ELCE707 HW02 Fuzzy Control of Inverted Pendulum –Simulation

M-B55515-2 Tang Qi

In this problem we will study the simulation of the fuzzy control system for the inverted pendulum (IP) studied in the tutorial introduction to fuzzy control. Use the model defined in Equation (2.25) on page 78 for the model for the pendulum. And use an appropriate numerical simulation technique for the nonlinear system and a small enough integration step size.

The building of system contains four steps, modeling moves of IP, designing fuzzy controller, solving the non-linear ODE for system, and tuning the systems with different memberships functions and rule bases to improve the system.

## Step 1: Modeling the Moving of IP

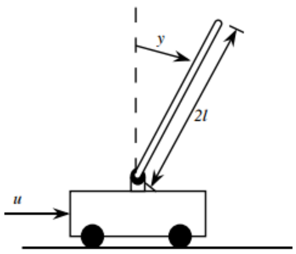


**Fig.01 Inverted Pendulum on a Cart**

Note that we don't need to model for the initial design of the fuzzy controller, but we need a model for mathematical analysis to evaluate the design. To simulate the fuzzy control system, we specify a mathematical model of the inverted pendulum as referred to Section 2.4 of book *Fuzzy Control* by Kevin M. et al:

$$\ddot{y} = \frac{9.8*(m+M)sin(y) - cos(y)[u + ml\dot{y}*sin(y)]}{l*[\frac{4}{3}(m+M) - m*cos^2(y)]} \qquad (1)$$

* Where m is the IP mass, M is the cart mass, u is the adding force and l is the half-length of the IP.

According to page 25, corresponding physical parameters of the IP are not provided, but one model for the inverted pendulum shown on page 25 is given in page 77, as

$$\ddot{y} = \frac{9.8\sin(y) - \cos(y)\left[\frac{\bar{u} + 0.25\dot{y}^2\sin(y)}{1.5}\right]}{0.5\left[\frac{4}{3} - \frac{1}{3}\cos^2(y)\right]}$$

$$\dot{\bar{u}} = -100\bar{u} + 100u \qquad (2)$$

* The second order filter on y (angular position) is the angular position of the pendulum, while the first order filter on u to produce ¯u represents an actuator.

## Step 2: Designing Fuzzy Controller

As indicated in Section 2.2, a rough transformation of linguistic description to rules in the rule base of fuzzy controller is as shown in Table. 01. Referring to the textbook, the membership functions are firstly in triangular and the error is in the range of $-\pi/2$ to $\pi/2$ while the change-in -error is in the range of $-\pi/4$ to $\pi/4$. The input membership functions with input values are as shown in .

**TABLE.01 Rule Table for the Inverted Pendulum with Rules That Are " ON" Highlighted:**

| "force" $u$ | | "change-in-error" $\dot{e}$ $-2$ | $-1$ | $0$ | $1$ | $2$ |
|---|---|---|---|---|---|---|
| "error" $e$ | $-2$ | 2 | 2 | 2 | 1 | 0 |
| | $-1$ | 2 | 2 | 1 | 0 | $-1$ |
| | $0$ | 2 | 1 | $\boxed{0}$ | $\boxed{-1}$ | $-2$ |
| | $1$ | 1 | 0 | $-1$ | $-2$ | $-2$ |
| | $2$ | 0 | $-1$ | $-2$ | $-2$ | $-2$ |

