Arm6x Manual

Concurrent Dynamics International

April 2014

Objectives

- Part I:
  - Build a model file (Arm6x.txt) to simulate an six link arm with base fixed to the ground

- Part II:
  - Build a Simulink model (Arm6x.mdl) that runs per Arm6x.txt with a given control system that moves the end-effector to a specified position
  - Examples
  - Exercises

License Restrictions

License type	Buildx.exe	Xsv01.dll
Enterprise	none	none
Project	Must stay with the object count specified by license	Runs with model_files with license specified object count

- Project license permits simulations of mechanisms with a specified object count in {bodies, wheels, forces} and in a unique configuration. No restrictions are placed on the mass property of bodies and wheels, and force placement and parameters or initial conditions.

How to Use This Manual

- This Arm6x manual is written for Enterprise license users where no restrictions are placed on the object counts {body, wheels, forces} in creating models.

- Arm6x.txt is a seed model_file for Enterprise license users to create other models such as: a Stanford arm, arm with end effectors, arms that grabs objects and so forth. I.e. changing an elbow pitch joint in Arm6x from rotary to prismatic joint makes Arm6x a Stanford arm

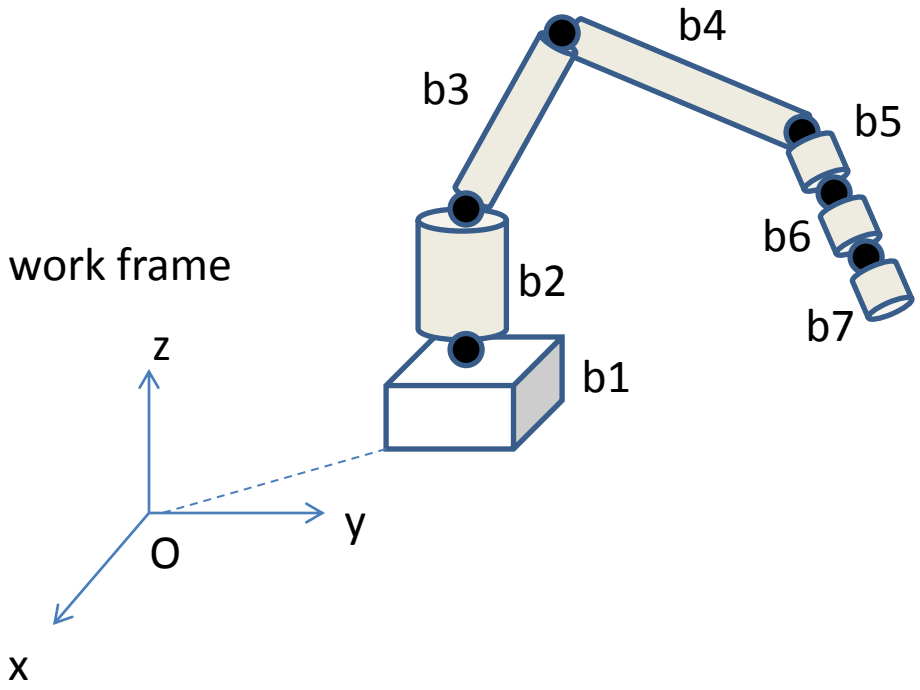
- CDI has many seed models to expedite the development of more complex mechanisms, i.e. a Stewart platform

- This manual is applicable to Project license users whose model object count is {7, 0, 0} and has a chain configuration as Arm6x

Part I Topics

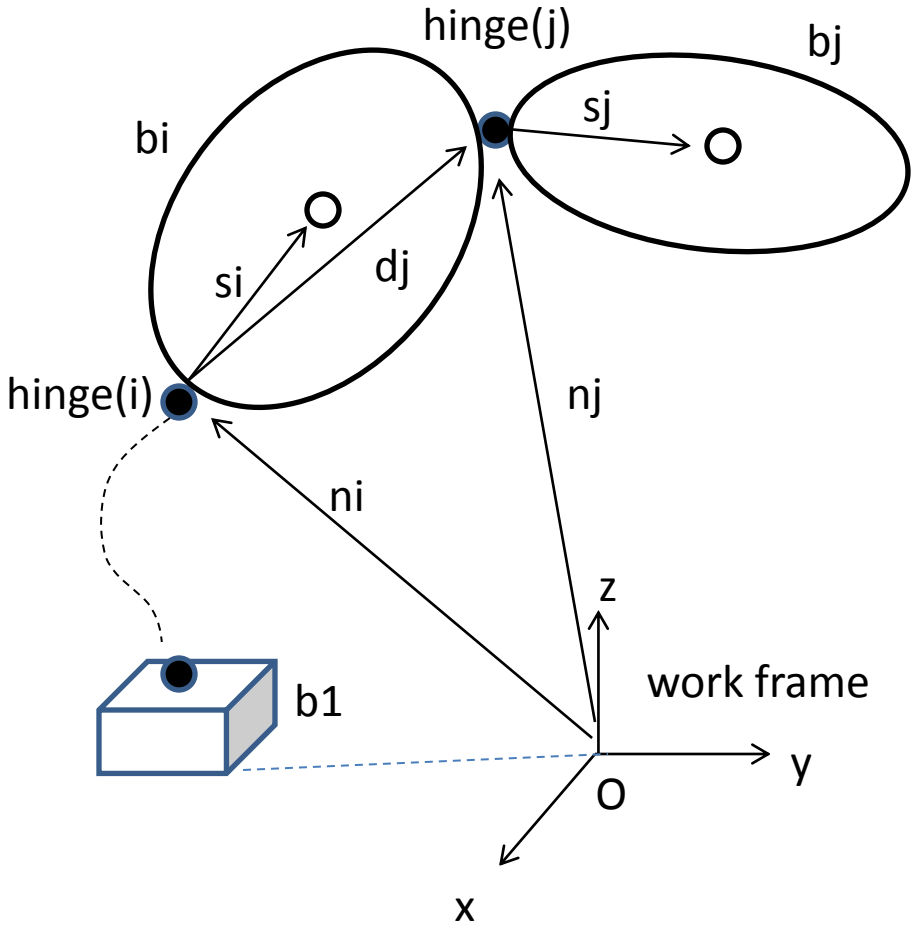
- Physical Model
- Buildx Tasks
- Key Files
- Main Menu
- Model Menus
- Body Menu
- B1 & B2 page …
- Body Actuation Signals
- Gravity Menu
- Dynamics Input
- Dynamics Output
- Plot Menu
- Simplot
- Save Model
- Exit Buildx
- Q & A

Arm6x Model



b1= ground
b2= shoulder (z)
b3= elbow1 (x)
b4= elbow2 (x)
b5= wrist pitch (x)
b6= wrist yaw (y)
b7= wrist roll (z)

Nomenclature



precedence:
b1= root body
bi= parent of bj
bj= child of bi

ni= hinge(i) position in wf
nj= hinge(j) position in wf
si= cm(i) position in bi_f
sj= cm(j) position in bj_f
dj= hinge(j) position in bi_f

hinge(i) is at origin of bi frame

On the Simulation

- We are building a model file in Part I to support a six-link robot arm simulation. All joints are rotary for now. Setup includes the definition of mass property and initial condition of the arm. It also includes the specification of dynamics input and output signals required by the arm control system.

- We will show in Part II two examples on how to develop and execute the mdl file to simulate the robot arm in moving from one configuration to another using a PD controller provided here.

- To expedite the control system development, we use the dynamics engine provided bj.gvec and bj.mc for the controller. The latter signals are normally computed by the controller given joint angles and they are then used to compute the bias torque component of the control signal.

- bj.gvec= bj's free axis in b1 frame, bj.mc=bj.branch.cm*bj.branch.mass

- Users are encouraged to close the Arm6x control loop with their own control system and compare performance against the one given here.

Buildx Tasks

- Build Arm6x.txt to define simulation parmeters:
  - mass property,model connectivity, degree of freedom
  - sim3xr.dll input/output/plot data
  - gravity, …etc

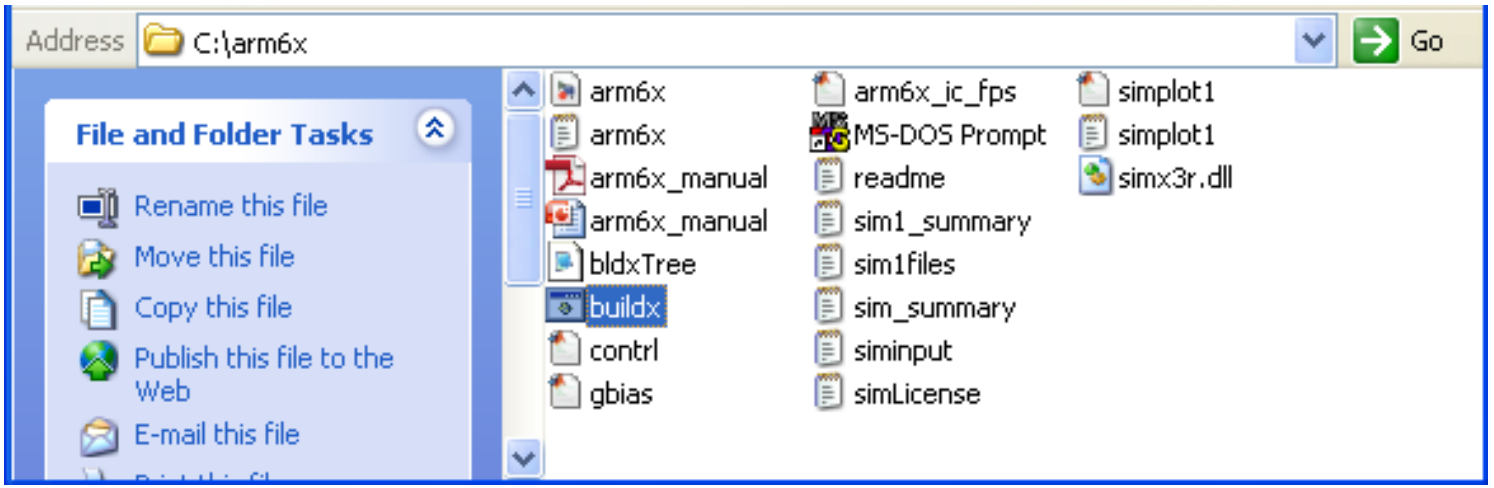
- Build simplot1.m to view sim results

Key Files in c:\Arm6x

- Buildx.bat='..\buildx'
- Input file pointer: siminput.txt
- Working files: sim1files.txt
- Model_file: Arm6x.txt
- Simplot file: simplot1.txt
- License file: simlicense.txt

