Delft University of Technology
Master of Science Thesis in Computer and Embedded Systems Engineering

LiDAR-Based State Estimation for Offshore Crane Lower-Block Localization

Timotheüs Ruben Zunderman

Embedded Systems

TU Delft Delft University of Technology Huisman

LiDAR-Based State Estimation for Offshore Crane Lower-Block Localization

Master of Science Thesis in Computer and Embedded Systems Engineering

Embedded Systems Group
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Van Mourik Broekmanweg 6, 2628 XE Delft, The Netherlands

Timotheüs Ruben Zunderman

June 22th, 2026

Author
Timotheüs Ruben Zunderman (tzunderman@student.tudelft.nl)
(timozunderman@gmail.com)
Title
LiDAR-Based State Estimation for Offshore Crane Lower-Block Localization
MSc Presentation Date
July 8th, 2026

Graduation Committee

prof. dr. Koen Langendoen (chairman)	Delft University of Technology
Dimitris Boskos	Delft University of Technology
Lo Stouten (direct supervisor)	Huisman Equipement B.V.

Abstract

The rapid growth of the offshore wind energy sector demands the deployment of increasingly large wind turbines, requiring specialized heavy-lift vessels equipped with massive rotary cranes. Safe and efficient load transfer during maritime operations is severely challenged by lower-block (hook) sway, induced by vessel and crane motions. Automated anti-sway control can mitigate these operational risks, but relies on accurate real-time lower-block position and velocity feedback. While state-of-the-art lower-block localization approaches typically assume a rigid crane structure or require hook-mounted active hardware, this thesis presents a real-time state estimation framework based on vessel- and boom-mounted sensors. At the massive scale of offshore cranes, structural boom elasticity introduces significant bending deflections that corrupt joint-encoder measurements. To account for these dynamics, a multi-body non-linear model incorporating a flexible boom alongside wave-induced vessel motion is derived. This model serves as the predictor step for a multi-rate filtering architecture that fuses boom-mounted 3D LiDAR data, encoders, and a vessel Motion Reference Unit. Both Extended (EKF) and Unscented (UKF) Kalman Filters are implemented and evaluated. By explicitly accounting for structural flexibility, the framework successfully limits tracking errors, achieving a position RMSE of 1.67–3.58 cm and a velocity RMSE of 0.66–5.52 cm/s, comfortably satisfying the requirements of 5.0–10.0 cm and 5.0–10.0 cm/s per axis, respectively. The EKF systematically outperforms the UKF, yielding an average 5.2% lower position RMSE and an 11% reduction in computational runtime, demonstrating the framework's viability for real-time industrial anti-sway control.

Preface

My fascination with electronic systems and their integration into larger, complex wholes began at a young age, ultimately guiding me to a Bachelor's degree in Electrical Engineering followed by a Master's in Computer and Embedded Systems Engineering. Driven by a desire to apply these concepts to practical, real-world engineering, I knew I wanted to complete my thesis within an industrial environment. Over the course of eight months, this thesis became far more than an academic milestone: it allowed me to immerse myself in the completely new topic of state estimation. I am deeply grateful to Huisman for trusting me with this topic, giving me the freedom to find my own way despite my background in embedded systems, and supporting me as I discovered the path that led to this final result. This project marked my entry into the world of offshore cranes, a domain I initially perceived as handling only massive, sluggish machinery. However, as I delved deeper into the project, I quickly learned to appreciate the incredible care, control, and finesse required by systems of this scale. The resulting work is written for fellow engineers, researchers, and anyone interested in how advanced estimation can bring precision to heavy industrial crane applications.

Navigating this journey was an incredible learning experience, but it is one I could not have completed alone. I want to take this opportunity to thank the people who supported me along the way. First and foremost, I want to thank my supervisors, Lo and Koen, who were always ready to guide me, answer my questions, and provide valuable feedback on my work. I am also very grateful to Dimitris for brainstorming with me on the dynamics and estimator components of the project, and for helping me formulate these concepts correctly. My thanks also go to my colleagues at Huisman, whose time, patience, and willingness to answer questions about topics entirely new to me were immensely helpful. In particular, I want to thank Remco for introducing me to Huisman, welcoming me to Wandelclub "De Grote Blaar", and getting coffee for me. Finally, I want to express my deepest appreciation to my family, roommates, and friends. Thank you for your patience when I was tired or stressed, for helping me keep things in perspective, and for simply being there whenever I needed a helping hand.

Timo Zunderman

Delft, The Netherlands
22nd June 2026

Contents

Chapter 1

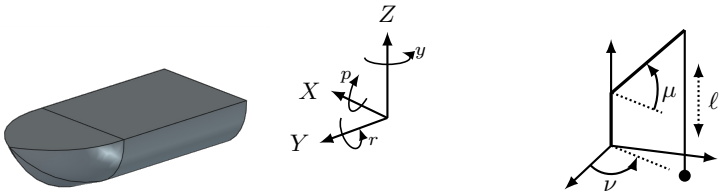
Introduction

1.1 Background

Global electricity demand is expected to grow by about 40% in the coming decade [16]. To meet this growing demand, new sources of electrical energy must be developed and deployed. To reduce carbon dioxide emissions caused by electricity generation, substantial research has focused on the development and construction of technologies that make use of renewable energy sources. Among these technologies, wind energy has emerged as an important contributor, converting the kinetic energy of the wind into electricity through large-scale turbines. In 2023, wind energy accounted for about 8% of global electricity generation [17], a share that is expected to increase to approximately 14% by 2030 [15].

Since the 1990s, wind turbines have increasingly been installed in offshore wind farms, where stronger and more consistent wind conditions, as well as the availability of large open areas, allow higher electricity production [28, 47]. Over the past decades, newly developed offshore wind farms generally contain a larger number of turbines with increasing dimensions [3]. Improvements in turbine design, particularly in blade length and hub height, have significantly increased power generation capacity [9]. The increasing turbine size directly impacts the installation complexity and the required capacity of lifting equipment.

The deployment of these massive offshore wind turbines necessitates large vessels equipped with specialized heavy-lift cranes. These vessels move in six degrees of freedom: surge, sway, heave, pitch, roll, and yaw, as can be seen in Figure 1.1(a). Cranes mounted on such installation vessels are typically rotary cranes with a main boom, capable of rotating around the vertical axis (slewing) and pivoting around a horizontal axis (luffing), while the load can be hoisted by one or more wire ropes (Figure 1.1(b)). Due to the large boom lengths and suspension heights, the lower-block (hook) and load behave as a pendulum with a long effective length, resulting in slow and lightly damped dynamics. The crane system is therefore underactuated, as only the boom motions can be directly controlled, while the motion of the load must be influenced indirectly.



(a) The six degrees of freedom of a vessel: translational movements comprising surge (Y), sway (X), and heave (Z), alongside rotational movements comprising pitch (p), roll (r), and yaw (y).

(b) The three operational degrees of freedom of a rotary crane: slewing around the vertical axis (ν), luffing the boom up and down (μ), and hoisting the load via the wire rope (ℓ).

Figure 1.1: **The degrees of freedom of a vessel and a rotary crane.**

During offshore lifting operations, the crane and load are subjected to various disturbances. Vessel motions induced by waves, wind forces acting on the load, and crane movements all contribute to undesired displacement of the lower-block, commonly referred to as sway. Excessive sway may lead to collisions with the vessel or turbine components, increase wear on crane structures, and create safety risks.

The installation phase represents a significant portion of the total investment cost of offshore wind projects, about 20–25% [37]. Due to the large size and mass of turbine components, installation operations are subject to strict safety limits, often defined by maximum allowable wave heights and wind speeds. These constraints result in narrow weather windows and can increase operational downtime and delays. Since installation vessels typically operate at high day rates, such delays directly impact project costs. Reducing sway has the potential to widen weather windows, decrease downtime, and lower installation expenses. Consequently, minimizing sway is essential for both safe and cost-effective offshore lifting operations.

Numerous sway reduction methods have been developed for land-based cranes, implementing feedforward strategies or feedback control loops using a wide variety of control principles and algorithms [14, 39]. Compared to other types of cranes, rotary cranes pose additional challenges due to their underactuated nature and the coupling between slewing, luffing, and load dynamics [11]. Offshore cranes further increase the complexity of sway mitigation, as vessel motions introduce additional degrees of freedom and act as unpredictable disturbances at the crane base, amplifying existing sway [1, 30]. In addition, the large scale of offshore cranes, with boom lengths exceeding 150 m, results in large moments of inertia and low achievable angular accelerations, limiting the crane's ability to react rapidly to sway. Furthermore, at this massive scale, the crane boom cannot be treated as a purely rigid body. Heavy loads and crane maneuvers induce significant structural bending and elastic oscillations that complicate the system dynamics even more.

The effectiveness of sway control depends strongly on accurate knowledge of the state of the lower-block, namely its position and velocity, relative to the vessel and crane tip. However, the large dimensions of offshore cranes make accurate localization of the lower-block challenging. Measurement distances may exceed 200 m, while centimeter-level accuracies are often required for safe and efficient operations. In addition, any localization system must be robust to environmental disturbances such as wind, rain, fog, and vibrations from onboard machinery. These requirements limit the applicability of conventional sensing solutions and motivate the need for specialized localization approaches specific to offshore crane operations.

Beyond control applications, accurate localization is also relevant for other aspects of offshore crane operations. The effective working radius of a crane changes continuously due to slewing motions combined with vessel pitch and roll, directly influencing load moments and permissible lifting limits. Furthermore, lifting heavy masses alters the vessel's center of gravity, which can affect stability margins during lifting operations. Reliable real-time knowledge of the lower-block position enables more precise estimation of these effects and supports safer operational decision-making.

Higher levels of automation and digitalization in offshore crane operations require precise localization. At present, crane operation depends heavily on the skill and experience of the operator, who must account for sway, changing working radii, and environmental disturbances. Existing support systems typically provide information such as crane angles, wire rope lengths, and camera views from selected perspectives. Even with these aids, safe and efficient operation remains demanding and requires extensive operator training. Partial automation and digital tools have the potential to reduce operator workload and training requirements.

A widely used method for reducing sway during offshore lifting operations is the use of tuggers. Tugger systems consist of one or more auxiliary wire ropes connecting the crane and the load (Figure 1.2). By actively controlling the tension in these ropes, additional forces can be applied to the load, allowing sway to be reduced. In some configurations, measurements of tugger rope lengths can also be used to infer information about the load position. However, the accuracy of such indirect estimates is often limited in practice, due to factors such as rope elasticity and slack. From a control and operational perspective, tuggers are also suboptimal as an anti-sway solution: the forces they can generate are typically limited, which reduces effectiveness for large loads and long effective pendulum lengths. Consequently, tuggers provide only partial sway mitigation and do not eliminate the need for accurate localization.

1.2 Research Context

This research was conducted in collaboration with Huisman Equipment B.V. [13],an industrial manufacturer of heavy lifting and handling systems for the marine energy sector. Executing this project within an industrial environment ensures that the developed state estimation framework is grounded in realistic oper-

Figure 1.2: **Tugger ropes connected to the load of a crane. These ropes can be used to reduce sway in crane operations.**

ational boundary conditions, physical constraints, and scale factors. Furthermore, this collaboration provided access to real-world structural parameters and engineering specifications used to construct and validate the simulation.

1.3 Problem Definition

The effectiveness of offshore anti-sway control relies heavily on high-accuracy, real-time state feedback of the lower-block (position and velocity). However, obtaining these accurate measurements is constrained by the operational environment and the sheer scale of the equipment. Measurement distances between the suspension point of the crane and the lower-block can exceed 200 m, and any localization hardware must withstand harsh offshore conditions, including wind, fog, and vibrations.

Beyond these environmental and hardware constraints, a fundamental challenge remains: because physical sensors typically provide only noisy, relative position data without direct velocity readings, the absolute state of the load must be mathematically reconstructed. This process of state estimation is dynamically highly complex. The suspension point of the pendulum is in constant motion due to a combination of wave-induced vessel kinematics and active crane maneuvers. Additionally, as previously established, the scale of modern offshore cranes means the boom structure cannot be accurately modeled as a perfectly rigid body. During lifting operations, the boom bends and oscillates under the load. This structural flexibility continuously alters the true kinematics of the suspension point, resulting in large offsets in the lower-block position.

Therefore, the core challenge lies in identifying suitable sensor technologies and measurement principles, and subsequently developing a state estimation framework capable of isolating the true pendulum dynamics from the complex, coupled motions of the vessel, the crane joints, and the flexing boom structure. This leads to the following primary research question:

RQ1. How can the real-time position and velocity of the lower-block be accurately estimated under the kinematic, dynamic, and environmental constraints of large-scale offshore lifting operations?

1.3.1 Evaluation Criteria

To systematically evaluate the suitability of existing localization technologies, a set of operational guidelines established by Huisman will be used. These baselines were formulated several years ago based on general offshore lifting experience. However, because the formal control-theoretic derivations for these values are undocumented, they serve only as a preliminary guideline for reviewing the state-of-the-art. A mathematical derivation of the exact requirements will be presented in Section 3.1.

One of the primary architectural restrictions in these guidelines is the avoidance of active sensors on the lower-block. While placing sensors directly on the hook is theoretically possible, it introduces severe operational drawbacks. Routing power and data cables along the hoist wire poses a high risk of entanglement and damage during complex lifting operations. Alternatively, relying on battery-powered sensors limits operational autonomy. Furthermore, transmitting real-time state data via wireless telemetry in offshore environments is highly susceptible to packet loss, limited bandwidth, and variable latency, which can degrade the stability of controllers. Therefore, crane- and vessel-mounted localization sensor architectures are strongly preferred.

Based on these operational considerations, Huisman suggests the localization system should target the following quantitative baselines and architectural constraints:

- A functional measurement range exceeding 200 m.
- A positional accuracy on the order of 10 cm along each coordinate axis.
- A velocity accuracy of approximately 10 cm/s along each coordinate axis.
- A minimum measurement update rate of 20 Hz.
- A sensor architecture where no active hardware is mounted on the lower-block.

These preliminary criteria provide the necessary context to assess the strengths and limitations of the sensing technologies discussed in Chapter 2.

1.4 Contributions

This thesis addresses the dual challenges of maritime sensor noise and unmodeled structural flexibility, thereby bridging a prominent gap in offshore crane literature, through the following primary contributions:

- **Sensor Evaluation and Architecture Design:** A systematic evaluation of localization technologies, leading to the design of a practical, crane- and vessel-mounted measurement framework. This architecture fuses relative boom-tip LiDAR measurements with existing vessel kinematics (Motion Reference Unit (MRU) and joint encoders), eliminating

the risks and challenges associated with sensors placed on the lower-block directly.

- **Integration of Structural Flexibility in Lower-Block Estimation:** The derivation of a dynamic model that explicitly accounts for structural boom bending by modeling a torsion spring. By incorporating these deformations into the state estimator, this work directly addresses a prominent gap in current offshore crane literature, which predominantly relies on rigid-body assumptions.
- **Comparative State Estimation Analysis:** The implementation, evaluation, and computational performance analysis of two non-linear state estimators: the Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF). This evaluation demonstrates the framework's capability to accurately reconstruct unmeasured velocity states and filter noise, successfully achieving a position Root Mean Square Error (RMSE) of 1.67–3.58 cm and a velocity RMSE of 0.66–5.52 cm/s. Furthermore, execution profiling validates that both algorithms execute 25 to 40 times faster than the physical timeline, confirming their computational efficiency.

1.5 Structure

The remainder of this thesis is organized as follows. Chapter 2 reviews the environmental challenges of marine operations and evaluates candidate tracking technologies based on literature. Subsequently, Chapter 3 translates maritime operational constraints into quantitative engineering requirements and details the selected system architecture. Chapter 4 establishes the mathematical formulation of the state estimation framework, detailing both the dynamic models and the filter algorithms. To evaluate this framework, Chapter 5 outlines the construction of the simulation environment, the simulated crane maneuvers, and the performance metrics. Chapter 6 delivers the performance evaluation of both filters. Then, Chapter 7 synthesizes these empirical findings against the initial design criteria to formulate a definitive architectural recommendation. Finally, Chapter 8 discusses the modeling limitations of this study and outlines trajectories for future research.

Chapter 2

Related Work

This chapter reviews the literature and established research surrounding lower-block localization. First, the physical and environmental non-idealities that complicate lower-block tracking are defined. Then, candidate localization technologies are evaluated against the preliminary operational criteria established in Section 1.3.1. This literature review provides the mechanical and architectural justification for the multisensor framework proposed in Chapter 3.

2.1 Operational and Environmental Challenges

The primary challenges in offshore crane localization originate from the physical behavior of the crane–load system and its harsh operating environment. Before evaluating specific localization technologies, these physical and environmental constraints are defined, as they serve as the primary criteria for sensor selection.

2.1.1 Non-Rigid Bodies

In many crane modeling and control approaches, the crane structure is assumed to behave as a rigid body. In practice, however, this assumption is often violated. Under heavy loads, structural components of the crane, particularly the boom, deflect due to their finite stiffness. This structural deformation is demonstrated in the Finite Element Method simulation shown in Figure 2.1, which illustrates the significant displacement a lattice boom can experience during operation. This behavior has been demonstrated in several studies, for example in [18, 25]. Depending on factors such as load mass, structural design, and working radius, these deflections can become significant.

The impact of structural flexibility is illustrated in [36], which observed vertical position errors of up to 1 m under load due to combined boom bending and wire rope elasticity. Because these deflections depend heavily on the crane's luffing angle [18] and the load, compensating the measurements for deflection, as the authors suggest, would require identifying offsets across a wide range of operating conditions, significantly increasing calibration complexity. Internal measurements at Huisman Equipment indicate that boom tip deflections in large offshore cranes can be on the order of 1 m. Consequently, accurately localizing

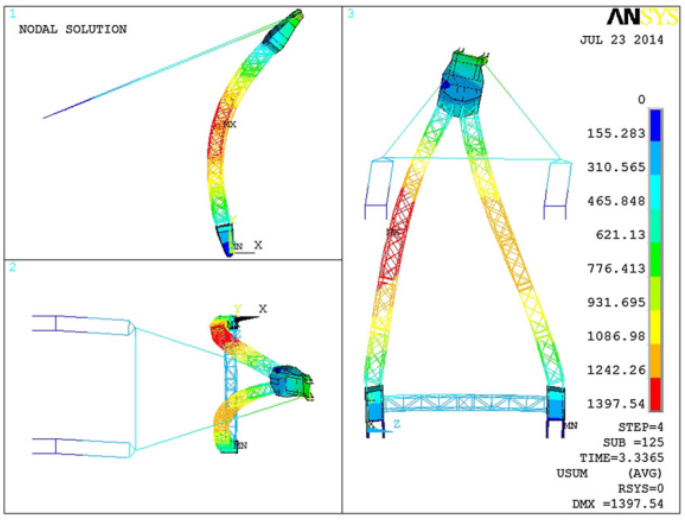


Figure 2.1: **Example of boom deflection in an offshore crane calculated using a Finite Element Method simulation. The colors represent the true magnitude of displacement, while the deformation of the structure is visually exaggerated for clarity. Retrieved from [25].**

the lower-block requires real-time estimation of the actual boom tip position, rather than relying on rigid-body kinematics and joint encoders.

2.1.2 Wire Ropes

Wire ropes are often modeled as massless, rigid links connecting the lower-block to the boom tip. However, this simplification introduces distinct physical non-idealities that affect specific measurement principles. First, wire ropes are elastic and stretch under load, as noted in [36]. For heavy offshore loads, this stretching directly corrupts the hoist-length measurements derived from winch encoders.

Second, when the mass of the wire rope is non-negligible relative to the combined mass of the lower-block and load, the rope can no longer be assumed to remain straight. Instead, it behaves as a distributed-mass system and may exhibit harmonic oscillations and bending [41, 43, 44], as illustrated in Figure 2.2. These effects primarily impact measurement approaches that derive the lower-block position based on the assumed the geometry of the rope at the boom tip. For sensing methods that directly measure the position of the lower-block itself, such as vision-based tracking or radio-based localization, rope oscillations and bending have little influence on the estimated position.

2.1.3 Vibrations

Besides low-frequency rope oscillations, high-frequency vibrations can also affect measurement quality. These vibrations may originate from multiple sources, including crane engines and motors, wind excitation, and structural resonance

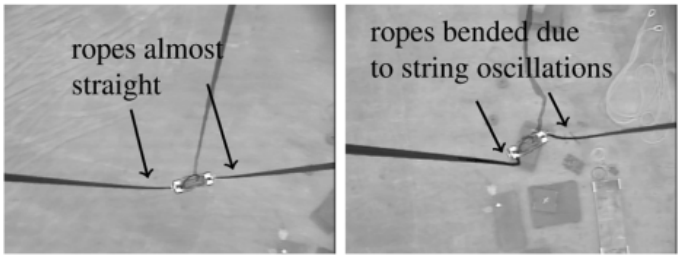


Figure 2.2: **Downward view from the suspension point with no rope oscillations (left) and heavy rope string oscillations (right). Note that multiple wire segments are visible due the use of a pulley system in the lower-block. Retrieved from [44].**

modes of the crane itself. Such effects have been reported to influence sensor performance in several crane-related studies, for example in [22, 32].

Vibrations primarily introduce sensor-level distortions rather than actual load motion. For optical sensors, cameras may suffer from motion blur, and LiDAR systems may produce distorted or shifted point clouds [49]. In contrast, Inertial Measurement Units (IMUs) directly measure these high-frequency accelerations. While these vibration components are undesirable for tracking low-frequency sway dynamics, they do not corrupt the underlying IMU measurements in the same manner as optical blur. Through signal processing techniques such as low-pass or Kalman filtering, this high-frequency measurement noise can often be effectively decoupled from the true pendulum dynamics, allowing it to be used to compensate distorted measurements from other sensors.

2.1.4 Weather Conditions

Weather conditions influence offshore crane operations in two different ways. First, adverse weather increases external disturbances. Strong winds acting on the load increase sway amplitudes, while wave-induced vessel motions affect the crane base and propagate through the structure. These effects make the load dynamics more complex and increase the demands on sensing and control.

Second, weather can directly reduce sensor performance. Visibility may be reduced by fog, rain, sea spray, or low clouds, leading to a loss of Line of Sight (LoS). Precipitation and spray can also introduce false returns or temporary signal loss in laser-based systems. While radar and radio-based sensors are inherently more robust to precipitation, severe weather can still reduce overall measurement reliability.

2.2 Localization Sensors

As shown in Section 1.1, extensive research has been conducted on reducing crane sway. Many active control schemes rely on the sway angles and angular velocities of the wire rope as primary feedback inputs. These quantities can

either be measured directly or derived from the relative distance between the boom tip and the lower-block.

When using relative distance measurements, the state of the crane, commonly referred to as its pose, must also be known. Estimating this pose requires additional sensors, such as rotary encoders to measure the slewing and luffing angles, and a Motion Reference Unit to measure the pitch and roll of the vessel. By combining the crane pose with the relative measurement to the lower-block, the position and velocity of the lower-block can be reconstructed.

This section reviews commonly used localization sensors, evaluating their mounting locations, practical limitations, and overall suitability for large-scale offshore cranes.

2.2.1 Accelerometer and Gyroscope

Inertial Measurement Units (IMUs) and Motion Reference Units (MRUs) utilize accelerometers and gyroscopes to measure linear accelerations and angular velocities. While MRUs perform onboard sensor fusion to directly output spatial orientation and translational motion, the underlying sensing principles are identical. Therefore, the terms IMU and MRU are used interchangeably for the remainder of this thesis. In crane applications, these sensors are typically mounted either near the suspension point at the boom tip [43, 44] or directly on the lower-block [33, 41, 42].

When mounted at the boom tip, an auxiliary mechanical structure is required to physically measure rope angles [44]. However, as established in Section 2.1.2, heavy wire ropes are subject to bowing and harmonic oscillations, which results in the measured top angle not providing an accurate representation of the true lower-block position. Furthermore, at the required measurement range of 200 m, even fractional angular measurement errors at the boom tip translate into significant positional errors at the lower-block, easily violating the 10 cm accuracy requirement.

IMUs mounted on the lower-block avoid this amplification, but introduce different practical challenges. These include variations in hook orientation due to tilting, as well as power supply and data transmission constraints, as explained in Section 1.3.1. To address power delivery, [41] proposes integrating a dynamo with a charging circuit that generates electrical power during hoisting.

A limitation of IMU-based lower-block localization is reported in [6], where an IMU mounted on the lower-block was used to estimate sway angles. The authors observed that the results became inaccurate in environments with a high presence of metallic structures. Although the exact cause was not explicitly stated, such decreases in accuracy are likely related to magnetic disturbances affecting the magnetometer measurements often used for IMU-based orientation estimation.

Finally, a fundamental mathematical limitation of all inertial sensors is integration drift. Obtaining a position estimate from an accelerometer requires double

integration over time, while obtaining an angle from a gyroscope requires single integration. Consequently, any small sensor bias or high-frequency measurement noise accumulates, leading to an unbounded drift in the estimated state.

Because of this inherent integration drift and the practical constraints of lower-block mounting, standalone IMUs cannot reliably fulfill the absolute positioning requirements of large-scale offshore lifting. However, they remain highly valuable in multisensor fusion architectures to capture high-frequency dynamics and vibrations, provided they are paired with sensors capable of distance measurement.

2.2.2 Camera

Cameras are widely used on offshore cranes, primarily to provide operators with additional visual perspectives. Because the hardware is often already integrated into practical crane setups, there is considerable research effort in extending camera systems beyond operator assistance toward automated load localization and sway monitoring. In crane applications, cameras are typically mounted near the boom tip and oriented downward to observe the lower-block. This downward perspective isolates the lower-block's motion against the background, simplifying target detection.

Several studies have utilized downward-facing cameras combined with machine learning to track the lower-block [6, 36, 52]. While positional accuracies of approximately 10 cm can be achieved at short distances of 10 m [6], all authors reported that performance degrades significantly as the distance from the camera to the lower-block increases. Furthermore, these optical systems proved highly sensitive to environmental factors, including decreased visual contrast [6] and sudden lighting variations [36].

Addressing the vibration challenges discussed in Section 2.1.3, Hassan and Miura [12] investigated the use of multiple cameras fused with IMUs to estimate the pose of the boom tip itself. The IMU was successfully used to compensate for high-frequency vibrations. Their results showed that stereo camera setups outperformed single-camera configurations, and that fusing camera and IMU data significantly improved estimation robustness. However, while this fusion approach effectively mitigates vibration, it remains fundamentally limited by distance. The authors only achieved centimeter-level accuracy at ranges of a few meters, meaning this solution cannot scale to fulfill the 200 m requirement of large offshore cranes.

Despite their flexibility and hardware availability, camera-based localization systems possess critical limitations for offshore crane applications. Their performance relies entirely on maintaining LoS, which can be easily blocked by rigging or the load itself. Crucially, their inherent sensitivity to distance-scaling errors, variable lighting, and adverse offshore weather conditions (e.g., heavy rain, dense fog, and glare) prevents standalone camera systems from reliably satisfying the 200 m range and 10 cm accuracy baselines.

2.2.3 Radar

Radar systems determine distance and velocity by analyzing the reflections and Doppler shifts of transmitted radio waves. In crane applications, radar sensors are typically mounted near the boom tip or on the crane mast to track the lower-block. Although rarely utilized for precise state estimation in crane literature, automotive radar demonstrates highly robust object detection up to 200 m even in adverse weather [35, 38].

Despite its robustness to adverse weather conditions, radar possesses significant limitations when applied to precise lower-block localization. The achievable range resolution is fundamentally limited by the signal bandwidth [8] and is typically in the decimeter range for automotive radar systems, with practical performance degrading at longer distances. Furthermore, the angular resolution of radar systems is constrained by antenna size and configuration [8]. Modern radar systems often use electronic scanning with phased array antennas to steer the beam without moving parts, resulting in limited angular resolution, reducing the ability to distinguish closely spaced objects [35]. As with camera systems, direct LoS is required. However, due to radar's lower spatial resolution, it is generally less sensitive to small occlusions than camera-based systems. Conversely, severe multipath reflections from the massive metallic crane structures and surrounding vessel installations may corrupt the signal, further complicating accurate target identification.

2.2.4 LiDAR

LiDAR systems measure the Time of Flight (ToF) of emitted laser pulses to generate high-resolution 2D or 3D point clouds. Because it utilizes short-wavelength laser light rather than radio waves, LiDAR achieves significantly higher spatial resolution than radar, enabling highly precise distance measurements and the detection of small features. In crane applications, LiDAR sensors are typically mounted on rigid structures, providing a direct view of the lower-block.

Several studies have investigated the use of laser-based distance measurements for crane applications. For example, Lee et al. [24] placed a reflector on the lower-block and a laser sensor near the suspension point to measure the distance between the two. The authors explicitly note that sway was not considered in their approach. As a result, the reflector may move out of the sensor's LoS in the presence of significant sway, preventing reliable distance measurements.

Modern 3D scanning LiDARs overcome the need for physical reflectors by capturing the entire workspace. Ul Hassan et al. [49] demonstrated this by mapping the surroundings of a crane. To overcome the limited field of view of fixed 3D LiDARs, a rotating 2D LiDAR was employed to reconstruct a full 3D point cloud. Crucially, an IMU was integrated to compensate for crane motion and high-frequency vibrations. In a related application, a boom-mounted rotating 2D LiDAR successfully tracked personnel entering a danger zone beneath a suspended load [22].

Beyond crane-specific applications, LiDAR is widely used in autonomous vehicles

due to its high point cloud density and accurate depth perception compared to cameras and radar. Commercial 3D LiDAR systems achieve detection ranges exceeding 200 m with centimeter-level precision. However, LiDAR is more sensitive to adverse weather conditions such as fog, rain, and spray. Qian et al. [38] showed that combining LiDAR with radar significantly improves detection performance in decreased visibility conditions compared to using either sensor alone.

Despite its high spatial resolution, LiDAR presents several challenges for lower-block localization in large offshore crane operations. Accurate lower-block tracking requires continuous LoS between the sensor and the load. Also, a boom-mounted LiDAR only provides the relative position between the boom tip and the lower-block. Extracting the lower-block location relative to vessel requires sensor fusion with the vessel's kinematic sensors and a way to incorporate the structural boom flexibility.

2.2.5 UWB

Ultra-Wideband (UWB) is a radio-based technology where active tags emit short pulses to fixed anchor sensors. Tag locations are determined via Time Difference of Arrival (TDOA), Angle of Arrival (AOA), or ToF. By strategically distributing anchors on the vessel and tags on the crane and lower-block, UWB can theoretically provide both relative and absolute position estimates, provided direct LoS is maintained.