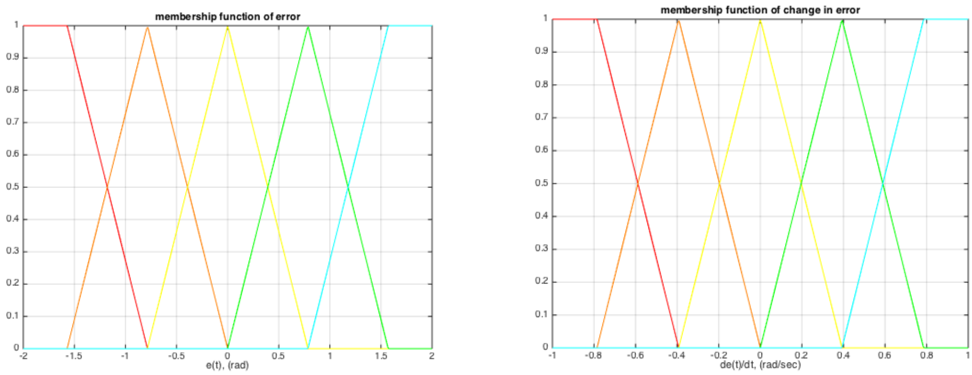


**Fig.02 Input Membership Functions with Input Values (e & e')**

Next is the inference step, which is determining conclusions according to the weight of each condition. Because we are using triangular membership function and each condition is of same weight, we are going to using the minimum of to represent the premise to certain situation and there should be always 2*2 conclusions reached (including repetitive conclusions).

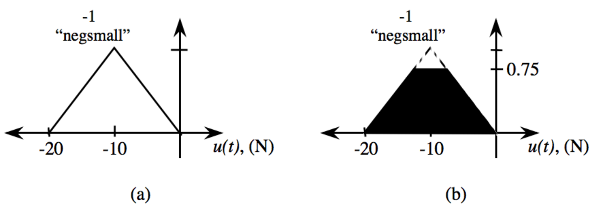


**Fig.03 One situation of inference**

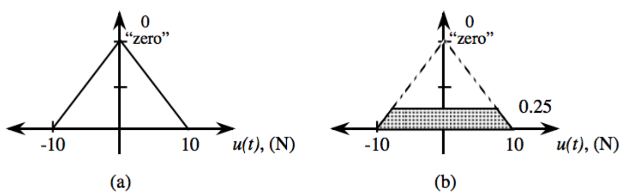


**Fig.04 the other situation of inference**

Following the book, we also use the minimum of the two membership functions to

represent the premise. The premises of the initial condition (y(0) = 0.1 radians (= 5.73 deg.), ˙y(0) = 0) is:
premise =

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0.1273 | 0 | 0 |
| 0 | 0 | 0.8727 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

Then we need to get the total area with regarding to the premises to get the force that we should imply onto the chart.

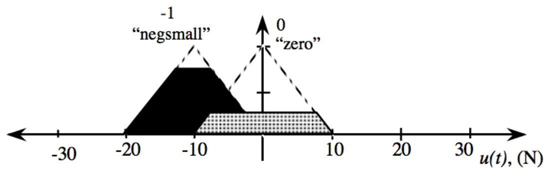


FIGURE 2.15 Implied fuzzy sets.

**Fig.05 interference to get decision**

The fuzzy controller uses the minimum operator to represent both the "and" in the premise and the implication and COG defuzzification).

**Step 3: solving the non-linear ODE for system**

The first design of the system is built as Fig. .Because it is hard to solve the ODE of the system analytically, indicated by the reference book, fourth order Runge-Kutta (RK4) method, which is superior than Eular Method for its high accuracy, and the integration step size h of 0.001 are used:

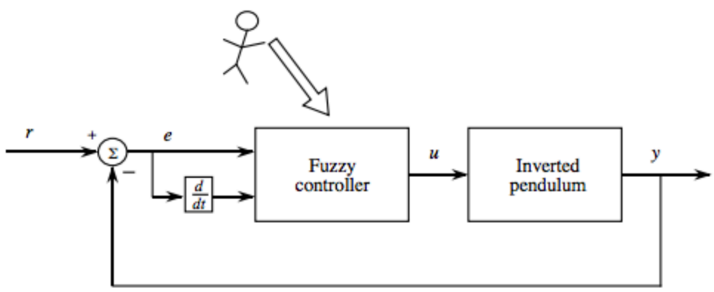


**Fig.06 rough fuzzy control system**

$$x(kh+h) = x(kh) + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$
\begin{aligned}
k_1 &= hF(x(kh), r(kh), kh) \\
k_2 &= hF\left(x(kh) + \frac{k_1}{2}, r\left(kh + \frac{h}{2}\right), kh + \frac{h}{2}\right) \\
k_3 &= hF\left(x(kh) + \frac{k_2}{2}, r\left(kh + \frac{h}{2}\right), kh + \frac{h}{2}\right) \\
k_4 &= hF\left(x(kh) + k_3, r(kh + h), kh + h\right)
\end{aligned}
$$

