ADAPTIVE CONTROLLER
FOR STABILIZING THE INVERTED PENDULUM SYSTEM

Slamet Rianto

University Undergraduate Research Fellow, 1994-95

Texas A&M University

Department of Electrical Engineering

APPROVED

Undergraduate Advisor ____________

Exec. Dir., Honors Program ____________

TABLE OF CONTENTS

ABSTRACT

The Inverted Pendulum System, which consists of a two-dimensional inverted pendulum mounted on a motor driven cart, is a very good model to describe an unstable system. The objective of our research is to keep the pendulum in a vertical position by asserting appropriate force to the cart. The force will then counter the torque caused by the movement of the pendulum. Using *Pole Placement* and *State Feedback* techniques, we can find that the required force is proportional to the angular displacement and angular velocity of the pendulum. This force is also influenced by the system parameters, i.e. the mass of the cart, the mass of the pendulum, and the length of the pendulum.

Unfortunately, in the real-world applications, most of the time the system parameters are not known or changing with time. In such a case, we need a controller that possesses the capability to continuously and automatically monitor any small changes in the system parameters, and then appropriately adjust itself to keep the system in a stable condition. Such a controller is called an *Adaptive Controller*. An adaptive controller can be designed as follows. First, a *Standard Parametric Model* containing the unknown parameters is developed. Then, an *On-line Parameter Estimation* technique is used to estimate the unknown parameters. Finally, the controller is designed based on the current estimates.

The block diagram as well as the MATLAB - SIMULINK circuitry of the designed adaptive controller is presented in this paper.

CHAPTER 1

INTRODUCTION

1.1 The Inverted Pendulum System

For many years, the inverted pendulum system has served as a benchmark for control system design. The system consists of a pendulum, which is simply a piece of rod with known center of gravity, mounted on a motor driven cart using a frictionless pivot. (see Figure 1)

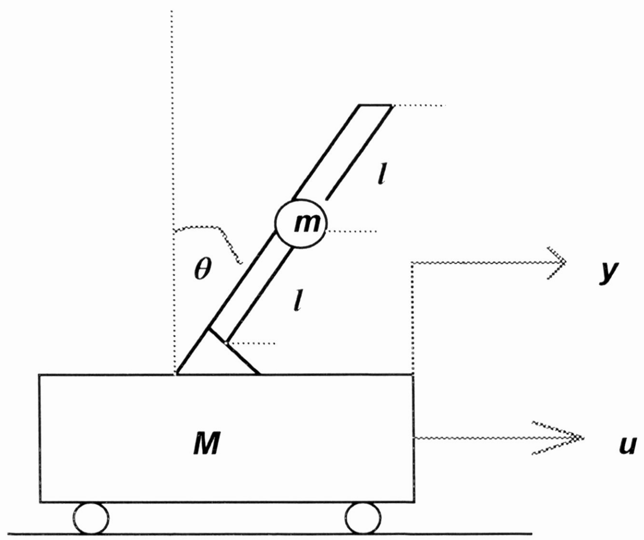

Figure 1: The Inverted Pendulum System

Here, ***M*** is the mass of the cart; ***m*** is the mass of the pendulum (the center of mass is located at the middle); ***θ*** is the angular displacement of the pendulum from the vertical position which corresponds to $\boldsymbol{\theta} = 0$; $\boldsymbol{u}$ is the force applied to the cart; and $\boldsymbol{y}$ is the position of the cart.

From the inverted pendulum system in Figure 1 above, it appears that it may be possible to force the pendulum to stay in the vertical position ($\boldsymbol{\theta} = 0$). However, because of the position of the pendulum, it is a little hard to do. Even if we are able to do it, the pendulum will not stay there long enough. A little disturbance would cause the pendulum to fall over in either direction. This is what we call an *unstable system*.

There are many unstable systems in real-world applications, ranging from the temperature of a chemical reactor to the dynamics of an airplane. Yet, for most cases, the instability of the system cannot be seen directly, without any analysis or calculations. In the case of the inverted pendulum system, however, the instability of the pendulum can be seen directly. We do not need to perform any calculations to determine if the pendulum is going to fall over. This makes the inverted pendulum system an intuitively natural model to describe an unstable system.

1.2 Stabilizing The Inverted Pendulum System

One of the tasks of a control engineer is to design a controller that stabilizes an unstable plant (system). The plant can be an amplifier that has an unstable gain, the cooling system in a nuclear reactor, or the automatic steering system in an automobile. In this paper, we will design a controller that stabilizes a model of an unstable plant: the inverted pendulum system.

