Report on Hybrid

Generated by MTT using :
(mtt -u -q -q Hybrid rep pdf)

Tue Aug 19 14:55:49 BST 2003

Contents

List of Figures

Part I

Hybrid

Chapter 1

Bounce

1.1 Bounce_abg.tex

MTT command:

```
mtt Bounce abg tex
```

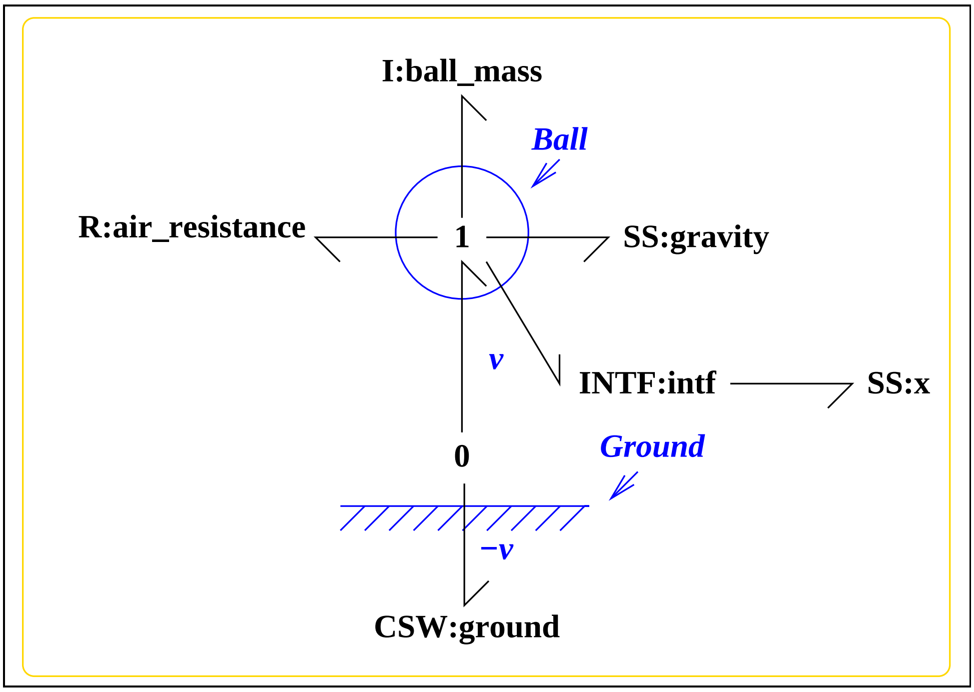

Figure 1.1: System **Bounce**: acausal bond graph

The acausal bond graph of system **Bounce**, togehter with a schematic diagram is displayed in Figure 1.1 (on page 9) and its label file is listed in Section 1.1.1 (on page 10). The subsystems are listed in Section 1.1.2 (on page 11).

The model uses the **CSW** switched **C** element to simulate contact with the ground. The corresponding switching function (See Section 1.5), is based on the height above the ground h as follows:

$$i_{sw} = \begin{cases} 0 & \text{if } h > 0 \\ -1 & \text{if } h \leq 0 \end{cases} \tag{1.1}$$

In other words, the component acts as an ideal spring when the ball is in contact with the ground yet has no effect when the ball is not in contact with the ground.

The ball is modelled as a point mass (the **I** component) and a linear resistance to motion (the (the **R** component).

The system was simulated for 100 time units and the resultant height is plotted in Figure 1.4. The ball was released at zero velocity from a height of ten units. The bounce height decreases due to the effect of the modelled air resistance.

1.1.1 Summary information

System Bounce::Bouncing ball example (hybrid) ¡Detailed description here¿

Interface information:

This component has no ALIAS declarations

Variable declarations:

This component has no PAR declarations

Units declarations:

This component has no UNITs declarations

The label file: Bounce_lbl.txt

```
%SUMMARY Bounce: Bouncing ball example (hybrid)
%DESCRIPTION <Detailed description here>
%% Label file for system Bounce (Bounce_lbl.txt)
```

```
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %% Version control history
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %% $Id: Bounce_lbl.txt,v 1.2 2001/10/05 11:24:30 gawthrop Exp $
% %% $Log: Bounce_lbl.txt,v $
% %% Revision 1.2  2001/10/05 11:24:30  gawthrop
% %% Updated for new mtt features - input.txt etc
% %%
% %% Revision 1.1  1997/09/11 09:54:22  peterg
% %% Initial revision
% %%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Each line should be of one of the following forms:
% a comment (ie starting with %)
% Component-name CR_name arg1,arg2,..argn
% blank



% Component type INTF
intf

% Component type CSW
ground lin state,k_g

% Component type I
ball_mass lin flow,m

% Component type R
air_resistance lin flow,r

% Component type SS
gravity SS external,internal
x SS external,0
```

1.1.2 Subsystems

- CSW: Switched C component (1) No subsystems.

- INTF: flow integrator (1) No subsystems.

1.1.3 CSW

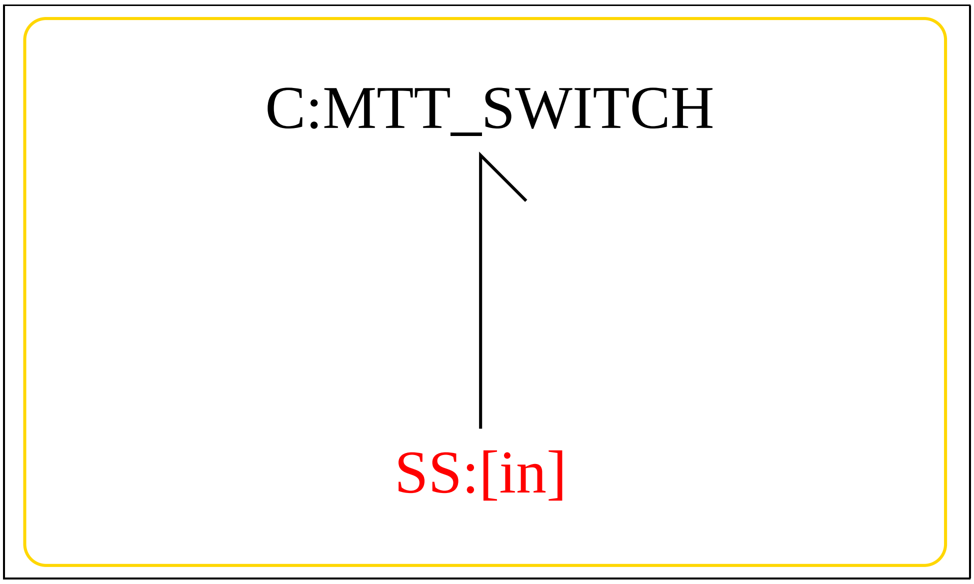

Figure 1.2: System **CSW**: acausal bond graph

The acausal bond graph of system **CSW** is displayed in Figure 3.2 (on page 42) and its label file is listed in Section 3.1.3 (on page 41). The subsystems are listed in Section 3.1.3 (on page 44).

Summary information

System CSW::Switched C component CSW acts as an C component except when the -s -c option is used. When the -s -c option is used: The component label is used as a variable name (eg Name) In the ode simulation: Name = 1 implies normal C (closed switch) Name = 0 implies state=0 (open switch) Name = -1 implies state¿0 (Diode)

Interface information:

Parameter $1 represents actual parameter **effort,c_s**

Parameter $1 represents actual parameter **lin**

Port out represents actual port **in**

Variable declarations:

This component has no PAR declarations

Units declarations:

This component has no UNITs declarations

The label file: CSW_lbl.txt

```
%SUMMARY CSW: Switched C component
%DESCRIPTION CSW acts as an C component except when the -s -c option is
%DESCRIPTION When the -s -c option is used:
%DESCRIPTION   The component label is used as a variable name (eg Name)
%DESCRIPTION   In the ode simulation:
%DESCRIPTION     Name = 1 implies normal C (closed switch)
%DESCRIPTION     Name = 0 implies state=0 (open switch)
%DESCRIPTION     Name = -1 implies state>0 (Diode)

%ALIAS out in
%ALIAS $1 lin
%ALIAS $1 effort,c_s

%% Label file for system CSW (CSW_lbl.txt)

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %% Version control history
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %% $Id: CSW_lbl.txt,v 1.3 1998/07/26 13:30:33 peterg Exp $
% %% $Log: CSW_lbl.txt,v $
% %% Revision 1.3  1998/07/26 13:30:33  peterg
% %% Added aliases
% %%
% %% Revision 1.2  1997/09/11 09:00:52  peterg
% %% More documentation.
% %%
% %% Revision 1.1  1997/06/16  10:55:20  peterg
% %% Initial revision
% %%
% %% Revision 1.1  1997/06/03 15:26:09  peterg
% %% Initial revision
% %%
```

```
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Each line should be of one of the following forms:
% a comment (ie starting with %)
% Component-name CR_name arg1,arg2,..argn
% blank


% Component type C
MTT_SWITCH lin effort,c_s

% Component type SS
[in] SS external,external
```

Subsystems

No subsystems.

1.1.4 INTF

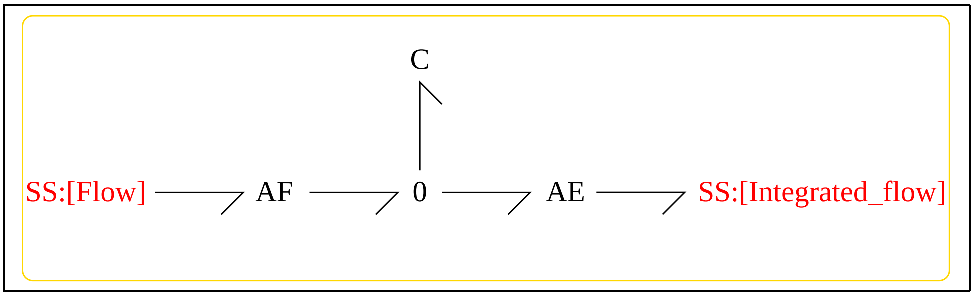

Figure 1.3: System **INTF**: acausal bond graph

The acausal bond graph of system **INTF** is displayed in Figure 2.3 (on page 29) and its label file is listed in Section 2.1.4 (on page 29). The subsystems are listed in Section 2.1.4 (on page 30).

INTF is a two-port component where the effort on port [out] is the integral of the flow on port [in].

