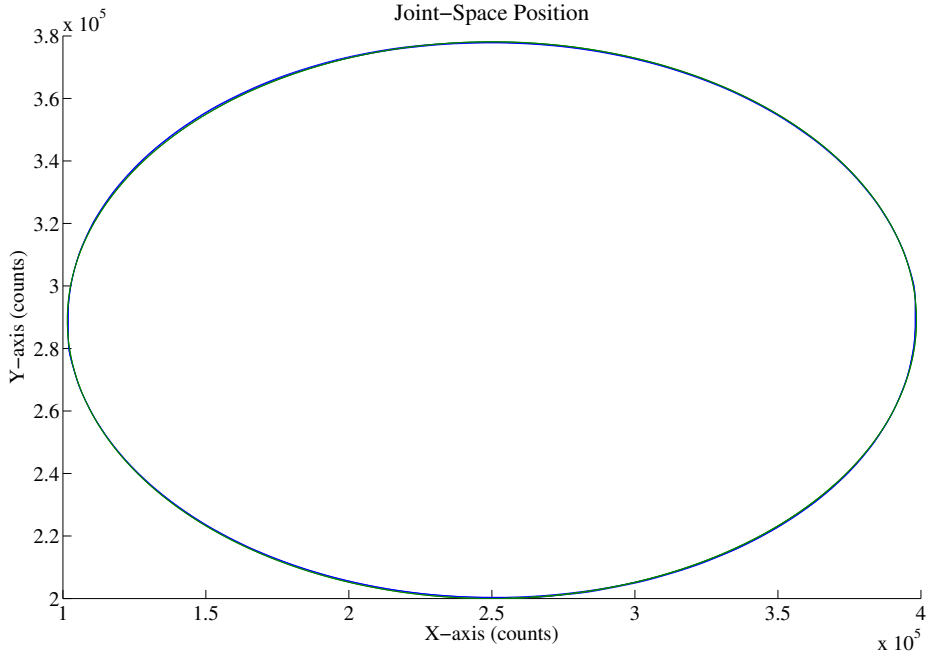


FIGURE 7.14: Circle trajectory of the EEP. The joint space position tracking is depicted.

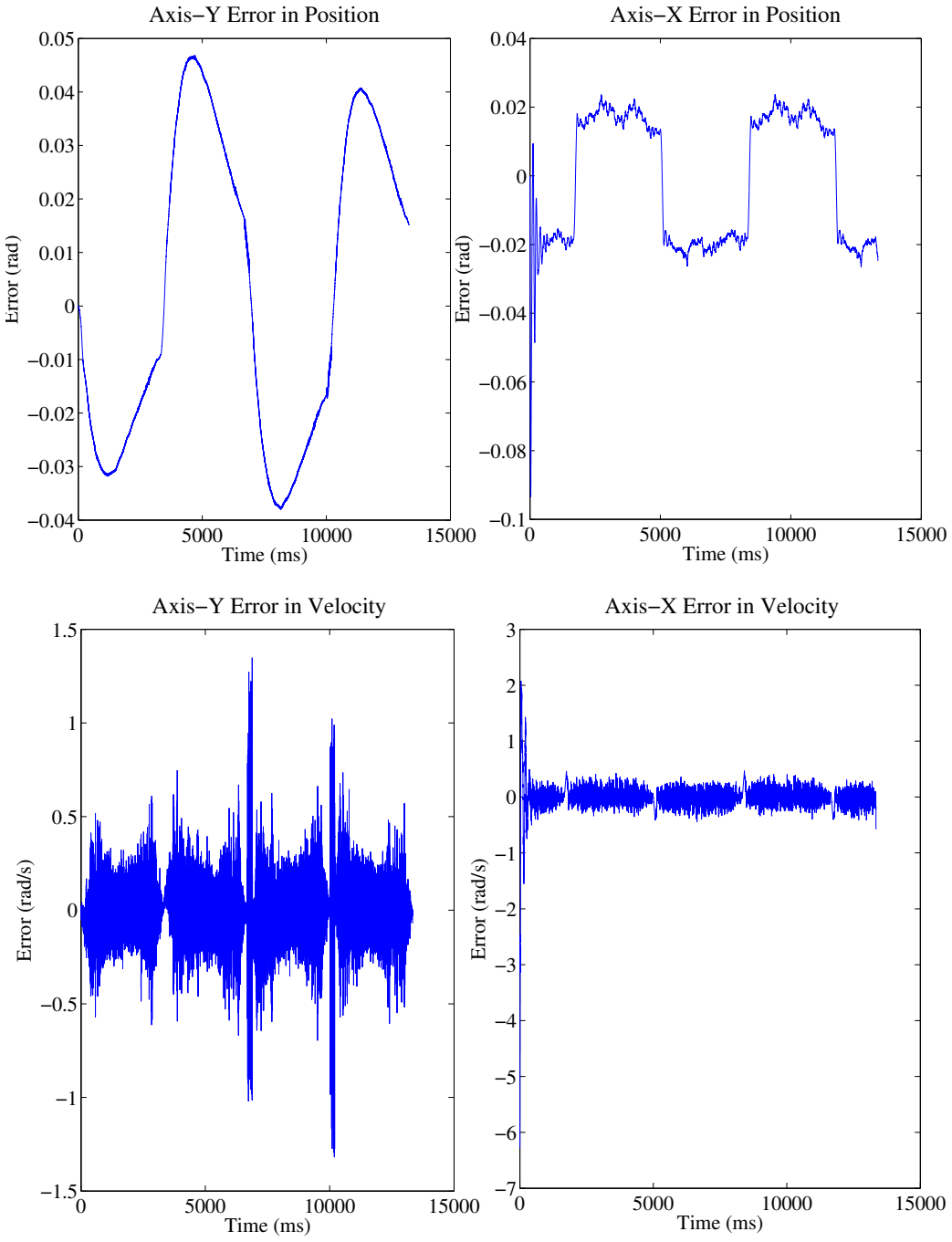


FIGURE 7.15: Tracking errors for a circle trajectory: position (top) and velocity (bottom) for axes Y (left) and X (right).

Chapter 8

Case Study: Drag Identification of Underwater Robots

In this chapter we present an application[1] case of the URETTS platform. It primarily involves what was discussed in Sections 1.2 and 3.1 of this thesis regarding modeling and control of underwater vehicles. Specifically, a new approach to estimating hydrodynamic drag forces is proposed that does not use a force sensor. The developed method involves processing the system responses and motor current measurements from the MACS. The sensor-less results are experimentally compared to measurements produced by a 6-DOF force/torque sensor.

8.1 Motivation

In this chapter we present a novel method for measuring drag forces through the processing of system response and motor current signals for each. This entails the use of several experiments, substantially different between them to cross-compare their respective results. If they produce similar results, these are interpreted to be valid measures of the drag forces.

The first experiment involves executing a particular manoeuvre twice, then processing the motor current signals. The second utilizes a model of the servomechanism

[1]The work described in this chapter was presented at the IEEE AIM-20214 conference in Besancon France in July, 2014, as an accepted paper submission, [52]

to predict the presence of drag needing only a single execution of any manoeuvre. This second experiment is cumbersome due to the need for executing identification experiments beforehand but require a single experiment. Provided that a model of the system is available, the same data set may then be used in together with SI techniques, when online real-time estimates may also be desired. To test the results of the experiments, their results are compared directly to the measurements of a force sensor.

8.2 System Modelling and Control

Complete analyses of the sub-system dynamics were presented in Chapter 3 and simple controllers for regulating the motion of the TC were described in Chapter 3. For the purpose of clarity, the descriptions are summarized again here. In this way the process of developing assumptions about the system and then interpreting the experimental results from this scope becomes clear.

However, the system model equations developed in the previous chapters are more general. The drag estimation experiments described below constitute a proof-of-concept stage in the development of the URETTS platform. For these reasons, the models and controller utilized are simplified so to capture the most dominant contributions to the dynamical behaviour of the overall system.

8.2.1 Load Unit Dynamics

The purpose of this work is identifying drag forces acting on submerged bodies to enable future research on developing model based controllers for robotic fish. The model assumed for these forces is that of (3.7), where the viscous drag is proportional to the square of velocity. Even though this equation applies for a body of any geometry, we consider a simplified set-up using an object of simple geometry to conduct experiments. A plastic cylinder of a fixed radius and height was used as the STO. The equation for viscous drag is adapted here in the following form:

$$F_D = D \cdot {v_y}^2 \cdot \mathrm{sgn}(v_y)\,, \quad D := \frac{1}{2} A \rho C_D \tag{8.1}$$

Where F_D is the external hydrodynamic force exerted on the STO, υ_y is its linear velocity along the Y axis of the tank and D is the effective coefficient of viscous drag which is summarizes the parameters in viscous drag. The motion only occurs along the longer Y axis of the tank (and so of the MACS) in order to take advantage of the greater total distance and produce larger data sets.

8.2.2 Carriage Dynamics

As described in Chapter 3 regarding assumption [MACS-A-6], we do not consider the coupling forces between axes of motion, as these are essentially negligible, enabling the modeling and control of each axis to be independent. Also, since the system exhibits oscillations we use the effective system dynamics, i.e. the parameters of the underlying slow-time scale to describe dynamics of the MACS (see Chapter 3 for details).

The dynamics of the carriage have been reduced to a simpler form which does not consider position dependent friction and also disregards the low velocity dynamics of friction. This is justified by the fact that the towing experiments are executed within the $[10, 50]\,{}^{cm}/_{s}$ range, which is well above the characteristic Stribeck velocity. We therefore we consider the following model for the slow time-scale dynamics:

$$J_{\text{eff},y}\ddot{\theta}_y + B_{\text{eff},y}\dot{\theta}_y + C_{\text{eff},,y} \cdot \text{sgn}(\dot{\theta}) = \tau_{m,y} \tag{8.2}$$

The above equations use the *effective* system parameters, expressed on the motor side. $J_{\text{eff},y}$ $(Kg \cdot m^2)$ is the effective inertia, $B_{\text{eff},y}$ (Nms) is the effective viscous friction coefficient and $C_{\text{eff},y}$ (Nm) is the effective kinetic (Coulomb) friction. Also, $\tau_{m,y}$ is the motor output torque. It is also important to mention again that current control is used to drive the motors. This then allows for direct force control on the motor and the input-output relationship is describe again here:

$$\tau_{m,y} = K_{T,y} K_{\text{amp},y} \nu_{c,y} \tag{8.3}$$

$K_{T,y}$ is the motor torque constant and $K_{\text{amp},y}$ is the current control gain of the motor driver amplifier. The model parameters are summarized in Table 8.1.

TABLE 8.1: Model Parameters for the Y axis Dynamics.

Quantity	**Symbol**	**Units**
Effective Drag Coefficient	D	$\frac{N \cdot s^2}{m^2}$
Effective Axis Inertia	$J_{\text{eff},y}$	$Kg \cdot m^2$
Viscous Friction Coefficient	$B_{\text{eff},y}$	$N \cdot s$
Coulomb Friction	$C_{\text{eff},y}$	$N \cdot m$
Gearbox Ratio	n_y	-
Sprocket Radius	R_y	m
Motor Torque	$\tau_{m,y}$	$N \cdot m$
Motor Torque Constant	$K_{T,y}$	$\frac{N \cdot m}{A}$
Amplifier Gain	$K_{\text{amp},y}$	$\frac{A}{V}$

8.2.3 Model Parameter Identification & Controller Design

Since (8.2) is linear with respect to the model parameters, a LLS technique is adopted for their identification. The reader is referred to Chapter 4 for further details concerning these methods as well as [20]. Specifically, just like in Chapter 4, the OLS algorithm is used. The identification experiments were executed similar to those in Chapter 7, using square-pulse waveform excitations of 60% duty cycle and a period of 1 sec. The resulting values from these identification experiments are presented in Table 8.2.

TABLE 8.2: Parameters Values for Y axis system.

Parameter	**Value**	**Units**
$J_{\text{eff},y}$	1.502e-5	$Kg \cdot m^2$
$B_{\text{eff},y}$	4.657e-6	$N \cdot s$
$C_{\text{eff},y}$	3.9e-3	$N \cdot m$
n_y	34.97	-
R_y	5.5	m
$K_{T,y}$	26.7e-3	$\frac{N \cdot m}{A}$
$K_{\text{amp},y}$	0.352	$\frac{A}{V}$

The controller used in the drag estimation experiments is the simple linear PID described in Chapters 4 and presented experimentally in Chapter 7. Selection of appropriate controller gains was initially done using the Ziegler-Nichols gain tuning method [64], and then further tuned empirically. This resulted in stiff velocity control which reduced the tracking error, achieving millimeter accuracy in tracking tasks. The values of the PID controller gains are those of the previous chapter, presented in Table 7.2

8.3 Experiment Design

Three methods are proposed and compared for measuring measure the acting hydrodynamic forces. The first is a direct measurement using the 6-DoF ATI force/torque sensor, while the other two constitute the estimation methods executed over the response data of the same experiment. The experiments consists of precisely repeated trapezoidal speed profiles along the y axis. In each instance, data is collected while the carriage tows the cylindrical STO.

Let $f(\theta)$ be the function representing the total loading torque on the y axis motor without the presence of hydrodynamic forces. If this system is in steady-state maintaining a constant velocity, this becomes $f_{ss}(\theta_{ss})$ and when drag forces are present, the torque balance equation becomes:

$$f_{ss}(\theta_{ss,y}) = \frac{R_y}{n_y} \cdot F_{D,y} + K_{T,y} K_{\text{amp},y} \nu_c \tag{8.4}$$

The proposed drag estimation methods are described as follows:

Method 1: *Using a Force Sensor.*

The object, whose drag force is to be estimated, is mechanically coupled to the towing carriage, with the force sensor placed exactly in between. The drag force is then effectively the force F_y, measured by the force sensor along the long tank axis during the steady-state phase of the trapezoidal profile executed by the controller. These Force Sensor Method (FSM) measurements however are conducted during all iterations of the experiments, including those of the estimation methods, so as to provide a benchmark for their performance.