Because the objective of the controller is to keep the inverted pendulum in a vertical position, the controller should produce a force ***u*** on the cart, which counters the torque produced by the movement of the pendulum. Therefore, if the pendulum swings to the *right*, the controller will assert a force ***u*** on the cart to the *right* to neutralize the torque, and vice versa. This way, any small tilt can be recovered without having the pendulum falling over.

If we were to make an educated guess, the controller force will be of the form

$$u = F(M, m, l)\theta + G(M, m, l)\dot{\theta}$$

where θ is the angular displacement, and $\dot{\theta}$ is the angular velocity of the pendulum. ***F(M, m, l)*** and ***G(M, m, l)*** are some functions of the *system parameters*, i.e. mass of the cart, mass of the pendulum, and the length of the pendulum. This conjecture makes sense because, as described in the previous paragraph, the controller force needs to know in what direction the force should be applied to the cart. In order to know the correct direction, the controller has to be supplied with the information about the movement of the pendulum (i.e. θ and $\dot{\theta}$ must be supplied). In addition to knowing the direction, the controller must also decide the magnitude of the force to be applied. This is taken care of by supplying the *system parameters* (***M***, ***m***, and ***l***) on the force equation. The information about θ and $\dot{\theta}$ is also useful for this matter. Bigger force is needed to neutralize the torque caused by faster swing of the pendulum.

The complete force equation for this problem, along with the design methodology will be described thoroughly in the following chapters.

1.3 The Need for an Adaptive Controller

Let us examine the force controller equation one more time.

$$u = F(M, m, l)\theta + G(M, m, l)\dot{\theta}$$

Notice that in order for the controller to work, we have to supply it with the *system parameters*: ***M***, ***m***, and ***l***. Unfortunately, in real-world problems, the system parameters are not known or keep changing with time. The design of the controller, therefore, is not as simple as when the system parameters are known. We need a controller which not only know, how to apply the appropriate force to stabilize the inverted pendulum, but is also capable of continuously monitoring any small changes in the system parameters, and then automatically adjust itself to keep the pendulum stable. Such a controller is called an *Adaptive Controller.*

In the case of the Inverted Pendulum System, the unknown system parameters are the mass of the cart (***M***), and the mass of the pendulum (***m***). We could have made the length of the pendulum (***l***) an unknown parameter as well; however, this would have made the system unrealistic. In most cases, the length of the pendulum is known or fixed.

In order to develop an adaptive controller, the design is divided into two stages:

1. Non-Adaptive Design

In this stage of the design, we assume that the system parameters, i.e. ***M*** and ***m*** are known. The design includes the use of techniques called *Pole Placement* and *State Feedback*. Those techniques are actually used to force the unstable pole(s) of the system to the stable region, i.e. the left half of the complex plane. The complete explanation and design methodology for this stage can be found in Chapter 2.

2. Adaptive Design

In this stage, the actual design of the adaptive controller is carried out. Because we do not know the system parameters, the parameters are estimated, giving us what are called *Parameter Estimates*. The parameter estimation is carried out by minimizing the error between the true parameters and the estimates, using the so called *Gradient Method*. The complete design methodology for this stage as well as the final design of the adaptive controller will be explained in Chapter 3.

CHAPTER 2

THE NON-ADAPTIVE DESIGN

In this chapter, we focus on the non-adaptive design, i.e. the design of the controller assuming that all system parameters are known. The main reason for undertaking such a study is that if we fail to solve the control problem for the known parameter case, then there is little or no hope that an adaptive controller would do anything better in the case where the parameters are unknown.

2.1 The State Space Representation

In order to analyze the stability of a system, we will represent the system in a state space form. This representation makes it easier to design the controller because we can see where the system poles are.

Let us begin with the mechanical equations describing the system dynamics. If we assume that θ is very small (i.e. $\sin\theta \sim \theta$ and $\cos\theta \sim 1$), and that there is neither friction at the pivot nor slip at the cart's wheels, the system dynamics can be described by the two equations:

$$(J + ml^2)\ddot{\theta} + ml\ddot{y} - mgl\theta = 0 \quad \text{...} (1)$$

$$ml\ddot{\theta} + (M + m)\ddot{y} = u \quad \text{...} (2)$$

where:

$J = \frac{ml^2}{3}$ is the moment of inertia of the pendulum

Equation (2) can be rewritten as:

$$(M+m)\ddot{y} = u - ml\ddot{\theta}$$

or,

$$\ddot{y} = \frac{u}{(M+m)} - \frac{ml}{(M+m)}\ddot{\theta} \quad \text{.......(3)}$$

