Time response of an RCR circuit subject to a sinusoidal pulse train (Laplace)

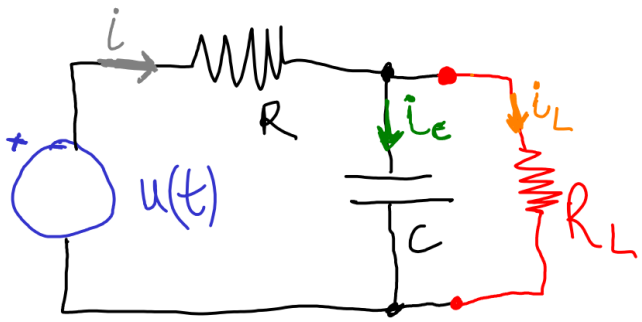
*This code was executed on Matlab **R2022b** (Linux)

Presentation in video: http://personales.upv.es/asala/YT/V/trenpulRCREN.html

Objectives: Model in state space and in transfer function form (plus initial conditions term) an electrical circuit. Obtain its time response to a repetated train of sinusoidal pulses as input.

Table of Contents

System Modelling (state space and transfer function)



These are the equations governing the above circuit:

$$\frac{dV_c}{dt}=\frac{1}{C}i_c,\quad i_c=i-i_L,\quad i=\frac{1}{R}(u-V_c),\quad i_L=\frac{1}{R_L}V_c$$

We may write normalised linear state and output equations as:

$$\frac{dV_c}{dt} = -\left(\frac{1}{RC} + \frac{1}{R_L C}\right) \cdot V_c + \frac{1}{RC} u$$

$$y = V_c$$

Let us give the parameters some numerical values:

```
R=0.5;RL=20;C=2e-2;
A=-1/(R*C)-1/(RL*C)
```

```
A = -102.5000
```

```
B=1/(R*C); C=1; D=0; %"ss"
```

The Laplace domain model will be:

$$Y(s) = (C(sI - A)^{-1}B + D) \cdot U(s) + C(sI - A)^{-1} \cdot x(0)$$

```
syms s
TransferFunction=simplifyFraction(C*1/(s-A)*B+D)
```

```
TransferFunction =
```

$$\frac{200}{2\,s + 205}$$

```
TCI=simplifyFraction(C*1/(s-A)) %this multiplies initial capacitor voltage
V_c(0)
```

```
TCI =
```

$$\frac{2}{2\,s + 205}$$

Single sinusoidal pulse and its Laplace transform

Time-domain input pulse

The sinusoidal pulse (a single semiperiod π/ω) is:

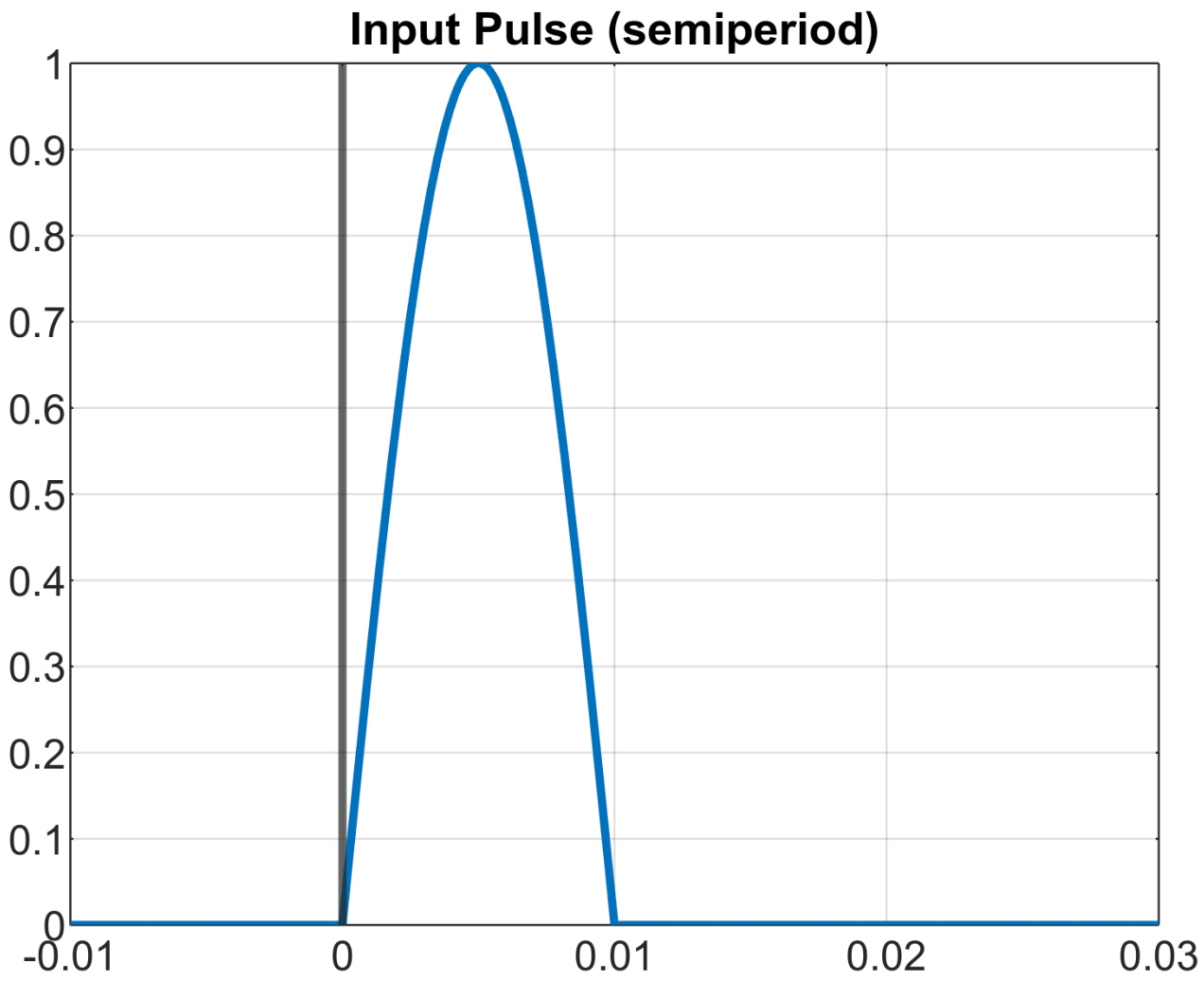
$$u(t) = \begin{cases} 0 & t < 0 \\ \sin(\omega t) & 0 \le t \le \frac{\pi}{\omega} \\ 0 & t > \frac{\pi}{\omega} \end{cases}$$

