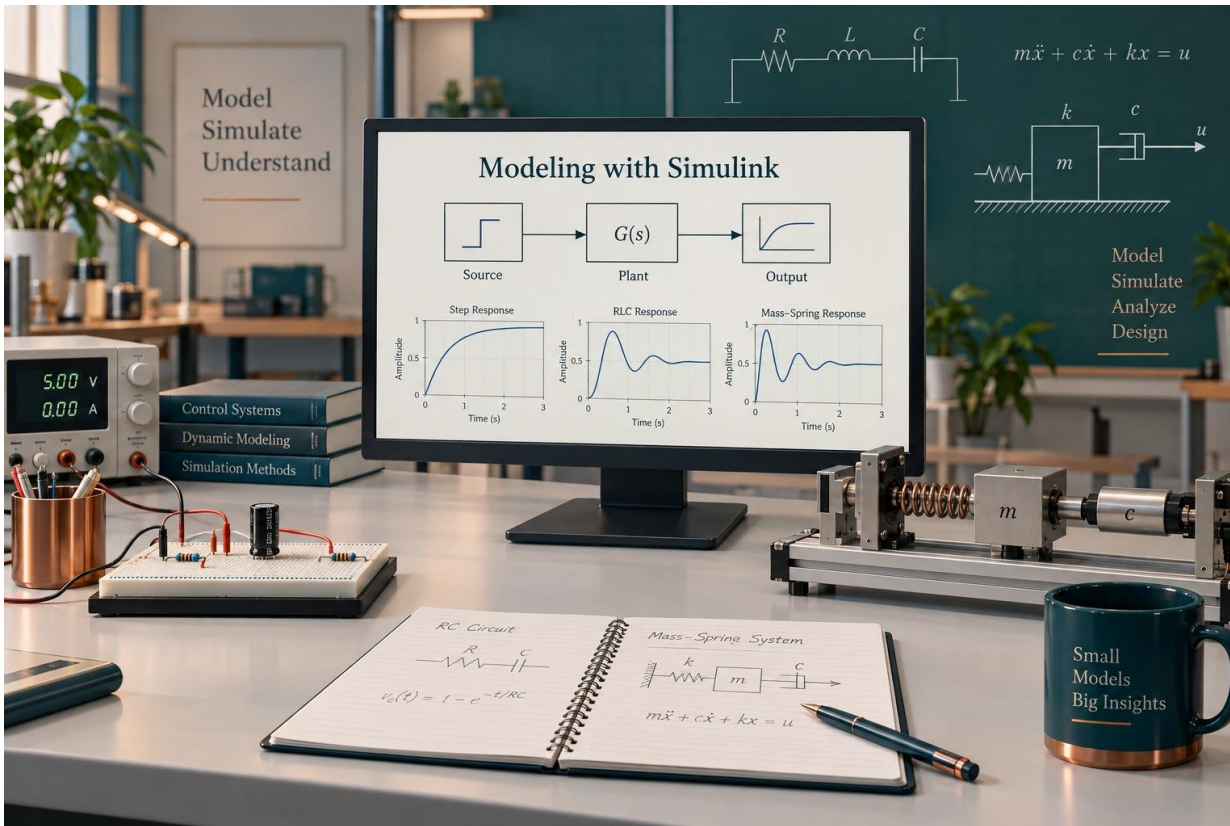


مدل سازی با سیمولینک



تصویر مفهومی آموزشی

ویژه مهندسی برق و مکانیک

مثال‌های کامل و پاسخ تمرین‌ها

محمدرضا خیری

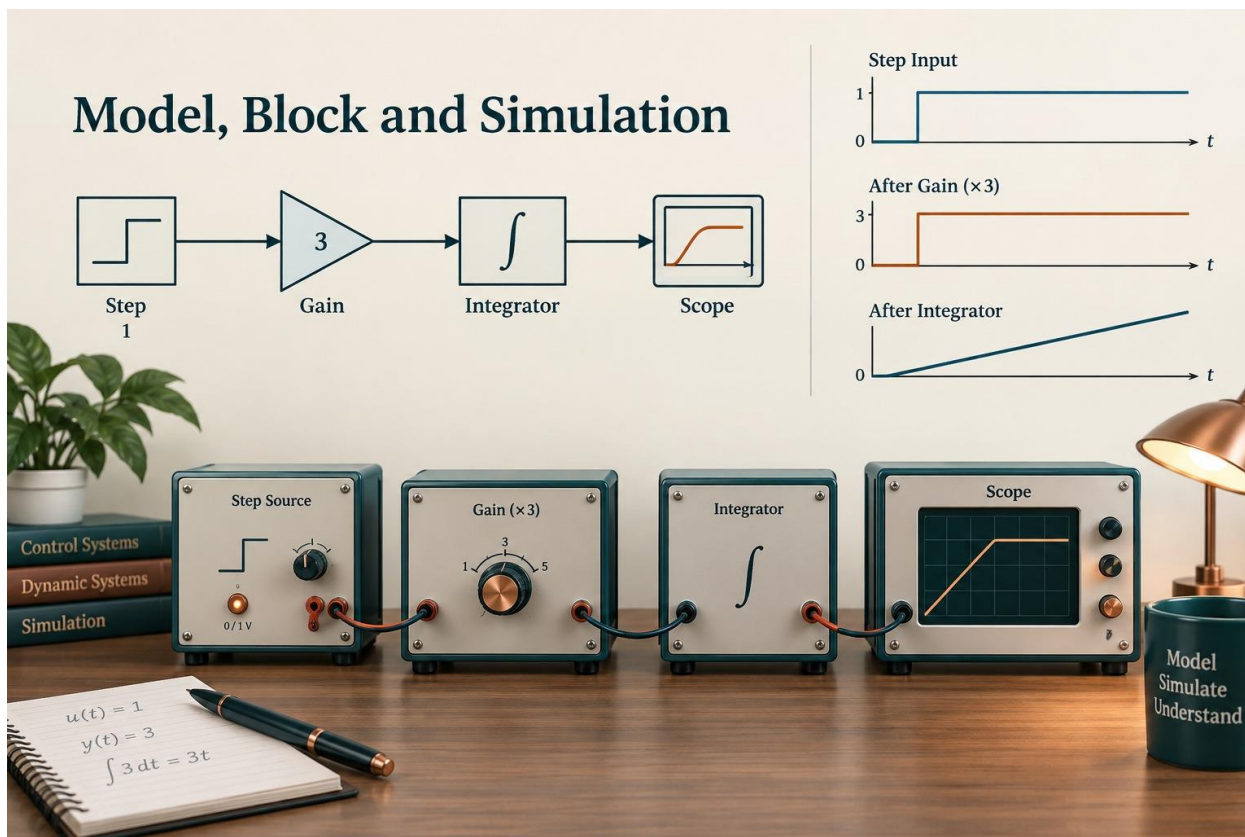
سرفصل‌های دوره

1. مدل، بلوک و شبیه‌سازی
2. سیستم‌های پیوسته برق و مکانیک
3. سیستم گسسته و کنترل
4. آزمون، پارامتر و پروژه

مدل، بلوک و شبیه‌سازی

فصل 1

مدل، مثال و پاسخ تشریحی



تصویر مفهومی آموزشی

مدل و سیگنال در Simulink

بلوک، یک عملیات یا دینامیک را توصیف می‌کند.
خط اتصال، سیگنال را از خروجی یک بلوک به ورودی بلوک دیگر می‌رساند.
زمان حل، نرخ نمونه‌برداری و شرایط اولیه، بخشی از تعریف مدل‌اند.

پیش‌نیاز نرم‌افزاری و مسیر اجرا

اجرای مدل‌ها به MATLAB و Simulink با مجوز معتبر نیاز دارد. اسکریپت‌های همراه، مدل را به صورت برنامه‌نویسی شده ایجاد می‌کنند. Simscape، Control System Toolbox و Stateflow مسیرهای اختیاری‌اند و در مثال‌های اصلی این دوره اجباری نیستند.

Scope و Integrator ،Gain ،Step

