MODEL-PREDICTIVE CONTROL OF A HYDRAULIC ACTIVE HEAVE COMPENSATION SYSTEM WITH HEAVE PREDICTION

by

Jeffrey Woodacre

Submitted in partial fulfillment of the
requirements for the degree of
Master of Applied Science

at

Dalhousie University
Halifax, Nova Scotia
August 2015

To my parents, who have always supported me.

Table of Contents

List of Tables

List of Figures

Abstract

This masters thesis presents the results of tracking ship heave motion with a heave prediction model-predictive controller (MPC) in the experimental actuation of a nonlinear, unloaded, full-scale active-heave compensation (AHC) hydraulic testbed. Implementing the MPC involves determining a system model for the AHC testbed and correcting for the nonlinear behavior of the AHC testbed. Multiple tuning parameters exist for MPC and so a single set of parameters was acceptably tuned and chosen to use for all experiments within this thesis work.

The experimental heave tracking results collected are compared to an AHC testbed simulator developed in MATLAB Simulink. A load is applied within the simulator to determine the AHC testbed response to operating under load conditions. For the experimental unloaded case, as well as the simulator loaded and unloaded cases, the MPC results are compared to a tuned PID controller in tracking of sine waves as well as four heave motion test cases.

The heave prediction MPC controller is found to track the test cases and sinusoidal references well while, additionally, outperforming the tuned PID controller in real-world experiments for all test cases and sine wave tracking. Two of the test cases introduced relatively high frequency components to the reference signal which the MPC is able to track, while the PID performance decreases dramatically with the addition of these high frequency components. Maintaining constant tuning parameters for each, the MPC is shown to be more robust for a range of operating conditions when compared to PID.

Within the simulator the MPC controller performance is reduced compared to the experimental testbed performance while the PID controller is able to better track two of the four test cases. The loss in MPC performance is attributed to different implementations of MPC between the simulator and the experimental setup. Applying a load within the AHC testbed simulator shows two important results: first, that a counter-balance valve is necessary for the AHC testbed system under load conditions, and second, a parallel integral controller may be needed with the MPC controller to ensure motor leakage does not affect performance.

List of Abbreviations and Symbols Used

Symbol	Description
A_i	Heave Amplitude for the i^{th} mode
a	Valve Opening Area
AC	Alternating Current
AHC	Active Heave Compensation / Compensator
B	Heave Mode State Matrix
$\mathbf{C}$	Output Matrix
ccm	Cubic Centimeter
C_d	Discharge Coefficient
cm	Centimeters
CoM	Center of Mass
D	Motor Displacement
D_p	Pump Derivative Constant
e	PID Error Term
f	Frequency [Hz]
FFT	Fast Fourier Transform
GPS	Global Positioning System
h	Heave Motion
IMU	Inertial Measurement Unit
in	inches
HPU	Hydraulic Power Unit
Hz	Hertz
I	Pump Integral Constant
K_p	Experimental Proportional Gain
kg	kilograms
L	Liters
$\mathbf{L}$	Heave Observer Gain Matrix
lbf	Pound Force

LQR	Linear Quadratic Regulator
m	Meters
min	minute
MPC	Model-Predictive Control / Controller
MRU	Motion Reference Unit
N	Newtons
N_c	Control Horizon
N_p	Prediction Horizon
Nm	Newton Meter
p	Pressure
P	MPC Control Action Weighting Parameter
P_p	Pump Proportional Constant
Pa	Pascals
PD	Proportional-Derivative
PHC	Passive Heave Compensation / Compensator
PI	Proportional-Integral
PID	Proportional-integral-derivative
psi	Pounds Per Square Inch
PV	Process Variable
Q	MPC State Weighting Parameter
q	Flow Rate
QP	Quadratic Programming
R	MPC Change in Control Action Weighting Parameter
rad	radian
rev(s)	Revolution(s)
RPM	Revolutions Per Minute
rps	Revolutions Per Second
RRC	Rolls-Royce Canada Limited
s	Seconds
StdDev	Standard Deviation
SP	Setpoint

t	Time
T_d	Experimental Derivative Time Constant
T_i	Experimental Integral Time Constant
T_{pred}	Time In The Future For Heave Prediction
TF	Transfer Function
ΔT	System Sampling Time
u	Control Action
USgpm	United States Gallons Per Minute
USV	Unmanned Surface Vehicle
v	Heave Offset
VFD	Variable Frequency Drive
v_{in}	Voltage Into Deadband Correction
V	Volts
x	Model States
ϕ	Phase Estimate
ρ	Fluid Density
ω	Angular Velocity
Φ	Discrete System Matrix
θ	Angle [rad]

Acknowledgements

The author would like to thank Dr. Bauer and Dr. Irani for all of their excellent support, guidance, and motivation throughout the course of this thesis work.

Funding was provided by Rolls-Royce Canada Limited (RRC) and the Atlantic Canada Opportunities Agency (ACOA).

Chapter 1

Introduction

The conditions when working at sea are well known to be treacherous. A corrosive salt environment, unpredictable weather, and vessel motions are just some of the issues encountered when working in what can be one of the most hostile operating environments on the planet. These issues as well as innumerable others need to be understood and planned for by any engineer who works in the field of ocean engineering to ensure the safety of each individual who works at sea. An important area of research in ocean engineering focuses on reducing the potential dangers associated with work performed at sea during high sea states; or, in other words, work performed during large ship motions.

As early as the 1960's research can be found regarding the effect of wave induced ship motion on cable suspended loads at sea [1]. It was shown that for large suspended loads being lowered to the ocean floor the suspending cable could theoretically break despite being designed with a reasonable static safety factor. When a cable under tension breaks there are two main concerns: first, the load is often lost or destroyed, and may fall near crew causing bodily harm; second, the snapped cable can retract quickly, whipping around and impacting both equipment and personnel, again causing both equipment damage and bodily harm. The reason a cable can theoretically break despite having a reasonable static safety factor is because of resonance in the load-cable system caused by heave motion of the surface vessel which can increase cable tension by upwards of 100 times [1]. A simple solution to avoid cable resonance would be to only perform lifting operations when the sea is calm; however, this sea state requirement could extend operations for days or weeks during storm seasons thus adding delays and becoming prohibitively expensive. Over-designing of the cable and lifting system could also be an option but this would significantly increase both weight and size of the lifting system — again increasing costs by requiring more power, more deck space, and potentially a larger vessel. Since most lifting operations at sea take place using a crane or winch, a more elegant engineering option has been introduced using a system which decouples the motion of the heaving ship from the motion of a suspended load, thus avoiding large increases in cable tension. This decoupling of load motion from ship heave motion is commonly known as heave compensation.

Within the field of modern ocean engineering, heave compensation is a broad term which can describe the decoupling of vessel motion in all three linear axis from either a physical load, as previously discussed, or a sensor measurement such as a depth dependent temperature profile. For the scope of this thesis, only decoupling of a physical load in the vertical direction will be considered. A physical load such as a towed sonar array can be decoupled from ship motion in two ways: through passive decoupling or active decoupling. Passive decoupling is a strictly mechanical process in which a spring-damper system is inserted in-line with the load as seen in Figure 1.1. The spring constant k and the damping constant c in Figure 1.1 are tuned to shift the resonant peak of the cable-load system outside of the wave frequency spectrum, as shown in Figure 1.2. Shifting the system resonance peak away from the wave frequency spectrum ensures wave motion cannot induce resonance in the system. Although desirable for their simplicity, passive heave compensators (PHCs) have been shown by Hatleskog and Dunnigan [2] to be no more than 80% effective in decoupling ship heave motion from load motion, meaning if a vessel were to heave vertically 1 m, then the load motion would be reduced by 80% to 0.2 m. Alternatively, an active heave compensation (AHC) system tested by Kuchler *et al.* [3] was able to achieve a 98% reduction in heave motion. An active heave compensator is a system which uses powered actuation of the load to ensure that load motion is decoupled from ship motion at all times. The focus of this thesis will be active heave compensation.

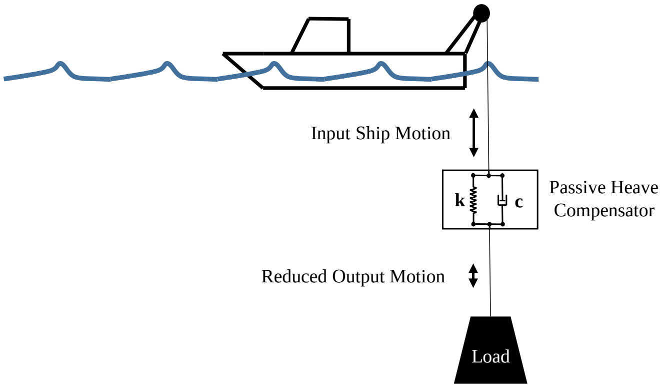


Figure 1.1: This schematic shows an example of a small vessel hauling a load using a passive heave compensator in line between the load and the vessel.

In Figure 1.3, a small ship is shown heaving vertically on a wave with either a crane or winch system supporting a load underwater. The bolded load depicts how

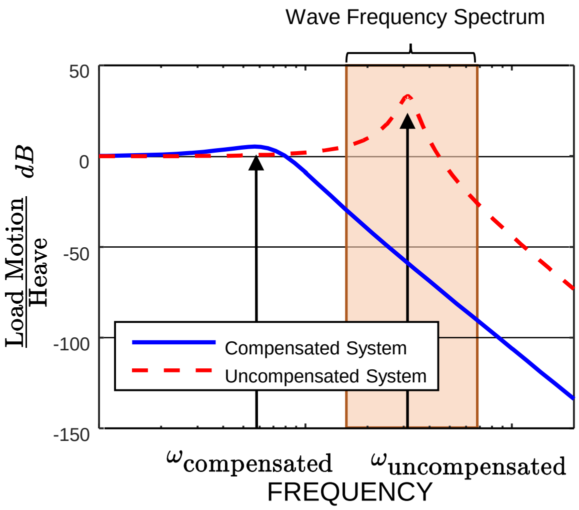


Figure 1.2: These Bode diagrams show an uncompensated (or poorly compensated) system operating within the wave spectrum as the dashed line, and a heave compensated system attenuating motion in the ocean wave spectrum as the solid line.

the load position would follow ship motion exactly without an AHC system. Having the load rigidly fixed to the ship would result in large dynamic loading of the cable due to forces required to move the load inertia. Alternatively, the grey dashed boxes show the load motion when an AHC system is being utilized. The cable supporting the load is reeled in or out as the ship heaves down or up, thus ensuring the load remains at the desired depth. With the AHC enabled the load remains vertically stationary from a fixed reference point and as there is no acceleration of the load so cable tension remains constant.

In a practical experimental system, however, it is impossible to totally decouple load motion from ship motion. The difficulties in decoupling load motion from ship motion using an AHC include issues such as: delays in measuring real-time ship motion, non-linear dynamics in both the load and winch system, inaccuracies modeling the system, and potential variations in load as in the case of tethered undersea autonomous robotics or oil drilling rigs. One important point avoided by many researchers is the non-linear properties of some actuators. Many AHC systems use hydraulic actuation as hydraulic power is commonly available at sea and is capable of handling very large loads. Although it is possible to obtain linear-response hydraulic components, they can be expensive and are generally less physically robust and require a cleaner environment than their less expensive, non-linear counterparts. It is important then, when studying AHC designs, to consider the non-linear hydraulic components as they are more practical for at-sea implementation. Unfortunately, the non-linearities in more robust hydraulic hardware can contribute significantly to the inability of an AHC to fully decouple vessel motion from load motion; so, in an at-sea system the non-linearities must be quantified and corrected for. *Given this observation, the first key objective of this thesis is to examine, quantify, and correct for the non-linear properties of low-cost hydraulic components within a hydraulic AHC system.*

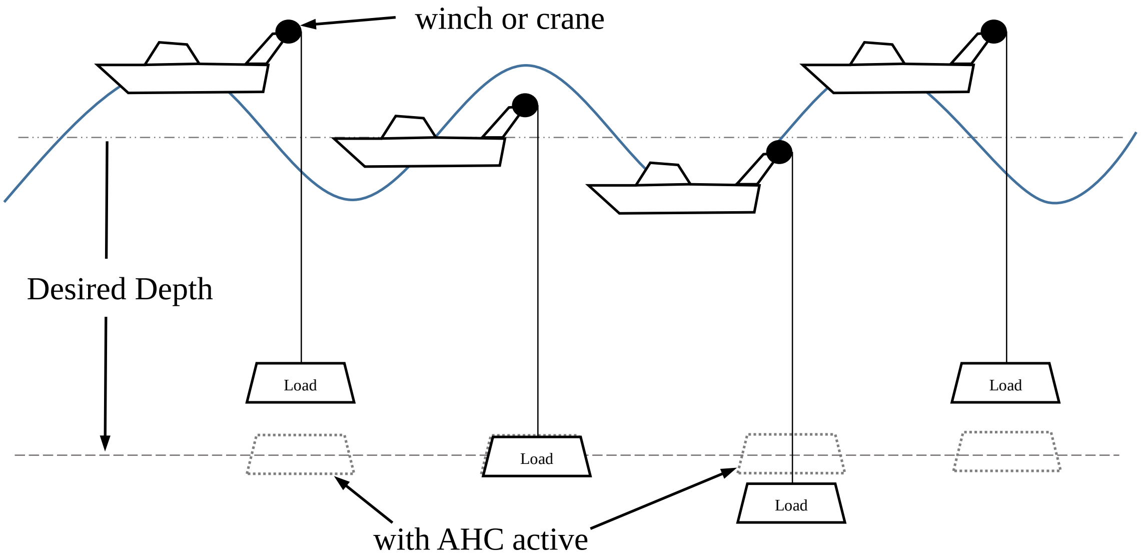


Figure 1.3: This schematic shows a vessel heaving vertically on an ocean wave. The dark load suspended underwater follows the vessel motion, showing the AHC disabled. With the AHC activated, the grey outlined load depict the load maintaining a constant depth.

Another important issue which will be addressed in this thesis is the use of predictive controllers as opposed to reactive controllers. In a review of the current state of AHC controller design it was found that most research groups develop reactive controllers based on PID, feedforward, state-space, or some combination of those three controller types to create an AHC system. Kuchler *et al.* [3] published a paper titled "Active Control for an Offshore Crane Using Prediction of the Vessels Motion"[3] where the authors implemented a wave-prediction algorithm to help correct for transport delays between actual ship motion and measurement of ship motion. Although they were successful in correcting for a time-delay within the system, one could go even further with their wave prediction algorithm and implement a control scheme which would use the future wave prediction data along with a model-based controller to predict future control actions. *Given this observation, the second key objective of this thesis is to implement a model predictive controller (MPC) combined with the wave prediction algorithm used by Kuchler et al. [3] to achieve improved decoupling of load motion from ship motion when compared to a more conventional reactive controller.*

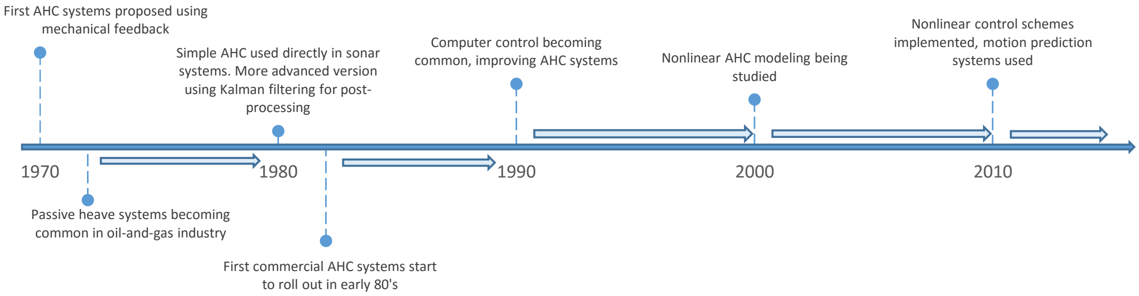
Additionally, most published works focus on simulation without physical validation using neither a reduced scale model nor a full-scale testbed. *It is, therefore, the third key objective of this thesis to physically validate an MPC controller design using a full-scale hydraulic testbed under zero load conditions for the implementation of an AHC system. A MATLAB Simulink model of the hydraulic testbed will also be created and used to compare MPC simulation results with results obtained from the hydraulic testbed.* The resulting controller design and Simulink model from this work could be used to assist design engineers in implementing and testing a hydraulic MPC controlled AHC system on vessels at sea under various conditions.

This thesis is divided into six chapters. Chapter 2 contains a literature review of AHC systems leading up to the current state-of-the-art in the field. Chapter 3 examines the testbed which was developed and used in this thesis work, describing in detail the hydraulic hardware used and identifying the non-linearities associated with the system components. Chapter 4 focused on the development of an MPC system and its implementation within LabVIEW for deployment during testing. The MPC was operated tracking sine waves, as well as four test cases, and was compared to an tuned PID controller for each situation. Also, the robustness of the MPC controller was examined and compared to the PID controller. In Chapter 5 a model was created in MATLAB Simulink of the hydraulic test equipment based on the results of Chapter 3 and Chapter 4. The results from Chapter 4 are compared to an MPC and a PID controller within the simulator environment and then the simulator will be modified to run at full load, representing operating conditions of a full AHC system. Chapter 6 summarizes the results and contributions of this thesis and suggests potential improvements and future work.

Chapter 2

Active Heave Compensation

The past 40 years have seen active heave compensation (AHC) systems become commonplace in many maritime operations. Figure 2.1 provides a brief timeline of AHC development starting with the first strictly mechanical AHC systems and ending with modern non-linear controlled AHCs. In this chapter, Section 2.1 will discuss the overall progression of AHC research starting in the early 1970's up to the current state of the field. Section 2.2 provides an overview of the two most common methods of actuation used for AHC systems — hydraulic actuation and electric actuation. In Section 2.3, the control methods applied to AHC are presented and discussed and finally, Section 2.4 summarizes the previous sections and concludes by suggesting that a hydraulic test bed combined with the use of model predictive control and wave prediction are a reasonable avenue to pursue in AHC research. This chapter is comprised mainly of excerpts from the review paper entitled "A Review of Vertical Motion Heave Compensation Systems" by J.K. Woodacre, R.J. Bauer, and R.A. Irani published August 2015 in Ocean Engineering [4].



First AHC systems proposed using mechanical feedback
Simple AHC used directly in sonar systems. More advanced version using Kalman filtering for post-processing
Computer control becoming common, improving AHC systems
Nonlinear AHC modeling being studied
Nonlinear control schemes implemented, motion prediction systems used
1970
1980
1990
2000
2010
Passive heave systems becoming common in oil-and-gas industry
First commercial AHC systems start to roll out in early 80's

Figure 2.1: An approximate timeline of heave compensation development.

2.1 Active Heave Compensation Research

Active heave compensation systems involve closed-loop control and require energy input. In an active system, ship heave motion is measured and relayed to a controller, which then moves an actuator to oppose the heave motion. So, if a ship heaves upward, the controller commands the load to move downward that same amount. For an active system, one of the greatest advantages is that the feedback variable is not limited to ship heave motion. Feedback can, for example, be based on the separation between two ships such as is used during payload transfer, or it can be a measured force from a load cell used to maintain a constant tension in the cable at all times during a lifting operation. Feedback can also be based on wave height which is most often used when a load transitions from air to water.

One of the first active heave systems was designed by Southerland in 1970 [5] where a spring-loaded tether was attached from a crane-boom on one ship to the deck of a second ship. A schematic of this system can be seen in Figure 2.2. As the tether pulled in and out, it moved a hydraulic proportional valve which adjusted the load, maintaining a constant height from the deck. The system shown in Figure 2.2 was fully integrated into the crane operation. A similar mechanically actuated system was patented in 1977 [14] but the system was packaged for retrofit onto cranes which were not heave compensated and could be hung from the crane, between the crane and the load.

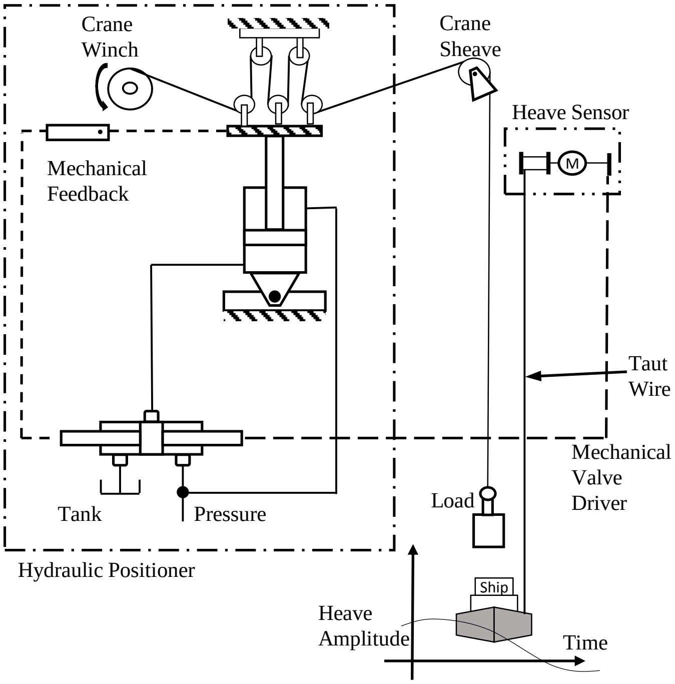


Figure 2.2: This system was presented by Southerland in 1970 as a method to transfer payload from ship-to-ship in the presense of significant waves. Figure reproduced from [5].

Little published work is found between 1980 and 1990 on mechanical AHC systems — likely because this time period occurred before real-time computer control was mature enough to integrate into a complicated system. Furthermore, in the 1980's passive systems were generally sufficient for the oil and gas industry, who were one of the main driving forces for initial heave compensation research. A patent by Barber in 1982 [15] does show a circuit-based AHC system where heave motion was sensed and a fixed circuit design was implemented to control heave motion, but a downside of the fixed circuit is that it cannot be changed. If control scheme changes need to be implemented it would require rework of the circuit board. So, although published works were sparse in this time period with respect to mechanical AHC systems, work on heave compensation theory and algorithm development did continue in the sonar field.

A patent by Hutchins [7] shows how a simple double-integrator circuit was used to convert accelerometer data into vertical motion data as part of a towed sonar array control circuit. In this case, the sonar array was used for mapping the ocean bottom. Having vertical position data allowed the sonar array to adjust the sonar pulse timing, effectively correcting for vertical motion on-board and demonstrating an early example of transitioning from mechanical feedback to electronic feedback in an AHC system (before computer control became dominant).

An improved method of correcting heave in sonar data was presentd by El-Hawary in 1982 [8]. The author analyzed sonar data using Fast Fourier Transform (FFT) analysis to determine the frequency components of ship heave and, through application of an optimized Kalman filter, was able to selectively remove heave motion in post-processing while retaining the ocean bottom profile. Due to the computation powered required, analysis could not be applied in real-time at the time of publication.

A patent granted in 1990 to Jones [16] is one of the first examples the present researcher could find of a microprocessor controlled AHC system. As it is a patent, details on the control method are limited; however, a patent by Hatleskog and Robicheux [12] does suggest the benefits of a microprocessor come mainly from adaptability. With mechanical hardware in place, the control parameters or control method can be changed by uploading new software to the controller. Operators could easily adjust control parameters on-the-fly, accounting for a wide range of loads or ocean conditions. The ability to modify software would be significantly less expensive than hardware changes, while also broadening the use of the control system so that it could potentially be used on large oil rigs, or adapted for smaller vessels which may want to use AHC for remotely operated vehicles. Software could also be written for accepting different sensor inputs depending on the AHC application which is appealing to users who may have multiple uses for an AHC system.

Large drilling rigs were one of the first adopters of AHC systems, as these AHC systems allowed the rigs to drill in a much wider range of weather conditions. When drilling at sea, there are a number of drilling vessel types — either floating or fixed in place — performing drilling operations at various depths. In the case where a vessel is floating, it is important to remove vessel heave motion from the entire drill string, where drill string is a term which often describes the entire drilling system from the ship down to the drill bit. Removal of heave motion from the drill string extends operational time and reduces fatigue on the drill and riser [17]. In 1998, Korde [17] performed an in-depth mathematical treatment of an AHC system used to stabilize the drill string for a drill ship. In his system, accelerometer data was used for position and force feedback in a hydraulic active position control system as well as an active vibration absorber. A more in-depth discussion of the system by Korde [17] will be performed in Section 2.3; however, note that simulation results show the system is able to fully decouple motion using a linear model. In 2008, Do and Pan [13] applied a nonlinear model and control scheme to actively compensate for heave motion in a similar drill string system to that which was examined previously by Korde [17]. In using a nonlinear model, Do and Pan [13] were unable to fully decouple ship heave from the drill string suggesting that using a linear system model may be too simplified to capture the full system dynamics.

Requiring more than simple acceleration measurements, modern systems often use an inertial measurement unit (IMU), also called a motion reference unit (MRU), to determine ship motion in real-time. Using 3-axis accelerometers, gyroscopes and potentially GPS as well as magnetometers an IMU determines ship motion based on algorithms similar to those presented by Godhaven [18] in 1998. Marine IMU's tend to be expensive to purchase, thus a promising low-cost GPS based alternative for measuring heave was presented in a paper by Blake *et al.* in 2008 [19]. Preliminary results show heave measurements with their device are comparable to those obtained from an IMU; however, the sampling rate of the GPS at the time of publication was limited to below 4 Hz which could be a concern when implementing high-speed control algorithms.

Control algorithms in an active heave system can be as simple as basic PID and pole-placement control, or as advanced as systems using Kalman filtering and observers to include complicated features like tether dynamics as part of the control scheme. In any control system, corrections for the inherent lag, perhaps introduced by the hydraulic system or through slow communication between the IMU and the control system, must be made to ensure ideal control. A system by Kyllingstad [20] for example, applied transfer function filters to correct for time and phase lag in their overall system. Alternatively, Kuchler *et al.* [3] used heave-prediction algorithms to predict vessel heave motion based on previous measurements and then applied control action based on these predicted motions. Now, as more advanced algorithms and better sensors are included in AHC systems, control quality improves; however, there are disadvantages to the inclusion of more advanced components such as increasing design and production cost as well as potentially introducing the need for specialized training for troubleshooting and repair.

2.2 Actuation

Primary actuation of most heave compensation systems is delivered by either hydraulic or electric drive systems. There are benefits and detriments to using both which will be discussed in the following sections.

2.2.1 Electric

An article in Offshore Magazine [21] mentions that alternating current (AC) driven heave compensation systems were introduced in the early 1990's. Electric heave compensation systems have increased in popularity due to their relatively high efficiency (estimated between 70% and 80% peak) [22] attributed to efficient control and motor systems as well as regenerative techniques used during braking [23]. Lack of an oil reservoir and low motor noise when compared to hydraulic systems is also appealing to consumers [22] who may not want to deal with oil replacement, potential leaks or fire hazards.

High power electric AC motors tend to be physically large, having a correspondingly large moment of inertia. A large inertia means large torques are needed to change motor speed when responding to transient behavior. In some situations it could be that, when changing speed, it is the motor inertia which dominates the required power, not the load itself.

The active heave system shown in Figure 2.3 uses an AC electric variable frequency drive (VFD), AC induction motor or motors, gearbox, sensor feedback and control system, as well as a braking system and potentially a cooling system. In an AC induction motor the motor speed is directly proportional to the supplied AC voltage frequency as described by the equation:

$$\omega_m = \frac{120f}{z} \tag{2.1}$$

with ω_m being motor speed in revolutions per minute (RPM), f being the AC voltage frequency in Hertz (Hz), and z being the number of motor poles. A VFD creates an AC voltage signal where the user may adjust the output frequency to drive the AC motor at an angular velocity as described in Equation 2.1.

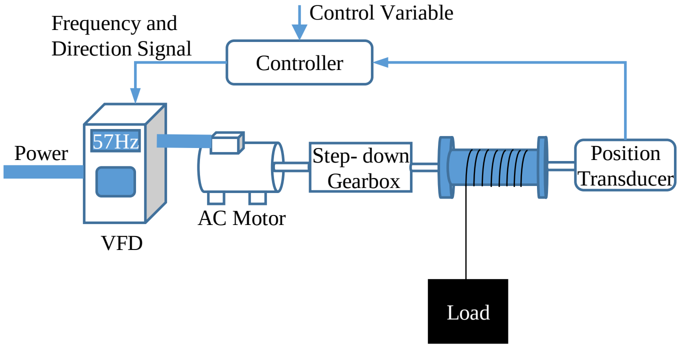


Figure 2.3: A simple AC drive winch system with feedback control.

If multiple actuators are needed, or multiple winches are to be installed, then the entire system must be replicated in full for each actuator as shown in Figure 2.4 where the system from Figure 2.3 has been replicated three times to create a multiple winch system. Replication of the full system is not ideal because the AC motors are large when compared to an equivalent power hydraulic motor. As an example, the Marathon Electric E213 100 horsepower electric motor weighs 1220 lbs [24], while the hydraulic Bosch-Rexroth MCR20 110 horsepower motor weighs 167 lbs [25].

The first alternating current electric AHC systems were likely powered by a VFD known as a scalar VFD. A scalar VFD maintains a constant voltage to frequency ratio to correct for reduced motor impedance at lower frequencies. A reduced impedance means that a lower voltage is required to maintain equivalent current and, therefore, torque. Scalar VFD's could lose torque during rapid speed changes forcing designers to oversize both the physical system and the power system [26]. Systems using a scalar VFD can provide their designed torque at a constant low speed [27]; however, for high-torque low-speed applications additional cooling is generally required for the motor since most AC motors rely on a fan directly connected to themselves to provide cooling. Additional cooling can be achieved by the addition of an externally driven fan or through fitting of the AC motor with an encasement and providing a water cooling system — both of which increase the total cost.

Modern VFD systems can now use vector control, also called field-oriented control, which more efficiently controls power delivery to the motors, resulting in better control and reducing the need to oversize motors [26]. Vector control also integrates

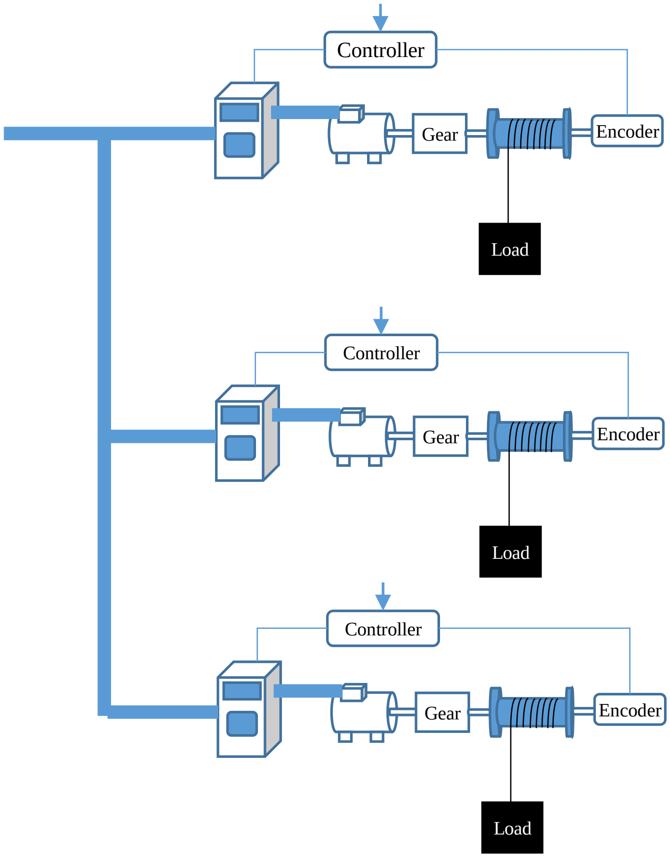


Figure 2.4: Multiple AC electric winches would require full duplication of the system in Figure 2.3.

regeneration into the electronics, allowing energy capture when decelerating, thereby increasing system efficiency. A current issue with energy capture in VFDs is storing the energy because if power is pushed into a ship's electrical grid when it cannot be used this excess power may disrupt other systems. Battery or capacitor bank storage is, therefore, needed which increases cost due to increased weight as well as additional storage space requirements. Reducing weight is one of the biggest design concerns on a ship which is why hydraulic systems can be very appealing.

2.2.2 Hydraulic

Hydraulic systems are well established in the marine industry. Hydraulic systems can be used for anything from opening large doors on a marine vessel to a simple winch on a fishing boat. As shown in Figure 2.5, hydraulic actuators provide the highest power to weight ratio of any actuator on the market as of 2010 [28]. This figure is incomplete because larger weight AC motors are not included; for example, the Marathon Electric E213 100 HP motor mentioned previously would appear at the star to the right of the AC motor block in Figure 2.5.

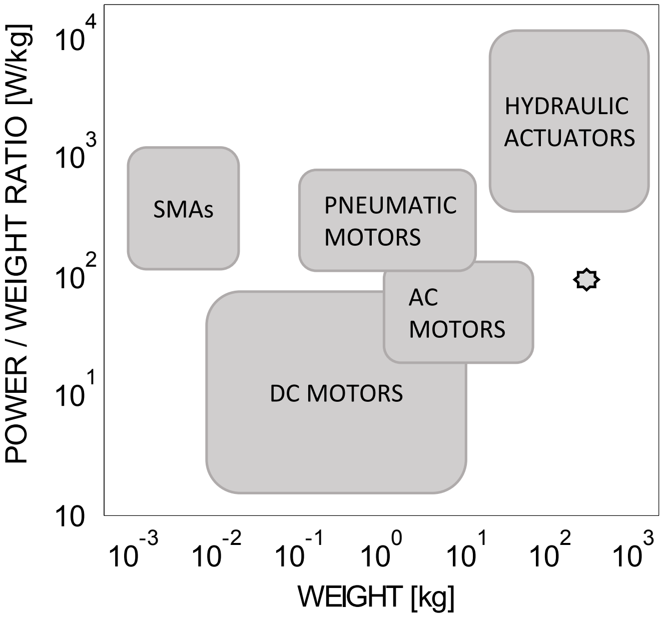


Figure 2.5: This Figure, reproduced from [28], shows that hydraulic systems can provide a higher actuator power density than AC drives. Here, SMA is short for shape-memory alloy.

The high power to weight ratio of hydraulic motors allows the actuator to maintain a small footprint at the point of actuation which can be appealing when deck space is limited. The downside to using hydraulic actuators is that a hydraulic power unit (HPU) must be placed somewhere aboard the ship. These HPUs can be large depending on the loads in question; however, it should be noted that one HPU can operate multiple actuators as shown in Figure 2.6. In Figure 2.6 each motor can be operated independently by operating their respective directional valves.

As mentioned, hydraulic systems are a well known and widely-used technology in the marine industry. Parts can be readily available so troubleshooting and repair of a hydraulic system can often be done quickly. In contrast, troubleshooting of electric systems can be more difficult and require specialized electrical training [22].

Figure 2.7 demonstrates two simple hydraulic circuits operating a motor. The upper circuit is an open-loop circuit, where fluid from the pump is regulated by a directional-valve as it travels to a motor, performs work, and returns to the open-air

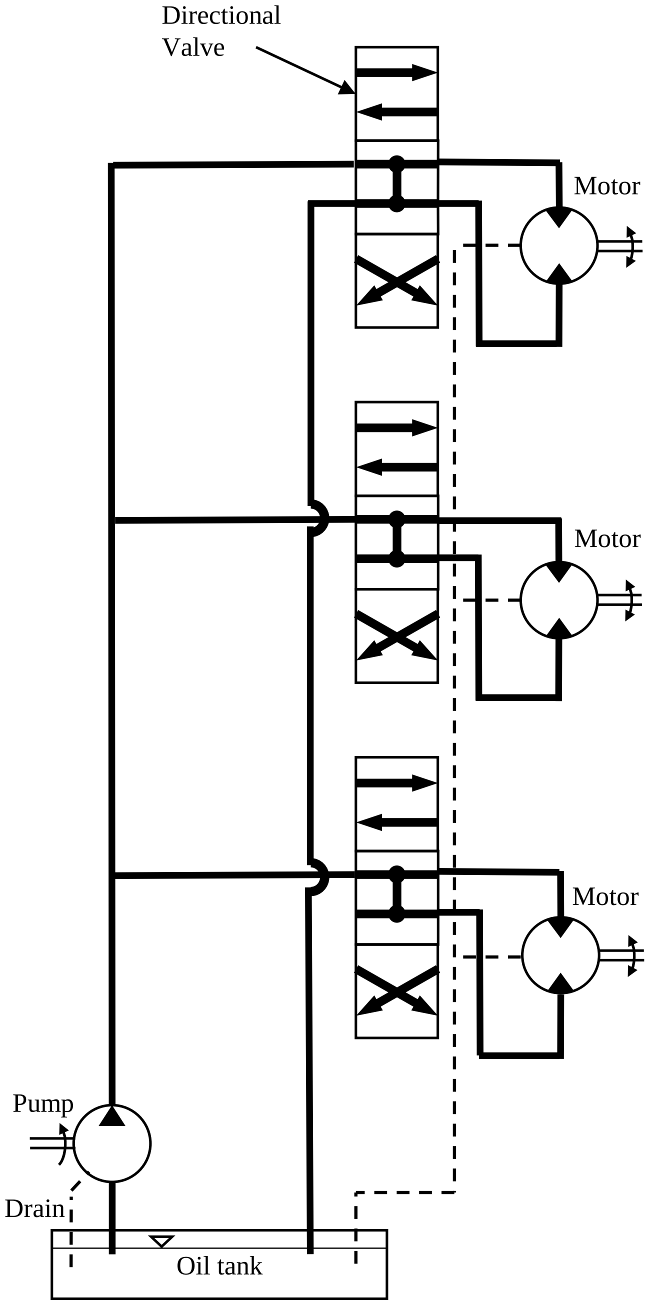
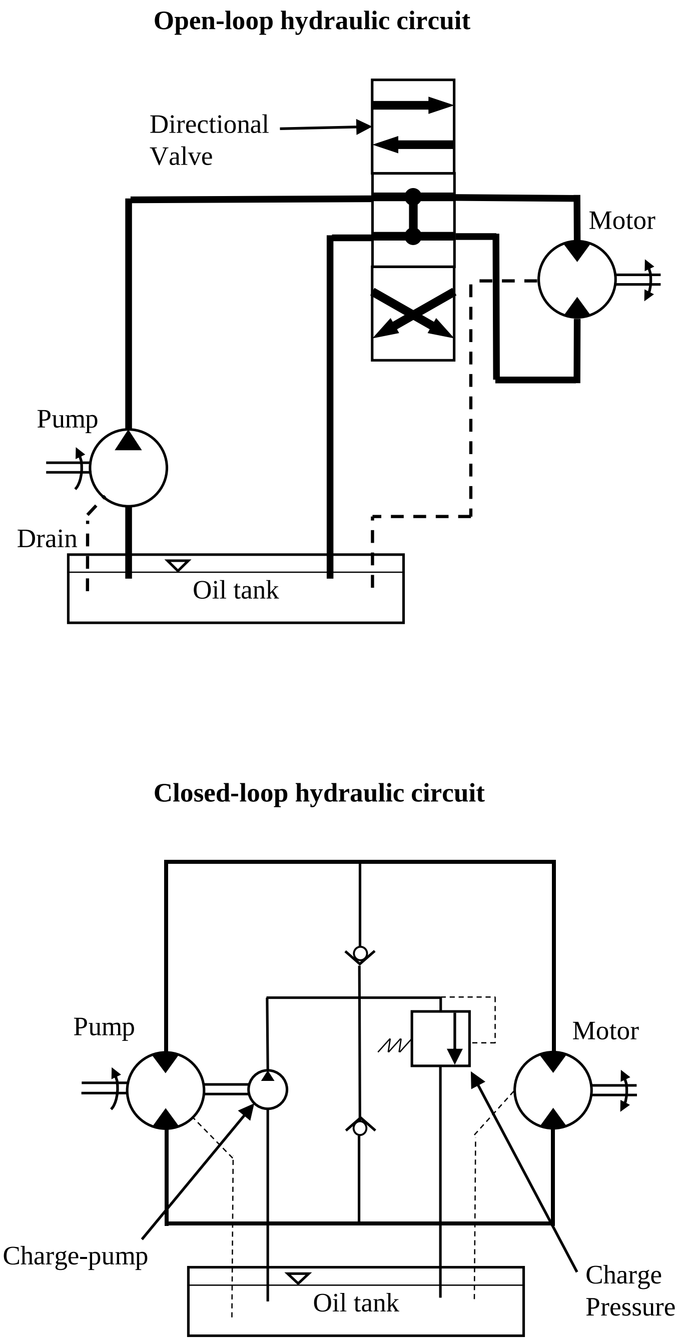


Figure 2.6: A single hydraulic pump can operate multiple motors; however, care must be taken if trying to operate each motor at the same time

reservoir. The lower circuit in Figure 2.7 is known as a closed loop circuit as fluid is regulated by the pump itself, travelling directly to the actuator, then returning to the pump. In a closed-loop system, the pump is able to provide flow in both directions, whereas an open-loop pump only provides flow in one direction.



Open-loop hydraulic circuit

Closed-loop hydraulic circuit

Figure 2.7: A simple open-loop hydraulic system (top), and a closed-loop hydraulic system (bottom) operating what could be a winch motor.

In an open-loop system, and hydraulic systems in general, the most significant downside is low efficiency. Depending on the design and operation, some open-loop systems can have an average efficiency as low as 10 to 35% [29]; however, efficiencies as low as these generally occur when operating a system far from maximum load. The lowest efficiency systems use a fixed displacement hydraulic pump delivering constant flow. Unused flow is diverted away from the load at significant energy cost, and a proportional valve controls how much useful flow is delivered to the motor. For a system which will only operate for short periods of time a fixed displacement pump may be acceptable — trading efficiency for simplicity, low initial cost of hardware, and ease of maintenance. In larger systems or systems which may run for extended periods of time, inefficiency can be very costly; therefore, a variable displacement hydraulic pump is preferred. Variable displacement pumps only deliver fluid when needed — better matching the process requirements and avoiding losses from dumping excess flow away from the load. A proportional control valve is used to moderate flow delivered to the load, however note that some proportional control valves may have a number of non-linear properties which need to be corrected for when used within a control loop. In systems using a variable displacement pump the most significant energy losses come from metering across the proportional control valve, and from pump and motor inefficiencies. These losses will be system specific and dependent on the stand-by pressure of the pump (where stand-by pressure is the pressure a variable displacement pump maintains when flow is not demanded). It would not be unreasonable to see efficiency numbers between 50% and 80% for a system using a proportional valve and a variable displacement pump. An alternative to having a proportional control valve is to use a closed-loop hydraulic system.

An efficiency of at least 80% can be realized in closed-loop systems [30]. Further efficiency increases can be realized when variable speed control is included on the closed-loop pump — reducing mechanical losses when flow is not required. Increased efficiency is enticing for designers; however, a closed-loop system has increased cost as a dedicated pump and motor are both needed for each actuator to operate independently at high efficiency.

In closed-loop cases, actuator speed is linearly controlled by pump output instead of the nonlinear response found in most proportional control valves which simplifies the control system for AHC. Increased cost for a closed-loop system, however, means proportional control valves are still commonly used and, as such, it is important to be able to model and control these valves and their systems accurately. In the next section, various control methodologies for active heave systems are examined.

2.3 Control

Using an AHC system, the goal is to actively remove as much of the ship heave motion as possible from the load or, in other words, to decouple ship motion from load motion using controllers and actuators. In 1970, one of the first AHC systems was presented by Southerland [5] using proportional control with mechanical feedback in a payload transfer situation. Recalling Figure 2.2, this mechanical feedback consists of a tether attached from a crane tip on one ship to the deck of a second ship. Motion of the second ship resulted in the tether pulling in, or letting out, moving a hydraulic valve either pulling the load up or letting it down. The work did not give experimental results on how effective the system was.

A report by Bennett in 1997 [31] mentions that a system used in the North Sea was able to reduce motion of 6 to 7 foot swells down to less than a 2 inch motion based on visual inspection — which is a 95% reduction. They do not, however, mention the type of control used, simply labelling the controller as a “computer”. The report by Bennett [31] presented results of implementing an AHC system which was purchased from a supplier, so it is reasonable that they would not know or be able to present the type of control used. In this case, the company supplying their AHC system would be unlikely to reveal the control algorithm.

As mentioned, the work by Southerland [5] presents a system idea, and the work by Bennett [31] presents final results of a system without details of the system itself. Often, if a group has funding to construct or purchase the experimental apparatus they may not want to fully reveal the design to protect their intellectual property. Due to the prohibitive cost in construction of an experimental apparatus, much of the work found in the literature presents a design, or a design with simulated results only.