Method 2: *Two-Step Motor Current Measurements.*

Assuming no previous knowledge of the plant model or system parameters, a PID control law can be applied. If the gains are appropriately chosen in order to ensure stiff trajectory tracking, then executing any same procedure twice, the first with the STO mounted below the EEP and in the water (input), while in the second, mounted on top of the EEP (input). By then taking the difference between (8.4) from each experiment, i.e. the Motor Current Difference (MCD), this should yield the drag force signal:

$$F_{D,y} = \frac{n_y}{R_y} K_{T,y} K_{\text{amp},y} (\nu_{c,1} - \nu_{c,2}) \tag{8.5}$$

Method 3: *Single-Step Model Based Estimation.*

This is the Model Based Drag Prediction (MBDP) method and considers the use of (4.8) and the model in (8.2) in place of $f(\theta)$ for the y axis dynamics. Using the parameters of Table 8.2, we proceed to predict the total forces produced only by the carriage system, which are then subtracted from the known input control torque. Even though this signal is also affected by other hydrodynamic forces during transient motion, these are negligible during steady-state. In this case, the drag force can be estimated through the following calculation:

$$F_{D,y} = \frac{n_y}{R_y} \left(K_{T,y} K_{\text{amp},y} \nu_c - J_{\text{eff},y} \ddot{\theta_y} - B_{\text{eff},y} \dot{\theta}_y - C_{\text{eff},,y} \cdot \text{sgn}(\dot{\theta}) \right) \tag{8.6}$$

8.4 Results

8.4.1 Experiment Execution

Following the methods described in the previous section, a set of trapezoidal speed profiles were executed along the Y axis of the tank using the MACS. The X axis was controlled so to remain centered in the middle of the tank's width. The symmetrical trapezoid profiles covered distances of up to $2.2m$, at steady-state speeds between $5^{cm}/_{s}$ and $55^{cm}/_{s}$ with $10^{cm}/_{s}$ increments. Acceleration to the target velocity was set to within less of a second and the profiles were modified appropriately so as to avoid excitation of the oscillation modes.

The tracking errors of the desired trajectories (position and velocity) are shown in Table 8.3, for both cases, i.e. with and without drag. It is clear that the controller achieves the desired trajectory with a sufficient degree of repeatability and accuracy. The errors are calculated from the raw values of the encoder i.e. they have not been filtered to remove noise and/or other disturbances.

TABLE 8.3: Trapezoid Experiment Trajectories.

Steady-state Velocity $[mm/s]$	**Mean Position Error** $[mm]$	**Mean Velocity Error** $[mm/s]$
50	3.36e-1	1.69e-2
150	5.81e-1	6.9e-2
250	8.17e-1	9.87e-2
350	1.8	3.8e-2
450	1.9	1.63e-2
550	3.9	2.31e-2

8.4.2 Method Comparison

Having assured that responses are are close enough to the commanded references, we present the results of using the sensor-less drag estimation methods and compare their performance to that of the force sensor. These are presented in Table 8.4 and the identified drag force curve, plotted against velocity are shown in Figure 8.2.

From Figure 8.2 it is clear that the method using differences in motor current produces results significantly closer to those by the FSM than to those by the MBDP method. The coefficients of drag which were identified via LS are also shown and compared in Table 8.4 with the results of the fitting.

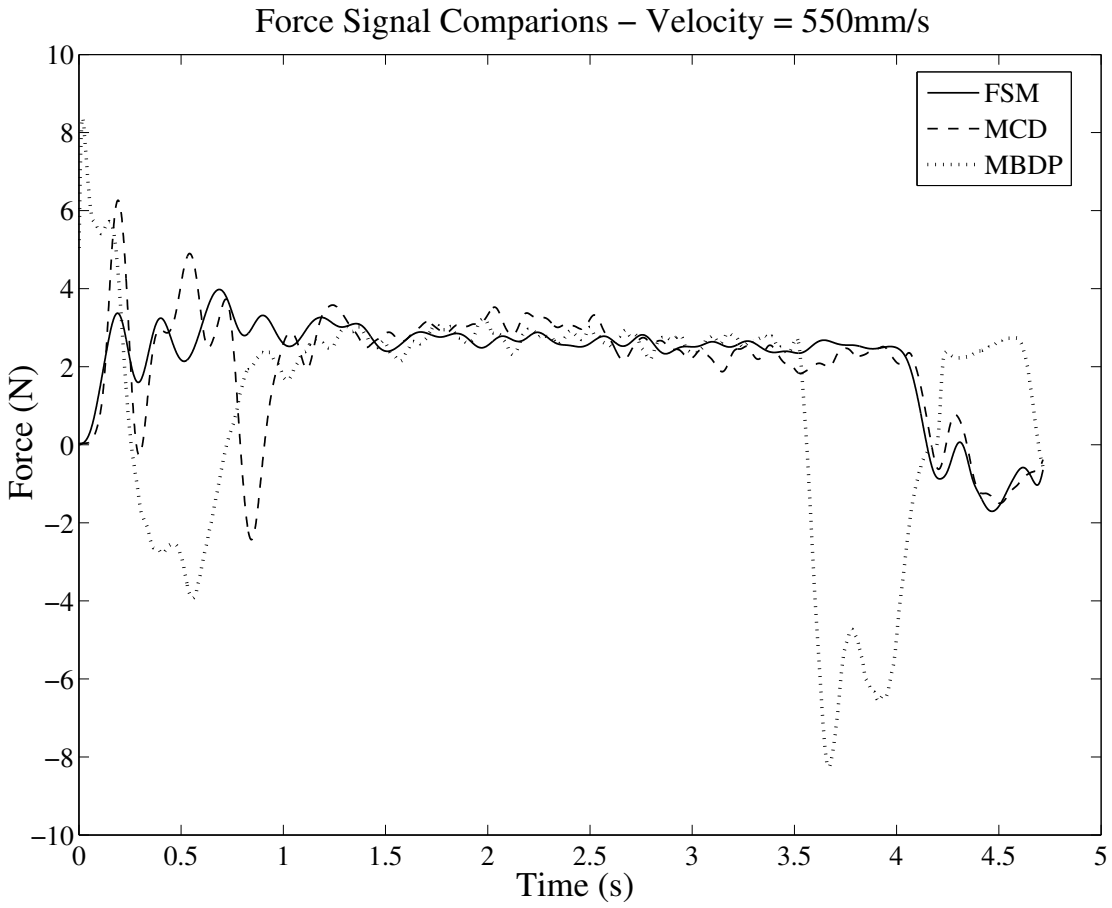


FIGURE 8.1: Measurements of the force sensor during trapezoidal motion.

TABLE 8.4: Drag Coefficient Comparison $[N \cdot s^2/m^2]$.

ATI Force Sensor (FSM)	**Motor Current Difference (MCD)**	**Model Based Drag Prediction (MBDP)**
8.78e+2	8.39e+2	7.97e+2

A force comparable to that of the FSM is detected in both estimation methods. However, the estimations deviated from the sensor measurements during the acceleration and deceleration phases. This was expected since transient responses are also affected by additional hydrodynamic effects such as added mass.

Furthermore, during the course of conducting identification experiments, it was observed (see Figure 8.1) that the loading forces increased as the carriage traversed the middle of the tank. This dependence on position was not accounted for in the system

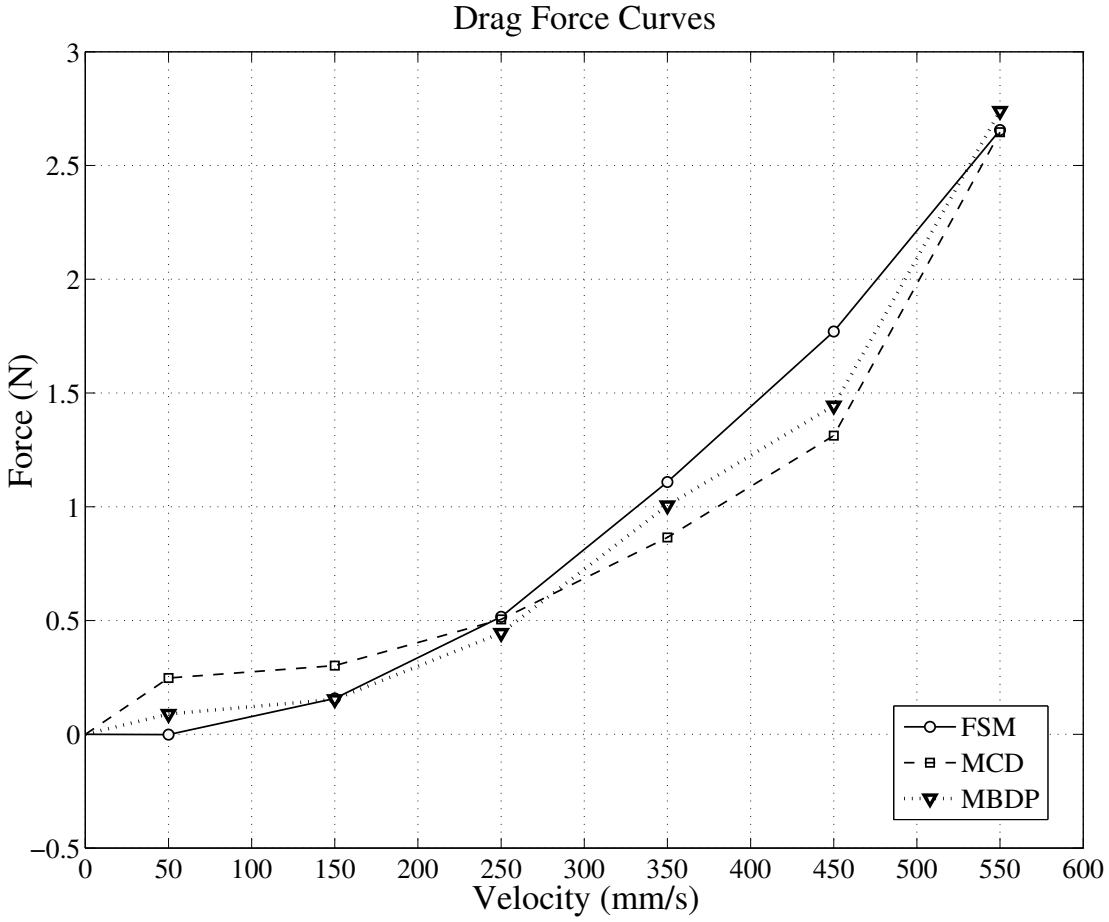


FIGURE 8.2: Measurements of the force sensor during trapezoidal.

model but justifies its inclusion in future research. This significantly contributed to the deviation of the model based approach from the results of the other methods.

Part V

Post-Mortem

Chapter 9

Conclusions and Future Work

This chapter summarizes the results of this thesis and presents new directions for continuing the current work. This stage of research involved the development and testing of the URETTS platform. To aid future research and development activities, we outline the directions in which the platform should be extended. This is with respect to both improvements to the exiting experimental system, in terms of hardware and software as well as future directions of research regarding its utilization in underwater robotics.

First we discuss the results of the work, commenting on the operation and functionality of the URETTS but also examine the experimental results presented in Chapters 6 and 7. The second and final section of this chapter outlines the recommended platform extensions and proposes future research prospects.

9.1 Thesis Conclusion

This thesis has demonstrated the development, testing and use of the URETTS platform for experiments involved in modeling of hydrodynamic forces. The platform in its current form serves mainly as a *proof-of-concept* breadboard for the development of experimental towing carriage systems. In order to determine the efficacy of the platform and therefore the outcome of this thesis, we must check whether the specified requirements have been met and asses the performance of the system in the various aspects of the intended application.

9.1.1 System Requirements Review

9.1.1.1 Functional Requirements

The functional requirements are met in the majority. All major functional requirements specified for the capabilities of the platform have been met or exceeded with the exception of the addition of a rotational DoF on the EEP. This enhancement to the capabilities of the system was not achieved due to time and functional constraints. The results of the requirements review is summarized in Table 9.1.

TABLE 9.1: URETTS Functional Requirements Review

Requirement	**Achieved** [Yes/No/Partially]
FR-1	Yes
FR-2	Yes
FR-3	Yes
FR-4	Yes
FR-5	Partially
FR-6	Yes
FR-7	Yes
FR-8	Yes

As the experiments were successfully executed without this capability we can determine that this does not significantly reduce the performance of the system. Therefore the inclusion of an additional rotational axis is left for a future iteration.

9.1.1.2 Operational Requirements

The operational requirements are summarized in Tables 9.2 and 9.3. From these we can determine that the implemented system meets in the most part the specified requirements for the EMCP and BBMC components. There are however certain aspects which must be addressed.

Firstly, the final control signal reaching the Maxon ADS motor amplifiers is not a truly DC variable signal. Thus, the final SNR of the amplifiers output PWM is not nominal such as when a true DC signal is applied at the input. This has not been observed to induce a measurable difference on the final output torque of the motor, but improving this SNR could possibly further improve both the resolution and the variance of the applied motor torque. This means that future experiments involving sensorless force estimation could benefit from the improved certainty in the applied torque output.

Secondly, the prototype of the Hall-sensor end-stops for the safety sub-system of the EMCP, was not completed. A single end-stop was used near the origin of the global frame for the *minimum* position of the TC. The overall design, required that a differential line *driver* be used to transmit the Hall-sensor output signals over the entire distance of the tank. Due once more to time constraints the system was not further developed and is left for the next iteration of the EMCP.

Lastly, due to the improper selection of the planetary gear heads and subsequent incompatibility of the selected SITOP, we cannot consider the use of external PSs as part of the URETTS design. Thus the requirements pertaining to the integration of an independent multi-channel PS are not met in full. This also is left for the next iteration of the URETTS.

Table 9.2: URETTS Operational Requirements Review - EMCP

Component	**Requirement**	**Specification**	**Achieved** [Yes/No/Partially]
EMCP	**EMCP-FR-1.**	a.	Yes
		b.	Yes
		c.	Yes
		d.	Yes
		e.	Yes
		f.	Yes
		g.	Yes
	EMCP-FR-2.	a.	Yes
		b.	Yes
		c.	Yes
		d.	Partially
		e.	Yes
	EMCP-FR-3.	a.	Yes
		b.	Yes
		c.	Yes
		d.	Yes
		e.	No
	EMCP-FR-4.	a.	Partially
		b.	Yes
		c.	Yes
		d.	Yes
		e.	Yes

TABLE 9.3: URETTS Operational Requirements Review - BBMC

Component	**Requirement**	**Specification**	**Achieved** [Yes/No/Partially]
BBMC	**BBMC-FR-1.**	a.	Yes
		b.	Yes
		c.	Yes
		d.	Yes
	BBMC-FR-2.	a.	Yes
		b.	Yes
		c.	Yes
		d.	Yes
		e.	Yes
		f.	Yes
		g.	Partially
	BBMC-FR-3.	a.	Yes
		b.	Yes
		c.	Yes
		d.	Yes
		e.	Partially

9.1.2 System Application

The results presented in Chapters 6 and 7 provide a sound basis for a successful outcome of both the design and the implementation of the URETTS platform. This conclusion is also supported by the fact that the system exceeded initial expectations with the achievement of experimental verification of sensorless force estimation.

Considering the overall cost achieved, the URETTS's system design and organization proves to be a promising concept. The design's high degree of modularity permits a level of interchangeability in the sub-system components which further enables us to

improve or change the components without significantly altering the system's operation (if at all).

Moreover, proving that the design can in fact be implemented with mid to low end components indicates that it can be maintainable over large periods of time with minimal maintenance costs. This, in combination with the aforementioned modularity, provides a sound long-term usability of the platform within CSL. The system can be utilized and subsequently upgraded when needed without having to be completely redesigned.

