Inv_Pendulum Manual

Concurrent Dynamics International

April 2014

Objectives

- Part I:
  - Build a model file (Inv_Pendulum.txt) to simulate an inverted pendulum on a moving cart
- Part II:
  - Build a Simulink model (Inv_Pendulum.mdl) that runs per Inv_Pendulum.txt with a given control system that moves the cart to a specified position while keeping pendulum upright or near upright
  - Examples
    1. move the cart to [0, 2, 0]
    2. move the cart to [0,10,0]

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

- Inv_Pendulum.txt is a seed model_file for Enterprise license users to create other models such as: multi-joint robot arm, dangling links, and so forth

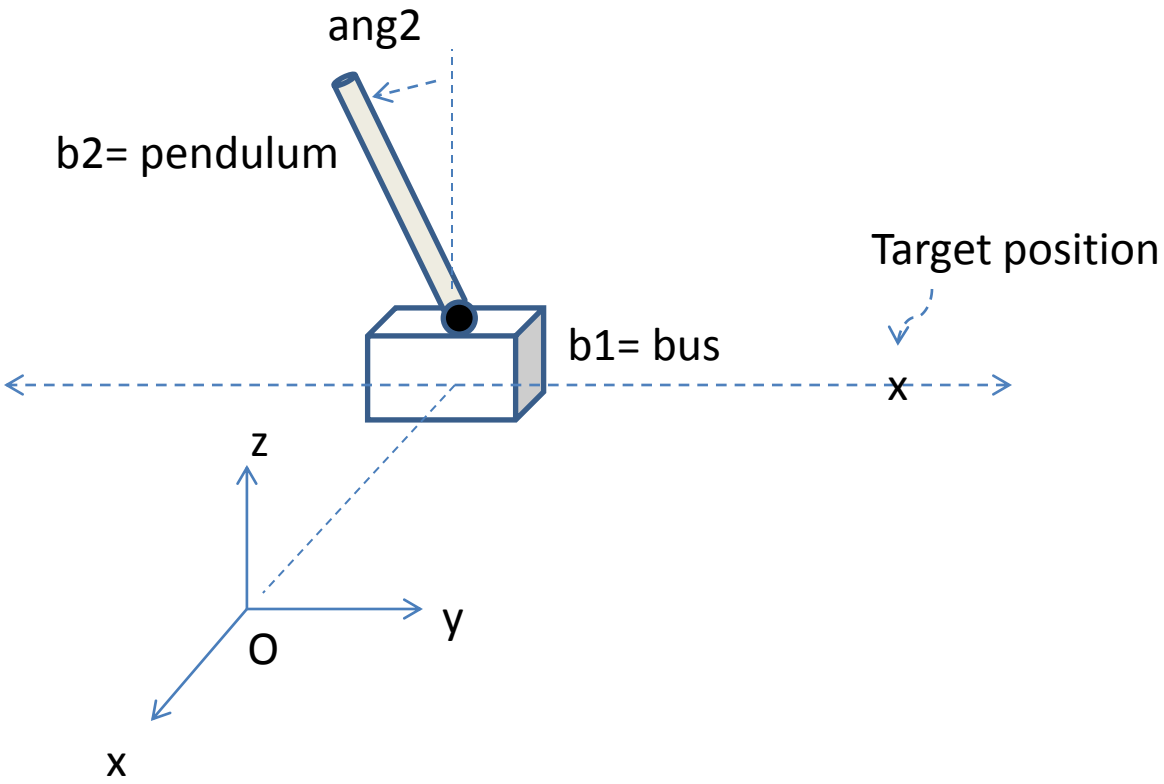
- CDI has many seed models to expedite the development of more complex mechanisms, i.e. a Stewart platform

- This manual is applicable to Project license users whose model object count is {3, 0, 0} and has a chain configuration as Inv_Pendulum

Part I Topics

- Physical model
- Buildx tasks
- Key files
- Main Menu
- Model Menus
- Body Menu
- All b1 data
- Body Actuation Signals
- Gravity Menu
- Dynamics input
- Dynamics output
- Plot data
- Simplot
- Save model data
- Exit Buildx
- Q & A

Inv_Pendulum Model



On the Simulation

- We are building a model file in Part I to support the simulation of a cart with an inverted pendulum on it. The cart is free to move along the the x-axis of the work space. Cart.force is the only control to move the cart to a specify point on the x-axis while keeping the pendulum upright. The tasks include the definition of mass property, joint dof, dynamics input and output signals required by the control system.

- Part II will show how to put together the mdl file to run two such simulations in Simulink.

- Users are encouraged to design and test the control system against models with diverse cart-pendulum mass properties.

- Users are encouraged to design a control system that moves the cart along some curved path in the x-y plane while keeping pendulum upright.

