MAE280A Report: Controller Design for Mobile Inverted Pendulum

Guoren Zhong
gzhong@eng.ucsd.edu

***Abstract*—In this work, we study the state feed back control problem of the mobile inverted pendulum (MIP) based on Zhu Zhuo's Master Thesis [1]. Intrinsically, the MIP system is nonlinear and unstable, but it can be linearized and model by a linear state space system with state variables θ, $\dot{\theta}$, and $\dot{\phi}$. Then, in order to stabilize the system, we introduce a state feed back controller. Due to this motivation, we design three controllers for MIP, and test their performance and robustness. Besides, we push the controllers to their limits and test their largest affordable initial angle θ of the MIP. Last but not least, we try to further improve the performance by altering the other initial conditions.**

***Index Terms*—Mobile inverted pendulum, equilibrium, control, stabilization.**

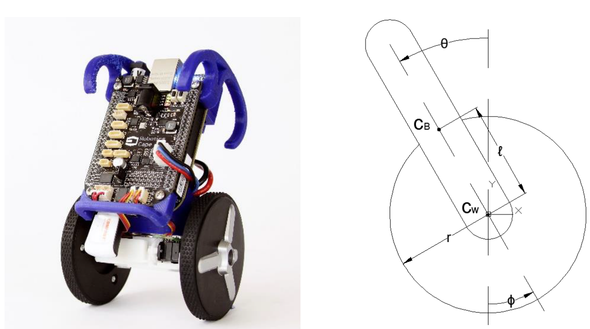
I. Introduction

MOBILE ROBOTS have drawn increasing research attention in the past few years, among which the two-wheeled mobile inverted pendulum (MIP) is one of the simplest. Despite this, MIP is still naturally nonlinear and unstable. Therefore, a controller is needed to keep the robot stable. In this work, we refer to the Master Thesis of Zhu Zhuo [1], then design our own controllers and test their performances and robustness to noises.

II. System modeling

In this section, we give a brief description to the reduced linearized used in this work.

A. MIP modeling and equilibriums



(a) (b)

Fig. 1: These are what the MIP looks like briefly. (a) is one of the simplest MIP model developed at the UCSD Coordinated Robotics Lab. (b) is the simplified MIP model that we can use for later discusses in this work.

According to Zhu, the system can be modeled by six parameters a, b, c, d, e, f for simplicity instead of a bunch of physical parameters. A simplified model is shown in Fig.1, where the state of MIP is described by θ, the body's angle, and ϕ, the wheel's angle, and their first-order derivatives. Besides, the voltage input is denoted as $V(t)$. Then, two equilibrium points are:

$$\begin{aligned}\text{Equilibrium 1: } &\theta(t)=0,\ \dot{\theta}(t)=0,\ \dot{\phi}(t)=0,\ V(t)=0.\\ \text{Equilibrium 2: } &\theta(t)=\pi,\ \dot{\theta}(t)=0,\ \dot{\phi}(t)=0,\ V(t)=0.\end{aligned} \tag{1}$$

Since Equilibrium 2 is stable and can be achieved without control, we would lay our emphasis on the unstable Equilibrium 1 in the following discussions. Intrinsically, the system is nonlinear and the state equation are shown in Eqn.3b:

$$\begin{bmatrix}\ddot{\theta}(t)\\ \ddot{\phi}(t)\\ \dot{\theta}(t)\\ \dot{\phi}(t)\end{bmatrix} = \begin{bmatrix}M_1\\ M_2\\ M_3\\ M_4\\ \dot{\theta}(t)\\ \dot{\phi}(t)\end{bmatrix}, \tag{2}$$