Start Buildx.exe

- Click c:\Arm6x\buildx.bat to start buildx.exe and see the Main Menu



Main Menu

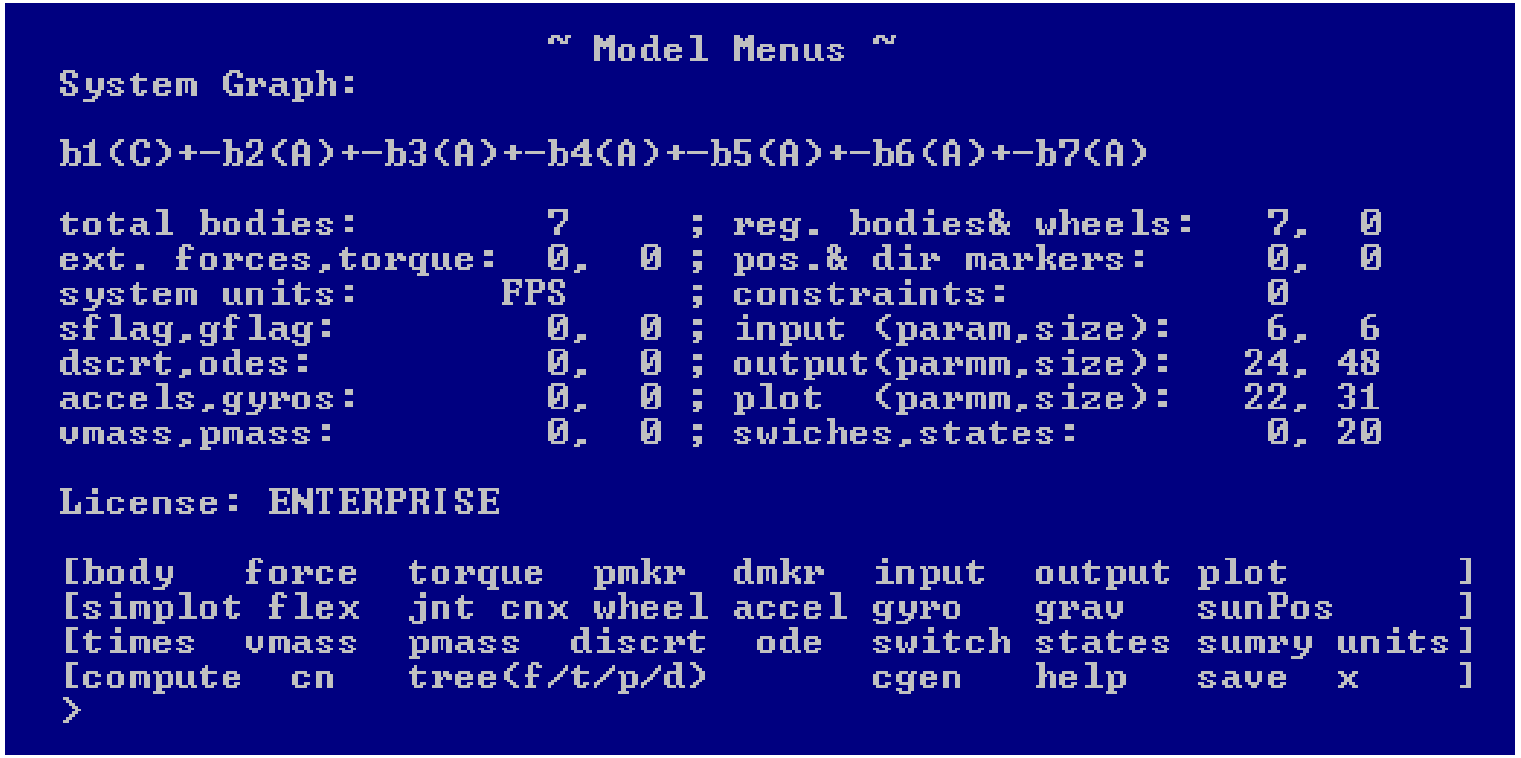
- See working files in sim1files.txt
- Type 'xmr' to go to Model Menus

Main Menu Commands

- Type 'xmr’ to go to Model Menus
- Type 'open’ to read a siminput file or a model file
- Type 'save’ to update siminput file or save current data to a model file
- Type 'model’ to change the model file name
- Type 'sumry’ to change summary file name
- Type 'mssg’ to change message file name
- Type 'plotdt’ to change the plot data sample period
- Type 'x’ to exit

Arm6x Model Menus

- See model parts size and choose menus to edit/browse



Info on Model Menus Page

- Block 1, top left: model tree diagram
- Block 2, model status:
  - {total bodies, … states}
- Block 3, menu commands:
  - {body, … x}
  - Type 'help' to see meaning of commands

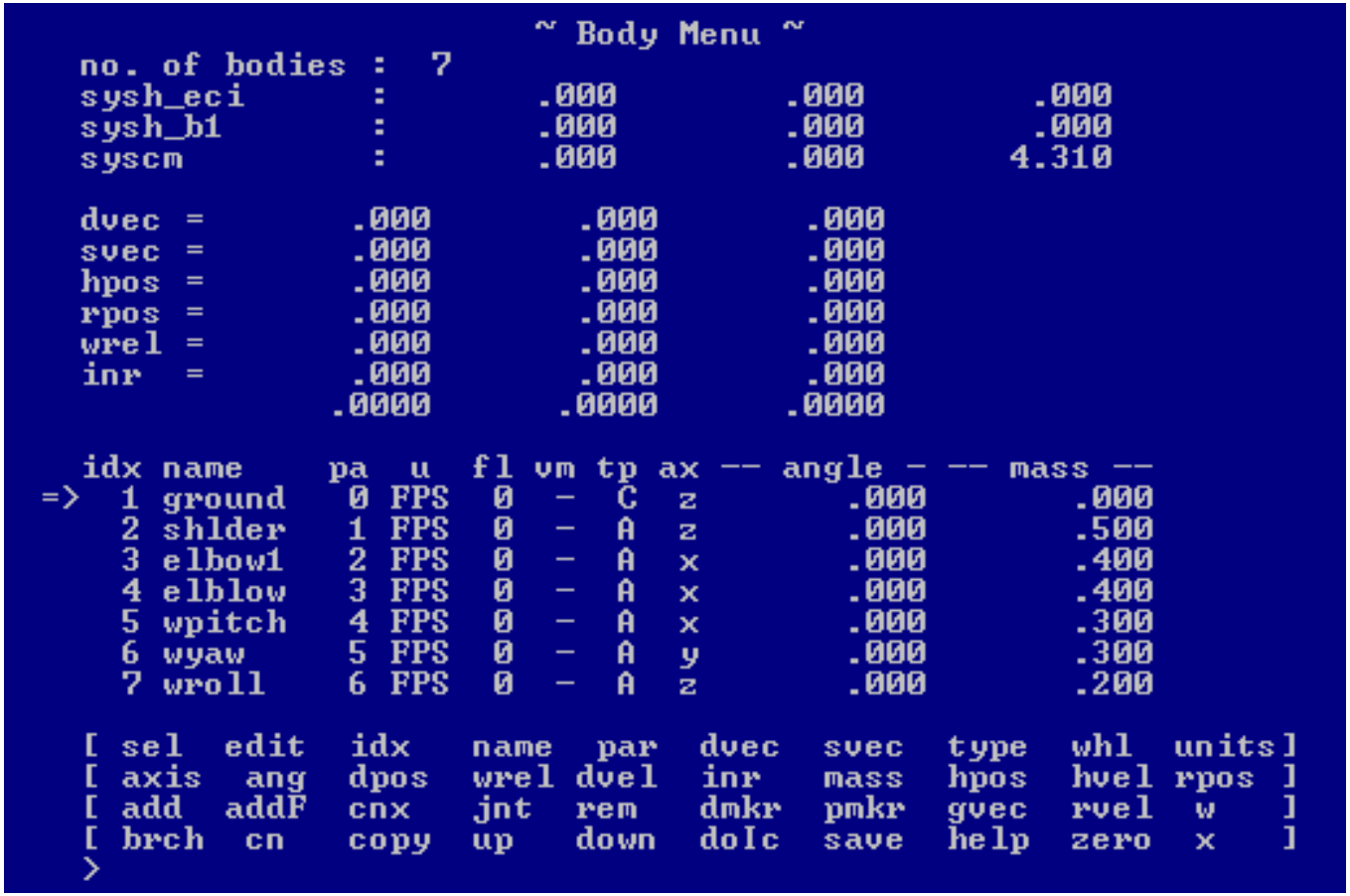
Model Menus Commands

- Type 'body’ to go to xsv.Body Menu
- Type 'force’ to go to xsv.force menu
- Type 'grav’ to go to xsv.Gravity Menu
- Type 'pmkr’ to go to xsv.position marker menu
- Type 'dmkr’ to go to xsv.direction marker menu
- Type 'input’ to go to xsv.input menu
- Type 'output’ to go to xsv.output menu
- Type 'plot’ to go to xsv.Plot Menu
- Type 'simplot’ to go to xsv.simPlot Menu
- Type 'time’ to go to xsv.timing menu

- Type 'help’ to get definition on data and commands
- Type 'x’ to exit menu

Body Menu

- See body summary, type ‘body’ from Model Menus page



Body Menu Commands

- Type 'add<j>' to add bodies to bj : i.e. 'add5' to add bodies to b(5)
- Type 'name<j>' to edit bj.name
- Type 'par<j>' to edit bj.parent
- Type 'type<j>' to edit motion type, bj.type: {a…h}
- Type 'axis<j>' to edit bj.axis: {x, y or z}
- Type 'ang<j>' to edit initial inboard bj.ang for 1 dof rotational joints
- Type 'wrel<j>' to edit bj.angular_rate
- Type 'mass<j>' to edit bj.mass
- Type 'svec<j>' to edit bj.svec
- Type 'dvec<j>' to edit bj.dvec
- Type 'inr<j>' to edit bj.inr
- Type 'edit<j>' to see all bj data
- Type 'help' to get definition on data and commands
- Type 'x' to exit menu

