

Uppgift 3 - Introduktion till Simulink®

ET1522, *Beräknings- och simuleringsteknik*

ET1531, *Reglerteknik*

ET1540, *Grundläggande programmering i Matlab*

Frida Gleisner, Anders Hultgren och Matz Lenells

August 28, 2019

Simulink är en toolbox i Matlabmiljön som kan användas vid simuleringar och analys av dynamiska system. Systemen som analyseras kan vara både linjära och olinjära samt kontinuerliga eller diskreta. Simulink har ett grafiskt användargränssnitt där användaren bygger upp en modell med förutbestämda block. Matlab och Simulink har blivit ett vanligt utvecklingsverktyg inom industrin.

1 Inlämningsuppgift

Delar av denna uppgiftsbeskrivning förutsätter kännedom om resten av handledningen.

För deltagare på sommarkursen ET1522 är det obligatoriskt att lösa uppgiften då metod 1 används. För deltagare på kursen ET1540 är det obligatoriskt att lösa uppgiften med såväl metod 1 som med metod 2. För deltagare på kursen ET1531 är det obligatoriskt att lösa uppgiften med såväl metod 1 som med metoderna 2 och 3.

Du ska simulera sitsens rörelse för en kontorsstol genom att rita ett blockschema i Simulink och skriva en m-fil. Du ska beräkna den så kallade fjäderkonstanten k så att sitsen sjunker 3,4 cm när en person som väger 75 kg sätter sig på stolen. Hur beräkningarna är gjorda ska framgå av din m-fil. Stolens sits antas väga 5 kg. Simuleringar ska göras för tre olika fall av dämpningskonstanten b : $b_1 = 400$ Ns/m, $b_2 = 1200$ Ns/m och $b_3 = 2400$ Ns/m.

Syftet med uppgiften är att presentera Simulink samt att du ska arbeta med en matematisk beskrivning av ett dynamiskt system, här bestående av en massa, en fjäder och en dämpare.

Provkör dina filer innan du skickar in dem och använd en version av Matlab från 2016 eller senare. Denna handledning utgår ifrån MATLAB_2018b. Beskrivningen av metod 1 har kompletterats så att den även fungerar om man använder MATLAB_2019a.

2 Modellering av en sitsens rörelse för en kontorsstol

2.1 Beskrivning av sitsens rörelse när en person sätter sig på stolen

När en person sätter sig på stolen sjunker sitsen nedåt och gungar eventuellt till för att slutligen ställa sig på en ny och lägre nivå. Sitsen sjunker beroende på att det i en kontorsstol

finns en så kallad gasfjäder. Vi vill beskriva hur mycket sitsen sjunker som funktion av tiden. Vi låter därför variabeln $y(t)$ beteckna sträckan som sitsen har rört sig nedåt från ursprungsläget.

Vi definierar nollnivån för $y(t)$ så att vid tidpunkten $t = 0$ så gäller att $y(0) = 0$. Vid tiden $t = 0$ tänker oss att sitsen är i vila och att en person sätter sig på stolen på ett sådant sätt att personens hela tyngd plötsligt verkar på stolen.

Vi inser att signalen $y(t)$ först ökar för att sedan övergå i en svängande rörelse som dämpas ut så att $y(t)$ så småningom kommer att anta ett nytt värde som vi antar i ett första exempel är fyra cm.

2.2 Kvalitativ beskrivning av sitsens rörelse

Gasfjädern ger en uppåtriktad kraft $F(y)$ mot stolsitsen som ökar med sitsens avstånd till jämviktsläget. (Här förutsätts att sitsen rör sig nedåt från jämviktsläget.) Detta betyder att om en tung person sitter på stolen så kommer sitsen efter insvängning att vara på en lägre nivå än sitsen hamnar på om en lätt person sätter sig på stolen.

2.3 Enkel matematisk beskrivning av sitsens rörelse

Hooks lag ger oss en enkel matematisk beskrivning av den kraft, $F(y)$, som gasfjädern ger, se ekvation (2.1).

$$F = k \cdot y + \ell, \quad \text{”Hookes lag”} \quad (2.1)$$

Här är k en konstant som kallas för fjäderkonstanten och ℓ är en parameter som svarar mot den kraft som gasfjädern verkar på stolsitsen när ingen sitter på stolen. Den givna ekvationen förutsätter att sitsen inte är i rörelse.

Vi betraktar nu två belastningsfall. I det första fallet sitter ingen person på stolen och då gäller att

$$m \cdot g = k \cdot 0 + \ell \quad (2.2)$$

där m betecknar sitsens vikt. Om vi antar att $m = 5$ kg så gäller att

$$\ell = m \cdot g \approx 5 \text{ kg} \cdot 9.8 \frac{\text{m}}{\text{s}^2} = 49 \text{ N}. \quad (2.3)$$

Detta värde behöver vi inte beräkna explicit som vi har gjort här. Beräkningen finns här bara för att det ska vara enklare att förstå vad ℓ är.

I det andra fallet sitter en person på stolen och då gäller att

$$(M + m) \cdot g = k \cdot s + \ell \quad (2.4)$$

där M betecknar personens vikt som vi antar är 75 kg och s betecknar hur lång sträcka som sitsen har sjunkit. Vi antar här att $s = 4$ cm.

Vi visar nu hur man från dessa två ekvationer kan beräkna fjäderkonstanten k . Vi skriver ut ekvationerna i de två fallen.

$$m \cdot g = k \cdot 0 + \ell \quad (2.5)$$

$$(M + m) \cdot g = k \cdot s + \ell \quad (2.6)$$

Dessa två ekvationer ger likheten

$$M \cdot g = k \cdot s \quad (2.7)$$

Denna ekvation ger i sin tur att fjäderkonstanten k ges av

$$k = \frac{M \cdot g}{s} = \frac{75 \cdot 10}{0.04} = 25 \cdot 750 = 18750 \text{ N/m} \quad (2.8)$$

2.4 Matematisk modell av en stolsits rörelse

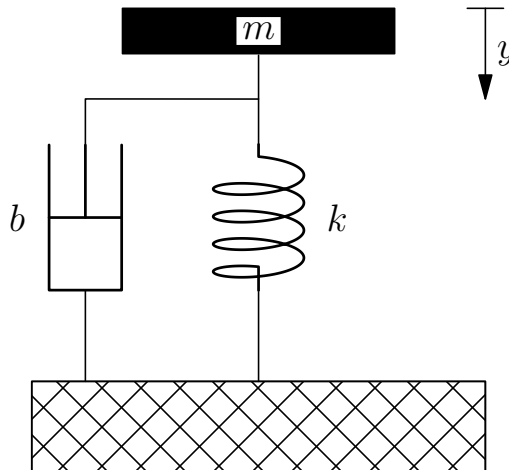


Figure 2.1: Enkel modell av en kontorsstol.

Figur 2.1 visar en enkel modell av en kontorsstol. Modellen består av sitsens massa, en ideal fjäder och en ideal dämpare. Fjäders modellerar kraften $k \cdot y + \ell$, se ekvation (2.1), och dämparen modellerar den kraft som uppstår i stolens gasfjäder på grund av att stolsitsens rörelse. Som redan sagts så väljer vi $y = 0$ vid jämviktsläget för stolen innan någon sätter sig på den.

Om någon sätter sig på stolen, någon med en vikt som inte överstiger den som stolen är byggd för, kommer sitsen att sjunka ner och eventuellt gunga till och finna ett nytt jämviktsläge där y är större än noll. Vi illustrerar detta med figur 2.2. Förändringen representeras av att massan ökar från m till $m + M$, där M representerar massan för personen som sätter sig på stolen, se figur 2.2. Stolens rörelse, efter det att en person har satt sig på stolen, beskrivs av differentialekvationen

$$(m + M)\ddot{y}(t) = -b\dot{y}(t) - ky(t) + Mg. \quad (2.9)$$

Den tillagda vikten, termen Mg , kan ses som en insignal till systemet, se figur 2.3. Om vi kallar insignalen för $u(t)$ kan ekvation (2.9) skrivas enligt

$$(m + M)\ddot{y}(t) + b\dot{y}(t) + ky(t) = u(t). \quad (2.10)$$

I vårt fall med kontorsstolen antar $u(t)$ två värden, $u = 0$ innan någon har satt sig på stolen och $u = Mg$ efter att personen har satt sig på stolen. Eftersom vi vill studera rörelsen hos systemet väljer vi att börja simuleringen när personen sätter sig på stolen vid tiden $t = 0$. Vi får

$$u(t) = \begin{cases} 0 & t < 0, \\ Mg & 0 \leq t. \end{cases} \quad (2.11)$$

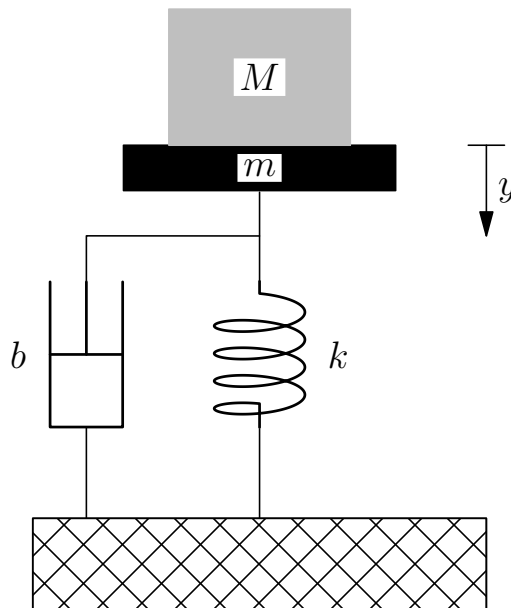


Figure 2.2: Modell av kontorsstol där massan för personen som sätter sig är tillagd.

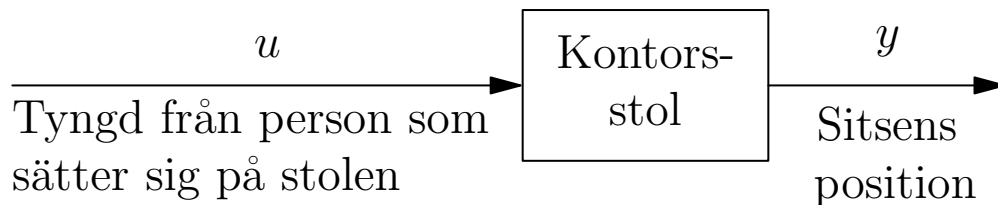


Figure 2.3: Kontorsstolen är ett system där tyngden från en person som sätter sig på sitsen kan ses som en insignal som ändrar sitsens position från dess initiala jämviktsposition.

Att lösningen $y(t)$ ska uppfylla ekvation (2.10) är inte tillräckligt för att bestämma lösningen entydigt, ytterligare villkor behövs. Normalt brukar man ange så kallade initialvillkor och i vårt fall antar vi att stolen står still i sitt jämviktsläge innan personen sätter sig varför vi skriver vårt system enligt

$$\begin{aligned}(m + M)\ddot{y}(t) + b\dot{y}(t) + ky(t) &= u(t), \\ \dot{y}(0) &= 0, \\ y(0) &= 0.\end{aligned}\tag{2.12}$$

3 Simulering av kontorsstolen med hjälp av Simulink

3.1 Filnamn

Vi kommer att i denna handledning använda två typer av filer. Dels m-filer vars filnamn alltid har ändelsen ".m", och dels filer som definierar blockscheman vars filnamn alltid har ändelsen ".slx".

Om du heter Lisa Berg så ska du kalla m-filen för `SimOfficeChairIntegratorsLisaBerg.m`

och slx-filen för

OfficeChairIntegratorsLisaBerg.slx.

Anpassa filnamnen så att de passar ditt namn så när som på att du inte ska använda bokstäverna å, ä eller ö. Det är första året som vi använder Canvas. Något som skiljer Canvas från den tidigare lärplattformen är att när Canvas tar emot filer från er och vi laddar ner samtliga filer på en gång till våra datorer så lägger den till era namn samt ett antal nummer. Sen händer det ofta att Matlab inte gillar filnamnen. Lägg därför gärna de två filerna i en mapp (kallas även för katalog eller binder) och kalla mappen för T3LisaBerg om du heter Lisa Berg annars använder du ditt namn. Lämna sedan in mappen.

Denna handledning presenterar tre olika metoder för att simulera kontorsstolens rörelse. Blockscheman ritas i en editor som hör till programmet Simulink.

