Proceedings of The Joint Canadian Society for Mechanical Engineering and CFD Society of Canada International Congress 2019
CSME-CFDSC Congress 2019
June 2-5, 2019, London, On, Canada

Kalman Filtering and PID Control of an Inverted Pendulum Robot

Claire Bourque, Andrew Lee, Elyse Hill, S. Andrew Gadsden*, Alex Bardelcik and Mohammad Biglarbegian
*Department of Mechanical Engineering
University of Guelph, Guelph, Ontario, Canada
Emails: cbourque, alee32, ehill06, gadsden, abardelc, mbiglarb@uoguelph.ca

***Abstract*—The inverted pendulum is a classical controls theory problem that is unstable and nonlinear. In this paper, the state space equations for this system were derived and then linearized using small angle approximations. Using a PID controller with the Kalman and Unscented Kalman filters, the system was simulated in Matlab and then programmed into an inverted pendulum robot built for experimentation. The performance of these filters and controllers were then compared. The results found that for both the simulations and the actual implementation, the Kalman filter produced better state estimates and allowed the system to resist interference and further improve system stability.**

I. Introduction

In this paper, the simulation and implementation of a PID controller and Kalman Filter in an inverted pendulum robot was performed. The purpose of this experiment was to control the inverted pendulum robot with the Kalman Filter and prove the Kalman Filter improves the system response of the robot. The system was first simulated using a PID controller, a Kalman Filter, and an Unscented Kalman Filter using Matlab. The mechanical system was first modeled using the Lagrange method and then the state space equations were used to simulate the system. To simulate the Kalman Filter, the system and measurement noise had to be found. The measurement noise could be found in the sensor manual while the system noise was approximated using the measurement noise. Next, using an Arduino, the Kalman filter was used to balance the movements of an Inverted Pendulum Robot. The sensors used to record the angle and speed of the robot were an ADXL345 accelerometer and a ITG 3200 gyroscope. The first sensor measured the acceleration of the system while the second measured the angular displacement and angular velocity. Using these sensors, the angular displacement of the robot was measured and controlled based on a setpoint of 0° from the vertical axis.

II. Literature Review

A. Inverted Pendulum Robot and PID Controller

An inverted pendulum is a classical controls problem that involves a nonlinear, unstable system with one input signal and several output signals. As such, PID controllers are often used to control inverted pendulum robots because they optimally model Single-Input-Single-Output (SISO) systems. In cases where there are multiple inputs to control, a Multiple-Input-Multiple-Output system is implemented because one PID-controller is not enough [1]. PID control is commonly implemented on inverted pendulum systems and has been found to be simple, effective, and robust according to the works of Cole et. al [2], Sondhia et. al [1], and Wang et. Al [3]. The experimental set up used in this paper was based on the work done by Cole et. al [2], the team that built the robot implemented in this experiment and provided a summary on the construction and programming of the segway-bot using PID control. In conjunction with this, the tutorial provided by Arduino that provides an example of simple programming for PID control was used to refine the system.

B. Inverted Pendulum Robot and Kalman Filter

Several articles on the application of the Kalman filter (KF), extended Kalman filter (EKF), and unscented Kalman filter (UKF) to the control of an inverted pendulum were reviewed to understand the effect of filters on the system. The KF has been shown to improve control of an inverted pendulum through the reduction of noise and improvement of the robustness of the system [4][5]. Further, the filter tends to be applied to the angle measurements of the system rather than the position error [5]. The EKF also improves control and stability of the system, evidenced by its ability to achieve results quickly and accurately [6]. Unlike the KF, the EKF can be used to estimate states in a non-linear system [8]. During the prediction stage, the EKF uses the Jacobian of the non-linear state equations to calculate the a priori state estimates. [6]. However, linearizing the system can result in lost information, which is why the UKF is often chosen for more complex systems. By using only the non-linear equations, the UKF is able to improve final estimation results for inverted pendulums, making it an optimal filter for the inherently non-linear system [7]. A significant amount of other literature has studied estimation theory combined with control theory, with applications to mechatronic systems [9-17].

III. Mechanical Model

In order to model the system, the state space model of the system was derived. This was accomplished by modeling the mechanical system using the Lagrange method. The parameters of the system are described in Table I. A free body diagram of a wheel and the body of the robot can be seen Figure 1. Using the free body diagram, the discrete state space equations were derived. The experimental setup used for this

TABLE I
LIST OF NOMENCLATURE

Symbol	Meaning
θ	Angle of rod with respect to vertical
φ	Angle of wheel with respect to vertical
z	Distance to center of gravity of rod
m_w	Mass of wheel
m_b	Mass of rod (body of robot)
r	Radius of wheel
g	Gravity
w	Thickness of robot body
J_1	Moment of inertia of the wheels
J_2	Coupled moment of inertia
J_3	Moment of inertia of the body of the robot
τ	Torque or input of the motor

experiment was built in-house, and is shown in Figure 2 to illustrate further the freed body diagram.

