February 14, 2024

6.3100 Lecture 4 Notes – Spring 2024

Experimental characterization of first order systems, and simulation tools

Dennis Freeman, Elfar Adalsteinsson, and Kevin Chen

Outline:

1. First order system: estimation of system properties
2. MATLAB tools for analyzing a first order system
3. Nominal and perturbation control signals
4. Review of complex numbers – interpretation using the complex plane

1. First order system: estimation of system properties

In the previous lectures, we discussed how to solve first order systems. If I give you the mathematical model of a system, then you can implement a proportional controller and find the optimal Kp. However, when you design a control system, how do you know the properties of the system in the first place? If I give you a robot, how do you figure out key parameters in your model? System identification is an area of study in which researchers measure the system properties through running simple experiments and observing system response.

We will introduce a very simple technique that is helpful for solving lab 1. First, a first order system is given by the equation:

$$y[n] = y[n-1] + \Delta T(\beta y[n-1] + \gamma c[n-1])$$

Here the parameters β and γ are system properties that we want to measure. We need to design a reasonable control input signal $c[n]$. Note that our goal is to measure β and γ, not stably control the system or closely follow a trajectory. Let's try two different options:

Feedback control:

$$u[n] = K_p(y_d[n] - y[n])$$

Given this feedback controller, the system equation becomes:

$$y[n] = y[n-1] + \Delta T(\beta y[n-1] + \gamma K_p(y_d[n-1] - y[n-1]))$$

$$y[n] = y[n-1] + \Delta T\big(\beta y[n-1] - \gamma K_p y[n-1]\big) + \gamma \Delta T K_p y_d[n-1]$$

$$y[n] - y[n-1] - \Delta T\big(\beta y[n-1] - \gamma K_p y[n-1]\big) = \gamma \Delta T K_p y_d[n-1]$$

$$y[n] = y[n-1](1 + \Delta T\beta - \Delta T\gamma K_p) + \gamma \Delta T K_p y_d[n-1]$$

Here the natural frequency is given by:

$$\lambda = 1 + \Delta T\beta - \Delta T\gamma K_p$$

This is a little bit problematic. The natural frequency changes as we change K_p. So it is not easy to measure the system β and γ. Let's try another controller.

Feedforward control:

$$u[n] = K_{ff} y_d[n]$$

Given this feedforward controller, the system equation becomes:

$$y[n] = y[n-1] + \Delta T(\beta y[n-1] + \gamma K_{ff} y_d[n-1])$$

$$y[n] = y[n-1] + \Delta T(\beta y[n-1]) + \gamma \Delta T K_{ff} y_d[n-1]$$

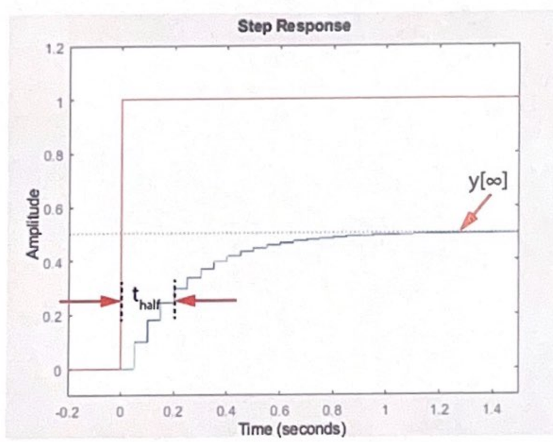
$$y[n] - y[n-1] - \Delta T(\beta y[n-1]) = \gamma \Delta T K_{ff} y_d[n-1]$$

$$y[n] = y[n-1](1 + \Delta T\beta) + \gamma \Delta T K_{ff} y_d[n-1]$$

Now this is much better because the natural frequency is given by:

$$\lambda = 1 + \Delta T\beta$$

To measure β and γ, we need to derive 2 relationships. A common approach is to look at the step response of a system. Here we let $y_d[n] = 1$ for n>0, and y[n]=0 for n=0. Suppose we run this experiment and obtain the following graph:



Graphically, we need to get two equations.

First, we can calculate λ by measuring the time y[n] uses to reach half of y[∞]. We have:

$$\lambda^{n^*} = 0.5$$

For the plot above, we measure $n^* = 3$. To calculate λ, we can use:

$$\lambda = \exp\left(\frac{1}{n^*}\log_e(0.5)\right)$$

We can back-calculate β:

$$\beta = \frac{\exp\left(\frac{1}{n^*}\log_e(0.5)\right) - 1}{\Delta T} = -4.1$$

Next, we need to solve for γ. We can solve for γ using the steady state condition. We measure y[∞] = 0.5. For large time, the equation becomes:

$$y[\infty] = y[\infty](1 + \Delta T\beta) + \gamma \Delta T K_{ff}$$

We obtain the equation:

$$\gamma = -\frac{y[\infty]\beta}{K_{ff}} = 2.1$$

You will practice this technique in lab 1.