3.2 Metod 1 - Systemet beskrivet med hjälp av två integratorer

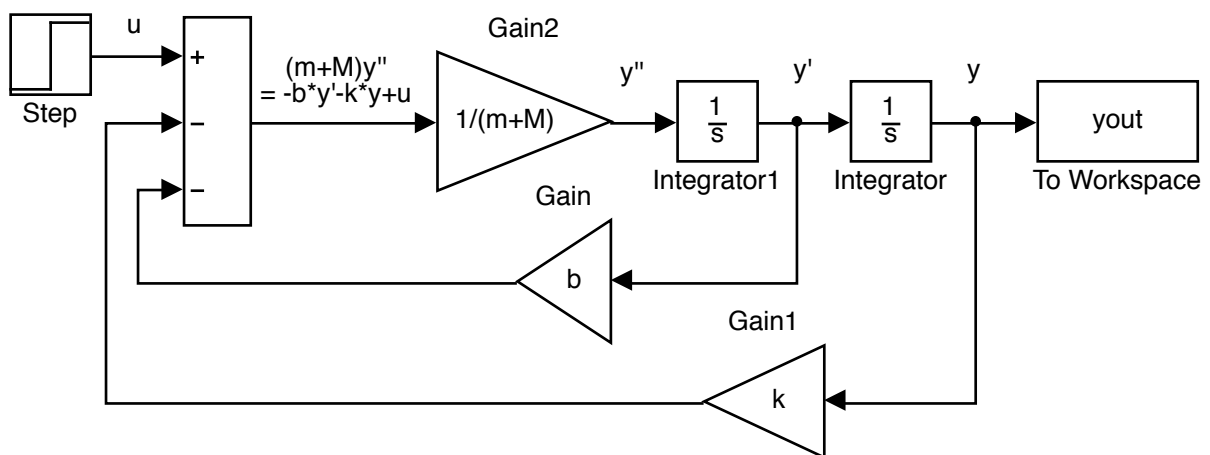


Figure 3.1: Blockschemat då integratorer används.

En metod för att simulera differentialekvation (2.12) är att skriva ekvationen på formen

$$(m + M)\ddot{y}(t) = -b\dot{y}(t) - ky(t) + u(t). \quad (3.1)$$

I blockschemat som visas i figur 3.1 har vi utgått från denna ekvation. I textkommentarer som finns i blockschemat används y' istället för \dot{y} och y'' istället för \ddot{y} .

Det finns två block i vilka det står $\frac{1}{s}$. Ett sådant block svarar mot integration och kallas för en integrator. Triangelarna svarar mot förstärkare och rektangeln med ett "+" och två "-"-tecken i sig är en summerator. Före den första integratorn har vi signalen $\ddot{y}(t)$ och vi ser därmed att blockschemat svarar mot ekvation (3.1).

3.2.1 Anvisningar för att skapa blockschemat i figur 3.1

1. Öppna ett simulinkfönster och skapa en slx-fil.

- Starta Simulink från Matlab genom att klicka på knappen för Simulink på verktygsraden eller genom att skriva "simulink" i kommandofönstret. Vid problem läs appendix B. Matlab svarar med att öppna ett fönster med titeln *Simulink Start Page*, se figur 3.2.

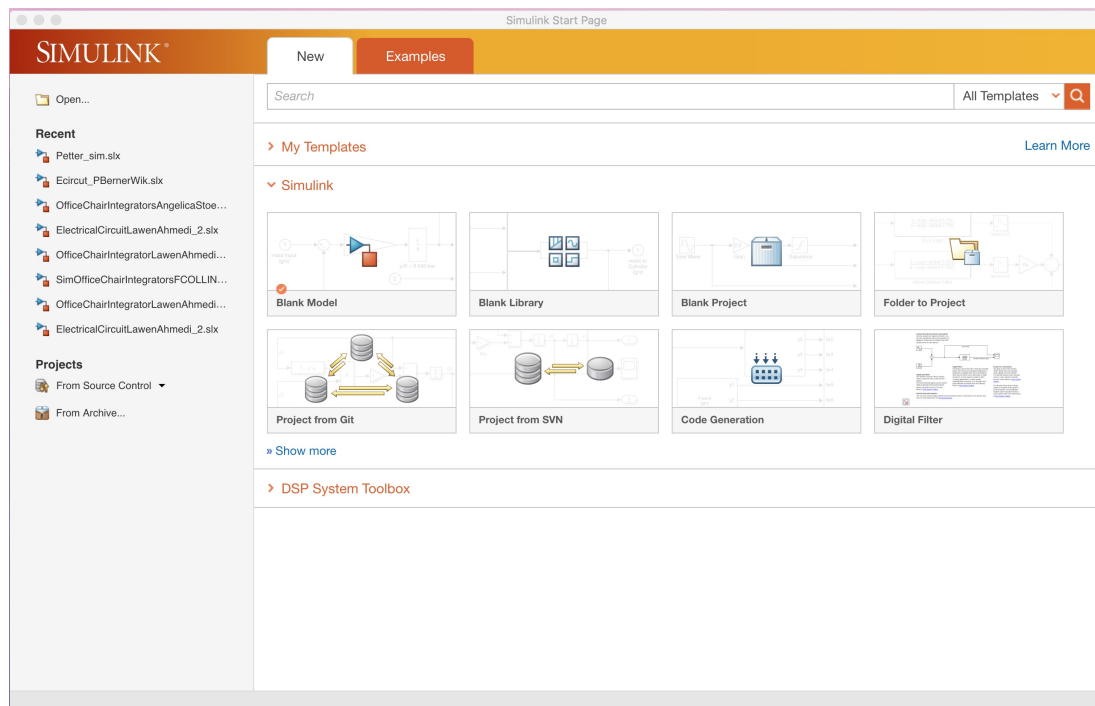


Figure 3.2: Startsidan för Simulink.

- (b) Under fliken *New* välj *Simulink* och klicka på det fönster som har undertexten *Blank Model*. Det kan ta en stund innan ett fönster kommer fram.
- (c) Ett tomt fönster kommer upp och fönstret har titeln *untitled*.
- (d) Filen sparas genom att klicka på menyposten *File -> Save As....* Kalla filen *OfficeChairIntegratorsLisaBerg.slx* om du heter Lisa Berg annars modifiera på lämpligt sätt. Filändelsen *.slx* finns redan och den ska du behålla.
- (e) Filen kan öppnas via Matlabs huvudfönster, den hittas tillsammans med övriga filer i underfönstret med rubriken *Current Folder*.

2. Placera funktionsblock i simulinkfönstret som figur 3.1 visar.

- (a) Klicka på menyposten *view -> Library Browser*. Då ska ett fönster med titeln *Simulink Library Browser* öppnas. Fönstret har två kolumner. I vänstra kolumnen ska det längst upp finns rubriken *Simulink*. Framför rubriken finns en triangel och den ska peka nedåt. Om den inte gör det klicka på den. När den pekar nedåt ska dryga tio underrubriker komma fram.
När man klickar på underrubriker i fönstrets vänstra del kommer det upp ett stort antal småfigurer, ikoner, med undertexter. En sådan liten ikon svarar mot något vi kallar för ett block eller funktionsblock och vi kallar ibland även ikonerna för ett block. Blocken kan dras till simulinkfönstret.
- (b) Från underrubriken *Sources* hämtas blocket *Step*. Under ikonerna ska underrubriken *Step* finnas.
- (c) Från *Math Operations* hämtas blocket *Add*. Öppna blocket *Add* genom att dubbelklicka på det. Skriv in de tre tecknen $+-$ i fältet "List of signs". Klicka på OK.
- (d) Från *Math Operations* hämtas blocket *Gain*. Det är möjligt att göra en kopia av ett block genom att högerklicka på det och dra ut en kopia som sedan släpps.

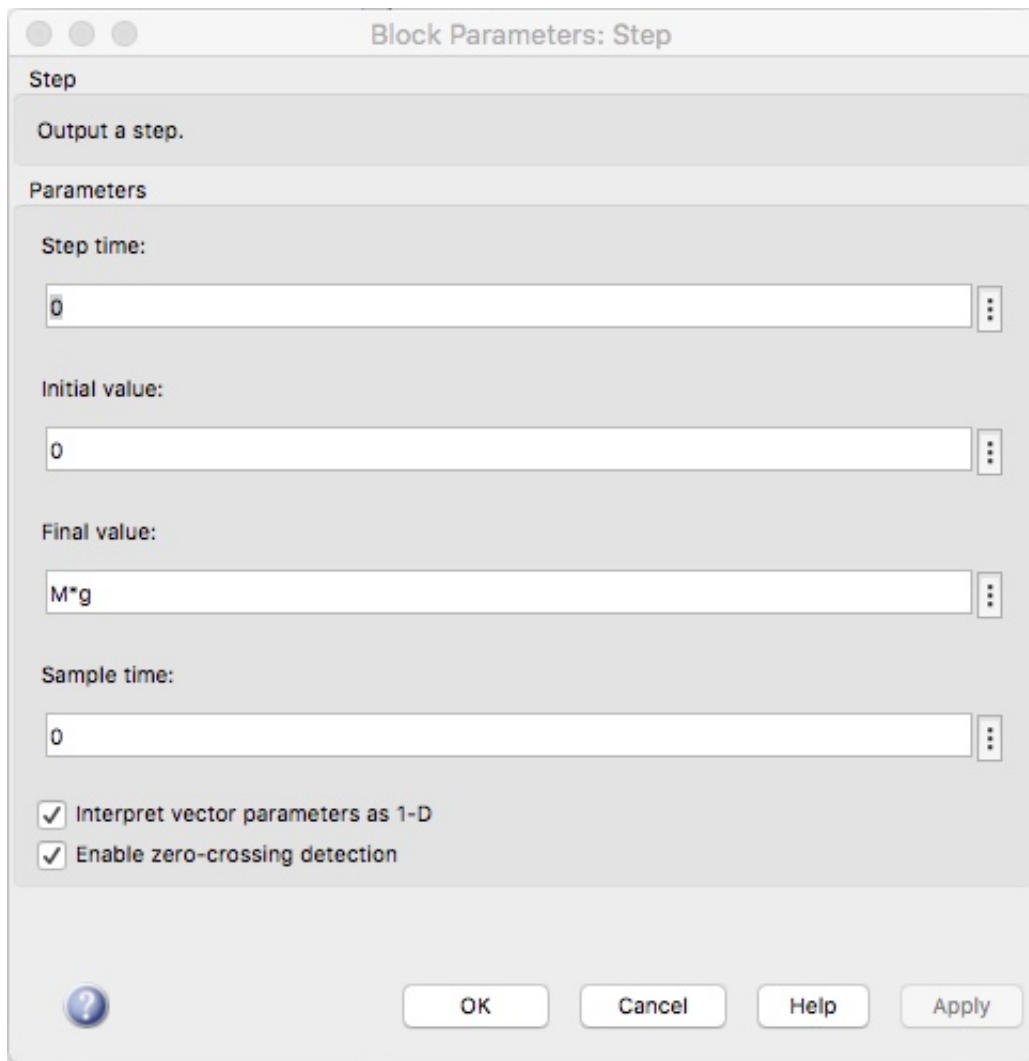


Figure 3.3: Genom att dubbelklicka på blocket *Step* i blockschemat "OfficeChairIntegrators" visas ett fönster som har vissa fält skrivbara. I detta fönster har textfälten för "Step time" och "Final Value" ändrats till 0 och $M * g$ respektive. Observera att felutskrift fås om M och g inte är definierade. För att definiera M så ge exempelvis kommandot » $M=75$ i kommandofönstret.

(På en mac hålls ctrl nedtryckt för att dra ut en kopia). Notera att Simulink själv särskiljer flera block av samma sort genom att numrera dem. Blockets namn har ingen betydelse för dess funktion. Två av de tre *Gain*-blocken behöver roteras. Detta görs genom att högerklicka på blocket och i snabbmenyn välja *Rotate & Flip -> Flip Block*. (Alternativt: Klicka på menyposten *Diagram -> Rotate & Flip -> Flip Block*.)

- (e) Från *Continuous* hämtas två block av typen *Integrator*.
- (f) Från *Sinks* hämtas blocket *To Workspace*.

Om inte underrubriken syns så klicka på ikonen. Då ska det komma upp tre prickar ovanför ikonen. Håll musmarkören ovanför prickarna. Då ska det komma upp fyra små ikoner. Välj andra från vänster. Då ska en underrubrik visas.

Ikonens storlek kan ändras på vanligt vis genom att ikonen markeras och man drar i något av hörnen.