Step ورودی پله را می‌سازد Gain . سیگنال را در ضریب ضرب می‌کند.
Integrator حالت پیوسته را از مشتق آن می‌سازد.
Scope برای مشاهده و To Workspace برای انتقال داده نتیجه به MATLAB استفاده می‌شوند.

ساخت تابع کمکی مدل

تابع `build_tf_model` در فایل همراه، ورودی `Step`، بلوک `Transfer Fcn` و خروجی `To Workspace` را وصل می‌کند.
کد کامل تابع در پیوست همین ارائه آمده است.
هر مثال پارامترهای فیزیکی خود را به این تابع می‌دهد و خروجی را با مرجع مستقل مقایسه می‌کند.

ساخت مدل در محیط گرافیکی

یک مدل خالی بسازید و Step، Transfer Fcn و Scope را از Library Browser اضافه کنید.
خروجی Step را به Plant و خروجی Plant را به Scope وصل کنید.
پارامتر مخرج، زمان پله و StopTime را پیش از Run مشخص کنید. اسکریپت همراه همین ساختار را با کد می‌سازد.

مدل بهره ثابت

ورودی پله یک را در بهره ۳ ضرب کنید. خروجی پس از آغاز پله چیست؟

```
model = 'gain_example';  
build_tf_model(model,3,1,2);  
out = sim(model);  
y = out.y;  
disp(y.Data(end));
```