These results clearly show that the URETTS system is a promising platform for use in the development of underwater robotic systems. Their multi-faceted operation provides many opportunities for researchers and designers/integrators to develop and *experimentally verify* their underwater systems with relatively simple and inexpensive means.

9.2 Platform Extensions

9.2.1 MACS Improvements

Improvements to the MACS include:

1. Modification of the TC in order to accommodate suspensions (spring-dampers) on the outer lateral rollers. These can be constructed using inexpensive RC model vehicle parts and will provide the necessary lateral support during motion without straining the lateral rollers and struts housing them.

2. Construction or procurement of a long rail-guide which will be mounted on the ceiling above the tank along its longer Y axis. This will guide the motor and encoder cables of the X axis motor, as well as other possible cables coming from the TC. One possibility is to use a system with small rollers, similar to a desk drawer mechanism in combination with a coiled, spring like cable for the connections. The coiled cable will be able to extend enough to accommodate the motion of the TC-EEP an retract so to avoid entering the water or obstructing motion.

3. Addition of a third θ rotation axis on the EEP. This will enable conducting experiments where to orientation of the STO or CSL-BRF will be controllable. Previous

procurements for the URETTS acquired a stepper motor and stepper driver, thus, these can be used to implement the additional axis.

4. Design and construction of a aluminium or Plexiglas mount for the third orientation axis.

5. Construction of a mount to house a camera above the tank in order to enable the use of computer vision algorithms for ground-truth state estimation of the carriage and CSL-BRF position.

9.2.2 EMCP Upgrades

The EMCP can be extended and improved in the following:

1. The addition of a stepper drive amplifier.

2. Procurement of a 12V DC power supply with at least 5A of maximum continuous output current.

3. Inclusion of additional PWM and quadrature encoder signal channels.

4. Design and construction of a final small form-factor PCB for the CSI using SMT/SMD electronics, which is to adapted as BeagleBone Cape expansion card.

5. Design and construction of a permanent enclosure box to fit the CSI, motor amplifiers, power supplies and signal connectors. This is to be permanently mounted on the base structure of the Y axis motor.

6. Selection and procurement of an integrated stereo-camera unit for the ground-truth state estimation of the TC-EEP and CSL-BRF.

9.2.3 BBMC Porting and Redesign

The changes which are to occur for the BBMC are:

1. Porting of the source code to run on a complete OS environment like Ubuntu or Angstrom Linux.

2. Transition to the BeagleBone Black.

3. Adaptation of the peripheral drivers in order to improve the existing drivers availbile in the ARM Linux kernel. This is primarily because a driver for eQEP (quadrature encoder peripheral) is still not available in Linux. There may be the need to also improve the eHRPWM and DMTimer drivers.

4. Adaptation of the hard real-time execution code to run on the dedicated PRU co-processors. This can be implemented as dedicated firmware which is loaded onto the ROM of each PRU and an appropriate driver/interface should be designed to communicate with the real-time engine or controller in order to send motor commands and receive internal controller state information (meta-data) etc.

5. Set-up of a private WiFi network for the BeagleBone and host computer. This network will only service the connection of the on-board systems of the robots to other computers.

6. Integration of the BBMC into the ROS middleware system on the BeagleBone (Black). The local ROS system can run a set of nodes to implement and manage the real-time motion control components, while a host computer running the master node can serve as a completely integrated monitoring and control station. This host system will receive the live data stream from the BeagleBone while executing experiments.

7. Creating of Graphical User Interface (GUI) for control and monitoring of the URETTS. This can be implemented either as a component within ROS using the available packages or as a standalone program that comminicates directly to the on-board BBMC via UDP sockets.

9.3 Future Research

9.3.1 Dynamics

This thesis presented a dynamical model of the MACS that was used to predict the behaviour of the system. Although this evolved in several directions, attempting to incorporate as may of the dynamical phenomena which were identified, the description is not complete. Specifically the additional steps required to completely describe the system are:

1. Identification of the equivalent spring-damper parameters for each axis. This requires the use of external sensors to measure the state of the TC and EEP, or could possibly be accomplished via mathematical state estimation.

2. Design of a state observer which can produce online estimates of the complete state of the system. This can either be done using a model based Luenberger-type observer or a sensor based Kalman filter. Inspiration for these can be found in [33].

3. Design of a Non-linear Disturbance Observer (NDO), as described in [49].

9.3.2 Advanced Controller

Furthermore, advanced control methods could be applied to improve tracking performance when executing towing experiments. Some initial directions to explore are:

1. Since closed-loop control is currently implemented in the joint space of the actuator, provided that exterior sensors can produce estimates of the ground-truth motion of the EEP, then a controller can be designed in the mechanisms task-space.

2. Use of a Cross-Coupling Controller (CCC) to improve the performance of the task-space trajectory tracking, i.e. the 2D Cartesian path of the end-effector.

3. Use of improved dynamical models which also describe the fast-time scale dynamics due to model elasticity.

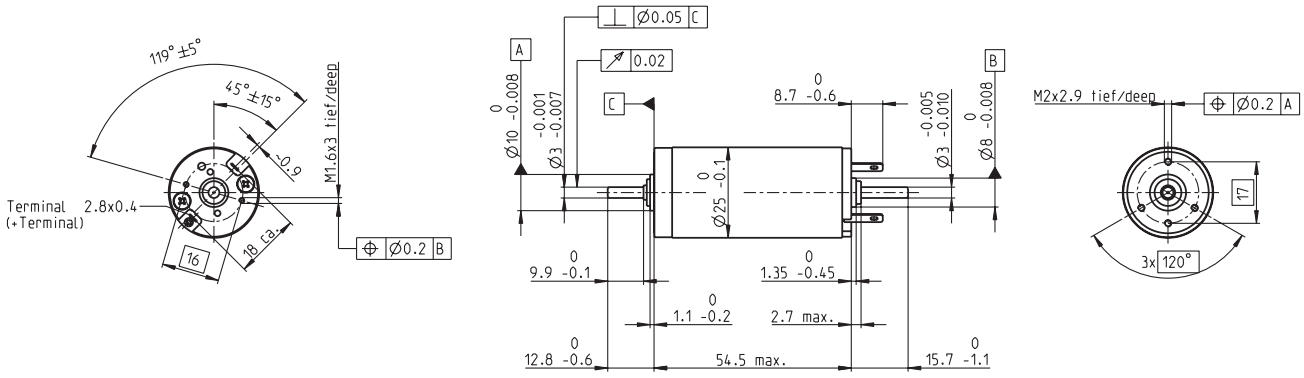
Part VI

Appendicies

Appendix A

Component Data-Sheets

RE 25 ∅25 mm, Graphite Brushes, 20 Watt



M 1:2

Stock program
Standard program
Special program (on request)

Part Numbers

according to dimensional drawing	118749	118750	**118751**	**118752**	118753	118754	**118755**	118756	118757
shaft length 15.7 shortened to 4 mm	302002	302003	302004	302005	302006	302007	302001	302008	302009
Motor Data									
Values at nominal voltage									
1 Nominal voltage	V	9	15	18	24	30	42	48	48	48
2 No load speed	rpm	10000	9660	10200	9560	9860	11100	10300	8240	5050
3 No load current	mA	110	60.8	53.9	36.9	30.5	25.2	20.1	15.2	8.52
4 Nominal speed	rpm	8970	8430	8850	8330	8640	9920	9160	7040	3830
5 Nominal torque (max. continuous torque)	mNm	11.1	20.5	22.9	26.3	26.7	27.1	27.7	28.7	30
6 Nominal current (max. continuous current)	A	1.5	1.5	1.46	1.16	0.968	0.784	0.653	0.536	0.343
7 Stall torque	mNm	232	225	220	243	249	283	264	209	129
8 Starting current	A	29.1	15.8	13.5	10.4	8.72	7.94	6.03	3.81	1.44
9 Max. efficiency	%	76	82	83	85	86	87	87	86	84
Characteristics									
10 Terminal resistance	Ω	0.309	0.952	1.33	2.32	3.44	5.29	7.96	12.6	33.4
11 Terminal inductance	mH	0.0275	0.0882	0.115	0.238	0.353	0.551	0.832	1.31	3.48
12 Torque constant	mNm/A	7.96	14.3	16.3	23.4	28.5	35.6	43.8	55	89.6
13 Speed constant	rpm/V	1200	670	586	408	335	268	218	174	107
14 Speed / torque gradient	rpm/mNm	46.5	44.7	48	40.3	40.4	39.8	39.6	39.8	39.7
15 Mechanical time constant	ms	5.68	4.87	4.77	4.55	4.47	4.4	4.37	4.37	4.35
16 Rotor inertia	gcm²	11.7	10.4	9.49	10.8	10.6	10.6	10.5	10.5	10.5

Specifications

Thermal data

17	Thermal resistance housing-ambient	14 K/W
18	Thermal resistance winding-housing	3.1 K/W
19	Thermal time constant winding	12.5 s
20	Thermal time constant motor	612 s
21	Ambient temperature	-30…+100°C
22	Max. permissible winding temperature	+125°C

Mechanical data (ball bearings)

23	Max. permissible speed	14000 rpm
24	Axial play	0.05 - 0.15 mm
25	Radial play	0.025 mm
26	Max. axial load (dynamic)	3.2 N
27	Max. force for press fits (static)	64 N
	(static, shaft supported)	800 N
28	Max. radial loading, 5 mm from flange	16 N

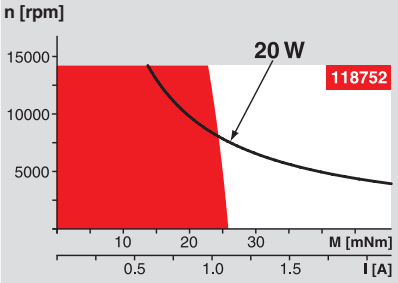
Other specifications

29	Number of pole pairs	1
30	Number of commutator segments	11
31	Weight of motor	130 g

Values listed in the table are nominal.
Explanation of the figures on page 71.

Option
Preloaded ball bearings

Operating Range



Comments

Continuous operation
In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient.
= Thermal limit.

Short term operation
The motor may be briefly overloaded (recurring).

Assigned power rating

maxon Modular System

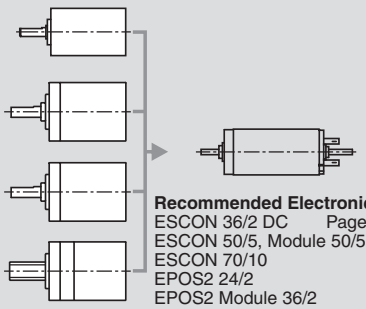
Overview on page 20 - 25

Planetary Gearhead
∅26 mm
0.75 - 4.5 Nm
Page 257

Planetary Gearhead
∅32 mm
0.75 - 6.0 Nm
Page 259/260/263

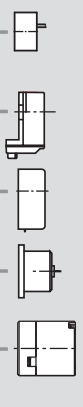
Koaxdrive
∅32 mm
1.0 - 4.5 Nm
Page 268

Spindle Drive
∅32 mm
Page 286–288



Recommended Electronics:

ESCON 36/2 DC	Page 320
ESCON 50/5, Module 50/5	321
ESCON 70/10	321
EPOS2 24/2	330
EPOS2 Module 36/2	330
EPOS2 24/5	331
EPOS2 50/5	331
EPOS2 P 24/5	334
EPOS3 70/10 EtherCAT	337
Notes	**22**



Encoder MR
128 - 1000 CPT,
3 channels
Page 302

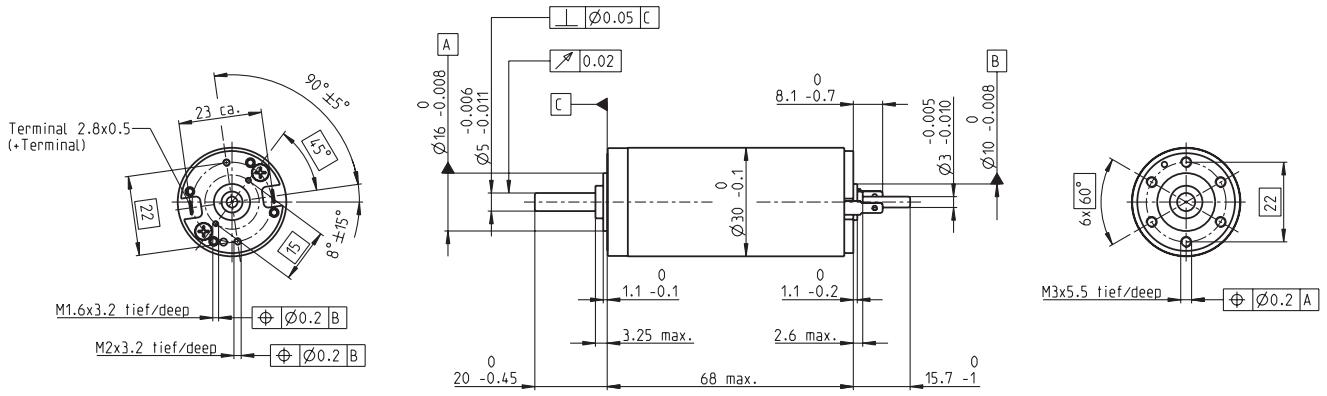
Encoder Enc
22 mm
100 CPT, 2 channels
Page 304

Encoder HED_ 5540
500 CPT,
3 channels
Page 305/307

DC-Tacho DCT
∅22 mm
0.52 V
Page 315

Brake AB 28
24 VDC
0.4 Nm
Page 348

RE 30 ∅30 mm, Graphite Brushes, 60 Watt



M 1:2

Stock program
Standard program
Special program (on request)

Part Numbers

according to dimensional drawing	310005	310006	**310007**	310008	**310009**
shaft length 15.7 shortened to 8.7 mm	268193	268213	268214	268215	268216
Motor Data					
Values at nominal voltage					
1 Nominal voltage	V	12	18	24	36	48
2 No load speed	rpm	8170	8590	8810	8590	8490
3 No load current	mA	301	213	165	106	78.6
4 Nominal speed	rpm	7630	7910	8050	7840	7760
5 Nominal torque (max. continuous torque)	mNm	51.6	75.5	85.6	86.6	89.7
6 Nominal current (max. continuous current)	A	4	4	3.47	2.28	1.74
7 Stall torque	mNm	852	1000	1020	1000	1050
8 Starting current	A	61.1	50.3	39.3	25.2	19.6
9 Max. efficiency	%	85	87	87	87	88
Characteristics						
10 Terminal resistance	Ω	0.196	0.358	0.611	1.43	2.45
11 Terminal inductance	mH	0.0345	0.0703	0.119	0.281	0.513
12 Torque constant	mNm/A	13.9	19.9	25.9	39.8	53.8
13 Speed constant	rpm/V	685	479	369	240	178
14 Speed / torque gradient	rpm/mNm	9.64	8.61	8.69	8.61	8.09
15 Mechanical time constant	ms	3.4	3.24	3.05	2.98	2.94
16 Rotor inertia	gcm²	33.7	35.9	33.5	33.1	34.7

Specifications

Thermal data

17	Thermal resistance housing-ambient	6.0 K/W
18	Thermal resistance winding-housing	1.7 K/W
19	Thermal time constant winding	16.3 s
20	Thermal time constant motor	525 s
21	Ambient temperature	-30…+100°C
22	Max. permissible winding temperature	+125°C

Mechanical data (ball bearings)

23	Max. permissible speed	12000 rpm
24	Axial play	0.05 - 0.15 mm
25	Radial play	0.025 mm
26	Max. axial load (dynamic)	5.6 N
27	Max. force for press fits (static)	110 N
	(static, shaft supported)	1200 N
28	Max. radial loading, 5 mm from flange	28 N

Other specifications

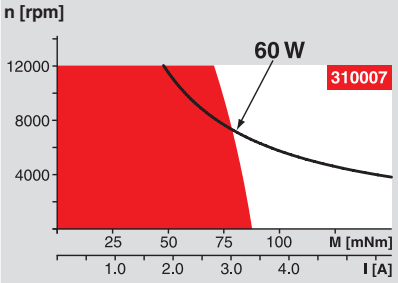
29	Number of pole pairs	1
30	Number of commutator segments	13
31	Weight of motor	260 g

Values listed in the table are nominal.
Explanation of the figures on page 71.