All B1 Data

- See all b1 data, type ‘edit1’ from Body Menu

```
                    ~ Body Menu ~
idx   name      par gflg  u  tp ax         ang          mass
  1 ground        0    0 FPS  d  x          .000   .00000E+00

> inertia(inr)= .00000E+00   .00000E+00   .00000E+00
                .00000E+00   .00000E+00   .00000E+00
> dVec         =        .000         .000         .000
  hngVec       =        .000         .000         .000
> sVec         =        .000         .000         .000
  rVec         =        .000         .000         .000
> dcm0         =   1.000000      .000000      .000000
                    .000000     1.000000      .000000
                    .000000      .000000     1.000000
  Euler seq    =123
  Euler angs   =        .000         .000         .000

  b2i matrix   =   1.000000      .000000      .000000
                    .000000     1.000000      .000000
                    .000000      .000000     1.000000

> wRel         =       .0000        .0000        .0000
> dPos         =        .000         .000         .000
> dVel         =        .000         .000         .000
  force on b   =        .000         .000         .000
  torque on b  =        .000         .000         .000

[idx  axis ang   b2i  dcm0 dpos  dvec dvel  gflag mass inr]
[name par  units o2i  dcm  type  svec wrel  help  save  x ]
>
```

B1(ground) Data for Arm6x

- b1 is grounded => b1.mass=0, b1.inr=0
- b1.attitude=b1.dcm0= identity matrix
- b1.rates= zeros

- Type 'idx2’ to see all b2 data

- Type 'x’ to exit this page

Info on b<j> Page

- Block 1: List of all attributes of body(j)

- Block 2: menu commands to change attributes of body(1) or to go to another Body Menu:
  - {Index, … x}
  - Type 'help': see data and command definitions
  - Type 'Idx<j>': goes to another body(j) page
  - Type 'x': exit this page

B2(shlder) Page

- See all b2 data, type ‘idx2’ from b1 page

```
                        ~ Body Menu ~
idx   name      par gflg   u   tp ax          ang          mass
  2 shlder        1    0 FPS   a  z           .000   .50000E+00

> inertia(inr)= .40000E+00   .40000E+00   .20000E+00
                .00000E+00   .00000E+00   .00000E+00
> dVec         =        .000         .000         .000
  hngVec       =        .000         .000         .000
> sVec         =        .000         .000        1.000
  rVec         =        .000         .000        1.000
> dcm0         =    1.000000     .000000     .000000
                    .000000    1.000000     .000000
                    .000000     .000000    1.000000
  Euler seq    =123
  Euler angs   =        .000         .000         .000

  b2i matrix   =    1.000000     .000000     .000000
                    .000000    1.000000     .000000
                    .000000     .000000    1.000000

> wRel         =       .0000        .0000        .0000
> dPos         =        .000         .000         .000
> dVel         =        .000         .000         .000
  force on b   =        .000         .000         .000
  torque on b  =        .000         .000         .000

[idx  axis ang   b2i  dcm0 dpos  dvec dvel  gflag mass inr]
[name par  units o2i  dcm  type  svec wrel  help  save  x ]
>
```

Inertia Menu

- Need define moi of each body bj.inertia about the bj.cm for all j
- Type 'inr’ from Body Menu page to see a summary of moi data

idx	name	ixx	iyy	izz
		ixy	ixz	iyz
=> 1	ground	.000	.000	.000
		.000	.000	.000
2	shlder	.400	.400	.200
		.000	.000	.000
3	elbow1	.200	.200	.100
		.000	.000	.000
4	elblow	.200	.200	.100
		.000	.000	.000
5	wpitch	.150	.150	.100
		.000	.000	.000
6	wroll	.200	.200	.100
		.000	.000	.000
7	wyaw	.150	.150	.100
		.000	.000	.000

- Type 'inr<j>’ to edit bj.inr
- Type 'x’ to exit menu

Dvec Menu

- Need define dvec(j), bj.hinge.position, in bj.parent frame, for all j
- Type dvec from Body Menu and see the dvec summary

```
   idx name        u ---------- ---dvec--- ----------
=>   1 ground    FPS        .000        .000        .000
     2 shlder    FPS        .000        .000        .000
     3 elbow1    FPS        .000        .000       2.000
     4 elblow2   FPS        .000        .000       2.000
     5 wpitch    FPS        .000        .000       2.000
     6 wroll     FPS        .000        .000        .500
     7 wyaw      FPS        .000        .000        .500
```

- Type 'dvec<j>’ to edit bj.dvec
- Type 'x’ to exit menu

Svec Menu

- Need define svec(j), bj.position.cm, in bj.local frame, for all j
- Type 'svec’ from Body Menu and see the svec summary

```
   idx name        u ---------- ---svec--- ----------
=>   1 ground    FPS        .000        .000        .000
     2 shlder    FPS        .000        .000       1.000
     3 elbow1    FPS        .000        .000       1.000
     4 elblow2   FPS        .000        .000       1.000
     5 wpitch    FPS        .000        .000        .250
     6 wroll     FPS        .000        .000        .250
     7 wyaw      FPS        .000        .000        .250
```

- Type 'svec<j>’ to edit bj.svec
- Type 'x’ to exit menu

Pos Summary

- See bj.cm is in b1 frame given bj.ang set for all j
- Type 'rpos' in Body Menu to see a summary of bj.cm

```
   idx name    ---------- ---rpos--- ----------
=>   1 ground        .000        .000        .000
     2 shlder       .000        .000       1.000
     3 elbow1       .000        .000       3.000
     4 elblow       .000        .000       5.000
     5 wpitch       .000        .000       6.250
     6 wroll        .000        .000       6.750
     7 wyaw         .000        .000       7.250
```

- Note: all cm's are initially on b1.z_axis for this example, given all bj.ang=0 for j

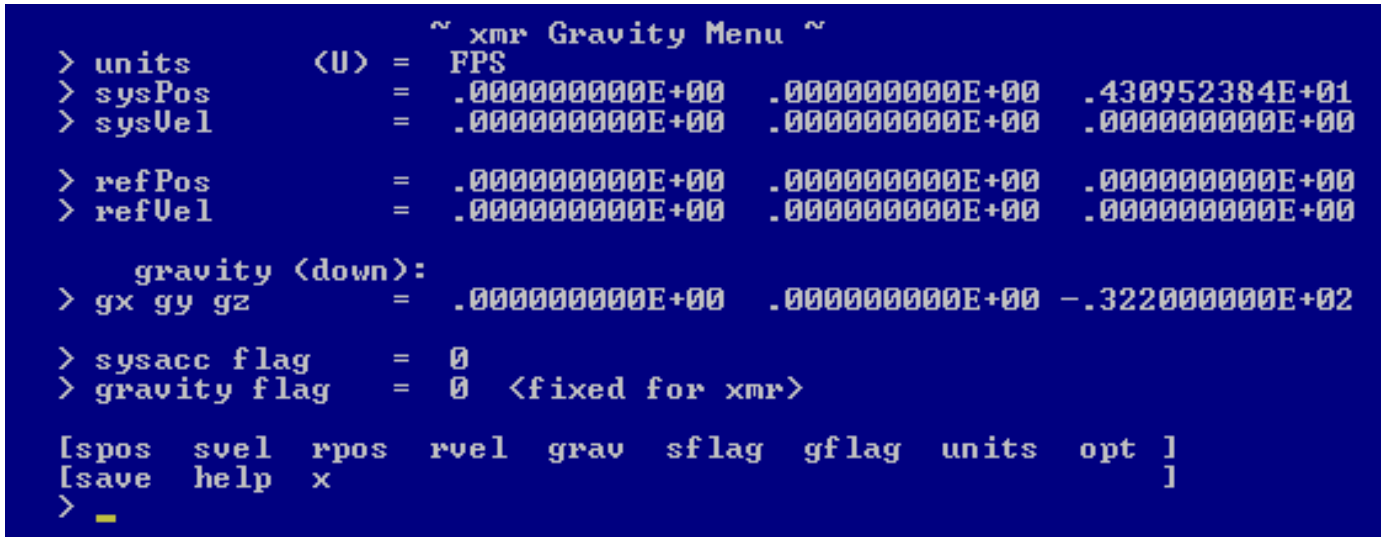
Body Actuation Signals

- Bj Inboard force or torque actuates that body and impacts the motion of the rest of the system. Accelerations can be specified for joints with prescribed motion
- The Dynamics Input signals for bj are processed based on bj.type as follows.

type	Size	Input	processing
A	1	Htqax,j	Bj.torque(axis)=Htqaxj
B	3	Htq,j	Bj.torque=Htqj
C	1	Wraccax,j	Bj.wracc(axis)=Wraccaxj
D	3	Wracc,j	Bj.wracc=wraccj
E	1	Frcax,j	Bj.force(axis)=frcaxj
F	3	Frc,j	Bj.force=frcj
G	1	Hraccax,j	Bj.hraccax=hraccaxj
H	3	Hracc,j	Bj.hracc=hraccj

Gravity Menu

- Need to define gravitational acceleration [gx,gy,gz] for the model
- Type 'grav' from Model Menus (page 11) to open Gravity Menu



- Type 'grav' here to edit [gx,gy,gz]
- Sflag=0 means (rpos,rvel) are prescribed by input signals
- =1 means (rpos,rvel) are force determined

Gravity Menu Commands

- Type 'spos’ to edit orbit position in workspace coordinates
- Type 'svel’ to edit total velocity in workspace coordinates
- Type 'rpos’ to edit b1.reference_position in workspace coordinates
- Type 'rvel’ to edit b1.reference_velocity in workspace coordinates
- Type 'grav’ to edit gravitational acceleration
- Type 'units’ to change the units of coordinates and mass properties
- Type 'sflag’ to run simulation in prescribed(spos,svel) mode or in force determined(spos,svel) mode
- Gflag=0 for xmr applications
- Type 'help’ to get definitions of data and commands
- Type 'x’ to exit menu
- Buildx automatically updates all menu parameters when one of them is altered, i.e. changing ‘rpos’ results in a new (spos)…etc.