Numerically, we'll work with a single 50Hz semi-period pulse.

```
omega=100*pi;
SemiPeriod=pi/omega
```

```
SemiPeriod = 0.0100
```

```
syms t real
pulse(t)= heaviside(t)*heaviside(SemiPeriod-t)*sin(omega*t);
fplot(pulse,[-0.01 0.03],LineWidth=2), grid on, title("Input Pulse
(semiperiod)"),xline(0,LineWidth=2)
```



Pulses in the Laplace domain

As the pulse coincides with a sinusoid in the first 0.01 seconds, we will use such signal (plus causality-related argumentations) later on.

```
u1(t)=sin(omega*t);
U1s=laplace(u1) %a sinusoidal wave starting at t=0   omega/(s^2+omega^2)
```

```
U1s =
```

$$\frac{100\,\pi}{s^2 + 10000\,\pi^2}$$

The single pulse transform is quite similar:

```
Pulse(s)=laplace(pulse(t))
```

```
Pulse(s) =
```

$$\frac{100\,\pi\,\left(\mathrm{e}^{-\frac{s}{100}}+1\right)}{s^2+10000\,\pi^2}$$

Pulse train in the laplace domain

As the pulse is zero exept in [0,0.01], letting $T = 0.01$, the train of pulses is

$$Train_{pulses}(t) = pulse(t) + pulse(t-T) + pulse(t-2T) + \ldots$$

If $Pulse(s)$ is the Laplace transform of the single pulse, the pulse train's transform will be:

$$Train_{pulses}(s) = pulse(s)\cdot(1 + e^{-Ts} + e^{-2Ts} + \ldots) = U(s)\frac{1}{1-e^{-Ts}}.$$

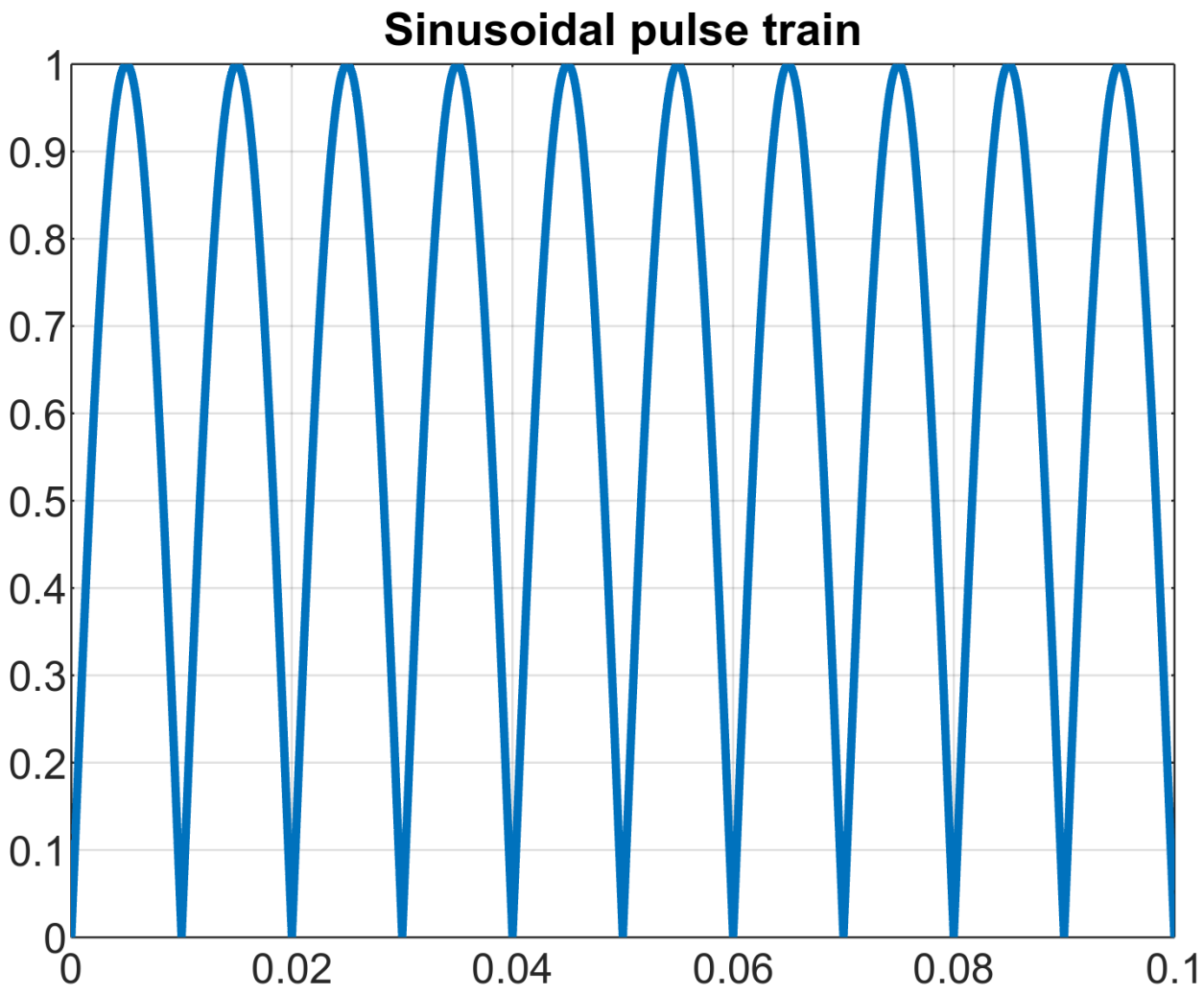
*The sum of a geometric progression formula was used.

```
Train(s)=Pulse(s)/(1-exp(-1/100*s))
```

```
Train(s) =
```

