

Rate-Based Pendulation Control System Study

VERSION 1.1

Dr. H. Schaub

Virginia Polytechnic Institute and State University

June 21, 2006

Table of Contents

1	Introduction	1
2	Ship Motion Estimation	5
2.1	Drifting Ship Frame Concept	5
2.2	Existing Solution: Filtering the Inertial Ship Motion	6
2.2.1	Coordinate System Layout	6
2.2.2	Transformation Algorithm	6
2.2.3	Numerical Filter Algorithm	7
2.3	Estimating Ship Motion Using IMU and Gyro Data	9
2.3.1	Ship Frame Angular Velocity Vector	9
2.3.2	Inertial Ship Acceleration Vector	9
2.3.3	Filtered Inertia Prime Ship Motion	10
2.3.4	Digital Filter Algorithm and Performance	11
2.4	Self-Tuning ω_c Algorithm	13
3	Control Solution	17
3.1	Existing Position Based Solution	17
3.1.1	Coordinate Frames	17
3.1.2	Position-Based Inverse Kinematics Solution	19
3.1.3	Control Flowchart	20
3.2	Rate-Based Control Solution Using Only Ship Motion Integration	22
3.3	Rate-Based Control Solution Using Velocity Kinematics	23
3.3.1	Velocity-Based Inverse Kinematics	23
3.3.2	Control Flowchart	25
4	Conclusions	29

Chapter 1

Introduction

The U.S. Navy is increasingly relying on commercial cargo vessels to transport supplies across the world. These cargo ships require commercial harbors with specialized cranes to off-load the containers. However, in many mission scenarios cargo must be delivered to coast lines which do not possess such commercial off-loading facilities. During Desert Storm operations, crane ships were employed to off-load the cargo from the container ships to smaller lighter vessels. Here the crane ship was anchored off-shore, while the cargo and lighter vessel were moored to it (see Fig. 1.1). However, it was found that even moderate sea states levels of 1 or 2 could cause dangerous amounts of payload pendulation. This caused frequent periods where the cargo transfer process was shut down until the sea conditions were calmer.



Figure 1.1: Cargo Ship Off-Loading Containers Onto Lighter Vessels.

Sandia National Laboratories won a contract to develop a prototype Pendulation Control System (PCS). The goal was to reduce payload pendulation and simplify the anchored-at-sea cargo transfer process for the crane crew. A

three-fold strategy was proposed. The existing crane control system is to be upgraded, and enhanced with new sensors, such that:

1. The payload motion is isolated from the crane ship motion. To accomplish this, the new ship motion sensors were installed. The new PCS will automatically compensate for ship motion and keep the ship motion from causing payload swing.
2. Transient payload swing is to be rejected. By measuring the payload swing angles, any payload swing caused by lift-off transients or crane servo limitations will be reduced to zero. This required installing a new swing sensor on the crane.
3. Operator commanded payload swing is avoided. The commanded crane speed (joystick) signals are modified such that the operator cannot cause payload swing to occur. This enhancement didn't require any new sensing hardware.

After first demonstrating this control concept on an in-house 1/16th scale motion platform and creating a hardware-in-the-loop simulation of the developed control algorithm, in October 2002 Sandia National Laboratories installed their first generation PCS solution onto the T-ACS 5 vessel (Flickertail State shown in Fig. 1.2). This installation was then tested and its performance tested versus the existing crane control modes (crane with RBTS, crane without RBTS). The PCS showed promising performance enhancements and the crew was able to transfer cargo at higher ship roll angles.



Figure 1.2: T-ACS 5 Vessel (Flickertail State).

The existing PCS solution is based on a position-based control strategy. This concept requires the knowledge of the rotational orientation, the ship translational position, as well as the crane and payload swing position angles. To measure the 6 degree-of-freedom ship positions, the POS/MV 320 system was purchased and installed. This is a highly accurate ship navigation system which can stream the current ship position and orientation across a serial line. To measure these six states, this system employs an inertial measurement unit (IMU), a rate gyro, as well as two GPS receivers. Due to operational constraints as to where the PCS was to function, differential GPS modes were not to be used. The horizontal position errors of this system varied between ± 1 meters to ± 10 meters, depending on the current accuracy of the GPS position solutions. The computed ship motion was very smooth, but these position errors appear as a random-walk component in the measurement signal. The PCS concept does further process the ship motion measurement to reduce the effect of these measurement errors on the payload motion. However, this effect can never

be canceled completely. Thus, even with a perfect crane servo sub-system which can perfectly implement the commanded crane speeds, this system will experience some amount of payload drift due to the ship motion sensor errors.

This report discusses an alternate approach to measuring the ship motion, and thus providing an alternate PCS algorithm. Instead of requiring the knowledge of the absolute ship position and orientation, this report investigates if it is possible to only measure ship translational acceleration and angular rate (IMU and gyro output) and still provide adequate PCS performance. The acceleration and rate measurement will also have biases and noise components. However, a process will be discussed how to compensate for these and integrate such measurements directly into the PCS algorithm. If no sensor errors are modeled, this new PCS concept will provide identical performance compared to the existing PCS concept. However, the IMU sensor is much cheaper than a full integrated navigation system such as the POS/MV. Thus, even if the ultimate rate-based PCS performance is comparable to the position-based

PCS solution, a substantial cost savings will be achieved when installing this systems on other crane ships.

The report is broken up into two primary sections. First the ship motion estimation process is discussed. Here the new methods are compared and contrasted to the existing ship motion processing solution. Next, the control algorithm changes our outlined. Again the new methods are compared and contrasted to the existing PCS solutions. Note that no performance evaluations are discussed in this report. Studying how the PCS performance will change is the scope of another report.

Chapter 2

Ship Motion Estimation

2.1 Drifting Ship Frame Concept

If a truly inertial coordinate frame \mathcal{I} is used to measure the ship motion, then slow drifting motion about the anchor point will cause the control to move the payload further and further away from the ship. Because the cargo and lighter vessels are to be attached to the crane ship, keeping the payload stationary with respect to an absolute inertial frame is overly restrictive and will in fact reduce the practicality of the PCS concept. Instead the global ship motion is mapped into another reference frame called the \mathcal{I}' frame. This reference frame is nominally aligned with the ship surge, sway and heave axes. As the ship drifts slowly, the \mathcal{I}' frame will drift along with it. The short-period motion of the ship is defined as any ship motion which has a frequency content about the 0.06-0.10 Hz bandwidth. This is the bandwidth at which the dominant amount of ship motion will occur. Further, this is also the bandwidth region of the natural payload pendulation frequency. Thus, it is important for the Pendulation Control System (PCS) to compensate for this short period ship motion. However, long period motion, or secular drift, should not be compensated for by the PCS. This slow motion will not be a driving factor in payload pendulation, and the cargo target vessels would be drifting along with the crane ship.

On the other hand, a long period motion, or secular drift, will cause only a minimal amount of payload swing. If the PCS system were to compensate for this slow drift, then the crane

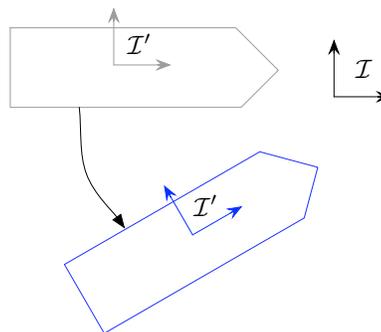


Figure 2.1: Illustration of the Slowly-Varying Inertia-Prime frame \mathcal{I}' .

would soon reach its maximum operational envelope. During the Turbo Patriot exercises, the Grand Canyon State ship was found to drift ten's of meters over half an hour. This drift was typically a slow rotation about the anchor point as the wind and wave directions were changing. Thus, this slow motion will not be a driving factor in payload pendulation. Further, because the cargo target vessels would be drifting along with the crane ship, it would not be beneficial to try to keep the payload at rest with respect to the absolute inertial frame \mathcal{I} .

The ship motion in the \mathcal{I}' frame will therefore only show what short-period motion the ship is performing. Any slower drifts or very high frequency components will be filtered out. Note that the ship sensor itself will also introduce some amount of static drift in its translational position measurement. This drift typically occurs at 0.005 - 0.02 Hz. By mapping the measured ship motion into the \mathcal{I}' frame, this artificial sensor induced drift will be filtered out to some degree as well.

2.2 Existing Solution: Filtering the Inertial Ship Motion

The current PCS solutions uses the POS/MV 320 inertial motion sensor to provide the six degrees of freedom information of the crane ship. This section illustrates for comparison purposes how the ship motion with respect to the \mathcal{I}' frame is computed. The following section will illustrate how the \mathcal{I}' ship motion is obtained if only accelerometer and rate gyro information is available.

2.2.1 Coordinate System Layout

The local inertial Earth coordinate frame \mathcal{I} is illustrated in Figure 2.2. This coordinate frame is oriented with the x axis aligned with the local north direction, while the y points towards the local west direction. The heading angle γ is defined as a clock-wise rotation about the local vertical axis. Note that the yaw angle ψ is also defined as a rotation about the local vertical axis, but with the opposite sign.

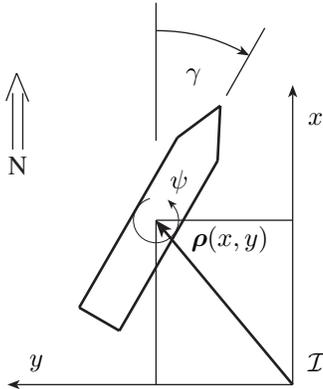


Figure 2.2: Illustration of the Local Earth Coordinate Frame

Let the vector $\rho(x, y)$ be the position vector of the ship relative to the \mathcal{I} frame origin. The vector components are taken in the \mathcal{I} frame.

Figure 2.3 shows the “moving inertia frame” \mathcal{I}' relative to the local inertia frame \mathcal{I} . The angle $\hat{\gamma}$ is the heading angle of the \mathcal{I}' frame, while $\hat{\rho}$ is the position vector of the \mathcal{I}' frame relative to the \mathcal{I} frame origin. The ship position vector

relative to the \mathcal{I}' frame is expressed simply by ρ' . The heading angle of the ship relative to the \mathcal{I}' frame is given by γ' .

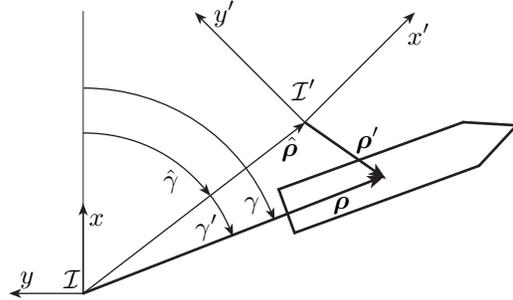


Figure 2.3: Illustration of the \mathcal{I}' Coordinate Frame

Note that if the ship were to only perform a constant drift (moving in a constant current for example), then γ' and ρ' would be zero. As the ship begins to do some short-period oscillatory motions due to interactions with waves and wind, then we would have non-zero γ' and ρ' states. The \mathcal{I}' frame can be visualized as a nominal reference frame about which the ship is currently yawing, surging and swaying.

2.2.2 Transformation Algorithm

Next we outline the algorithm which will translate the inertial (x, y, γ) into the \mathcal{I}' frame (x', y', ψ') states. Note that (x, y) are the local inertial North-West coordinates, while (x', y') are the ship relative surge and sway coordinates. The first step is to filter and process the heading angle. Let $F(\cdot)$ be some generic filter function that will remove any low frequency motions (i.e. a type of high-pass or band-pass filter). The exact digital filtering algorithm is discussed in more detail later on. The heading angle of the ship relative to the \mathcal{I}' frame is then computed using

$$\gamma' = F(\gamma) \quad (2.1)$$

The filtered ship yaw angle is then computed simply using

$$\psi' = -\gamma' \quad (2.2)$$

The \mathcal{I}' heading angle $\hat{\gamma}$ relative to the \mathcal{I} frame is now computed using

$$\hat{\gamma} = \gamma - \gamma' \quad (2.3)$$

Let ${}^{\mathcal{I}}\boldsymbol{\rho}$ be the ship position vector with the vector components taken in the \mathcal{I} frame. The rotation matrix $[\mathcal{I}'\mathcal{I}]$, which maps vector components in the \mathcal{I} frame to vector components in the \mathcal{I}' frame, is given by

$$[\mathcal{I}'\mathcal{I}] = \begin{bmatrix} \cos \hat{\gamma} & -\sin \hat{\gamma} & 0 \\ \sin \hat{\gamma} & \cos \hat{\gamma} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

The position vector ${}^{\mathcal{I}}\boldsymbol{\rho}$ is then rotated into \mathcal{I}' vector components using

$${}^{\mathcal{I}'}\boldsymbol{\rho} = [\mathcal{I}'\mathcal{I}] {}^{\mathcal{I}}\boldsymbol{\rho} \quad (2.5)$$

The filtered \mathcal{I}' ship motion $\boldsymbol{\rho}'$ is then obtained by applying the filter $F()$ to this position vector.

$${}^{\mathcal{I}'}\boldsymbol{\rho}' = (x', y', h') = F({}^{\mathcal{I}'}\boldsymbol{\rho}) \quad (2.6)$$

Note that the vertical height measurement is also filtered in this process. The static drift errors of the vertical measurement are relatively large with a GPS aided system. By filtering the vertical position measurement with a highpass or bandpass filter, the oscillatory heave motion h' is extracted.