Dynamics Input

- Need input data to simx3r.dll to actuate the vehicle dynamics during run time
- Type 'input' from Model Menus to open the input menu
- Use 'newlist' to get a suggested list for the current model
- Use 'add' and 'rem' command to modify current input list (udata)
- Arm6x input list is as follows:

```
Udata list:

 1) htqax,2          ! 2) htqax,3          ! 3) htqax,4
 4) htqax,5          ! 5) htqax,6          ! 6) htqax,7
```

- htqax, 2:7= joint torque

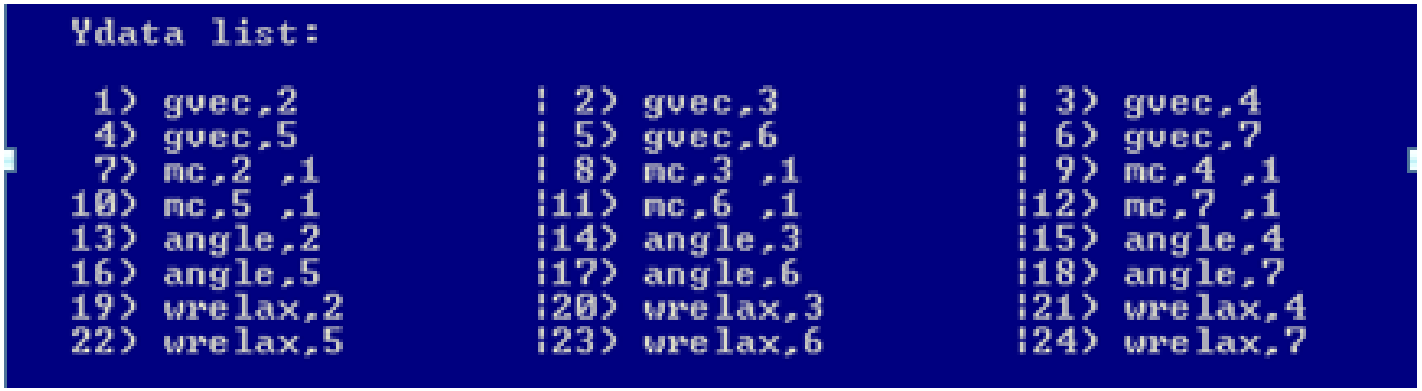
Input Menu Commands

- Type 'add’ to add new variables to the end of udata list
- Type 'add<j>’ to insert new variables at udata(j)
- Type 'rem’ to remove a group of variables
- Type 'rem<j>’ to remove udata(j)
- Type 'chg<j>’ to change udata(j)
- Type 'len’ to see ordinal position of udata and their length
- Type 'x’ to exit udata menu

- A variable selection menu appears on commands {add, chg}
- Type 'sel<j>’ to select var(j) to add or chg
- Type 'x’ to return to udata menu

Dynamics Output

- Need output from simx3r.dll to drive the control system during run time
- Type 'output' from Model Menus to open the output menu
- Use 'newlist' to get a suggested list for the current model
- Use 'add' and 'rem' command to modify current output list (ydata)
- Arm6x output list is as follows:



- gvec, 2:7 = bj.free_axis for j= 2:7
- mc, 2:7 = msum(j)*bj.subtree_cm.pos for j= 2:7
- angle, 2:7 = bj.joint.ang for j= 2:7
- wrelax, 2:7 = bj.joint.rate for j= 2:7

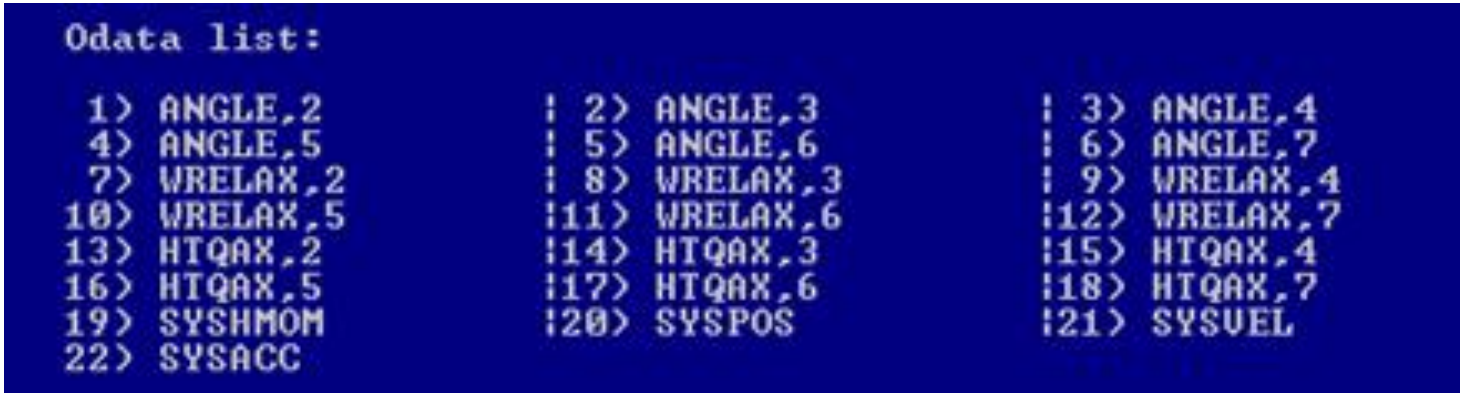
Output Menu Commands

- Type 'add<j>’ to insert new variables at ydata(j)
- Type 'rem’ to remove a group of variables
- Type 'rem<j>’ to remove ydata(j)
- Type 'chg<j>’ to change ydata(j)
- Type 'len’ to see ordinal position of udata and their length
- Type 'x’ to exit ydata menu

- A variable selection menu appears on commands {add, chg}
- Type 'sel<j>’ to select var(j) to add or chg
- Type 'x’ to return to ydata menu

Plot Data

- Need to save selected data from dynamics engine to plotfile during run time
- Type 'plot' from Model Menus to open the Plot Menu
- Use 'newlist' to get a suggested list for the current model
- Use 'add' and 'rem' command to modify current plot data list (odata)
- Arm6x plot list is as follows:



- angle, 2:7= bj.angle
- wrelax, 2:7= bj.ang_rate for j=2:7
- htqax, 2:7= bj.hinge_torque for j=2:7, … etc.

Plot Data Commands

- Type 'add’ to add new variables to the end of odata list
- Type 'add<j>’ to insert new variables at odata(j)
- Type 'rem’ to remove a group of variables
- Type 'rem<j>’ to remove odata(j)
- Type 'chg<j>’ to change odata(j)
- Type 'len’ to see ordinal position of udata and their length
- Type 'x’ to exit odata menu

- A variable selection menu appears on commands {add, chg}
- Type 'sel<j>’ to select var(j) to add or chg
- Type 'x’ to return to odata menu

Simplot

- Need simplot1.m to view sim results, then type ‘simplot’ from Model Menus (page 11) or Plot Menu to build it (page 35)
- all plot data are selected in Plot Menu
- A. steps from simPlot Menu:
  1. Type 'add’ to add figures and respond with ‘4’ to create 4 figures
  2. Type 'title1’ to set figure (1) title: i.e. reply with ‘system’
  3. For ‘title2’ ,‘title3’ and ‘title4’ commands, respond with ‘angles’,’rate’ and ‘torque’
  4. Type 'vars1’ to define variables to be plotted in fig 1. this opens the plot variables page
- B. steps from vars menu:
  1. Type 'addv19’ and reply with 4to add 4 variables starting with the variable(19), syshmom
  2. Type 'addp’ and respond with ‘1,4’ to create six subplots for figure(1)
  3. Type 'format’ and respond with ‘2,2’ to plot 4 subplots in 2 rows and 2 columns format
  4. Type 'x’ to go back to simPlot Menu
- Repeat steps A.3 and all B steps with proper indexing to define subplots of other figures for sat4w6j2a (i.e. ‘angles’, ‘rates’ and ‘torque’)
- Final steps from simPlot Menu:
  1. Type 'save’ and reply with ‘simplot1.txt’ to save simplot data to simplot1.txt
  2. Type 'make’ to create simplot1.m, see completion message
  3. Type 'x’ to exit simPlot Menu. Simplot1.m is ready.

Save Model Data

- Need to save model after model parts data have been altered
- Go to Model Menus (page 11) or any menu that has the ‘save’ command and type ‘save’
- Choose one of 3 options:
  1. Save to current model file
  2. Save to another model file
  3. Cancel

Exit Buildx

- 3 ways to exit buildx:
  - Go to Model Menus and type ‘q’ <return>
  - Go to Main Menu and type ‘x’
  - Click the ‘x’ on top right corner of the buildx window

- Reminder: Save model data if changes have been done to the current model. See page 37.

Q & A

- Can one add and delete bodies, wheels and forces?
  - yes if you have enterprise license, and no if you have a project license
- Are all Project licenses restricted to object count {7, 0, 0}?
  - no, for example oneCylinderEngine Project license has an object count of {4, 0, 1} and a unique parent-child relation between bodies, wheels and forces
- Must all joints in Arm6x be rotational?
  - no. any joint can be either rotational or translational
  - Arm6x Project license requires that (b2:b7).joint be 1 dof
  - use 'type<j>' to define the relative motion of bj
- How to model an arm with less than 6 joints under Arm6x Project license?
  - let's say your arm model is chain(b1:b5) and b6:b7 are not needed
  - set b(6:7).type to c, set b(6:7).mass to 0, set b(6:7).inr to zero
- Can free axes in Arm6x be different from those given?
  - yes, use 'axis<j>' command from Body Menu to change bj.axis

- How can one see all the available input, output and plot variables when choosing them from udata, ydata and odata menus?
  - all available variable list is shown when one types ‘add’ command from the menu
  - Type 'defj’ to get the definition of variable(j) in that list
  - Type 'selj’ from the add menu to select variable(j) to the list

- How does one change the plot data sample period?
  - Type 'plotdt’ from the Main Menu or from the times menu to do that

- Why are there ‘dt’ and other time specification in the times menu?
  - those time specifications are not used for the Simulink applications, they are for the Fortran and C implementation of xsv01 engine

Part II Topics

- Sim3xr.dll functionality
- Key files
- Arm6x.mdl
- Control system
- Running Arm6x.mdl
- Viewing sim results
- Example1
- Example2
- Adjustable sim parameters
- Exercises
- Simulation Notes
- Summary

Sim3xr.dll Functionality

- Sim3xr.dll solves the equations of motion of a mechanism required by model_file
- It reads actuation signals (udata) from the control system in Simulink workspace
- It sends motion signals (ydata) to Simulink workspace for control system input
- It outputs selected data (odata) to plotfile for post-sim viewing