2. MATLAB tools for analyzing a first order system

Most 1st order systems are simple enough that we can solve them manually. However, as we will see in the next few weeks, the algebra becomes tedious for higher order system. Numerical tools are useful for solving control systems. We are going to use MATLAB in this course. The reason that we use MATLAB is that it has very convenient packages, and many physical systems are controlled by MATLAB/Simulink.

The code below generates the system identification plot on page 2.

```
%define variables
Kff = 1;
beta = -4;
gamma = 2;
dt = 1/20;

%define the numerator and denomination of the transfer function
%will be covered in detail next week
den = [1, -(1+dt*beta)];
num = [0 dt*gamma*Kff];

%open a new figure
close all; figure(2); hold on

%form a discrete time control system
sys = tf(num,den,dt,'variable','z^-1');
step(sys,1.5) %simulate for 1.5 second

%plot reference data
axis([-0.2 1.5 -0.1 1.2])
plot([-0.2 0],[0 0],'r-')
plot([0 0],[0 1],'r-')
plot([0 1.5],[1 1],'r-')
```

The lines of code that may look puzzling relate to the definition of a transfer function (tf) with a numerator and a denominator. We will explain the transform techniques in the next 2 weeks of class. For now, it is sufficient to use this function. It comes from "pattern matching".

$$y[n] = y[n-1](1 + \Delta T\beta) + \gamma\Delta T K_{ff} y_d[n-1]$$

$$y[n] - y[n-1](1 + \Delta T\beta) = \gamma\Delta T K_{ff} y_d[n-1]$$

The denominator relates to the coefficients in front of y[n] and y[n-1]. Here they are given by :

$$den = [1, \quad -\Delta T\beta - 1]$$

The numerator relates to the coefficients in front of $y_d[n]$ and $y_d[n-1]$. Here they are given by:

$$num = [0, \quad \Delta T\gamma K_{ff}]$$

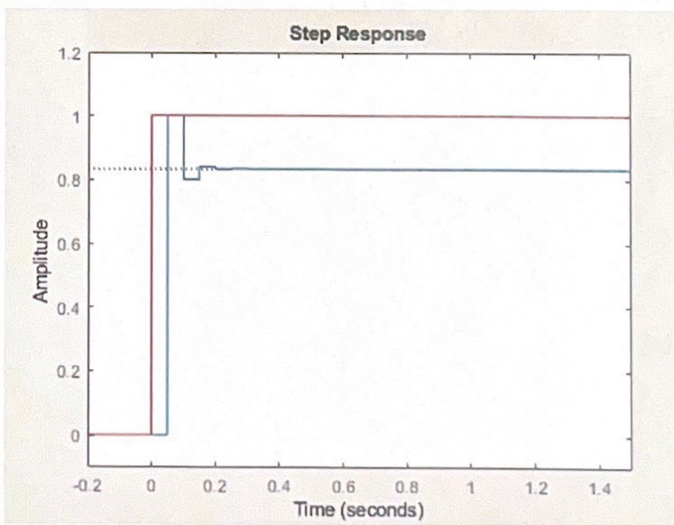
We can easily modify this code to study proportional feedback control. Under feedback control, the system equation is given by:

$$y[n] - y[n-1]\left(1 + \Delta T\beta - \Delta T\gamma K_p\right) = \gamma\Delta T K_p y_d[n-1]$$

Now we only need to change the denominator and numerator lines:

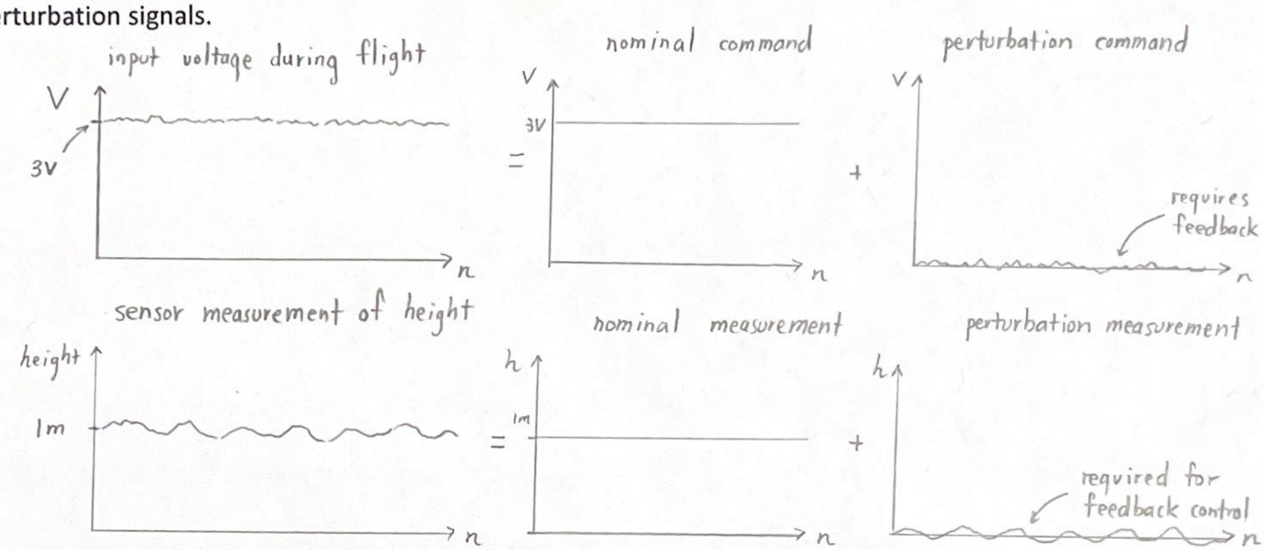
```
%change to proportional controller
den = [1, -(1+dt*beta)+dt*gamma*Kp];
num = [0 dt*gamma*Kp];
```