2.2.3 Numerical Filter Algorithm

A 2nd order digital bandpass filter $F()$ is used to remove the static and low frequency components of the heading and inertial (x, y, z) ship motion. Let x_k be the current ship state measurement, while x_{k-i} is the i^{th} previous measurement. Similarly, let y_k be the equivalent filtered ship state at the current time step, while y_{k-i} are the i^{th} previous filtered states. Let ω_c (in units of rad/s) be the center frequency of the bandpass filter, BW (in units of rad/s) be the allowable bandwidth, while ξ is the non-dimensional damping factor. The digital sampling period is h .

The recursive digital filter algorithm is

$$y_k = \frac{1}{a_0} \left[\begin{aligned} & y_{k-1}(4(4 - h^2\omega_c^2)(4 + h^2\omega_c^2 \\ & \quad + 2hBW\xi)) \\ & + y_{k-2}(2(-48 + 4h^2BW^2 + 8h^2\omega_c^2 \\ & \quad - 3h^4\omega_c^4)) \\ & + y_{k-3}(4(4 - h^2\omega_c^2)(4 + h^2\omega_c^2 \\ & \quad - 2hBW\xi)) \\ & + y_{k-4}(-4h^2BW^2 - (4 + h^2\omega_c^2)^2 \\ & \quad + 4BWh(4 + h^2\omega_c^2)\xi) \\ & + 4h^2BW^2(x_k - 2x_{k-2} + x_{k-4}) \end{aligned} \right] \quad (2.7)$$

where

$$a_0 = 4h^2BW^2 + (4 + h^2\omega_c^2)^2 + 4hBW(4 + h^2\omega_c^2)\xi \quad (2.8)$$

Note that this filter is applied to the heading angle γ , as well as the ${}^{\mathcal{I}'}\boldsymbol{\rho}$ states. The algorithm was derived using the trapezoidal rule to express the differential operator. The corresponding Laplace domain filter transfer function is

$$\frac{Y(s)}{X(s)} = \frac{BW^2 s^2}{(s^2 + \omega_c^2)^2 + 2BWs\xi(s^2 + \omega_c^2) + BW^2 s^2} \quad (2.9)$$

The transfer function of this digital filter is illustrated in Figure 2.4. Here a center frequency ω_c of 1/12 seconds was chosen (12 second nominal ship period), a bandwidth BW of 0.1 Hz, and damping coefficient ξ value of 0.707. Note that with this critical damping coefficient, there is a flat region around ω_c in the magnitude plot where effectively no amplification is applied to the input signal. Higher and lower frequency components of the input signal are filtered out. The efficiency of the filtering depends on the frequency spread between the center frequency and the to be rejected frequency.

Note that the POSMV sensor system provides rather smooth sensor signals. Thus, it would not be necessary to filter out the higher

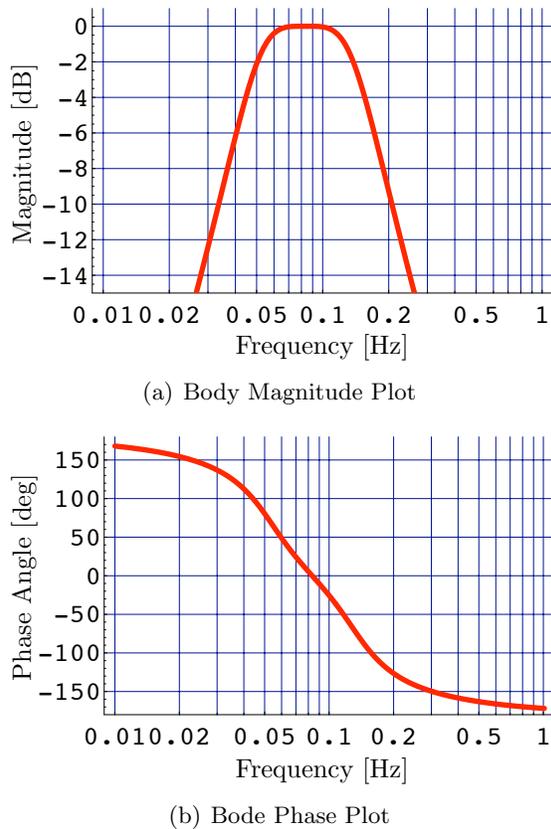


Figure 2.4: 2nd Order Bandpass Filter Bode Plots.

frequencies along with the static and low frequency signal components. A lowpass filter could also have removed the static and low frequency signal components. However, a low pass filter will always produce a non-zero phase lag in the filtered states. At the drop-off frequency (magnitude drops below -3dB) this phase error is already 45 degrees! To keep this phase angle error to a minimum, a bandpass filter was chosen instead of the low pass filter. Think of the bandpass filter as a smart combination of a lowpass and highpass filter. The lowpass filter will always produce a lag, a highpass filter a lead phase angle in the filtered signal. The bandpass filter will act as a highpass filter for low frequencies (lead) and lowpass filter for high frequencies (lag). At the bandpass center frequency ω_c there is zero phase angle error as shown in Figure 2.4(b). Since the ship will roll at a known, relatively fixed frequency, we can set the bandpass center frequency to be equal

to the dominant ship roll frequency. This allows us to obtain a filtered ship motion signal whose dominant motion component will have little or no phase error.

If the ship frequency is different than ω_c , than either a lag or lead phase error will be present. The higher order the filter is, the stronger this center frequency sensitivity will be. A first order filter will be able to asymptotically reject any constant term of the input signal. Thus, any ship motion biases will be completely rejected. However, a constant linear motion in the input signal will result in a constant bias in the filtered signal. The magnitude of this bias is depended on the ship drift rate and the filter bandwidth parameter. To avoid such errors during constant drifting, a second order bandpass filter was chosen. The second order filter can reject both constant and linearly drifting signal components as illustrated in the following numerical simulation.

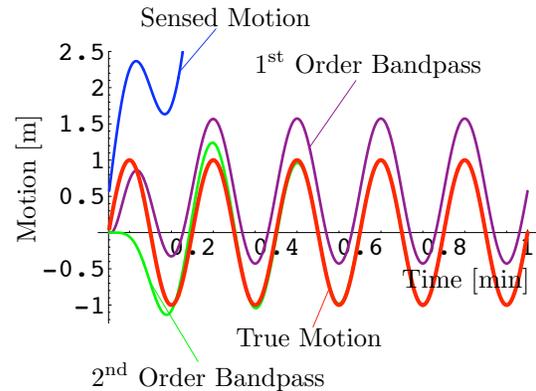


Figure 2.5: Illustration of Bandpass Filtering the Ship Motion Position Data.

The ship motion is simulated as a 1 meter sinusoidal motion with a 12 second period. The sensed motion signal is corrupted with a constant 0.5 meter bias, as well as a linear 0.25 meter per second drift. Figure 2.5 compared the true and sensed ship motion to 1st and 2nd order bandpass filtered data. The filters have a center frequency ω_c of 1/12 Hz, a bandwidth BW of 0.1 Hz, and a damping coefficient ξ of 0.707. Due to the linear drift of the sensed ship motion, the 1st bandpass filtered data doesn't asymptotically converge to the true motion. Even with

the relatively slow 0.25 m/s motion, the steady-state bias is noticeable. In contrast, the 2nd order bandpass filtered data is able to completely reject this linear drift, as would be desired for the PCS application. Note that the filter transients no longer visible after 30 seconds.

2.3 Estimating Ship Motion Using IMU and Gyro Data

Using a PCS concept which requires the absolute six degree of freedom ship states surge, sway, heave, yaw, pitch and roll is rather expensive. The POSMS 320 sensor uses an extensive Kalman filter to included GPS signals to bound the ship position state errors. Without the GPS measurements, integrating noisy gyro and IMU data would quickly cause the perceived motion to radically deviate from the true ship motion. The rate-based PCS concept investigates using only IMU and rate gyro information to compute the ship motion with respect to introduced drifting \mathcal{I} . Note that the ship motion doesn't need to be known with respect to the true inertial frame, only with respect to the drifting frame. If the resulting PCS performance is acceptable, then the cost of the required ship motion sensor would be dramatically lowered. Thus, in this development the ship motion sensor is assumed to provide gyro angular rate measurements, IMU accelerometer measurements, as well as direct pitch angle θ and roll angle ϕ measurement. The only attitude coordinate not directly measured will be the yaw angle ψ .

2.3.1 Ship Frame Angular Velocity Vector

The rate gyro sensor measured the instantaneous angular velocity vector of the sensor or ship frame \mathcal{S} with respect to the inertial frame \mathcal{I} . Let the ship fixed frame $\mathcal{S} : \{\hat{s}_1, \hat{s}_2, \hat{s}_3\}$ be the frame of the rate gyro. This frame is nominally aligned such that \hat{s}_1 points towards the bow, \hat{s}_2 points towards port, and \hat{s}_3 points up-

ward as illustrated in Figure 2.6. However, this alignment is not mandatory with the PCS strategy.

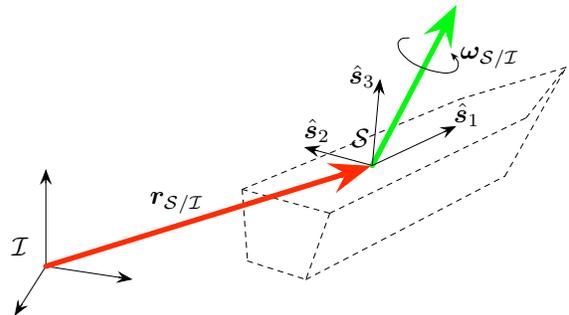


Figure 2.6: Illustration of Ship Translation and Rotation With Respect to Inertial Frame.

The angular velocity vector

$$\omega_{S/I} = \omega_1 \hat{s}_1 + \omega_2 \hat{s}_2 + \omega_3 \hat{s}_3 \quad (2.10)$$

is typically expressed in \mathcal{S} frame components as shown. The three vector components ω_i are the three angular rates that the gyro sensor will provide. These body angular velocities can be mapped into yaw rate $\dot{\psi}$, pitch rate $\dot{\theta}$ and roll rate $\dot{\phi}$ using

$$\begin{pmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{pmatrix} = \begin{bmatrix} 0 & \frac{s\phi}{c\theta} & \frac{c\phi}{c\theta} \\ 0 & c\phi & -s\phi \\ 1 & s\phi \tan\theta & c\phi \tan\theta \end{bmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} \quad (2.11)$$

with $c\alpha = \cos \alpha$, $s\alpha = \sin \alpha$ and $\tan \alpha = \tan \alpha$. We won't need the pitch and roll rates in the rate-based PCS solution. However, we will require the yaw rate to be able to compute the current yaw angle. The roll and pitch angles are assumed to be directly available from the ship motion sensor.