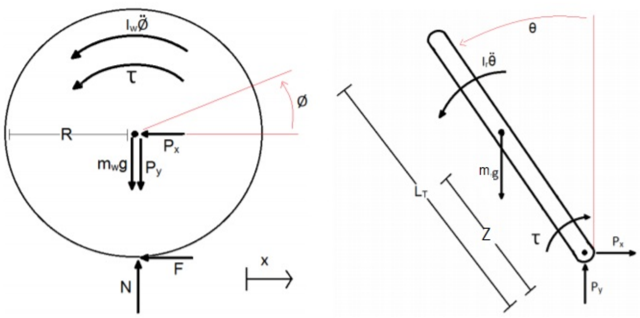


Fig. 1. Free Body Diagram of System

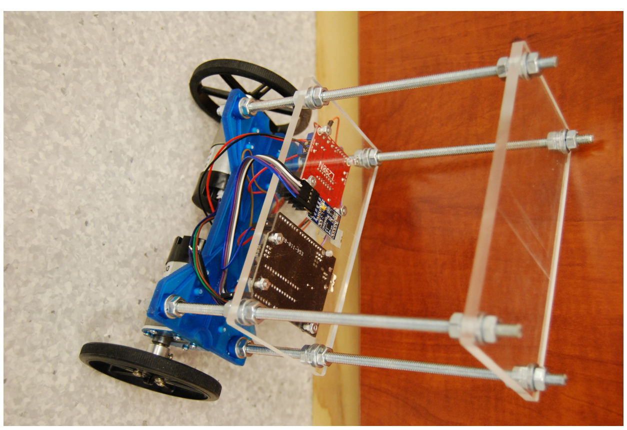


Fig. 2. Experimental Setup Used to Simulate an Inverted Pendulum

$$\frac{d}{dt}\begin{bmatrix}\varphi\\ \theta\\ \dot{\varphi}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}1 & 0 & T & 0\\ 0 & 1 & 0 & T\\ 0 & \frac{-J_2 g m_b z}{J_1 J_3 - J_2^2}T & 1 & 0\\ 0 & \frac{J_1 g m_b z}{J_1 J_3 - J_2^2}T & 0 & 1\end{bmatrix}\begin{bmatrix}\varphi\\ \theta\\ \dot{\varphi}\\ \dot{\theta}\end{bmatrix} + \begin{bmatrix}0\\ 0\\ \frac{J_3 + J_2}{J_1 J_3 - J_2^2}T\\ \frac{J_1 - J_2}{J_1 J_3 - J_2^2}T\end{bmatrix}\begin{bmatrix}0\\ 0\\ \tau\\ \tau\end{bmatrix}$$

$$C_s = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\end{bmatrix}$$

In the above state space representation, J_1, J_2, and J_3 are represented by the following equations:

$$J_1 = (m_b + m_w)r^2 + I_w$$
$$J_2 = m_b r z$$
$$J_3 = m_b z^2 + I_b$$

where:

$$I_w = m_w r^2$$
$$I_b = \frac{1}{12} m_b (2 \times z + 2 \times w)^2$$

IV. SYSTEM SIMULATION

A. PID Controller

Following derivation of the state space equations, the system was simulated with a PID controller to determine if the state space equations accurately modeled the inverted pendulum system. The values found in Table II were used in the simulation and represent actual measurements taken from the inverted pendulum robot. However, it is important to note that this only compares the linearization of the model but not its accuracy. To evaluate the model properly, the simulated results should be compared with the experimental results. However, in this paper, the simulation was used to tune a PID controller initially before implementation on the experimental setup. The system

TABLE II
LIST OF PARAMETERS

Parameter	Value	Description
T	0.01	Period of system, in seconds
m_w	0.1	Mass of wheel, in kilograms
m_b	1.82	Mass of rod (body of robot), in kilograms
r	0.045	Radius of wheel, in meters
g	9.81	Gravity, in meters per seconds squared
w	0.09	Thickness of robot body, in meters
z	0.08	Distance to center of gravity of rod, in meters
τ	1	Torque or input of the motor, in newtons
t	0:T:5	Arbitrary time interval, in seconds
x	$\begin{bmatrix}0 & 0.1 & 0 & 0\end{bmatrix}'$	Initialization of x

was initially calibrated and tuned without the incorporation of noise in order to clearly model ideal conditions. Both the linear and nonlinear state space equations were used in the model in order to ensure that they approximated the same values. Initially, the model displayed a typical dynamic response, oscillating around the setpoint ($\theta = 0°$) until it eventually achieves steady state after several oscillations. The gains were tuned to minimize steady state error (Table III), resulting in an ideal inverted pendulum response where the angle of the body decreases until a steady state value of $\theta \approx 0°$ is achieved. The value of K_i is equivalent to zero in this case because the alterations to K_p and K_d already minimized the steady state error, meaning K_i was not needed.This position would be maintained for the duration of the simulation as no noise or other destabilizing occurrences would be present (see Figure 3a).