$$-\frac{100\,\pi\,\left(\mathrm{e}^{-\frac{s}{100}}+1\right)}{(s^2+10000\,\pi^2)\,\left(\mathrm{e}^{-\frac{s}{100}}-1\right)}$$

```
fplot(ilaplace(Train),[0 0.1],LineWidth=2), grid on, title("Sinusoidal pulse
train")
```



Time response to this pulse makes Matlab a bit crazy:

```
Y_try1=ilaplace(TransferFunction*Train(s))
```

```
Y_try1 =
```

$$-20000\,\pi\left(\frac{41\left(\frac{\sigma_5\,(\sigma_1-1)\,\mathrm{i}}{400\,\pi}-\frac{\mathrm{e}^{100\,\pi\,t\,\mathrm{i}}\,(\sigma_1-1)\,\mathrm{i}}{400\,\pi}\right)}{5\,\sigma_8}+\frac{\sigma_5\left(\sigma_3-\sigma_1\,\sigma_3-\sigma_1\left(\left(\frac{\partial}{\partial s}\,\lfloor s\rfloor\right)\Big|_{s=-100\,t}\right)\right)-\sigma_1+\sigma_1\,\sigma}{50\,\sigma_8}\right.$$

where

$$\sigma_1 = (-1)^{\lfloor -100\,t\rfloor}$$

$$\sigma_2 = \mathrm{e}^{200\,\pi\,\left(t-\frac{1}{100}\right)\mathrm{i}}$$

$$\sigma_3 = \mathrm{e}^{200\,\pi\,t\,\mathrm{i}}$$

$$\sigma_4 = 100\,\pi\,\left(t-\frac{1}{100}\right)\mathrm{i}$$

$$\sigma_5 = \mathrm{e}^{-100\,\pi\,t\,\mathrm{i}}$$

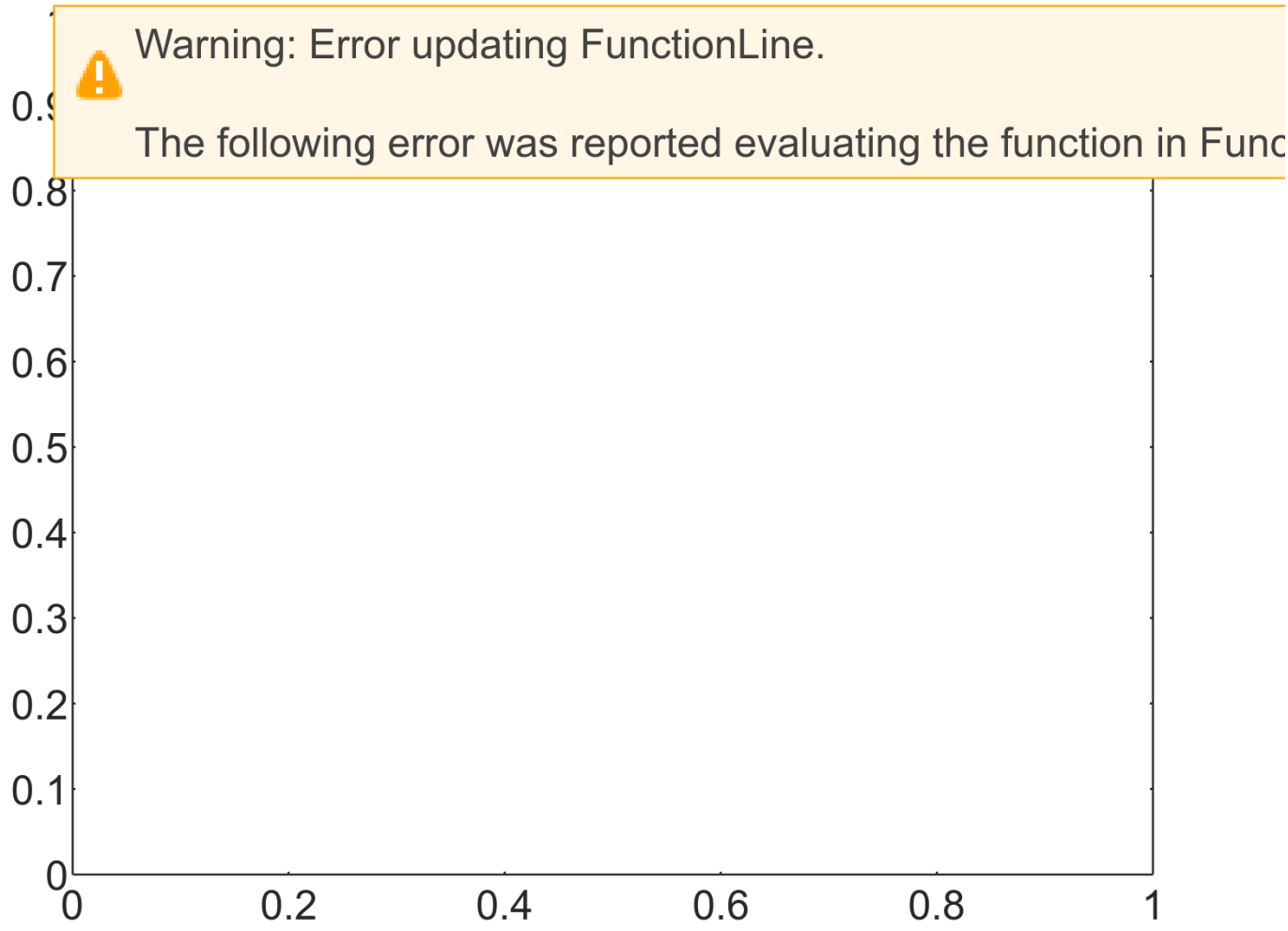
$$\sigma_6 = 25\,\sigma_8\,\left(\mathrm{e}^{41/40}-1\right)$$