⚠ Tolerances may vary from the standard specification.

Option
Preloaded ball bearings

Operating Range



Comments

Continuous operation
In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient.
= Thermal limit.

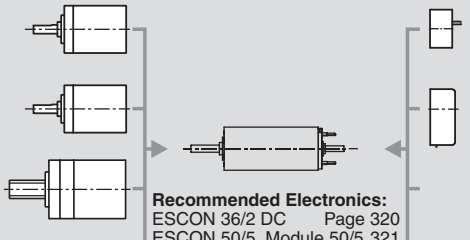
Short term operation
The motor may be briefly overloaded (recurring).

Assigned power rating

maxon Modular System

Overview on page 20 - 25

Planetary Gearhead
∅32 mm
0.75 - 6.0 Nm
Page 259–265
Koaxdrive
∅32 mm
1.0 - 4.5 Nm
Page 268
Spindle Drive
∅32 mm
Page 286–288



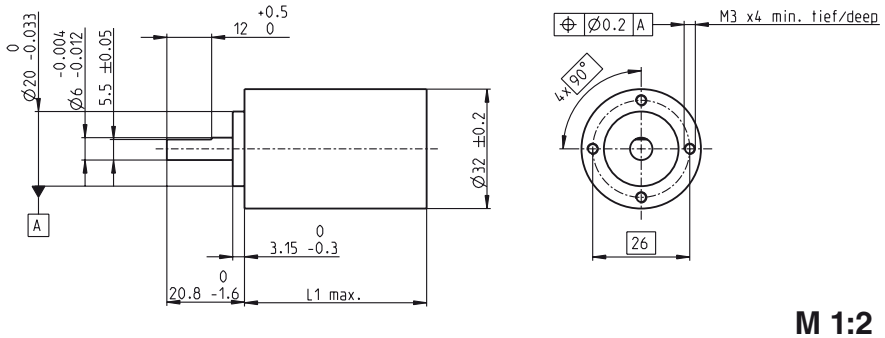
Encoder MR
256 - 1024 CPT,
3 channels
Page 303
Encoder HED_ 5540
500 CPT,
3 channels
Page 305/309

Recommended Electronics:

ESCON 36/2 DC	Page 320
ESCON 50/5, Module 50/5	321
ESCON 70/10	321
EPOS2 Module 36/2	330
EPOS2 24/5	331
EPOS2 50/5	331
EPOS2 P 24/5	334
EPOS3 70/10 EtherCAT	337
Notes	**22**

Planetary Gearhead GP 32 BZ ∅32 mm, 0.75–4.5 Nm

Low Backlash



M 1:2

Technical Data

Planetary Gearhead			straight teeth
Output shaft			stainless steel, hardened
Bearing at output			ball bearing
Radial play, 5 mm from flange			max. 0.1 mm
Axial play			max. 0.7 mm
Max. permissible axial load			120 N
Max. permissible force for press fits			120 N
Sense of rotation, drive to output			=
Recommended input speed			< 4000 rpm
Recommended temperature range			-40…+100°C
Number of stages	1	2	3
Max. radial load, 12 mm from flange	70 N	140 N	210 N

Stock program
Standard program
Special program (on request)

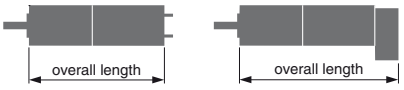
Article Numbers

	358975	351942	358331	357988	358335	358385	358512	358513	358515	358516

Gearhead Data

1	Reduction		3.7 : 1	5.2 : 1	19 : 1	27 : 1	35 : 1	71 : 1	100 : 1	139 : 1	181 : 1	236 : 1
2	Reduction absolute		63/17	57/11	3591/187	3249/121	1539/44	226233/3179	204687/2057	185193/1331	87723/484	41553/176
3	Max. motor shaft diameter	mm	5.5	3	3	3	3	3	3	3	3	3
4	Number of stages		1	1	2	2	2	3	3	3	3	3
5	Max. continuous torque	Nm	0.75	0.75	2.25	2.25	2.25	4.5	4.5	4.5	4.5	4.5
	Max. continuous torque within the preloading	Nm	0.5	0.5	1.1	1.1	1.1	1.7	1.7	1.7	1.7	1.7
6	Intermittently permissible torque at gear output	Nm	1.1	1.1	3.2	3.2	3.2	6.2	6.2	6.2	6.2	6.2
7	Max. efficiency	%	85	85	80	80	80	70	70	70	70	70
8	Weight	g	150	150	190	190	190	240	240	240	240	240
9	Average backlash no load	°	0.15	0.15	0.35	0.35	0.35	0.5	0.5	0.5	0.5	0.5
10	Mass inertia	gcm²	1.25	1.25	0.75	0.75	0.75	0.7	0.7	0.7	0.7	0.7
11	Gearhead length L1*	mm	33.5	33.5	43.6	43.6	43.6	53.1	53.1	53.1	53.1	53.1

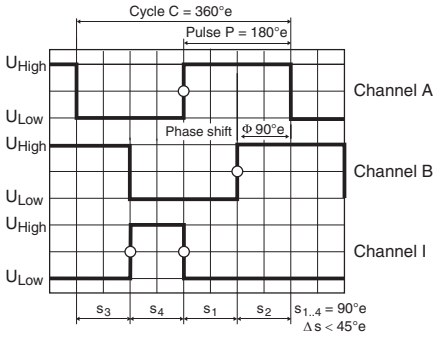
*for EC 32 L1 is + 6.4 mm, for RE 30 L1 is + 1.0 mm



maxon Modular System

+ Motor	Page	+ Sensor/Brake	Page	Overall length [mm] = Motor length + gearhead length + (sensor/brake) + assembly parts									
RE 25	77/79			88.1	88.1	98.2	98.2	98.2	107.7	107.7	107.7	107.7	107.7
RE 25	77/79	MR	272	99.1	99.1	109.2	109.2	109.2	118.7	118.7	118.7	118.7	118.7
RE 25	77/79	Enc 22	274	102.2	102.2	112.3	112.3	112.3	121.8	121.8	121.8	121.8	121.8
RE 25	77/79	HED_ 5540	276/278	108.9	108.9	119.0	119.0	119.0	128.5	128.5	128.5	128.5	128.5
RE 25	77/79	DCT 22	286	110.4	110.4	120.5	120.5	120.5	130.0	130.0	130.0	130.0	130.0
RE 25, 20 W	78			76.6	76.6	86.7	86.7	86.7	96.2	96.2	96.2	96.2	96.2
RE 25, 20 W	78	MR	272	87.6	87.6	97.7	97.7	97.7	107.2	107.2	107.2	107.2	107.2
RE 25, 20 W	78	HED_ 5540	277/280	97.4	97.4	107.5	107.5	107.5	117.0	117.0	117.0	117.0	117.0
RE 25, 20 W	78	DCT 22	286	98.9	98.9	109.0	109.0	109.0	118.5	118.5	118.5	118.5	118.5
RE 25, 20 W	78	AB 28	330	110.7	110.7	120.8	120.8	120.8	130.3	130.3	130.3	130.3	130.3
RE 25, 20 W	78	HED_ 5540 / AB 28	277/330	127.9	127.9	138.0	138.0	138.0	147.5	147.5	147.5	147.5	147.5
RE 25, 20 W	79	AB 28	330	122.2	122.2	132.3	132.3	132.3	141.8	141.8	141.8	141.8	141.8
RE 25, 20 W	79	HED_ 5540 / AB 28	276/330	139.4	139.4	149.5	149.5	149.5	159.0	159.0	159.0	159.0	159.0
RE 30, 60 W	80			102.6	102.6	112.7	112.7	112.7	122.2	122.2	122.2	122.2	122.2
RE 30, 60 W	80	MR	273	114.0	114.0	124.1	124.1	124.1	133.6	133.6	133.6	133.6	133.6
RE 35, 90 W	81			104.6	104.6	114.7	114.7	114.7	124.2	124.2	124.2	124.2	124.2
RE 35, 90 W	81	MR	273	116.0	116.0	126.1	126.1	126.1	135.6	135.6	135.6	135.6	135.6
RE 35, 90 W	81	HED_ 5540	276/278	125.3	125.3	135.4	135.4	135.4	144.9	144.9	144.9	144.9	144.9
RE 35, 90 W	81	DCT 22	286	122.7	122.7	132.8	132.8	132.8	142.3	142.3	142.3	142.3	142.3
RE 35, 90 W	81	AB 28	330	140.7	140.7	150.8	150.8	150.8	160.3	160.3	160.3	160.3	160.3
RE 35, 90 W	81	HEDS 5540 / AB 28	276/330	157.9	157.9	168.0	168.0	168.0	177.5	177.5	177.5	177.5	177.5
A-max 26	101-108			78.3	78.3	88.4	88.4	88.4	97.9	97.9	97.9	97.9	97.9
A-max 26	101-107	MEnc 13	285	85.4	85.4	95.5	95.5	95.5	105.0	105.0	105.0	105.0	105.0
A-max 26	102-108	MR	272	87.1	87.1	97.2	97.2	97.2	106.7	106.7	106.7	106.7	106.7
A-max 26	102-108	Enc 22	275	92.7	92.7	102.8	102.8	102.8	112.3	112.3	112.3	112.3	112.3
A-max 26	102-108	HED_ 5540	277/278	96.7	96.7	106.8	106.8	106.8	116.3	116.3	116.3	116.3	116.3
A-max 32	109/111			96.5	96.5	106.6	106.6	106.6	116.1	116.1	116.1	116.1	116.1
A-max 32	110/112			95.1	95.1	105.2	105.2	105.2	114.7	114.7	114.7	114.7	114.7
A-max 32	110/112	MR	273	106.3	106.3	116.4	116.4	116.4	125.9	125.9	125.9	125.9	125.9
A-max 32	110/112	HED_ 5540	277/278	115.9	115.9	126.0	126.0	126.0	135.5	135.5	135.5	135.5	135.5
EC 32, 80 W	156			100.2	100.2	110.3	110.3	110.3	119.8	119.8	119.8	119.8	119.8
EC 32, 80 W	156	HED_ 5540	277/279	118.6	118.6	128.7	128.7	128.7	138.2	138.2	138.2	138.2	138.2
EC 32, 80 W	156	Res 26	287	120.3	120.3	130.4	130.4	130.4	139.9	139.9	139.9	139.9	139.9
MCD EPOS, 60 W	325			153.6	153.6	163.7	163.7	163.7	173.2	173.2	173.2	173.2	173.2
MCD EPOS P, 60 W	325			153.6	153.6	163.7	163.7	163.7	173.2	173.2	173.2	173.2	173.2

Encoder MR Type ML, 128–1000 CPT, 3 Channels, with Line Driver



Direction of rotation cw (definition cw p. 70)

Stock program
Standard program
Special program (on request)

Part Numbers	225771	225773	225778	225805	225780
Type					
Counts per turn	128	256	500	512	1000
Number of channels	3	3	3	3	3
Max. operating frequency (kHz)	80	160	200	320	200
Max. speed (rpm)	37500	37500	24000	37500	12000

maxon Modular System

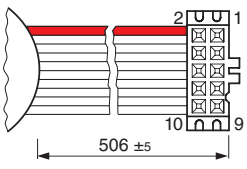
+ Motor	Page	+ Gearhead	Page	+ Brake	Page	Overall length [mm] / ● see Gearhead				
RE 25	99/101					65.5	65.5	65.5	65.5	65.5
RE 25	99/101	GP 26, 0.75 - 2.0 Nm	257			●	●	●	●	●
RE 25	99/101	GP 32, 0.75 - 6.0 Nm	259-264			●	●	●	●	●
RE 25	99/101	KD 32, 1.0 - 4.5 Nm	268			●	●	●	●	●
RE 25	99/101	GP 32 S	286-288			●	●	●	●	●
RE 25, 20 W	100					54.0	54.0	54.0	54.0	54.0
RE 25, 20 W	100	GP 26, 0.75 - 2.0 Nm	257			●	●	●	●	●
RE 25, 20 W	100	GP 32, 0.75 - 6.0 Nm	259-264			●	●	●	●	●
RE 25, 20 W	100	KD 32, 1.0 - 4.5 Nm	268			●	●	●	●	●
RE 25, 20 W	100	GP 32 S	286-288			●	●	●	●	●
A-max 26	126-132					53.5	53.5	53.5	53.5	53.5
A-max 26	126-132	GP 26, 0.75 - 4.5 Nm	257			●	●	●	●	●
A-max 26	126-132	GS 30, 0.07 - 0.2 Nm	258			●	●	●	●	●
A-max 26	126-132	GP 32, 0.75 - 6.0 Nm	259-264			●	●	●	●	●
A-max 26	126-132	GS 38, 0.1 - 0.6 Nm	269			●	●	●	●	●
A-max 26	126-132	GP 32 S	286-288			●	●	●	●	●
RE-max 29	155-158					53.5	53.5	53.5	53.5	53.5
RE-max 29	155-158	GP 32, 0.75 - 6.0 Nm	260-264			●	●	●	●	●
RE-max 29	155-158	GP 32 S	286-288			●	●	●	●	●
EC-max 30, 40 W	192							54.2		54.2
EC-max 30, 40 W	192	GP 32, 1 - 8.0 Nm	264/266					●		●
EC-max 30, 40 W	192	KD 32, 1.0 - 4.5 Nm	268					●		●
EC-max 30, 40 W	192	GP 32 S	286-288					●		●
EC-max 30, 60 W	193							76.2		76.2
EC-max 30, 60 W	193	GP 32, 8 Nm	266					●		●
EC-max 30, 60 W	193	KD 32, 1.0 - 4.5 Nm	268					●		●
EC-max 30, 60 W	193	GP 42, 3 - 15 Nm	271					●		●
EC-4pole 30	201							59.2		59.2
EC-4pole 30	201	GP 32, 4.0 - 8.0 Nm	266					●		●
EC-4pole 30	201	GP 42, 3 - 15 Nm	271					●		●
EC-4pole 30	202							76.2		76.2
EC-4pole 30	202	GP 32, 4.0 - 8.0 Nm	266					●		●
EC-4pole 30	202	GP 42, 3 - 15 Nm	271					●		●

Technical Data

Supply voltage V_{CC}	5 V ± 5%
Output signal	TTL compatible
Phase shift Φ	90°e ± 45°e
Index pulse width	90°e ± 45°e
Operating temperature range	-25…+85°C
Moment of inertia of code wheel	≤ 0.7 gcm²
Output current per channel	max. 5 mA

The index signal I is synchronised with channel A or B.