Summary information

System INTF::flow integrator Port [in]: Flow to be integrated Port [out]: Effort = integral of flow on port [in]

Interface information:

Port in represents actual port **Flow**

Port out represents actual port **Integrated_flow**

Variable declarations:

This component has no PAR declarations

Units declarations:

This component has no UNITs declarations

The label file: INTF_lbl.txt

```
%% Label file for system INTF (INTF_lbl.txt)
%SUMMARY INTF: flow integrator
%DESCRIPTION Port [in]: Flow to be integrated
%DESCRIPTION Port [out]: Effort = integral of flow on port [in]


% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %% Version control history
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %% $Id: INTF_lbl.txt,v 1.3 1998/07/16 07:35:10 peterg Exp $
% %% $Log: INTF_lbl.txt,v $
% %% Revision 1.3  1998/07/16 07:35:10  peterg
% %% Aliased version
% %%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% Port aliases
%ALIAS in Flow
%ALIAS out Integrated_flow
```

```
% Argument aliases

%% Each line should be of one of the following forms:
%       a comment (ie starting with %)
%       component-name cr_name arg1,arg2,..argn
%       blank

% ---- Component labels ----

% Component type SS
[Flow] SS external,external
[Integrated_flow] SS external,external
```

Subsystems

No subsystems.

1.2 Bounce_struc.tex

MTT command:

```
mtt Bounce struc tex
```

List of inputs for system Bounce			
	Component	System	Repetition
1	gravity	Bounce__gravity	1

List of outputs for system Bounce			
	Component	System	Repetition
1	x	Bounce__x	1

List of states for system Bounce			
	Component	System	Repetition
1	mttC	Bounce__intf__mttC	1
2	MTT_SWITCH	Bounce__ground__MTT_SWITCH	1
3	ball_mass	Bounce__ball_mass	1

1.3 Bounce_ode.tex

MTT command:

```
mtt Bounce ode tex
```

$$\begin{aligned} \dot{x}_1 &= \frac{x_3}{m} \\ \dot{x}_2 &= \frac{(-x_3)}{m} \\ \dot{x}_3 &= \frac{(k_g m x_2 - m u_1 - x_3 r)}{m} \end{aligned} \tag{1.2}$$

$$y_1 = x_1 \tag{1.3}$$

1.4 Bounce_simpar.txt

MTT command:

```
mtt Bounce simpar txt
```

```
# Simulation parameters for system Bounce (Bounce_simpar.txt)
# Generated by MTT on Sat Jul 25 15:57:56 BST 1998.
###############################################################
## Version control history
###############################################################
## $Id: Bounce_simpar.txt,v 1.2 2000/12/28 17:45:24 peterg Exp $
## $Log: Bounce_simpar.txt,v $
## Revision 1.2  2000/12/28 17:45:24  peterg
## To RCS
##
## Revision 1.1  1998/10/01 19:21:04  peterg
## Initial revision
##
###############################################################
FIRST=0.0;
LAST=10.0;
DT=0.02;
STEPFACTOR=1;
```

1.5 Bounce_input.txt

MTT command:

```
mtt Bounce input txt
```

```
# Numerical parameter file (Bounce_input.txt)
# Generated by MTT at Wed Jul  2 10:47:49 BST 1997

# %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
# %% Version control history
# %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
# %% $Id: Bounce_input.txt,v 1.5 2003/06/11 15:58:14 gawthrop Exp
# %% $Log: Bounce_input.txt,v $
# %% Revision 1.5  2003/06/11 15:58:14  gawthrop
# %% Updated examples for latest MTT.
# %%
# %% Revision 1.4  2001/10/05 11:24:29  gawthrop
# %% Updated for new mtt features - input.txt etc
# %%
# %% Revision 1.3  2000/05/18 11:30:14  peterg
# %% Moved switch logic to logic.txt
# %%
# %% Revision 1.2  1998/10/01 19:20:33  peterg
# %% Cahged switch function
# %%
# %% Revision 1.1  1998/07/25 18:35:02  peterg
# %% Initial revision
# %%
# %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

# Set the inputs
## Removed by MTT on Fri Oct  5 10:37:56 BST 2001: mttu(1) = 9.81;


## Removed by MTT on Tue Jun 10 10:22:08 BST 2003: bounce_gravity
= 9.81; # Added by MTT on Fri Oct 05 10:37:59 BST 2001
bounce__gravity = 9.81;
```

1.6 Bounce_logic.txt

MTT command:

```
mtt Bounce logic txt
```

```
# -*-octave-*- Put Emacs into octave-mode
# Simulation parameters for system Bounce (Bounce_logic.txt)
# Generated by MTT on Thu May 18 12:29:15 BST 2000.
###############################################################
## Version control history
###############################################################
## $Id: Bounce_logic.txt,v 1.4 2003/06/11 15:58:24 gawthrop Exp $
## $Log: Bounce_logic.txt,v $
## Revision 1.4  2003/06/11 15:58:24  gawthrop
## Updated examples for latest MTT.
##
## Revision 1.3  2002/09/29 13:14:12  geraint
## Updated names.
##
## Revision 1.2  2000/12/28 17:45:24  peterg
## To RCS
##
## Revision 1.1  2000/05/18 11:29:58  peterg
## Initial revision
##
###############################################################


# Set the switches

## On when below ground level
bounce__ground__mtt_switch_logic = (bounce__intf__mttc < 0);
```

1.7 Bounce_numpar.txt

MTT command:

```
mtt Bounce numpar txt
```

```
# Numerical parameter file (Bounce_numpar.txt)
# Generated by MTT at Wed Jul  2 10:47:47 BST 1997

# %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
# %% Version control history
# %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
# %% $Id: Bounce_numpar.txt,v 1.2 2001/10/05 11:24:30 gawthrop Exp
# %% $Log: Bounce_numpar.txt,v $
# %% Revision 1.2  2001/10/05 11:24:30  gawthrop
# %% Updated for new mtt features - input.txt etc
# %%
# %% Revision 1.1  2000/12/28 17:45:24  peterg
# %% To RCS
# %%
# %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

# Parameters
k_g =  100.0; # Ground stiffness
m =  1.0; # Ball mass
r =  1.0; # Air resistance
```

1.8 Bounce_odeso.ps

MTT command:

```
mtt Bounce odeso ps
```

This representation is given as Figure 1.4 (on page 21).

1.9 Bounce_rep.txt

MTT command:

```
mtt Bounce rep txt
```

```
# Outline report file for system Bounce (Bounce_rep.txt)

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %% Version control history
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

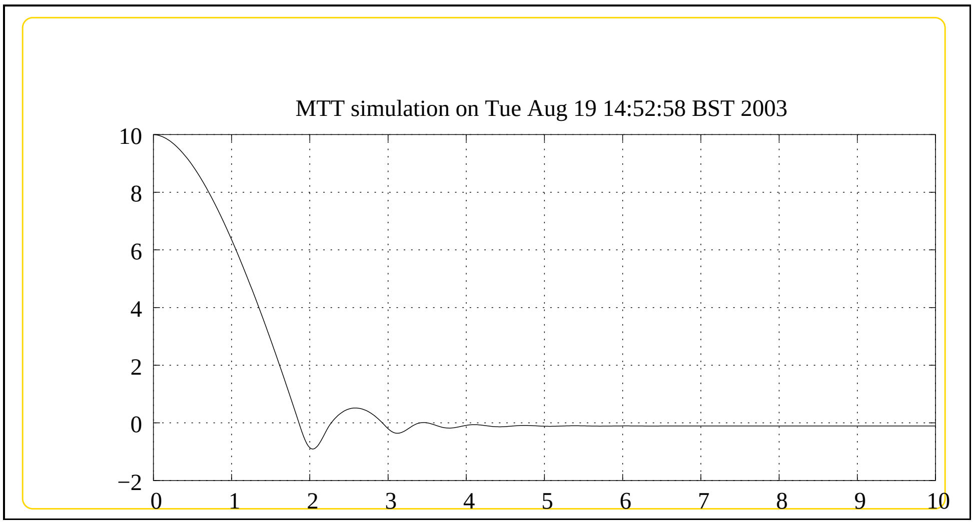

Figure 1.4: System **Bounce**, representation odeso (-noargs)

```
% %% $Id: Bounce_rep.txt,v 1.2 2001/10/05 11:24:30 gawthrop Exp $
% %% $Log: Bounce_rep.txt,v $
% %% Revision 1.2  2001/10/05 11:24:30  gawthrop
% %% Updated for new mtt features - input.txt etc
% %%
% %% Revision 1.1  2000/05/19 07:09:19  peterg
% %% Initial revision
% %%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

mtt Bounce abg tex
mtt Bounce struc tex
mtt Bounce ode tex


mtt Bounce simpar txt
mtt Bounce input txt
mtt Bounce logic txt
mtt Bounce numpar txt

mtt Bounce odeso ps

mtt Bounce rep txt
```

Chapter 2

BouncingRod

2.1 BouncingRod_abg.tex

MTT command:

```
mtt BouncingRod abg tex
```

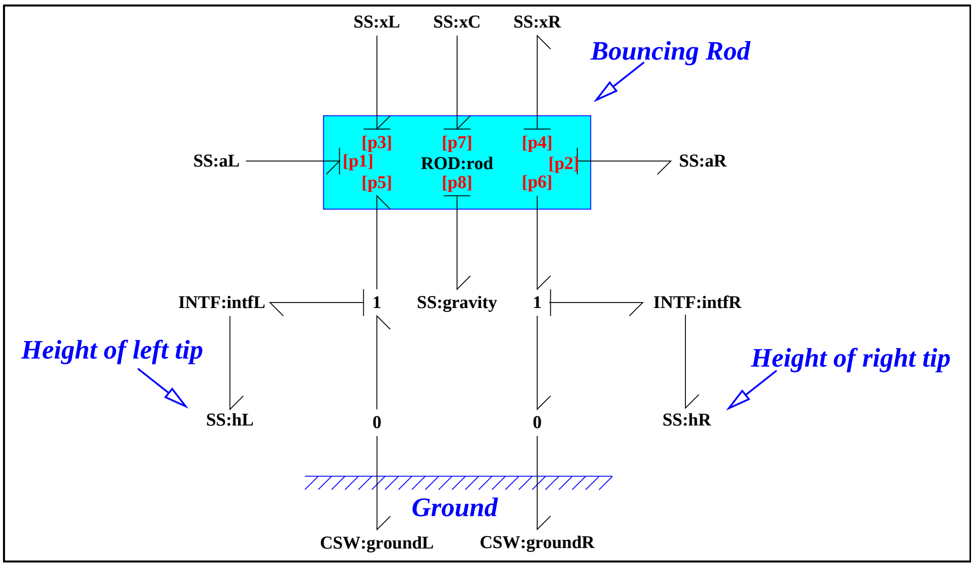

Figure 2.1: System **BouncingRod**: acausal bond graph

The acausal bond graph of system **BouncingRod** is displayed in Figure 2.1 (on page 23) and its label file is listed in Section 2.1.1 (on page 24). The subsystems are listed in Section 2.1.2 (on page 26).

The system consists of a uniform rod of mass 1kg, length 2m (and therefore of inertia about the mass centre of $\frac{1}{3}$kgm^2. The rod is released at an angle of $\frac{\pi}{4}$ from the vertical, the mass centre is 10m above the ground and all velocities are initially zero. The gravitational constant is taken as unity.
The ground is modeled as an ideal compliance in the vertical direction with compliance of 0.1mN^{-1} and it is assumed that contact takes place at the rod tips only. There is no horizontal resistance to motion. This idealised setup is modeled by a two **CSW** components, one for each rod tip, modulated by the height of each rod tip above the ground: each **CSW** is off when the corresponding height is positive.
The system was simulated for 100 time units and the resultant height of each tip is plotted in Figure 2.5. The rod was released at zero velocity from a height of ten units and at an angle of $\frac{\pi}{4}$ radians to the vertical. The oscillations in height are due to rod rotation about its mass centre. The bounce height changes due to energy transfer to and from the rod rotation about its mass centre.

2.1.1 Summary information

System BouncingRod::Two-dimensional bouncing rod (hybrid) A uniform rod bounces on a compliant surface – the CSW component is used.

Interface information:

This component has no ALIAS declarations

Variable declarations:

This component has no PAR declarations

Units declarations:

This component has no UNITs declarations

The label file: BouncingRod_lbl.txt

```
%SUMMARY BouncingRod: Two-dimensional bouncing rod (hybrid)
%DESCRIPTION A uniform rod bounces on a compliant surface -- the
%DESCRIPTION CSW component is used.