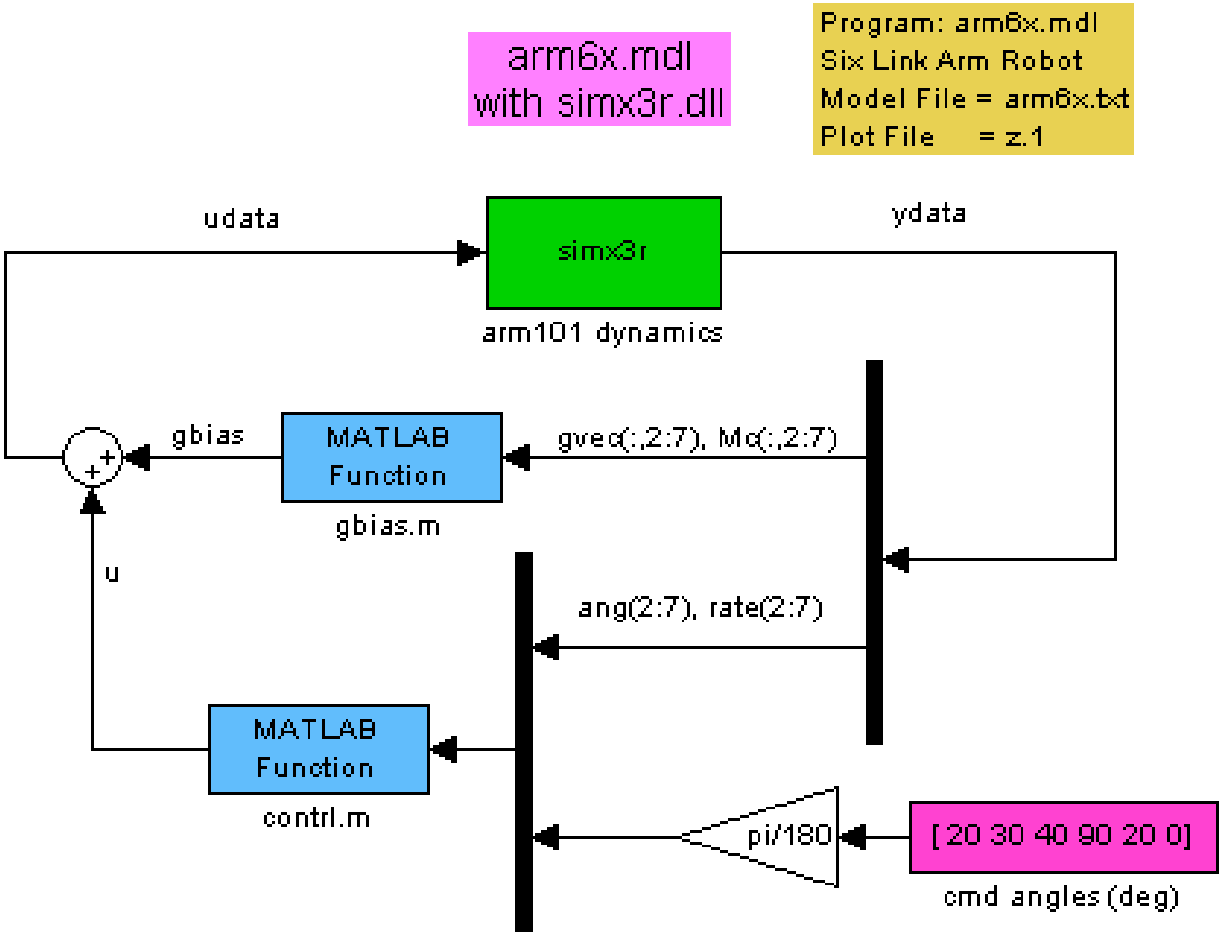
Key Files

- Input file pointer: siminput.txt
- Working files definition: sim1files.txt
- Model_file: Arm6x.txt
- License file: simLicense.txt
- Simulink program: Arm6x.mdl
- Simulation engine: sim3xr.dll
- Control scripts: contrl.m, gbias.m

Arm6x.mdl

- Need Arm6x.mdl to run an sim3xr.dll based dynamics simulation according to Arm6x.txt
- Sends ydata to contrl.m and is actuated by udata from the latter
- Control system is user supplied and application specific

- Click c:\Arm6x\Arm6x.mdl to open it in Simulink workspace



Control System

- Need gbias.m and contrl.m to map ydata to udata to cause Arm6x to move the joint angles to specified angles in Arm6x.mdl

- Control system i/o:
  - Input: gvec2:7,mc2:7,angle2:7,wrelax2:7 (ydata)
  - Output: htqax2:7 (udata)

- Summing outputs from gbias.m and contrl.m yields hinge torque,htqax2:7

Arm6x Contrl.m Code

```
function u=contrl(y)
 % Objective: move joint angles to commanded angles

angle = y( 1: 6) ; % joint angles
rate  = y( 7:12) ; % joint rates
cmd   = y(13:18) ; % command angles
kp    = 3.  ;
kv    = 3.5 ;

for i=1:6,
      u(i) = kp*(cmd(i)-angle(i))-kv*rate(i) ; % PD controller
end
```

Arm6x Gbias.m Code

```
function u=gbias(y)
% compute gravity bias for each joint
% extract input signals from u array
grav  = [0  0  -32.2]; %gravitational acceleration (fps)
gvec(:,1)= y(1:3); % joint free axes
gvec(:,2)= y(4:6);
gvec(:,3)= y(7:9);
gvec(:,4)= y(10:12);
gvec(:,5)= y(13:15);
gvec(:,6)= y(16:18);
Mc(:,1)  = y(19:21) ;% joint mass sum times joint cm from hinge
Mc(:,2)  = y(22:24);
Mc(:,3)  = y(25:27);
Mc(:,4)  = y(28:30);
Mc(:,5)  = y(31:33);
Mc(:,6)  = y(34:36);

for i=1:6,
   u(i)= -gvec(:,i)'*(cross( Mc(:,i),grav))';  % grav bias torque for rot joints
End
```

Example 1

- Setup given so far is the preparation of example1

- Example 1 objective:
  - Move arm angles from [0, 0, 0, 0, 0, 0] to command angles of [20, 30, 40, 90, 20, 0] degs

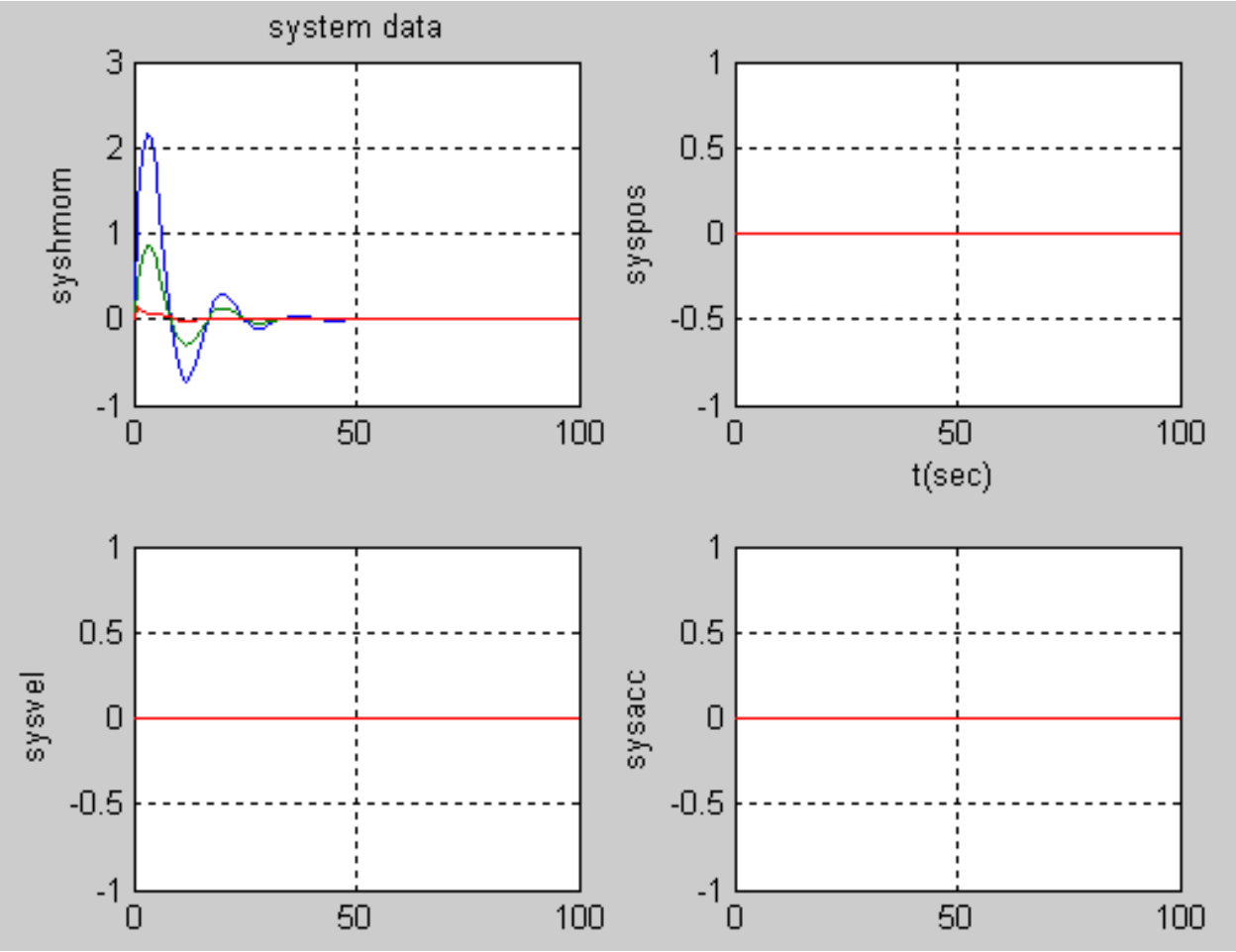
Run Arm6x.mdl

- Click c:\Arm6x\Arm6x.mdl to open the Simulink program
- Open the simulation dialog box to set configuration parameters as needed,i.e.:
  - solver options > step size, method
  - end time
- Exit dialog box and click the run button (black triangle) on control bar to start simulation

