

Mathematics for Robotics and Control -

Assignment 4: Modelling in the Time Domain

```
In [3]: import sys
# Windows and Linux so far have no problem with the inline backend...
if not "win" in sys.platform and not "linux" in sys.platform:
    %pylab
# ...on other platforms we did run into problems in the past, so let's
else:
    %pylab inline
```

Welcome to pylab, a matplotlib-based Python environment [backend: module://IPython.kernel.zmq.pylab.backend_inline].
For more information, type 'help(pylab)'.

Assignment 4.1

Answer the following questions:

- I. What are some advantages of modelling a system in the time domain vs. the frequency domain?

Some functions are easier in their respective domains. Convolution in the time domain is multiplication in the frequency domain. Multiple functions can be combined (added or super imposed) in the time domain but they can be seperated into their frequency components and the information is seperable in the frequency domain. In the frequency domain the stability and transient response is quickly available. The time domain approach has the advantage that it can be applied to non Linear Time Invariant systems.

- II. Name two additional forms on how to represent a system in state space form, apart from the Controllable Canonical Form. Give reasons as to why you would pick each form and *briefly* describe their structure, i.e. how to obtain them.

i. Observable Canonical Form:

The Observable Canonical Form guarentees an 'Observable' system. Every state will have some effect on the output and therefor ever state/transition is detectable, the system can not move from one state to another without in some way effecting the output.

ii. Cascade Form / Parallel Form:

The Cascade or Parallel forms refer to how the system is represented in a block diagram. They are non canonical forms, but can be preferred depending on how the system is being implemented, or how the system being modelled is.

- III. Assume you have a linear algebra package installed on your computer and a system model in state space, i.e. the matrices A , B , C , and D . How can you quickly obtain the poles of

said system without converting it back to the frequency domain?

Very quick, MatLab provides this functionality through at least 3 different toolboxes (Control Systems Toolbox, System Identification Toolbox and Signal Processing Toolbox). The python class `statesp.StateSpace` has a method `pole()` which will "Compute the poles of a state space system."

```
In [6]: ?statesp.StateSpace.pole
Type:      instancemethod
String Form:
File:      /opt/anaconda/lib/python2.7/site-packages/control/statesp.py
Definition: statesp.StateSpace.pole(self)
Docstring: Compute the poles of a state space system.
```

- IV. Given a system model in SSR, how can you determine if the system is controllable? Construct a simple example and give some code to demonstrate.
- V. Like question 4, but now talk about observability.

Assignment 4.1 took me minutes.

Assignment 4.2

Take the system from the most recent lab class and...

- I. ...determine the transfer function that represents the whole system

First we look at the inner loop containing, $G_1 = \frac{5}{s(1+2)}$ and negative self feedback. This simplifies to a new block N_1 .

$N_1 =$

- II. ...convert it to a state space representation, use the desired angle as input and the actual angle and wheel position as state variables.
- III. ...simulate the system *manually* in Python, assuming $x_0 = 0$, using the following angle values as inputs:



- I. ...during your simulation, pick K so as to create an unstable, underdamped, overdamped and critically damped system. Include your results for each choice of K .

```
In [5]: import numpy as np
u_t = np.concatenate([
    [0.]*10,
    [-np.pi*0.5]*1,
    [0.]*10,
    [np.pi*0.5]*1,
```

```

    [0]*8,
    [np.pi*0.5]*1,
    [0.]*20,
    [-np.pi*0.5]*1,
    [0.]*15,
    [-np.pi*0.5]*1,
    [0.]*15,
    ])
print u_t

```

```

[ 0.      0.      0.      0.      0.      0.
 0.
 0.      0.      0.     -1.57079633  0.      0.
 0.
 0.      0.      0.      0.      0.      0.
 0.
 1.57079633  0.      0.      0.      0.      0.
 0.
 0.      0.      1.57079633  0.      0.      0.
 0.
 0.      0.      0.      0.      0.      0.
 0.
 0.      0.      0.      0.      0.      0.
 0.
 0.      0.     -1.57079633  0.      0.      0.
 0.
 0.      0.      0.      0.      0.      0.
 0.
 0.      0.      0.      0.     -1.57079633  0.
 0.
 0.      0.      0.      0.      0.      0.
 0.
 0.      0.      0.      0.      0.      0.]

```

The submission for this assignment is the generated system output. Be sure to include it in your submission!

Assignment 4.2 took me minutes.

Assignment 4.3

Last week, we discussed how a Pole/Zero - plot allows you to quickly judge the stability of a system. What plot could you use to inspect the damping properties of second-order differential equations? Find and explain plots for the underdamped, overdamped and critically damped case.

You can use a plot of the system response vs time. Or just examine the roots in the Pole Zero plot directly. The Critically damped case will reach a steady state end value in a reasonable time without over shooting. the underdamped case oscillate about the steady state value and may converge or diverge. The overdamped case may not reach the steady state, or will do so very slowly.



source: <http://lpsa.swarthmore.edu/Transient/TransInputs/TransStep.html>

Assignment 4.3 took me minutes.

Assignment 4.4

Convert the following diagrams to a single transfer function representing the complete system.

System A 

System B 

**** System A ****

No example given in lecture, lab or notes provided how to deal with overlapping loops.

My solution is on pen and paper but the result involved moving the G_4 around so the loops didn't overlap. Here is answer, see attached image for work.

$$H(s) = \frac{G_1 G_2 G_6 (G_5 + G_3 G_4)}{1 + H_1 G_4 G_6 + H_2 G_1 G_2 (G_3 G_4 + G_5) G_6}$$

**** System B ****

Assume $D(s) = 0$. And that $D(s)$ is a disturbance signal.

Create new block $N_1(s)$, the numerator of the feedback loop.

$$N_1(s) = \left(\frac{K}{s+3} + D(s) \right) \frac{2}{s}$$

Then combine the terms of the feedback loop

$$H(s) = \frac{N_1(s)}{1 + (s+1)(N_1(s))}$$

Set $D(s) = 0$, and simplify.

$$H(s) = \frac{2Ks^2 + 6Ks}{s^4 + 6s^3 + 9s^2 + 4K^2}$$

Assignment 4.4 took me minutes.

Use this button to create a .txt file containing the time in minutes you spent working on the assignments. Make sure to include your name in the textbox below. The file will be created in the current directory.

Student's name:

Save timings