In a 1998 paper, Korde [17] presented a full linear drill-string model and developed a control system using accelerometers and an actuated harmonic absorber. Figure 2.8 shows the actuated part of Korde’s system with the central actuator acting on M_m (the vibration absorber) while the other two actuators act on M_c (where M_c combines the mass of the drill string and the block holding the string to the actuators). Korde’s system applies feed-forward control based on direct accelerometer measurements to control the vibration absorber, as well as double-integrating the accelerometer data for position control of both sets of actuators. This type of vibration absorber is similar to that used in multistory buildings to reduce seismic and wind vibration [32]. Theoretical results show that this system can fully decouple load motion from

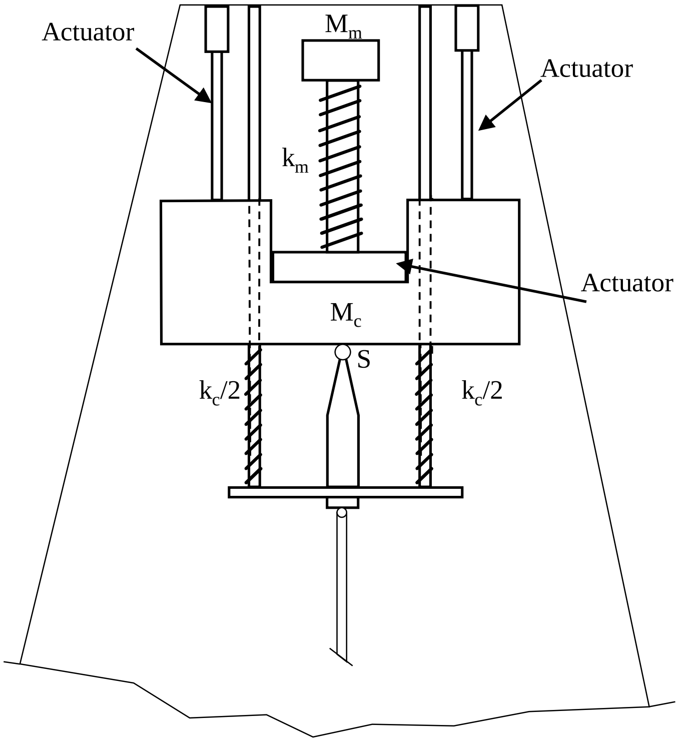


Figure 2.8: Here, the actuated portion of Korde's [17] system is shown. The harmonic absorber M_m and the support block M_c are shown with their respective actuators. Figure reproduced from [17].

ship motion; however, the theoretical full decoupling results are based on idealized calculations and the author mentions that a real-world system may require online estimates of system parameter changes to obtain ideal controller performance.

Time domain simulations of a similar vibration absorber system were presented by Li and Liu [33] in 2009, where the authors used a linear quadratic regulator (LQR) to actuate the vibration absorber and the block holding their drill string. An LQR controller is a state feedback controller which optimizes controller gains by solving a quadratic minimization problem. The optimization is based on weighting parameters. Li and Liu [33] were able to show a heave motion decoupling of up to 84% with the potential to achieve further decoupling with additional iterations of weighting parameters in the LQR system.

Built upon a similar linear drill-string model as used by Korde [17], Hatleskog and Dunnigan [34] derive a linear transfer-function model for an active-passive hybrid system using feedfoward control on displacement (as opposed to Korde who used acceleration) as well as a PD feedback loop with respect to actuator position. The Hatleskog and Dunnigan system is mechanically simpler as a vibration absorber is not used in this case. The design and considerations for Hatleskog and Dunnigan's [34] system are presented, but not simulated or implemented in their paper. Hatleskog and Dunnigan expect the system to be 90% to 95% effective, attributing any deviations from 100% to potential sensor error. It should be noted that Hatleskog and Dunnigan discuss using a closed-loop hydraulic system, as mentioned in Section 2.2.2 of this paper, to ensure a linear system response. A linear response, meaning that the actuator motion is directly proportional to the control signal, makes control design much less complicated.

In both the paper by Korde [17] and the paper by Hatleskog and Dunnigan [34], friction is considered linear. This assumption is rarely accurate in real-world applications, but is often used for simplicity. Pan and Do in 2008 [13] correct for any linearized friction inaccuracies by modeling the total force on their hydraulic actuator as:

$$m_H \ddot{x}_H = A_H p_H - b_h \dot{x}_H + \tilde{\Delta},$$

where $m_H \ddot{x}_H$ represents the total force on their actuator, $A_H p_H$ models the force due to hydraulic pressure, $b_h \dot{x}_H$ models linear friction, and $\tilde{\Delta}$ is a state dependent disturbance term meant to account for nonlinear friction and other unmodeled forces. In this case the disturbance is not measurable so an observer is used. Additionally, Pan and Do build their system model to include a proportional control valve which, due to a flow across the valve being proportional to $\sqrt{\Delta p}$ where Δp is the pressure drop across the valve, the system is inherently nonlinear [35]. Although the system could be linearized, Pan and Do chose to apply a nonlinear control scheme using Lyapunov's direct method. In using nonlinear control they were able to maintain the model's accuracy. For their simulation, Pan and Do obtained system parameters from Korde [17] and the simulations show a load motion of less than 0.1 m deviation for a significant wave height of 4 m — or an approximately 97.5% motion decoupling.

A simple Proportional-PI controller is used by Gu *et al.* [36] for control of a hydraulic hoisting rig meant to lower heavy loads to the sea-floor. In their controller design shown in Figure 2.9, Gu *et al.* [36] use PI control as part of a closed-loop velocity control scheme for heave compensation, while proportional control is used in the outer control loop as position control to lower the load. In simulations, the controller was able to reduce a 1 m, 0.1 Hz sinusoidal heave motion input to approximately 1 cm — or a 99% decoupling. Although this simulation predicts excellent performance, it should be noted that a pure sinusoidal input is an idealized heave signal, and it would be preferential to provide the system response for a full spectrum of ocean waves. Additionally, when moving from simulation to a physical implementation, time-delay in system components may become a concern.

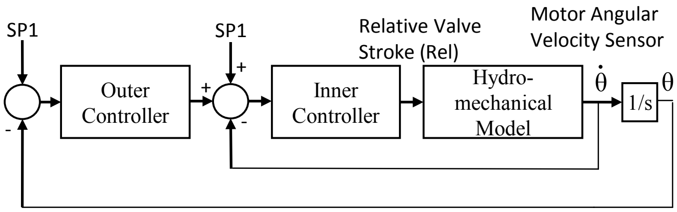


Figure 2.9: The control scheme used by Gu *et al.* [36] is shown here, with an inner velocity control loop, and an outer position control loop.

In the work by Hatleskog and Dunnigan from 2007 [34], it is briefly mentioned that a predictive controller may be helpful in creating an AHC system that approaches 100% effectiveness in heave motion decoupling. Reasoning is not given as to how prediction may improve performance; nevertheless, it is possible that a predictive controller could be useful in systems where a significant but consistent and known time-lag exists between heave measurement and actual motion. Prediction could also be used to partially correct for a large phase lag within the controller structure.

Hastlekog and Dunnigan go on to say that heave motion of a vessel is "...essentially unpredictable with a high probability of significant predictive error". Halliday *et al.* [37] in 2006 published work providing a method for using Fast Fourier Transforms (FFT) to accurately predict wave motion within 10% approximately 10 seconds (s) into the future and up to 50 meters (m) away from the point of measurement. Although Halliday *et al.* intended to use short-term wave prediction to increase efficiency of wave-energy collectors, their work is easily adaptable to predicting short-term ship motion using IMU data. Neupert *et al.* [38], at a conference in 2008, presented work to this effect.

Neupert *et al.* [38] present a system using heave motion prediction as part of the control methodology for an AHC crane. Figure 2.10 shows a simplified schematic of their heave prediction system. This heave prediction algorithm is the basis for the wave prediction algorithm used in conjunction with MPC in this thesis.

To predict ship motion, ship heave data from an IMU data ($w(t)$ in Figure 2.10) is collected for a set amount of time and an FFT is performed. Peak detection is performed on the FFT and the dominant peaks are determined, initializing an observer with the peak height A_{obs}, frequency f_{obs}, and phase ϕ_{obs}. A Kalman filter updates the value of A_{obs} in real-time while the other values are held constant until the next FFT is performed. Using a Kalman filter to update dominant peaks instead of performing an FFT every time step saves considerable computing power. When the FFT is performed again, some peaks may be removed or added to the observer depending on the data. The values for amplitude A_{obs}, frequency f_{obs}, and phase ϕ_{obs} are used by the prediction algorithm to predict future heave motion. The primary purpose of prediction in this controller is to help in dealing with known time delays between sensors and actuators which is important in systems with long delays as delay will introduce phase lag in a system, hindering a controller's ability to respond quickly.

Neupert *et al.* [38] use a linearized model of crane dynamics along with the pole-placement control method to set load position. The authors apply a simple observer using a mass-spring-damper model to calculate actual load position during operation. For a relatively stiff cable, this observer is likely unnecessary as the cable will not stretch appreciably and load motion will match actuator motion. Consider following equation:

$$\frac{X_L}{X_H} = \frac{cs + k}{m_L s^2 + cs + k} \tag{2.2}$$

where x_H is the ship heave, x_L is the load displacement, m_L is the load mass, k is the

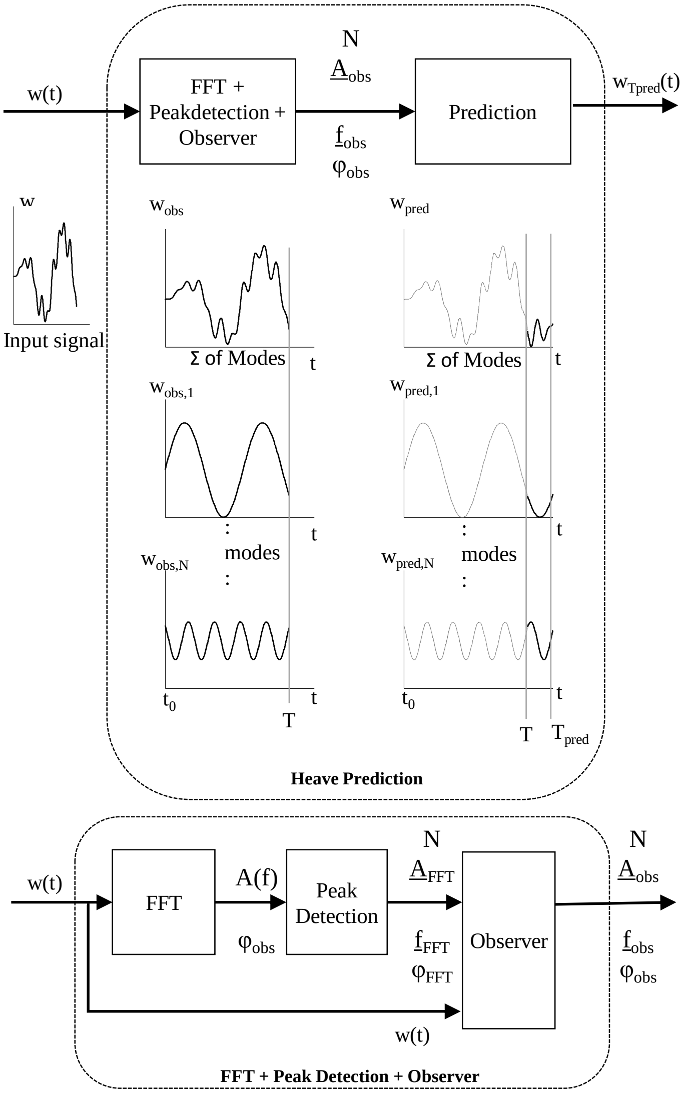


Figure 2.10: A simplified schematic of the Neupert *et al.* method for heave prediction is shown here. Figure taken from [38].

spring constant for the cable holding the load and c is the system damping. If k is dominant in the numerator and denominator suggesting a rigid cable, then Equation 2.2 can be simplified to $\frac{X_L}{X_H} = 1$ which means that load motion X_L matches actuator motion X_H. A dominant k would be representative of a load held at a shallow depth since cable mass, length and damping would be relatively small. For considerable depth an observer becomes useful as k is no longer dominant in the transfer function and the load motion will be out of phase with ship motion.

Neupert *et al.* [38] perform simulations showing that their state feedback controller can track a step-input to within ± 3 cm with a ship heave motion of approximately 0.5 m. In the follow-up work to Neupert *et al.* [38], Kuchler *et al.* [3] present the data seen in Figure 2.11 showing that, with a larger heave motion, a load motion of less than ± 3 cm is no longer attainable. In region A of Figure 2.11, from $t = 0$ to 250 s, the controller is inactive. Region B of Figure 2.11 shows the state-feedback control active, but heave prediction is unused. Region C of Figure 2.11 shows state-feedback and heave prediction being used together. Based on a performance factor that the authors introduced, namely $\int_{t_0}^{t_0+250} \Delta {z_p}^2 dt$, energy in the load is reduced by 83% for the nonpredictive controller and energy is reduced by 98.2% for the predictive controller — showing a clear improvement when using heave prediction. Similar results are shown for experimental results; however, the same performance factor cannot be used as values are not reported for the heave motion.

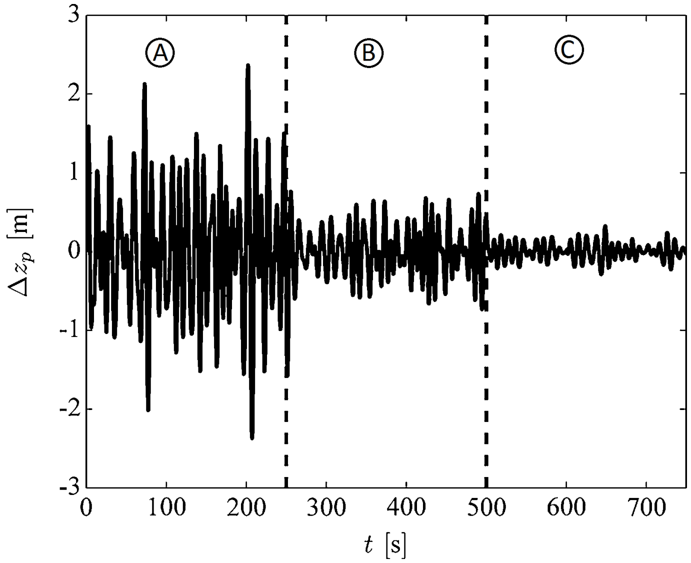


Figure 2.11: Simulation results showing load motion split into three sections: no controller, controller without prediction, and controller with prediction. Figure is taken from [3].

In their experimental system, Kuchler *et al.* [3] report a delay of approximately 0.7 s between sensor measurements and actuator response. It is possible that the inability of their controller to completely decouple load from heave motion is caused by the prediction algorithm error when trying to predict 0.7 s into the future. Reducing the system delay may further increase the ability of the system to reject heave from the load motion. Additionally, the use of state-feedback can be thought of as applying a filter to the system. When applying a filter, it is not always possible to completely decouple the output from the input. Feed-forward control is often applied to complement state-feedback controllers where it can lead to zero-error moving reference tracking in ideal circumstances [39].

2.4 Discussion and Conclusion

It is the present researchers opinion that the inability for many controllers to totally compensate for heave motion may not only be due to sensor lag, as mentioned by Kuchler *et al.* [3], but also inherent phase lag in a system. It is well known that simple PID and pole-placement-based controllers cannot perfectly track a sinusoidal moving reference because the controller's inherent phase lag ensures some delay in the system. This inability to perfectly track is especially true for reactive controllers tracking a moving reference. While the addition of a feed-forward component to both PID and pole-placement controllers can overcome delay due to system phase lag, allowing perfect tracking of a sinusoidal reference in ideal circumstances [39], coupling the inherent system phase lag with additional time delay can lead to significant system delays in the phase diagram, which cannot be easily compensated for. A possible option to correct for large phase lag is the use of a predictive controller, specifically model-predictive control (MPC). A model-predictive controller relies on a system model to determine optimal controller output by solving a quadratic optimization problem. Additionally, MPC is capable of utilizing knowledge of future set-point changes, a process called previewing, to react to set-point changes prior to the changes occurring. This previewing coupled with the wave prediction algorithm applied by Kuchler *et al.* [3] are an avenue of research which is yet to be pursued in literature. The potential to improve upon the current state of controllers in AHC research and design a novel controller not seen in the field are both sufficient reason to further pursue the design and implementation of an MPC system using wave prediction which, as mentioned in Chapter 1 was the second key goal of this thesis work. Recall that the first key goal was to determine the non-linear hydraulic valve properties and the third key goal was the creation of a simulator to test the AHC testbed under load.

Although electrically actuated AHC systems are becoming more prevalent, hydraulic AHC systems are still very commonly used due to their high power-to-weight ratio and easy integration into vessels at sea. It will be one focus of this thesis to examine fully the interaction between MPC and an un-loaded, full scale hydraulic testbed. What this means is that a hydraulic pump, valve, and motor which could be operate a 2000 lbf (8900 N) AHC system will be used in the work presented within Chapters 3 and 4. Although the motor itself will not be operated under load, the unloaded system should react very similarly to a loaded system due to the nature of the hydraulic valve and pump used. This will be further explored in the following chapter where the testbed is described in detail.

Chapter 3

Hydraulic Testbed Identification

This chapter provides an in-depth examination of each of the main hydraulic components used to complete this thesis work and describes how these components can be assembled to create an AHC system testbed. The non-linearities of the hydraulic valve component are examined and quantified to allow the implementation of a linear control scheme which will be the focus of Chapter 4. Furthermore, the identified properties of the system will be used in Chapter 5 for the creation of a MATLAB Simulink model of the AHC testbed. The hydraulic equipment used in the completion of this thesis work was provided by Rolls-Royce Canada Limited (RRC).

3.1 Hydraulic Circuit and Component Background

The hydraulic circuit used for the testbed in this thesis work is shown in Figure 3.1. RRC specified that this AHC would ideally be installed on a vessel with a pre-existing hydraulic pump and reservoir; so, for this reason an open-loop hydraulic system was chosen. As described in Chapter 2, an open-loop hydraulic system consists of a pump, an open-to-air oil tank, a valve, and an actuator — a hydraulic motor for the scope of this thesis work. For the AHC design in this thesis the motor is attached via direct-drive to a winch drum that is capable of hauling the heave compensated load in or out, as desired.

The schematic used in Figure 3.1 shows a load sensing pump which can provide variable flow to the 4-way, 3-position proportional valve. A load sensing pump operates by providing enough flow to ensure the pump outlet pressure — which we will call p_{pump} — is maintained at a fixed, adjustable amount above the load sense line pressure, which we will call p_{LS}. We can define a value, p_{drop} such that $p_{drop} = p_{pump} - p_{LS} > 0$. Figure 3.1 shows that the load sense line measures fluid pressure after the valve, while the pump creates a pressure before the valve; so, p_{drop} is actually the pressure change across the proportional valve, meaning this system maintains a constant pressure difference across the valve. To understand why maintaining a constant p_{drop} is important, we need to discuss flow through an opening.

In Figure 3.2 an opening of area A restricts fluid flow along a pipe. In this case

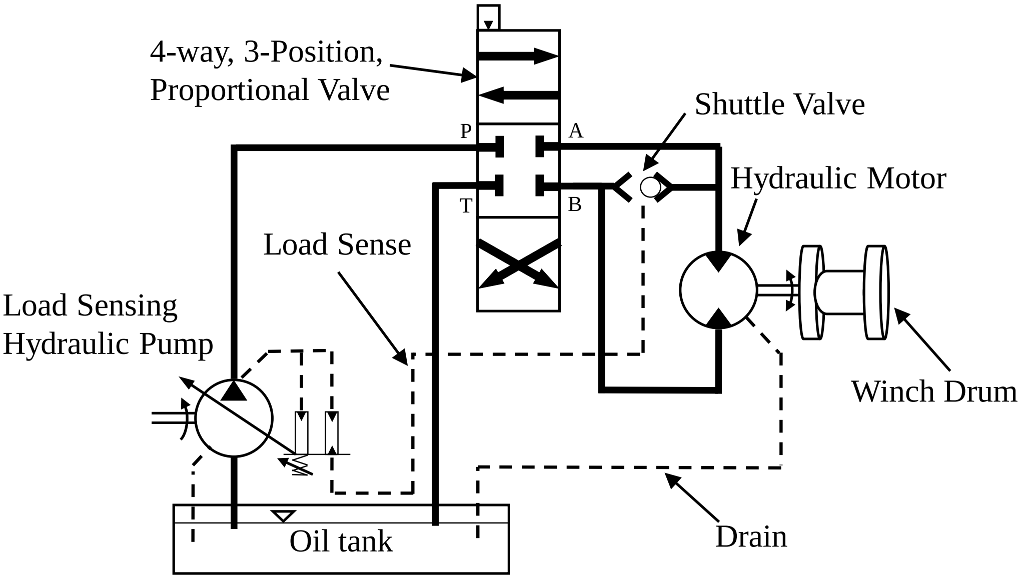


Figure 3.1: A simplified open-loop hydraulic circuit used for the AHC testbed is shown here. A pump supplies hydraulic fluid to a directional valve, which can both limit and change direction of the fluid leading up to the hydraulic motor, allowing the speed and direction of the motor to be controlled.

flow must be conserved, however, there is an associated energy loss with forcing the fluid through an opening. This energy loss manifests as the pressure drop $p_1 - p_2$. An equation known as the orifice equation can be used to describe the relationship between flow rate and pressure drop in Figure 3.2 as follows:

$$q = C_d a \sqrt{\frac{2(p_1 - p_2)}{\rho}} \tag{3.1}$$

where

q	=	Flow rate $[m^3/s]$
a	=	Opening area $[m^2]$
C_d	=	Discharge coefficient (related to opening geometry) [unitless]
p_1	=	Pressure before opening [Pa]
p_2	=	Pressure after opening [Pa]
ρ	=	Fluid density [kg/m^3]

What we see in this equation is that flow rate through an opening is proportional to the opening area and also proportional to the square root of the pressure drop across the opening. We assume the discharge coefficient to be a constant. So, if we can maintain a constant pressure drop across an opening then the flow rate will be

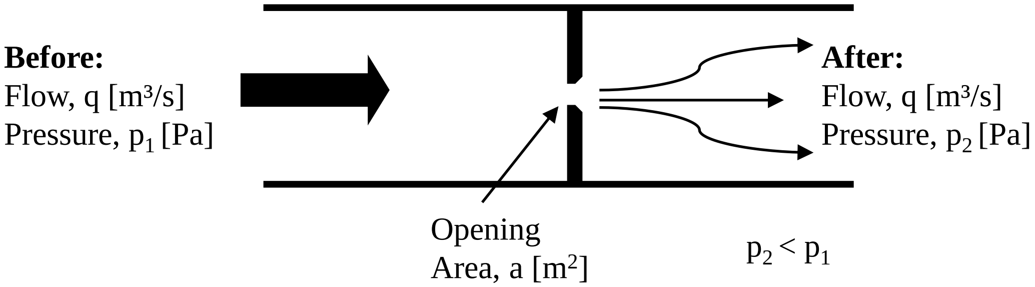


Figure 3.2: An opening of area a is restricting the flow q. Flow is maintained through the system, however, a pressure drop of $p_2 - p_1$ will exist when forcing fluid through the opening.

proportional to the area of the opening alone. Looking back at Figure 3.1, we have already established that the pressure drop across our proportional valve is being held constant by the load sensing pump, so we can now say that the flow rate across the proportional valve will be proportional to the valve opening only.

The 4-way, 3-position proportional valve in Figure 3.1 is drawn as being in the neutral position. For this valve, being in the neutral position means that all four ports are blocked and the motor is locked in position. With all ports blocked the pump remains idle, providing only enough flow to counter any leakage in the system and maintain pressure at port P. Figure 3.3 provides a section view of the inside of a standard 4-way, 3-position closed center proportional valve in neutral.

Inside of a proportional valve there is a cylindrical spool which directs flow between the ports by moving along a single axis. In Figure 3.3 the valve spool is positioned such that flow from port P is unable to flow to either port A or B. Actuation of the spool can be performed electronically, hydraulically, mechanically, or pneumatically depending on valve choice. For the testbed in this thesis the valve will be actuated electronically. To direct flow from port P to port A the spool must be moved to the right. Figure 3.4 shows that when the spool is actuated to the right pressurized fluid can move from port P to port A. Also note, when the valve moves to the right port B is now open to port T allowing fluid to return to the oil tank through port B. Similarly, if we want fluid to flow from port P to B we can shift the valve to the left which will open a connection between ports P and B and ports A and T. As the valve is shifted more in either direction the opening between ports increases in proportion to the spool position. From Equation 3.1 this increase in opening size means that the flow

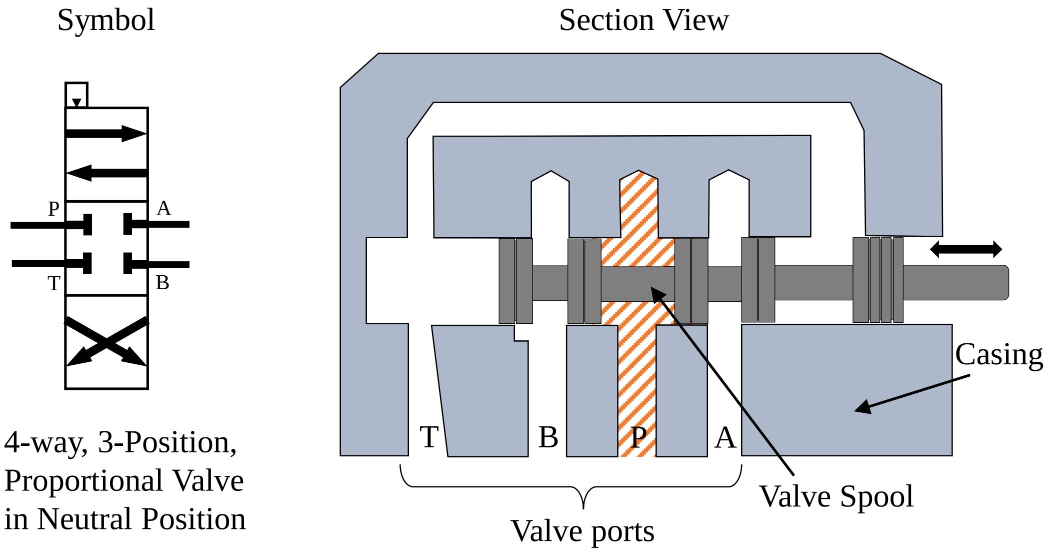


Figure 3.3: Here we see the symbol for a 4-way, 3 position closed center proportional valve on the left, with a section view of the valve's inside on the right. While in neutral, the valve blocks flow from leaving port P. The crossed lines indicate high pressure fluid in this region.

rate through the valve will also be proportional to spool position and as we control spool position electronically, flow rate will be proportional to an electronic signal we provide regardless of the motor load. For this system, shifting the valve spool either left or right changes the direction of fluid flow allowing control of the motor direction. In examining the valve used in our testbed it is necessary to determine the relationship between the electronic control signal and the valve opening, and thus also the flow rate through the valve. One other important property of the valve spool we are interested in is the deadband which is a property related to the construction of the spool itself.

In Figure 3.5 a detail view of the edge between port P and port A shows that the spool must travel a small distance before flow can develop between the ports. This small distance is the deadband and it corresponds to a spool position where changes in the electronic control signal to the valve do not produce a change in flow until the spool moves away from this zone. It is possible to have a valve made without a deadband, but due to the machining precision required there is an increase in both cost and lead time. Additionally, a deadband is sometimes desirable as it reduces leakage between the pump port P and the other valve ports while in the neutral position. Most inexpensive valves come with a deadband by default. To properly identify the testbed valve and ensure accurate control, the deadband will need to be

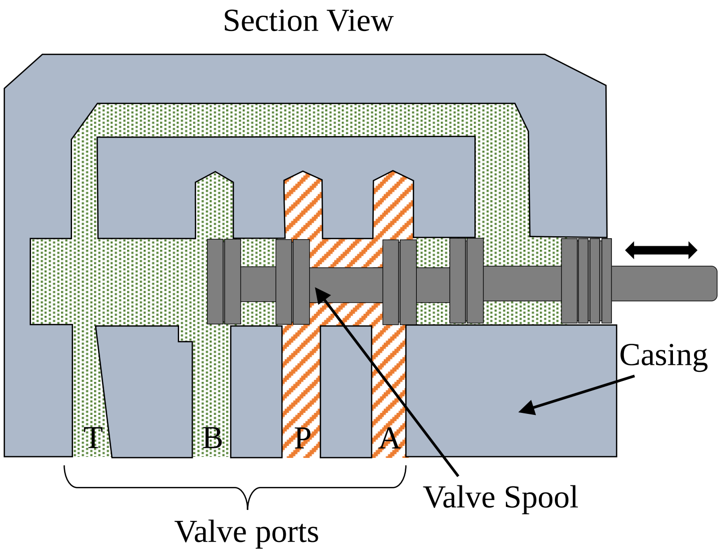


Figure 3.4: Here the valve spool has shifted, allowing fluid flow from port P to A, and from port B to T.

quantified.

The discussion so far has focused on identifying the properties related to fluid flow through our valve and ensuring that it is independent of the motor load. Understanding the flow rate through our valve is important because hydraulic motors are specified based on two properties: the relationship between fluid pressure and torque, and the relationship between flow rate and angular velocity. The relationship between flow rate and angular velocity for a hydraulic motor is called the motor displacement and is usually specified in ccm — cubic centimeters per revolution, or in in^3/rev. This flow rate to angular velocity relationship is linear, so if we can determine flow rate through the valve we can determine motor speed and vice versa. At the end of this chapter we will have determined the relationship between valve control signal and motor speed and identified a model to use for MPC control of the testbed.

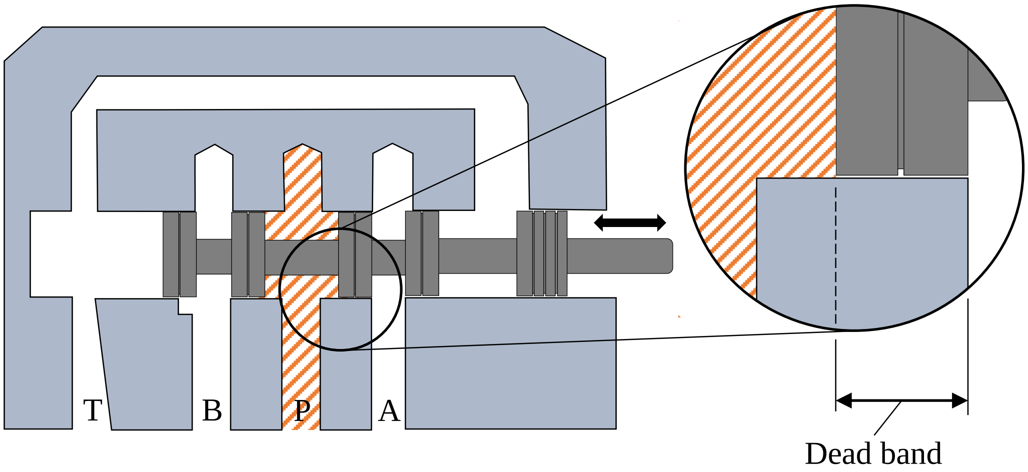


Figure 3.5: The detail view in this Figure shows a small distance which the spool must travel before fluid can flow from port P to port A. This area is known as the deadband.

3.2 AHC Requirements and Hydraulic Component Specifications

The hydraulic testbed was designed to suit an AHC system based on the general recommendations of RRC in Table 3.1.

Table 3.1: AHC Requirements for Hydraulic Test Bed

Requirement	**Rating**
Operating Load [lbf (N)]	2000 (8900)
Peak Ship Center of Mass (CoM) Heave Motion [m]	±2.5
Peak Ship CoM Heave Period [s]	7.5
Peak Ship CoM Heave Velocity [m/s]	2.1
System Peak Pressure [psi (bar)]	3000 (200)
Winch Drum Diameter [in (m)]	16 (0.4046)

The operating load was chosen to simulate cable tension as experienced by a winch dragging a submerged object. Neither the dimensions nor depth of the object were provided. The peak ship center of mass (CoM) heave motion, period, and velocity values were generated from the output of a ShipMo3D [40] simulation — software which simulates ship motion at sea, provided the ship dimensions and sea-state are given. In the above case, sea state 5 was chosen where sea state 5 contains ocean waves from 2.5 m to 4 m in height with a period in the range of 8 s. The vessel was 12 m long with a draft of 0.75 m. These values were chosen as the specific dimensions and conditions which an unmanned surface vehicle (USV) may experience while operating at sea. The cable drum attached to the winch motor was specified with a diameter of 16 in, or 40.64 cm providing enough information to calculate the required motor torque as well as the required motor displacement. Generally, in hydraulic design, one starts by specifying the actuator and then moving backward through the hydraulic system.

Given the drum radius of 0.2032 m and an operating load of 8900 N the operating torque of the motor was required to be at least 1808 Nm. The only option available which could provide 1808 Nm of torque with the 3000 psi peak system pressure was a Black Bruin Model BB4-800ccm hydraulic radial piston motor. The motor could also achieve 1808 Nm of torque on a reduced pressure of 1500 psi if connected to a 2:1 gear ratio transmission which broadens the potential application to systems with a lower peak operating pressure. Note, a 2:1 gear ratio would double the flow requirements.

The Black Bruin BB4-800 motor has a displacement of 800 ccm or 0.8 liters (L) per revolution (rev). With a 0.4046 m diameter drum and a maximum line speed of 2.1 m/s the drum is required to rotate at a maximum speed of 1.65 revolutions per second (rps) or 99 revolutions per minute (rpm). At 99 rpm, and 0.8 L/rev a peak flow rate of 79.2 L/min, or 20.9 USgal/min. At RRC a hydraulic power unit (HPU) capable of supplying 0 - 43 USgal/min at 0 - 3100 psi is used to emulate an HPU at sea. This HPU regulates flow from a load sensing pump using a Danfoss PVG-120 Proportional Valve — a valve which is relatively inexpensive, robust, and capable of handling the flow rates and pressures needed by the BB4-800 motor in the testbed.

The PVG-120 originally had only a mechanical lever to control flow, so an analog voltage input electronic control module was installed, allowing computer control of the valve. For a full parts list of the HPU and PVG120 valve see Appendix A. The testbed BB4-800 motor can be seen in Figure 3.6 attached to a rotary encoder via the highlighted 1:2 chain-sprocket reduction. In the side view to the right of the fixed mount is where hydraulic hoses (not visible in the image) connect from the HPU to the motor providing the flow necessary for the motor to rotate.

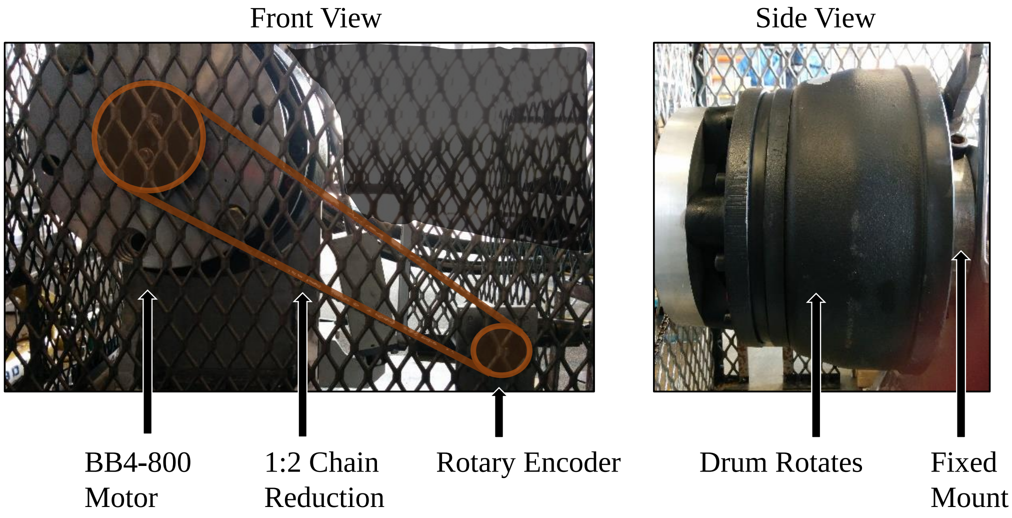


Figure 3.6: The Black Bruin Model BB4-800 motor is rigidly attached to a frame and connected to an encoder via a 1:2 chain-sprocket reduction as shown in the Front view. In the side view the mounting point on the right is visible. The motor is encased in a metal cage for safety reasons.

3.3 Instrumentation and Data Acquisition

To identify the testbed system a rotary encoder, two pressure sensors and a flow sensor were used. Table 3.2 provides all relevant technical specifications and model numbers for each device. Each device was connected to the testbed as labeled in Figure 3.7. The pressure transducers were used to determine the hydraulic motor load and to check that the pressure drop across the valve remained constant during operations. The Hedland flow meter was to be used to compare to the flow as calculated based on the hydraulic motor speed, however, the refresh rate of the flow meter was on the order of two seconds so it was not useful in real-time or transient applications. Flow rates are, instead, calculated based on the motor rotation speed. The pressure sensor data and flow rate data are used in Chapter 5 when creating a MATLAB Simulink model of the testbed.

The rotary encoder was configured for 7200 counts per encoder revolution, however, the encoder was read in quadrature meaning both rising and falling edges of the encoder signal were counted thus increasing the resolution by a factor of four to 28,800 counts per encoder revolution. The encoder was additionally linked to the hydraulic motor via a 2:1 reduction from encoder to motor, so the overall resolution

Table 3.2: Sensors in Testbed

Sensor	Model Number	Specifications
Pressure Sensor	Rexroth HM20-2X/250-H-K35	Pressure Range: 0 - 250 bar Output: 0.1 - 10 V
Flow Sensor	Hedland MR Flow Transmitter	Flow rate: 4 - 55 USgpm Pressure Range: 0 - 241 bar Output Range: 0 - 10 V
Encoder	BEI H250-SS	Resolution: 7200 counts/rev

of the system was 57,600 counts per revolution of the hydraulic motor. To read from the encoder and each transducer, as well as to control the PVG-120 valve, an NI myRIO model 1900 embedded system was used.

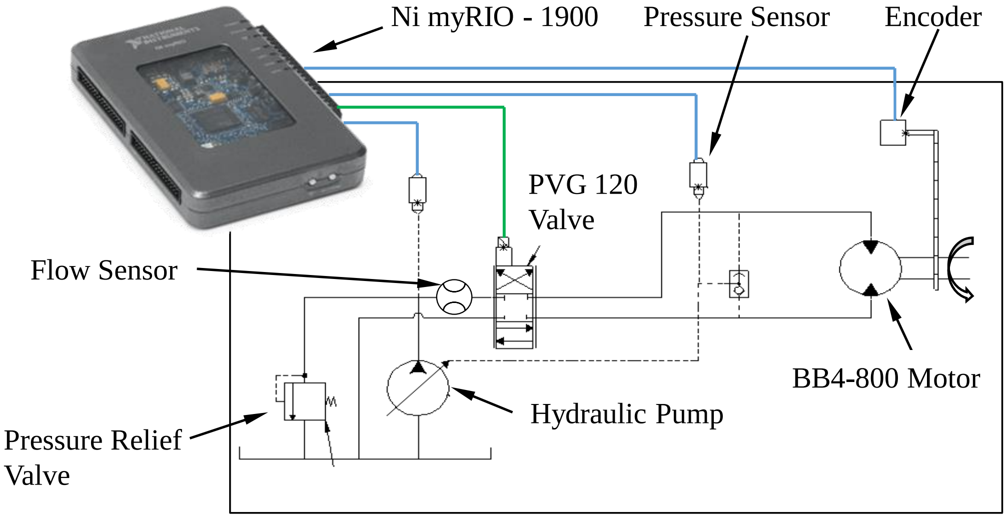


Figure 3.7: Each sensor used in the testbed is connected as shown. The NI myRIO reads each sensor and sends a control signal to the PVG-120.

The NI myRIO is an embedded hardware device with both analog and digital input/output channels. The myRIO allows NI LabVIEW programs to be run on-board in real-time. Having LabVIEW programs run in real-time means that when acquiring data or controlling a device the program will be consistent in timing of operations and loops, ensuring all data is collected within a single loop cycle and that the loop completes in exactly the time frame specified. This consistent timing is preferred when compared to a PC data acquisition loop which may delay loop operations due to the need to schedule other background processes on the CPU.

LabVIEW code was written to provide ramp, step, and sinusoidal control signals to the PVG-120 valve where the ramp input allowed identification of the valve deadband, the step inputs allowed identification of the valve transfer function model, and the sinusoidal inputs allowed the creation of a frequency response to compare to the transfer function model. The PVG-120 valve was powered with a 12 V supply (12.44 V actual) which, based on the specifications, means that the valve input signal range was 3 - 9 V, with neutral being at 6 V. An input signal below 6 V and outside of the deadband would rotate the motor in the negative direction (encoder counts decrease) and an input signal above 6 V and outside of the deadband would rotate the motor in the positive direction (encoder counts increase). First the ramp data will be examined and the deadband identified.

3.4 Deadband and Hysteresis

Figure 3.8 shows the raw encoder counts and voltage data for a linear voltage ramp from 3 V to 9 V and then back to 3 V with 60 seconds for each ramp direction. The count data is directly proportional to position, however, a relationship between control voltage and motor angular velocity is needed since, as discussed in Section 3.1, it is the motor angular velocity, not position, which is proportional to control voltage. Figure 3.9 plots angular velocity as a function of control voltage to the PVG valve. The angular velocity was calculated by differentiating the counts data in Figure 3.8. Scaling the differentiated counts data to convert from counts per second to motor revolutions per second leads to a plot of motor angular velocity versus control voltage as shown in Figure 3.9.

Examining the overall response in Figure 3.9 a deadband is clearly visible, however, also note that the blue and red curves, which occur during positive voltage ramps, are distinctly and repeatably different when compared to the negative voltage ramp shown by the green curve. This difference between curves, known as hysteresis, is common in proportional valves and can be caused by "...friction between a proportional valve spool and bore, flow forces acting against he spool, residual magnetism in the armature, gravity, as well as by inertia of the spool itself" [41]. Hysteresis introduces difficulties when trying to control an actuator reproducibly so some valve manufacturers apply dither — a high frequency (relative to the system response), low amplitude sinusoidal signal superimposed onto the valve control signal — to keep the

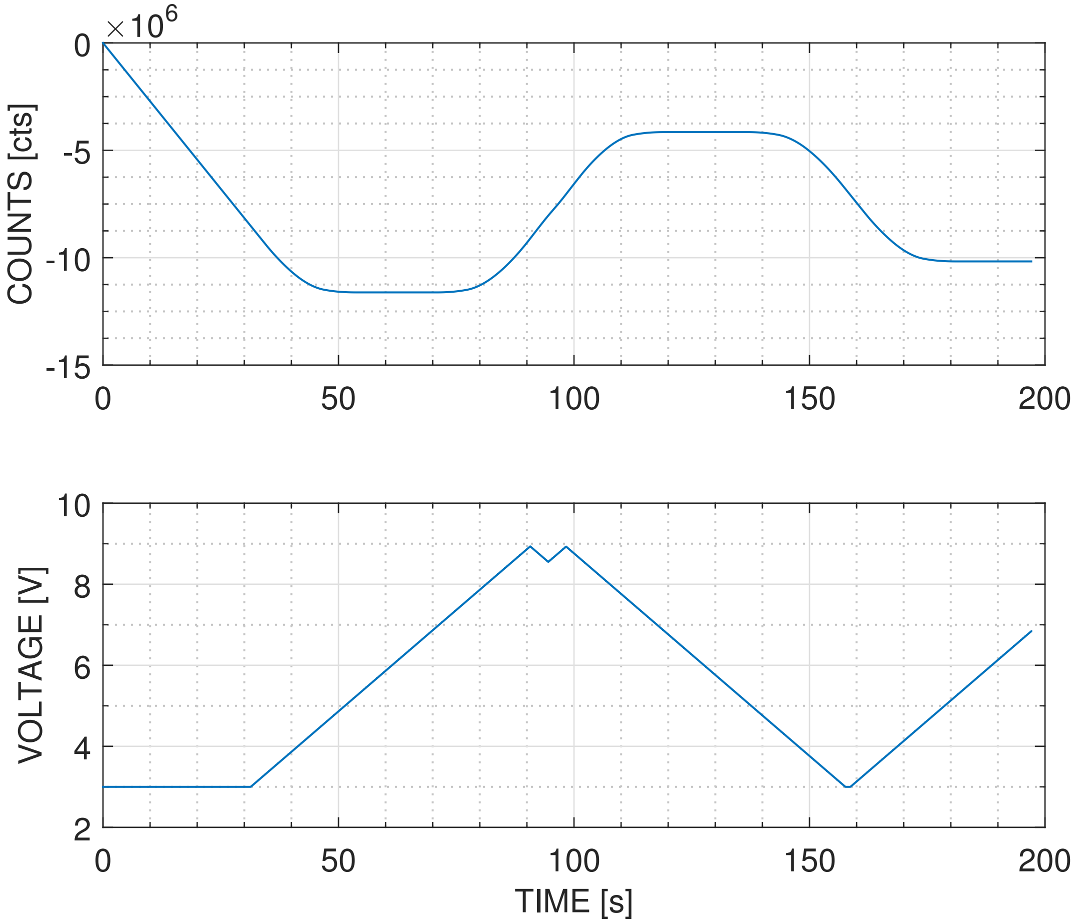


Figure 3.8: The upper plot shows raw encoder count versus time data during a 60 second control signal ramp from 3 V to 9 V. The lower plot shows the ramp voltage versus time.

spool moving a small amount to try to minimize static friction effects. The PVG-120 valve does not have dither added by the manufacturer so it was necessary to perform multiple ramp tests with a manually added dither signal to determine appropriate dither settings. The results are shown in Figure 3.10 where each of the six graphs plot angular velocity versus control voltage for the following dither cases: no dither, 20 Hz dither at 0.15 V peak signal, 50 Hz dither at 0.15 V peak signal, 15 Hz at 0.15 V peak signal, 50 Hz at 0.3 V peak signal, and 100 Hz at 0.3 V peak signal.

