Nonlinear Controller Design for Stabilizing an Inverted Pendulum Cart System on an Unknown Inclination

David Kun[a,*]

[a] *Neil Armstrong Hall of Engineering Rm. 3200A, 701 W. Stadium Ave., West Lafayette, IN 47907*

Abstract

The inverted pendulum cart system has attracted much research focus from the Control society. It is a seemingly simple system with complex nonlinearities, analogous to many industrial applications. This paper begins by examining the effects of an uncertain rail slope and unknown Coulomb friction force on a simple linear controller's performance. Subsequently, a second-order sliding mode controller is designed in order to mitigate the effects of these uncertain disturbances. Finally, the performance of the two controllers is compared in various simulations, which are constructed in Matlab's Simulink.

Keywords: Sliding Mode Control, Nonlinear Control, Backstepping Design, Pendulum Cart System

1. Introduction

The inverted pendulum cart (IPC) system is a well known nonlinear under-actuated system that has been widely used to test control strategies. Various approaches have been taken to stabilize the IPC, including PID and LQR stabilization [7], swing-up and stabilization control using neural networks [1], backstepping design [2], robust adaptive control [4], sliding-mode control [8], and many others. Further work has been done to stabilize a double inverted pendulum [5] and even a triple inverted pendulum [6]. However, published papers addressing IPC control in the presence of an unknown rail slope are either scarce or non-existent.

First, this paper demonstrates that a small rail inclination does not have a significant impact on a simple controller's performance. However, as the rail inclination is increased, performance decreases rapidly. If the cart's Coulomb friction force is also included in simulations, it exacerbates the controller response and convergence.

Second, a sliding mode controller is designed to overcome the uncertainties in rail angle and Coulomb friction. This nonlinear controller shows a significant improvement in overall performance, including mitigation of oscillatory response due to friction, which is apparent in the linear controller.

Finally, this paper concludes with a summary of the results from each controller, and a comparison of their performance. Suggestions for future research work are provided at the end.

2. Model Description

Figure 1, below, presents a simple diagram of the inverted pendulum cart system used in this paper. The nomenclature is explained on the following page.

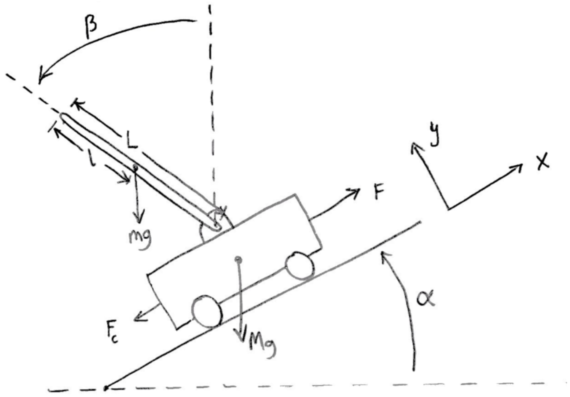

Figure 1: Free Body Diagram

[*] Corresponding author
Email address: dkun@purdue.edu (David Kun)
URL: https://engineering.purdue.edu/HSL/ (David Kun)

Preprint submitted to ME 677, Prof. Bin Yao *January 13, 2014*

Nomenclature:

β **:** Pendulum angle from upward vertical
α **:** Rail angle from horizontal
x **:** Cart position on rail
M **:** Cart mass
m **:** Pendulum mass
M_{eq} **:** Total equivalent inertia
L **:** Rod length
l **:** Rod half-length
I **:** Rod rotational inertia
b_r **:** Rod joint damping coefficient
b_{eq} **:** Total equivalent damping
F_c **:** Coulomb friction from track
μ_c **:** Coulomb friction coefficient
F **:** Force on cart
K **:** Input-to-force gain
g **:** Gravitational acceleration

Define the total translational and rotational inertia as:

$$M_t = M_{eq} + m$$
$$I_t = I + ml^2$$

respectively, Using Lagrangian dynamics, a theoretical model of the pendulum cart system can be derived (similar to the derivation in [9]). The resulting system dynamics can be written as follows:

$$\mathcal{M}\ddot{q} + \mathcal{B}\dot{q} + \mathcal{C} = \mathcal{K}u - \mathcal{F}, \qquad q = \begin{bmatrix} x \\ \beta \end{bmatrix} \tag{1}$$

where,

$$\mathcal{M} = \begin{bmatrix} M_t & -ml\cos(\beta-\alpha) \\ -ml\cos(\beta-\alpha) & I_t \end{bmatrix}$$
$$\mathcal{B} = \begin{bmatrix} b_{eq} & ml\sin(\beta-\alpha)\dot{\beta} \\ 0 & b_r \end{bmatrix}$$
$$\mathcal{C} = \begin{bmatrix} M_t g\sin(\alpha) \\ -mglsin(\beta) \end{bmatrix}, \mathcal{K} = \begin{bmatrix} K \\ 0 \end{bmatrix}, \mathcal{F} = \begin{bmatrix} F_c \\ 0 \end{bmatrix}$$

Here we define K and u as the gain and input, respectively, and Ku is regarded as the control force F in Figure 1. The Coulomb friction force $(F_c(\dot{x}))$ is modeled as shown in Figure 2, when used in Matlab and Simulink simulations.