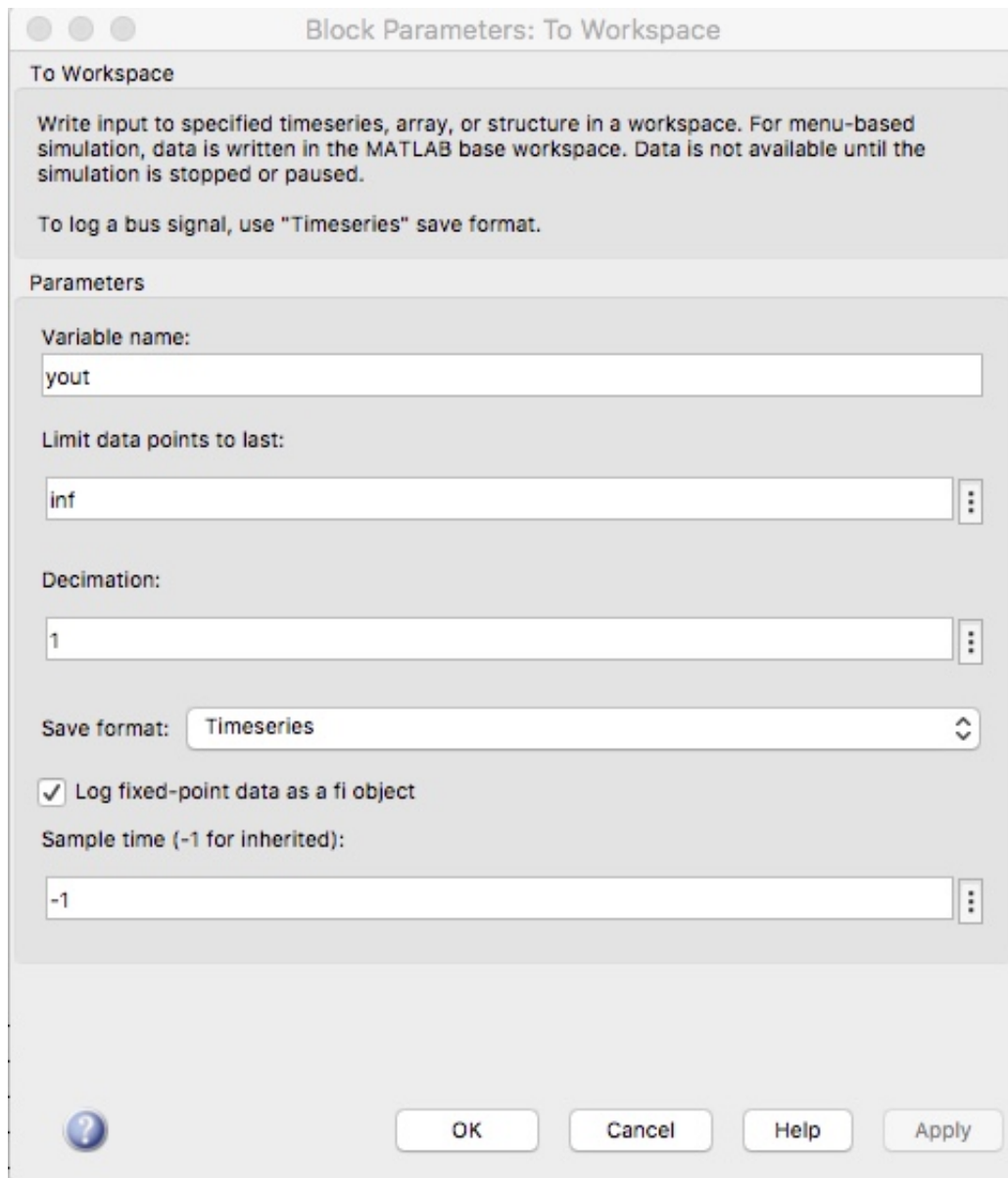


Figure 3.4: Genom att dubbelklicka på blocket To Workspace i blockschemat "OfficeChairIntegrators" visas ett fönster som har vissa fält skrivbara. I detta fönster har textfältet för "Variable name" ändrats till *yout*.

3. Sammanlänka funktionsblocken

Håll markören över symbolen för output ">", markören ändrar då form till ett +, sedan kan en linje dras till symbolen för input på ett annat block ">", (symbolerna blir roterade om blocket är roterat).

En linje kan även dras från input till output eller från input till en tidigare dragen linje.

För att radera ett block eller en koppling klickar du på blocket eller kopplingen och trycker sedan på delete.

4. Justera parametrar för funktionsblocken. Öppna respektive block genom att dubbelklicka på det. Så länge parametrarna som skrivs in inte är definierade i en m-fil kommer Simulink ge felmeddelanden, dessa försvinner när m-filen körs.

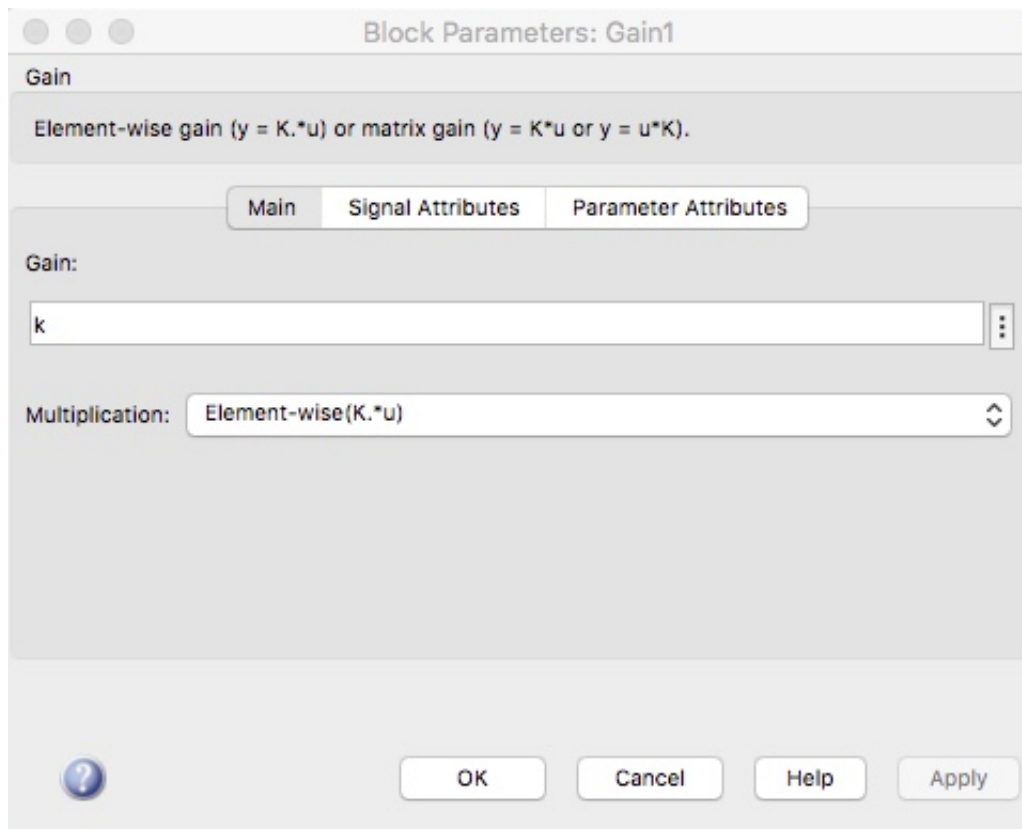


Figure 3.5: Genom att dubbelklicka på blocket `Gain` i blockschemat "OfficeChairIntegrators" visas ett fönster som har vissa fält skrivbara. I detta fönster har textfältet för 'Gain:' ändrats till k .

- (a) Öppna stegfunktionsblocket, då kommer ett fönster upp med titeln "Block Parameters: Step", se figur 3.3. Sätt "Step time" till 0, vilket betyder att personen sätter sig på stolen vid $t = 0$ när simuleringen startar. Den externa kraften som appliceras när sitsen trycks ner är $M \cdot g$, alltså ska "Final value" sättas till $M \cdot g$. Klicka på OK.
- (b) Öppna blocket `To Workspace`, se figur 3.4, och skriv in `yout` som namnet på variabeln. Kontrollera att "Save format" är "Timeseries". Om det inte är det och om man inte kan välja detta format så beror det på att den använda simulinkversion är gammal. Välj då formatet "Structure". I programkoden som finns nedan finns kommentarer som anger hur du ska förändra programmet för att det ska fungera.
- (c) Öppna blocket `Gain`, som är kopplat till utgången av det högra integratorblock, se figur 3.5, och skriv in parametern k .
- (d) Öppna blocket `Gain`, som är kopplat till utgången av det vänstra integratorblocket, och skriv in parametern b .
- (e) Öppna blocket `Gain` vars utgång är kopplat till ingången av det vänstra integratorblocket, och skriv in parametern $1/(m + M)$. Markera blocket genom att klicka på det, öka blockets storlek genom att dra i ett av hörnen så att texten $1/(m + M)$ framträder.

5. Lägg till noteringar i fönstret

I blockschemat i figur 3.1 finns noteringar, eller kommentarer. Dessa noteringar påverkar inte programmet. Det finns två typer av noteringar. Den ena typen är text som kan stå under ett block, dubbelklicka på texten för att ändra den. Den andra typen av noteringar är text på valfritt tomt ställe i schemat. Exempelvis står det u mellan Step- och Add-blocken. För att skapa en sådan notering gör så här: Dubbelklicka på ett ställe i diagrammet, men inte i ett block. Då kommer det fram en ruta som du kan skriva i. Skriv några tecken. Tryck return och texten syns i diagrammet. Klicka i texten och fram kommer en minieditor och du kan nu skriva en kommentar med flera rader.

3.2.2 Kommandofilen för version R2018b av MATLAB samt tidigare versioner

(Matlab R2019a fungerar ej på samma sätt som R2018b. Därför ges här två versioner av m-filen SimOfficeChairIntegrators.m. Om du använder R2019a, gå till avsnitt 3.2.3.)

Slutligen ska vi skapa en kommandofil i Matlab, en m-fil. Denna fil kommer att innehålla definitioner av parametrar, köra simuleringen och presentera resultatet av simuleringen. En m-fil skapas genom att i Matlabs file-meny välja "new – script". Nedan följer ett exempel på ett lämpligt innehåll av m-filen. (Om kod kopieras från ett pdf-dokument till en m-fil får man i allmänhet problem med enkla apostrofer. I så fall kan man i Matlabs editor byta ut pdf-textens apostrofer mot Matlabs apostrofer.)

Följande m-fil genererar tre simuleringar och plottar tre kurvor i två diagram. Diagrammet görs i en enkel version och en annan version som visar några olika sätt att kontrollera ett diagrams utseende. Den andra versionen ska betraktas som överkurs. Om du undrar över något kommando så använd Matlabs hjälpfunktion genom att skriva help följt av kommandot i kommandofönstret. Exempelvis ger kommandot

```
>> help sim
```

information om kommandot sim.

```
% SimOfficeChairIntegrators.m
%
% 2018 dec 03. Frida Gleisner och Matz Lenells. Kommentarererna på
% svenska är från 2013 och framåt.
% Bearbetning av ett program som gjordes av Anders Hultgren och
% Staffan Karlsson och som utvecklades under 1990-talet och framåt.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This file contains data about the simulation of the office chair
%
% The name of this file should NOT be the same as the one
% containing the simulink model, the slx-file, earlier called mdl-file.
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear % 2019 aug 28: Clear variables and functions from memory.
% I allmanhet bra att
% gora detta eftersom man inte riskerar att gamla data ligger kvar. En
% stor fara med att lata gamla data finnas kvar ar om man utvecklar en
% m-fil i en sadan milja. Man provkor sedan m-filen.
% Antag att m-filen da fungerar pa grund av att gamla
% data finns i arbetsminnet, dvs Workspace. Om man sedan startar om matlab
% och kor m-filen direkt efter start sa fungerar inte m-filen eftersom
% gamla data inte finns kvar. Efter varje omstart av matlab sa rensas
```

```

% arbetsminnet.

% Gravitation acceleration
g=9.8; % [m/s^2]
% Mass of office chair seat
m=5; % [kg]
% Mass of person taking a seat on the office chair
M=75; % [kg]
% Damping constant, value has to be entered before running simulation
b1=400; b2=1200; b3=2400; % [Ns/m]
% Spring constant
k=25920; % [N/m]
% Simulation duration in seconds
simtime=3; % [s]
% Väljer parametrar för simuleringarna så att de plottade kurvorna blir
% släta.
optionvec1 = simset('AbsTol',1e-9, 'RelTol', 1e-6);

optionvec2 = simset(optionvec1,'MaxStep',simtime/500);
b=b1;
% sim-kommandot är ett kommando vars funktion
% MathWorks har ändrat under årens lopp.
% Med kommandot
% >>help sim
% får du omfattande information om kommandot.
% Anropet här ger att variabeln "time1" innehåller
% simuleringens tidsvektor som i detta fall innehåller 510 element.
% Variabeln "yout.Data" innehåller motsvarande datapunkter.
% Exempelvis gäller att element 117 i Variabeln "yout.Data"
% innehåller uppskattade värdet av y(time1[117]).
% Parametern 'OfficeChairIntegrators' talar om
% vilken differentialekvation som "sim" ska lösa.
% Parametern "simtime" ger simuleiringsstiden.
% parametern "optionvec2" innehåller de parametrar som styr
% simuleringen. Exempelvis feltoleranser.
time1=sim('OfficeChairIntegratorsLisaBerg',simtime, optionvec2);
% Vid utveckling av matlabprogram får man ibland felutskrift
% pga att en variabel har en annan struktur än den man tror.
% Antag att man är osäker på variabeln "time1".
% Då kan man i kommandofönstret skriva >>time1
% och trycka return. I vårt fall skulle då 510 värden skrivas
% ut i kommandofönstret. Därför om man tror att variabeln
% är en vektor är det i allmänhet bättre att skriva följande
% programrad. "sizetime" kommer då bli en radvektor med
% två tal som anger matrisdimensionen för "time1".
% Se utskrift från kommandofönstret som finns direkt efter
% denna m-fil.
sizetime=size(time1)

% Blocket "To Workspace" kan lagra variabler i olika format. Default är
% "Timeseries". Kommandot >>yout skriver ut formatet, dvs variabelns
% datastruktur i kommandofönstret.