پاسخ مدل بهره ثابت

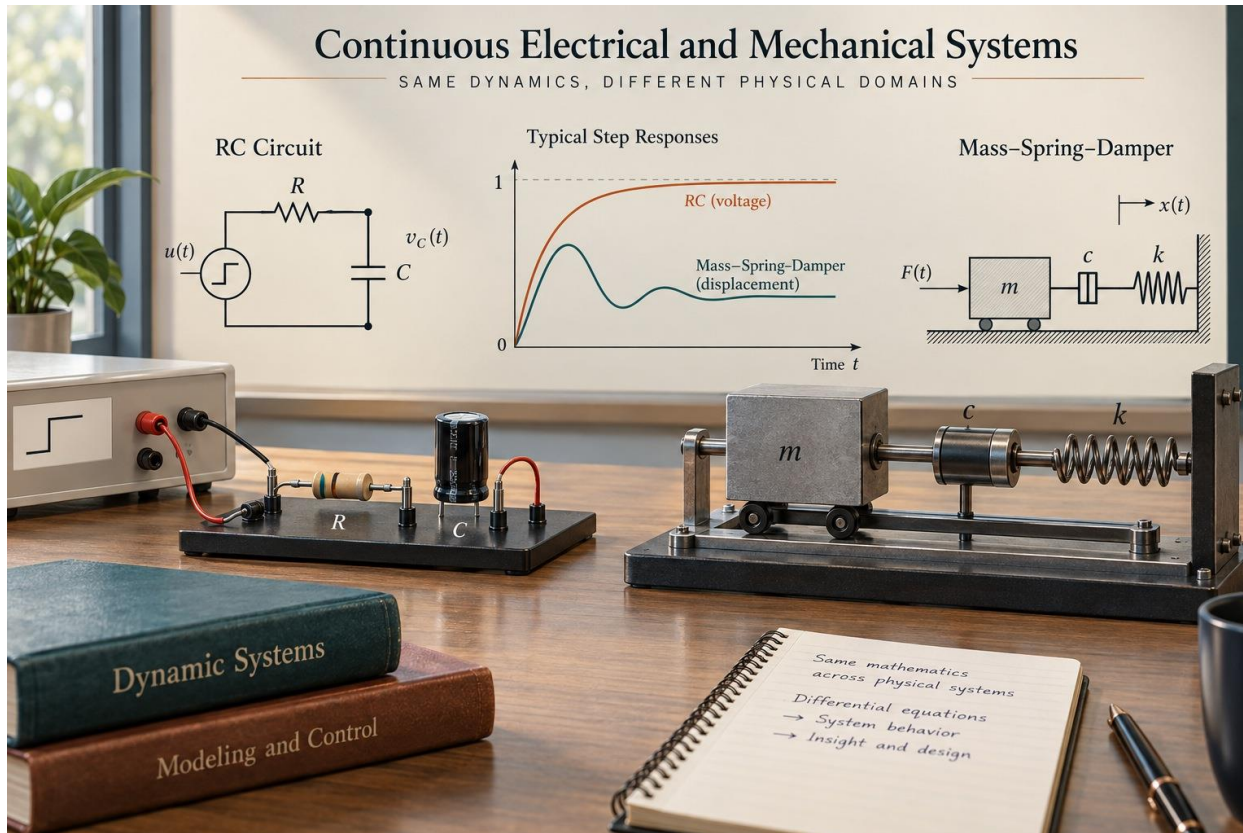
خروجی: ۳

مدل با Transfer Fcn بدون حالت هم می‌تواند یک بهره ثابت را نشان دهد. برای نمایش مستقیم مفهوم، بلوک Gain نیز مناسب است.

سیستم‌های پیوسته برق و مکانیک

فصل 2

مدل، مثال و پاسخ تشریحی



تصویر مفهومی آموزشی

مدل مرتبه اول و شرایط اولیه

مدل $\dot{y} + y = u$ یک حالت دارد.

برای شرایط اولیه صفر و ورودی پله واحد، $y = 1 - \exp(-t/\tau)$.
تغییر شرایط اولیه، پاسخ گذرا را عوض می‌کند؛ مقدار اولیه بخشی از قرارداد مدل است.

مدل شارژ خازن

برای $R=1000$ اهم و $C=100$ میکروفاراد، پاسخ پله واحد را در 0.5 ثانیه بررسی کنید.

