Georgia Institute of Technology, Technical Report, GT-GOLEM-2014-004, 2014

Towards Stable Balancing

Munzir Zafar, Can Erdogan, Mike Stilman

1 Introduction

The purpose of this document is to establish a technical framework to control Krang in its stable balancing state. We will first discuss related work in inverted pendulum-like systems to get acquianted with the literature. Next, we will present a simple inverted pendulum model, analyze the dynamics and simulate it. We will evaluate the effect of different *control gains* in state feedback control and analyze the effect of *noise* in output feedback control. Lastly, we will study the effect of *modeling errors* in mass measurements.

2 Related Work

One of the first detailed work on a balancing robot is by Ha et al [2] where they approach the modeling problem directly from the Lagrangian mechanics. Although their formalization skips quite a few steps, the paper is ideal for trajectory control.

Another example of a two-wheeled balancing robot is the robot JOE [1]. The authors present a detailed explanation of their pole placement approach in the linearized system. The interesting points of the paper are (1) the decoupling of the motions around the vertical and lateral axes and (2) the effect of backlash on estimating body velocity from encoders.

Salerno et al. introduce Quasimoro, a quasiholonomic balancing robot in [5] and [4]. The most important difference between Quasimoro and JOE is the nonlinear control approach adopted. Also, the papers outline a detailed explanation of holonomic/nonholonomic systems and geometric/kinematic constraints, concepts that can be important in analyzing Krang. Note that this work apparently led to balancing wheelchair applications [3].

3 Simple Modeling

Imagine an actuated wheel with a rod on top of it and we would like to balance the rod. For the time being, we are going to ignore position control and just focus on stable balancing. Let x be the position of the wheel, and $\ddot{x}$ be the acceleration $\bar{a}_A$ of point A common to both the wheel and the rod. Let θ be the angle of the rod from the positive vertical axis. Let M and m be the masses of the wheel and the rod respectively and let l be the distance of the center of mass B of the rod from the point of rotation A on the wheel. We now define two frames of references: Frame XYZ is the world frame of reference fixed in the world and frame xyz is fixed on the rod with origin at the rotation point A. This is shown in Figure 1.

We refer to the unit vectors along XYZ as $\mathbf{I}$, $\mathbf{J}$ and $\mathbf{K}$ and the unit vectors along xyz as $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$. The following relationships hold:

$$\mathbf{i} = cos\theta\mathbf{I} - sin\theta\mathbf{J} \tag{1}$$

$$\mathbf{j} = sin\theta\mathbf{I} + cos\theta\mathbf{J} \tag{2}$$

$$\mathbf{I} = cos\theta\mathbf{i} + sin\theta\mathbf{j} \tag{3}$$

$$\mathbf{J} = -sin\theta\mathbf{i} + cos\theta\mathbf{j} \tag{4}$$

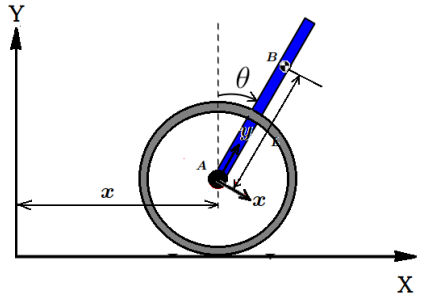

Figure 1: Simple wheeled pendulum model

3.1 Kinematics

The angular velocity and angular acceleration of the rod can be written as:

$$\bar{\omega}_{B/A} = -\dot{\theta}\mathbf{k} \tag{5}$$

$$\bar{\alpha}_{B/A} = -\ddot{\theta}\mathbf{k} \tag{6}$$

The resultant acceleration of point B in frame XYZ can be written as the sum of acceleration of point A and the tangential and centrifugal acceleration of point B with respect to point A:

$$\begin{aligned}
\bar{a}_B &= \bar{a}_A + (\bar{\alpha}_{B/A} \times \bar{r}_{B/A}) + (\bar{\omega}_{B/A} \times (\bar{\omega}_{B/A} \times \bar{r}_{B/A})) \\
\bar{a}_B &= \ddot{x}\mathbf{I} + (-\ddot{\theta}\mathbf{k} \times l\mathbf{j}) + (-\dot{\theta}\mathbf{k} \times (-\dot{\theta}\mathbf{k} \times l\mathbf{j})) \\
\bar{a}_B &= \ddot{x}\mathbf{I} + l\ddot{\theta}\mathbf{i} + (-\dot{\theta}\mathbf{k} \times l\dot{\theta}\mathbf{i}) \\
\bar{a}_B &= \ddot{x}\mathbf{I} + l\ddot{\theta}\mathbf{i} - l\dot{\theta}^2\mathbf{j}
\end{aligned} \tag{7}$$

Using equations (1-4) we can write the acceleration purely in one frame of reference. In frame xyz, we have:

$$\bar{a}_B = (\ddot{x}cos\theta + l\ddot{\theta})\mathbf{i} + (\ddot{x}sin\theta - l\dot{\theta}^2)\mathbf{j} \tag{8}$$

In frame XYZ, we have:

$$\bar{a}_B = (\ddot{x} + l\ddot{\theta}cos\theta - l\dot{\theta}^2 sin\theta)\mathbf{I} + (-l\ddot{\theta}sin\theta - l\dot{\theta}^2 cos\theta)\mathbf{J} \tag{9}$$

3.2 Dynamics

To analyze the dynamics of the system, we are going to use a free-body diagram approach as shown in Figure 2. The idea is to model all the substantial forces acting on the two objects and determine their relationship. The wheel is being driven by a motor fixed on the rod. The wheel has the driving torque τ, the friction torque τ_f, the forces R_x and R_y that are the reactions to the constraint forces acting on the rod, a rolling friction between the wheel and ground f_G prevents the wheel from slipping, the weight of the wheel Mg and the normal N from the ground. For the rod, we have the weight of the rod mg, the unknown constraint forces R_x and R_y, and the external force F_{ext} acting along x-axis at B. Note also that the rod is going to experience the equal and opposite torques τ and τ_f being applied on the wheel. Also, I, I_w and I_m denote the inertia of the rod, wheel and motor respectively. The gear ratio of the motor is η.

The goal is to determine the state equations in terms of the known quatities. For this, we apply Newton-Euler equations on the wheel and the rod. Evaluating torque equation on the wheel-motor system about Z-axis, and noting that three torques $-\tau$, τ_f and $f_G r$ are acting on it, while the rotational acceleration of the wheel is $-\frac{\ddot{x}}{r}$ and that of the motor is $-(\frac{\ddot{x}}{r} - \ddot{\theta})$, we have

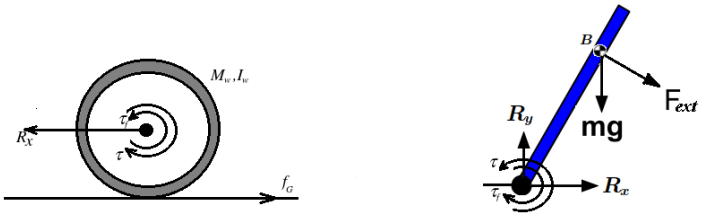

Figure 2: Free-body diagram

$$\begin{aligned}\mathbf{K}.(\sum \bar{M}_A) &= I_{zz}\alpha_z \\ -\tau + \tau_f + f_G r &= -I_w\frac{\ddot{x}}{r} - \eta^2 I_m(\frac{\ddot{x}}{r} - \ddot{\theta})\end{aligned} \tag{10}$$

Applying Newton's equation $\sum \bar{F} = m\bar{a}$ along X-axis on the wheel, i.e. taking the dot-product of the vector equation with unit vector $\mathbf{I}$, we get

$$\begin{aligned}\mathbf{I}.(\sum \bar{F}_A &= M\bar{a}_A) \\ f_G - R_x &= M\ddot{x} \\ f_G &= M\ddot{x} + R_x\end{aligned} \tag{11}$$

Applying Newton's equation on the rod along X-axis, we get

$$\begin{aligned}\mathbf{I}.(\sum \bar{F}_B &= m\bar{a}_B) \\ \mathbf{I}.(R_x\mathbf{I} + F_{ext}\mathbf{i}) &= m\ddot{x} + ml\ddot{\theta}cos\theta - ml\dot{\theta}^2 sin\theta \\ R_x + F_{ext}cos\theta &= m\ddot{x} + ml\ddot{\theta}cos\theta - ml\dot{\theta}^2 sin\theta \\ R_x &= m\ddot{x} + ml\ddot{\theta}cos\theta - ml\dot{\theta}^2 sin\theta - F_{ext}cos\theta\end{aligned} \tag{12}$$

Substituting R_x from equation 12 in 11 and then the resulting f_G from equation 11 in 10, and using $\tau = K_f u$ we get

$$\begin{aligned}-\tau + \tau_f + (M\ddot{x} + m\ddot{x} + ml\ddot{\theta}cos\theta - ml\dot{\theta}^2 sin\theta - F_{ext}cos\theta)r &= -I_w\frac{\ddot{x}}{r} - \eta^2 I_m(\frac{\ddot{x}}{r} - \ddot{\theta}) \\ [(m+M)r + I_w/r + \eta^2 I_m/r]\ddot{x} + (mrlcos\theta - \eta^2 I_m)\ddot{\theta} &= K_f u - \tau_f + F_{ext}rcos\theta + mrl\dot{\theta}^2 sin\theta\end{aligned} \tag{13}$$

where u is the current supplied to the wheel-motors of the wheel and is our control input and K_f is the conversion factor that converts this current to the actual force τ applied on the wheel.

Secondly, we evaluate the torques on the rod-motor system about z-axis at point B:

$$\begin{aligned}\mathbf{k}.(\sum \bar{M}_B) &= I_{zz}\alpha_z \\ \tau - \tau_f + \mathbf{k}.(\bar{r}_{A/B} \times R_x\mathbf{I} + \bar{r}_{A/B} \times R_y\mathbf{J}) &= -I\ddot{\theta} + \eta^2 I_m(\ddot{x}/r - \ddot{\theta}) \\ \tau - \tau_f + \mathbf{k}.[(-l\mathbf{j}) \times R_x(cos\theta\mathbf{i} + sin\theta\mathbf{j}) + (-l\mathbf{j}) \times R_y(-sin\theta\mathbf{i} + cos\theta\mathbf{j})] &= -I\ddot{\theta} + \eta^2 I_m(\ddot{x}/r - \ddot{\theta}) \\ \tau - \tau_f + R_x lcos\theta - R_y lsin\theta &= -I\ddot{\theta} + \eta^2 I_m(\ddot{x}/r - \ddot{\theta})\end{aligned} \tag{14}$$

Now, applying Newton-Euler equations along the x-axis on the rod gives:

$$\mathbf{i}.(\sum \bar{F}_B = m\bar{a}_B)$$
$$\mathbf{i}.(R_x\mathbf{I} + R_y\mathbf{J} - mg\mathbf{J} + F_{ext}\mathbf{i}) = m\ddot{x}cos\theta + ml\ddot{\theta}$$
$$R_xcos\theta - R_ysin\theta + mgsin\theta + F_{ext} = m\ddot{x}cos\theta + ml\ddot{\theta}$$
$$R_xcos\theta - R_ysin\theta = -F_{ext} - mgsin\theta + m\ddot{x}cos\theta + ml\ddot{\theta} \tag{15}$$

The left-hand side of this equation appears in equation 14. Upon substitution, we get

$$\tau - \tau_f + (-F_{ext} - mgsin\theta + m\ddot{x}cos\theta + ml\ddot{\theta})l = -I\ddot{\theta} + \eta^2 I_m(\ddot{x}/r - \ddot{\theta})$$
$$(mlcos\theta - \eta^2 I_m/r)\ddot{x} + (ml^2 + I + \eta^2 I_m)\ddot{\theta} = -K_f u + \tau_f + F_{ext}l + mglsin\theta \tag{16}$$

Equations 13 and 16 give the description of the dynamics of the system in terms of the known parameters. These will be utilized in the later sections for linearization, state-space equations and simulation. Now, we assign the variables as follows: (1) $x_1 = \theta$, (2) $x_2 = \dot{\theta}$, (3) $x_3 = x$ and (4) $x_4 = \dot{x}$, and rewrite the equations in the state-space form:

$$\dot{x}_1 = x_2 \tag{17}$$
$$\dot{x}_3 = x_4 \tag{18}$$
$$\begin{bmatrix} \dot{x}_2 \\ \dot{x}_4 \end{bmatrix} = \frac{1}{\Delta}\begin{bmatrix} [(m+M)r + I_w/r + \eta^2 I_m/r] & -(mlcosx_1 - \eta^2 I_m/r) \\ -(mrlcosx_1 - \eta^2 I_m) & (ml^2 + I + \eta^2 I_m) \end{bmatrix}\begin{bmatrix} -K_f u + \tau_f + F_{ext}l + mglsinx_1 \\ K_f u - \tau_f + F_{ext}rcosx_1 + mrlx_2^2 sinx_1 \end{bmatrix} \tag{19}$$

where, $\Delta = [(m+M)r + I_w/r + \eta^2 I_m/r](ml^2 + I + \eta^2 I_m) - r(mlcosx_1 - \eta^2 I_m/r)^2$

