Exercise 5: Time delay and Pade’ approximations (Solutions)

Time-delays are very common in control systems. The Transfer function of a time-delay is defined as:

\[ H(s) = e^{-ts} \]

In some situations it is necessary to substitute \( e^{-ts} \) with an approximation, e.g., the Padé-approximation:

\[
e^{-ts} \approx \frac{1 - k_1 s + k_2 s^2 + \cdots \pm k_n s^n}{1 + k_1 s + k_2 s^2 + \cdots + k_n s^n}
\]

i.e., we need to have the system on transfer function form.

1. order system with time-delay:

A 1.order transfer function with time-delay may be written as:

\[ H(s) = \frac{K}{Ts + 1} e^{-ts} \]

Step Response:

A step response for a 1.order system with time delay has the following characteristics:

MathScript has a built-in function called `pade` for creating transfer functions for time-delays:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pade</code></td>
<td>Incorporates time delays into a system model using the Padé</td>
<td><code>&gt;[num, den] = pade(delay, order)</code></td>
</tr>
</tbody>
</table>
approximation method, which converts all residuals. You must specify the delay using the set function. You also can use this function to calculate coefficients of numerator and denominator polynomial functions with a specified delay.

\[ [A, B, C, D] = \text{pade}(\text{delay}, \text{order}) \]

<table>
<thead>
<tr>
<th>Sys_order1</th>
<th>( K=4; \ T=3; \ \text{delay}=5; )</th>
</tr>
</thead>
<tbody>
<tr>
<td>set</td>
<td>( H = \text{sys}_\text{order1}(K, T, \text{delay}) )</td>
</tr>
<tr>
<td>series</td>
<td>( H = \text{set}(H_1, \text{'inputdelay'}, \text{delay}); )</td>
</tr>
<tr>
<td></td>
<td>( H = \text{series}(H_1, H_2); )</td>
</tr>
</tbody>
</table>

### Task 1: Time delay using approximation

Given the following system:

\[ H(s) = e^{-2s} \]

### Task 1.1

Create Pade’ approximations for this system using approximations with different orders. Use approximations with order 1, 2, 3, 4 and 10 using the built-in `pade` function in MathScript.

Plot the step response for these approximations.

Discuss the results.

**Solution:**

MathScript code:

\[
\begin{align*}
t &= [0:0.1:10] \\
\text{delay} &= 2; \\
\text{n} &= 1; \\
H_1 &= \text{pade}(\text{delay}, \text{n}) \\
\text{n} &= 2; \\
H_2 &= \text{pade}(\text{delay}, \text{n}) \\
\text{n} &= 3; \\
H_3 &= \text{pade}(\text{delay}, \text{n}) \\
\text{n} &= 4; \\
H_4 &= \text{pade}(\text{delay}, \text{n}) \\
\text{n} &= 10; \\
H_5 &= \text{pade}(\text{delay}, \text{n}) \\
\text{step}(H_1, H_2, H_3, H_4, H_5, t)
\end{align*}
\]

Step response:
A more elegant solution would be to use a “for loop” to implement this:

```matlab
clear, clc
t = 0:0.1:10;
delay = 2;
n = [1,2,3,4,10];
colors = ['b', 'c', 'g', 'k', 'r'];
N = length(n);
for i=1:N
    H = pade(delay, n(i));
```
Which gives the same results.

**Task 1.2**

Plot the step response for the exact solution \( H(s) = e^{-2s} \) using “pen and paper”.

You can also find the exact step response in MathScript using e.g., the `sys_order1` function. Compare the results.

Compare the exact solution with an approximation of order 50.

Discuss the results.