View Sim Results

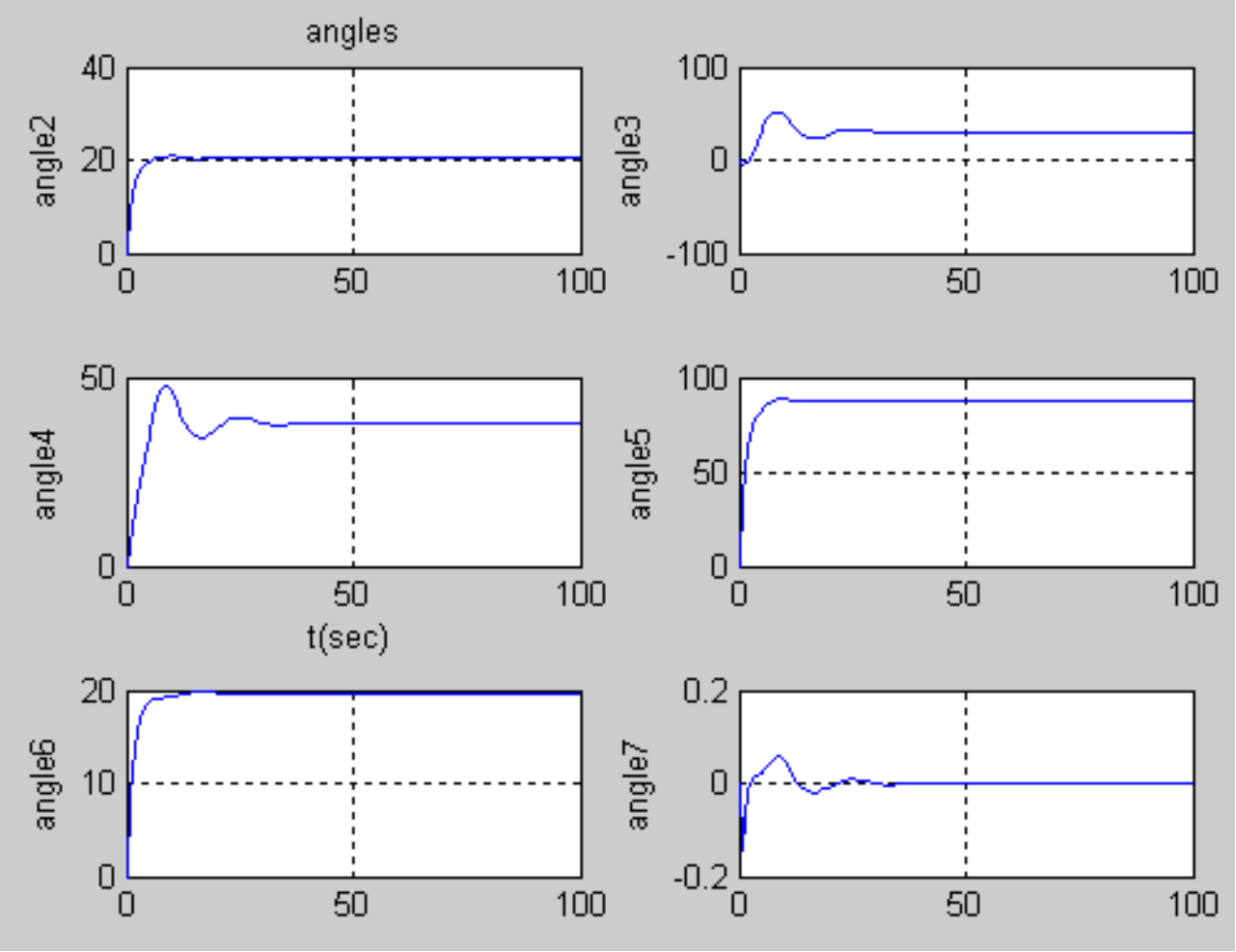
- Plot_file (i.e. z.1) has the time response of selected odata. See p: 29
- Type 'load z.1’ from Matlab window to read in sim result
- Type 'simplot1(z)’ to view result
- Simplot1.m is a script that can be constructed easily using buildx.exe (see simPlot Menu)

Fig.1 System Motion-1



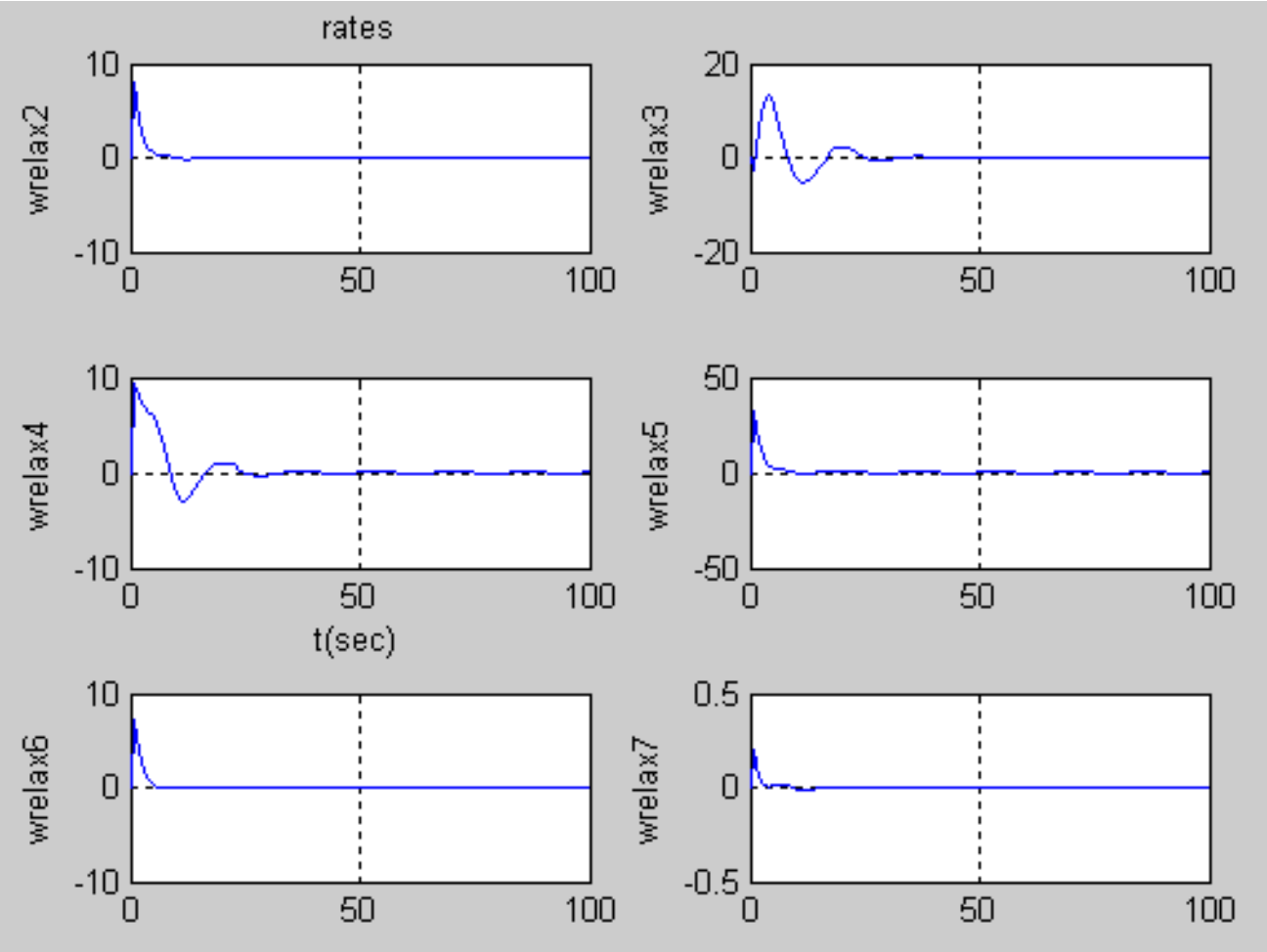
- ground base motion:
  syspos=0
  sysvel=0
  sysacc=0

Fig.2 Joint Angles-1



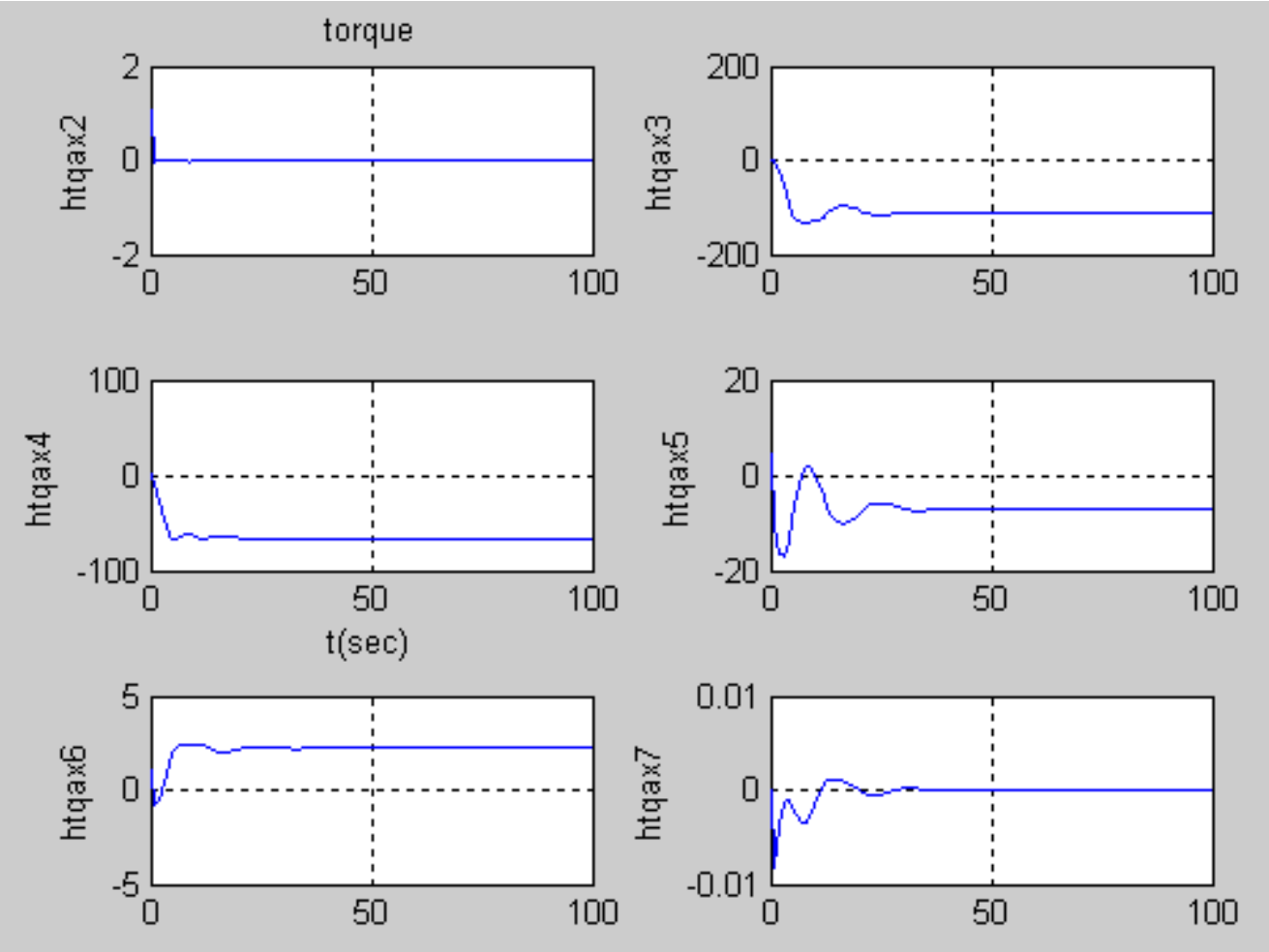
- initial angles:
  [0, 0, 0, 0, 0, 0] deg
- ss joint angles:
  [20, 30, 40, 90, 20, 0] deg per command