The initial condition be $y(0) = 0.1$ radians (= 5.73 deg.), $\dot{y}(0) = 0$, and the initial condition for the actuator state is zero $u(0)=0$ . All conditions are considered to be ideal and in theoretical. Simulating in Matlab 2014b, we get the exact result of fuzzy controller balancing inverted pendulum, as the first design in page 79. Plots of angular position and input force to time are shown in Fig.07, with corresponding codes appended in Appendix A.

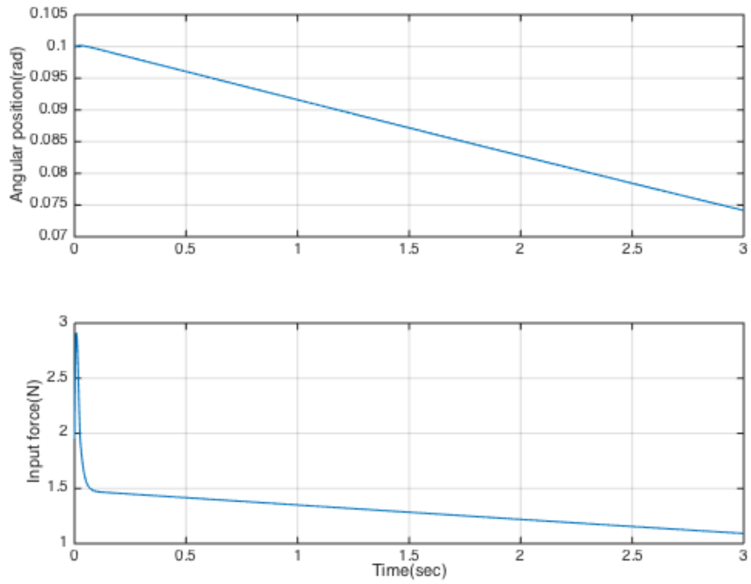


**Fig.07 response of rough system e= 0.1, e'=0**

From the figure we can see that it takes more than 3 seconds for the angular position of IP to be less than 0.075 rad, which is too long for a real-time system with interval of 0.01s for iterative steps. Thus, we improve the system via following steps.

**(a) Verify all the simulation results of Section 2.4.1 (i.e., use all the same parameters as used there and reproduce all the simulation results show).**

We use standard ideas from control engineering to conclude that we ought to try to tune the "derivative gain." To do this we introduce gains on the proportional and derivative terms, and at the same time we also put a gain h between the fuzzy controller and the inverted pendulum, as shown in Fig.

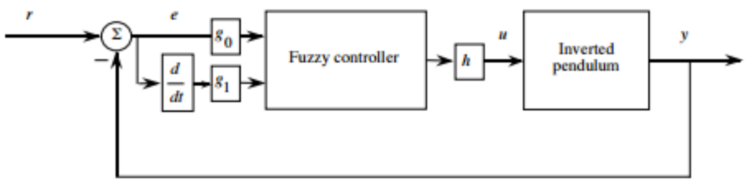


**Fig.08 improved fuzzy control system design for IP**

We choose g0 = 1, g1 = 0.1, and h = 1 and we can get a relatively improved result compared to the first design.