Now, substituting (3) to (1)

$$(J + ml^2)\ddot{\theta} + ml[\frac{u}{(M+m)} - \frac{ml}{(M+m)}\ddot{\theta}] - mgl\theta = 0$$

or,

$$(J + ml^2)\ddot{\theta} - \frac{(ml)^2}{(M+m)}\ddot{\theta} + \frac{ml}{(M+m)}u - mgl\theta = 0$$

$$[(J + ml^2) - \frac{(ml)^2}{(M+m)}]\ddot{\theta} - mgl\theta + \frac{ml}{(M+m)}u = 0$$

$$\ddot{\theta} = [\frac{(mgl)(M+m)}{(J+ml^2)(M+m)-(ml)^2}]\theta - [\frac{ml}{(J+ml^2)(M+m)-(ml)^2}]u \quad \text{.........(4)}$$

Let us define:

$$\beta = \sqrt{\frac{(mgl)(M+m)}{(J+ml^2)(M+m)-(ml)^2}}$$

$$\gamma = \frac{ml}{(J+ml^2)(M+m)-(ml)^2}$$

Now, equation (4) can be rewritten as:

$$\ddot{\theta} = \beta^2\theta - \gamma u \quad \text{...(5)}$$

If we define the state variables ***x1*** and ***x2*** as ***x1*** = θ and ***x2*** = $\dot{\theta}$,

the state space equations become:

$$\dot{x}1 = x2$$

$$\dot{x}2 = \beta^2 x1 - \gamma u$$

which can be rewritten in vector-matrix form as:

$$\begin{bmatrix} \dot{x}1 \\ \dot{x}2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \beta^2 & 0 \end{bmatrix} \begin{bmatrix} x1 \\ x2 \end{bmatrix} + \begin{bmatrix} 0 \\ -\gamma \end{bmatrix} u$$

$$\begin{bmatrix} y1 \\ y2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x1 \\ x2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u$$

Here, we choose the outputs ***y1*** and ***y2*** of the state-space system to be equal to the state variables ***x1*** and ***x2*** respectively.

A typical state-space description of the system is in terms of four matrices (***A***, ***B***, ***C***, and ***D***). In our case, these matrices are:

$$A = \begin{bmatrix} 0 & 1 \\ \beta^2 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$C = \begin{bmatrix} -\gamma & 0 \\ 0 & -\gamma \end{bmatrix}$$

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

It is sometimes convenient to describe a system using a transfer function and/or block diagram instead of using a bunch of matrices in the state-space description. The conversion from state-space system to transfer function representation can be done using the *Leverrier Algorithm* [2]. However, since our state-space system is in a *companion form*, it is simple enough to convert it just by inspection. Figure 2.1 describes the representation of the inverted pendulum system using block diagram generated by the transfer function. Notice that the system has an input ***u*** and two outputs ***y1*** and ***y2***. Notice also that, because of the way we set it up, ***y1*** = ***x1*** = $\boldsymbol{\theta}$, and ***y2*** = ***x2*** = $\dot{\theta}$

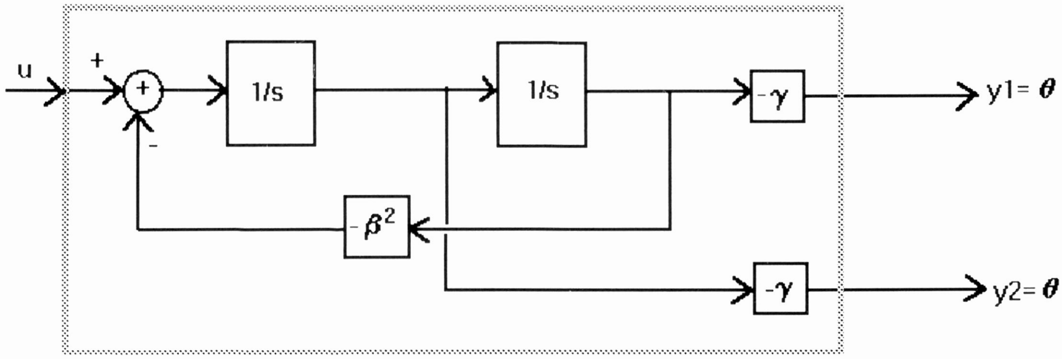

Figure 2.1 Block Diagram of The Inverted Pendulum System

2.2 The State Feedback System and Pole Placement Technique

From the state-space system described in section 2.1, we can infer (from matrix ***A***) that the characteristic equation for the inverted pendulum system is:

$$s^2 - \beta^2 = 0$$

If we look at the definition of $\boldsymbol{\beta}$ in section 2.1, we can see that β^2 is always positive. Therefore, $\boldsymbol{\beta}$ is always real. Hence, the system poles, ***p1*** and ***p2***, are:

$$p1 = +\beta$$
$$p2 = -\beta$$

Notice that one of the system poles, i.e. ***p1***, is located in the right hand side of the complex plane (unstable region). This makes the inverted pendulum system unstable.