Fig.3 Joint Rates-1



- ss joint rates:
  [0, 0, 0, 0, 0, 0]
  deg/s

Fig.4 Joint Torque-1



- ss joint torque: non zero torque for joints [3, 4, 5, 6]

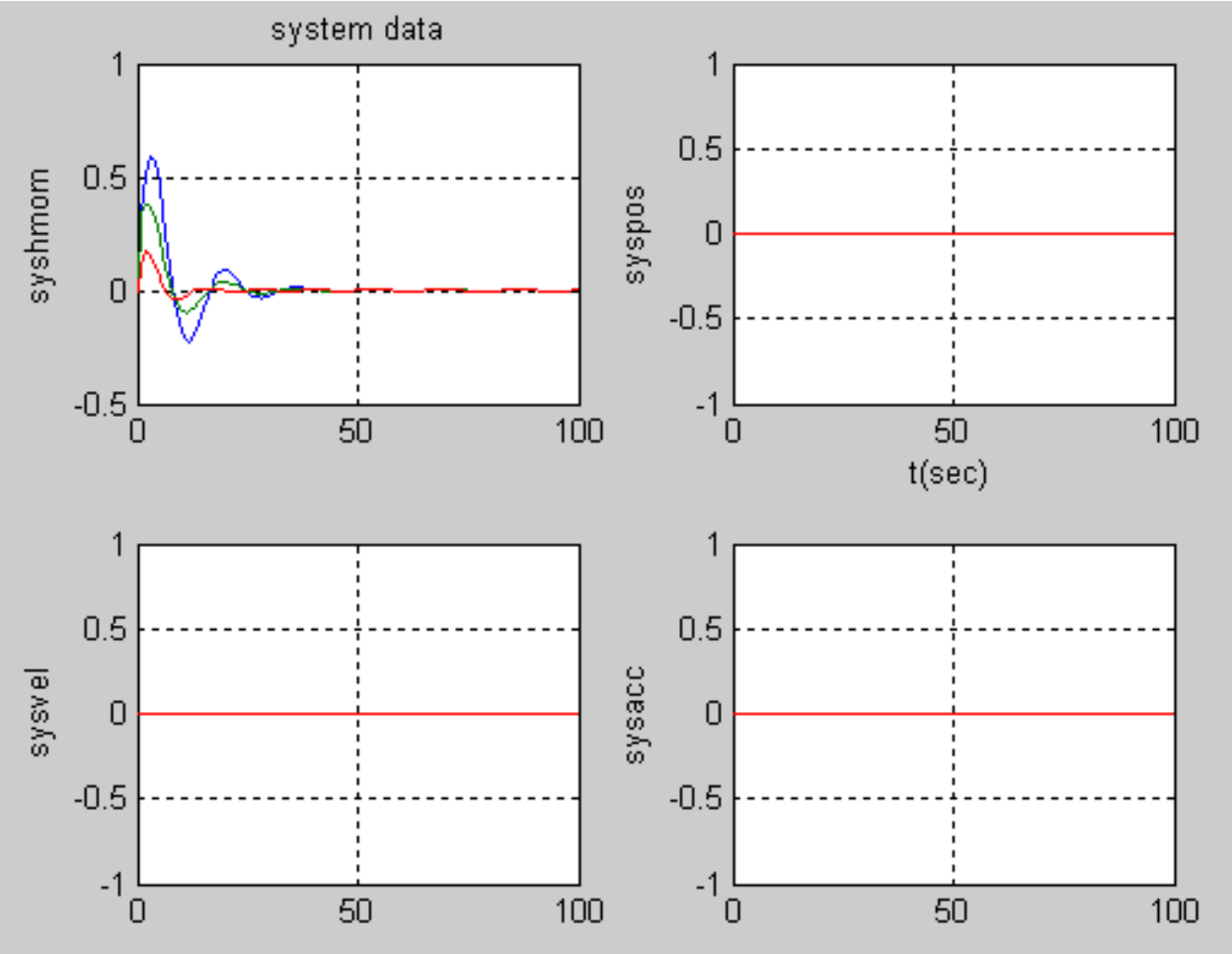
Example 2

- Example 2 objective:
  - move arm angles(2:7) from [10, 20, 30, 40, 50, 60] to command angles of [20, 30, 40, 90, 20, 0] degs

- Procedure:
  - go to Body Menu and use the commands ‘ang2’ to change joint angle of b2 to 10 deg
  - Type 'ang3’ to change joint angle of b(3) to 20 deg
  - and so forth
  - Type 'save’ to save the model file
  - Type 'q’ to quit buildx

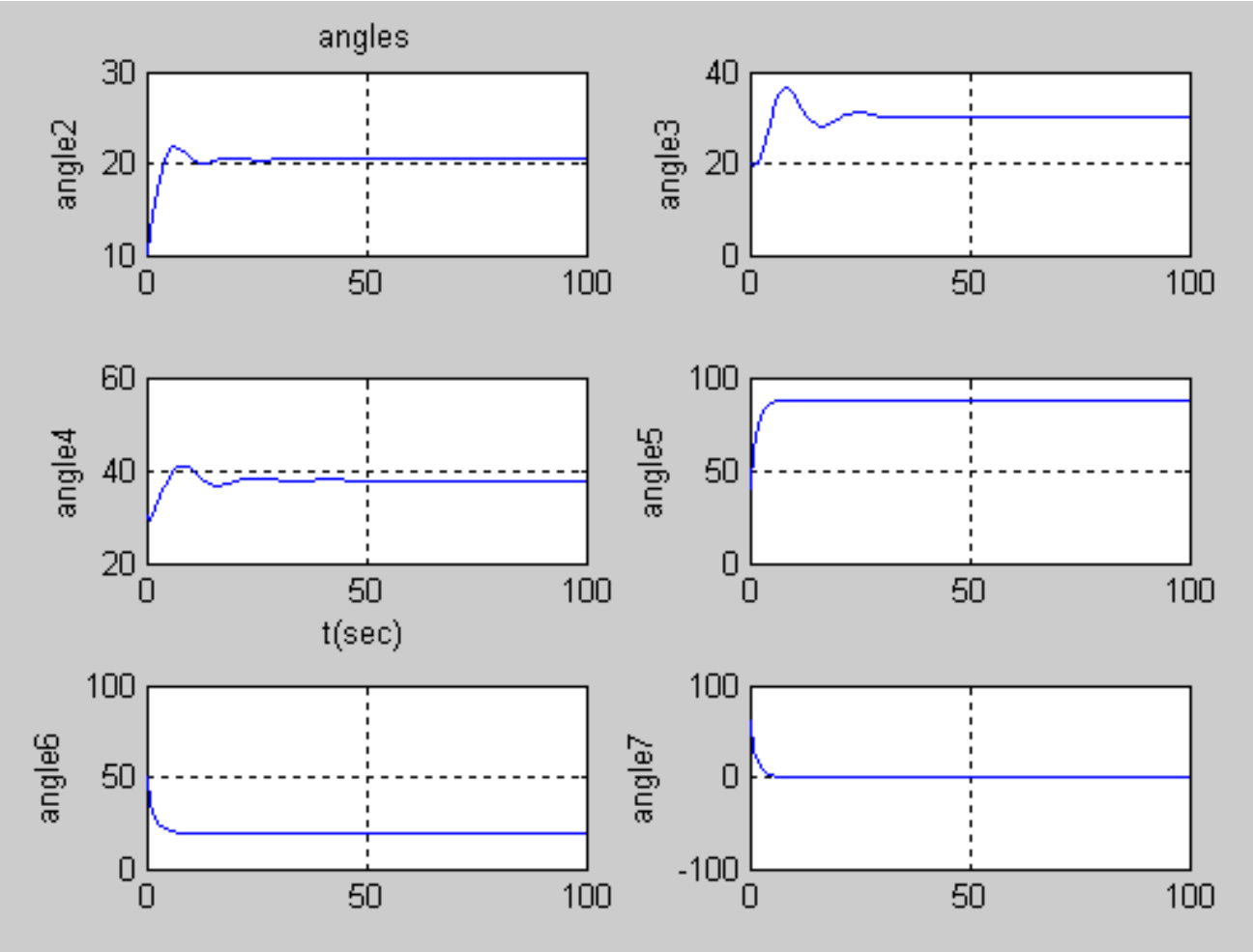
- Run Arm6x.mdl (see page 50)
- View results (see page 51)