2.3.2 Inertial Ship Acceleration Vector

The IMU measurement will provide the true acceleration vector \mathbf{a} that the ship motion sensor is experiencing. Let $\mathbf{r}_{S/I}$ be the inertial position vector of the ship motion sensor, and \mathbf{g} be the local gravitational vector, then the measured acceleration is expressed as

$$\mathbf{a} = \ddot{\mathbf{r}}_{S/I} + \mathbf{g} \quad (2.12)$$

Note that the Earth is treated as a non-rotating object here. In Integrated-Navigation-System (INS) problems the Earth's rotation must be included. However, for the PCS required ship motion information, the Earth rotation is treated as a very low frequency perturbation which will not contribute to payload pendulation. The vector components of the measured acceleration will be taken with respect to the ship frame \mathcal{S} .

$${}^{\mathcal{S}}\mathbf{a} = a_1\hat{\mathbf{s}}_1 + a_2\hat{\mathbf{s}}_2 + a_3\hat{\mathbf{s}}_3 \quad (2.13)$$

Because the PCS solution requires the inertial ship motion $\mathbf{r}_{\mathcal{S}/\mathcal{I}}$ in inertial frame components, we solve for the inertial ship acceleration using

$$\mathcal{I}\ddot{\mathbf{r}}_{\mathcal{S}/\mathcal{I}} = [\mathcal{I}\mathcal{S}] {}^{\mathcal{S}}\mathbf{a} - \mathcal{I}\mathbf{g} \quad (2.14)$$

where

$$\mathcal{I}\mathbf{g} = \begin{pmatrix} 0 \\ 0 \\ 9.81 \end{pmatrix} \text{ m/s} \quad (2.15)$$

The rotation matrix $[\mathcal{I}\mathcal{S}]$ requires the ship yaw, pitch and roll angles (ψ, θ, ϕ) .

$$[\mathcal{I}\mathcal{S}] = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix}$$

The roll and pitch angles will be measured directly from the ship motion sensor, while the yaw angle ψ will need to be computing from the yaw rate expression in Eq. (2.11).

The required inertial ship motion sensor velocity vector $\dot{\mathbf{r}}_{\mathcal{S}/\mathcal{I}}$ and position vector $\mathbf{r}_{\mathcal{S}/\mathcal{I}}$ are then found by integration this acceleration expression. However, care must be taken with this integration. Sensor biases and noise will cause un-aided integration schemes to go unstable. The next section will discuss how to stabilize this process.

2.3.3 Filtered Inertia Prime Ship Motion

The rate-based PCS concept will require both the inertial ship motion sensor motion $\mathbf{r}_{\mathcal{S}/\mathcal{I}}$ and

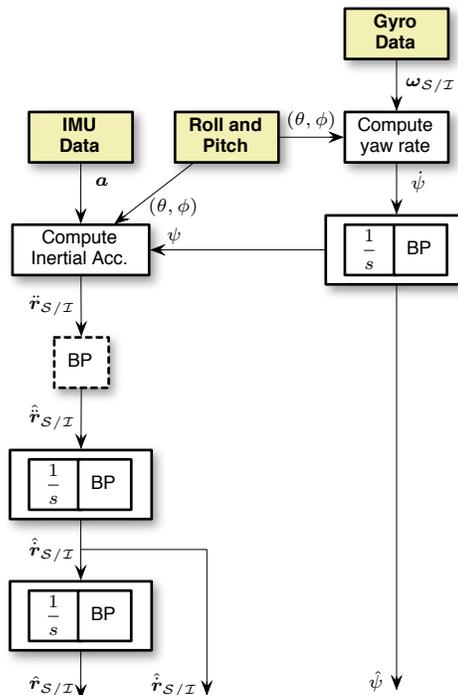


Figure 2.7: Flowchart Illustration How IMU and Gyro Data is Integrate and Filtered.

velocity $\dot{\mathbf{r}}_{\mathcal{S}/\mathcal{I}}$, as well as the sensor rotation rate $\omega_{\mathcal{S}/\mathcal{I}}$ and attitude (ψ, θ, ϕ) to determine the inverse-velocity-kinematics. Using the IMU and gyro data, this section will outline how these states can be computed. In particular, we don't require the ship motion, velocity and heading with respect to the absolute inertial frame \mathcal{I} . Instead, we will again take advantage of the drifting \mathcal{I}' which yields nominally zero surge, sway, heave and yaw states. The pitch and roll angles are assumed to be measured directly and are not filtered in this process. This is important because the roll and pitch angle determines the local gravity vector direction. Further, the roll and pitch angles will not be oscillating about zero. The crane ship will typically have a non-zero tilt depending on the sea state and the loading of the vessel.

Figure 2.7 illustrates how the sensor measurements will be integrated and filtered to obtain the required ship motion states. The gyro measurement vector $\omega_{\mathcal{S}/\mathcal{I}}$ is used along with Eq. (2.11) to compute the yaw rate $\dot{\psi}$ of the ship sensor frame. This rate is then integrated

and filtered at the same time using a digital recursion formula. The algorithm is explained in detail later on. The filter is a first order bandpass filter which will reject any static offsets, or low-frequency components of the filtered yaw angle $\hat{\psi}$. The resulting yaw angle will have a nominal value of zero. This makes this filtered yaw angle $\hat{\psi}$ equivalent to the previous yaw angle ψ' with respect to the \mathcal{I}' frame. The filtered yaw angle no longer provides the ship heading with respect to north, but with respect to the drifting \mathcal{I}' frame. If the gyro measurement has a bias error, then this will cause the unfiltered integration to have an erroneous linear growth term. The 1st order bandpass filter will reduce this linear error growth to a constant offset. The magnitude of this offset depends on the gyro bias amplitude and the filter gain settings. This offset should have a minimal impact on the PCS performance since the method is insensitive to have a static yaw offset.

With a yaw angle $\hat{\psi}$ estimated, we are now able to map the IMU measured acceleration vector \mathbf{a} and compute the inertial acceleration vector $\ddot{\mathbf{r}}_{S/\mathcal{I}}$. This vector is then filtered and integrated in a series of steps to obtain estimated inertial velocity $\hat{\dot{\mathbf{r}}}_{S/\mathcal{I}}$ and estimated inertial position $\hat{\mathbf{r}}_{S/\mathcal{I}}$. Each filtering step is a first order bandpass filter which removes static offsets, as well as low and high frequency components relative to the filter center frequency (set to dominant ship roll frequency). The integrations and filtering steps are completed as one. There is an optional bandpass filter step shown in Figure 2.7 which would process directly the $\ddot{\mathbf{r}}_{S/\mathcal{I}}$ states. If active, then this step will remove any bias from the acceleration measurements. An acceleration bias will cause a quadratic error drift in the position measurement. The two integration steps with bandpass filtering only yield a combined second order filter. Thus, these two filters will enable the accelerometer bias to cause a constant offset in the ship position measurement. However, initial studies of the PCS performance shows it be reasonably insensitive to such biases. If the dashed bandpass filter box is active, then this additional filter step will remove the position

bias completely. However, the more filtering steps are included, the more sensitive the PCS performance will become to the bandpass filter frequency not match precisely the ship motion frequency. A separate rate-based performance study will investigate this issue further.

Finally, please note that the estimated $\hat{\dot{\mathbf{r}}}_{S/\mathcal{I}}$ and $\hat{\mathbf{r}}_{S/\mathcal{I}}$ vectors are identical to the \mathcal{I}' motion computed in the position-based PCS strategy. The integrated and filtered ship motion will have a nominally zero value, thus representing surge, sway, heave and yaw motion with respect to a slowly moving reference frame.

2.3.4 Digital Filter Algorithm and Performance

To integrate the rate and acceleration measurements, we assume that the sensor measurements are available at constant time steps h . This won't be the case with the actual ship sensor, where the sensor update frequency and the control update frequency are different. To provide the integration and filtering routine with state measurements at constant intervals, an estimation routine is implemented which extrapolates all measured states to the next control time step. This estimator allows provides increased PCS robustness to intermittent sensor communication issues.

The Laplace domain transfer function of the 1st order bandpass filter is given by

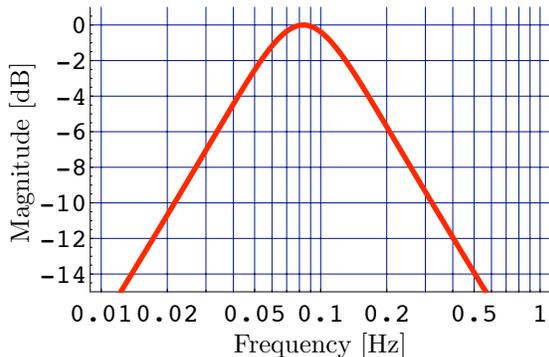
$$\frac{Y(s)}{X(s)} = \frac{sBW}{s^2 + BWs + \omega_c^2} \quad (2.16)$$

where $X(s)$ is the filter input signal, and $Y(s)$ is the filtered state. If the state is being integrated and filtered at the same time, then the transfer function is given by

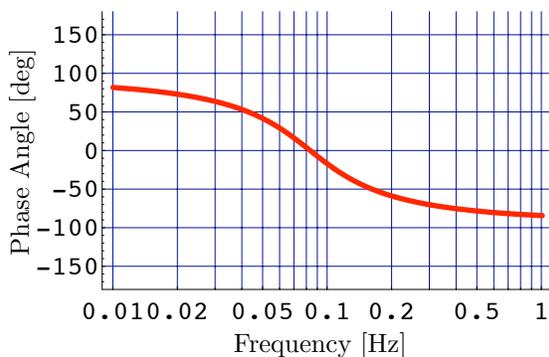
$$\frac{Y(s)}{X(s)} = \frac{sBW}{s^2 + BWs + \omega_c^2} \frac{1}{s} \quad (2.17)$$

The bode plots of the 1st order bandpass filter are shown in Figure 2.8. Here the center frequency ω_c is set to 1/12 Hz, and the bandwidth parameter BW is set to 0.1 Hz. Note that the first order bandpass filter is not as effective in rejecting lower and higher frequency

components as the second order bandpass filter in Figure 2.4. However, the phase angle introduced with the filter is less sensitive to differences between the ship motion frequency and the filter center frequency.



(a) Body Magnitude Plot



(b) Bode Phase Plot

Figure 2.8: 1st Order Bandpass Filter Bode Plots.

Using the trapezoidal rule to digitally approximate the differential operator s , we find the following recursive formula to bandpass filter a signal: The recursive digital filter algorithm is

$$y_k = \frac{1}{4 + 2BWh + h^2\omega_c^2} \left[\begin{aligned} &y_{k-1}(8 - 2h^2\omega_c^2) \\ &+ y_{k-2}(-4 + 2BWh - h^2\omega_c^2) \\ &+ 2hBW(x_k - x_{k-2}) \end{aligned} \right] \quad (2.18)$$

The recursive formula for the integration and

bandpass process is:

$$y_k = \frac{1}{4 + 2BWh + h^2\omega_c^2} \left[\begin{aligned} &y_{k-1}(8 - 2h^2\omega_c^2) \\ &+ y_{k-2}(-4 + 2BWh - h^2\omega_c^2) \\ &+ h^2BW(x_k + 2x_{k-1} + x_{k-2}) \end{aligned} \right] \quad (2.19)$$

In Figure 2.7, Eq. (2.18) is used in the block labeled BP, while Eq. (2.19) is used in the blocks labeled both with BP and the integration expression $1/s$.

To illustrate the performance of these digital filters, the following simulation is conducted. The true ship motion has a 1 meter amplitude with a 12 second period. The sensed ship acceleration is biased with 0.02 m/s^2 . This is larger than typical IMU acceleration biases, but helps illustrate how the different integration and filtering options differ in performance. Figure 2.9 illustrates the resulting estimated velocity and positions.

The blue line shows the results if the biased acceleration measurements are integrated directly without applying any filtering. As expected, the velocity shows a linear error growth, while the estimated position errors grow quadratically. The purple line shows the estimated results if the velocity and position integration steps are performed simultaneously with a bandpass filter pass. The original acceleration measurements are used unfiltered here. As predicted, the estimated velocity and position states have a small bias. The amplitude of this bias will depend on the acceleration bias, as well as on the filter gains used. The filter gains are set to the same value as in the current PCS implementation. If the IMU acceleration bias is set to a smaller value than the 0.02 m/s^2 used in this simulation, than the estimated velocity and position biases would be difficult to see in this performance illustration. Finally, the green line shows the results if the acceleration measurement is filtered first using a bandpass filter, and then integrated and filtered as in the previous case. Here the original sensor bias does not cause any steady-state

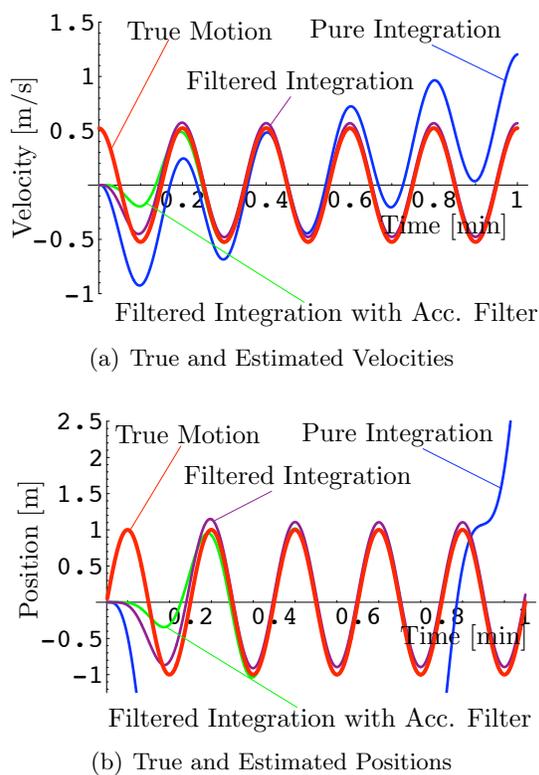


Figure 2.9: Illustration of Integrating Biased Acceleration Measurements to find Velocity and Position Estimates.

biases in estimated velocity and position measurements. However, the initial transients in the estimated states take longer to converge. It would be preferable to use the integration and filtering solution which does not include the additional bandpass filter on the acceleration measurement. However, if it is found that the resulting biases cause performance issues, than this filter addition is a possible solution to be investigated.