UWB has been successfully applied in crane-related contexts, although its use for direct lower-block localization remains limited. In [48], UWB tags were attached to equipment and personnel to track their positions on a construction site, achieving a localization accuracy of approximately 10–15 cm. Similarly, Zhang et al. [53] used UWB for crane pose estimation and collision avoidance, localizing both the boom tip and the lower-block with an accuracy of approximately 25 cm at short distances of 5–10 m.

Other studies have demonstrated UWB's capability at much longer ranges. Liu et al. [29] introduced temperature compensation for long-range UWB measurements, reducing the ranging error to 4.4 cm at distances up to 400 m. This compensation also improved the statistical properties of the measurement noise. Furthermore, [31] reported a theoretical accuracy limit of approximately 2 cm for TDOA-based UWB localization. By incorporating phase information, the authors achieved sub-centimeter accuracy at 10 m and an error of 3.7 cm at 30 m.

Compared to conventional narrowband radio or radar systems, UWB's short pulses make it robust to multipath effects and signal reflections from metallic structures [48, 53]. However, its application to offshore crane localization is severely hindered by practical and geometric constraints. Reliable operation of TDOA- and AOA-based systems still depends on maintaining LoS between the tag and multiple anchors. In addition, UWB requires the placement of fixed anchors, which becomes non-trivial when the load is positioned above open water. Lastly, the active nature of UWB tags requires an onboard power source. While modern UWB systems are energy-efficient, with battery lifetimes

spanning months to years, they still require active hardware on the lower-block.

2.2.6 GNSS

Global Navigation Satellite Systems (GNSS) measure signal ToF to provide absolute positioning. While Real-Time Kinematics (RTK) achieves centimeter-level precision using a local reference station, Precise Point Positioning (PPP) eliminates local infrastructure requirements by solely relying on precise satellite models. In crane applications, receivers can be mounted on various components provided a clear LoS to the sky is maintained.

High-precision GNSS has been successfully applied to offshore lifting. Maes et al. [32] fused RTK-GNSS with an IMU to track a wind turbine blade, achieving an absolute position error of just 1–2 cm. However, RTK requires a base station within tens of kilometers [50, 51], severely restricting its use in deep-water wind farms. While PPP offers a shore-independent alternative with a centimeter-level horizontal accuracy and a vertical accuracy on the order of 10 cm [51], it suffers from long initialization times—often exceeding 20 minutes to mathematically resolve carrier-phase ambiguities [5]. Hybrid techniques, such as PPP-RTK networks, have been proposed to mitigate this by broadcasting regional atmospheric corrections, reducing the convergence time to minutes or seconds [27].

Despite its capability for high precision under ideal conditions, GNSS possesses severe operational limitations that restrict its suitability for lower-block localization in offshore crane operations. First, GNSS requires uninterrupted LoS to multiple satellites in the sky, making the system highly vulnerable to intentional or unintentional signal blockage and blind spots created by the crane mast, boom, and lifted components. Second, the heavy presence of metallic structures and the sea surface induce multipath reflections, which can corrupt the phase measurements required for RTK or PPP accuracy. Furthermore, standard GNSS receivers typically output data at frequencies of 1 to 10 Hz, falling short of the 20 Hz minimum update rate required for stable anti-sway control. Finally, because GNSS antennas must be mounted directly on the tracked object, localizing the lower-block explicitly requires active hardware and wireless telemetry on the hook.

2.3 Synthesis and Comparison

To reliably control crane sway, the localization system must satisfy performance requirements from Section 1.3.1: e.g. a functional range of 200 m, a positional accuracy of 10 cm, and a minimum update rate of 20 Hz. Based on the synthesis in Table 2.1, the majority of evaluated technologies fundamentally fail to meet these constraints.

Standalone IMUs suffer from unbounded integration drift, where position errors grow proportionally to time squared. Cameras become highly sensitive to pixel-level noise at distances up to 200 m, making 10 cm accuracy highly unlikely. Similarly, while radar provides robust range measurements, its limited angular resolution is insufficient for centimeter-level lateral positioning. UWB systems

struggle to guarantee 10 cm accuracy at a 200 m range given the restricted anchor geometry on a vessel. Finally, typical marine GNSS receivers operate below the required 20 Hz update rate and are susceptible to signal occlusion caused by the crane geometry.

After evaluating these performance metrics, LiDAR emerges as the only technology capable of consistently providing centimeter-level relative position measurements at the required 200 m range. A boom tip-mounted 3D LiDAR with downward view also satisfies the secondary operational preference to avoid instrumenting the lower-block. By eliminating the need for active sensors, batteries, and wireless data transmission on the hook, it avoids operational risks and autonomy issues. Therefore, LiDAR is selected as the primary sensing technology for this thesis.

However, relying on a standalone boom-mounted LiDAR introduces specific kinematic challenges. LiDAR systems typically operate around 10–30 Hz and do not measure velocity directly. Furthermore, because LiDAR only measures the relative position to the lower-block, the calculated absolute pendulum state is highly vulnerable to unmodeled boom deflection and vessel motion. To overcome these challenges and reconstruct the unmeasured velocity and absolute position of the lower-block, the LiDAR must be integrated into a multisensor fusion architecture alongside crane joint encoders and a vessel MRU. The mathematical derivation of the exact system requirements and the design of this state estimation architecture are detailed in Chapter 3.

Table 2.1: **Comparative analysis of sensor technologies for lower-block localization.** Sensitivity denotes the impact on measurement accuracy. Cell colors indicate suitability against the evaluation criteria set in Section 1.3.1: green denotes favorable performance, orange denotes marginal or conditional performance, and red denotes poor performance or a direct violation of the criteria.

	Typical accuracy level	Typical update rate	Sensor location	Sensitivity to boom deflection[c]	Sensitivity to wire rope effects	Sensitivity to vibrations	Sensitivity to weather	Key advantages	Key limitations
IMU	N/A[a]	100–1000 Hz	Suspension point	High	High	High	Low	Can be used to filter out vibrations for other sensor types	Inaccuracies near the suspension point get amplified, possible interference caused by metal objects
			Lower-block	Low	Low	High	Low		Power delivery and data transmission, tilting, possible interference caused by metal objects
Camera	dm-m	>30 FPS[b]	Suspension point	High	Low	High	High	Often already present	Lighting conditions, visual contrast, LoS, lower accuracy at larger distances
Radar	dm	10–50 Hz	Boom	High	Low	Low	Low	Direct velocity measurement possible, relatively resilient to multipath	Resolution, LoS
			Crane mast or vessel	Low	Low	Low	Low		
LiDAR	cm	10–30 Hz	Boom	High	Low	Medium	High	Resolution	LoS
			Crane mast or vessel	Low	Low	Medium	High		
UWB	dm	10–100 Hz	Suspension point and lower-block	Low	Low	Low	Low	Relatively resilient to multipath, easy to track multiple	Power delivery and data transmission, LoS
GNSS	cm-dm	1–20 Hz	Suspension point and lower-block	Low	Low	Low	Medium	Absolute positioning	Power delivery and data transmission, LoS, susceptible to radio interference

[a] An IMU's position accuracy is not listed because its position estimate is derived from integration, which causes rapid drift without correction from an external sensor.

[b] Frames Per Second.

[c] Boom-mounted sensors measure relative position. Determining the absolute load position requires the exact crane pose, which is corrupted by structural deflection.

Chapter 3

System Overview

This chapter established the engineering specifications and structural architecture of the proposed state estimation framework. First, the requirements and assumptions are defined to establish the boundaries of the system. Then, the overall multisensor architecture is proposed to satisfy these requirements. Finally, the reference frames that will be used to derive the dynamic model are defined.

3.1 Requirements and Assumptions

To successfully design and evaluate the state estimation framework, the overarching operational objectives and environmental constraints must be defined. These parameters are first used to mathematically derive the quantitative performance requirements. Finally, the necessary system modeling assumptions are established to map these requirements to a tractable mathematical representation.

3.1.1 Operational Requirement

Maximum sway The primary operational objective defined by Huisman is to limit absolute lower-block sway to a maximum of 0.5 m during offshore lifting operations in the future.

3.1.2 Environmental Constraints

The environmental constraints define the working area of the crane and are based on typical operational ranges used by Huisman.

- Sway angles relative to the vessel remain within $\pm 15^\circ$ in both horizontal directions under worst-case environmental conditions.
- The considered operational lower-block distance from the boom tip or suspension point is between 10–200 m. Rope lengths below 10 m are not representative for large offshore lifting operations, while lengths exceeding 200 m are outside the intended scope.
- The considered operational luffing angles are 15°–75°.
- There is no mechanical or operational limit on slewing angles.
- Wave-induced vessel motions are assumed to be bounded by a pitch and roll amplitude of 2° with a wave period of at least 4 s during crane operations.

- Weather conditions are within safe operational limits, meaning that there is no major impact on the performance of sensors.

3.1.3 Performance Requirements

The operational requirement and environmental constraints are used to derive the following performance requirements in this section.

Update rate According to Franklin et al. [7], a digital control loop bandwidth should encompass 2–5 times the highest dominant frequency of the system dynamics, while Åström and Wittenmark [40] demonstrate that the measurement sampling rate should be 10–20 times this control bandwidth to minimize discretization effects. For small angles, the natural frequency of a pendulum system can be calculated by

$$f = \frac{1}{2\pi}\sqrt{\frac{g}{\ell}}, \tag{3.1}$$

where f is the pendulum frequency, $g = 9.81\,\mathrm{m/s^2}$ is the gravitational acceleration, and ℓ is the length of the wire rope. This results in a natural frequency range of approximately 0.035–0.16 Hz.

Another dominant frequency component is wave-induced motion, which directly excites the pendulum dynamics. The upper bound for the pitch and roll frequency is 0.25 Hz (period of 4 s), which is higher than the 0.16 Hz natural frequency of the crane. Therefore, this maximum wave frequency is used to determine the measurement frequency. Using the guidelines above results in a recommended frequency range of 5–25 Hz. An update rate of 25 Hz is selected to provide sufficient margin against unmodeled dynamics and to ensure robust digital implementation, which aligns well with the guideline described in Section 1.3.1.

Position accuracy According to [7], sensor accuracy should be 5–10 times better than the required position control accuracy. Applying this to the 0.5 m operational sway limit yields a required position accuracy of 5–10 cm per axis.

Velocity accuracy The velocity measurement must be accurate enough to allow the control system to react to sway dynamics effectively. A common practice is to select a velocity accuracy that is 1–2% of the maximum expected velocity. For a pendulum with a static suspension point, this maximum velocity is calculated by

$$v_{\mathrm{max,sway}} = \sqrt{2g\ell(1-\cos\theta_0)}, \tag{3.2}$$

where v is the velocity of the mass, g is the gravitational acceleration, ℓ is the pendulum length, and θ_0 is the initial sway angle. This maximum is reached when the mass is directly below its suspension point. While the environmental worst-case sway limit was defined as 15°, active precision lifting operations generally halt or engage anti-sway control before such extremes are reached. Therefore, a nominal operational maximum of 5° is used for this derivation. Using a smaller angle conservatively limits $v_{\mathrm{max,sway}}$, which consequently results in a stricter accuracy requirement. Using the maximum considered length (200 m) and a sway angle of 5°, $v_{\mathrm{max,sway}}$ becomes 3.86 m/s.

In the worst case, the crane is slewing at full speed while the pendulum is swinging with the operational maximum sway. The velocities of both components are added together, resulting in a higher total velocity. The slewing speed of such cranes is approximately 0.3 rpm, or 0.0314 rad/s, according to Huisman. Since it is not realistic to slew at the maximum speed when the boom is horizontal, a boom radius of 50 m is considered. This gives $v_{\text{max,slew}} = r\omega_{\text{max,slew}} = 1.57\,\text{m/s}$. This results in a combined maximum lateral velocity of 5.43 m/s.

Luffing and hoisting velocities act primarily in the radial and vertical directions, whereas maximum sway velocity acts tangentially. Therefore, the combination of slewing and sway provides the most conservative basis for estimating maximum velocity. Applying the 1–2% baseline to the 5.43 m/s maximum yields a target velocity accuracy requirement of 5–10 cm/s alongside each individual axis.

Real-time execution bounds To support integration within digital control infrastructure, the state estimation framework must be compatible with real-time execution. The selected output update rate of 25 Hz defines a strict maximum processing time budget of 40 ms per execution loop. However, minimizing actual computational execution time is highly preferred to minimize latency within an active anti-sway control loop.

Requirements Summary The derivations for these requirements yield results mostly consistent with the initial guidelines provided by Huisman (Section 1.3.1). This alignment validates the mathematical approach and establishes a firm set of engineering targets for the remainder of this thesis. To summarize, the multisensor architecture must be designed to meet the following quantitative baselines: an output update rate of at least 25 Hz, a positional position estimation accuracy of 5–10 cm per axis, and a velocity estimation accuracy of 5–10 cm/s per axis, while maintaining compatibility with real-time execution.

3.1.4 System Modeling Assumptions

To mathematically represent the crane and vessel for state estimation, the following modeling assumptions are made:

- The vessel is approximated as a rigid body with six degrees of freedom.
- Vessel pitch and roll occur around a single, fixed center of rotation.
- Vessel yaw is not explicitly modeled. Following the operational envelope defined by Huisman, the vessel is assumed to maintain a fixed heading.
- Vessel translational motions (surge, sway, and heave) are omitted from the kinematic equations. Due to the crane's scale, boom-tip acceleration is heavily dominated by the moment-arm effect of vessel rotations, allowing translations to be neglected.
- The crane boom behaves as a flexible (non-rigid) body.
- The suspension point is assumed to geometrically coincide with the boom tip.
- The lower-block and its attached load are combined and modeled as a single point mass.
- Wire ropes are modeled as massless, rigid links. While heavy ropes exhibit elastic stretch and bowing (Section 2.1.2), unmodeled boom deflection would introduce significantly larger positional offsets. Therefore, to constrain the

scope and complexity of the dynamic model, compensating for boom flexibility is prioritized, and rope elasticity is excluded.
- High-frequency structural vibrations are not explicitly modeled in the crane's system dynamics. As explained in Section 2.1.3, these vibrations primarily introduce sensor-level noise rather than macroscopic load motion. Due to the effect of unmodeled boom deflections being larger, these vibrations are excluded as well.
- Sensor noise is assumed to be zero-mean additive white Gaussian noise.
- Sensor data are assumed to be pre-processed. For example, LiDAR measurements are assumed to provide directly usable relative position estimates derived from point-cloud processing.

3.2 Proposed Architecture

As concluded in Chapter 2, a boom-mounted 3D LiDAR is the optimal primary sensor for lower-block localization. However, the derived performance requirements in Section 3.1.3 (i.e., a 25 Hz update rate and 5–10 cm/s velocity accuracy per axis) highlight the limitations of using LiDAR as a standalone device. Commercially available LiDAR systems typically operate at update rates around 10–30 Hz and do not directly measure velocity.

Furthermore, the LiDAR only measures the relative distance to the lower-block. During offshore operations, the boom tip itself moves continuously due to wave-induced vessel motions, active crane maneuvers, and unmodeled structural boom deflection. To accurately determine the absolute sway of the load and distinguish it from the motion of the boom tip, the system must compensate for the movement of the sensor.

Therefore, the LiDAR data must be combined with measurements from crane joint encoders and an MRU for vessel orientation and angular rates. Because these sensors are standard equipment on modern offshore installation vessels, this architecture does not require additional hardware beyond the LiDAR itself. By fusing these combined sensor inputs within a model-based state estimator, the noisy position measurements can be filtered and the unmeasured velocity states accurately reconstructed to meet the requirements. The design of this state estimation architecture is detailed in Chapter 4.

3.2.1 Data Flow

The logical data flow of the proposed architecture is illustrated in Figure 3.1. In real-world operation, the boom tip LiDAR provides noisy measurements of the relative position of the lower-block. Together with information about the vessel orientation and crane pose, these measurements are fed into the estimator alongside a predictive dynamic model of the crane's dynamics. The recursive estimator continuously propagates the system's states using a non-linear dynamic model and corrects these states as new sensor data arrive, ultimately outputting a filtered 3D position and reconstructing the unmeasured velocity vectors.

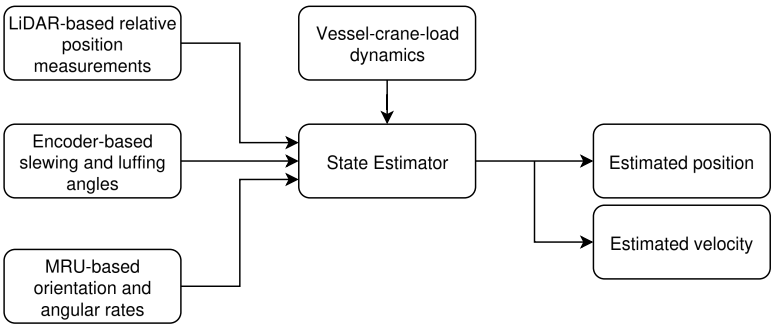


Figure 3.1: **Data flow of the proposed architecture. The recursive state estimator fuses real-time relative position data from a boom-mounted LiDAR with measurements from joint-integrated rotary encoders, a vessel-based MRU, and its own past state estimates.**

3.2.2 Simulation Framework

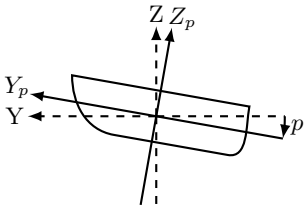
To develop and validate the proposed state estimation architecture, a combination of MATLAB/Simulink and Python is utilized. Simulink serves as the primary simulation environment, as it is an industry standard for control systems and natively supports a variety of discrete-time filters. It is used to simulate the "ground truth" physics of the crane, generate synthetic sensor data with injected noise, and execute the discrete state estimation algorithms. Furthermore, Python scripts are employed alongside the simulation to symbolically derive the complex non-linear dynamics and to visualize the 3D kinematic results.

3.3 Reference Frames

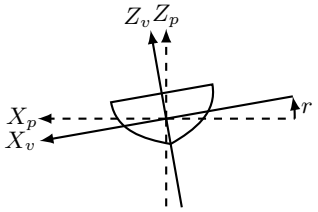
To derive the mathematical equations describing the swinging motion of the lower-block, four distinct reference frames are established to model the coupled crane–vessel system, which are based on [11]. While the vessel and crane could mathematically share a single coordinate system, keeping them separate ensures the model directly mirrors the physical sensor topology. This isolates the passive, wave-induced disturbances from the active mechanical rotations. Consequently, this structure simplifies the forward kinematic transformation chain used to derive the dynamic equations, mapping coordinates through the following sequence: *Inertial Frame* $\rightarrow$ *Vessel Frame* $\rightarrow$ *Crane Frame* $\rightarrow$ *Load Frame.*

The Inertial Frame The Inertial Frame serves as the absolute, earth-fixed coordinate system. Its Z-axis is aligned opposite to the gravity vector (pointing upwards), while the X and Y axes lie in the horizontal plane. All absolute positions and velocities, including the final estimated state of the lower-block, are evaluated relative to this global frame.

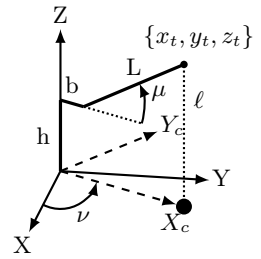
The Vessel Frame The Vessel Frame is body-fixed to the center of rotation of the vessel. Its Y-axis points towards the bow, its X-axis points towards the starboard side, and its Z-axis points upward relative to the deck. As illustrated in Figures 3.2(a) and 3.2(b), the Vessel Frame is rotated relative to the Inertial



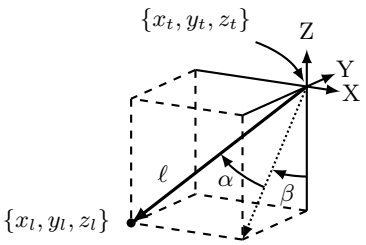
(a) Pitching rotation of the vessel: rotation around the X-axis of the Inertial Frame in the positive direction. This results in the bow moving up for a positive pitch.



(b) Rolling rotation of the vessel: rotation around the pitched Y_p-axis in the positive direction. The bow points out of the paper in this figure and a positive roll results in the port side of the vessel moving up.



(c) The crane components in the Crane Frame, where h is the height of the mast, b is the joint offset from the Z-axis, L is the boom length, ℓ is the wire rope length, μ is the luffing angle, and ν is the slewing angle. The base axes X, Y, and Z align with the orientation of the Vessel Frame, while X_c and Y_c represent the axes after the slewing rotation.



(d) Definition of the Euler angles, α and β, used to describe the location of the lower-block $\{x_l, y_l, z_l\}$ relative to the boom tip suspension point $\{x_t, y_t, z_t\}$. The orientation is defined by two subsequent rotations: a primary rotation around the fixed X-axis by β, followed by a secondary rotation around the new Y-axis by α.

Figure 3.2: **Coordinate rotations and parameter definitions of the vessel–crane–load system.**

Frame by the vessel's pitch around the X-axis and roll around the pitched Y-axis. Defining the axes in this manner creates a right-handed coordinate system, where a positive pitch results in the bow moving up.

The Crane Frame The origin of this frame is located at the base of the crane where it is attached to the vessel. This frame is oriented around its Z-axis such that the boom of the crane points along the X-axis of the Crane Frame. The origin of the Crane Frame is offset by a constant translational vector relative to the origin of the Vessel Frame, since the rotational center of the vessel is unlikely to align with the base of the crane. With the slewing and luffing angles defined, the pose of the crane structure is fixed relative to the vessel deck, as can be seen in Figure 3.2(c).

The Load Frame The origin of the Load Frame is dynamically located at the boom tip (the suspension point) within the Crane Frame, where the axes of the Load Frame align with those of the Crane Frame. From this origin, two rotational sway angles (α and β) and the length of the wire rope (ℓ) define the location of the lower-block point mass relative to the boom tip, as shown in Figure 3.2(d).

Chapter 4

State Estimation

A straightforward approach to obtaining the velocity of the lower-block is through numerical differentiation of the position measurements. However, differentiating a noisy signal amplifies high-frequency noise. Given the system requirements of a 25 Hz update rate (an sampling period of 0.04 s) and a positional accuracy of 0.1 m, simple differentiation would yield a velocity noise magnitude of approximately $0.1/0.04 = 2.5\,\mathrm{m/s}$. Because this error is more than an order of magnitude larger than the required velocity accuracy, direct differentiation is infeasible. Therefore, a more robust state estimation technique is required to reconstruct the unmeasured velocity states accurately and provide a clean signal for an anti-sway controller.

4.1 Estimators

Several filtering techniques can be employed to estimate velocity from noisy position data. This section evaluates these methods against the non-linear kinematics of the offshore crane to select the most suitable algorithms for the state estimation framework.

The simplest approach is combining numerical differentiation with a low-pass filter to smooth the resulting high-frequency noise spikes. However, aggressive low-pass filtering introduces significant time delays into the estimated signal, which degrades control performance. To avoid this lag, model-based estimators are preferred. These algorithms use a mathematical representation of the system to predict how the states evolve, reducing reliance on raw numerical differentiation.

The Kalman Filter (KF) [21] is the standard model-based estimator, which is mathematically optimal for linear systems subject to Gaussian noise. However, the swinging dynamics of an offshore crane are inherently non-linear. To address non-linear system dynamics, variants such as the Extended Kalman Filter (EKF) [45] and Unscented Kalman Filter (UKF) [20] are commonly used. The EKF handles non-linearities by calculating Jacobian matrices to linearize the system around the current state estimate at each time step. In contrast, the UKF uses the Unscented Transform to propagate a minimal set of sample points (sigma points) through the non-linear equations, capturing the system's mean and variance without explicit linearization. The EKF is generally computationally less demanding, while the

UKF often provides improved estimation accuracy for strongly nonlinear systems, as the Unscented Transform typically captures nonlinear transformations more accurately than the first-order linearization employed by the EKF [20, 45]. However, comparative studies have shown that the relative performance of both filters is highly application-dependent and that the UKF does not universally outperform the EKF [4].

For systems subject to extreme non-linearities and non-Gaussian noise, Particle Filters (PF) [10] offer an alternative by utilizing sequential Monte Carlo simulations. However, PFs require simulating a vast number of particles, demanding significant computational power [45]. Because the measurement noise in this application can be reasonably approximated as Gaussian, the PF is deemed excessively computationally heavy. Therefore, the EKF and UKF are selected as the candidate estimators. To determine the optimal balance between tracking accuracy and computational efficiency for this specific offshore crane architecture, both will be comparatively evaluated in simulation.

Regardless of their specific approaches to handling non-linearities, both the EKF and UKF operate on a shared recursive, two-step algorithm. This architecture establishes the structure for the remainder of this chapter:

- **Prediction Step:** The filter uses the mathematical physics model to project the current state forward in time. Because physical models are inherently imperfect, the filter tracks both the estimated state vector and its associated uncertainty. During the prediction step, this uncertainty grows to reflect model inaccuracies. Defining this step requires the continuous dynamic model (Section 4.2), the state-space representation (Section 4.3), and a numerical discretization method (Section 4.4).
- **Correction Step:** When a new sensor reading arrives, the filter propagates the predicted state through a measurement model to generate an predicted sensor reading. It then computes the Kalman Gain: a dynamic weighting matrix that balances the accumulated uncertainty of the dynamical model against the known noise profile of the sensor. This gain is multiplied by the difference between the actual and predicted sensor readings, to update and correct the predicted state estimate. Defining this correction step requires explicit measurement models mapping the state to the physical sensors (Section 4.5).

By continuously repeating this cycle, the filter fuses the noisy sensor data with the predictive physics model. The exact software configuration and tuning process are detailed in Section 4.6.

4.2 Dynamics

Since both the EKF and UKF require a mathematical representation of the swinging motions of the lower-block, a dynamic model must be established. A crane load suspended by a wire rope fundamentally behaves as a spherical pendulum. Therefore, modeling a simplified 3D pendulum serves as the baseline kinematic framework. This section first establishes this foundational model, which is subsequently expanded to describe the motion of the lower-block for a crane with a static base, a vessel-mounted crane, and finally a vessel-mounted crane featuring a flexible boom.

4.2.1 3D Pendulum

To describe the location of the lower-block load combination relative to its suspension point, a specific Euler angle parameterization (α and β) is utilized based on [11]. These angles describe two subsequent rotations, as explained in Section 3.3 and illustrated in Figure 3.2(d). Using this rotation sequence, the position of the pendulum mass is given by:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \ell \begin{bmatrix} -\sin\alpha \\ \cos\alpha\sin\beta \\ -\cos\alpha\cos\beta \end{bmatrix}. \tag{4.1}$$

Standard spherical coordinate systems are avoided here because they present a mathematical singularity when the pendulum hangs straight down, resulting in the azimuth angle being undefined at the crane's equilibrium state. Conversely, the parameterization in (4.1) only encounters a singularity when $\beta = \pm\frac{\pi}{2}$. Because this condition corresponds to the lower-block swinging completely horizontal to the boom, the model remains well-defined throughout all realistic offshore lifting scenarios.

Differentiating the position with respect to time yields the velocity of the lower-block:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \ell \begin{bmatrix} -\dot{\alpha}\cos\alpha \\ -\dot{\alpha}\sin\alpha\sin\beta + \dot{\beta}\cos\alpha\cos\beta \\ \dot{\alpha}\sin\alpha\cos\beta + \dot{\beta}\cos\alpha\sin\beta \end{bmatrix}. \tag{4.2}$$

The kinetic energy T of the point mass representing the lower-block load combination is given by:

$$T = \frac{1}{2}mv^2 = \frac{1}{2}m\left(\dot{x}^2 + \dot{y}^2 + \dot{z}^2\right) = \frac{1}{2}m\ell^2\left(\dot{\alpha}^2 + \dot{\beta}^2\cos^2\alpha\right), \tag{4.3}$$

and its potential energy V is given by:

$$V = mgh = mg(\ell + z) = mg\ell(1 - \cos\alpha\cos\beta). \tag{4.4}$$

where m is the mass of the lower-block. Using the kinetic and potential energies, the Lagrangian L is defined as:

$$L = T - V = \frac{1}{2}m\ell^2\left(\dot{\alpha}^2 + \dot{\beta}^2\cos^2\alpha\right) - mg\ell(1 - \cos\alpha\cos\beta). \tag{4.5}$$

The equations describing the motions for this system are derived using the Euler-Lagrange equation:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} = \frac{\partial L}{\partial q_i}, \tag{4.6}$$

where L is the Lagrangian and q_i represents the generalized coordinates (the independent degrees of freedom of the system). For this system, the generalized coordinates are α and β. An equation for a generalized coordinate is called an Equation of Motion (EoM). For α, the partial derivatives of the Lagrangian are evaluated as:

$$\begin{aligned} &\frac{\partial L}{\partial \dot{\alpha}} = m\ell^2\dot{\alpha} \quad \Longrightarrow \quad \frac{d}{dt}\frac{\partial L}{\partial \dot{\alpha}} = m\ell^2\ddot{\alpha}, \\ &\frac{\partial L}{\partial \alpha} = -m\ell^2\dot{\beta}^2\sin\alpha\cos\alpha - mg\ell\sin\alpha\cos\beta. \end{aligned} \tag{4.7}$$

Equating these terms yields the EoM for α:

$$\ddot{\alpha} = -\dot{\beta}^2 \sin\alpha\cos\alpha - \frac{g}{\ell}\sin\alpha\cos\beta. \tag{4.8}$$

For β, the partial derivatives are:

$$\begin{aligned} \frac{\partial L}{\partial \dot{\beta}} &= m\ell^2\dot{\beta}\cos^2\alpha \quad \Longrightarrow \quad \frac{d}{dt}\frac{\partial L}{\partial \dot{\beta}} = m\ell^2\left(\ddot{\beta}\cos^2\alpha - 2\dot{\alpha}\dot{\beta}\sin\alpha\cos\alpha\right), \\ \frac{\partial L}{\partial \beta} &= -mg\ell\cos\alpha\sin\beta, \end{aligned} \tag{4.9}$$

which results in:

$$\ddot{\beta} = 2\dot{\alpha}\dot{\beta}\tan\alpha - \frac{g}{\ell}\frac{\sin\beta}{\cos\alpha}. \tag{4.10}$$

(4.8) and (4.10) thus form the complete set of non-linear EoMs for the 3D pendulum.