The *Pole Placement Technique* is one of the most common methods to stabilize an unstable system. In this technique, we literally force the pole(s) that are located in the unstable region to move to the stable region. Therefore, in our case, we need to move the pole ***p1*** to the left hand side of the complex plane.

To make our analysis simple, let us define the desired poles, ***pd1*** and ***pd2***, which are, of course, located in the stable region. Our purpose is, therefore, to force the original poles ***p1*** and ***p2*** to move to ***pd1*** and ***pd2*** respectively. Let us pick the values for ***pd1*** and ***pd2***.

$$pd1 = -10$$
$$pd2 = -20$$

Hence, the desired characteristic polynomial would be:

$$(s+10)(s+20) = s^2 + 30s + 200$$

And, therefore, the desired matrix ***Ad*** in companion form is:

$$Ad = \begin{bmatrix} 0 & 1 \\ -200 & -30 \end{bmatrix}$$

Now, instead of forcing the system poles ***p1*** and ***p2*** to the desired poles ***pd1*** and ***pd2*** individually, we can force the closed loop state-space matrix ***A*** to be the desired matrix ***Ad***, and get the same result. This can be done with the help of a feedback called *State Feedback System*. The State Feedback has the following form:

$$u = -\begin{bmatrix} k1 & k2 \end{bmatrix} \begin{bmatrix} x1 \\ x2 \end{bmatrix}$$

Notice that the State Feedback $\boldsymbol{u}$ is a function of both state variables $\boldsymbol{x1}$ and $\boldsymbol{x2}$, which are also the output of the state-space system ($\boldsymbol{y1}$ and $\boldsymbol{y2}$) because of the way we chose matrix C in section 2.1. This choice of output makes the implementation of the controller much easier because we can construct the state feedback by taking the signal from the output of the system, which is $\boldsymbol{y1}$ and $\boldsymbol{y2}$, and then multiplying them with either $\boldsymbol{k1}$ or $\boldsymbol{k2}$ and adding them together to obtain the state feedback $\boldsymbol{u}$. (see Figure 2.2). Here, $\boldsymbol{r}$ is just a reference signal.

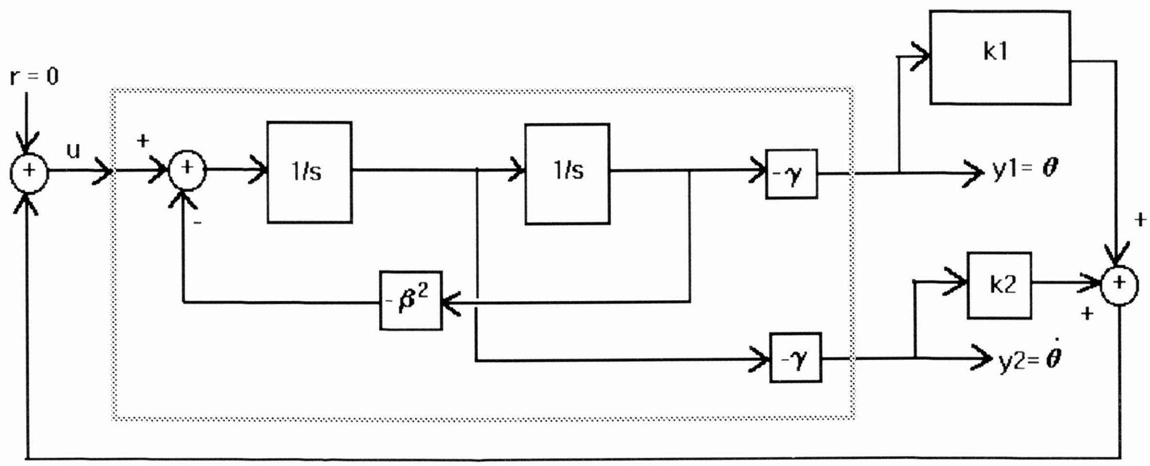

Figure 2.2 Implementation of The Controller Using State Feedback Technique

The next step is to determine ***k1*** and ***k2***. They can be determined by solving a simple matrix equation as follows:

$$[Ad][x] = [A][x] + [B][u]$$

$$\begin{bmatrix} 0 & 1 \\ -200 & -30 \end{bmatrix}\begin{bmatrix} x1 \\ x2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \beta^2 & 0 \end{bmatrix}\begin{bmatrix} x1 \\ x2 \end{bmatrix} - \begin{bmatrix} 0 \\ -\gamma \end{bmatrix}\begin{bmatrix} k1 & k2 \end{bmatrix}\begin{bmatrix} x1 \\ x2 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 \\ -200 & -30 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \beta^2 & 0 \end{bmatrix} - \begin{bmatrix} 0 \\ -\gamma \end{bmatrix}\begin{bmatrix} k1 & k2 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 \\ -200 & -30 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \beta^2 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ -\gamma k1 & -\gamma k2 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 \\ -200 & -30 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \beta^2 + \gamma k1 & \gamma k2 \end{bmatrix}$$