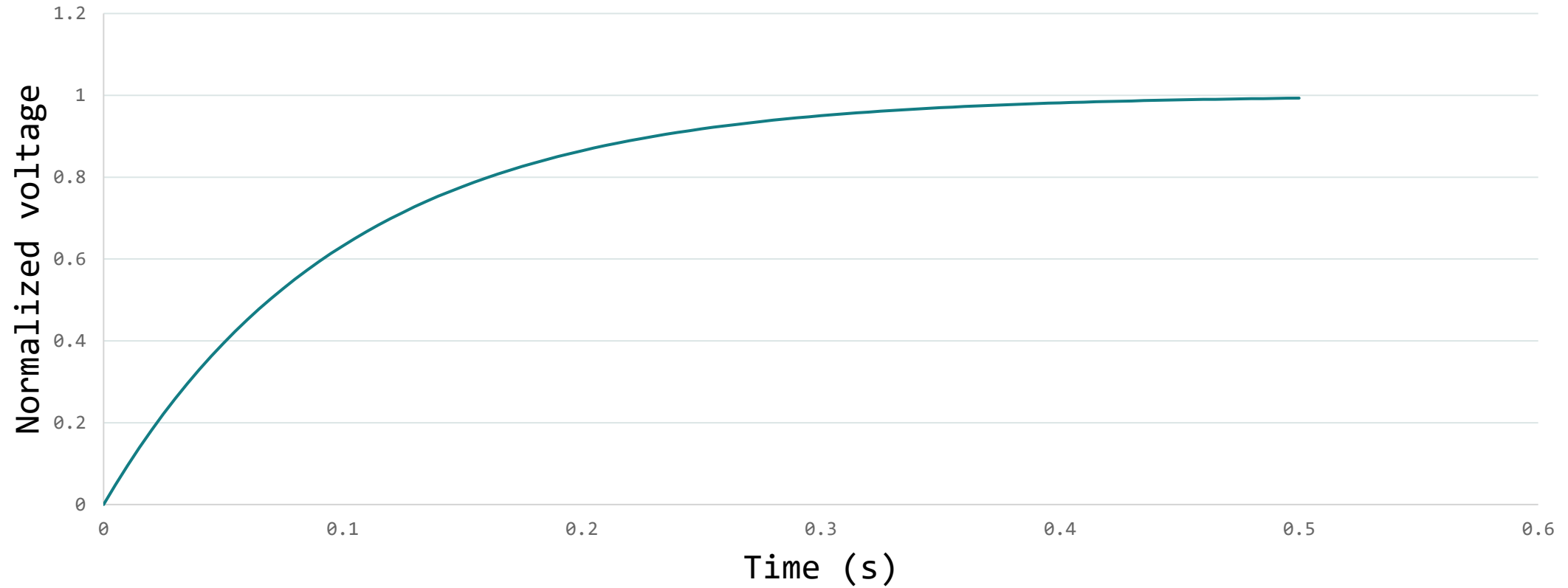
```
R = 1000; C = 100e-6; tau = R*C;  
model = 'rc_example';  
build_tf_model(model,1,[tau 1],0.5);  
out = sim(model); y = out.y;  
reference = 1-exp(-0.5/tau);  
disp([y.Data(end) reference]);
```

پاسخ مدل شارژ خازن

پاسخ مرجع در 0.5 ثانیه ≈ 0.993262
برای پله 5 ولت، خروجی پنج برابر می شود.

تابع انتقال ولتاژ خازن به ورودی برابر $1/(RCs+1)$ است. شرایط اولیه در مدل صفر است.

پاسخ تحلیلی مرجع



این نمودار از رابطه تحلیلی مثال ساخته شده است.

مدل جرم و فنر

برای $m=1$ ، $c=2$ و $k=4$ ، پاسخ مکان به نیروی پله واحد را شبیه‌سازی کنید.

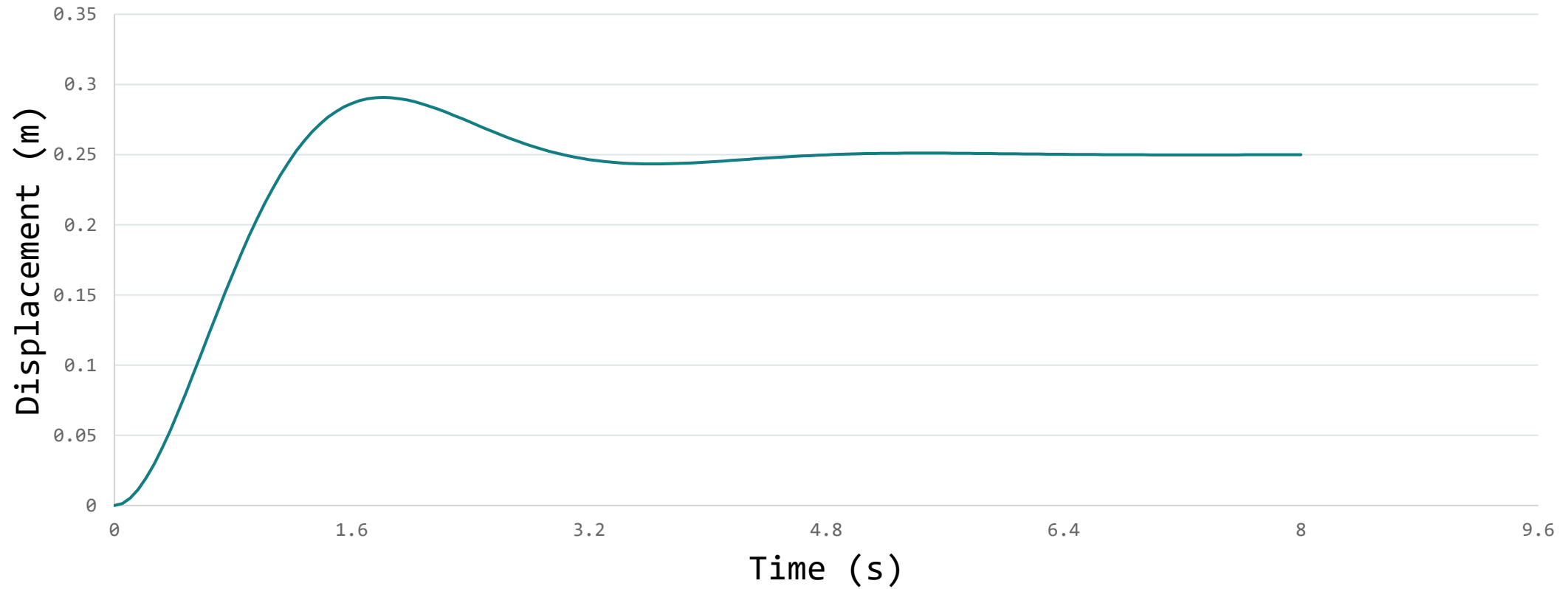
```
m = 1; c = 2; k = 4;  
model = 'spring_example';  
build_tf_model(model,1,[m c k],8);  
out = sim(model); y = out.y;  
staticDisplacement = 1/k;  
disp([y.Data(end) staticDisplacement]);
```