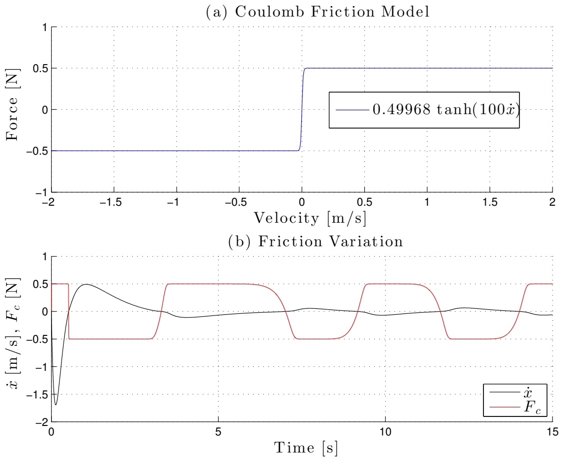

Figure 2: (a) Friction, F_c, Model and (b) Example

3. LQR Controller

To begin, we derive a simple LQR control algorithm to stabilize the pendulum cart system. In *this* design process, we assume that the Coulomb friction on the cart is negligible, i.e. $F_c = 0$, and the rail inclination is at a nominal value of zero, i.e. $\alpha = 0$. With these assumptions, we can solve Equation (1) for the second derivatives, which yields:

$$\begin{aligned} \ddot{q} &= \mathcal{M}^{-1}\left\{-\mathcal{B}\dot{q} - \mathcal{C} + \mathcal{K}u\right\} \\ &= \frac{1}{\gamma}\begin{bmatrix} I_t & ml\cos(\beta-\alpha) \\ ml\cos(\beta-\alpha) & M_t \end{bmatrix} \times \\ &\left\{\begin{bmatrix} b_{eq} & ml\sin(\beta-\alpha)\dot{\beta} \\ 0 & b_r \end{bmatrix}\dot{q} - \begin{bmatrix} M_t g\sin(\alpha) + Ku \\ -mglsin(\beta) \end{bmatrix}\right\} \end{aligned} \tag{2}$$

where,

$$\gamma = M_t I_t - (ml\cos(\beta-\alpha))^2 \tag{3}$$

Substituting $\alpha = 0$ and linearizing the equations yields the following linear system:

$$\dot{X}_{lqr} = A_{lqr}X_{lqr} + B_{lqr}u$$
$$A_{lqr} = \frac{1}{\gamma_1}\begin{bmatrix} 0 & 0 & \gamma_1 & 0 \\ 0 & 0 & 0 & \gamma_1 \\ 0 & m^2l^2g & -I_t b_{eq} & -mlb_r \\ 0 & M_t mgl & -mlb_{eq} & -M_t b_r \end{bmatrix}$$
$$B_{lqr} = \frac{1}{\gamma_1}\begin{bmatrix} 0 \\ 0 \\ I_t K \\ mlK \end{bmatrix}, X_{lqr} = \begin{bmatrix} x \\ \beta \\ \dot{x} \\ \dot{\beta} \end{bmatrix}, \gamma_1 = M_t I_t - m^2 l^2$$

As a further improvement to the LQR controller, we can augment the system with an integral state, $\dot{\lambda} = x_d - x$, so when $\dot{\lambda} \to 0$, $x \to x_d$. Since the focus of this work is stabilization of the pendulum from initial perturbation, we can set $x_d = 0$, without loss of generality. Then the new augmented system is given by:

$$A_a = \begin{bmatrix} & & & & 0 \\ & & & & 0 \\ & A_{lqr} & & & 0 \\ & & & & 0 \\ -1 & 0 & 0 & 0 & 0 \end{bmatrix}, B_a = \begin{bmatrix} B_{lqr} \\ 0 \end{bmatrix}$$

$$X_a = \begin{bmatrix} X_{lqr} \\ \int (x_d - x)dt \end{bmatrix}, \dot{X}_a = A_a X_a + B_a u$$

In this controller, the input u is given by:

$$u = -K_1 X_a$$

where the gains $K_1 = [k_1, k_2, k_3, k_4, k_5]^T$ are chosen by using the LQR method, which minimizes the cost function:

$$\min\{\int_0^\infty X_a^* Q X_a + Ru^2 \text{ d}t\}$$

subject to $u = -K_1 X_a$. The weight matrices are chosen to be:

$$Q = \text{diag}([1, 10, 0, 0, 1]), \qquad R = 0.1$$

to emphasize the control effort on the pendulum angle, with some (smaller) emphasis on the cart position and the integral of cart position. The resulting gains are given by:

$$K_1 = \begin{bmatrix} -8.49, 48.46, -15.38, 9.67, 3.16 \end{bmatrix}$$

This concludes the LQR controller design.