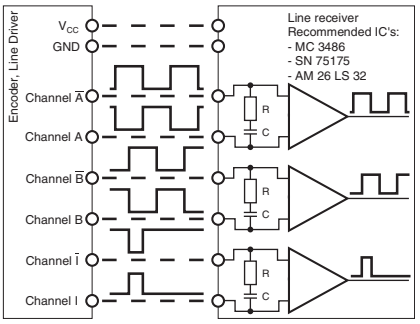
Pin Allocation



1	N.C.
2	V_{CC}
3	GND
4	N.C.
5	Channel $\overline{A}$
6	Channel A
7	Channel $\overline{B}$
8	Channel B
9	Channel $\overline{I}$ (Index)
10	Channel I (Index)

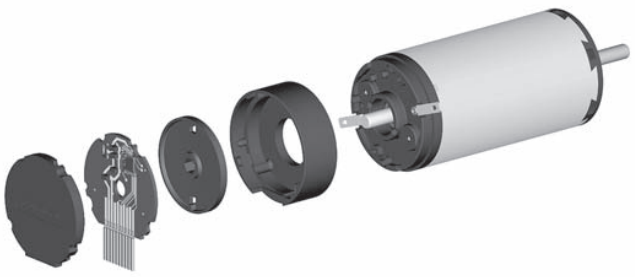
DIN Connector 41651/
EN 60603-13
flat band cable AWG 28

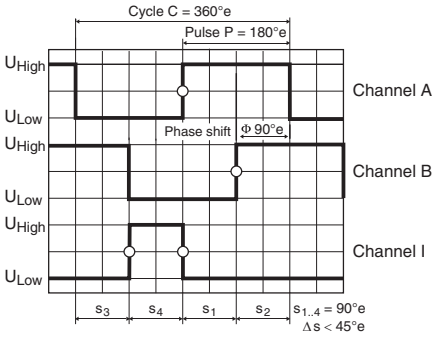
Connection example



Terminal resistance R = typical 120 Ω
Capacitor C ≥ 0.1 nF per m line length

Encoder MR Type L, 256–1024 CPT, 3 Channels, with Line Driver





Direction of rotation cw (definition cw p. 70)

Stock program
Standard program
Special program (on request)

Part Numbers	225783	228452	225785	228456	225787
Type					
Counts per turn	256	500	512	1000	1024
Number of channels	3	3	3	3	3
Max. operating frequency (kHz)	80	200	160	200	320
Max. speed (rpm)	18750	24000	18750	12000	18750

maxon Modular System

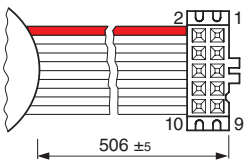
+ Motor	Page	+ Gearhead	Page	+ Brake	Page	Overall length [mm] / ● see Gearhead				
RE 30, 15 W	102					79.4	79.4	79.4	79.4	79.4
RE 30, 15 W	102	GP 32, 0.75 - 4.5 Nm	261			●	●	●	●	●
RE 30, 60 W	103					79.4	79.4	79.4	79.4	79.4
RE 30, 60 W	103	GP 32, 0.75 - 4.5 Nm	259			●	●	●	●	●
RE 30, 60 W	103	GP 32, 0.75 - 6.0 Nm	261-265			●	●	●	●	●
RE 30, 60 W	103	GP 32 S	286-288			●	●	●	●	●
RE 35, 90 W	104					82.4	82.4	82.4	82.4	82.4
RE 35, 90 W	104	GP 32, 0.75 - 4.5 Nm	259			●	●	●	●	●
RE 35, 90 W	104	GP 32, 0.75 - 6.0 Nm	261-265			●	●	●	●	●
RE 35, 90 W	104	GP 32, 4.0 - 8.0 Nm	266			●	●	●	●	●
RE 35, 90 W	104	GP 42, 3 - 15 Nm	270			●	●	●	●	●
RE 35, 90 W	104	GP 32 S	286-288			●	●	●	●	●
RE 40, 150 W	105					82.4	82.4	82.4	82.4	82.4
RE 40, 150 W	105	GP 42, 3 - 15 Nm	270			●	●	●	●	●
RE 40, 150 W	105	GP 52, 4 - 30 Nm	273			●	●	●	●	●
A-max 32	134/136					72.7	72.7	72.7	72.7	72.7
A-max 32	134/136	GP 32, 0.75 - 6.0 Nm	261-265			●	●	●	●	●
A-max 32	134/136	GS 38, 0.1 - 0.6 Nm	269			●	●	●	●	●
A-max 32	134/136	GP 32 S	286-288			●	●	●	●	●
EC-max 40, 70 W	194					73.9	73.9	73.9	73.9	73.9
EC-max 40, 70 W	194	GP 42, 3 - 15 Nm	271			●	●	●	●	●
EC-max 40, 120 W	195					103.9	103.9	103.9	103.9	103.9
EC-max 40, 120 W	195	GP 52, 4 - 30 Nm	274			●	●	●	●	●
EC-i 40, 50 W	216					41.9	41.9	41.9	41.9	41.9
EC-i 40, 50 W	216	GP 32, 1 - 6 Nm	264			●	●	●	●	●
EC-i 40, 50 W	216	GP 32 S	286-288			●	●	●	●	●
EC-i 40, 70 W	217					51.9	51.9	51.9	51.9	51.9
EC-i 40, 70 W	217	GP 32, 1 - 6 Nm	264			●	●	●	●	●
EC-i 40, 70 W	217	GP 32 S	286-288			●	●	●	●	●

Technical Data

Supply voltage V_{CC}	5 V ± 5%
Output signal	TTL compatible
Phase shift Φ	90°e ± 45°e
Index pulse width	90°e ± 45°e
Operating temperature range	-25…+85°C
Moment of inertia of code wheel	≤ 1.7 gcm²
Output current per channel	max. 5 mA

The index signal I is synchronised with channel A or B.

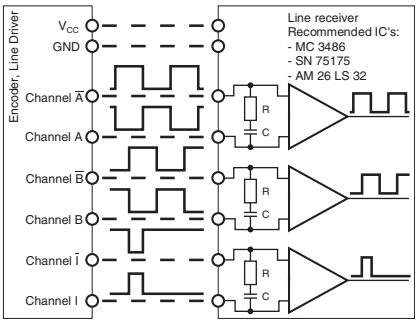
Pin Allocation



1	N.C.
2	V_{CC}
3	GND
4	N.C.
5	Channel $\overline{A}$
6	Channel A
7	Channel $\overline{B}$
8	Channel B
9	Channel $\overline{I}$ (Index)
10	Channel I (Index)

DIN Connector 41651/
EN 60603-13
flat band cable AWG 28

Connection example



Terminal resistance R = typical 120 Ω
Capacitor C ≥ 0.1 nF per m line length

maxon motor

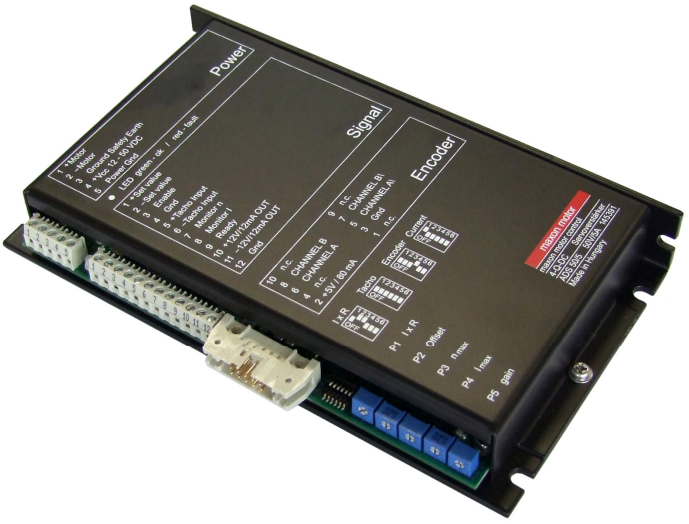
maxon motor control 4-Q-DC Servoamplifier ADS 50/5

Order number 145391

Operating Instructions July 2009 Edition

The ADS 50/5 is a powerful servoamplifier for driving permanent magnet DC motors up to 250 Watts.
Four modes can be selected by DIP switches on the board:

- Speed control using tacho signals
- Speed control using encoder signals
- IxR compensated speed control
- Torque or current control



The ADS 50/5 is protected against excess current,
excess temperature and short circuit on the motor winding. With the FET power transistors incorporated in the servoamplifier, an efficiency of up to 95 % is achieved. A built in motor choke combined with the high PWM frequency of 50 kHz allows the connection of motors with a very low inductivity. In most cases an external choke can be omitted.
Thanks to the wide input power supply range of 12 - 50 VDC, the ADS 50/5 is very versatile and can be used with various power supplies. The aluminium housing makes installation simple, with terminal markings for easy connection.

Table of Contents

The latest edition of these operating instructions may be downloaded from the internet as a PDF-file under www.maxonmotor.com, category «Service & Downloads», order number 145391 or in the e-shop http://shop.maxonmotor.com.

2 Performance Data

2.1 Electrical data

Nominal supply voltage $+V_{cc}$	12 … 50 VDC
Absolute minimum supply voltage $+V_{cc\ min}$	11.4 VDC
Absolute maximum supply voltage $+V_{cc\ max}$	52.5 VDC
Max. output voltage	$0.9 \cdot V_{CC}$
Max. output current I_{max}	10 A
Continuous output current I_{cont}	5 A
Switching frequency	50 kHz
Efficiency	95 %
Band width current controller	2.5 kHz
Built-in motor choke	150 µH / 5 A

2.2 Inputs

Set value	-10 ... +10 V (R_i = 20 kΩ)
Enable	+4 ... + 50 VDC (R_i = 15 kΩ)
Input voltage DC tacho “Tacho Input”	min. 2 VDC, max. 50 VDC (R_i = 14 kΩ)
Encoder signals “Channel A, A\, B, B\”	max. 100 kHz, TTL level

2.3 Outputs

Current monitor “Monitor I”, short-circuit protected	-10 ...+10 VDC (R_O = 100 Ω)
Speed monitor “Monitor n”, short-circuit protected	-10 ...+10 VDC (R_O = 100 Ω)
Status reading “READY”	
Open collector, short-circuit protected	max. 30 VDC ($I_L \leq$ 20 mA)

2.4 Voltage outputs

Aux. voltage, short-circuit protected	+12 VDC, -12 VDC, max. 12 mA (R_O = 1 kΩ)
Encoder supply voltage	+5 VDC, max. 80 mA