Measurement and system noise were then added to the system to create a more realistic simulation (Figure 3b). These noise values were assumed to be Gaussian and white. Measurement noise stemmed from the errors produced through the position tracking provided by the gyroscope and accelerometer sensors implemented on the inverted pendulum robot. The values for measurement noise were found within the specification manuals for the sensors used for the inverted pendulum robot. System noise stemmed from any possible interference that could affect the system measurements. This includes things such as friction of the table and wheels, loose bolts in the apparatus, or unwanted vibrations in the system. These values were approximated since it is very difficult to completely understand the total amount of system noise the robot experiences. The measurement noise values were used to approximate the system noise values. This experiment assumed most of the noise would stem from the sensors, thus the system noise was approximated to several factors less than the measurement noise. The covariance matrices for system (Q) and measurement (R) noise can be found in Table IV.

TABLE III
GAIN VALUES

Gain	Value
K_p	1.1
K_i	0
K_d	0.2

TABLE IV
COVARIANCE MATRICES

System	Measurement
$Q = \begin{bmatrix} 6\times10^{-7} & 0 & 0 & 0 \\ 0 & 6\times10^{-7} & 0 & 0 \\ 0 & 0 & 6\times10^{-7} & 0 \\ 0 & 0 & 0 & 6\times10^{-7} \end{bmatrix}$	$R = \begin{bmatrix} 6\times10^{-4} & 0 & 0 & 0 \\ 0 & 6\times10^{-4} & 0 & 0 \\ 0 & 0 & 6\times10^{-4} & 0 \\ 0 & 0 & 0 & 6\times10^{-4} \end{bmatrix}$

B. Kalman Filters applied to PID Controller

In order to linearize the system model, small angle approximations were made. Both the linear and non-linear system models were simulated and the results were recorded over a ten second interval to confirm that both models behaved similarly for small angles. Thus, the linearized model was assumed to be the true model and was used to calculate the *a priori* state estimates and *a priori* state covariance in the Kalman Filter. Using the non-linear equations in the EKF would not significantly improve the accuracy of the state estimates for small angles. Since the physical system is only expected to operate within small angles, the EKF was not implemented. If this assumption was not made, then the EKF would have to be applied in place of the KF since when large angles are encountered, the system would behave nonlinearly. This issue did not apply to the UKF, which uses the nonlinear equations without linearizing them. The discretized, nonlinear state space equations can be found below:

$$\frac{d}{dt}\begin{bmatrix} \varphi \\ \theta \\ \dot{\varphi} \\ \dot{\theta} \end{bmatrix} =$$

$$\begin{bmatrix} 1 & 0 & \varphi T & 0 \\ 0 & 1 & 0 & \theta T \\ 0 & \frac{J_2 \sin(\theta)\cos\theta g m z + J_2 \dot{\theta}^2}{(J_2\cos(\theta))^2 - J_1 J_3} T & 1 & \frac{J_2 J_3 \sin(\theta)\dot{\theta}^2}{(J_2\cos(\theta))^2 - J_1 J_3} T \\ 0 & \frac{\sin(\theta) J_1 m_b z + (1 + J_2^2 g \cos(\theta)\sin(\theta)\dot{\theta}^2}{(J_2\cos(\theta))^2 - J_1 J_3} T & 0 & \frac{1 + J_2^2 g \sin(\theta)\cos(\theta)\dot{\theta}^2}{(J_2\cos(\theta))^2 - J_1 J_3} T + 1 \end{bmatrix} \begin{bmatrix} \varphi \\ \theta \\ \dot{\varphi} \\ \dot{\theta} \end{bmatrix}$$