$$\sigma_7 = (-1)^{\sigma_9}-1$$

$$\sigma_8 = 1600\,\pi^2+1681$$

$$\sigma_9 = \lfloor 1-100\,t\rfloor$$

```
fplot(Y_try1)
```



So, we'll do it by hand.

[RECAP] Time response to single pulse , via causality/state piecewise approach

Response to nonzero initial conditions

Initial capacitor voltage:

```
V0=0.95;
```

The time response to the whole sinusoid (infinite duration) and given initial conditions will be identical to the pulse response we are seeking, as causal systems cannot depend on "future" input waveform features:

```
Y1=TransferFunction*U1s;
```

```
%time response to sinusoid with zero initial conditions V0.
vpa(partfrac(Y1,FactorMode='real'),5)
```

```
ans =
```

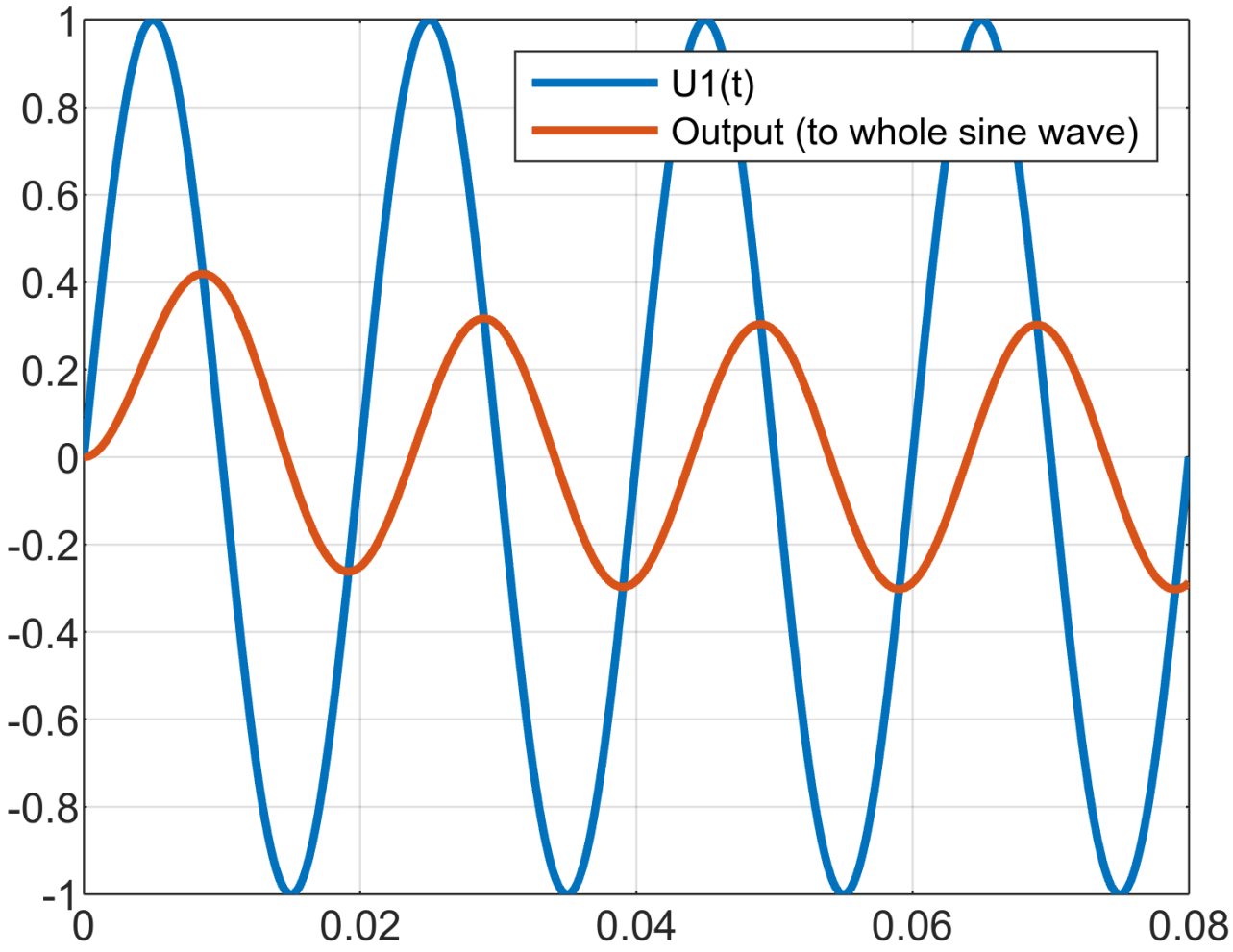
$$\frac{0.57537}{2.0\,s + 205.0} - \frac{0.28769\,s - 29.488}{s^2 + 98696.0}$$

```
Y1_t(t)=ilaplace(Y1); vpa(Y1_t,4)
```

```
ans(t) =
```

$$0.09386\sin(314.2\,t) - 0.2877\cos(314.2\,t) + 0.2877\,e^{-102.5\,t}$$

```
Y1_tIC=Y1_t+ilaplace(TCI*V0); %plug in in the initial condition effect.
fplot([u1;Y1_t],[0 0.08],LineWidth=2), grid on, legend("U1(t)","Output (to
whole sine wave)")
```