Buildx Tasks

- Build a model file to define simulation parmeters:
  - Mass property,model connectivity, degree of freedom, dynamics input/output/plot data, gravity, …etc

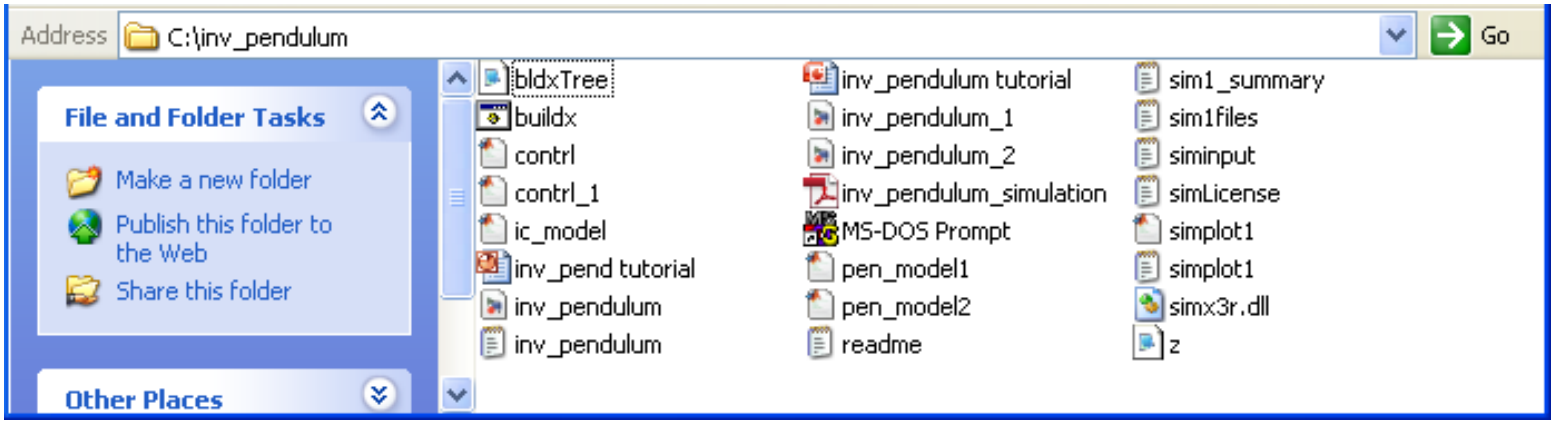
- Edit & browse the attributes of model components
  - bodies,wheels,forces,markers,gravity,initial state,…

Key Files

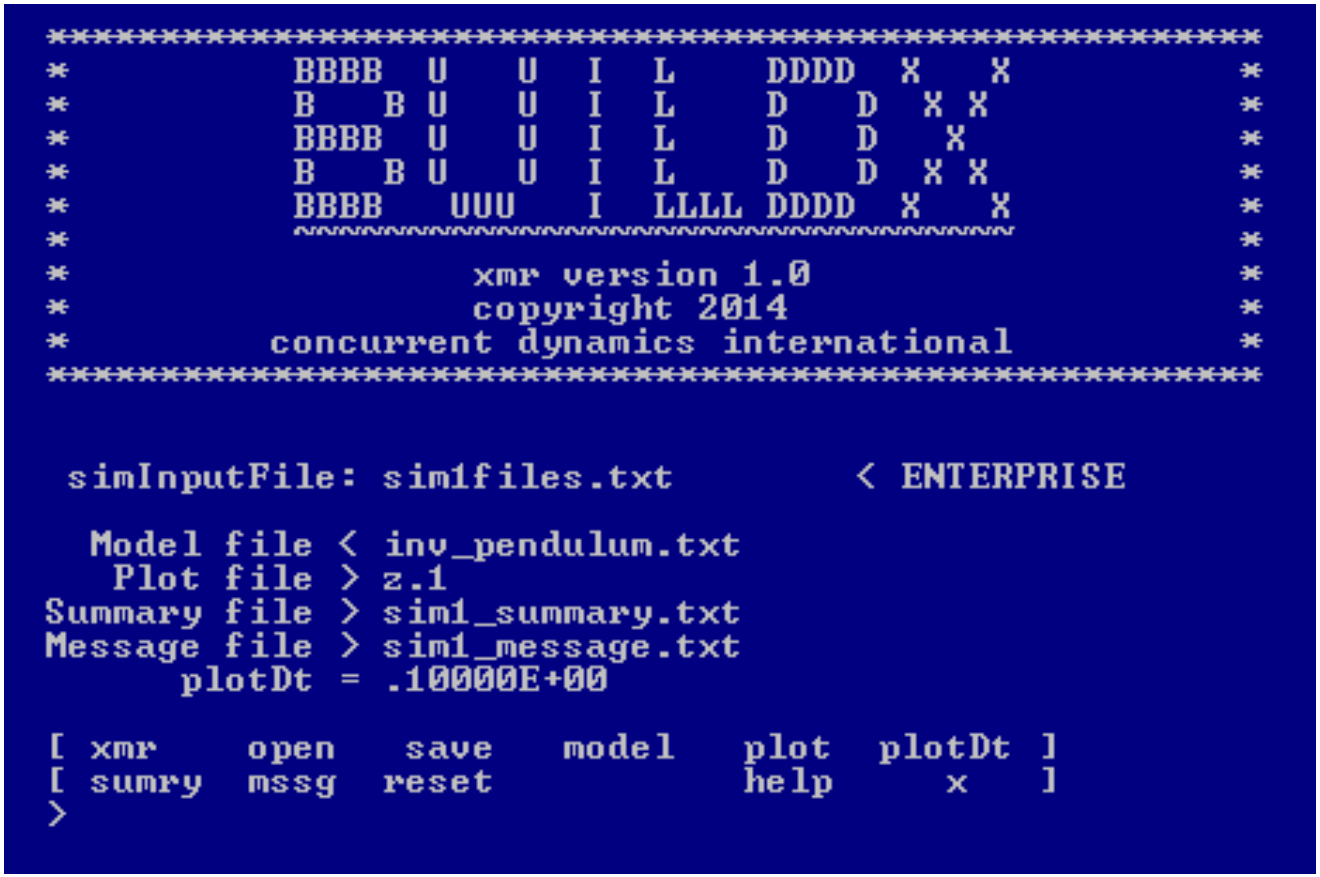
- Buildx.bat='..\buildx'
- Input file pointer: siminput.txt
- Working files: sim1files.txt
- Model_file: Inv_Pendulum.txt
- License file: simlicense.txt