```

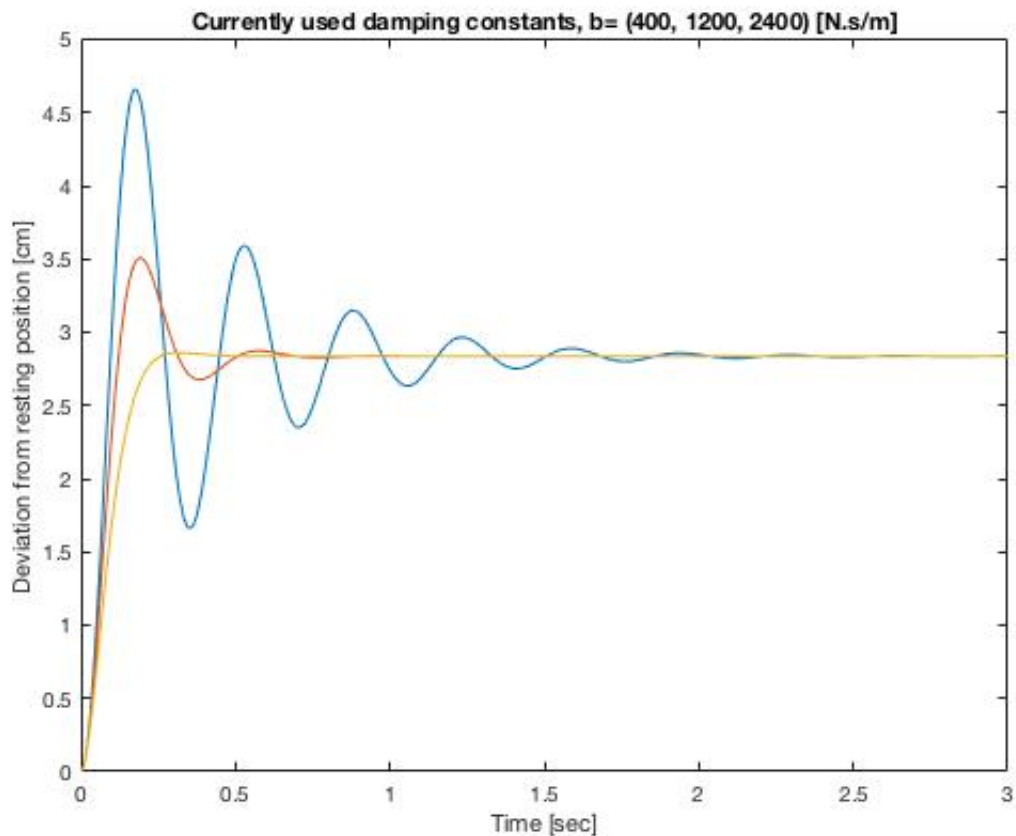


Figure 3.6: Diagram som visar stolsitsens läge, $y(t)$, för tre olika värden på dämpningsfaktorn b .

```
% Ett annat format som tidigare var default är "Array".
% I avsnitt 3.3.3: Frivillig uppgift som behandlar valet "Array"
% som kan göras i blocket "To Workspace"
% kan du se hur man använder data från ett block som använder formatet
% "Array".

yout
% Föregående kommando ger datastrukturen. Av dess utseende
% sluter vi oss till att vi får data med följande tilldelningssats.
x1=yout.Data;
% Om yout inte har formatet "Timeseries" utan har formatet
% "Structure", sätt då x1=yout.signals.values; Motsvarande för x2 och x3.
% För att man ska kunna veta hur tilldelningen till x1 ska göras så kan man
% först skriva kommandot >>yout. Då får man veta vilken datastruktur
% yout har på översta nivån. Man gissar att "yout.signals" innehåller
% data varför man skriver kommandot >>yout.signals. Därvid ser man att
% variabeln "yout.signals" har posten values.
% Vi simulerar systemet en andra gång.
b=b2;
time2=sim('OfficeChairIntegratorsLisaBerg',simtime, optionvec2);
x2=yout.Data;
%x2=yout.signals.values;
```

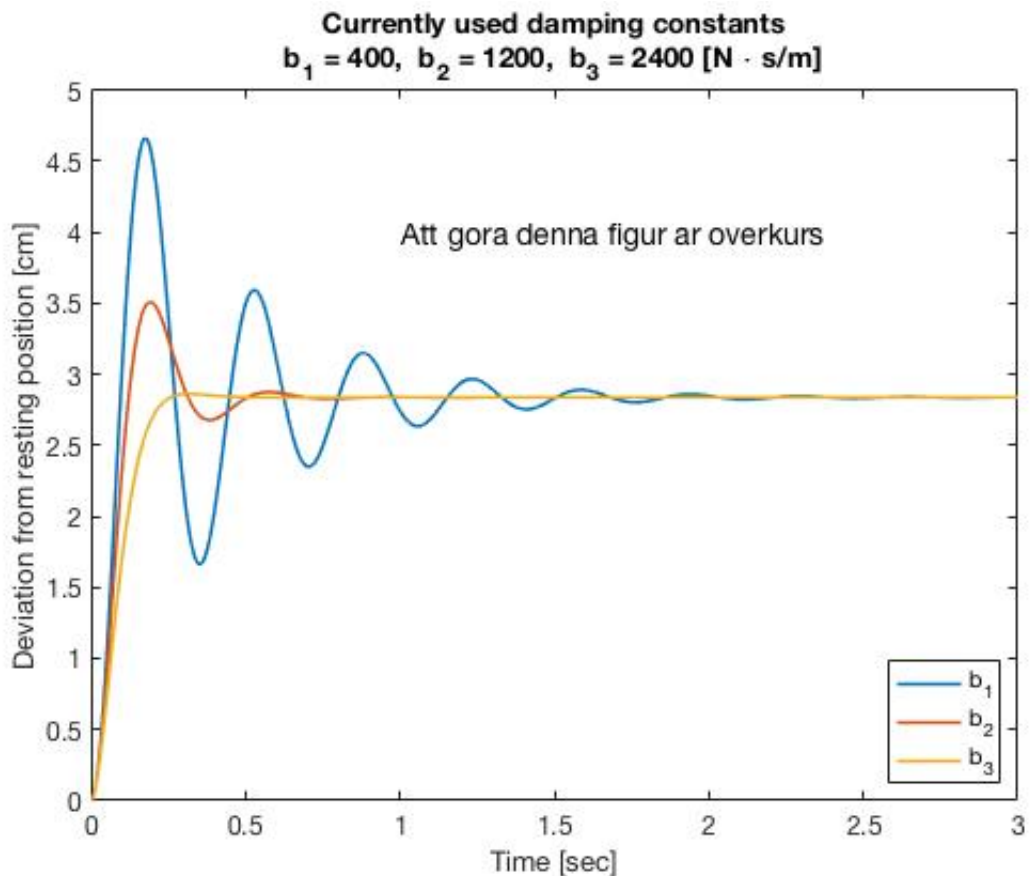


Figure 3.7: Diagram som visar stolsitsens läge, $y(t)$, för tre olika värden på dämpningsfaktorn b . Till skillnad från när diagrammet i figur 3.6 görs så används här kommandon för att styra diagrammets utformning.

```
% Vi simulerar systemet en tredje gång.
b=b3;
time3=sim('OfficeChairIntegratorsLisaBerg',simtime, optionvec2);
x3=yout.Data;
%x3=yout.signals.values;
% Plot the simulation result, time versus deviation of seat from resting
% position. Here the deviation is scaled from meters to centimetres
figure(1)
plot(time1,x1.*100, time2,x2.*100, time3,x3.*100)
% Label of the plot
% De tre punkterna som finns sist i nasta kommandorad anger for
% systemet (interpretatorn) att foljande rad hor till kommandot.
% Gor man bara radbyte sa kommer systemet att protestera.
title(['Currently used damping constants, b= (',num2str(b1),', ',',...
      num2str(b2),', ', ', num2str(b3),', ') [N.s/m]'])
% Guide text of y-axis
ylabel('Deviation from resting position [cm]')
% Guide text of x-axis
xlabel('Time [sec]')
```

% Denna del ar overkurs. Har gors ytterligare ett diagram som visar några

```

% olika sätt att kontrollera ett diagrams utseende.
% Se dokumentation: Documentation Home -> MATLAB -> Graphics ->
% -> Formatting and Annotation

figure(2)
lines = plot(time1,x1.*100, time2,x2.*100, time3,x3.*100);
lines(1).LineWidth = 1.5;
lines(2).LineWidth = 1.5;
lines(3).LineWidth = 1.5;
% Label of the plot
rowstr1 = 'Currently used damping constants';
rowstr2 = ['b_1 = ',num2str(b1),', ', ' ',...
          ' b_2 = ',num2str(b2),', ', ' ', ' b_3 = ',num2str(b3), ' ...
          ' [N \cdot s/m]'];
title({rowstr1, rowstr2}, 'FontSize',14)
% Följande kommando fungerar inte på alla datorer. Därför har det
% kommenterats bort och ersatts av ett som fungerar på alla datorer.
%legend('b_1', 'b_2', 'b_3', 'location', 'southeast', 'FontSize',14)
legend('b_1', 'b_2', 'b_3', 'location', 'southeast');
% Guide text of y-axis
ylabel('Deviation from resting position [cm]', 'FontSize',14)
% Guide text of x-axis
xlabel('Time [sec]', 'FontSize',16)
ax = gca;
ax.FontSize = 12
% Bring the plot window to the foreground
shg

```

När man kör programmet ovan så fås följande utskrift i Matlabs kommandofönster (Command Window) under förutsättning att variabeln "yout" har strukturen "Timeseries".

```

>> SimOfficeChairIntegrators

sizetime =

    510      1

timeseries

Common Properties:
    Name: ''
    Time: [510x1 double]
    TimeInfo: [1x1 tsdata.timemetadata]
    Data: [510x1 double]
    DataInfo: [1x1 tsdata.datametadata]

More properties, Methods

>>

```

Av utskriften ovan syns att variabeln *yout* i sig innehåller tidsvektorn och vi hade kunnat använda vektorn *yout.Time* istället för *time1* som vi använde.

3.2.3 Kommandofilen som genererar simuleringarna när R2019a används

Slutligen ska vi skapa en kommandofil i Matlab, en m-fil. Denna fil kommer att: innehålla definitioner av parametrar, köra simuleringen och presentera resultatet av simuleringen. En m-fil skapas genom att i Matlabs file-meny välja "new – script". Nedan följer ett exempel på ett lämpligt innehåll av m-filen. (Om kod kopieras från ett pdf-dokument till en m-fil får man i allmänhet problem med enkla apostrofer. I så fall kan man i Matlabs editor byta ut pdf-textens apostrofer mot Matlabs apostrofer.)

Följande m-fil genererar tre simuleringar och plottar tre kurvor i två diagram. Diagrammet görs i en enkel version och en annan version som visar några olika sätt att kontrollera ett diagrams utseende. Den andra versionen ska betraktas som överkurs. Om du undrar över något kommando så använd Matlabs hjälpfunktion genom att skriva `help` följt av kommandot i kommandofönstret. Exempelvis ger kommandot

```
>> help sim
```

information om kommandot `sim`.

```
% SimOfficeChairIntegrators.m
%
% OBSERVERA! FUNGERAR EJ FOR R2018b OCH DARMED TROLIGEN EJ HELLER FOR
% TIDIGARE VERSIONER.

% 2019 aug 28. Frida Gleisner och Matz Lenells. Kommentarererna på svenska
% ar fran 2013 och framat.
% Bearbetning av ett program som gjordes av Anders Hultgren och
% Staffan Karlsson och som utvecklades under 1990-talet och framat.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This file contains data about the simulation of the office chair
%
% The name of this file should NOT be the same as the one
% containing the simulink model, the slx-file, earlier called mdl-file.
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear % Clear variables and functions from memory. I allmanhet bra att
% gora detta eftersom man inte riskerar att gamla data ligger kvar. En
% stor fara med att lata gamla data finnas kvar ar om man utvecklar en
% m-fil i en sadan milja. Man provkor sedan m-filen.
% Antag att m-filen da fungerar på grund av att gamla
% data finns i arbetsminnet, dvs Workspace. Om man sedan startar om matlab
% och kor m-filen direkt efter start sa fungerar inte m-filen eftersom
% gamla data inte finns kvar. Efter varje omstart av matlab sa rensas
% arbetsminnet.
% Gravitation acceleration
g=9.8;
% Mass of office chair seat
m=5;
% Mass of person taking a seat on the office chair
M=75;
% Damping constant, value has to be entered before running simulation
b1=400; b2=1200; b3=2400;
% Spring constant
k=25920;
```

```

% Simulation duration in seconds
simtime=3;
% Väljer parametrar för simuleringarna så att de plottade kurvorna blir
% släta.
optionvec1 = simset('AbsTol',1e-9, 'RelTol', 1e-6);

optionvec2 = simset(optionvec1,'MaxStep',simtime/500);
b=b1;
% sim-kommandot är ett kommando vars funktion
% MathWorks har ändrat under arens lopp.
% Med kommandot
% >>help sim
% får du omfattande information om kommandot.

% Parametern 'OfficeChairIntegrators' talar om
% vilken differentialekvation som "sim" ska lösa.
% Parametern "simtime" ger simuleringstiden.
% parametern "optionvec2" innehåller de parametrar som styr
% simuleringen. Exempelvis feltoleranser.
simOut = sim('OfficeChairIntegrators',simtime, optionvec2);
% Vid utveckling av matlabprogram får man ibland felutskrift
% pga att en variabel har en annan struktur än den man tror.
% Antag att man är osäker på variabeln "simOut".
% Då kan man i kommandofönstret skriva >> simOut
% och trycka return. Har skriver vi direkt i koden
% simOut och när detta program körs så skriver programmet ut datastrukturen
% för simOut.

simOut

% I detta fall får man utskriften (så när som på %-tecknen)

%simOut =

%   Simulink.SimulationOutput:
%
%               tout: [507x1 double]
%               yout: [1x1 timeseries]
%
%   SimulationMetadata: [1x1 Simulink.SimulationMetadata]
%   ErrorMessage: [0x0 char]

% Vektorn tout finns default i simOut och innehåller de tider som
% integrationsrutinen använder när den simulerar vår differentialekvation.
% Om man skriver kommandot
% >> simOut.tout
% så skrivs cirka 500 tidsvärden ut på skärmen. Det första värdet är 0 och
% det sista värdet är 3.0 eftersom simtime=3.

% Datastrukturen yout finns som en understruktur pga vårt block
% "To Workspace".