3.2.1 Friction term τ_f

We make an assumption on the friction term that it comprises of three components: the static coulomb friction, the kinetic columb friction and the linear viscous friction.

$$\tau_f = \eta_f(u, x) + bx_4$$

where bx_4 represents the linear viscous friction due to air resistance and is linearly dependent on the speed of the wheel $x_4 = \dot{x}$. And η_f represents the coulomb friction. This friction term is a nonlinear function of the states x and the control input u. When the system is moving (i.e. $|x_4| > 0$) this firction is equal to kinetic coulomb friction $\mu_k(m+M)gr\,sign(x_4)$, where μ_k is the co-efficient of friction in the gear box. This friction changes direction depending on the sign of the wheel speed x_4. When the the wheel is not moving i.e. when $x_4 = \dot{x}_4 = 0$ then equation 13 gives

$$\eta_f|_{x_4=\dot{x}_4=0} = K_f u - (mrlcos\theta - \eta^2 I_m)\ddot{\theta} + F_{ext}rcos\theta + mrl\dot{\theta}^2 sin\theta$$
$$\eta_f|_{x_4=\dot{x}_4=0} = K_f u - (mrlcosx_1 - \eta^2 I_m)\dot{x_2} + F_{ext}rcosx_1 + mrlx_2^2 sinx_1 \tag{20}$$

$\dot{x}_2$ can be determined in terms of other states by substituting $\ddot{x} = 0$ in equation 16. We get,

$$\ddot{\theta}|_{x_4=\dot{x}_4=0} = \frac{-K_f u + \tau_f + F_{ext}l + mglsin\theta}{ml^2 + I + \eta^2 I_m}$$
$$\dot{x}_2|_{x_4=\dot{x}_4=0} = \frac{-K_f u + \tau_f + F_{ext}l + mglsinx_1}{ml^2 + I + \eta^2 I_m} \tag{21}$$

Substituting this in equation 20, we get

$$\eta_f|_{x_4=\dot{x}_4=0} = \frac{(ml^2 + I + \eta^2 I_m)(K_f u + F_{ext}rcosx_1 + mrlx_2^2 sinx_1) + (mrlcosx_1 - \eta^2 I_m)(-K_f u + F_{ext}l + mglsinx_1)}{ml^2 + I + 2\eta^2 I_m - mrlcosx_1} = \Gamma(u,x)(say) \tag{22}$$

Equation 22 gives the expression the coulomb friction when the robot is not moving. The robot starts moving when the right hand side of this equation exceeds $\mu_s(m+M)g$.

$$\eta(u,x) = \begin{cases} \mu_k(m+M)gr\,sign(x_4) & : |x_4| > 0 \\ \Gamma(u,x) & : x_4 = 0\ and\ |\Gamma(u,x)| < \mu_s(m+M)g \\ \mu_s(m+M)gr\,sign(\Gamma(u,k)) & : x_4 = 0\ and\ |\Gamma(u,x)| > \mu_s(m+M)g \end{cases} \tag{23}$$

3.3 Linearization and the state-space form

Given the dynamics equations, we want to linearize the system and put it in the state-space form to use LQR technique in order to determine the feedback control gains. If we linearize around $\theta = 0$ and use the infinitesmally small difference ϕ as the new variable, we would approximate the nonlinear functions $cos(\theta)$ and $sin(\theta)$ as follows: $cos(\theta) \approx 1$ and $sin(\theta) \approx \phi$. Also, we do not take into account the force F_{ext} when linearizing the system because F_{ext} is only modeled as an unknown disturbance to the system when simulating the system. For the friction term τ_f only the linear component $b\dot{x}$ is substituted. So, the linearized equations turn out to be:

$$(ml^2 + I + \eta^2 I_m)\ddot{\phi} + (ml - \eta^2 I_m/r)\ddot{x} = -K_f u + b\dot{x} + mgl\phi \tag{24}$$

$$(mrl - \eta^2 I_m)\ddot{\phi} + [(m+M)r + I_w/r + \eta^2 I_m/r]\ddot{x} = K_f u - b\dot{x} \tag{25}$$

To make the final expression compact, let's define the coefficients of the states occuring on the left-hand sides of the above equations: $A = ml^2 + I + \eta^2 I_m$, $B = ml - \eta^2 I_m/r$, $C = mrl - \eta^2 I_m$ and $D = (m+M)r + I_w/r + \eta^2 I_m/r$. And let $E = AD - BC$. The resulting state-space form is as follows:

$$\begin{bmatrix} \dot{\phi} \\ \ddot{\phi} \\ \dot{x} \\ \ddot{x} \end{bmatrix} = \frac{1}{AD - BC} \left(\begin{bmatrix} 0 & 1 & 0 & 0 \\ Dmgl/E & 0 & 0 & (B+D)b/E \\ 0 & 0 & 0 & 1 \\ -Cmgl/E & 0 & 0 & -(A+C)b/E \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \\ x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ -(B+D)/E \\ 0 \\ (A+C)/E \end{bmatrix} K_f u \right)$$

4 Simulation

For simulating the inverted pendulum on the wheel whose equations are derived above, we encoded the equations 17 to 19 in an M-file named *invPendulumOnCartFext.m* listed below:

```
function xdot = invPendulumOnCartFext(t,x)
global m b g M l I TorquePerAmp u Fext Iw Im eta r
Fext=0; u =0;
% x=[phi phidot x xdot]
del=((m+M)*r+Iw/r+eta^2*Im/r)*(m*l^2+I+eta^2*Im)-r*(m*l*cos(x(1))-eta^2*Im/r)^2;
xdot(1) = x(2);
xdot(2) = (1/del)*(((m+M)*r+Iw/r+eta^2*Im/r)*(-TorquePerAmp*u+b*x(4)+...
    Fext*l+m*g*l*sin(x(1)))-(m*l*cos(x(1))-eta^2*Im/r)*(TorquePerAmp*u...
    -b*x(4)+Fext*r*cos(x(1))+m*r*l*x(2)^2*sin(x(1))));
xdot(3) = x(4);
xdot(4) = (1/del)*(-(m*r*l*cos(x(1))-eta^2*Im)*(-TorquePerAmp*u+b*x(4)+...
    Fext*l+m*g*l*sin(x(1)))+(m*l^2+I+eta^2*Im)*(TorquePerAmp*u-b*x(4)+...
    Fext*r*cos(x(1))+m*r*l*x(2)^2*sin(x(1))));
xdot = xdot';
```

This code accepts the current state and based on the present values determines the gradients of the state. This function can now be solved using a solver such as *ode45* in MATLAB. We used the following code to simulate the system:

```
global m b g M l I u Fext TorquePerAmp Iw Im eta r

% Constants
M = 9.32;   % Mass of the cart (the two wheels) [kg]
m = 135.18; % Mass of the pendulum (the rest of the robot) [kg]
b = 0.8;    % Co-efficient of friction for the cart (taken from a web site)
I = 30 ;    % mass moment of inertia (approximation)
g = 9.8;
l = 0.25;   % Length to pendulum Center of Mass
            % (Our center of mass whose mean value is empirically found out
            % to be 0.25 m)
TorquePerAmp = 2 * 0.85;   % No. of Wheels = 2
                           % TorquePerAmp = 0.85 Nm/A
r = 10 * 0.0254;      % Wheel radius = 10 in
Iw = 4.66*(0.8*r)^2;  % wheelmass * (0.8*wheel_radius)^2
Im = 1.6e-4;     % rotor inerita [this rating is for BLY343S-48V-3200, with
                 % rotor inertia = 0.02266 oz-in-sec2.
                 % The motor we have is BLY343D-24V-2000, with no specs.
eta = 15;        % Gear Ratio

% Simulate
tf=19; dt=0.005; t=0;
x0 = [0; 0.1; 0; 0];
X=[]; T=[]; U = [];
u=0;
[T,X] = ode45('invPendulumOnCartFext',[0,tf],x0);

figure(1); subplot(221); plot(T,X(:,1)*180/pi);
title('phi'); xlabel('time (sec)'); ylabel('phi (degrees)'); grid on;
figure(1); subplot(222); plot(T,X(:,2)*180/pi);
title('phidot'); xlabel('time (sec)'); ylabel('phidot (deg/sec)'); grid on;
figure(1); subplot(223); plot(T,X(:,3));
title('x'); xlabel('time (sec)'); ylabel('x (m)'); grid on;
figure(1); subplot(224); plot(T,X(:,4));
title('xdot'); xlabel('time (sec)'); ylabel('xdot (m/s)'); grid on;


if(1)
  axis_speed_gain=0.1; axis_center=0; plot_memory=20; plot_step=2;
  figure(2);
  clf;
  for j = 1 : plot_step : size(X,1)
      for c=1:plot_memory
          k=j-plot_step*(plot_memory-c);
          if(k>0)
              color=(1-exp(-(plot_memory-c)/(plot_memory/5)))*[0.8 0.8 0.8];
              linewidth = 1+4*(exp(-(plot_memory-c)/(plot_memory/5)));
              plot([X(k,3);X(k,3)+3*l*sin(X(k,1))],[0; 3*l * cos(X(k,1))] ...
              , '-', 'Color',color,'linewidth', linewidth);  hold on;
              plot(X(k,3),0,'o','MarkerSize',5,'Color',color,'linewidth', linewidth);
              set(gca, 'fontsize', 20); hold on;
          end
      end
      axis_err=X(j,3)-axis_center;
      axis_speed=axis_speed_gain*axis_err;
      axis_center=axis_center+axis_speed;
      axis([axis_center-1.5 axis_center+1.5, -1 1]);
      str=sprintf('time=%.3f sec',T(j));
      text(axis_center+0.05,0.85,str, 'fontsize', 24);
      grid on;
      pause;
      clf;
  end
end
```

In the above two code listings, the term *TorquePerAmp* is used in place of K_f in the foregoing equations. All the following simulation results will be generatd by small variations in the above code listings. The specific change will be mentioned as required.

4.1 Uncontrolled System

When the system defined by state equations of the nonlinear system is simulated setting u and F_{ext} to zero and initial state to $x_{t=0} = \begin{bmatrix} 0 & 0.1 & 0 & 0 \end{bmatrix}^T$, we observe the behavior depicted in Figures 3 and 4

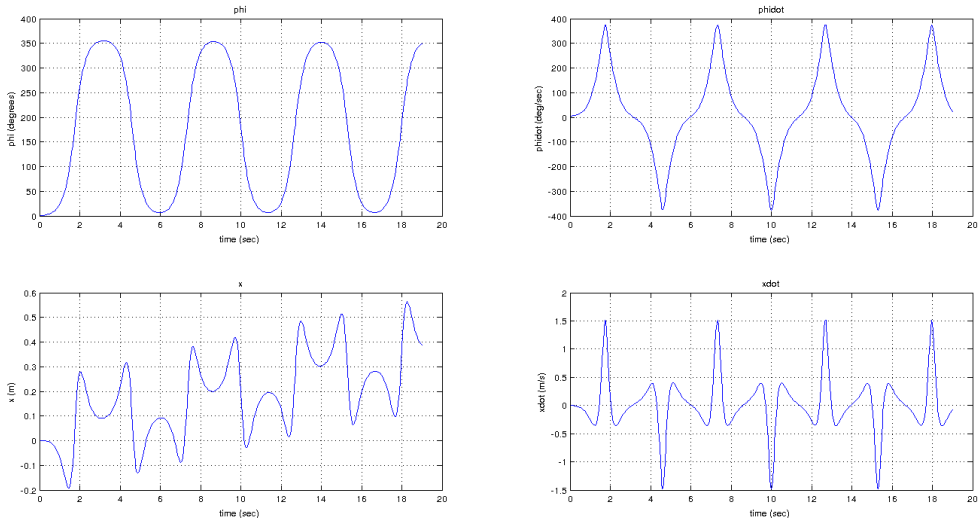

Figure 3: Plots of the states of the uncontrolled system

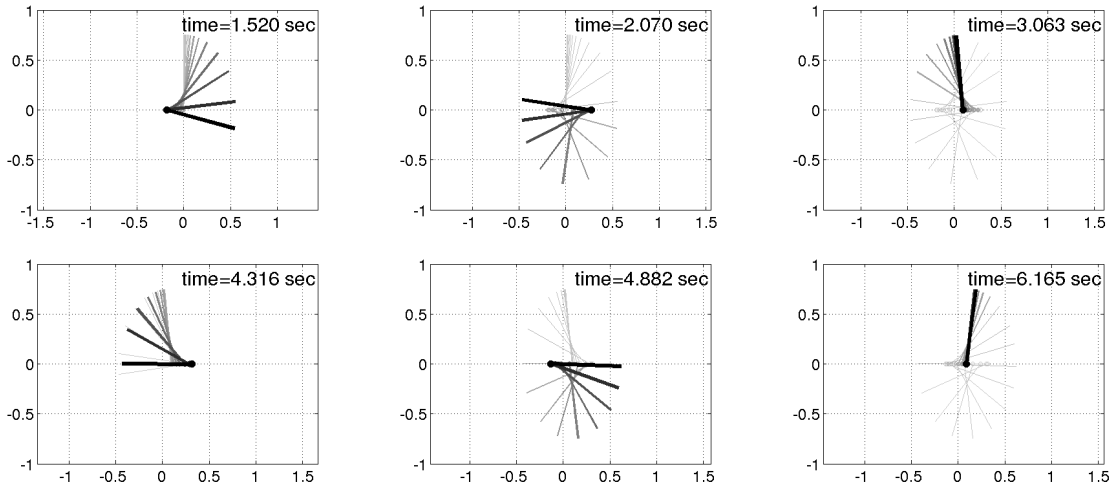

Figure 4: Snapshots of the simulation of the inverted pendulum on wheel at different time instances. The darkest line represents the most current state each snapshot, while the grey lines represents previous positions