where

$$\begin{aligned}M_1 =& -\frac{1}{ac-b^2\cos\theta(t)^2}(-ad\cdot\sin\theta(t)+ae\cdot V(t)\\ &+be\cdot\cos\theta(t)\cdot V(t)+aj\dot{\theta}(t),\\ M_2 =&\ bj\cdot\cos\theta(t)\cdot\dot{\theta}(t)+b^2\cos\theta(t)\cdot\sin\theta(t)\cdot\dot{\theta}(t)^2\\ &-aj\cdot\dot{\phi}(t)-bj\cdot\cos\theta(t)\cdot\dot{\phi}(t),\\ M_3 =& -\frac{\sec\theta(t)}{b^2-ac\cdot\sec\theta(t)^2}(-bd\cdot\sin\theta(t)+be\cdot V(t)\\ &+ce\cdot\sec\theta(t)\cdot V(t),\\ M_4 =&\ bj\dot{\theta}(t)+cj\cdot\sec\theta(t)\cdot\dot{\theta}(t)+bac\cdot\tan\theta(t)\cdot\dot{\theta}(t)^2\\ &-bj\cdot\dot{\phi}(t)-cj\cdot\sec\theta(t)\cdot\dot{\phi}(t),\end{aligned}$$

B. Linearization

As we can see from above, the nonlinear system is quite complicated. To simplify the problem, we applied linearization techniques to the system, and model the linearized system using Eqn.3:

$$\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\end{bmatrix} = \begin{bmatrix}\dot{\theta}\\ \dot{\phi}\\ \theta\\ \phi\end{bmatrix},\quad \begin{bmatrix}y_1\\ y_2\end{bmatrix} = \begin{bmatrix}\dot{\theta}\\ \dot{\phi}\end{bmatrix},\quad u_1 = V \tag{3}$$

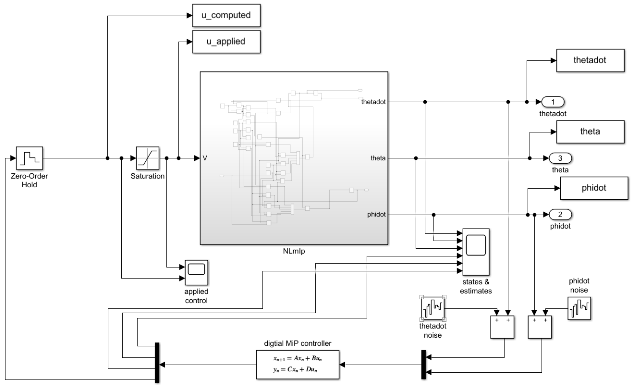


Fig. 2: The MIP model built in Simulink.

After linearizing the model with respect to Equilibrium 1 in Eqn.1, we can further simplify the system by reducing $x_4 = \phi$ from it since it has no impact on any other states. The resulting reduced linear state equation is shown in Eqn.4.

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \end{bmatrix} = \begin{bmatrix} -\frac{(a+b)j}{-b^2+ac} & \frac{(a+b)j}{-b^2+ac} & \frac{ad}{-b^2+ac} \\ -\frac{(b+c)j}{b^2-ac} & \frac{(b+c)j}{b^2-ac} & \frac{bd}{b^2-ac} \\ 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} -\frac{(a+b)e}{-b^2+ac} \\ \frac{(b+c)e}{-b^2+ac} \\ 0 \end{bmatrix} \cdot u \tag{4}$$

Based on the results in Zhu's thesis, the state space model can be specified as below in Eqn.5.

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad \mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}, \tag{5}$$

where

$$\mathbf{A} = \begin{bmatrix} -13.53 & 13.53 & 175.5 \\ 17.74 & -17.71 & -115.6 \\ 1 & 0 & 0 \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} -90.46 \\ 118.6 \\ 0 \end{bmatrix}, \ \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \ \mathbf{D} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

C. *Convert to discrete-time*

Since we would like to consider this problem in discrete-time, we need to apply a conversion to the system in Eqn.5 using MATLAB's "c2d()" command with a sample rate of 100 Hz. The resulting discrete-time state space system is shown in Eqn.6.