As we have a change in the input "formula" at $t = 0.01$, switching to "zero", the above solution is only valid in the first 0.01 seconds, being meaningless from that time onwards. For $t \geq 0.01$ we will have a "free" response (zero input) from the initial conditions that the first semi-period have left the system at:

```
CondIniFragment2=vpa(Y1_tIC(0.01),6)
```

$$\texttt{CondIniFragment2} = 0.731763$$

Obviously, if we were simulating a system with order >1, we would need the output and a certain number of its derivatives at $t = 0.01$, or the whole state at that instant (not just the output). In this case, for simplicity, we don't need that because it's fisrt order stuff.

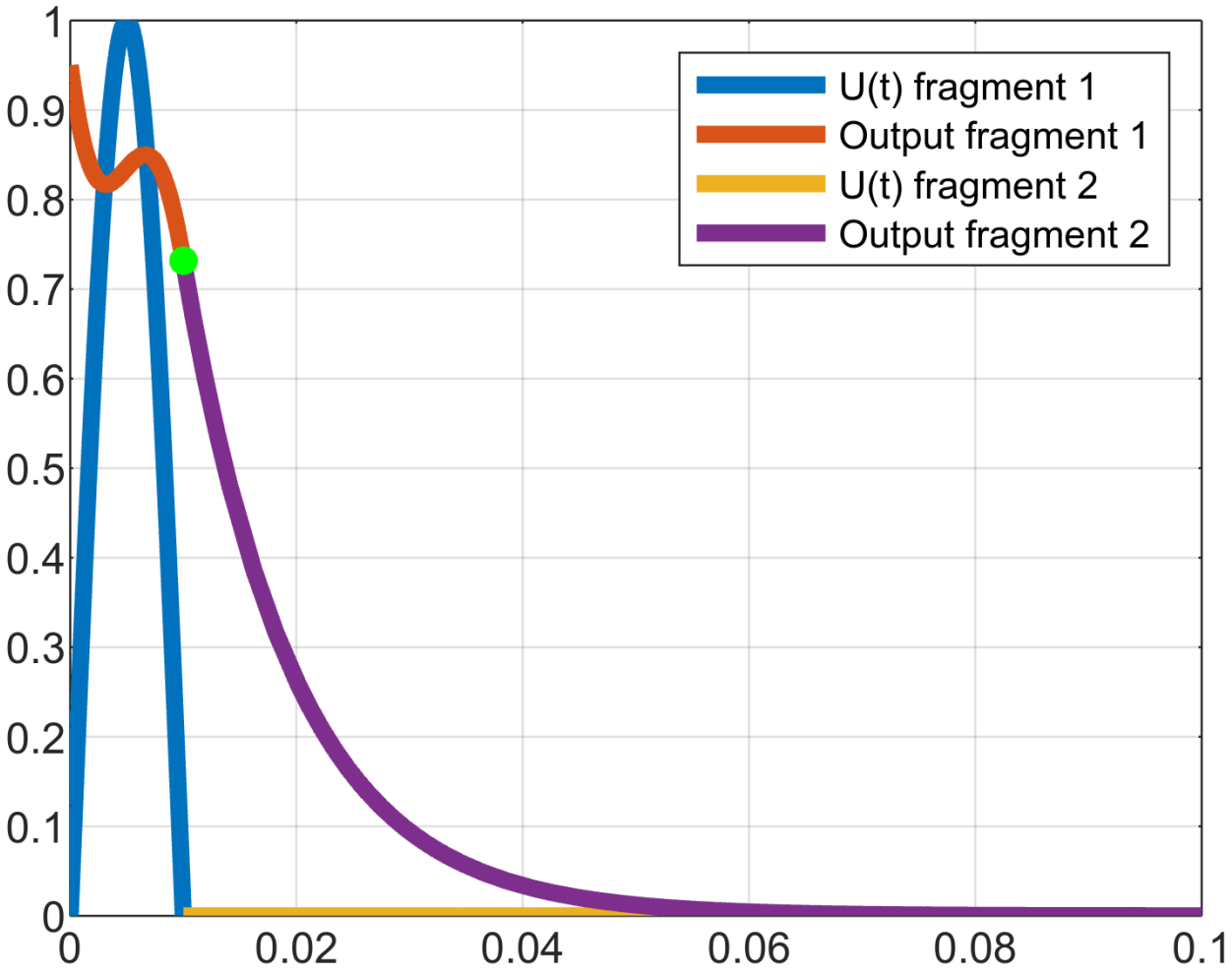
The free response with "initial" conditions equal to the "final" ones of the previous fragment is:

```
syms t_new %new time variable
Y_t_Fragment2(t_new)=ilaplace(TCI*CondIniFragment2,t_new);
vpa(Y_t_Fragment2,5)
```

$$\texttt{ans(t\_new)} = 0.73176\,\mathrm{e}^{-102.5\,t_{\mathrm{new}}}$$

But, beware, actually "zero" of this new clock is 0.01 of the "official" timing; so we need to be careful when writing/plotting what we just obtained:

```
fplot([u1;Y1_tIC],[0 0.01],LineWidth=4), hold on
fplot([0;Y_t_Fragment2(t-0.01)],[0.01 0.1],LineWidth=4),
plot(0.01,CondIniFragment2,'.g',MarkerSize=20)
hold off, grid on
legend("U(t) fragment 1","Output fragment 1","U(t) fragment 2","Output
fragment 2")
```



Of course, both methods give the same solution, as expected.