4.2.2 Crane with a Static Base

To incorporate crane kinematics, the Crane Frame (denoted by C, with corresponding subscript c in mathematical notation) is used, where the crane can freely slew, luff, and hoist, as defined in Section 3.3 and depicted in Figure 3.2(c). Using these definitions, the location of the lower-block relative to the suspension point at the boom tip can be calculated by (4.1). The absolute location of the suspension point t in the Crane Frame C is defined by its coordinates p_{tc}:

$$p_{tc} = \begin{bmatrix} x_{tc} \\ y_{tc} \\ z_{tc} \end{bmatrix} = \begin{bmatrix} b + L\cos\mu \\ 0 \\ h + L\sin\mu \end{bmatrix}. \tag{4.11}$$

To transform this location from the rotated Crane Frame to an unrotated base frame, a temporary Inertial Frame (I) is established with its origin at the crane base. The rotation matrix R_{ic} is then applied to map coordinates from C to I:

$$p_{ti} = \begin{bmatrix} \cos\nu & -\sin\nu & 0 \\ \sin\nu & \cos\nu & 0 \\ 0 & 0 & 1 \end{bmatrix} p_{tc} = R_{ic}p_{tc}. \tag{4.12}$$

This yields the following absolute locations for the suspension point p_{ti} and the load p_{li} in the temporary Inertial Frame I:

$$p_{ti} = \begin{bmatrix} x_{ti} \\ y_{ti} \\ z_{ti} \end{bmatrix} = R_{ic}\begin{bmatrix} x_{tc} \\ y_{tc} \\ z_{tc} \end{bmatrix} = \begin{bmatrix} (b + L\cos\mu)\cos\nu \\ (b + L\cos\mu)\sin\nu \\ h + L\sin\mu \end{bmatrix}, \tag{4.13}$$

$$p_{li} = \begin{bmatrix} x_{li} \\ y_{li} \\ z_{li} \end{bmatrix} = \begin{bmatrix} x_{ti} \\ y_{ti} \\ z_{ti} \end{bmatrix} + \ell R_{ic}\begin{bmatrix} -\sin\alpha \\ \cos\alpha\sin\beta \\ -\cos\alpha\cos\beta \end{bmatrix}. \tag{4.14}$$

To formulate (4.14), the relation from (4.1) is utilized to determine the relative lower-block location in the Load Frame. This relative vector is then rotated by the crane's slewing matrix (R_{ic}) and added to the absolute suspension point location.

After calculating the derivative of (4.14) with respect to time, (4.3), (4.4), (4.5), and (4.6) can be used to calculate the EoMs. Due to the number of time-varying variables (α, β, μ, ν, and ℓ) and the repeated application of the chain rule, the number of terms grows immensely, making manual derivation highly susceptible to algebraic errors. Therefore, a symbolic computation script was written in Python (see Appendix B) to derive the final EoMs, resulting in (A.1) and (A.2).

4.2.3 Vessel-Mounted Crane

Integrating vessel motions caused by waves requires the use of the Vessel Frame (V). The orientation of the vessel relative to the Inertial Frame is given by the rotation matrix R_{iv}:

$$\begin{aligned} R_{iv} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos p & -\sin p \\ 0 & \sin p & \cos p \end{bmatrix} \begin{bmatrix} \cos r & 0 & \sin r \\ 0 & 1 & 0 \\ -\sin r & 0 & \cos r \end{bmatrix} \\ &= \begin{bmatrix} \cos r & 0 & \sin r \\ \sin p \sin r & \cos p & -\sin p \cos r \\ -\cos p \sin r & \sin p & \cos p \cos r \end{bmatrix}. \end{aligned} \tag{4.15}$$

The transformation matrix defined in (4.12) now functions as the crane–to–vessel rotation matrix (R_{vc}). As explained in 3.3, the kinematic chain includes a positional offset. Thus, (4.13) and (4.14) are expanded to:

$$p_{ti} = \begin{bmatrix} x_{ti} \\ y_{ti} \\ z_{ti} \end{bmatrix} = R_{iv} \left(\begin{bmatrix} c_x \\ c_y \\ c_z \end{bmatrix} + R_{vc} \begin{bmatrix} x_{tc} \\ y_{tc} \\ z_{tc} \end{bmatrix} \right), \tag{4.16}$$

$$p_{li} = \begin{bmatrix} x_{li} \\ y_{li} \\ z_{li} \end{bmatrix} = \begin{bmatrix} x_{ti} \\ y_{ti} \\ z_{ti} \end{bmatrix} + \ell R_{iv} R_{vc} \begin{bmatrix} -\sin \alpha \\ \cos \alpha \sin \beta \\ -\cos \alpha \cos \beta \end{bmatrix}, \tag{4.17}$$

where c_x, c_y, and c_z represent the positional offsets from the vessel's center of rotation to the crane base. Using the same symbolic computation script as for the static base, the expanded EoMs for a crane on a floating vessel can be derived. This gives (A.3) and (A.4).

4.2.4 Vessel-Mounted Crane with a Flexible Boom

To account for structural flexibility, the rigid-body dynamic model must be further expanded. Based on operational insights from Huisman on the crane type of interest, the most dominant structural deformation occurs as lateral boom deflection. Because these cranes show significantly higher stiffness in the luffing direction compared to the slewing direction, it is sufficient to capture the first mode of the boom. This can be done by modeling a virtual torsion spring acting purely in the slewing direction at the mast-boom joint, as was advised by Huisman.

This flexibility is incorporated by introducing a new generalized coordinate, ν_f, which represents the angle of deflection of the boom from its neutral position within the Crane Frame (Figure 4.1). With this addition, the kinematic position of the

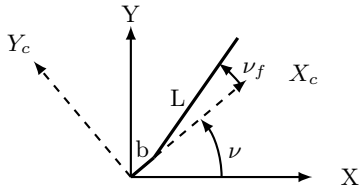


Figure 4.1: **Top view of the crane parts in the Crane Frame. ν_f is the angle of deflection from the neutral position, which aligns with X_c. b stays aligned with X_c.**

boom tip relative to the Crane Frame, originally defined in (4.11), is updated to:

$$p_{tc} = \begin{bmatrix} x_{tc} \\ y_{tc} \\ z_{tc} \end{bmatrix} = \begin{bmatrix} b + L\cos\mu\cos\nu_f \\ L\cos\mu\sin\nu_f \\ h + L\sin\mu \end{bmatrix}. \tag{4.18}$$

Due to this change, the coordinate transformations required to map the system states to the Inertial Frame in (4.13) must be updated accordingly.

To derive the EoMs for this expanded system using the Euler-Lagrange method, the system's energy equations must be modified to account for both the elastic deformation and the structural inertia. To capture the dynamic coupling between the structure and the load, the boom's flexible inertia is modeled as an equivalent point mass, m_b, located at the suspension point. The total kinetic energy T is computed as the sum of the kinetic energies of both masses:

$$T = \frac{1}{2}m_b\left(\dot{x}_{ti}^2 + \dot{y}_{ti}^2 + \dot{z}_{ti}^2\right) + \frac{1}{2}m\left(\dot{x}_{li}^2 + \dot{y}_{li}^2 + \dot{z}_{li}^2\right). \tag{4.19}$$

For the potential energy, the virtual torsion spring stores energy defined as $V_f = \frac{1}{2}k_f\nu_f^2$, where k_f is the torsional stiffness constant. Consequently, the total potential energy of the system becomes:

$$V = V_g + V_f = mgz_{li} + m_b g z_{ti} + \frac{1}{2}k_f\nu_f^2, \tag{4.20}$$

where V_g is the gravitational potential energy and V_f is the potential energy stored in the spring.

Furthermore, structural damping within the boom must be accounted for to prevent the modeled spring from oscillating infinitely. This non-conservative energy loss is mathematically modeled using the Rayleigh dissipation function, R, defined as:

$$R = \frac{1}{2}c_f\dot{\nu}_f^2, \tag{4.21}$$

where c_f is the torsional damping coefficient. The Euler-Lagrange equations are then expanded to include this dissipation term:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} + \frac{\partial R}{\partial \dot{q}_i} = 0. \tag{4.22}$$

Table 4.1: **The numerical values used in the simulations**

Parameter	m [kg]	m_b [kg]	L [m]	b [m]	h [m]	c_x [m]	c_y [m]	c_z [m]	k_f [Nm/rad]	c_f [Js/rad]
Value	1e6	2e6	150 m	10 m	40 m	50 m	-20 m	10 m	2e11	2e9

Rather than algebraically solving for each individual coordinate acceleration ($\ddot{\alpha}$, $\ddot{\beta}$, $\ddot{\nu}_f$) to obtain explicit analytical formulas, the EoMs are preserved in a compact matrix-vector form. Solving for these accelerations symbolically would require the analytical inversion of the heavily coupled, non-linear expressions describing the vessel motion and boom flexibility. This process introduces severe symbolic expression swell, yielding equations that are substantially more complex and computationally intensive to evaluate online. Instead, the system is factored into a standard mass matrix and forcing vector, with the structure of these terms detailed in Appendix A.3, where the inversion is done numerically during the simulation.

Based on crane data from Huisman, the values used in the simulations can be found in Table 4.1. To visualize the different EoMs or the impact of the values on the trajectory of the crane and lower-block, the Python script explained in Appendix D can be used.

4.3 State-Space Representation

As Appendix A shows, multiple time-varying variables are required to continuously calculate the sway accelerations $\ddot{\alpha}$ and $\ddot{\beta}$ and the flexible boom deflection acceleration $\ddot{\nu}_f$. To implement this in a Kalman filter framework, these variables must be mapped into a continuous-time state-space model of the form:

$$\dot{\mathbf{x}} = f(\mathbf{x}) + \mathbf{w}(t), \tag{4.23}$$

where $\mathbf{x}$ is the state vector, f is called the transition function, and $\mathbf{w}(t)$ is the zero-mean Gaussian process noise vector. The full state of the vessel–crane–load system is defined by a 21-dimensional state vector $\mathbf{x}$:

$$\mathbf{x} = \begin{bmatrix} \alpha & \dot{\alpha} & \beta & \dot{\beta} & \mu & \dot{\mu} & \ddot{\mu} & \nu & \dot{\nu} & \ddot{\nu} & \nu_f & \dot{\nu}_f & \ell & \dot{\ell} & \ddot{\ell} & p & \dot{p} & \ddot{p} & r & \dot{r} & \ddot{r} \end{bmatrix}^T. \tag{4.24}$$

This representation yields the continuous-time state transition function $f(\mathbf{x})$ formulated as:

$$\begin{aligned} \dot{\mathbf{x}} &= f(\mathbf{x}) + \mathbf{w}(t) \\ &= \begin{bmatrix} \dot{\alpha} & f_{\ddot{\alpha}}(\mathbf{x}) & \dot{\beta} & f_{\ddot{\beta}}(\mathbf{x}) & \dot{\mu} & \ddot{\mu} & 0 & \dot{\nu} & \ddot{\nu} & 0 & \dot{\nu}_f & f_{\ddot{\nu}_f}(\mathbf{x}) & \dot{\ell} & \ddot{\ell} & 0 & \dot{p} & \ddot{p} & 0 & \dot{r} & \ddot{r} & 0 \end{bmatrix}^T + \mathbf{w}(t), \end{aligned} \tag{4.25}$$

where $f_{\ddot{\alpha}}(\mathbf{x})$, $f_{\ddot{\beta}}(\mathbf{x})$, and $f_{\ddot{\nu}_f}(\mathbf{x})$ correspond to the derived non-linear EoMs.

The state vector in (4.24) combines the EoMs of the pendulum (α, β) and the flexible boom deflection (ν_f) with the driven kinematic states of the crane structure (μ, ν, ℓ) and the wave-induced vessel rotations (p, r). The acceleration states ($\ddot{\mu}$, $\ddot{\nu}$, $\ddot{\ell}$, $\ddot{p}$, $\ddot{r}$) are explicitly included in this vector because they are required to evaluate the EoMs within the transition function. Since these higher-order kinematic accelerations of

the crane structure and the vessel hull are not directly measured by the sensors, they must be estimated online by the filter to ensure accurate dynamic tracking during active maneuvering and wave excitation. The time derivatives of these accelerations (the system jerks) are set to zero because there is no analytical model defined to deterministically predict how they change over time. Instead, these variations are modeled as stochastic processes driven entirely by the continuous-time process noise vector $\mathbf{w}(t)$. Defining the process noise at this higher derivative level prevents abrupt, step-like changes in the estimated accelerations. This ensures continuous acceleration profiles over time, which results in smoother velocity and displacement estimates for the system. This also directly mirrors the physical reality of the heavy crane, where structural inertia prevents instantaneous changes in acceleration.

4.4 Discretization

Because the EKF and UKF operate as discrete-time algorithms, the continuous model $f(\mathbf{x})$ must be discretized. For real-time applications, the integration algorithm must have a strictly deterministic execution time to guarantee that calculations finish within the available time. Therefore, the 4th-order Runge-Kutta (RK4) method is employed. Unlike complex solvers that use variable step sizes or iterative loops, RK4 is a commonly used, explicit algorithm that executes exactly four fixed mathematical evaluations of the transition function per time step [45]. This ensures deterministic execution times on embedded hardware. By achieving high-accuracy predictions for highly non-linear swinging motions at a strictly bounded computational cost, the RK4 algorithm provides a good balance between mathematical accuracy and real-time execution constraints [45].

4.5 Measurement Models

In addition to the state transition model $f(\mathbf{x})$, the estimators require measurement models. These models provide explicit mathematical mappings that translate the internal state vector $\mathbf{x}$ into the predicted sensor outputs, since not all sensors measure states directly.

As the selected physical sensors operate at independent data rates (detailed in Section 4.5.1), the Simulink implementation utilizes a multi-rate architecture (Section 4.6.3). Therefore, instead of a single measurement function, three independent measurement models are defined. During execution, the estimator dynamically applies the specific measurement function only when new data from that respective sensor arrives.

Boom-Mounted LiDAR The LiDAR does not directly measure the sway angles (α and β). Instead, it measures the 3D Cartesian coordinates (x, y, and z) of the lower-block relative to the suspension point. To map the state vector to the LiDAR's output, the non-linear Forward Kinematics relation defined in (4.1) is applied directly:

$$h_{\text{LiDAR}}(\mathbf{x}) = \ell \begin{bmatrix} -\sin\alpha \\ \cos\alpha \sin\beta \\ -\cos\alpha\cos\beta \end{bmatrix}. \tag{4.26}$$

Crane Joint Encoders The rotary encoders directly measure the luffing and slewing angles of the crane. Therefore, their measurement model is a direct linear mapping of the respective state variables:

$$h_{\text{Encoders}}(\mathbf{x}) = \begin{bmatrix} \mu \\ \nu \end{bmatrix}. \tag{4.27}$$

MRU Similarly, the MRU directly measures the vessel's pitch, roll, and their respective rates, resulting in the following linear measurement model:

$$h_{\text{MRU}}(\mathbf{x}) = \begin{bmatrix} p \\ \dot{p} \\ r \\ \dot{r} \end{bmatrix}. \tag{4.28}$$

4.5.1 Sensor Characteristics

To evaluate the estimator in a realistic simulation environment, the measurement models are parametrized using the specifications of commercially available sensors typically used by Huisman. These specifications define the independent update rates and the measurement noise profiles used in the filter design.

Boom-Mounted LiDAR To track the relative position of the lower-block, a robust 3D LiDAR suited for marine environments is selected: the Ouster OS1 MAX [34]. This sensor provides point-cloud data at an update rate of 5–40 Hz, depending on the selected resolution. For this thesis, 20 Hz is selected, as this provides a good balance between resolution and update rate. Although this rate is slightly lower than the 25 Hz target derived in Section 3.1.3, the multi-rate filter architecture compensates for this difference, as detailed in Section 4.6.3. Its positional measurement noise has a standard deviation of $\sigma_{\text{LiDAR}} = 1.5\,\text{cm}$, where it is assumed that after processing the point cloud, the extracted x-, y-, and z-coordinates share the same measurement noise.

Crane Joint Encoders The slewing and luffing angles of the crane are measured using standard industrial absolute rotary encoders. These sensors typically provide high-frequency updates. For this thesis, the update rate for the joint encoders is set at 100 Hz. The mechanical backlash and structural play in the crane joints dominate the measurement uncertainty compared to the electrical noise and angular quantization of the sensors themselves. Therefore, an effective standard deviation of $\sigma_{\text{Encoders}} = 0.1^\circ$ is used, based on internal data at Huisman.

MRU Vessel kinematics are captured using a marine-grade MRU, such as the Kongsberg MRU 5 [23]. This unit outputs pitch, roll, and their respective angular rates at a frequency of 200 Hz. The noise profile for the angular position is $\sigma_{\text{MRU\_pos}} = 0.02^\circ$, and the angular rate noise is $\sigma_{\text{MRU\_vel}} = 0.025^\circ/\text{s}$.

4.6 Filter Implementation

In Simulink, the Control System Toolbox contains EKF and UKF blocks, removing the need to manually implement the algorithms. To use these blocks, some

configuration has to be done, which is explained in this section.

4.6.1 Non-Linearity Handling

As briefly stated in Section 4.1, the EKF linearizes the system around the current state estimate, allowing the standard KF equations to be applied to a locally linear approximation. In Simulink, this linearization step can either be performed by supplying the Jacobian matrices and evaluating the partial derivatives in real time during the simulation, or by letting the EKF block approximate the Jacobians numerically. In this work, the Jacobians required for the EKF are computed numerically using the functionality provided by the Simulink block. Given the high dimensionality of the state vector and the nonlinear nature of the coupled vessel–pendulum dynamics, deriving and maintaining analytical Jacobians within the Simulink environment would introduce significant implementation complexity. The numerical approximation ensures consistency with the implemented process and measurement models, while remaining sufficiently accurate at the selected update rate.

The UKF, on the other hand, selects a minimal set of sample points, called sigma points, around the current estimate. These sigma points are propagated through the nonlinear model, after which the predicted state and uncertainty are reconstructed from the transformed points. When a new measurement becomes available, the sigma points are passed through the measurement model to obtain predicted measurements, which are then compared to the actual measurements to correct the state estimate. The default sigma point selection parameters provided by the Simulink UKF block are used. These parameters define the spread and weighting of the sigma points to accurately capture the mean and covariance of the transformed states. Although alternative parameterizations were tested, these standard values yielded the best results.

4.6.2 Covariance Matrices and Tuning

Kalman filters are tuned through the selection of covariance matrices, which represent the relative confidence in the system model versus the measurements. The process noise covariance matrix, Q, describes the uncertainty in the dynamic model, including unmodeled vessel motion and external disturbances. The measurement noise covariance matrix, R, represents the uncertainty associated with the sensors. By adjusting these matrices, the estimator is tuned to rely more heavily on either the model-based predictions or the incoming sensor data.

State Transition Covariance Matrix (Q) Because the state vector contains kinematic derivatives (velocity and acceleration), the process noise must be scaled correctly across the time step T_s. This is achieved using kinematic tracking models [26], which assume that unmodeled disturbances act as white noise on the highest-order derivative of each state. This noise is then propagated to the lower derivatives through integration.

For the non-linear pendulum sway states α and β and the boom deflection angle ν_f, the highest modeled derivative is acceleration. This is formally represented using

the Discrete White Noise Acceleration (DWNA) model [26]. The unmodeled disturbance is treated as a continuous acceleration noise (σ_{a}), which, when integrated over the sample time, yields the following block-diagonal entries for the sway angles and velocities:

$$Q_{\alpha,\beta,\nu_f} = \mathrm{diag}\left(\left(\frac{1}{2}\sigma_{\mathrm{a}}T_s^2\right)^2, (\sigma_{\mathrm{a}}T_s)^2\right). \tag{4.29}$$

Conversely, the crane kinematics (μ, ν, ℓ) and vessel rotations (p, r) include acceleration as explicit states in the state vector. Therefore, the Discrete White Noise Jerk (DWNJ) model is applied. Here, the unknown disturbance acts on the jerk (σ_j). Integrating this jerk yields the noise covariance for position, velocity, and acceleration:

$$Q_{\mathrm{kinematic}} = \mathrm{diag}\left(\left(\frac{1}{6}\sigma_j T_s^3\right)^2, \left(\frac{1}{2}\sigma_j T_s^2\right)^2, (\gamma\sigma_j T_s)^2\right). \tag{4.30}$$

The base standard deviations (σ) were derived from the physical operational limits of the system. The pendulum acceleration noise was configured to $\sigma_a = 10^{-4}\,\mathrm{rad/s^2}$ and boom deflection noise to $\sigma_a = 0.003\,\mathrm{rad/s^2}$. For the vessel, the theoretical maximum jerk was calculated by taking the derivative of the worst-case pitch and roll amplitude (2°) over the minimum considered wave period (4 s), yielding approximately $0.14\,\mathrm{rad/s^3}$. This was conservatively increased in the implementation to $\sigma_{\mathrm{pr\_jerk}} = 0.5\,\mathrm{rad/s^3}$ to ensure the filter remains responsive to unpredictable, harsher wave impacts.

For the crane joints, the jerk noise was approximated by taking the operational maximum acceleration limits ($a_{\max}$) and normalizing them over the sample time, formulated as $\sigma_j = 0.5 \cdot a_{\max}/T_s$. The defined acceleration limits were $a_\mu = 0.0005\,\mathrm{rad/s^2}$, $a_\nu = 0.01\,\mathrm{rad/s^2}$, and $a_\ell = 0.1125\,\mathrm{m/s^2}$, which were based on actual crane data provided by Huisman.

Finally, during simulation tuning, an empirical scaling factor (γ) was introduced to the acceleration covariance blocks. Setting $\gamma = 0.1$ for the crane joints significantly improved filter stability by preventing the estimator from over-reacting to numerical noise in the unmeasured acceleration states. This aligns physically with the massive structural inertia of the crane, which naturally damps high-frequency acceleration transients. For the vessel and pendulum states, a scaling factor of $\gamma = 1.0$ was maintained. The complete 21×21 diagonal Q matrix is formed by concatenating these discrete blocks.

Measurement Covariance Matrices (R) For the measurement model, a diagonal matrix is used to define the noise covariance for each sensor, assuming that the individual measurement axes are uncorrelated. The diagonal entries correspond to the measurement variances, populated by converting the sensor standard deviations (defined in Section 4.5.1) into standard SI base units (meters and radians). During this conversion, a conservative safety margin was applied to the values to account for unmodeled mechanical vibrations and hardware quantization. This results in the following explicit matrices:

$$R_{\mathrm{LiDAR}} = \mathrm{diag}\left(\left(1.5 \cdot 10^{-2}\right)^2, \left(1.5 \cdot 10^{-2}\right)^2, \left(1.5 \cdot 10^{-2}\right)^2\right), \tag{4.31}$$

$$R_{\text{Encoders}} = \text{diag}\left(\left(2 \cdot 10^{-3}\right)^2, \left(2 \cdot 10^{-3}\right)^2\right), \tag{4.32}$$

$$R_{\text{MRU}} = \text{diag}\left(\left(4 \cdot 10^{-4}\right)^2, \left(5 \cdot 10^{-4}\right)^2, \left(4 \cdot 10^{-4}\right)^2, \left(5 \cdot 10^{-4}\right)^2\right). \tag{4.33}$$

To ensure a rigorous comparative evaluation between the EKF and UKF, identical Q and R matrices are utilized for both algorithms. This guarantees a fair baseline comparison, ensuring that any observed performance differences are strictly attributable to how each algorithm handles the non-linear system dynamics, rather than discrepancies in filter tuning.

4.6.3 Multi-Rate Architecture

An asynchronous multi-rate software execution scheme is required due to the significant sampling frequency differences among the sensors. Utilizing a conventional single-rate filter structure would either require artificially downsampling the high-frequency vessel data (200 Hz) or oversampling the LiDAR channel (20 Hz). This trade-off either removes usable high-fidelity data or wastes processing cycles computing redundant corrections on duplicate data.

To handle these asynchronous data streams without introducing custom scheduling overhead, the state estimation architecture utilizes the native multi-rate execution features of the Simulink filtering blocks. Within the discrete-time execution engine of these blocks, the state transition (prediction) step is configured to run continuously at a base loop frequency of 200 Hz, while the measurement correction tasks are triggered conditionally on event-driven task boundaries as new sensor data become available. This setup introduces a strict structural constraint: to ensure exact data synchronization within Simulink's discrete-time scheduling framework, the base prediction rate must be chosen as an exact integer multiple of all incoming sensor update rates. Since the selected hardware interfaces operate at 200 Hz, 100 Hz, and 20 Hz, the 200 Hz base rate satisfies this alignment requirement perfectly. This block configuration provides clean state estimates at a continuous 200 Hz, seamlessly exceeding the 25 Hz anti-sway controller requirement.

Fast prediction combined with slower measurement updates leads to increased covariance growth between corrections, as process noise accumulates over time. However, due to the geometric sensitivity of the crane system, small variations in pitch and roll can result in significant displacement of the suspension point. This justifies the use of a 200 Hz prediction rate, ensuring vessel kinematics are tracked continuously while accepting the naturally slower measurement updates of the 20 Hz LiDAR.

Chapter 5

Simulation Framework and Performance Metrics

To evaluate the performance of the model-based state estimators under realistic offshore operating conditions, a simulation framework is developed in MATLAB/Simulink. The framework simulates the coupled non-linear system dynamics, emulates realistic sensor measurements, executes the estimation algorithms, and logs performance metrics. This chapter details the architectural setup of this environment, defines the operational evaluation scenarios, and establishes the metrics used to quantify estimation accuracy.

5.1 Simulation Architecture

To guarantee an unbiased evaluation, the simulation framework decouples the physical system simulation from the digital state estimation software. As illustrated by the functional pipeline in Figure 5.1, the architecture is organized into four processing stages that replicate the hardware-software boundary of a physical crane deployment: ground truth data generation, kinematic mapping, sensor emulation, and the state estimation pipeline. The low-level Simulink block diagrams and subsystem routing interfaces of this architecture are detailed in Appendix C.

5.1.1 Ground Truth Data Generation

To evaluate estimator accuracy, an absolute reference baseline, the ground truth, is required. The physical system is simulated by continuously evaluating the full, coupled non-linear EoMs derived for the vessel, crane, and flexible boom. Within this subsystem, the continuous-time state derivatives are integrated numerically to propagate the true state vector over time.

Unlike the state estimators, which do not have access to the true acceleration inputs and must stochastically model unmeasured accelerations via process noise, the ground truth simulation is driven entirely deterministically. This means that input trajectories are fully prescribed as time-varying functions. Specifically, predefined acceleration profiles are fed directly as inputs into the non-linear EoMs. The simulation engine numerically integrates these accelerations forward in time to compute

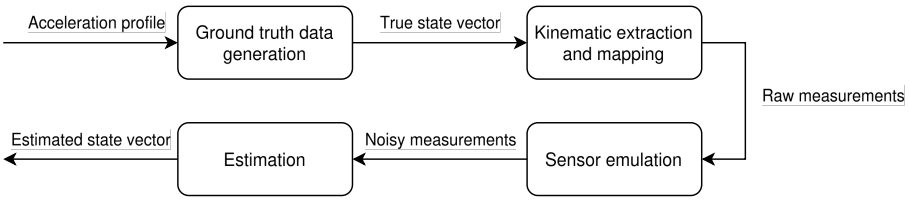


Figure 5.1: **Functional block diagram of the simulation framework, illustrating the data pipeline from deterministic state generation to digital state estimation.**

the true, continuous velocity and displacement states of the coupled vessel–crane system.

5.1.2 Kinematic Extraction and Mapping

This step extracts the raw crane joint states (slewing, luffing, and hoisting) along with the vessel states (pitch, roll, and their respective angular rates) and routes them directly to the sensor emulation subsystem. Crucially, it evaluates the forward kinematics to calculate the time-varying relative position vector between the boom tip and the lower-block. This vector represents the true physical trajectory of the load relative to the crane's LiDAR mounting location, providing the exact geometric baseline required to emulate the digital LiDAR sensor measurements.

5.1.3 Sensor Emulation

To approximate the operational environment, the true extracted data generated by the physical model are modified to mimic the finite resolution, sampling frequency, and measurement uncertainty of the actual instrumentation. Following the standard measurement model employed in Kalman filtering and sensor fusion, sensor errors are represented as additive zero-mean white Gaussian noise [45], with the noise power scaled to match the specific standard deviation and update rate of each sensor [19].

5.1.4 Estimator Pipeline

The discretized, multi-rate sensor streams are routed into the state estimation pipeline. For this simulation, the EKF and UKF blocks from the MATLAB Control System Toolbox are utilized. These blocks are parameterized with the covariance structures, dynamical model, and fixed-step solver among other configuration parameters described in Chapter 4. Operating independently from the ground truth physical model, the pipeline uses only the noisy, sampled sensor data to estimate the state of the crane.

5.2 Evaluation Scenarios

To evaluate the tracking capabilities of the estimators, four distinct operational scenarios were designed. These scenarios are structured to isolate specific operational situations, allowing for a clear analysis of how different system dynamics

influence the overall estimation accuracy, ranging from static lifts to multi-axis maneuvers under active wave disturbances. In accordance with the modeling assumptions from Section 3.1.4, wave-induced vessel motions are restricted strictly to angular pitch and roll variations, while surge, sway, heave, and yaw are ignored across all evaluated cases. To visualize the scenarios, the Python script from Appendix D can be used.

5.2.1 Scenario 1: Nominal Ship-to-Ship Transfer

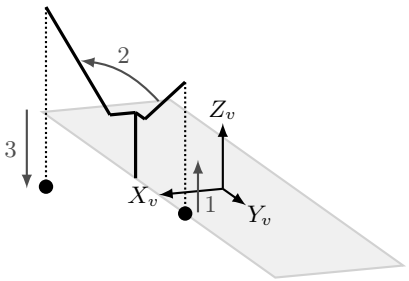
The first scenario establishes a baseline under ideal environmental conditions, assuming a static vessel with no wave- or crane-induced pitch or roll. The maneuver replicates a standard ship-to-ship lifting operation executed over a total simulation duration of 700 s. The lower-block starts at rest on the deck with an initial slewing angle of 90° and a luffing angle of 30°. The operation is split into three sequential phases: a vertical hoisting phase, a simultaneous slewing and luffing phase accelerating to 0° and 60°, respectively, and a final vertical lowering phase, as visualized in Figure 5.2(a). Because the slewing target is reached before the luffing target, this sequence provides step-changes in the acceleration profiles. This allows for the clear observation of baseline kinematic convergence and isolates how the estimators handle discrete transitions in joint movement without external disturbances.

5.2.2 Scenario 2: Environmental Disturbance Rejection

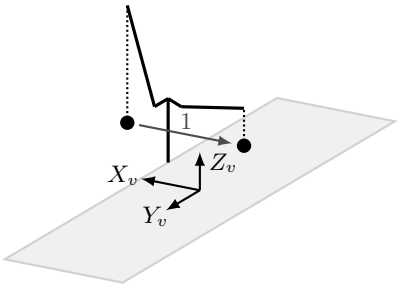
To isolate the impact of wave-induced vessel motions, this scenario executes the exact crane trajectory defined in Scenario 1 over an identical 700 s duration while introducing active vessel dynamics. The vessel motion is parameterized via continuous sinusoidal pitch and roll disturbances with an amplitude of 1° and a characteristic period of 8 s, representing a typical offshore operational baseline specified by Huisman. By maintaining crane kinematics perfectly identical to the baseline scenario, any subsequent degradation in state estimation accuracy can be directly attributed to the non-linear coordinate transformations required by the filters to decouple vessel rotations from the true lower-block states.