$$\mathbf{x}_{n+1} = \mathbf{A_D}\mathbf{x}_n + \mathbf{B_D}\mathbf{u}_n, \quad \mathbf{y}_n = \mathbf{C_D}\mathbf{x}_n + \mathbf{D_D}\mathbf{u}_n, \tag{6}$$

where

$$\mathbf{A_D} = \begin{bmatrix} 0.8917 & 0.1165 & 1.581 \\ 0.1475 & 0.8475 & -0.9253 \\ 0.00942 & 0.00061 & 1.008 \end{bmatrix},$$

$$\mathbf{B_D} = \begin{bmatrix} -0.7791 \\ 1.02 \\ -0.0041 \end{bmatrix}, \ \mathbf{C_D} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \ \mathbf{D_D} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

And the eigenvalues for $\mathbf{A_D}$ is 0.7046, 1.0918, and 0.9509 respectively, which agree with our statement that the system is unstable. Therefore, a controller is needed to stabilize the system.

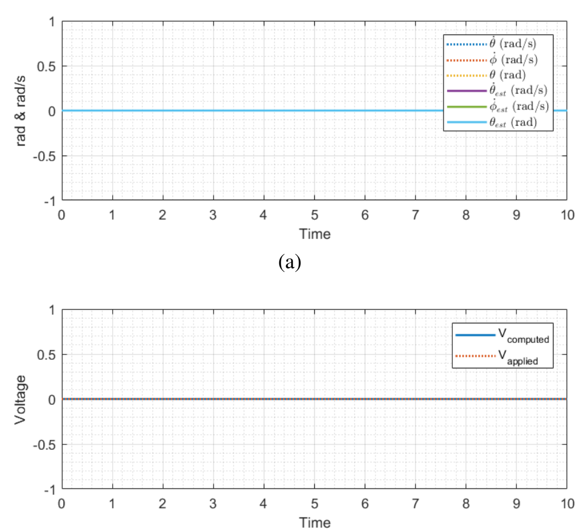
III. State Estimate Feedback Controller

In this section, we show some controller designs and run equilibrium tests on them. Then, we test their performance on stabilization with and without noise.

A. *Controller designs*

First, the general MIP model built in *Simulink* is shown in Fig.2, where, our focus, a controller module can be seen at the bottom. Besides, two noise modules for $\dot{\theta}$ and $\dot{\phi}$ at the bottom right are used to test the robustness against noise. And the states and estimates at each step would be recorded by the scope module right above the noise modules. Additionally, another scope module on the left is responsible to record the voltage input applied to the system, and this voltage is set to be within the range $[-6V, 6V]$ for all designs.

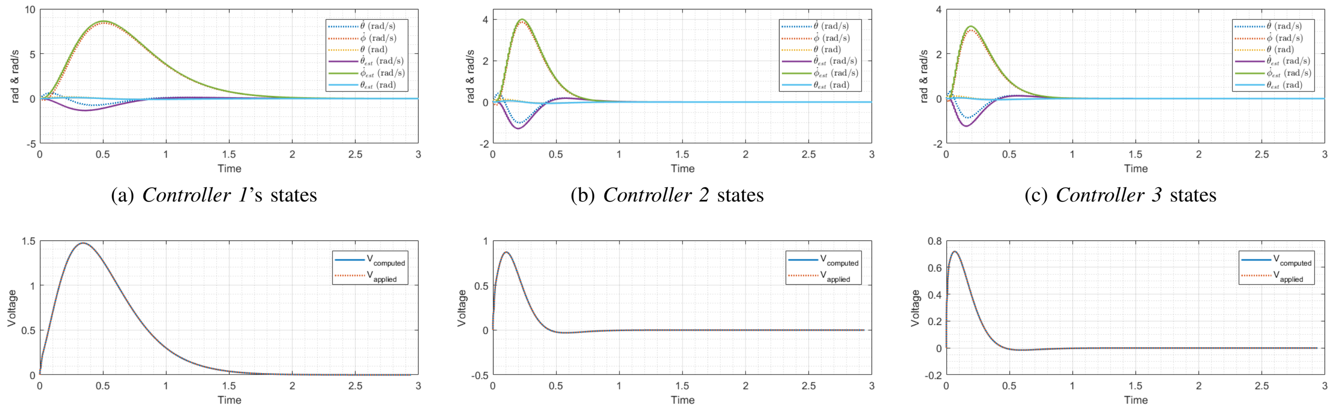
In general, we design state feedback controllers by applying the pole placement technique. Specifically, we can choose $\mathbf{K} \in \mathbb{C}^{1\times 3}$ such that the eigenvalues of $(\mathbf{A_D} - \mathbf{B_D}\mathbf{K})$ are within the stable region ($|\lambda| < 1$). Moreover, since some states cannot be measured directly, an observer is needed to give estimations on the states. Similarly, we choose $\mathbf{L}$ such that $(\mathbf{A_D} - \mathbf{L}\mathbf{C_D})$ has eigenvalues within the stable region. To achieve the above, we use the MATLAB's 'place()' command to find $\mathbf{K}$ and $\mathbf{L}$ that place the poles at desired positions. For convenience, we denote the poles as P_K and P_L in later discussions. Below are the three designs we made, each of which is an attempt to improve from the last one.