2.4 Self-Tuning ω_c Algorithm

To make the bandpass filter function less sensitive to the center frequency ω_c , a self-tuning algorithm is used. Doing so we are able to make use of some of the characteristics of the typical ship motion of ship motion sensor signal.

1. The dominant ship motion is a near-sinusoidal signal. The frequency spread of

the ship motion is centered around a narrow peak.

2. The frequencies don't change very fast. There might be small variations to the roll and other ship motion periods as the ship reacts to changing wind, current, wave or cargo loading conditions. However, this change will be rather slow and subtle.
3. Due to the narrow frequency band exhibited by the ship motion, we have a lower period P_{min} and upper period P_{max} between which the dominant ship periods must lie. These two values provide lower and upper constraints on the actual ship roll period estimator.

The algorithm will estimate the ship roll period, because this is assumed to be the dominant ship motion mode. Figure 2.10 shows the flow chart of the self-tuning ω_c algorithm. While this algorithm directly estimates the current motion period P , note that $\omega_c = 2\pi/P$. The general idea of the algorithm is to watch for the rate state \dot{y} to have three zero crossings and measure the time between the first and the third crossing. This time interval is the desired ship motion period. This simple concept is expanded to make the algorithm more robust to transient non-sinusoidal behavior and measurement noise.

The algorithm assumes that a lower period limit P_{min} and upper period limit P_{max} are available. The parameter h is the digital filter sampling time. The integer parameter DT contains the number of digital filter time steps that may occur before the upper waiting period limit of P_{max} has passed. If no complete sin wave has been detected after P_{max} seconds, then the search is reset. This avoids being stuck in an unsuccessful, partial search.

The algorithm could have tracked two zero crossings instead of three. However, using three crossings is more robust. Assume that the sinusoidal motion has a small off-set to the nominal zero rate value as illustrated in Figure 2.11. By tracking only two zero crossings the ship motion period estimates could be both too large

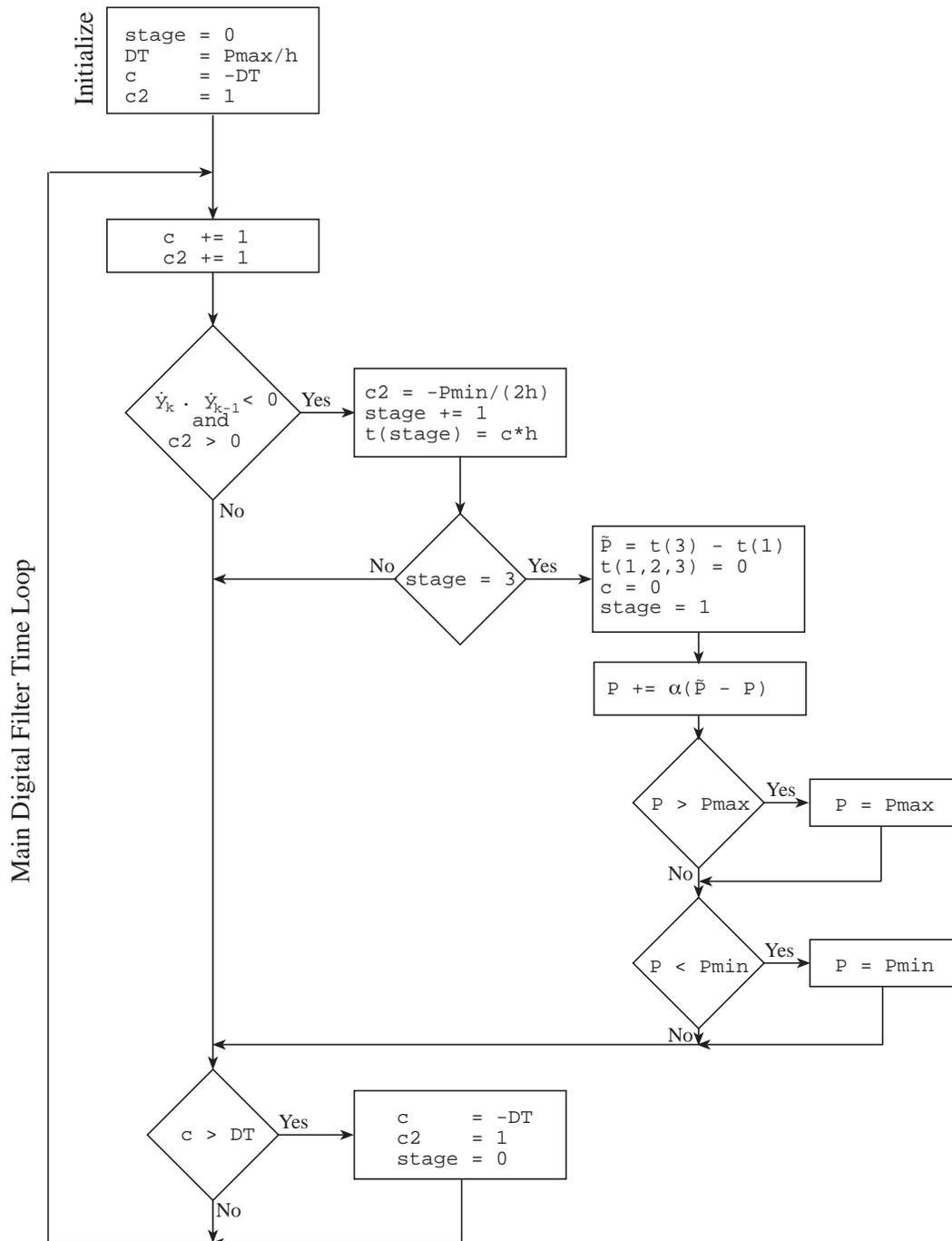


Figure 2.10: Flow Diagram of the Self-Tuning ω_c Algorithm

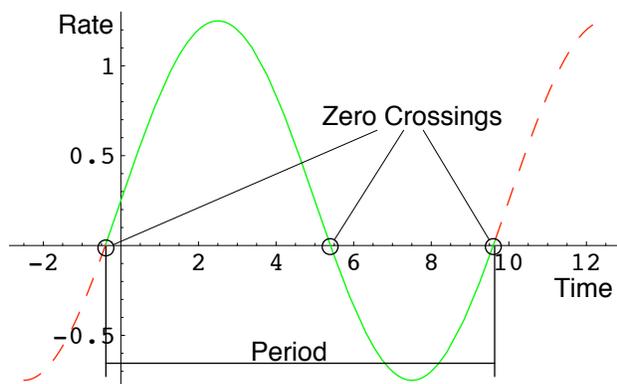


Figure 2.11: Illustration of Estimating the Ship Period if the Rate Measurement have a Bias.

or too small. This is the case even without any noise or other measurement errors. By tracking three crossings and differencing the third and first crossing, we are able to extract the true motion period even with this static offset present in the ship motion.

Instead of tracking the roll rate $\dot{\phi}$ zero crossings, the algorithm could also have tracked the integrated and filtered roll state ϕ zero crossings. However, because the rates are measured directly with the rate gyro sensor, this is more direct approach. Using the acceleration state zero crossing was avoided to due the noise issues when performing a numerical differentiation.

The three integer parameters c , $c2$ and $stage$ control the general flow of the period estimation algorithm. The integer c counts the h time steps since the current period search was started. The integer $c2$ counts the steps since the last zero crossing was detected. This parameter is used to avoid having sensor noise cause two erroneous zero crossings. The parameter $stage$ keeps track of how many zero crossings have been encountered during the current period search. Each time a rate zero crossing is detected, the stage counter is incremented by one.

During the algorithm initialization, the parameter $stage$ is set to zero because no zero crossings have been encountered. The filter step counter c is set equal to $-DT$. This allows the period search routine twice the normal time to search for three zero crossings during the *first*

search only. If the ship motion has just passed through a zero crossing before the initialization, having $c = -DT$ during the initialization still allows us to capture the next three zero crossings. The parameter $c2$ is initialized to the value 1. Any positive value of $c2$ allows the algorithm to detect a zero crossing.

After each digital filter time step h , the counters c and $c2$ are incremented by 1. If $c2$ is positive and a zero crossing has been detected in $\dot{\phi}$, then the zero crossing counter $stage$ is incremented by one. The time of this zero crossing is recorded relative to when the current period search was initiated. Further, the parameter $c2$ is reset to $-P_{min}/(2h)$. This causes the algorithm to wait for $P_{min}/2$ seconds before it accepts another zero crossing. Since P_{min} is the minimum ship motion period that is expected, $P_{min}/2$ is the minimum time that may pass between the zero crossing of a pure sinusoidal motion of this period. The concern here is that if a noisy rate signal is used in this algorithm, then the noise could cause for some erroneous zero crossings. With the $c2$ counter, the algorithm waits for a small time before it starts to consider zero crossings again.

If three zero crossings have been detected, then the estimated ship motion period \tilde{P} is computed as the time difference between the third and first zero crossing. The time values $t(i)$ and the counter c are reset to zero. By resetting the $stage$ parameter to 1, and not zero, we are telling the algorithm to use the last zero crossing as the first zero crossing for the new period search.

The current ship motion period P is then updated in the algorithm. Instead of applying the entire difference between the current estimated and stored ship motion period, only a fraction α is used here with $0 < \alpha < 1$. This update equation is a standard technique used in gradient search algorithms. Adding the parameter α enhances the robustness of the algorithm to noise and other errors. The smaller α is, the longer it will take the algorithm to converge to a true, constant ship motion period. The benefit here is that any noise induced erroneous period estimates \tilde{P} will have a smaller effect on

the stored ship motion period P . On the other side, a larger value of α allows the algorithm to converge to a true ship motion period faster at the expense of decreased robustness to noise and measurement errors. A tradeoff study is performed to see what values of α are suitable for the chosen ship motion sensor and the associated sensor noise levels.

If the stored ship motion period P lies outside of the range $[P_{\min}, P_{\max}]$, then P is clipped to the closest period limit. This increases the robustness of the estimation algorithm by projecting any estimated period values to values within a pre-determined range of period values.

The last step of the self-tuning ω_c algorithm makes sure that not too much time has passed since the current zero crossing search was initiated. If more than P_{\max} time has passed, then the search counter c is reset to $-DT$. The counter $c2$ is set to 1. This avoids any waiting time before zero crossings are registered. The parameter `stage` is set to zero. This means that any last zero crossing measurement is discarded and the search algorithm starts out a fresh new search for three zero crossings.

The attitude roll, pitch and yaw rates are measured directly with the rate gyro instruments. Thus, these rate measurements could be used directly in this estimation technique. However, to estimate the surge, sway and heave translational periods, the IMU accelerometer information would have to be filtered and integrated. This would reduce the measurement noise and bias effects. If the IMU noise is small enough, then the accelerometer information could be used directly.

The presented self-tuning bandpass ship motion sensor filter can have the period estimation routine implemented for each state that is being filtered. However, this is excessive for the present ship motion measurement problem. The ship degrees of freedom steady-state sinusoidal periods do not differ that much from another. In particular, being anchored at sea, the translational motion is rather small. Thus, it is easier to estimate either the dominant roll or pitch motion. The code is setup to be able to estimate either.