The following parameters (Table 1) were used in all simulations presented in this paper:

The following figures show the performance of the LQR controller with different unknown rail inclinations. The simulations were run using Matlab's Simulink and S-functions (see Appendix A). In the first two plots (Figure 3), there was no Coulomb friction present. In the next plots (Figure 4), the Coulomb friction is included. The initial offset of the pendulum is 45 degrees, i.e. $X_a(0) = [0, 45^o, 0, 0, 0]^T$. In both cases, the LQR controller manages to stabilize the pendulum quite quickly (although with a large overshoot in both pendulum angle and cart position). However, when Coulomb friction is included in the simulations, the LQR controller is not able to converge to the zero positions (of cart and pendulum). Instead, it oscillates about those points.

Table 1: Simulation Parameters

Parameter	Value
M_{eq}	$1.0424\ kg$
m	$0.231\ kg$
M_t	$1.2734\ kg$
I	$0.0079\ kg \cdot m^2$
I_t	$0.03155\ kg \cdot m^2$
L	$0.64\ m$
l	$0.32\ m$
b_r	$0.00014\ kg/s$
b_{eq}	$9.582\ kg/s$
μ_c	0.04
g	$9.81\ m/s^2$
K	1.7189

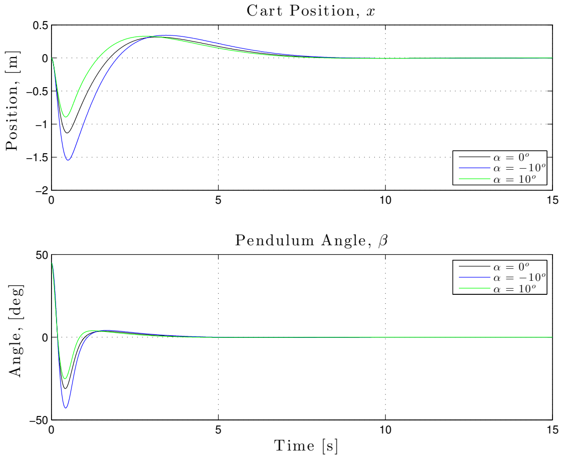

Figure 3: LQR Controller, No Coulomb Friction

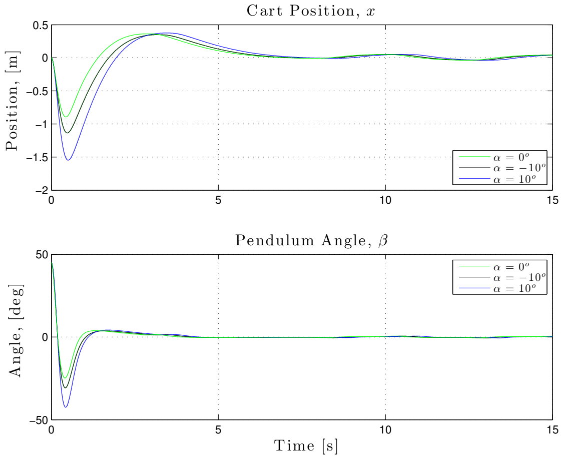

Figure 4: LQR Controller, with Coulomb Friction

These results motivate the design of a better-performing nonlinear controller. The approach used in this paper is similar to a sliding mode controller (SMC) design in [8]. The method and implementation are explained in the following section.

4. Sliding Mode Controller

The inverted pendulum cart system dynamics from Equation (1) can be placed in the following form:

$$\ddot{q} = \frac{1}{\gamma}\begin{bmatrix} I_t & ml\cos(\beta-\alpha) \\ ml\cos(\beta-\alpha) & M_t \end{bmatrix} \times \left\{ \begin{bmatrix} b_{eq} & ml\sin(\beta-\alpha)\dot{\beta} \\ 0 & b_r \end{bmatrix} \dot{q} - \begin{bmatrix} M_t g\sin(\alpha) + Ku - F_c(\dot{x}) \\ -mglsin(\beta) \end{bmatrix} \right\} \tag{4}$$

Unlike the previous section, we are including the Coulomb friction term,

$$F_c = \mu_c M_t g \ \mathrm{sign}(\dot{x}) \approx 0.4997 \ \mathrm{sign}(\dot{x})$$

where $\mu_c = 0.04$ was obtained from [3] and is shown in Table 1. The rail inclination angle (α) is unknown, so if we assume that the inertia matrix $\mathcal{M}$ does not vary much, i.e.

$$\begin{aligned} ml\cos(\beta-\alpha) &\approx ml\cos(\beta) \\ ml\sin(\beta-\alpha) &\approx ml\sin(\beta) \end{aligned}$$

we can compensate for this additional uncertainty in the controller design while greatly simplifying the derivation of the controller. Note that the major source of uncertainty in the dynamics comes from the gravitational effect on the IPC ($M_t g\sin(\alpha)$) and the friction on the cart (F_c). For brevity, we will denote:

$$\begin{aligned} \cos(\theta) &\triangleq C_\theta \\ \sin(\theta) &\triangleq S_\theta \end{aligned}$$