(a)

(b)

Fig. 3: These show the results of equilibrium tests. (a) presents the state values and the estimates. (b) presents the voltage computed and voltage applied. These plots in fact work for all three controllers since they have the same plot for equilibrium tests.



(a) *Controller 1*'s states (b) *Controller 2* states (c) *Controller 3* states

(d) *Controller 1*'s voltages (e) *Controller 2*'s voltages (f) *Controller 3*'s voltages

Fig. 4: These are the simulation results in a time span of $t = 3s$ of three controllers under the conditions that $\theta_0 = 0.1rad$ and $t = 3$. (a)(b)(c) plot the states of each controller, and (d)(e)(f) plot the voltage input for each design.

1) Design 1: In the first design, we set the poles as:

$$P_{K1} = P_{L1} = [0.95 \quad 0.94 \quad 0.93]. \tag{7}$$

Thus, the corresponding $\mathbf{K}$ and $\mathbf{L}$ are:

$$\mathbf{K}_1 = \begin{bmatrix} -0.1156 & -0.1731 & -3.3744 \end{bmatrix}, \quad \mathbf{L}_1 = \begin{bmatrix} -0.2059 & -0.2697 \\ 0.2395 & 0.1332 \\ 0.0021 & -0.0168 \end{bmatrix} \tag{8}$$

2) Design 2: In order to make some improvement, we change both poles to:

$$P_{K2} = P_{L2} = [0.9 \quad 0.88 \quad 0.86], \tag{9}$$

which are slightly slower than those in the first design. And the corresponding $\mathbf{K}$ and $\mathbf{L}$ are:

$$\mathbf{K}_2 = \begin{bmatrix} -0.5421 & -0.3382 & -7.3041 \end{bmatrix}, \quad \mathbf{L}_2 = \begin{bmatrix} -0.0502 & -0.2088 \\ 0.1835 & 0.1575 \\ 0.0035 & -0.0268 \end{bmatrix} \tag{10}$$

3) Design 3: In the third design, we keep $\mathbf{L}$ the same as those in *design 2*. However, we try to find a better $\mathbf{K}$ by solving the discrete algebraic Riccati Equation. This can be done in MATLAB though the command "dare()". In our case, the Q and R are set as:

$$Q = [0 \quad 1 \quad 0], \quad R = 0.01, \tag{11}$$

which give us:

$$\mathbf{K}_3 = \begin{bmatrix} -2.4054 & -1.0362 & -24.7557 \end{bmatrix}. \tag{12}$$

B. Equilibrium tests

First, we ran the equilibrium tests on all three designs, and recorded the state values, state estimates, and voltages applied. Specifically, all initial values are set to 0 as well as the noise power. The results are shown in Fig.3. As expected, under the equilibrium condition, all the states, estimates, and voltages stay at 0.