```



```

% Blocket "To Workspace" kan lagra variabler i olika format. Default ar
% "Timeseries". Kommandot >> simOut.yout skriver ut formatet, dvs
% variabelns datastruktur i kommandofönstret. Har skriver vi direkt i koden
% simOut.yout och nar detta program körs så skriver programmet ut
% datastrukturen för simOut.yout.
simOut.yout %Detta fungerar inte när R2018b används.

% Matlab skriver ut följande på skarmen

%    timeseries

%    Common Properties:
%           Name: ''
%           Time: [507x1 double]
%           TimeInfo: [1x1 tsdata.timemetadata]
%           Data: [507x1 double]
%           DataInfo: [1x1 tsdata.datametadata]

%    More properties, Methods

% Slut på utskriften. Som synes innehåller datastrukturen flera olika
% understrukturer. Tiden ligger i Time, dvs i simOut.yout.Time och
% y-vardena ligger i simOut.yout.Data.
% Man kan plotta ut y-vardena som funktion av tiden med kommandot
% >> plot(simOut.yout.time, simOut.yout.Data)
% (Man kan alltså erhålla tidsvardena på två olika sätt, med simOut.tout
% och med simOut.yout.Time.)
% Nedan använder vi ett enklare sätt att plotta denna funktion.
yout1 = simOut.yout;

% Vi simulerar systemet en andra gång.
b=b2;
simOut = sim('OfficeChairIntegrators',simtime, optionvec2);
yout2 = simOut.yout;

% Vi simulerar systemet en tredje gång.
b=b3;
simOut = sim('OfficeChairIntegrators',simtime, optionvec2);

yout3 = simOut.yout;

% Plot the simulation result, time versus deviation of seat from resting
% position. Here the deviation is scaled form meters to centimetres
figure(1)
hold off

plot(yout1*100)
hold on
plot(yout2*100)
hold on
plot(yout3*100)
hold off

```

```

% Label of the plot
% De tre punkterna som finns sist i nasta kommandorad anger for
% systemet (interpretatorn) att foljande rad hor till kommandot.
% Gor man bara radbyte sa kommer systemet att protestera.
title(['Currently used damping constants, b= (',num2str(b1),', ',...
      num2str(b2),', ', ', num2str(b3), ' ) [N.s/m]'])
% Guide text of y-axis
ylabel('Deviation from resting position [cm]')
% Guide text of x-axis
xlabel('Time [sec]')

% Denna del ar overkurs. Har gors ytterligare ett diagram som visar några
% olika satt att kontrollera ett diagrams utseende.

% Se dokumentation: Documentation Home -> MATLAB -> Graphics ->
% -> Formatting and Annotation
figure(2)
t1 = yout1.Time;
x1 = yout1.Data;
t2 = yout2.Time;
x2 = yout2.Data;
t3 = yout3.Time;
x3 = yout3.Data;
lines = plot(t1, x1*100, t2, x2*100, t3, x3*100);
lines(1).LineWidth = 1.5;
lines(1).Color = 'blue';
lines(2).LineWidth = 1.5;
lines(2).Color = 'red';
lines(3).LineWidth = 1.5;
lines(3).Color = 'magenta';

% Label of the plot
rowstr1 = 'Currently used damping constants';
rowstr2 = ['b_1 = ',num2str(b1),', ', ',...
          ' b_2 = ',num2str(b2),', ', ', ' b_3 = ',num2str(b3), ' ...
          ' [N \cdot s/m]'];
title({rowstr1, rowstr2}, 'FontSize',14)
% Kommentar fran hosten 2018.
% Foljande kommando fungerar inte pa alla datorer. Darfor har det
% kommenterats bort och ersatts av ett som fungerar pa alla datorer.
% legend('b_1', 'b_2', 'b_3', 'location', 'southeast', 'FontSize',14)
legend('b_1', 'b_2', 'b_3', 'location', 'southeast');
text(1, 0.04*100, 'Att göra denna figur är överkurs', 'FontSize',14)
% Guide text of y-axis
ylabel('Deviation from resting position [cm]', 'FontSize',14)
% Guide text of x-axis
xlabel('Time [sec]', 'FontSize',16)
ax = gca;
ax.FontSize = 12;
% Bring the plot window to the foreground
shg

```

3.3 Metod 2 - Systemet beskrivet på tillståndsform

M-filen som finns i detta avsnitt är gjord för matlab R2018b och fungerar nästan säkert inte för R2019a.

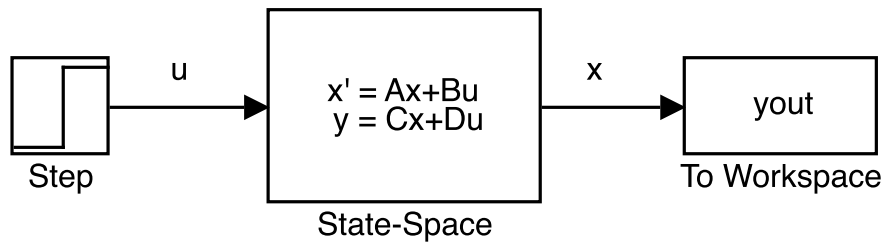


Figure 3.8: Blockschema då systemet beskrivs på tillståndsform.

Block Parameters: State-Space

State Space

State-space model:
 $\dot{x}/dt = Ax + Bu$
 $y = Cx + Du$

Parameters

A:
A

B:
B

C:
C

D:
D

Initial conditions:
0

Absolute tolerance:
auto

State Name: (e.g., 'position')
"

Buttons: ? OK Cancel Help Apply

Figure 3.9: Genom att dubbelklicka på blocket *State-Space* i blockschemat "OfficeChairState" skrivs ett fönster ut som har vissa fält skrivbara. I detta fönster har textfälten för "A", "B", "C" och "D" ändrats till "A", "B", "C" och "0" respektive.

Vi inför hjälpvariablerna $x_1 = y$ och $x_2 = \dot{y}$ och kan då skriva om differentialekvation

(2.10) enligt

$$(m+M)\dot{x}_2(t) + bx_2(t) + kx_1(t) = u(t), \quad (3.2)$$

där

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= -\frac{k}{m+M}x_1 - \frac{b}{m+M}x_2 + \frac{1}{m+M}u(t). \end{aligned} \quad (3.3)$$

Hjälpvariablerna x_1 och x_2 kallas för tillstånd och ekvationerna ovan kallas för tillståndsekvationer. För att göra vår andra typ av simulering i Simulink skriver vi tillståndsekvationerna på matrisform. Tillstånden läggs in i m-filen som matriser. För ekvation 3.3 fås matriserna

$$\underbrace{\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix}}_{x'} = \underbrace{\begin{pmatrix} 0 & 1 \\ \frac{-k}{m+M} & \frac{-b}{m+M} \end{pmatrix}}_A \underbrace{\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}}_x + \underbrace{\begin{pmatrix} 0 \\ \frac{1}{m+M} \end{pmatrix}}_B u. \quad (3.4)$$

Simulink har blocket State-Space som är gjort för simulering av linjära system skrivna på tillståndsform, se figur 3.8. Ekvation (3.4) svarar mot övre raden i blocket State-Space. För att Simulink ska ge oss läget på stolen vill vi ha y som utsignal. Variabeln motsvaras av x_1 och fås genom matrismultiplikationen

$$y = \underbrace{\begin{pmatrix} 1 & 0 \end{pmatrix}}_C \underbrace{\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}}_x. \quad (3.5)$$

Vårt C är detsamma som C i rad två i blocket State-Space.

3.3.1 Beskrivning av hur man ritar blockschemat i figur 3.8

1. Öppna ett nytt modellfönster enligt instruktionerna i avsnitt 3.2.1, spara filen som OfficeChairStatesLisaBerg.slx där du istället för Lisa Berg skriver ditt namn.
2. Kopiera och använd blocken Step och To Workspace från din tidigare modell, då behöver blocken inte justeras.
3. Blocket State-Space hittar du under "Continuous" i fönstret "Simulink Library Browser". Dubbelklicka på blocket och skriv in "A", "B", "C" och "D" i de fyra översta fälten enligt figur 3.9.

3.3.2 Kommandofilen som genererar simuleringarna

```
% SimOfficeChairState.m
% 2018 nov 27. Frida Gleisner och Matz Lenells. Se kommentarer till
% SimOfficeChairIntegrators.m
clear % 2019 aug 28.
g=9.8;
m=5;
M=75;
b1=400; b2=1200; b3=2400;
k=25920;
```

```

simtime=3;
A=[0, 1;
-k/(m+M), -b1/(m+M)];
B= [ 0 ; 1/(m+M)];
C=eye(2); % eye(2) är enhetsmatrisen för 2x2-matriser.
%Kommandot C = [1, 0; 0, 1]; fungerar lika bra.
D = zeros(2, 1);
E = eye(2); % Om man väljer blocket "Descriptor State-Space"
% istället för blocket "State-Space" så måste man ange matrisen E.
optionvec1 = simset('AbsTol',1e-9, 'RelTol', 1e-6);
optionvec2 = simset(optionvec1,'MaxStep',simtime/5000);
%Väljer parametrar för simuleringarna så att plottade kurvor blir släta.
b=b1;
time1=sim('OfficeChairState',simtime, optionvec2);
sizetime=size(time1)
yout.Name='yout';
yout
x1=yout.Data(:, 1);
A=[0, 1;
-k/(m+M), -b2/(m+M)];
time2=sim('OfficeChairState',simtime, optionvec2);
x2=yout.Data(:, 1);
A=[0, 1;
-k/(m+M), -b3/(m+M)];
time3=sim('OfficeChairState',simtime, optionvec2);
x3=yout.Data(:, 1);
lines = plot(time1,x1.*100, time2,x2.*100, time3,x3.*100);
lines(1).LineWidth = 1.5;
lines(2).LineWidth = 1.5;
lines(3).LineWidth = 1.5;
title(['Currently used damping constants,b= (',num2str(b1),', ',',...
num2str(b2),', ', ', num2str(b2), ' ) [N.s/m]'], 'FontSize',14)
legend('b_1', 'b_2', 'b_3', 'location', 'southeast', 'FontSize',14)
ylabel('Deviation from resting position [cm]', 'FontSize',14)
xlabel('Time [sec]', 'FontSize',14)
ax = gca;
ax.FontSize = 12;
shg

```

3.3.3 Frivillig uppgift som behandlar valet "Array" som kan göras i blocket "To Workspace"

Blockschema 3.10 är en utvidgning av blockschema 3.8. Det som har kommit till är blocket "To Workspace 1" där "variable name" är x .