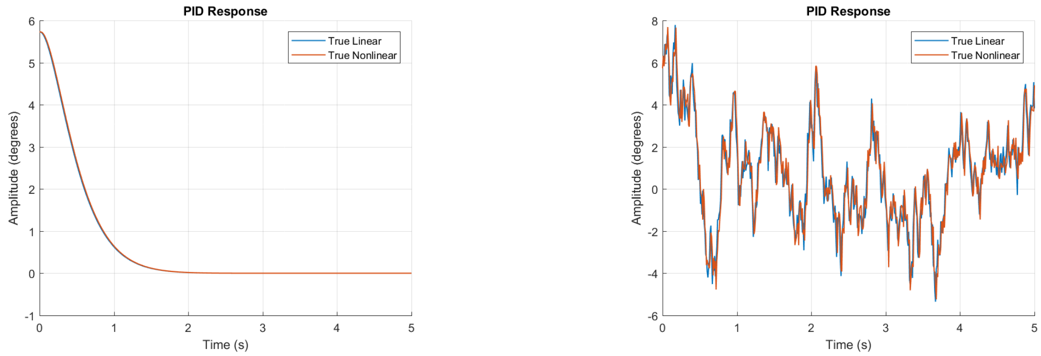
$$+ \begin{bmatrix} 0 \\ 0 \\ \frac{J_3 + J_2\cos(\theta)}{J_1 J_3 - J_2^2} \\ \frac{-J_1 - J_2\cos(\theta)}{J_1 J_3 - J_2^2} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \tau \\ \tau \end{bmatrix}$$

$$C_s = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Figure 4 displays the filters applied to the system and Table V displays the root-mean squared errors of the filters. Application of the KF revealed that the filter was able to accurately track the system, never deviating very much from the true approximations. The UKF varied in response when applied. While on some trials it would appear to track the system perfectly, on others its values would explode, resulting in incorrect approximations. From the reviewed theory, the UKF should improve the system approximation. This suggests that there is an issue in the code used to approximate with the UKF. However, given the linear approximation made on the system, it is reasonable that the KF would be the best estimator since it is optimal for linear systems. Since the implementation on the real system involves improved body angle control by applying the KF, a final simulation comparing the measurement data was also conducted. This can be seen in Figure 5. The efficacy of the Kalman filter was measured by tracking body body angle error of the PID controller where the setpoint was 0°. In one scenario, the sensor measurement was used to calculate the body angle error of the PID controller. The other scenario used the Kalman estimate to calculate the the body angle error. The RMSE value of the PID error decreases, as seen in Table VI, illustrating that applying the KF to the inverted pendulum system should produce a better response. The UKF's response when applied in both cases was unexpected. Because it is unstable, it makes sense that it is unable to approximate the measurements in Fig. 5a. When

TABLE V
RMSE VALUES

State	KF (rad^2)	UKF (rad^2)
θ	2.44×10^{-3}	2.53
φ	5.22×10^{-4}	1.55×10^{-3}
$\dot{\theta}$	1.65×10^{-3}	3.94×10^{-1}
$\dot{\varphi}$	1.86×10^{-2}	2.58×10^{-2}



(a) (b)

Fig. 3. Initial Simulation Results: (a) Ideal PID Response (without noise), (b) Ideal PID Response (with noise).

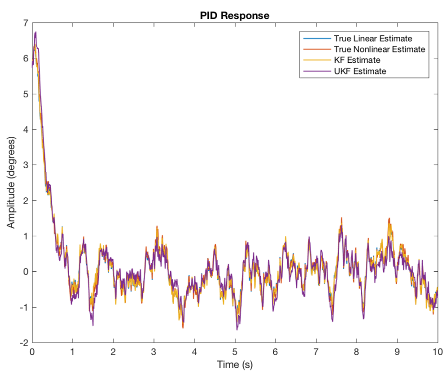


Fig. 4. System with Filters Applied

the KF was used in the PID controller (Fig. 5b), the UKF was able to produce more accurate estimates. However, it is currently unknown why it has this effect.

TABLE VI
RMSE VALUES FOR MEASUREMENT COMPARISON

Without Kalman Error	$4.78 \times 10^{-2\circ}$
With Kalman Error	$3.66 \times 10^{-2\circ}$

V. SYSTEM IMPLEMENTATION

A. System Model Accuracy vs. Kalman Filter Efficacy

In order to tune and implement the KF into the inverted pendulum robot, the value of variables in the initial state space equations were changed. Eq. 1 represents the system matrix constant (F) which multiplied the angular position of the body of the robot. From the initial measurements in Table II, this factor was equal to 75.6. Eq. 2 represents the input matrix constant (B) and has a value of 0.0015. Initial simulation of the system model with these values was problematic, resulting in an inaccurate system model that poorly balanced. After implementing this model in Arduino, it was found that the errors occurring in the system were the result of using incorrect measurement values.