C. Stabilization tests

In this part, we examine the performance of three controllers. For the settings, we now change the initial angle to $\theta_0 = 0.1rad$, and leave all the others unchanged. The simulation results are shown in Fig.4. From the plots, we see that all three controllers work fine under this condition, where the curves converge within the 3-second time span. For *controller 1*, it takes approximately 2 seconds to converge, while it only takes less than 1 second for the other two. Although *controller 2* and *controller 3* have comparable converging time, the latter one has a smaller variation. Besides, from the voltage curves in Fig.4d,4e, and 4f, we also observe a decreasing magnitudes of voltage inputs and they are all within the saturation limits $[-6V, 6V]$. Therefore, we can conclude that all three controllers are capable of stabilizing the MIP system. However, *controller 3* seems to be the best among them.

D. Capability test

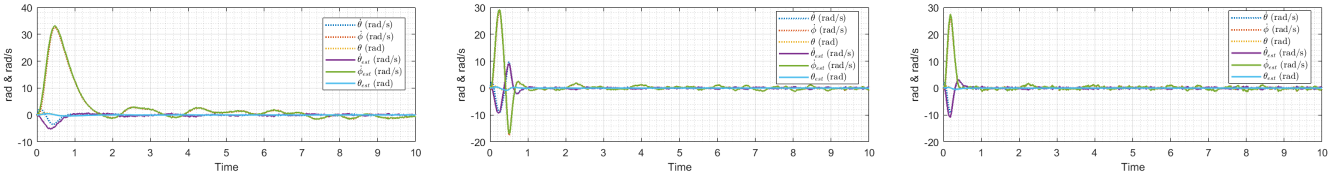
Now, we want to push each controller to limit and find out the largest bias of initial angle θ_0 they can take. Still, we keep all the other settings the same while increase θ_0 gradually. The maximum initial angle for each is shown in Table.I. From the results, we again see that *controller 3* has the best performance since it can take the largest initial angle.

Controller No.	1	2	3
max θ_0 (rad)	0.36	0.57	0.65

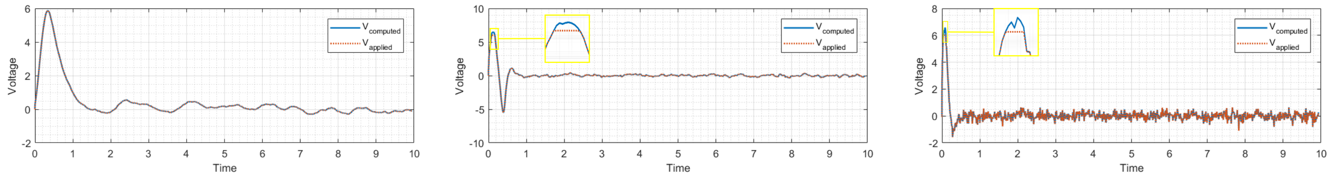
TABLE I: Maximum initial angles θ_0 of each controller.

E. Robustness to noise

After all the tests above, we would like to see how robust the three controllers against noise. First, we set θ_0 to the maximum values of each controller as given in table.I, and increase the measurement noise powers for both $\dot{\theta}$ and $\dot{\phi}$ to



(a) *Controller 1*'s states (b) *Controller 2*'s states (c) *Controller 3*'s states



(d) *Controller 1*'s voltages (e) *Controller 2*'s voltages (f) *Controller 3*'s voltages

Fig. 5: These plots show the performance of each controller at its maximum θ_0 under a noise power of 0.0005 for both $\dot{\theta}$ and $\dot{\phi}$ within a time span of $t = 10s$. (a)(b)(c) plot the states of each controller, and (d)(e)(f) plot the voltage input for each design.