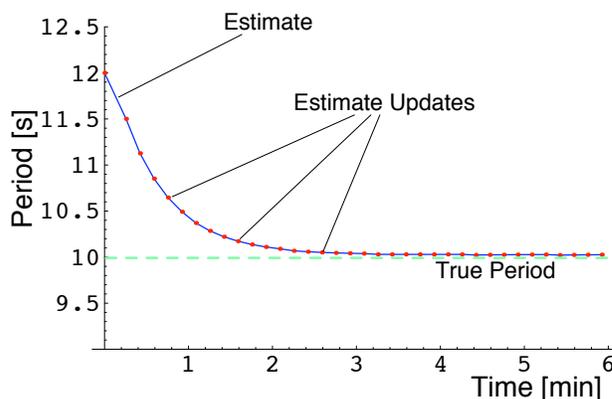


Figure 2.12: Illustration of Ship Period Estimation Routine.

The performance of the ship period estimation routine is illustrated in Figure 2.12. The true roll motion is set to have a period of 10 seconds, while the initial roll period estimate is set to 12 seconds. The update parameter α is set to 0.25. The period estimation error has been reduced to a fraction after only a 1 minute period. Due to the slow motion of the ship (about 10 second period), the period updates only occur about every 10 seconds. Gradually the estimate ship period converges to the true value. To make this convergence process more aggressive, the α value could be increased. However, the ship period should not change often (several times over a time span of minutes). The more conservative α value of 0.25 assures that any sensor noise induced erroneous readings will have a minimal impact on the integration and filtering process.

Chapter 3

Control Solution

3.1 Existing Position Based Solution

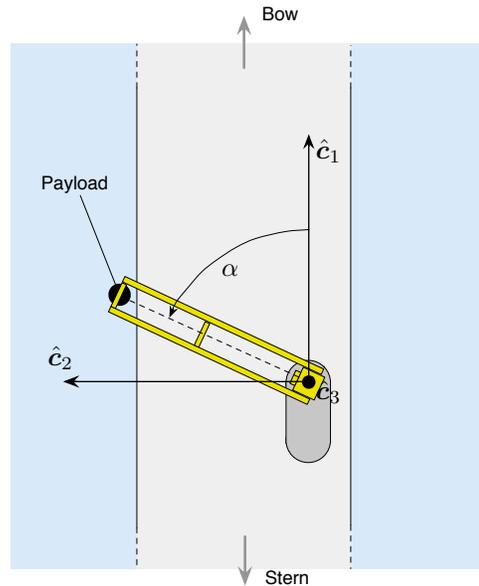
The existing position-based PCS solution, developed by Sandia National Laboratories and found in Reference 1, is outlined in this section. The two rate-based PCS solutions will make use of this position-based solution to find a stabilizing control.

3.1.1 Coordinate Frames

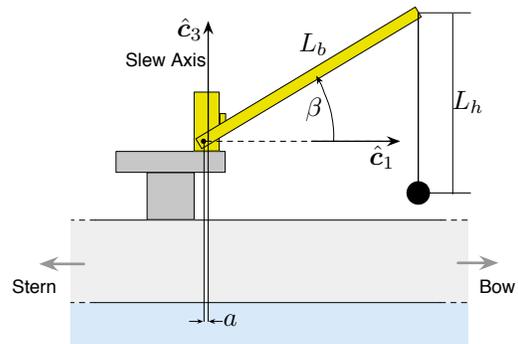
To describe the position-based PCS solution, several coordinate frames must be introduced. Let $\mathcal{C} : \{\hat{c}_1, \hat{c}_2, \hat{c}_3\}$ be the crane frame illustrated in Figure 3.1. The first \hat{c}_1 axis indicates the zero slew direction. If the boom is aligned with \hat{c}_2 , then the boom has rotated a slew angle α of $+90^\circ$. The \hat{c}_3 unit direction vector is aligned with the slew rotation axis of the crane.

The origin of the \mathcal{C} frame is on the crane slew axis, even with the boom hinge point as shown in Figure 3.1(b). Note that the crane frame is drawn with \hat{c}_1 pointing towards the bow, and \hat{b}_3 being vertical compared to the ship. However, this precise alignment is not required by the algorithm. Through careful survey measurements the relative orientation of the crane frame with respect to the ship sensor frame \mathcal{S} has been determined.

The boom length is expressed through the parameter L_b , while the hoist length L_h is defined as the distance from the boom tip to the payload. As noted earlier, the boom luffing hinge point is not at the crane frame \mathcal{C} origin. The hinge point is in the local-horizontal (\hat{c}_1, \hat{b}_2)



(a) Top View of Crane Frame \mathcal{C}



(b) Side View of Crane Frame \mathcal{C}

Figure 3.1: Crane Frame Illustration.

plane, but removed a distance $a > 0$ behind the slew axis.

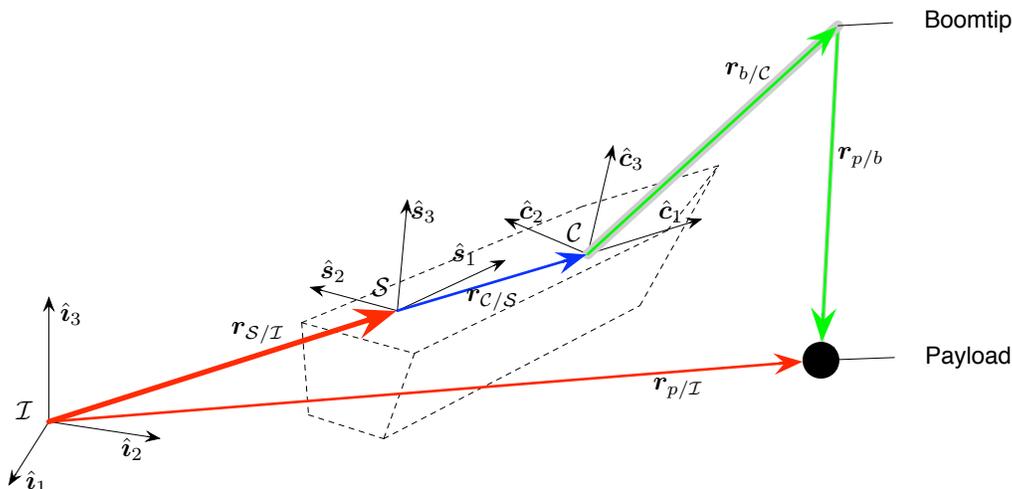


Figure 3.2: Illustration of Inertial, Ship and Crane Coordinate Frames.

Two additional coordinate frames used are the inertial frame \mathcal{I} and the ship sensor frame \mathcal{S} . The inertial frame \mathcal{I} used here could be a true inertial frame, or the slowly drifting frame \mathcal{I}' introduced earlier. We won't make any distinction in the following development. It is assumed that a ship motion sensor is measuring the translation and rotation motion of the ship (modeled as a rigid body) with respect to this inertial frame. The ship sensor frame $\mathcal{S} : \{\hat{s}_1, \hat{s}_2, \hat{s}_3\}$ is assumed to be mounted on a fixed location with respect to the rigid vessel as shown in Figure 3.2. Note that \hat{s} does not have to be pointing perfectly to the bow, nor does \hat{s}_3 have to be perfectly aligned with a ship normal vector. Instead, through careful surveying measurements, it is assumed that the relative position and orientation of the crane frame with respect to this ship sensor frame have been determined. Because the true ship sensor roll and pitch angle are measured, and used without being processed by a bandpass filter, it is still possible to determine the local gravity direction with respect to the ship frame.

To find the position-based inverse kinematics solution, we are given a desired (nominal) inertial payload position $\tilde{\mathbf{r}}_{p/\mathcal{I}}$ and would like to determine the require crane slew α , luff β and hoist L_h states that would ideally place the payload at this location given the current ship posi-

tion and orientation. To illustrate this answer, we express the inertia payload position vector as

$$\mathbf{r}_{p/\mathcal{I}} = \mathbf{r}_{\mathcal{S}/\mathcal{I}} + \mathbf{r}_{\mathcal{C}/\mathcal{S}} + \mathbf{r}_{b/\mathcal{C}} + \mathbf{r}_{p/b} \quad (3.1)$$

Note that these position descriptions are simply vectors and no coordinate frame component choice has been made yet. The ship sensor position vector is assumed to be given in inertial frame \mathcal{I} components. The coordinate frame with respect to which the vector components are taken is illustrated through a left superscript. The ship sensor 3×1 position vector is then written as ${}^{\mathcal{I}}\mathbf{r}_{\mathcal{S}/\mathcal{I}}$.² The position vector of the crane frame relative to the ship sensor frame will be expressed in \mathcal{S} components through ${}^{\mathcal{S}}\mathbf{r}_{\mathcal{C}/\mathcal{S}}$. The position vector of the boom tip with respect to the crane frame is expressed using \mathcal{C} frame components as

$${}^{\mathcal{C}}\mathbf{r}_{b/\mathcal{C}} = \begin{pmatrix} (L_b \cos \beta - a) \cos \alpha \\ (L_b \cos \beta - a) \sin \alpha \\ L_b \sin \beta \end{pmatrix} \quad (3.2)$$

The position vector of the payload relative to the boom tip is simply expressed using \mathcal{I} components as

$${}^{\mathcal{I}}\mathbf{r}_{p/b} = \begin{pmatrix} 0 \\ 0 \\ -L_h \end{pmatrix} \quad (3.3)$$

because the inertial $\hat{\mathbf{i}}_3$ is chosen such that it aligns with the local gravity force direction. The relative orientations of the inertial \mathcal{I} , ship sensor \mathcal{S} and crane \mathcal{C} coordinate frames will be expressed through 3×3 rotation matrices.² For example, let the rotation matrix $[IS]$ map a vector with \mathcal{S} frame components into a vector with inertial frame components. Similarly $[SC]$ is defined as mapping crane frame vector components into ship frame components. Using the ship yaw, pitch and roll angles, we will be able to compute the $[IS]$ rotation matrix. The constant rotation matrix $[SC]$ is found through the calibration angles determined from surveying the crane frame position relative to the ship sensor frame. Finally, $[IC]$ is found using matrix product

$$[IC] = [IS][SC] \quad (3.4)$$

To get the inverse transformation of a rotation matrix, we simply transpose the matrix (thus $[CI] = [IC]^{-1} = [IC]^T$).

3.1.2 Position-Based Inverse Kinematics Solution

With the previous coordinate frames defined, we are now able to express the position-based inverse kinematics solution. Eq. (3.1) is rewritten using specific coordinate frames as

$$\begin{aligned} \mathcal{I}\tilde{\mathbf{r}}_{p/\mathcal{I}} &= \mathcal{I}\mathbf{r}_{\mathcal{S}/\mathcal{I}} + [IS] \mathcal{S}\mathbf{r}_{\mathcal{C}/\mathcal{S}} \\ &+ [IC] \mathcal{C}\mathbf{r}_{b/\mathcal{C}} + \mathcal{I}\mathbf{r}_{p/b} \end{aligned} \quad (3.5)$$

The position vector $\tilde{\mathbf{r}}_{p/\mathcal{I}}$ is assumed to be the desired (nominal) inertial payload position. Assuming no payload swing is present, the payload will hang straight down along the gravity direction vector. For the position-based inverse kinematics solution, we assume that the payload is not swinging and determine the ideal crane states to keep the payload at the desired inertial position. Let us introduce the gravity direction vector $\hat{\mathbf{g}}$ as

$$\hat{\mathbf{g}} = -\hat{\mathbf{i}}_3 = \begin{pmatrix} g_1 \\ g_2 \\ g_3 \end{pmatrix} = [CI] \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} \quad (3.6)$$

where g_i are the crane frame \mathcal{C} vector components of the unit vector $\hat{\mathbf{g}}$. Using $\hat{\mathbf{g}}$, we can write the payload position vector with respect to the boom tip as

$$\mathcal{C}\mathbf{r}_{p/b} = L_h \mathcal{C}\hat{\mathbf{g}} \quad (3.7)$$

Eq. (3.5) can be rewritten in \mathcal{C} frame components as

$$\begin{aligned} [CI] (\mathcal{I}\tilde{\mathbf{r}}_{p/\mathcal{I}} - \mathcal{I}\mathbf{r}_{\mathcal{S}/\mathcal{I}} - [IS] \mathcal{S}\mathbf{r}_{\mathcal{C}/\mathcal{S}}) \\ = \mathcal{C}\mathbf{r}_{b/\mathcal{C}} + L_h \mathcal{C}\hat{\mathbf{g}} \end{aligned} \quad (3.8)$$

Let us introduce the vector $\boldsymbol{\xi}$ as

$$\mathcal{C}\boldsymbol{\xi} = [CI] (\mathcal{I}\tilde{\mathbf{r}}_{p/\mathcal{I}} - \mathcal{I}\mathbf{r}_{\mathcal{S}/\mathcal{I}} - [IS] \mathcal{S}\mathbf{r}_{\mathcal{C}/\mathcal{S}}) \quad (3.9)$$

The inverse kinematic problem needs to solve the vector equation

$$\mathcal{C}\boldsymbol{\xi} = \begin{pmatrix} (L_b \cos \beta - a) \cos \alpha + L_h g_1 \\ (L_b \cos \beta - a) \sin \alpha + L_h g_2 \\ L_b \sin \beta + L_h g_3 \end{pmatrix} \quad (3.10)$$

for the crane states α , β and L_h . This nonlinear equation can be solved explicitly, but requires a complicated solution of a quintic polynomial function. The actual algorithm is not shown here in this report.