Start Buildx.exe

- Click c:\Inv_Pendulum\Buildx.bat to start Buildx.exe and see the Main Menu



Main Menu



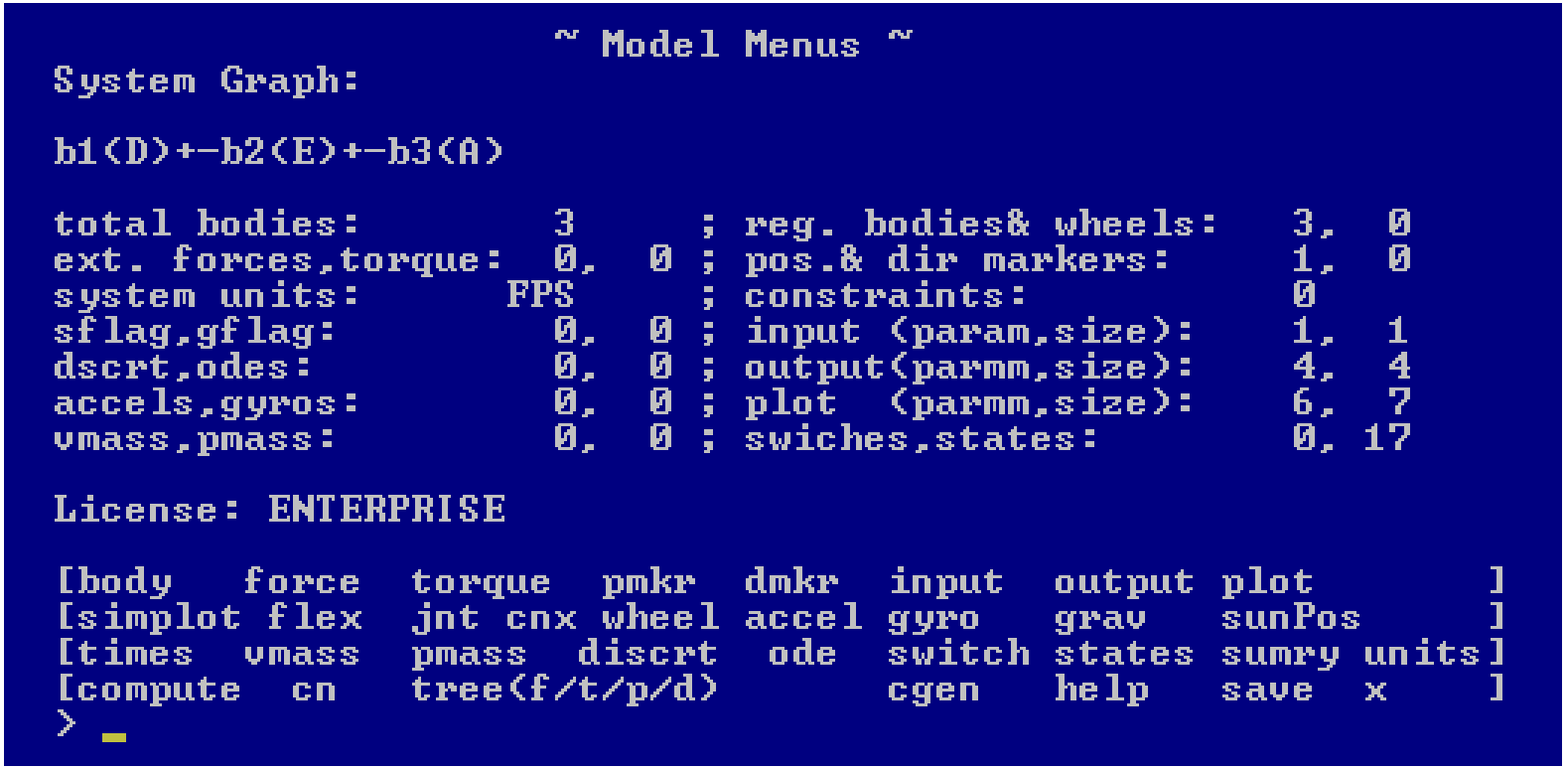
- See working files in sim1files.txt
- Type 'xmr' to go to Model Menus