2.5 Trim potentiometers

IxR compensation
Offset
n_{max}
I_{max}
gain

2.6 LED indicator

2 coloured LED	READY / ERROR
green = ok, red = error	

2.7 Ambient temperature- / Humidity range

Operating	-10 ... +45°C
Storage	-40 ... +85°C
Non condensating	20 ... 80 %

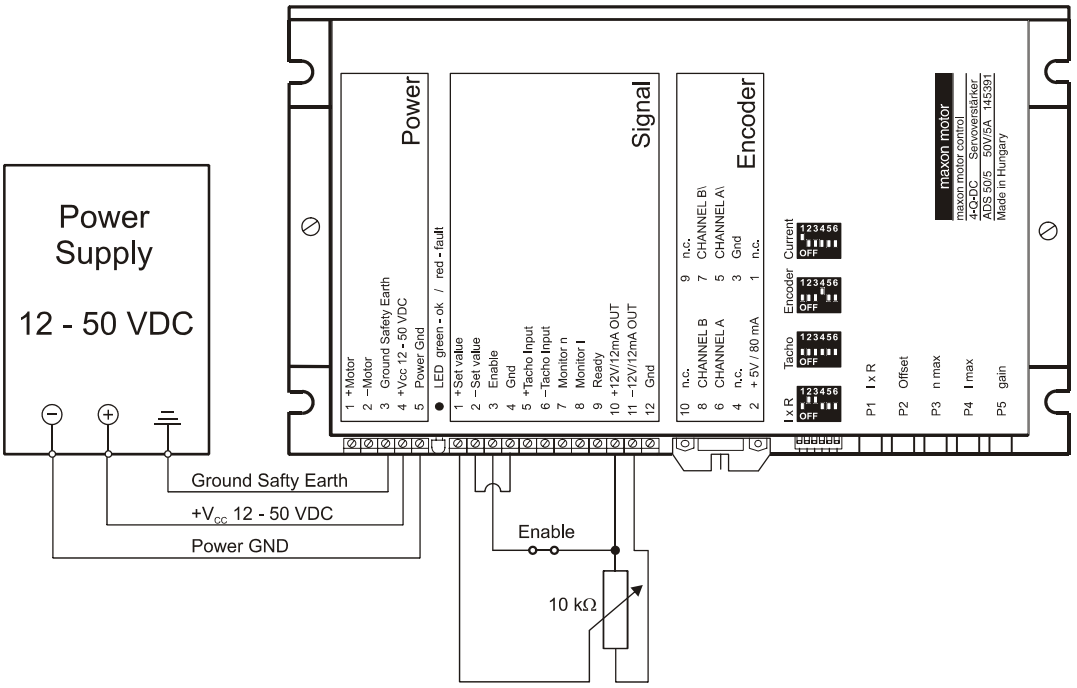
2.8 Mechanical data

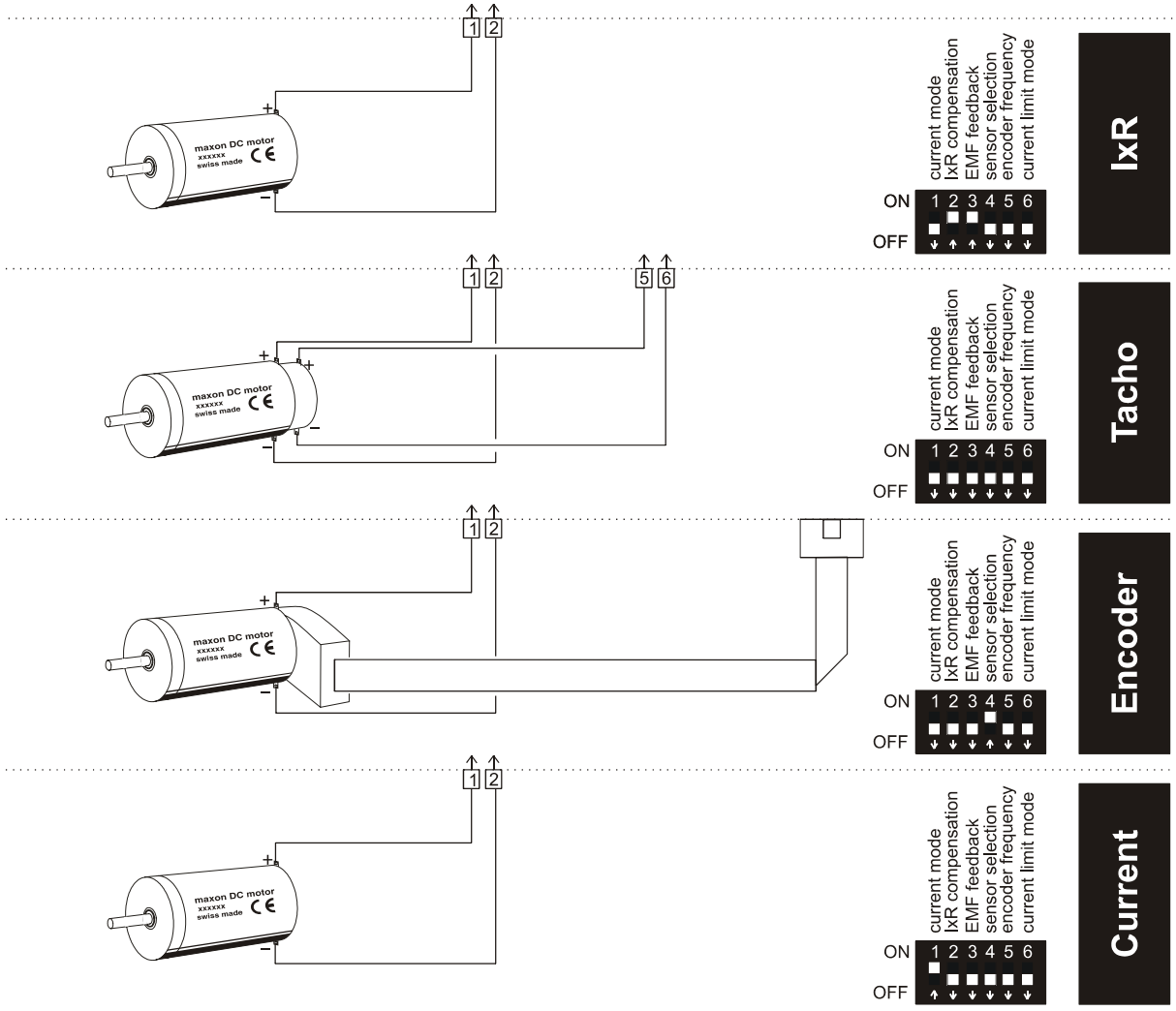
Weight	approx. 360 g
Dimensions	see dimension drawing
Mounting plate	for M4 screws

2.9 Terminal

PCB-clamps	Power (5 poles), Signal (12 poles)
Pitch	3.81 mm
suitable for wire cross section	0.14 - 1 mm^2 multiple-stranded wire or
	0.14 - 1.5 mm^2 single wire
Encoder	Plug DIN41651
	for flat cable, pitch 1.27 mm, AWG 28

3 Minimum External Wiring for Different Modes of Operation



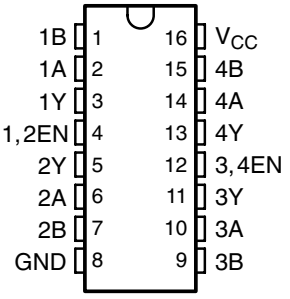


SN65175, SN75175
QUADRUPLE DIFFERENTIAL LINE RECEIVERS

SLLS145C – OCTOBER 1990 – REVISED NOVEMBER 2006

- **Meet or Exceed the Requirements of ANSI Standard EIA/TIA-422-B, RS-423-B, and RS-485**
- **Meet ITU Recommendations V.10, V.11, X.26, and X.27**
- **Designed for Multipoint Bus Transmission on Long Bus Lines in Noisy Environments**
- **3-State Outputs**
- **Common-Mode Input Voltage Range –12 V to 12 V**
- **Input Sensitivity . . . ±200 mV**
- **Input Hysteresis . . . 50 mV Typ**
- **High Input Impedance . . . 12 kΩ Min**
- **Operate From Single 5-V Supply**
- **Low-Power Requirements**
- **Plug-In Replacement for MC3486**

D OR N PACKAGE
(TOP VIEW)



description

The SN65175 and SN75175 are monolithic quadruple differential line receivers with 3-state outputs. They are designed to meet the requirements of ANSI Standards EIA/TIA-422-B, RS-423-B, and RS-485, and several ITU recommendations. These standards are for balanced multipoint bus transmission at rates up to 10 megabits per second. Each of the two pairs of receivers has a common active-high enable.

The receivers feature high input impedance, input hysteresis for increased noise immunity, and input sensitivity of ±200 mV over a common-mode input voltage range of ±12 V. The SN65175 and SN75175 are designed for optimum performance when used with the SN75172 or SN75174 quadruple differential line drivers.

The SN65175 is characterized for operation from –40°C to 85°C. The SN75175 is characterized for operation from 0°C to 70°C.

FUNCTION TABLE
(each receiver)

DIFFERENTIAL A – B	ENABLE	OUTPUT Y
$V_{ID} \geq 0.2$ V	H	H
-0.2 V $< V_{ID} < 0.2$ V	H	?
$V_{ID} \leq -0.2$ V	H	L
X	L	Z
Open circuit	H	?

H = high level, L = low level, ? = indeterminate,
X = irrelevant, Z = high impedance (off)

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

Copyright © 2006, Texas Instruments Incorporated

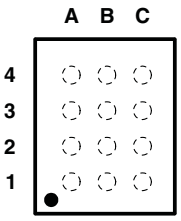
4-BIT BIDIRECTIONAL VOLTAGE-LEVEL TRANSLATOR WITH AUTOMATIC DIRECTION SENSING AND ±15-kV ESD PROTECTION

Check for Samples: TXB0104

FEATURES

- 1.2 V to 3.6 V on A Port and 1.65 V to 5.5 V on B Port ($V_{CCA} \leq V_{CCB}$)
- V_{CC} Isolation Feature – If Either V_{CC} Input Is at GND, All Outputs Are in the High-Impedance State
- OE Input Circuit Referenced to V_{CCA}
- Low Power Consumption, 5-µA Max I_{CC}
- I_{off} Supports Partial-Power-Down Mode Operation
- Latch-Up Performance Exceeds 100 mA Per JESD 78, Class II
- ESD Protection Exceeds JESD 22
  - A Port
    - 2500-V Human-Body Model (A114-B)
    - 1500-V Charged-Device Model (C101)
  - B Port
    - ±15-kV Human-Body Model (A114-B)
    - 1500-V Charged-Device Model (C101)

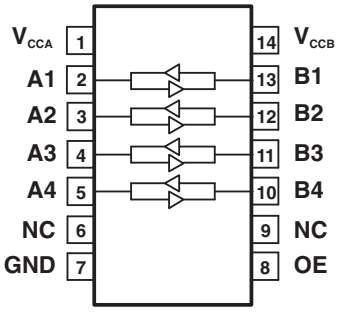
GXU/ZXU PACKAGE
(TOP VIEW)



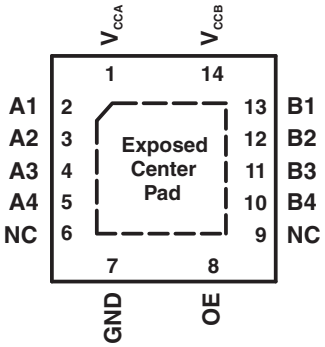
TERMINAL ASSIGNMENTS
(GXU/ZXU Package)

	A	B	C
4	A4	GND	B4
3	A3	OE	B3
2	A2	V_{CCA}	B2
1	A1	V_{CCB}	B1

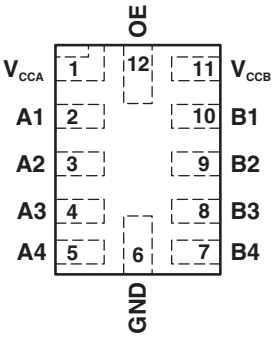
D OR PW PACKAGE
(TOP VIEW)



RGY PACKAGE
(TOP VIEW)



RUT PACKAGE
(TOP VIEW)



A. N.C. − No internal connection

B. For RGY, if the exposed center pad is used, it must only be connected as a secondary ground or left electrically open.

C. Pull up resistors are not required on both sides for Logic I/O.

D. If pull up or pull down resistors are needed, the resistor value must be over 50 kΩ.

E. 50 kΩ is a safe recommended value, if the customer can accept higher Vol or lower Voh, smaller pull up or pull down resistor is allowed, the draft estimation is Vol = Vccout × 4.5k/(4.5k + Rpu) and Voh = Vccout × Rdw/(4.5k + Rdw).

F. If pull up resistors are needed, please refer to the TXS0104 or contact TI.

G. For detailed information, please refer to application note SCEA043.

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

NanoFree is a trademark of Texas Instruments.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of the Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

Copyright © 2006–2012, Texas Instruments Incorporated

Low Power Quad Channels Digital Isolators

Check for Samples: ISO7640FM, ISO7641FM

FEATURES

- **Signaling Rate: 150 Mbps**
- **Low Power Consumption, Typical I_{CC} per Channel (3.3 V Supplies):**
  - **ISO7640FM: 2 mA at 25 Mbps**
  - **ISO7641FM: 2.4 mA at 25 Mbps**
- **Low Propagation Delay: 7 ns Typical**
- **Output Defaults to Low-state in fail-safe mode**
- **Wide Temperature Range: –40°C to 125°C**
- **50 KV/µs Transient Immunity, Typical**
- **Long Life with SiO_2 Isolation barrier**
- **Operates From 2.7V, 3.3 V and 5 V Supply and Logic Levels**
- **Wide Body SOIC-16 Package**

APPLICATIONS

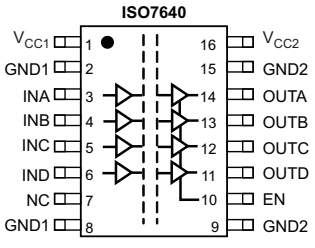
- **Optocoupler Replacement in:**
  - **Industrial Fieldbus**
    - **Profibus**
    - **Modbus**
    - **DeviceNet™ Data Buses**
  - **Servo Control Interface**
  - **Motor Control**
  - **Power Supplies**
  - **Battery Packs**

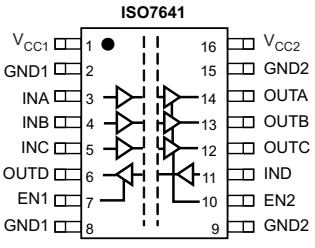
SAFETY AND REGULATORY APPROVALS

- **6000 V_{PK} / 4243 V_{RMS} for 1 Minute per UL 1577 (approved)**
- **VDE Approval for DIN EN 60747-5-2 (VDE 0884 Rev. 2), 1414 V_{PK} Working Voltage (approved)**
- **CSA Component Acceptance Notice 5A, IEC 60601-1 Medical Standard (approved)**
- **5 KV_{RMS} Reinforced Insulation per TUV for EN/UL/CSA 60950-1 and EN/UL/CSA 61010-1 (approved)**

DESCRIPTION

ISO7640FM and ISO7641FM provide galvanic isolation up to 6 KV_{PK} for 1 minute per UL and VDE. These devices are also certified up to 5 KV_{RMS} Reinforced isolation at a working voltage of 400 V_{RMS} per end equipment standards EN/UL/CSA 60950-1 and 61010-1. ISO7640F and ISO7641F are quad channel isolators; ISO7640F has four forward and ISO7641F has three forward and one reverse direction channels. Suffix F indicates that output defaults to Low-state in fail-safe conditions (see Table 1). M-Grade devices are high speed isolators capable of 150 Mbps data rate with fast propagation delays





Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

UNLESS OTHERWISE NOTED this document contains PRODUCTION DATA information current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

Copyright © 2011–2013, Texas Instruments Incorporated

8-Channel CMOS Logic to High Voltage Level Translator

Data Sheet **ADG3123**

FEATURES

- 2.3 V to 5.5 V input voltage range
- Output voltage levels (V_{DDA} and V_{DDB} to V_{SS} ≤ 35 V)
  - Low output voltage levels: down to −24.2 V
  - High output voltage levels: up to +35 V
- Rise/fall time: 12 ns/19.5 ns typical
- Propagation delay: 80 ns typical
- Operating frequency: 100 kHz typical
- Ultralow quiescent current: 65 µA typical
- 20-lead, Pb-free, TSSOP package

APPLICATIONS

- Low voltage to high voltage translation
- TFT-LCD panels
- Piezoelectric motor drivers

FUNCTIONAL BLOCK DIAGRAM

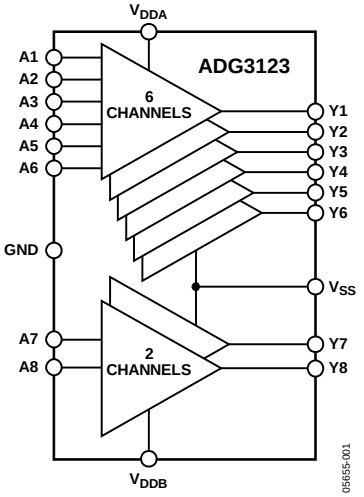


Figure 1.

GENERAL DESCRIPTION

The ADG3123 is an 8-channel, noninverting CMOS to high voltage level translator. Fabricated on an enhanced LC²MOS process, the device is capable of operating at high supply voltages while maintaining ultralow power consumption.

The internal architecture of the device ensures compatibility with logic circuits running from supply voltages within the 2.3 V to 5.5 V range. The voltages applied to Pin V_{DDA}, Pin V_{DDB}, and Pin V_{SS} set the logic levels available at the outputs on the Y side of the device. Pin V_{DDA} and Pin V_{DDB} set the high output level for Pin Y1 to Pin Y6 and for Pin Y7 to Pin Y8, respectively. The V_{SS} pin sets the low output level for all channels. The ADG3123 can provide output voltages levels down to −24.2 V for a low input level and up to +35 V for a high input logic level. For proper operation, V_{DDB} must always be greater than or equal to V_{DDA} and the voltage between the Pin V_{DDB} and Pin V_{SS} should not exceed 35 V.