%% Label file for system BouncingRod (BouncingRod_lbl.txt)
```

```
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %% Version control history
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %% $Id: BouncingRod_lbl.txt,v 1.6 2001/10/05 11:24:34 gawthrop Exp $
% %% $Log: BouncingRod_lbl.txt,v $
% %% Revision 1.6  2001/10/05 11:24:34  gawthrop
% %% Updated for new mtt features - input.txt etc
% %%
% %% Revision 1.5  2000/12/04 08:36:23  peterg
% %% Declare pi
% %%
% %% Revision 1.4  2000/05/18 10:30:39  peterg
% %% New SS form
% %%
% %% Revision 1.3  1997/09/11 09:55:25  peterg
% %% Added documentation
% %%
% %% Revision 1.2  1997/09/11 08:37:29  peterg
% %% Added description
% %%
% %% Revision 1.1  1997/07/06 16:18:37  peterg
% %% Initial revision
% %%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Each line should be of one of the following forms:
% a comment (ie starting with %)
% Component-name CR_name arg1,arg2,..argn
% blank



% Component type CSW
groundL lin effort,epsilon
groundR lin effort,epsilon

% Component type INTF
intfL
intfR

% Component type ROD
```

```
rod none l;l;j;m

% Component type SS
aL SS 0,internal
aR SS 0,internal
gravity SS external,internal
hL SS external,0
hR SS external,0
xC SS 0,internal
xL SS 0,internal
xR SS 0,internal
```

2.1.2 Subsystems

- CSW: Switched C component (2) No subsystems.
- INTF: flow integrator (2) No subsystems.
- ROD: rigid rod in two dimensions (1)
  - INTF: flow integrator (1)

2.1.3 CSW

The acausal bond graph of system **CSW** is displayed in Figure 3.2 (on page 42) and its label file is listed in Section 3.1.3 (on page 41). The subsystems are listed in Section 3.1.3 (on page 44).

Summary information

System CSW::Switched C component CSW acts as an C component except when the -s -c option is used. When the -s -c option is used: The component label is used as a variable name (eg Name) In the ode simulation: Name = 1 implies normal C (closed switch) Name = 0 implies state=0 (open switch) Name = -1 implies state¿0 (Diode)

Interface information:

Parameter $1 represents actual parameter **effort,c_s**

Parameter $1 represents actual parameter **lin**

Port out represents actual port **in**

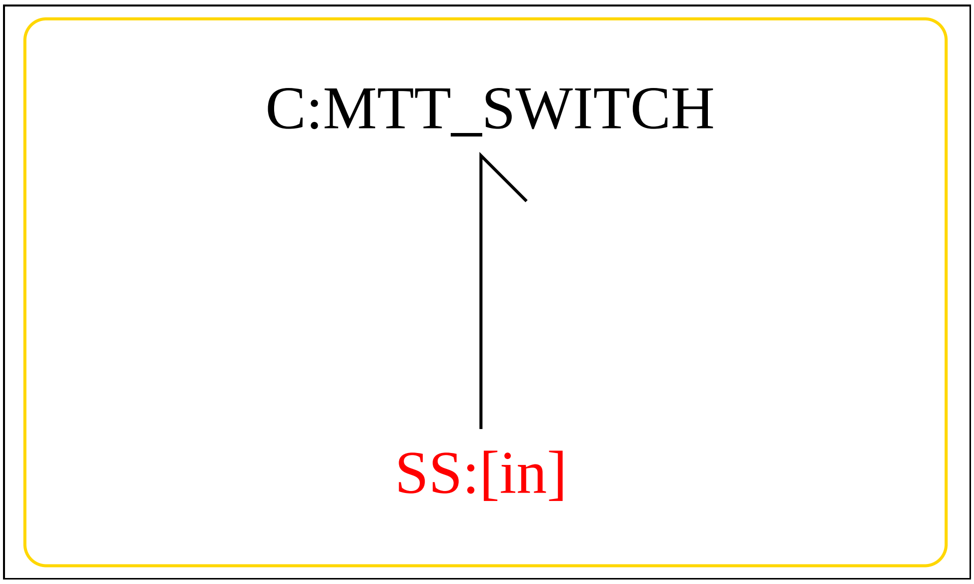

Figure 2.2: System **CSW**: acausal bond graph

Variable declarations:

This component has no PAR declarations

Units declarations:

This component has no UNITs declarations

The label file: CSW_lbl.txt

```
%SUMMARY CSW: Switched C component
%DESCRIPTION CSW acts as an C component except when the -s -c option is
%DESCRIPTION When the -s -c option is used:
%DESCRIPTION   The component label is used as a variable name (eg Name)
%DESCRIPTION   In the ode simulation:
%DESCRIPTION     Name = 1 implies normal C (closed switch)
%DESCRIPTION     Name = 0 implies state=0 (open switch)
%DESCRIPTION     Name = -1 implies state>0 (Diode)

%ALIAS out in
%ALIAS $1 lin
%ALIAS $1 effort,c_s
```

```
%% Label file for system CSW (CSW_lbl.txt)

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %% Version control history
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %% $Id: CSW_lbl.txt,v 1.3 1998/07/26 13:30:33 peterg Exp $
% %% $Log: CSW_lbl.txt,v $
% %% Revision 1.3  1998/07/26 13:30:33  peterg
% %% Added aliases
% %%
% %% Revision 1.2  1997/09/11 09:00:52  peterg
% %% More documentation.
% %%
% %% Revision 1.1  1997/06/16  10:55:20  peterg
% %% Initial revision
% %%
% %% Revision 1.1  1997/06/03 15:26:09  peterg
% %% Initial revision
% %%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Each line should be of one of the following forms:
% a comment (ie starting with %)
% Component-name CR_name arg1,arg2,..argn
% blank



% Component type C
MTT_SWITCH lin effort,c_s

% Component type SS
[in] SS external,external
```

Subsystems

No subsystems.

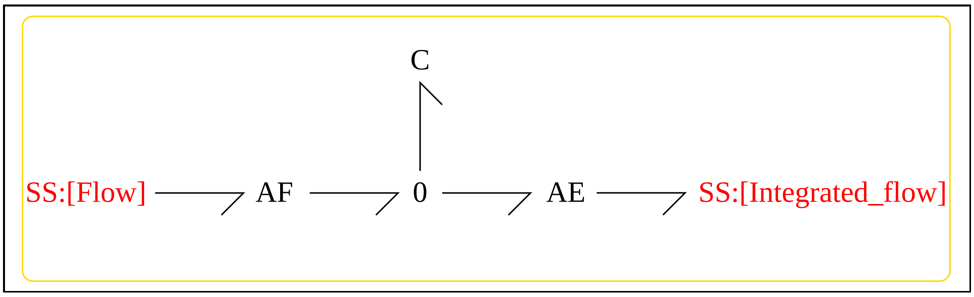

Figure 2.3: System **INTF**: acausal bond graph

2.1.4 INTF

The acausal bond graph of system **INTF** is displayed in Figure 2.3 (on page 29) and its label file is listed in Section 2.1.4 (on page 29). The subsystems are listed in Section 2.1.4 (on page 30).

INTF is a two-port component where the effort on port [out] is the integral of the flow on port [in].

Summary information

System INTF::flow integrator Port [in]: Flow to be integrated Port [out]: Effort = integral of flow on port [in]

Interface information:

Port in represents actual port **Flow**

Port out represents actual port **Integrated_flow**

Variable declarations:

This component has no PAR declarations

Units declarations:

This component has no UNITs declarations

The label file: INTF_lbl.txt

```
%% Label file for system INTF (INTF_lbl.txt)
%SUMMARY INTF: flow integrator
%DESCRIPTION Port [in]: Flow to be integrated
%DESCRIPTION Port [out]: Effort = integral of flow on port [in]


% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %% Version control history
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %% $Id: INTF_lbl.txt,v 1.3 1998/07/16 07:35:10 peterg Exp $
% %% $Log: INTF_lbl.txt,v $
% %% Revision 1.3  1998/07/16 07:35:10  peterg
% %% Aliased version
% %%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% Port aliases
%ALIAS in Flow
%ALIAS out Integrated_flow

% Argument aliases

%% Each line should be of one of the following forms:
%       a comment (ie starting with %)
%       component-name cr_name arg1,arg2,..argn
%       blank