Full response to pulse train

It would be done in pieces... Seeing in what condition "each pulse" ends and repeating calculations for the "next pulse" in the I.C. in which the said pulse ended.

Causality: Since a "pulse" coincides with the "sine" in the first 0.01 seconds, the responses in that interval will be identical. We will work with the response Y1_t with the sine input and not with the response before the complete pulse.

First fragment will have initial conditions equal to zero, by assumption:

```
Yfragment1(t)=Y1_t(t); vpa(Yfragment1(t),4) %Y1_t is response to
sin(100·pi·t) and zero initial cond.
```

ans = $0.09386\sin(314.2\,t) - 0.2877\cos(314.2\,t) + 0.2877\,e^{-102.5\,t}$

```
V1=Yfragment1(0.01); vpa(V1,4) %final conditions of fragment 1.
```

ans = 0.3909

Second fragment will have initial conditions equal to the "final" conditions of fragment one, so we'll add the free response term:

```
Yfragment2(t)=Y1_t(t)+ilaplace(V1*TCI); vpa(Yfragment2(t),4)
```

ans = $0.09386\sin(314.2\,t) - 0.2877\cos(314.2\,t) + 0.6786\,e^{-102.5\,t}$

```
V2=Yfragment2(0.01); vpa(V2,4)
```

ans = 0.5312

Third fragment will have initial conditions equal to the "final" conditions of fragment 2... and so on

```
Yfragment3(t)=Y1_t(t)+ilaplace(V2*TCI);
V3=Yfragment3(0.01); vpa(V3,4)
```

ans = 0.5815

```
Yfragment4(t)=Y1_t(t)+ilaplace(V3*TCI);
V4=Yfragment4(0.01); vpa(V4,4)
```

ans = 0.5995

```
Yfragment5(t)=Y1_t(t)+ilaplace(V4*TCI);
V5=Yfragment5(0.01); vpa(V5,4)
```

ans = 0.606

```
Yfragment6(t)=Y1_t(t)+ilaplace(V5*TCI);
V6=Yfragment6(0.01); vpa(V6,4)
```

ans = 0.6083

```
Yfragment7(t)=Y1_t(t)+ilaplace(V6*TCI);
V7=Yfragment7(0.01); vpa(V7,4)
```

ans = 0.6092

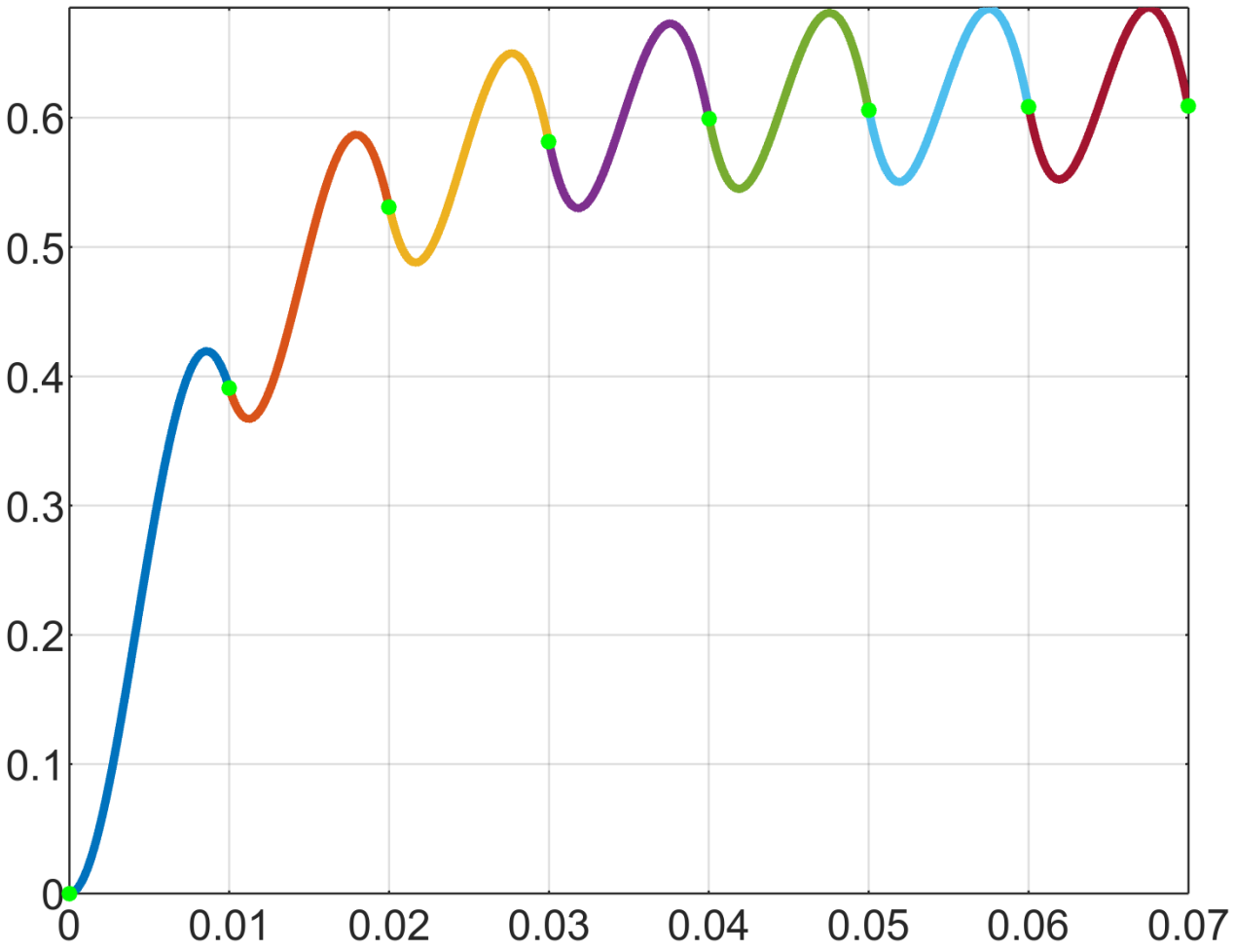
```
Yfragment8(t)=Y1_t(t)+ilaplace(V7*TCI);
V8=Yfragment8(0.01); vpa(V8,4) %a "for" loop would be a smarter way of doing
this, I know. Avoid copy-paste in production code.
```

ans = 0.6095

This does converge to a non-sinusoidal (input isn't a sinusoid so there will be harmonics) steady-state regime we will compute later on.