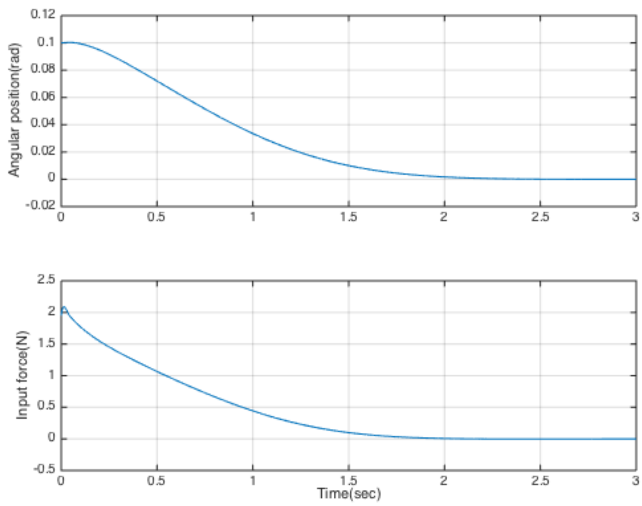


**Fig.09 triangular MF with scaling gains g0=1, g1=0.1, and h=1**

**(b) Repeat (a) for the case where we use Gaussian membership functions. Use product to represent the premise and implication and COG defuzzification. This problem demonstrates that changing membership function shapes and the inference strategy can have a significant impact on performance. Once you have completed (a) for all its parts, tune the scaling gains g0, g1 and h.**

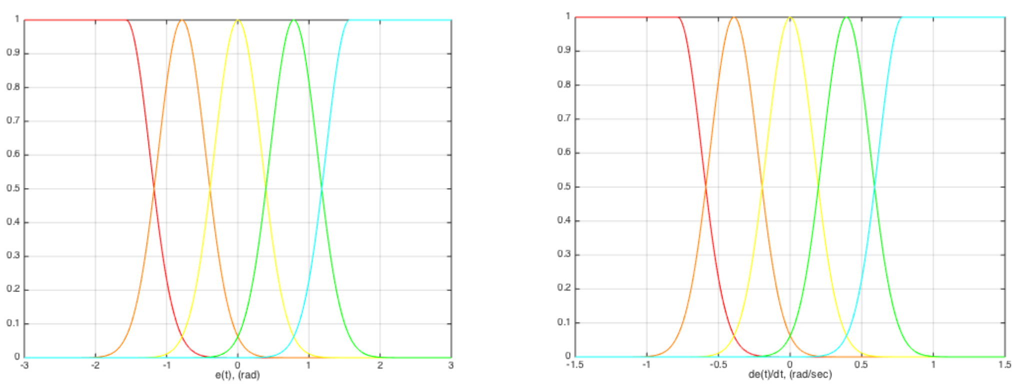


**Fig.10 Gaussian MF**

Still, we can use the minimum of two membership functions to replace the operation and the center of gravity (COG) method to do defuzzification. To be in accordance with part a, we we let g0=1, g1=0.1, h=1.

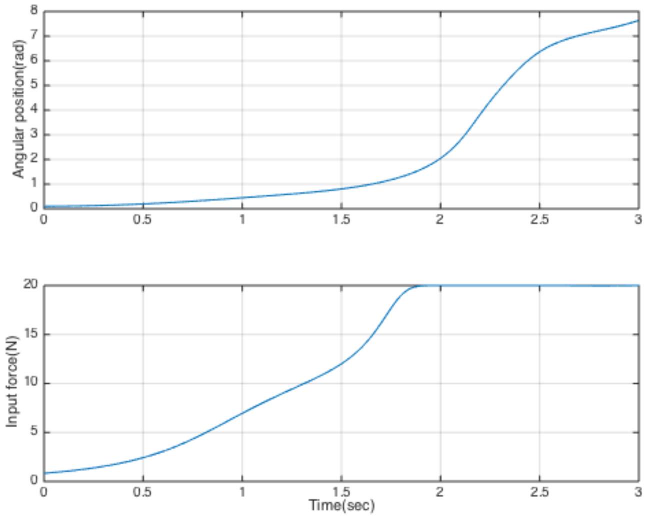


**Fig.11 Gaussian outputs with scaling gains g0=1, g1=0.1, and h=1**

Tuning the scaling gains of the following trials in Table

| g0 | g1 | h |
|---|---|---|
| 1 | 0. 1 | 2 |
| 1 | 0. 2 | 2 |
| 2 | 0. 2 | 2 |

We have following figures of results for comparison.