پاسخ مدل جرم و فنر

جابجایی ماندگار مرجع 0.25 : متر
خروجی در زمان محدود ممکن است هنوز دقیقاً ماندگار نشده باشد.

مدل از $m\ddot{x} + c\dot{x} + kx = F$ به دست می‌آید. مقدار ماندگار برای نیروی ثابت واحد، $1/k$ است.

پاسخ تحلیلی مرجع



این نمودار از رابطه تحلیلی مثال ساخته شده است.

مدار RLC سری

برای $L=10$ میلی‌هانی، $C=100$ میکروفاراد و $R=10$ اهم، مدل ولتاژ خازن به ورودی را بسازید.

```
L = 0.01; C = 100e-6; R = 10;  
model = 'rlc_example';  
build_tf_model(model,1,[L*C R*C 1],0.03);  
out = sim(model);  
wn = 1/sqrt(L*C);  
zeta = R/2*sqrt(C/L);  
disp([wn zeta]);
```

پاسخ مدار RLC سری

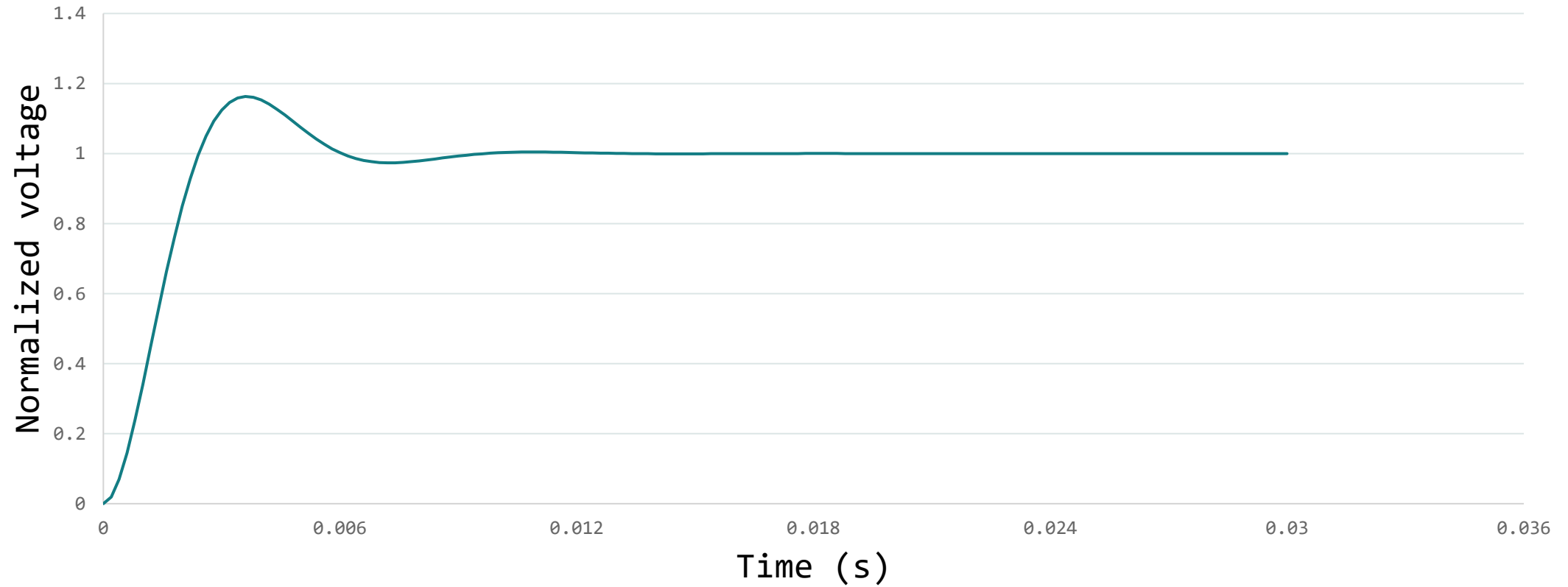
$$\omega_n = 1000 \text{ rad/s}$$

$$\zeta = 0.5$$

بهره ماندگار ولتاژ خازن: یک

مخرج تابع انتقال $LCs^2 + RCs + 1$ است. شکل پاسخ در این مثال کم میرایی است.

پاسخ تحلیلی مرجع



این نمودار از رابطه تحلیلی مثال ساخته شده است.