Notice that, in the cases seen in Figure 3.10, adding dither appears to extend the maximum angular velocity attainable by the motor by up to 0.55 revolutions per second (rps). This velocity increase is misleading as some ramp data without dither was also able to reach the same maximum angular velocity as ramp data with dither. The valve spool simply does not always reach the same end point without dither, and

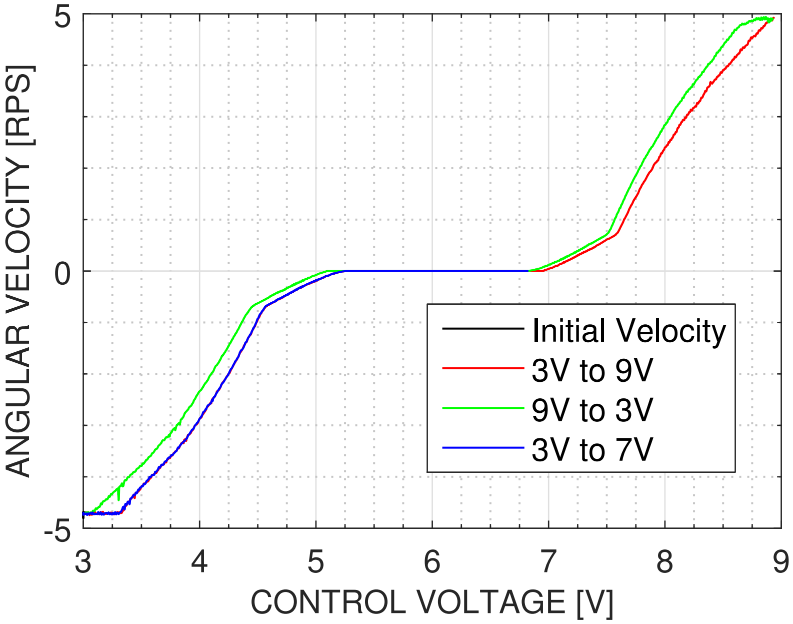


Figure 3.9: This plot shows the differentiated data from Figure 3.8 versus voltage. It has been color coded by ramp direction to show the valve response hysteresis.

regardless it is not of importance as the peak required speed of the motor is only 99 rpm (1.65 rps) based on the system requirements from Section 3.2. Examining the data collected, a dither signal of 50 Hz, 0.3 V_{pk} was chosen as it provided the largest reduction in hysteresis compared to the other dither signals. This 50 Hz, 0.3 V_{pk} dither was permanently added to all control signals passing to the hydraulic valve. With the hysteresis corrected the valve deadband could be accurately identified.

To determine the edges of the deadband the valve control voltage was linearly cycled up and down eight times from 6.9 to 7.3 V as well as from 5.5 to 5.1 V. Figure 3.11 plots the motor angular velocity response to these linearly cycled voltage signals where the green dots occur during ramp up, and the red x's during ramp down. The edge of the deadband is defined as the control voltage at which the motor velocity begins to change. In each data set there is a spread of voltages where the motor velocity reaches zero which could be chosen as the edge of the deadband. Due to the number of possible choices the midpoint of each spread was chosen to define when the deadband ended. The upper edge of the deadband was determined to be 7.125 V while the lower edge of the deadband was found to be 5.37 V. With the deadband identified, it will be compensated for in Chapter 4 through application of a control algorithm, allowing a more linear control of the AHC system. As an aside, note that the data collected during a positive ramp is slightly below the negative ramp data

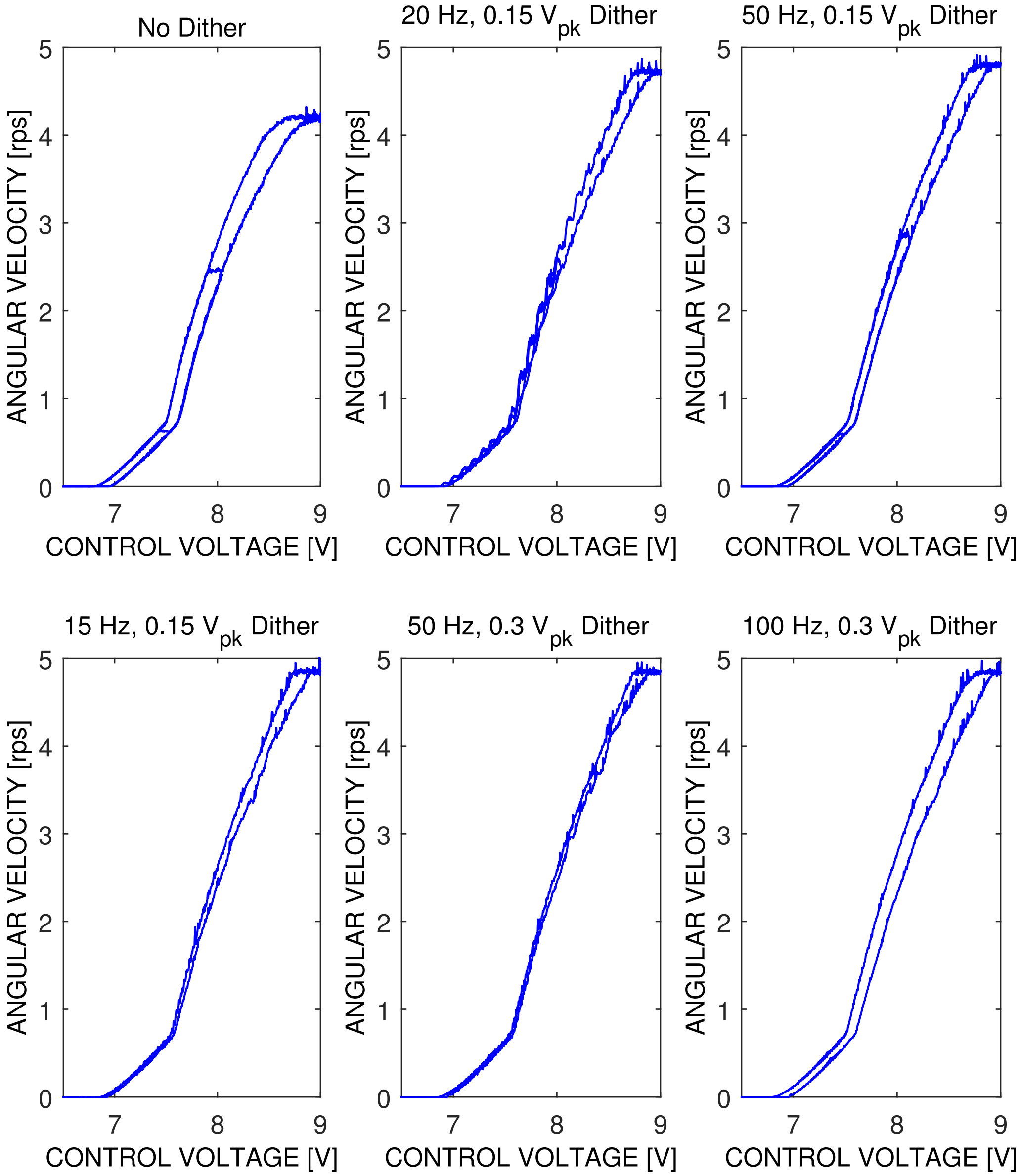


Figure 3.10: Six plots show the effects of adding different frequencies and amplitudes of a dither signal during voltage ramp. The option which reduces hysteresis effects most significantly is a 50 Hz, 0.3V sinusoidal signal added to the control voltage.

suggesting the valve hysteresis was not completely removed by adding dither.

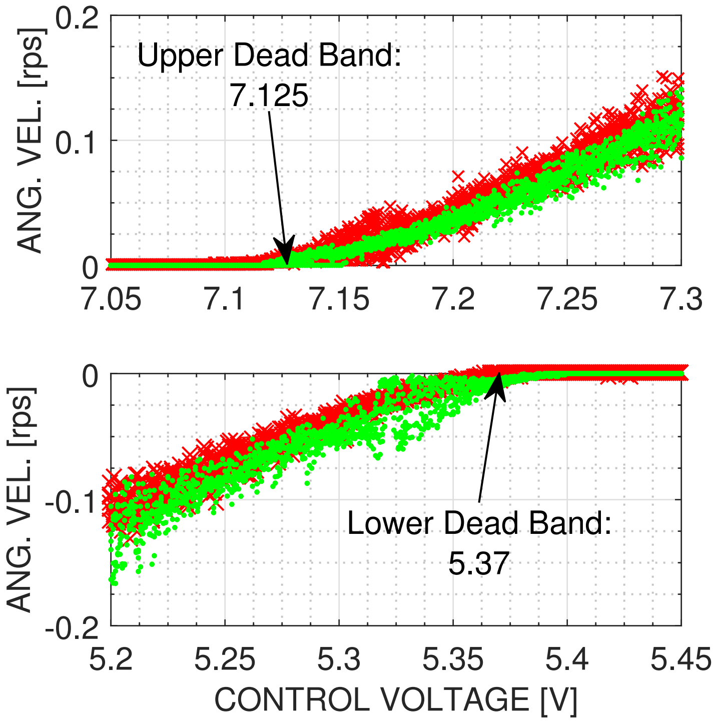


Figure 3.11: Here the upper and lower PVG-120 deadband limits are identified based on when the velocity goes to zero.

3.5 System Model and Response Non-Linearity

A two pole, one zero transfer function (TF) model of the AHC testbed was determined based on step response data using the MATLAB System Identification toolbox. Figure 3.12 shows the testbed system step response with an identified transfer function model overlay. The identified transfer function was,

$$\frac{\omega[rps]}{V} = \frac{1597s + 26.93}{60(s^2 + 23.8s + 0.3919)} \tag{3.2}$$

where ω is angular velocity in rps and V is voltage. Separating the denominator into its roots the transfer function becomes,

$$\frac{\omega[rps]}{V} = \frac{1597(s + 0.01686)}{60(s + 23.78)(s + 0.01648)} = \frac{26.6167}{s + 23.78} \tag{3.3}$$

where the zero and pole of $(s + 0.01686)$ and $(s + 0.01648)$ cancel each other in the final step as they are close to identical, leading to a simpler first order system.

In Figure 3.13 a comparison between the steady state angular velocity response of Equation 3.3 and the testbed steady state response versus control voltage is given. In

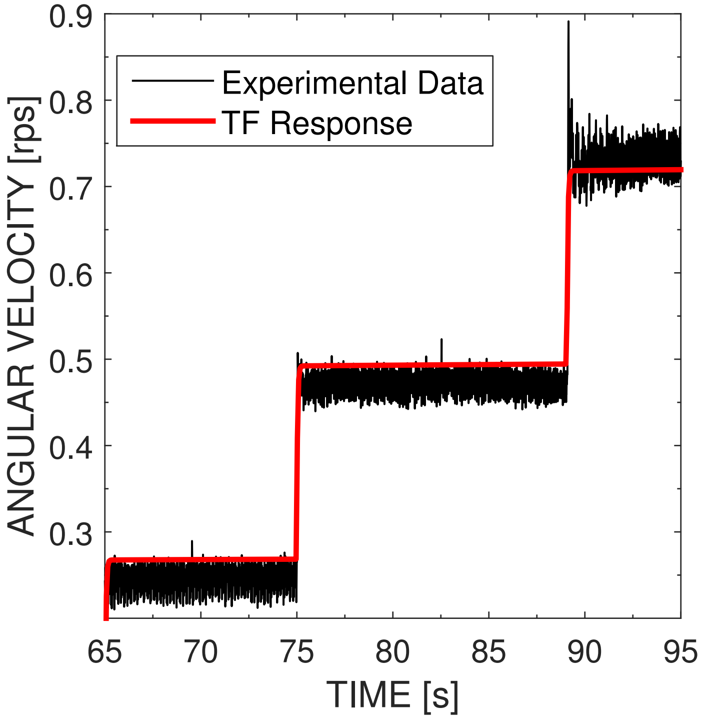


Figure 3.12: A comparison between the experimental data and the TF response data shows errors in gains which are explained by deviations in linearity of the valve at low voltages.

this plot, the system response shows that two distinct gains regions exist for the AHC testbed: one inside of the control voltage range $[-0.7, 0.7]$ which the identified TF model reasonably matches, and one outside of the control voltage range $[-0.7, 0.7]$ where the linear TF response deviates from the actual system response. This divergence cannot be corrected for within the TF model itself as the LabVIEW MPC implementation used in this thesis work requires a linear model. Instead, a solution allowing the use of the linear TF model for the full range of motor speed will be presented in Chapter 4.

To further validate the linear TF model, data to construct a frequency response for the PVG-120 and BB4-800 system was collected after all experiments were performed. A 0.275 V_{pk} sine wave was input for frequencies from 0.1 Hz to 10 Hz and the output angular velocity was measured. The voltage was kept within the region where the TF model and the system model steady-state responses match. In Figure 3.14 the collected frequency response and the TF response are plotted. At 0.62 rad/s the identified TF gain is $10^{0.9747/20} = 1.11$ and the frequency response gain is $10^{0.6335/20} = 1.08$ which corresponds to a 4% disagreement. The corner frequency error at -3 dB for the TF response is 29.47 rad/s, while the experimental data shows a corner frequency of 50.27 rad/s — a disagreement of 70%. Later, in Chapter 4, it is shown that when using the identified TF model for MPC control the controller is robust against errors of ±60% on the model corner frequency identification. This robustness against corner frequency errors suggests that the discrepancy in corner frequencies seen in Figure 3.14 may be acceptable.

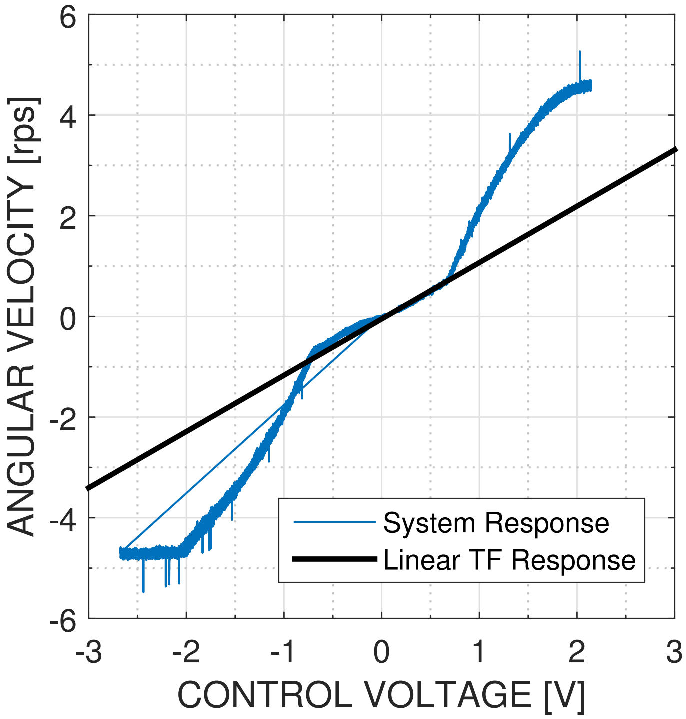


Figure 3.13: A comparison between the actual testbed velocity versus control voltage and the TF velocity versus control voltage data shows the two responses diverge near a control signal of 1 V.

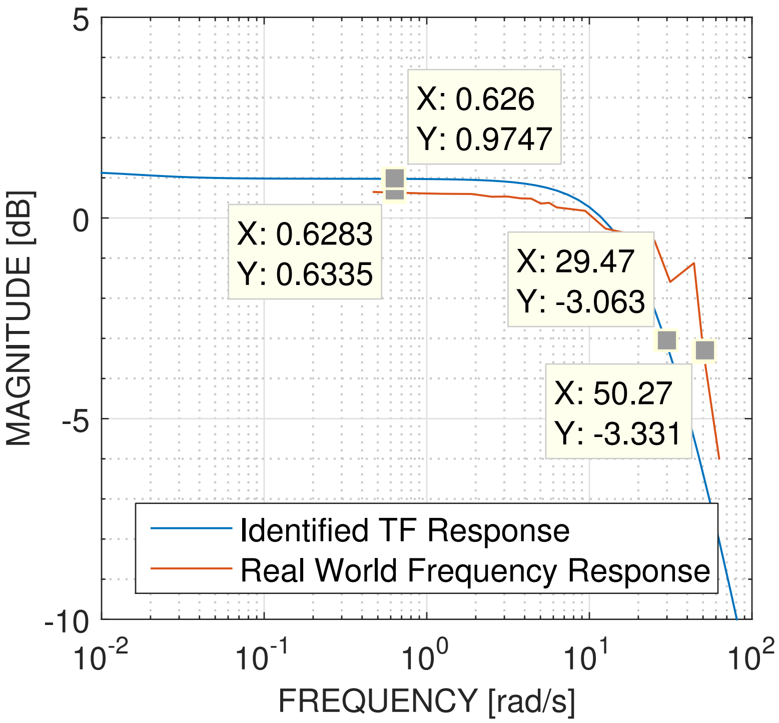


Figure 3.14: A comparison between the experimental data and the TF response data shows errors in gains which are explained by deviations in linearity of the valve at low voltages. Conservatively, the -3 dB point occurs at 29.47 rad/s (4.69 Hz) where control signals above this frequency will not track well.

3.6 Conclusion

In this chapter the AHC testbed was introduced and characterized by identifying the valve deadband, as well as a transfer function model for the full AHC system at low velocities. The full system steady-state response was also measured, however, Figure 3.13 demonstrated that above one revolution per second the TF model and the system response deviate from each other. This deviation occurs due to the valve opening not increasing linearly with spool position. As mentioned previously, a solution allowing the use of the linear TF model for the full range of motor speeds will be presented in Chapter 4, allowing a linear MPC controller to operate the AHC testbed.

Chapter 4

Model Predictive Control with Wave Prediction and Actuation of the AHC Testbed

This chapter presents the results of implementing model-predictive control (MPC) in conjunction with a heave prediction algorithm to operate an un-loaded, full-scale AHC testbed. Section 4.1 of this chapter is an overview of MPC and shows a brief example detailing how two of the tuning parameters affect the aggressiveness of the controller. Section 4.2 of this chapter details the theory behind the heave prediction algorithm used to allow control based on future heave motions. Section 4.3 explains the methods used to correct for valve non-linearities identified in Chapter 3 and examines data obtained from actuating the AHC testbed to determine appropriate tunings for both MPC and PID controllers. Section 4.4 of this chapter compares the tuned MPC controller to the tuned PID controller under four distinct test cases. Section 4.5 of this chapter examines the robustness of the MPC controller to model errors. Section 4.6 concludes with a discussion of the results presented in this chapter.

4.1 Theoretical Background and Behavior of a Model-Predictive Controller

Model-predictive control is a discrete algorithm in control theory which determines the optimal controller output necessary to reach a set-point through the minimization of a quadratic cost function. In contrast to proportional-integral-derivative (PID) control, MPC requires that a system model be identified for the controller to function. Given an identified system model, MPC minimizes the cost function J where,

$$\begin{gathered} J = \sum_{i=0}^{N_p} x_i^T Q x_i + \sum_{i=0}^{N_c} u_i^T P u_i + \sum_{i=0}^{N_c} \Delta u_i^T R \Delta u_i \\ x_{min} \leq x_i \leq x_{max} \\ u_{min} \leq u_i \leq u_{max} \end{gathered} \tag{4.1}$$

Here, Q, P, and R are weighting parameters for the model state error x, the controller output u, and the rate of change for the controller output Δu, respectively. N_p is the prediction horizon over which the controller allows the model to evolve and N_c is the control horizon, or how many time-steps forward the control action is calculated. It is required that $N_p \geq N_c$, and for $i \geq N_c$, u_i and Δu_i are held constant. Choice of N_c and N_p will depend on the system sampling time and the time scale of the system dynamics.

To minimize the cost function a type of mathematical optimization problem called Quadratic programming (QP) is solved. The simple explanation is that the QP algorithm considers the set of all possible values of u over the control horizon and determines which combination of values of u such that the vector $[u_0, u_1, ..., u_{N_c}]$ of future control actions subject to the constraints on u and x will minimize the cost function over the prediction horizon.

The function of a model predictive controller can be best explained by examining Figure 4.1 where there are three regions of time to consider: the past, where the previous system output and the past controller action are found, the current time k, and the future, where the predicted system output is calculated based on the future optimal control action.

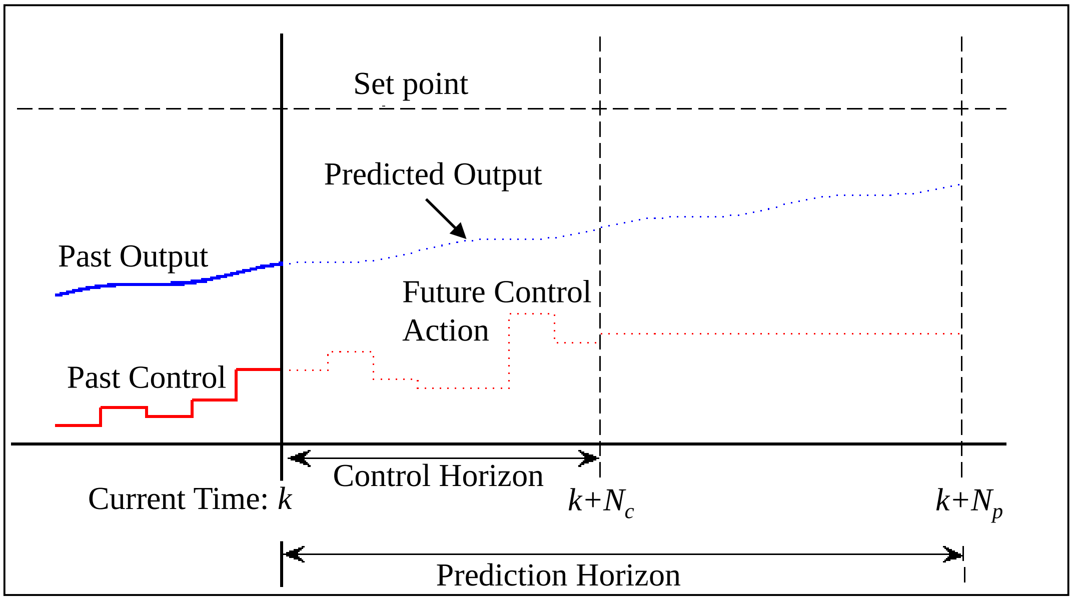


Figure 4.1: The behavior of a MPC system is seen with the control and prediction horizons shown. The future action is calculated based on the current state at time k.

Before time k we have the previous system response and controller action. At time k the controller solves the QP problem of minimizing J and determines the optimal values of u from u_k to u_{k+N_c} needed to reach the set-point. These control actions are shown between time k and $k + N_c$. The systems estimated reaction to these control actions is also seen in this time range. After $k + N_c$ the controller action is held constant at u_{k+N_c} and the system model is allowed to evolve until $k + N_p$.

The calculation of u is computationally intensive, and a larger value of N_c means a longer calculation. Keeping N_c low can allow the system to run in real-time which is important for many applications.

If there is significant time-lag in the system, then the system will not react to control actions immediately and it may not be necessary to calculate many values of u. For example if $N_c = 10$, $N_p = 10$, and there is a 5 time-step delay in the system, then any applied controller action will not be seen in the system output until the fifth time-step. After this point any control action applied will not appear in the system output until after the prediction horizon, so it is of no advantage to calculate control values past $N_c = 5$ as they will not be seen.

Smaller values of N_c result in a less aggressive (slower) control action. Figure 4.2 shows this less aggressive control action in detail where a simple MPC controller is tracking a step input for a simple inertia-damper system described by the transfer function

$$G(s) = \frac{1}{Js + B}$$

The larger plot in Figure 4.2 is a representative example for each of the smaller plots with the same axis scale set-point and system labels. Each smaller plot corresponds to a value of the prediction horizon N_p and the control horizon N_c. Notice that for a given value of N_p, smaller control horizons results in less aggressive control action. Or conversely, for a given control horizon, larger values of the prediction horizon give less aggressive action.

Consider the plot where $N_p = 1$ and $N_c = 1$. Since the controller is only looking one step ahead, the optimal control action in this case is going to be the one which reaches the set-point at the next time-step with no consideration for the time-step following that. Now, if we applied the previously determined control action we might realize it actually causes a very large overshoot at two time-steps ahead. If the controller had been looking two steps ahead ($N_p = 2$) it would have accounted for upcoming overshoot and applied a less aggressive control.

Often control algorithms do not ‘know’ when a set-point change is going to occur

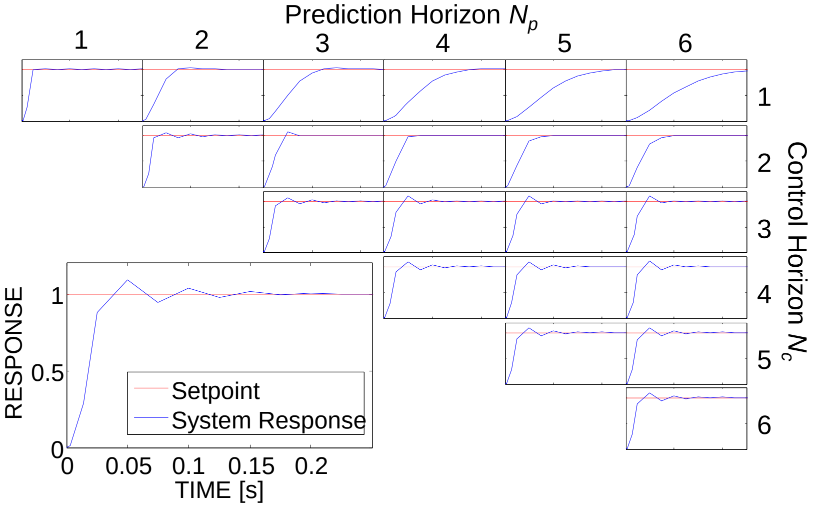


Figure 4.2: The behavior of an MPC controller is shown here for various values of N_p and N_c. Overall, longer prediction horizons and shorter control horizons result in less aggressive control action.

or, for a moving reference, how the set-point is going to evolve. In Figure 4.1 the set-point was constant; however, if the set-point was varying within the control horizon, the changing set-point can be considered during the QP minimization of J and preemptive control action can be applied to follow the changing set-point. Providing future information of a moving reference is called set-point previewing. An MPC controller with previewing is demonstrated in the lowest plot of Figure 4.3 where the responses of three controllers to a step input are given for a simple first-order system.

The upper two plots of Figure 4.3 show the step response of a PID and an MPC controlled system without previewing. Notice the upper two responses begin to change after the step input has occurred. In the lower plot, the system response begins to change prior to the step input demonstrating that the MPC controller uses knowledge of the upcoming step input attempting to reach the desired set-point at the exact time the step occurs instead of after. The ability of an controller to react to upcoming set-point changes can be especially useful in situations where a constantly moving reference must be tracked. In the following section a heave prediction algorithm will be presented which, when combined with MPC allows set-point previewing of ship motion at sea.

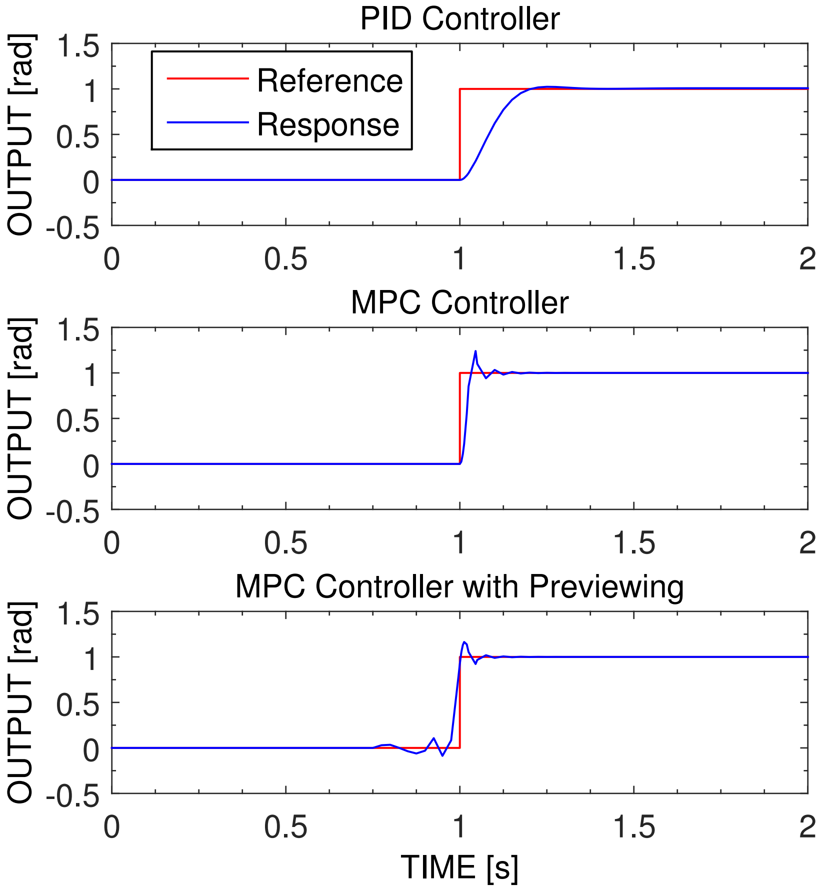


Figure 4.3: The behavior of three controllers tracking a step-input are shown here with the lower controller using set-point previewing while the middle and upper controllers do not use previewing. The use of previewing can be seen in the lower plot with the controller reacting prior to the step-input occurring.

4.2 Heave Prediction Model

Within this section, a heave prediction algorithm will be presented. It is of note, however, that although the algorithm presented is used to predict only heave motion, the algorithm can estimate any periodic, sinusoidal input signal, not just heave; thus it could be used as a generic set-point prediction algorithm.

As mentioned, the key to heave prediction is the assumption that ship heave is periodic and, therefore, can be decomposed into a set of N sine waves with an additional offset term to account for any deviation from zero mean. Following the work of Kuchler *et al.* [3] heave motion can then be written as,

$$h(t) = \sum_{i=1}^{N} A_i \sin(2\pi f_i t + \phi_i) + v(t) \tag{4.2}$$

with A_i, f_i and ϕ_i being the amplitude, frequency, and phase for each mode of the ship heave and $v(t)$ being the offset term. The frequency values f_i are determined based on a fast-fourier transform (FFT) of the previous 120 seconds of ship heave data with f_i values chosen corresponding to the largest peak values of the FFT spectrum. The amplitude and phase values at these peak frequencies are used to initialize a discrete observer model

$$\begin{aligned} \hat{\mathbf{x}}_{k+1} &= \Phi\hat{\mathbf{x}}_k + \mathbf{L}(h_k - \hat{h}_k), \qquad \hat{\mathbf{x}}_0 = \mathbf{x}_0 \\ \hat{h}_k &= \mathbf{C}\hat{\mathbf{x}}_k \end{aligned} \tag{4.3}$$

where k indicates the current time-step, $\hat{\mathbf{x}}$ is the vector of observed states, $\mathbf{x}_0$ is the vector of initial states, Φ is the discrete system matrix, $\mathbf{L}$ is the vector of observer gains, $\hat{h}$ is the estimated ship heave motion, and $\mathbf{C}$ is the system output matrix. The variables in Equation (4.3) are derived as follows:

$$\dot{x}_i = \underbrace{\begin{bmatrix} 0 & 1 \\ -(2\pi f_i(t_0))^2 & 0 \end{bmatrix}}_{\mathrm{B}_i(t_0)} x_i$$

$$h_i(t) = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{C_i(t_0)} x_i \quad i = 1, ..., N \tag{4.4}$$

are the continuous equations describing a single ship heave mode with states

$$x_i(t) = \begin{bmatrix} A_i(t)\sin(2\pi f_i t + \phi_i(t)) \\ 2\pi A_i(t) f_i \cos(2\pi f_i t + \phi_i(t)) \end{bmatrix} \tag{4.5}$$

and initial conditions

$$x_i(t_0) = \underbrace{\begin{bmatrix} A_i(t_0)\sin(\phi_i(t_0)) \\ 2\pi A_i(t_0) f_i \cos(\phi_i(t_0)) \end{bmatrix}}_{x_{i,0}} \tag{4.6}$$

Equations (4.4) and (4.5) are combined for each of the N modes to create

$$\dot{x} = \underbrace{\begin{bmatrix} B_1 & 0 & \dots & \dots & 0 \\ 0 & B_2 & \ddots & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & B_N & 0 \\ 0 & \vdots & \vdots & 0 & 0 \end{bmatrix}}_{\mathbf{B}} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \\ v \end{bmatrix}}_{\mathbf{x}}$$

$$h(t) = \underbrace{\begin{bmatrix} C_1 & C_2 & \dots & C_N & 1 \end{bmatrix}}_{\mathbf{C}} x \tag{4.7}$$

which is discretized to

$$\mathbf{x}_{k+1} = \underbrace{\exp(\mathbf{B}\Delta T)}_{\Phi} \mathbf{x}_k$$

$$h_k = \mathbf{C}\mathbf{x}_k \tag{4.8}$$

where ΔT is the sampling time of the system. As already shown in Equation (4.3) the observer is written as,

$$\begin{aligned} \hat{\mathbf{x}}_{k+1} &= \Phi\hat{\mathbf{x}}_k + \mathbf{L}(h_k - \hat{h}_k), \qquad \hat{\mathbf{x}}_0 = \mathbf{x}_0 \\ \hat{h}_k &= \mathbf{C}\hat{\mathbf{x}}_k \end{aligned}$$

with the observer gain

$$\mathbf{L} = \begin{bmatrix} 0.2 & 3 & 0.2 & 3 & \cdots & 0.2 & 3 & 0.1 \end{bmatrix} \tag{4.9}$$

being chosen through trial-and-error within the scope of this work. For any single mode the two observer state estimates can be written as,

$$\begin{aligned} \hat{x}_{i,1} &= A_{\text{Obs},i} \sin\left(2\pi f_i t + \phi_{\text{Obs},i}\right) \\ \hat{x}_{i,2} &= 2\pi A_{\text{Obs},i} f_i \cos(2\pi f_i t + \ \phi_{\text{Obs},i}) \end{aligned} \tag{4.10}$$

which can be rearranged to solve for $\phi_{\text{Obs},i}$ and $A_{\text{Obs},i}$ as follows:

$$\phi_{\text{Obs},i} = \arctan\left(\frac{2\pi f_i \hat{x}_{i,1}}{\hat{x}_{i,2}}\right) - 2\pi f_i t \tag{4.11}$$

$$A_{\text{Obs},i} = \frac{\hat{x}_{i,1}}{\sin(2\pi f_i t + \phi_{\text{Obs},i})} \tag{4.12}$$

where $i = 1, ..., \text{N}$ for N frequency modes identified. The number of modes and their frequencies can change over time so when a new mode is introduced or an old mode removed the heave model should be reinitialized with the new modes. Within the scope of this thesis it is assumed the modes do not change in a single run, reducing the complexity of implementation for the heave prediction algorithm.

To predict future heave motion Equations (4.11) and (4.12) are used to re-create Equation (4.2), resulting in

$$h_{\text{pred}}(t + T_{\text{pred}}) = \sum_{i=1}^{N} A_{\text{Obs},i} \sin(2\pi f_i (t + T_{\text{Pred}}) + \phi_{\text{Obs},i}) + v \tag{4.13}$$

where h_{pred} is the heave prediction T_{pred} seconds into the future from current time t.

4.3 MPC and PID Controller Implementation and Tuning

When performing AHC tests fixed values for the MPC parameters N_p, N_c, Q, P and R were used to ensure comparable results across all tests. All testing was performed with 50 msec time steps (20 Hz sampling rate). To determine optimal MPC parameters, the standard deviation of the error between actual motor position and motor position set-point was used as the metric for determining how well a controller performed. For PID tuning, the controllers ability to track a step input was used. Before tuning control parameters however, corrections for the non-linear valve response were made.

4.3.1 Adjusting the Control Signal for Non-linear Valve Response

Figure 4.4 is a flowchart showing how the MPC and PID controllers operate the PVG-120 proportional valve. First, a control voltage is generated at each controller time-step and delivered to the non-linear gain correction function. The non-linear gain correction function is explained by examining Figure 4.5.

If the controller output is 2 V (point A in Figure 4.5) then following the arrow from point A to point B shows that the controller is demanding a velocity of 2 rps based on the linear model which the MPC is using. Providing a control voltage of

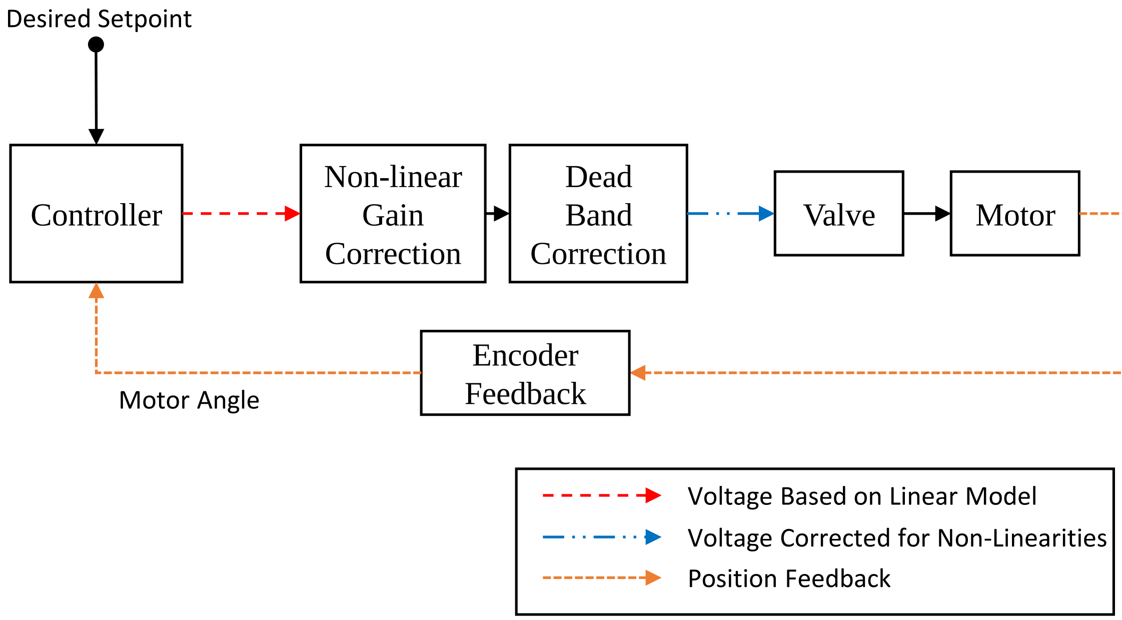


Figure 4.4: This flow diagram provides the logic followed in converting the controller signal to a signal to actuate the control valve. The conversions shown here are used for both the MPC and PID controllers.

2 V directly to the control valve, after deadband correction, would result in a motor speed of 4.5 rps seen at point B′, not 2 rps as desired at point B. A new voltage is defined by following the path from points B to C, and down to D finding that a voltage of 1 V is required to achieve the desired velocity. Within the non-linear gain correction block in Figure 4.4 this gain correction is performed by a 1D lookup table and the adjusted voltage is used by the deadband correction block which relies on the deadband identification performed in Chapter 3.

In Chapter 3 the valve voltage deadband was identified to be from 5.37 V to 7.125 V. In practice working with the AHC testbed it was determined that values of 5.34 V to 7.14 V achieved better results — an improvement attributed to ensuring that a non-zero control signal will always result in motor motion. The deadband correction block in Figure 4.4 follows the simple logic that, for the voltage into the block v_{in}

$$\begin{aligned} \text{if } |v_{in}| < 0.005, \text{then} \quad & v_{\text{out}} = 6 \\ \text{else if } v_{in} > 0, \text{then} \quad & v_{\text{out}} = v_{\text{in}} + 7.14 \\ \text{else if } v_{in} < 0, \text{then} \quad & v_{\text{out}} = v_{\text{in}} + 5.34 \end{aligned}$$

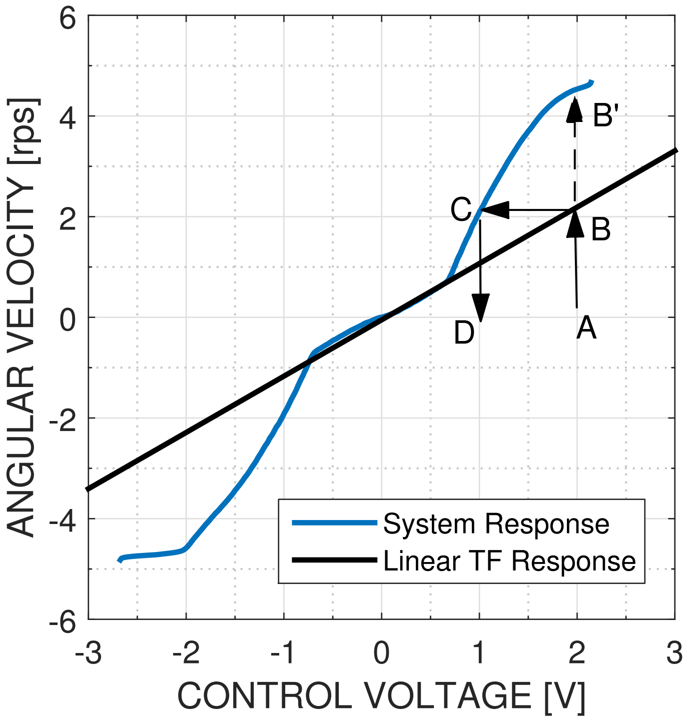


Figure 4.5: A comparison of the linear model used by MPC and the actual system response is shown. If the controller provides 2 V (point A), then the controller is commanding a velocity of approximately 2 rps (point B) based on the TF, but a velocity of 4 rps based on the system response (point B). For the system to rotate at a velocity of 2 rps, following from point C to point D we see a control voltage of 1 V would be appropriate.

where the first line was necessary to reduce valve instability caused by small oscillations around 0 V. Once the non-linear corrections are made the corrected control signal can be used by the valve to actuate the motor, rotating the motor and changing the encoder measurement which is then provided as feedback to the controller.

4.3.2 Test Case Data for AHC Testbed

There are four data sets seen in Figure 4.6 used as a moving reference to test the tracking ability of the MPC and PID controllers on the AHC testbed. The Benchmark case data was taken from Kuchler *et al.* [3] to compare the MPC controller with wave prediction from this thesis to the non-linear controller with wave prediction used by Kuchler *et al.*. Test Case A was provided by RRC to represent simple heave motion of a vessel at sea. In Test Case B a high frequency sine wave was superimposed onto the Test Case A data to represent a smaller vessel moving at sea or a vessel experiencing more erratic wave motions. The fourth data set, Test Case C, contains a signal of slowly modulating frequency to represent different frequency modes being introduced into the heave motion over time.

The data sets provided by RRC contain 450 seconds of heave data, where the first 120 seconds were used to initialize the heave prediction FFT while the next 120 seconds of data were used to compare the PID and MPC controllers. Additionally, two more data sets were provided, derived from the Test Case A and B data, where high frequency 20 Hz, 0.05 peak amplitude noise was superimposed to represent noise in the heave motion sensors. For the Benchmark Case there was 215 seconds of total data, allowing only 95 seconds of data to compare the PID and MPC controllers. To compare the controllers, values for the controller tune-able parameters must be chosen.

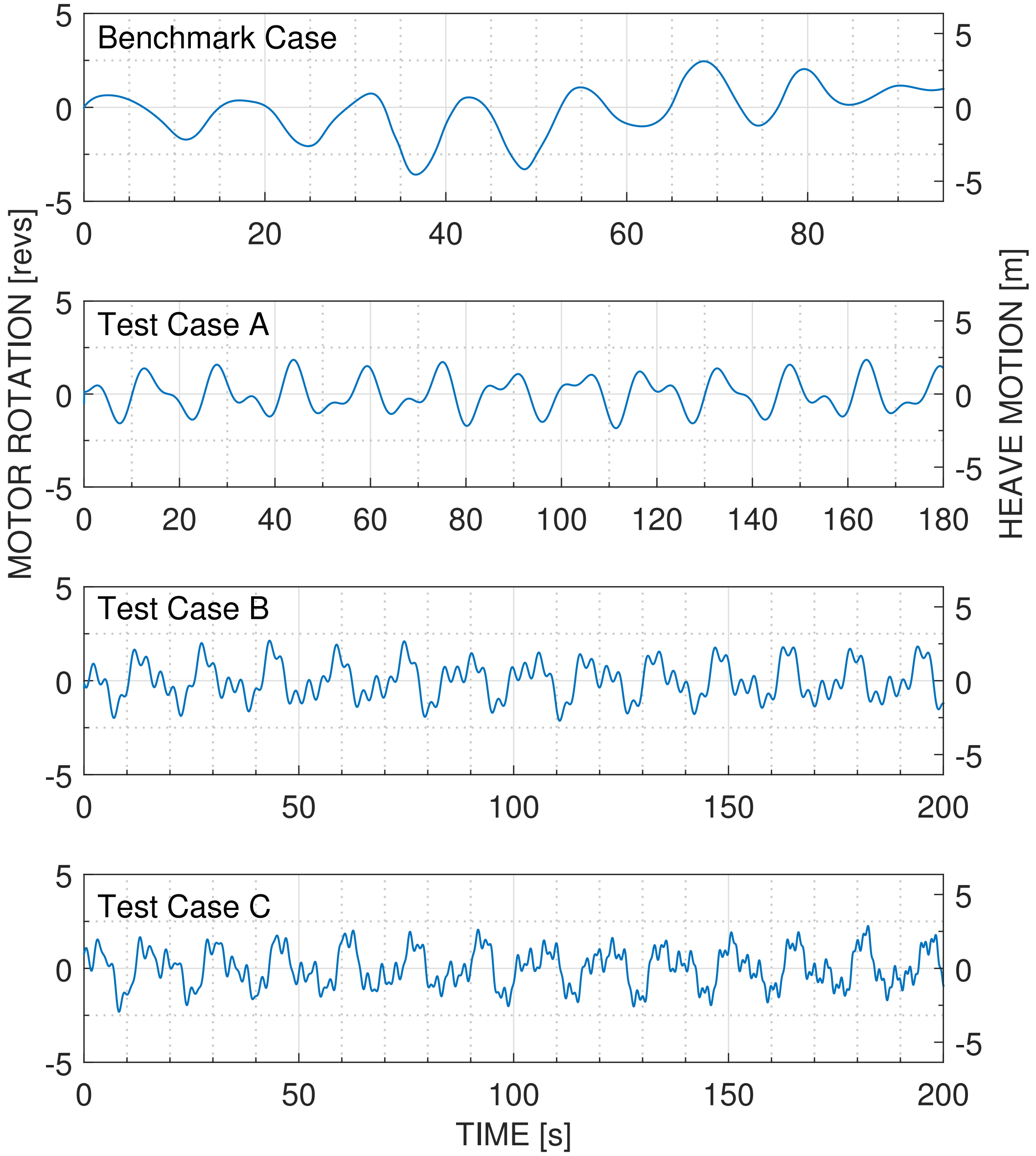


Figure 4.6: Benchmark Case data was taken from the paper by Kuchler *et al.* [3]. Test cases A, B, and C were provided by RRC to represent three at-sea motion scenarios.