5.2.3 Scenario 3: Coordinated Continuous Path

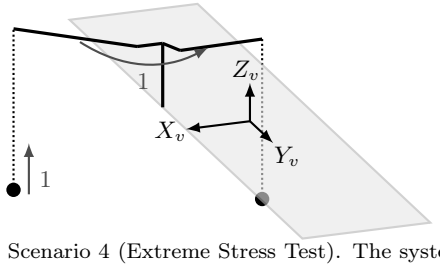
The third scenario evaluates the estimators during a continuous tracking maneuver under the same active wave conditions. Starting at a height of 50 m above the vessel's center of rotation on the starboard side, the crane's joint velocities are continuously modulated to sweep the load across the deck to the port side (see Figure 5.2(b)), taking 650 s to complete. This is a common operation encountered during the process of upending a monopile. Because the crane operates via rotational joints, traversing a linear path across the vessel deck requires highly non-linear coordination between the slewing, luffing, and hoisting kinematics. This continuous sweeping movement results in constantly changing accelerations that inherently excite the unactuated pendulum dynamics of the lower-block, testing the estimator's ability to maintain tracking accuracy when the joint coordinates are highly coupled and constantly shifting.



(a) Crane movements for Scenario 1 (Nominal Ship-to-Ship Transfer) and Scenario 2 (Environmental Disturbance Rejection). The maneuver executes three sequential phases: hoisting of the lower-block (1), simultaneous slewing and luffing movements over the water (2), and a final vertical lowering phase to the target deck (3).



(b) Crane movements for Scenario 3 (Coordinated Continuous Path). The load is commanded to follow a straight-line from the starboard to the port side at a constant height above the deck (1).



(c) Crane movements for Scenario 4 (Extreme Stress Test). The system is initialized at its maximum physical reach envelope. Simultaneous maximum-velocity commands are applied across all crane actuators to aggressively excite structural boom flexibility under active wave conditions (1).

Figure 5.2: **Crane paths for the evaluated simulation scenarios. Trajectories are represented within the reference Vessel Frame (X_v, Y_v, Z_v), with active wave-induced vessel motions excluded for clarity.**

5.2.4 Scenario 4: Extreme Stress Test

The final scenario serves as an unrealistic stress test designed to evaluate the mathematical limits of the estimation frameworks over a condensed 100 s window. The system initializes at an extended operational envelope, featuring an initial 200 m hoisting length, a 0° slewing angle, and a 15° luffing angle, positioning the load over 60 m below deck level. From this structurally sensitive state, simultaneous maximum-velocity commands are applied to all actuators under wave conditions with an increased amplitude of 2° and an 8 s period. Doubling the wave amplitude relative to the baseline intentionally scales the vessel's angular velocities and linear accelerations at the boom tip, increasing the inertial stress on the system dynamics. The joints are driven aggressively toward target states of 110° slewing, 20° luffing, and a 180 m wire rope length, as depicted in Figure 5.2(c). This combination yields the highest possible structural accelerations and lateral velocities, aggressively exciting the structural flexibility of the crane boom. This case is specifically designed to observe potential linearization breakdown, directly comparing the performance limits of the EKF's first-order approximation against the UKF's

sigma-point sampling.

5.3 Performance Metrics

To quantify and compare the performance of the EKF and UKF frameworks against the derived requirements, statistical metrics are computed by comparing the state estimates against the ground truth.

To evaluate tracking precision and capture steady-state offsets caused by unmodeled boom deflections or filtering lag, the Root Mean Square Error (RMSE) is utilized. The RMSE is computed independently for translational position and velocity across the global axes (X, Y, Z) in the Inertial Frame:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(s_{\text{true}}(k) - \hat{s}(k)\right)^2}, \tag{5.1}$$

where N is the total number of discrete simulation steps, s_{true} is the true ground truth state of the lower-block (position or velocity along the Inertial Frame axes), and $\hat{s}$ is the estimated state of the lower-block.

While the RMSE provides an aggregate measure of tracking accuracy over a full operational profile, it can obscure short-duration error spikes. To capture the worst-case tracking performance, the Maximum Absolute Error (MaxAE) is also evaluated for the position and velocity on a per-axis basis for each scenario:

$$\text{MaxAE} = \max_{k} |s_{\text{true}}(k) - \hat{s}(k)|. \tag{5.2}$$

Tracking the worst-case transient error is essential because standard average metrics completely smooth over high-frequency tracking spikes caused by sudden wave impacts or structural decelerations. Isolating these peak deviations ensures that the filter does not introduce brief, localized tracking divergences that could otherwise feed corrupted state estimates into the anti-sway automation pipeline and destabilize the control system.

The formal tracking requirements established for this system are evaluated strictly against the RMSE metrics rather than the MaxAE bounds. While the RMSE measures the continuous statistical standard deviation of the tracking error, the MaxAE captures isolated, worst-case transient spikes driven by extreme maneuvering impulses. Consequently, the MaxAE is treated as an auxiliary diagnostic tool for safety margins rather than a rigid threshold for baseline filter validation.

Chapter 6

Results

This chapter presents and evaluates the tracking performance and computational efficiency of the state estimation frameworks. First, the baseline difference between the rigid boom model and the flexible boom model is analyzed to establish how the true system trajectory is generated. Next, the tracking accuracy of the EKF and the UKF is demonstrated and compared across a variety of operational scenarios and estimator configurations. Finally, the runtime performance of the estimators is evaluated to assess their real-time execution viability.

6.1 Structural Model Difference

Before evaluating the filters, the divergence between the rigid and flexible boom model is quantified. This comparison shows the tracking errors that are introduced purely by omitting structural flexibility from the equations, before any sensor data or filtering is applied to correct the state variables.

Figure 6.1 illustrates the position divergence between the rigid and flexible models for both the boom tip and the lower-block during Scenario 2 (Disturbance Rejection). As shown by the blue line, the structural flexibility introduces an oscillation at the boom tip, resulting in a maximum position difference of 0.91 m between the rigid model and the flexible model. These oscillations correspond to the 8 s period of the waves acting on the vessel structure. Because the steel boom has a fixed stiffness, this tip error remains within a bounded envelope throughout the simulation.

Notably, the distinct peak in the boom tip distance just after the 100 s mark, which is also the maximum peak, directly corresponds to the deceleration of the crane's slewing motion. The sudden reduction in rotational velocity creates a large inertial force that deflects the boom structure. This structural deflection acts as a mechanical impulse on the load suspension point. As shown by the orange line, the lower-block position error is relatively small until this event occurs. Following the deceleration, the impulse causes the lower-block trajectory in the flexible model to drift out of phase with the rigid model's prediction. This timing shift accumulates over the remainder of the scenario, eventually creating a large position error of 7.46 m. Because wave-induced vessel motions continuously excite the pendulum, this difference is likely to be sustained.

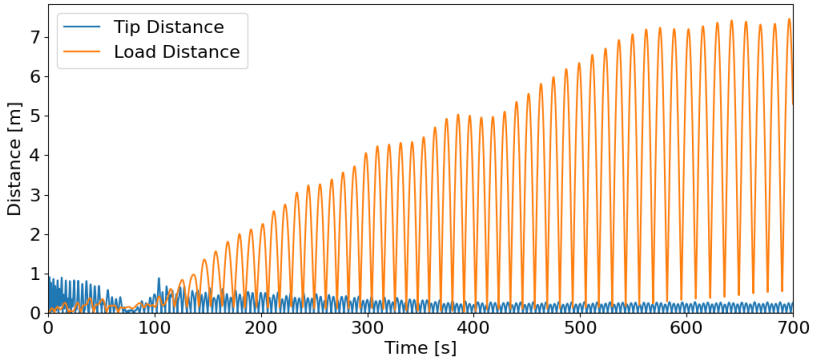


Figure 6.1: **Distance difference between the rigid model and the flexible model under wave disturbances (Scenario 2: Disturbance Rejection). The boom tip difference remains within a steady boundary (maximum of 0.91 m), while the unactuated lower-block position error accumulates over time (maximum of 7.46 m) due to a slewing deceleration impulse at 100 s. Continuous wave-induced vessel motions excite the load dynamics, preventing the difference from decreasing.**

This analysis shows that assuming a completely rigid crane causes severe tracking degradation during active maneuvers. To ensure the evaluation is realistic, the flexible model is used exclusively to generate the ground truth trajectory. The following sections will evaluate how effectively the EKF and UKF estimators can use noisy sensor measurements to estimate the state of the lower-block.

6.2 Performance Metrics

To evaluate the tracking performance of the state estimation pipeline, four distinct configurations are compared for each scenario. Both the EKF and UKF algorithms are evaluated using two different internal dynamical models: the simplified rigid model and the flexible model. Utilizing both formulations allows for a direct assessment of the trade-off between tracking accuracy and computational complexity, since the rigid model is simpler compared to the flexible model. Because the true initial states of the crane are unknown to the estimators at startup, the first five seconds of each simulation run are excluded from the calculated performance metrics. This initialization window grants the filters sufficient time to converge from their initial values to a stable, realistic state estimate.

6.2.1 Impact of the Internal Model Configuration

The most prominent trend across all four operational scenarios is the severe performance degradation that occurs when the estimators rely on a rigid internal model. When the true system experiences structural flexibility under wave disturbances, omitting these dynamics from the estimator equations leads to large tracking errors. As demonstrated in Table 6.1, this effect is especially visible in the horizontal plane, which aligns with the primary directions of load sway. For example, in Scenario 2

Table 6.1: **Comparison of tracking performance across all scenarios evaluated via Root Mean Square Error (RMSE).To ensure alignment with the performance criteria, tracking errors are evaluated individually for each axis. The data contrast the EKF and UKF implementations under rigid and flexible internal model configurations. The best-performing configuration for each metric is highlighted in bold.**

	Configuration		**Position Error [cm]**			**Velocity Error [cm/s]**		
Scenario	Model	Est.	x	y	z	$\dot{x}$	$\dot{y}$	$\dot{z}$
Scenario 1: Nominal Transfer	Rigid	EKF	3.048	4.262	1.769	1.154	1.654	3.016
		UKF	3.052	4.264	1.783	1.155	1.655	3.017
	Flexible	**EKF**	**1.811**	**2.001**	**1.667**	**0.633**	**1.049**	**2.883**
		UKF	1.881	2.212	1.707	0.757	1.375	2.885
Scenario 2: Disturbance Rejection	Rigid	EKF	15.03	29.18	3.934	7.704	14.71	4.516
		UKF	15.03	29.18	3.945	7.704	14.71	4.516
	Flexible	**EKF**	**1.832**	**1.996**	**1.690**	**0.640**	**1.067**	**2.951**
		UKF	1.912	2.207	1.739	0.764	1.388	2.954
Scenario 3: Coordinated Path	Rigid	EKF	12.56	4.524	1.863	6.461	2.212	2.742
		UKF	12.56	4.524	1.864	6.461	2.212	2.742
	Flexible	**EKF**	**2.005**	**2.093**	**1.817**	**1.007**	**0.843**	**2.683**
		UKF	2.178	2.182	1.819	1.294	1.018	2.683
Scenario 4: Stress Test	Rigid	EKF	39.83	45.39	7.854	20.41	23.49	12.15
		UKF	39.83	45.39	7.859	20.41	23.49	12.15
	Flexible	**EKF**	**2.292**	**2.246**	**3.576**	**1.251**	**1.268**	**5.517**
		UKF	2.497	2.485	3.581	1.559	1.586	5.513

(Disturbance Rejection), switching from a flexible internal model to a rigid internal model causes the y-axis error to increase from around 2 cm to 29 cm. A similar degradation is observed in Scenario 4 (Stress Test), where the rigid configurations yield position errors as high as around 40 cm and 45 cm in the x and y axes, respectively. The vertical tracking performance (z-axis) is also negatively affected by the rigid assumption, though the absolute error growth is smaller since the boom structure undergoes no vertical deflection in this model.

6.2.2 Comparison of Estimators Under the Rigid Model

When evaluating the rigid model configurations in Table 6.1 and Table 6.2, the choice of filtering algorithm has almost no impact on tracking accuracy. The RMSE values for the EKF and UKF are virtually identical across all scenarios. The MaxAE metrics in Table 6.2 follow the exact same trend, showing an average difference of less than 0.2% between the two algorithms when restricted to a rigid internal model. This uniform behavior indicates that the errors introduced by the unmodeled structural flexibility completely dominate the filtering process, when an estimator is forced to operate with a severe model mismatch. As a result, the mathematical differences between the first-order linearization of the EKF and the sigma-point propagation of the UKF become irrelevant to the final tracking output.

Table 6.2: **Comparison of tracking performance across all scenarios evaluated via Maximum Absolute Error (MaxAE). The data contrast the EKF and UKF implementations under rigid and flexible internal model configurations. The best-performing configuration for each metric is highlighted in bold.**

	Configuration		**Position Error [cm]**			**Velocity Error [cm/s]**		
Scenario	Model	Est.	x	y	z	$\dot{x}$	$\dot{y}$	$\dot{z}$
Scenario 1: Nominal Transfer	Rigid	EKF	38.91	31.26	8.637	16.44	15.63	17.86
		UKF	38.91	31.25	8.760	16.44	15.63	17.80
	Flexible	**EKF**	**8.541**	**9.659**	**7.584**	**5.650**	**5.538**	**15.34**
		UKF	10.52	11.39	7.657	7.697	7.794	15.20
Scenario 2: Disturbance Rejection	Rigid	EKF	91.76	100.3	15.88	44.49	47.40	28.94
		UKF	91.76	100.3	15.96	44.49	47.40	28.90
	Flexible	**EKF**	**8.962**	**9.467**	**7.977**	**5.914**	**5.429**	**14.96**
		UKF	10.83	11.21	8.412	7.755	7.716	14.80
Scenario 3: Coordinated Path	Rigid	EKF	39.95	14.37	8.056	20.20	6.503	13.78
		UKF	39.96	14.34	8.017	20.20	6.480	13.76
	Flexible	**EKF**	**8.363**	**8.786**	**8.125**	**4.050**	**4.105**	**12.85**
		UKF	8.915	9.449	8.004	5.273	5.076	12.75
Scenario 4: Stress Test	Rigid	EKF	106.3	113.9	35.32	47.33	58.40	57.59
		UKF	106.3	113.9	35.34	47.33	58.40	57.61
	Flexible	**EKF**	**9.873**	**9.007**	**12.37**	**5.170**	**5.975**	**23.28**
		UKF	11.11	11.77	12.28	7.061	8.565	23.19

6.2.3 Comparison of Estimators Under the Flexible Model

When the estimators are configured with the flexible internal model, the unmodeled structural errors are removed. Under these conditions, a consistent trend emerges where the EKF slightly, but systematically outperforms the UKF in both average and peak tracking precision. Looking at the position RMSE in Table 6.1, the EKF achieves a lower tracking error across almost all coordinates and scenarios, outperforming the UKF by an average of 5.2% in position error. The largest difference occurs in the horizontal velocity states ($\dot{x}$ and $\dot{y}$), where the UKF shows up to 10% higher estimation errors during active movements.

This performance gap is even more pronounced when analyzing the peak transient deviations via the MaxAE metric in Table 6.2. When using the flexible model, the EKF reduces peak horizontal position errors by 14.3% on average compared to the UKF. For instance, for Scenario 4, the EKF limits the maximum y-axis deviation to 9.007 cm, whereas the UKF allows a peak deviation of 11.77 cm (a 30.7% increase in peak error). This indicates that the EKF maintains a tighter tracking envelope during sudden transitions and maximum velocity phases.

Crucially, the flexible internal model makes tracking performance independent of environmental state. While rigid errors tripled under wave action, the flexible filters keep horizontal MaxAE under 9.5 cm in Scenario 2 (Disturbance Rejection) and res-

Table 6.3: **Summary of minimum, mean, and maximum RMSE and MaxAE metrics aggregated across all scenarios. The data contrast the EKF and UKF estimation frameworks restricted to the flexible internal model configuration.**

		Position Error [cm]			**Velocity Error [cm/s]**		
Est.	Metric	min	mean	max	min	mean	max
EKF	RMSE	1.667	2.085	3.576	0.663	1.816	5.517
	MaxAE	7.584	9.059	12.37	4.050	9.021	23.28
UKF	RMSE	1.707	2.200	3.581	0.757	1.981	5.513
	MaxAE	7.657	10.13	12.28	5.076	10.24	23.19

ist error expansion during the coupled maneuvers of Scenario 3 (Continuous Path). Even when pushed to its physical limits in Scenario 4 (Stress Test), horizontal errors remain below 12 cm, with minor vertical deviations isolated to 12.37 cm for z and 23.28 cm/s for $\dot{z}$.

Finally, Table 6.3 condenses these tracking boundaries specifically for the flexible model configurations. The summary confirms that both filters successfully satisfy the operational design criteria, maintaining the RMSE well within the target requirements of 5–10 cm for position and 5–10 cm/s for velocity. Within this envelope, the EKF systematically holds the lower boundary for both average tracking precision and peak transient errors across all evaluated operational profiles.

6.3 Runtime Performance

To evaluate the computational efficiency of the EKF and UKF frameworks, execution times were profiled to isolate the processing cost of the algorithms. During profiling, the total runtime of the simulation was recorded. Subsequently, the simulation was run again with the estimation pipeline deactivated to capture the runtime of the overhead. Subtracting these leaves the execution duration consumed strictly by the state estimation.

The simulations were executed on an HP ZBook Studio G5 equipped with an Intel Core i7-9750H processor. When deployed on an actual vessel, different hardware and production-optimized code implementations will be used. Consequently, these measurements serve as a comparative indication of computational effort rather than absolute benchmarks.

Table 6.4 summarizes the performance profiles. As expected, the execution time for the true system dynamics remains highly consistent across configurations within the same scenario, varying by only a few tenths of a second. This consistency confirms that changing the internal filter configuration does not alter the underlying simulation environment.

Analyzing the isolated estimation times reveals two distinct trends. First, utilizing a simplified rigid internal model reduces estimation computation time by approxim-

Table 6.4: **Computational execution profiling and normalized runtime ratios for the state estimators, averaged over five independent execution runs. The normalized time represents the execution time expressed as a percentage of the physical simulated duration, where values below 100% indicate faster-than-real-time execution.**

Scenario	Configuration		Total Time [s]	Overhead Time [s]	Estimation Time [s]	Normalized Time [s/s]
	Model	Est.				
Scenario 1: Nominal Transfer	Rigid	EKF	28.93	10.49	18.44	2.6%
		UKF	32.62	10.44	22.17	3.2%
	Flexible	EKF	36.57	10.13	26.45	3.8%
		UKF	36.88	10.73	26.15	3.7%
Scenario 2: Disturbance Rejection	Rigid	EKF	29.42	10.69	18.72	2.7%
		UKF	30.71	10.80	19.90	2.8%
	Flexible	EKF	34.14	10.91	23.23	3.3%
		UKF	38.40	10.65	27.75	4.0%
Scenario 3: Coordinated Path	Rigid	EKF	27.37	9.624	17.75	2.7%
		UKF	28.51	9.902	18.60	2.9%
	Flexible	EKF	31.67	9.755	21.91	3.4%
		UKF	33.78	9.736	24.05	3.7%
Scenario 4: Stress Test	Rigid	EKF	4.327	1.891	2.436	2.4%
		UKF	4.565	1.881	2.684	2.7%
	Flexible	EKF	4.821	1.943	2.878	2.9%
		UKF	5.181	1.837	3.344	3.3%

ately 30% compared to the flexible model configuration. Second, the EKF systematically requires less processing time than the UKF, showing an average decrease of roughly 11%.

Interestingly, this performance gap between the two algorithms is noticeably smaller than theoretically expected. Because the current UKF configuration utilizes 43 sigma points, the state transition function (defined in (4.25)) must be evaluated as many times at every integration step, whereas the EKF only evaluates this function a single time. However, the EKF is forced to numerically calculate its system Jacobians. Due to the high dimensionality of the state vector, this numerical differentiation introduces significant computational overhead, balancing out the execution cost and bringing the total runtime of the EKF close to that of the UKF.

To establish a baseline for real-time feasibility, the final column of Table 6.4 presents the normalized execution time, defined as the percentage of the available physical simulation timeline consumed by the filter calculations. This metric highlights that the calculation time is mostly dependent on the internal model of the estimator. Across all evaluated configurations, the algorithms require only 2.4%–4.0% of the actual duration of the crane maneuvers to execute, satisfying the real-time execution compatibility requirement while leaving substantial computational headroom.

Chapter 7

Conclusions

This thesis addressed the critical problem of heavy-lift offshore crane lower-block (hook) localization, where wave-induced vessel motions excite unactuated load sway and massive structural boom elasticity corrupts conventional joint-encoder measurements. To overcome the limitations of state-of-the-art approaches that rely on vulnerable hook-mounted active hardware or assume an unrealistic rigid crane structure, a real-time state estimation framework was developed that does not rely on mounting active sensors on the lower-block. This work derived a coupled, non-linear multi-body dynamic model incorporating a flexible boom alongside wave-induced vessel motion to serve as the predictor step for a multi-rate filtering architecture. By fusing boom-mounted 3D LiDAR data, joint encoders, and a vessel Motion Reference Unit (MRU), the framework successfully reconstructs lower-block position and velocity states under active environmental disturbances.

The results demonstrate that accounting for structural boom flexibility is important for reliable offshore lower-block localization. Under model-matching simulation conditions, these results confirm that the core mathematical architecture is sound and capable of decoupling complex structural states. While an estimator utilizing a rigid boom model suffers from tracking errors that scale up rapidly during aggressive operations, the flexible model explicitly accounts for these transient structural states. Consequently, the flexible framework maintains a stable, centimeter-level error envelope regardless of whether the crane is executing nominal load transfers or navigating wave-induced vessel motions.

7.1 Requirement Validation

To evaluate the effectiveness of the Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) estimators, their tracking metrics were validated against the core target design requirements. Table 7.1 synthesizes these design targets against the achieved operational boundaries, evaluated on a per-axis basis to align with industrial criteria. Both estimation frameworks successfully fulfill all quantitative tracking requirements across the full envelope of operational scenarios. The achieved position and velocity Root Mean Square Errors (RMSE) reside comfortably within the strict target thresholds required for active anti-sway control. Furthermore, execution profiling on a standard processor (Intel Core i7-9750H) reveals that both

Table 7.1: **Requirement validation matrix for the EKF and UKF offshore crane lower-block localization frameworks. Accuracy values represent the performance envelope (minimum to maximum RMSE) across all evaluated axes and scenarios. Real-time execution denotes the percentage of the total available computational time budget consumed.**

Requirement Component	Target Boundary	Achieved (EKF)	Achieved (UKF)
Update Rate	$\geq 25\,\text{Hz}$	200 Hz	200 Hz
Position Accuracy	5.0–10.0 cm	1.67–3.58 cm	1.71–3.58 cm
Velocity Accuracy	5.0–10.0 cm/s	0.66–5.52 cm/s	0.76–5.51 cm/s
Real-Time Execution	$\leq 100\%$	$\leq 3.8\%$	$\leq 4.0\%$

filters consume less than 4.0% of the available real-time allocation window, confirming their computational viability and leaving significant processing margin for sensor processing and control loops.

7.2 Architectural Recommendation

Based on the comparative analysis, the EKF is recommended over the UKF as the optimal estimator for this localization architecture due to two primary engineering factors:

Higher Tracking Precision The EKF systematically minimized tracking errors within this framework, achieving an average 5.2% lower position RMSE and a 14.3% reduction in peak transient errors (Maximum Absolute Error). This unexpected performance gain occurs because the crane's smooth trajectories allow the EKF's first-order linearizations to remain highly accurate, whereas the UKF's deterministic sampling provides no additional benefit under the selected tuning parameters. Notably, this outcome remains specific to the evaluated simulation profiles, noise models, and filter tuning parameters utilized in this study; alternative operational conditions or a physical, real-world deployment could alter the comparative performance of the two estimators.

Reduced Computational Load The EKF demonstrated an approximate 11% reduction in estimation runtime compared to the UKF. Minimizing the state estimator's processing overhead is highly advantageous for physical deployment, as it preserves valuable computation time for upstream sensor pre-processing and downstream anti-sway control loops.

Chapter 8

Limitations and Future Work

8.1 Limitations of the Study

While the state estimation framework successfully validates the core tracking requirements, several modeling simplifications and environmental abstractions bound the scope of this study, as listed in this section.

Process Model Simplifications To maintain a manageable mathematical representation, several physical phenomena were simplified or omitted from the dynamics. First, the vessel's center of rotation was assumed to be stationary, neglecting how it shifts dynamically due to changing mass distributions as the crane boom and load move. Because boom-tip acceleration is heavily dominated by vessel pitch and roll moment-arms at this scale, any pivot shift distorts the estimator's dominant acceleration terms. Secondarily, vessel translational motions (surge, sway, heave) and yaw were excluded, which could introduce unmodeled acceleration components under severe sea states or translational drift. Finally, the internal model treats wire ropes as massless, rigid links, neglecting axial elasticity and high-frequency string oscillations, while structural vibrations were omitted entirely. As demonstrated by the rigid-model analysis, operating under such severe model mismatches can induce tracking degradation and destabilize the estimator during aggressive operations.

Sensor and Environmental Abstractions The simulation environment relies on idealized measurement behaviors that abstract away some harsh offshore realities. The framework emulates sensor uncertainties using zero-mean additive white Gaussian noise and assumes that the primary LiDAR data are perfectly pre-processed into clean 3D coordinate vectors. In physical offshore conditions, however, sensor noise is rarely strictly Gaussian. Furthermore, LiDAR performance degrades under environmental disturbances such as sea spray, heavy rain, and dense fog. These maritime conditions introduce data dropouts, erroneous reflections, and variable processing latencies that were not captured within this simulation framework. In a physical implementation, such point cloud distortions and data gaps would introduce measurement outliers or temporary measurement deprivation, directly corrupting the filter correction step and risking estimator divergence if left unmitigated.

8.2 Future Work

To expand upon the contributions of this thesis and bridge the gap toward industrial deployment, the following research directions are organized in a proposed order of engineering execution. This progression moves systematically from model validation to real-world sensor robustness and eventual control system integration.

Dynamic Model Verification The analytical equations of motion derived in this study should be verified against an independent multi-body physics engine. While this thesis demonstrated that the EKF and UKF achieve high tracking precision when the internal model aligns with the simulation environment, validation on an external platform is a critical next step. Testing the framework on an independent physics engine would confirm whether the derived flexible model accurately captures coupled structural states before transitioning to physical hardware.

Model Linearization and Simplification The analytical equations incorporating structural flexibility states are highly intractable and computationally dense, limiting direct physical interpretation and increasing estimator runtime. As revealed during execution profiling, dynamically calculating the numerical Jacobians for these high-dimensional states introduces significant computational overhead. To eliminate this overhead and minimize the computational footprint, future research should investigate symbolic model reduction techniques or localized, operating-point linearization, following methodologies similar to those established for flexible robotic mechanisms, as presented in [2]. Simplifying these symbolic expressions would also facilitate physical insight into the coupled dynamics of the crane system. Furthermore, linearizing the non-linear plant dynamics would open up the possibility of deploying mathematically optimal linear estimators, such as the standard Kalman Filter.

Latency and Sensor Dropouts Real-world implementations must handle asynchronous sensor dropouts and a combination of distinct latencies. Specifically, the framework needs to manage non-constant measurement latencies, LiDAR preprocessing delays, and downstream estimation lag before data reach the anti-sway loop. Expanding the framework to incorporate delayed-measurement compensation techniques is critical to ensure the filters can use late measurements and preserve tracking stability when a sensor signal is lost.

Feed-Forward of Operator and Motor Data The current framework operates strictly reactively, observing motions driven by external dynamics. Tracking precision could be improved by introducing an input vector ($\mathbf{u}$) to incorporate feed-forward data. Integrating operator joystick inputs and motor drive torques into the state transition step would allow the filter to predict structural and lower-block movements before they are registered by the sensors.

Anti-Sway Control Integration The state estimator should ultimately be integrated within a control framework, such as Model Predictive Control or a Linear Quadratic Regulator. Testing the estimator and controller combination will verify if this high tracking precision successfully reduces load sway without overloading the actuators or causing unwanted structural vibrations.

Bibliography

[1] Roberto Caporali. Anti-Sway Control on a Harbor Crane System Using a Command Smoothing Iterative Method. *Int. J. Innov. Res. Dev.*, 11(7):21–31, August 2022.

[2] Alessandro De Luca and Bruno Siciliano. Closed-Form Dynamic Model of Planar Multilink LightWeight Robots. *IEEE Trans. on Syst., Man, and Cybern.*, 21(4):826–839, August 1991.

[3] H. Díaz and C. Guedes Soares. Review of the Current Status, Technology and Future Trends of Offshore Wind Farms. *Ocean Eng.*, 209:107381:1–107381:21, August 2020.

[4] Luigi D'Alfonso, Walter Lucia, Pietro Muraca, and Paolo Pugliese. Mobile Robot Localization via EKF and UKF: A Comparison Based on Real Data. *Robot. and Auton. Syst.*, 74:122–127, December 2015.

[5] Mohamed Elsheikh, Umar Iqbal, Aboelmagd Noureldin, and Michael Korenberg. The Implementation of Precise Point Positioning (PPP): A Comprehensive Review. *Sensors*, 23(21):8874:1–8874:25, October 2023.

[6] Yihai Fang, Jingdao Chen, Yong K. Cho, Kinam Kim, Sijie Zhang, and Esau Perez. Vision-Based Load Sway Monitoring to Improve Crane Safety in Blind Lifts. *J. Struct. Integr. Maint.*, 3(4):233–242, November 2018.

[7] Gene Franklin, David Powell, and Abbas Emami-Naeini. *Feedback Control of Dynamic Systems*. Pearson, 8th edition, 2019.

[8] Jonah Gamba. *Radar Signal Processing for Autonomous Driving*. Signals and Communication Technology. Springer Nature, Singapore, 2020.

[9] Mario González, Dylan Jones, Andressa Santiso, Negar Akbari, David Melo, Luana Nogueira, and Rafael Vasconcelos. Offshore Wind and Energy Transition: Lessons Learned, Progress, and Trends. *Annu. Rev. Environ. Resour.*, 50:409–432, October 2025.