$$F = \frac{J_1 g m_b z}{J_1 J_3 - J_2^2} \tag{1}$$

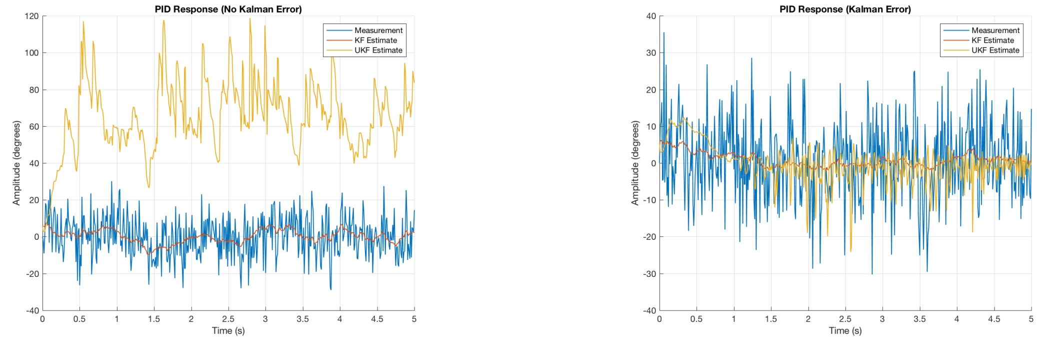
$$B = \frac{J_1 - J_2}{J_1 J_3 - J_2^2} \tag{2}$$

Figure 6a uses the incorrect system model. When compared with the true states, the incorrect system model produces estimates with high error. Aforementioned, the initial values in Table II ranged from being measured and approximated. This caused some inaccuracies in the model constants F and B that adversely affected the system. The inaccurate estimates caused the PID controller to produce the wrong torque needed to control the system, resulting in an unstable system that does not balance. The instability of the system indicates the importance of using proper parameter measurement values in the system model. The inertia of the pendulum system as well as the effect of torque on that system are vital to its control. Thus, accurate models are necessary to ensure the system is stable. The system model variables were then adjusted through trial and error until the correct system response was achieved. Figure 6b illustrates the system response with the adjusted system model. Using values which accurately model the system produces more accurate body angle errors for the PID controller.

B. Kalman Filter Process

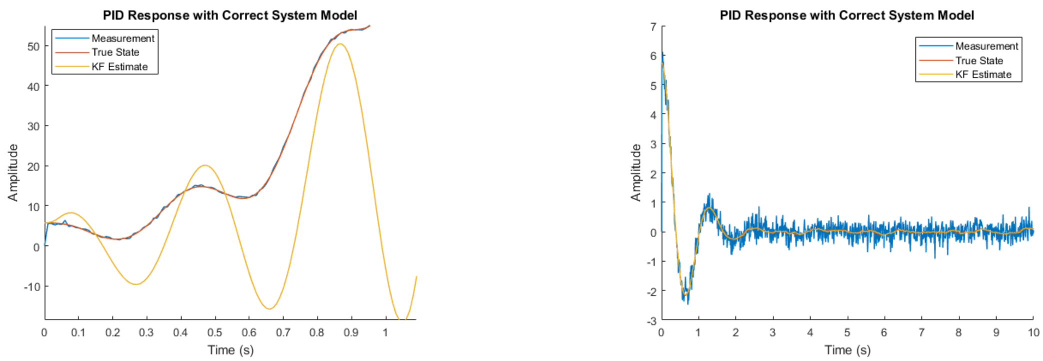
As described, the KF is a predictor-corrector estimation strategy that yields optimal estimates to linear systems and measurements. A nonlinear form of the KF is known as the extended KF, and essentially first-order Taylor series approximations are used to create Jacobians or linear matrices. The process is similar between the KF and EKF. The KF process is described as follows. The first two equations below represent the prediction stage and predict the state estimates and covariances, respectively.

$$\hat{x}_{k+1|k} = F\hat{x}_{k|k} + Bu_k \tag{3}$$



(a) (b)

Fig. 5. Simulation Comparison to Measurement Values: (a) without Kalman Error; (b) with Kalman Error.



(a) (b)

Fig. 6. Simulated System Models: (a) with Poor System Measurements; (b) with Adjusted System Measurements.

$$P_{k+1|k} = FP_{k|k}F^T + Q_k \tag{4}$$

The next section is referred to as the update stage. The following equations are used iteratively with the prediction stage to calculate the updated state estimates and covariances, respectively. The first equation is used to calculate an innovation covariance used to generate the Kalman gain. The H term refers to the measurement matrix. Note that the upated state estimates are used by the PID controller to generate the control signal based on the error (difference between the desired and estimated states).