So,

$$k1 = \frac{-(200 + \beta^2)}{\gamma}$$

$$k2 = \frac{-30}{\gamma}$$

Therefore, the state feedback ***u*** is:

$$u = \frac{(200 + \beta^2)}{\gamma} x1 + \frac{30}{\gamma} x2$$

The complete implementation of the controller using the Pole Placement and State Feedback Technique is illustrated on the following figure:

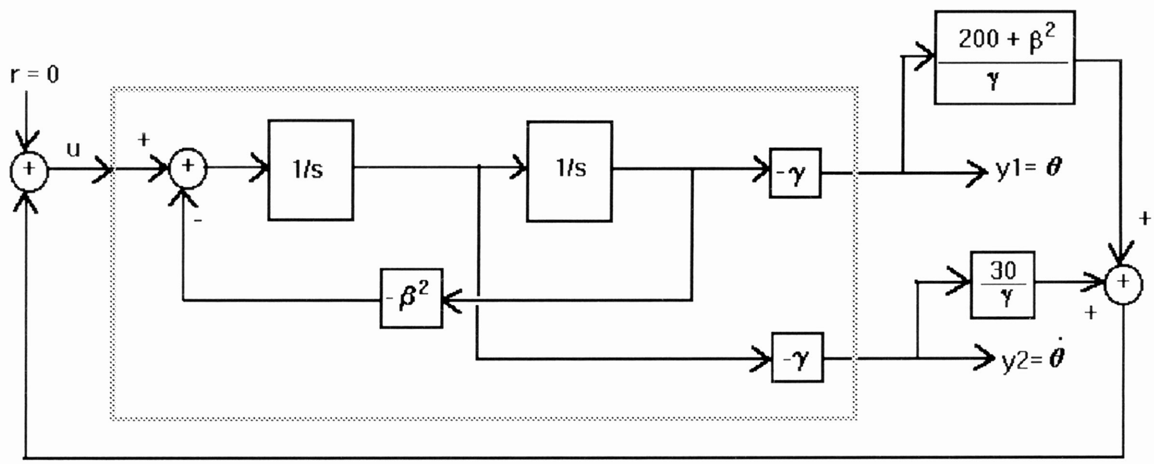

Figure 2.3 The Complete Implementation of Non-Adaptive Controller

CHAPTER 3

THE ADAPTIVE CONTROLLER

3.1 Introduction

In the previous chapter, the system parameters of the inverted pendulum system, i.e. the mass of the cart M and the mass of the pendulum m, were assumed to be known. The construction of the controller are based on the values of those parameters. In the case of an adaptive controller, however, the system parameters are not known; therefore, instead of using the true system parameters, we will be using the *estimates* of those parameters. These estimates will be updated on-line using a recursive estimation procedure.

In designing the adaptive controller, we will use the same state-feedback controller structure developed in Chapter 2. The only difference is that we will be updating $k1$ and $k2$ based on the estimates that we obtain. Since we do not know the true values of the parameters, we cannot use them to update $k1$ and $k2$. Fortunately, we know (can measure) the values of the outputs, $y1$ and $y2$, and the input u. Thus, we will be using these values ($y1$, $y2$, and u) to develop an *Adaptive Law* that will govern the construction of the parameter estimates. Based on this estimates, we can then update $k1$ and $k2$.

Basically, the adaptive controller can be designed using the following steps:

1. The system parameters, M and m will be related to the unknown parameters in the *Standard Parametric Model*.

2. An *On-line Parameter Estimation Technique* will be used to estimate M and m based on the measurements of ***y1***, ***y2***, and ***u***.

3. The controller will then be designed based on the current estimates.

3.2 The Standard Parametric Model for The Inverted Pendulum System

Recall that the state-space representation for the Inverted Pendulum System is:

$$A = \begin{bmatrix} 0 & 1 \\ \beta^2 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$C = \begin{bmatrix} -\gamma & 0 \\ 0 & -\gamma \end{bmatrix}$$

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Now, if we consider only the first output (i.e. ***y1***), the matrix ***C*** will become:

$$C = \begin{bmatrix} -\gamma & 0 \end{bmatrix}$$

Therefore, we can describe the system in transfer function form as:

$$y1(s) = \frac{-\gamma}{s^2 - \beta^2} u(s)$$

or,

$$s^2[y1] - \beta^2 y1 = -\gamma u \qquad (3.1)$$

In this stage, we want all the signals in equation (3.1) being either measurable signals or the parameters that we want to estimate. The polynomial looks all right because we can measure both ***y1*** and ***u***, and we can consider β^2 and γ to be the parameters we want to estimate (both β^2 and γ are functions of the unknown system parameters, ***M*** and ***m***). However, we really do not want the term ($s^2[y1]$) because this means we take the double derivative of the output ***y1***. If we take the derivative of a signal, the high frequency noise of that signal will be amplified. Therefore, any operations that include taking derivatives of a signal must be avoided.