To specify the desired inertial payload position relative to the inertial \mathcal{I} , the nominal ship concept is introduced. The nominal ship frame is at rest with zero translation and rotation relative to the inertial frame. Associated with the nominal ship frame are nominal crane states. Think of these as the average values of the true, time-varying crane states. If the operator commands a slew-left maneuver, then the nominal slew angle is increased. This results in the payload moving through inertial space as if the ship was not translating and rotating. The nominal payload position also assumes that no payload swing is present. Given nominal slew $\tilde{\alpha}$ and luff $\tilde{\beta}$ states, the nominal boom tip position relative to the crane frame \mathcal{C} is expressed as

$$\mathcal{C}\tilde{\mathbf{r}}_{b/\mathcal{C}} = \begin{pmatrix} (L_b \cos \tilde{\beta} - a) \cos \tilde{\alpha} \\ (L_b \cos \tilde{\beta} - a) \sin \tilde{\alpha} \\ L_b \sin \tilde{\beta} \end{pmatrix} \quad (3.11)$$

Using the nominal hoist length \tilde{L}_h , the position vector of the payload relative to the boom tip is expressed as

$${}^{\mathcal{I}}\tilde{\mathbf{r}}_{p/b} = \begin{pmatrix} 0 \\ 0 \\ -\tilde{L}_h \end{pmatrix} \quad (3.12)$$

Thus, the nominal inertial payload position is computed using

$${}^{\mathcal{I}}\tilde{\mathbf{r}}_{p/\mathcal{I}} = ({}^{\mathcal{S}}\tilde{\mathbf{r}}_{c/\mathcal{S}} + [SC] c\tilde{\mathbf{r}}_{b/c}) + {}^{\mathcal{I}}\delta\mathbf{r} + {}^{\mathcal{I}}\tilde{\mathbf{r}}_{p/b} \quad (3.13)$$

where $\delta\mathbf{r}$ are prescribed boom tip motions to damping out any existing swing. The Cartesian damping corrections are not covered in detail in this report. Simply assume that $\delta\mathbf{r}$ is a small position vector which will asymptotically go to zero as the swing angles go to zero. At first glance it might appear that we are adding up vector components mixed across the \mathcal{S} and \mathcal{I} frames. However, recall that the nominal ship frame is assumed to have the same attitude as the inertial frame. Thus, for the nominal position calculations, we find that $[IS]$ is the identity matrix (no orientation difference). Given the current nominal crane states $\tilde{\alpha}$, $\tilde{\beta}$ and \tilde{L}_h , as well as the Cartesian damping correction $\delta\mathbf{r}$ to the boom tip motion, Eq. (3.13) is used to compute the nominal payload position $\tilde{\mathbf{r}}_{p/\mathcal{I}}$.

3.1.3 Control Flowchart

A general overview of the position-based PCS concept is shown in Figure 3.3. The input signals are the user joystick signal, the crane joint absolute and incremental encoder values, the swing resolver values, as well as the POS/MV ship motion measurements surge, sway, heave, yaw, pitch and roll with respect to some inertial frame \mathcal{I} . The ship position states are processed through an estimator algorithm which provides states at equal control time steps. Next, the current estimated ship states are filtered and translated to slowly moving \mathcal{I}' frame. These ship position states are finally provided to the position based PCS algorithm, which returns

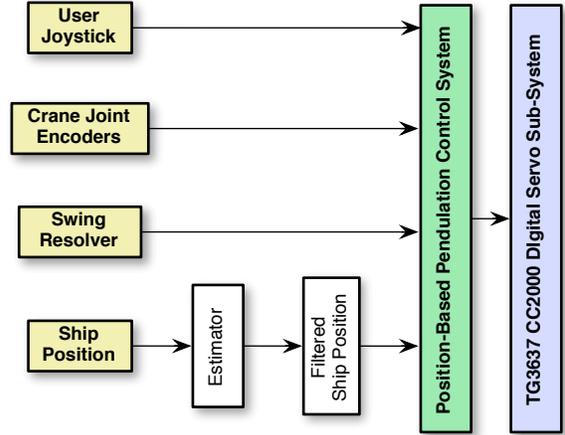


Figure 3.3: Illustration of General Position-Based PCS Concept.

crane servo rates to the CC2000 digital control system.

A flowchart of the existing position-based PCS algorithm is shown in Figure 3.4. The inputs to the PCS are shown in yellow boxed and include the user joystick commands (commanded nominal crane motion), the crane joint encoder information (both absolute and incremental encoder readings), the swing sensor resolver information, as well as the ship motion sensor (surge, sway, heave, yaw, pitch and roll states).

Let us begin the flowchart discussion by following the joystick signal and monitoring how they get processed. The user provides joystick signal which determine how fast the crane joint (slew, luff or hoist length) are to change. These nominal crane joint speed commands are clipped if the crane is too close to a joint limit. Next, the joystick signal is processing by a self-tuning input shaping algorithm.³ Here the input signal is filtered to remove the frequency content which is close to the payload pendulation frequency. The resulting commanded crane rates are then integrated over one control time step to yield new nominal crane states. Recall that the nominal crane is assumed to be on a non-moving ship. The nominal ship and crane states are used to compute the nominal, inertial payload position.

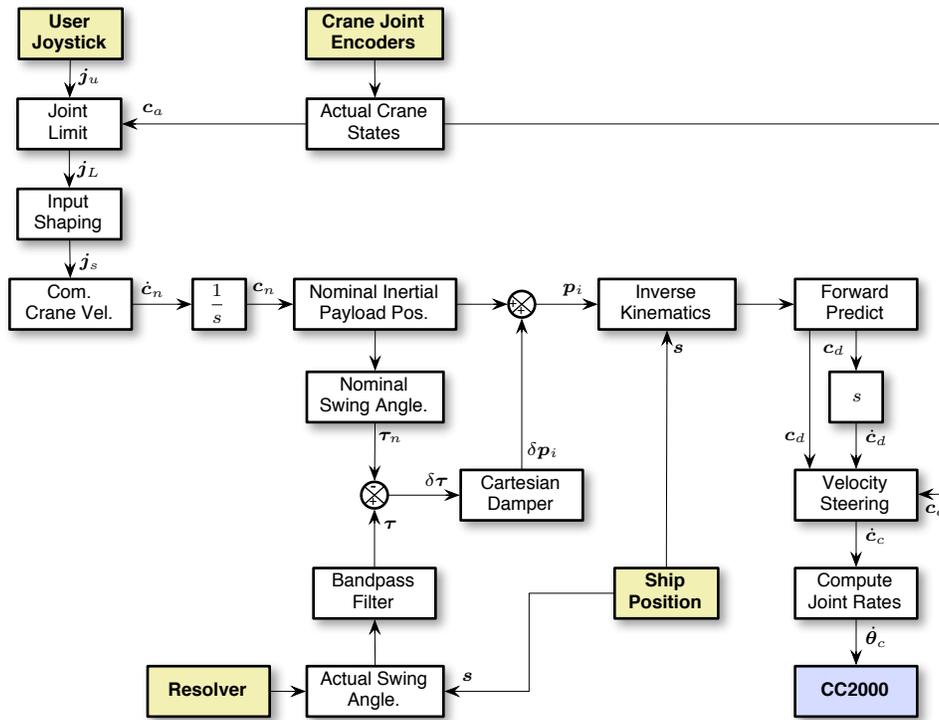


Figure 3.4: Flow Chart of the Position-Based PCS Solutions.

The nominal crane motion determines how the boom tip will horizontally move through the inertial space. Even with an input-shaped velocity command, there will be some non-zero payload deflection with respect to the gravity vector. The nominal crane states are used to compute how much nominal swing is expected for the given maneuver. For example, if the user commands the crane to move to the left, then the payload will lag the boom tip motion slightly to begin with. As the constant rotation to the left is continued, the payload will asymptotically move back to a zero swing angle condition. These payload deflections are expected and required for input-shaped maneuvers to succeed. To avoid having the damper try to cancel these required swing angles, the actual swing angles are differenced with the nominal swing angles to compute the payload swing error states. A Cartesian damping strategy is used to compute a control solution to dampen out any erroneous swing. By tracking the nominally expected swing angles, we are able to decouple the input shaper from the swing damper

and provide a system that is more responsive and better performing.

The Cartesian damper computes a required boom tip correction vector $\delta \mathbf{r}$ which determines how much the boom should move relative to the nominal position to dampen out any swing. The damper is setup such that the damping corrections $\delta \mathbf{r}$ will go to zero as the swing angle become zero. The damping correction of the payload is added to the nominal, inertial payload position to find the current ideal inertial payload position. Using the current ship position states, the position-based inverse kinematics problem is solved for the commanded crane slew, luff and hoist states that would result in the payload being in the desired inertial position. The commanded crane states are passed through a forward prediction filter which is capable of compensating for very small communication delays.

The next step is to turn these commanded crane states into a crane state rate command. By differencing the commanded crane states with the result of the previous command time

step, the commanded crane states are numerically differentiated. A low pass filter is used in this step as well to avoid amplifying any noise introduced in the commanded signal. This filtered differentiation step will thus introduce a phase lag into the commanded velocity signal. The amplitude of this lag is controlled through the filter gains. Next, the velocity steering law compared the actual crane states with the commanded crane states. If this crane state error is non-zero, then a proportional feedback law is used to drive such tracking errors to zero. In control language terms, an integral feedback terms is imposed on the open-loop velocity command to avoid secular drift. This velocity steering term also has setup to handle servo saturation situations smoothly.

The final step is to map the slew, luff and hoist rates to actual slew gear, luff drum and hoist drum speed commands. The final answer is sent to the digital CC2000 crane control system. It receives these commands and sends appropriate command currents to the sub-servo systems to track the provided speed command.

The performance of the shown position-based PCS concept is determined through the accuracy of the various sensors, as well as the accuracy of the hydraulic crane servo sub-systems. Note that the complete ship position information is required to compute a solution. Further, any noise introduced by the sensors becomes amplified through the commanded crane rate differentiation process. The low-pass filter does help reduce the resulting noise level, but an additional phase lag is introduced. The PCS performance is sensitive to having a large phase lag in the motion compensation component of the control law. Because the final commanded crane position is differentiated (computed to compensate for both ship motion, perform the user commanded motion, as well as compensate for damping), the filter parameters must be set very conservatively to avoid harsh performance penalties. The following rate-based PCS concepts will provide equivalent control solutions assuming that the sensor have no errors, and the numerical differentiations are perfect. PCS performance

difference will only manifest themselves once realistic sensor errors are introduced.

3.2 Rate-Based Control Solution Using Only Ship Motion Integration

In section 2.3 a method is outlined to integrate the ship motion sensor accelerometer and rate gyro data to obtain the ship motion with respect to the slowly drifting \mathcal{I}' frame. One rate-based PCS concept is to use this integration and selective filtering process to obtain estimates of the current ship motion position states, and provide these to the same position-based PCS algorithm. The integrated-filtered ship states are equivalent to the \mathcal{I}' states. This concept is outlined in Figure 3.5

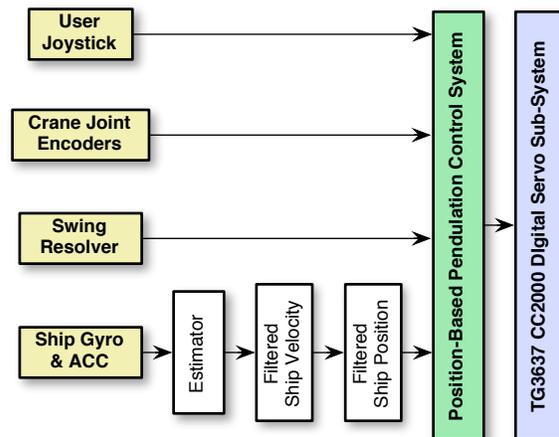


Figure 3.5: Illustration of Rate-Based PCS Concept using Ship Motion Integration.