[10] Neil J. Gordon, David J. Salmond, and Adrian F. M. Smith. Novel Approach to Nonlinear/non-Gaussian Bayesian State Estimation. *IEE Proc. F Radar Signal Process.*, 140(2):107–113, April 1993.

[11] Thomas Gustafsson. *Modelling and control of rotary crane systems*. Ph.d. thesis, Luleå University of Technology, Luleå, Sweden, 1993.

[12] Mahmood Hassan and Jun Miura. Sensor Pose Estimation and 3D Mapping for Crane Operations Using Sensors Attached to the Crane Boom. *IEEE Access*, 11:90298–90308, August 2023.

[13] Huisman Equipment B.V. `https://www.huismanequipment.com/en`, 2026. Last accessed: June 10, 2026.

[14] Paweł Hyla. The Crane Control Systems: A Survey. In *Proc. 17th Int. Conf. Methods Models Autom. Robot. (MMAR)*, pages 505–509, Międzyzdroje, Poland, August 2012. IEEE.

[15] International Energy Agency. Renewables 2024. Technical report, International Energy Agency, Paris, France, October 2024.

[16] International Energy Agency. World Energy Outlook 2025. Technical report, International Energy Agency, Paris, France, November 2025.

[17] International Renewable Energy Agency. Renewable Energy Statistics 2025. Technical report, International Renewable Energy Agency, Abu Dhabi, United Arab Emirates, July 2025.

[18] Adamiec-Wójcik Iwona, Brzozowska Lucyna, Drąg Łukasz, Metelski Marek, and Wojciech Stanisław. Modelling 3D Dynamics of Offshore Lattice Jib Cranes by Means of the Rigid Finite Element Method. *J. Ocean Eng. Mar. Energy*, 9(3):495–513, August 2023.

[19] Kshitij Jerath, Sean Brennan, and Constantino Lagoa. Bridging the Gap Between Sensor Noise Modeling and Sensor Characterization. *Meas*, 116:350–366, February 2018.

[20] Simon J. Julier and Jeffrey K. Uhlmann. New extension of the Kalman filter to nonlinear systems. In *Signal Process., Sensor Fusion, and Target Recognit. VI*, pages 182–193, Orlando, FL, USA, April 1997. SPIE.

[21] Rudolf. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *J. Basic Eng.*, 82(1):35–45, March 1960.

[22] Yusuke Kawasaki and Jun Miura. Collision Warning by Rotating 2D LiDAR for Safe Crane Operation. In Ivan Petrovic, Emanuele Menegatti, and Ivan Marković, editors, *Intell. Auton. Syst. 17*, pages 330–341. Springer Nature, Cham, Germany, 2023.

[23] Kongsberg. `https://www.kongsberg.com/what-we-do/ocean-space/inertial-solutions/mru/mru-5/`, 2026. Last accessed: May. 6, 2026.

[24] Ghang Lee, Hong-Hyun Kim, Chi-Joo Lee, Sung-Il Ham, Seok-Heon Yun, Hunhee Cho, Bong Keun Kim, Gu Taek Kim, and Kyunghwan Kim. A Laser-Technology-Based Lifting-Path Tracking System for a Robotic Tower Crane. *Automat. Constr.*, 18(7):865–874, November 2009.

[25] Wenjun Li, Jiong Zhao, Zhen Jiang, Wei Chen, and Qicai Zhou. A Numerical Study of the Overall Stability of Flexible Giant Crane Booms. *J. Constr. Steel Res.*, 105:12–27, February 2015.

[26] X. Rong Li and Vesselin P. Jilkov. Survey of Maneuvering Target Tracking. Part I. Dynamic Models. *IEEE Trans. on Aerosp. and Electron. Syst.*, 39(4):1333–1364, October 2003.

[27] Xingxing Li, Jiaxin Huang, Xin Li, Zhiheng Shen, Junjie Han, Linyang Li, and Bo Wang. Review of PPP–RTK: Achievements, Challenges, and Opportunities. *Satell. Navig.*, 3:28:1–28:22, December 2022.

[28] Yi Li, Xuan Huang, Kong Fah Tee, Qiusheng Li, and Xiao-Peng Wu. Comparative Study of Onshore and Offshore Wind Characteristics and Wind Energy Potentials: A Case Study for Southeast Coastal Region of China. *Sustain. Energy Technol. and Assessments*, 39:100711:1–100711:13, June 2020.

[29] Sheng Liu, Hongze Yang, Zhiqiang Mei, Xiuying Xu, and Qianen He. Ultra-Wideband High Accuracy Distance Measurement Based on Hybrid Compensation of Temperature and Distance Error. *Meas.*, 206:112276:1–112276:8, January 2023.

[30] Biao Lu, Yongchun Fang, Ning Sun, and Xiangyu Wang. Antiswing Control of Offshore Boom Cranes With Ship Roll Disturbances. *IEEE Trans. Control Syst. Technol.*, 26(2):740–747, March 2018.

[31] Junqi Ma, Fusang Zhang, Beihong Jin, Chang Su, Siheng Li, Zhi Wang, and Jiazhi Ni. Push the Limit of Highly Accurate Ranging on Commercial UWB Devices. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 8(2):62:1–62:27, June 2024.

[32] K. Maes, G. De Roeck, and G. Lombaert. Motion Tracking of a Wind Turbine Blade During Lifting Using RTK-GPS/INS. *Eng. Structures*, 172:285–292, October 2018.

[33] Jörg Neupert, Eckhard Arnold, Klaus Schneider, and Oliver Sawodny. Tracking and anti-sway control for boom cranes. *Control Eng. Pract.*, 18(1):31–44, January 2010.

[34] Ouster Inc. `https://ouster.com/products/hardware/os1-max`, 2026. Last accessed: May. 6, 2026.

[35] Sujeet Milind Patole, Murat Torlak, Dan Wang, and Murtaza Ali. Automotive Radars: A Review of Signal Processing Techniques. *IEEE Signal Process. Mag.*, 34(2):22–35, March 2017.

[36] Leon C. Price, Jingdao Chen, Jisoo Park, and Yong K. Cho. Multisensor-Driven Real-Time Crane Monitoring System for blind Lift Operations: Lessons Learned From a Case Study. *Automat. Constr.*, 124:103552:1–103552:14, April 2021.

[37] Thomas Prässler and Jan Schaechtele. Comparison of the Financial Attractiveness Among Prospective Offshore Wind Parks in Selected European Countries. *Energy Policy*, 45:86–101, June 2012.

[38] Kun Qian, Shilin Zhu, Xinyu Zhang, and Li Erran Li. Robust Multimodal Vehicle Detection in Foggy Weather Using Complementary Lidar and Radar Signals. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 444–453, Nashville, TN, USA, August 2021. IEEE/CVF.

[39] Liyana Ramli, Z. Mohamed, Auwalu M. Abdullahi, H.I. Jaafar, and Izzuddin M. Lazim. Control Strategies for Crane Systems: A Comprehensive Review. *Mech. Syst. and Signal Process.*, 95:1–23, October 2017.

[40] Karl Åström and Bjorn Wittenmark. *Computer-Controller Systems: Theory and Design*. Dover Publications Inc., 3rd edition, 2011.

[41] Florentin Rauscher, Samuel Nann, and Oliver Sawodny. Motion Control of an Overhead Crane Using a Wireless Hook Mounted IMU. In *2018 Annu. Amer. Control Conf.*, pages 5677–5682, Milwaukee, WI, USA, June 2018. IEEE.

[42] O. Sawodny, H. Aschemann, J. Kumpel, C. Tarin, and K. Schneider. Anti-Sway Control For Boom Cranes. In *Proc. Amer. Control Conf. (ACC)*, pages 244–249, Anchorage, AK, USA, May 2002. IEEE.

[43] Ulf Schaper, Conrad Sagert, Oliver Sawodny, and Klaus Schneider. A load position observer for cranes with gyroscope measurements. *IFAC Proc. Vol.*, 44(1):3563–3568, January 2011. 18th IFAC World Congress.

[44] Ulf Schaper, Oliver Sawodny, Michael Zeitz, and Klaus Schneider. Load Position Estimation for Crane Anti-Sway Control Systems. *J. Dyn. Syst. Meas. Control*, 136(3):031013:1–031013:7, May 2014.

[45] Daniel Simon. *Optimal State Estimation: Kalman, H∞, and Nonlinear Approaches*. John Wiley & Sons, 2006.

[46] SymPy Development Team. `https://docs.sympy.org/latest/index.html`, 2025. Last accessed: June 11, 2026.

[47] Shahrior Tanvir and Amin Etminan. Comparative Analysis of Offshore and Onshore Wind Turbines: Efficiency, Design, and Environmental Impact. *Wind Eng.*, 50(1):200–215, February 2026.

[48] Jochen Teizer, Manu Venugopal, and Anupreet Walia. Ultrawideband for Automated Real-Time Three-Dimensional Location Sensing for Workforce, Equipment, and Material Positioning and Tracking. *Transp Res. Rec.*, 2081(1):56–64, January 2008.

[49] Mahmood Ul Hassan, Dipankar Das, and Jun Miura. 3D Mapping for a Large Crane Using Rotating 2D-Lidar and IMU Attached to the Crane Boom. *IEEE Access*, 11:21104–21116, February 2023.

[50] Domingos Sárvio Magalhães Valente, Abdul Momin, Tony Grift, and Alan Hansen. Accuracy and Precision Evaluation of Two Low-Cost RTK Global Navigation Satellite Systems. *Comput. Electron. Agric.*, 168:105142:1–105142:8, January 2020.

[51] Xingyuan Yan, Chenchen Liu, Mi Jiang, Meng Yang, Wei Feng, Min Zhong, and Lingzhi Peng. Performance Analysis of Oceanographic Research Vessel Precise Point Positioning Based on BDS/GNSS RTK Receivers. *Meas.*, 211:112637:1–112637:13, April 2023.

[52] Yufeng Yin, Xiaoyan Liu, Guancheng Li, Qing Fan, and Sanglan Li. Accurate Crane Hook Tracking and Height Measurement With Monocular Vision. *IEEE Trans. Instrum. Meas.*, 74:5047414:1–5047414:14, October 2025.

[53] C. Zhang, A. Hammad, and S. Rodriguez. Crane Pose Estimation Using UWB Real-Time Location System. *J. Comput. Civ. Eng.*, 26(5):625–637, September 2012.

Appendix A

Equations of Motion

A.1 Crane with a Static Base

Below are the EoMs describing the movement of a swinging lower-block for a crane with a static base.

$$\begin{aligned}\ddot{\alpha} = \frac{1}{l}\Big(&- 2\dot{\alpha}\dot{l} - \ddot{\nu}\ell\sin\beta - g\sin\alpha\cos\beta - \dot{\nu}^2 b\cos\alpha - 2\dot{\nu}\dot{l}\sin\beta \\ &+ \ddot{\nu}b\sin\alpha\sin\beta - L\ddot{\mu}\sin\mu\cos\alpha - \dot{\beta}^2\ell\sin\alpha\cos\alpha \\ &- L\dot{\mu}^2\cos\alpha\cos\mu - L\dot{\nu}^2\cos\alpha\cos\mu + L\ddot{\nu}\sin\alpha\sin\beta\cos\mu \\ &+ \dot{\nu}^2\ell\sin\alpha\cos\alpha\cos^2\beta + L\dot{\mu}^2\sin\alpha\sin\mu\cos\beta \\ &- L\ddot{\mu}\sin\alpha\cos\beta\cos\mu - 2\dot{\beta}\dot{\nu}\ell\cos^2\alpha\cos\beta \\ &- 2L\dot{\mu}\dot{\nu}\sin\alpha\sin\beta\sin\mu\Big)\end{aligned} \tag{A.1}$$

$$\begin{aligned}\ddot{\beta} = \frac{1}{l\cos\alpha}\Big(&- g\sin\beta - \ddot{\nu}b\cos\beta - 2\dot{\beta}\dot{l}\cos\alpha + L\dot{\mu}^2\sin\beta\sin\mu \\ &- L\ddot{\mu}\sin\beta\cos\mu - L\ddot{\nu}\cos\beta\cos\mu + \ddot{\nu}\ell\cos\alpha\cos\beta\tan\alpha \\ &+ \dot{\nu}^2\ell\sin\beta\cos\alpha\cos\beta + 2\dot{\alpha}\dot{\beta}\ell\cos\alpha\tan\alpha \\ &+ 2\dot{\alpha}\dot{\nu}\ell\cos\alpha\cos\beta + 2L\dot{\mu}\dot{\nu}\sin\mu\cos\beta \\ &+ 2\dot{\nu}\dot{l}\cos\alpha\cos\beta\tan\alpha\Big)\end{aligned} \tag{A.2}$$

A.2 Vessel-Mounted Crane

Below are the EoMs describing the movement of a swinging lower-block for a crane mounted on a moving vessel.

$$\begin{aligned}\ddot{\alpha} = \frac{1}{l}\Big(&- 2\dot{\alpha}\dot{l} - \ddot{\nu}\ell\sin\beta - \dot{\nu}^2 b\cos\alpha - \dot{p}^2 b\cos\alpha - 2\dot{\nu}\dot{l}\sin\beta \\ &+ \ddot{\nu}b\sin\alpha\sin\beta + \ddot{r}c_x\sin\alpha\cos\beta + \ddot{r}c_z\cos\alpha\cos\nu \\ &+ \ddot{r}h\cos\alpha\cos\nu + g\sin\nu\sin p\cos\alpha + \dot{r}^2 c_z\sin\alpha\cos\beta \\ &+ \dot{r}^2 h\sin\alpha\cos\beta - L\ddot{\mu}\sin\mu\cos\alpha - \ddot{p}\ell\sin\beta\sin r - \ddot{r}\ell\cos\beta\cos\nu \\ &- \dot{\beta}^2\ell\sin\alpha\cos\alpha - \dot{r}^2\ell\sin\alpha\cos\alpha - L\dot{\mu}^2\cos\alpha\cos\mu\end{aligned}$$

$$
\begin{aligned}
&- L\dot{\nu}^2 \cos\alpha \cos\mu - L\dot{p}^2 \cos\alpha \cos\mu - \dot{r}^2 b \cos\alpha \cos^2\nu \\
&- \dot{p}^2 c_x \cos\alpha \cos\nu - \dot{r}^2 c_x \cos\alpha \cos\nu - \dot{p}^2 c_y \sin\nu \cos\alpha \\
&- 2\dot{l}\dot{p} \sin\beta \sin r - 2\dot{l}\dot{r} \cos\beta \cos\nu + L\ddot{\nu} \sin\alpha \sin\beta \cos\mu \\
&+ \ddot{p}\ell \sin\nu \cos\beta \cos r + \ddot{p} b \sin\alpha \sin\beta \sin r + \ddot{p} c_x \sin\nu \sin r \cos\alpha \\
&+ L\ddot{r} \sin\mu \cos\alpha \cos\nu + \ddot{r} b \sin\alpha \cos\beta \cos\nu + g \sin\alpha \sin\beta \sin p \cos\nu \\
&+ \dot{\nu}^2 \ell \sin\alpha \cos\alpha \cos^2\beta + \dot{p}^2 \ell \sin\alpha \cos\alpha \cos^2\beta \\
&+ \dot{p}^2 \ell \sin\alpha \cos\alpha \cos^2 r + L\dot{\mu}^2 \sin\alpha \sin\mu \cos\beta \\
&+ L\dot{r}^2 \sin\alpha \sin\mu \cos\beta + \dot{p}^2 b \cos\alpha \cos^2\nu \cos^2 r \\
&+ \dot{p}^2 c_x \cos\alpha \cos\nu \cos^2 r + \dot{p}^2 c_x \sin\alpha \sin\beta \sin\nu \\
&+ \dot{r}^2 c_x \sin\alpha \sin\beta \sin\nu + \dot{p}^2 c_z \sin\alpha \cos\beta \cos^2 r \\
&+ \dot{p}^2 h \sin\alpha \cos\beta \cos^2 r - L\ddot{\mu} \sin\alpha \cos\beta \cos\mu - \ddot{p} c_y \sin r \cos\alpha \cos\nu \\
&- \ddot{p} c_y \sin\alpha \cos\beta \cos r - \ddot{p} c_z \sin\nu \cos\alpha \cos r - \ddot{p} h \sin\nu \cos\alpha \cos r \\
&- \ddot{r} c_z \sin\alpha \sin\beta \sin\nu - \ddot{r} h \sin\alpha \sin\beta \sin\nu - g \sin r \cos\alpha \cos\nu \cos p \\
&- g \sin\alpha \cos\beta \cos p \cos r - \dot{r}^2 \ell \sin\beta \sin\nu \cos\nu - L\dot{r}^2 \cos\alpha \cos\mu \cos^2\nu \\
&- \dot{p}^2 c_y \sin\alpha \sin\beta \cos\nu - 2\dot{\beta}\dot{\nu}\ell \cos^2\alpha \cos\beta \\
&- 2\dot{\nu}\dot{p} b \sin r \cos\alpha + 2\dot{\nu}\dot{r}\ell \sin\nu \cos\beta + 2\dot{l}\dot{p} \sin\nu \cos\beta \cos r \\
&+ 2\dot{r}^2 \ell \sin\alpha \cos\alpha \cos^2\nu + L\ddot{p} \sin\alpha \sin\beta \sin r \cos\mu \\
&+ \ddot{p} c_x \sin\alpha \sin\beta \sin r \cos\nu + \ddot{p} c_y \sin\alpha \sin\beta \sin\nu \sin r \\
&+ L\ddot{r} \sin\alpha \cos\beta \cos\mu \cos\nu + g \sin\alpha \sin\beta \sin\nu \sin r \cos p \\
&+ \dot{p}^2 \ell \sin\beta \sin\nu \cos\nu \cos^2 r + \dot{p}^2 \ell \sin r \cos\beta \cos\nu \cos r \\
&+ L\dot{p}^2 \cos\alpha \cos\mu \cos^2\nu \cos^2 r + L\dot{p}^2 \sin\alpha \sin\mu \cos\beta \cos^2 r \\
&+ \dot{r}^2 b \sin\alpha \sin\beta \sin\nu \cos\nu + \dot{p}^2 c_z \sin r \cos\alpha \cos\nu \cos r \\
&+ \dot{p}^2 h \sin r \cos\alpha \cos\nu \cos r - L\ddot{p} \sin\mu \sin\nu \cos\alpha \cos r \\
&- \ddot{p} b \sin\alpha \sin\nu \cos\beta \cos r - \ddot{p} c_z \sin\alpha \sin\beta \cos\nu \cos r \\
&- \ddot{p} h \sin\alpha \sin\beta \cos\nu \cos r - L\ddot{r} \sin\alpha \sin\beta \sin\mu \sin\nu \\
&- \dot{r}^2 \ell \sin\alpha \cos\alpha \cos^2\beta \cos^2\nu - \dot{p}^2 c_x \sin\alpha \sin\beta \sin\nu \cos^2 r \\
&- \dot{p}^2 c_x \sin\alpha \sin r \cos\beta \cos r - 2\dot{\beta}\dot{p}\ell \sin r \cos^2\alpha \cos\beta \\
&- 2\dot{\beta}\dot{r}\ell \sin\alpha \sin\nu \cos\alpha - 2L\dot{\mu}\dot{\nu} \sin\alpha \sin\beta \sin\mu \\
&- 2L\dot{\nu}\dot{p} \sin r \cos\alpha \cos\mu - 2\dot{\nu}\dot{r}\ell \sin\nu \cos^2\alpha \cos\beta \\
&- 2\dot{\nu}\dot{r} b \sin\alpha \sin\nu \cos\beta - 2\dot{p}\dot{r}\ell \sin\beta \cos^2\alpha \cos r \\
&- 2\dot{p}\dot{r}\ell \sin\beta \cos^2\nu \cos r - 2\dot{p}^2 \ell \sin\alpha \cos\alpha \cos^2\beta \cos^2 r \\
&- 2\dot{p}^2 \ell \sin\alpha \cos\alpha \cos^2\nu \cos^2 r + 2\dot{\beta}\dot{r}\ell \sin\beta \cos^2\alpha \cos\nu \\
&+ 2L\dot{\mu}\dot{r} \cos\alpha \cos\mu \cos\nu + 2\dot{\nu}\dot{p}\ell \cos\beta \cos\nu \cos r \\
&+ 2\dot{p}\dot{r} c_x \sin\nu \cos\alpha \cos r + 2\dot{p}\dot{r} c_z \sin\nu \sin r \cos\alpha \\
&+ 2\dot{p}\dot{r} h \sin\nu \sin r \cos\alpha + 2\dot{r}^2 \ell \sin\beta \sin\nu \cos^2\alpha \cos\nu \\
&+ \dot{p}^2 \ell \sin\alpha \cos\alpha \cos^2\beta \cos^2\nu \cos^2 r + L\dot{p}^2 \sin\mu \sin r \cos\alpha \cos\nu \cos r \\
&+ L\dot{r}^2 \sin\alpha \sin\beta \sin\nu \cos\mu \cos\nu - L\ddot{p} \sin\alpha \sin\nu \cos\beta \cos\mu \cos r
\end{aligned}
$$

$$
\begin{aligned}
&- L\ddot{p}\sin\alpha\sin\beta\sin\mu\cos\nu\cos r - \dot{p}^2 b\sin\alpha\sin\beta\sin\nu\cos\nu\cos^2 r \\
&- \dot{p}^2 b\sin\alpha\sin r\cos\beta\cos\nu\cos r - \dot{p}^2 c_z\sin\alpha\sin\beta\sin\nu\sin r\cos r \\
&- \dot{p}^2 h\sin\alpha\sin\beta\sin\nu\sin r\cos r - 2\dot{\beta}\dot{p}\ell\sin\beta\sin\nu\cos^2\alpha\cos r \\
&- 2\dot{\beta}\dot{p}\ell\sin\alpha\cos\alpha\cos\nu\cos r - 2L\dot{\mu}\dot{p}\sin\nu\cos\alpha\cos\mu\cos r \\
&- 2L\dot{\mu}\dot{p}\sin\alpha\sin\beta\sin\mu\sin r - 2L\dot{\mu}\dot{r}\sin\alpha\sin\mu\cos\beta\cos\nu \\
&- 2L\dot{\mu}\dot{r}\sin\alpha\sin\beta\sin\nu\cos\mu - 2\dot{\nu}\dot{p}\ell\cos^2\alpha\cos\beta\cos\nu\cos r \\
&- 2\dot{\nu}\dot{p}b\sin\alpha\cos\beta\cos\nu\cos r - 2L\dot{\nu}\dot{r}\sin\alpha\sin\nu\cos\beta\cos\mu \\
&- 2\dot{p}\dot{r}\ell\sin\nu\sin r\cos^2\alpha\cos\beta - 2\dot{p}^2\ell\sin\beta\sin\nu\cos^2\alpha\cos\nu\cos^2 r \\
&- 2\dot{p}^2\ell\sin r\cos^2\alpha\cos\beta\cos\nu\cos r + 2\dot{\nu}\dot{p}\ell\sin\alpha\sin r\cos\alpha\cos^2\beta \\
&+ 2L\dot{p}\dot{r}\sin\mu\sin\nu\sin r\cos\alpha + 2\dot{p}\dot{r}b\sin\alpha\sin\beta\cos^2\nu\cos r \\
&+ 2\dot{p}\dot{r}b\sin\nu\cos\alpha\cos\nu\cos r + 2\dot{p}\dot{r}c_x\sin\alpha\sin\beta\cos\nu\cos r \\
&+ 2\dot{p}\dot{r}c_z\sin\alpha\sin\beta\sin r\cos\nu + 2\dot{p}\dot{r}h\sin\alpha\sin\beta\sin r\cos\nu \\
&+ 4\dot{p}\dot{r}\ell\sin\beta\cos^2\alpha\cos^2\nu\cos r \\
&- L\dot{p}^2\sin\alpha\sin\beta\sin\nu\cos\mu\cos\nu\cos^2 r \\
&- L\dot{p}^2\sin\alpha\sin r\cos\beta\cos\mu\cos\nu\cos r \\
&- L\dot{p}^2\sin\alpha\sin\beta\sin\mu\sin\nu\sin r\cos r \\
&- 4\dot{p}\dot{r}\ell\sin\alpha\sin\nu\cos\alpha\cos\nu\cos r \\
&- 2L\dot{\mu}\dot{p}\sin\alpha\sin\beta\cos\mu\cos\nu\cos r \\
&- 2L\dot{\nu}\dot{p}\sin\alpha\cos\beta\cos\mu\cos\nu\cos r \\
&- 2\dot{\nu}\dot{r}\ell\sin\alpha\sin\beta\cos\alpha\cos\beta\cos\nu \\
&+ 2L\dot{\mu}\dot{p}\sin\alpha\sin\mu\sin\nu\cos\beta\cos r \\
&+ 2L\dot{p}\dot{r}\sin\alpha\sin\beta\cos\mu\cos^2\nu\cos r \\
&+ 2L\dot{p}\dot{r}\sin\nu\cos\alpha\cos\mu\cos\nu\cos r + 2L\dot{p}\dot{r}\sin\alpha\sin\beta\sin\mu\sin r\cos\nu \\
&- 2\dot{p}\dot{r}\ell\sin\alpha\sin\beta\sin r\cos\alpha\cos\beta\cos\nu \\
&+ 2\dot{\nu}\dot{p}\ell\sin\alpha\sin\beta\sin\nu\cos\alpha\cos\beta\cos r \\
&+ 2\dot{p}\dot{r}\ell\sin\alpha\sin\nu\cos\alpha\cos^2\beta\cos\nu\cos r \\
&+ 2\dot{p}^2\ell\sin\alpha\sin\beta\sin\nu\sin r\cos\alpha\cos\beta\cos r\Big)
\end{aligned}
\tag{A.3}
$$

$$
\begin{aligned}
\ddot{\beta} = \frac{1}{l\cos\alpha}\Big(&\ddot{r}c_x\sin\beta + \dot{r}^2 c_z\sin\beta + \dot{r}^2 h\sin\beta - \ddot{\nu}b\cos\beta - 2\dot{\beta}\dot{l}\cos\alpha \\
&+ \ddot{r}b\sin\beta\cos\nu + \ddot{r}c_z\sin\nu\cos\beta + \ddot{r}h\sin\nu\cos\beta + L\dot{\mu}^2\sin\beta\sin\mu \\
&+ L\dot{r}^2\sin\beta\sin\mu + \dot{p}^2 c_y\cos\beta\cos\nu + \dot{p}^2 c_z\sin\beta\cos^2 r \\
&+ \dot{p}^2 h\sin\beta\cos^2 r - L\ddot{\mu}\sin\beta\cos\mu - L\ddot{\nu}\cos\beta\cos\mu - \ddot{p}b\sin r\cos\beta \\
&- \ddot{p}c_y\sin\beta\cos r - \ddot{r}\ell\sin\nu\cos\alpha - g\sin p\cos\beta\cos\nu - g\sin\beta\cos p\cos r \\
&- \dot{p}^2 c_x\sin\nu\cos\beta - \dot{r}^2 c_x\sin\nu\cos\beta - 2\dot{l}\dot{r}\sin\nu\cos\alpha \\
&+ \ddot{\nu}\ell\cos\alpha\cos\beta\tan\alpha + \ddot{p}c_z\cos\beta\cos\nu\cos r + \ddot{p}h\cos\beta\cos\nu\cos r \\
&+ L\ddot{r}\sin\mu\sin\nu\cos\beta + L\ddot{r}\sin\beta\cos\mu\cos\nu + \dot{\nu}^2\ell\sin\beta\cos\alpha\cos\beta \\
&+ \dot{p}^2\ell\sin\beta\cos\alpha\cos\beta + L\dot{p}^2\sin\beta\sin\mu\cos^2 r \\
&+ \dot{p}^2 c_x\sin\nu\cos\beta\cos^2 r - \ddot{p}\ell\cos\alpha\cos\nu\cos r - L\ddot{p}\sin r\cos\beta\cos\mu
\end{aligned}
$$

$$
\begin{aligned}
&- \ddot{p} b \sin\beta \sin\nu \cos r - \ddot{p} c_x \sin r \cos\beta \cos\nu - \ddot{p} c_y \sin\nu \sin r \cos\beta \\
&- g \sin\nu \sin r \cos\beta \cos p - \dot{r}^2 b \sin\nu \cos\beta \cos\nu - \dot{p}^2 c_x \sin\beta \sin r \cos r \\
&- 2\dot{\nu}\dot{r}\ell \cos\alpha \cos\nu - 2\dot{\nu}\dot{r} b \sin\beta \sin\nu - 2\dot{l}\dot{p} \cos\alpha \cos\nu \cos r \\
&+ 2\dot{\alpha}\dot{\beta}\ell \cos\alpha \tan\alpha + 2\dot{\alpha}\dot{\nu}\ell \cos\alpha \cos\beta \\
&+ 2L\dot{\mu}\dot{\nu} \sin\mu \cos\beta + 2\dot{\nu}\dot{l} \cos\alpha \cos\beta \tan\alpha \\
&+ \ddot{p}\ell \sin r \cos\alpha \cos\beta \tan\alpha + L\ddot{p} \sin\mu \cos\beta \cos\nu \cos r \\
&+ \dot{p}^2 \ell \sin\nu \sin r \cos\alpha \cos r + \dot{p}^2 b \sin\nu \cos\beta \cos\nu \cos^2 r \\
&+ \dot{p}^2 c_z \sin\nu \sin r \cos\beta \cos r + \dot{p}^2 h \sin\nu \sin r \cos\beta \cos r \\
&- L\ddot{p} \sin\beta \sin\nu \cos\mu \cos r - \ddot{r}\ell \sin\beta \cos\alpha \cos\nu \tan\alpha \\
&- \dot{r}^2 \ell \sin\beta \cos\alpha \cos\beta \cos^2\nu - L\dot{r}^2 \sin\nu \cos\beta \cos\mu \cos\nu \\
&- \dot{p}^2 b \sin\beta \sin r \cos\nu \cos r - 2\dot{\alpha}\dot{r}\ell \sin\beta \cos\alpha \cos\nu \\
&- 2L\dot{\mu}\dot{r} \sin\beta \sin\mu \cos\nu - 2\dot{\nu}\dot{p} b \sin\beta \cos\nu \cos r \\
&- 2L\dot{\nu}\dot{r} \sin\beta \sin\nu \cos\mu - 2\dot{l}\dot{r} \sin\beta \cos\alpha \cos\nu \tan\alpha \\
&- 2\dot{p}\dot{r} b \cos\beta \cos^2\nu \cos r - 2\dot{p}\dot{r} c_x \cos\beta \cos\nu \cos r \\
&- 2\dot{p}\dot{r} c_z \sin r \cos\beta \cos\nu - 2\dot{p}\dot{r} h \sin r \cos\beta \cos\nu \\
&- 2\dot{p}^2 \ell \sin\beta \cos\alpha \cos\beta \cos^2 r + 2\dot{\alpha}\dot{p}\ell \sin r \cos\alpha \cos\beta \\
&+ 2\dot{\alpha}\dot{r}\ell \sin\nu \cos\alpha \tan\alpha + 2L\dot{\mu}\dot{p} \sin\mu \sin r \cos\beta \\
&+ 2L\dot{\mu}\dot{r} \sin\nu \cos\beta \cos\mu + 2\dot{\nu}\dot{p}\ell \sin\nu \cos\alpha \cos r \\
&+ 2\dot{\nu}\dot{r}\ell \cos\alpha \cos^2\beta \cos\nu + 2\dot{l}\dot{p} \sin r \cos\alpha \cos\beta \tan\alpha \\
&+ \ddot{p}\ell \sin\beta \sin\nu \cos\alpha \cos r \tan\alpha + \dot{p}^2 \ell \sin\beta \cos\alpha \cos\beta \cos^2\nu \cos^2 r \\
&+ \dot{r}^2 \ell \sin\nu \cos\alpha \cos\beta \cos\nu \tan\alpha + L\dot{p}^2 \sin\nu \cos\beta \cos\mu \cos\nu \cos^2 r \\
&+ L\dot{p}^2 \sin\mu \sin\nu \sin r \cos\beta \cos r - L\dot{p}^2 \sin\beta \sin r \cos\mu \cos\nu \cos r \\
&- 2\dot{\nu}\dot{p}\ell \sin\nu \cos\alpha \cos^2\beta \cos r - 2L\dot{\nu}\dot{p} \sin\beta \cos\mu \cos\nu \cos r \\
&- 2L\dot{p}\dot{r} \cos\beta \cos\mu \cos^2\nu \cos r - 2L\dot{p}\dot{r} \sin\mu \sin r \cos\beta \cos\nu \\
&- 2\dot{p}^2 \ell \sin\nu \sin r \cos\alpha \cos^2\beta \cos r + 2\dot{\alpha}\dot{p}\ell \cos\alpha \cos\nu \cos r \tan\alpha \\
&+ 2\dot{\alpha}\dot{p}\ell \sin\beta \sin\nu \cos\alpha \cos r + 2L\dot{\mu}\dot{p} \cos\beta \cos\mu \cos\nu \cos r \\
&+ 2L\dot{\mu}\dot{p} \sin\beta \sin\mu \sin\nu \cos r + 2\dot{\nu}\dot{p}\ell \sin\beta \sin r \cos\alpha \cos\beta \\
&+ 2\dot{\nu}\dot{r}\ell \sin\beta \sin\nu \cos\alpha \tan\alpha + 2\dot{l}\dot{p} \sin\beta \sin\nu \cos\alpha \cos r \tan\alpha \\
&+ 2\dot{p}\dot{r}\ell \sin r \cos\alpha \cos^2\beta \cos\nu + \dot{p}^2 \ell \sin\beta \sin r \cos\alpha \cos\nu \cos r \tan\alpha \\
&- \dot{p}^2 \ell \sin\nu \cos\alpha \cos\beta \cos\nu \cos^2 r \tan\alpha \\
&+ 2\dot{\nu}\dot{p}\ell \sin\beta \cos\alpha \cos\nu \cos r \tan\alpha \\
&+ 2\dot{p}\dot{r}\ell \cos\alpha \cos\beta \cos^2\nu \cos r \tan\alpha \\
&+ 2\dot{p}\dot{r}\ell \sin\beta \sin\nu \cos\alpha \cos\beta \cos\nu \cos r \Big)
\end{aligned}
\tag{A.4}
$$

A.3 Vessel-Mounted Crane with Flexible Boom

As established in Section 4.2.4, evaluating the expanded Euler-Lagrange equations for the crane system yields a complex set of coupled, non-linear differential equa-

tions. To prevent the symbolic expressions from swelling further, these equations are not solved analytically for individual coordinate accelerations. Instead, they are factored into a matrix-vector form:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} = \mathbf{F}(\mathbf{q}, \dot{\mathbf{q}}), \tag{A.5}$$

where:

- $\mathbf{q} = [\alpha, \beta, \nu_f]^T$ is the 3×1 vector of generalized coordinates representing the two sway angles and the lateral boom deflection angle, respectively.
- $\mathbf{M}(\mathbf{q})$ is the 3×3 symmetric mass matrix, representing the configuration-dependent inertia properties of the system.
- $\mathbf{F}(\mathbf{q}, \dot{\mathbf{q}})$ is the 3×1 generalized forcing vector, incorporating gravitational forces, Coriolis forces, centrifugal forces, and non-conservative generalized dissipation forces.