0.0005, then ran simulations for a time span of $t = 10s$. The results are shown in Fig.5, where we can see all the controllers are still working fine, despite some fluctuations due to the noise. Also, as observed in Fig.5e and 5f, there appear a small period of time where the voltage needed exceeds the saturation value. This is apparently not an appealing phenomenon, but the system adjusts well after that period and goes back to normal quickly.

After that, we set θ_0 back to 0, and test the maximum noise that the controllers can take before they blow up. The results are shown in Table.II, where *controller 2* and *controller 3* are comparable to each other, and both of them are far better than *controller 1*.

Controller No.	1	2	3
max $\dot{\theta}$ noise power	0.0094	0.035	0.04
max $\dot{\phi}$ noise power	0.024	0.065	0.04

TABLE II: Maximum noise levels of each controller.

IV. Further improvements

In Zhu's thesis, he claimed that the maximum initial angle is 59°, approximately $1.03rad$, which is far greater than our controllers. In order to further improve the limit, we now try to change the some other initial values. For simplicity, we would only attempt on *controller 2* and *3* since we have seen that *controller 1* is clearly not competitive comparing to the other two. Also, since the parameters like ϕ and $\dot{\phi}$ do not directly affect the body angle θ, we did not expect to have significant improvements by changing them. Thus, we only tried to modify $\dot{\theta}_0$, $\dot{\theta}_{est,0}$, and $\theta_{est,0}$ for our improvement purpose.

A. Modifying initial body angle velocity

Firstly, we consider changing the initial value of $\dot{\theta}_0$ from 0 to some negative value. Since we have $\theta_0 > 0$, we could introduce an initial velocity $\dot{\theta}_0 < 0$ for compensation, which may help stabilizing at early stage. After several tests, the results show that there are indeed some improvements in the maximum θ_0. And the maximum improvements are:

- *Controller 2*: $\dot{\theta}_0 = -8.5rad \Rightarrow \theta_{0,\max} = 0.76rad$,
- *Controller 3*: $\dot{\theta}_0 = -6.5rad \Rightarrow \theta_{0,\max} = 0.80rad$,

From which we see some obvious improvements from before. And the above simulations are shown in Fig.6a and 6d.

B. Modifying initial body angle velocity estimation

Secondly, again since we have $\theta_0 > 0$, we may change the initial estimation of $\dot{\theta}_{est,0}$ to compensate that. The motivation of this is we make the controller think that the initial bias of θ is caused by an initial velocity $\dot{\theta}_0$ so that it knows what control to apply. After several tests, we found the maximum improvements below:

- *Controller 2*: $\dot{\theta}_{est,0} = 9.7rad/s \Rightarrow \theta_{0,\max} = 1.09rad$,
- *Controller 3*: $\dot{\theta}_{est,0} = 3.0rad/s \Rightarrow \theta_{0,\max} = 0.95rad$.

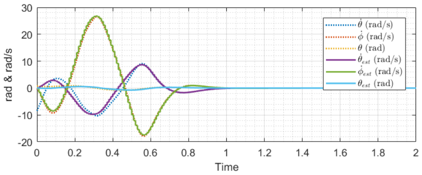
The results show that $\theta_{0,\max}$'s of both controllers become significantly larger. What's more, we found an initial value that enables *controller 2* to receive larger θ_0 than that in Zhu's Thesis. Although this could make the variation much larger when the initial angle is small, it is still acceptable as long as it converges. In other words, this setting of *controller 2* can afford the initial angle $\theta_0 \in [0, 1.09rad]$. And the above simulations are shown in Fig.6b and 6e.

C. Modifying initial body angle estimation

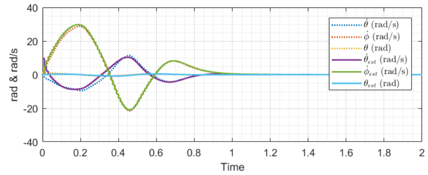
Apart from those mentioned above, we also try to increase the initial angle estimate $\theta_{est,0}$ to some positive values since we would actually have a positive initial angle θ_0. In other words, we are helping the system to do estimations, and the optimal results are obtain as followed after a series of trials:

- *Controller 2*: $\theta_{est,0} = 0.55rad \Rightarrow \theta_{0,\max} = 1.03rad$,
- *Controller 3*: $\theta_{est,0} = 0.26rad \Rightarrow \theta_{0,\max} = 0.978rad$.