4.3.3 Tuning of MPC Weighting Parameters

The first MPC parameters examined were the state error weight Q and the control signal rate of change weight R from the cost function in Equation 4.1. The control signal weight P was set to 0 because the cost associated with u, the input signal, is related to energy input into the system and we are not interested in limiting energy input. A position feedback MPC controller was used for the work in this thesis so a single integration of the velocity model obtained in Chapter 3 provides the transfer function

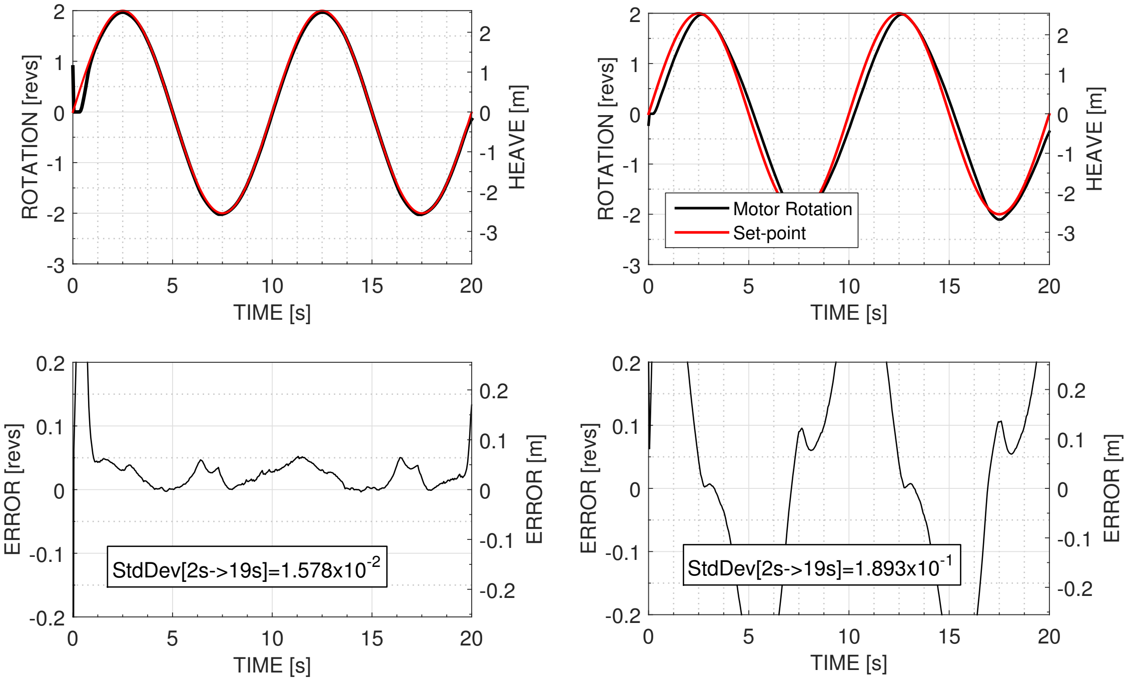
$$\frac{\theta[revs]}{V} = \frac{26.6167}{s^2 + 23.78s} \tag{4.14}$$

used for the MPC controller where θ is the motor rotation in revolutions, and V is in volts.

To test the MPC controller for acceptable Q and R values a sine wave at 0.1 Hz with an amplitude of 2 revolutions was used as the reference and the horizons, N_p and N_c, were set to 10 and 1, respectively. Properly tuned Q and R values will minimize the error standard deviation. Table 4.1 summarizes the results of testing a range of Q and R values, showing that $Q = 10$ and $R = 0.5$ provide the best performance compared to the other values tested. Figure 4.7a shows the motion data for sine testing with two axis: the left showing direct motor rotation and error, and the right showing the conversion from motor rotation to heave motion compensation given a 16 inch (0.4064 m) diameter winch drum. Comparing Figure 4.7a with the sub-optimal results from $Q = 5$ and $R = 10$ in Figure 4.7b there exists a distinct delay between the set-point and the motor position caused by putting more weight on reducing changes in control signal. For $R >> Q$ the MPC controller becomes unable to follow a moving reference due to the heavy weight on changes to the controller output. Figure 4.8 shows the MPC control action for Q = 10 and R = 0.5, Q = 10 and R = 10, and Q = 0.5 and R = 10. When the cost function parameter Q is weighted heavily the control action is most aggressive as shown by the large initial voltage spike. In contrast, when R is weighted more heavily than Q, then control action is slow and less aggressive. The best controller in Table 4.1 also happens to be the most aggressive, suggesting an aggressive controller provides the best tracking results in the sine tracking case.

Table 4.1: Identifying Tuned MPC Cost Parameters

Q	**R**	**Error StdDev [revs]**
10	*0.5*	*0.0158*
10	1	0.0205
10	3	0.0363
10	5	0.0568
10	7	0.0770
10	10	0.100
5	10	0.189
3	10	0.283
1	10	0.661
0.5	10	0.939



(a) Tracking a moving reference with acceptable cost function weights of $Q = 10$ and $R = 0.5$.

(b) Tracking a moving reference with poor cost function weights of $Q = 5$ and $R = 10$.

Figure 4.7: A comparison between acceptable cost function weights (a) and poorly chosen cost function weights (b).

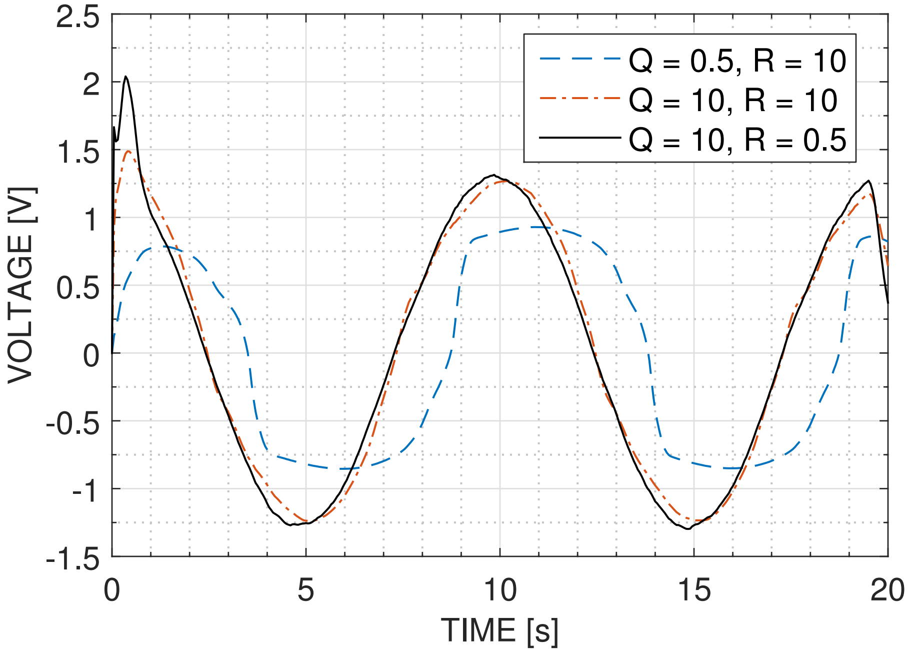


Figure 4.8: Three control action curves are shown for different MPC cost weights. The most aggressive behavior occurs for Q = 10 and R = 0.5 demonstrated by the large initial voltage spike. When R is heavily weighted, the control action is very slow to change.

4.3.4 Tuning of MPC Horizons

Tuning of the MPC controller prediction and control horizons N_p and N_c was performed by minimizing the tracking error standard deviation for tracking the Benchmark Data which was presented in Section 4.3.2. Two tests were performed for tracking the Benchmark Data: the first test was to use MPC and track the Benchmark Data while providing the controller with the exact future heave data for previewing; the second test provided the MPC controller with predicted future heave data instead of the exact future heave data for previewing. Once values of N_p and N_c were chosen MPC performance was tested for Test Case A and Test Case B to determine if the Benchmark Data horizon values also minimized error for the two test cases.

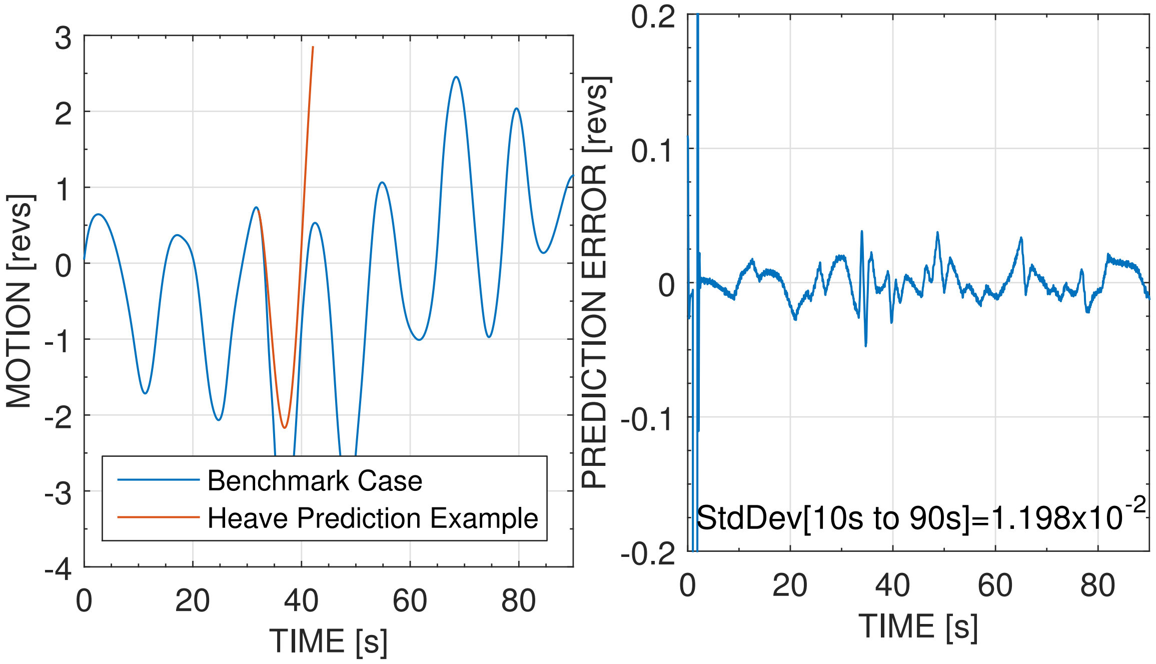
Table 4.2 summarizes the results of each test providing the tracking error standard deviation for optimal previewing in the optimal column and for heave prediction previewing in the heave column. The Benchmark Data results as well as the Test Case A and B results are given with all units in revolutions. In testing with the Benchmark Data, $N_p = 5$ and $N_c = 1$ were chosen as the best horizon options as these values reduce error by 38% over the next best settings of $N_p = 5$ and $N_c = 3$. In Test Case A, heave prediction has a 3% lower error standard deviation for $N_p = 5$ and $N_c = 5$ than for the best values from the Benchmark Case, but for Test Case B the $N_p = 5$ and $N_c = 1$ values again show the lowest error by 6%. Notice that in each case the optimal prediction errors are lower than the heave prediction errors. Figure 4.9 shows why the optimal prediction performs better compared to the heave predition.

Figure 4.9a shows the full set of Benchmark Case heave data in blue, and the heave prediction data from the heave prediction algorithm in red where the prediction algorithm started predicting at 32 s. Notice that as time progresses, the heave prediction data deviates from the actual test case data. The deviation between the heave prediction and the actual heave motion data is the reason that the optimal prediction errors in Table 4.2 are lower than the heave prediction errors.

For each test case the heave data was processed providing a future prediction of heave motion at each time-step based on the heave prediction algorithm developed in Section 4.2. Figure 4.9a shows this prediction at a single time-step of the Benchmark Data and, as expected, moving forward in time from the start of the prediction reduces the prediction accuracy as indicated by the curves diverging. This divergence occurs because the system is not completely predictable. In Figure 4.9b the error between each heave prediction and the actual heave data at 0.25 s (5 time steps) in the future

Table 4.2: Identifying ideal MPC Horizons using tracking error standard deviation [revs].

N$_p$	**N$_c$**	**Benchmark Case**		**Test Case A**		**Test Case B**	
		Optimal	Heave	Optimal	Heave	Optimal	Heave
5	*1*	***0.0129***	***0.0157***	*0.0131*	*0.0137*	*0.0218*	*0.0336*
5	3	0.0199	0.0218	0.0107	0.0138	0.0289	0.0356
5	5	0.0204	0.0233	0.0107	0.0133	0.0288	0.0359
10	1	0.0129	0.0245	0.0182	0.0342	0.0515	0.0886
10	3	0.0211	0.0245	0.0117	0.0141	0.0246	0.0364
10	5	N/A	N/A	0.0119	0.0137	0.0290	0.0361
10	10	0.0207	0.0235	0.0112	0.0135	0.0286	0.0365



(a) Benchmark Data with a representative wave prediction curve at t=32s plotted for 10 seconds in the future.

(b) Error between actual Benchmark Data and wave prediction values for Test Case A at 0.25 s (5 time steps) in the future.

Figure 4.9: An example of wave prediction for a single time-step is shown in (a), with the error at 0.5 s into the future for every time step in (b).

is shown at each time-step, with the average error being 0.0120 motor revolutions (0.015 m). The large initial error in Figure 4.9b is due to the observer states not having converged on the actual states until 3 s. This initial observer error is why the error standard deviation is calculated from 10 s onward, to ensure time for the observer states to converge on the actual states. The heave prediction observer gains

L for the prediction in Figure 4.9 were chosen aggressively as the data is noise free allowing the observer to converge quickly on the states. In Section 4.4 it will be shown that, for data sets containing noise, aggressive heave observer gains result in poorer performance as the noise reduces accuracy of the observer state estimates.

4.3.5 PID Tuning

Tuning of the PID controller parameters was performed via trial-and-error. Just as the MPC controller output was adjusted for the system deadband and non-linear gain as described in Section 4.3.1, the PID controller output was also adjusted. Furthermore, the equations used for the PID controller output as implemented in LabVIEW were:

$$\begin{aligned} e(k) &= SP(k) - PV(k) \\ u_p(k) &= K_p e(k) \\ u_i(k) &= u_i(k-1) + \frac{K_p}{T_i}\Big(\frac{e(k)+e(k-1)}{2}\Big)\Delta T \\ u_d(k) &= -K_p \frac{T_d}{\Delta T}\Big(PV(k) - PV(k-1)\Big) \\ u(k) &= u_p(k) + u_i(k) + u_d(k) \end{aligned} \tag{4.15}$$

where

$SP(k)$	=	Setpoint at the current time step, k [revs]
$PV(k)$	=	Process variable at the current time step,k [revs]
$e(k)$	=	Error between setpoint and process variable [revs]
K_p	=	Controller gain [V/rev]
T_i	=	Integral time constant [min]
T_d	=	Derivative Time Constant [min]
ΔT	=	Control Loop Time [min]
$u_{p,i,d}(k)$	=	Proportional, Integral, and Derivative control terms [V]
$u(k)$	=	Controller output [V]

The PID integral term was additionally clamped between ±3 V to ensure integral wind-up did occur which could make the controller unstable. The controller gain term K_p was initially tested at 5, with integral and derivative terms disabled, however, it was found that a value of $K_p = 8$ showed improved results for tracking a step-input.

The derivative term was then included in testing, and it was found $T_d = 0.0005$ resulted in reasonable step tracking without overly damping the system or becoming unstable. Finally, integral action was added to the controller and a value of $T_i = 0.005$ achieved better step tracking compared to the P or PD controllers. Figure 4.10 shows the results of step tracking for the P, PD, and PID controllers which were found to have the best tracking results and Table 4.3 provides a summary of the best tuned PID values, as well as the integral of the tracking error which was used to determine the quality of tracking.

In Figure 4.10 the proportional controlled motor (top) overshoots the set-point by 0.155 revs before undershooting the set-point and finally reaching steady-state. Improving upon the proportional controller, the PD controller (center) reduces overshoot to 0.037 revs and no longer under-shoots the set-point. Finally, the PID controller reduces over-shoot to 0.027 revs and again, avoids additional under-shoot of the set-point.

Table 4.3: PID Tuning Parameters and Error Integral

K_p	T_i	T_d	**Integral of Error**
8	∞	0	-1.237
8	∞	0.0005	-1.176
8	0.005	0.0005	-1.042

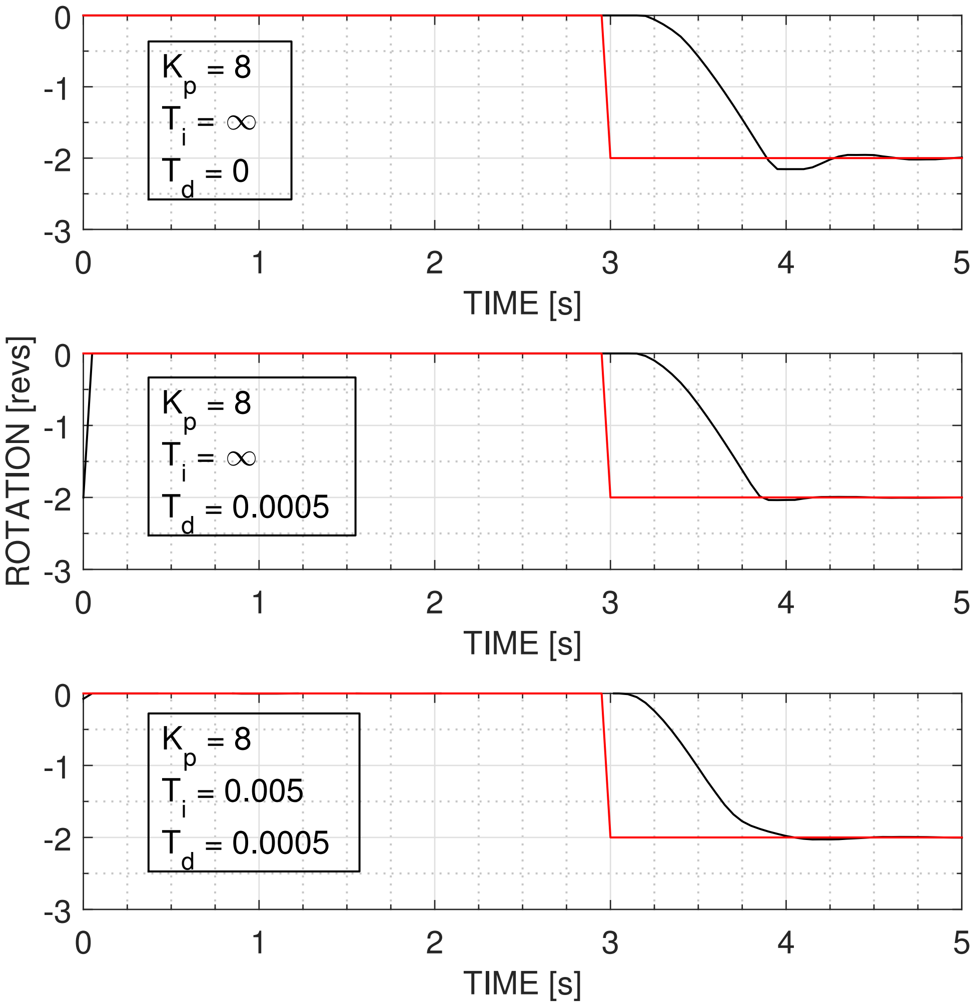


Figure 4.10: The step-responses for each of the cases shown in Table 4.3 are presented here

4.4 MPC Performance Compared to PID for AHC Testbed Actuation

With acceptable parameters determined for both MPC and PID controllers, comparisons can be made between the two controllers for the simple case of a sine reference input, and also for the test cases as described in Section 4.3.2. To numerically compare the PID and MPC controllers, the standard deviation of the tracking error is used as a metric. It is desired to have a low error standard deviation as this means reduced unwanted motion while ignoring any constant offset.

4.4.1 Sine Tracking

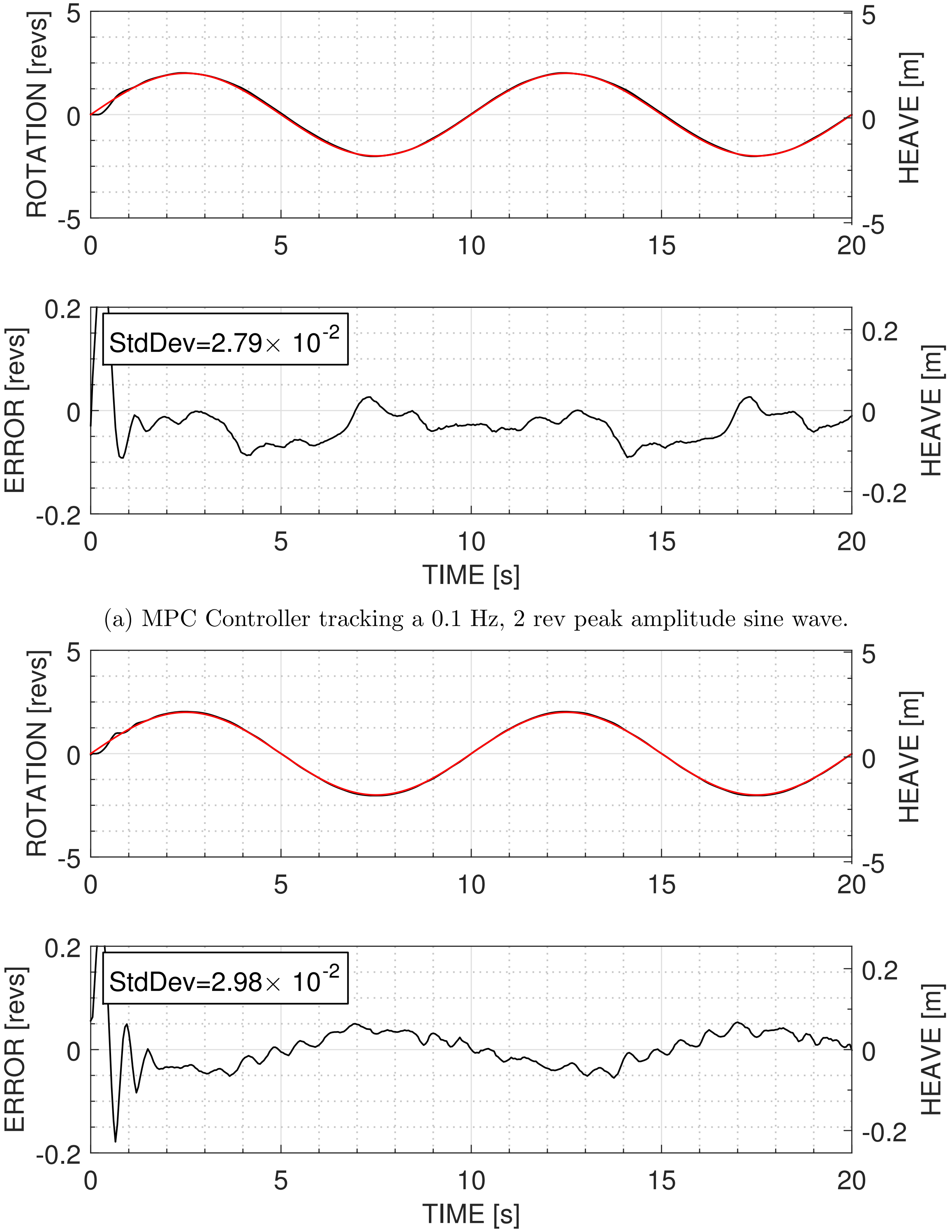
The tuned MPC and PID controllers are shown tracking a 0.1 Hz, 2 revolution peak sine wave in Figure 4.11 where the X-axis is time in seconds, the left Y-axis is the motor rotation in revolutions, and the right Y-axis is the heave motion in meters which would correspond to the given motor rotation from the left Y-axis with a 0.4046 m diameter drum attached to the motor. For the 0.1 Hz sine wave, the MPC controller appears to be comparable to the PID controller with only a 1.90×10^{-3} revs difference between their standard deviations. Ignoring initial transients, the largest tracking errors in Figure 4.11a occur when the system was rotating at a negative velocity between times $t = 4$ s to 6 s and $t = 14$ s to 16 s. This large error at these times suggests that the system model and non-linear corrections used for the MPC controller are not symmetric and cannot correct as well when the motor rotates at a negative velocity. At each of the large error times the system is lagging behind the set-point meaning the actual system rotates slower in the negative direction than the TF model used by the MPC controller. Figure 4.5, seen previously, confirms this slower negative angular velocity, where in the linear region, for a given negative control voltage, the actual system will rotate slower than the TF model would specify.

For the PID controller in Figure 4.11b the largest tracking errors occur when the motor is stoping and changing direction at times $t = 2.5$ s, 7.5 s, 12.5 s, and 17.5 s. At each of these times the system overshoots the set-point showing the PID action consistently lags behind set-point changes.

Table 4.4 shows that as the tracked sine frequency is increased from 0.1 Hz to 0.2 Hz the PID controller provides poorer tracking results with error increasing from 2.98×10^{-2} revs to 1.076×10^{-1} revs — a change of 7.78×10^{-2} revs or 261%, while the MPC controller is more robust under changing frequencies with an error increase from 2.79×10^{-2} revs to 3.60×10^{-2} revs — a change of 8.1×10^{-3} revs or 29%. The error difference column provides the difference at each frequency between the two controller errors. As frequency increases the error between the controllers also increases, showing that the PID controller performance degrades quicker than the MPC controller.

Table 4.4: MPC and PID Tracking Error for Ramping Sine Frequency

Sine Freq. [Hz]	**MPC Err. [revs]**	**PID Err. [revs]**	**Err. Diff. [revs]**
0.1	0.0279	0.0298	0.0019
0.125	0.0242	0.0457	0.0215
0.15	0.0270	0.0651	0.0381
0.2	0.0360	0.1076	0.0716



(a) MPC Controller tracking a 0.1 Hz, 2 rev peak amplitude sine wave.

(b) PID Controller tracking a 0.1 Hz, 2 rev peak amplitude sine wave.

Figure 4.11: A comparison between tuned MPC and PID controllers tracking a 0.1 Hz, 2 rev peak amplitude sine wave.

4.4.2 Test Cases

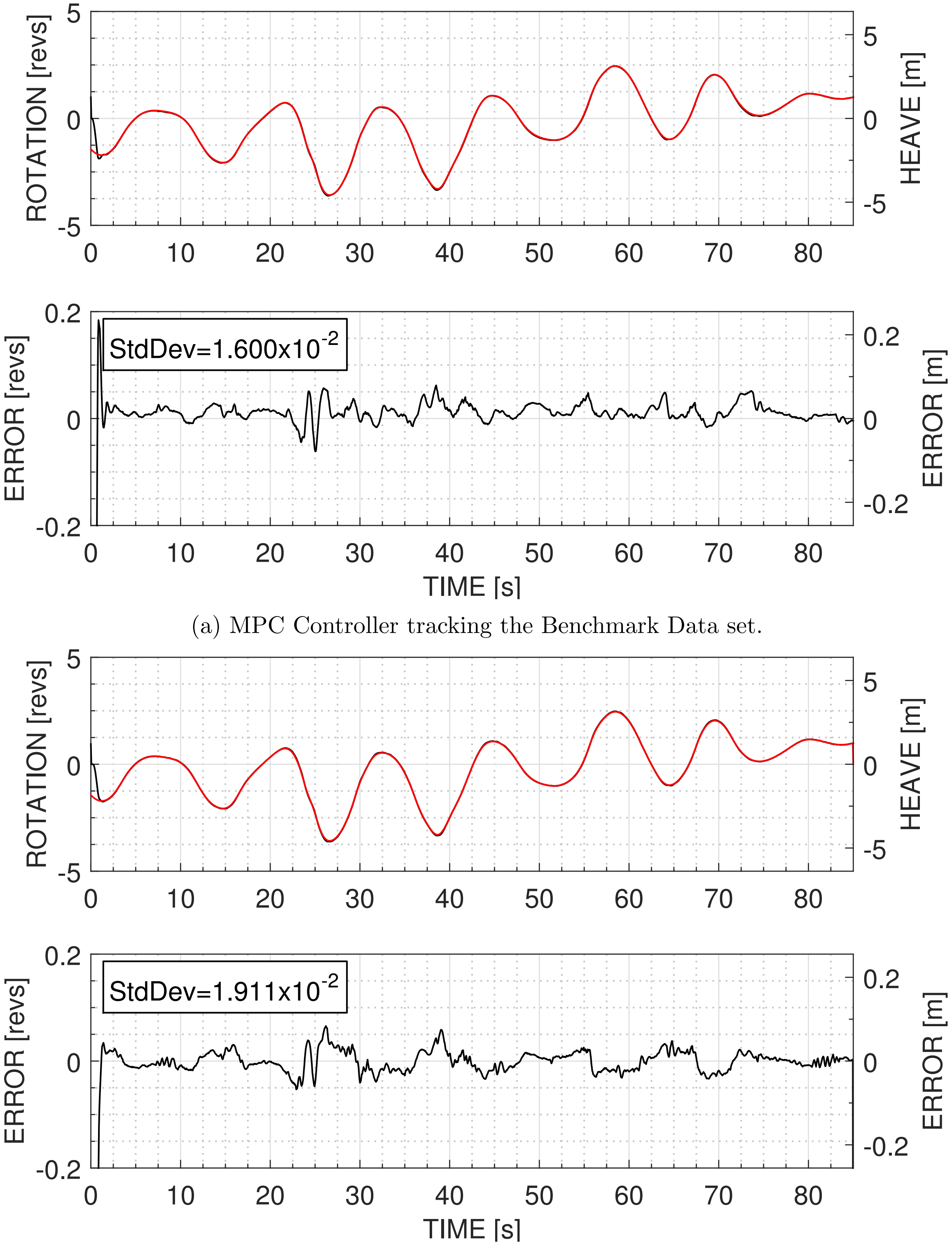
Figures 4.12 to 4.15 show the MPC controller and the PID controller experimental reference tracking results for the Benchmark case and for Test Cases A, B, and C. The upper plot in each figure gives the motor set-point data and the system response with the left Y-axis representing motor revolutions and the right Y-axis corresponding to the equivalent heave motion for a 16 in (0.4064 m) diameter winch drum. The lower plot in each figure shows the tracking error of the controller where, again, the left Y-axis shows error in motor revolutions and the right Y-axis shows the equivalent error for a 16 in (0.4064 m) diameter winch drum. In each case, the better performing controller will have a lower error standard deviation as printed in each of the error figures.

Figure 4.12 shows the results of the MPC and PID controllers tracking the Benchmark Data from Kuchler *et al.* [3]. Figure 4.12a shows the MPC controller tracking the Benchmark test case with an error standard deviation of 1.600×10^{-2} revs, or 2.042×10^{-2} m in heave, while in Figure 4.12b the PID controller tracks the Benchmark test case with a larger error standard deviation of 1.911×10^{-2} rev, or 2.434×10^{-2} m in heave — an increase of 3.20×10^{-3} rev, or 4.09×10^{-3} m. As a point of comparison, Kuchler *et al.* [3] using wave prediction and a non-linear control design were able to achieve an error standard deviation of 5.3×10^{-2} m on a loaded AHC testbed for similar heave motion to the Benchmark test. Having a loaded testbed and different test data makes the comparison to Kuchler *et al.* difficult, but the standard deviation value of 5.3×10^{-2} m, or 4.2×10^{-2} revs shows that the error values achieved by the MPC and PID controllers are reasonable.

Figure 4.13 shows the results of the MPC and PID controllers tracking the Test Case A data. Recall that the Test Case A data was provided by RRC to represent simple heave motion of a vessel at sea. Figure 4.13a shows the MPC controller tracking Test Case A with an error standard deviation of 1.377×10^{-2} rev, or 1.758×10^{-2} m in heave while in Figure 4.13b the PID controller tracks Test Case A with a larger error standard deviation of 1.704×10^{-2} rev, or 2.176×10^{-2} m in heave — an increase of 3.27×10^{-3} rev, or 4.18×10^{-3} m. For both the Benchmark test data and the Test Case A data, the MPC controller shows improved tracking over the PID controller by a very similar amount: 3.20×10^{-3} rev compared to 3.27×10^{-3} rev. The Benchmark and Test Case A sets are comparable, each with relatively low frequency smooth motions, so it is reasonable that the difference between MPC and PID in each case would also be similar.

Figure 4.14 shows the results of the MPC and PID controllers tracking the Test Case B data. Recall that the Test Case B data was provided by RRC to represent simple heave motion of a vessel at sea with an additional higher frequency component added. Figure 4.14a shows the MPC controller tracking Test Case A with an error standard deviation of 2.908×10^{-2} rev, or 3.710×10^{-2} m in heave while in Figure 4.14b the PID controller tracks Test Case A with a larger error standard deviation of 7.235×10^{-2} rev, or 9.237×10^{-2} m in heave — an increase of 4.327×10^{-2} rev, or 5.524×10^{-2} m. Adding a higher frequency component has increased the PID error by 149% over the MPC error whereas in the previous sets the error increased by 19% for the Benchmark test data and by 24% for the Test Case A data. This large increase in error for the PID controller matches what was seen in Section 4.4.1 where increasing the frequency of a tracked sine increased tracking error for PID more than MPC, suggesting the MPC controller is robust over a wider range of potential heave motions compared to the PID controller.

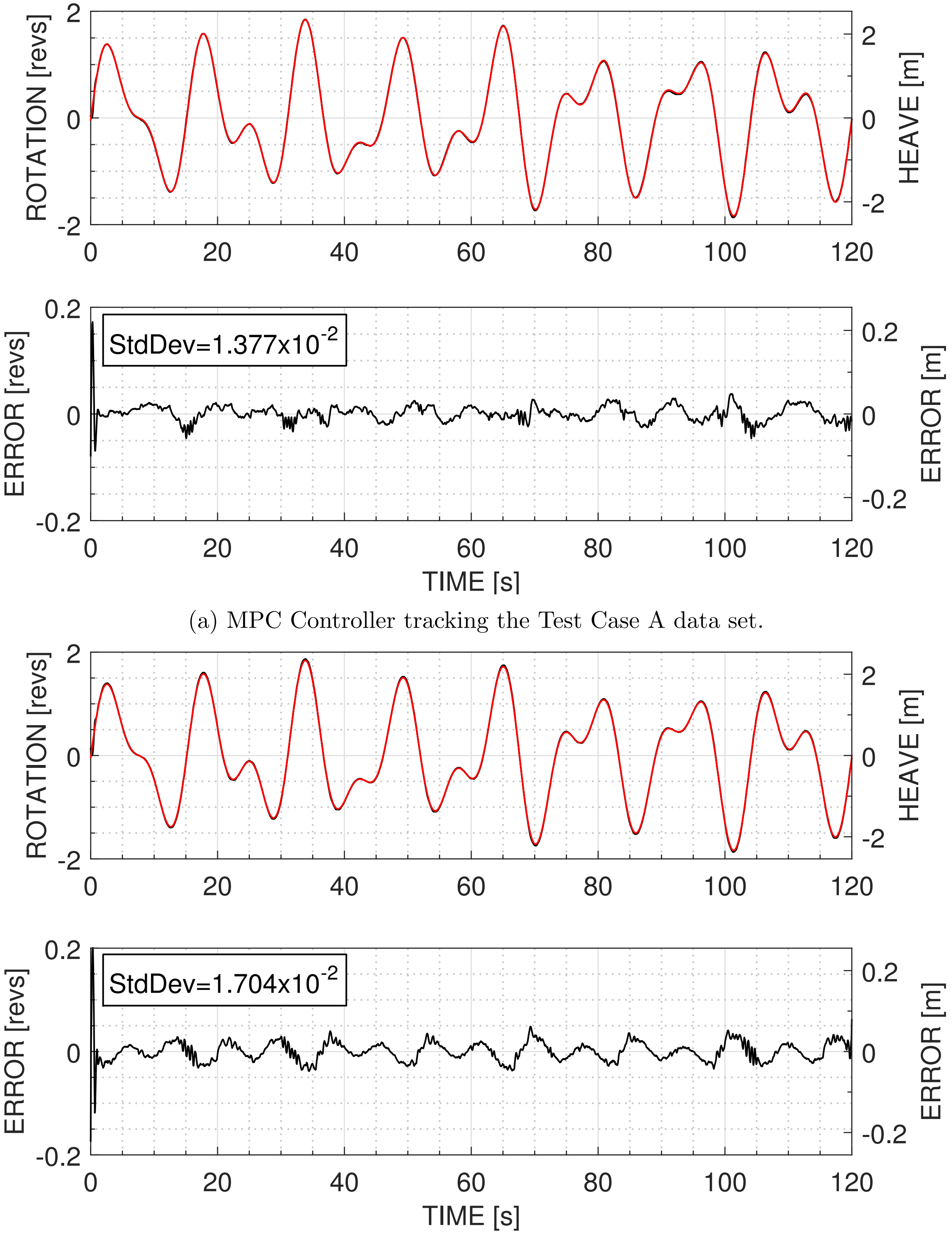
Figure 4.15 shows the results of the MPC and PID controllers tracking the Test Case C data. The Test Case C data represents changing conditions for a vessel at sea by adding a sinusoidal component with increasing frequency as time progresses. The added sinusoidal component in Test Case C is additionally of higher frequency than the added component in Test Case B. Figure 4.15a shows the MPC controller tracking Test Case C with an error standard deviation of 8.765×10^{-2} rev, or 1.119×10^{-1} m in heave while in Figure 4.15b the PID controller tracks Test Case C with a larger error standard deviation of 1.528×10^{-1} rev, or 1.950×10^{-1} m in heave — an increase of 6.511×10^{-2} rev, or 8.312×10^{-2} m. In Test Case C, similar to the previous cases the MPC controller shows improved results over the PID controller; however, both controllers have difficulty accounting for the small, high frequency components introduced in Test Case C as seen by the large errors and sharp changes in the error plot. These sharp, quick changes in the error plots occur during direction changes in the rotation graphs where the controllers both overshoot the set-point. The most likely cause of these overshoots is the system being physically unable to change speed as rapidly as desired. Also notice that the error plot amplitudes appears to grow as time progresses for each case. Again, this is explained by the increasing frequency component adding further inability for the system to physically compensate for the direction changes.



(a) MPC Controller tracking the Benchmark Data set.

(b) PID Controller tracking the Benchmark Data set.

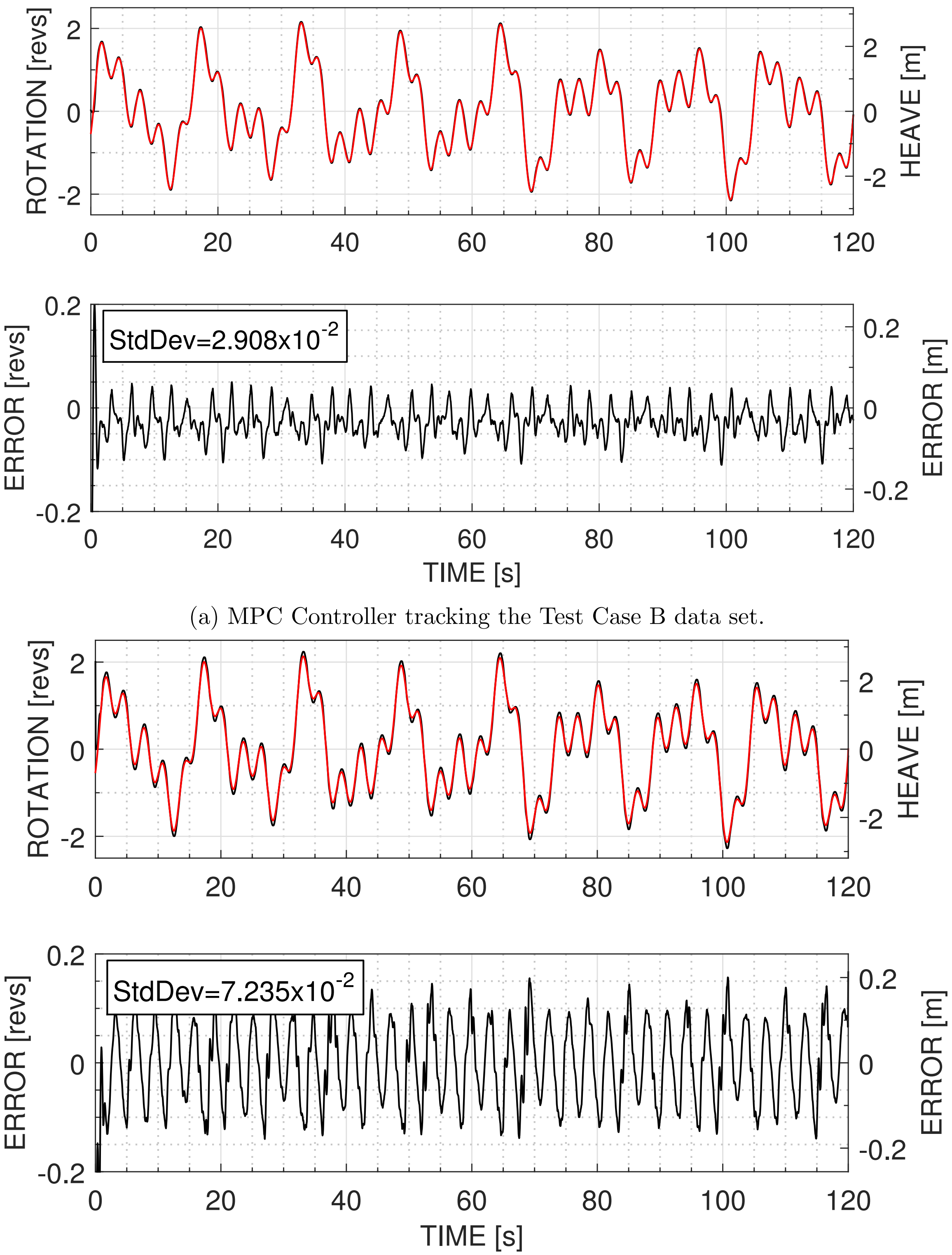
Figure 4.12: A comparison between tuned MPC and PID controllers tracking the Benchmark Data; data used by Kuchler *et al.*[3] for testing their wave prediction algorithm.



(a) MPC Controller tracking the Test Case A data set.

(b) PID Controller tracking the Test Case A data set.