Figur 3.4 visar det fönster som används för att bestämma parametrarna för block "To Workspace". Till höger om textfältet "Save format:" finns ett annat skrollbart textfält där användaren kan välja olika format för att spara data i. I figur 3.4 så visas att man där har valt "Timeseries" vilket är default. Ett annat format är "Array" och det formatet användes under många år som default. I fallet med blocket "To Workspace 1" som finns i figur 3.10 så har formatet valts till "Array". För att plotta data i detta fall kan man lägga följande kod efter koden som visas i avsnitt 3.3.2.

```
figure(12)
```

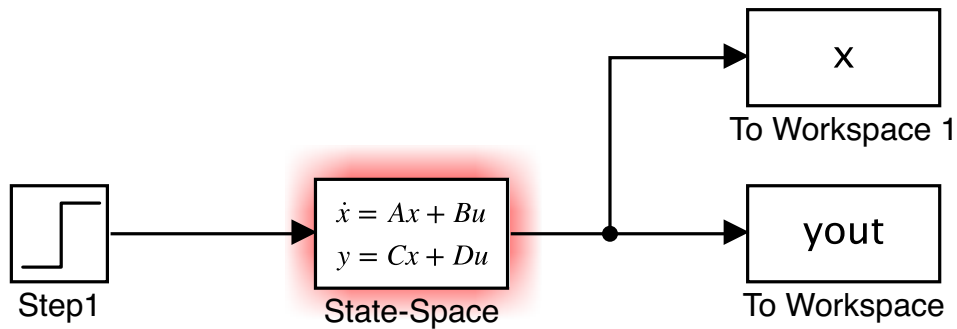


Figure 3.10: Ett blockschema som är utvidgning av det schema som finns i figur 3.8. Tillägget består av blocket "To Workspace 1" som sparar tillståndsvektor x .

```

plot(time3, x(:,1)*100, time3, x(:,2)*100) % Ger ett diagram med tva
% kurvor. Inget mer behöver skrivas för att diagrammet ska plottas.

% Nedan finns kommandon för att diagrammet ska bli enklare att läsa.

% Skapande av en text som innehåller 'x_1' tillsammans med en pil.
% Kurvorna som presenteras består av punkter samt av linjer mellan dessa
% punkter. För användaren betyder det att när man vill veta y-värdet
% (ordinatans värde) för en viss tidpunkt t så behöver man göra en
% interpolation mellan två punkter i diagrammet. En sådan interpolation
% kan göras med kommandot "interp1".
timeForWritingLabel1 = 1; %[s] Tidpunkt för placering av 'x_1'.
interpolate_data1 = interp1(time3, x(:,1)*100, timeForWritingLabel1)
text(timeForWritingLabel1, interpolate_data1 - 1, '\uparrow x_1', ...
      'FontSize', 14 )

% Skapande av en text som innehåller 'x_2' tillsammans med en pil.
timeForWritingLabel2 = 0.15;
interpolate_data2 = interp1(time3, x(:,2)*100, timeForWritingLabel2)
text(timeForWritingLabel2, interpolate_data2, '\leftarrow x_2', ...
      'FontSize', 14)

% Utskrift av titel för diagrammet.
title(['Tillståndsvektorn x och dess två signaler i fallet då b=', ...
      num2str(b3), ' [N.s/m]'])

% Angivande av skalor för y-axeln. Observera att det finns två rader.
columnStr1 = 'x_1 Avvikelse från viloposition [cm]';
columnStr2 = 'x_2 Avvikelsen hastighet från viloposition [cm/s]';
ylabel({columnStr1, columnStr2}, 'FontSize', 14)

xlabel('Time [sec]', 'FontSize', 14)

% Kommandon för att axlarnas graderingar ska skrivas med en viss
% fontstorlek.
ax = gca;
ax.FontSize = 12;

```

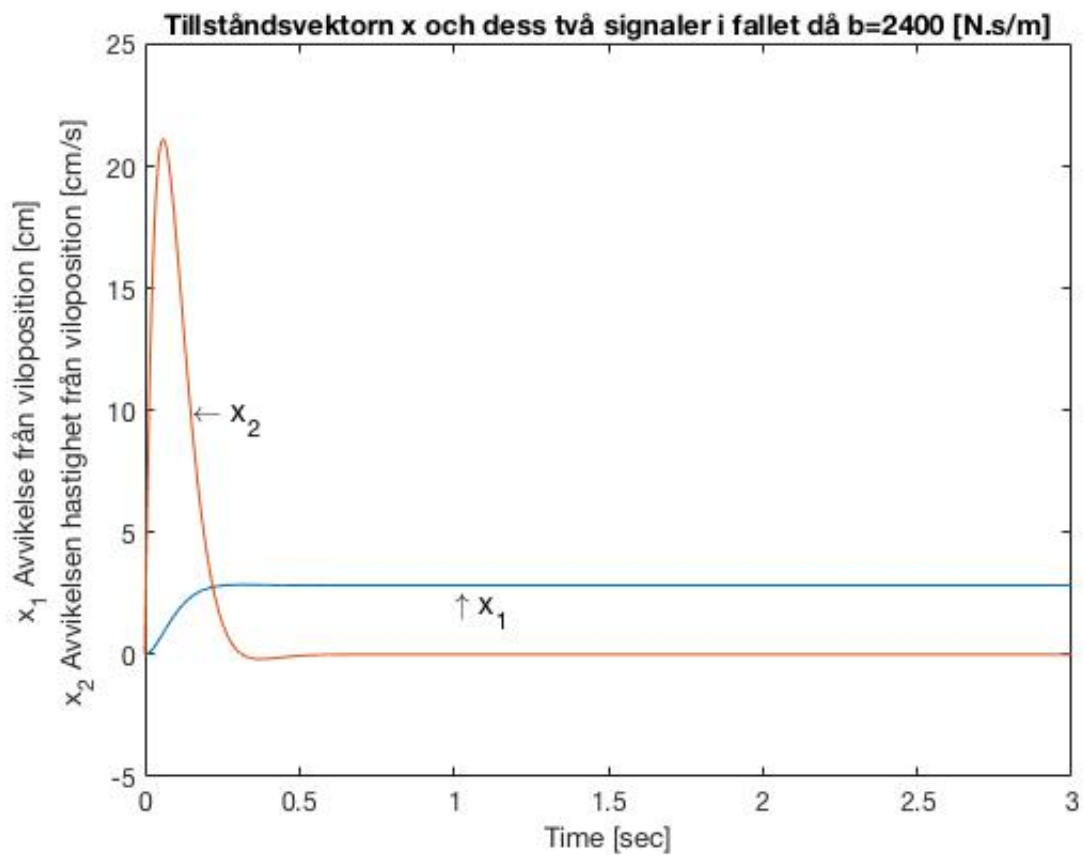


Figure 3.11: Diagram som fås när koden i avsnitt 3.3.3 körs. Diagrammet visar graferna för de två tillståndsvariablerna x_1 och x_2 i fallet med $b = 2400$.

Koden ovan ger figur 3.11, som visar två olika skalor i y-led samt texter, i detta fall " $\leftarrow x_1$ " och " $\uparrow x_2$ " i själva diagrammet.

3.3.4 Kommentar om val av lösare och parametrar vid simulering

Det finns olika numeriska algoritmer för att lösa en differentialekvation. För enkla typexempel fungerar nästan alltid den lösare som är default.

3.4 Metod 3 - Systemet beskrivet med sin överföringsfunktion

M-filen som finns i detta avsnitt är gjord för matlab R2018b och fungerar nästan säkert inte för R2019a.

Vi applicerar laplacetransformen på ekvation (2.12) och får

$$s^2Y(s)(m+M) + sY(s)b + kY(s) = U(s). \quad (3.6)$$

Överföringsfunktionen

$$G(s) = \frac{Y(s)}{U(s)} = \frac{1}{(m+M)s^2 + bs + k}$$

ger ett uttryck för hur insignalen $u(t)$ påverkar utsignalen $x(t)$. Överföringsfunktionen $G(s)$ illustreras i figur 3.12. För att simulera systemet beskrivet i figur 3.12 så använder

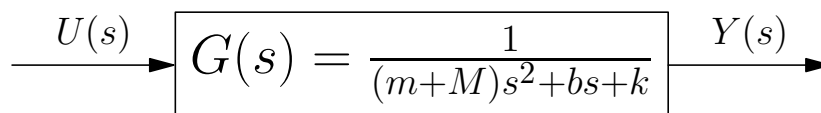


Figure 3.12: Överföringsfunktion för kontorsstolen.



Figure 3.13: Blockschemat då systemet beskrivs med sin överföringsfunktion.

man i Simulink blockschemat som visas i figur 3.13 och skriver dessutom en kommandofil, en m-fil, som anropar den programkod som svarar mot blockschemat i figur 3.13. Här följer anvisningar för att rita blockschemat. Därefter följer koden som finns i m-filen. Koden i m-filen genererar för tre olika b-värden simuleringar av stolsystemet samt ett diagram som visar stolsitsens läge, $y(t)$, som funktion av tiden, se figur 3.7.

3.4.1 Beskrivning av hur man ritar blockschemat i figur 3.13

1. Öppna ett nytt modellfönster enligt instruktionerna i avsnitt 3.2.1, spara filen som SimuOfficeChairSimple.slx.
2. Kopiera och använd blocken Step och To Workspace från din tidigare modell, då behöver blocken inte justeras.

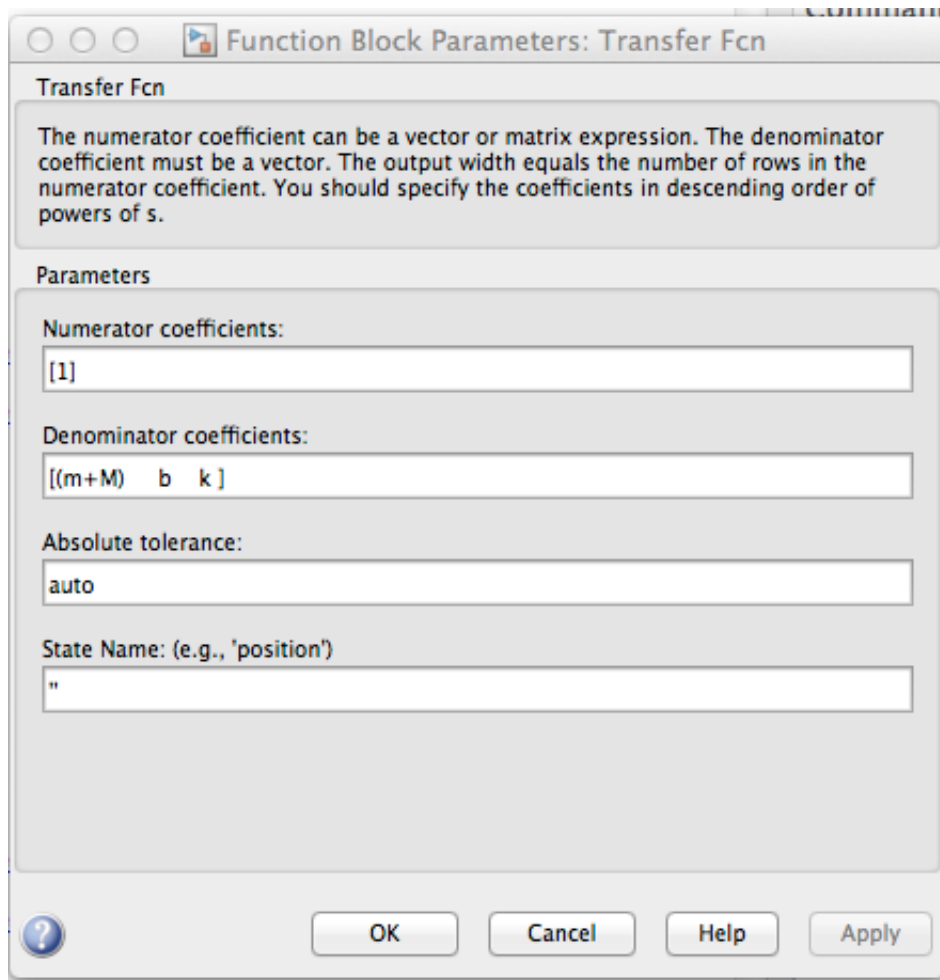


Figure 3.14: Genom att dubbelklicka på blocket `Transfer Fcn` i blockschema "OfficeChairSimple" skrivs ett fönster ut som har vissa fält skrivbara. I detta fönster har textfälten för "Numerator coefficients" och "Denominator coefficients" ändrats till `[1]` och radvektorn `[(m+M) b k]` respektive.

3. Blocket `Transfer Fcn` hittar du under "Continuous" i fönstret "Simulink Library Browser". Dubbelklicka på blocket och i fältet "Numerator coefficients" skrivs `[1]` samt i fältet "Denominator coefficients" skrivs `[(m+M) b k]`, se figur 3.14. Markera blocket genom att klicka på det, öka blockets storlek genom att dra i ett av hörnen så att texten framträder.

3.4.2 Kommandofilen som genererar simuleringarna

```
%SimOfficeChairTrfFnc.m

% 2013 juli. Frida Gleisner och Matz Lenells. Se kommentarer till
% SimOfficeChairIntegrators.m
clear % 2019 aug 28.
g=9.8;
m=5;
M=75;
b1=400; b2=1200; b3=2400;
k=25920;
```

```

simtime=3;
optionvec1 = simset('AbsTol',1e-9, 'RelTol', 1e-6);
optionvec2 = simset(optionvec1,'MaxStep',simtime/500);
b=b1;
time1=sim('OfficeChairTrfFnc',simtime, optionvec2);
sizetime=size(time1)
yout.Name='yout';
yout
x1=yout.Data;
b=b2;
time2=sim('OfficeChairTrfFnc',simtime, optionvec2);
x2=yout.Data;
b=b3;
time3=sim('OfficeChairTrfFnc',simtime, optionvec2);
x3=yout.Data;
plot(time1,x1.*100, time2,x2.*100, time3,x3.*100)
title(['Currently used damping constants,b= (',num2str(b1),',', ' ',...
      num2str(b2),',', ' ', num2str(b2),', ' ) [N.s/m]'])
ylabel('Deviation from resting position [cm]')
xlabel('Time [sec]')
shg

```

3.5 Simulering av olinjära system

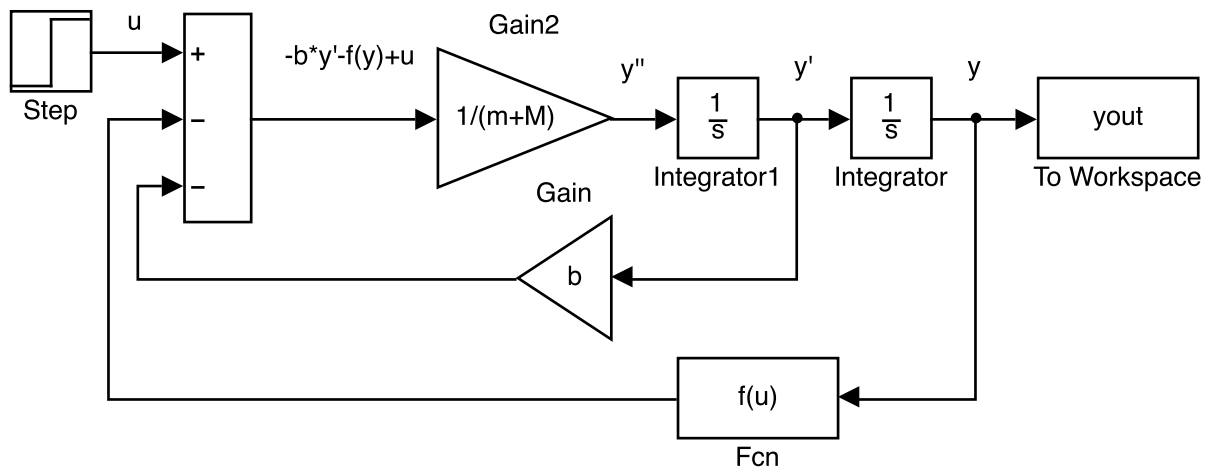


Figure 3.15: Blockschemat för ett olinjärt system.