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

Model Menus

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



- See 3 bodies, 0 wheels, 0 external forces

Model Menus Info

- 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 Menu

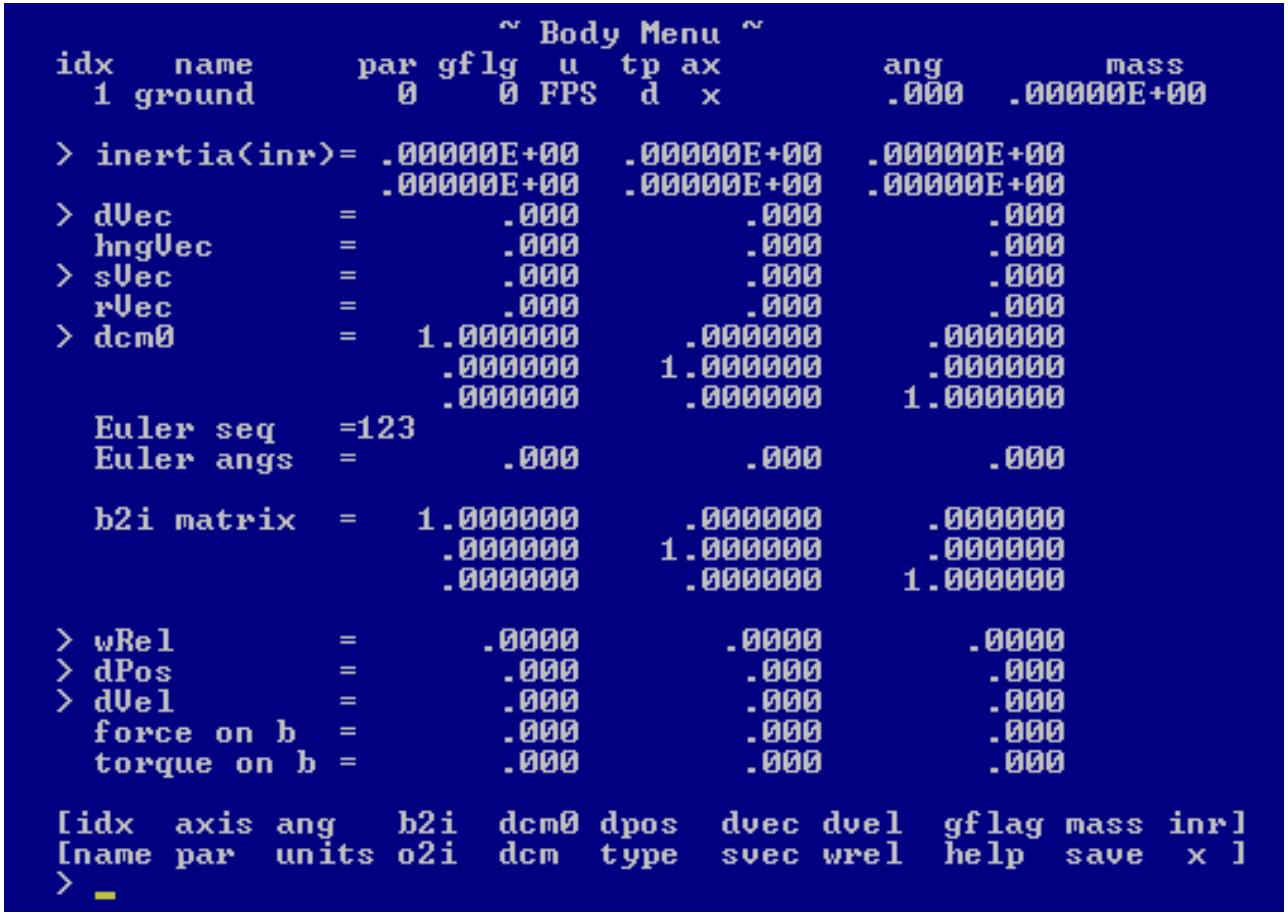
```
                          ~ Body Menu ~
   no. of bodies :  3
   sysh_eci      :          .000          .000          .000
   sysh_b1       :          .000          .000          .000
   syscm         :          .000          .000         3.000

   dvec =          .000          .000          .000
   svec =          .000          .000          .000
   hpos =          .000          .000          .000
   rpos =          .000          .000          .000
   wrel =          .000          .000          .000
   inr  =          .000          .000          .000
                  .0000         .0000         .0000

   idx name     pa  u   fl vm tp ax -- angle - -- mass --
=>   1 ground    0 FPS   0  -  D  x         .000         .000
     2 cart      1 FPS   0  -  E  y         .000        5.000
     3 pendlm    2 FPS   0  -  A  x         .000        1.000

   [ sel  edit  idx   name  par  dvec  svec  type  whl  units]
   [ axis  ang  dpos  wrel dvel  inr   mass  hpos  hvel rpos ]
   [ add  addF  cnx   jnt  rem   dmkr  pmkr  gvec  rvel  w   ]
   [ brch  cn   copy  up   down  doIc  save  help  zero  x   ]
   >
```