The plot is:

```
fplot(Yfragment1(t),[0 0.01],LineWidth=2), grid on, hold on
fplot(Yfragment2(t-.01),[0.01 0.02],LineWidth=2),
fplot(Yfragment3(t-.02),[0.02 0.03],LineWidth=2),
fplot(Yfragment4(t-.03),[0.03 0.04],LineWidth=2),
fplot(Yfragment5(t-.04),[0.04 0.05],LineWidth=2),
fplot(Yfragment6(t-.05),[0.05 0.06],LineWidth=2),
fplot(Yfragment7(t-.06),[0.06 0.07],LineWidth=2),
plot(0:0.01:0.07,[0 V1 V2 V3 V4 V5 V6 V7],'.g',MarkerSize=11)
hold off
```



Steady state regime (periodic but non-sinusoidal) to pulse train

Each "cycle" must start in the same initial tension as it finished.

We'll name such steady-state tension at each 0.01 seconds as V_{ss}:

```
syms V_ss real
Yperm(t)=Y1_t(t)+ilaplace(TCI*V_ss) ;
vpa(Yperm(t),4)
```

ans = $0.09386\sin(314.2\,t) - 0.2877\cos(314.2\,t) + 0.2877\,\mathrm{e}^{-102.5\,t} + V_{ss}\,\mathrm{e}^{-102.5\,t}$

```
vpa(Yperm(0),4)
```

ans = V_{ss}

```
vpa(Yperm(0.01),4)
```

ans = $0.3588\,V_{ss} + 0.3909$

```
sol=vpa(solve(Yperm(0.01)==Yperm(0)),6)
```

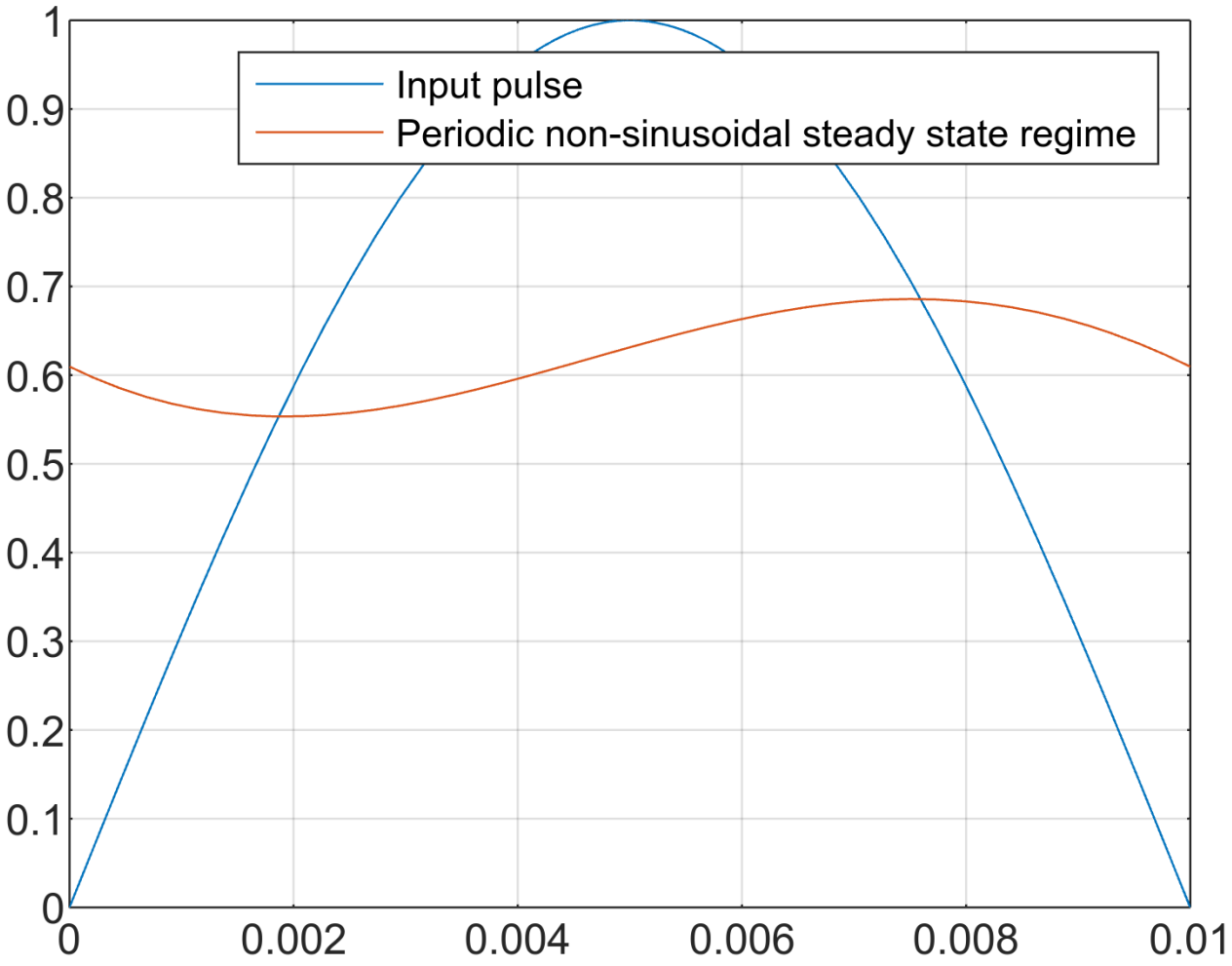
sol = 0.609644

So, replacing this value, we obtain the final steady-state periodic regime:

```
Yfinal(t)=subs(Yperm(t),V_ss,sol);
vpa(Yfinal,4)
```