Avsnitten 3.3 och 3.4 visar att systemet (2.10) kan simuleras med hjälp av blockscheman som är mycket enklare än blockschemat i figur 3.1. Metoderna som visas i avsnitten 3.3 och 3.4 kan dock inte utvidgas till att behandla olinjära system, utom i mycket speciella fall av olinjäriteter, exempelvis att insignalen kvadreras. Metod 1, beskriven i avsnitt 3.2, kan dock modifieras för att klara olinjära system eftersom blockschemat i figur 3.1 är så flexibelt. Låt oss studera fallet med en olinjär fjäder så att vårt system beskrivs av ekvationen

$$(m + M)\ddot{z}(t) + b\dot{z}(t) + k_1z(t) + k_2z^3(t) = u(t). \quad (3.7)$$

Här betecknar z avståndet ner till fjädern räknat från fjäderns viloläge. När ingen person sitter på stolen så gäller att $u(t) = mg$. När en person sitter på stolen så ökar kraften och

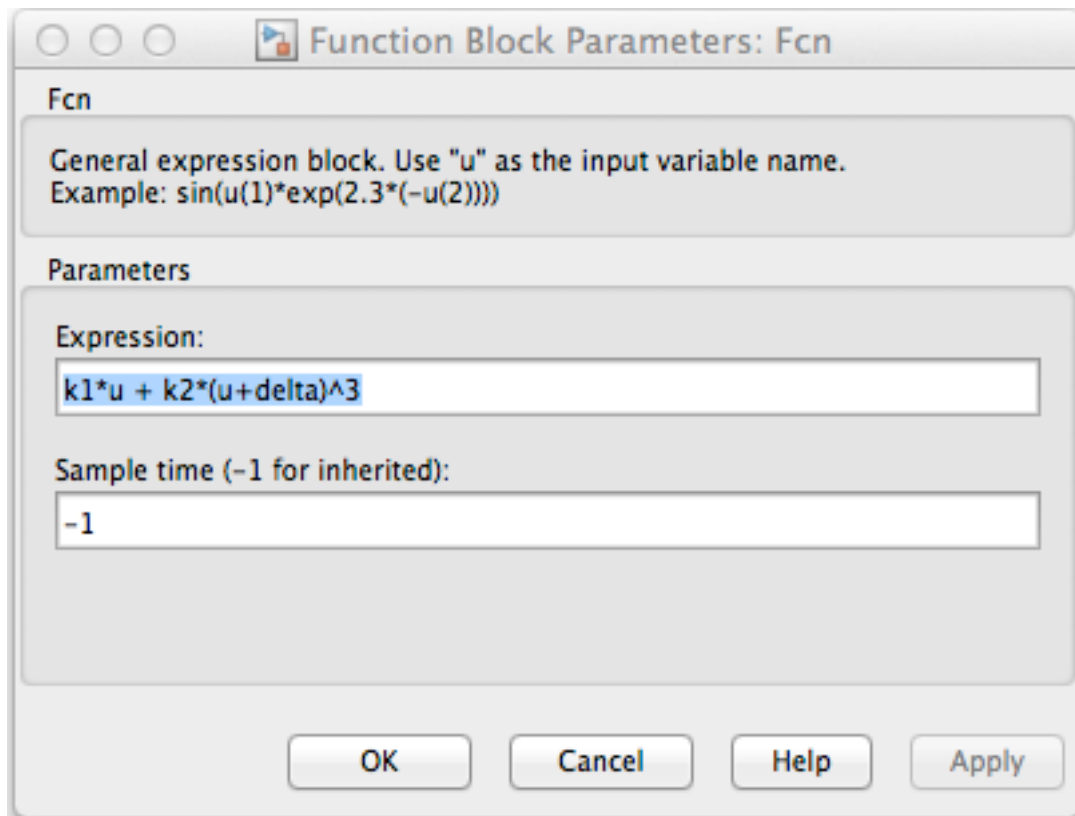


Figure 3.16: Om man i schemat, se figur 3.15, dubbelklickar på blocket Fcn kommer detta fönster upp.

då blir insignalen $u(t) = (m + M)g$.

En stol där ingen sitter kommer efter ett kort tag att ställa in sig så att $z(t)$ blir konstant. Vi betecknar konstanten med δ . Då gäller att

$$k_1\delta + k_2\delta^3 = mg. \quad (3.8)$$

I fallet när $k_2 = 0$ svarande mot det linjära fallet så är det enkelt att beräkna δ . Om $k_2 \neq 0$ så är det besvärligare att bestämma ett numeriskt värde på δ .

Det kan vara praktiskt att göra som vi gjort tidigare, mäta stolens position relativt viloläget $z = \delta$. Som förut betecknar vi detta avstånd med y och här gäller att $z = y + \delta$. Ekvation (3.7) kan då skrivas som

$$(m + M)\ddot{y}(t) + b\dot{y}(t) + k_1(y(t) + \delta) + k_2(y(t) + \delta)^3 = (M + m)g. \quad (3.9)$$

i fallet då en person sitter på stolen. Man kan skriva om denna ekvation som

$$(m + M)\ddot{y}(t) + b\dot{y}(t) + k_1y(t) + k_2(y(t) + \delta)^3 = (M + m)g - k_1\delta. \quad (3.10)$$

Blockschemat i figur 3.1 modifieras till blockschemat i figur 3.15 för att klara av olinjäriteten i ekvation (3.10). I blocket Fcn ska fältet "Expression" fyllas i enligt vad figur 3.16 visar. I blockshemat används beteckning u för två olika storheter, dels för insignalen och dels som argument i den olinjära funktionen $f(u)$. Insignalen $u(t)$ är ett steg med storleken $Mg + mg - k_1\delta$.

För att beskriva olinjära system kan man också använda något som kallas för S-function.

A Härledning av ekvation (2.9)

Antag att ingen sitter på stolen och att den befinner sig i jämvikt. Stolsitsen dras nedåt av jordens dragningskraft med kraften mg . Jämfört med fjäderns neutrala längd så är den förkortad med längden $\delta = mg/k$. Newtons andra lag säger att massan går accelerationen är lika med summa krafter som verkar på massan. Ekvationen blir

$$(m + M)\ddot{y}(t) = -b\dot{y}(t) - k(\delta + y(t)) + Mg + mg. \quad (\text{A.1})$$

I detta fall verkar tre krafter på massan. På grund av jordens dragningskraft verkar kraften $Mg + mg$ nedåt, fjädern ger kraften $k(\delta + y(t))$ som är riktad uppåt och dämparen ger kraften $b\dot{y}(t)$ och den är riktad så att den motverkar rörelse. Konstanten b kallas för friktionskoefficienten. Om man vill vara tydlig med vilken friktion som åsyftas så kan man kalla b för friktionskoefficienten för den viskösa friktionen. Vi ser att ekvationerna (A.1) och (2.9) är ekvivalenta.

B Matlabs fönster

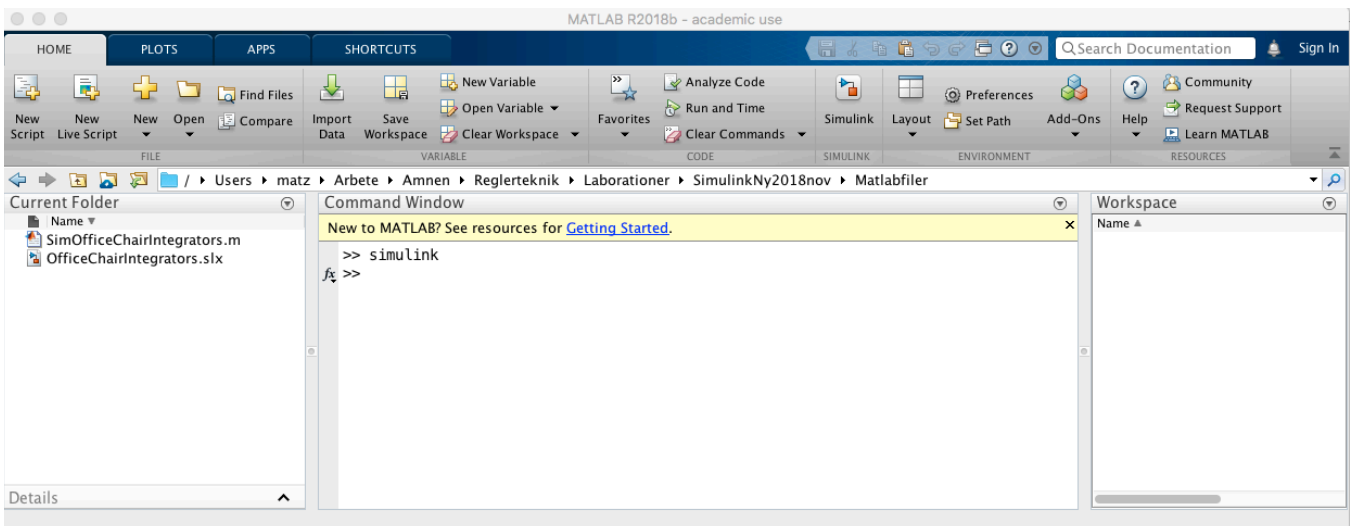


Figure B.1: Matlabs huvudfönster visat när defaultutformningen används. Utformningen kan väljas mha menyn som visas om man klickar på knappen "Layout" som finns i verktygspanelen, se nära övre kanten och en bit till höger om mitten.



Figure B.2: Icon för Simulink.

På webbsidan

https://se.mathworks.com/help/matlab/learn_matlab/desktop.html

ger Mathworks en kort introduktion till Matlab. Där visas också en bild på hur Matlabs huvudfönster ser ut. Figur B.1 visar nästan en identisk bild till den som finns på webbsidan. En sak som skiljer är att i figuren finns en "Simulink"-ikon, se figur B.2 för att se hur ikonen ser ut. Du hittar ikonen strax till höger om mitten och nära fönstrets övre kant.

Överst i fönstret finns en list med fönstrets namn, "MATLAB R2018b - academic use". Därunder finns en list där man bland annat kan välja olika vyer, såsom "HOME", "PLOT** etc, för det som vi här kallar för verktygspanel. Verktygspanelen innehåller många olika knappar med vilka man styra programmet. Om man håller muspekaren stilla över en av knapparna så kommer det fram en hjälptext som ger en kort information om knappen. Under verktygspanelen finns en list med olika knappar och informationer. Om man klickar på femte knappen från vänster, den som har en grön pil som går snett nedåt vänster, kommer det upp ett filsökningsfönster, "Browse for folder". Den knappen är användbar för att ställa in programmet så att "Current Folder" blir den önskade. Figuren visar fallet när katalogen är

/Users/matz/Arbete/Amnen/Reglerteknik/Laborationer/SimulinkNy2018nov/Matlabfiler

Filvägen beror på den dator som används. De tre underfönstren som syns på den nedre halvan av huvudfönstret är "Current Folder", "Command Window" och "Workspace".

C Modellering av stolens rörelse när tillstånd används

Vid modellering kan det oftast vara praktiskt att använda något som kallas tillstånd. Här visas hur stolens rörelse kan modelleras med tillstånd. Låt läget beskrivas av x_1 där vi här utgår ifrån att $x_1 = 0$ då fjädern är viloläge. Positiv riktning är nedåt. Med beteckningar som i avsnitt A betyder det att $x_1 = y + \delta$. Låt hastigheten beskrivas av $x_2 = \dot{x}_1 = \dot{y} + \dot{\delta} = \dot{y}$. Här refererar vi till y och derivatan av y , \dot{y} , för att man lätt ska se hur detta modellbygge hänger ihop med det tidigare modellbygget.

Låt P beteckna ett tvärsnitt av stolpen som håller upp stolsitsen, se figur 2.2. Tre krafter verkar i tvärsnittet P :

$$(M + m)g \quad \text{tyngdkraften} \quad (\text{C.1})$$

$$-kx_1 \quad \text{fjäderkraften} \quad (\text{C.2})$$

$$-bx_2 \quad \text{viskösa friktionskraften} \quad (\text{C.3})$$

Krafterna är alla riktade rakt nedåt. Enligt Newtons andra lag gäller att

$$(M + m) \frac{dx_2}{dt} = -kx_1 - bx_2 - (M + m)g \quad (\text{C.4})$$

Denna ekvation tillsammans med ekvationen $x_2 = \frac{dx_1}{dt}$ ger följande system av differentialekvationer.

$$\begin{cases} \frac{dx_1}{dt} = x_2 \\ \frac{dx_2}{dt} = -\frac{k}{M+m}x_1 - \frac{b}{M+m}x_2 - g \end{cases} \quad (\text{C.5})$$

Detta ekvationssystem kan skrivas på följande sätt

$$\begin{pmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\frac{k}{M+m} & -\frac{b}{M+m} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} g \quad (\text{C.6})$$

Vi inför ett antal storheter och vi definierar derivatan för en kolonnvektor \mathbf{x} .