Then the resulting equations are given by:

$$\begin{bmatrix} \ddot{x} \\ \ddot{\beta} \end{bmatrix} = -\frac{1}{\gamma}\begin{bmatrix} I_t b_{eq} & I_t m l S_\beta \dot{\beta} + b_r m l C_\beta \\ b_{eq} m l C_\beta & M_t b_r + m^2 l^2 C_\beta S_\beta \dot{\beta} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{\beta} \end{bmatrix} + \frac{1}{\gamma}\begin{bmatrix} m^2 l^2 g S_\beta C_\beta - I_t M_t g S_\alpha - I_t F_c \\ M_t m g l S_\beta - Mtmgl S_\alpha C_\beta - F_c m l C_\beta \end{bmatrix} + \frac{1}{\gamma}\begin{bmatrix} I_t K \\ m l C_\beta K \end{bmatrix} u \tag{5}$$

where,

$$\gamma = M_t I_t - m l C_\beta$$

In order to continue with the controller design, the system needs to be re-written in a regular form,

$$\begin{aligned} \ddot{\eta} &= g(\eta, \dot{\eta}, \xi, \dot{\xi}) \\ \ddot{\xi} &= f(\eta, \dot{\eta}, \xi, \dot{\xi}) + u \end{aligned} \tag{6}$$

by implementing a diffeomorphic state space transformation. We begin by defining:

$$\eta = x - \frac{I_t}{ml}\ln\left(\frac{1+\sin(\beta)}{\cos(\beta)}\right), \qquad |\beta| < \frac{\pi}{2} \tag{7}$$

Taking the first and second derivatives results in:

$$\begin{aligned} \dot{\eta} &= \dot{x} - \frac{I_t}{mlC_\beta}\dot{\beta} \\ \ddot{\eta} &= \ddot{x} - \frac{I_t}{mlC_\beta}\tan(\beta)\dot{\beta}^2 - \frac{I_t}{mlC_\beta}\ddot{\beta} \end{aligned} \tag{8}$$

Substituting $\ddot{x}, \ddot{\beta}$ from Equation (5) into Equation (8) and simplifying the results yields:

$$\ddot{\eta} = -\left(g + \frac{I_t\dot{\beta}^2}{mlC_\beta}\right)\tan(\beta) + \frac{b_r\dot{\beta}}{mlC_\beta} \tag{9}$$

Then the system can be written in normal form as:

$$\begin{cases} \dot{\eta} &= \dot{\eta} \\ \ddot{\eta} &= -\left(g + \frac{I_t\dot{\beta}^2}{mlC_\beta}\right)\tan(\beta) + \frac{b_r\dot{\beta}}{mlC_\beta} \\ \dot{\beta} &= \dot{\beta} \\ \ddot{\beta} &= \frac{1}{\gamma}\Big(M_t m g l S_\beta - m^2 l^2 S_\beta C_\beta \dot{\beta}^2 - M_t b_r \dot{\beta} \\ & -b_{eq} m l C_\beta \dot{x} + m l C_\beta K u \\ & - F_c m l C_\beta - M_t m g l S_\alpha C_\beta\Big) \end{cases} \tag{10}$$

Note that the last two terms in Equation (10), i.e. $\Delta_0 = (-F_c m l C_\beta - M_t m g l S_\alpha C_\beta)/\gamma$, represent unmatched terms in the system, and the controller will be designed to attenuate their effects.

We design a fictitious output (ξ) that will keep the system ($\eta, \dot{\eta}$) minimum phase with respect to the output when $b_r\dot{\beta} = 0$. Let,

$$\xi = \tan(\beta) - \lambda_1\eta - \lambda_2\dot{\eta} \tag{11}$$

for $\lambda_1, \lambda_2 > 0$.

From Equations (9), (10), and (11), we see that as $\eta, \dot{\eta}, \xi \to 0$, both β and x also converge to zero. Therefore, we can design a sliding motion on $\xi = 0$

in order to stabilize x and β under the minimum phase assumption. We begin by taking the first and second derivatives of ξ, was follows:

$$\begin{aligned}\dot{\xi} &= \sec^2(\beta)\dot{\beta} - \lambda_1\dot{\eta} - \lambda_2\ddot{\eta}\\ &= \sec^2(\beta)\dot{\beta} - \lambda_1\dot{\eta} +\\ &\quad \lambda_2\left(g\tan(\beta) + \frac{I_t\dot{\beta}}{mlC_\beta}\tan(\beta) - \frac{b_r\dot{\beta}}{mlC_\beta}\right)\end{aligned}$$

The second derivative is more lengthy,

$$\begin{aligned}\ddot{\xi} &= 2\tan(\beta)\sec^2(\beta)\dot{\beta}^2 + \sec^2(\beta)\ddot{\beta} - \lambda_1\ddot{\eta} +\\ &\quad \lambda_2\Bigg(g\sec^2(\beta)\dot{\beta} + \frac{2I_t}{mlC_\beta}\tan(\beta)\dot{\beta}\ddot{\beta} +\\ &\quad \frac{I_t}{mlC_\beta}\tan^2(\beta)\dot{\beta}^3 + \frac{I_t}{mlC_\beta}\sec^2(\beta)\dot{\beta}^3 -\\ &\quad \frac{b_r}{mlC_\beta}\ddot{\beta} - \frac{b_r}{mlC_\beta}\tan(\beta)\dot{\beta}^2\Bigg)\end{aligned}$$