% ---- Component labels ----

% Component type SS
[Flow] SS external,external
[Integrated_flow] SS external,external
```

Subsystems

No subsystems.

2.1.5 ROD

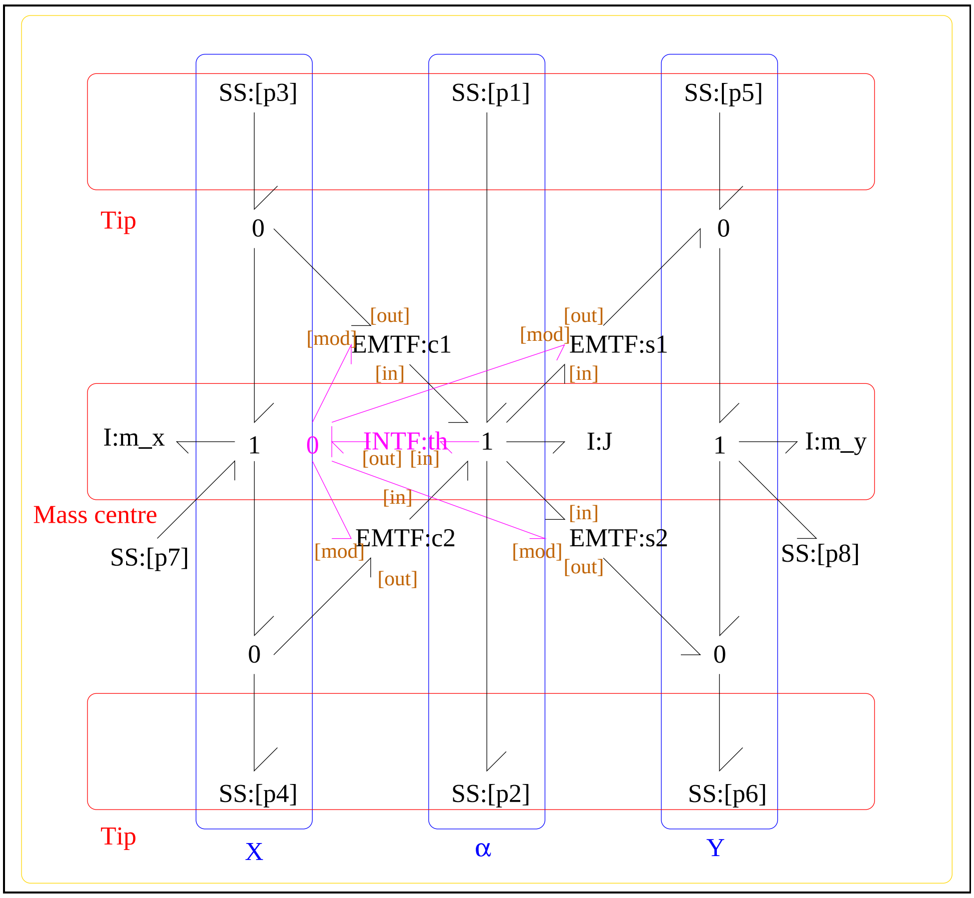

Figure 2.4: System **ROD**: acausal bond graph

The acausal bond graph of system **ROD** is displayed in Figure 2.4 (on page 31) and its label file is listed in Section 2.1.5 (on page 31). The subsystems are listed in Section 2.1.5 (on page 34).
This is a special version just for this problem.

Summary information

System ROD::rigid rod in two dimensions Port [1]: Angular torque/velocity - end 1 Port [2]: Angular torque/velocity - end 2 Port [3]: x force/velocity - end 1 Port [4]: x force/velocity - end 2 Port [5]: y force/velocity - end 1 Port [6]: y force/velocity - end 2 Port [7]: x force/velocity - centre Port [8]: y force/velocity - centre

Parameter 1: length from end 1 to mass centre Parameter 2: length from end 2 to mass centre Parameter 3: inertia about mass centre Parameter 4: mass See Section 10.2 of ”Metamodelling”

Interface information:

This component has no ALIAS declarations

Variable declarations:

This component has no PAR declarations

Units declarations:

This component has no UNITs declarations

The label file: ROD_lbl.txt

```
%SUMMARY ROD: rigid rod in two dimensions
%DESCRIPTION Port [1]: Angular torque/velocity - end 1
%DESCRIPTION Port [2]: Angular torque/velocity - end 2
%DESCRIPTION Port [3]: x force/velocity - end 1
%DESCRIPTION Port [4]: x force/velocity - end 2
%DESCRIPTION Port [5]: y force/velocity - end 1
%DESCRIPTION Port [6]: y force/velocity - end 2
%DESCRIPTION Port [7]: x force/velocity - centre
%DESCRIPTION Port [8]: y force/velocity - centre
%DESCRIPTION
%DESCRIPTION Parameter 1: length from end 1 to mass centre
%DESCRIPTION Parameter 2: length from end 2 to mass centre
%DESCRIPTION Parameter 3: inertia about mass centre
%DESCRIPTION Parameter 4: mass
%DESCRIPTION See Section 10.2 of "Metamodelling"


%% Label file for system ROD (ROD_lbl.txt)



% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %% Version control history
```

```
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %% $Id: ROD_lbl.txt,v 1.2 2000/12/28 17:45:59 peterg Exp $
% %% $Log: ROD_lbl.txt,v $
% %% Revision 1.2  2000/12/28 17:45:59  peterg
% %% To RCS
% %%
% Revision 1.1  1996/11/07  10:57:17  peterg
% Initial revision
%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Each line should be of one of the following forms:
% a comment (ie starting with %)
% Component-name CR_name arg1,arg2,..argn
% blank

%Inertias
J lin flow,$3
m_x lin flow,$4
m_y lin flow,$4

%Integrate angular velocity to get angle
th

%Modulated transformers
s1 lsin flow,$1
s2 lsin flow,$2
c1 lcos flow,$1
c2 lcos flow,$2

% ports

[p1] SS external,external
[p2] SS external,external
[p3] SS external,external
[p4] SS external,external
[p5] SS external,external
[p6] SS external,external
[p7] SS external,external
[p8] SS external,external
```

Subsystems

- INTF: flow integrator (1) No subsystems.

2.2 BouncingRod_struc.tex

MTT command:

```
mtt BouncingRod struc tex
```

List of inputs for system BouncingRod			
	Component	System	Repetition
1	gravity	BouncingRod__gravity	1

List of outputs for system BouncingRod			
	Component	System	Repetition
1	hL	BouncingRod__hL	1
2	hR	BouncingRod__hR	1

List of states for system BouncingRod			
	Component	System	Repetition
1	MTT_SWITCH	BouncingRod__groundL__MTT_SWITCH	1
2	MTT_SWITCH	BouncingRod__groundR__MTT_SWITCH	1
3	mttC	BouncingRod__intfL__mttC	1
4	mttC	BouncingRod__intfR__mttC	1
5	J	BouncingRod__rod__J	1
6	m_x	BouncingRod__rod__m_x	1
7	m_y	BouncingRod__rod__m_y	1
8	mttC	BouncingRod__rod__th__mttC	1

2.3 BouncingRod_ode.tex

MTT command:

```
mtt BouncingRod ode tex
```

$$
\begin{aligned}
\dot{x}_1 &= \frac{(\sin(x_8) l m x_5 - j x_7)}{(jm)} \\
\dot{x}_2 &= \frac{(\sin(x_8) l m x_5 + j x_7)}{(jm)} \\
\dot{x}_3 &= \frac{(-\sin(x_8) l m x_5 + j x_7)}{(jm)} \\
\dot{x}_4 &= \frac{(\sin(x_8) l m x_5 + j x_7)}{(jm)} \\
\dot{x}_5 &= \frac{(-\sin(x_8) l (x_1 + x_2))}{\varepsilon} \\
\dot{x}_6 &= 0 \\
\dot{x}_7 &= \frac{(-\varepsilon u_1 + x_1 - x_2)}{\varepsilon} \\
\dot{x}_8 &= \frac{x_5}{j}
\end{aligned} \tag{2.1}
$$

$$
\begin{aligned}
y_1 &= x_3 \\
y_2 &= x_4
\end{aligned} \tag{2.2}
$$

2.4 BouncingRod_numpar.txt

MTT command:

```
mtt BouncingRod numpar txt
```

```
# Numerical parameter file (BouncingRod_numpar.txt)
# Generated by MTT at Fri Jul  4 14:09:11 BST 1997

# %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
# %% Version control history
# %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
# %% $Id: BouncingRod_numpar.txt,v 1.3 2001/10/05 11:24:34 gawthrop Exp
# %% $Log: BouncingRod_numpar.txt,v $
# %% Revision 1.3  2001/10/05 11:24:34  gawthrop
# %% Updated for new mtt features - input.txt etc
# %%
# %% Revision 1.2  2000/12/28 17:45:59  peterg
# %% To RCS
```

```
# %%
# %% Revision 1.1  1997/07/06 16:17:27  peterg
# %% Initial revision
# %%
# %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

# Parameters
epsilon =  0.01;
l =  1.0; # Default value
m =  1.0; # Default value
j =  l*l*m/3;
```

2.5 BouncingRod_input.txt

MTT command:

```
mtt BouncingRod input txt
```

```
# Numerical parameter file (BouncingRod_input.txt)
# Generated by MTT at Fri Jul  4 14:09:14 BST 1997

# %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
# %% Version control history
# %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
# %% $Id: BouncingRod_input.txt,v 1.3 2003/06/11 15:58:51 gawthrop
# %% $Log: BouncingRod_input.txt,v $
# %% Revision 1.3  2003/06/11 15:58:51  gawthrop
# %% Updated examples for latest MTT.
# %%
# %% Revision 1.2  2000/12/04 08:27:42  peterg
# %% New version
# %%
# %% Revision 1.1  1997/07/06 16:17:48  peterg
# %% Initial revision
# %%
# %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
# Set the inputs
## Removed by MTT on Mon Dec  4 08:27:04 GMT 2000: u(1) =
9.81; # gravity (Default value)
## Removed by MTT on Tue Jun 10 10:33:30 BST 2003: bouncingrod_gravity
= 9.81; # Added by MTT on Mon Dec 04 08:27:10 GMT 2000
bouncingrod__gravity = 9.81; # Gravity
```

2.6 BouncingRod_logic.txt

MTT command:

```
mtt BouncingRod logic txt
```

```
# -*-octave-*- Put Emacs into octave-mode
# Simulation parameters for system BouncingRod (BouncingRod_logic.txt)
# Generated by MTT on Thu May 18 12:14:35 BST 2000.
###############################################################
## Version control history
###############################################################
## $Id: BouncingRod_logic.txt,v 1.3 2003/06/11 15:59:04 gawthrop Exp $
## $Log: BouncingRod_logic.txt,v $
## Revision 1.3  2003/06/11 15:59:04  gawthrop
## Updated examples for latest MTT.
##
## Revision 1.2  2002/09/29 15:31:39  geraint
## Updated names.
##
## Revision 1.1  2000/12/28 17:45:59  peterg
## To RCS
##
###############################################################

# Set the switches
## Removed by MTT on Tue Jun 10 10:45:53 BST 2003: bouncingrod_groundl_m
= (bouncingrod_intfl_mttc <= 0);
## Removed by MTT on Tue Jun 10 10:45:53 BST 2003: bouncingrod_groundr_m
= (bouncingrod_intfr_mttc <= 0);
bouncingrod__groundl__mtt_switch_logic = (bouncingrod__intfl__mttc <= 0)
bouncingrod__groundr__mtt_switch_logic = (bouncingrod__intfr__mttc <= 0)
```

2.7 BouncingRod_odeso.ps

MTT command:

```
mtt BouncingRod odeso ps
```

This representation is given as Figure 2.5 (on page 38).

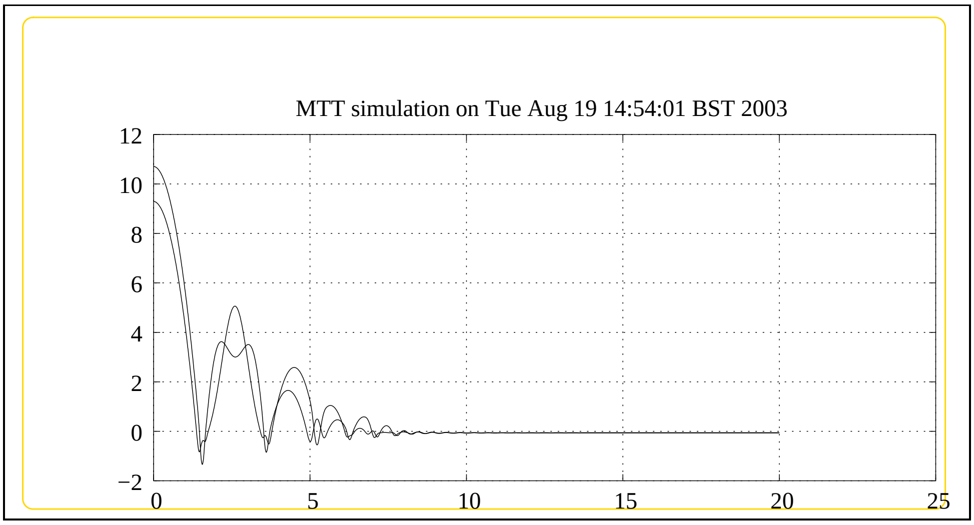

Figure 2.5: System **BouncingRod**, representation odeso (-noargs)

Chapter 3

Clutch

3.1 Clutch_abg.tex

MTT command:

```
mtt Clutch abg tex
```

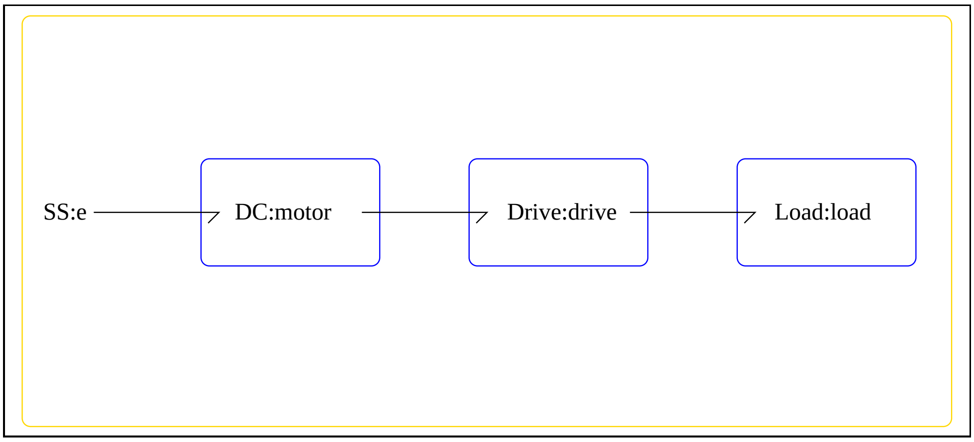

Figure 3.1: System **Clutch**: acausal bond graph

The acausal bond graph of system **Clutch** is displayed in Figure 3.1 (on page 39) and its label file is listed in Section 3.1.1 (on page 40). The subsystems are listed in Section 3.1.2 (on page 41). The details of the DC motor and the load are hidden behind a word bond graph so as to focus on the clutch mechanism as modelled by the **CSW** component. It is natural to model the clutch shaft by a compliance to absorb the shock of engaging the clutch; therefore the drive model contains a **CSW** component.

Figure 3.6, shows the angular velocities of the motor and load. The clutch is engaged from time t given by $0 < t < 15$; and a unit input voltage is applied at time $t = 0$. All initial conditions are zero.

3.1.1 Summary information

System Clutch::Simple model of a clutch system (Hybrid) Uses CSW component to model the clutch/flexi shaft

Interface information:

This component has no ALIAS declarations

Variable declarations:

This component has no PAR declarations

Units declarations:

This component has no UNITs declarations

The label file: Clutch_lbl.txt

```
%SUMMARY Clutch: Simple model of a clutch system (Hybrid)
%DESCRIPTION Uses CSW component to model the clutch/flexi shaft
%% Label file for system Clutch (Clutch_lbl.txt)

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %% Version control history
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %% $Id: Clutch_lbl.txt,v 1.2 2000/05/18 10:32:04 peterg Exp $
% %% $Log: Clutch_lbl.txt,v $
% %% Revision 1.2  2000/05/18 10:32:04  peterg
% %% New SS form
% %%
% %% Revision 1.1  1997/09/11 09:38:48  peterg
% %% Initial revision
% %%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Each line should be of one of the following forms:
```

```
% a comment (ie starting with %)
% Component-name CR_name arg1,arg2,..argn
% blank



% Component type DC
motor lin k_m;l_a;r_a;j_m;b_m

% Component type Drive
drive

% Component type Load
load

% Component type SS
e SS external,internal
```

3.1.2 Subsystems

- DC: DC motor (or generator) (1) No subsystems.
- Drive: The clutch/shaft drive subsystem (1)
  - CSW: Switched C component (1)
- Load: An inertial/frictional load (1) No subsystems.

3.1.3 CSW

The acausal bond graph of system **CSW** is displayed in Figure 3.2 (on page 42) and its label file is listed in Section 3.1.3 (on page 41). The subsystems are listed in Section 3.1.3 (on page 44).

Summary information

System CSW::Switched C component CSW acts as an C component except when the -s -c option is used. When the -s -c option is used: The component label is used as a variable name (eg Name) In the ode simulation: Name = 1 implies normal C (closed switch) Name = 0 implies state=0 (open switch) Name = -1 implies state¿0 (Diode)

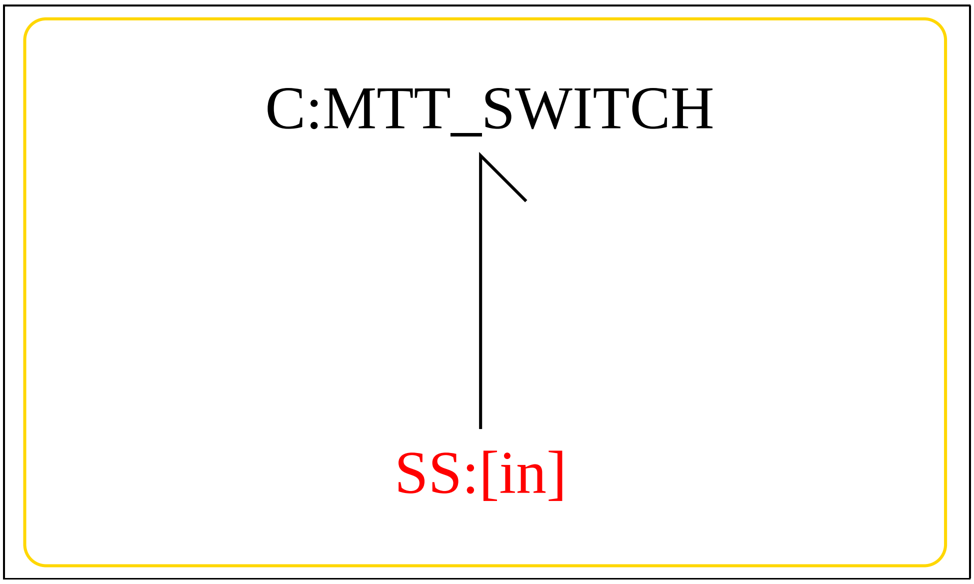

Figure 3.2: System **CSW**: acausal bond graph

Interface information:

Parameter $1 represents actual parameter **effort,c_s**

Parameter $1 represents actual parameter **lin**

Port out represents actual port **in**

Variable declarations:

This component has no PAR declarations

Units declarations:

This component has no UNITs declarations

The label file: CSW_lbl.txt

```
%SUMMARY CSW: Switched C component
%DESCRIPTION CSW acts as an C component except when the -s -c opti
%DESCRIPTION When the -s -c option is used:
%DESCRIPTION   The component label is used as a variable name (eg
%DESCRIPTION   In the ode simulation:
```

```
%DESCRIPTION     Name = 1 implies normal C (closed switch)
%DESCRIPTION     Name = 0 implies state=0 (open switch)
%DESCRIPTION     Name = -1 implies state>0 (Diode)

%ALIAS out in
%ALIAS $1 lin
%ALIAS $1 effort,c_s

%% Label file for system CSW (CSW_lbl.txt)

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %% Version control history
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %% $Id: CSW_lbl.txt,v 1.3 1998/07/26 13:30:33 peterg Exp $
% %% $Log: CSW_lbl.txt,v $
% %% Revision 1.3  1998/07/26 13:30:33  peterg
% %% Added aliases
% %%
% %% Revision 1.2  1997/09/11 09:00:52  peterg
% %% More documentation.
% %%
% %% Revision 1.1  1997/06/16  10:55:20  peterg
% %% Initial revision
% %%
% %% Revision 1.1  1997/06/03 15:26:09  peterg
% %% Initial revision
% %%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Each line should be of one of the following forms:
% a comment (ie starting with %)
% Component-name CR_name arg1,arg2,..argn
% blank



% Component type C
MTT_SWITCH lin effort,c_s

% Component type SS
[in] SS external,external
```

Subsystems

No subsystems.

3.1.4 DC

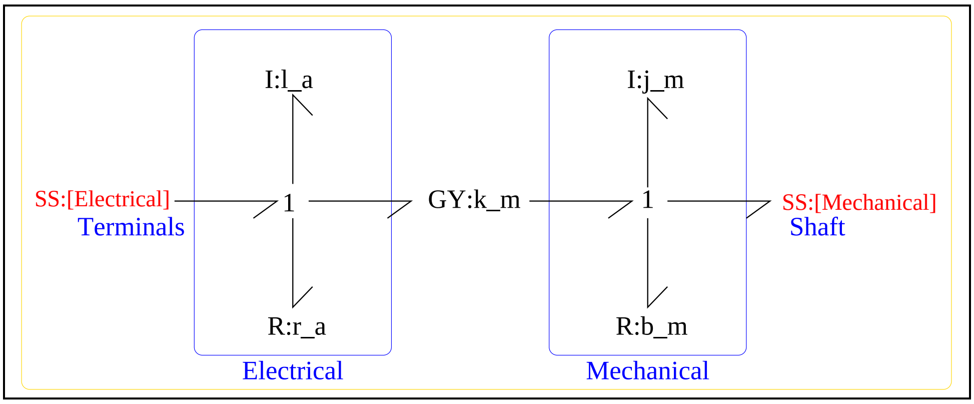

Figure 3.3: System **DC**: acausal bond graph

The acausal bond graph of system **DC** is displayed in Figure 3.3 (on page 44) and its label file is listed in Section 3.1.4 (on page 44). The subsystems are listed in Section 3.1.4 (on page 47).

Index	Parameter
1	Motor gain (k_m)
2	Armature inductance (l_a)
3	Armature resistance (r_a)
4	Inertia (j_m)
5	Friction coefficient (b_m)

Table 3.1: DC motor parameters

DC is a two-port component representing a DC motor. It has the 5 parameters listed in Table 3.1 (on page 44).

Summary information

System DC::DC motor (or generator)

Interface information:

Parameter $1 represents actual parameter **k_m**

Parameter $2 represents actual parameter **l_a**

Parameter $3 represents actual parameter **r_a**

Parameter $4 represents actual parameter **j_m**

Parameter $5 represents actual parameter **b_m**

Port in represents actual port **Electrical**

Port out represents actual port **Mechanical**

Variable declarations:

This component has no PAR declarations

Units declarations:

Port Electrical has domain electrical

- **Effort units** volt
- **Flow units** amp

Port Mechanical has domain rotational

- **Effort units** N*m
- **Flow units** radians/s

The label file: DC_lbl.txt