This is, in a sense, verification of the derived equations and the implementation in MATLAB. That's because, in the real world, we expect the system to oscillate between the extremities depicted in the figures. This is because the pendulum in the absence of control input u and external force F_{ext} should oscillate. Other interesting phenomena observed is the fact that as the pendulum oscillates the wheel also oscillates. This happens as a result of the radial component of the weight of the pendulum. The tangential component of the weight is moving the rod while the radial component if moving the wheel. As the time proceeds we also notice that the wheel center position is shifted to a positive direction. This is because as the system is losing energy due to ground friction the amount of energy left in the system

in every next half cycle is smaller than the previous. So the displacement caused by the first half cycle is dominating the resultant direction of displacement.

4.2 Response Of Controlled System to varying Impulsive Forces

We now close the loop using gains derived by applying LQR on the linearized model of the system. The gains calculated using LQR are $K = \begin{bmatrix} -225.8715 & -70.5301 & -1.0000 & -7.8310 \end{bmatrix}^T$ and the loop is closed by setting $u = -K(x - x_{ref})$ where $x_{ref} = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T$. Since we are not interested in controlling the position x of the system, we keep its control gain equal to zero hence the control gain vector becomes $K = \begin{bmatrix} -225.8715 & -70.5301 & 0 & -7.8310 \end{bmatrix}^T$. We keep the initial condition of the system to zero but we apply an impulsive force such that $F_{ext} = F_{impulse}$ for $t < 0.1sec$ anf then $F_{ext} = 0$ for $t > 0.1sec$. Figures 5 and 6 show the resultant behavior of the system. The following observations are made:

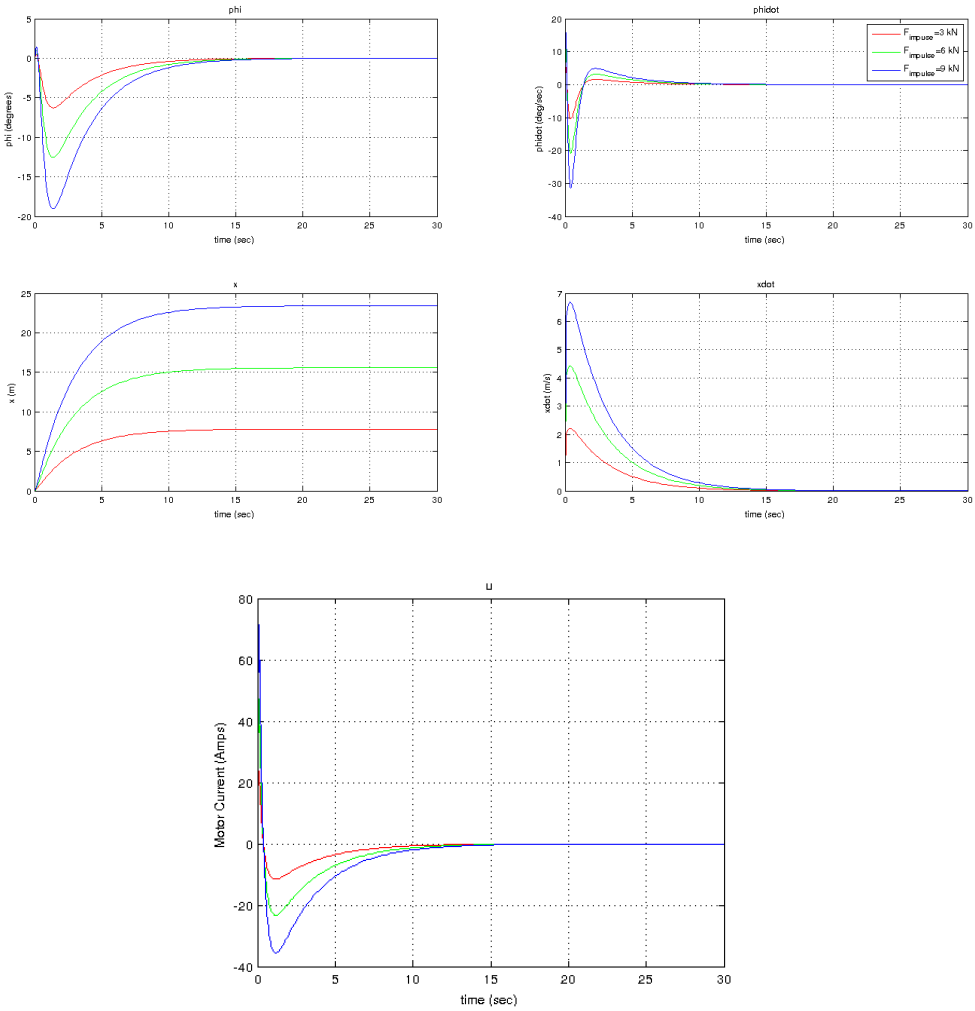

Figure 5: Plots of the states of the controlled system subjected to varying impulsive forces

1. The closed loop system is stable
2. As F_{ext} is increased, the settling time of the system remains unchanged and is about 10 seconds.
3. Since the three states ϕ, $\dot{\phi}$, and $\dot{x}$ are the only controlled states they all returned back to zero after the system has settled while tbe position has changed to a non-zeor value signifying that the system undergoes a resultant displacement in the direction of the impulsive force.

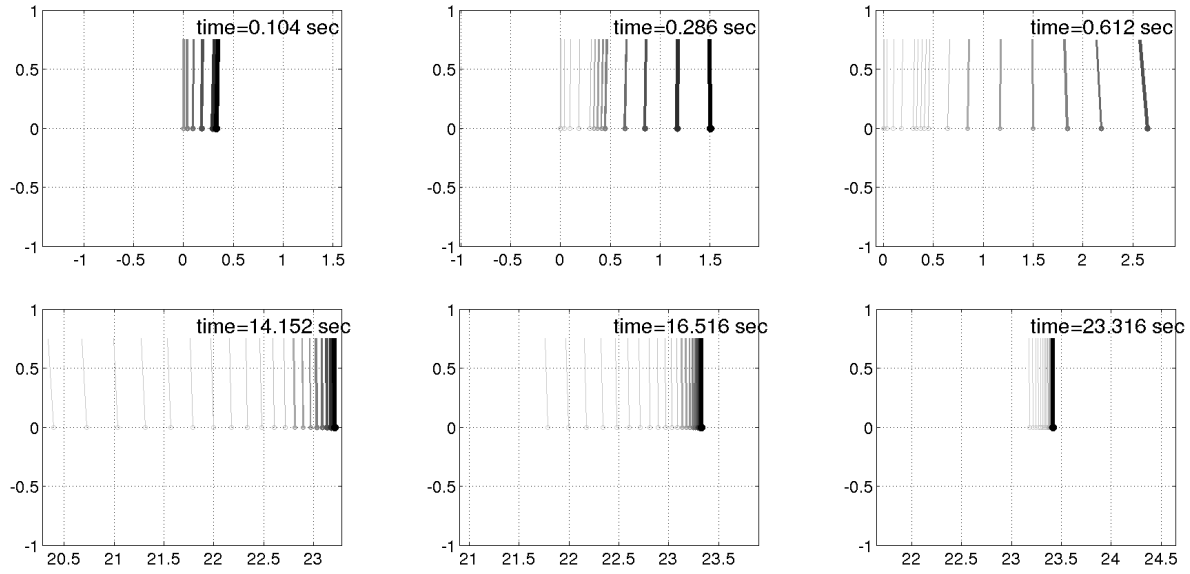

Figure 6: Snapshots of the simulation of the controlled system different time instances after it's been subjected to an impulsive force

4. As the magnitude of the force $F_{impulse}$ is increased we observe a proportional increase in the peak values of the states

Point number 2 above however is a bit unrealistic. In general when we kick our robot, we observe that it undergoes larger displacements and the settling time usually increases with the increase of force in the kick. This shows that there is a discrepancy in our modeling of the system. This discrepancy is visible when we look at the plot of the control input u. We see that the motor current keeps increasing with increaing force. And reaches as high as 100A for the case when $F_{ext} = 9000N$. This however is not possible for the real system we are trying to control. The upper limit on the value of motor current is $50A$. Thus we next show the response of the same system in the presence of saturation limit on u.

4.3 Effect of Saturation limits on Control Input u

We now move on to introduce the saturation limit on the control input u to the system. So $-50A < u < 50A$ is the limit on the values of u at all times. The resulting plots are shown in 7. We observe that as the force impulse increases to $9000N$ we have increase in the settling time of the system. Also, since the value of u is now limited we see the peak values of the state reaching to higher levels beyond the ones we saw earlier in the absence of the saturation limit on u.

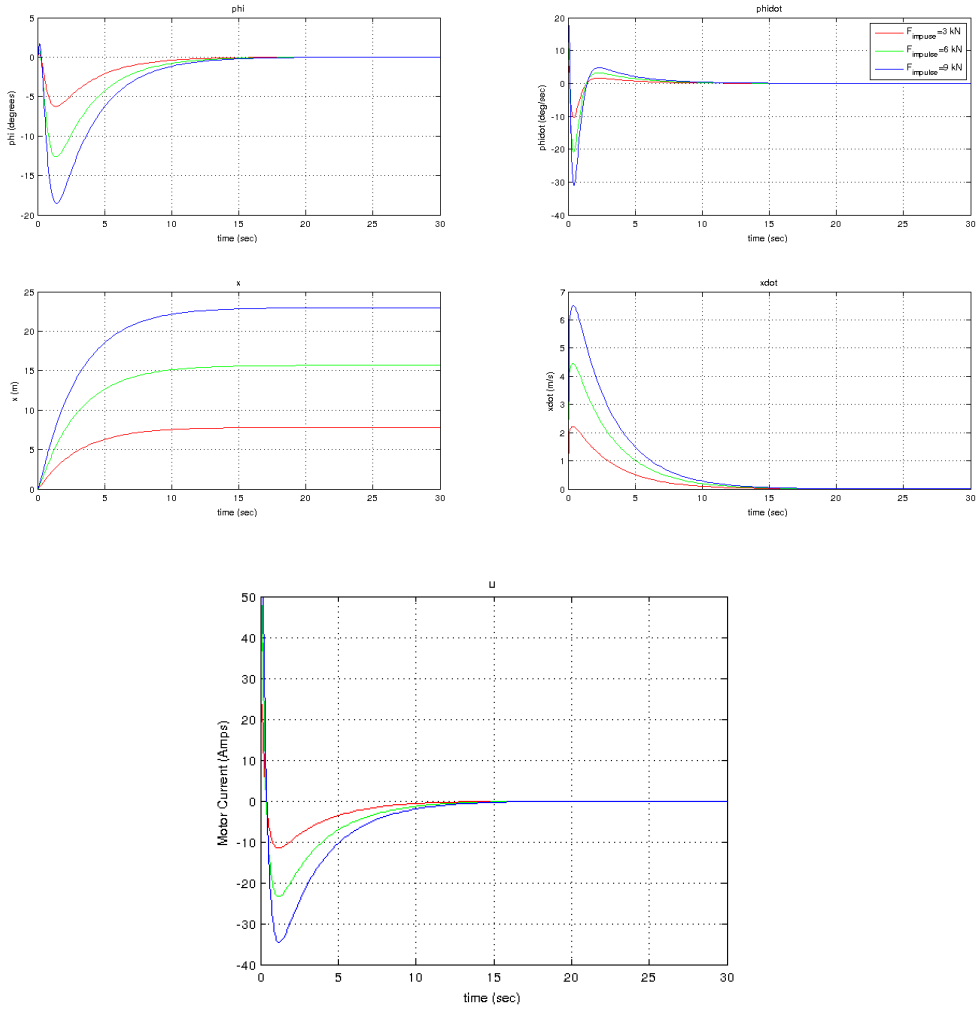

Figure 7: Plots of the states of the controlled system with saturation limits on the control input u, subjected to varying impulsive forces

4.4 Effect of Discretization of Control Input u

Another aspect of the real world that we didn't introduce in our is the effect of discretization of the control input u. Even though our real systen is a continuous system, the feedback control of the system is taking place inside a computer in an iterative fashion. The time of each iteration is the interval between successive updates of the control input u. This time is currently $2ms$ but it can change based on the updates in the alogorithms and so forth. We want to know how this discretization of u affects the performance of the system. So we plot the responses of the system in Figure 8 for time intervals T_s ranging from 2 ms to 500 sec. We note that upto 50ms the response of the system (to an impuslive force of 2000N) closely matches the response of the actual system. From 50 ms to 200 ms we begin to see a rise in the overshoots and peaks of the system. Increasing the interval beyond 200 ms causes instability and the system blows up.