Before substituting $\ddot{\beta}$ into the above equation, let us write:

$$\ddot{\beta} = \beta_1 + \beta_2 u + \beta_3$$

where,

$$\begin{aligned}\beta_1 &= \frac{1}{\gamma}\Big(M_t mglS_\beta - m^2l^2S_\beta C_\beta\dot{\beta}^2\\ &\qquad -M_t b_r\dot{\beta} - b_{eq}mlC_\beta\dot{x}\Big)\\ \beta_2 &= \frac{1}{\gamma}\left(mlC_\beta K\right)\\ \beta_3 &= \frac{1}{\gamma}\left(-F_c mlC_\beta - M_t mglS_\alpha C_\beta\right)\end{aligned}$$

Note that β_1 is known, β_2 is the input gain, and β_3 represents uncertainties. Also define:

$$c_1 = \frac{1}{mlC_\beta}\left(2I_t\tan(\beta)\dot{\beta} - b_r\right)$$

Therefore,

$$\begin{aligned}\ddot{\xi} &= 2\tan(\beta)\sec^2(\beta)\dot{\beta}^2 -\\ &\quad \lambda_1\left(\frac{b_r\dot{\beta}}{mlC_\beta} - g\tan(\beta) - \frac{I_t\dot{\beta}^2}{mlC_\beta}\tan(\beta)\right) +\\ &\quad \sec^2(\beta)\beta_1 + \sec^2(\beta)\beta_2 u + \sec^2(\beta)\beta_3 +\\ &\quad \lambda_2 c_1\beta_1 + \lambda_2 g\sec^2(\beta)\dot{\beta} +\\ &\quad \frac{\lambda_2}{mlC_\beta}\Big(I_t\tan^2(\beta)\dot{\beta}^3 + I_t\sec^2(\beta)\dot{\beta}^3 -\\ &\qquad b_r\tan(\beta)\dot{\beta}^2\Big) + \lambda_2 c_1\beta_2 u + \lambda_2 c_1\beta_3\end{aligned}$$

We can now group together the known terms, input gain, and uncertainties in the following form:

$$\ddot{\xi} = \mu + z + \tau \tag{12}$$

where μ is known, z is uncertain, and τ is the input gain. Using Equation (12), we have the following definitions:

$$\begin{aligned}\mu &= 2\tan(\beta)\sec^2(\beta)\dot{\beta}^2 -\\ &\quad \lambda_1\left(\frac{b_r\dot{\beta}}{mlC_\beta} - g\tan(\beta) - \frac{I_t\dot{\beta}^2}{mlC_\beta}\tan(\beta)\right) +\\ &\quad \left(\sec^2(\beta) + \lambda_2 c_1\right)\beta_1 + \lambda_2 g\sec^2(\beta)\dot{\beta} +\\ &\quad \frac{\lambda_2}{mlC_\beta}\Big(I_t\tan^2(\beta)\dot{\beta}^3 + I_t\sec^2(\beta)\dot{\beta}^3\\ &\qquad -b_r\tan(\beta)\dot{\beta}^2\Big)\\ z &= \left(\sec^2(\beta) + \lambda_2 c_1\right)\beta_3\\ \tau &= \left(\sec^2(\beta) + \lambda_2 c_1\right)\beta_2 u\end{aligned}$$

From the fictitious output in Equation (11), we can rearrange:

$$\tan(\beta) = \xi + \lambda_1\eta + \lambda_2\dot{\eta}$$

and combine with Equation (9) to obtain:

$$\ddot{\eta} = -\left(g + \frac{I_t\dot{\beta}^2}{mlC_\beta}\right)\left(\xi + \lambda_1\eta + \lambda_2\dot{\eta}\right) + \frac{b_r\dot{\beta}}{mlC_\beta} \tag{13}$$

The system described by Equation (12) and Equation (13) is in a form similar to Equation (6) when $b_r\dot{\beta} = 0$. Looking at Equation (12), we would like to design:

$$\tau = -\mu - \Gamma_1\text{sign}(\xi) - \Gamma_2\text{sign}(\dot{\xi}) - \Gamma_3\xi - \Gamma_4\dot{\xi}$$

Then we must choose:

$$u = \frac{-\mu - \Gamma_1\text{sign}(\xi) - \Gamma_2\text{sign}(\dot{\xi}) - \Gamma_3\xi - \Gamma_4\dot{\xi}}{\left(\sec^2(\beta) + \lambda_2 c_1\right)\beta_2} \tag{14}$$

As is proved in Section 2 of [8], for robust stability, we must have:

$$\Delta < \Gamma_2 < \Gamma_1 - \Delta \tag{15}$$

$$\Gamma_3, \Gamma_4 \geq 0 \tag{16}$$

where Δ is the upper bound of the matched disturbance, z, i.e. $|z| < \Delta$ for almost all $t, \beta, \dot{\beta}$. Then the quasi-homogeneous controller (14) ensures finite time stability of the $(\xi, \dot{\xi})$ system.