**Solution:**

Using \( n = 50 \) gives:

MathScript code:
```
clear, clc
t = 0:0.1:10;
delay = 2;
n = 50;
H = pade(delay, n);
step(H, t)
```

The “exact” is when \( n \to \infty \). We can do it as follows in MathScript:
clear, clc
t = 0:0.1:10;
delay = 2;

% Pade Approximation
n=50,
H_pade = pade(delay, n);
step(H_pade, t)

% Exact Solution
H_exact = sys_order1(1,0,delay)
hold on
step(H_exact, 'r', t)

This gives:

![Plot1](image)

**Discussion:** \( n \) specifies the order of the Pade approximation polynomial functions. A higher order results in a more accurate approximation of the delay but also increases the order of the resulting model. A large order can make the model too complex to be useful!!

**Task 2: 1.order system with delay**

Given a 1.order transfer function with time-delay:

\[
H(s) = \frac{K}{Ts + 1} e^{-\tau s}
\]

Where \( K = 4, T = 2, \tau = 3 \), i.e.:
Task 2.1

Draw a sketch of the step response for this system using “pen and paper”.

Solution:

Sketch of the step response for the system:

Task 2.2

Find the step response for this system using MathScript using a 5.order approximation and the \texttt{pade} function.

Compare the results with the exact solution (using e.g., the \texttt{sys\_order1} function).

Solution:

MathScript Code:

```matlab
clear
clc
t=0:0.1:10;
K = 4;
T = 2;
delay = 3;

% Method 1 - Exact solution--------------
H = sys_order1(K, T, delay)
figure(1)
```
step(H, t)

% Method 2 - Solution with Pade'

num = [K];
den = [T, 1];
H1 = tf(num, den)

n=5
H2 = pade(delay, n)
H_usingpade = series(H1, H2)

figure(2)
step(H_usingpade, t)

This gives the following results:

Note!

When using the approximation, we need to split the function in 2 different parts:

\[ H_1(s) = \frac{K}{Ts + 1} \]

\[ H_2(s) = e^{-Ts} \]

Where we only use the \texttt{pade} function for \( H_2 \), while for \( H_1 \) we can use the ordinary \texttt{tf} function. Finally we can set them together using the \texttt{series} function in MathScript:

\[ H(s) = H_1(s) H_2(s) = \frac{K}{Ts + 1} e^{-Ts} \]

**Task 3: Pade’ Approximation**
Note! In this task we shall note use the built-in pade function, but create our own approximation using the definition itself:

\[ e^{-ts} \approx \frac{1 - k_1 s + k_2 s^2 + \ldots \pm k_n s^n}{1 + k_1 s + k_2 s^2 + \ldots + k_n s^n} \]

where:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( n = 1 )</td>
<td>( n = 2 )</td>
</tr>
<tr>
<td>( k_1 = \frac{\tau}{2}, \text{ other } k_i = 0 )</td>
<td>( k_1 = \frac{\tau}{2}, k_2 = \frac{\tau}{12}, \text{ other } k_i = 0 )</td>
</tr>
</tbody>
</table>

**Task 3.1**

Set up the mathematical expressions, i.e., find the transfer functions for a 1.order and 2.order Pade’-approximation (pen & paper).

Set the time-delay \( \tau = 5 \), i.e.

\[ H(s) = e^{-5s} \]

**Solution:**

We have:

\[ e^{-ts} \approx \frac{1 - k_1 s + k_2 s^2 + \ldots \pm k_n s^n}{1 + k_1 s + k_2 s^2 + \ldots + k_n s^n} \]

Where:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( n = 1 )</td>
<td>( n = 2 )</td>
</tr>
<tr>
<td>( k_1 = \frac{\tau}{2}, \text{ other } k_i = 0 )</td>
<td>( k_1 = \frac{\tau}{2}, k_2 = \frac{\tau}{12}, \text{ other } k_i = 0 )</td>
</tr>
</tbody>
</table>

**For a 1.order Pade’ approximation \((n = 1)\) we get the following transfer function:**

We get:

\[ e^{-ts} \approx \frac{1 - k_1 s}{1 + k_1 s} \]

Where:

\[ k_1 = \frac{\tau}{2} = \frac{5}{2} = 2.5 \]