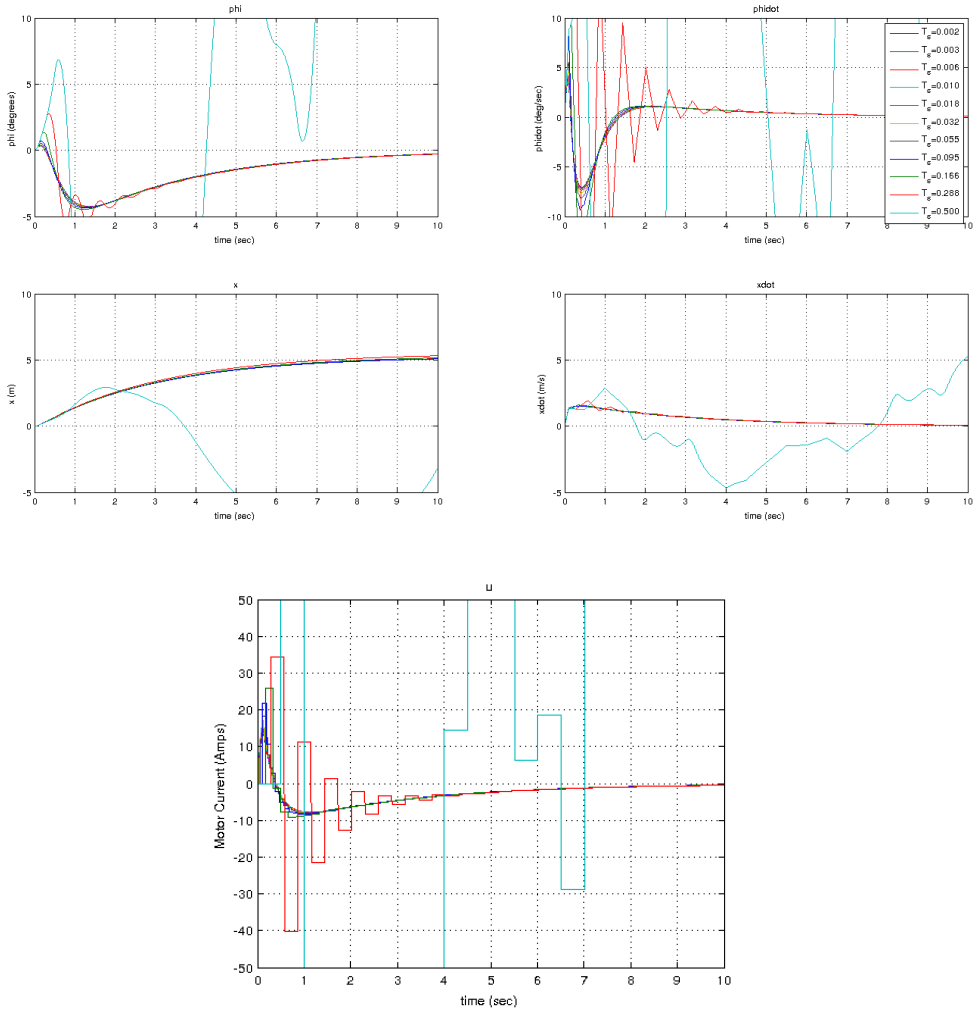

Figure 8: Plot of the response of the system when the control input is discrete with time intervals T_s ranging from 2 ms to 500ms

4.5 Effect of changing Control Gains

4.5.1 Changing K_P

Now we want to determine the sensitivity of our system to the values of the control gains. We refer to the first two elements of our control gain vector K as K_P and K_D. For the purpose of finding the effect of changing K_P, we keep K_D fixed at the opitmum value returned by LQR (134) and we use $F_{impulse} = 2000N$ to test the system. We make the following observations about the effect of changing the gain values K_P from 250 to 650.

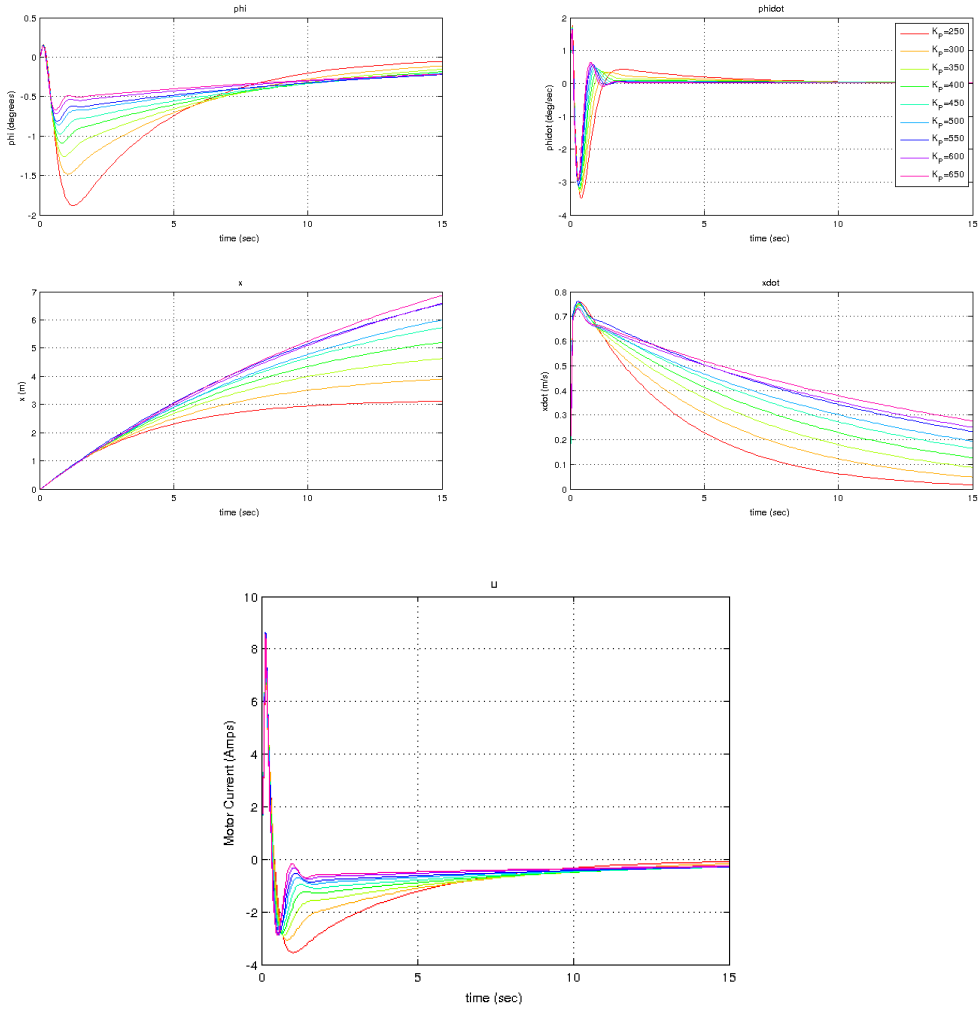

Figure 9: Plots of the states of the controlled system with differing values of the control gains K_P

Using smaller K_P (250) causes:

1. a higher settling time (about 15 seconds)
2. oscillations in phi (one and a half cycle until settling)
3. displacement of one meter
4. motor current peaks at 30A, oershoots to -5A before stabilizing

Increasing K_P upto the optimum 400 corresponds to

1. decreasing settling time (best achieved is 6 seconds)

2. oscillation go away
3. resulting displacement remains the same
4. motor current peak remains the same, overshoot decreases to -1A

Increasing K_P more (upto 650) has following effects:

1. Settling time further decreases (goes as low as 2 seconds)
2. Oscillations go away.
3. resulting displaement decreases by just a bit
4. Motor current starts peaking more (goes as high as 33A), and overshoot in both phi and motor current increase to -3Nm.

4.5.2 Changing K_D

Now we study the effect of changing the second gain in our control gain vector K_D. We keep the value of K_P fixed to its optimum -400 and we fix $F_{impulse}$ to 2000N. We then change K_D from -14 to -374 in step of -30.

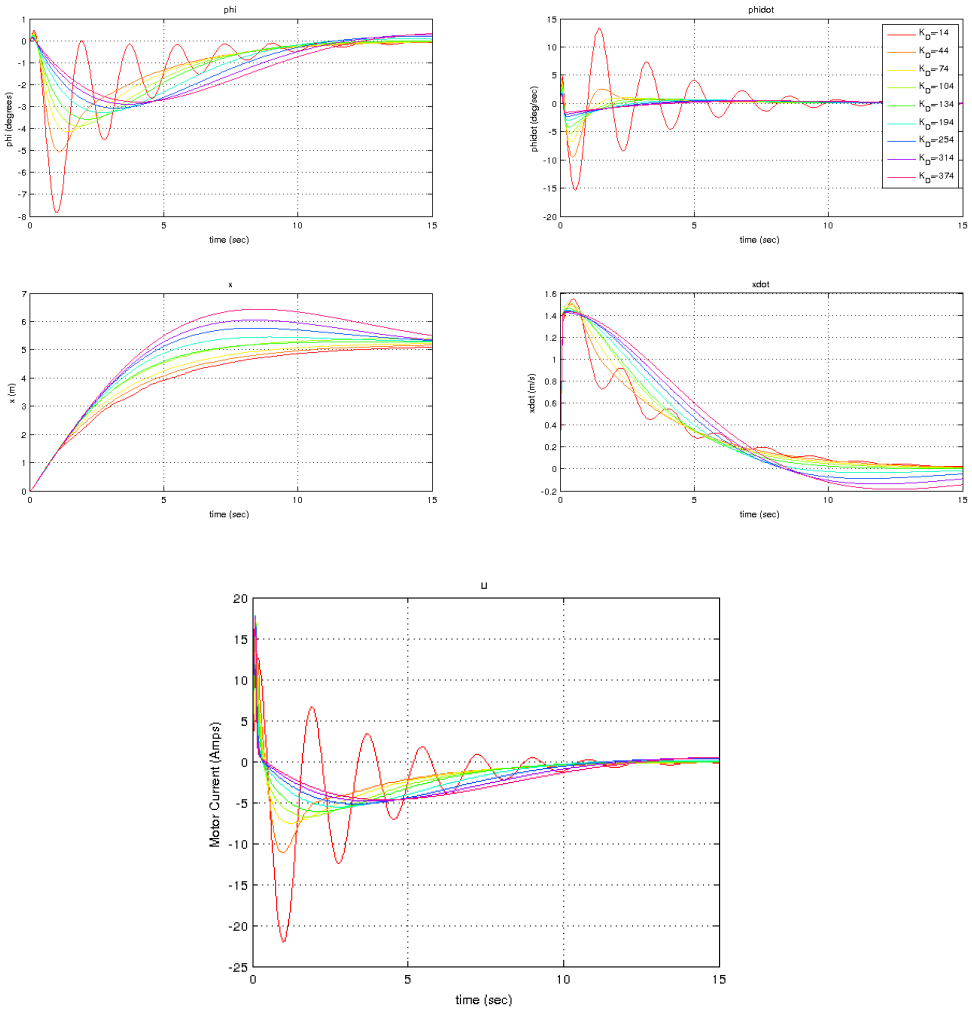

Figure 10: Plots of the states of the controlled system with differing values of the control gains K_P

The follwoing is what we observe: As we change K_D we note that the settling time of the system remains unchanged to 6 seconds. Also, the resultant displacement of the wheel remains unchanged to 0.8m.