$$S_{k+1} = HP_{k+1|k}H^T + R_k \tag{5}$$

$$K_{k+1} = P_{k+1|k}H^T S_{k+1}^{-1} \tag{6}$$

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1}(z_{k+1} - H\hat{x}_{k+1|k}) \tag{7}$$

$$P_{k+1|k+1} = (I - K_{k+1}H)P_{k+1|k} \tag{8}$$

C. Kalman Filter System Tuning

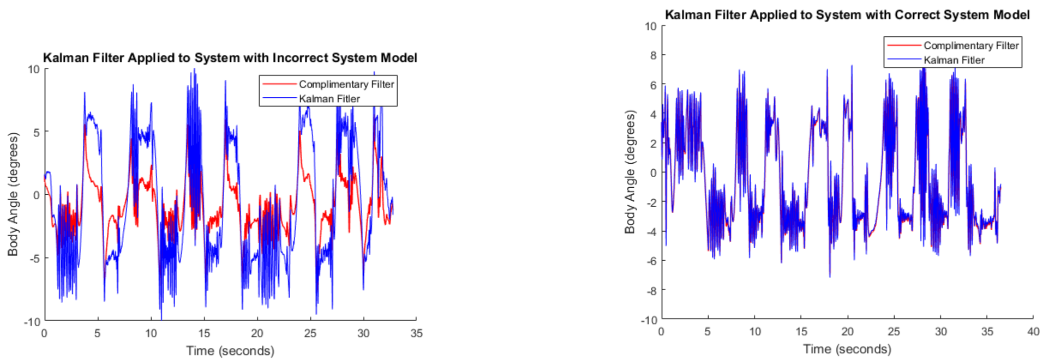
The Kalman filter was initially tuned in the Arduino assuming a zero measurement covariance. The system constants were tuned until the Kalman estimates matched the complementary filter estimates. Initially, F was assumed to be 75.6 based on measurements of the physical system and B was assumed to be 0.015 based on the simulation. These system constants overestimated the effect of the torque on the body angle. Thus, the Kalman estimates produced much higher angles than the complementary filter as seen in Figure 7a. F was tuned for the KF to a value of 61 while B was tuned to a value of 0.001. After the model was corrected, the KF estimates matched the complementary filter estimates (Figure 7b).

D. Covariance Matrix

After tuning the system matrices, the covariance matrices were tuned to further improve the estimates. The measurement covariance matrix is not constant for all angles and thus needed to be calculated for different angles. The calculated values were taken in 5° increments and can be found in Table VII. At small angles ($<10°$), the covariance values were relatively similar in magnitude. At angles $\geq 10°$, however, the magnitude increases substantially.

E. PID Tuning

The PID controller was retuned for the inverted pendulum robot. The gains for the implementation were tuned in the same order as in the Matlab simulations: The proportional gain was tuned first while setting all other gains to zero.



(a) (b)

Fig. 7. Implemented System Models: (a) with Poor System Measurements; (b) with Adjusted System Measurements.

TABLE VII
COVARIANCE VALUES

Degrees	Covariance
0	$R = 4.7801 \times 10^{-4}$
5	$R = 5.7013 \times 10^{-4}$
10	$R = 1.1378 \times 10^{-3}$

TABLE VIII
INITIAL AND TUNED PARAMETERS

Initial Parameters	**Tuned Parameters**
$Q = \begin{bmatrix} 6 \times 10^{-7} & 0 \\ 0 & 6 \times 10^{-7} \end{bmatrix}$	$Q = \begin{bmatrix} 2 \times 10^{-7} & 0 \\ 0 & 2 \times 10^{-7} \end{bmatrix}$
$R = 6 \times 10^{-4}$	$R = 2 \times 10^{-5}$
System Constant = 75.6	System Constant = 61
$B = 0.015$	$B = 0.001$
K_p = 70	K_p = 12
K_d = 0.25	K_d = 0.1

Then, the derivative gain was slowly adjusted to smooth the oscillations. Finally, integral gains were omitted because the system is unstable and never expected to settle completely with 0 steady state error. The PID was tuned initially on the complementary filter, which combines sensor data from both the accelerometer and gyroscope. Listing 1 displays the system update equation can be seen for the complementary filter. Using both the sensor data from the accelerometer and gyroscope, a full picture of how the system reacted while a controller or filter was maintaining its upright position can be seen. The KF was then applied and also had to be tuned until the parameter values in Table VIII were achieved. These parameters created a stable system. The variables' greatest change in magnitude came from tuning the system noise and the value for the input matrix constant.

Listing 1. Code Equation for Complementary Filter

```
C1= 0.98
C2 = 0.02
Complementary(k)= C1*(Complementary(k-1)
+ gyroscope(k)*T) + C2*(accelerometer(k));
```

VI. EXPERIMENTAL RESULTS

The PID controller was implemented using the complimentary filter estimates as input followed by using the Kalman filter estimates as the input (represented in Figures 8a and 8b, respectfully). The initial body angle offset was set to approximately 4° with a setpoint of 0° which represents a perfectly upright robot. The PID controller was allowed to run until the inverted pendulum robot reached steady state. The RMSE values of both trials were calculated and are displayed in Table IX.