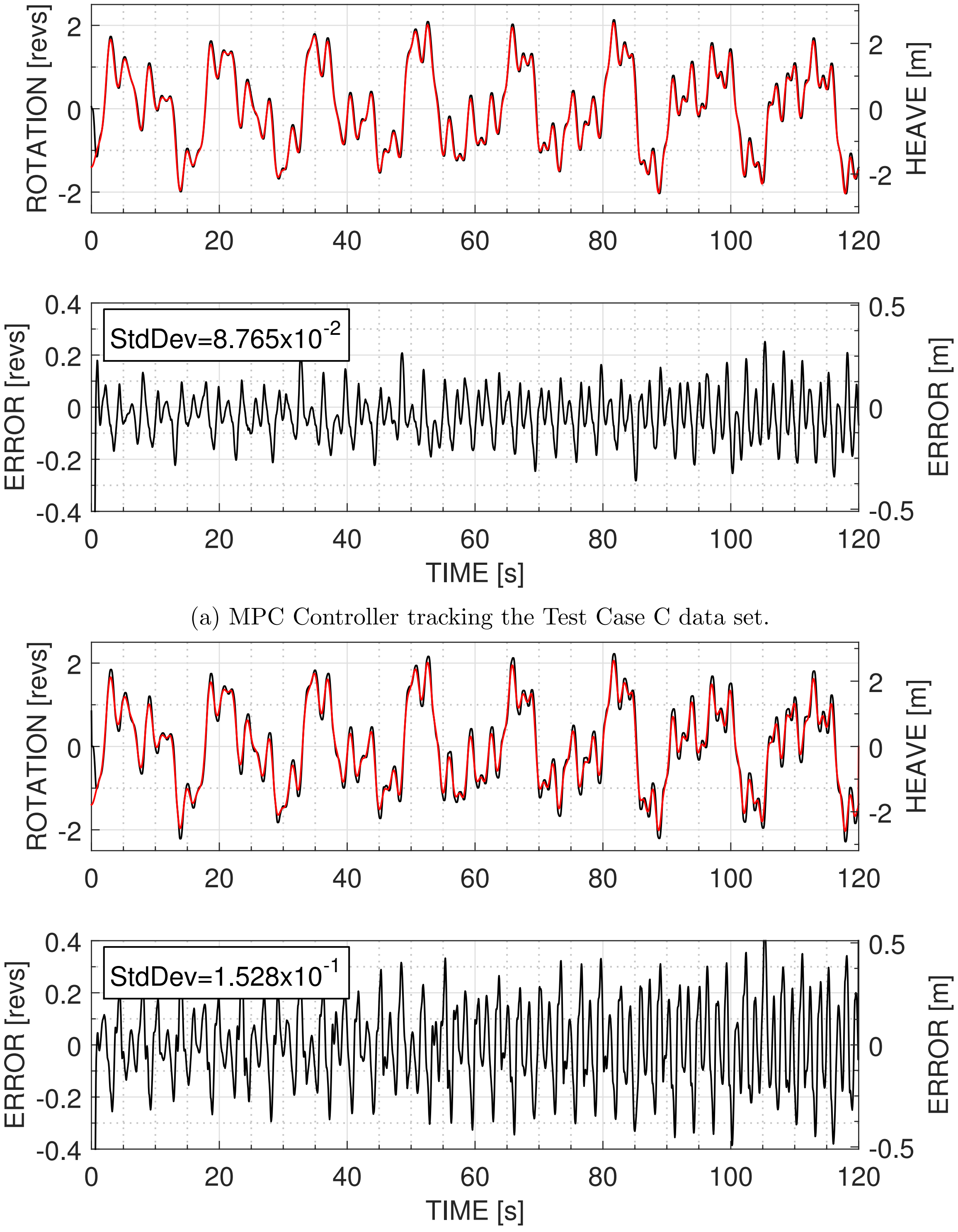
Figure 4.13: A comparison between tuned MPC and PID controllers tracking Test Case A; a signal provided by RRC to be representative of ship heave motion at sea.



(a) MPC Controller tracking the Test Case B data set.

(b) PID Controller tracking the Test Case B data set.

Figure 4.14: A comparison between tuned MPC and PID controllers tracking Test Case B; the Test Case A heave data with an additional high frequency heave component added.



(a) MPC Controller tracking the Test Case C data set.

(b) PID Controller tracking the Test Case C data set.

Figure 4.15: A comparison between optimized MPC and PID controllers tracking Test Case C; the Test Case A heave data with an additional modulating frequency component added.

4.4.3 Test Cases with Added Noise

As mentioned in Section 4.3.2, the RRC provided Test Cases A and B heave data were also generated with 0.05 peak amplitude, 20 Hz noise meant to represent a heave measurement signal with noise. Figure 4.16 shows a representative sample of Test Case A both with and without the above mentioned noise, where the time scale is reduced to 5 s of data to better display the difference between each data set. In an ideal scenario any noise added to the heave signal would be filtered prior to the controllers, however, it is important to analyze the results of a noisy signal for the case where filtering is not possible.

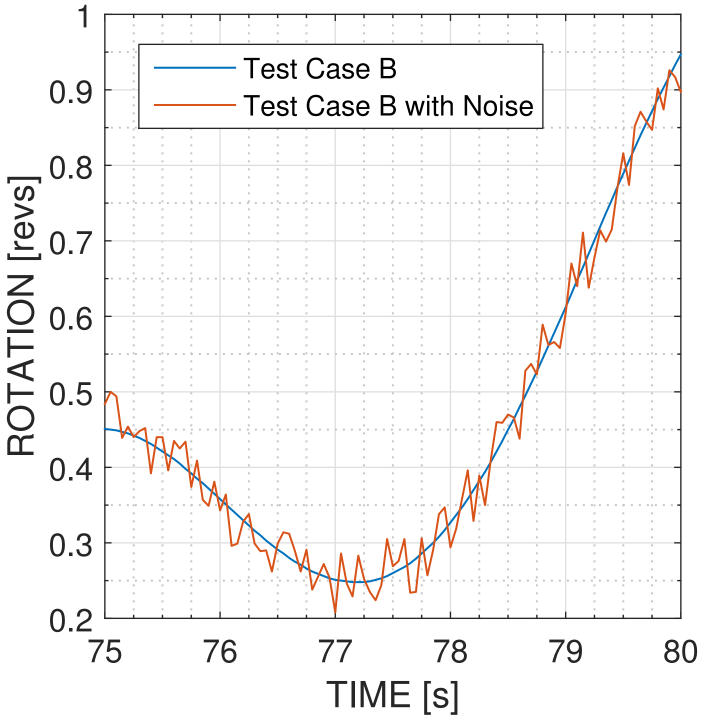
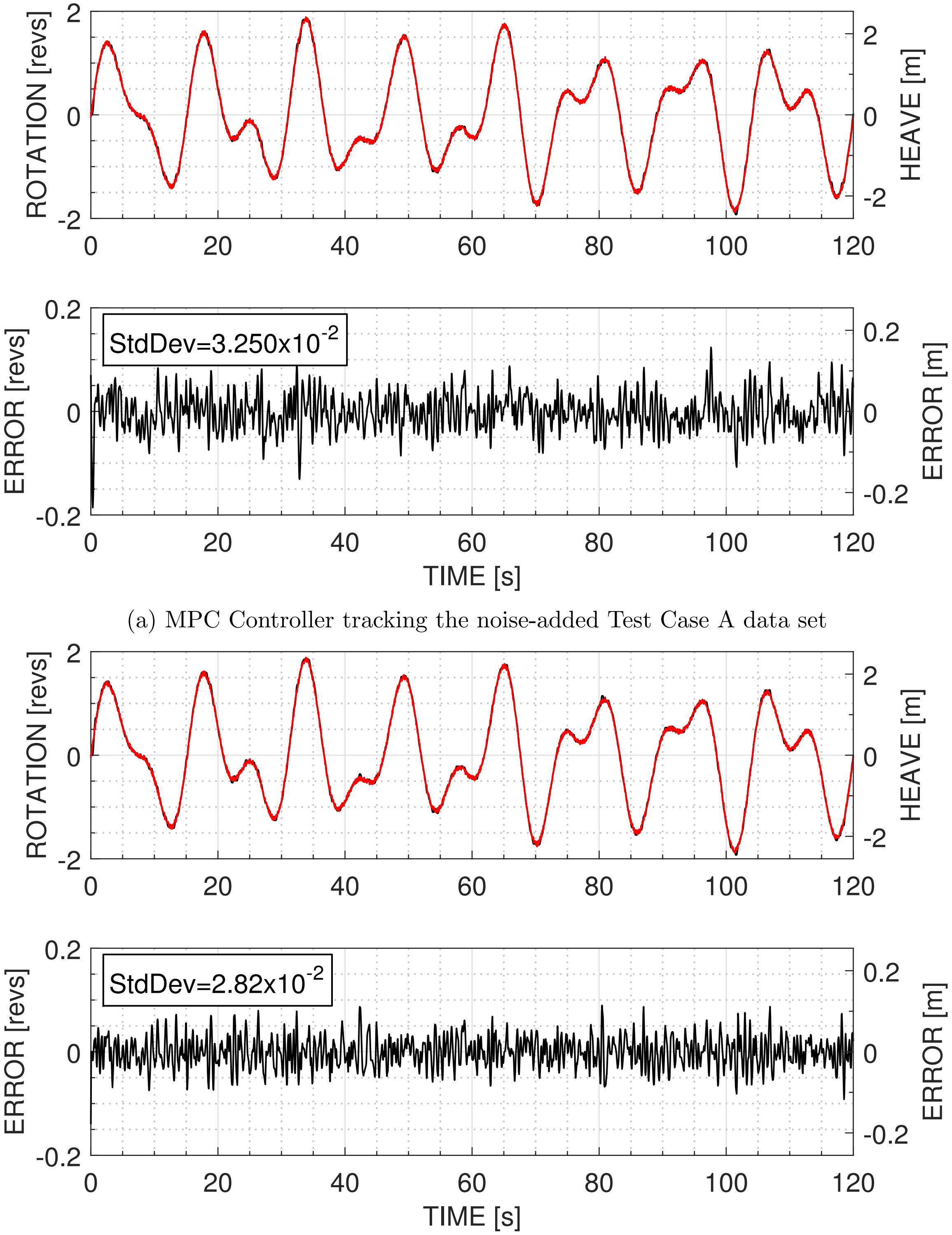


Figure 4.16: Test Case A is shown here with and without the 0.05 peak amplitude, 20 Hz noise added for a narrow time region.

Comparing Figure 4.13, the controller response for Test Case A without noise, to Figure 4.17, the controller response for Test Case A with noise, we see that adding noise has increased the tracking error for the MPC controller by 1.873×10^{-2} revs (2.391×10^{-2} m), and for the PID controller by 1.116×10^{-2} revs (1.424×10^{-2} m). A larger increase in tracking error for the MPC controller when compared to the PID controller suggests that MPC is more sensitive to signal noise than PID for the given conditions.

In Test Case B with added noise Figure 4.18 shows an increase in error standard deviation for both controllers compared to the noise-free test case in Figure 4.14; however, because the PID error for Test Case B was already 4.327×10^{-2} revs (5.524×10^{-2} m) larger than the MPC error, the data in Figure 4.18 shows that the MPC controller tracks better than the PID controller. Note that that increase in error from the clean test case to the test case with noise for MPC in Test Case B was 1.781×10^{-2} revs while for PID it was smaller at 8.25×10^{-3} rev, again suggesting that MPC has a higher sensitivity to noise for the controllers used.

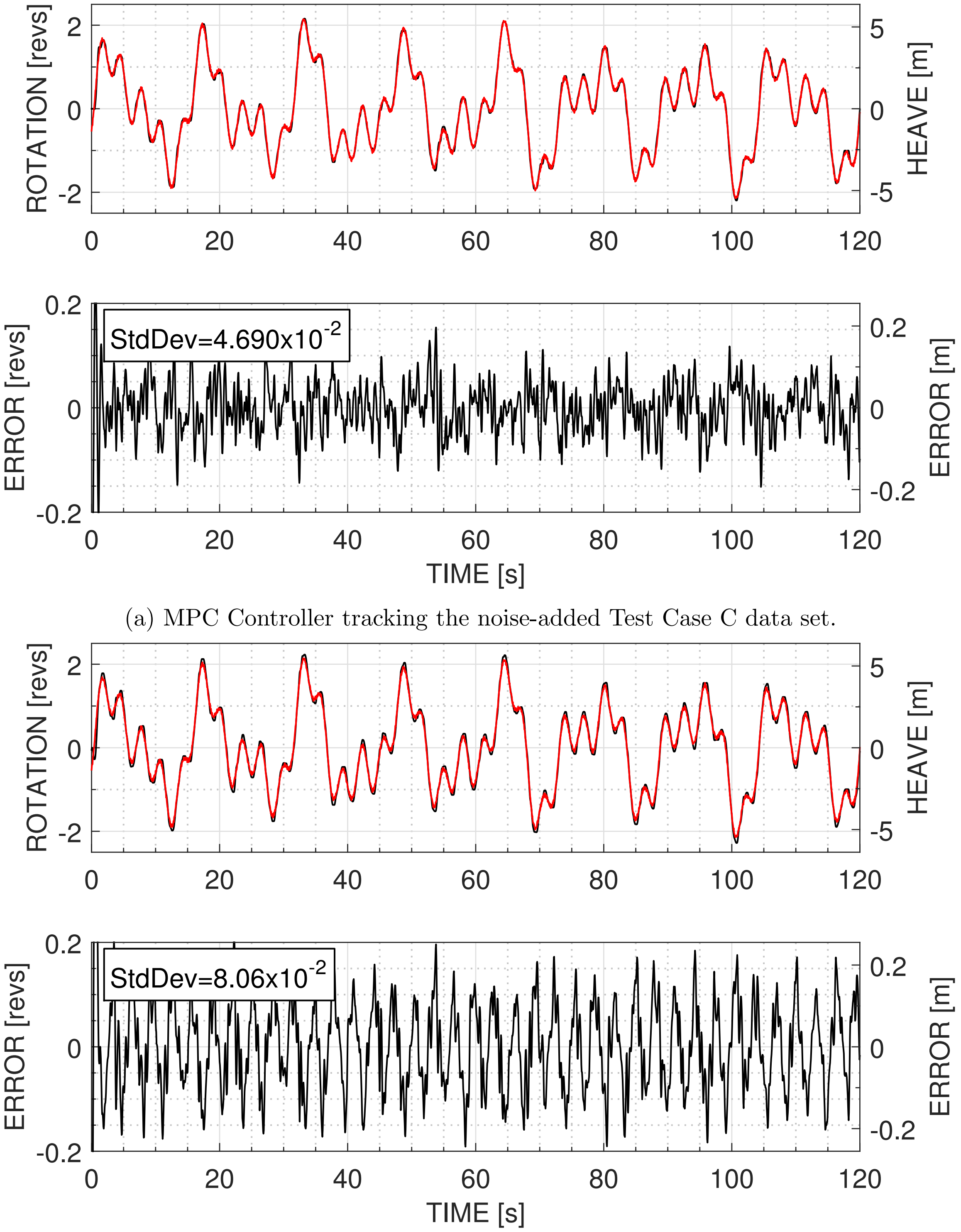
The higher noise sensitivty for MPC can be explained by Figure 4.19 where Figure 4.19a shows heave prediction error 5 steps (0.25 s) into the future without signal noise for Test Case A, and Figure 4.19b shows heave prediction error 5 steps into the future with signal noise for Test Case A. Recall that the MPC controller uses a heave prediction algorithm to provide future heave predictions. The heave prediction algorithm relies on an observer to update the estimate for heave amplitude and phase at each time step, based on the measured heave motion. The observer gains affect how well the observer converges to accurate estimates of the heave amplitude and phase. With relatively low observer gains, the observer estimation will converge more slowly to accurate state estimates, but will be more robust against noise. The observer gains in this thesis are relatively high, so, the system converges quickly to accurate estimates, however, the estimates are going to be more negatively impacted by noise in the heave signal as is common with high gain observers. By adding noise, the average error in heave prediction for 5 steps into the future increases by 0.06417 revs, or approximately a factor of 8. This increase in error means that the MPC controller previewing data is providing poorer future heave prediction values, which result in control actions providing poorer tracking.



(a) MPC Controller tracking the noise-added Test Case A data set

(b) PID Controller tracking the noise-added Test Case A data set.

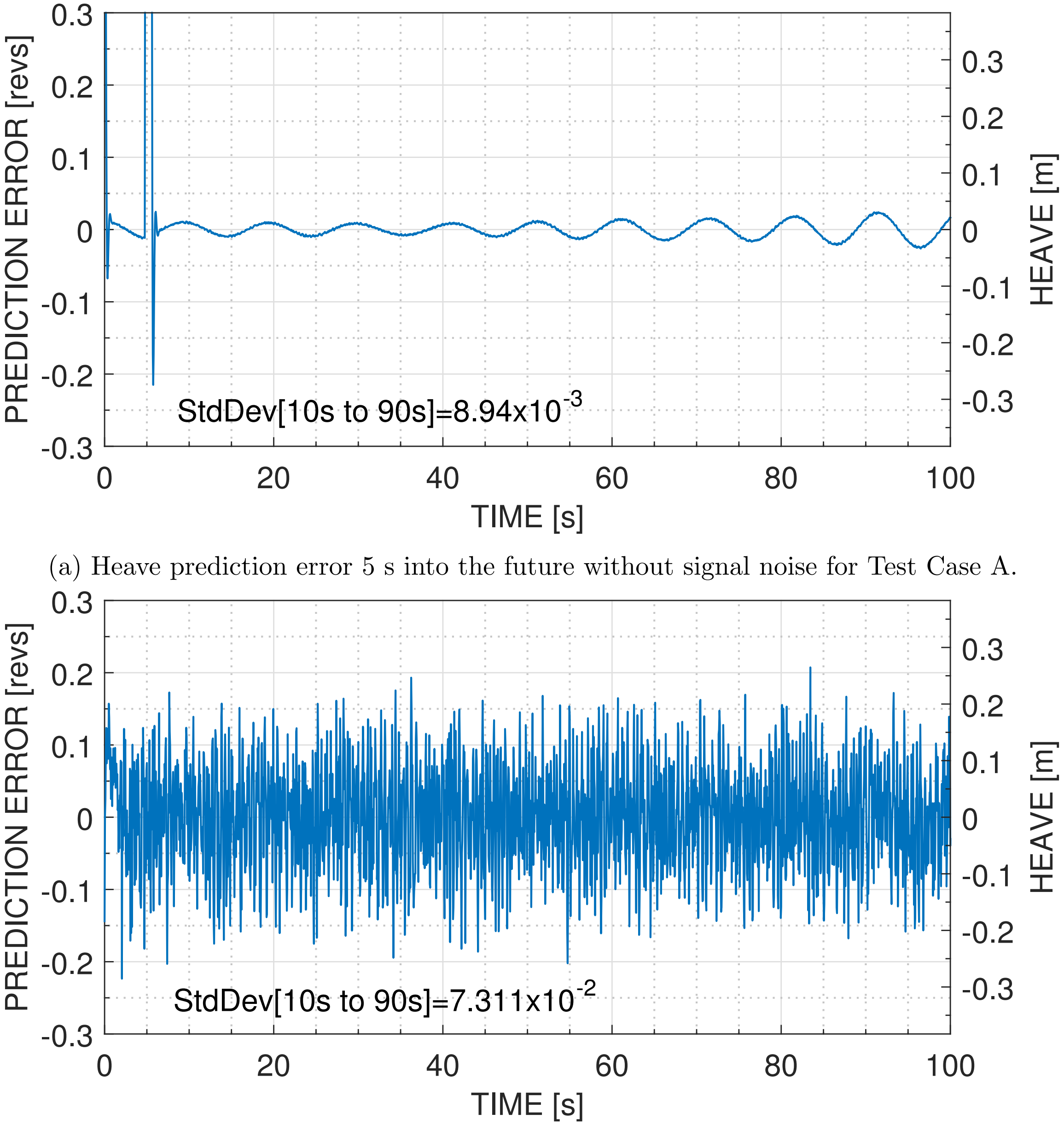
Figure 4.17: A comparison between optimized MPC and PID controllers tracking Test Case A data with added 20 Hz, 0.05 peak amplitude noise.



(a) MPC Controller tracking the noise-added Test Case C data set.

(b) PID Controller tracking the noise-added Test Case C data set.

Figure 4.18: A comparison between optimized MPC and PID controllers tracking Test Case C data with added 20 Hz, 0.05 peak amplitude noise.



(a) Heave prediction error 5 s into the future without signal noise for Test Case A.

(b) Heave prediction error 5 s into the future without signal noise for Test Case A.

Figure 4.19: A comparison between heave prediction errors 5 s in the future for Test Case A with and without noise. The error increases by a factor of 8 with the noise-added signal.

4.5 MPC Robustness to Model Errors

Model-predictive control relies upon identification of a system model. In Figure 4.20 the effects of changing the AHC testbed identified model corner frequency are shown by plotting error standard deviations for modified corner frequencies in each test case. The original TF model used for the MPC controller was

$$\frac{\omega}{V} = \frac{26.6167}{s^2 + 23.78s} = \frac{1.12}{(s)\left(\frac{s}{23.78} + 1\right)} \tag{4.16}$$

where the corner frequency is 23.78 rad/s. The modified corner frequencies were obtained by using the original model corner frequency 23.78 rad/s (3.785 Hz) and multiplying by 0.1, 0.6, 0.8, 1.2, 1.4, and 10.

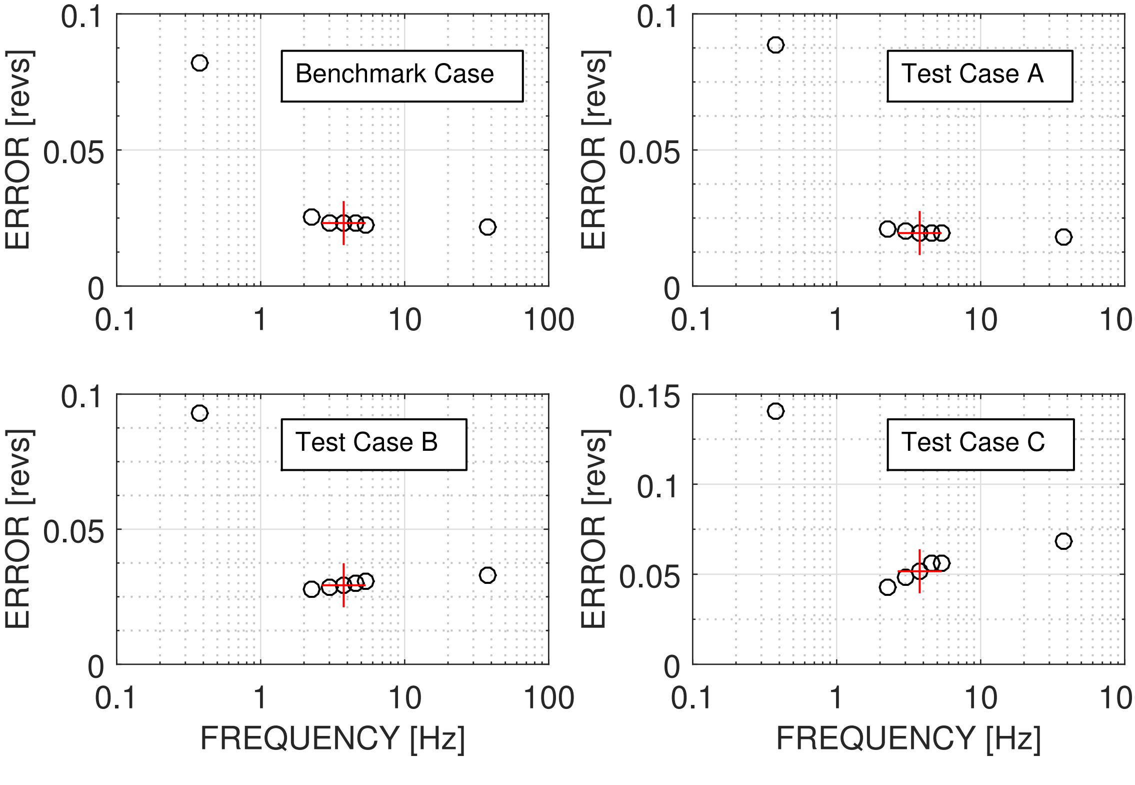


Figure 4.20: Here we compare the effects of changing the MPC model corner frequency for the four test cases. In each plot the corner frequency is changed by a factor of 0.1, 0.6, 0.8, 1.2, 1.4 and 10. The original model corner frequency is identified by a red +.

For the Benchmark Case as well as each Test Case, A, B, and C, reducing the corner frequency by a decade had the largest effect increasing the tracking error by 5.925×10^{-2}, 6.915×10^{-2}, 6.357×10^{-2} and 8.909×10^{-2} revs from the identified system error respectively. It is unlikely that a model corner frequency would be so poorly identified in a system as to be a decade lower or higher than the actual system. Focusing on only the range from 0.6 to 1.4 multiples of the model corner frequencies shows Test Case D exhibits the widest range of errors ranging from 4.234×10^{-2} revs to 5.570×10^{-2} — a 31% difference. This error difference of 31% for a poorly identified MPC model is below the 74% error difference between MPC and PID for Test Case D suggesting that, for this system, MPC control with identified model errors provides improved control over the PID controller used.

4.6 Conclusion

In this Chapter an MPC controller and a PID controller were designed and compared under a number of test conditions. It was shown that as higher frequency components are introduced into a moving reference signal the MPC controller designed in Section 4.3 is more robust in tracking the reference signal when compared to the designed PID controller. This robustness is important as the operating conditions for an AHC system may not always be known and the operators cannot be relied upon to know how to tune a system for changing conditions. For the PID controller, although it is likely that values for K_p, T_i and T_d could be found to improve tracking under a range of conditions, an algorithm would be required to assess the optimal PID tuning values which would add another layer of complexity to the control design.

In Section 4.4.3, it was noted that the MPC controller suffers from a high sensitivity to reference signal noise due to the heave prediction algorithm having difficulty accurately predicting future motion. This difficulty in providing an accurate heave prediction resulted in Test Case A with added noise showing a higher tracking error for MPC when compared to the PID controller. As was mentioned at the beginning of Section 4.4.3, a low-pass filter would likely be used in a commercial system at sea, removing noise from the heave data and allowing the heave prediction algorithm to operate as expected.

Section 4.5 presented results showing the robustness of MPC to errors in the identified system model, allowing for some amount of variance in the system identification process. Allowing some model variation is important because the MPC controller and heave prediction algorithm implemented are generic, and could be applied to other AHC systems with different valves and motors so long as a system model was available. If the system model was not properly identified the MPC controller would still function and provide heave decoupling.

With the robustness of the MPC controlled AHC testbed proven, the next step in testing is to apply a load to the AHC testbed; however, since testing the full scale AHC testbed under load is not within the scope of this thesis work, Chapter 5 will focus on the creation of a MATLAB Simulink model of the AHC testbed which can be loaded without requiring additional hardware.

Chapter 5

Creating a Simulator Model to Represent the Active Heave Compensator Testbed

In Chapter 4 results were presented for an unloaded, full-scale hydraulic active heave compensation testbed. Loading of the testbed was outside of the scope of this thesis; however, it is important to know how the system might react were it to be loaded. In this chapter, a simulated version of the testbed is created using MATLAB Simulink and the MATLAB Simulink SimHydraulics toolbox.

MATLAB Simulink is a visual programming environment where blocks representing real-world objects such as motors, or mathematical concepts such as a derivative, can be placed within the programming window and connected together. Simulink allows for the creation of a system model through intuitive diagrams instead of through lines of code representing systems of differential equations. For the simulator constructed in this thesis the hydraulic motor, valve, and pump, highlighted in Figure 5.1 by red dashed lines, are identified and recreated. Figure 5.2 shows the motor velocity data used for simulator component identification where the motor velocity shown is caused by a control voltage being held at 3 V until time $t = 23$ s, followed by the voltage being linearly ramped from 3 V to 9 V over 60 seconds.

In Section 5.1, a model of the Black Bruin BB4-800 motor is created based on manufacturer specifications and experimental testing data. Section 5.2 presents a model for the PVG-120 4-way, 3-position hydraulic control valve based on experimental data. In Section 5.3 a pump model is created to provide flow based on pressure difference across the PVG-120 valve model. Section 5.4 implements an MPC controller in simulation and provides comparisons between the simulator results and the experimental results from Chapter 4 to validate the simulator. Section 5.5 then shows the results of actuating the simulator while under load. Finally, Section 5.6 concludes the chapter by summarizing the key results presented within.

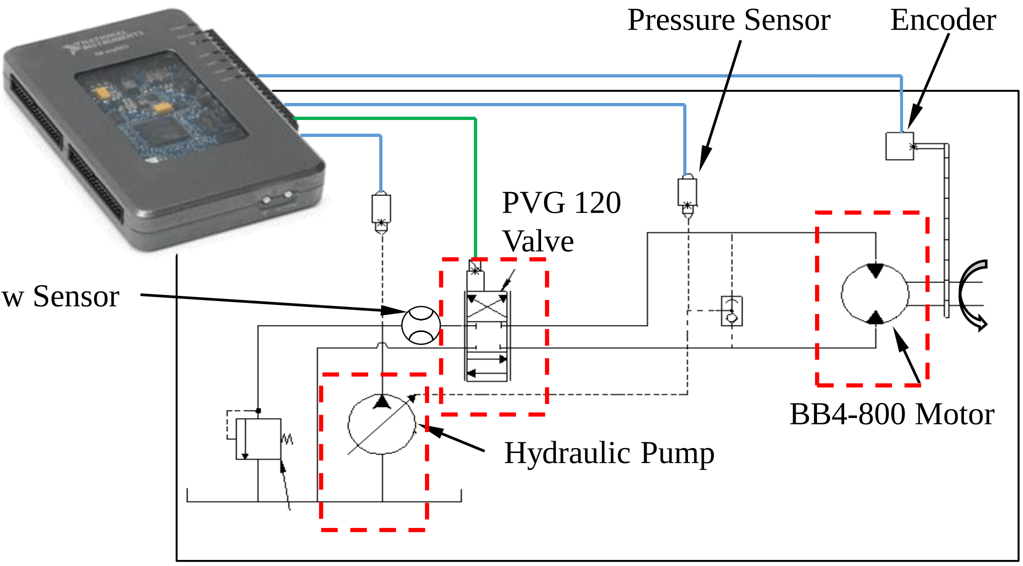


Figure 5.1: In this figure the three components highlighted by red-dashed lines will be recreated withing MATLAB Simulink.

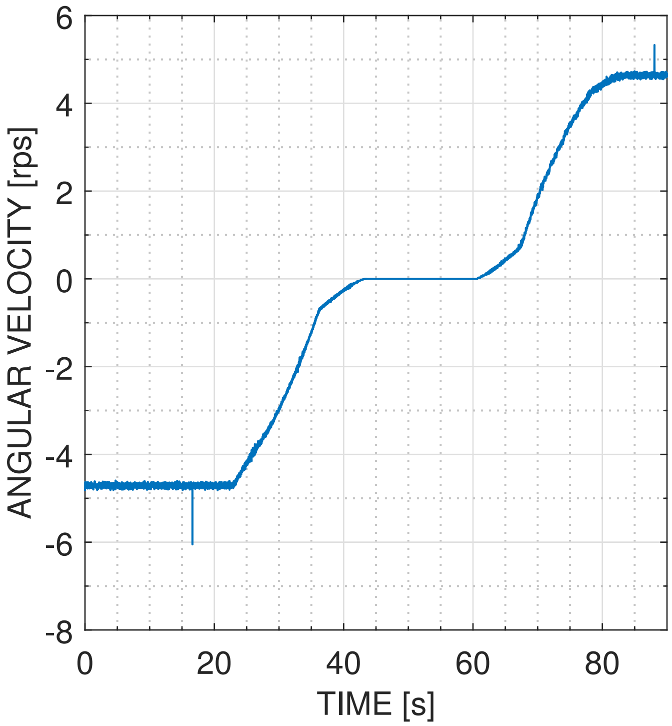


Figure 5.2: The PVG-120 valve voltage was held at 3 V and then ramped linearly from 3 V to 9 V over the time period $t = 23$ s to $t = 83$ s. The motor angular velocity for this time period is shown here.

5.1 Black Bruin BB4-800 Hydraulic Motor Model

When modeling a hydraulic motor the rotational inertia, displacement, friction, and oil leakage rate are the four properties required to create an accurate model. For the BB4-800 motor the inertia, 0.60 kgm^2, the displacement, 0.8 L/rev, and the oil leakage rates shown in Figure 5.3 were all supplied by the manufacturer design guides [42, 43]. The displacement is the amount of fluid needed to rotate the motor through a single revolution, and the oil leakage rate provided in L/min as a function of pressure in bar is the amount of fluid provided to the motor which contributes to lubrication instead of to rotation of the motor. Information about motor friction is not provided by the manufacturer so a method to model friction must be implemented.

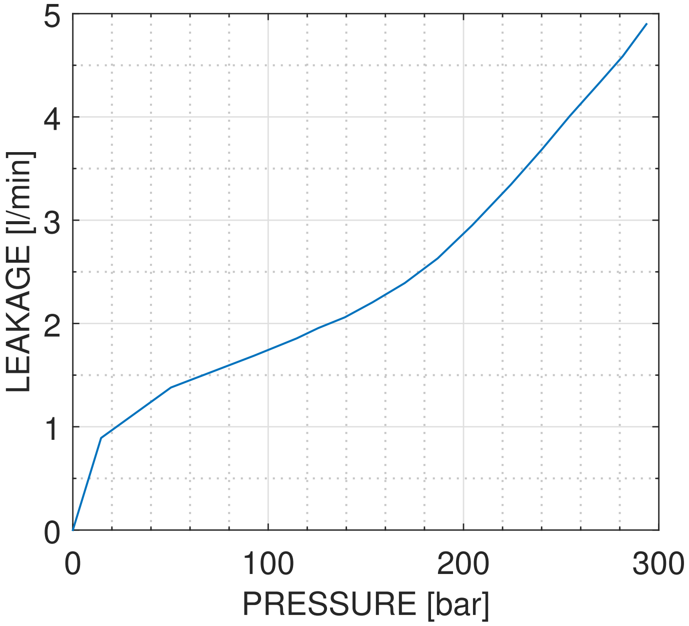


Figure 5.3: The Black Bruin BB4-800 motor leakage in L/min as a function of pressure in bar was provided in the Black Bruin Motor Design Guide [42]

A rotational friction model provided within MATLAB Simulink combining Stribeck, Coulomb, and viscous friction components was used to model friction for the BB4-800 motor. Figure 5.4 shows this friction model where the vertical axis is friction torque and the horizontal axis is the motor angular velocity. The Stribeck friction is the friction needed to first start moving from rest, shown in Figure 5.4 by the Break-away Friction torque and by V_L which gives the angular velocity of peak break-away torque. The Coulomb Friction is a constant offset which provides a friction torque for all motor speeds, and the Viscous Friction is the linear angular velocity dependent part of the friction which begins to take effect at the transition velocity v_T.

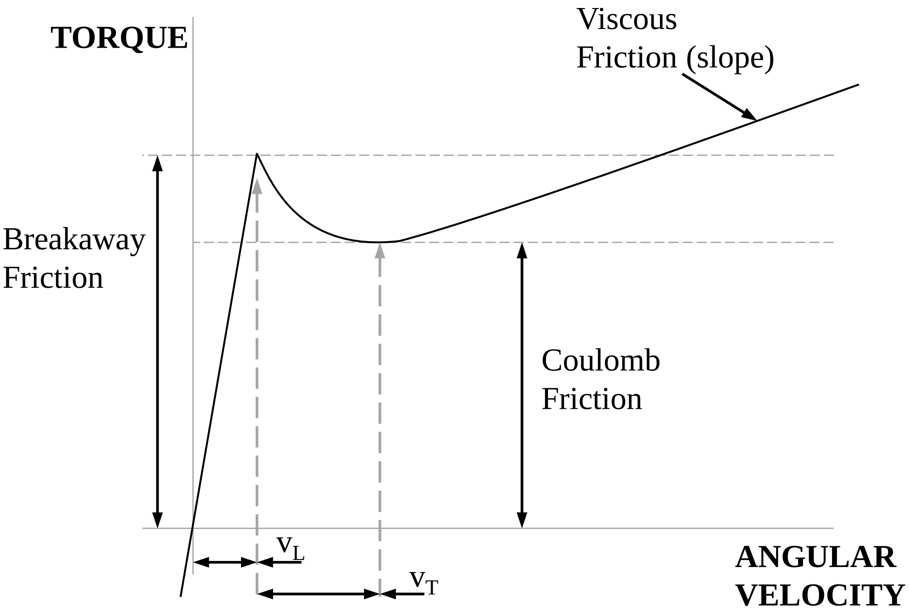


Figure 5.4: The motor friction model used by MATLAB Simulink is shown here.

Using experimental data, the overall friction torque on the unloaded BB4-800 motor can be calculated from pressure sensor data as:

$$T = \frac{1}{2\pi} p_{\text{Load}} D \tag{5.1}$$

where T is the friction torque in Newton-meters (Nm), p_{Load} is the pressure sensor measurement on the load in Pascals (Pa), and D is the motor displacement in meters cubed (m^3).

To identify the five properties: Breakaway Friction, Coulomb Friction, Viscous Friction slope, V_L and V_T, a parameter estimation toolbox within MATLAB was used to determine the friction properties based on a linear ramp signal applied to the AHC testbed. Figure 5.5 shows the simple MATLAB Simulink model of the motor torque used to identify the five torque properties. The block labeled 'A' applied an angular velocity matching the motion data seen in Figure 5.2. This motion data is applied to the motor inertia at 'B', and the Rotational Friction block at 'C' which contains all of the five friction properties described previously. These friction properties at 'C' are modified by the MATLAB response optimization algorithm until the model friction torque, measured at 'D', matches the calculated experimental friction torque based on Equation (5.1) as closely as possible. Figure 5.6 compares the friction torque from identifying the parameters in the Figure 5.5 model compared to the torque calculated from an experimental data set. The Simulink Model parameters identified based on those shown in Figure 5.4 are:

Breakaway Friction	=	245.26 [Nm]
Coulomb Friction	=	54.34 [Nm]
Viscous Friction	=	21.74 [Nm/(rad/s)]
V_L	=	0.6713 [rad/s]
V_T	=	19.25 [rad/s]

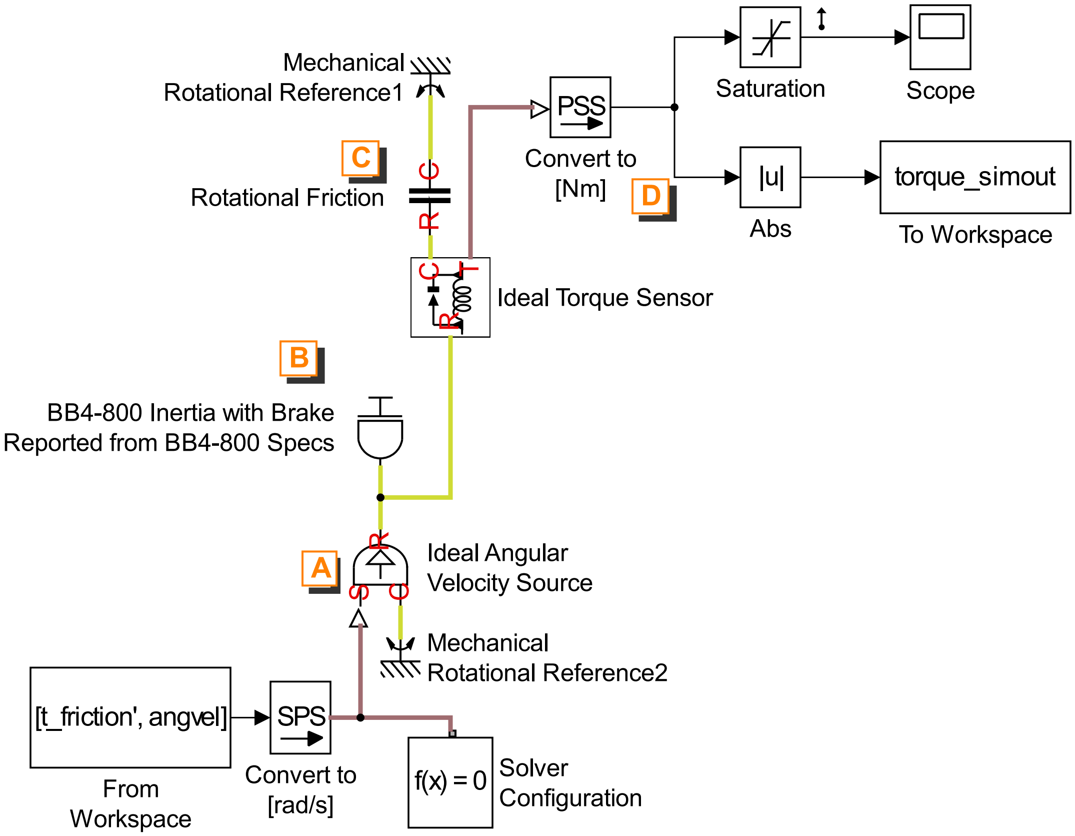


Figure 5.5: This simulation models the AHC testbed motor friction to determine the friction properties based on experimental data.

The actual system torque in Figure 5.6 matches well with the simulation friction torque values below 600 Nm — a torque which corresponds to a motor speed of 4.2 rps. Recall that the actual system torque is calculated from Equation (5.1) where

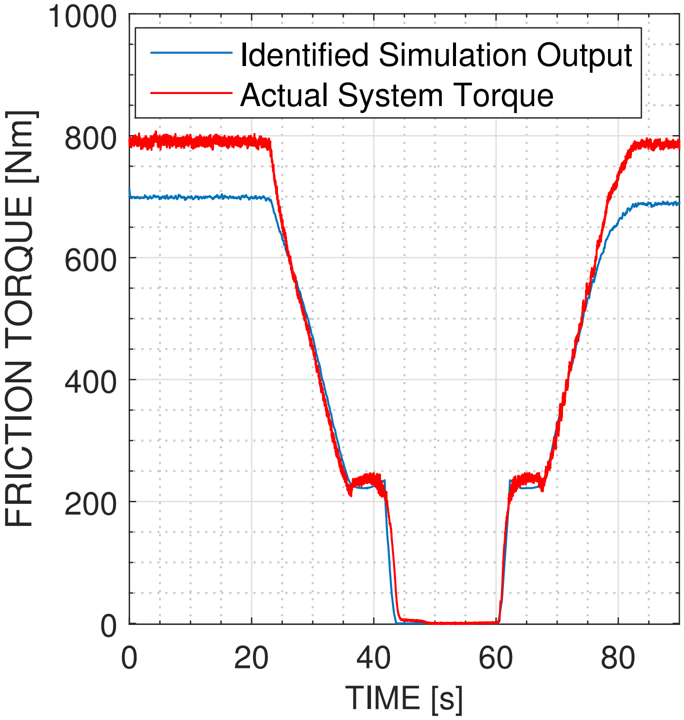


Figure 5.6: A comparison of the identified simulation friction torque compared to the actual friction torque shows good match for torque values below 700 Nm.

p_{Load} is the pressure measured from only one side of the hydraulic motor; however, the pressure p_{Load} should actually be a measurement of the pressure difference across the motor. At low speeds (below 4.2 rps), the pressure at the pump outlet is low and may not contribute to p_{Load}. For speeds above 4.2 rps, the hydraulic fluid returning from the motor may be at a higher pressure due to flow restriction, meaning if this return pressure were accounted for the Actual System Torque in Figure 5.5 would be reduced above 600 Nm. Measurement of the outgoing flow pressure could improve the match between Identified Simulation Output and Actual System Torque.

5.2 PVG-120 Valve Model

A custom 4-way, 3-position PVG-120 proportional valve model was created in Simulink because the PVG-120 valve is not symmetric with respect to flow restriction where symmetric means the pressure drop for flow from ports P to A/B equals the pressure drop for flow from A/B to T. Recall Figure 5.7 from Chapter 3 shows a proportional valve with the spool positioned to allow flow from port P to A and from port B to T. The PVG-120 valve has very little pressure drop for flow from ports B to T compared to P to A. This larger pressure drop for input flow is known as metering-in where metering-in allows a pump to create a constant pressure drop across the valve ports P and A by monitoring pressure only at port A. The return flow to port B is not limited and, therefore, very little pressure drop from port B to T is generated. Note that if the spool in Figure 5.7 were positioned to allow flow from P to B, the P to B flow would be metered while the A to T flow would be unrestricted. The metered opening will be identified from this point on as 'from P', and the unmetered opening as 'to T'. The Simulink model used for the valve is shown in Figure 5.8 where the four orifices, P to A, P to B, A to T, and B to T are each labeled.

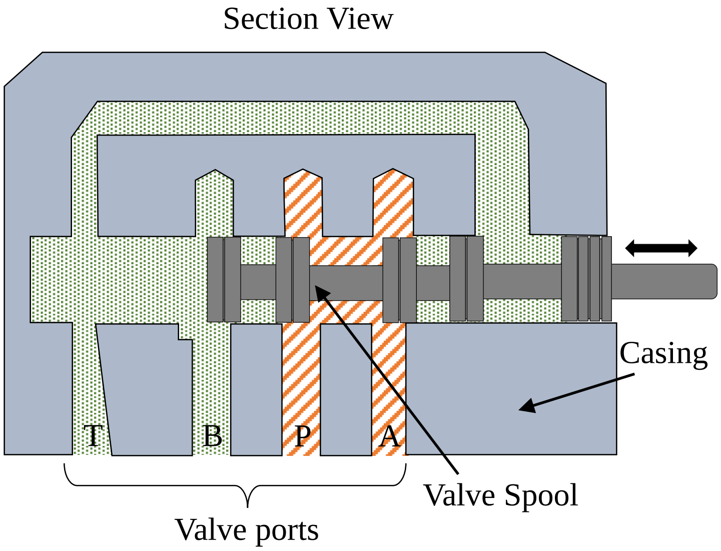


Figure 5.7: Here the valve spool has shifted, allowing fluid flow from port P to A, and from port B to T.

In Figure 5.8 an incoming control signal on S_pos results in orifices P to A and B to T opening, while orifices A to T and P to B remain closed. A signal on the S_neg line results in P to B and A to T opening, while orifices B to T and P to A remain closed. To model the PVG-120 proportional valve orifice openings within Simulink it is necessary to know the area of the valve openings from P and to T as a function of the control voltage.

For the openings to T the manufacturer specification sheet [44] provides the maximum opening as 0.723 cm^2, decreasing linearly to zero as the valve is closed. The manufacturer provided value must be used in this case as a pressure sensor was not installed at the to T port to confirm. To measure the port opening from P recall from

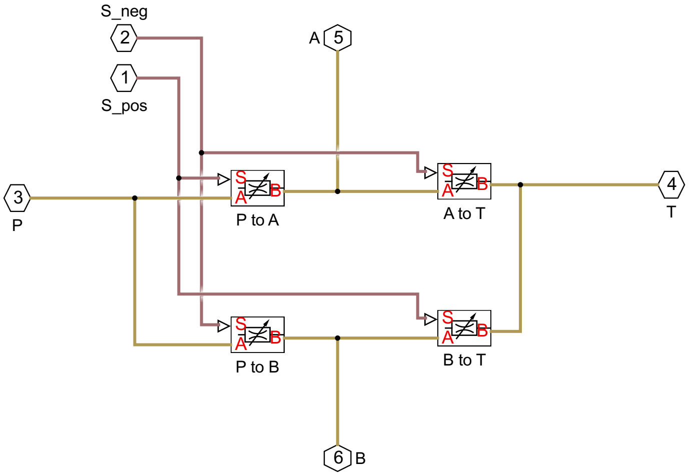


Figure 5.8: The PVG-120 Simulink model consists of four orifices, shown here, directing flow from port P to either A or B, then back to T. The control signals on S_pos and S_neg determine which orifices open.