In order to avoid this problem, we can *filter* both sides of the equation by a stable, strictly proper transfer function $1/\lambda(s)$ where

$$\lambda(s) = s^2 + 30s + 200$$

is an arbitrary second order polynomial, with stable zeros. This polynomial is often called *Filter*. Therefore, When we filter both sides of the equation (3.1), we get

$$\frac{s^2}{\lambda(s)}[y1] - \beta^2 \frac{1}{\lambda(s)}[y1] = -\gamma \frac{1}{\lambda(s)}[u]$$

or:

$$\frac{s^2}{\lambda(s)}[y1] = \beta^2 \frac{1}{\lambda(s)}[y1] - \gamma \frac{1}{\lambda(s)}[u]$$

or:

$$\frac{s^2}{\lambda(s)}[y\ 1] = \begin{bmatrix}\beta^2 & -\gamma\end{bmatrix} \begin{bmatrix} \frac{1}{\lambda(s)}[y\ 1] \\ \frac{1}{\lambda(s)}[u] \end{bmatrix} \qquad (3.2)$$

If we let:

$$M^* = \begin{bmatrix}\beta^2 & -\gamma\end{bmatrix}^T$$

$$\psi = \begin{bmatrix} \frac{1}{\lambda(s)}[y\ 1] & \frac{1}{\lambda(s)}[u] \end{bmatrix}^T$$

$$z = \frac{s^2}{\lambda(s)}[y\ 1]$$

we can rewrite the equation (3.2) as:

$$z = M^{*T}\psi$$

This equation is the *Standard Parametric Model*. This model will form the basis for developing the parameter estimator described in the next section.

3.3 The On-line Parameter Estimation

Recall the Standard Parametric Model from the previous section:

$$z = M^{*T}\psi$$

This model represents the relationship between the input signal ψ and the output signal z, with M^* as the unknown constant. In other words, we only have access to the measurements of ψ and z, but not M^*.

Now, let M be the estimate of M^* and $\hat{z}$ be the estimate of z based on M. Thus:

$$\hat{z} = M^T \psi$$

and the estimation error is:

$$\varepsilon 1 = z - \hat{z} = z - M^T \psi$$

Let us choose a cost function $J(M)$, which is a representation of both estimation error and parameter error.

$$J(M) = \frac{\varepsilon 1^2}{2} = \frac{(z - M^T \psi)^2}{2}$$

Now, we want to minimize the cost function because when $J(M)$ is minimized, M will become a good estimate for M^*. The most widely known method to minimize the cost function is the so called *Gradient Method* [1]. In this method, we track the cost function $J(M)$ in the direction of its steepest descent. The result is the following differential equation:

$$\dot{M} = -\Gamma \nabla J(M)$$

where:

$\Gamma = \Gamma^T > 0$ is a scaling matrix

and $\nabla J(M)$ is the gradient of ***J (M)***, which is simply the derivative Jacobian of ***J (M)*** with respect to ***M***. Therefore,

$$\nabla J(M) = \frac{dJ(M)}{dM} = -(z - M^T \psi)\psi$$

and the final differential equation is:

$$\dot{M} = \Gamma(z - M^T \psi)\psi = \Gamma \varepsilon 1 \psi \,, \quad M(0) = \text{Mo}$$

This differential equation is the parameter estimator or *Adaptive Law*. The parameter estimates obtained from this adaptive law will be used to design the adaptive controller in the next section.

3.4 The Development of the Adaptive Controller

Let us rewrite the Adaptive Law:

$$\dot{M} = \Gamma(z - M^T \psi)\psi = \Gamma \varepsilon 1 \psi$$

Recall the definition of ***M****, ***ψ***, and ***z*** from section 3.2.

$$M^* = \begin{bmatrix} \beta^2 & -\gamma \end{bmatrix}^T$$

$$\psi = \begin{bmatrix} \frac{1}{\lambda(s)}[y\,1] & \frac{1}{\lambda(s)}[u] \end{bmatrix}^T$$

$$z = \frac{s^2}{\lambda(s)}[y\,1]$$

Remember that ***M*** is the estimate of ***M****.