Fig.5 System Motion-2



- ground base motion:
  syspos=0
  sysvel=0
  sysacc=0

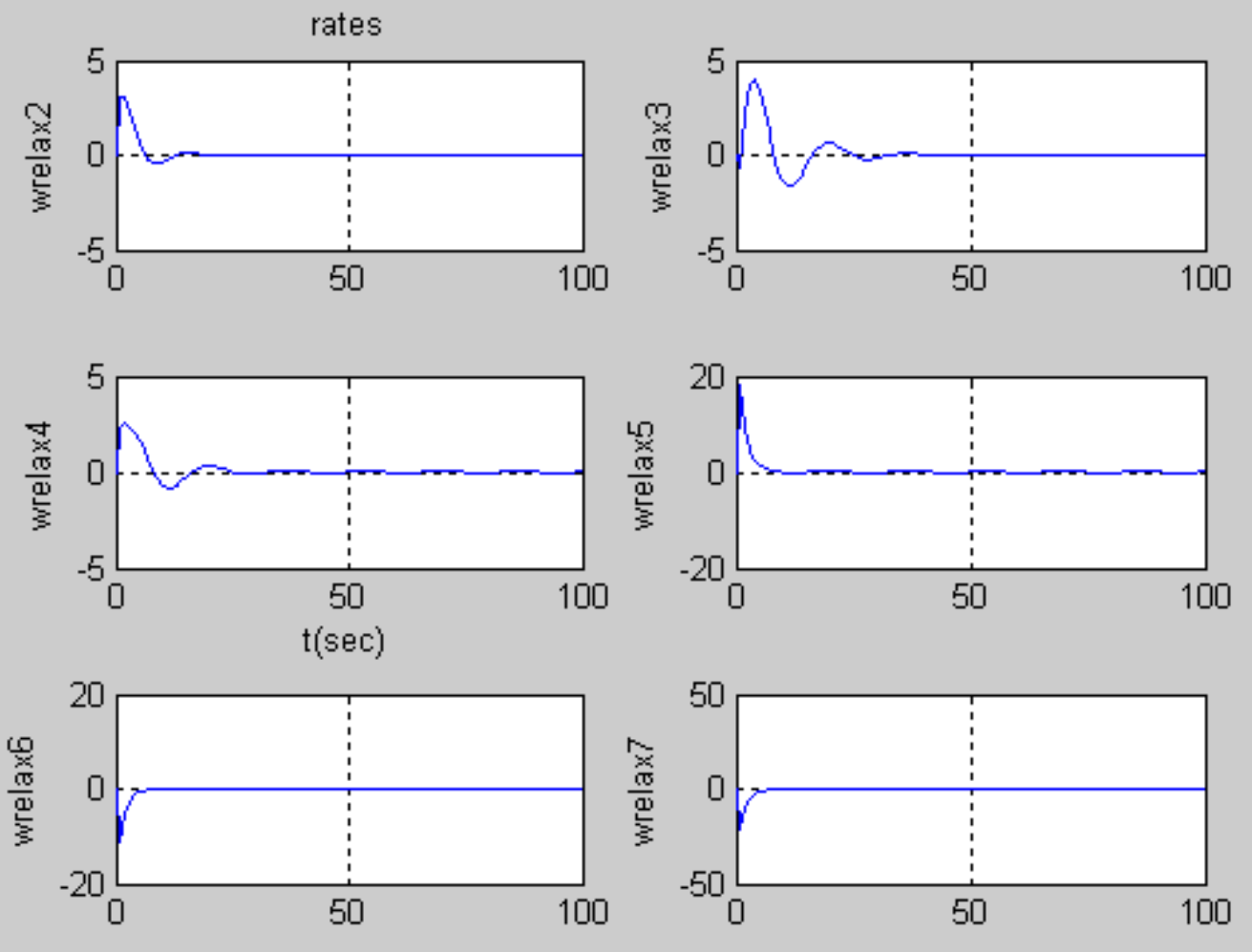
Fig.6 Joint Angles-2



- initial angles:
  [10, 20, 30, 40, 50, 60] deg

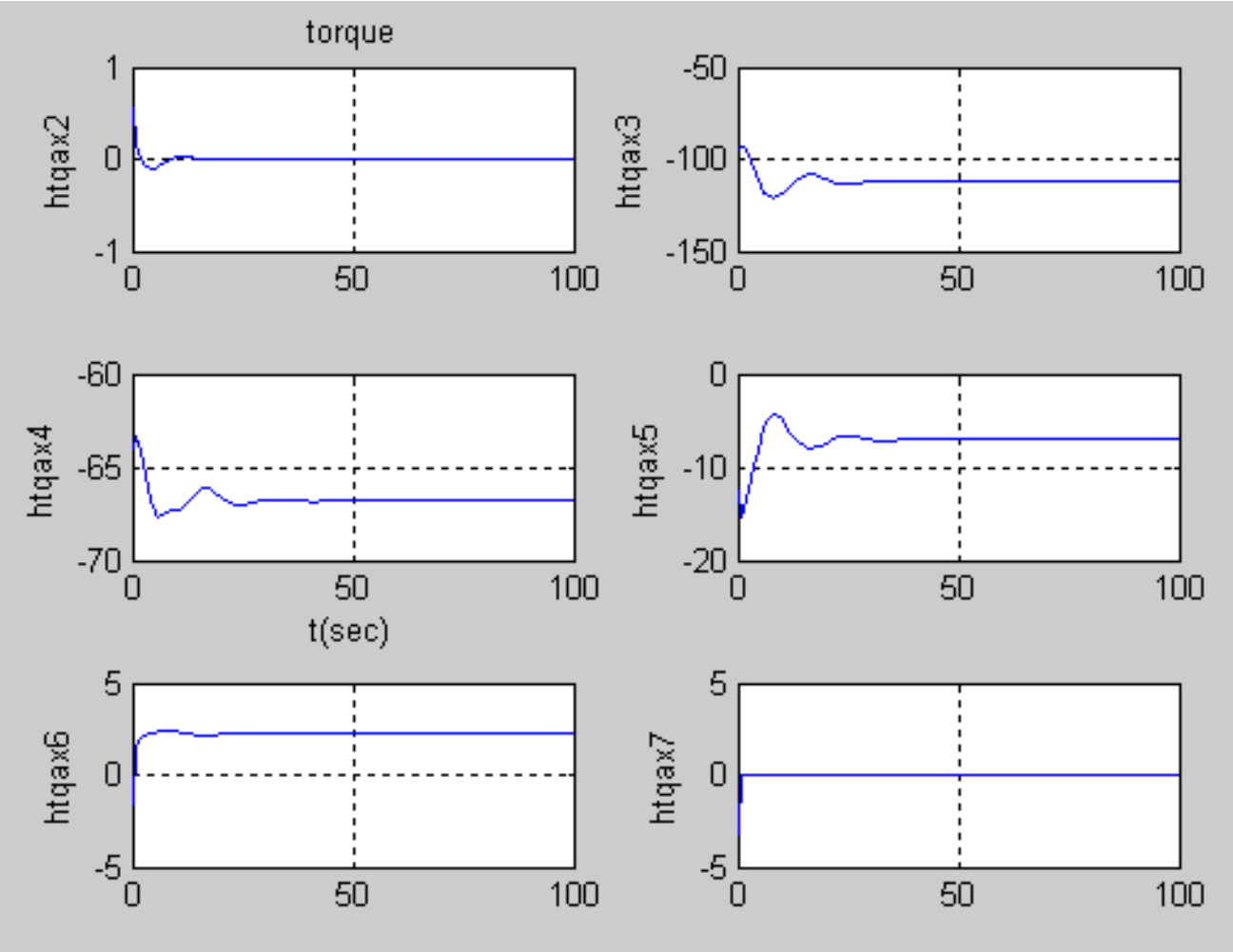
- ss joint angles:
  [20, 30, 40, 90, 20, 0] deg per command

Fig.7 Joint Rates-2



- ss joint rates:
  [0, 0, 0, 0, 0, 0]
  deg/s

Fig.8 Joint Torque-2



- ss joint torque: non zero torque for joints [3, 4, 5, 6]

Adjustable Sim Parameters

- Dtplot: plot data sample period
    a. Type 'plotdt' from the Main Menu to edit

- Dt: simulation integration step size
    a. Change it under the 'simulation' button in the Simulink window
    b. dt=0.01 for Arm6x, try larger stepsize until response diverges
    c. Generally choose dt < 1/(5*f) where f=freq of fastest process in model

- T: simulation period
    a. Change 'stop time' under the 'simulation' button in the Simulink program

- Integration method:
    a. Arm6x works with ode2 and rk4.

Exercises

actions	parameters	reference
change mass property	mass, inr, svec, dvec	pages:17-26
change initial condition	ang,wrel,wrelax, dcm0	pages:17-18
change joint type, axis	type, axis	pages:17-18
change gravity	gx, gy, gz	pages:27-28
add /remove bodies*		pages:17-18
modify input (udata)		pages:30-31
modify output (ydata)		Pages:32-33
modify plot (odata)		pages:34-35
modify simplot1.m		page:36

* Not for project licenses

actions	changes	control system config
Design your own arm control system for Arm6x	•may need new ydata	•could be different from given control system
Modify Arm6x to a Stanford arm	•change b3.(type,axis,mass prop) •need change ydata •need change udata •need change odata list and simplot1.m	•Need change gbias.m and contrl.m to accommodate the prismatic joint(3)

Simulation Notes

Subject	Arm6x	comments
gforces	comment applies to all bodies in Arm6x	gravity forces are auto-computed for all bodies in model by simx3r.dll so no xf's are needed for them
wheels	no reaction wheels needed for Arm6x	wheels can be invoked to represent angular momentum due to high speed motors
geometry	Arm6x is chain of 7 bodies with base locked to workspace and a 1dof joint between bodies	position and orientation of body parts are defined by their dvec, svec ,dcm0 and joint coordinates
b1.dcm0	comment applies here	b1.dcm0 is the orientation of base body in workspace
mass, inertia	These can be zero for b1 if it is locked to workspace	mass and inertia can be set to zero for non-terminal bodies to represent ideal massless joints

Summary

- Two examples given here demonstrated that simx3r.dll can simulate a ground fixed 6 link robot arm. The arm control shown is simply to move from one configuration to another. More complex motion can be implemented by scheduling the commands to move along a specified path.
- The value of Arm6sim is that its mass property and joint motion type can be varied to fit the robotic arm of interest . User can define a variety of Dynamics Input/Output signals to design and test his application specific control system. Another perspective is that Arm6sim can be used to examine the trade space in the design of a multi-linked robot arm, its operations and control system.