It is shown in [8], that when $b_r\dot{\beta} \neq 0$, we lose the asymptotic stability property. However, by tuning λ_1 and λ_2, we can decrease the radius of the ball to which the system converges, and the system is proved to be input-to-state practically stable.

5. Results

This section presents some simulation results to compare the performance of the sliding mode controller (SMC) and linear quadratic controller (LQR). The tuned parameters $(\lambda 1, \lambda_2)$ were chosen by an iterative procedure based on the system's response to perturbations, and are defined as follows:

$$\lambda_1 = 0.1, \qquad \lambda_2 = 0.2$$

The maximum disturbance was calculated from $|z|$ in Equation (12) with the following maximum expected values:

Table 2: Maximum Disturbance Calculation

Parameter	Symbol	Value
Rail inclin.	α_{max}	30^o
Pend. angle	β_{max}	50^o
Pend. angular rate	$\dot{\beta}_{max}$	1 rad/s
Friction	$F_{c,max}$	0.5

Then,

$$\begin{aligned}
&\Gamma_2 > \Delta = \left|\left(\sec^2(\beta_{max}) + \lambda_2 c_1\right)\beta_3\right| \\
&\Gamma_2 > 23.1\ , \\
&\quad \text{and} \\
&\Gamma_1 > \Gamma_2 + \Delta = \Gamma_2 + 23.1\ , \\
&\quad \text{and} \\
&\Gamma_3, \Gamma_4 \geq 0
\end{aligned}$$

In these simulations, the chosen values are:

$$\begin{aligned}
\Gamma_1 &= 47.0 \\
\Gamma_2 &= 23.4 \\
\Gamma_3 &= 30.0 \\
\Gamma_4 &= \ \ 5.0
\end{aligned}$$

Figure 5 ($\beta(0) = 25^o$) and Figure 6 ($\beta(0) = 45^o$) show a comparison of the LQR and SMC controllers' performance with 0 degrees rail inclination and with Coulomb friction present. In these simulation conditions, the SMC demonstrates a superior performance as it stabilizes the pendulum with the same pendulum angle overshoot, smaller cart overshoot, and much faster convergence of both pendulum angle and cart position.

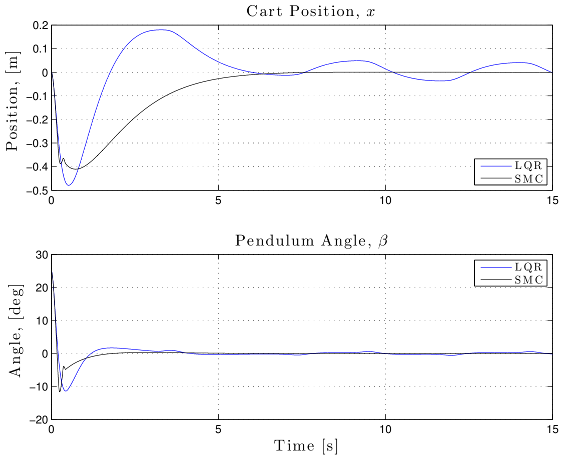

Figure 5: Rail Inclination $\alpha = 0^o$, Initial Pendulum Angle $\beta(0) = 25^o$

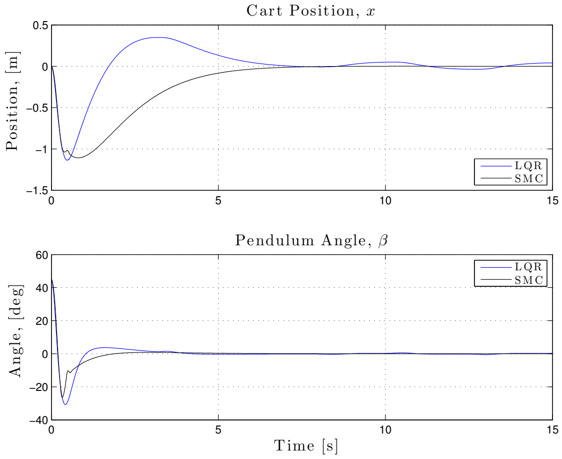

Figure 6: Rail Inclination $\alpha = 0^o$, Initial Pendulum Angle $\beta(0) = 45^o$

Now we introduce the rail inclination to the pendulum cart system. In Figure 7, the rail is inclined forward by ten degrees, i.e. $\alpha = -10^o$, while the pendulum angle is initially inclined backwards, i.e. $\beta(0) = 20^o$.