Then we get:

\[ e^{-ts} \approx H_{\text{pade}}(s) = \frac{1 - k_1 s}{1 + k_1 s} = \frac{1 - \frac{5}{2} s}{1 + \frac{5}{2} s} = \frac{1 - 2.5 s}{1 + 2.5 s} \]

**For a 2.order Pade’ approximation \((n = 2)\) we get the following transfer function:**
We get:

\[ e^{-ts} \approx \frac{1 - k_1 s + k_2 s^2}{1 + k_1 s + k_2 s^2} \]

Where:

\[ k_1 = \frac{\tau}{2} = \frac{5}{2} = 2.5 \]
\[ k_2 = \frac{\tau^2}{12} = \frac{25}{12} = 2.083 \]

Then we get:

\[ e^{-ts} \approx H_{pade}(s) = \frac{1 - k_1 s + k_2 s^2}{1 + k_1 s + k_2 s^2} = \frac{1 - \frac{\tau}{2} s + \frac{\tau^2}{12} s^2}{1 + \frac{\tau}{2} s + \frac{\tau^2}{12} s^2} = \frac{1 - 2.5s + 2.083s^2}{1 + 2.5s + 2.083s^2} \]

**Task 3.2**

Define the transfer function for a 1.order and 2.order Pade’-approximation using the tf function in MathScript.

Use the step function to plot the responses.

**Solution:**

MathScript Code:

```matlab
clear, clc
t=0:0.1:10;
delay=5;

% 1.order approx. using tf function
k1=delay/2;
num=[-k1, 1];
den=[k1, 1];
H2=tf(num, den)
figure(1)
step(H2,t)
title('1.order approx using tf function')

% 2.order approx. using tf function
k1=delay/2;
k2=delay^2/12;
num=[k2, -k1, 1];
den=[k2, k1, 1];
H2=tf(num, den)
figure(3)
```

---

EE4107 - Cybernetics Advanced
Note!

1. order Pade’ is given by:

\[ e^{-ts} \approx H_{pade1}(s) = \frac{1 - k_1 s}{1 + k_1 s} \]

To use it in MathScript we have to have it on the following form:

\[ e^{-ts} \approx H_{pade1}(s) = \frac{-k_1 s + 1}{k_1 s + 1} \]

For 2. order Pade’:

\[ e^{-ts} \approx H_{pade2}(s) = \frac{1 - k_1 s + k_2 s^2}{1 + k_1 s + k_2 s^2} \]

In order to use in MathScript:

\[ e^{-ts} \approx H_{pade2}(s) = \frac{k_2 s^2 - k_1 s + 1}{k_2 s^2 + k_1 s + 1} \]

The results become:

<table>
<thead>
<tr>
<th>1. order approximation:</th>
<th>2. order approximation:</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Graph" /></td>
<td><img src="image2.png" alt="Graph" /></td>
</tr>
</tbody>
</table>

Task 3.3

Do you get the same results using the `pade` function? Discuss the results.

Solution:

We compare with using the `pade` function in MathScript:

```matlab
clear, clc
t=0:0.1:10
```
delay=5;

% 1.order approx. using tf
k1=delay/2;
num=[-k1, 1];
den=[k1, 1];

H2=tf(num, den)
figure(1)
step(H2,t)
title('1.order approx using tf function')

% 1.order using pade
n=1;
H2_pade = pade (delay,n)

figure(2)
step(H2_pade,t)
title('1.order approx using pade function')

% 2.order approx. using tf
k1=delay/2;
k2=delay^2/12;
num=[k2, -k1, 1];
den=[k2, k1, 1];

H2=tf(num, den)
figure(3)
step(H2,t)
title('2.order approx using tf function')

% 2.order using pade
n=2;
H2_pade = pade (delay,n)

figure(4)
step(H2_pade,t)
title('2.order approx using pade function')

We get the same results.

**Additional Resources**

- [http://home.hit.no/~hansha/?lab=mthscript](http://home.hit.no/~hansha/?lab=mthscript)

Here you will find tutorials, additional exercises, etc.