非線性控制第九章作業

李佳玲 F44079029

Consider the following system



We’re going to design adaptive control  in order to track the following linear model reference, with the system parameters being unknown:



(a)

The control signal is written as



Assuming the parameters  are known, we’re going to acquire control law parameters  which will ensure the transfer function of both  and  being identical.

Plugging  into the original system gives





We can then obtain the parameters  by comparing this with the model reference:



(b)

Unlike in (a), we actually don’t know the parameters , therefore, we’re going to change the control law into



where  stand for the estimation of  , respectively.

Define the tracking error and parameters error as



We can then express the estimated parameters as follows:



Plugging this into the system gives





Afterwards, we apply the definition of tracking error:





The equation above shows the relationship between tracking error and parameters error.

Now, we need to have three other differential equations since there are four time variables but only one equation.

We can do that by assuming



where  are undecided parameters.

To make the tracking error , we need to apply Lyapunov stability theorem for choosing proper .

For the Lyapunov function, we choose



where  are free parameters (自由選定參數).

Then, taking the derivative of  gives



In order to satisfy the condition , we can set  as



Plugging this back into the equation gives



Therefore, we know that when , we can ensure , but we can’t promise that  will be as well. This means that this kind of gain-adjusting mode can achieve our goal of  but can’t perfectly estimate the parameters .

(c)

Let , the system then becomes





where 

Alongside with our first ODE, we also have the rest of the four ODE:



where , and  is chosen here.

Now, we’re going to use Matlab to perform numerical simulation and attest our result with a couple of different input  .

Firstly,  is applied.

|  |  |
| --- | --- |
|  |  |
|  |  |

From the figures, we can see that tracking error  is successfully achieved. On the other hand, all the parameters error  doesn’t converge to 0. There exists a difference between the real value of the parameters and our estimation of them, which means that we will never estimate the parameters perfectly when applying step input .

Secondly, we use  instead to see how this will affect the error.

|  |  |
| --- | --- |
|  |  |
|  |  |

From the figures, we can see that when using , although it takes much longer for tracking error to converge to 0, it still successfully achieves that at the end. What’s more, all three of the parameters error  are a lot closer to 0. For , it can even estimate the parameter pretty correctly since the error is almost 0.

When we compare the two different input  and , both tracking error will satisfy the condition . In terms of parameters error , they won’t be as close to 0 like tracking error does, but they do become closer to 0 when using .

Before jumping into conclusion, I’m going to try out different input that’s more “complicated” than  such as changing the amplitude and frequency and even combine them together, hoping to estimate the error more quickly as well as accurately.

 :

|  |  |
| --- | --- |
|  |  |
|  |  |

And also :

|  |  |
| --- | --- |
|  |  |
|  |  |

Finally, we can infer that when we’re trying to estimate the parameters, we can do it better by using a more “complicated” signal to “excite” the system at higher frequency and therefore estimate the parameters more correctly. When we used the step input , it’s just too simple and quite boring actually, and as we saw in the figures, our estimation of parameters weren’t even close.

When looking at the figures, we see that both  and  achieved 0 faster than  does. There are several possibilities for this situation. Anyhow, all three of them as well as tracking error will converge to 0 when using more complicated signals, as shown above, and complete our goal.