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(ground) Data

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



B1(ground) Data

- 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

B<j> Page Info

- 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

All B2(cart) Data

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

```
                        ~ Body Menu ~
idx   name      par gflg   u   tp ax         ang          mass
  2 cart          1    0 FPS   e  y          .000   .50000E+01

> inertia(inr)= .10000E+01   .10000E+01   .10000E+01
                .00000E+00   .00000E+00   .00000E+00
> dVec         =       .000         .000         .000
  hngVec       =       .000         .000         .000
> sVec         =       .000         .000        3.000
  rVec         =       .000         .000        3.000
> 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 to see a summary of moi data

	idx	name	ixx	iyy	izz
			ixy	ixz	iyz
	1	ground	.000000E+00	.000000E+00	.000000E+00
			.000000E+00	.000000E+00	.000000E+00
=>	2	cart	.100000E+01	.100000E+01	.100000E+01
			.000000E+00	.000000E+00	.000000E+00
	3	pendlm	.100000E+01	.100000E+01	.100000E+01
			.000000E+00	.000000E+00	.000000E+00

- 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  .000000E+00  .000000E+00  .000000E+00
 =>  2 cart      FPS  .000000E+00  .000000E+00  .000000E+00
     3 pendlm    FPS  .000000E+00  .000000E+00  .000000E+00
```

- 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  .000000E+00  .000000E+00  .000000E+00
 =>  2 cart      FPS  .000000E+00  .000000E+00  .300000E+01
     3 pendlm    FPS  .000000E+00  .000000E+00  .300000E+01
```

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

Pos Summary

- Type 'rpos' in Body Menu to see a summary of bj.cm in b1 frame

```
    idx name    ---------- ---rpos--- ----------
      1 ground        .000       .000       .000
 =>   2 cart          .000       .000      3.000
      3 pendlm        .000       .000      3.000
```

- Note: all cm's are on b1.z_axis, given all bj.ang=0 for Inv_Pendulum

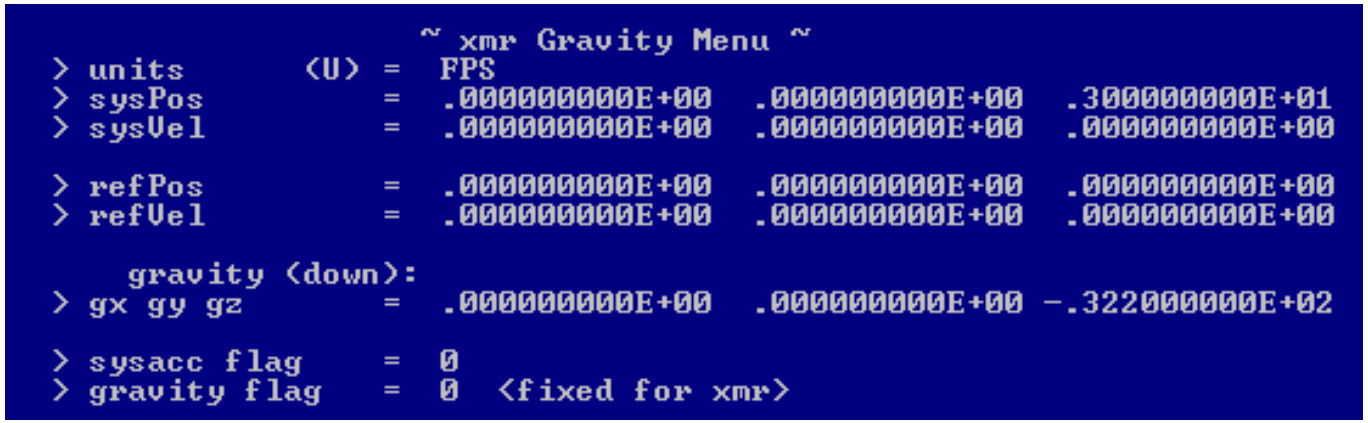
Body Actuation Signals

- 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 depends on bj.type. They are processed 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 define gravitational acceleration [gx,gy,gz] for the model
- Type 'grav' from Model Menus (page 13) 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 by default
- 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 mechanism 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)
- Inv_Pendulum input is just the lateral force on the cart: hfax,2

```
Udata list:
 1) hfax,2
```

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)
- Inv_Pendulum output list is as follows:

```
 Ydata list:

  1) angle,3          ¦ 2) hposrax,2        ¦ 3) wrelax,3
  4) hvelrax,2
```

- angle,3= b3.ang
- linear position of cart=hposrax,2= b2.pos
- wrelax,3=b3.angular rate(axial)
- linear rate of cart=hvelrax,2=b2.rate

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)
- Inv_Pendulum plot list is as follows:

```
Odata list:

 1) HPOSRAX,2        ¦ 2) ANGLE,3          ¦ 3) HVELRAX,2
 4) WRELAX,3         ¦ 5) HFRCAX,2         ¦ 6) HTQAX,3
```

- hposrax,2= b2.linear position
- angle,3= b3.ang
- hvelrax,2=b2.linear rate
- wrelax,3= b3.ang_rate
- hfrcax,2= b2.force_axial
- htqax, 3= b3.hinge_torque

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 13) or plot menu to build it (page 32)
- 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 Inv_Pendulum (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 13) 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 the model data if changes have been done to the current model. See page 35.

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 {3, 0, 0}?
  - no, for example oneCylinderEngine Project license has an object count of {5, 0, 1} and a unique parent-child relation between bodies, wheels and forces
- Is Inv_Pendulum Project license limited to just this pendulum model?
  - No, one can model a two link object
  - Change b2.type to ‘a’, and adjust b2.mass property
- How does one change the pendulum joint to a 3 dof rotational hinge?
  - Go to Body Menu and set b3.type=b
  - Important: always adjust your controller for mass property changes or changes to the joint type or 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

- Simx3r.dll Functionality
- Key Files
- Inv_Pendulum.mdl
- Control System
- Running Inv_Pendulum.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 the model_file
- Reads actuation signals (udata) from the control system in Simulink workspace
- Sends motion signals (ydata) to Simulink workspace for control system input
- Output selected data (odata) to plotfile for post-sim viewing

Key Files

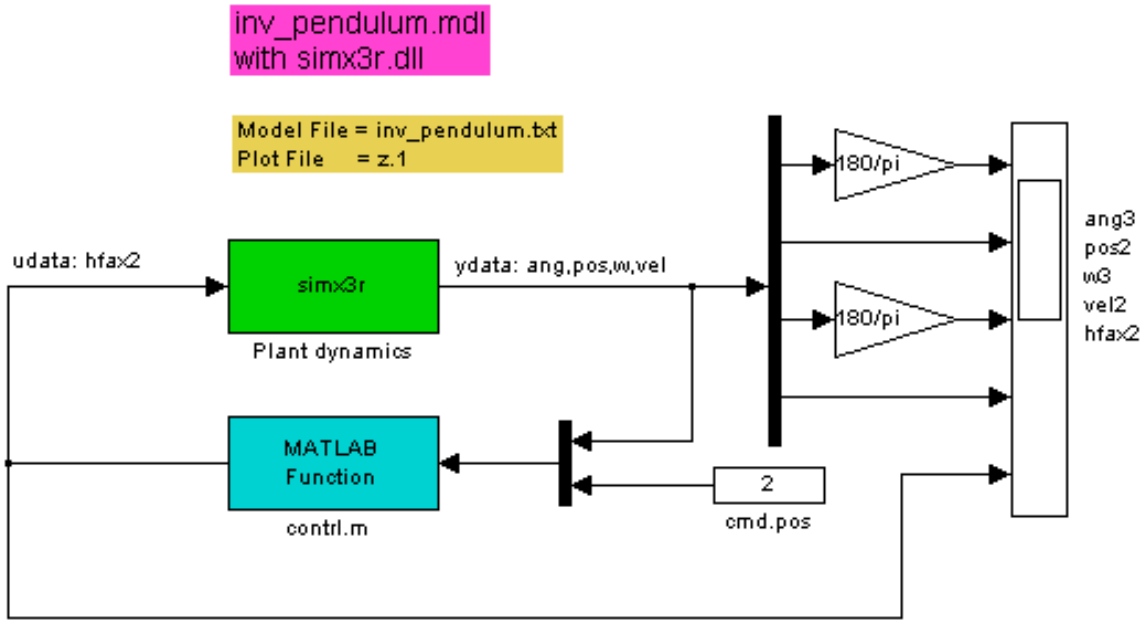
- Input file pointer: siminput.txt
- Working files definition: sim1files.txt
- Model_file: Inv_Pendulum.txt
- License file: simLicense.txt
- Simulink program: Inv_Pendulum.mdl
- Simulation engine: sim3xr.dll
- Control scripts: contrl.m

Inv_Pendulum.mdl

- Need Inv_Pendulum.mdl to run a sim3xr.dll based dynamics simulation according to Inv_Pendulum.txt
- Sends ydata to contrl.m and is actuated by udata from the latter
- Control system is user supplied and application specific
- Click on c:\Inv_Pendulum\Inv_Pendulum.mdl to open it in Simulink workspace

Inv_Pendulum.mdl

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



Control System

- Need contrl.m to define actuation signals (udata) to move the cart to a commanded position with pendulum standing upright on the cart

- Its input signals are motion signals (ydata) from xsv01.dll and a commanded cart final position

- I/O for this contrl.m:
  - Input: angle3, hposrax2, wrelax3, hvelrax2, cmd
  - Output: hfax2

Contrl.m Code

```
function f=contrl(y)
global m1 m2 L Ix g
%
% input
ang = y(1); % pendulum angle about x axis
pos = y(2); % cart position on y axis
w   = y(3); % pendulum rate about x axis
vel = y(4); % cart velocity on y axis
cmd=y(5); % commanded cart position

% kinematics and gains
sa  = sin(ang);
ca  = cos(ang);
kp  = 10^2; % omega^2
kv  = 20  ; % 2*omega
```

```
% generalized mass matrix
M   =[   Ix     -m2*L*ca  ;
       -m2*L*ca   (m1+m2) ];

B   =[0  1]'   ; % force coefficients
C   =[0  1]    ; % constraint coefficients

% rhs of force equation
beta=[ m2*L*sa*g   -m2*L*sa*w^2]';

% rhs of constraint equation
gama = L*ca*(-kv*w -kp*ang) +4*vel +1*(pos-cmd);