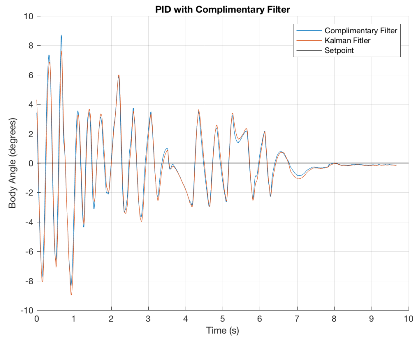
TABLE IX
PID CONTROLLER RMSE VALUES

PID Controller Input	RMSE of Error from Setpoint
Complimentary Filter Estimate	68.4210°
Kalman FIlter Estimate	65.0810°

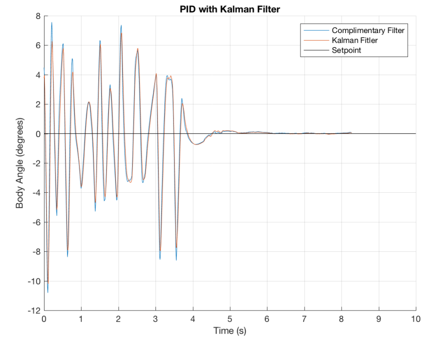
VII. DISCUSSION

A. Experimental Analysis

The RMSE of the error from the setpoint was 68.412° and 65.081° for the complimentary filter and Kalman filter respectively. In addition, the complimentary filter estimates produced a steady error of approximately 0.15° while the Kalman filter had virtually no steady state error. Finally, the settling time using the complimentary filter was approximately 7.8 seconds while the settling time using the Kalman filter was approximately 4.5 seconds. The Kalman filter appears to offer an improvement, over the complimentary filter, as predicted in the simulated results. However, the results are very comparable when considering that there was high variability in the experimental setup. Some of the main contributors of variability were the power and USB cables. The cables produced an uncertain amount of positive torque and damping when they dragged behind the robot. The Kalman filter estimates were



(a)



(b)

Fig. 8. PID Controller Results: (a) Using Complimentary Filter Estimate as Input; (b) Using Kalman Filter Estimate as Input.

relatively consistent with the complimentary filter estimates and offered only marginal improvement when used in the PID controller as shown in Fig. 8. Thus it is reasonable to expect that the Kalman filter would perform comparably with the complementary filter. While the Kalman filter is able to smooth out some of the noise, it appears to lag slightly behind the complimentary filter during rapid oscillations. Thus, when the angular velocity is positive, the Kalman estimates tend to be lower than the complimentary filter estimates. The opposite is true for negative angular velocities. The PID response shows several oscillations before finally being able to settle close to the setpoint. The magnitude of the oscillations do not monotonically decreases as expected in a second order system model. There are several contributing factors for the additional oscillations. The motor driver was not consistent in its output and often failed to output the correct torque based on the setpoint error. In addition, there was significant backlash in the motor-wheel system which was not modeled into the controller or system matrix of the Kalman filter. The backlash would often cause several degrees of movement before torque was actually applied to the system.

B. System Problems

There were several physical limitations with the system that greatly impacted the results. One issue was that the wires connected to the power supply created a great deal of interference with the system. In ideal situations, the controller or filter should be the only stabilizing factor on the robot. In practice, however, as the robot travelled away from the power supply, the connecting wires would act like a spring and become tense before pulling the robot backwards, causing a reaction force on the robot that artificially controlled its movements. Another issue stemmed from the motors. There were discrepancies with the performance between the individual motors. The motor response of the left motor was lacking compared to the right motor. The left motor was occasionally unresponsive which made the robot travel in circles instead of in a forward and backward motion. The unresponsiveness could be due to the amount of current entering the right motor compared to the left motor. Another problem with controlling the robot was that the motor could not output small values of torque to correct small body angle errors. This is because the DC motors required a certain threshold of voltage in order to operate. For small angle errors, the PID controller would output a small voltage to correct the body angle. However, the motors outputted zero torque because the voltage was below the required threshold. Thus, the motors would not activate until a certain body angle error was achieved. One final issue with the robot was its sensitivity to disturbances. This was due to the robot's physical design which caused it to have a small moment of inertia and a low center of gravity. To improve the controllability of the robot, more weight should be added higher up on the body of the robot.

C. Future Work

Several additions can be made to the inverted pendulum robot to improve performance. One addition would involved incorporating more controllers, such as for the position or the yaw of the system. A linear-quadratic regulator controller could also be added in place of a PID controller to alter system performance. Additionally, the estimation techniques need to be refined as well. The UKF needs to be corrected in order to correctly approximate the system. Fixing the UKF will allow for incorporation of other estimation techniques, such as the interacting multiple model, that could produce more refined results. Finally, several physical aspects of the system should be adjusted based on the limitations mentioned in the previous section. This would include making the robot powered by a battery to allow the wires to be removed and redesigning the system to increase its mass and its the moment of inertia

VIII. Conclusions

Experimentation showed that implementation of a Kalman filter with a tuned PID controller improved the response of an inverted pendulum robot. The simulated and implemented results also further demonstrated that an accurate system model is important to good overall system performance. If any estimates are inaccurate, the system model will not correctly correspond with the actual system, resulting in inaccuracies and possibly instabilities. Additional experimentation should examine effects of different controllers and estimators in the system. In future experiments, several modifications can be made to the inverted pendulum robot in order to improve performance such as removing the wires from the system, correcting the unresponsive motor, and increasing the inertia of the system.