Like before, the sliding mode controller stabilizes the pendulum angle more quickly, with less overshoot, and without exhibiting the oscillations that the LQR controller experiences. However, the system converges to a non-zero velocity of -0.131 m/s,

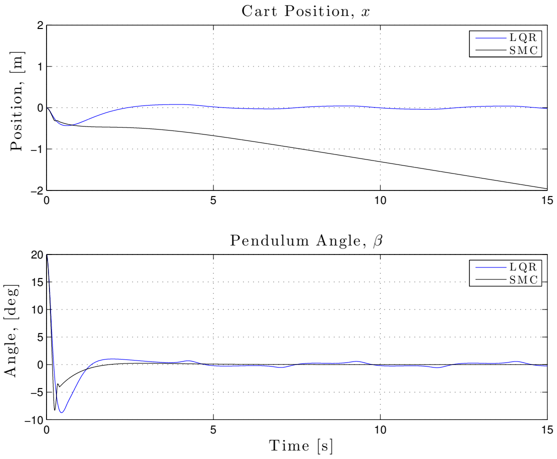

Figure 7: Rail Inclination $\alpha = -10^o$, Initial Pendulum Angle $\beta(0) = 20^o$

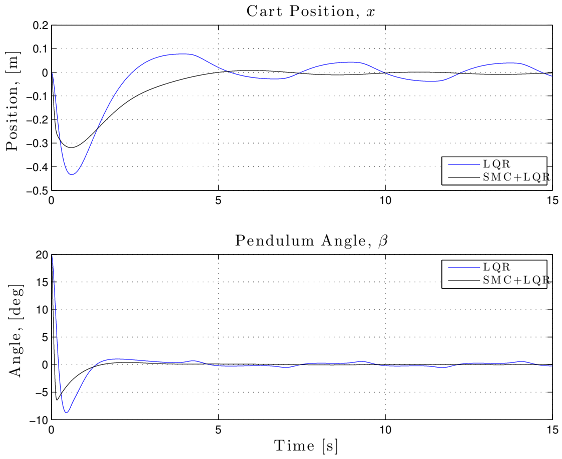

Figure 8: Rail Inclination $\alpha = -10^o$, Initial Pendulum Angle $\beta(0) = 20^o$

and therefore the cart continues to slide up the rail. Similarly, with a larger (smaller) rail inclination, the velocity converges to a larger (smaller) value.

The best method to deal with this steady state error is to incorporate an adaptive control method for parameter estimation, where the unknown parameter is $\sin(\alpha)$, from the constant disturbance due to gravity, $M_t mgl\cos(\beta)\sin(\alpha)$.

Due to time constraints, however, a simpler solution is taken in this paper. One can take advantage of the benefits of both the integral LQR controller (convergence in the presence of unknown rail inclination) and the SMC controller (mitigation of oscillations and smaller transient overshoot) by summing their control efforts.

$$u = u_{SMC} + u_{LQR}$$

where u_{SMC} is from Equation (14), and $u_{LQR} = -K_1 X_a$ as explained in Section 3.

Since the SMC controller is quite good at stabilizing the pendulum, the LQR controller that will be added to it can be modified to focus on the cart position. This is done by changing the weight matrix, Q, as follows:

$$Q = \text{diag}\left([10, 0, 100, 0, 100]\right)$$

As shown in Figure 8, the combined efforts of SMC and LQR provide a better system response than only using the LQR controller. Like before, the rail is inclined at -10 degrees, and the pendulum is initially at 20 degrees.

Finally, it is noted that this method requires some tuning of the LQR controller weights, in order to accommodate various rail inclinations and to improve performance. Therefore, this method is not truly robust to rail inclination, but is essentially robust to unknown Coulomb friction disturbances on the cart.

6. Conclusion and Future Work

Evidently, nonlinear control of the inverted pendulum cart system can be significantly better than its nonlinear counterpart, both in terms of transient performance (e.g. overshoot) and steady state performance (e.g. oscillation about desired cart/pendulum position).

However, the nonlinear control comes with a great increase in complexity, both in the design and implementation of the controller.

Further work needs to be done in order to make a controller that is robust to uncertain rail inclinations. This will involve an Adaptive Robust Control (ARC) design, which will achieve robust stability and robust performance for the inverted pendulum cart system on an inclined rail.

Appendix A. Simulation Setup

The Simulink model, which was used for all simulations, is shown in Figure A.9.

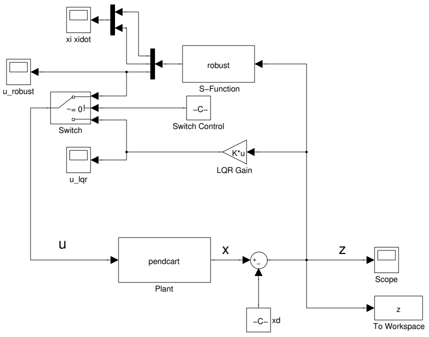

Figure A.9: Simulink Model

The *mdlDerivatives* function of the S-function block (above), *pendcart*, is shown below.

```
function sys = mdlDerivatives(t,x,u)
%
% M*q_dd + C*q_d + D = [K;0]*u - [Fc;0]
%
% q = [ x, beta, x_dot, beta_dot ]^T

global M_eq m g Mt I l It
global b_eq br K alpha FcA FcB

M = [ Mt, -m*l*cos(x(2)-alpha);
      -m*l*cos(x(2)-alpha), It ];
C = [ b_eq, m*l*sin(x(2)-alpha)*x(4);
      0, br ];
D = [ Mt*g*sin(alpha);
      -m*g*l*sin(x(2)) ];
Fc=FcA*tanh(FcB*x(3));

vec=M\(-C*[x(3);x(4)]-D+[K;0]*u(1)-[Fc;0]);

sys(1) = x(3);
sys(2) = x(4);
sys(3) = vec(1);
sys(4) = vec(2);
sys(5) = -x(1);
```