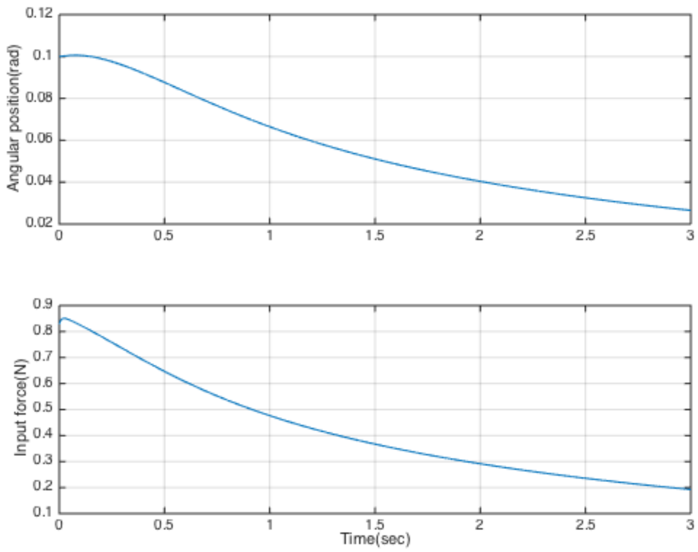


**Fig.11 Gaussian MF with scaling gains g0=1, g1=0.1, and h=2**

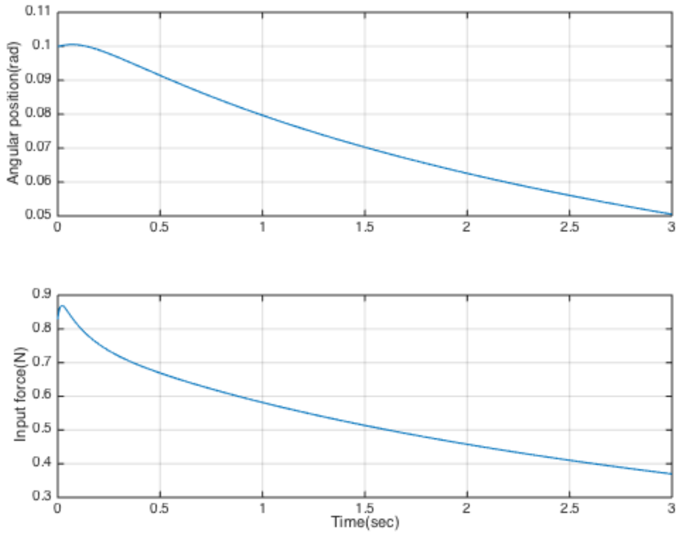


**Fig.12 Gaussian MF with scaling gains g0=1, g1=0.2, and h=2**

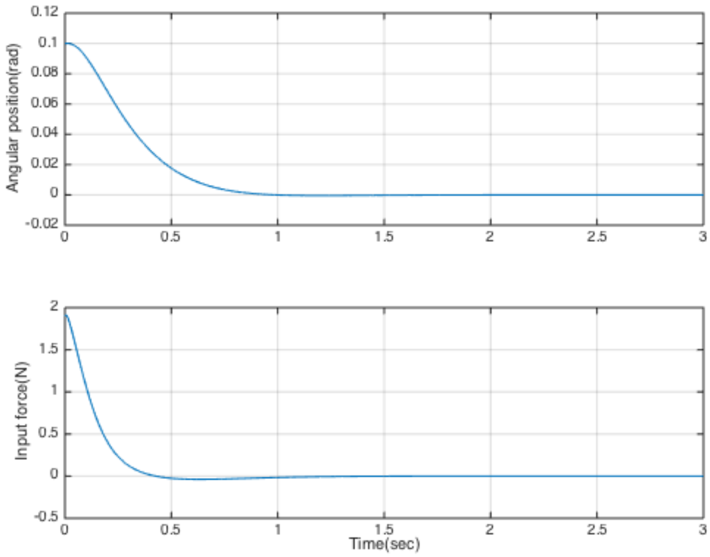


**Fig.13 Gaussian MF with scaling gains g0=2, g1=0.1, and h=2**

From current observation, we can find g0 and h are better to be larger while g1 is better to be smaller. To prove it, each parameter is examined as individual factors here.