مدل حرارتی مرتبه اول

سامانه حرارتی با ثابت زمانی ۱۰ ثانیه و بهره یک را برای ورودی پله واحد شبیه‌سازی کنید.

```
tau = 10;  
model = 'thermal_example';  
build_tf_model(model,1,[tau 1],50);  
out = sim(model);  
reference = 1-exp(-5);  
disp([out.y.Data(end) reference]);
```

پاسخ مدل حرارتی مرتبه اول

در ۵۰ ثانیه $0.993262 \approx$ از تغییر نهایی

این مدل خطی آموزشی، رفتار همه سامانه‌های حرارتی را توصیف نمی‌کند. پارامتر باید با داده یا مشخصات فیزیکی همان سامانه تعیین شود.

مدل فضای حالت موتور DC

با فرض $L=J=1$ و $R=b=1$ و ثابت‌های موتور 0.1 ، ماتریس حالت را برای جریان و سرعت بسازید.

```
L = 1; R = 1; J = 1; b = 1;  
Kt = 0.1; Ke = 0.1;  
A = [-R/L -Ke/L; Kt/J -b/J];  
B = [1/L; 0]; C = [0 1]; D = 0;  
poles = eig(A);  
steady = -A\B;  
disp(poles); disp(C*steady+D);
```

پاسخ مدل فضای حالت موتور DC

قطب‌ها: $-1 \pm 0.1i$
سرعت ماندگار برای ورودی واحد ≈ 0.0990099

حالت‌ها جریان و سرعت زاویه‌ای‌اند. بلوک State-Space می‌تواند A، B، C و D را بگیرد. مثال با فرض‌های خطی و واحدهای SI است.

ورودی دلخواه از Workspace

یک ورودی رمپ با نمونه‌های زمان صفر تا یک بسازید تا به بلوک From Workspace داده شود.

```
t = (0:0.1:1)';  
u = 2*t;  
inputSignal = timeseries(u,t);  
disp([inputSignal.Time(end) inputSignal.Data(end)]);  
% Set From Workspace Data to inputSignal.
```

پاسخ ورودی دلخواه از Workspace

در آخرین نمونه: زمان ۱ ثانیه و مقدار ۲

زمان‌ها باید مرتب باشند و سیاست درونیابی و رفتار پس از آخرین نمونه مشخص شود.

حل‌کننده و تفرانس

مدل RC را با تفرانس مشخص اجرا کنید و خطای آخرین نمونه را در برابر پاسخ تحلیلی محاسبه کنید.

```
model = 'solver_example';  
build_tf_model(model,1,[0.1 1],0.5);  
set_param(model,'Solver','ode45', ...  
    'RelTol','1e-7','AbsTol','1e-9');  
out = sim(model);  
errorFinal = abs(out.y.Data(end)-(1-exp(-5)));  
disp(errorFinal);
```

پاسخ حل‌کننده و تفرانس

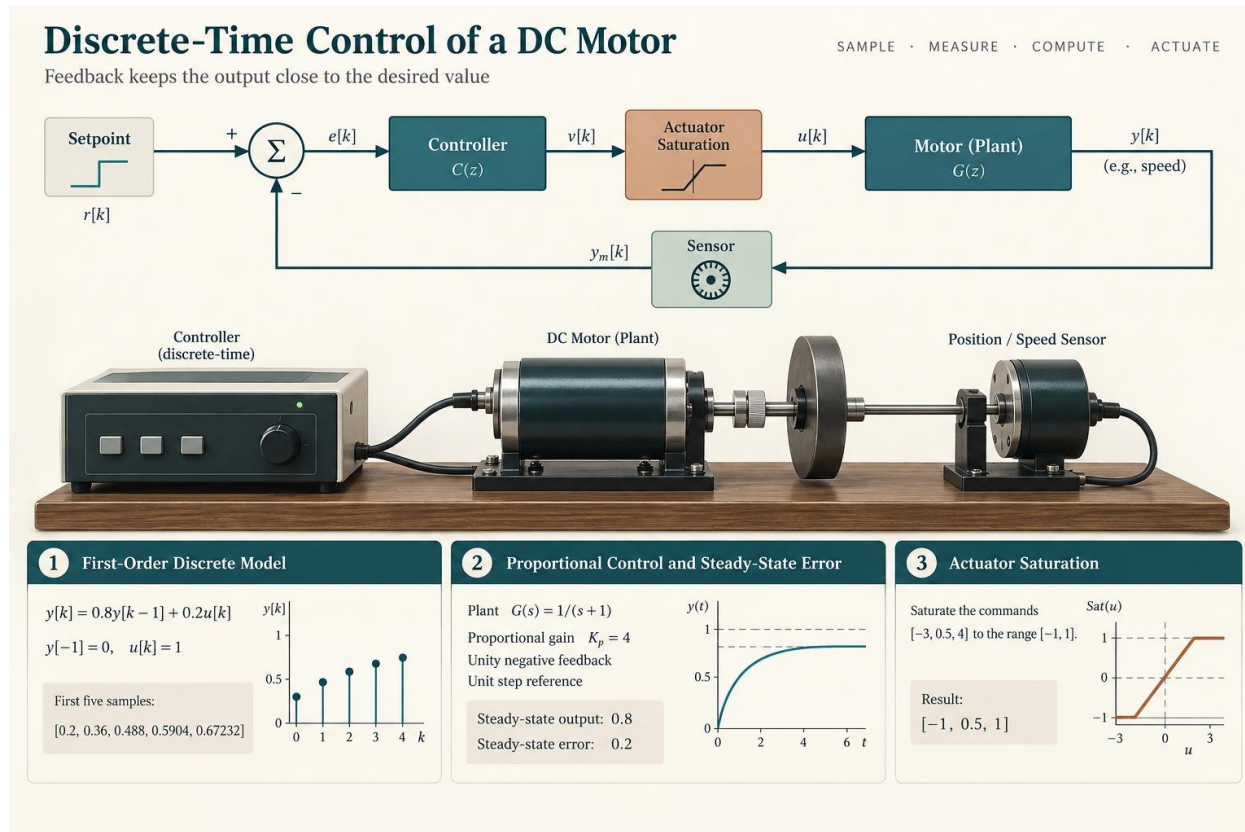
خطای آخرین نمونه باید از اجرای واقعی گزارش شود.
کاهش تفرانس معمولاً هزینه محاسباتی را افزایش می‌دهد.