The low output impedance of the channels guarantees fast rise and fall times even for significant capacitive loads. This feature, combined with low propagation delay and low power consumption, makes the ADG3123 an ideal driver for TFT-LCD panel applications.

The ADG3123 is guaranteed to operate over the −40°C to +85°C temperature range and is available in a compact, 20-lead TSSOP, Pb-free package.

PRODUCT HIGHLIGHTS

1. Compatible with a wide range of CMOS logic levels.
2. High output voltage levels.
3. Fast rise and fall times coupled with low propagation delay.
4. Ultralow power consumption.
5. Compact, 20-lead TSSOP, RoHS-compliant package.

Rev. B Document Feedback

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700 ©2005–2013 Analog Devices, Inc. All rights reserved.
Technical Support www.analog.com

Sitara™ AM335x ARM® Cortex™-A8 Microprocessors (MPUs)

Check for Samples: AM3359, AM3358

1 Device Summary

1.1 Features

- Highlights
  - Up to 1-GHz Sitara™ ARM® Cortex™-A8 32-Bit RISC Microprocessor
    - NEON™ SIMD Coprocessor
    - 32KB of L1 Instruction and 32KB Data Cache with Single-Error Detection (parity)
    - 256KB of L2 Cache with Error Correcting Code (ECC)
  - mDDR(LPDDR), DDR2, DDR3, DDR3L Support
  - General-Purpose Memory Support (NAND, NOR, SRAM) Supporting Up to 16-bit ECC
  - SGX530 3D Graphics Engine
  - LCD and Touchscreen Controller
  - Programmable Real-Time Unit and Industrial Communication Subsystem (PRU-ICSS)
  - Real-Time Clock (RTC)
  - Up to Two USB 2.0 High-Speed OTG Ports with Integrated PHY
  - 10, 100, 1000 Ethernet Switch Supporting Up to Two Ports
  - Serial Interfaces Including:
    - Two Controller Area Network Ports (CAN)
    - Six UARTs, Two McASPs, Two McSPI, and Three I2C Ports
  - 12-Bit Successive Approximation Register (SAR) ADC
  - Up to Three 32-Bit Enhanced Capture Modules (eCAP)
  - Up to Three Enhanced High-Resolution PWM Modules (eHRPWM)
  - Crypto Hardware Accelerators (AES, SHA, PKA, RNG)
- MPU Subsystem
  - Up to 1-GHz ARM® Cortex™-A8 32-Bit RISC Microprocessor
  - NEON™ SIMD Coprocessor
  - 32KB of L1 Instruction Cache with Single-Error Detection (parity)
  - 32KB of L1 Data Cache with Single Error-Detection (parity)
  - 256KB of L2 Cache with Error Correcting Code (ECC)
  - 176KB of On-Chip Boot ROM
  - 64KB of Dedicated RAM
  - Emulation and Debug
    - JTAG
  - Interrupt Controller (up to 128 interrupt requests)
- On-Chip Memory (Shared L3 RAM)
  - 64 KB of General-Purpose On-Chip Memory Controller (OCMC) RAM
  - Accessible to all Masters
  - Supports Retention for Fast Wake-Up
- External Memory Interfaces (EMIF)
  - mDDR(LPDDR), DDR2, DDR3, DDR3L Controller:
    - mDDR: 200-MHz Clock (400-MHz Data Rate)
    - DDR2: 266-MHz Clock (532-MHz Data Rate)
    - DDR3: 400-MHz Clock (800-MHz Data Rate)
    - DDR3L: 400-MHz Clock (800-MHz Data Rate)
    - 16-Bit Data Bus
    - 1 GB of Total Addressable Space
    - Supports One x16 or Two x8 Memory Device Configurations
  - General-Purpose Memory Controller (GPMC)
    - Flexible 8-Bit and 16-Bit Asynchronous Memory Interface with Up to seven Chip Selects (NAND, NOR, Muxed-NOR, SRAM)
    - Uses BCH Code to Support 4-Bit, 8-Bit, or 16-Bit ECC
    - Uses Hamming Code to Support 1-Bit ECC
  - Error Locator Module (ELM)
    - Used in Conjunction with the GPMC to

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

Sitara, SmartReflex, DSP/BIOS, XDS are trademarks of Texas Instruments.
Cortex, NEON are trademarks of ARM Ltd or its subsidiaries.
ARM is a registered trademark of ARM Ltd or its subsidiaries.
EtherCAT is a registered trademark of EtherCAT Technology Group.
POWERVR SGX is a trademark of Imagination Technologies Limited.
All other trademarks are the property of their respective owners.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of the Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

Copyright © 2011–2013, Texas Instruments Incorporated

- Locate Addresses of Data Errors from Syndrome Polynomials Generated Using a BCH Algorithm
  - Supports 4-Bit, 8-Bit, and 16-Bit per 512-byte Block Error Location Based on BCH Algorithms
- Programmable Real-Time Unit and Industrial Communication Subsystem (PRU-ICSS)
  - Supports protocols such as EtherCAT®, PROFIBUS, PROFINET, EtherNet/IP™, and more
  - Peripherals Inside the PRU-ICSS
    - One UART Port with Flow Control Pins, Supports Up to 12 Mbps
    - Two MII Ethernet Ports that Support Industrial Ethernet, such as EtherCAT
    - One MDIO Port
    - One Enhanced Capture (eCAP) Module
- Power Reset and Clock Management (PRCM) Module
  - Controls the entry and Exit of Stand-By and Deep-Sleep Modes
  - Responsible for Sleep Sequencing, Power Domain Switch-Off Sequencing, Wake-Up Sequencing and Power Domain Switch-On Sequencing
  - Clocks
    - Integrated 15-35 MHz High-Frequency Oscillator Used to Generate a Reference Clock for Various System and Peripheral Clocks
    - Supports Individual Clock Enable and Disable Control for Subsystems and Peripherals to Facilitate Reduced Power Consumption
    - Five ADPLLs to Generate System Clocks (MPU Subsystem, DDR Interface, USB and Peripherals [MMC and SD, UART, SPI, I2C], L3, L4, Ethernet, GFX [SGX530], LCD Pixel Clock)
  - Power
    - Two Non-Switchable Power Domains (Real-Time Clock [RTC], Wake-Up Logic [WAKE-UP])
    - Three Switchable Power Domains (MPU Subsystem [MPU], SGX530 [GFX], Peripherals and Infrastructure [PER])
    - Implements SmartReflex™ Class 2B for Core Voltage Scaling Based On Die Temperature, Process Variation and Performance (Adaptive Voltage Scaling [AVS])
    - Dynamic Voltage Frequency Scaling (DVFS)
- Real-Time Clock (RTC)
  - Real-Time Date (Day-Month-Year-Day of Week) and Time (Hours-Minutes-Seconds) Information
  - Internal 32.768-kHz Oscillator, RTC Logic and 1.1-V Internal LDO
  - Independent Power-on-Reset (RTC_PWRONRSTn) Input
  - Dedicated Input Pin (EXT_WAKEUP) for External Wake Events
  - Programmable Alarm Can be Used to Generate Internal Interrupts to the PRCM (for Wake Up) or Cortex-A8 (for Event Notification)
  - Programmable Alarm Can be Used with External Output (PMIC_POWER_EN) to Enable the Power Management IC to Restore Non-RTC Power Domains
- Peripherals
  - Up to Two USB 2.0 High-Speed OTG Ports with Integrated PHY
  - Up to Two Industrial Gigabit Ethernet MACs (10, 100, 1000 Mbps)
    - Integrated Switch
    - Each MAC Supports MII, RMII, RGMII and MDIO Interfaces
    - Ethernet MACs and Switch Can Operate Independent of Other Functions
    - IEEE 1588v2 Precision Time Protocol (PTP)
  - Up to Two Controller-Area Network (CAN) Ports
    - Supports CAN Version 2 Parts A and B
  - Up to Two Multichannel Audio Serial Ports (McASP)
    - Transmit and Receive Clocks Up to 50 MHz
    - Up to Four Serial Data Pins per McASP Port with Independent TX and RX Clocks
    - Supports Time Division Multiplexing (TDM), Inter-IC Sound (I2S), and similar Formats
    - Supports Digital Audio Interface Transmission (SPDIF, IEC60958-1, and AES-3 Formats)
    - FIFO Buffers for Transmit and Receive (256 bytes)
  - Up to Six UARTs
    - All UARTs Support IrDA and CIR Modes
    - All UARTs Support RTS and CTS Flow Control
    - UART1 Supports Full Modem control
  - Up to Two Master and Slave McSPI Serial Interfaces
    - Up to Two Chip Selects
    - Up to 48 MHz
  - Up to Three MMC, SD, and SDIO Ports

Copyright © 2011–2013, Texas Instruments Incorporated

- 1-Bit, 4-Bit and 8-Bit MMC, SD, and SDIO Modes
- MMCSD0 has dedicated Power Rail for 1.8-V or 3.3-V Operation
- Up to 48-MHz Data Transfer Rate
- Supports Card Detect and Write Protect
- Complies with MMC4.3 and SD and SDIO 2.0 Specifications

– Up to Three I2C Master and Slave Interfaces
- Standard Mode (up to 100 kHz)
- Fast Mode (up to 400 kHz)

– Up to Four Banks of General-Purpose IO (GPIO)
- 32 GPIOs per Bank (Multiplexed with Other Functional Pins)
- GPIOs Can be Used as Interrupt Inputs (Up to Two Interrupt Inputs per Bank)

– Up to Three External DMA Event Inputs That Can Also be Used as Interrupt Inputs

– Eight 32-Bit General-Purpose Timers
- DMTIMER1 is a 1-ms Timer Used for Operating System (OS) Ticks
- DMTIMER4 - DMTIMER7 are Pinned Out

– One Watchdog Timer

– SGX530 3D Graphics Engine
- Tile-Based Architecture Delivering Up to 20 Million Polygons per second
- Universal Scalable Shader Engine is a Multi-Threaded Engine Incorporating Pixel and Vertex Shader Functionality
- Advanced Shader Feature Set in Excess of Microsoft VS3.0, PS3.0 and OGL2.0
- Industry Standard API Support of Direct3D Mobile, OGL-ES 1.1 and 2.0, OpenVG 1.0, and OpenMax
- Fine-Grained Task Switching, Load Balancing and Power Management
- Advanced Geometry DMA Driven Operation for Minimum CPU Interaction
- Programmable High-Quality Image Anti-Aliasing
- Fully Virtualized Memory Addressing for OS Operation in a Unified Memory Architecture

– LCD Controller
- Up to 24-Bits Data Output; 8-Bits per Pixel (RGB)
- Resolution Up to 2048x2048 (With Maximum 126-MHz Pixel Clock)
- Integrated LCD Interface Display Driver (LIDD) Controller
- Integrated Raster Controller
- Integrated DMA Engine to Pull Data from the External Frame Buffer without Burdening the Processor via Interrupts or a Firmware Timer
- 512-Word Deep Internal FIFO
- Supported Display Types:
  – Character Displays - Uses LCD Interface Display Driver (LIDD) Controller to Program these Displays
  – Passive Matrix LCD Displays - Uses LCD Raster Display Controller to Provide Timing and Data for Constant Graphics Refresh to a Passive Display
  – Active Matrix LCD Displays - Uses External Frame Buffer Space and the Internal DMA Engine to Drive Streaming Data to the Panel

– 12-Bit Successive Approximation Register (SAR) ADC
- 200K Samples per Second
- Input Can be Selected from any of the Eight Analog Inputs Multiplexed Through an 8:1 analog Switch
- Can be Configured to Operate as a 4-wire, 5-wire, or 8-wire Resistive Touch Screen Controller (TSC) Interface

– Up to Three 32-Bit Enhanced Capture Modules (eCAP)
- Configurable as Three Capture Inputs or Three Auxiliary PWM Outputs

– Up to Three Enhanced High-Resolution PWM Modules (eHRPWM)
- Dedicated 16-Bit Time-Base Counter with Time and Frequency Controls
- Configurable as Six Single-Ended, Six Dual-Edge Symmetric, or Three Dual-Edge Asymmetric Outputs

– Up to Three 32-Bit Enhanced Quadrature Encoder Pulse (eQEP) Modules

- Device Identification
  – Contains Electrical fuse Farm (FuseFarm) of Which Some Bits are Factory Programmable
    - Production ID
    - Device Part Number (Unique JTAG ID)
    - Device Revision (readable by Host ARM)
- Debug Interface Support
  – JTAG and cJTAG for ARM (Cortex-A8 and PRCM), PRU-ICSS Debug
  – Supports Device Boundary Scan
  – Supports IEEE 1500
- DMA
  – On-Chip Enhanced DMA Controller (EDMA) has Three Third-Party Transfer Controllers (TPTC) and One Third-Party Channel Controller (TPCC), Which Supports Up to 64 Programmable Logical Channels and Eight QDMA Channels. EDMA is Used for:
    - Transfers to and from On-Chip Memories

Copyright © 2011–2013, Texas Instruments Incorporated

    - Transfers to and from External Storage (EMIF, General-Purpose Memory Controller, Slave Peripherals)
- Inter-Processor Communication (IPC)
  - Integrates Hardware-Based Mailbox for IPC and Spinlock for Process Synchronization Between the Cortex-A8, PRCM, and PRU-ICSS
    - Mailbox Registers that Generate Interrupts
      - Four Initiators (Cortex-A8, PRCM, PRU0, PRU1)
    - Spinlock has 128 Software-Assigned Lock Registers
- Security
  - Crypto Hardware Accelerators (AES, SHA, PKA, RNG)
- Boot Modes
  - Boot Mode is Selected via Boot Configuration Pins Latched on the Rising Edge of the PWRONRSTn Reset Input Pin
- Packages:
  - 298-Pin S-PBGA-N298 Via Channel™ package (ZCE Suffix), 0.65-mm Ball Pitch
  - 324-Pin S-PBGA-N324 package (ZCZ Suffix), 0.80-mm Ball Pitch

1.2 Applications

- Gaming Peripherals
- Home and Industrial Automation
- Consumer Medical Appliances
- Printers
- Smart Toll Systems
- Connected Vending Machines
- Weighing Scales
- Educational Consoles
- Advanced Toys

Copyright © 2011–2013, Texas Instruments Incorporated

1.3 Description

The AM335x microprocessors, based on the ARM Cortex-A8, are enhanced with image, graphics processing, peripherals and industrial interface options such as EtherCAT and PROFIBUS. The device supports the following high-level operating systems (HLOSs) that are available free of charge from TI:

- Linux®
- Android™

The AM335x microprocessor contains these subsystems:

- Microprocessor unit (MPU) subsystem based on the ARM Cortex-A8 microprocessor.
- POWERVR SGX™ Graphics Accelerator subsystem for 3D graphics acceleration to support display and gaming effects.
- The Programmable Real-Time Unit and Industrial Communication Subsystem (PRU-ICSS) is separate from the ARM core, allowing independent operation and clocking for greater efficiency and flexibility. The PRU-ICSS enables additional peripheral interfaces and real-time protocols such as EtherCAT, PROFINET, EtherNet/IP, PROFIBUS, Ethernet Powerlink, Sercos, and others.