If we let $\hat{\beta}^2$ and $\hat{\gamma}$ to be the estimates of β^2 and γ respectively, and if we choose $\Gamma = \Gamma^T = I$ (Identity), we can rewrite the adaptive law using the following matrix equation:

$$\begin{bmatrix} \dot{\hat{\beta}}^2 \\ -(\dot{\hat{\gamma}}) \end{bmatrix} = \left(\frac{s^2}{\lambda(s)}[y1] - \begin{bmatrix} \hat{\beta}^2 & -\hat{\gamma} \end{bmatrix} \begin{bmatrix} \frac{1}{\lambda(s)}[y1] \\ \frac{1}{\lambda(s)}[u] \end{bmatrix}\right) \begin{bmatrix} \frac{1}{\lambda(s)}[y1] \\ \frac{1}{\lambda(s)}[u] \end{bmatrix}$$

$$\begin{bmatrix} \dot{\hat{\beta}}^2 \\ -(\dot{\hat{\gamma}}) \end{bmatrix} = \left(\frac{s^2}{\lambda(s)}[y1] - \hat{\beta}^2 \frac{1}{\lambda(s)}[y1] + \hat{\gamma} \frac{1}{\lambda(s)}[u]\right) \begin{bmatrix} \frac{1}{\lambda(s)}[y1] \\ \frac{1}{\lambda(s)}[u] \end{bmatrix}$$

Therefore, the individual estimates are:

$$\dot{\hat{\beta}}^2 = (\frac{s^2}{\lambda(s)}[y1] - \hat{\beta}^2 \frac{1}{\lambda(s)}[y1] + \hat{\gamma} \frac{1}{\lambda(s)}[u])\frac{1}{\lambda(s)}[y1]$$

$$\dot{\hat{\gamma}} = (\hat{\beta}^2 \frac{1}{\lambda(s)}[y1] - \frac{s^2}{\lambda(s)}[y1] - \hat{\gamma} \frac{1}{\lambda(s)}[u])\frac{1}{\lambda(s)}[u]$$

The implementation of the adaptive controller is described in Figure 3.1 on the following page.