During numerical simulation, the instantaneous accelerations are resolved at each time step by computing $\ddot{\mathbf{q}} = \mathbf{M}^{-1}\mathbf{F}$ using numerical linear system solvers, bypassing the need for a costly symbolic matrix inversion.

A.3.1 System Matrices and Vectors

The components of the mass matrix $\mathbf{M}(\mathbf{q})$ are given by the symmetric matrix:

$$\mathbf{M}(\mathbf{q}) = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33}. \end{bmatrix} \tag{A.6}$$

Due to the symmetry of the mass matrix, $M_{12} = M_{21}$, $M_{13} = M_{31}$, and $M_{23} = M_{32}$.

The total generalized forcing vector $\mathbf{F}(\mathbf{q}, \dot{\mathbf{q}})$ is composed of the conservative and inertial forces derived directly from the system Lagrangian ($\mathbf{F}_L$), combined with the non-conservative structural damping derived via the Rayleigh dissipation function ($\mathbf{F}_{\text{damping}}$):

$$\mathbf{F}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{F}_L(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{F}_{\text{damping}}(\dot{\mathbf{q}}), \tag{A.7}$$

$$\mathbf{F}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} F_{L,1}(\mathbf{q}, \dot{\mathbf{q}}) \\ F_{L,2}(\mathbf{q}, \dot{\mathbf{q}}) \\ F_{L,3}(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -c_f \dot{\nu}_f \end{bmatrix}. \tag{A.8}$$

where $\mathbf{F}_L$ contains the Coriolis, centrifugal, gravitational, and torsional spring stiffness forces ($-k_f \nu_f$). The torsional damping coefficient c_f operates strictly as a non-conservative force opposing the velocity of the boom deflection coordinate ν_f in the third row.

A.4 Symbolic Expressions

The explicit, unsimplified symbolic expressions for each unique element of the mass matrix and the Lagrangian forcing vector are detailed below:

$$M_{11} = l^2 m,$$

$$
\begin{aligned}
M_{12} &= 0, \\
M_{13} &= L\ell m\left(-\sin\alpha\sin\beta\cos\nu_f + \sin\nu_f\cos\alpha\right)\cos\mu, \\
M_{22} &= l^2 m\cos^2\alpha, \\
M_{23} &= L\ell m\cos\alpha\cos\beta\cos\mu\cos\nu_f, \\
M_{33} &= L^2\left(m+m_b\right)\cos^2\mu.
\end{aligned}
$$

Similarly, the entries of the forcing vector derived from the system Lagrangian (prior to appending the explicit Rayleigh damping term) are defined as:

$$
\begin{aligned}
F_{L,1} = &-\ddot{\nu}l^2 m\sin\beta - 2\dot{\alpha}\dot{l}\ell m - \ddot{p}l^2 m\sin\beta\sin r - \ddot{r}l^2 m\cos\beta\cos\nu \\
&- \dot{\nu}^2 b\ell m\cos\alpha - \dot{p}^2 b\ell m\cos\alpha - \dot{\beta}^2 l^2 m\sin\alpha\cos\alpha \\
&- \dot{r}^2 l^2 m\sin\alpha\cos\alpha - 2\dot{\nu}\dot{l}\ell m\sin\beta + \ddot{\nu}b\ell m\sin\alpha\sin\beta \\
&+ \ddot{p}l^2 m\sin\nu\cos\beta\cos r + \ddot{r}c_x\ell m\sin\alpha\cos\beta + \ddot{r}c_z\ell m\cos\alpha\cos\nu \\
&+ \ddot{r}h\ell m\cos\alpha\cos\nu + g\ell m\sin\nu\sin p\cos\alpha + \dot{r}^2 c_z\ell m\sin\alpha\cos\beta \\
&+ \dot{r}^2 h\ell m\sin\alpha\cos\beta + \dot{\nu}^2 l^2 m\sin\alpha\cos\alpha\cos^2\beta \\
&+ \dot{p}^2 l^2 m\sin\alpha\cos\alpha\cos^2\beta + \dot{p}^2 l^2 m\sin\alpha\cos\alpha\cos^2 r \\
&- \dot{r}^2 b\ell m\cos\alpha\cos^2\nu - \dot{p}^2 c_x\ell m\cos\alpha\cos\nu - \dot{r}^2 c_x\ell m\cos\alpha\cos\nu \\
&- \dot{p}^2 c_y\ell m\sin\nu\cos\alpha - \dot{r}^2 l^2 m\sin\beta\sin\nu\cos\nu \\
&- 2\dot{\beta}\dot{\nu}l^2 m\cos^2\alpha\cos\beta - 2\dot{l}\dot{p}\ell m\sin\beta\sin r \\
&- 2\dot{l}\dot{r}\ell m\cos\beta\cos\nu + 2\dot{\nu}\dot{r}l^2 m\sin\nu\cos\beta \\
&+ 2\dot{r}^2 l^2 m\sin\alpha\cos\alpha\cos^2\nu + \ddot{p}b\ell m\sin\alpha\sin\beta\sin r \\
&+ \ddot{p}c_x\ell m\sin\nu\sin r\cos\alpha + L\ddot{r}\ell m\sin\mu\cos\alpha\cos\nu \\
&+ \ddot{r}b\ell m\sin\alpha\cos\beta\cos\nu + g\ell m\sin\alpha\sin\beta\sin p\cos\nu \\
&+ L\dot{\mu}^2\ell m\sin\alpha\sin\mu\cos\beta + L\dot{r}^2\ell m\sin\alpha\sin\mu\cos\beta \\
&+ \dot{p}^2 b\ell m\cos\alpha\cos^2\nu\cos^2 r + \dot{p}^2 c_x\ell m\cos\alpha\cos\nu\cos^2 r \\
&+ \dot{p}^2 c_x\ell m\sin\alpha\sin\beta\sin\nu + \dot{r}^2 c_x\ell m\sin\alpha\sin\beta\sin\nu \\
&+ \dot{p}^2 c_z\ell m\sin\alpha\cos\beta\cos^2 r + \dot{p}^2 h\ell m\sin\alpha\cos\beta\cos^2 r \\
&+ \dot{p}^2 l^2 m\sin\beta\sin\nu\cos\nu\cos^2 r + \dot{p}^2 l^2 m\sin r\cos\beta\cos\nu\cos r \\
&- L\ddot{\mu}\ell m\sin\mu\cos\alpha\cos\nu_f - L\ddot{\mu}\ell m\sin\alpha\cos\beta\cos\mu \\
&- L\ddot{\nu}\ell m\sin\nu_f\cos\alpha\cos\mu - \ddot{p}c_y\ell m\sin r\cos\alpha\cos\nu \\
&- \ddot{p}c_y\ell m\sin\alpha\cos\beta\cos r - \ddot{p}c_z\ell m\sin\nu\cos\alpha\cos r \\
&- \ddot{p}h\ell m\sin\nu\cos\alpha\cos r - \ddot{r}c_z\ell m\sin\alpha\sin\beta\sin\nu \\
&- \ddot{r}h\ell m\sin\alpha\sin\beta\sin\nu - g\ell m\sin r\cos\alpha\cos\nu\cos p - g\ell m\sin\alpha\cos\beta\cos p\cos r \\
&- L\dot{\mu}^2\ell m\cos\alpha\cos\mu\cos\nu_f - L\dot{\nu_f}^2\ell m\cos\alpha\cos\mu\cos\nu_f \\
&- L\dot{\nu}^2\ell m\cos\alpha\cos\mu\cos\nu_f - L\dot{p}^2\ell m\cos\alpha\cos\mu\cos\nu_f \\
&- L\dot{r}^2\ell m\cos\alpha\cos\mu\cos\nu_f - \dot{p}^2 c_y\ell m\sin\alpha\sin\beta\cos\nu \\
&- \dot{r}^2 l^2 m\sin\alpha\cos\alpha\cos^2\beta\cos^2\nu - 2\dot{\beta}\dot{p}l^2 m\sin r\cos^2\alpha\cos\beta \\
&- 2\dot{\beta}\dot{r}l^2 m\sin\alpha\sin\nu\cos\alpha - 2\dot{\nu}\dot{p}b\ell m\sin r\cos\alpha \\
&- 2\dot{\nu}\dot{r}l^2 m\sin\nu\cos^2\alpha\cos\beta - 2\dot{p}\dot{r}l^2 m\sin\beta\cos^2\alpha\cos r
\end{aligned}
$$

$$
\begin{aligned}
&- 2\dot{p}\dot{r}l^2 m \sin\beta \cos^2\nu \cos r - 2\dot{p}^2 l^2 m \sin\alpha \cos\alpha \cos^2\beta \cos^2 r \\
&- 2\dot{p}^2 l^2 m \sin\alpha \cos\alpha \cos^2\nu \cos^2 r + 2\dot{\beta}\dot{r}l^2 m \sin\beta \cos^2\alpha \cos\nu \\
&+ 2\dot{\nu}\dot{p}l^2 m \cos\beta \cos\nu \cos r + 2\dot{l}\dot{p}\ell m \sin\nu \cos\beta \cos r \\
&+ 2\dot{r}^2 l^2 m \sin\beta \sin\nu \cos^2\alpha \cos\nu + L\ddot{\nu}\ell m \sin\alpha \sin\beta \cos\mu \cos\nu_f \\
&+ \ddot{p}c_x \ell m \sin\alpha \sin\beta \sin r \cos\nu + \ddot{p}c_y \ell m \sin\alpha \sin\beta \sin\nu \sin r \\
&+ L\ddot{r}\ell m \sin\alpha \cos\beta \cos\mu \cos(\nu + \nu_f) + g\ell m \sin\alpha \sin\beta \sin\nu \sin r \cos p \\
&+ L\dot{p}^2 \ell m \cos\alpha \cos\mu \cos\nu_f \cos^2 r + L\dot{p}^2 \ell m \sin\alpha \sin\mu \cos\beta \cos^2 r \\
&+ L\dot{r}^2 \ell m \sin\nu \sin(\nu + \nu_f) \cos\alpha \cos\mu + \dot{r}^2 b\ell m \sin\alpha \sin\beta \sin\nu \cos\nu \\
&+ \dot{p}^2 c_z \ell m \sin r \cos\alpha \cos\nu \cos r + \dot{p}^2 h\ell m \sin r \cos\alpha \cos\nu \cos r \\
&+ \dot{p}^2 l^2 m \sin\alpha \cos\alpha \cos^2\beta \cos^2\nu \cos^2 r \\
&- L\ddot{\mu}\ell m \sin\alpha \sin\beta \sin\mu \sin\nu_f - L\ddot{p}\ell m \sin\nu_f \sin r \cos\alpha \cos\mu \\
&- L\ddot{p}\ell m \sin\mu \sin\nu \cos\alpha \cos r - \ddot{p}b\ell m \sin\alpha \sin\nu \cos\beta \cos r \\
&- \ddot{p}c_z \ell m \sin\alpha \sin\beta \cos\nu \cos r - \ddot{p}h\ell m \sin\alpha \sin\beta \cos\nu \cos r \\
&- L\ddot{r}\ell m \sin\alpha \sin\beta \sin\mu \sin\nu - L\dot{\mu}^2 \ell m \sin\alpha \sin\beta \sin\nu_f \cos\mu \\
&- L\dot{\nu_f}^2 \ell m \sin\alpha \sin\beta \sin\nu_f \cos\mu - L\dot{\nu}^2 \ell m \sin\alpha \sin\beta \sin\nu_f \cos\mu \\
&- L\dot{p}^2 \ell m \sin\alpha \sin\beta \sin\nu_f \cos\mu - \dot{p}^2 c_x \ell m \sin\alpha \sin\beta \sin\nu \cos^2 r \\
&- \dot{p}^2 c_x \ell m \sin\alpha \sin r \cos\beta \cos r - 2\dot{\beta}\dot{p}l^2 m \sin\beta \sin\nu \cos^2\alpha \cos r \\
&- 2\dot{\beta}\dot{p}l^2 m \sin\alpha \cos\alpha \cos\nu \cos r - 2L\dot{\nu_f}\dot{\nu}\ell m \cos\alpha \cos\mu \cos\nu_f \\
&- 2\dot{\nu}\dot{p}l^2 m \cos^2\alpha \cos\beta \cos\nu \cos r - 2\dot{\nu}\dot{r}b\ell m \sin\alpha \sin\nu \cos\beta \\
&- 2\dot{p}\dot{r}l^2 m \sin\nu \sin r \cos^2\alpha \cos\beta \\
&- 2\dot{p}^2 l^2 m \sin\beta \sin\nu \cos^2\alpha \cos\nu \cos^2 r \\
&- 2\dot{p}^2 l^2 m \sin r \cos^2\alpha \cos\beta \cos\nu \cos r + 2L\dot{\mu}\dot{\nu_f}\ell m \sin\mu \sin\nu_f \cos\alpha \\
&+ 2L\dot{\mu}\dot{\nu}\ell m \sin\mu \sin\nu_f \cos\alpha + 2L\dot{\mu}\dot{r}\ell m \cos\alpha \cos\mu \cos\nu \\
&+ 2\dot{\nu}\dot{p}l^2 m \sin\alpha \sin r \cos\alpha \cos^2\beta + 2\dot{p}\dot{r}c_x \ell m \sin\nu \cos\alpha \cos r \\
&+ 2\dot{p}\dot{r}c_z \ell m \sin\nu \sin r \cos\alpha + 2\dot{p}\dot{r}h\ell m \sin\nu \sin r \cos\alpha \\
&+ 4\dot{p}\dot{r}l^2 m \sin\beta \cos^2\alpha \cos^2\nu \cos r \\
&+ L\ddot{p}\ell m \sin\alpha \sin\beta \sin r \cos\mu \cos\nu_f + L\dot{p}^2 \ell m \sin\mu \sin r \cos\alpha \cos\nu \cos r \\
&+ L\dot{r}^2 \ell m \sin\alpha \sin\beta \sin\nu \cos\mu \cos(\nu + \nu_f) \\
&- L\ddot{p}\ell m \sin\alpha \sin(\nu + \nu_f) \cos\beta \cos\mu \cos r \\
&- L\ddot{p}\ell m \sin\alpha \sin\beta \sin\mu \cos\nu \cos r \\
&- L\dot{p}^2 \ell m \sin\nu \sin(\nu + \nu_f) \cos\alpha \cos\mu \cos^2 r \\
&- \dot{p}^2 b\ell m \sin\alpha \sin\beta \sin\nu \cos\nu \cos^2 r - \dot{p}^2 b\ell m \sin\alpha \sin r \cos\beta \cos\nu \cos r \\
&- \dot{p}^2 c_z \ell m \sin\alpha \sin\beta \sin\nu \sin r \cos r - \dot{p}^2 h\ell m \sin\alpha \sin\beta \sin\nu \sin r \cos r \\
&- 4\dot{p}\dot{r}l^2 m \sin\alpha \sin\nu \cos\alpha \cos\nu \cos r \\
&- 2L\dot{\mu}\dot{\nu_f}\ell m \sin\alpha \sin\beta \sin\mu \cos\nu_f \\
&- 2L\dot{\mu}\dot{\nu}\ell m \sin\alpha \sin\beta \sin\mu \cos\nu_f - 2L\dot{\mu}\dot{p}\ell m \sin\nu \cos\alpha \cos\mu \cos r \\
&- 2L\dot{\mu}\dot{r}\ell m \sin\alpha \sin\mu \cos\beta \cos(\nu + \nu_f) \\
&- 2L\dot{\mu}\dot{r}\ell m \sin\alpha \sin\beta \sin\nu \cos\mu
\end{aligned}
$$

$$
\begin{aligned}
&- 2L\dot{\nu}_f\dot{\nu}\ell m \sin\alpha \sin\beta \sin\nu_f \cos\mu \\
&- 2L\dot{\nu}_f\dot{p}\ell m \sin r \cos\alpha \cos\mu \cos\nu_f \\
&- 2L\dot{\nu}_f\dot{r}\ell m \sin\alpha \sin(\nu+\nu_f) \cos\beta \cos\mu \\
&- 2L\dot{\nu}\dot{p}\ell m \sin r \cos\alpha \cos\mu \cos\nu_f - 2\dot{\nu}\dot{p}b\ell m \sin\alpha \cos\beta \cos\nu \cos r \\
&- 2L\dot{\nu}\dot{r}\ell m \sin\alpha \sin(\nu+\nu_f) \cos\beta \cos\mu \\
&- 2\dot{\nu}\dot{r}l^2 m \sin\alpha \sin\beta \cos\alpha \cos\beta \cos\nu \\
&+ 2L\dot{\mu}\dot{p}\ell m \sin\mu \sin\nu_f \sin r \cos\alpha + 2L\dot{p}\dot{r}\ell m \sin\mu \sin\nu \sin r \cos\alpha \\
&+ 2\dot{p}\dot{r}b\ell m \sin\alpha \sin\beta \cos^2\nu \cos r + 2\dot{p}\dot{r}b\ell m \sin\nu \cos\alpha \cos\nu \cos r \\
&+ 2\dot{p}\dot{r}c_x\ell m \sin\alpha \sin\beta \cos\nu \cos r + 2\dot{p}\dot{r}c_z\ell m \sin\alpha \sin\beta \sin r \cos\nu \\
&+ 2\dot{p}\dot{r}h\ell m \sin\alpha \sin\beta \sin r \cos\nu \\
&- L\dot{p}^2\ell m \sin\alpha \sin\beta \sin\nu \cos\mu \cos^2 r \cos(\nu+\nu_f) \\
&- L\dot{p}^2\ell m \sin\alpha \sin r \cos\beta \cos\mu \cos r \cos(\nu+\nu_f) \\
&- L\dot{p}^2\ell m \sin\alpha \sin\beta \sin\mu \sin\nu \sin r \cos r \\
&- 2L\dot{\mu}\dot{p}\ell m \sin\alpha \sin\beta \cos\mu \cos\nu \cos r \\
&- 2L\dot{\mu}\dot{p}\ell m \sin\alpha \sin\beta \sin\mu \sin r \cos\nu_f \\
&- 2L\dot{\nu}_f\dot{p}\ell m \sin\alpha \cos\beta \cos\mu \cos r \cos(\nu+\nu_f) \\
&- 2L\dot{\nu}_f\dot{p}\ell m \sin\alpha \sin\beta \sin\nu_f \sin r \cos\mu \\
&- 2L\dot{\nu}\dot{p}\ell m \sin\alpha \cos\beta \cos\mu \cos r \cos(\nu+\nu_f) \\
&- 2L\dot{\nu}\dot{p}\ell m \sin\alpha \sin\beta \sin\nu_f \sin r \cos\mu \\
&- 2\dot{p}\dot{r}l^2 m \sin\alpha \sin\beta \sin r \cos\alpha \cos\beta \cos\nu \\
&+ 2L\dot{\mu}\dot{p}\ell m \sin\alpha \sin\mu \sin(\nu+\nu_f) \cos\beta \cos r \\
&+ 2\dot{\nu}\dot{p}l^2 m \sin\alpha \sin\beta \sin\nu \cos\alpha \cos\beta \cos r \\
&+ 2L\dot{p}\dot{r}\ell m \sin\nu \cos\alpha \cos\mu \cos r \cos(\nu+\nu_f) \\
&+ 2L\dot{p}\dot{r}\ell m \sin\alpha \sin\beta \cos\mu \cos\nu_f \cos r \\
&+ 2L\dot{p}\dot{r}\ell m \sin\alpha \sin\beta \sin\mu \sin r \cos\nu \\
&+ 2\dot{p}\dot{r}l^2 m \sin\alpha \sin\nu \cos\alpha \cos^2\beta \cos\nu \cos r \\
&+ 2\dot{p}^2 l^2 m \sin\alpha \sin\beta \sin\nu \sin r \cos\alpha \cos\beta \cos r \\
&- 2L\dot{p}\dot{r}\ell m \sin\alpha \sin\beta \sin\nu \sin(\nu+\nu_f) \cos\mu \cos r, \\
F_{L,2} = &-\ddot{r}l^2 m \sin\nu \cos^2\alpha - 2\dot{\beta}\dot{l}\ell m \cos^2\alpha + \ddot{\nu}l^2 m \sin\alpha \cos\alpha \cos\beta \\
&+ \ddot{r}c_x\ell m \sin\beta \cos\alpha + \dot{r}^2 c_z\ell m \sin\beta \cos\alpha + \dot{r}^2 h\ell m \sin\beta \cos\alpha \\
&+ \dot{\nu}^2 l^2 m \sin\beta \cos^2\alpha \cos\beta + \dot{p}^2 l^2 m \sin\beta \cos^2\alpha \cos\beta \\
&- \ddot{\nu}b\ell m \cos\alpha \cos\beta - \ddot{p}l^2 m \cos^2\alpha \cos\nu \cos r \\
&- 2\dot{\nu}\dot{r}l^2 m \cos^2\alpha \cos\nu - 2\dot{l}\dot{r}\ell m \sin\nu \cos^2\alpha \\
&+ 2\dot{\alpha}\dot{\beta}l^2 m \sin\alpha \cos\alpha + 2\dot{\alpha}\dot{\nu}l^2 m \cos^2\alpha \cos\beta \\
&+ \ddot{p}l^2 m \sin\alpha \sin r \cos\alpha \cos\beta + \ddot{r}b\ell m \sin\beta \cos\alpha \cos\nu \\
&+ \ddot{r}c_z\ell m \sin\nu \cos\alpha \cos\beta + \ddot{r}h\ell m \sin\nu \cos\alpha \cos\beta \\
&+ L\dot{\mu}^2\ell m \sin\beta \sin\mu \cos\alpha + L\dot{r}^2\ell m \sin\beta \sin\mu \cos\alpha \\
&+ \dot{p}^2 c_y\ell m \cos\alpha \cos\beta \cos\nu + \dot{p}^2 c_z\ell m \sin\beta \cos\alpha \cos^2 r
\end{aligned}
$$

$$
\begin{aligned}
&+ \dot{p}^2 h\ell m \sin\beta \cos\alpha \cos^2 r + \dot{p}^2 l^2 m \sin\nu \sin r \cos^2\alpha \cos r \\
&- L\ddot{\mu}\ell m \sin\beta \cos\alpha \cos\mu - \ddot{p}b\ell m \sin r \cos\alpha \cos\beta \\
&- \ddot{p}c_y \ell m \sin\beta \cos\alpha \cos r - \ddot{r}l^2 m \sin\alpha \sin\beta \cos\alpha \cos\nu \\
&- g\ell m \sin p \cos\alpha \cos\beta \cos\nu - g\ell m \sin\beta \cos\alpha \cos p \cos r \\
&- \dot{p}^2 c_x \ell m \sin\nu \cos\alpha \cos\beta - \dot{r}^2 c_x \ell m \sin\nu \cos\alpha \cos\beta \\
&- \dot{r}^2 l^2 m \sin\beta \cos^2\alpha \cos\beta \cos^2\nu - 2\dot{\alpha}\dot{r}l^2 m \sin\beta \cos^2\alpha \cos\nu \\
&- 2\dot{l}\dot{p}\ell m \cos^2\alpha \cos\nu \cos r - 2\dot{p}^2 l^2 m \sin\beta \cos^2\alpha \cos\beta \cos^2 r \\
&+ 2\dot{\alpha}\dot{p}l^2 m \sin r \cos^2\alpha \cos\beta + 2\dot{\alpha}\dot{r}l^2 m \sin\alpha \sin\nu \cos\alpha \\
&+ 2\dot{\nu}\dot{l}\ell m \sin\alpha \cos\alpha \cos\beta + 2\dot{\nu}\dot{p}l^2 m \sin\nu \cos^2\alpha \cos r \\
&+ 2\dot{\nu}\dot{r}l^2 m \cos^2\alpha \cos^2\beta \cos\nu + L\ddot{\mu}\ell m \sin\mu \sin\nu_f \cos\alpha \cos\beta \\
&+ \ddot{p}c_z \ell m \cos\alpha \cos\beta \cos\nu \cos r + \ddot{p}h\ell m \cos\alpha \cos\beta \cos\nu \cos r \\
&+ \ddot{p}l^2 m \sin\alpha \sin\beta \sin\nu \cos\alpha \cos r + L\ddot{r}\ell m \sin\mu \sin\nu \cos\alpha \cos\beta \\
&+ L\ddot{r}\ell m \sin\beta \cos\alpha \cos\mu \cos(\nu + \nu_f) \\
&+ L\dot{\mu}^2 \ell m \sin\nu_f \cos\alpha \cos\beta \cos\mu + L\dot{\nu_f}^2 \ell m \sin\nu_f \cos\alpha \cos\beta \cos\mu \\
&+ L\dot{\nu}^2 \ell m \sin\nu_f \cos\alpha \cos\beta \cos\mu + L\dot{p}^2 \ell m \sin\beta \sin\mu \cos\alpha \cos^2 r \\
&+ L\dot{p}^2 \ell m \sin\nu_f \cos\alpha \cos\beta \cos\mu + \dot{p}^2 c_x \ell m \sin\nu \cos\alpha \cos\beta \cos^2 r \\
&+ \dot{p}^2 l^2 m \sin\beta \cos^2\alpha \cos\beta \cos^2\nu \cos^2 r \\
&+ \dot{r}^2 l^2 m \sin\alpha \sin\nu \cos\alpha \cos\beta \cos\nu - L\ddot{\nu}\ell m \cos\alpha \cos\beta \cos\mu \cos\nu_f \\
&- \ddot{p}b\ell m \sin\beta \sin\nu \cos\alpha \cos r - \ddot{p}c_x \ell m \sin r \cos\alpha \cos\beta \cos\nu \\
&- \ddot{p}c_y \ell m \sin\nu \sin r \cos\alpha \cos\beta - g\ell m \sin\nu \sin r \cos\alpha \cos\beta \cos p \\
&- \dot{r}^2 b\ell m \sin\nu \cos\alpha \cos\beta \cos\nu - \dot{p}^2 c_x \ell m \sin\beta \sin r \cos\alpha \cos r \\
&- 2\dot{\nu}\dot{p}l^2 m \sin\nu \cos^2\alpha \cos^2\beta \cos r - 2\dot{\nu}\dot{r}b\ell m \sin\beta \sin\nu \cos\alpha \\
&- 2\dot{l}\dot{r}\ell m \sin\alpha \sin\beta \cos\alpha \cos\nu \\
&- 2\dot{p}^2 l^2 m \sin\nu \sin r \cos^2\alpha \cos^2\beta \cos r \\
&+ 2\dot{\alpha}\dot{p}l^2 m \sin\beta \sin\nu \cos^2\alpha \cos r \\
&+ 2\dot{\alpha}\dot{p}l^2 m \sin\alpha \cos\alpha \cos\nu \cos r \\
&+ 2\dot{\nu}\dot{p}l^2 m \sin\beta \sin r \cos^2\alpha \cos\beta \\
&+ 2\dot{\nu}\dot{r}l^2 m \sin\alpha \sin\beta \sin\nu \cos\alpha + 2\dot{l}\dot{p}\ell m \sin\alpha \sin r \cos\alpha \cos\beta \\
&+ 2\dot{p}\dot{r}l^2 m \sin r \cos^2\alpha \cos^2\beta \cos\nu + L\ddot{p}\ell m \sin\mu \cos\alpha \cos\beta \cos\nu \cos r \\
&+ \dot{p}^2 b\ell m \sin\nu \cos\alpha \cos\beta \cos\nu \cos^2 r \\
&+ \dot{p}^2 c_z \ell m \sin\nu \sin r \cos\alpha \cos\beta \cos r + \dot{p}^2 h\ell m \sin\nu \sin r \cos\alpha \cos\beta \cos r \\
&+ \dot{p}^2 l^2 m \sin\alpha \sin\beta \sin r \cos\alpha \cos\nu \cos r \\
&- L\ddot{p}\ell m \sin r \cos\alpha \cos\beta \cos\mu \cos\nu_f \\
&- L\ddot{p}\ell m \sin\beta \sin(\nu + \nu_f) \cos\alpha \cos\mu \cos r \\
&- L\dot{r}^2 \ell m \sin\nu \cos\alpha \cos\beta \cos\mu \cos(\nu + \nu_f) \\
&- \dot{p}^2 b\ell m \sin\beta \sin r \cos\alpha \cos\nu \cos r \\
&- \dot{p}^2 l^2 m \sin\alpha \sin\nu \cos\alpha \cos\beta \cos\nu \cos^2 r
\end{aligned}
$$