Copyright © 2011–2013, Texas Instruments Incorporated

Appendix B

System Configurations

B.1 EMCP Connections

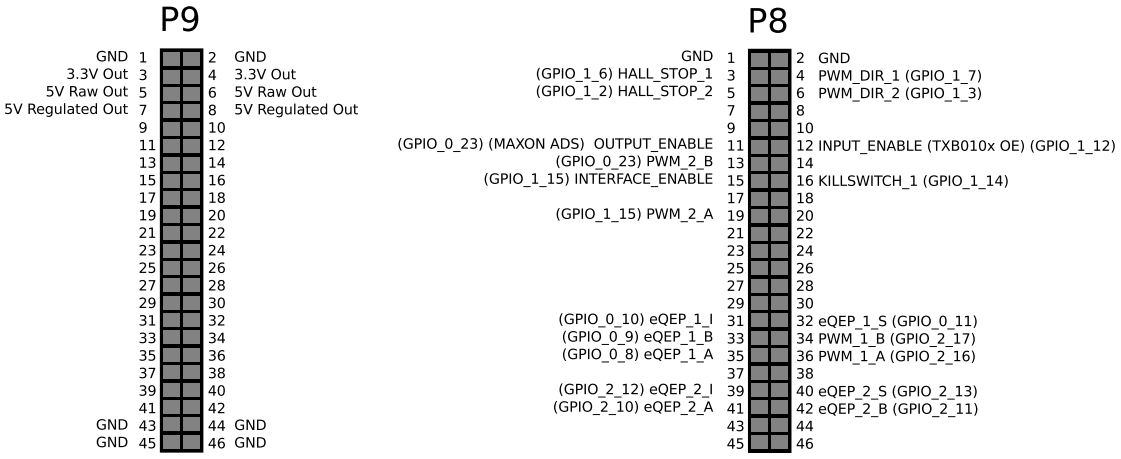


FIGURE B.1: Overview of the BeagleBone connections in the EMCP.

Bibliography

[1] Dimitrios Tzeranis, Evangelos Papadopoulos, and George Triantafyllou. On the design of an autonomous robot fish. In *Proceedings of 11th IEEE Mediterranean Conference on Control and Automation (MED'03), Rhodes, June*. Citeseer, 2003.

[2] E. Papadopoulos, E. Apostolopoulos, and P. Tsigkourakos. Design, control, and experimental performance of a teleoperated robotic fish. In *Control and Automation, 2009. MED '09. 17th Mediterranean Conference on*, pages 766–771, June 2009. doi: 10.1109/MED.2009.5164636.

[3] R.H. Tizard and B.G.V. Harrington. An electronic speed control for the towing carriage of a ship-model testing tank. *Proceedings of the IEE - Part II: Power Engineering*, 97(59):651–662, October 1950. doi: 10.1049/pi-2.1950.0189.

[4] Hua ming Wang, Yong he Xie, Jun mei Liu, Zao jian Zou, and Wei He. Experimental and numerical study of hydrodynamic forces on ship in oblique motion. In *Remote Sensing, Environment and Transportation Engineering (RSETE), 2011 International Conference on*, pages 328–331, June 2011. doi: 10.1109/RSETE.2011.5964280.

[5] C. Aage and L.W. Smitt. Hydrodynamic manoeuvrability data of a flatfish type auv. In *OCEANS '94. 'Oceans Engineering for Today's Technology and Tomorrow's Preservation.' Proceedings*, volume 3, pages III/425–III/430 vol.3, Sep 1994. doi: 10.1109/OCEANS.1994.364236.

[6] Junzhi Yu and Long Wang. Parameter optimization of simplified propulsive model for biomimetic robot fish. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 3306–3311, April 2005. doi: 10.1109/ROBOT.2005.1570620.

[7] J. Pereira and A. Duncan. System identification of underwater vehicles. In *Underwater Technology, 2000. UT 00. Proceedings of the 2000 International Symposium on*, pages 419–424, 2000. doi: 10.1109/UT.2000.852581.

[8] Yuh-Lin Hwang. Hydrodynamic modeling of lmrs unmanned underwater vehicle and tow tank test validation. In *OCEANS 2003. Proceedings*, volume 3, pages 1425–1430 Vol.3, Sept 2003. doi: 10.1109/OCEANS.2003.178071.

[9] F. Lopez Pena, M. Miguez Gonzalez, A. Deibe, D. Pena, and F. Orjales. An autonomous scale ship model for towing tank testing. In *Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), 2013 IEEE 7th International Conference on*, volume 01, pages 25–29, Sept 2013. doi: 10.1109/IDAACS.2013.6662633.

[10] Jenhwa Guo and Forng-Chen Chiu. Maneuverability of a flat-streamlined underwater vehicle. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 1, pages 897–902 vol.1, 2001. doi: 10.1109/ROBOT.2001.932664.

[11] D. Shea, C. Williams, Moqin He, P. Crocker, N. Riggs, and R. Bachmayer. Design and testing of the marport sqx-500 twin-pod auv. In *Autonomous Underwater Vehicles (AUV), 2010 IEEE/OES*, pages 1–9, Sept 2010. doi: 10.1109/AUV.2010.5779657.

[12] Zheng Chen, S. Shatara, and Xiaobo Tan. Modeling of biomimetic robotic fish propelled by an ionic polymer metal composite caudal fin. *Mechatronics, IEEE/ASME Transactions on*, 15(3):448–459, June 2010. ISSN 1083-4435. doi: 10.1109/TMECH.2009.2027812.

[13] Dong Sun, Xiaoyin Shao, and Gang Feng. A model-free cross-coupled control for position synchronization of multi-axis motions: Theory and experiments. *Control Systems Technology, IEEE Transactions on*, 15(2):306–314, March 2007. ISSN 1063-6536. doi: 10.1109/TCST.2006.883201.

[14] M. Tomizuka. Model based prediction, preview and robust controls in motion control systems. In *Advanced Motion Control, 1996. AMC '96-MIE. Proceedings., 1996 4th International Workshop on*, volume 1, pages 1–6 vol.1, Mar 1996. doi: 10.1109/AMC.1996.509370.

[15] M. Fischer and M. Tomizuka. Application and comparison of alternative position sensors in high-accuracy control of an x-y table. In *Advanced Motion Control, 1996. AMC '96-MIE. Proceedings., 1996 4th International Workshop on*, volume 2, pages 494–499 vol.2, Mar 1996. doi: 10.1109/AMC.1996.509298.

[16] Hu Jwu-Sheng Tomizuka Masayoshi. Synchronization of two motion control axes under adaptive feedforward control. *Journal of Dynamic Systems, Measurement and Control*, 11(0022-0434):pages, June 1992. doi: 10.1115/1.2896515. URL http://dx.doi.org/10.1115/1.2896515.

[17] Ho Seong Lee and M. Tomizuka. Robust motion controller design for high-accuracy positioning systems. *Industrial Electronics, IEEE Transactions on*, 43(1):48–55, Feb 1996. ISSN 0278-0046. doi: 10.1109/41.481407.

[18] Yoram Koren. Cross-coupled biaxial computer control for manufacturing systems. *Journal of Dynamic Systems, Measurement, and Control*, 102(4):265–272, 1980.

[19] Kira L Barton and Andrew G Alleyne. A cross-coupled iterative learning control design for precision motion control. *Control Systems Technology, IEEE Transactions on*, 16(6):1218–1231, 2008.

[20] Lennart Ljung. *System identification*. Springer, 1998.

[21] Karel J Keesman. *System identification: an introduction*. Springer, 2011.

[22] Münchhof Marco Isermann Rolf. *Identification of Dynamic Systems*, volume XXV. Springer, 2011. URL http://www.springer.com/physics/complexity/book/978-3-540-78878-2.

[23] Brian Armstrong-Hélouvry, Pierre Dupont, and Carlos 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.

[24] Henrik Olsson, Karl J Åström, Carlos Canudas de Wit, Magnus Gäfvert, and Pablo Lischinsky. Friction models and friction compensation. *European journal of control*, 4(3):176–195, 1998.

[25] Brian Armstrong-Helouvry. *Control of machines with friction*, volume 128. Springer, 1991.

[26] M. Tomizuka. On the compensation of friction forces in precision motion control. In *Motion Control Proceedings, 1993., Asia-Pacific Workshop on Advances in*, pages 69–74, Jul 1993. doi: 10.1109/APWAM.1993.316194.

[27] Chasparis G. *Design of a servomechanism control system with friction compensation.* NTUA, 2011.

[28] Evangelos G Papadopoulos and Georgios C Chasparis. Analysis and model-based control of servomechanisms with friction. *Journal of dynamic systems, measurement, and control*, 126(4):911–915, 2004.

[29] Bruno Siciliano and Wayne J Book. A singular perturbation approach to control of lightweight flexible manipulators. *The International Journal of Robotics Research*, 7(4):79–90, 1988.

[30] Alessandro De Luca. Dynamic control of robots with joint elasticity. In *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on*, pages 152–158. IEEE, 1988.

[31] Petar Kokotovic, Hassan K Khali, and John O'reilly. *Singular perturbation methods in control: analysis and design*, volume 25. Siam, 1999.

[32] K. Mitesh, K. Majid, and M. Manivannan. An adaptive-method for velocity estimation using time-to-digital converter. In *Field-Programmable Technology (FPT), 2011 International Conference on*, pages 1–4, Dec 2011. doi: 10.1109/FPT.2011.6132713.

[33] R. Petrella, M. Tursini, L. Peretti, and M. Zigliotto. Speed measurement algorithms for low-resolution incremental encoder equipped drives: a comparative analysis. In *Electrical Machines and Power Electronics, 2007. ACEMP '07. International Aegean Conference on*, pages 780–787, Sept 2007. doi: 10.1109/ACEMP.2007.4510607.

[34] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5, 2009.

[35] Sylvain Joyeux, Rachid Alami, Simon Lacroix, and Roland Philippsen. A plan manager for multi-robot systems. *The International Journal of Robotics Research*, 28(2):220–240, 2009.

[36] Gene F Franklin, Michael L Workman, and Dave Powell. *Digital control of dynamic systems*. Addison-Wesley Longman Publishing Co., Inc., 1997.

[37] Chi-Tsong Chen. *Linear system theory and design*. Oxford University Press, Inc., 1995.

[38] Norman S Nise. *CONTROL SYSTEMS ENGINEERING, (With CD)*. John Wiley & Sons, 2007.

[39] Jonathan Valvano. *Embedded Systems: Real-Time Operating Systems for ARM Cortex-M Microcontrollers*. CreateSpace Independent Publishing Platform, 2012.

[40] Jean J Labrosse. *Microc/OS-II*. R & D Books, 1998.

[41] Robert Love. *Linux system programming: talking directly to the kernel and C library*. " O'Reilly Media, Inc.", 2013.

[42] Christopher Hallinan. *Embedded Linux primer: a practical, real-world approach*. Pearson Education India, 2007.

[43] Doug Abbott. *Linux for embedded and real-time applications*. Newnes, 2011.

[44] Jonathan Corbet, Alessandro Rubini, and Greg Kroah-Hartman. *Linux device drivers*. " O'Reilly Media, Inc.", 2005.

[45] Phillip A Laplante. *Real-time systems design and analysis*. Wiley, 1993.

[46] DF Kurtulus, F Scarano, and L David. Unsteady aerodynamic forces estimation on a square cylinder by tr-piv. *Experiments in fluids*, 42(2):185–196, 2007.

[47] Gerasimos K Politis. Application of a bem time stepping algorithm in understanding complex unsteady propulsion hydrodynamic phenomena. *Ocean Engineering*, 38(4): 699–711, 2011.

[48] Wai Leung Chan and Taesam Kang. Simultaneous determination of drag coefficient and added mass. *Oceanic Engineering, IEEE Journal of*, 36(3):422–430, 2011.

[49] Arthur J Krener and Witold Respondek. Nonlinear observers with linearizable error dynamics. *SIAM Journal on Control and Optimization*, 23(2):197–216, 1985.

[50] Magnus Linderoth, Andreas Stolt, Anders Robertsson, and Rolf Johansson. Robotic force estimation using motor torques and modeling of low velocity friction disturbances. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 3550–3556. IEEE, 2013.

[51] J Edward Colgate and Kevin M Lynch. Mechanics and control of swimming: a review. *Oceanic Engineering, IEEE Journal of*, 29(3):660–673, 2004.

[52] Vassilios Tsounis, Michail Makrodimitris, and Evangelos Papadopoulos. A low cost, sensor-less drag force estimation methodology via measuring of motor currents. In *Advanced Intelligent Mechatronics (AIM), 2014 IEEE/ASME International Conference on*, pages 354–359. IEEE, 2014.

[53] Benjamin S Blanchard, Wolter J Fabrycky, and Walter J Fabrycky. *Systems engineering and analysis*, volume 4. Prentice Hall Englewood Cliffs, New Jersey, 1990.

[54] Ntouskas S. *Design of Towing Tank Carriage*. NTUA, 2011.

[55] Herbert Goldstein. *Classical mechanics*, volume 4. Pearson Education India, 1962.

[56] Cornelius Lanczos. *The variational principles of mechanics*, volume 4. Courier Dover Publications, 1970.

[57] Haim Baruh. *Analytical dynamics*. WCB/McGraw-Hill Boston, 1999.

[58] Reza N Jazar. *Advanced Dynamics: Rigid Body, Multibody, and Aerospace Applications*. John Wiley & Sons, 2011.

[59] Bruno Siciliano and Oussama Khatib. *Springer handbook of robotics*. Springer, 2008.

[60] George Keith Batchelor. *An introduction to fluid dynamics*. Cambridge university press, 2000.

[61] Hassan K Khalil and JW Grizzle. *Nonlinear systems*, volume 3. Prentice hall Upper Saddle River, 2002.

[62] Lorenzo Sciavicco and Bruno Siciliano. *Modelling and control of robot manipulators*. Springer, 2000.

[63] Bruce Carter. *Op Amps for Everyone, Fourth Edition*. Newnes, Newton, MA, USA, 4th edition, 2013. ISBN 0123914957, 9780123914958.

[64] Karl Johan Åström and Tore Hägglund. *Advanced PID control*. ISA-The Instrumentation, Systems, and Automation Society; Research Triangle Park, NC 27709, 2006.