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \quad \frac{d\mathbf{x}}{dt} = \begin{pmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \end{pmatrix} \quad A = \begin{pmatrix} 0 & 1 \\ -\frac{k}{M+m} & -\frac{b}{M+m} \end{pmatrix} \quad B = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (\text{C.7})$$

Med dessa storheter kan ekvationssystemet skrivas som

$$\frac{d\mathbf{x}}{dt} = A\mathbf{x} + Bg \quad (\text{C.8})$$

D Simulink och insignal-utsignalsystem

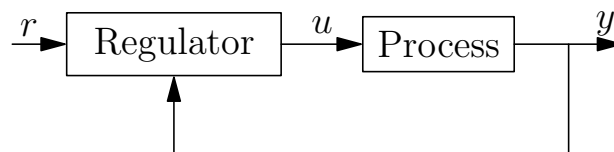


Figure D.1: Blockschemat för ett typiskt reglersystem. I den här framställningen tänker vi oss att blockens beteenden beskrivs av differentialekvationer. Gemensamt för blocken är att de kan ha insignaler och utsignaler.

Ett typiskt reglertekniskt system innehåller en process som ska styras samt en regulator, se blockschemat i figur D.1. Såväl processen som regulatorn är oftast dynamiska system, och sådana tänker vi oss här beskrivna av ordinära differentialekvationer. Blocket "Process" har insignalen u och utsignalen y medan blocket "Regulator" har de två insignalerna r och y och utsignalen u . Simulink är gjort för att kunna simulera reglertekniska system där blockscheman som i figuren kan användas. Användaren ritat blockscheman i en speciell editor.

Dynamiska system kan simuleras direkt i Matlab. Man kan även programmässigt foga ihop system av delsystem där varje delsystem beskrivs av differentialekvationer. På så sätt kan man programmässigt skapa system som i Simulink skapas genom att visuellt koppla ihop olika delsystem. Användandet av Simulink motiveras av att ett blockschema är mycket enklare att förstå än programkod som gör blockschemats kopplingar.

Vi vill betrakta ekvation (2.9) som ett system med en insignal och en utsignal. Positionen y är en naturlig utsignal, däremot finns ingen naturlig insignal. Vi väljer därför att betrakta tyngdkraften från massan M som en insignal u . För att få ett typiskt reglertekniskt system så skulle vi kunna införa en kraft F som påverkar stolsitsen och vi skulle då få $u = Mg + F$. Närliggande exempel är system för aktiv dämpning som förekommer i många sammanhang. Se exempelvis filmen (Kontrollerat 2018 nov 27. Okänt om produkten saluförs.)

<http://www.youtube.com/watch?v=eSi6J-QK1lw>.

E Reproducerbarhet av simuleringssessioner

Grundläggande för tekniskt och vetenskapligt arbete är reproducerbarhet. Gjorda simuleringar ska kunna återupprepas baserat på den information som ges i en rapport. Ett problem med att använda Simulink är att nödvändig information kan döljas i de block som används. För att lösa detta problem har Mathworks, som har Matlab och Simulink

som sina produkter, infört möjligheter att ställa alla parametrar programmässigt. Exempelvis kan man från den m-fil som driver simuleringen i avsnitt 3.2 ställa "Final Value" i blocket Step, se figur 3.1. Vidare har man definierat datastrukturer som om sådana används gör att omfattande information sparas efter en simulering. Exempelvis har vi sett att signaler som sparas med dataformatet "Timeseries" inte bara innehåller den primära signalen utan även variabelns tidsvektor.

Nedan visas hur man kan göra programmässig tilldelning av värdet för "Final Value" i blocket Step, se figur 3.1. Koden är sådan att du direkt kan kopiera koden och lägga in den i m-filen som finns i avsnitt 3.2.

```
% Here follows instructions of how to find
% the parameter names of different blocks's
% parameters. Matlab uses sometimes different
% names for the same parameter. One name in the
% dialog box and another one in commands.
% In the Step box you find the name
% "Final Value" but in the command you use the
% name "After".
%
% There is two methods to obtain the information.
%
% Method 1:
% a) Double click on the the block Step. Then
% a pane with the title "Block Parameters: Step"
% should be visible.
%
% b) Right click on "'Final value" in the pane.
% A text field with the text "What's this?".
% Left click on the text and then another pane
% should be visible. The titel of the pane is
% "Final value". In this pane ther is a table with
% the title "Programmatic Use". It has a row
% with the text: "Block Parameter: After".
%
% Method 2:
% I use "/" as a delimiter between search
% phrases.
% a) Go first to Help --> Documentation
% b ) Then you go down the tree along the
% search path:
% Simulink / Modeling /Configure Models / Blocks
% At this point you should see a window to the right
% with the head line "Blocks".
% c) Go down to the
% subsections "Topics" and "Block Parameters".
% Under "Block Parameters" you will see a list
% of links.
% d) Click on the last one
% "Block-Specific Parameters"
% e) Choose then
```

```
% "Sources Library Block Parameters"
% and you will then look into a long list of
% parameter names. The list is divided in sections.
% f) Go down to a section named "Step".
% There you find the parameter name "After".
%
% To programmatic assign the text "M*g" to the row
% below the text 'Final value' in the block Step 1
% you use the command:
set_param('OfficeChairIntegrator/Step','After','M*g');
% Open the block 'OfficeChair/Step' and look what the
% "Final value" is now. It should be "M*g".
```

Vår erfarenhet av att ställa blocks parametrar programvarumässigt är att arbetet med att göra m-filerna blir så omfattande och att m-filerna i sig blir så stora att det är opraktiskt att göra det i undervisningen.

F Obsoleta programkonstruktioner i Matlab

Mathworks har svårt för att ta bort programkonstruktioner som företaget anser är obsoleta. Om programkonstruktioner tas bort så behöver en användares gamla program skrivas om ifall användaren ska kunna uppdatera sin Matlabversion. Detta kan vara ett skäl till att Mathworks väljer att låta obsoleta programkonstruktioner finnas kvar. Detta gör att Matlab och Simulink blir allt omfångsrikare och svåröverskådligare, samt att språket alltmer bryter mot den princip inom datalogin som säger att ett programspråk inte ska tillåta många olika sätt att lösa samma problem. Dessutom har användarna lärt sig att Mathwork inte är snara att ta bort obsoleta programkonstruktioner.

G Skalning av tiden i en differentialekvation

Inför den skalade tiden $\tau = \frac{1}{\beta}t$. Då gäller att

$$\frac{dy(t(\tau))}{d\tau} = \frac{dy}{dt} \frac{dt}{d\tau} = \beta \frac{dy}{dt} \Rightarrow \frac{dy}{dt} = \frac{1}{\beta} \frac{dy(t(\tau))}{d\tau} \quad (\text{G.1})$$

och att

$$\frac{d^2y(t(\tau))}{d\tau^2} = \frac{d}{d\tau} \left(\beta \frac{dy}{dt} \right) = \beta \frac{d^2y}{dt^2} \frac{dt}{d\tau} = \beta^2 \frac{d^2y}{dt^2} \Rightarrow \frac{d^2y}{dt^2} = \frac{1}{\beta^2} \frac{d^2y(t(\tau))}{d\tau^2}. \quad (\text{G.2})$$

Ekvation (2.10) blir uttryckt i tiden τ

$$(m+M) \frac{1}{\beta^2} \frac{d^2y(t(\tau))}{d\tau^2} + b \frac{1}{\beta} \frac{dy(t(\tau))}{d\tau} + ky(t(\tau)) = u(t). \quad (\text{G.3})$$

Multiplikation med β^2 ger

$$(m+M) \frac{d^2y(t(\tau))}{d\tau^2} + \beta b \frac{dy(t(\tau))}{d\tau} + \beta^2 ky(t(\tau)) = \beta^2 u(t). \quad (\text{G.4})$$

Om man ändrar k till k_{ny} och b till b_{ny} och om $\beta^2 k_{ny} = k$ och $\beta b_{ny} = b$ så fås att parameterbytet bara ger en skalning i tiden.

H Beträktelse av massas rörelse hängande i en ideal fjäder

Anders Hultgren har i föreläsningen som ges för kursen ET1540, *Grundläggande programmering i Matlab* betraktat fallet med en massa hängande i en gummisnodd. Fallet är intressant eftersom man kan betrakta det som ett typfall samtidigt som det är svårt att se en gummisnodd som en ideal fjäder. Här följer ett försök till analys.

Antag att man har en ideal gummisnodd med fjäderkonstanten k . Häng massan m i snodden. Den massan sjunker sträckan y_1 (stationärt) $mg = ky_1$. Introducera variabeln z given av

$$y = y_1 + z \quad (\text{H.1})$$

Dra nu vikten ner till y_2 där $y_2 > y_1 > 0$.

Systemets totala energi vid platsen y_2 :

Potentiell energi sätter vi till 0.

Energi i snodden: $ky_2^2/2$

Kinetisk energi: 0.

Om vi släpper vikten vid y_2 så kommer vikten att gunga upp och ner kring jämvikt-släget y_1 .

Hela tiden gäller att $W_p + W_k = ky_2^2/2$

dvs $mg(y_2 - y) + ky^2/2 + mv^2/2 = ky_2^2/2$

Motsvarande ode: $mdv/dt = -ky + mg$

dvs $md^2/dt^2 + ky = mg$

Frekvensen ω ges av: $\omega = \sqrt{k/m}$

Om vi antar att lösningen är $y(t) = y_1 + A \cos \omega t$

Här gäller att $y_2 - y_1 = A$.

Då gäller att $v(t) = -A\omega \sin \omega t$

Då fås av energiekvationen att

$$mg(y_2 - (y_1 + A \cos \omega t)) + k(y_1 + A \cos \omega t)^2/2 + m(A\omega \sin \omega t)^2/2 = ky_2^2/2 \quad (\text{H.2})$$

$$mg(y_2 - y_1 - A \cos \omega t) + k(y_1 + A \cos \omega t)^2/2 + \frac{1}{2}mA^2\omega^2 \sin^2 \omega t = ky_2^2/2 \quad (\text{H.3})$$

Om vi tänker oss ett experiment där man låter en massa hänga i en gummisnodd och sen för ner massan en bit så har vi situationen där man känner A och m medan y_1 och k inte är kända. Parametern ω antar vi ges av $\omega = \sqrt{k/m}$.

Experimentet förväntas ge ω och därmed vet vi k . y_1 ges av

$$\begin{cases} mg = ky_1 \\ m\omega^2 = k \end{cases} \Leftrightarrow \begin{cases} mg = ky_1 \\ m\omega^2 = \frac{mg}{y_1} \end{cases} \Rightarrow y_1 = g/\omega^2 \quad (\text{H.4})$$

H.1 Energin magasinerad i fjädern

När massan är i vila och $y = y_1$ så gäller att man kan tänka sig att den ideal fjädern har magasinerat energi

$$ky_1^2/2 = \frac{mg}{y_1}y_1^2/2 = \frac{mgy_1}{2} \quad (\text{H.5})$$

Tänk att massan är upphängt i ett oelastiskt viktlöst snöre och att den ideala fjädern är fästad i snörets fria ände. Om systemet är i vila och man klipper av snöret så kommer energin i fjädern att frigöras och omvandlas till värme.

H.2 Bestämning av fjäderkonstanten k på annat sätt

Låt systemet beskrivet ovan vara i sitt vilostånd. Fäst en vikt m_2 , som exempelvis är en tiondel av m , på vikten m . Låt det nya systemet inta sitt viloläge. Mät sträckan Δ mellan de två vilolägena. Då gäller att $k\Delta = m_2g$. Jämför med värdet $k = m\omega^2$.

H.3 Kontroll av att ansatta lösningen $y(t) = y_1 + A \cos \omega t$ uppfyller energilagen

Vi utvecklar ekvation (H.3) ovan för att se vad den ger för likheter.

$$mg(A - A \cos \omega t) + k(y_1 + A \cos \omega t)^2/2 + \frac{1}{2}mA^2\omega^2 \sin^2 \omega t = k(y_1 + A)^2/2 \quad (\text{H.6})$$

Vi utnyttjar att $mg = ky_1$ samt att $\omega = \sqrt{k/m}$. Då fås

$$y_1A - y_1A \cos \omega t + \frac{1}{2}(y_1 + A \cos \omega t)^2 + \frac{1}{2}A^2 \sin^2 \omega t = y_1^2/2 + y_1A + A^2/2 \quad (\text{H.7})$$

Vi förenklar ett steg till

$$\frac{1}{2}y_1^2 + \frac{1}{2}A^2 \cos^2 \omega t + \frac{1}{2}A^2 \sin^2 \omega t = y_1^2/2 + A^2/2 \quad (\text{H.8})$$

Härmed har vi visat att likhet gäller.