The system is unstable when K_D is small. As we increase K_D the system repsonse is pretty sensitive in terms of the oscillations. As we move closer to the LQR optimum -134 the oscillations are minimized. We also observe that the peak current is reducing as we increase the gain in the below-optimum range.

As we reach the optimum -134, we observe that there is still some overshoot in the current and the states ϕ and $\dot{\phi}$. The oscillations however are gone.

Increasing K_D beyond the optimum removes all overshoots in ϕ, $\dot{\phi}$ and u, which is actually very good. But one thing that is not good is the value of motor current peaks. The current starts peaking more from the optimum 30A to as high as 50A when the value of K_D is increased from the optimum.

4.6 Effect of changing I and l on Control Gains

This study is done to see how much the gain values on ϕ and $\dot{\phi}$ depend on the inertia value of the system I and the center of mass of the system l. We note that the values of the optimum gains decrease with increase in the inertia of the system I and increases with the increase in the distance of the Center of Mass of the system. The value of K_P remains between 360 and 460. The value of K_D however varies alot with changing I and l. It goes from 50 (for high I and low l) to 380 (for low I and high l). This indicates it is important to have a good idea of the parameters I and l on the real system. And as the robot moves we should be able to change the values of the gains to adjust to the changing dynamics of the system.

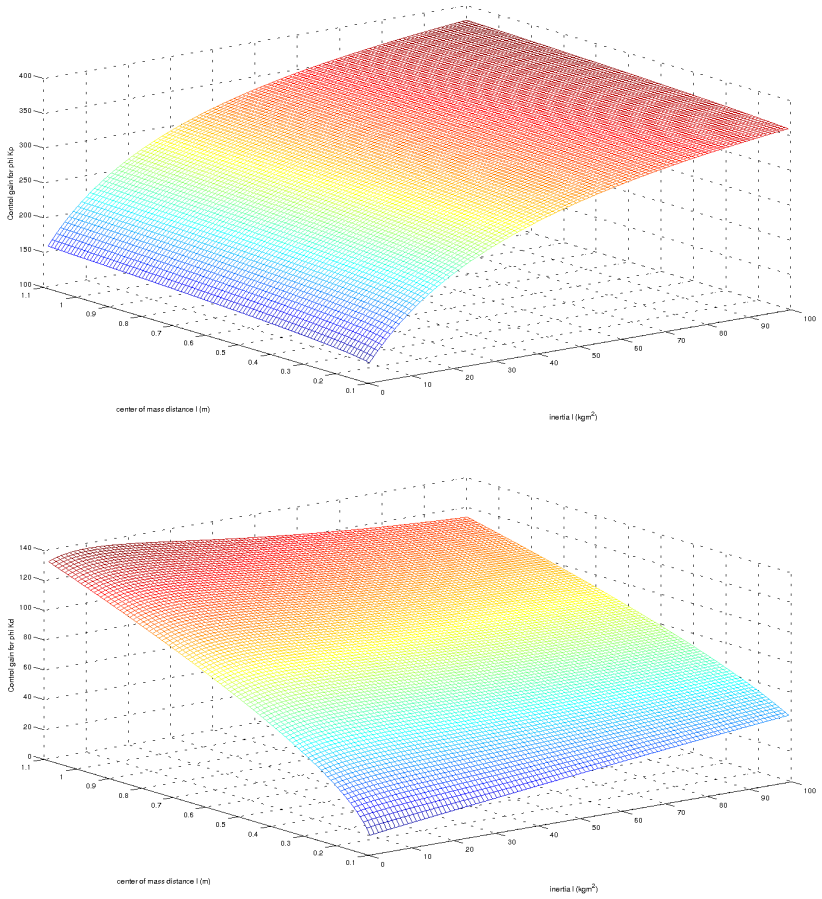

Figure 11: Mesh plots of the gains K_P and K_D as a function of the inertia I and COM distance l

4.7 Region Of Attraction

Given the parameters we have defined for our system we want to figure out the region of attraction (ROA) of our system. More specifically, we want to find out what is the initial value of the angle ϕ from which the system can recover its equilibrium. For this purpose we try running the simulation at different initial values of the state variable ϕ and observe at what value the system goes unstable. It turns out that the system is able to recover its equilibrium until 21.8 degrees. As the initial value is increased further, the system goes unstable (Figure 12).

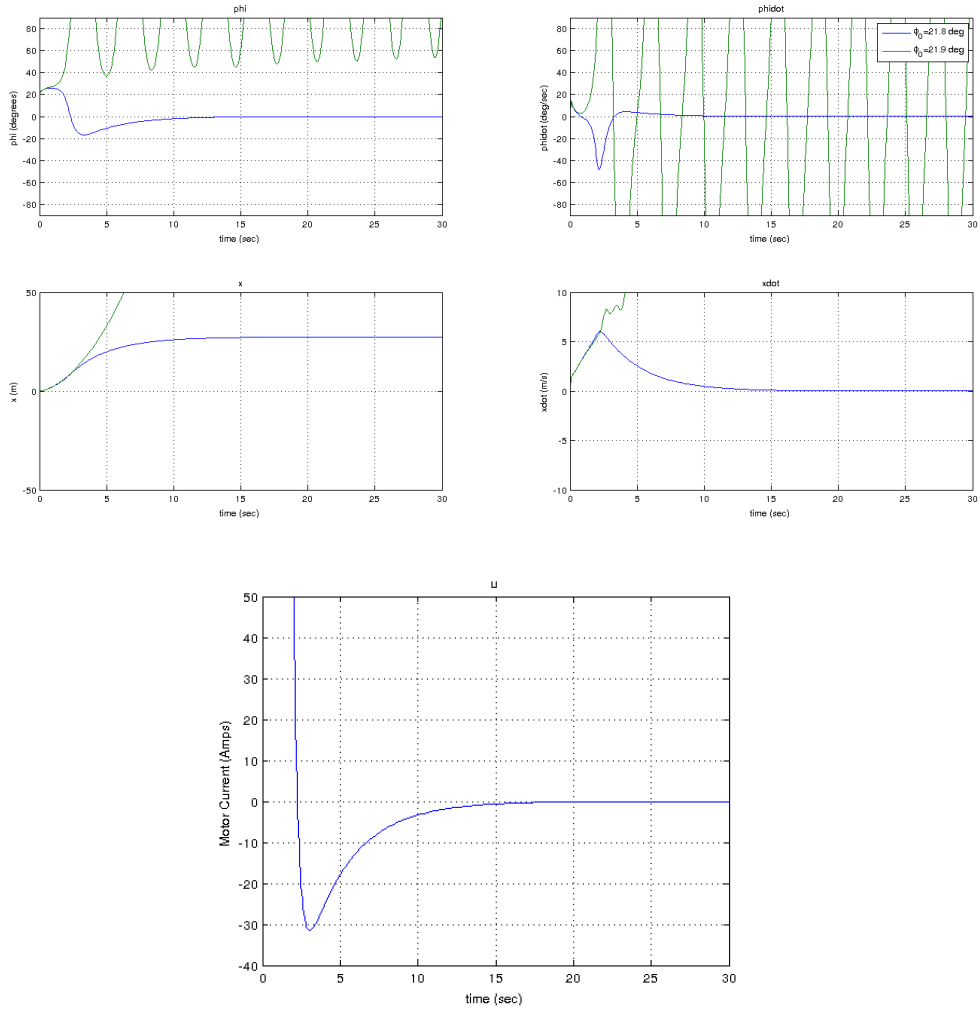

Figure 12: The response of the system when the initial value of ϕ is 11.2 and 11.3 degrees away from the equilibrium. We see that at 11.3 degrees the system blows up

The problem with this angle however is that it takes about 32.5 meters of back-travelling by the wheel to get back to the equilibrium. This is impracticable because the wheel may have obstacles along its path. We don't want the distance to be traveled beyond a maximum of 7 meters. Table 1 shows how the displacement of x in the steady state is affected by the initial value of the angle ϕ_0.

We can easily imagine that our system, given the assumed system dynamics can not be expected to rise from its initial position unless the COM is within 8-9 degrees from the vertical line above the point of contact of the wheel on ground.

This, however, may be different in Krang where the area of contact of the wheels from the ground is more than just a single point. It is a rectangular region. So this initial value may be more relaxed for starting Krang from initial position.

ϕ_0 **(degrees)**	$x_{t\to\infty}$ **(meters)**
1.000	5.481
2.000	5.631
3.000	5.762
4.000	5.957
5.000	6.186
6.000	6.289
7.000	6.466
8.000	6.697
9.000	6.791
10.000	6.972
11.000	7.010
12.000	7.356
13.000	7.296
14.000	7.975
15.000	8.354
16.000	8.855
17.000	9.383
18.000	9.955
19.000	11.040
20.000	12.447
21.000	14.520

Table 1: The relationship of the initial state of ϕ and the amount of displacement of the wheel before stability

4.8 Effect of Modeling Errors

We now study the effect of modelling error on the control input u. We are interested in four particular parameters mass of the rod m, inertia of the rod I, distance l of the center of mass and finally and the estimation of the parameter ϕ. Even though ϕ is a state variable, we treat the error in the estimation of ϕ as a modeling error because it is a result of estimation of the coordinates of the center of mass of the robot based on the mass model of the robot.

4.8.1 Effect of modeling error in m

We find the effect of modeling error of mass m by choosing to find the system gains based on the incorrect masses that range from 20% to 180% of the actual mass of the system and see the effect of on the system response. The plots of the states and the control input are shown in Figur 4.8.1. We don't see a major change in the bahvior of the system except for when the mass is modelled too small as compared to the actual mass. In general smaller mass estimation leads to higher peaks, more overshoots, higher resultant displacements on the state variables and motor current and higher settling times. On the converse larger mass estimations are associated with opposite effects on the state variables. But the effect on the motor current u is the same i.e. higher peaks, overshoots and settling times. But, the differences in the behavior are very small and canbe neglected.

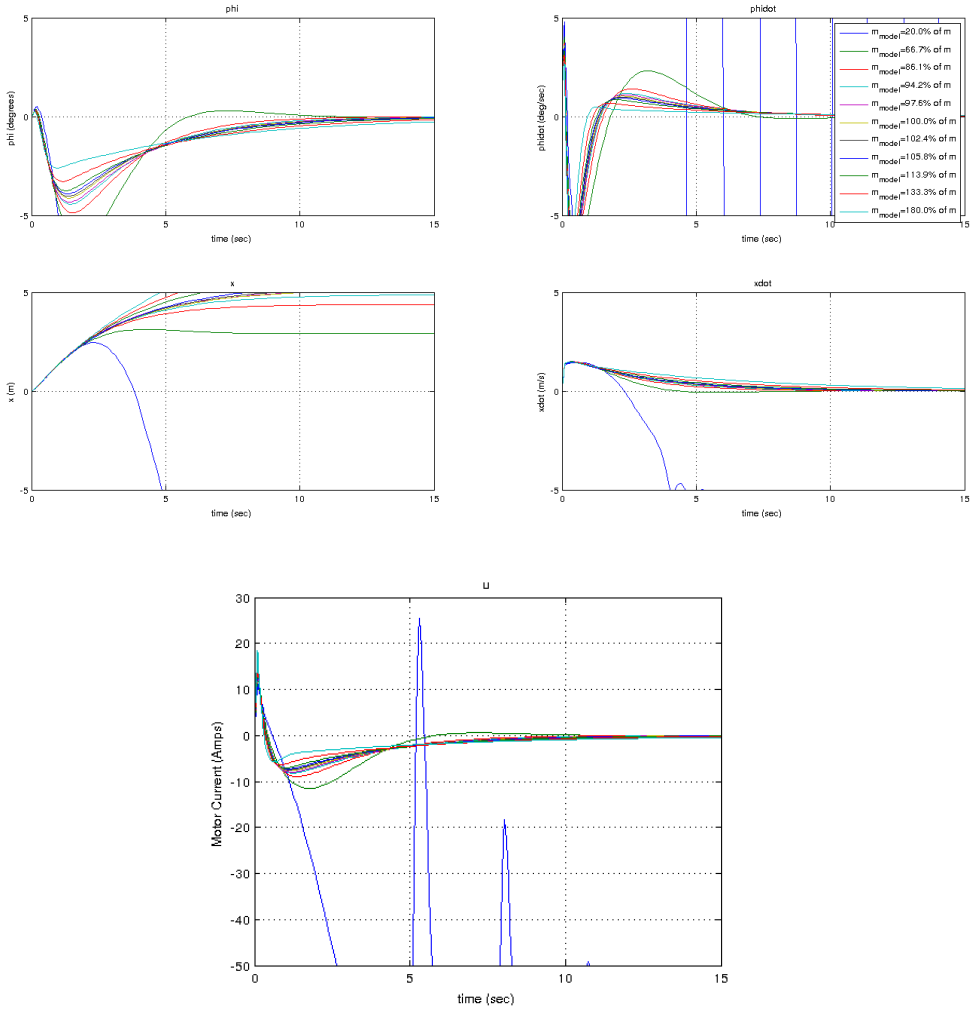

Figure 13: The behavior of the system when control gains are based on modeled mass m_{model} that ranges from 20% to 180% of the actual mass m

4.8.2 Effect of modeling error in I

We adopt the same methodology as adopted in the previous subsection. Control gains of the system are derived from modeled inertia parameter I_{model} that is not equal to the actual inertia I of the system. The response of the system for 10 different values of I_{model} ranging from 20% of the actual inertia to 180% is shown in Figure 14. The response of the system doesn't change much except when I_{model} is very small in which case we see oscillations in the state veriables as well as the control input. For the rest of the values, however, we see very little change in the response of the system.