انتخاب حل‌کننده به پیوستگی، سختی مسئله و نرخ نمونه‌برداری بستگی دارد. کوچک کردن گام به‌تنهایی مدل فیزیکی غلط را اصلاح نمی‌کند.

سیستم گسسته و کنترل

فصل 3

مدل، مثال و پاسخ تشریحی



تصویر مفهومی آموزشی

کنترل PID و مسیرهای توسعه

کنترل PID از سهم تناسبی، انتگرالی و مشتقی استفاده می‌کند. فیلتر مشتق، اشباع، anti-windup و نویز اندازه‌گیری بر رفتار واقعی اثر دارند. پاسخ تمرین: با افزودن انتگرال، خطای ثابت می‌تواند برای برخی گیاه‌ها کاهش یابد، اما پایداری و اشباع باید دوباره بررسی شوند.

نرخ نمونه برداری و گام حل

Sample time نرخ به روزرسانی یک بخش گسسته است.
گام حل کننده، شبکه محاسبه شبیه سازی را تعیین می کند.
برای مدل دارای چند نرخ، سازگاری تبادل سیگنال و Rate Transition را بررسی کنید.

مدل گسسته مرتبه اول

رابطه $y[k] = 0.8y[k-1] + 0.2u[k]$ با $y[-1] = 0$ و ورودی واحد را برای پنج نمونه حساب کنید.

```
u = ones(1,5); y = zeros(1,5); previous = 0;
for k = 1:5
    y(k) = 0.8*previous+0.2*u(k);
    previous = y(k);
end
disp(y);
```

پاسخ مدل گسسته مرتبه اول

[0.2, 0.36, 0.488, 0.5904, 0.67232]

برای بلوک Discrete Transfer Fcn با فرم چند جمله‌ای z ، $\text{num}=[0.2 \ 0]$ و $\text{den}=[1 \ -0.8]$ همین پاسخ بدون تأخیر اضافی را می‌دهد. نرخ نمونه‌برداری نیز باید تعیین شود.

کنترل تناسبی و خطای ماندگار

برای گیاه $1/(s+1)$ و $K_p=4$ در بازخورد منفی واحد، پاسخ مرجع واحد و خطای ماندگار را بیابید.

```
Kp = 4;  
model = 'feedback_example';  
build_tf_model(model,Kp,[1 1+Kp],5);  
out = sim(model);  
yss = Kp/(1+Kp); ess = 1-yss;  
disp([yss ess]);
```

پاسخ کنترل تناسبی و خطای ماندگار

خروجی ماندگار 0.8 :
خطای ماندگار 0.2 :

مدل حلقه بسته $K_p/(s+1+K_p)$ است. این نتیجه برای کنترل P و فرض‌های همین گیاه است.

اشباع محرک

فرمان‌های $[-3, 0.5, 4]$ را به محدوده $[-1, 1]$ محدود کنید.

```
command = [-3 0.5 4];  
lower = -1; upper = 1;  
limited = min(max(command, lower), upper);  
disp(limited);  
% Saturation: LowerLimit=-1, UpperLimit=1
```

پاسخ اشباع محرک

$[-1, 0.5, 1]$

اشباع یک غیرخطی بودن مهم است. در کنترل انتگرالی، anti-windup باید جدا طراحی و آزموده شود.

آزمون، پارامتر و پروژه

فصل 4

مدل، مثال و پاسخ تشریحی



تصویر مفهومی آموزشی

زیرسیستم و مدل بزرگ

بلوک‌های دارای یک مسئولیت مشخص را در Subsystem قرار دهید.
ورودی و خروجی، واحد و نرخ نمونه‌برداری هر زیرسیستم را مشخص کنید.
برای مدل بزرگ، مدیریت داده، نسخه‌بندی و آزمون‌های بازگشتی اهمیت پیدا می‌کنند.