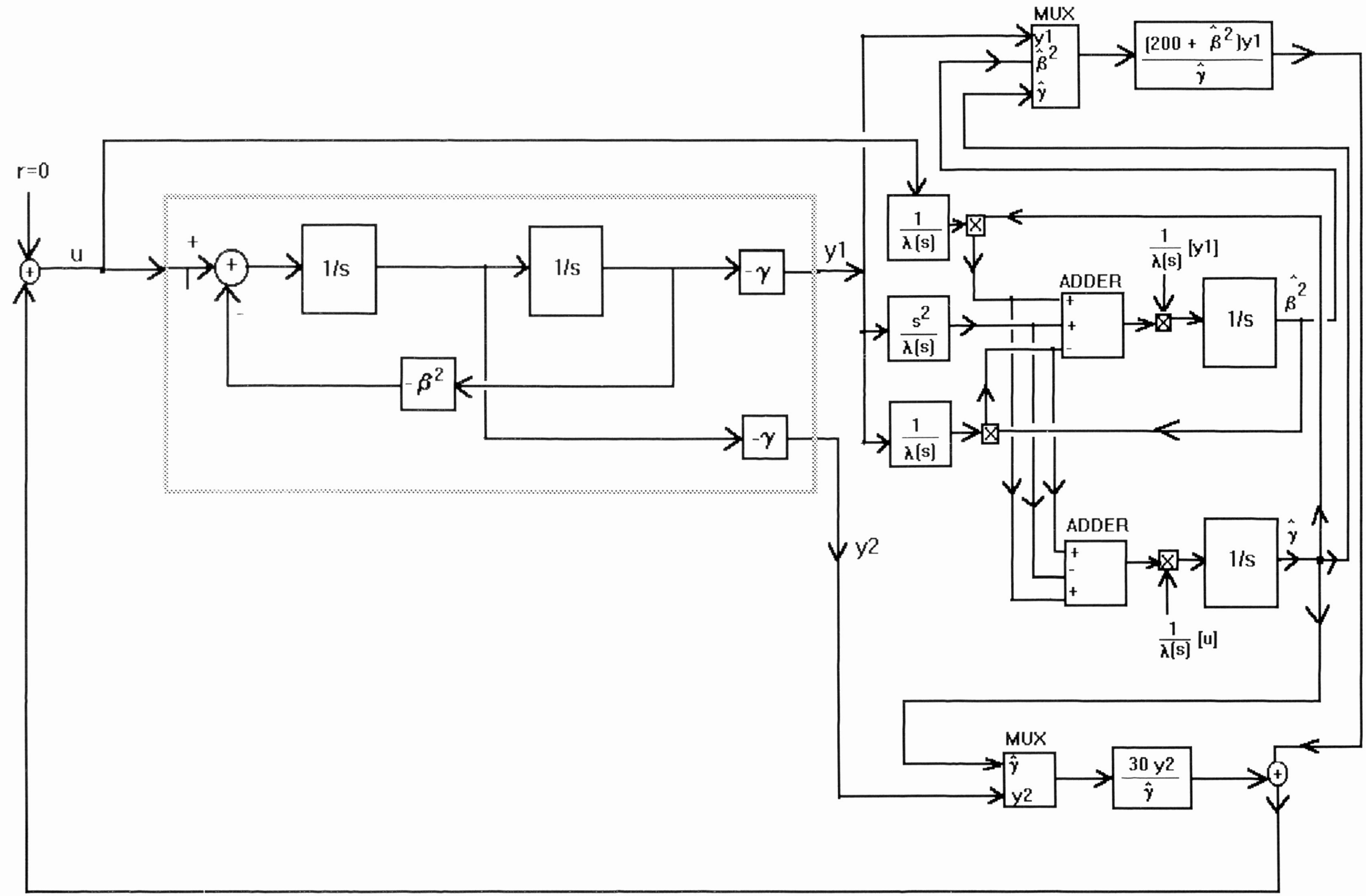

Figure 3.1 The Implementation of Adaptive Controller

3.5 Direction of Adaptation

In some cases, we have to know how the parameter estimates behave when we put the adaptive controller to work. The behavior of the parameter estimates could affect the performance of the controller. The adaptive controller that we have just designed is a perfect example of such a case. Consider the block diagram on Figure 3.1. Notice that we always use the parameter estimate $\hat{\gamma}$ in the denominator of the functions

$$\frac{(200+\hat{\beta}^2)y1}{\hat{\gamma}} \quad \text{and} \quad \frac{30y2}{\hat{\gamma}}.$$

As we can see, this could get very ugly if $\hat{\gamma}$ is zero. Therefore, we need to make sure that $\hat{\gamma}$ (t) does not pass through zero.

What we need to do first is to check whether $\hat{\gamma}$ is actually going to zero in some time t. Since we know the sign of γ , which is positive, we can do the checking by setting up a program that will monitor the value of $\hat{\gamma}$ at every time t. If the value of $\hat{\gamma}$ is some number δ close to zero (e.g. δ ~ 0.001), the program will check $d\hat{\gamma}(t)/dt$ from the adaptive law. If $d\hat{\gamma}(t)/dt$ is positive, then the direction of adaptation is pointing to the positive direction, and we don't need to worry about it because $\hat{\gamma}$ will not pass through zero. If $d\hat{\gamma}(t)/dt$ is negative, we need to stop the adaptation because $\hat{\gamma}$ will otherwise pass through zero. The intuitive discussion given above can be justified using a rigorous analysis [1].

CHAPTER 4

CONCLUSION

Even though it looks simple, the Inverted Pendulum System plays an important role in the field of Control Engineering because of its ability to perfectly represent an unstable system. The presence of a model in research is very important. A model can help researchers develop a theory. Furthermore, by using a model, researchers can test a theory they just developed to see whether it will work in the real world or not.

In the research that I have done, I designed an adaptive controller for a model (inverted pendulum system). Because the controller works on the model, and because the model is a perfect one to describe an unstable system, I believe that a similar controller may also be applied for stabilizing real-world systems. There are some problems, however, because in the model, everything is assumed to be ideal. Some instances of these assumptions are no noise, no friction, no loss, no slips, etc. Further studies on this subject are, therefore, required to perfect the adaptive controller and to make it applicable in a real world environment.

REFERENCES

[1] Ioannou, P. A., and Datta, A., *Robust Adaptive Control: A Unified Approach,* Proceedings to the IEEE, Vol. 79, No. **12**, pp. 1736-1768, December 1991.

[2] Chen, C., *Analog and Digital Control System Design*, Sanders College Publishing. Orlando, 1993.

[3] Siebert, W. M., *Circuits, Signals, and Systems,* Mc Graw-Hill. New York, 1986.

[4] Ogata, K., *Modern Control Engineering*, Prentice Hall, Inc. New Jersey, 1970.

[5] Higdon, D. T., and R. H. Cannon, Jr., *On the Control of Unstable Multiple-Output Mechanical System*, ASME Paper no. **63**-WA-148, 1963.

APPENDIX
THE MATLAB - SIMULINK BLOCK DIAGRAM

The Adaptive Controller Block is the same as in Figure 3.1 on the paper