$$
\begin{aligned}
&- 2L\dot{\mu}\dot{r}\ell m \sin\beta \sin\mu \cos\alpha \cos(\nu+\nu_f) \\
&- 2L\dot{\nu_f}\dot{r}\ell m \sin\beta \sin(\nu+\nu_f) \cos\alpha \cos\mu \\
&- 2\dot{\nu}\dot{p}b\ell m \sin\beta \cos\alpha \cos\nu \cos r \\
&- 2L\dot{\nu}\dot{r}\ell m \sin\beta \sin(\nu+\nu_f) \cos\alpha \cos\mu \\
&- 2\dot{p}\dot{r}b\ell m \cos\alpha \cos\beta \cos^2\nu \cos r - 2\dot{p}\dot{r}c_x\ell m \cos\alpha \cos\beta \cos\nu \cos r \\
&- 2\dot{p}\dot{r}c_z\ell m \sin r \cos\alpha \cos\beta \cos\nu - 2\dot{p}\dot{r}h\ell m \sin r \cos\alpha \cos\beta \cos\nu \\
&+ 2L\dot{\mu}\dot{\nu_f}\ell m \sin\mu \cos\alpha \cos\beta \cos\nu_f \\
&+ 2L\dot{\mu}\dot{\nu}\ell m \sin\mu \cos\alpha \cos\beta \cos\nu_f + 2L\dot{\mu}\dot{r}\ell m \sin\nu \cos\alpha \cos\beta \cos\mu \\
&+ 2L\dot{\nu_f}\dot{\nu}\ell m \sin\nu_f \cos\alpha \cos\beta \cos\mu \\
&+ 2\dot{\nu}\dot{p}l^2 m \sin\alpha \sin\beta \cos\alpha \cos\nu \cos r \\
&+ 2\dot{l}\dot{p}\ell m \sin\alpha \sin\beta \sin\nu \cos\alpha \cos r \\
&+ 2\dot{p}\dot{r}l^2 m \sin\alpha \cos\alpha \cos\beta \cos^2\nu \cos r \\
&+ L\dot{p}^2\ell m \sin\nu \cos\alpha \cos\beta \cos\mu \cos^2 r \cos(\nu+\nu_f) \\
&+ L\dot{p}^2\ell m \sin\mu \sin\nu \sin r \cos\alpha \cos\beta \cos r \\
&- L\dot{p}^2\ell m \sin\beta \sin r \cos\alpha \cos\mu \cos r \cos(\nu+\nu_f) \\
&- 2L\dot{\nu_f}\dot{p}\ell m \sin\beta \cos\alpha \cos\mu \cos r \cos(\nu+\nu_f) \\
&- 2L\dot{\nu}\dot{p}\ell m \sin\beta \cos\alpha \cos\mu \cos r \cos(\nu+\nu_f) \\
&- 2L\dot{p}\dot{r}\ell m \cos\alpha \cos\beta \cos\mu \cos\nu_f \cos r \\
&- 2L\dot{p}\dot{r}\ell m \sin\mu \sin r \cos\alpha \cos\beta \cos\nu \\
&+ 2L\dot{\mu}\dot{p}\ell m \cos\alpha \cos\beta \cos\mu \cos\nu \cos r \\
&+ 2L\dot{\mu}\dot{p}\ell m \sin\mu \sin r \cos\alpha \cos\beta \cos\nu_f \\
&+ 2L\dot{\mu}\dot{p}\ell m \sin\beta \sin\mu \sin(\nu+\nu_f) \cos\alpha \cos r \\
&+ 2L\dot{\nu_f}\dot{p}\ell m \sin\nu_f \sin r \cos\alpha \cos\beta \cos\mu \\
&+ 2L\dot{\nu}\dot{p}\ell m \sin\nu_f \sin r \cos\alpha \cos\beta \cos\mu \\
&+ 2\dot{p}\dot{r}l^2 m \sin\beta \sin\nu \cos^2\alpha \cos\beta \cos\nu \cos r \\
&+ 2L\dot{p}\dot{r}\ell m \sin\nu \sin(\nu+\nu_f) \cos\alpha \cos\beta \cos\mu \cos r, \\
F_{L,3} = &-\dot{\nu_f}c_f - \nu_f k_f - L^2\ddot{\nu}m\cos^2\mu - L^2\ddot{\nu}m_b\cos^2\mu \\
&- L^2\ddot{p}m \sin r \cos^2\mu - L^2\ddot{p}m_b \sin r \cos^2\mu \\
&+ L\ddot{r}c_z m \sin(\nu+\nu_f)\cos\mu + L\ddot{r}hm\sin(\nu+\nu_f)\cos\mu \\
&+ L^2\ddot{r}m\sin\mu\sin(\nu+\nu_f)\cos\mu + L\ddot{r}c_z m_b\sin(\nu+\nu_f)\cos\mu \\
&+ L\ddot{r}hm_b\sin(\nu+\nu_f)\cos\mu + L^2\ddot{r}m_b\sin\mu\sin(\nu+\nu_f)\cos\mu \\
&+ L\dot{p}^2c_y m\cos\mu\cos(\nu+\nu_f) + L\dot{p}^2c_y m_b\cos\mu\cos(\nu+\nu_f) \\
&- L\ddot{\nu}bm\cos\mu\cos\nu_f - L\ddot{\nu}bm_b\cos\mu\cos\nu_f - L\ddot{l}m\sin\alpha\sin\nu_f\cos\mu \\
&- Lgm\sin p\cos\mu\cos(\nu+\nu_f) - Lgm_b\sin p\cos\mu\cos(\nu+\nu_f) \\
&- L\dot{\nu}^2bm\sin\nu_f\cos\mu - L\dot{p}^2bm\sin\nu_f\cos\mu - L\dot{r}^2bm\sin\nu_f\cos\mu \\
&- L\dot{p}^2c_x m\sin(\nu+\nu_f)\cos\mu - L\dot{r}^2c_x m\sin(\nu+\nu_f)\cos\mu \\
&- L^2\dot{r}^2m\sin\nu\cos^2\mu\cos\nu - L^2\dot{r}^2m\sin\nu_f\cos^2\mu\cos\nu_f \\
&- L\dot{\nu}^2bm_b\sin\nu_f\cos\mu - L\dot{p}^2bm_b\sin\nu_f\cos\mu - L\dot{r}^2bm_b\sin\nu_f\cos\mu
\end{aligned}
$$

$$
\begin{aligned}
&- L\dot{p}^2 c_x m_b \sin(\nu+\nu_f)\cos\mu - L\dot{r}^2 c_x m_b \sin(\nu+\nu_f)\cos\mu \\
&- L^2\dot{r}^2 m_b \sin\nu\cos^2\mu\cos\nu - L^2\dot{r}^2 m_b \sin\nu_f\cos^2\mu\cos\nu_f \\
&+ 2L^2\dot{\mu}\dot{\nu_f} m\sin\mu\cos\mu + 2L^2\dot{\mu}\dot{\nu_f} m_b\sin\mu\cos\mu \\
&+ 2L^2\dot{\mu}\dot{\nu} m\sin\mu\cos\mu + 2L^2\dot{\mu}\dot{\nu} m_b\sin\mu\cos\mu \\
&+ 2L^2\dot{\mu}\dot{r} m\sin(\nu+\nu_f)\cos^2\mu \\
&+ 2L^2\dot{\mu}\dot{r} m_b\sin(\nu+\nu_f)\cos^2\mu + 2L^2\dot{p}\dot{r} m\cos^2\mu\cos r \\
&+ 2L^2\dot{p}\dot{r} m_b\cos^2\mu\cos r + L\ddot{\nu}\ell m\sin\alpha\cos\mu\cos\nu_f \\
&+ L\ddot{p}c_z m\cos\mu\cos r\cos(\nu+\nu_f) + L\ddot{p}h m\cos\mu\cos r\cos(\nu+\nu_f) \\
&+ L^2\ddot{p}m\sin\mu\cos\mu\cos r\cos(\nu+\nu_f) \\
&+ L\ddot{p}c_z m_b\cos\mu\cos r\cos(\nu+\nu_f) + L\ddot{p}h m_b\cos\mu\cos r\cos(\nu+\nu_f) \\
&+ L^2\ddot{p}m_b\sin\mu\cos\mu\cos r\cos(\nu+\nu_f) + L\dot{\alpha}^2\ell m\sin\alpha\sin\nu_f\cos\mu \\
&+ L\dot{\nu}^2\ell m\sin\alpha\sin\nu_f\cos\mu + L\dot{p}^2\ell m\sin\alpha\sin\nu_f\cos\mu \\
&+ L\dot{r}^2\ell m\sin\alpha\sin\nu_f\cos\mu + L\dot{p}^2 bm\sin\nu_f\cos\mu\cos^2 r \\
&+ L\dot{p}^2 c_x m\sin(\nu+\nu_f)\cos\mu\cos^2 r + L^2\dot{p}^2 m\sin\nu\cos^2\mu\cos\nu\cos^2 r \\
&+ L^2\dot{p}^2 m\sin\nu_f\cos^2\mu\cos\nu_f\cos^2 r + L\dot{p}^2 bm_b\sin\nu_f\cos\mu\cos^2 r \\
&+ L\dot{p}^2 c_x m_b\sin(\nu+\nu_f)\cos\mu\cos^2 r \\
&+ L^2\dot{p}^2 m_b\sin\nu\cos^2\mu\cos\nu\cos^2 r \\
&+ L^2\dot{p}^2 m_b\sin\nu_f\cos^2\mu\cos\nu_f\cos^2 r - L\ddot{l}m\sin\beta\cos\alpha\cos\mu\cos\nu_f \\
&- L\ddot{p}bm\sin r\cos\mu\cos\nu_f - L\ddot{p}c_x m\sin r\cos\mu\cos(\nu+\nu_f) \\
&- L\ddot{p}c_y m\sin r\sin(\nu+\nu_f)\cos\mu - L\ddot{p}bm_b\sin r\cos\mu\cos\nu_f \\
&- L\ddot{p}c_x m_b\sin r\cos\mu\cos(\nu+\nu_f) \\
&- L\ddot{p}c_y m_b\sin r\sin(\nu+\nu_f)\cos\mu - Lgm\sin r\sin(\nu+\nu_f)\cos\mu\cos p \\
&- Lgm_b\sin r\sin(\nu+\nu_f)\cos\mu\cos p - L\dot{r}^2 bm\sin\nu\cos\mu\cos(\nu+\nu_f) \\
&- L\dot{r}^2 bm_b\sin\nu\cos\mu\cos(\nu+\nu_f) - 2L\dot{\alpha}\dot{l}m\sin\nu_f\cos\alpha\cos\mu \\
&- 2L^2\dot{p}\dot{r} m\cos^2\mu\cos^2\nu\cos r - 2L^2\dot{p}\dot{r} m\cos^2\mu\cos^2\nu_f\cos r \\
&- 2L^2\dot{p}\dot{r} m_b\cos^2\mu\cos^2\nu\cos r - 2L^2\dot{p}\dot{r} m_b\cos^2\mu\cos^2\nu_f\cos r \\
&+ 2L^2\dot{\mu}\dot{p} m\cos^2\mu\cos r\cos(\nu+\nu_f) + 2L^2\dot{\mu}\dot{p} m\sin\mu\sin r\cos\mu \\
&+ 2L^2\dot{\mu}\dot{p} m_b\cos^2\mu\cos r\cos(\nu+\nu_f) \\
&+ 2L^2\dot{\mu}\dot{p} m_b\sin\mu\sin r\cos\mu + 2L\dot{\nu}\dot{l}m\sin\alpha\cos\mu\cos\nu_f \\
&+ 2L^2\dot{r}^2 m\sin\nu\sin\nu_f\sin(\nu+\nu_f)\cos^2\mu \\
&+ 2L^2\dot{r}^2 m_b\sin\nu\sin\nu_f\sin(\nu+\nu_f)\cos^2\mu \\
&+ L\ddot{p}\ell m\sin\alpha\sin r\cos\mu\cos\nu_f + L\dot{\alpha}^2\ell m\sin\beta\cos\alpha\cos\mu\cos\nu_f \\
&+ L\dot{\beta}^2\ell m\sin\beta\cos\alpha\cos\mu\cos\nu_f + L\dot{\nu}^2\ell m\sin\beta\cos\alpha\cos\mu\cos\nu_f \\
&+ L\dot{p}^2\ell m\sin\beta\cos\alpha\cos\mu\cos\nu_f \\
&+ L\dot{r}^2\ell m\sin\alpha\sin\nu\cos\mu\cos(\nu+\nu_f) \\
&+ L\dot{p}^2 bm\sin\nu\cos\mu\cos^2 r\cos(\nu+\nu_f) \\
&+ L\dot{p}^2 c_z m\sin r\sin(\nu+\nu_f)\cos\mu\cos r
\end{aligned}
$$

$$
\begin{aligned}
&+ L\dot{p}^2 hm \sin r \sin(\nu + \nu_f) \cos\mu \cos r \\
&+ L^2 \dot{p}^2 m \sin\mu \sin r \sin(\nu + \nu_f) \cos\mu \cos r \\
&+ L\dot{p}^2 b m_b \sin\nu \cos\mu \cos^2 r \cos(\nu + \nu_f) \\
&+ L\dot{p}^2 c_z m_b \sin r \sin(\nu + \nu_f) \cos\mu \cos r \\
&+ L\dot{p}^2 h m_b \sin r \sin(\nu + \nu_f) \cos\mu \cos r \\
&+ L^2 \dot{p}^2 m_b \sin\mu \sin r \sin(\nu + \nu_f) \cos\mu \cos r \\
&- L\ddot{\nu}\ell m \sin\beta \sin\nu_f \cos\alpha \cos\mu \\
&- L\ddot{r}\ell m \sin(\nu + \nu_f) \cos\alpha \cos\beta \cos\mu \\
&- L\dot{p}^2 \ell m \sin\alpha \sin\nu_f \cos\mu \cos^2 r - 2L\dot{\beta}\dot{l} m \cos\alpha \cos\beta \cos\mu \cos\nu_f \\
&- 2L\dot{\nu}\dot{l} m \sin\beta \sin\nu_f \cos\alpha \cos\mu - 2L\dot{\nu}\dot{p} b m \sin\nu_f \sin r \cos\mu \\
&- 2L\dot{\nu}\dot{p} b m_b \sin\nu_f \sin r \cos\mu \\
&- 2L\dot{l}\dot{r} m \sin(\nu + \nu_f) \cos\alpha \cos\beta \cos\mu - 2L\dot{p}\dot{r} b m \cos\mu \cos\nu_f \cos r \\
&- 2L\dot{p}\dot{r} c_x m \cos\mu \cos r \cos(\nu + \nu_f) \\
&- 2L\dot{p}\dot{r} c_z m \sin r \cos\mu \cos(\nu + \nu_f) \\
&- 2L\dot{p}\dot{r} h m \sin r \cos\mu \cos(\nu + \nu_f) \\
&- 2L^2 \dot{p}\dot{r} m \sin\mu \sin r \cos\mu \cos(\nu + \nu_f) - 2L\dot{p}\dot{r} b m_b \cos\mu \cos\nu_f \cos r \\
&- 2L\dot{p}\dot{r} c_x m_b \cos\mu \cos r \cos(\nu + \nu_f) \\
&- 2L\dot{p}\dot{r} c_z m_b \sin r \cos\mu \cos(\nu + \nu_f) \\
&- 2L\dot{p}\dot{r} h m_b \sin r \cos\mu \cos(\nu + \nu_f) \\
&- 2L^2 \dot{p}\dot{r} m_b \sin\mu \sin r \cos\mu \cos(\nu + \nu_f) \\
&- 2L^2 \dot{p}^2 m \sin\nu \sin\nu_f \sin(\nu + \nu_f) \cos^2\mu \cos^2 r \\
&- 2L^2 \dot{p}^2 m_b \sin\nu \sin\nu_f \sin(\nu + \nu_f) \cos^2\mu \cos^2 r \\
&+ 2L\dot{\alpha}\dot{\nu}\ell m \cos\alpha \cos\mu \cos\nu_f + 2L\dot{\alpha}\dot{l} m \sin\alpha \sin\beta \cos\mu \cos\nu_f \\
&+ 2L\dot{l}\dot{p} m \sin\alpha \sin r \cos\mu \cos\nu_f \\
&+ L\dot{r}^2 \ell m \sin\beta \sin\nu \sin(\nu + \nu_f) \cos\alpha \cos\mu \\
&- L\ddot{p}\ell m \cos\alpha \cos\beta \cos\mu \cos r \cos(\nu + \nu_f) \\
&- L\ddot{p}\ell m \sin\beta \sin\nu_f \sin r \cos\alpha \cos\mu \\
&- L\dot{p}^2 \ell m \sin\alpha \sin\nu \cos\mu \cos^2 r \cos(\nu + \nu_f) \\
&- 2L\dot{\beta}\dot{\nu}\ell m \sin\nu_f \cos\alpha \cos\beta \cos\mu \\
&- 2L\dot{l}\dot{p} m \cos\alpha \cos\beta \cos\mu \cos r \cos(\nu + \nu_f) \\
&- 2L\dot{l}\dot{p} m \sin\beta \sin\nu_f \sin r \cos\alpha \cos\mu \\
&+ 2L\dot{\alpha}\dot{\beta}\ell m \sin\alpha \cos\beta \cos\mu \cos\nu_f \\
&+ 2L\dot{\alpha}\dot{\nu}\ell m \sin\alpha \sin\beta \sin\nu_f \cos\mu \\
&+ 2L\dot{\alpha}\dot{p}\ell m \sin r \cos\alpha \cos\mu \cos\nu_f \\
&+ 2L\dot{\alpha}\dot{r}\ell m \sin\alpha \sin(\nu + \nu_f) \cos\beta \cos\mu \\
&+ 2L\dot{\beta}\dot{r}\ell m \sin\beta \sin(\nu + \nu_f) \cos\alpha \cos\mu \\
&+ 2L\dot{\nu}\dot{p}\ell m \sin\alpha \sin\nu_f \sin r \cos\mu + 2L\dot{p}\dot{r}\ell m \sin\alpha \cos\mu \cos\nu_f \cos r \\
&+ 2L\dot{p}\dot{r} b m \sin\nu \sin(\nu + \nu_f) \cos\mu \cos r
\end{aligned}
$$

$$
\begin{aligned}
&+ 2L\dot{p}\dot{r}bm_b \sin\nu \sin(\nu+\nu_f)\cos\mu\cos r \\
&+ 4L^2\dot{p}\dot{r}m \sin\nu \sin\nu_f \cos^2\mu \cos r \cos(\nu+\nu_f) \\
&+ 4L^2\dot{p}\dot{r}m_b \sin\nu \sin\nu_f \cos^2\mu \cos r \cos(\nu+\nu_f) \\
&- L\dot{p}^2\ell m \sin\beta \sin\nu \sin(\nu+\nu_f)\cos\alpha\cos\mu\cos^2 r \\
&- L\dot{p}^2\ell m \sin r \sin(\nu+\nu_f)\cos\alpha\cos\beta\cos\mu\cos r \\
&- 2L\dot{\beta}\dot{p}\ell m \sin\nu_f \sin r \cos\alpha\cos\beta\cos\mu \\
&- 2L\dot{p}\dot{r}\ell m \sin\alpha\sin\nu\sin(\nu+\nu_f)\cos\mu\cos r \\
&+ 2L\dot{\alpha}\dot{p}\ell m \sin\alpha\cos\beta\cos\mu\cos r\cos(\nu+\nu_f) \\
&+ 2L\dot{\alpha}\dot{p}\ell m \sin\alpha\sin\beta\sin\nu_f\sin r\cos\mu \\
&+ 2L\dot{\beta}\dot{p}\ell m \sin\beta\cos\alpha\cos\mu\cos r\cos(\nu+\nu_f) \\
&+ 2L\dot{\nu}\dot{p}\ell m \sin\beta\sin r\cos\alpha\cos\mu\cos\nu_f \\
&+ 2L\dot{p}\dot{r}\ell m \sin r\cos\alpha\cos\beta\cos\mu\cos(\nu+\nu_f) \\
&+ 2L\dot{p}\dot{r}\ell m \sin\beta\sin\nu\cos\alpha\cos\mu\cos r\cos(\nu+\nu_f).
\end{aligned}
$$

Appendix B

Symbolic Dynamics Calculator

To eliminate the tedious and error-prone nature of manual analytical derivations, a framework was developed in Python using the `SymPy` symbolic mathematics library [46]. The framework automates the kinematic chain configuration, frame transformations, velocity derivations, and energy formulations required to compute the complex system matrices across diverse operational configurations. Ultimately, it utilizes an automated code-generation pipeline to export optimized, execution-ready equations directly compatible with Python, MATLAB/Simulink, and LaTeX typesetting engines.

B.1 Kinematic Chain and Reference Frame Architecture

The software structures the vessel–crane–load system as a series of sequential reference frames and relative position vectors, following the configuration established in Section 3.3 and Section 4.2. The four distinct reference systems (Inertial Frame, Vessel Frame, Crane Frame, and Load Frame) are implemented using the `ReferenceFrame` class within SymPy's mechanics module.

Position vectors mapping the distances between joint links are defined using symbolic parameters. The kinematic chain starts at the inertial origin O, maps to the vessel-mounted crane base via offset constants (c_x, c_y, c_z), continues through the vertical mast height (h) and mast-boom offset (b), tracks the boom length (L) with the structural deflection angle (ν_f), and terminates at the lower-block via the wire rope length (ℓ). Linear and angular velocities are automatically resolved by evaluating symbolic time derivatives relative to the designated inertial frame using build-in differentiation methods.

B.2 Multi-Configurational Design (Dynamics Level)

To verify the symbolic derivation pipeline, the calculator implements a modular, scalable architecture governed by a `DynamicsLevel` enumeration. This allows the software to scale systematically from a basic static pendulum up to the full vessel-mounted flexible-boom model:

```
class DynamicsLevel(Enum):
    StaticBasePendulum = 0
    StaticBaseCrane = 1
    StaticPitchRoll = 2
    Vessel = 3
    StaticBaseCraneFlexibleBoom = 4
    StaticPitchRollFlexibleBoom = 5
    VesselFlexibleBoom = 6
```

Conditional logic within the framework determines whether parameters such as luffing (μ), slewing (ν), rope length (ℓ), and vessel wave states (p, r) are treated as static constants or time-variant `dynamicsymbols`. This capability enables the validation of individual mechanical phenomena during system development.

B.3 Symbolic Matrix Extraction

Once the positions and velocities are fully established, the framework calls a generalized Euler-Lagrange solver method. This method formulates the global system Lagrangian by isolating the kinetic and potential energies of both the lower-block point mass and the concentrated boom tip mass.

Rather than executing a computationally expensive analytical matrix inversion symbolically, the calculator extracts and isolates the equations directly into a configuration-dependent mass matrix $\mathbf{M}(\mathbf{q})$ and a generalized forcing vector $\mathbf{F}(\mathbf{q}, \dot{\mathbf{q}})$ (see Appendix A.3 for the mathematical details). The resulting symbolic structures are passed to algebraic simplification algorithms, applying SymPy's `trigsimp`, `expand`, and `factor` methods to eliminate redundant trigonometric expressions and collect matching groupings. The finalized matrices are subsequently parsed by an automated translation function that maps pythonic syntax (such as `**` for powers) to target-compliant syntax (such as `^`) while organizing the raw expressions into structured math blocks.

B.4 Computational Optimization via Common Subexpression Elimination

A core contribution of the scripting pipeline is the automation of real-time deployment code preparation. Raw, unoptimized symbolic expressions for the flexible-boom configuration easily expand into thousands of individual floating-point operations. This makes direct numerical evaluation within high-frequency estimation loops computationally heavy due to CPU overhead constraints.

When the factorization pipeline is enabled, the flattened matrix structures are passed into a specialized Common Subexpression Elimination (CSE) method. This optimization algorithm parses the global execution tree to detect repeating algebraic combinations, extracting them into a sequential array of temporary intermediate cache variables which are subsequently substituted back into the terminal system matrices. Crucially, the routine overrides default automated simplification behaviors by filtering out and rejecting trivial replacements where a temporary variable is assigned to a basic constant, a base dynamic symbol, or a first-order time

derivative. This structural filtering reduces the overall floating-point operational count, turning complex symbolic formulations into high-efficiency functions ready for compilation.

Appendix C

MATLAB Implementation Framework

To evaluate the state estimation algorithms under controlled conditions, a simulation and filtering environment was developed within MATLAB/Simulink. This framework is responsible for propagating the continuous-time system dynamics to generate the system ground truth, emulating physical sensors, and executing the discrete state estimation filters.

The entire simulation workspace is initialized and managed via an automated set of auxiliary MATLAB scripts. Prior to model execution, an initialization script defines the structural geometric and mass parameters, compiles the physical sensor variances into a global environment structure, and initializes the filter covariance matrices ($\mathbf{P}_0$, $\mathbf{Q}$, and $\mathbf{R}$). Furthermore, to bridge the gap between symbolic derivation and numerical execution, the complex mass matrices $\mathbf{M}(\mathbf{x})$ and generalized forcing vectors $\mathbf{F}(\mathbf{x})$ generated by the symbolic tool (Appendix B) are automatically parsed into optimized MATLAB code, preventing manual transcription errors.

All continuous-time integrations within the system subsystems are executed using a fixed-step fourth-order Runge-Kutta numerical solver (`ode4`) with a base simulation step size of $\Delta t_{\text{sim}} = 0.005\,\text{s}$ (200 Hz).

C.1 Simulink Subsystem Topology

The Simulink model architecture is structured into modular subsystems to decouple the environmental physics from the discrete estimation algorithms. This modular layout simplifies the debugging process and provides a clean interface for executing comparative tracking studies.

Figure C.1 illustrates the complete top-level data loop inside the Simulink canvas. The process moves sequentially through five major operational phases: continuous state propagation, true 3D lower-block coordinate calculation, sensor emulation, discrete state estimation, and estimated 3D lower-block coordinate calculation.

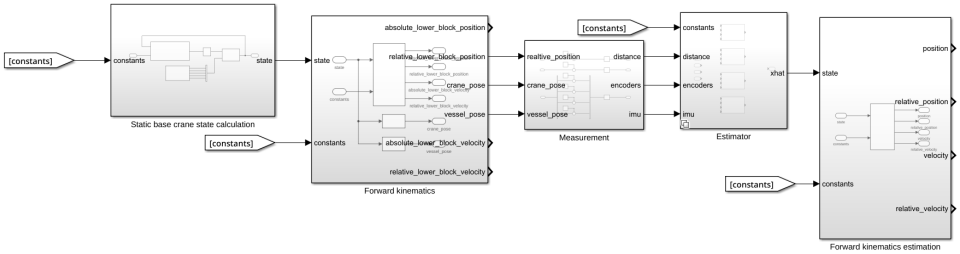


Figure C.1: **Global top-level Simulink architecture, illustrating the routing between the continuous-time plant, true 3D lower-block coordinate calculation, measurement emulation, discrete filter blocks, and estimated 3D lower-block coordinate calculation.**

C.1.1 Ground Truth Data Generation

The ground truth generation subsystem acts as the virtual physical system, computing the continuous-time evolution of the vessel–crane–load system. State propagation relies on the numerical integration of a non-linear, state-space formulation:

$$\dot{\mathbf{x}} = f(\mathbf{x}). \tag{C.1}$$

To ensure code reusability and eliminate any mathematical differences between the true system physics and the internal filter models, an identical state-space structure is utilized across both the ground truth simulation and the estimation algorithms. Within this unified architecture, the global state vector $\mathbf{x}$ is augmented to include the drive and wave accelerations ($\ddot{\mu}$, $\ddot{\nu}$, $\ddot{\ell}$, $\ddot{p}$, $\ddot{r}$) as internal states. While the estimators model these unmeasured acceleration dynamics, the ground truth subsystem handles them deterministically by directly overriding these specific elements inside the state vector with prescribed profiles before evaluating the equations.

As shown in Figure C.2, the subsystem groups these equations within a `Dynamics` block, which can be toggled between the rigid crane model and the flexible-boom model. This block evaluates the system dynamics to output the state derivative vector $\dot{\mathbf{x}}$, which is subsequently passed through a continuous integrator block ($1/s$) to resolve the state vector $\mathbf{x}$ for the next integration step.

Simultaneously, a deterministic `Scenario` block reads predefined time series of the five acceleration profiles from an external file. A localized function block streams these values directly into the integration loop to overwrite the acceleration states, ensuring that the continuous integration is driven by a physically consistent, predefined reference trajectory.