No code change to the control algorithm would be needed. The estimation step would need to be expanded to estimate the three accelerometer measurements, the three rate gyro measurements, as well as the pitch and roll angles. The ship motion filtering process is replaced with an entirely new algorithm which will integrate and filter the ship motion measurements to obtain first filtered velocities, and finally filtered ship position states, as outlined in section 2.3.

As with any solution based on measuring the ship acceleration and angular rates, this solution will be sensitive to the numerical integration issues of corrupted measurements. If the measurement biases and noise levels are small enough, then the resulting estimated ship motion should be sufficient to provide a robust PCS solution.

Because this control solution using the existing position-based PCS algorithm outlined in Figure 3.4, it still requires that the required crane states are numerically differentiated to obtain servo crane rates. Without careful filtering in this differentiation process, any measurement noise in the ship attitude, motion, or swing angles will be amplified.

3.3 Rate-Based Control Solution Using Velocity Kinematics

The following rate-based PCS solution will assume that the ship motion is sensed using accelerometer and rate gyro information, as well as true roll and pitch angles, and compute the current ship motion rates. These rates are then used to directly compute required crane rates to compensate for the measured ship motion using a velocity-based inverse kinematics routine.

3.3.1 Velocity-Based Inverse Kinematics

In section 3.1.1 the inertial coordinate frame \mathcal{I} , the ship sensor coordinate frame \mathcal{S} , and the crane coordinate frame \mathcal{C} were introduced. In section 3.1.2 the position-based inverse kinematics routine is outlined. This section will use these previous results to derive the velocity-based inverse kinematics routine.

The inertial payload position $\mathbf{r}_{p/\mathcal{I}}$ is found using Eq. (3.1)

$$\mathbf{r}_{p/\mathcal{I}} = \mathbf{r}_{\mathcal{S}/\mathcal{I}} + \mathbf{r}_{\mathcal{C}/\mathcal{S}} + \mathbf{r}_{b/\mathcal{C}} + \mathbf{r}_{p/b}$$

The inertial payload velocity $\dot{\mathbf{r}}_{p/\mathcal{I}}$ is then found by taking the inertial derivative of this equa-

tion.

$$\dot{\mathbf{r}}_{p/\mathcal{I}} = \dot{\mathbf{r}}_{\mathcal{S}/\mathcal{I}} + \dot{\mathbf{r}}_{\mathcal{C}/\mathcal{S}} + \dot{\mathbf{r}}_{b/\mathcal{C}} + \dot{\mathbf{r}}_{p/b} \quad (3.14)$$

Let us use the following notation. The derivative $\frac{S_d \mathbf{x}}{dt}$ is said to be the derivative of \mathbf{x} as seen by the \mathcal{S} frame, where \mathcal{S} could be a frame rotating with an inertial rate $\boldsymbol{\omega}_{\mathcal{S}/\mathcal{I}}$. Because the $\mathbf{r}_{\mathcal{C}/\mathcal{S}}$ vector will be expressed using \mathcal{S} frame components, and $\mathbf{r}_{b/\mathcal{C}}$ will be expressed using \mathcal{C} frame components, the inertial payload velocity is expressed as:

$$\begin{aligned} \dot{\mathbf{r}}_{p/\mathcal{I}} = & \dot{\mathbf{r}}_{\mathcal{S}/\mathcal{I}} + \frac{S_d}{dt} (\mathbf{r}_{\mathcal{C}/\mathcal{S}}) + \boldsymbol{\omega}_{\mathcal{S}/\mathcal{I}} \times \mathbf{r}_{\mathcal{C}/\mathcal{S}} \\ & + \frac{C_d}{dt} (\mathbf{r}_{b/\mathcal{C}}) + \boldsymbol{\omega}_{\mathcal{S}/\mathcal{I}} \times \mathbf{r}_{b/\mathcal{C}} + \dot{\mathbf{r}}_{p/b} \end{aligned} \quad (3.15)$$

We make use of the fact here that $\boldsymbol{\omega}_{\mathcal{S}/\mathcal{I}} = \boldsymbol{\omega}_{\mathcal{C}/\mathcal{I}}$. Note that the ship angular velocity vector can be expressed as

$$\boldsymbol{\omega}_{\mathcal{S}/\mathcal{I}} = \begin{matrix} S \\ \left(\begin{array}{c} \omega_1 \\ \omega_2 \\ \omega_3 \end{array} \right) \end{matrix} \quad (3.16)$$

where $(\omega_1, \omega_2, \omega_3)$ are the three rate measurements of the ship gyro sensor. Further, note that since $\mathbf{r}_{\mathcal{C}/\mathcal{S}}$ is a constant vector as seen by the ship sensor frame, we find that

$$\frac{S_d}{dt} (\mathbf{r}_{\mathcal{C}/\mathcal{S}}) = 0 \quad (3.17)$$

The crane boom tip position vector $\mathbf{r}_{b/\mathcal{C}}$ is expressed in \mathcal{C} frame components using the slew angle α and luff angle β as (see Eq. (3.2)):

$${}^C \mathbf{r}_{b/\mathcal{C}} = \begin{pmatrix} (L_b \cos \beta - a) \cos \alpha \\ (L_b \cos \beta - a) \sin \alpha \\ L_b \sin \beta \end{pmatrix}$$

The derivative of this boom tip position vector as seen by the \mathcal{C} frame is

$$\begin{aligned} \frac{C_d}{dt} ({}^C \mathbf{r}_{b/\mathcal{C}}) = & \\ & {}^C \begin{pmatrix} -(L_b c \beta - a) s \alpha \dot{\alpha} - L_b s \beta c \alpha \dot{\beta} \\ (L_b c \beta - a) c \alpha \dot{\alpha} - L_b s \beta s \alpha \dot{\beta} \\ L_b c \beta \dot{\beta} \end{pmatrix} \end{aligned} \quad (3.18)$$

where the short-hand $c\alpha = \cos \alpha$ and $s\alpha = \sin \alpha$ is used.

The payload position vector $\mathbf{r}_{p/b}$ relative to the boom tip is expressed in inertial \mathcal{I} vector components in Eq. (3.7) as:

$$\mathbf{r}_{p/b} = \begin{matrix} \mathcal{I} \\ \left(\begin{array}{c} 0 \\ 0 \\ -L_h \end{array} \right) \end{matrix} = L_h \hat{\mathbf{g}}$$

where $\hat{\mathbf{g}} = -\hat{\mathbf{i}}_3$ is the local gravity unit direction vector. The inertial derivative of this vector is expressed as

$$\dot{\mathbf{r}}_{p/b} = \begin{matrix} \mathcal{I} \\ \left(\begin{array}{c} 0 \\ 0 \\ -\dot{L}_h \end{array} \right) \end{matrix} = \dot{L}_h \hat{\mathbf{g}} \quad (3.19)$$

The gravity vector is trivially expressed in the inertial frame \mathcal{I} . However, we need this vector to be expressed in crane frame \mathcal{C} vector components. Using the rotation matrix $[IC]$, we find that

$${}^c \hat{\mathbf{g}} = \begin{pmatrix} g_1 \\ g_2 \\ g_3 \end{pmatrix} = [IC]^T \mathcal{I} \hat{\mathbf{g}} = [IC]^T \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} \quad (3.20)$$

Substituting Eqs. (3.30) and (3.19) into Eq. (3.14), we find

$$\begin{aligned} & \mathcal{N} \dot{\mathbf{r}}_{p/\mathcal{I}} - \mathcal{N} \dot{\mathbf{r}}_{S/\mathcal{I}} - [IS] \left({}^S \boldsymbol{\omega}_{S/\mathcal{I}} \times {}^S \mathbf{r}_{C/S} \right. \\ & \quad \left. + {}^S \boldsymbol{\omega}_{S/\mathcal{I}} \times [SC] {}^c \mathbf{r}_{b/c} \right) \\ & = [IC] \left(\frac{c_d}{dt} ({}^c \mathbf{r}_{b/c}) + [CN] \mathcal{N} \dot{\mathbf{r}}_{p/b} \right) \end{aligned} \quad (3.21)$$

where we are now explicitly writing vector components with respect to the \mathcal{I} , \mathcal{S} and \mathcal{C} frame. The 3×3 matrix $[IC]$ is the rotation matrix mapping crane frame vector components to inertial frame components. The matrices $[SC]$ and $[IS]$ are equivalent rotation matrices between the \mathcal{S} , \mathcal{C} and \mathcal{I} frames. The left hand side of this equation is known if we know the prescribed inertial payload velocity $\mathcal{N} \dot{\mathbf{r}}_{p/\mathcal{I}}$, the inertial ship motion $\mathcal{N} \dot{\mathbf{r}}_{S/\mathcal{I}}$, the ship attitude matrix $[IS]$, the ship rotation rate ${}^S \boldsymbol{\omega}_{S/\mathcal{I}}$, as well

as the current boom tip position vector ${}^c \mathbf{r}_{b/c}$. The right hand side of this equation contains all terms that depend on the in-question slew, luff and hoist rates. Let us introduce the vector $\boldsymbol{\xi}$ as

$${}^c \boldsymbol{\xi} = [CI] \left(\mathcal{N} \dot{\mathbf{r}}_{p/\mathcal{I}} - \mathcal{N} \dot{\mathbf{r}}_{S/\mathcal{I}} - [IS] \left({}^S \boldsymbol{\omega}_{S/\mathcal{I}} \times {}^S \mathbf{r}_{C/S} \right. \right. \\ \left. \left. + {}^S \boldsymbol{\omega}_{S/\mathcal{I}} \times [SC] {}^c \mathbf{r}_{b/c} \right) \right) \quad (3.22)$$

Then the payload velocity condition can be written compactly as

$${}^c \boldsymbol{\xi} = \frac{c_d}{dt} ({}^c \mathbf{r}_{b/c}) + \dot{L}_h {}^c \hat{\mathbf{g}} \quad (3.23)$$

This vector equation can be expressed in matrix form as

$${}^c \boldsymbol{\xi} = \underbrace{\begin{bmatrix} -(L_b - a)c\beta s\alpha & -L_b s\beta c\alpha & g_1 \\ (L_b - a)c\beta c\alpha & -L_b s\beta s\alpha & g_2 \\ 0 & L_b c\beta & g_3 \end{bmatrix}}_{[A]} \begin{pmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{L}_h \end{pmatrix} \quad (3.24)$$

Given the sensor measurements and a nominal inertial payload velocity, we can compute the $\boldsymbol{\xi}$ vector. Using the above equation, the velocity-based inverse kinematic solution requires that

$$\begin{pmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{L}_h \end{pmatrix} = [A]^{-1} {}^c \boldsymbol{\xi} \quad (3.25)$$

Let the determinant Δ of the $[A]$ matrix be given by

$$\begin{aligned} \Delta & = L_b(L_b \cos \beta - a)(g_1 \cos \alpha \cos \beta \\ & \quad + g_2 \sin \alpha \cos \beta + g_3 \sin \beta) \end{aligned} \quad (3.26)$$

The matrix inverse is then given by Eq. (3.27).