Chapter 3 equation

$$q = C_d a \sqrt{\frac{2(p_1 - p_2)}{\rho}} \tag{5.2}$$

can be used to describe the flow through a valve opening where

q	=	Flow rate [m^3/s]
a	=	Opening area [m^2]
C_d	=	Discharge coefficient (related to opening geometry) [unitless]
p_1	=	Pressure before valve [Pa]
p_2	=	Pressure after valve [Pa]
ρ	=	Fluid density [kg/m^3]

The total valve flow rate q is calculated from the motor angular velocity data shown in Figure 5.2 by the relationship, $q = D\frac{\omega}{2\pi}$ where ω is the motor angular velocity in rad/s and D is the motor displacement in L/rev, and the pressure sensors shown previously in Figure 5.1 are used to determine the pressure drop from port P. With the flow and pressure drop calculated, $C_d a$ can be determined; however, without knowing the value for C_d the model in Figure 5.9 is used to determine C_d within a custom PVG-120 valve model.

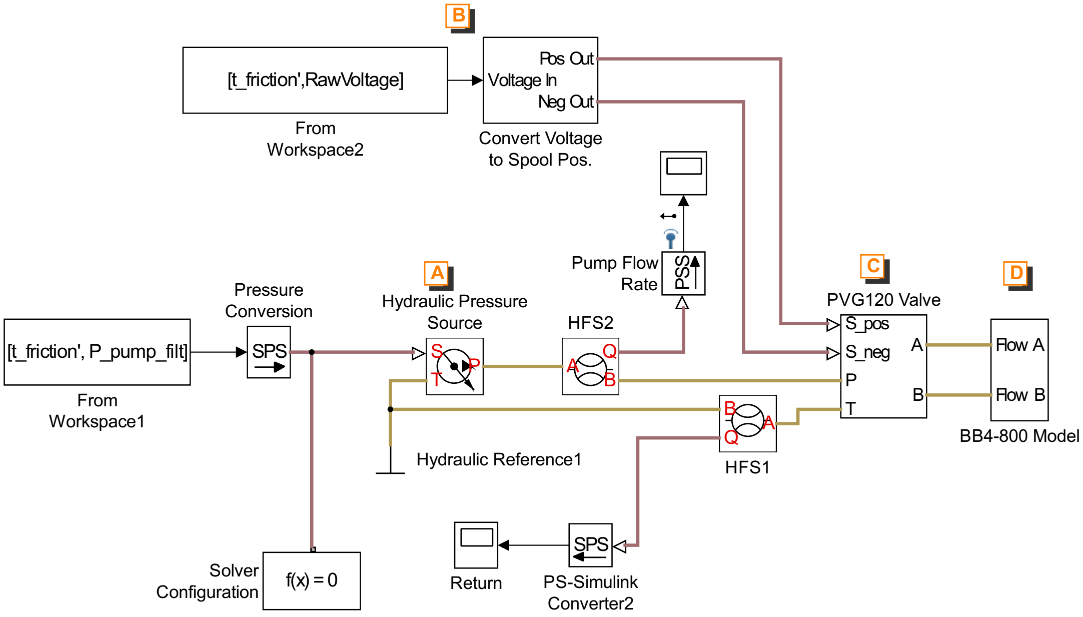


Figure 5.9: The AHC testbed data is loaded by the ‘From Workspace 1/2’ blocks and provides a control voltage and pressure to the PVG-120 valve model, which directions flow to the motor model defined in Section 5.1.

In Figure 5.9, a hydraulic pressure source at ‘A’ matches the experimental pump pressure data. The control voltage to the valve model is generated at ‘B’ and provided to the valve model at ‘C’, where the PVG-120 Valve block models the four valve orifices mentioned at the beginning of this section with $C_d A$ included for the from P orifices. At point ‘D’ the motor BB4-800 motor model identified in Section 5.1 creates a load on the system to match the experimental pressure.

The same response optimization procedure used to determine the motor friction properties in Section 5.1 was used to identify C_d as 0.83, and a comparison between the model motor angular velocity and the AHC testbed angular velocity shown in

Figure 5.10 shows good agreement between the model and the AHC testbed data. Small deviations between the simulation data and the experimental data could be due to variations in C_d, which, since it is related to opening geometry likely varies with valve opening, or due to the inability to measure the true opening area for flow to T.

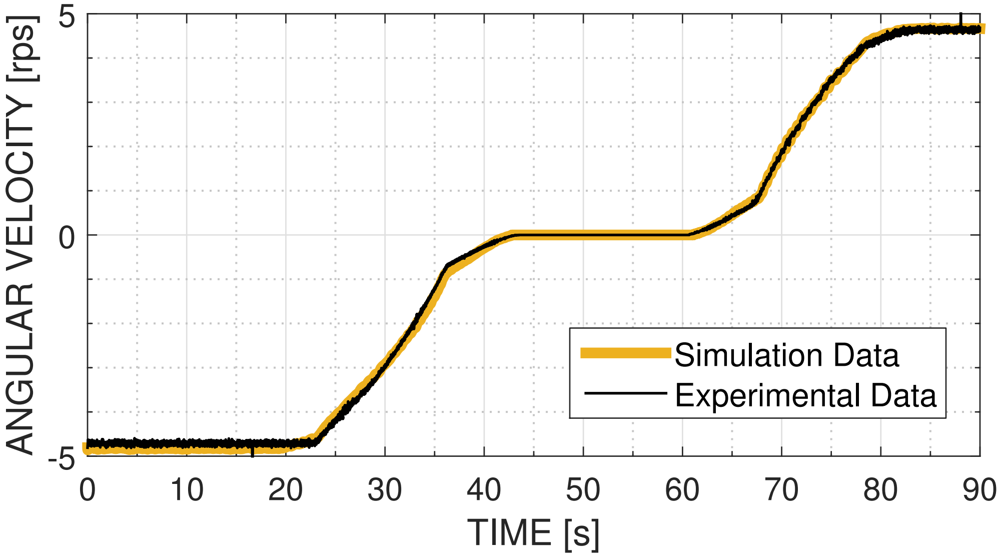


Figure 5.10: The valve model angular velocity is compared to the AHC testbed with the identified valve and motor models.

5.3 Pump Identification

The AHC testbed pump is an Eaton PVM141 load sensing pump. The important thing to note about a load sensing pump is that it operates by providing enough flow to ensure the pump outlet pressure is maintained at a fixed, adjustable amount above the load pressure. For the Eaton PVM141 load-sensing pump, the only specifications supplied by the manufacturer relevant to modeling were the maximum output flow rate of 243 L/min, and the maximum pump pressure of 5000 psi. In the AHC testbed the load-sensing pump maintains a constant pressure drop across the PVG-120 proportional valve. Since little information about the pump internals is provided by the manufacturer, the pump was modeled as an ideal hydraulic flow source utilizing a PID controller to maintain a pressure drop across the simulation valve model — the same pressure drop which was measured on the AHC testbed.

The PID model used within MATLAB Simulink to represent the hydraulic pump was

$$q(s) = \Big(P_p + I\frac{1}{s} + D_p\frac{N}{1+N\frac{1}{s}}\Big)e \tag{5.3}$$

where q is the desired flow, P_p is the proportional constant, I is the integral constant, D is the derivative constant, N is a filter coefficient to ensure the derivative term is not noisy, and e is the error between the desired pressure drop and the actual pressure drop across the valve model. The PID parameters were identified through trial-and-error using the test data previously shown in Figure 5.2. Figure 5.11 shows the final AHC testbed simulator results using the tuned PID controller as a pump, and the AHC testbed data from Figure 5.2.

Following the experimental data in Figure 5.2, note that at times $t = 22$ s, 35 s, and 65 s there are small, but noticeable deviations between the testbed data and the simulator results. These deviations between simulator data and testbed data are expected as the PID controller does not operate identically to the actual AHC testbed pump. Differences in the pump reaction timing, a more damped pump response for the AHC testbed, as well as cumulative errors from the identified motor and valves models all contribute to the deviations seen in Figure 5.11.

Despite the differences between simulator and testbed data, the results in Figure 5.11 show the simulator is capable of operating similarly to the unloaded full-scale AHC testbed. In Section 5.4 MPC is implemented within the simulator and results are presented for a comparison between the simulator and the AHC testbed data for sinusoidal reference tracking as well as the test cases from Chapter 4.

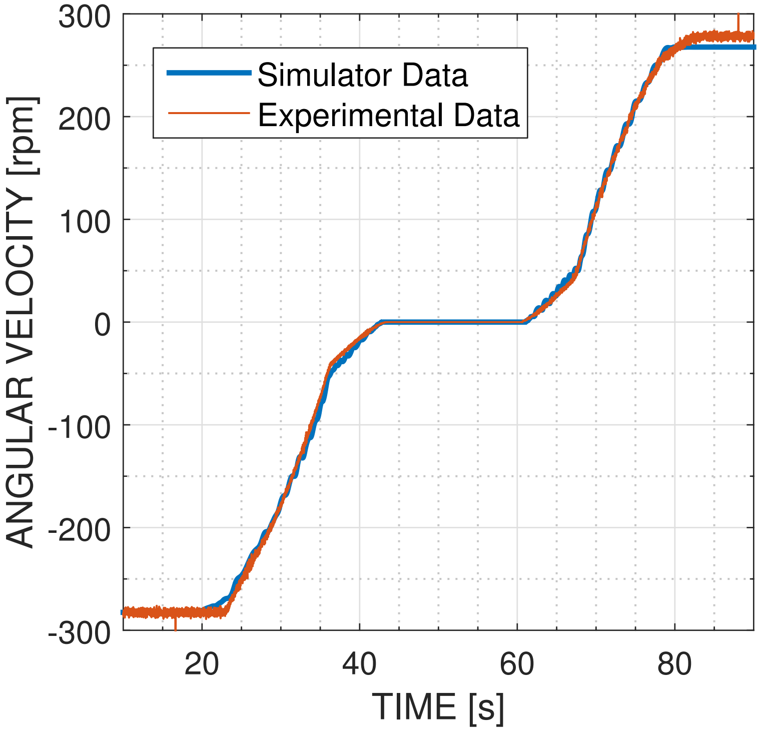


Figure 5.11: The full simulator angular velocity is compared to the AHC testbed for the Figure 5.2 test data.

5.4 Simulator and Unloaded AHC Testbed Comparison

Within the simulator the PID and MPC controllers as well as the non-linear corrections were implemented as shown in the Figure 5.12 flow diagram. For either one of the controllers, the set-point and current motor angle data are provided to the controller and the controller generates an output control action based on the data received. The non-linear gain correction block was described in Chapter 4 as a correction between the linear TF system model and the non-linear response of the valve itself, ensuring the motor angular velocity matches the angular velocity commanded by the controller. The deadband correction block then adjusts the control signal to ensure any positive control signal results in the valve opening and the motor rotating in one direction, while a negative control signal results in valve opening in the opposite direction — effectively removing the valve deadband. Both, the PID and MPC controllers were implemented in the simulator with the same tuning parameters used for experiments in Chapter 4.

It should be mentioned that the simulator deadband width differs from the experimental deadband width identified in Chapter 3. In Chapter 3 the deadband was

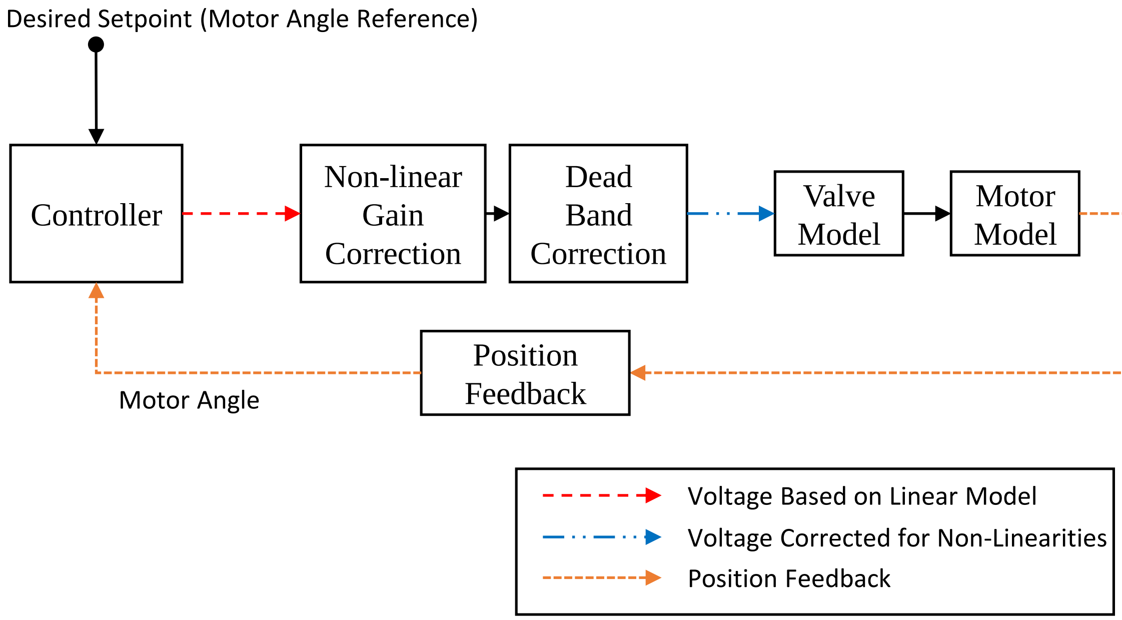

Desired Setpoint (Motor Angle Reference)
Controller
Non-linear Gain Correction
Dead Band Correction
Valve Model
Motor Model
Position Feedback
Motor Angle
Voltage Based on Linear Model
Voltage Corrected for Non-Linearities
Position Feedback

Figure 5.12: A flow diagram of the simulator control logic shows the controller receiving a reference and motor angle signal, followed by the controller output being corrected for non-linearities in the system. This logic is identical to that followed when performing experiments on the AHC testbed.

determined to range from 5.370 V to 7.125 V, while for the simulator data the deadband was determined to range from 5.286 V to 7.145 V. This difference between deadbands was caused by an unreliable power supply used to power the valve when collecting the initial system identification data. The power supply voltage would drift over time and, as the valve deadband was sensitive to small changes in supply voltage, this drift caused the deadband to differ for the system identification data shown in Figure 5.2 compared to the band identified in Chapter 3. It is important to note that, although the deadband correction is different between the simulation and experimental implementation, there is no effect on the dynamics of the system as only the deadband width is modeled in the valve model, not the dynamics of spool motion within the deadband.

5.4.1 Sine Reference Tracking

Figure 5.13 shows a comparison for tracking a 0.1 Hz, 2 rev amplitude sine wave between MPC operating on the unloaded AHC testbed and MPC within the simulator. Looking at the error plot, simulator tracking error follows closely the AHC testbed error, where at times $t = 7$ s, 8 s, and 18 s features in each curve match almost identically. The tracking error standard deviation calculated from time $t = 2$ s to 19 s is calculated as 2.847×10^{-2} revs for the simulator and 2.791×10^{-2} revs for the AHC testbed, showing a difference of 6×10^{-4} revs, suggesting good agreement between the simulator model and the AHC testbed for this frequency of sine wave. Figure 5.14 shows a comparison for tracking a 0.1 Hz, 2 rev amplitude sine wave with PID operating on the unloaded AHC testbed and within the simulator.

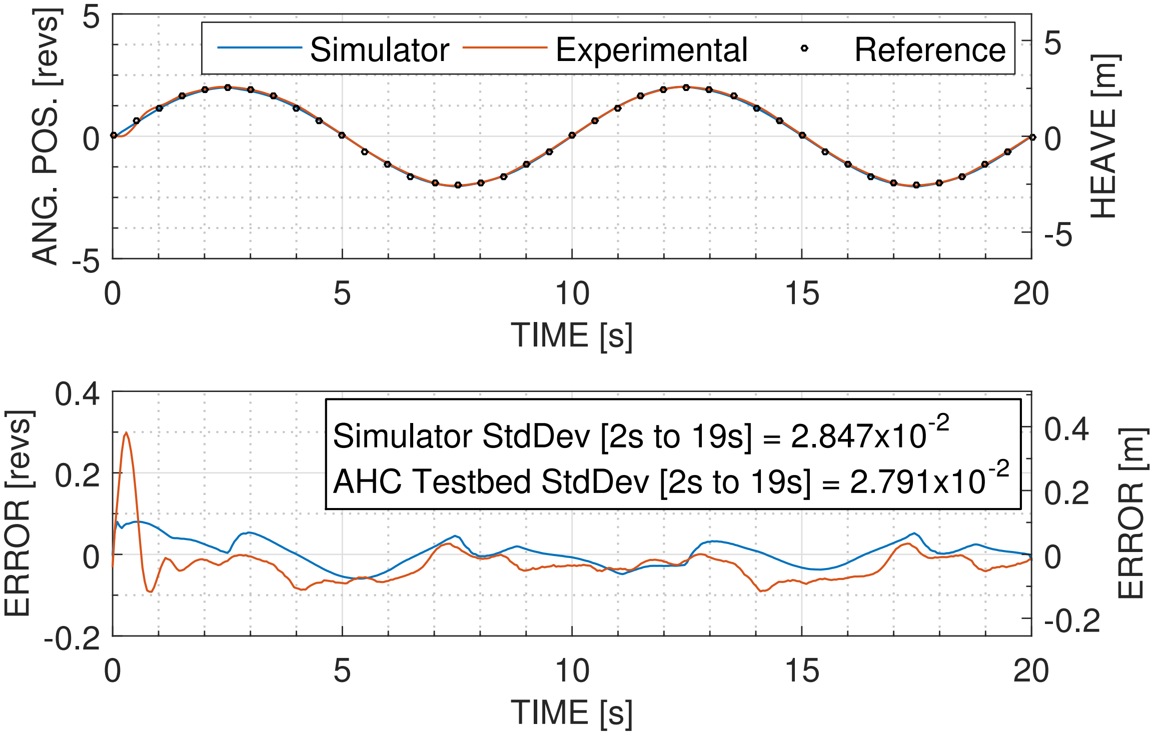


Figure 5.13: The simulator output is shown with the AHC testbed results for MPC tracking a 0.1 Hz, 2 rev sine wave.

The error curves in Figure 5.14 show that the experimental and simulator PID controllers match very closely. The similarity between these PID error curves contrasts the MPC error curves where, although the MPC curves have some matching features, the MPC error curves are clearly different. It is possible that differences in MPC implementation between Simulink and LabVIEW can explain the poor matching between MPC error curves compared to the matching between PID error curves. A custom MPC implementation which could be created in both Simulink and LabVIEW would confirm this. The MPC controller may also be more sensitive to small deviations in the error signal when compared to the PID.

The error standard deviations of 3.011×10^{-2} revs and 2.910×10^{-2} revs for the

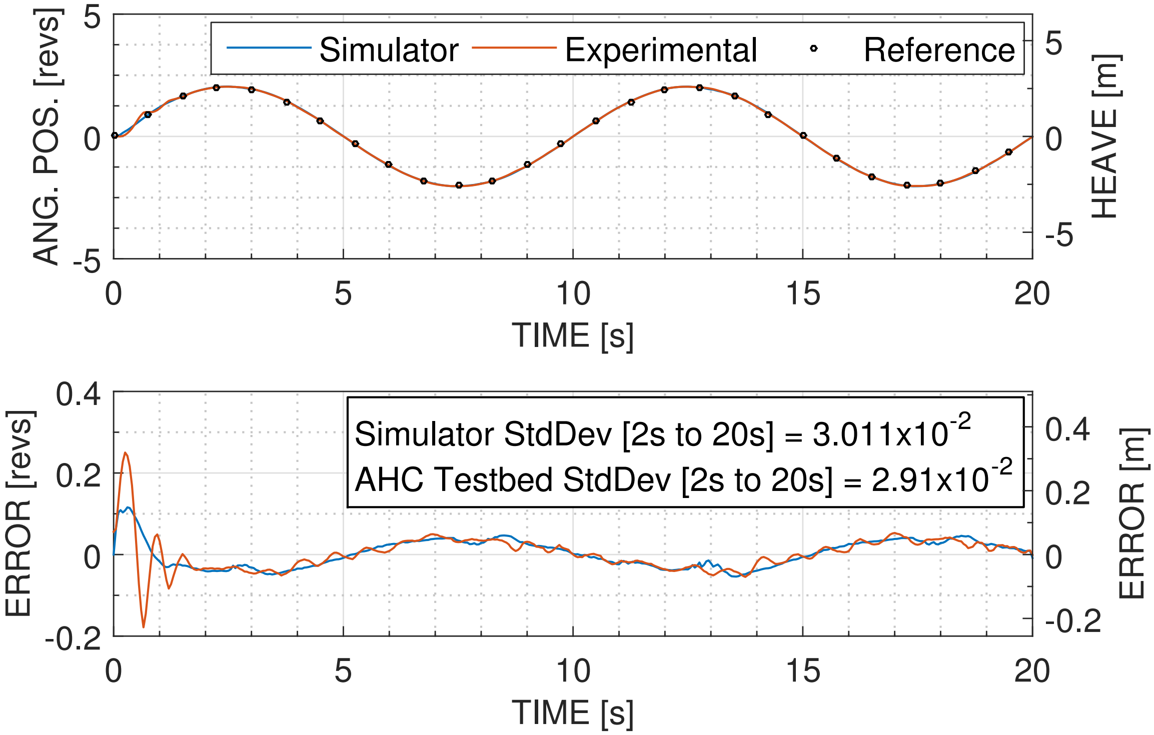


Figure 5.14: The simulator output is shown with the AHC testbed results for PID tracking a 0.1 Hz, 2 rev sine wave.

PID simulator and experimental data are higher than the error standard deviations of 2.847×10^{-2} revs and 2.791×10^{-2} revs for the MPC controller which follows the trend seen in Chapter 4 that PID under performs in error tracking compared to MPC. For MPC tracking of a 0.125 Hz sine wave, Figure 5.15 shows an increase in error standard deviation for the simulator. This increase in error indicates a reduced tracking ability of MPC in the simulator compared to the AHC testbed with increasing sine frequency.

In Figure 5.15 the error standard deviation from $t = 2$ s to 20 s for the simulator is 3.879×10^{-2} revs while for the AHC testbed data the standard deviation is 2.543×10^{-2} revs — an increase of 1.336×10^{-2} revs. The MPC error plots in Figure 5.15 for a 0.125 Hz sine wave no longer follow each other as closely as the error plots shown previously in Figure 5.13 for a 0.1 Hz sine wave, whereas again, Figure 5.16 shows the PID error plots tracking closely with each other for a 0.125 Hz sine wave.

The 0.125 Hz sine tracking error standard deviation for MPC is 3.879×10^{-2} revs for simulation and 2.543×10^{-2} revs for experiment, again, showing lower values compared to the 4.791×10^{-2} revs and 4.472×10^{-2} revs for the PID simulation and experimental error curves. The lower MPC error is important as it shows the MPC

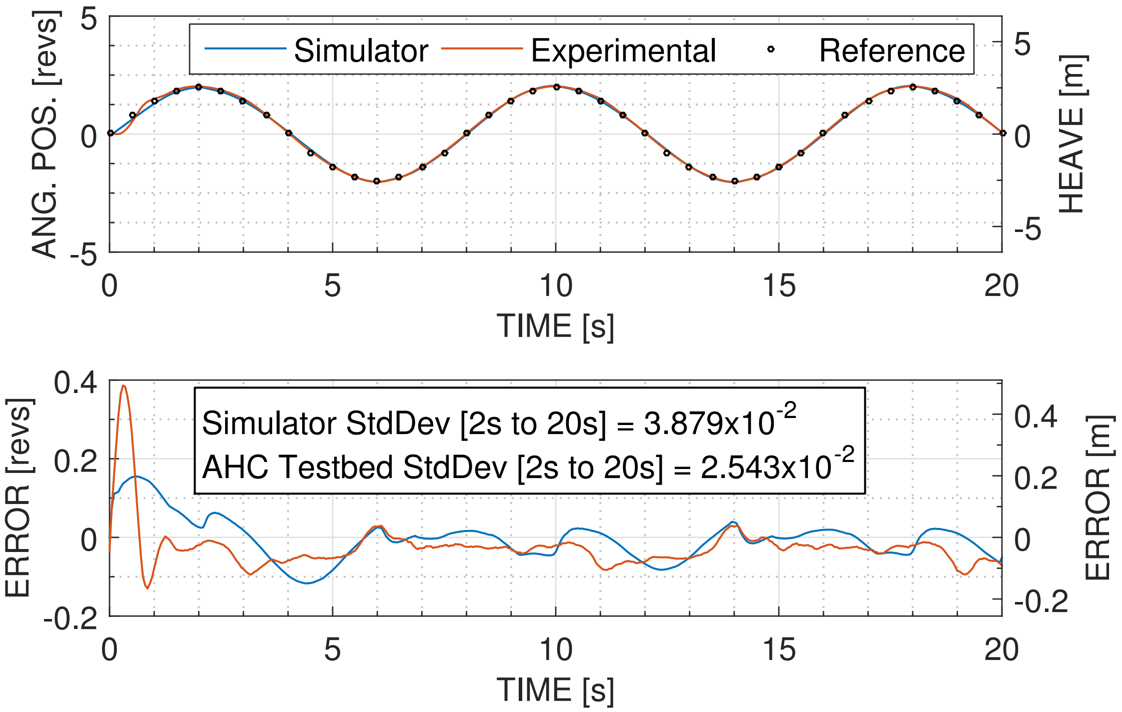


Figure 5.15: The simulator output is shown with the AHC testbed results for MPC tracking a 0.125 Hz, 2 rev sine wave.

controller is still superior in this case under both experimental and simulation conditions compared to the PID controller.

Another increase in the MPC tracking error standard deviation occurs when moving from tracking a 0.125 Hz sine wave to tracking a 0.150 Hz sine wave. In Figure 5.17 the simulator error standard deviation for 0.15 Hz sine tracking is 6.317×10^{-2} revs, an increase of 2.438×10^{-2} revs from the 3.879×10^{-2} revs 0.125 Hz sine wave tracking error standard deviation. The difference between simulator and experimental error curve standard deviations also increases at 0.15 Hz, changing from 2.658×10^{-2} revs for the experimental error to 6.317×10^{-2} for the simulation error — a difference of 3.659×10^{-2} revs. Figure 5.18 shows that for PID the error tracking a 0.15 Hz sine wave rises from 6.386×10^{-2} revs to 7.126×10^{-2} between experimental and simulation curves — a difference of 7.4×10^{-3} revs.

While the behaviour of PID remains consistent between the 0.1 Hz and 0.15 Hz cases, with simulation data reasonably following experimental data, the MPC data curves begin to deviate from each other as frequency increases; again, suggesting the Simulink and LabVIEW MPC implementations differ. At this point, it is important to look at how the simulator tracks the test case data sets used in Chapter 4 as these

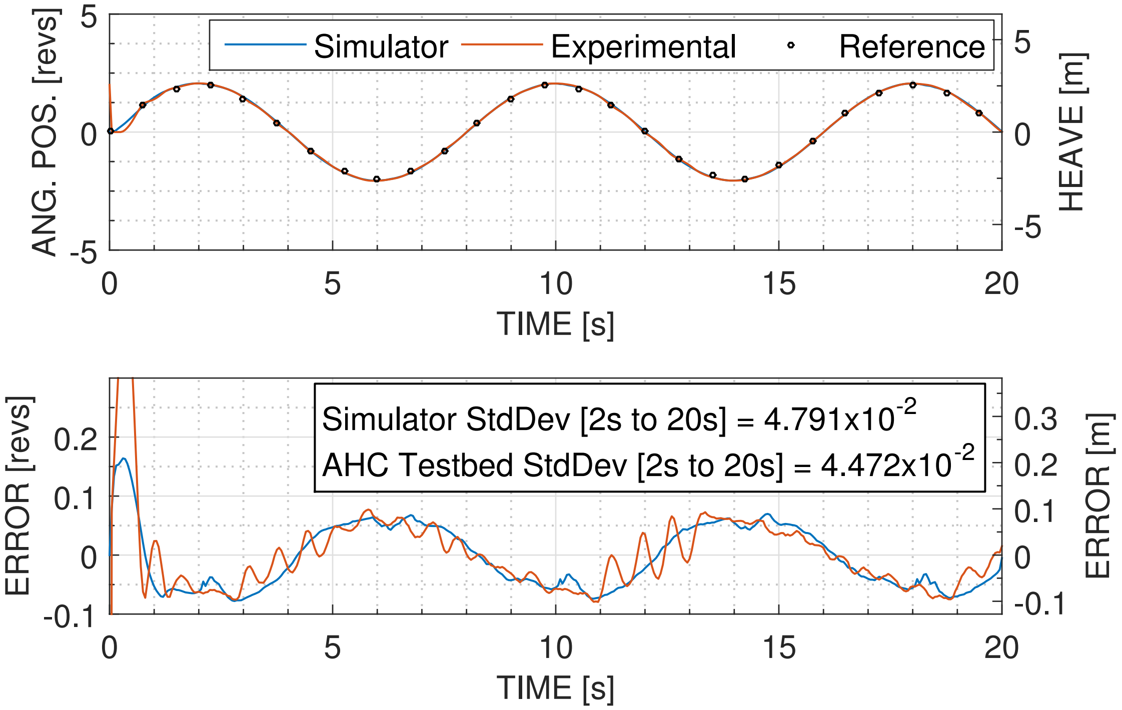


Figure 5.16: The simulator output is shown with the AHC testbed results for PID tracking a 0.125 Hz, 2 rev sine wave.

data sets may provide further insight into the simulator output.

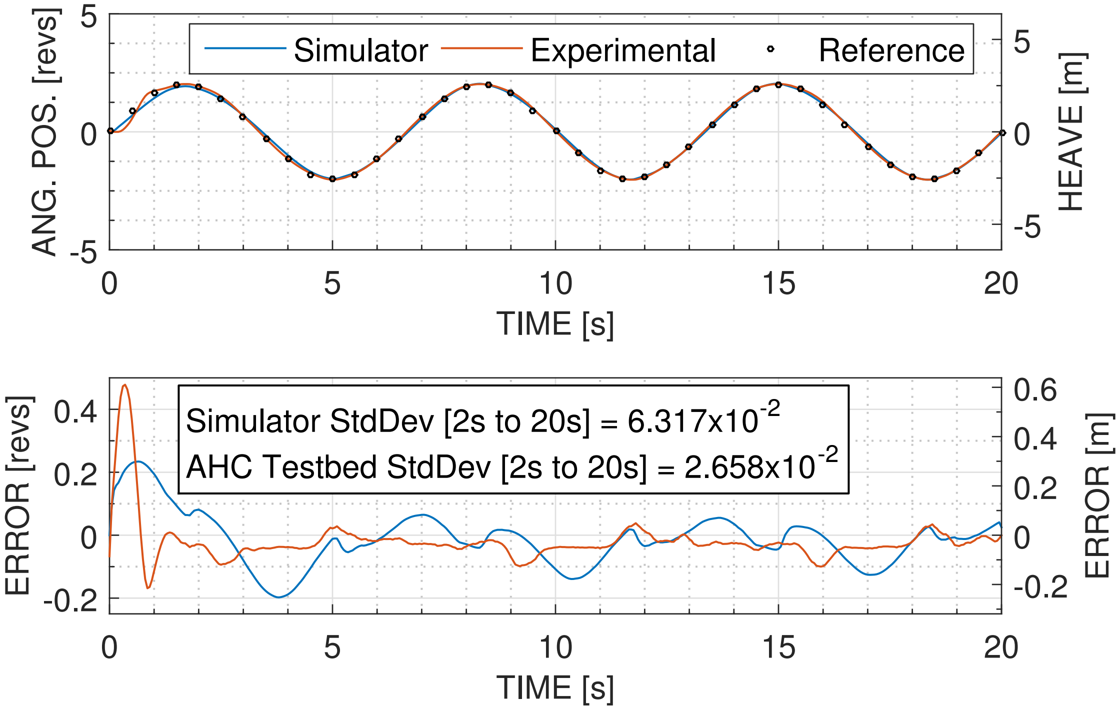


Figure 5.17: The simulator output is shown with the AHC testbed results for MPC tracking a 0.150 Hz, 2 rev sine wave.

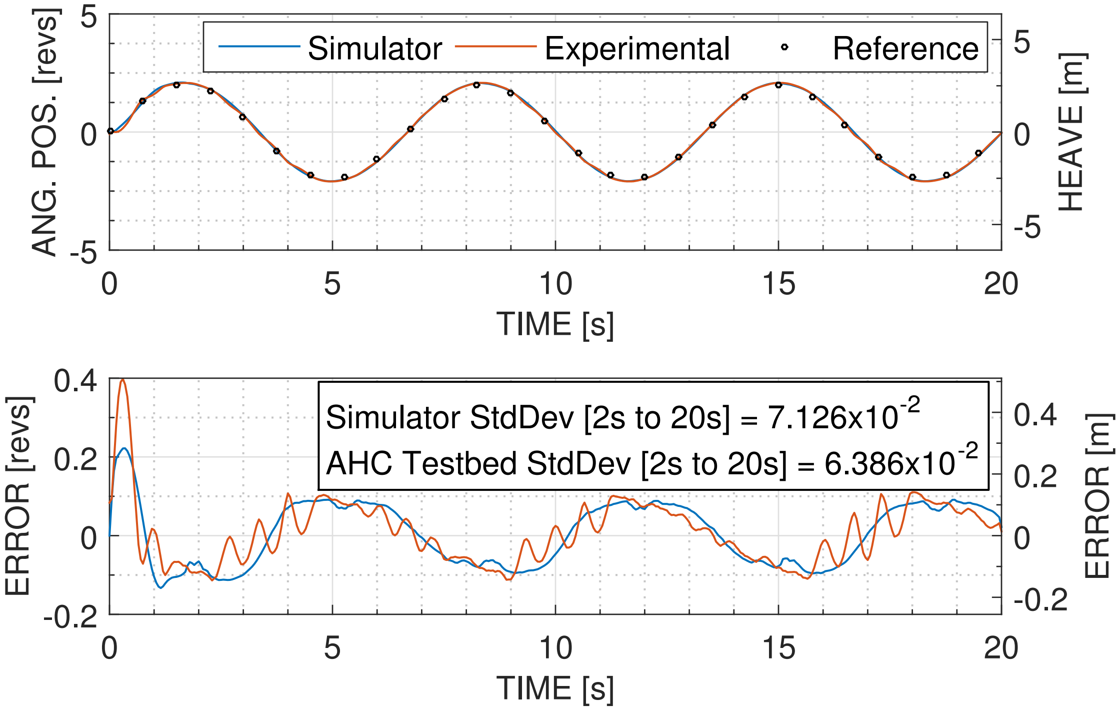


Figure 5.18: The simulator output is shown with the AHC testbed results for PID tracking a 0.150 Hz, 2 rev sine wave.

5.4.2 Benchmark Case Tracking

Figure 5.19 shows MPC tracking the Benchmark case for the AHC testbed and the simulator. For the simulator, an error standard deviation of 2.835×10^{-2} revs is calculated based on data from time $t = 5$ s to 80 s, whereas for the AHC testbed a lower error of 1.306×10^{-2} revs is calculated. As seen for MPC sine tracking in the simulator, MPC tracking of the Benchmark case shows a poorer tracking result where features of the error curve do not closely match. Figure 5.20 shows that, for PID, error between simulator and experimental data again closely track one another. Also notice that, for the simulator error standard deviation, the PID with an error of 2.009×10^{-2} revs tracks better compared to MPC with a 2.835×10^{-2} revs error. In Chapter 4, PID only outperformed MPC for the case where noise was added to the reference signal. This difference in performance further highlights the importance of creating a consistent MPC implementation across both the simulation and experimental platforms. In tracking of Test Case A it is found that, again, the PID error curves closely track each other and MPC within the simulator under-performs when compared to both the experimental MPC as well as the experimental and simulator PIDs.

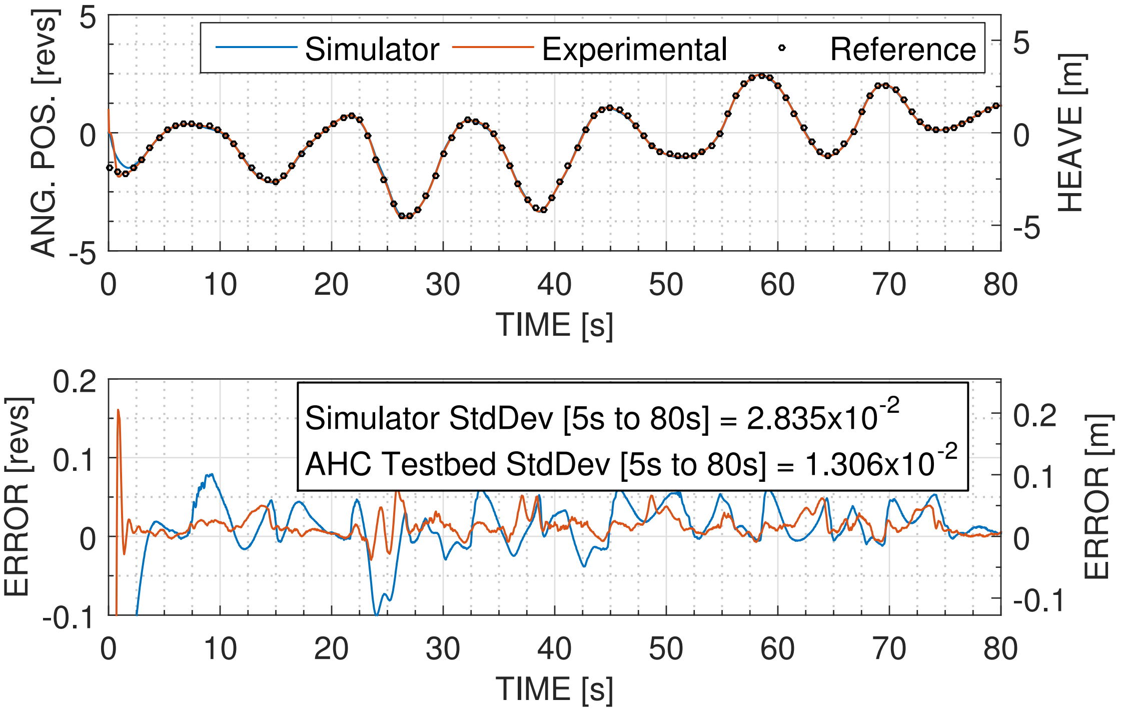


Figure 5.19: A comparison between angular position of the AHC testbed and the simulator for MPC tracking the Benchmark case shows good agreement between the AHC testbed and simulator results; however, a clear increase in error occurs for the simulator.

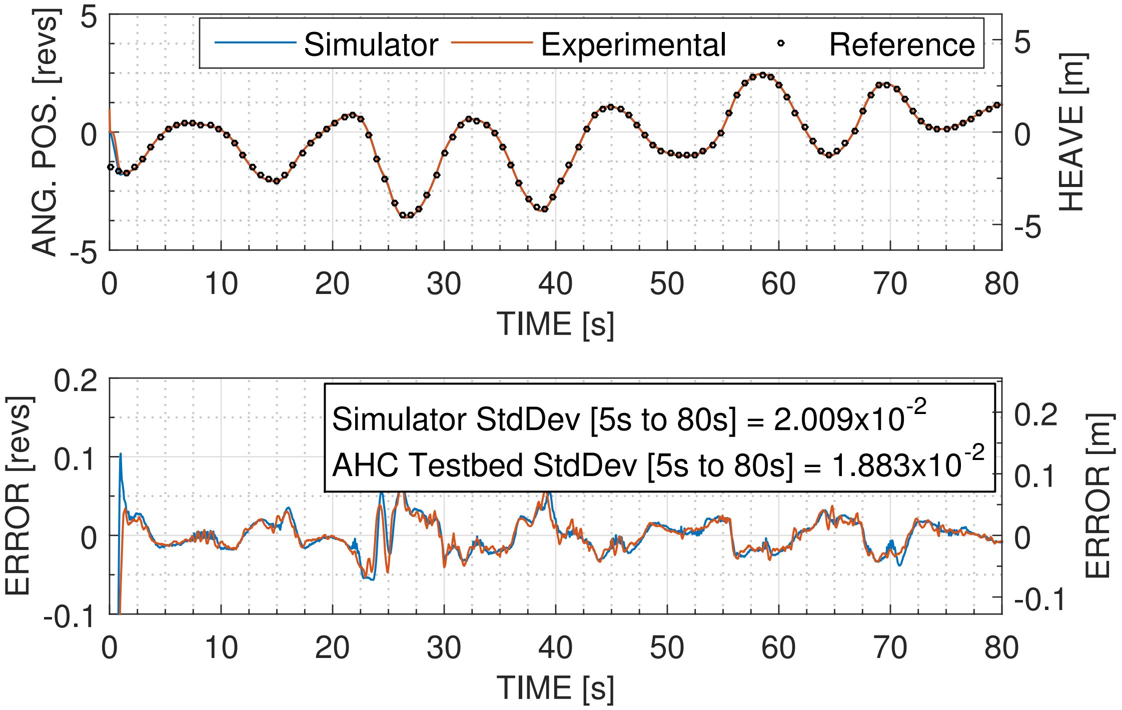


Figure 5.20: The PID tracking curves show good agreement, where the error curves again minimally deviates between simulation and experimental.

5.4.3 Test Case A Tracking

Figure 5.21 shows the Test Case A tracking error standard deviation for MPC between experiment and simulator increasing from 1.310×10^{-2} revs to 2.129×10^{-2} revs. The simulator error curve does not track with the experimental error curve, showing few features in common between the two curves. Figure 5.22 shows the PID error standard deviation between experiment and simulator increasing from 1.649×10^{-2} revs to 1.712×10^{-2} revs which is an increase of only 6.3×10^{-4} revs compared to the increase of 3.439×10^{-2}, and again, the PID error curves track well with each other. Figure 5.23 shows the Test Case B MPC tracking results for the simulator and the AHC testbed where previously unseen peaks in the simulator error curve show additional differences between the AHC testbed and simulator results.

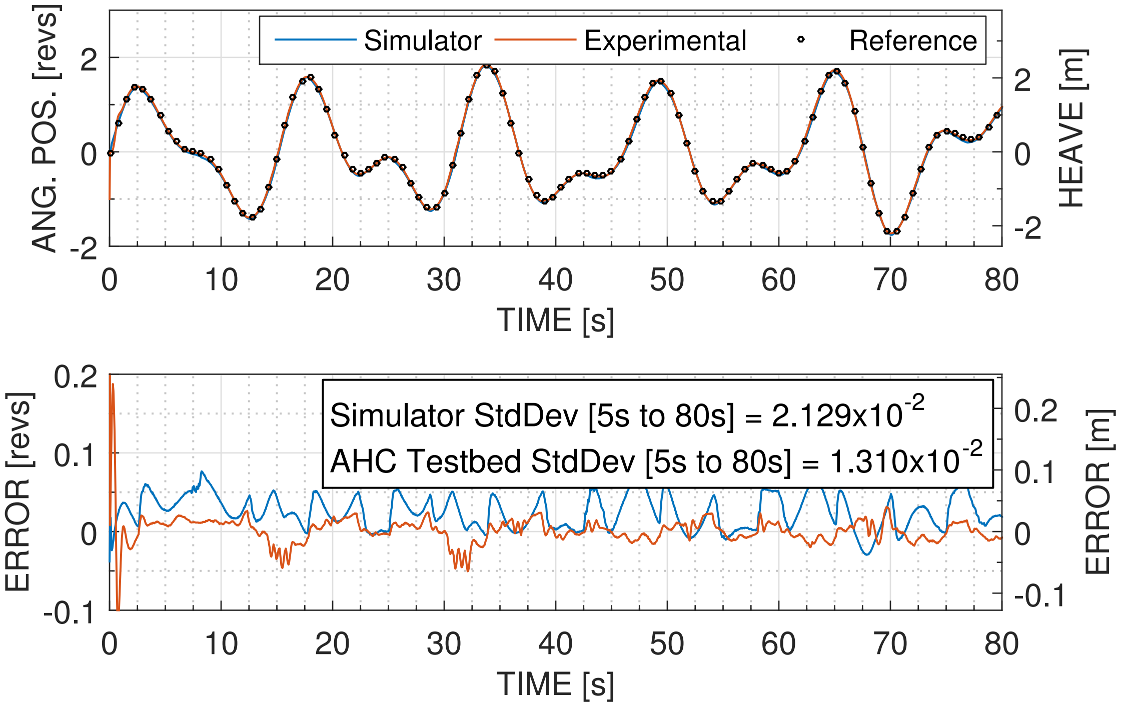


Figure 5.21: A comparison between angular position of the AHC testbed and the simulator for MPC tracking Test Case A shows reasonable agreement

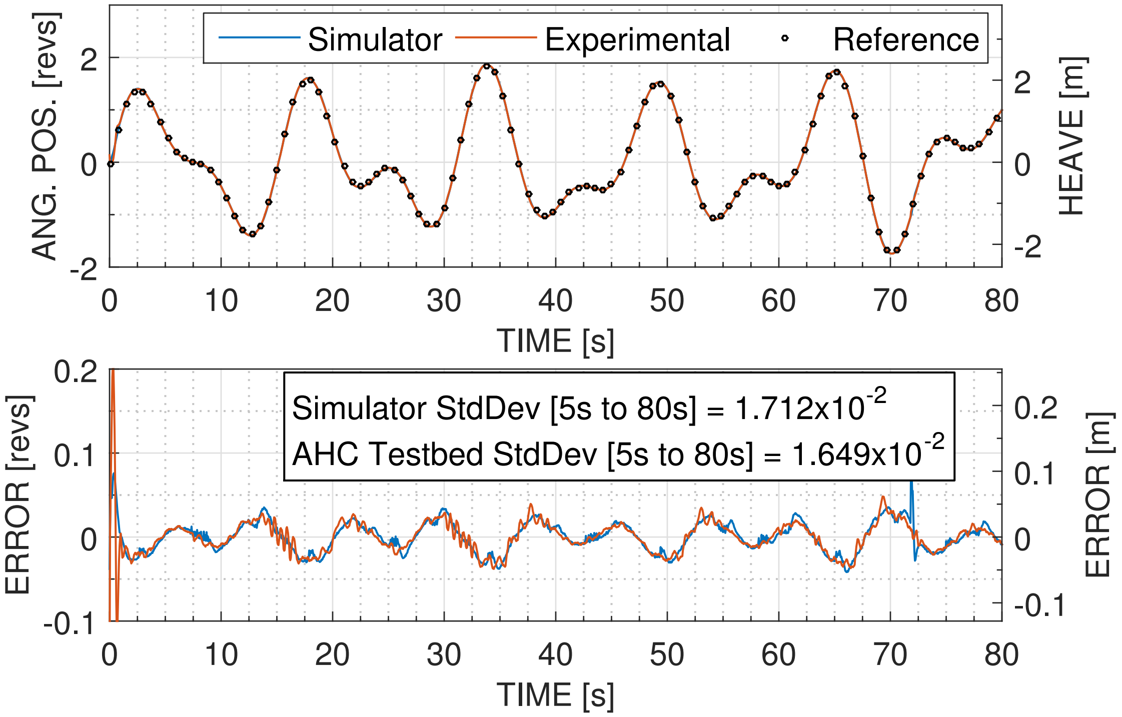


Figure 5.22: A comparison between angular position of the AHC testbed and the simulator for PID tracking Test Case A shows the simulator and experimental responses are almost identical.