```
%SUMMARY DC: DC motor (or generator)

%% Port Alias
%ALIAS in Electrical
%ALIAS out Mechanical

%% Unit definition
%UNITS Electrical electrical volt amp
%UNITS Mechanical rotational N*m radians/s
```

```
%ALIAS $1 k_m
%ALIAS $2 l_a
%ALIAS $3 r_a
%ALIAS $4 j_m
%ALIAS $5 b_m


%% Label file for system DC (DC_lbl.txt)

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %% Version control history
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %% $Id: DC_lbl.txt,v 1.7 2001/02/05 03:07:10 geraint Exp $
% %% $Log: DC_lbl.txt,v $
% %% Revision 1.7  2001/02/05 03:07:10  geraint
% %% angular displacement units: changed rads to radians
% %%
% %% Revision 1.6  2000/11/16 09:45:51  peterg
% %% Added unit definitions
% %%
% %% Revision 1.5  1998/07/26 12:49:24  peterg
% %% Corrected some errors
% %%
% %% Revision 1.4  1998/07/26 12:45:33  peterg
% %% Added ports
% %%
% %% Revision 1.3  1998/07/22 12:01:17  peterg
% %% Aliased ports and parameters.
% %%
% %% Revision 1.2  1996/12/04  16:01:42  peterg
% %% Documantation added.
% %%
% %% Revision 1.1  1996/12/04  16:00:56  peterg
% %% Initial revision
% %%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Each line should be of one of the following forms:
% a comment (ie starting with %)
% Component-name CR_name arg1,arg2,..argn
% blank
```

```
%Motor gain
k_m lin flow,k_m

% Electrical components
%Inductance
l_a lin effort,l_a

%Resistance
r_a lin flow,r_a

% Mechanical components
%Inertia
j_m lin flow,j_m

%Friction
b_m lin flow,b_m

% Ports
[Electrical] SS external,external
[Mechanical] SS external,external
```

Subsystems

No subsystems.

3.1.5 Drive

The acausal bond graph of system **Drive** is displayed in Figure 3.4 (on page 48) and its label file is listed in Section 3.1.5 (on page 47). The subsystems are listed in Section 3.1.5 (on page 49).

Summary information

System Drive::The clutch/shaft drive subsystem ¡Detailed description here¿

Interface information:

This component has no ALIAS declarations

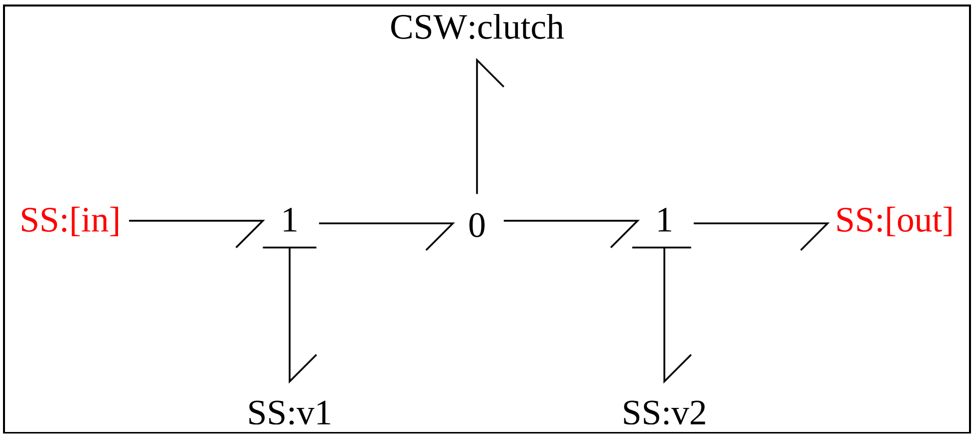

Figure 3.4: System **Drive**: acausal bond graph

Variable declarations:

This component has no PAR declarations

Units declarations:

This component has no UNITs declarations

The label file: Drive_lbl.txt

```
%SUMMARY Drive: The clutch/shaft drive subsystem
%DESCRIPTION <Detailed description here>
%% Label file for system Drive (Drive_lbl.txt)

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %% Version control history
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %% $Id: Drive_lbl.txt,v 1.3 2002/09/29 13:46:46 geraint Exp $
% %% $Log: Drive_lbl.txt,v $
% %% Revision 1.3  2002/09/29 13:46:46  geraint
% %% Added missing SS entries.
% %%
% %% Revision 1.2  2000/05/18 10:32:32  peterg
% %% New SS form
% %%
% %% Revision 1.1  1997/09/11 09:39:16  peterg
% %% Initial revision
```

```
% %%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Each line should be of one of the following forms:
% a comment (ie starting with %)
% Component-name CR_name arg1,arg2,..argn
% blank



% Component type CSW
clutch lin effort,epsilon

% Component type R
% r_d lin flow,r_d

% Component type SS
v1 SS 0,external
v2 SS 0,external
[in] SS external,external
[out] SS external,external
```

Subsystems

- CSW: Switched C component (1) No subsystems.

3.1.6 Load

The acausal bond graph of system **Load** is displayed in Figure 3.5 (on page 50) and its label file is listed in Section 3.1.6 (on page 49). The subsystems are listed in Section 3.1.6 (on page 52).

Summary information

System Load::An inertial/frictional load ¡Detailed description here¿

Interface information:

This component has no ALIAS declarations

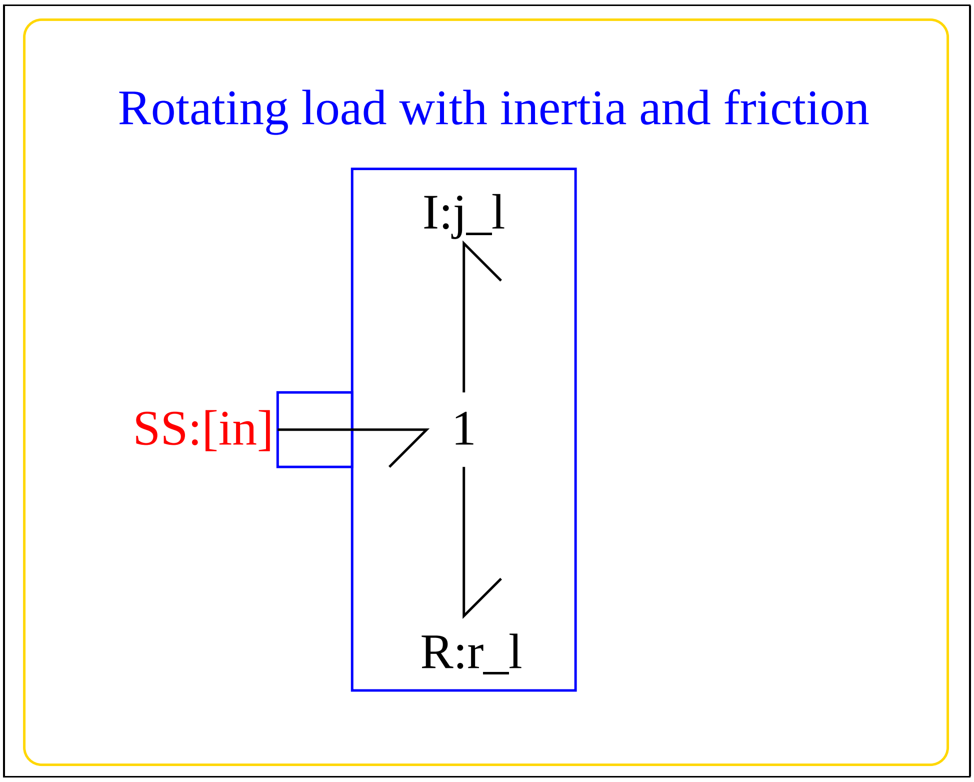

Figure 3.5: System **Load**: acausal bond graph

Variable declarations:

This component has no PAR declarations

Units declarations:

This component has no UNITs declarations

The label file: Load_lbl.txt

```
%SUMMARY Load: An inertial/frictional load
%DESCRIPTION <Detailed description here>
%% Label file for system Load (Load_lbl.txt)

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %% Version control history
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %% $Id: Load_lbl.txt,v 1.2 2002/09/29 13:46:46 geraint Exp $
% %% $Log: Load_lbl.txt,v $
% %% Revision 1.2  2002/09/29 13:46:46  geraint
% %% Added missing SS entries.
% %%
% %% Revision 1.1  1997/09/11 09:39:29  peterg
% %% Initial revision
% %%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Each line should be of one of the following forms:
% a comment (ie starting with %)
% Component-name CR_name arg1,arg2,..argn
% blank



% Component type I
j_l lin flow,j_l

% Component type R
r_l lin flow,r_l

% Component type SS
[in] SS external,external
```

Subsystems

No subsystems.

3.2 Clutch_struc.tex

MTT command:

```
mtt Clutch struc tex
```

List of inputs for system Clutch			
	Component	System	Repetition
1	e	Clutch__e	1

List of outputs for system Clutch			
	Component	System	Repetition
1	v1	Clutch__drive__v1	1
2	v2	Clutch__drive__v2	1

List of states for system Clutch			
	Component	System	Repetition
1	l_a	Clutch__motor__l_a	1
2	j_m	Clutch__motor__j_m	1
3	MTT_SWITCH	Clutch__drive__clutch__MTT_SWITCH	1
4	j_l	Clutch__load__j_l	1

3.3 Clutch_ode.tex

MTT command:

```
mtt Clutch ode tex
```

$$\begin{aligned}
\dot{x}_1 &= \frac{(j_m l_a u_1 - j_m x_1 r_a - k_m l_a x_2)}{(j_m l_a)} \\
\dot{x}_2 &= \frac{(-b_m \varepsilon l_a x_2 + \varepsilon j_m k_m x_1 - j_m l_a x_3)}{(\varepsilon j_m l_a)} \\
\dot{x}_3 &= \frac{(j_l x_2 - j_m x_4)}{(j_l j_m)} \\
\dot{x}_4 &= \frac{(-\varepsilon x_4 r_l + j_l x_3)}{(\varepsilon j_l)}
\end{aligned} \tag{3.1}$$

$$y_1 = \frac{x_2}{j_m}$$
$$y_2 = \frac{x_4}{j_l} \tag{3.2}$$

3.4 Clutch_dm.tex

MTT command:

```
mtt Clutch dm tex
```

$$A = \begin{pmatrix} \frac{(-r_a)}{l_a} & \frac{(-k_m)}{j_m} & 0 & 0 \\ \frac{k_m}{l_a} & \frac{(-b_m)}{j_m} & \frac{(-1)}{\varepsilon} & 0 \\ 0 & \frac{1}{j_m} & 0 & \frac{(-1)}{j_l} \\ 0 & 0 & \frac{1}{\varepsilon} & \frac{(-r_l)}{j_l} \end{pmatrix} \tag{3.3}$$

$$B = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{3.4}$$

$$C = \begin{pmatrix} 0 & \frac{1}{j_m} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{j_l} \end{pmatrix} \tag{3.5}$$

$$D = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{3.6}$$

3.5 Clutch_simpar.txt

MTT command:

```
mtt Clutch simpar txt
```

```
# Simulation parameters for system Clutch (Clutch_simpar.txt)
# Generated by MTT on Sun Jul 26 13:26:44 BST 1998.
###############################################################
## Version control history
###############################################################
## $Id: Clutch_simpar.txt,v 1.2 2001/10/05 11:24:35 gawthrop Exp $
## $Log: Clutch_simpar.txt,v $
## Revision 1.2  2001/10/05 11:24:35  gawthrop
## Updated for new mtt features - input.txt etc
##
## Revision 1.1  1999/12/21 09:13:19  peterg
## Initial revision
##
###############################################################

LAST=20.0;
DT=0.1;
STEPFACTOR=1;
```

3.6 Clutch_input.txt

MTT command:

```
mtt Clutch input txt
```

```
# Numerical parameter file (Clutch_input.txt)
# Generated by MTT at Thu Jul  3 12:17:46 BST 1997

# %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
# %% Version control history
# %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
# %% $Id: Clutch_input.txt,v 1.3 2003/04/12 19:23:15 geraint Exp $
# %% $Log: Clutch_input.txt,v $
# %% Revision 1.3  2003/04/12 19:23:15  geraint
# %% Updated names with double underscore.
# %%
# %% Revision 1.2  2002/09/29 13:47:35  geraint
# %% Updated names.
# %%
```

```
# %% Revision 1.1  2000/05/19 07:10:03  peterg
# %% Initial revision
# %%
# %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

# Set the inputs
clutch__e = 1.0; # e (Default value)
```

3.7 Clutch_logic.txt

MTT command:

```
mtt Clutch logic txt
```

```
# -*-octave-*- Put Emacs into octave-mode
# Simulation parameters for system Clutch (Clutch_logic.txt)
# Generated by MTT on Thu May 18 12:29:15 BST 2000.
###############################################################
## Version control history
###############################################################
## $Id: Clutch_logic.txt,v 1.4 2003/04/12 19:23:15 geraint Exp $
## $Log: Clutch_logic.txt,v $
## Revision 1.4  2003/04/12 19:23:15  geraint
## Updated names with double underscore.
##
## Revision 1.3  2002/09/29 13:47:35  geraint
## Updated names.
##
## Revision 1.2  2001/10/05 11:24:35  gawthrop
## Updated for new mtt features - input.txt etc
##
## Revision 1.1  2000/05/19 07:11:36  peterg
## Initial revision
##
## Revision 1.1  2000/05/18 11:29:58  peterg
## Initial revision
##
###############################################################

# Set the switches
```

```
clutch__drive__clutch__mtt_switch_logic = ((t >= 5)&&(t < 15)); #
```

3.8 Clutch_numpar.txt

MTT command:

```
mtt Clutch numpar txt
```

```
# Numerical parameter file (Clutch_numpar.txt)
# Generated by MTT at Thu Jul  3 12:12:35 BST 1997

# %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
# %% Version control history
# %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
# %% $Id: Clutch_numpar.txt,v 1.1 2000/12/28 17:46:44 peterg Exp $
# %% $Log: Clutch_numpar.txt,v $
# %% Revision 1.1  2000/12/28 17:46:44  peterg
# %% To RCS
# %%
# %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

# Parameters
b_m =  0.1; # Default value
epsilon =  0.01;
j_l =  1.0; # Default value
j_m =  10.0; # Default value
k_m =  1.0; # Default value
l_a =  0.1; # Default value
r_a =  1.0; # Default value
r_l =  5.0; # Default value
```

3.9 Clutch_odeso.ps

MTT command:

```
mtt Clutch odeso ps
```

This representation is given as Figure 3.6 (on page 57).

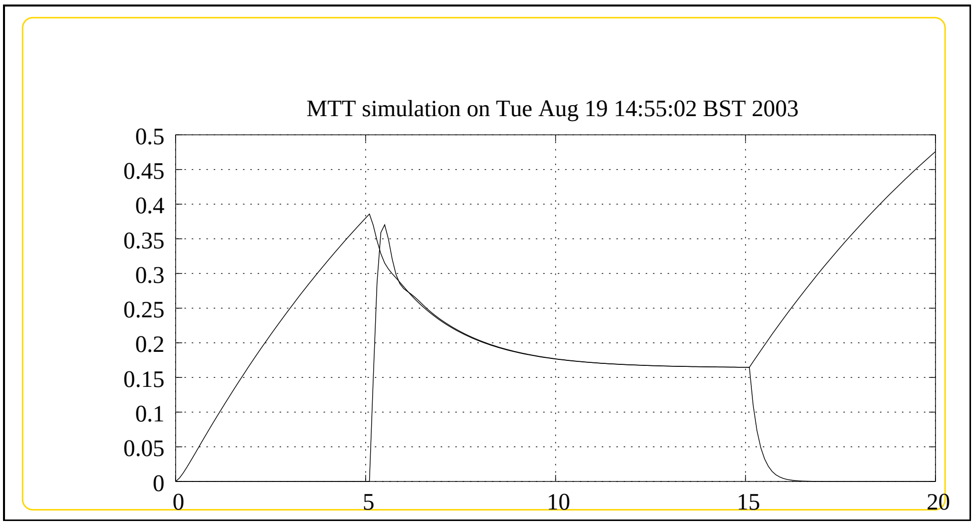

Figure 3.6: System **Clutch**, representation odeso (-noargs)

3.10 Clutch_rep.txt

MTT command:

```
mtt Clutch rep txt
```

```
# Outline report file for system Clutch (Clutch_rep.txt)

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %% Version control history
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %% $Id: Clutch_rep.txt,v 1.3 2001/10/05 11:24:35 gawthrop Exp $
% %% $Log: Clutch_rep.txt,v $
% %% Revision 1.3  2001/10/05 11:24:35  gawthrop
% %% Updated for new mtt features - input.txt etc
% %%
% %% Revision 1.2  2000/05/19 07:16:07  peterg
% %% Added logic
% %%
% %% Revision 1.1  1999/02/21 08:18:25  peterg
% %% Initial revision
% %%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
mtt  Clutch abg tex
mtt  Clutch struc tex
mtt  Clutch ode tex
mtt  Clutch dm tex

mtt  Clutch simpar txt
mtt  Clutch input txt
mtt  Clutch logic txt
mtt  Clutch numpar txt

mtt  Clutch odeso ps

mtt  Clutch rep txt
```

Chapter 4

Weirs

4.1 Weirs_abg.tex

MTT command:

```
mtt Weirs abg tex
```

The acausal bond graph of system **Weirs** is displayed in Figure 4.1 (on page 60) and its label file is listed in Section 4.1.1 (on page 61). The subsystems are listed in Section 4.1.2 (on page 62).

Aircraft fuel tanks are often fitted with baffles to reduce fuel slosh. A simple model relating to such a system is shown in Figure 4.1 (on page 60) which corresponds to a single tank containing two dividing weirs. Liquid with flow rate f enters the left-hand compartment; liquid leaks out of the centre compartment at a flow rate determined by gravity and the properties of the corresponding orifice. The Bond Graph appearing in Figure 4.1 (on page 60) represents each of the three compartments by a **C** component (labelled tank1 to tank3), the corresponding pressures are measured by the **SS** elements p1–p3. The leak is represented by the **R** component labelled leak. The flows over the two weirs are represented by the four **ISW** components; each weir has a separate **ISW** component for each flow direction. Each **ISW** component is switched by the appropriate level.

The system was simulated for 20 time units and the resultant level of each tank partition is plotted in Figure 4.4 (on page 73). Each partition has unit cross section, and the two weir heights are 1 and 2 respectively; the inflow f is given by:

$$f = \begin{cases} 1 & \text{if } t \leq 10 \\ 0 & \text{if } t > 10 \end{cases} \tag{4.1}$$

and the leak resistance is linear with flow resistance 5.

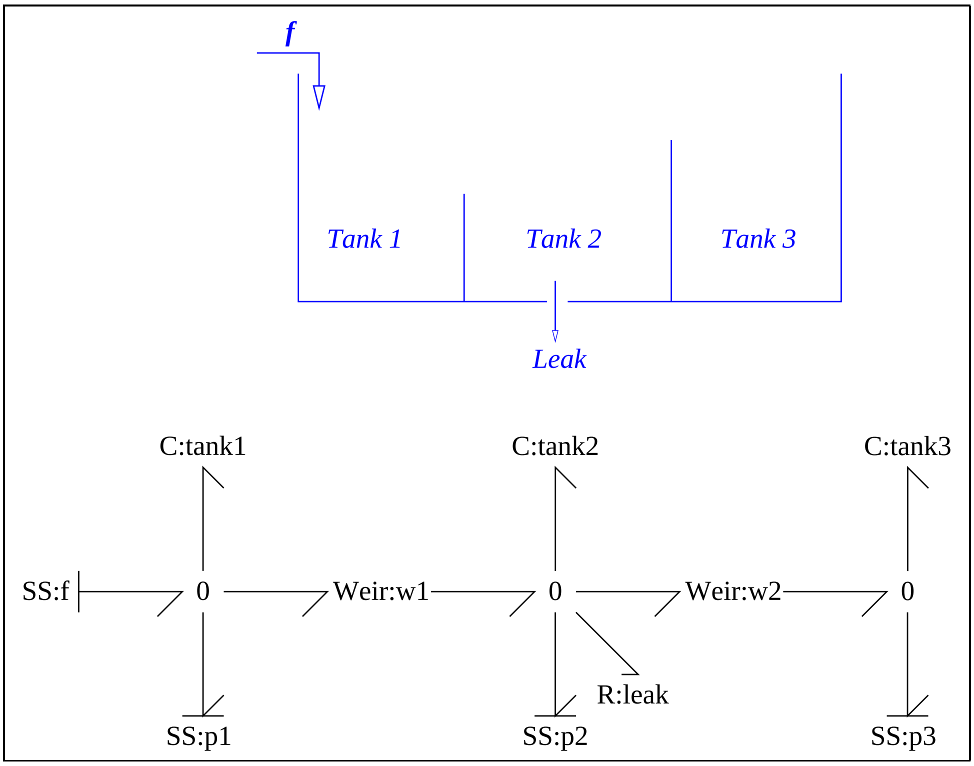

Figure 4.1: System **Weirs**: acausal bond graph

4.1.1 Summary information

System Weirs:Double-weir tanks system (hybrid) ¡Detailed description here¿

Interface information:

Parameter $1 represents actual parameter **tank1**

Parameter $2 represents actual parameter **tank2**

Parameter $3 represents actual parameter **tank3**

Parameter $4 represents actual parameter **leak**

Variable declarations:

This component has no PAR declarations

Units declarations:

This component has no UNITs declarations

The label file: Weirs_lbl.txt