The velocity-based inverse kinematic solution in Eq. (3.25) requires a nominal inertial payload motion. Using the nominal crane concept from the position-based inverse kinematic solution, we can compute a similar nominal inertial payload velocity vector $\dot{\mathbf{r}}_{p/\mathcal{I}}$. The nominal

$$[A]^{-1} = \frac{1}{\Delta} \begin{bmatrix} -L_b(g_2c\beta + g_3sas\beta) & L_b(g_1c\beta + g_zcas\beta) & L_b(g_1s\alpha - g_2c\alpha)s\beta \\ -(L_b c\beta - a)g_3c\alpha & -(L_b c\beta - a)g_3s\alpha & (L_b c\beta - a)(g_1c\alpha + g_2s\alpha) \\ L_b(L_b c\beta - a)c\alpha c\beta & L_b(L_b c\beta - a)c\beta s\alpha & L_b(L_b c\beta - a)s\beta \end{bmatrix} \quad (3.27)$$

inertial payload position was computed using Eq. (3.13)

$$\tilde{\mathbf{r}}_{p/\mathcal{I}} = (\tilde{\mathbf{r}}_{c/s} + \tilde{\mathbf{r}}_{b/c}) + \delta\mathbf{r} + \tilde{\mathbf{r}}_{p/b}$$

where $\delta\mathbf{r}$ are the Cartesian damping corrections to the inertial (x, y) motion of the payload to damping out any payload swing. The nominal boom tip vector $\tilde{\mathbf{r}}_{b/c}$ is expressed using the nominal slew angle $\tilde{\alpha}$ and luff angle $\tilde{\beta}$ in Eq. (3.11) using crane frame \mathcal{C} vector components as

$${}^c\tilde{\mathbf{r}}_{b/c} = \begin{pmatrix} \left(L_b \cos \tilde{\beta} - a \right) \cos \tilde{\alpha} \\ \left(L_b \cos \tilde{\beta} - a \right) \sin \tilde{\alpha} \\ L_b \sin \tilde{\beta} \end{pmatrix}$$

The nominal payload position $\tilde{\mathbf{r}}_{p/b}$ with respect to the boom tip is expressed in Eq. (3.12) using inertial frame \mathcal{I} vector components as:

$${}^{\mathcal{I}}\tilde{\mathbf{r}}_{p/b} = \begin{pmatrix} 0 \\ 0 \\ -\tilde{L}_h \end{pmatrix}$$

Taking the inertial derivative of $\tilde{\mathbf{r}}_{p/\mathcal{I}}$, the nominal inertial payload velocity vector is

$$\dot{\tilde{\mathbf{r}}}_{p/\mathcal{I}} = \left(\dot{\tilde{\mathbf{r}}}_{b/c} \right) + \delta\dot{\mathbf{r}} + \dot{\tilde{\mathbf{r}}}_{p/b} \quad (3.28)$$

where the short-hand notation $\frac{{}^{\mathcal{I}}d\mathbf{x}}{dt} \equiv \dot{\mathbf{x}}$ is used. Because the nominal ship is not rotating, we find that the nominal ship sensor and nominal crane frame are inertial frames as well. The derivative of the boom tip position vector $\tilde{\mathbf{r}}_{b/c}$ is

$${}^c\dot{\tilde{\mathbf{r}}}_{b/c} = \begin{pmatrix} \left(L_b \cos \tilde{\beta} - a \right) \dot{\cos \tilde{\alpha}} \\ \left(L_b \cos \tilde{\beta} - a \right) \dot{\sin \tilde{\alpha}} \\ L_b \dot{\sin \tilde{\beta}} \end{pmatrix} \quad (3.29)$$

$$\begin{aligned} \frac{{}^{\mathcal{N}}d}{dt} ({}^c\mathbf{r}_{b/c}) &= \frac{c_d}{dt} ({}^c\mathbf{r}_{b/c}) = c\dot{\mathbf{r}}_{b/c} = \\ & c \begin{pmatrix} - \left(L_b c \tilde{\beta} - a \right) s \tilde{\alpha} \dot{\tilde{\alpha}} - L_b s \tilde{\beta} c \tilde{\alpha} \dot{\tilde{\beta}} \\ \left(L_b c \tilde{\beta} - a \right) c \tilde{\alpha} \dot{\tilde{\alpha}} - L_b s \tilde{\beta} s \tilde{\alpha} \dot{\tilde{\beta}} \\ L_b c \tilde{\beta} \dot{\tilde{\beta}} \end{pmatrix} \end{aligned} \quad (3.30)$$

The inertia damping correction rates $\delta\dot{\mathbf{r}}$ must still be computed. The algorithm uses the existing damping correction algorithm which produces $\delta\mathbf{r}$ vectors, which are then numerically differentiated to obtain the $\delta\dot{\mathbf{r}}$ damping correction rates. With this rate-based PCS concept, the only sensor states that will need to be differentiated are the swing sensor states. However, the swing-resolver measurements are fine-resolution digital measurements with very little noise. The inertial swing states are then passed through a bandpass filter to remove any secular terms or biases. Thus, the damping corrections which are computed using these filtered swing angles are very clean, and the numerical differentiation is not expected to significantly amplify any noise.

3.3.2 Control Flowchart

The general flowchart of the rate-based PCS strategy with velocity-based inverse kinematics is shown in Figure 3.6. The ship motion is still being sensed using accelerometer and rate gyro sensors. The estimated \mathcal{I}' ship motion is obtained after careful integration and bandpass filtering of the states. However, instead of only using the ship position states, both the inertial position and velocity ship states are passed on to the new PCS control algorithm.

Figure 3.7 illustrates the new rate-based PCS control algorithm. The flow chart is discussed by tracking what happens to the user crane joystick signals. As before, these signals are translated into commanded nominal crane ve-

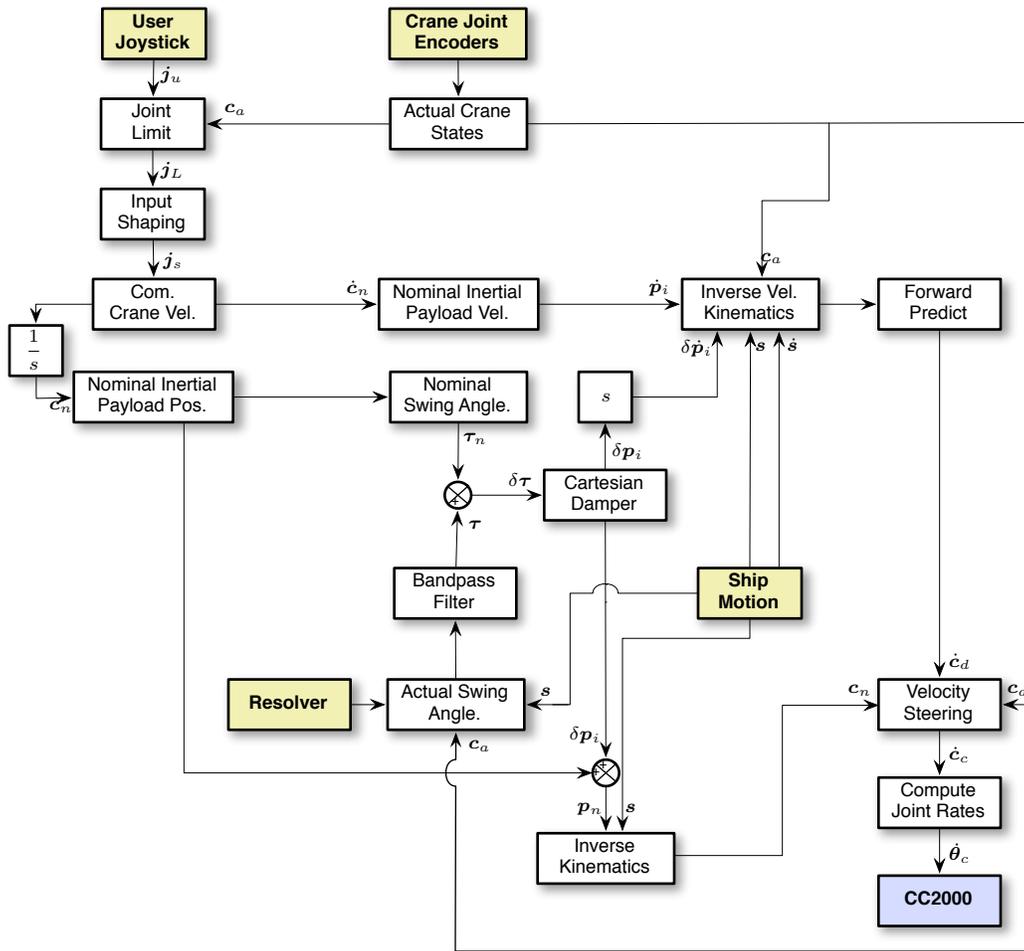


Figure 3.6: Flow Chart of the Rate-Based PCS Solutions.

locities. However, instead of immediately integrating this rates to obtain new nominal crane states, the crane rates \dot{c}_n are used directly to compute the nominal inertial payload velocity vector. The crane encoder input signals are used to compute the actual crane states c_a , while the swing resolve input signal is used to compute filtered, inertial swing angles, and a corresponding damping correction term $\delta\dot{r}$. These states, along with the current ship position states s and rates \dot{s} , are then used to determine the inverse velocity kinematics solution. The previous position-based inverse kinematic solution is computed as before. However, the nominal crane states c_n are now used in the velocity steering law which will track the desired crane rates \dot{c}_d and bounds the difference

between the nominal crane states c_n and the actual crane states c_a . Without this feedback term on c_n , only measuring ship rates, we would have secular drift issues where the actual crane states would deviate the desired states.

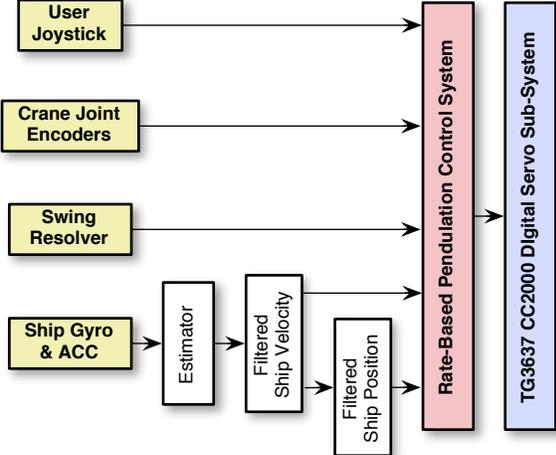


Figure 3.7: Illustration of Rate-Based PCS Concept using Velocity Kinematics.

Chapter 4

Conclusions

Two control methodologies to drive the PCS with only IMU (rate gyro and accelerometer) information are presented. Both methods require filtered ship position states (surge, sway, heave, yaw, pitch and roll). A digital algorithm is presented to integrate and filter the ship motion sensor data and obtain the ship motion with respect to a slowly drifting inertia frame \mathcal{I}' . This frame drifts with the nominal ship motion, where yaw and translation are measured relative to this frame. The integration is performed synchronously with a bandpass filtering process. The end result is equivalent to the \mathcal{I}' frame ship motion that is currently computed with the existing PCS installation. Differences will be due to having different sensor error behaviors. Changes to the ship sensor communication routine will be required for all control methods discussed in this report. The filtering, integration, and estimation routines will all need to be updated.

One velocity-based PCS strategy is use the integrated and filtered ship motion sensor data, and feed it directly to the existing position-based PCS solution. With perfect sensor data, and perfect ship motion filtering, the integrated solution and the velocity based solution are equivalent. This strategy has the benefit that no change to the actual control algorithm will need to be made. As such, this strategy should also work with the deck-tracking algorithm without changes to the control code. However, the current position based PCS strategy requires a numerical differentiation step of the commanded crane states. The commanded crane states are computed using *information*

from all sensor. Thus, any sensor noise in the ship position states or the swing angles, or the discretization issues of the crane encoder signals, become amplified by this differentiation process. This is the reason that a lowpass filter is combined with this differential operator to reduce the noise amplification, yet not introduce too much phase lag.

The second strategy creates a new velocity-based inverse kinematics control strategy. Here the ship velocity is estimated, which is then combined with the user operator joystick crane speed command, to create directly the commanded crane speeds. Note that the commanded crane states are not numerically differentiated in this process. The only differentiation that occurs with this strategy is that of the Cartesian damping corrections. However, the swing angles are measured very accurately with digital resolvers. These states are then fed through a bandpass filter to compute the inertial swing angles about the local gravity vector. Thus, the swing states used in the control algorithm are very smooth, and numerically differentiating them should not be an issue. However, this strategy will require an extensive rewrite of the actual control algorithm. Only control crane rates, any errors in the implemented speed commands will lead to an unstable solution. Thus, the previous position-based inverse kinematics solution is computed at the same time and used in the velocity-steering law to bound the crane state errors.

Both velocity-based control methods have been implemented into the CraneSim simulation. Future work will model ship IMU sensors

in more detail be including biases and noise levels typical with ship motion sensors being considered.

Bibliography

- [1] Robinett, R. D., Groom, K. N., Feddema, J. T., and Parker, G. G., “Control system and method for payload control in mobile platform cranes,” December 17 2002, Patent Number 4,596,765.
- [2] Schaub, H. and Junkins, J. L., *Analytical Mechanics of Space Systems*, AIAA Education Series, Reston, VA, October 2003.
- [3] Parker, G. G., Groom, K. N., Hurtado, J. E., Feddema, J., Robinett, R. D., and Leban, F., “Experimental verification of a command shaping boom crane control system,” *IEEE Proceedings of the American Control Conference*, San Diego, CA, 1999, pp. 86–90.