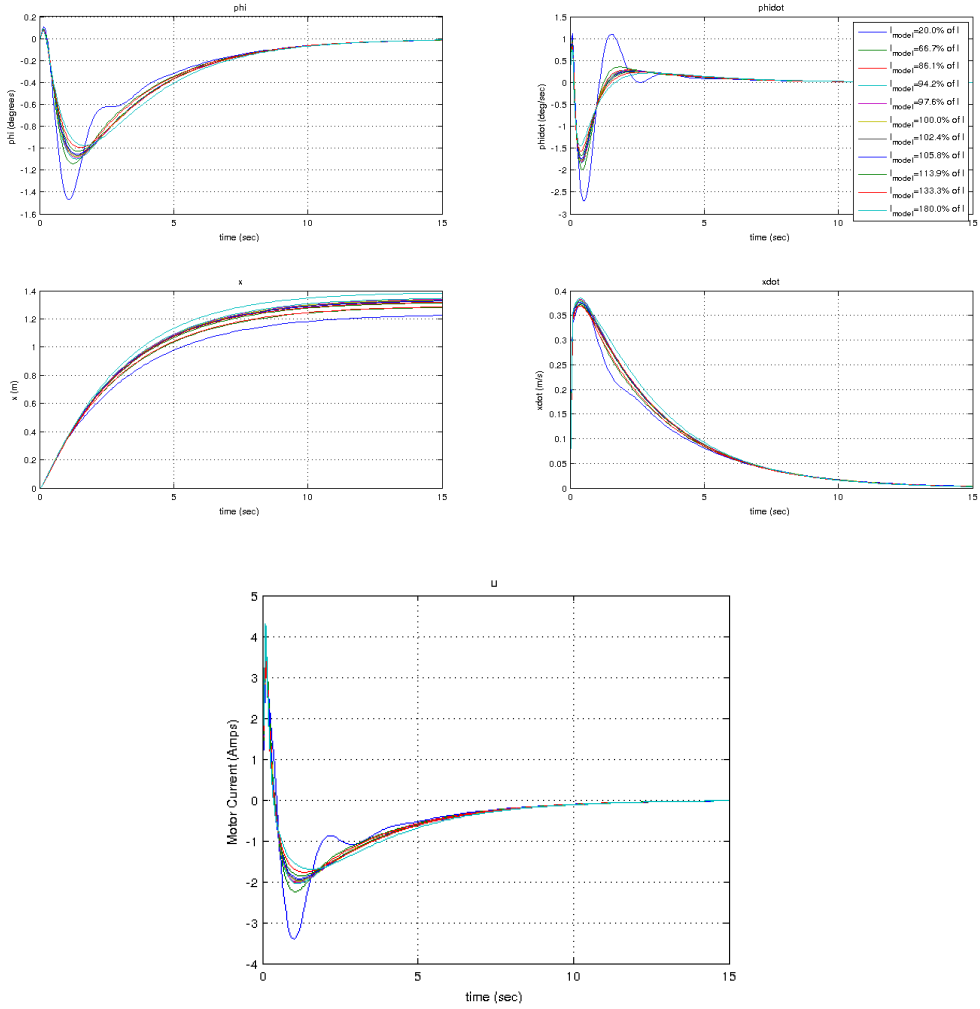

Figure 14: The behavior of the system when control gains are based on modeled mass I_{model} that ranges from 20% to 180% of the actual mass I

4.8.3 Effect of modeling error in l

We adopt the same methodology as adopted in the previous subsection. Control gains of the system are derived from modeled COM distance parameter l_{model} that is not equal to the actual COM distance l of the system. The response of the system for 10 different values of l_{model} ranging from 20% of the actual COM distance to 180% is shown in Figure 15. The response of the system doesn't change much except when l_{model} is very small, in which case, we see smaller peaks and overshoots in the state variables but at the cost of very high values of the motor current values. For the rest of the values, however, we see very little change in the response of the system.

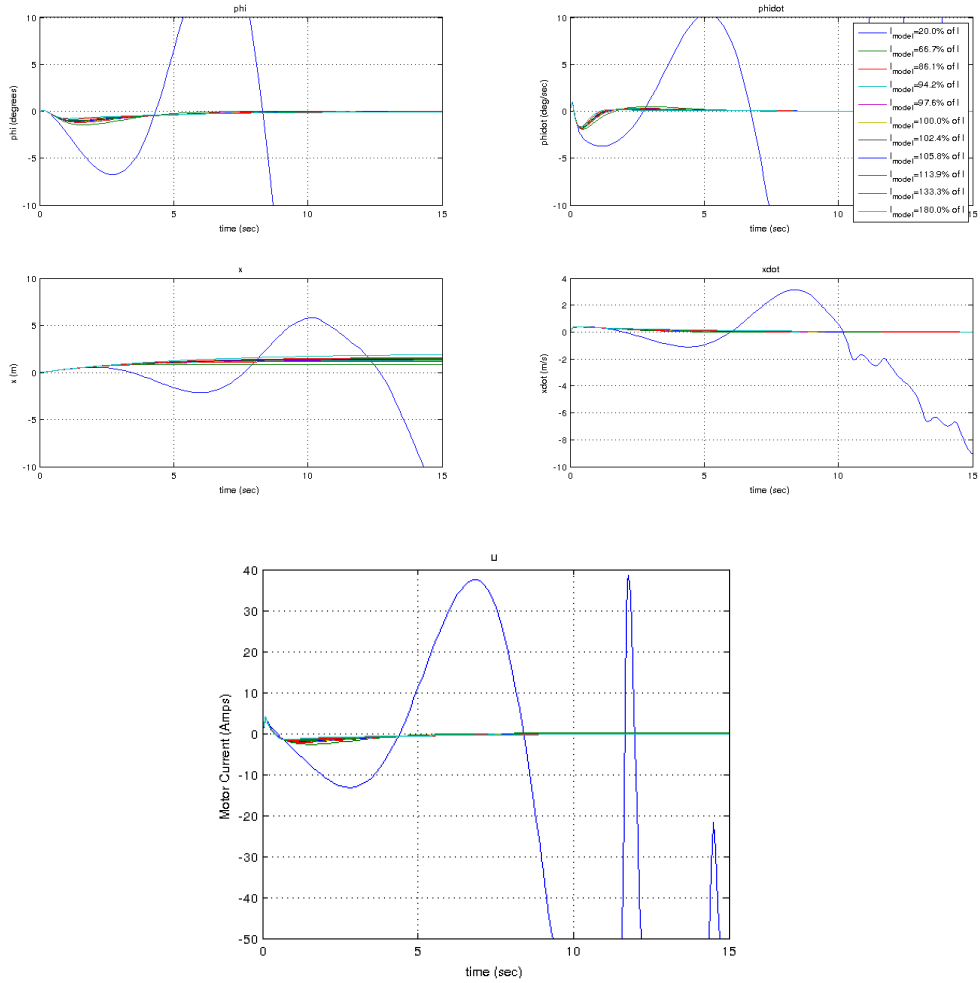

Figure 15: The behavior of the system when control gains are based on modeled mass l_{model} that ranges from 20% to 180% of the actual mass l

4.8.4 Effect of modeling error in Center Of Mass (resulting in error in ϕ)

Studying the effect of modeling error in ϕ is very important. In the real system of Krang, ϕ represents the angle from the vertical of the line connecting the estimated COM of Krang to the wheel axis. So ϕ is estimated in each iteration based on the current positions of all the joints and the masses and center of masses of all the links connecting the joints. Clearly, there are a number of variables on the accuracy of which depends the accuracy of the estimation of ϕ. And thus, the theoretical assumption that we have measured the state ϕ is just that — a theoretical assumption. In general, at best we get a $\hat{\phi}$ which is 0.5 degrees away from the system's actual ϕ. This $\hat{\phi}$ is what we will then use to calculate the state error for feedback control.

$$u = -K(\hat{x} - x_{ref}) \tag{26}$$

$$\hat{x} = \begin{bmatrix} \hat{\phi} \\ \dot{\hat{\phi}} \\ \hat{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} \phi \\ \dot{\phi} \\ x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} \tilde{\phi} \\ \dot{\tilde{\phi}} \\ \tilde{x} \\ \dot{\tilde{x}} \end{bmatrix} \tag{27}$$

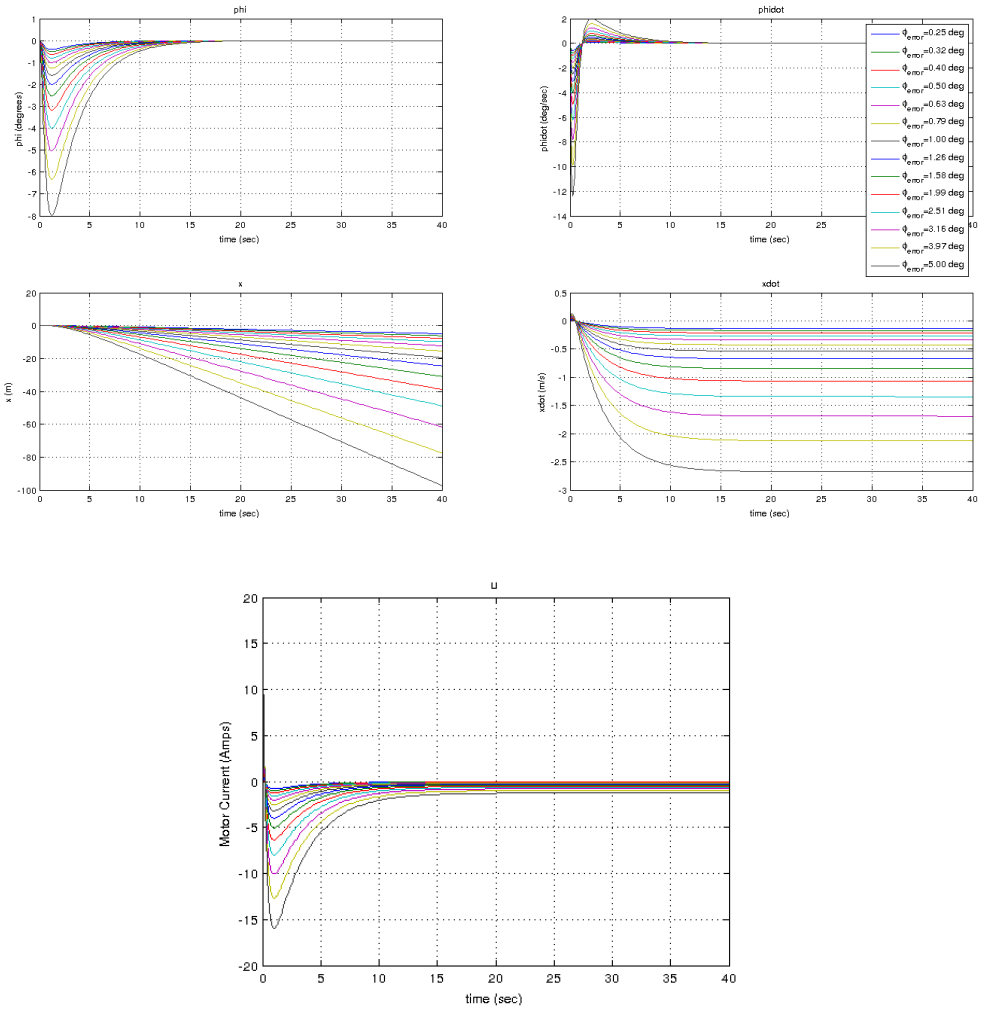

Figure 16: The behavior of the system that results from an incorrect estimation of the Center of Mass of the system resulting in an error in the angle ϕ. The responses are shown for the case when the system start from equilibrium position in the absence any external force but an incorrect estimate $\hat{\phi} = \phi + \tilde{\phi}$ with $\tilde{\phi}$ ranging from 0.5 to 5 degrees

For now, we assume $\dot{\tilde{\phi}} = \tilde{x} = \dot{\tilde{x}} = 0$ i.e. the error in measurement of all the states except the state ϕ zero.