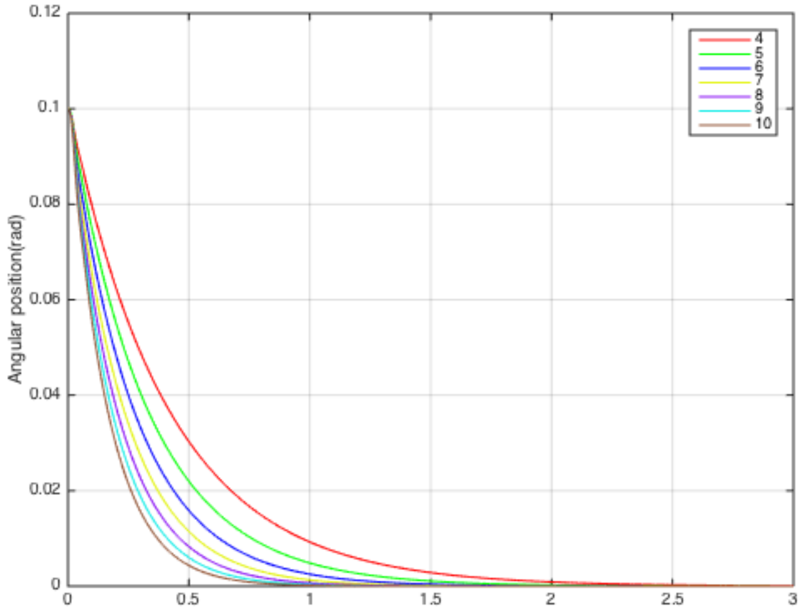


**Fig.14 error gain g0=4-10, g1=0.8, h=6**

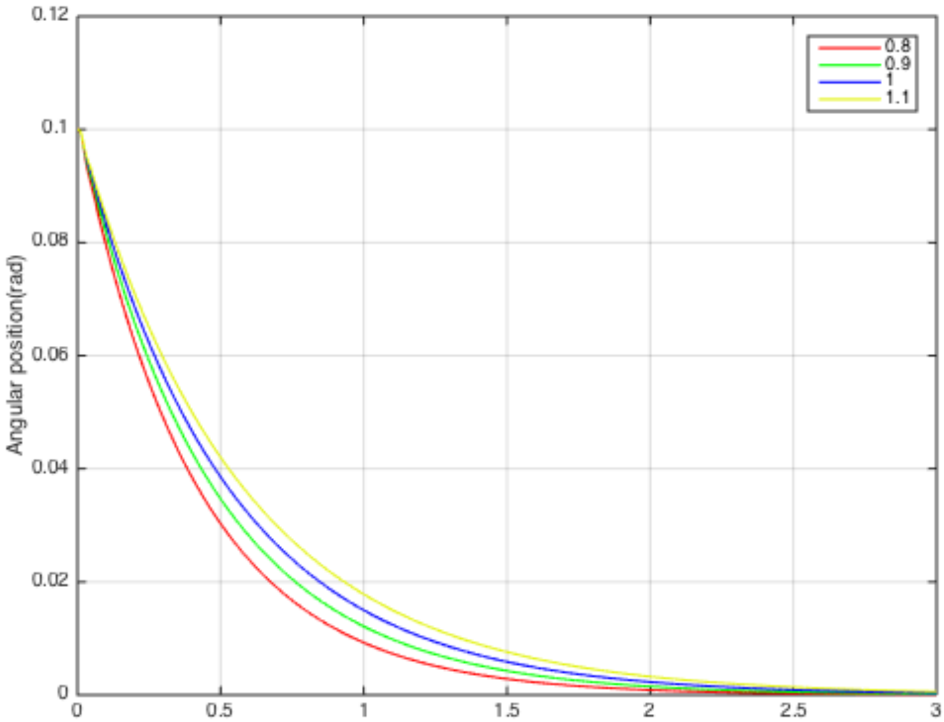


**Fig.15 error gain g0=4, g1=0.8-1.4, h=6**

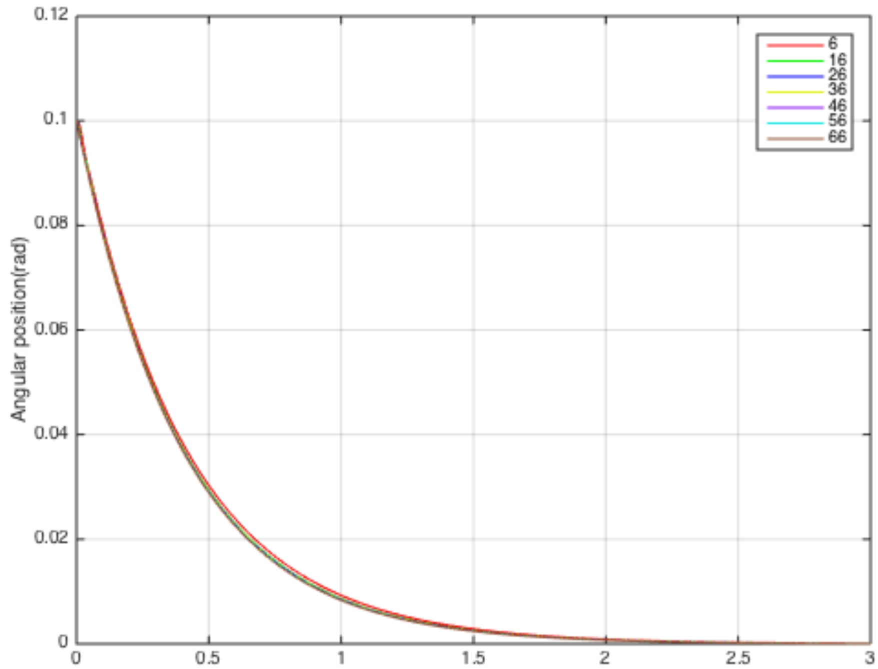


**Fig.16 error gain g0=4, g1=0.8, h=6-66**

**(c) Repeat (a) for the case where we use 49 rules, as in Exercise 2.4(b) (use triangular membership functions).**

Here we define the variables into 7 categories: XXL (7), XL (6), L (5), M (4), R (3), XR (2), XXR (1).

**TABLE.02 Refined Rule Table for the Inverted Pendulum:**

| F | | e' | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| e | 1 | 7 | 7 | 7 | 7 | 6 | 5 | 4 |
| | 2 | 7 | 7 | 7 | 6 | 5 | 4 | 3 |
| | 3 | 7 | 7 | 6 | 5 | 4 | 3 | 2 |
| | 4 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| | 5 | 6 | 5 | 4 | 3 | 2 | 1 | 1 |
| | 6 | 5 | 4 | 3 | 2 | 1 | 1 | 1 |
| | 7 | 4 | 3 | 2 | 1 | 1 | 1 | 1 |