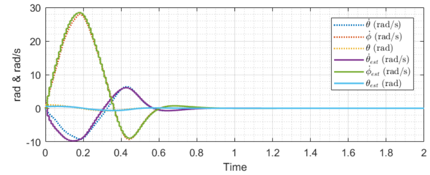
Again, we achieve some notable improvements. The simulations of these are shown in Fig.6c and 6f.



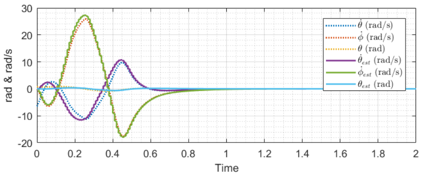
(a) *C2*: $\dot{\theta}_0 = -8.5rad/s,\ \theta_0 = 0.76rad$



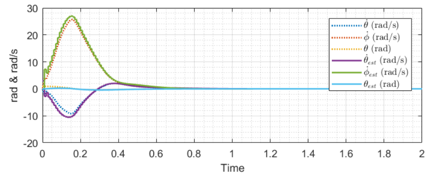
(b) *C2*: $\dot{\theta}_{est,0} = 9.7rad/s,\ \theta_0 = 1.09rad$



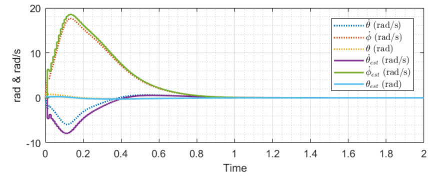
(c) *C2*: $\theta_{est,0} = 0.55rad,\ \theta_0 = 1.03rad$



(d) *C3*: $\dot{\theta}_0 = -6.5rad/s,\ \theta_0 = 0.80rad$



(e) *C3*: $\dot{\theta}_{est,0} = 3.0rad/s,\ \theta_0 = 0.95rad$



(f) *C3*: $\theta_{est,0} = 0.26rad,\ \theta_0 = 0.978rad$

Fig. 6: The states of *controller 2* and *3* with altered initial values, where the noise power is 0. (a)(b)(c) plot the states of *controller 2* (*C2* for short above) with modification on $\dot{\theta}_0$, $\dot{\theta}_{est,0}$, and $\theta_{est,0}$ respectively. (d)(e)(f) plot the states of *controller 3* (*C3* for short above) with modification on $\dot{\theta}_0$, $\dot{\theta}_{est,0}$, and $\theta_{est,0}$ respectively.

V. Conclusion

In conclusion, we design three controllers for the MIP robot system, and find out the maximum capability of each controller. Besides, we test the robustness of them against measurement noise, and the results show that they can all work find under some small noises. But as the noise power increases, *controller 1* loses stability quickly while the other two can afford much larger noise. Additionally, we try some further improvements by modifying the initial values and the outcome is quite satisfying, where *controller 2* is able to take an even larger θ_0 than that in Zhu's thesis. In summary, among all three designs, *controller 3* seems to perform the best under normal settings. However, with altering settings of initial conditions, *controller 2* could also have an outstanding performance. Future improvement can be taking ϕ into consideration, which could make the system more complicated but in the mean time stronger since we access to more information.

Acknowledgement

I would like to thank Professor Bob Bitmead for introducing us to the topic of linear system and control. Through MAE280A I gain some general ideas about systems, controllability, observavility, etc., which are definitely helpful for my future study and research.

References

[1] Zhuo, Zhu. *LQG Controller Design of the Mobile Inverted Pendulum*. University of California, San Diego, 2017.