The response of the system based on this error in the modeling is shown in Figure 16. We choose 15 different errors $\tilde{\phi}$ from 0 to 5 degrees. We don't apply any external impulsive force and start the at zero initial condition. We observe the following:

1. The system initially tries to bring $\hat{\phi}$ to zero by making ϕ equal to the negative of the error $\tilde{\phi}$.
2. Since this is an unstable position $\dot{\phi}$ increases because of the torques acting on the system due to the pendulum's weight. In order to bring that to zero motor current u shoots up and enevtually the system turns back to its actual equilibrium position $\phi = 0$.
3. Since $\phi = 0$ means $\hat{\phi} = \tilde{\phi}$, the components of u due to this error remains non-zero and is compensated by a steady-state non-zero $\dot{x}$, resulting in a small steady state u
4. The steady state values of $\dot{x}$ and u are proportional to the error $\tilde{\phi}$. By onservation $\dot{x}_{t=\infty} = -0.8\tilde{\phi}$ and $u_{t=\infty} = -1.01\tilde{\phi}$ where $\tilde{\phi}$ is in degrees.

If we introduce position control, the observation is shown in Figure 17. Here, a similar set of observations is made except that the error term of ϕ is compensated not by a steady state $\dot{x}$ but a steady state x and steady-state u is ten times smaller and only results due to a very small steady-state error in ϕ. So $\phi_{t=\infty} = -0.0076\tilde{\phi}$, $u_{t=\infty} = -0.02\tilde{\phi}$ and $x_{t=\infty} = -7\tilde{\phi}$ and when $\tilde{\phi}$ is in degrees.

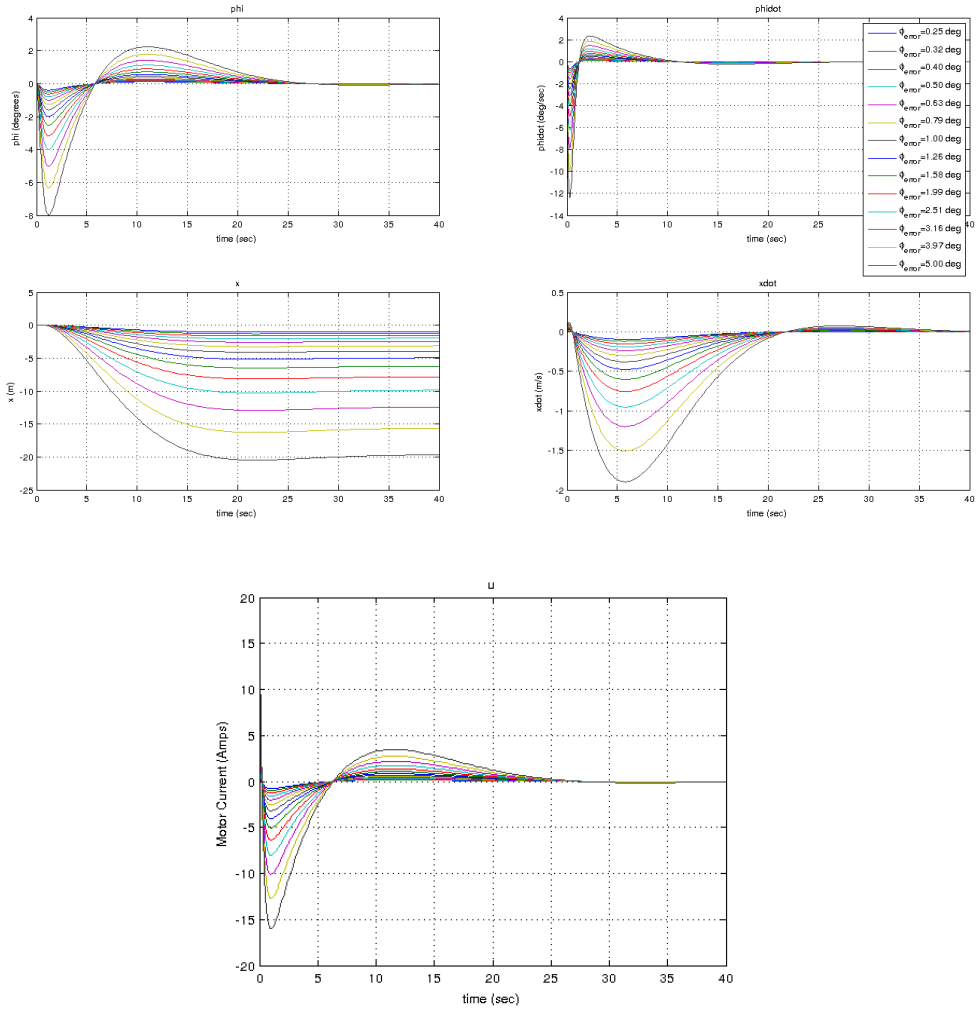

Figure 17: The behavior of the system with position control, that results from an incorrect estimation of the Center of Mass of the system resulting in an error in the angle ϕ. The responses are shown for the case when the system starts from equilibrium position in the absence any external force but an incorrect estimate $\hat{\phi} = \phi + \tilde{\phi}$ with $\tilde{\phi}$ ranging from 0.5 to 5 degrees.

4.9 Effect of Noise in State Measurement

When a noise of 2 degrees is introduced in the measurement of ϕ (see Figure 18), we see that the system stays stable. But in the absence of position control, we see that the act of stabilizing the system forces the position of the wheel x to keep changing. We also observe that noise in ϕ causes the control input u to change randomly. As a result the whole system undergoes random oscillations, as can be observed in the plots of $\dot{\phi}$ and $\dot{x}$ but the system stays stable.

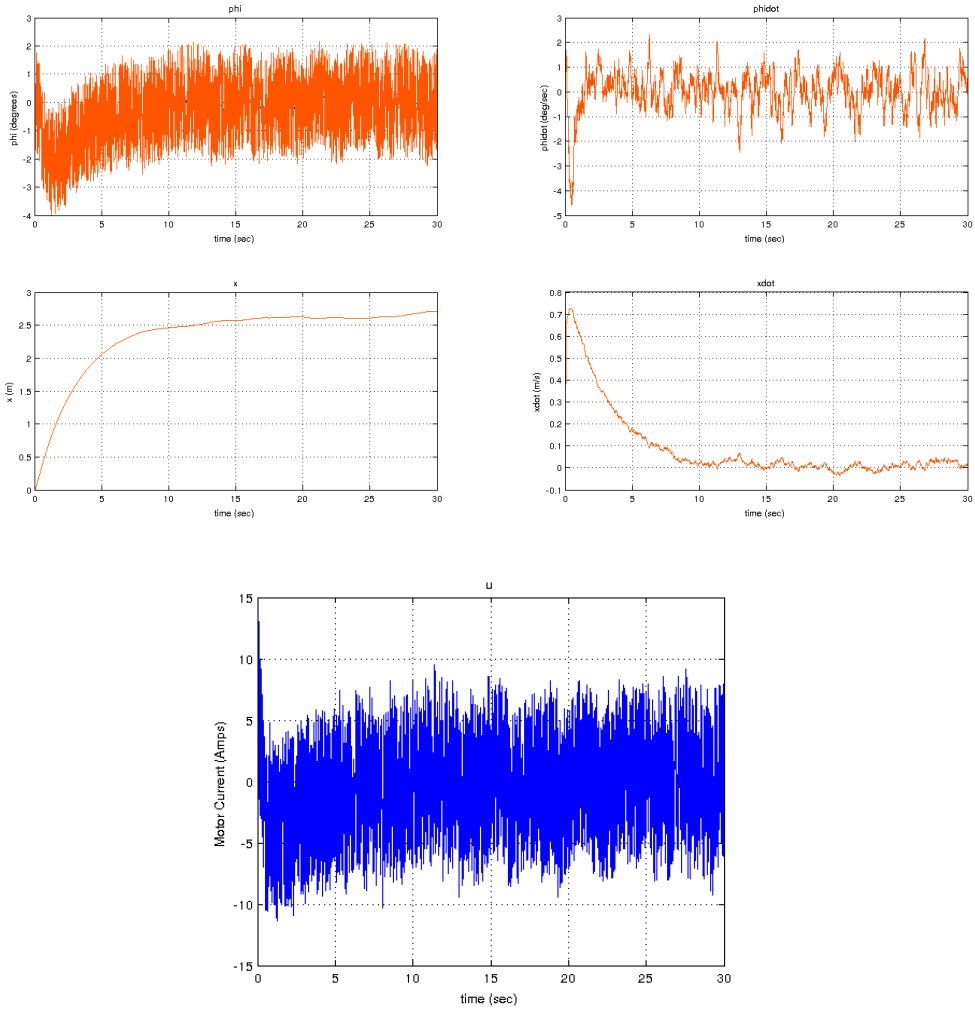

Figure 18: Response of the controlled system in the presence of noise in the measurement of ϕ

Increasing the simulation time further (from 30 sec to 60 sec) (Figure 19) reveals that x doesn't just keep moving in the positive direction but the movements in x are random in nature. And x pretty much oscillates around the zero position as time goes on. Increasing the noise to 5 degrees still keeps the system stable. It only leads to a larger wandering of x.

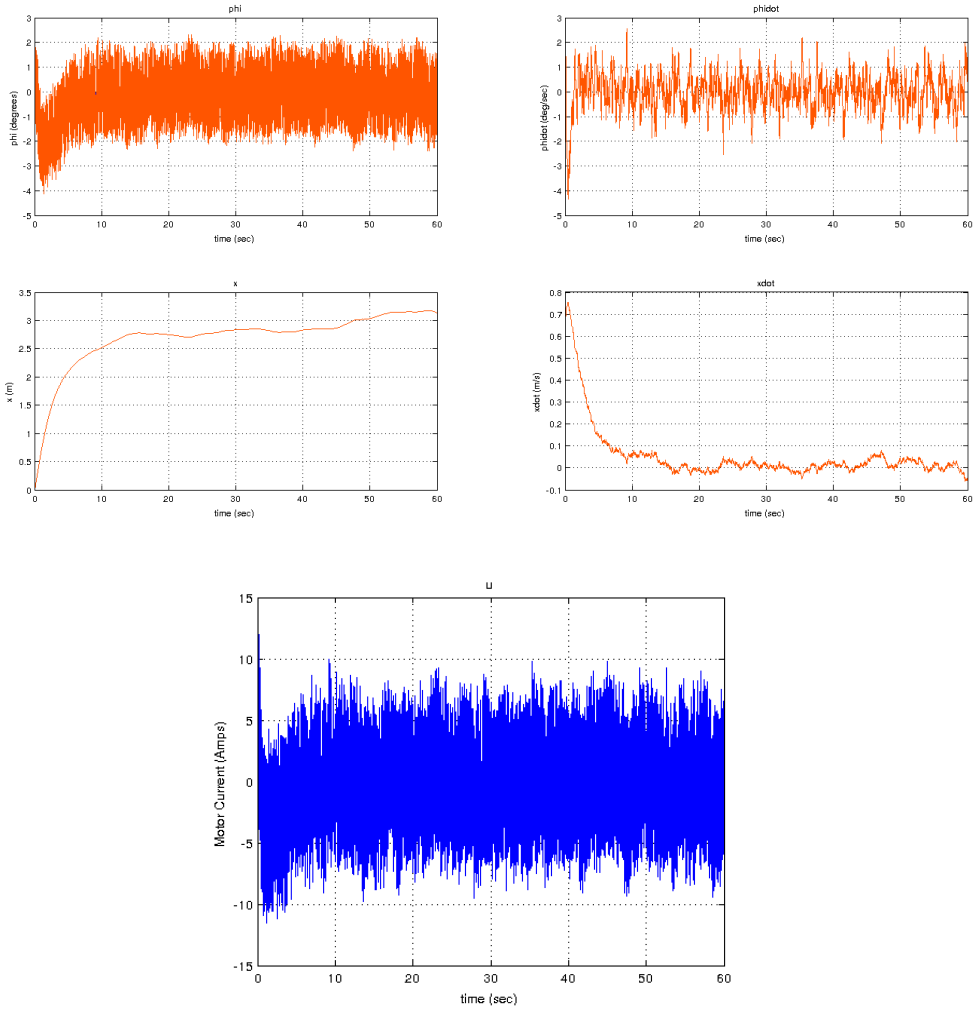

Figure 19: Response of the controlled system in the presence of noise in the measurement of ϕ with simulation time increased to 60 sec to see whether x stabilizes but it doesn't

Similar behaviors are observed even when force is increased to 3000N (Figure 20). Just that the initial displacement of x is large in the positive direction (i.e. the direction of the force) before the wandering behavior starts.