We can re-run the simulation and get the following graph:

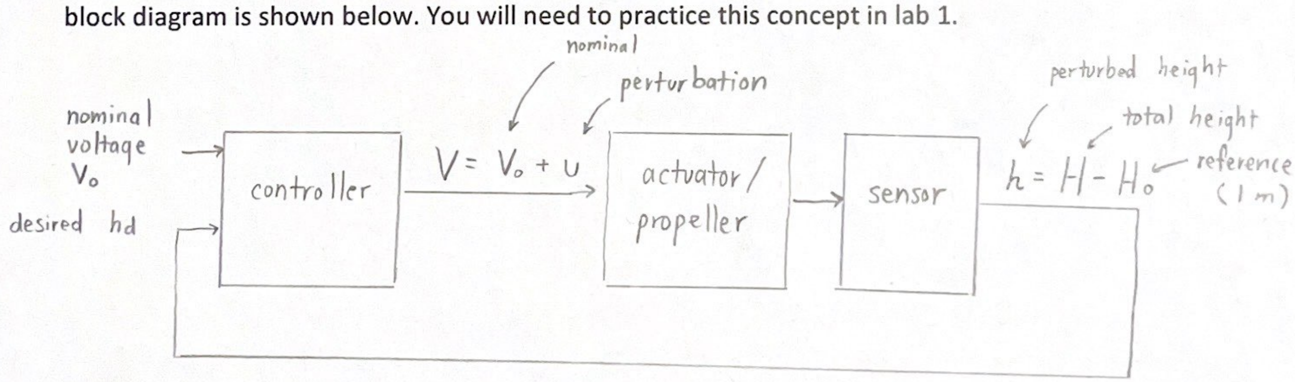


3. Nominal and perturbation control signals

Often time we control a system relative to an equilibrium state. For example, consider a quadrotor that is flying, and we want to control it to hover around a setpoint 1 m above ground. We have an altitude sensor that measures instantaneous height. We can view this problem as controlling a system near an equilibrium state, where there is a large nominal input command and a large nominal altitude setpoint. These nominal commands (driving voltage) and sensor output (height) do not require feedback control. What we are interested in are the "perturbation" control signals and sensor output (deviation from the setpoint height), which we will use feedback control. The two graphs below illustrate the relationship between nominal and perturbation signals.

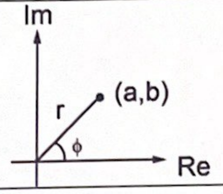


When we draw block diagrams, we need to specify which are the nominal quantities and which are the perturbation quantities we control. For this simple quadrotor example, the corresponding block diagram is shown below. You will need to practice this concept in lab 1.



4. Review of complex numbers

Complex numbers are mathematical tools for us to analyze high order DT problems (later we will also use complex numbers to analyze continuous time problems). Here is a quick summary of complex numbers. A complex number consists of a real part a and an imaginary part b. It can be thought as a tuple of two numbers (a, b), or it can be represented as a magnitude (r) and an angle (ϕ) in the complex plane.

	z = a + jb z = re$^{i\phi}$	$r = \sqrt{a^2 + b^2}$ $\phi = \tan^{-1}(b/a)$	$a = rcos\phi$ $b = rsin\phi$

In this course, we use j to denote the imaginary number. Note here are two important relations:

$$j^2 = -1 \text{ and } \frac{1}{j} = \frac{j}{j^2} = -j$$

We can use the polar form to evaluate the solution and analyze stability. Specifically, we have:

$$\lambda^n = (a + jb)^n = \left(re^{j\phi}\right)^n = r^n e^{jn\phi}$$

The phase $e^{jn\phi}$ has an amplitude of 1, and the term r^n determines whether the system is stable or unstable. Specifically, for the system to converge, we must impose $r < 1$. The key takeaway is that for the system to be stable, the amplitude of all natural frequencies must be less than 1.

Let's review an example introduced in lecture 2. Consider the step response of a control system where the natural frequency varies along the real-axis in the complex plane. Cases 1-7 correspond to the following time-domain solutions.