پروژه نهایی برق و مکانیک

برق: مقایسه پاسخ RC و RLC با ثابت‌ها و شرایط اولیه معلوم.

مکانیک: پاسخ جرم و فنر و اثر میرایی.

پاسخ نمونه RC : در پنج ثابت زمانی به 0.993262 از مقدار نهایی می‌رسد؛ جابجایی ماندگار جرم و فنر برای نیروی واحد برابر $1/k$ است.

حلقه جبری و مدل فیزیکی

اتصال مستقیم بلوک‌ها ممکن است وابستگی هم‌زمانی ایجاد کند که به حل حلقه جبری نیاز دارد. برای رفع مشکل، ابتدا معنای فیزیکی و ساختار معادله را بررسی کنید. افزودن تأخیر صرفاً برای حذف هشدار، ممکن است مدل را عوض کند.

تغییر پارامتر و مقایسه

ثابت‌های زمانی 0.1، 0.2، و 0.5 ثانیه را از نظر پاسخ در $t=1$ مقایسه کنید.

```
taus = [0.1 0.2 0.5];
results = zeros(size(taus));
for k = 1:numel(taus)
    model = sprintf('sweep_%d',k);
    build_tf_model(model,1,[taus(k) 1],1);
    out = sim(model);
    results(k) = out.y.Data(end);
end
disp(results);
disp(1-exp(-1./taus));
```

پاسخ تغییر پارامتر و مقایسه

مرجع تحلیلی $[0.999955, 0.993262, 0.864665]$ $\approx$

مدلهایی با ثابت زمانی کوچکتر، برای ورودی و شرایط اولیه یکسان سریعتر به مقدار نهایی نزدیک میشوند.

اجرای کنترل شده با SimulationInput

برای یک مدل موجود، StopTime را بدون تغییر دائمی تنظیمات اصلی به یک ثانیه تغییر دهید.

```
model = 'siminput_example';  
build_tf_model(model,1,[0.1 1],0.5);  
simIn = Simulink.SimulationInput(model);  
simIn = simIn.setModelParameter('StopTime','1');  
out = sim(simIn);  
disp(out.y.Time(end));
```

پاسخ اجرای کنترل شده با SimulationInput

آخرین زمان خروجی: یک ثانیه

SimulationInput تنظیمات یک اجرا را بسته بندی می کند. این روش برای مقایسه اجراها و تغییر پارامتر کنترل شده مفید است.

آزمون بازگشتی مدل

برای پاسخ RC در ۵ ثابت زمانی، شرطی بنویسید که خطای آخرین نمونه از ۰.۰۰۱ کمتر باشد.

```
model = 'regression_example';  
build_tf_model(model,1,[0.1 1],0.5);  
out = sim(model);  
expected = 1-exp(-5);  
observed = out.y.Data(end);  
assert(abs(observed-expected) < 1e-3);  
disp('RC regression test passed');
```

پاسخ آزمون بازگشتی مدل

در اجرای معتبر، پیام موفقیت آزمون چاپ می‌شود.
در صورت خطای بیش از حد، assert اجرای برنامه را متوقف می‌کند.

حد آزمون باید به نیاز مهندسی و روش حل مرتبط باشد. این آزمون فقط یک سناریو را پوشش می‌دهد.

کد کامل سازنده مدل

```
function build_tf_model(model,num,den,stopTime)
    if bdIsLoaded(model) || isfile([model '.slx'])
        error('Choose a new model name in a fresh work folder');
    end
    new_system(model);
    add_block('simulink/Sources/Step',[model '/Input'], ...
        'Time','0','Before','0','After','1');
    add_block('simulink/Continuous/Transfer Fcn', ...
        [model '/Plant'],'Numerator',mat2str(num), ...
        'Denominator',mat2str(den));
    add_block('simulink/Sinks/To Workspace', ...
        [model '/Output'],'VariableName','y', ...
        'SaveFormat','Timeseries');
    add_line(model,'Input/1','Plant/1');
```

کد کامل سازنده مدل

```
add_line(model,'Plant/1','Output/1');  
set_param(model,'StopTime',num2str(stopTime), ...  
    'Solver','ode45','ReturnWorkspaceOutputs','on');  
Simulink.BlockDiagram.arrangeSystem(model);  
save_system(model,[model '.slx']);  
end
```

منابع و مستندات رسمی

mathworks.com/help/matlab/ref/assert.html

mathworks.com/help/simulink/slref/discretetransferfcn.html

mathworks.com/help/simulink/slref/fromworkspace.html

mathworks.com/help/simulink/slref/saturation.html

mathworks.com/help/simulink/slref/sim.html

mathworks.com/help/simulink/slref/simulink.simulationinput.html

mathworks.com/help/simulink/slref/statespace.html

منابع و مستندات رسمی

mathworks.com/help/simulink/slref/sum.html

mathworks.com/help/simulink/slref/transferfcn.html

mathworks.com/help/simulink/ug/approach-modeling-programmatically.html

mathworks.com/help/simulink/ug/choose-a-solver.html