5.4.4 Test Case B Tracking

Figure 5.23 plots the results of tracking Test Case B for the AHC testbed and the simulator. Peaks in amplitude above 0.2 revs in the simulator error curve at $t = 32$ s, 37 s, 45s, 52 s, and 67 s inflate the simulator error standard deviation shown in Figure 5.23 as 5.181×10^{-2} revs; however, calculating the simulator standard deviation from $t = 3.5$ s to 30 s, a range which avoids the peaks, gives an error value of 3.230×10^{-2} revs — a reduction of 1.95×10^{-2} revs.

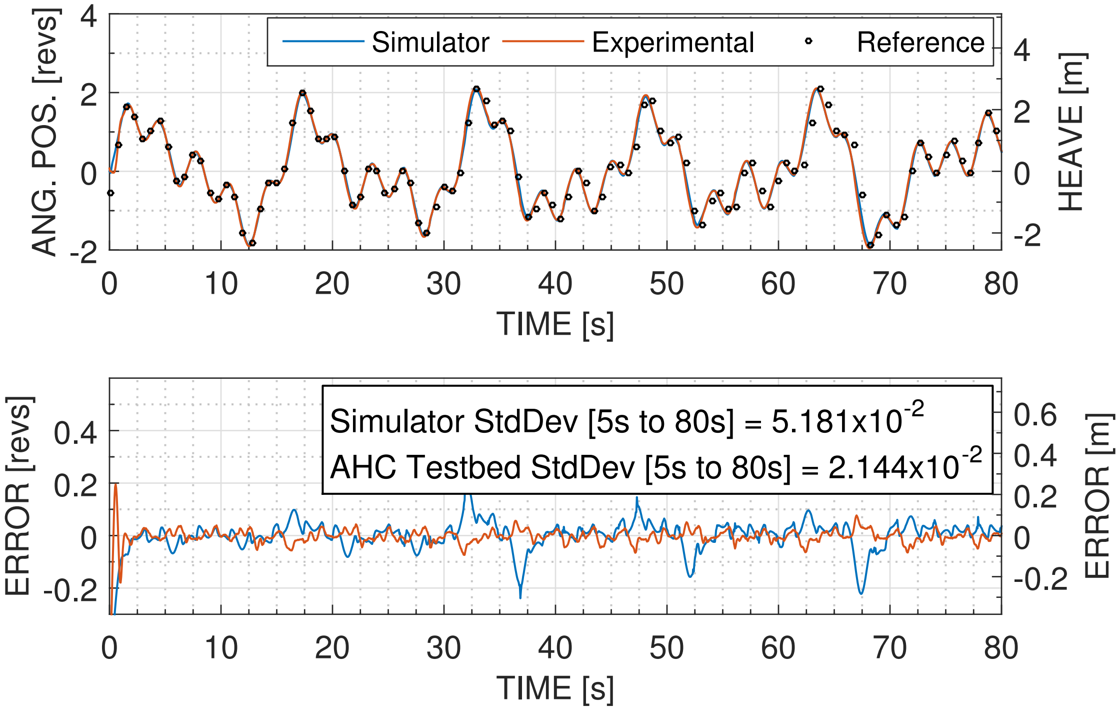


Figure 5.23: A comparison between angular position of the AHC testbed and the simulator for MPC tracking Test Case B shows reasonable agreement, although large peaks in the simulator tracking error inflate the simulator error standard deviation.

At the location of each error peak, examination of the position curves shows that in each case the simulator undershoots the set-point compared to the AHC testbed. This observation could suggest that the MATLAB Simulink MPC acts less aggressively compared to the LabVIEW MPC when using identical tuning parameters, or perhaps the pump model used in the simulator is not tracking pressure changes properly, resulting in reduced pressure drop across the valve model. In Figure 5.24 the ability of PID when tracking Test Case B is comparable between the simulator and experimental curves; however, note that the simulator Test Case B tracking error — ignoring the error peaks — of 3.23×10^{-2} revs is lower than the PID simulator tracking error of 7.134×10^{-2} revs by a factor of 2.2.

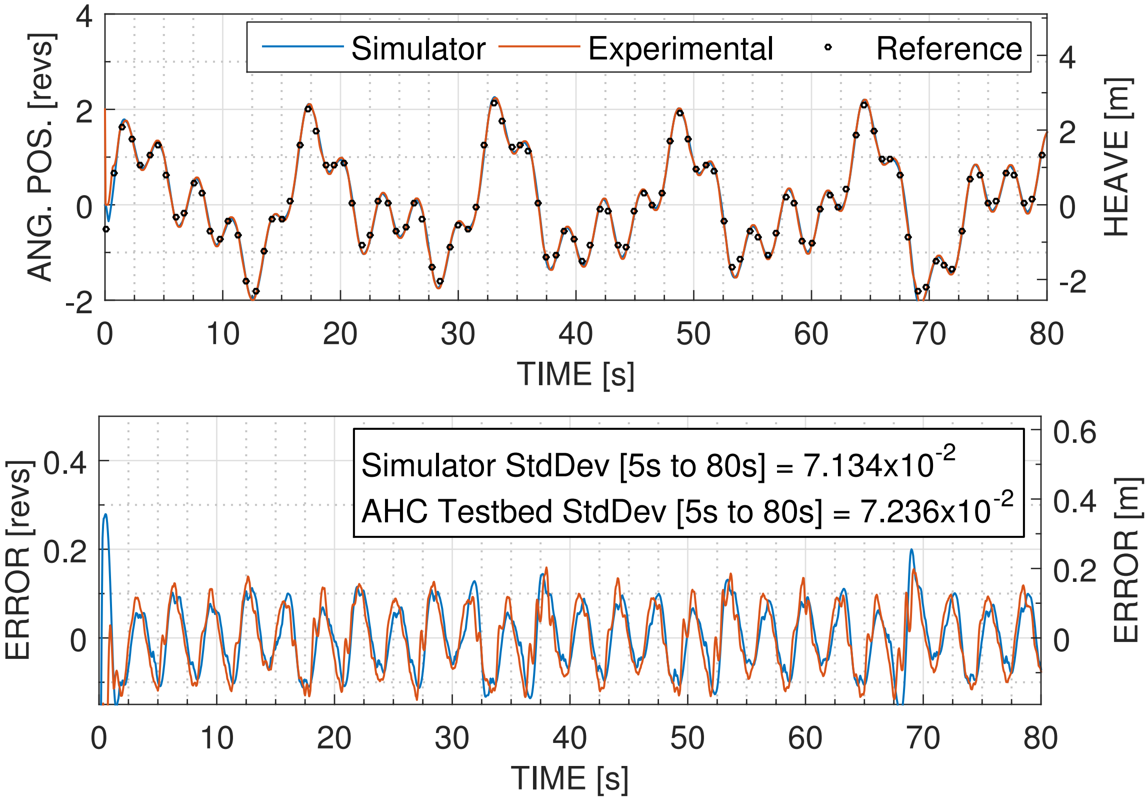


Figure 5.24: A comparison between angular position of the AHC testbed and the simulator for PID tracking Test Case B shows the error curves following each other closely.

The tracking of Test Case B showing better performance for MPC when compared to PID is similar to results seen in Chapter 4, where adding higher frequency components to the reference reduced the effectiveness of PID more than MPC. Figures 5.25 and 5.26 show a similar result as, again, the MPC tracks with reduced error compared to PID for both simulator and experimental data sets.

5.4.5 Test Case C Tracking

In Figure 5.25 peaks in the simulator error plot similar to, but larger than, those seen in the Figure 5.23 error plot are visible. The largest peak at $t = 50$ s reaches a values of -4.789×10^{-1} revs. Inspecting the angular position plot in Figure 5.25, the simulator consistently undershoots the AHC testbed data at most peaks and valleys. This consistent undershoot further suggests that the MPC is acting in a less aggressive manner compared to the AHC testbed system. Furthermore, taking the error standard deviation for time spans where peaks do not occur such as between $t =$ 20 s to 30 s and between $t = 55$ s to 70 s results in simulator error standard deviations of $5.718{\times}10^{-2}$ revs and $3.255{\times}10^{-2}$ revs — both reductions compared to the AHC testbed standard deviation. Figure 5.26 shows a PID error standard deviation of $1.234{\times}10^{-1}$ revs in simulation and $1.191{\times}10^{-1}$ revs for experimental data.

The error plots in Figure 5.26, again, track well with each other. The PID simulator error of $1.234{\times}10^{-1}$ revs is $3.736{\times}10^{-2}$ revs above the MPC error of $8.604{\times}10^{-2}$, however, recall the MPC error is inflated by the peaks in the Figure 5.25 simulator error plot. A comparison to the MPC simulator error value calculated between $t =$ 20 s to 30 s and the PID error value results in a difference between PID and MPC of $6.622{\times}10^{-2}$ revs. This large difference between MPC and PID error standard deviations suggest, as previously mentioned, that PID is less robust to adding new frequency components to the reference. At this point the simulator has been shown to reasonably compare to the AHC testbed under no load, so loading the simulator should lead to results similar to what would be seen loading the full-scale AHC testbed.

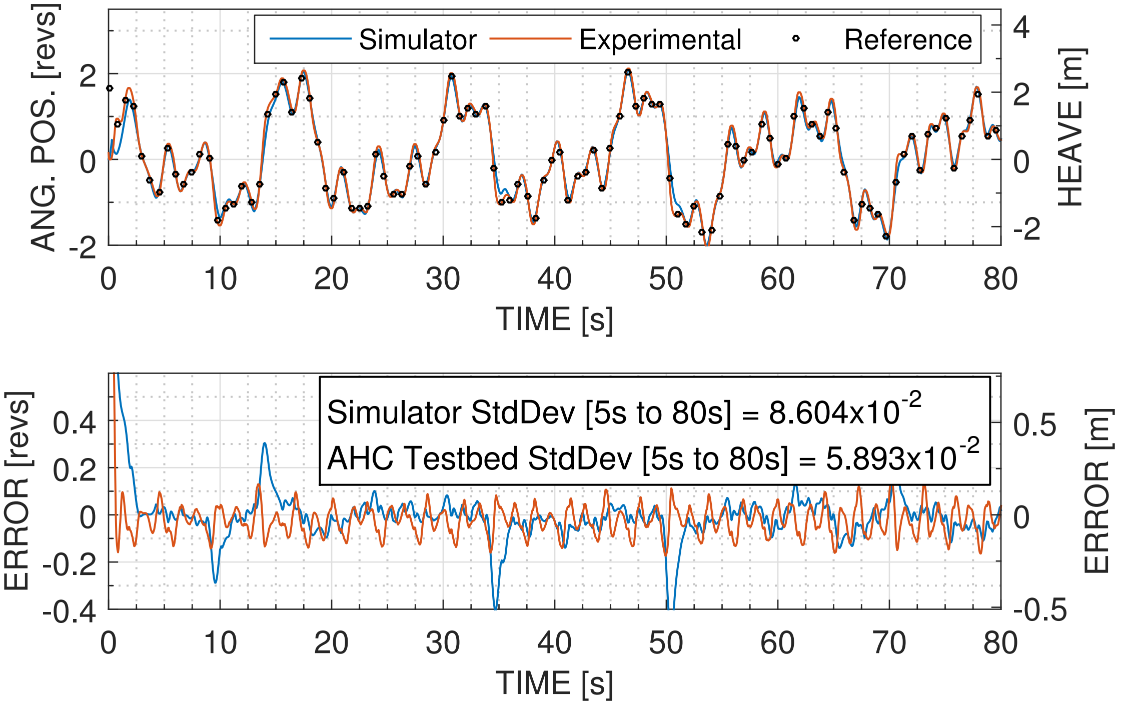


Figure 5.25: A comparison between angular position of the AHC testbed and the simulator for MPC tracking Test Case C shows similar peaks in the error plot to Figure 5.23.

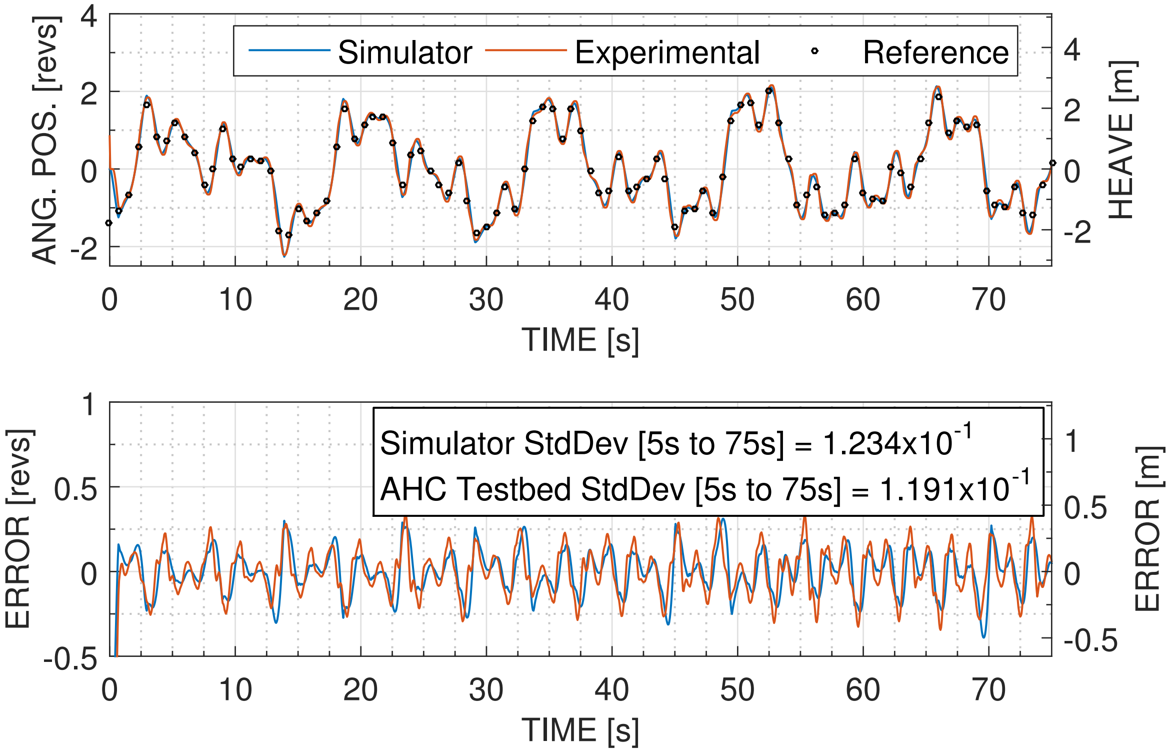


Figure 5.26: PID tracking of Test Case C for the unloaded simulator shows good agreement between simulator and experiment, and errors larger than when tracking with MPC.

5.5 AHC Simulator Operating Under Load Conditions

Loading of the full-scale AHC testbed was outside of the scope of this thesis work; however, within the AHC testbed simulator a load can be applied to the hydraulic motor to study how the system response could be affected.

5.5.1 Sine Tracking and a Counter-Balance Valve

In Chapter 3 Section 3.2 the load requirements for an AHC system as provided by Rolls-Royce Canada Limited (RRC) were given as 2000 lbf (8900 N) pulling on a winch cable attached to a 16 in (0.4046 m) diameter winch drum. The drum was assumed to be made from lightweight composite, and therefore the inertia was negligible. To hold this load a torque of 1800 Nm is needed and this torque is provided by the hydraulic motor. Figure 5.27 shows the result of MPC tracking a 0.1 Hz 2 rev amplitude sine wave while applying an 1800 Nm torque load to the simulator motor and compares the loaded simulator to an unloaded simulator tracking the same reference.

For times $t = 0$ s to 2.5 s, 7.5 s to 12.5 s, and 17.5 s to 20 s the loaded simulator

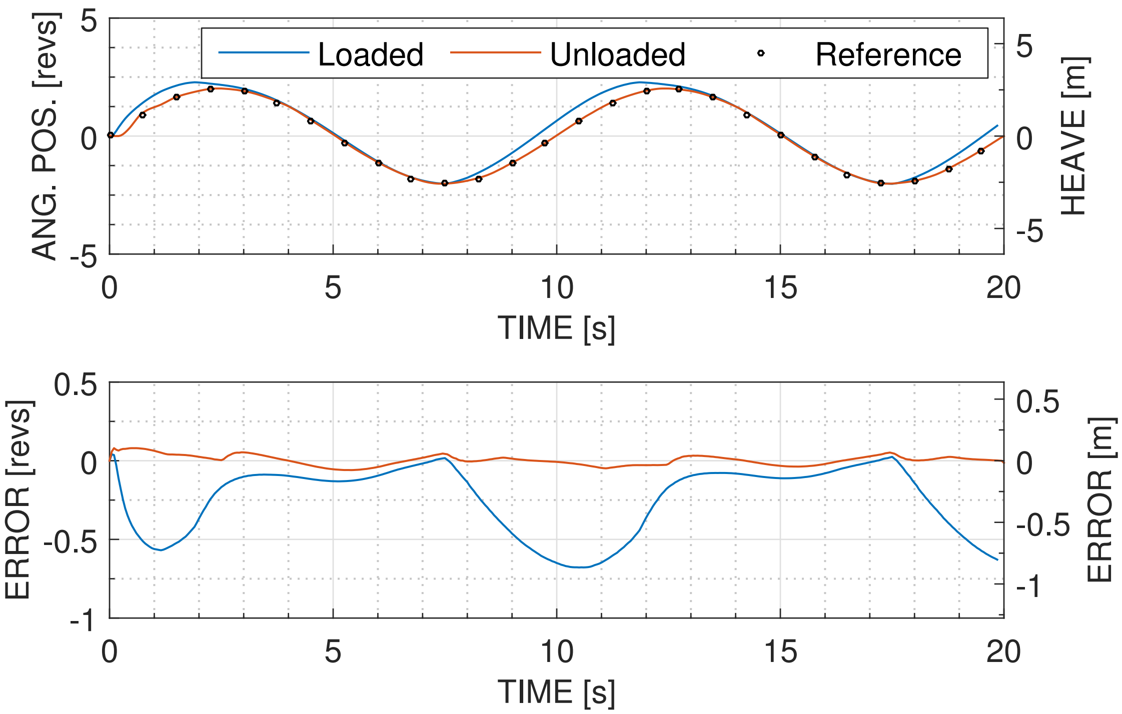


Figure 5.27: The angular position curve here with MPC tracking shows the loaded simulator position clearly leading the unloaded simulator. The leading of the simulator is caused by the load rotating the motor further than desired.

data shows errors below -0.5 revs while the unloaded simulator does not fall below 0.06 revs. These large errors are caused when the motor rotates in the same direction as the 1800 Nm torque is being applied, which causes the motor to act like a pump, driving hydraulic fluid freely through the valve. When rotating in the opposite direction to the applied torque the pump flow opposes the torque and can be properly regulated by the valve. Recall from Section 5.2 that the PVG-120 valve does not restrict flow to the tank, therefore, this explains why the loaded motor can freely drive fluid to the tank when rotating in the direction of the applied torque. The problem of a loaded actuator forcing undesired fluid flow is common in the field of hydraulics and can cause a load to run away and can also cause damage to equipment due to cavitation. The solution, to ensure a load does not drive the pump, is to restrict flow out from the actuator using a counter-balance valve. A schematic of the inside of a counter-balance is shown in Figure 5.28.

A counter-balance valve blocks flow in one direction until a pressure threshold is reached, at which point the valve opens allowing flow. The pressure threshold to open the valve is set above the expected load pressure so that additional pump pressure is required to open the valve. In Figure 5.28 the left schematic shows a closed

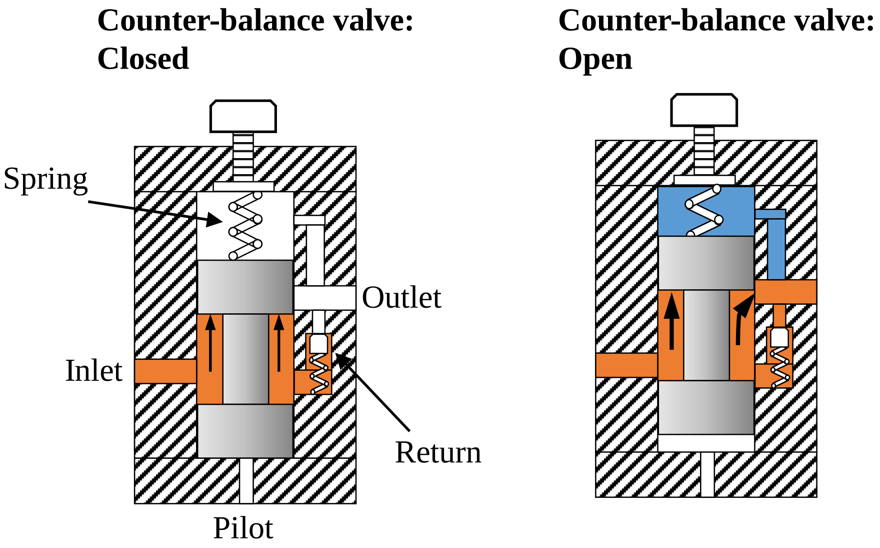


Figure 5.28: A counter-balance valve is shown in the closed, and open positions. The valve opens when inlet pressure becomes large enough to push the spool upward, allowing flow to the outlet.

counter-balance valve where the inlet pressure is insufficient to overcome the spring force acting on the spool. The spring force can usually be adjusted by compressing the spring more or less which, in Figure 5.28, is done by turning a knob on top of the valve. In the rightmost schematic, the inlet pressure has overcome the spring force and flow can now move to the outlet. If the pressure increases the spool will move up further allowing more flow to the outlet. The pilot port allows a hydraulic force to be applied to the spool from a point other than the valve inlet. Also note the return line which allows unrestricted flow from the outlet to the inlet. For the simulator, the properties of a Rexroth VBSN-12A (Model: 04.52.25-03-57-20-00) [45] counter-balance valve were used as this valve allowed 120 L/min flow rates and could hold a load pressure of up to 5000 psi, whereas the load pressure due to the 1800 Nm load was calculated to be only 2000 psi. Figure 5.29 shows a comparison between the loaded and unloaded simulator results with a counter-balance added to the loaded simulator.

Comparing the error curve for a loaded simulator with a counter-balance valve in Figure 5.29 to the error curve for a loaded simulator without a counter-balance valve in Figure 5.27 the large errors between $t = 0$ s to 2.5 s, 7.5 s to 12.5 s, and 17.5 s to

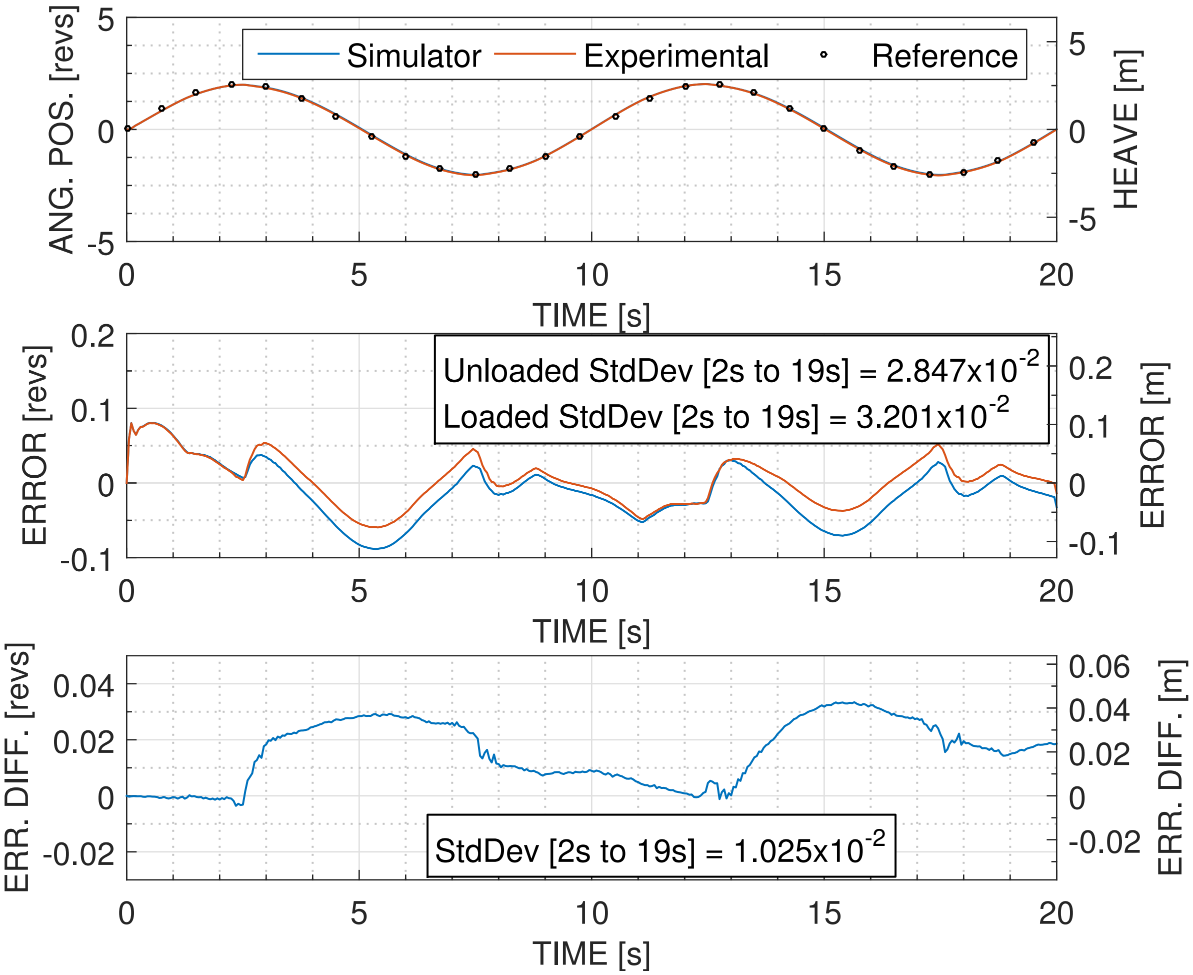


Figure 5.29: The addition of a counter-balance valve reduced the loaded simulator MPC tracking error significantly compared to Figure 5.27. The lower plot shows a clear increase in error during negative motion in the position plot.

20 s in Figure 5.27 are removed by the counter-balance valve. Deviations between the loaded and unloaded curve in the loaded counter-balance simulation occur between t = 2.5 s to 7.5 s and t = 12.5 s to 17.5 s, showing the loaded simulator response lagging behind the unloaded response. The lag is clearly visible in the lower plot where the two lines in the error plot are subtracted to show the difference between the loaded and unloaded curve. This lag is caused by leakage in the motor.

The pressure required to lift the load causes hydraulic fluid to be forced around the working portion of the motor. The fluid forced around the working portion of the motor is lost and does not contribute to the motor rotation, resulting in a lower motor speed and therefore motion lag. The increase of error standard deviation in Figure 5.29 from 2.847×10^{-2} revs for the unloaded case to 3.201×10^{-2} revs for the loaded case is directly caused by the motor leakage. Fluid lost to leakage is negligible on the unloaded simulator as the pressures are much lower. Figure 5.30 shows the PID controller tracking a 0.1 Hz 2 rev sine wave for the loaded and unloaded simulators. The lower plot shows that the deviation seen for MPC due to motor leakage does not occur for PID.

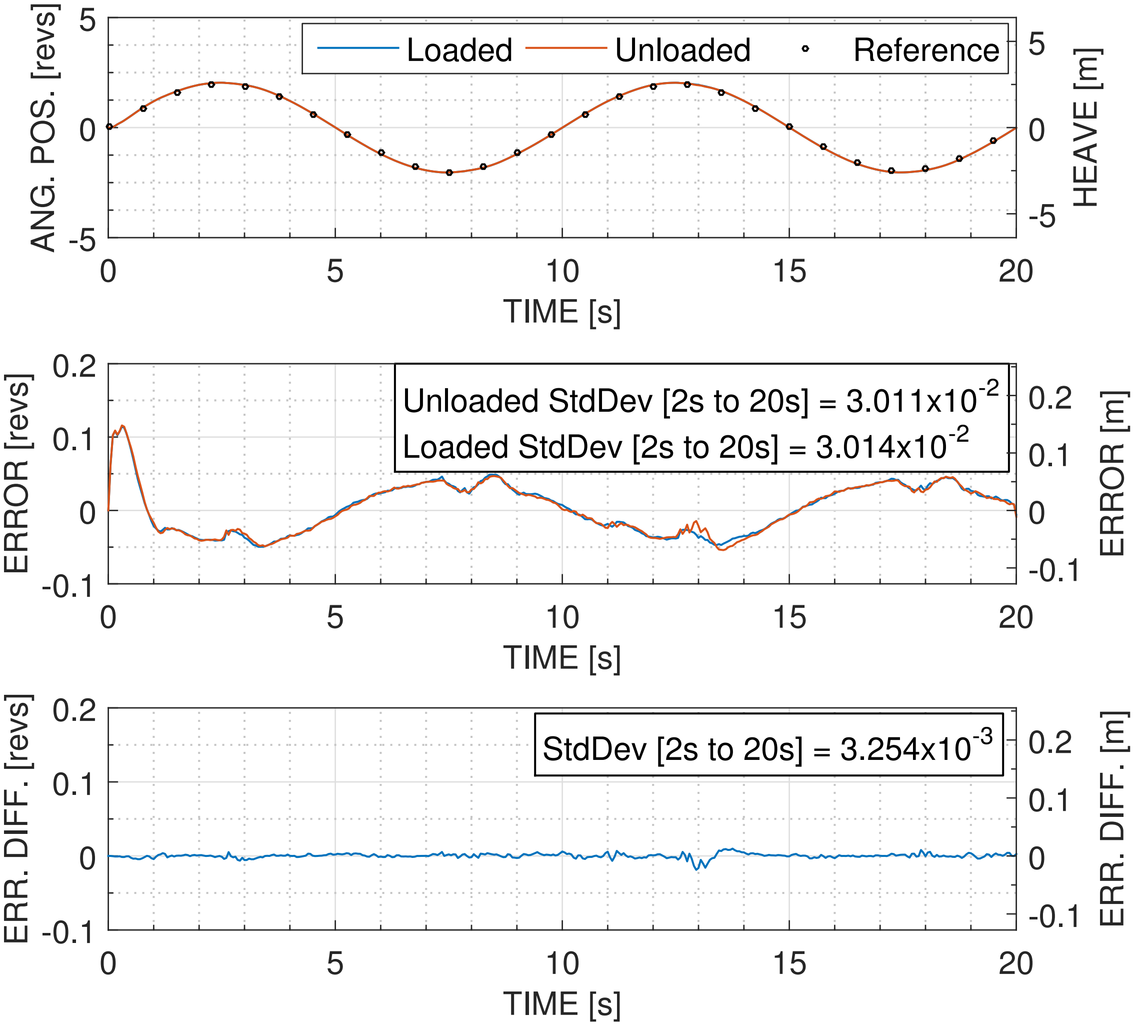


Figure 5.30: The integral term of PID reduces deviations between the loaded and unloaded simulator curves.

The integral term of the PID controller is able to correct for the reduced flow caused by motor leakage, whereas, MPC only applies correction terms based on the model. Since the large load reduces effective flow to the motor, this essentially changes the gain on the TF model used by the MPC as the gain represents amount of flow to the motor. To correct for this apparent change in TF gain, one could add an integral controller to be used in parallel with the MPC whereby the MPC would still correct for upcoming changes in set-point while the integral controller would correct for slowly changing constant offsets.

The error standard deviation difference between the loaded and unloaded simulator for PID tracking of the 0.1 Hz 2 rev sine reference is 3×10^{-4} revs whereas for the MPC, the same error difference is 3.54×10^{-3}, showing a factor of 11.8 difference between MPC and PID. This difference suggests that PID is less affected by loading of the simulator. In the following section, MPC tracking the Benchmark Case data in the loaded simulator shows the same lag behavior seen with MPC in the Figure 5.29 0.1 Hz sine error curve.

5.5.2 Benchmark Case

In the error plot in Figure 5.31, the loaded case error curve is shifted down when compared to the unloaded curve at every point for a decreasing angular position, again showing that the loaded system lags when the motor rotates at a negative angular velocity which corresponds to lifting of a load. As seen for the 0.1 Hz 2 rev sine tracking case, when lowering the load (positive angular position slope) there is better agreement between the unloaded and loaded error curves. Figure 5.32 shows the PID controller error curves track each other closely between the loaded and unloaded cases in the simulator.

As mentioned in Section 5.5.1, the PID loaded and unloaded error curves tracking so closely with one another is likely due to the integral term of the PID controller accounting for the constant offset due to motor leakage. For each of the test cases presented further in this chapter, results are similar; the MPC controller error curves deviate from each other while lifting the load, and the PID controller shows very little deviation between the loaded and unloaded error curves.

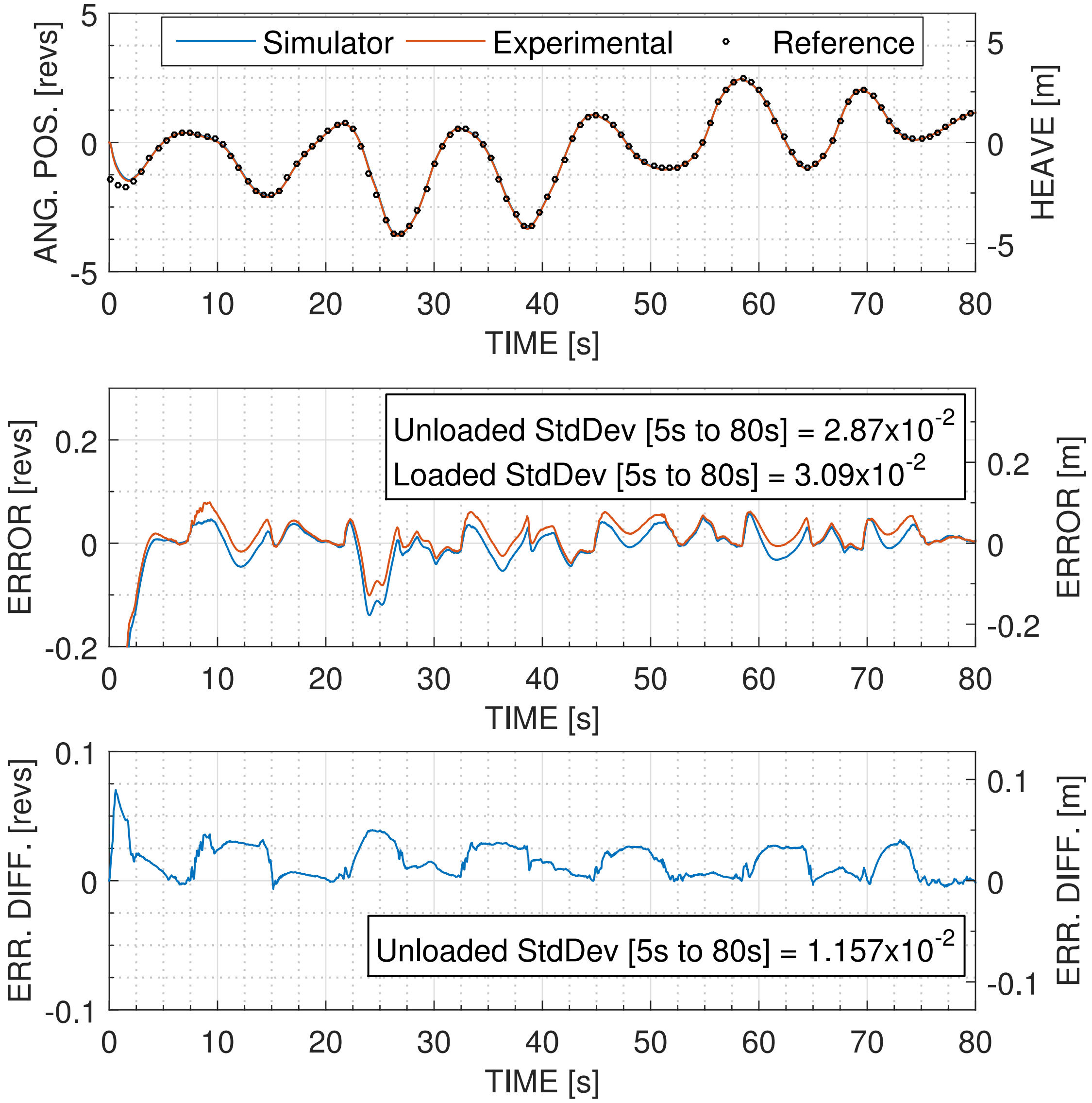


Figure 5.31: The loaded and unloaded simulator data for MPC tracking of the Benchmark case is shown here. The addition of a counter-balance valve reduced the loaded simulator error significantly compared to Figure 5.27.

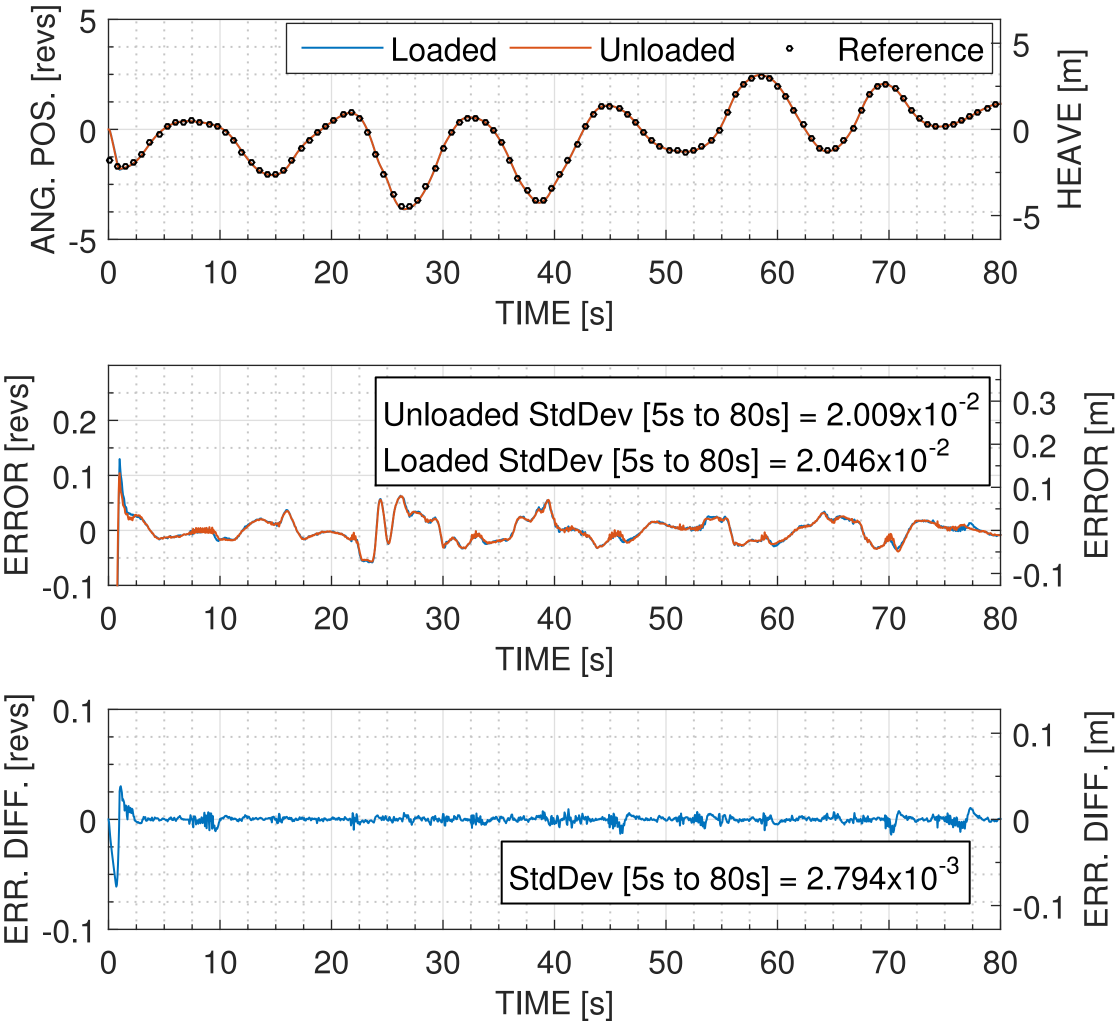


Figure 5.32: The PID controller tracking the Benchmark case, both loaded and unloaded, in the simulator. Very little difference is shown between the error curves.

5.5.3 Test Case A

Figure 5.33 shows that, for the MPC controller tracking Test Case A, the loaded error curve deviates from the unloaded case curve mainly when raising the load.

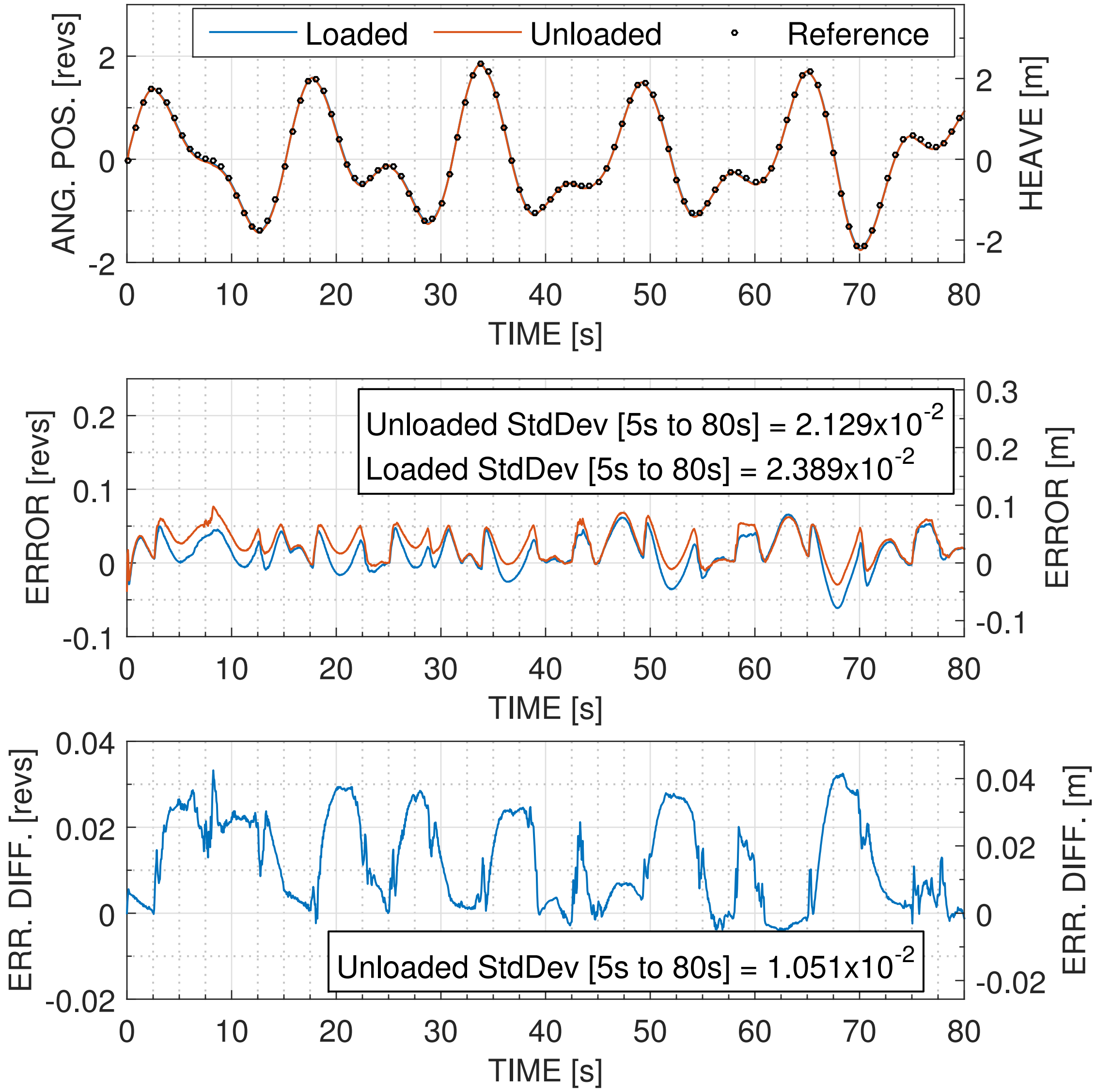


Figure 5.33: The loaded and unloaded simulator results for tracking Test Case A with MPC are shown here. The loaded error curve is shifted down due to motor leakage, seen clearly in the lower plot for times when the angular position is decreasing.