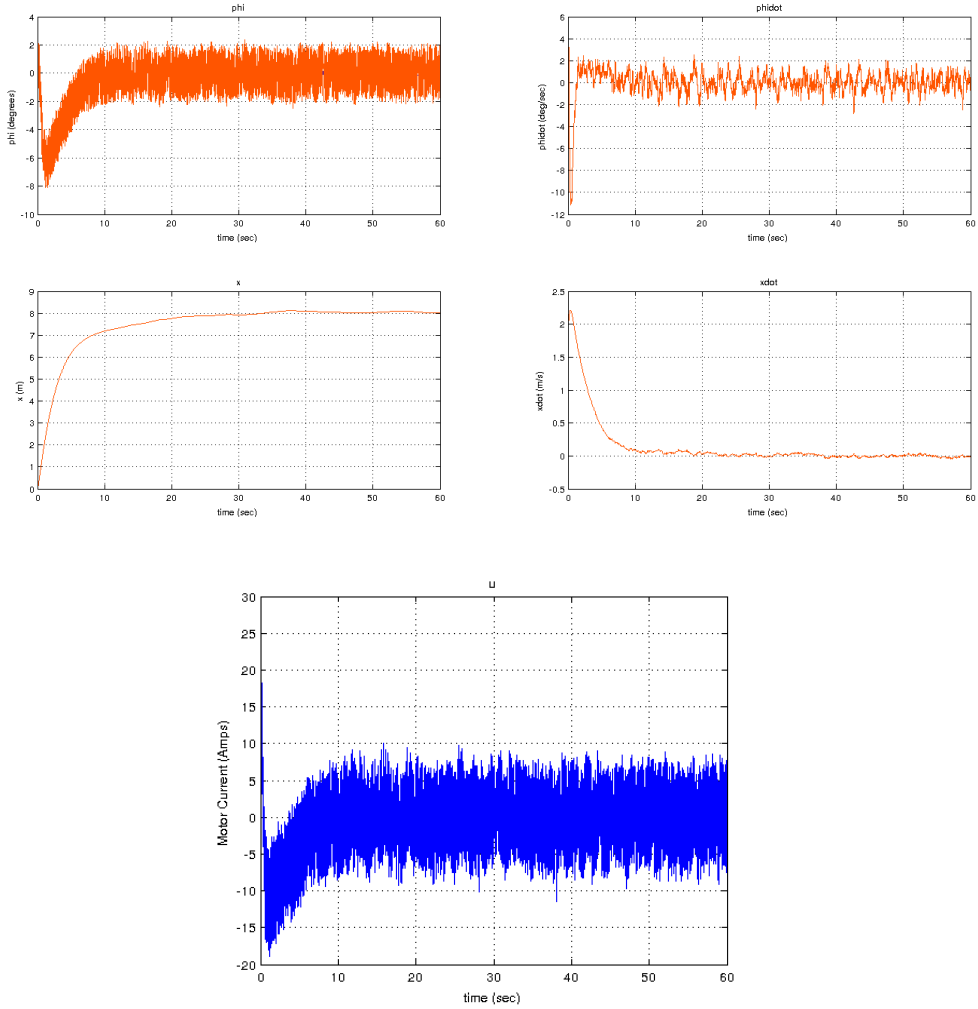

Figure 20: $F_{impulse}$ increased to 3000N to see the whether the system with measurement noise stays stable even in the presence of high impulsive force

Increasing the noise too high can lead the system to go unstable. (Figure 21) At noise level equaling 20 degrees, we see the that system blows up.

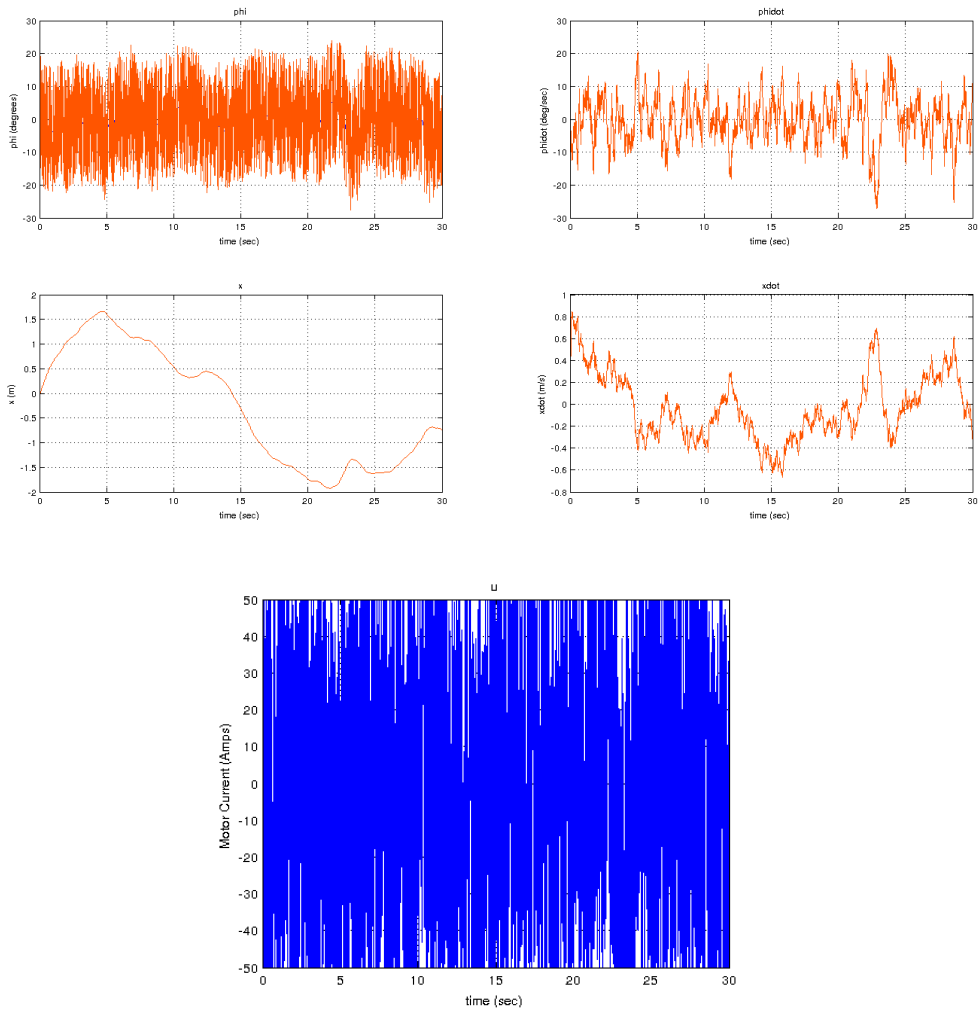

Figure 21: Response of the controlled system in the presence of high noise in the measurement of ϕ. The system blows up.

4.9.1 Effect of Filtering the Noise in State Measurement

We saw the effect of noise in the measurement of ϕ. We observed that it results in a wandering behavior in the position x and oscillations in other controlled variables as well as the control input u. In the real system (i.e. Krang) we have used Kalman filtering to filter out correct signal from the noisy one. For the purpose of this study, we modeled the discrete nature of the control, as is present in the real system and we set the sampling time $T_s = 2ms$. We used ϕ and $\dot{\phi}$ as the state veriables, as well as the output variables. The matrices used for the purpose of Kalman filter are as under:

$$A = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, Q = \begin{bmatrix} 1e-3 & 0 \\ 0 & 1e-3 \end{bmatrix} \tag{28}$$

where A, B and C define the state-space model for the measured quantities R is the observation noise covariance and Q is the process noise covariance.

We start the system with zero initial state and apply a 500N of impulsive force in the beginning. The reponse is shown in 22. We see that the disturbances in the state variables are much smaller. The wandering behavior of x is much more stabilized. And the disturbance in the motor current u is also minimized.

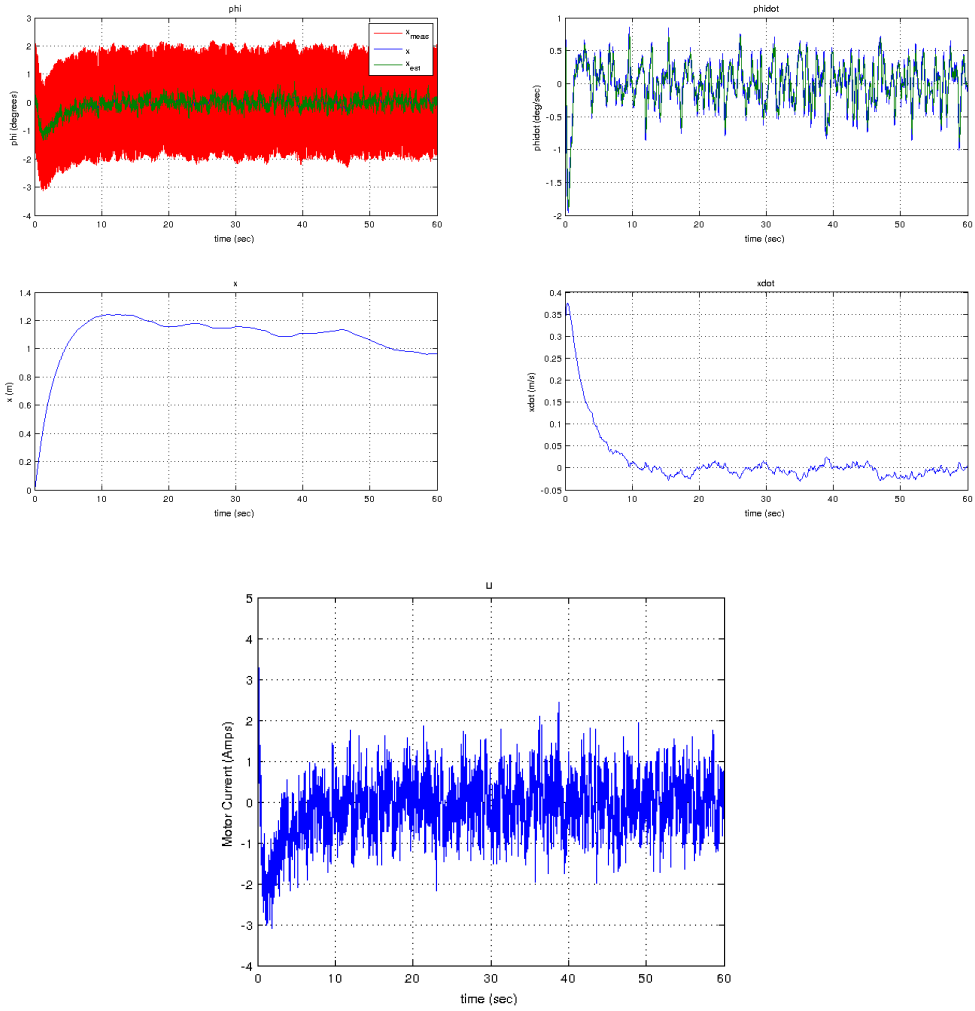

Figure 22: When the noise in the measurement in filtered, the response of the system is improved.

References

[1] Felix Grasser, Aldo D'Arrigo, Silvio Colombi, and Alfred C Rufer. Joe: a mobile, inverted pendulum. *Industrial Electronics, IEEE Transactions on*, 49(1):107–114, 2002.

[2] Yun-Su Ha and Shin'ichi Yuta. Trajectory tracking control for navigation of the inverse pendulum type self-contained mobile robot. *Robotics and autonomous systems*, 17(1):65–80, 1996.

[3] Alessio Salerno. Quasimoro—a telerobot for the augmentation of wheelchair users. In *American Society of Mechanical Engineers*. ASME, 2008.

[4] Alessio Salerno and Jorge Angeles. On the nonlinear controllability of a quasiholonomic mobile robot. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 3, pages 3379–3384. IEEE, 2003.

[5] Alessio Salerno, Svetlana Ostrovskaya, and Jorge Angeles. The development of quasiholonomic wheeled robots. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 4, pages 3514–3520. IEEE, 2002.