The *mdlDerivatives* function of the S-function block, *robust*, which is for the fictitious states η and ξ, is shown next.

```
function sys = mdlDerivatives(t,x,u)
%
% Fictitious System
% X = [eta; etadot; xi]
% u = [x; beta; xdot; betadot]

global m g It l br lambda1 lambda2

sys(1) = x(2);
sys(2) = -( g + It*u(4)^2/(m*l*cos(u(2))) ) * ...
 tan(u(2)) + br*u(4)/(m*l*cos(u(2)));
sys(3) = sec(u(2))^2 * u(4) - ...
 lambda1*x(2) - lambda2*sys(2);
```

The *mdlOutputs* function of the S-function block, *robust*, which computes and outputs the control input for the *pendcart* system, is shown below.

```
function sys = mdlOutputs(t,x,u)
%
global Mt M_eq m g I l It b_eq br K
global lambda1 lambda2
global Gamma1 Gamma2 Gamma3 Gamma4

Xdot  = mdlDerivatives(t,x,u);
xidot = Xdot(3);

mlc   = m*l*cos(u(2));
gamma = Mt*It - mlc^2;
beta1 = (1/gamma) * (Mt*m*g*l*sin(u(2)) - ...
 m^2*l^2*sin(u(2))*cos(u(2))*u(4)^2 - ...
 Mt*br*u(4) - b_eq*mlc*u(3));
beta2 = (1/gamma) * (mlc*K);
c1    = (2*It*tan(u(2))*u(4) - br) / mlc;

mu = 2*tan(u(2))*(sec(u(2))^2)*u(4)^2 - ...
     lambda1 * (br*u(4)/mlc - g*tan(u(2)) - ...
      It*u(4)^2*tan(u(2))/mlc) + ...
     (sec(u(2))^2 + lambda2*c1) * beta1 + ...
     lambda2 * g*sec(u(2))^2*u(4) + ...
     (lambda2/mlc) * ( It*u(4)^3 * (tan(u(2))^2 ...
     +sec(u(2))^2) - br*tan(u(2))*u(4)^2 );

den    = (sec(u(2))^2 + lambda2*c1) * beta2;

sys(1) = x(3);
sys(2) = xidot;
sys(3) = (1/den) * ...
         ( -mu -   ...
           Gamma1*tanh(100*x(3)) - ...
           Gamma2*tanh(100*xidot) - ...
           Gamma3*x(3) - ...
           Gamma4*xidot );
```

Bibliography

[1] Charles W Anderson. Learning to control an inverted pendulum using neural networks. *Control Systems Magazine, IEEE*, 9(3):31–37, 1989.

[2] A Benaskeur and A Desbiens. Application of adaptive backstepping to the stabilization of the inverted pendulum. In *Electrical and Computer Engineering, 1998. IEEE Canadian Conference on*, volume 1, pages 113–116. IEEE, 1998.

[3] Sue Ann Campbell, Stephanie Crawford, and Kirsten Morris. Friction and the inverted pendulum stabilization problem. *Transactions of the ASME-G-Journ DynamicSystems Measurement Control*, 130(5):054502, 2008.

[4] Chaio-Shiung Chen and Wen-Liang Chen. Robust adaptive sliding-mode control using fuzzy modeling for an inverted-pendulum system. *Industrial Electronics, IEEE Transactions on*, 45(2):297–306, 1998.

[5] K Furuta, T Okutani, and H Sone. Computer control of a double inverted pendulum. *Computers & Electrical Engineering*, 5(1):67–84, 1978.

[6] GA Medrano-Cerda. Robust stabilization of a triple inverted pendulum-cart. *International Journal of Control*, 68(4):849–866, 1997.

[7] Ahmad Nor Kasruddin Nasir, Mohd Ashraf Ahmad, and Mohd Fua'ad Rahmat. Performance comparison between lqr and pid controllers for an inverted pendulum system. In *AIP Conference Proceedings*, volume 1052, page 124, 2008.

[8] Samer Riachy, Yuri Orlov, Thierry Floquet, Raul Santiesteban, and Jean-Pierre Richard. Second-order sliding mode control of underactuated mechanical systems i: Local stabilization with application to an inverted pendulum. *International Journal of Robust and Nonlinear Control*, 18(4-5):529–543, 2008.

[9] Shubhobrata Rudra and Ranjit Kumar Barai. Robust adaptive backstepping control of inverted pendulum on cart system. *equilibrium*, 5(1), 2012.