References

[1] S. Sondhia, R. R. Pillai, S. S. Hedge, S. Chakole and V. Vora, *"Development of Self Balancing Robot with PID Control,"* International Journal of Robotics, vol. 7, no. 1, pp. 1-6, 2017.

[2] N. Cole, A. Cataford, S. Hockin, A. Gaiero and S. Rossi, *"Self Balancing Bot,"* ENGG*4480 W17, pp. 1-28, 2017.

[3] J. Wang, *Simulation studies of inverted pendulum based on PID controllers,"* Simulation Modeling Practice and Theory, vol. 19, no. 1, pp. 440-449, 2011.

[4] W. Zhang, G. Li and W. Lirong, *"Research on the Control Method of Inverted Pendulum Based on Kalman Filter,"* in 2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing, Dalian, 2014.

[5] H. Lee and S. Jung, *"Balancing and navigation control of a mobile inverted pendulum robot using sensor fusion of low cost sensors,"* Mechatronics, vol. 22, no. 1, pp. 95-105, 2012.

[6] S. Hoseini and J. Poshtan, *"Fault tolerant control applied on an inverted pendulum by using extended Kalman filter,"* in 2007 International Conference on Intelligent and Advanced Systems, Kuala Lumpur 2007.

[7] M. Zheng, K. Ikeda and T. Shimomura, *"Parameter Estimation of Rotary Inverted Pendulum based on Unscented Kalman Filter,"* in 2006 SICE-ICASE International Joint Conference, Tokushima, 2006.

[8] S. Bonadies, N. Smith, N. Niewoehner, A. S. Lee, A. M. Lefcourt, and S. A. Gadsden, *"Development of Proportional-Integral-Derivative and Fuzzy Control Strategies for Navigation in Agricultural Environments,"* ASME Journal of Dynamic Systems, Measurement, and Control, Vol. 140, No. 6, 2017.

[9] J. Kim, K. Chang, B. Schwarz, A. S. Lee, M. Al-Shabi, and S. A. Gadsden, *"Dynamic Model and Motion Control of a Robotic Manipulator,"* Journal of Robotics, Networking and Artificial Life, Vol. 4, No. 2, pages 138-141, 2017.

[10] S. A. Gadsden, S. R. Habibi, and T. Kirubarajan, *"Kalman and Smooth Variable Structure Filters for Robust Estimation,"* IEEE Transactions on Aerospace and Electronic Systems, Vol. 50, No. 2, pages 1038-1050, 2014.

[11] S. A. Gadsden and A. S. Lee, *"Advances of the Smooth Variable Structure Filter: Square-Root and Two-Pass Formulations,"* Journal of Applied Remote Sensing, Vol. 11, No. 1, 2017.

[12] J. Goodman, J. Kim, A. S. Lee, S. A. Gadsden, and M. Al-Shabi, *"A Variable Structure-Based Estimation Strategy Applied to an RRR Robot System,"* Journal of Robotics, Networking and Artificial Life, Vol. 4, No. 2, pages 142-145, 2017.

[13] H. H. Afshari, S. A. Gadsden, and S. R. Habibi, *"Gaussian Filters for Parameter and State Estimation: A General Review and Recent Trends,"* Signal Processing, Vol. 135, pages 218-238, 2017.

[14] A. Cataford, S. A. Gadsden, K. Turpie, and M. Biglarbegian, *"AirLUSI: Estimation, Filtering, and PID Tracking Simulation,"* 2018 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), Quebec City, Quebec, 2018.

[15] B. Xu, Y. Lyu, and S. A. Gadsden, *"Estimation and Control of a Double-Inverted Pendulum,"* 2018 Canadian Society for Mechanical Engineering (CSME) International Congress, Toronto, Ontario, 2018.

[16] S. A. Gadsden, *"An Adaptive PID Controller Based on Bayesian Theory,"* 2017 ASME Dynamic Systems and Control Conference, Tysons Corner, Virginia, 2017.

[17] M. Al-Shabi, A. Cataford, and S. A. Gadsden, *"Quadrature Kalman Filters with Applications to Robotic Manipulators,"* 2017 IEEE International Symposium on Robotics and Intelligent Sensors, Ottawa, Ontario, 2017.