% solve for cart force subject to constraint
R= C*(M\B);
f= 1/R*(gama -C*(M\beta));
```

Run Inv_Pendulum.mdl

- 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

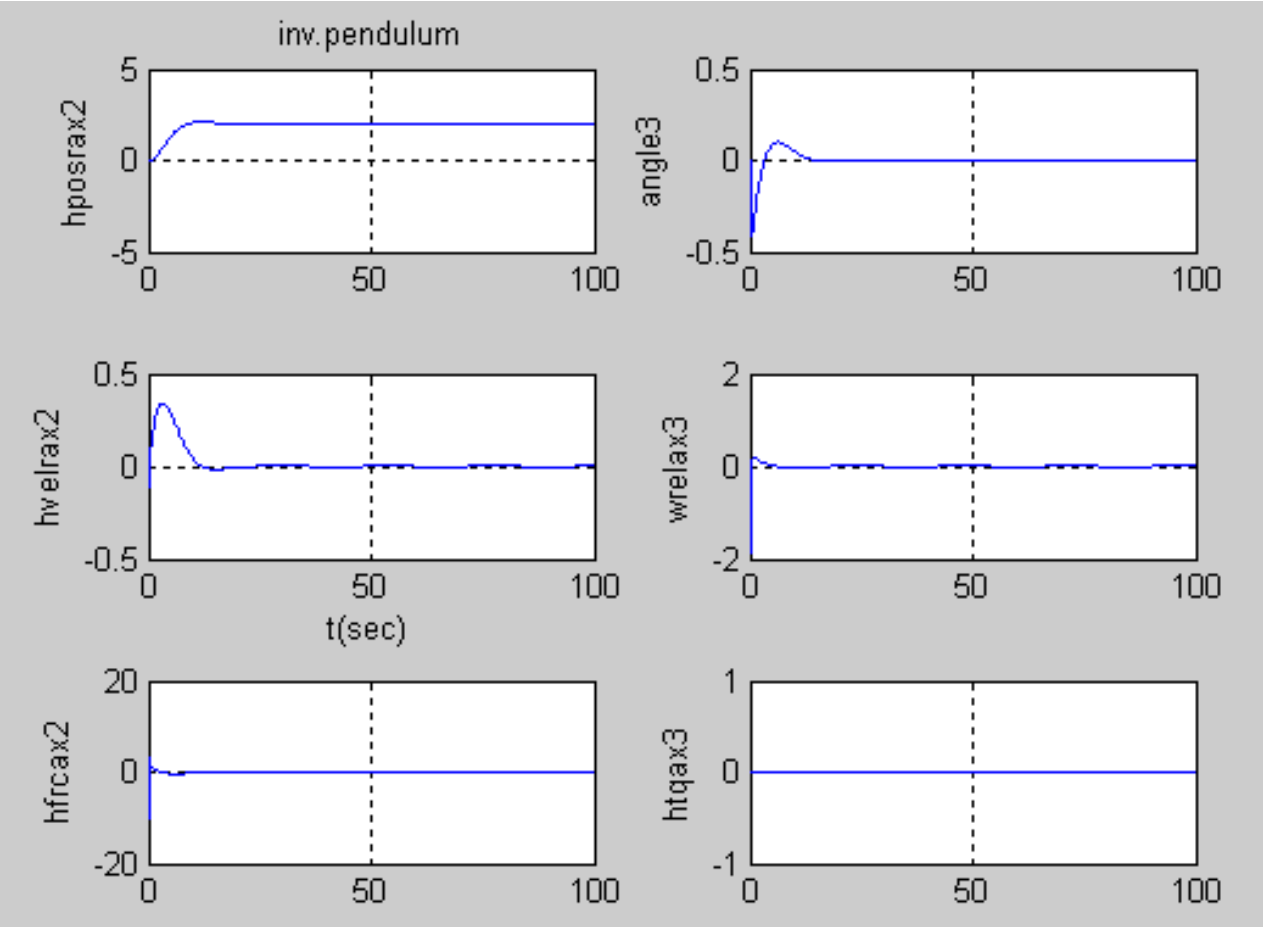
View Sim Results

- 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)

Example1

- Use Buildx to set b2.posr=0, b3.ang=0, b2.velr=0, b3.wrel=0 as initial condition
- Set cart.cmd.pos=2 in Inv_Pendulum.mdl
- Run Inv_Pendulum.mdl
- View sim result using simplot1.m
- Try with cart.cmd.pos=-2

Fig.1 Arm Motion-1



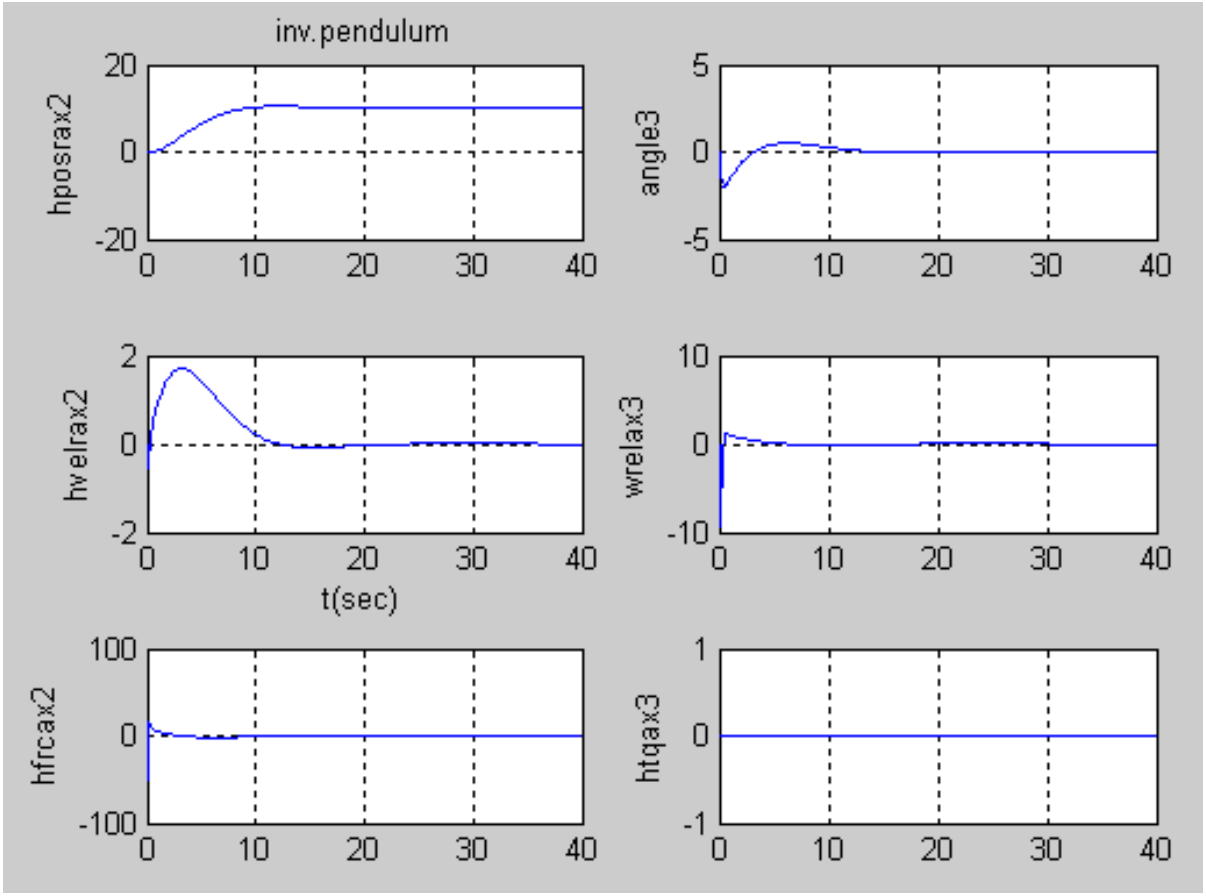
hposrax2= cart.pos
hvelrax2 = cart.vel
gfrcax2 = cart.force
angle3 = pend.tilt
wrelax3 =pend.tilt.rate
htqax3 =pend.torque

Steady state:
cart: cart.pos=2
cart.vel=0
Pend:tilt & rate=0

Example2

- Use Buildx to set b2.posr=0, b3.ang=0, b2.velr=0, b3.wrel=0 as initial condition
- Set cart.cmd.pos=10 in Inv_Pendulum.mdl
- Run Inv_Pendulum.mdl
- View sim result using simplot1.m
- Try with b2.posr=0, b3.ang=0, b2.velr=1, b3.wrel=-1 as initial condition

Fig.2 Arm Motion-2



hposrax2= cart.pos
hvelrax2 = cart.vel
gfrcax2 = cart.force
angle3 = pend.tilt
wrelax3 =pend.tilt.rate
htqax3 =pend.torque

Steady state:
cart: cart.pos=10
cart.vel=0
Pend:tilt & rate=0

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 program
    b. dt=0.01 for arm6a.mdl, 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. several methods are available from the 'simulation' control dialog in the Simulink window

- Inv_Pendulum works with Ode2 and rk4.

Exercises

actions	parameters	reference
change mass property	mass, inr, svec, dvec	pages:16-24
change initial condition	ang,wrel,wrelax, dcm0	pages:16-17
change joint type, axis	type, axis	pages:16-17
change gravity	gx, gy, gz	Pages: 26-27
add /remove bodie*		pages:16-17
modify input (udata)		pages:29-30
modify output (ydata)		pages:31-32
modify plot (odata)		pages:33-34
modify simplot1.m		page:35

* Not for project licenses

actions	changes	control system config
design your own Inv_Pendulum control system	•may need new ydata	•can be different from given control system

Simulation Notes

Subject	Arm6x	comments
gforces	comment applies to all bodies in arm6x	gravity forces are auto-computed for a given mechanism 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 3 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 controlled motion of an inverted pendulum on top of a cart. The cart was shown to move to commanded position in a smooth manner using a PD controller.
- The value of Inv_pendulum is that its mass property can be varied to fit the experiment of interest. The condition of a successful experiment depends on several factors such as the mass property of the pendulum, its length, the mass of the cart, the control algorithm and the bandwidth of the controller. User can define a variety of Dynamics Input/Output signals to design and test his application specific control system. Another view is that Inv_Pendulum can be used to examine the trade space in the design of an inverted pendulum experiment, its operations and control system.