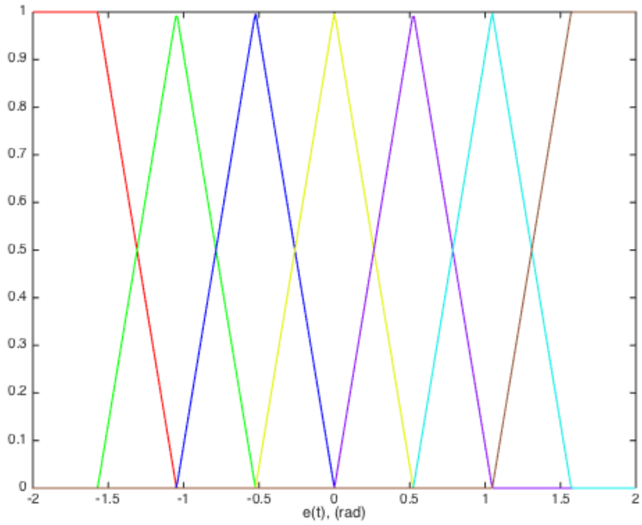


**Fig.17 Input membership function of angular position**

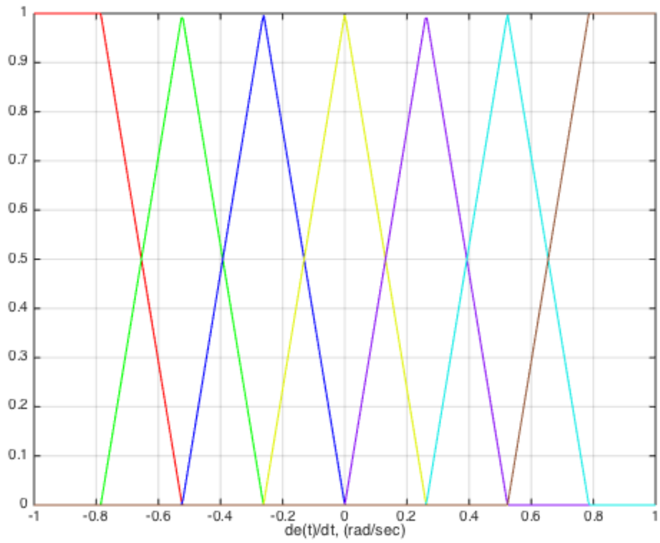


**Fig.18 Input membership function of angular speed**

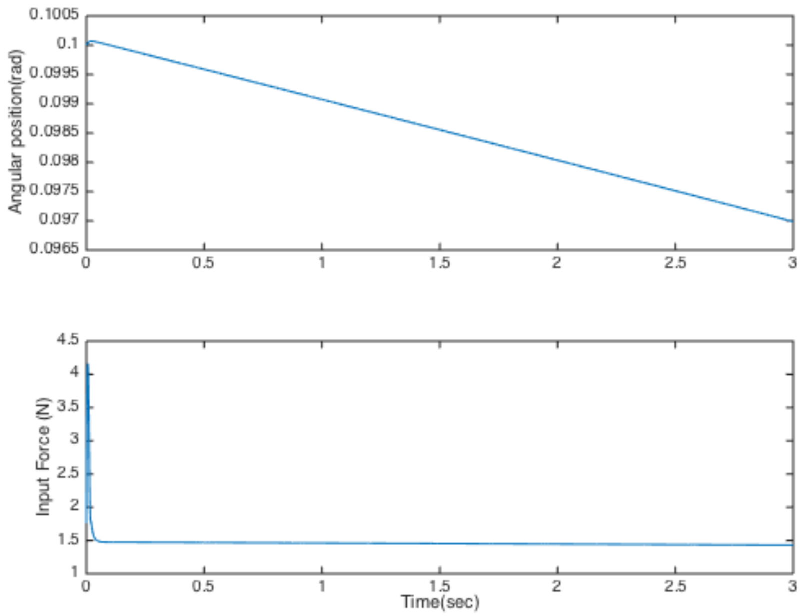


**Fig.19 triangular MF with 7*7 rule bases**

From the above plots, we cannot find obvious difference of Triangular membership function and Gaussian membership function. But as Gaussian is continuous in time domain, the outputs should be smoother compared with triangular membership functions.