C.1.2 Forward Kinematics and Metric Extracti

Prior to executing the sensor emulation stage, the uncorrupted physical quantities tracked by the sensors must be extracted from the continuous state vector $\mathbf{x}$. This extraction is handled inside the forward kinematics subsystem shown in Figure C.3.

This subsystem contains the structural transformations required to translate the states into physical positions and orientations. An embedded MATLAB function

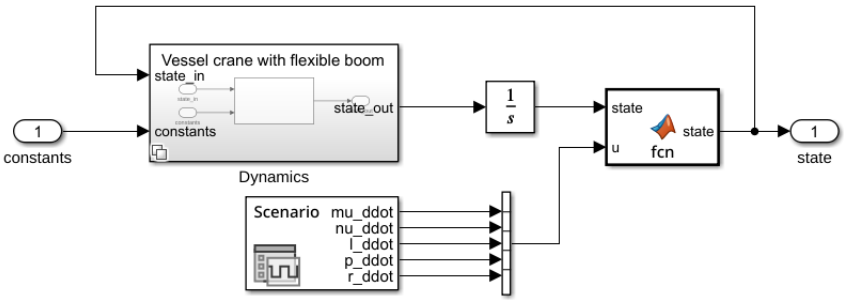


Figure C.2: **Internal topology of the ground truth generation subsystem, showing the continuous integrator feedback loop driven by prescribed input acceleration profiles.**

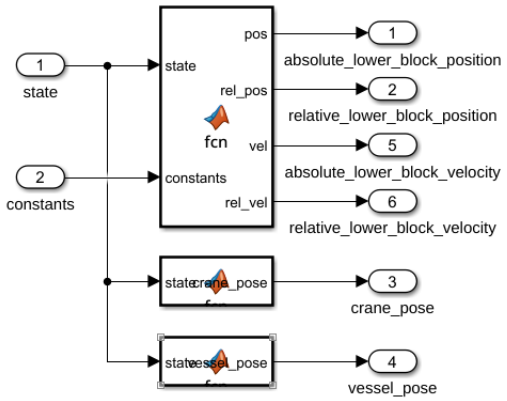


Figure C.3: **Forward kinematics extraction subsystem layout, using embedded MATLAB functions to resolve true 3D Cartesian tracking parameters, crane orientations, and vessel pose from the state vector.**

block computes the real-world 3D Cartesian coordinates of the system, outputting the absolute and relative positions and velocities of the suspended lower-block. Concurrently, the measurement models are used to output the real-time crane pose and vessel pose. These raw, ideal signals serve as the true inputs for the measurement emulation and data-logging components.

C.1.3 Sensor Emulation

The sensor emulation subsystem distorts the measured values to accurately mimic the physical limitations and imperfections of real-world instrumentation. This block models hardware measurement uncertainties and introduces discrete time-sampling bounds.

As illustrated in Figure C.4, the raw position, crane pose, and vessel orientation signals are routed into parallel instrumentation channels. Each channel corrupts the true signal by adding Gaussian noise. The continuous noise profiles are generated using Simulink `Band-Limited White Noise` blocks. To maintain physical

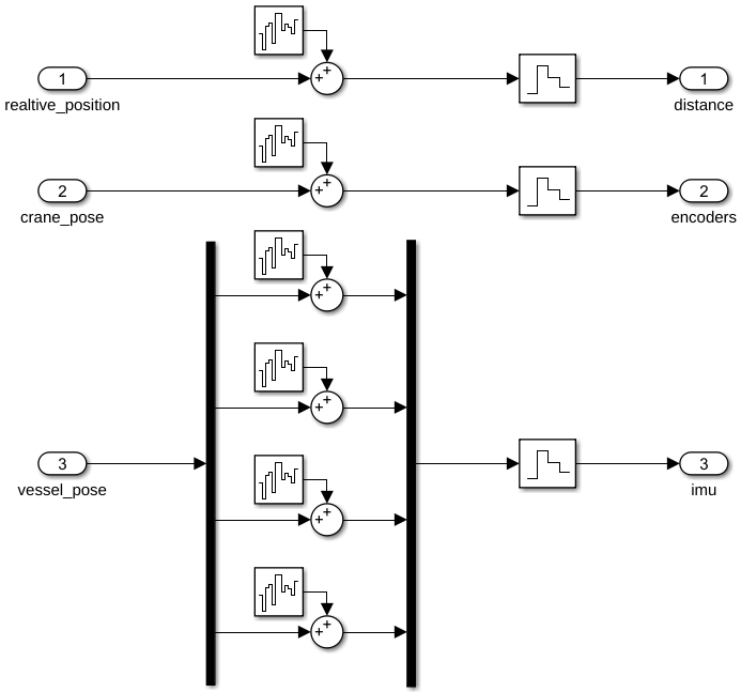


Figure C.4: **Detailed sensor emulation pipeline showing the injection of the Band-Limited White Noise combined with discrete Zero-Order Hold sampling blocks.**

consistency when changes are made to the numerical solver step size, the internal noise power spectral density parameter P is explicitly scaled within the block configuration based on the targeted sensor variance σ^2 and the specific block execution period Δt according to the analytical relationship:

$$P = \sigma^2 \cdot \Delta t. \tag{C.2}$$

Following noise injection, the corrupted continuous signals are passed through `Zero-Order Hold` (ZOH) blocks. These blocks sample the continuous data streams to physical operating frequencies (e.g., 20 Hz or 100 Hz ranges depending on the sensor characteristics), generating the discrete measurement profiles that are ultimately used by the discrete filtering algorithms.

C.1.4 State Estimation Subsystem

The discrete, noisy measurement streams are fed directly into the state estimation subsystem, which contains the implementations of the filtering frameworks.

As depicted in Figure C.5, the subsystem is organized into an itemized model matrix containing four independent filtering configurations: the EKF tracking rigid crane dynamics, the UKF tracking rigid crane dynamics, the EKF tracking flexible-boom dynamics, and the UKF tracking flexible-boom dynamics. This multi-model layout allows the user to route active signals to any desired filter block configuration.

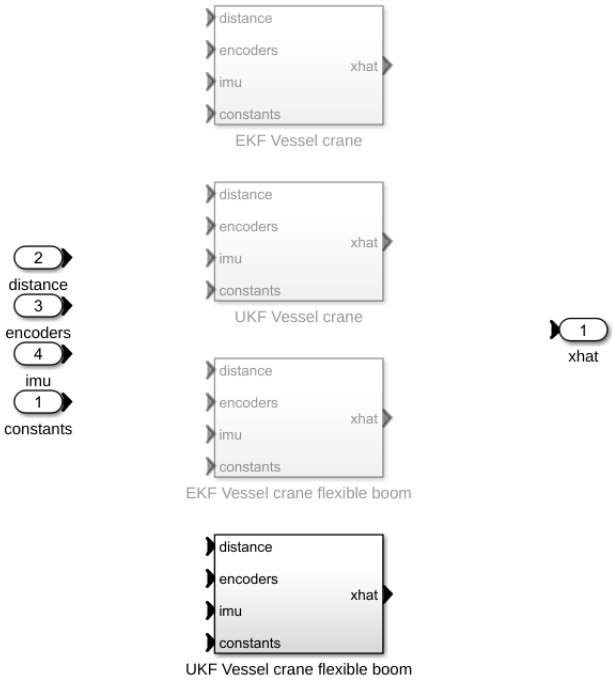


Figure C.5: **State estimation block matrix, showing the modular subsystem routing for evaluating EKF and UKF performance variants under matching or mismatched plant models.**

Crucially, the internal non-linear process model embedded within the filter blocks can be selected independently of the system model selected in the ground truth generation stage (Section C.1.1). This decoupling provides a tool for analyzing filter performance and robustness under conditions of structural model mismatch, such as running a rigid-body estimator configuration while the true system simulation is experiencing elastic boom deflection.

C.1.5 Forward Kinematics of the Estimation

Because the ultimate metrics of interest, namely the positioning accuracy and velocity tracking of the suspended lower-block, are real-world 3D Cartesian coordinates rather than direct entries in the system's state vector, a final reconstruction step is required.

The forward kinematics estimation subsystem maps the discrete, filtered state estimate vector $\hat{\mathbf{x}}$ output by the estimator block back into 3D space. By evaluating the relative frame translations and rotation chains established in Chapter 4, this block computes the estimated absolute inertial position, relative position, absolute velocity, and relative velocity of the lower-block.

C.1.6 Constants and Data Logging Pipelines

To maintain a clean and readable model canvas, auxiliary tasks like parameter routing and dataset compilation are isolated into dedicated support blocks.

Figure C.6 displays the internal layouts of these auxiliary subsystems. Figure C.6(a) highlights the constants vector serialization routine, which bundles individual system parameters (such as gravity g, sample period T_s, mass properties m and m_b, geometric offsets L, b, h, c_x, c_y, c_z, and flexibility coefficients k_f, c_f) into a single structural bus variable. This bus can be distributed cleanly across the canvas without cluttering the model layout with cross-cutting signal lines.

Figure C.6(b) shows the configuration of the data logger subsystem. This block aggregates the true system state trajectories, estimated state streams, and calculated 3D tracking metrics into a unified multi-channel bus. This bus is routed to a localized `To Workspace` (`out.data`) logging block, which caches the time histories in the MATLAB workspace. This cached data is then exported to standardized data formats for analysis and 3D rendering within the Python visualization framework (Appendix D).

C.2 Automation and Post-Processing Pipeline

To ensure complete experimental reproducibility and eliminate manual configuration overhead, the graphical Simulink framework is backed by an automated programmatic scripting pipeline developed in MATLAB. This underlying software layer acts as an execution engine that manages the simulation lifecycle across four distinct operational phases: workspace parameterization, input profile preprocessing, multi-dimensional batch simulation, and transient-filtered metric extraction.

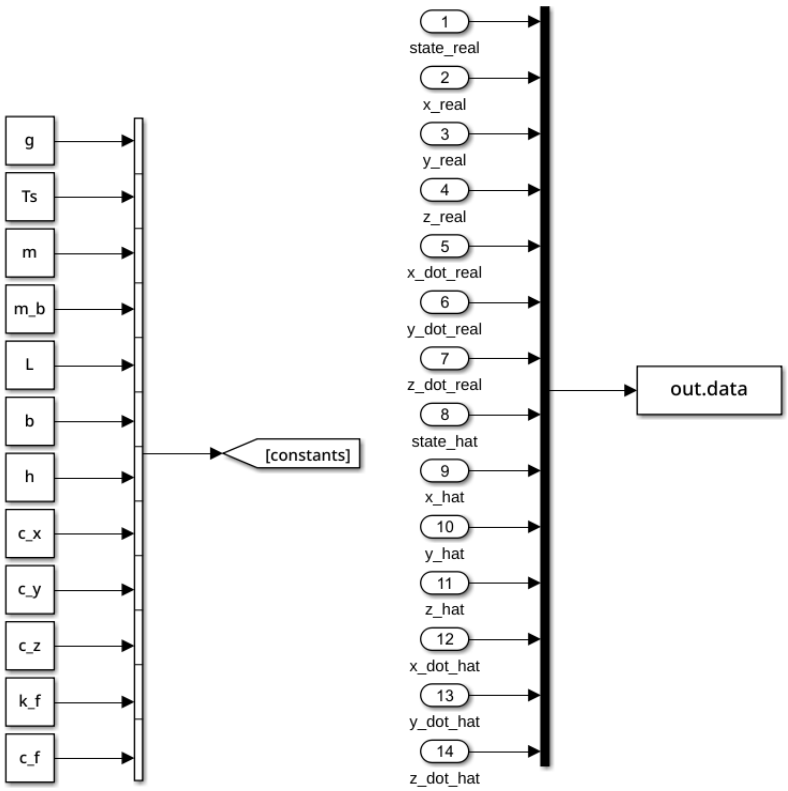
C.2.1 Workspace Initialization and Filter Parameterization

Before the numerical solver executes the model, a centralized initialization script constructs the complete MATLAB workspace environment. Rather than hard-coding physical properties, the script utilizes a structured data-ingestion pipeline that reads standardized JSON scenario files. It parses these configuration profiles to extract the system's geometric constants (L, b, h), mass properties (m, m_b), crane-base coordinate offsets (c_x, c_y, c_z), and localized boom elasticity characteristics (k_f, c_f).

In addition to defining the deterministic plant variables, this initialization script configures the stochastic state-space properties required by the discrete estimators. It sets up the initial error covariance matrix $\mathbf{P}_0$, the measurement noise covariance matrix $\mathbf{R}$, and the discrete-time process noise covariance matrix $\mathbf{Q}$, as explained in Section 4.6.2.

C.2.2 Automated Batch Simulation Execution Engine

To evaluate and compare the state estimators under a wide range of operational conditions, a master batch simulation script was developed. This script replaces



(a) Constants vector serialization. (b) Data logger aggregation bus.

Figure C.6: **Auxiliary subsystems layout, illustrating the automated matrix parameters serialization block and the consolidated real-time dataset logging pipeline.**

manual model execution with a fully automated, multi-loop parameter sweep.

The batch engine contains an execution loop that makes use of four nested testing dimensions:

- **Scenario Configuration Profile:** Iterates through diverse environmental operational paths (e.g., pure hoisting, coupled slewing-luffing, and active multi-axis wave disturbances).
- **System Dynamics Model:** Toggles the true underlying system generation block between the rigid crane configuration and the flexible-boom configuration.
- **Estimator Algorithm Selection:** Switches the active filtering block architecture between the EKF and the UKF.
- **Estimator Process Model Configuration:** Toggles the internal process model embedded within the filter between rigid equations and flexible equations.

During each iteration, the script updates the model's simulation options parameter structure, executes the core command `sim(model_name, options)`, isolates the

resulting time histories from the output structures, and routes them to the post-processing pipeline. This automated execution loop is essential for analyzing filter response and tracking performance under structural model mismatches.

C.2.3 Metric Calculation and Data Serialization

Once a simulation run concludes, the raw time histories are processed by a dedicated statistical analysis script to evaluate tracking performance.

When an estimator is initialized, it typically experiences a brief initialization transient as the state estimates $\hat{\mathbf{x}}$ converge from their initial guesses toward the true system states $\mathbf{x}$. To prevent these large, temporary initial errors from biasing the steady-state performance metrics, the script applies a filter that discards the first 5 seconds of simulation time. Using this dataset, the script computes two primary statistical performance metrics: the RMSE (see (5.1)) and the MaxAE (see (5.2)).

The calculated tracking statistics are compiled into data tables within the MATLAB workspace, allowing for rapid performance comparisons between the EKF and UKF architectures. Finally, an export script flattens the multi-channel data streams into structured, comma-separated value (CSV) files. These files can be used by the Python visualization tool (Appendix D) for 3D trajectory rendering and visual analysis.

Appendix D

Equation of Motion Visualizer Framework

To verify the analytical derivations of the EoMs, a 3D visualization tool was developed in Python. This tool implements the geometric constraints of the vessel-mounted offshore crane system.

The Python framework served two main purposes during development. First, it facilitated the visual validation of whether the derived EoMs exhibited physically consistent behavior under wave disturbances. Because the equations are too complex for direct analytical interpretation, and the implementation of an external physics engine was infeasible within the project's timeline, qualitative visual inspection provided the primary way of ensuring structural and behavioral correctness.

Second, the tool facilitated a direct comparison of ground truth simulations. By executing identical configurations independently in both Python and MATLAB/Simulink, the results from both environments could be cross-checked. Comparing these independent results allowed for isolating mathematical differences and programming errors across both implementations.

D.1 Simulation Configuration

The visualizer configures the simulation parameters by loading standardized JSON scenario profiles (same file as MATLAB/Simulink). A primary challenge in executing these simulations is in balancing the step-size constraints of the numerical integrator against the visual update frequency of the graphics engine. While the graphical frontend operates at a conventional visual refresh frequency (e.g., 25 Hz, corresponding to a frame period of 0.04 s), the underlying physical model requires a substantially smaller time step to maintain stability during numerical integration (e.g., 100 Hz, or 0.01 s).

To resolve this difference, the framework reads the target frames per second (FPS) and the maximum allowable simulation time step from the configuration file. It

then uses a ceiling function to calculate an integer sub-stepping ratio:

$$\Delta t_{\text{vis}} = \frac{1}{\text{target\_fps}}, \tag{D.1}$$

$$\text{substep} = \left\lceil \frac{\Delta t_{\text{vis}}}{\Delta t_{\max}} \right\rceil, \tag{D.2}$$

$$\Delta t_{\text{sim}} = \frac{\Delta t_{\text{vis}}}{\text{substep}}. \tag{D.3}$$

By selecting an adjusted simulation time step that is an integer divisor of the visual playback period, the framework guarantees that each rendered frame aligns precisely with a calculated state vector. This mathematical alignment prevents timing drift and ensures that transient dynamics are captured accurately on screen without requiring artificial data interpolation.

D.2 Numerical Simulation and Integration Pipeline

To bridge the gap between structural parameters and visual rendering, the framework implements a numerical simulation engine. This pipeline handles state tracking, continuous input interpolation, and multi-model configuration loops.

D.2.1 State Space Representation and Input Vector

The dynamic layout of the multi-body vessel–crane–load system is captured by a generalized 16-dimensional state space vector, implemented as an overloaded object container using Python's `NamedTuple` (`State`). The global state vector is explicitly structured as:

$$\mathbf{x} = \begin{bmatrix} \alpha & \dot{\alpha} & \beta & \dot{\beta} & \mu & \dot{\mu} & \nu & \dot{\nu} & \nu_f & \dot{\nu}_f & \ell & \dot{\ell} & p & \dot{p} & r & \dot{r} \end{bmatrix}^T, \tag{D.4}$$

where α and β represent sway angles, μ is the luffing angle, ν is the slewing angle, ν_f is boom deflection, ℓ is the hoisting cable length, and p, r are the wave-induced pitch and roll displacements of the vessel. To keep the integration math concise and readable, the `State` class overrides standard algebraic operators (`__add__`, `__mul__`), permitting direct vector arithmetic during updates.

Input commands are passed into the EoMs via an acceleration input vector (`Inputs`):

$$\mathbf{u} = \begin{bmatrix} \ddot{\mu} & \ddot{\nu} & \ddot{\ell} & \ddot{p} & \ddot{r} \end{bmatrix}^T. \tag{D.5}$$

D.2.2 Runge-Kutta Integration and Input Ingestion

The forward simulation is driven by the `simulate` function, which loops through the discrete time array using a classic fourth-order Runge-Kutta (RK4) algorithm. Because command histories or sea-state data profiles are often logged at arbitrary or variable frequencies, the framework evaluates the input accelerations at each fractional RK4 sub-step (k_1, k_{23}, k_4) using continuous linear interpolation (`np.interp`). The interpolation routine ensures smooth acceleration transitions across high-frequency integration intervals, avoiding artificial step discontinuities.

D.2.3 Model Configurations and JIT Compilation

To allow for modular testing and component isolation, the framework incorporates an enumeration flag (`ODE_Function`) that dynamically selects between six environmental and physical configurations:

- `STATIC_BASE_CRANE`: The crane base is fixed at the Inertial / Vessel origin. The crane can luff, slew, and hoist.
- `STATIC_VESSEL`: The crane base is offset from the Inertial / Vessel origin. In contrast to the previous model, the vessel may have a static pitch and roll.
- `VESSEL`: The `STATIC_VESSEL` model, but now with dynamic pitch and roll.
- `STATIC_BASE_CRANE_FLEXIBLE_BOOM`: The `STATIC_BASE_CRANE` model, but with a flexible boom.
- `STATIC_VESSEL_FLEXIBLE_BOOM`: The `STATIC_VESSEL` model, but with a flexible boom.
- `VESSEL_FLEXIBLE_BOOM`: The `VESSEL` model, but with a flexible boom.

Each selected model variant isolates its explicit structural configurations into an underlying matrix solver function. This function builds up the system's mass matrix $\mathbf{M}(\mathbf{x})$ and force vector $\mathbf{F}(\mathbf{x}, \mathbf{u})$ to solve for the target accelerations of the generalized coordinates, as detailed in Appendix A.3.

Because the symbolic matrix structures are dense and execution speed is vital for interactive processing, these numerical evaluation steps are optimized using Numba's Just-In-Time compiler via the `@njit(cache=True)` decorator. This compiles the complex linear algebra matrix evaluations (`np.linalg.solve`) directly into cached machine code, avoiding Python interpreter overhead.

D.3 Kinematic Pose Reconstruction

The core geometric engine of the visualizer is managed by the `CranePose` class, which maps the state vector into a 3D visual space. Rather than evaluating the dense, fully expanded symbolic expressions output by the automated dynamics calculator (Appendix B), the visualizer reconstructs the system's instantaneous pose step-by-step using the explicit, sequential rotation and translation matrices established in Section 4.2.

Implementing the visual reconstruction via an independent forward kinematics pipeline provides a crucial diagnostic cross-verification mechanism. Because the automated Lagrangian derivation framework yields symbolic expressions that are too complex to verify via manual analytical inspection, driving the 3D graphics with a separate, modular geometric coordinate chain ensures that any structural derivation errors, sign mismatches, or algebraic discrepancies in the EoMs will immediately manifest as unphysical behavior or visual breaks on screen.

D.3.1 Vessel Reference and Base Motion

To provide a basic visual reference for the vessel, a simple rectangular plane is defined in the local vessel frame using estimated length and beam dimensions. To simulate wave-induced deck motions, the coordinates of this rectangle are transformed into the earth-fixed inertial frame at each time step. This is accomplished

by applying the time-varying rotation matrix R_{iv}, which is updated continuously using the instantaneous pitch (p) and roll (r) states:

$$p_i = R_{iv} p_v. \tag{D.6}$$

This straightforward representation avoids unnecessary computational overhead while clearly showing how the rotational movements of the vessel's deck propagate through the crane foundation.

D.3.2 Sequential Kinematic Chain Implementation

Once the moving base frame is established, the framework constructs the physical components of the crane assembly sequentially from the deck upward. The Python architecture chains a series of relative translations and frame rotations to evaluate the exact inertial coordinates of each structural link.

First, the absolute position of the crane base is calculated. The framework transforms the offsets (c_x, c_y, c_z) defined within the local Vessel Frame into absolute inertial coordinates using the vessel-to-inertial transformation:

$$p_{\text{base}} = R_{iv} \begin{bmatrix} c_x \\ c_y \\ c_z \end{bmatrix}. \tag{D.7}$$

Next, the position of the boom tip is resolved. The visualizer accomplishes this by first defining the tip coordinates relative to the local Crane Frame, denoted as p_{tc}. This operation synthesizes the crane mast height (h), the mast-to-boom offset (b), the luffing angle (μ), and the boom length (L), while treating the lateral elastic deflection (ν_f) as a localized rotational disturbance in the slewing direction:

$$p_{tc} = \begin{bmatrix} b + L\cos\mu\cos\nu_f \\ L\cos\mu\sin\nu_f \\ h + L\sin\mu \end{bmatrix}. \tag{D.8}$$

To isolate the final absolute coordinates of the boom tip in the Inertial Frame (p_{ti}), this local vector is rotated into the Vessel Frame via the crane-to-vessel rotation matrix R_{vc}, added to the crane-base offsets, and subsequently transformed by the vessel rotation matrix R_{iv}:

$$p_{ti} = p_{\text{base}} + R_{iv} R_{vc} p_{tc}. \tag{D.9}$$

Isolating ν_f within the local vector makes it possible to visually distinguish and inspect how structural boom flexibility deflects the tip trajectory independently of the rigid-body slewing and luffing inputs.

D.3.3 Cable and Lower-Block Projection

The terminal link in the forward kinematics chain determines the absolute position of the lower-block. The orientation of the hoisting cable relative to the boom tip suspension point is modeled using the sway angles α and β. The visualizer maps these angles to a 3D unit direction vector within the local Load Frame:

$$\Omega = \begin{bmatrix} -\sin\alpha \\ \cos\alpha\sin\beta \\ -\cos\alpha\cos\beta \end{bmatrix}. \tag{D.10}$$

Given an instantaneous rope hoist length ℓ, the absolute coordinates of the lower-block (p_{li}) are fully determined by projecting along this directional vector from the calculated inertial boom tip position p_{ti} using the combined rotational transformation chain:

$$p_{li} = p_{ti} + \ell R_{iv} R_{vc} \begin{bmatrix} -\sin\alpha \\ \cos\alpha \sin\beta \\ -\cos\alpha \cos\beta \end{bmatrix}. \tag{D.11}$$

This sequential reconstruction ensures that every geometric component aligns with the underlying mechanical structure, providing a reliable baseline for the dual-model rendering validation.

D.4 Graphical Rendering Engine and Dual-Model Synchronization

To render the reconstructed kinematic states in real time, a 3D graphics engine was developed using the `PyQt6` framework and `pyqtgraph.opengl`. By using an OpenGL-backed canvas widget (`GLViewWidget`), the tool offloads rendering calculations to the GPU. This optimization ensures a steady visual refresh rate that matches the target playback frame rate.

D.4.1 Geometric Asset Abstraction

The structural components of the vessel and crane are abstracted into geometric primitives. This approach simplifies the scene while preserving physics clarity:

- **Vessel:** Represented as a flat rectangular plane via a `GLMeshItem`, which rotates dynamically to mirror the wave-induced movements.
- **Crane Mast and Boom Structures:** Modeled as single-segment skeletal line primitives using `GLLinePlotItem`. This abstraction highlights the rigid orientation changes induced by slewing (ν) and luffing (μ), alongside the elastic deflections (ν_f).
- **Hoisting Cable:** Rendered as a single linear segment stretching continuously between the calculated inertial boom tip p_{ti} and the lower-block position p_{li}.
- **Lower-Block:** Rendered as a point mass at the end of the cable using a localized `GLScatterPlotItem`.

By using these low-overhead primitives, the application can quickly update the vertices on the fly.

D.4.2 Dual-Model Synchronization and Error Amplification

A key feature of this visualization framework is its ability to render two separate simulation trajectories simultaneously within the same spatial environment. This synchronized presentation proved highly valuable for cross-checking the independent Python integration outputs against the MATLAB/Simulink reference logs, as well as qualitatively evaluating the estimation performance of the EKF and UKF architectures derived in Chapter 4.

The framework instantiates two separate kinematic instances: a reference model (representing the ground truth simulation trajectory) and an estimation model

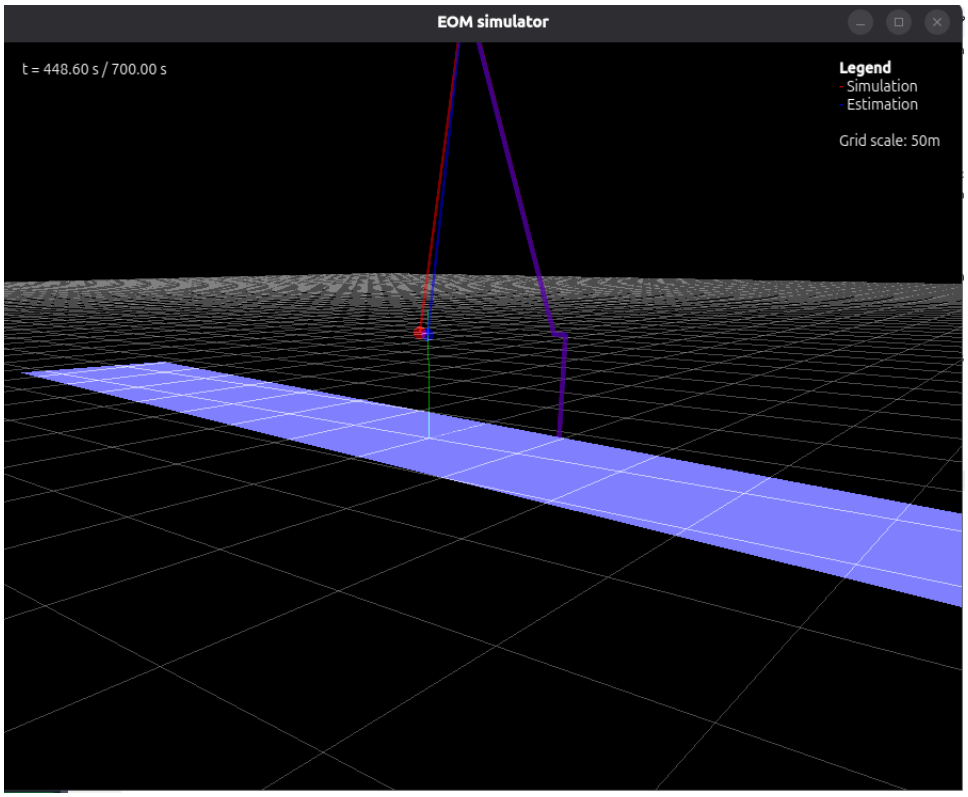


Figure D.1: **Graphical interface of the 3D EoM visualizer. The reference simulation trajectory is rendered in red, while the filtered state estimation model is superimposed in blue. The interactive workspace supports real-time camera manipulations (panning and zooming) with an automated loop routine for continuous trajectory inspection.**

(representing the filtered state estimates). To preserve visual clarity during concurrent display, the framework uses a color-coded rendering scheme, displaying the ground truth trajectory in red and the estimated model in blue. Any tracking lag, coordinate alignment mismatches, or numerical drift between the two environments appears immediately as a visible separation between the two models.

Furthermore, small estimation errors in the sway angles (α, β) can be difficult to spot visually because the hoisting cable is very long. To address this, the visualizer includes an error amplification function. This feature scales up the spatial differences between the simulated lower-block position ($p_{li,\mathrm{sim}}$) and the estimated lower-block position ($p_{li,\mathrm{est}}$) by a user-defined factor κ:

$$p_{li,\mathrm{rendered}} = \kappa\, p_{li,\mathrm{est}} + (1-\kappa)\, p_{li,\mathrm{sim}}. \tag{D.12}$$

By scaling up these small variations, it is possible to visually isolate and diagnose tiny tracking differences and subtle non-linear modeling errors that might otherwise show up only as minor background noise in raw data plots.

D.4.3 Real-Time Playback

The synchronization loop is driven by a high-resolution `QtCore.QTimer` configured to trigger at precise intervals matching the visual frame period. Upon each timer tick event, the playback engine computes the current simulation timestamp (t_{play}), scaled by an optional user-defined acceleration factor.

Since the simulation datasets are logged at discrete intervals, the framework maps the continuous playback time to a discrete array index i using a localized minimum distance search mapping:

$$i = \arg\min_{j} |t_j - t_{\text{play}}|. \tag{D.13}$$

Upon resolving the instantaneous frame index i, the respective state vectors for both the reference and estimation datasets are passed directly to their corresponding `CranePose` instances. These instances update their respective forward kinematic pipelines, modify the underlying OpenGL vertices, and command a canvas repaint.