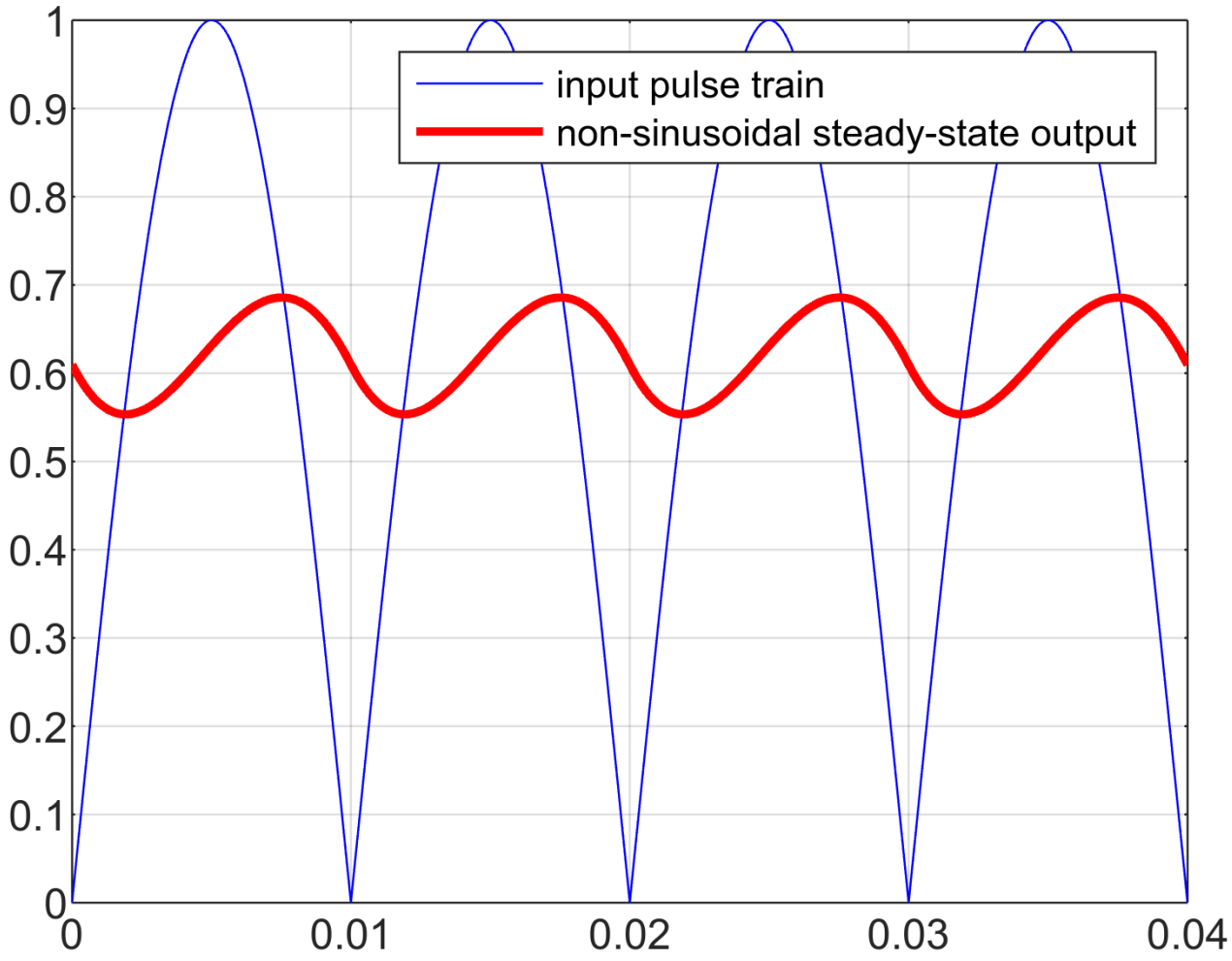
ans(t) = $0.09386\sin(314.2\,t) - 0.2877\cos(314.2\,t) + 0.8973\,\mathrm{e}^{-102.5\,t}$

```
fplot([u1; Yfinal],[0 0.01]), grid on, legend("Input pulse","Periodic non-
sinusoidal steady state regime")
```



Once reached, this periodic steady state would repeat forever:

```
fplot(u1,[0 0.01],Color="blue"), grid on
hold on
fplot(Yfinal,[0 0.01],LineWidth=2,Color="red"), grid on
fplot(u1(t-0.01),[0.01 0.02],Color="blue"), grid on
fplot(u1(t-0.02),[0.02 0.03],Color="blue"), grid on
fplot(u1(t-0.03),[0.03 0.04],Color="blue"), grid on
fplot(Yfinal(t-0.01),[0.01 0.02],LineWidth=2,Color="red"), grid on
fplot(Yfinal(t-0.02),[0.02 0.03],LineWidth=2,Color="red"), grid on
fplot(Yfinal(t-0.03),[0.03 0.04],LineWidth=2,Color="red"), grid on
hold off, legend("input pulse train", "non-sinusoidal steady-state output")
```


input pulse train
non-sinusoidal steady-state output
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.01
0.02
0.03
0.04