The loaded error curve in Figure 5.33 is shifted down compared to the unloaded case when lifting the load. This shift was seen previously for the Benchmark case. Note that, other than shifting the error curve, adding the counter-balance valve and 1800 Nm load have little effect on the loaded curve shape or features when compared to the unloaded curve. Figure 5.34 shows the same result seen previously when comparing the loaded and unloaded simulator PID cases; the PID error curves track each other with little noticeable error.

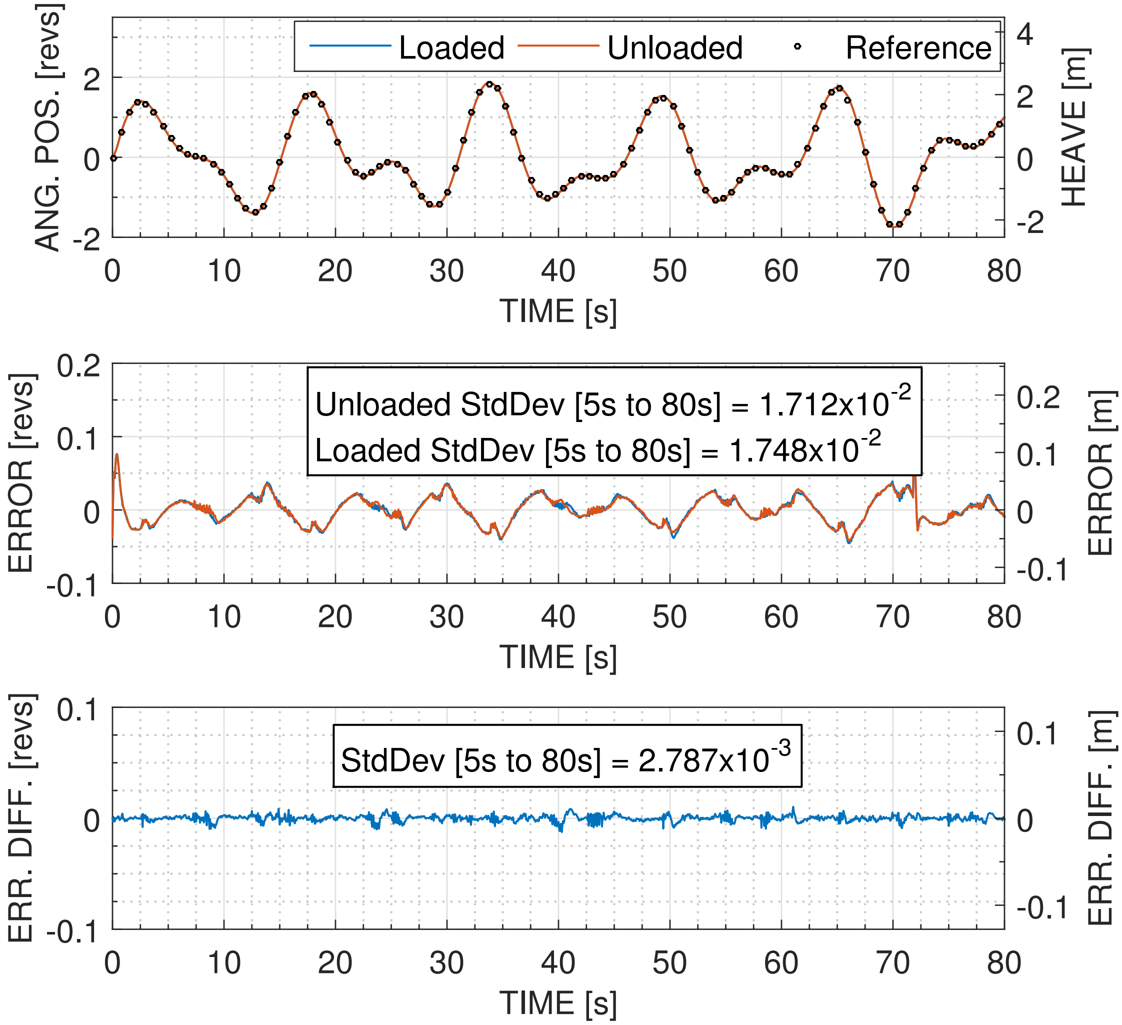


Figure 5.34: The loaded and unloaded simulator PID tracking Test Case A is shown here. The loaded error curve is shifted down due to motor leakage, seen clearly in the lower plot.

5.5.4 Test Case B

Figure 5.35 shows the loaded and unloaded simulator MPC tracking curves for Test Case B. Note the error curve peaks at t = 32 s, 37 s, 52 s, and 67 s maintain their shape between both curves. The Test Case B MPC error curves show a standard deviation difference of 1.056×10^{-2} revs between the loaded and unloaded simulator cases. Looking back at Test Case A, the loaded simulator MPC error standard deviation is 2.389×10^{-2} revs while the unloaded error standard deviation is 2.129×10^{-2} — a change of 2.6×10^{-3} revs. For the Benchmark cases, the difference between the loaded and unloaded simulator error standard deviations was 2.2×10^{-3} revs. Note that for the Benchmark Case, Test Case A, and Test Case B, the difference between the loaded and unloaded simulator errors varies, suggesting the offset caused by leakage is situation dependent. The PID tracking of Test Case B for the loaded and unloaded simulator shown in Figure 5.36 provides little further insight into the system.

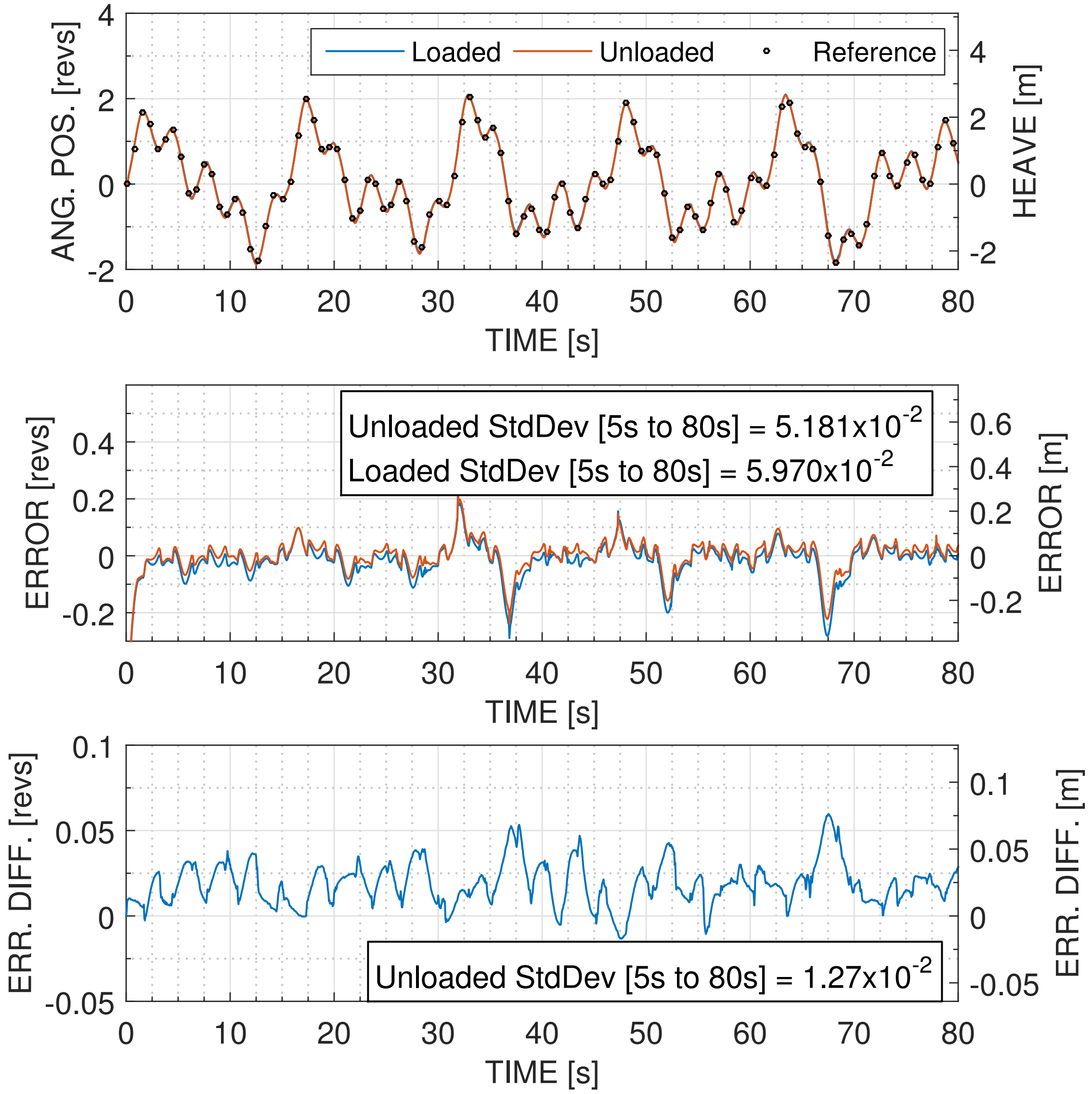


Figure 5.35: The loaded and unloaded simulator results for tracking Test Case B with MPC are shown here. The loaded error curve is shifted down due to motor leakage, but, otherwise the curves have identical features and follow each other.

Similar to the previous test data, the PID error curves deviate very little from each

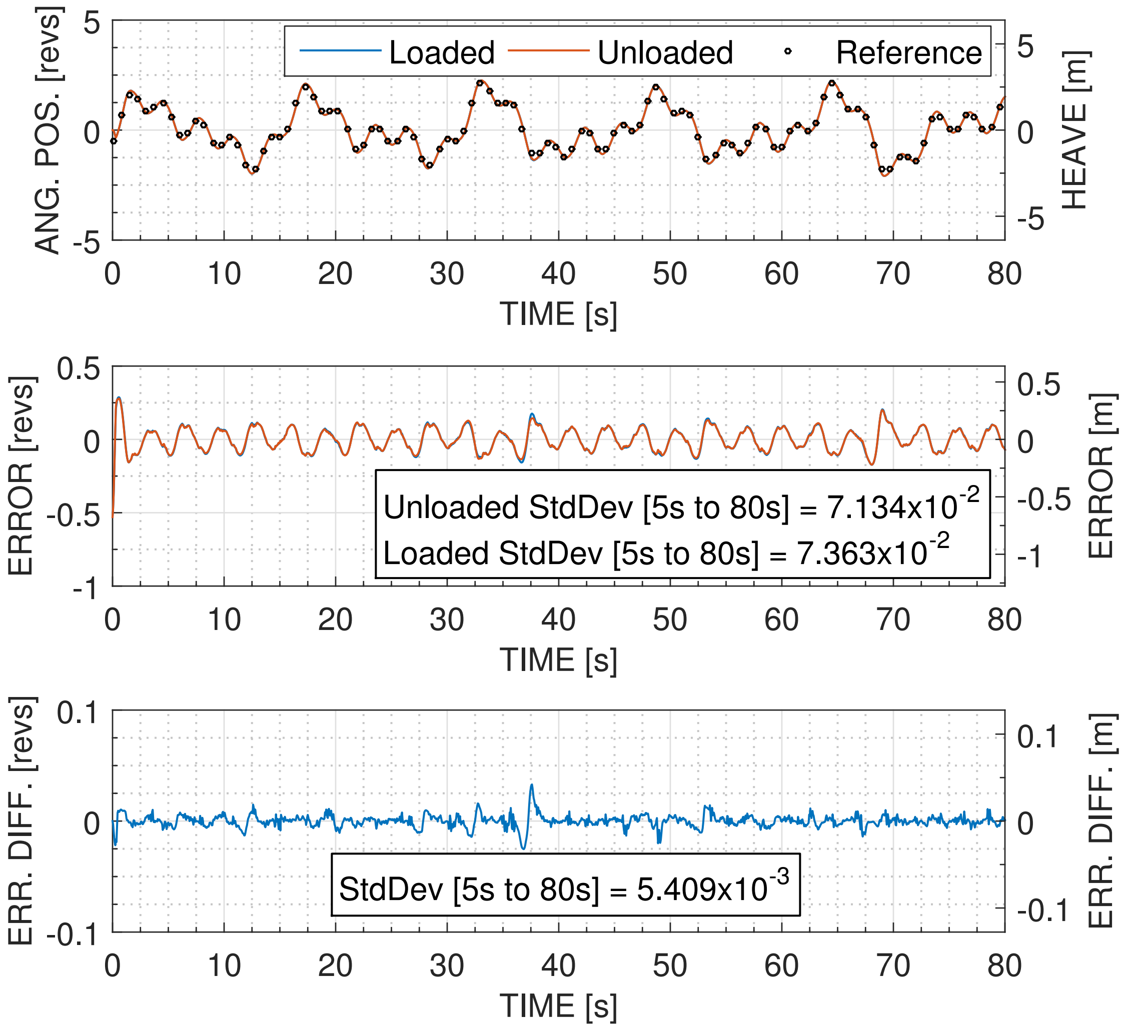


Figure 5.36: The loaded and unloaded simulator PID tracking Test Case B is shown here. The difference between error curves between the loaded and unloaded cases is minimal.

other, with an error standard deviation difference of 2.29×10^{-3} revs. This difference, an increase from the 3×10^{-4} revs difference for the Benchmark case and the 3.6×10^{-4} revs difference for Test Case A, is attributable to the added high frequency component to Test Case B.

5.5.5 Test Case C

Figure 5.37 shows the loaded and unloaded simulator results for PID tracking of Test Case C. For Test Case C, the PID error curve standard deviations differ by 1.1×10^{-3} revs, which again, is a larger difference when compared to the Benchmark data and Test Case A data, neither of which contained high frequency components.

Figure 5.38 shows the loaded and unloaded simulator results for MPC tracking of Test Case C. As with MPC tracking Test Case B, peaks in the error curves tracking Test Case C at times t = 10 s, 15 s, 35 s, and 50 s match well between the unloaded and loaded case, and again, as with the previous MPC tracking cases the loaded error curve is shifted with respect to the unloaded curve showing the motor lags behind the test case data when lifting a load.

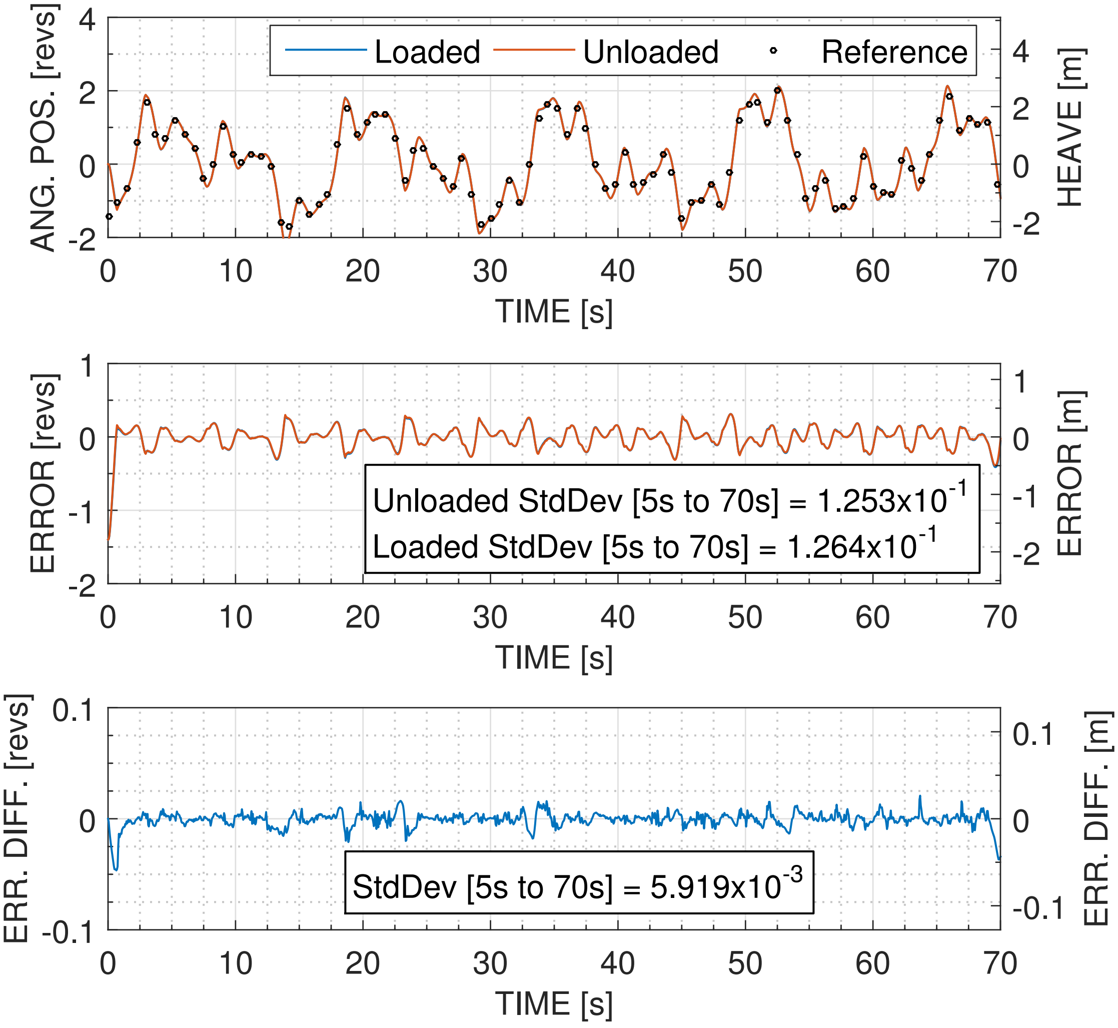


Figure 5.37: The loaded and unloaded simulator tracking Test Case C with PID is shown here.

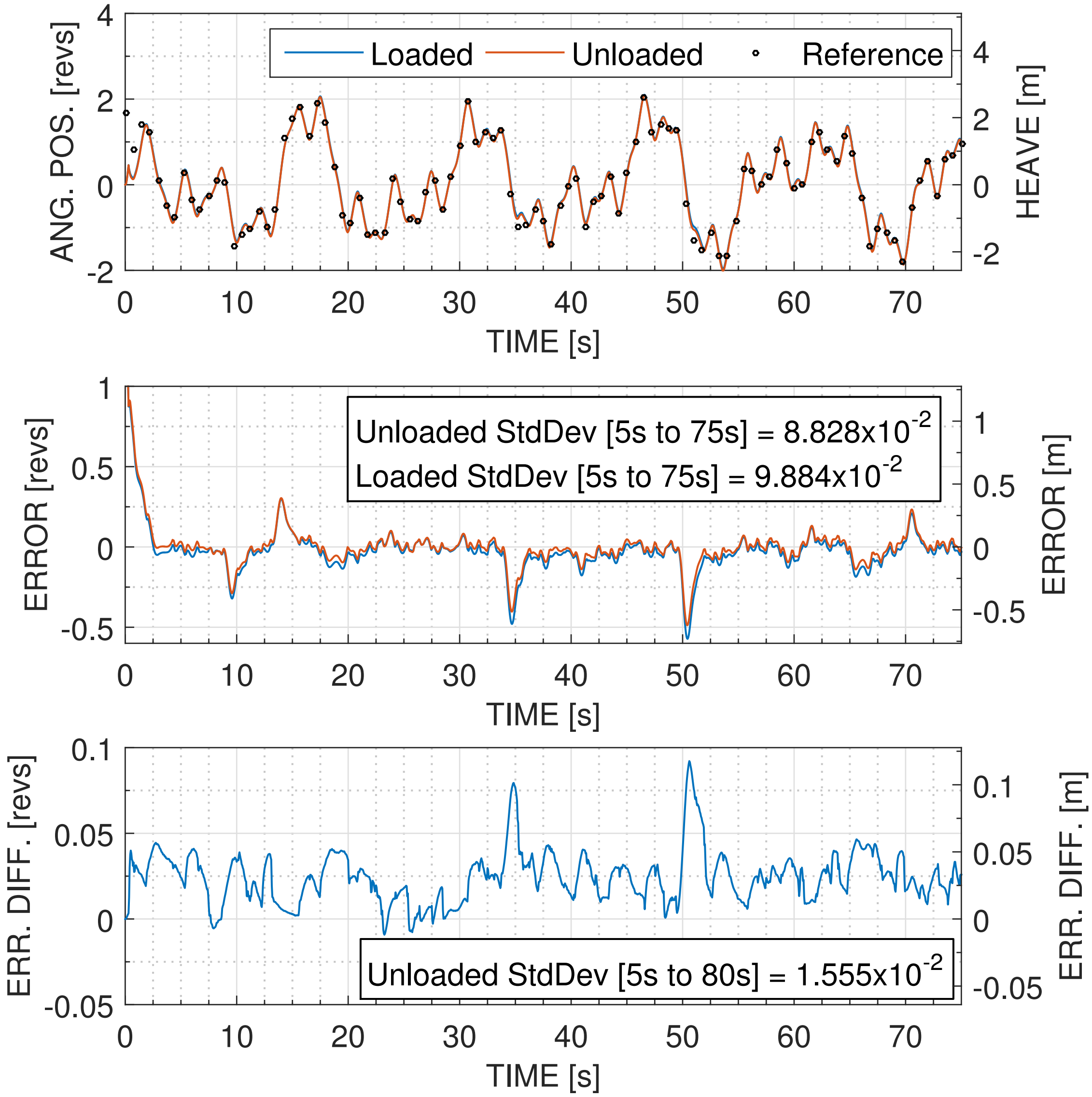


Figure 5.38: The loaded and unloaded simulator tracking Test Case C with MPC is shown here. The loaded error curve is shifted down due to motor leakage.

5.6 Conclusion

In this chapter a simulator for the unloaded full-scale AHC testbed was created and a load was applied to determine how the system would operate. Based on experimental data collected on the AHC testbed, Section 5.1 outlined the creation of a motor model for the BB4-800 motor, where a MATLAB Simulink response optimization algorithm matched the model behavior to the real motor behavior by adjusting friction parameters. In Section 5.2 flow data, pressure data, and the valve specification sheet data was used to model the PVG-120 proportional valve openings as a function of control voltage, creating a good approximation of the valve which was used to direct flow from the pump. Since the pump internals were not well known, Section 5.3 showed how a PID controller could approximate the AHC testbed pump instead of physically modeling the pump as was done for the motor and valve. This PID pump model likely resulted in some inconsistencies between the AHC testbed data and simulator results; however, Section 5.4 clearly shows deviations between the two are relatively small and the simulator results reasonably agree with the AHC testbed results.

In Section 5.4, MPC and PID controllers were used with the unloaded simulator to track sinusoidal moving references as well as the Benchmark data and Test Cases A, B, and C. The results were compared to the experimental results for the unloaded AHC testbed where it was found that tracking of a 0.15 Hz 2 rev sine wave began to show a large deviation between the simulator and experimental AHC testbed results for MPC. The PID controller showed good agreement between simulator and experiment when tracking each reference sine wave, regardless of frequency. When examining the Benchmark and Test Cases A, B, and C, the PID controller tracking ability within the simulator matched almost identically to the experimental results. The MPC controller error curves did not match as well as PID and, for the Benchmark case as well as Test Case A, PID tracked better than MPC. This improved PID tracking over MPC was only seen in the simulator, as Chapter 4 clearly shows MPC tracking better in all cases. An examination of the MPC implementation within MATLAB Simulink and LabVIEW is recommended to study the discrepancy between experimental and simulation tracking results.

Section 5.5 presented the results of applying a load to the simulated AHC testbed and tracking the test cases with MPC and PID control. Two important results were discovered: the first result was that a counter-balance valve would be needed when operating the AHC testbed under load. The second result was that when operating with MPC control under load the motor will lag behind the tracked motion when lifting a load, due to leakage in the motor, whereas the PID controller's integral term avoided this lag. These two results from Section 5.5 will allow future work on this project involving the application of a load to the full-scale AHC testbed to move more quickly, as there are two clearly defined issues which can be addressed.

Ideally, the loaded simulator behavior seen in this chapter will be seen when applying a load to the full-scale AHC testbed, further verifying that the simulator is an accurate representation of the full-scale AHC testbed. PID control shows promise in that it is load independent within the simulator environment, however, despite being load independent the PID control does not track as well for Test Cases B and C, due to the high frequency components. The addition of a parallel integral term to the MPC controller may reduce overall error for MPC tracking.

Chapter 6

Conclusions

The three key objectives of this work were:

1. To quantify and correct for the non-linear properties of a low-cost hydraulic components within a hydraulic AHC system
2. To implement a heave motion prediction MPC controller and actuate an unloaded full-scale AHC testbed under four test cases comparing the MPC results to a PID controller under the same conditions
3. To create a simulation of the AHC testbed to determine how the system operates under load conditions.

This chapter summarizes the main results of this thesis with respect to the three key objectives, and concludes with potential avenues for future work related to this project.

6.1 Objective One: Non-linear Hydraulic Properties

Chapter 3 presented results identifying a fricton model for a Black Bruin BB4-800 hydraulic motor and identified two non-linearities related to to a PVG-120 proportional control valve: the dead-band, and the non-linear response of flow through the valve with respect to the input control voltage. The friction model was utilized in Chapter 5 when creating a simulation model of the AHC testbed, while algorithms to correct for the valve non-linearities were applied in Chapter 4 to linearize the system for MPC control.

6.2 Objective Two: Heave Prediction MPC Control of an AHC Testbed

An AHC testbed model to use with the heave prediction MPC controller was identified in Chapter 3 and MPC was then implemented to actuate the unloaded full-scale AHC testbed in Chapter 4. It was shown that for tracking sine waves of 0.1 Hz, 0.125 Hz, 0.15 Hz, and 0.2 Hz that the MPC controller consistently tracked the sine wave with improved tracking error when compared to a tuned PID controller. For the Benchmark case and three test cases the heave prediction MPC controller was able to track the motion with reduced error compared to the PID controller. When noise was added to Test Cases A and B the ability for MPC to motion track was reduced due to increased error in the heave prediction algorithm; however, in a situation where noise became a significant issue a low-pass filter could normally be implemented for an AHC system. Another key point where noise may play a role is during model identification.

MPC requires a model but model identification is not always accurate due to issues such as signal noise or unknown phenomenon. It was shown at the end of Chapter 4 that if a first-order system model has corner frequencies identified within ±40% the MPC tracking ability is minimally effected. In the extreme, a corner frequency reduced by a factor of 10 leads to unacceptable tracking results for all cases, while a corner frequency increased by a factor of 10 shows reasonable tracking for tracking slow motion without high frequency components.

6.3 Objective Three: Simulator Results

In Chapter 5 the development of a simulator matching the physical properties of the unloaded, full-scale AHC testbed was presented. MPC was implemented on the simulator and tracking of the Benchmark case as well as the test cases showed similar tracking error magnitudes when compared to the experimental MPC tracking, however, there were noticeable differences. For PID implemented within the simulator the simulator tracking data and the experimental tracking data overlapped almost identically, suggesting the MPC implementation within MATLAB Simulink differs from that used during experiments.

Applying a load within the simulator and tracking the test cases with MPC provided two important results: a counter-balance valve was required in the hydraulic system to ensure the motor does not over-run, and motor leakage due to the high load pressure caused the loaded system motion to lag behind the unloaded system motion. This lag when using MPC can likely be corrected by adding a parallel integral controller to the MPC controller. A PID controller was implemented within the loaded simulator and it was found that the same lag noticed when using MPC did not occur. The integral action of the PID controller removed any offset.

In tracking the test cases under load, MPC still performed better than PID for cases where higher speed motions were demanded of the system. It is possible that were the simulator MPC implementation similar to the experimental MPC implementation then the simulator results would show MPC consistently outperforming compared to PID, as was seen for the experimental results in Chapter 4.

6.4 Future Work

This thesis work shows that MPC provides better results tracking a noise-free moving reference when compared to a tuned PID controller, but avenues to improve upon work and to follow other research paths always present themselves during a project. The following are a few suggestions for improvements and future work based on the work within this thesis:

1. The MPC controller used in this thesis work ignored valve spool dynamics when crossing the dead-band. An improvement to the current system could be to create a model for spool position as a function of control voltage and apply a non-linear gain based on spool position to better predict motor flow.

2. In Chapter 5 the HPU pump was modeled as a PID controller. An accurate pump model could lead to better agreement between simulator results and AHC testbed results.

3. Chapter 5 revealed the need for a counter-balance valve for the AHC testbed under load. With simulator results showing how the system may react under load, the real-world full-scale AHC testbed can be loaded to obtain results for comparison to the loaded simulator.

4. A cross-platform implementation of MPC for both MATLAB Simulink and LabVIEW would allow more consistent comparison between the two.

5. A ship at sea does not simply heave vertically; there are other motions to consider when performing AHC. Considering more degrees-of-freedom for ship motion and determining how those affect a heave signal could lead down new avenues of research.

6. It was found that the heave prediction algorithm was sensitive to signal noise. A low-pass filter should be applied to noisy heave data and then the heave prediction algorithms prediction capabilities should be compared when given the noisy signal, and the filtered signal.

Bibliography

[1] A. D. Little, “Stress analysis of ship-suspended heavily loaded cables for deep underwater emplacements,” *Department of the Navy, Bureau of Ships*, Aug 1963.

[2] J. Hatleskog and M. Dunnigan, “Heave compensation simulation for non-contact operations in deep water,” in *OCEANS 2006*, pp. 1–6, Sept 2006.

[3] S. Kuchler, T. Mahl, J. Neupert, K. Schneider, and O. Sawodny, “Active control for an offshore crane using prediction of the vessels motion,” *Mechatronics, IEEE/ASME Transactions on*, vol. 16, no. 2, pp. 297–309, 2011.

[4] J. Woodacre, R.J.Bauer, and R. Irani, “A review of vertical motion heave compensation systems,” *"Ocean Engineering*, vol. 104, no. 0, pp. 140 – 154, 2015.

[5] A. Southerland, “Mechanical systems for ocean engineering,” *Naval Engineers Journal*, vol. 82, no. 5, pp. 63–74, 1970.

[6] B. Butler, “Heave compensation,” in *Second Annual European Meeting of the Society of Petroleum Engineers of AIME*, Society of Petroleum Engineers, 1973.

[7] R. Hutchins, “Heave compensation system,” May 1978. US Patent 4,091,356.

[8] F. El-Hawary, “Compensation for source heave by use of a kalman filter,” *Oceanic Engineering, IEEE Journal of*, vol. 7, no. 2, pp. 89–96, 1982.

[9] E. Kidera, “At-sea handling and motion compensation,” in *OCEANS ’83, Proceedings*, pp. 766–770, 1983.

[10] R. A. Sullivan, M. R. Davenport, and R. E. Clements, “Multipurpose active/-passive motion compensation system,” in *16th Annual Offshore Technology Conference*, pp. 235–240, Offshore Technology Conference, April 1984.

[11] A. Hellrand, L. K. Moen, and T. Faanes, “Crane control system with active heave compensation and constant tension modes onboard the vessel stena wel/servicer,” in *20th Annual Offshore Technology Conference*, pp. 519–530, 1990.

[12] L. Robichaux and J. Hatleskog, “Semi-active heave compensation system for marine vessels,” 1993. US Patent 5,209,302.

[13] K. Do and J. Pan, “Nonlinear control of an active heave compensation system,” *Ocean Engineering*, vol. 35, pp. 558–571, April 2008.

[14] J. P. Blanchet and T. J. Reynolds, “Crane hook heave compensator and method of transferring loads,” 1977. US Patent No. 4003472.

[15] N. Barber, "Control means for motion compensation devices," Sept 1982. US Patent 4,349,179.

[16] A. Jones and T. Cherbonnier, "Active reference system," Oct 1990. US Patent 4,962,817.

[17] U. Korde, "Active heave compensation on drill-ships in irregular waves," *Ocean Engineering*, vol. 25, no. 7, pp. 541–561, 1998.

[18] J. M. Godhaven, "Adaptive tuning of heave filter in motion sensor," in *OCEANS '98 Conference Proceedings*, vol. 1, pp. 174–178 vol.1, 1998.

[19] S. Blake, C. Hill, T. Moore, C. Hide, and D. Park, "A heave compensation algorithm based on low cost gps receivers," *The Journal of Navigation*, vol. 61, no. 2, p. 291, 2008.

[20] A. Kyllingstad, "Method and apparatus for active heave compensation," Sept 2012. US Patent 8,265,811.

[21] "Active heave drilling drawworks system goes to work," *Offshore Magazine*, vol. 59, no. 4, 1999.

[22] V. D. Angelis, "Comparison study of electric, electro-hydraulic, and hydraulic drive science winches," in *11th European Research Vessel Operators Meeting 2009*, 2009.

[23] J. Kang, "Regenerative power units save energy," *Control Engineering (Online)*, March 13 2013.

[24] Marathon Motors, *Marathon Motors Product Catalog*, 2013.

[25] Bosch Rexroth, *Radial Piston Motor (Multi-Stroke) MCR20 Product Sheet*, 05.2012 ed., 2012.

[26] K. Godbole, "Field oriented control reduces motor size, cost and power consumption in industrial applications," September 2006.

[27] R. Parekh, "Ac induction motor fundamentals," *Microchip Technology Inc. Application Notes*, 2003.

[28] A. Nespoli, S. Besseghini, S. Pittaccio, E. Villa, and S. Viscuso, "The high potential of shape memory alloys in developing miniature mechanical devices: A review on shape memory alloy mini-actuators," *Sensors and Actuators A: Physical*, vol. 158, pp. 149–160, 3 2010.

[29] X. Liang and T. Virvalo, "What's wrong with energy utilization in hydraulic cranes," in *Proceedings of the 5th International Conference on Fluid Power Transmission and Control*, p. 419, 2001.

[30] P. Jones, “Maximizing hydraulic efficiency,” *Design Engineering (online)*, May 2012.

[31] P. Bennett, “Active heave: the benefits to operations as seen in the north sea,” in *Proceedings of the 1997 SPE/IADC Drilling Conference*, (Amsterdam, Neth), pp. 207–217, Sedco Forex, March 1997.

[32] G. Lee-Glauser, G. Ahmadi, and L. Horta, “Integrated passive/active vibration absorber for multistory buildings,” *Journal of Structural Engineering*, vol. 123, no. 4, pp. 499–504, 1997.

[33] L. Li and S. Liu, “Modeling and simulation of active-controlled heave compensation system of deep-sea mining based on dynamic vibration absorber,” in *Mechatronics and Automation, 2009. ICMA 2009. International Conference on*, pp. 1337–1341, Aug 2009.

[34] J. Hatleskog and M. Dunnigan, “Active heave crown compensation sub-system,” in *OCEANS 2007 - Europe*, pp. 1–6, June 2007.

[35] B. Eryilmaz and B. H. Wilson, “Unified modeling and analysis of a proportional valve,” *Journal of the Franklin Institute*, vol. 343, pp. 48–68, 1 2006.

[36] P. Gu, A. A. Walid, Y. Iskandarani, and H. R. Karimi, “Modeling, simulation and design optimization of a hoisting rig active heave compensation system,” *International Journal of Machine Learning and Cybernetics*, vol. 4, no. 2, pp. 85–98, 2013.

[37] J. R. Halliday, D. G. Dorrell, and A. Wood, “A fourier approach to short term wave prediction,” in *The Sixteenth International Offshore and Polar Engineering Conference*, May 2006.

[38] J. Neupert, T. Mahl, B. Haessig, O. Sawodny, and K. Schneider, “A heave compensation approach for offshore cranes,” in *American Control Conference, 2008*, pp. 538–543, 2008.

[39] F. Lewis, *Applied Optimal Control and Estimation: Digital Design and Implementation.* Digital signal processing series, Prentice Hall, 1992.

[40] K. McTaggart, *ShipMo3D Version 3.0 User Manual for Creating Ship Models.* Defense R&D Canada - Atlantic, 2011.

[41] Vickers Incorporated, *Vickers Industrial Hydraulics Manual*, 4th ed., 1999.

[42] Sampro Hydraulics Ltd., *Black Bruin Hydraulic Motors: Design Guide*, preprint 2012-10-31 ed., 2012.

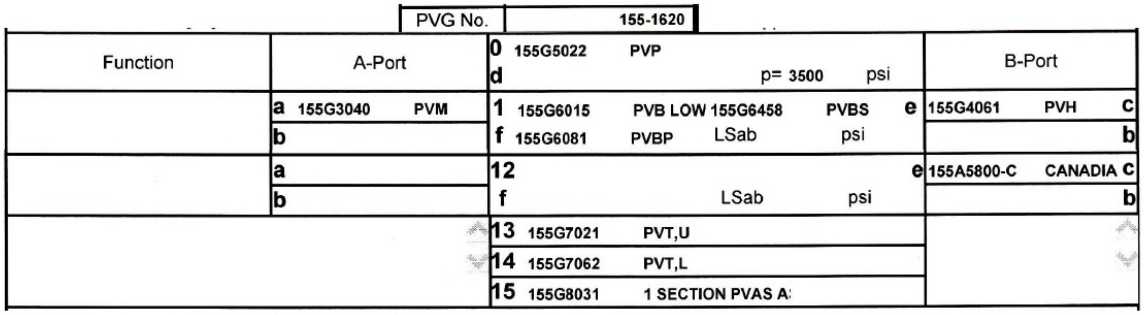
[43] Sampro Hydraulics Ltd., *Black Bruin Hydraulic Motors: Design Guide*, 0805: web ed., 2005.

[44] Danfoss, *Technical Information Proportional Valve Group PVG 120*, 520l0356 ed., Dec. 2013.

[45] Rexroth Bosch Group, *Counterbalance, standard poppet type differential area*, 18320-03/01.10 ed., Jan. 2010.

Appendix A

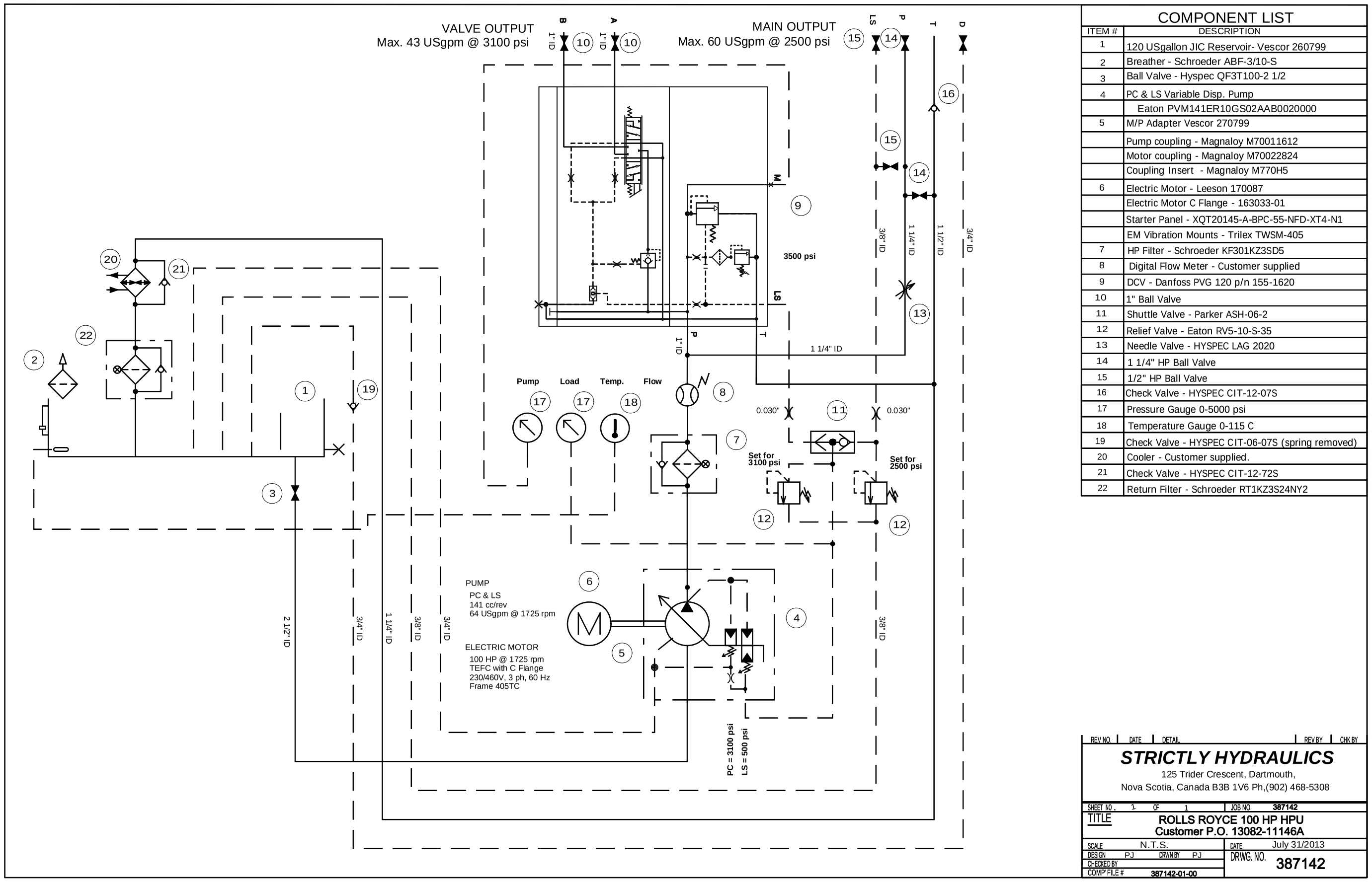
Valve Specifications and HPU Hydraulic Circuit



PVG No. 155-1620

Function	A-Port	0 155G5022 PVP d p= 3500 psi	B-Port
	a 155G3040 PVM	1 155G6015 PVB LOW 155G6458 PVBS e	155G4061 PVH c
	b	f 155G6081 PVBP LSab psi	b
	a	12 e	155A5800-C CANADIA c
	b	f LSab psi	b
		13 155G7021 PVT,U	
		14 155G7062 PVT,L	
		15 155G8031 1 SECTION PVAS A:	

Figure A.1: This build sheet provides the necessary components item numbers to purchase the exact PVG 120 valve used within this thesis work.



COMPONENT LIST

ITEM #	DESCRIPTION
1	120 USgallon JIC Reservoir- Vescor 260799
2	Breather - Schroeder ABF-3/10-S
3	Ball Valve - Hyspec QF3T100-2 1/2
4	PC & LS Variable Disp. Pump
	Eaton PVM141ER10GS02AAB0020000
5	M/P Adapter Vescor 270799
	Pump coupling - Magnaloy M70011612
	Motor coupling - Magnaloy M70022824
	Coupling Insert - Magnaloy M770H5
6	Electric Motor - Leeson 170087
	Electric Motor C Flange - 163033-01
	Starter Panel - XQT20145-A-BPC-55-NFD-XT4-N1
	EM Vibration Mounts - Trilex TWSM-405
7	HP Filter - Schroeder KF301KZ3SD5
8	Digital Flow Meter - Customer supplied
9	DCV - Danfoss PVG 120 p/n 155-1620
10	1" Ball Valve
11	Shuttle Valve - Parker ASH-06-2
12	Relief Valve - Eaton RV5-10-S-35
13	Needle Valve - HYSPEC LAG 2020
14	1 1/4" HP Ball Valve
15	1/2" HP Ball Valve
16	Check Valve - HYSPEC CIT-12-07S
17	Pressure Gauge 0-5000 psi
18	Temperature Gauge 0-115 C
19	Check Valve - HYSPEC CIT-06-07S (spring removed)
20	Cooler - Customer supplied.
21	Check Valve - HYSPEC CIT-12-72S
22	Return Filter - Schroeder RT1KZ3S24NY2

REV NO. DATE DETAIL REV BY CHK BY

STRICTLY HYDRAULICS
125 Trider Crescent, Dartmouth,
Nova Scotia, Canada B3B 1V6 Ph,(902) 468-5308

SHEET NO. 1 OF 1 JOB NO. 387142
TITLE ROLLS ROYCE 100 HP HPU
Customer P.O. 13082-11146A
SCALE N.T.S. DATE July 31/2013
DESIGN PJ DRWN BY PJ DRWG. NO. 387142
CHECKED BY
COMP FILE # 387142-01-00

Figure A.2: Strictly Hydraulics HPU.