```
%% Label file for system Weirs (Weirs_lbl.txt)
%SUMMARY Weirs Double-weir tanks system (hybrid)
%DESCRIPTION <Detailed description here>

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %% Version control history
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %% $Id: Weirs_lbl.txt,v 1.2 2001/10/05 11:24:35 gawthrop Exp $
% %% $Log: Weirs_lbl.txt,v $
% %% Revision 1.2  2001/10/05 11:24:35  gawthrop
% %% Updated for new mtt features - input.txt etc
% %%
% %% Revision 1.1  1999/03/09 00:12:12  peterg
% %% Initial revision
% %%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Port aliases

% Argument aliases
%ALIAS $1 tank1
%ALIAS $2 tank2
%ALIAS $3 tank3
%ALIAS $4 leak

%% Each line should be of one of the following forms:
%       a comment (ie starting with %)
%       component-name cr_name arg1,arg2,..argn
%       blank

% ---- Component labels ----

% Component type C
tank1 lin effort,c_1
tank2 lin effort,c_2
tank3 lin effort,c_3

% Component type R
leak lin flow,r_l

% Component type SS
f SS internal,external
p1 SS external,0
p2 SS external,0
p3 SS external,0

% Component type Weir
w1      none epsilon_i;epsilon_r
w2      none epsilon_i;epsilon_r
```

4.1.2 Subsystems

- (2)

  - ISW: Switched I component (1)

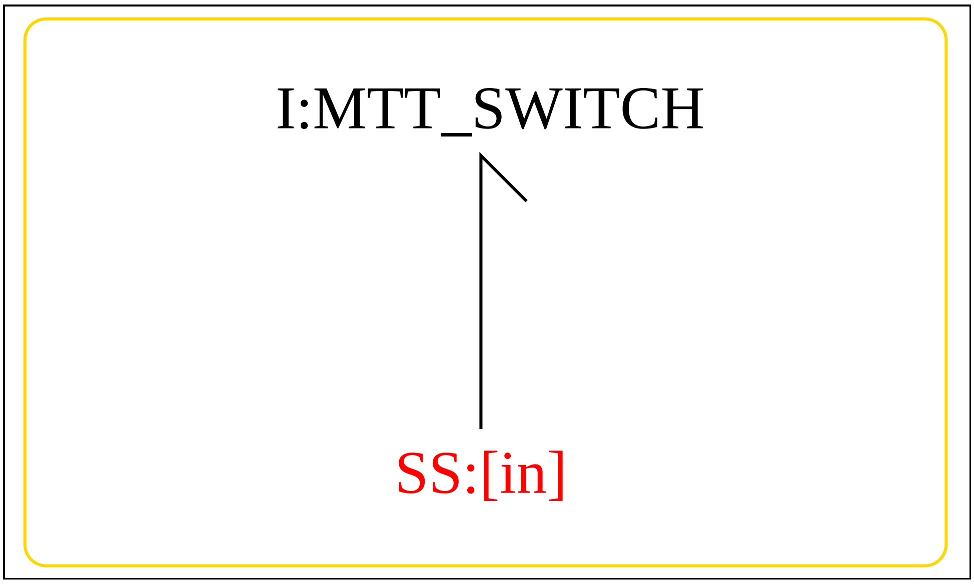

Figure 4.2: System **ISW**: acausal bond graph

4.1.3 ISW

The acausal bond graph of system **ISW** is displayed in Figure 4.2 (on page 63) and its label file is listed in Section 4.1.3 (on page 63). The subsystems are listed in Section 4.1.3 (on page 65).

Summary information

System ISW::Switched I component ISW acts as an I component except when the -s -c option is used. When the -s -c option is used: The component label is used as a variable name (eg Name) In the ode simulation: Name = 1 implies normal I (closed switch) Name = 0 implies state=0 (open switch) Name = -1 implies state¿0 (Diode)

Interface information:

Parameter $1 represents actual parameter **flow,i_s**

Parameter $1 represents actual parameter **lin**

Port out represents actual port **in**

Variable declarations:

This component has no PAR declarations

Units declarations:

This component has no UNITs declarations

The label file: ISW_lbl.txt

```
%SUMMARY ISW: Switched I component
%DESCRIPTION ISW acts as an I component except when the -s -c opti
%DESCRIPTION When the -s -c option is used:
%DESCRIPTION   The component label is used as a variable name (eg
%DESCRIPTION   In the ode simulation:
%DESCRIPTION     Name = 1 implies normal I (closed switch)
%DESCRIPTION     Name = 0 implies state=0 (open switch)
%DESCRIPTION     Name = -1 implies state>0 (Diode)

%ALIAS out in
%ALIAS $1 lin
%ALIAS $1 flow,i_s



%% Label file for system ISW (ISW_lbl.txt)

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %% Version control history
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %% $Id: ISW_lbl.txt,v 1.2 1998/07/26 13:00:47 peterg Exp $
% %% $Log: ISW_lbl.txt,v $
% %% Revision 1.2  1998/07/26 13:00:47  peterg
% %% Added alaises
% %%
% %% Revision 1.1  1998/07/26 12:57:00  peterg
% %% Initial revision
% %%
% %% Revision 1.1  1997/06/03 15:26:09  peterg
% %% Initial revision
% %%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%% Each line should be of one of the following forms:
% a comment (ie starting with %)
% Component-name CR_name arg1,arg2,..argn
% blank



% Component type I
MTT_SWITCH lin flow,i_s

% Component type SS
[in] SS external,external
```

Subsystems

No subsystems.

4.1.4 Weir

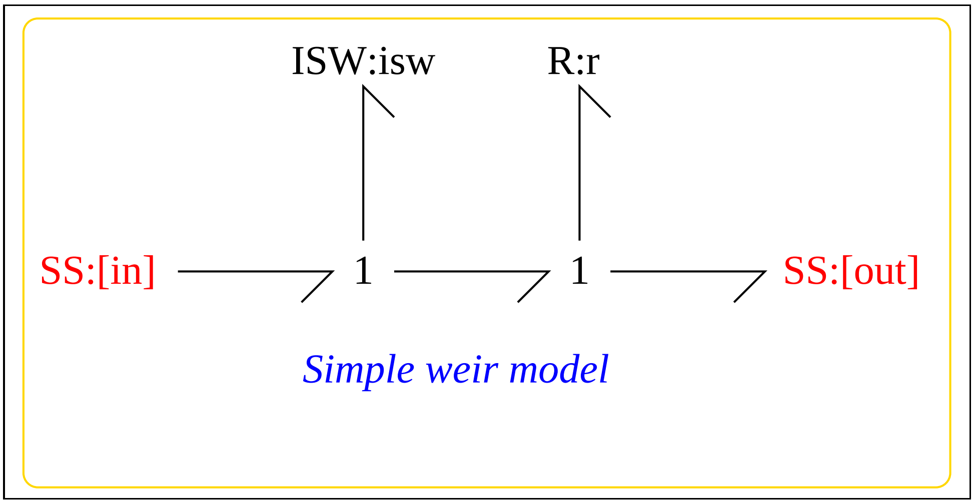

Figure 4.3: System **Weir**: acausal bond graph

The acausal bond graph of system **Weir** is displayed in Figure 4.3 (on page 65) and its label file is listed in Section 4.1.4 (on page 66). The subsystems are listed in Section 4.1.4 (on page 68).

The weir is modelled by an **ISW** component in series with an **R** component. Physicaly, the former represents the inertia of the fluid together with the switching effect of the weir; the latter represents the flow resistance.
The switching logic is on if the level on either side of the weir reaches the level of the weir.

Summary information

System Weir: Detailed description here

Interface information:

Parameter $1 represents actual parameter **i_s**

Parameter $2 represents actual parameter **r**

Port in represents actual port **in**

Port out represents actual port **out**

Variable declarations:

This component has no PAR declarations

Units declarations:

This component has no UNITs declarations

The label file: Weir_lbl.txt

```
#SUMMARY Weir
#DESCRIPTION Detailed description here

## System Weir, representation lbl, language txt
## File Weir_lbl.txt
## Generated by MTT on Fri Oct  5 10:50:46 BST 2001

  ######################################
  ##### Model Transformation Tools #####
  ######################################

  ###############################################################
  ## Version control history
```

```
  ###############################################################
  ## $Id: Weir_lbl.txt,v 1.2 2001/10/05 11:24:35 gawthrop Exp $
  ## $Log: Weir_lbl.txt,v $
  ## Revision 1.2  2001/10/05 11:24:35  gawthrop
  ## Updated for new mtt features - input.txt etc
  ##
  ## Revision 1.2  2001/07/03 22:59:10  gawthrop
  ## Fixed problems with argument passing for CRs
  ##
  ###############################################################

## Port aliases
#ALIAS in in
#ALIAS out out

## Argument aliases
#ALIAS $1 i_s
#ALIAS $2 r

## Each line should be of one of the following forms:
##      a comment (ie starting with #)
##      component-name cr_name arg1,arg2,..argn
##      blank

## ---- Component labels ----

## Component type ISW
isw lin flow,i_s

## Component type R
r lin flow,r

## Component type R
[in] SS external,external
[out] SS external,external
```

Subsystems

- ISW: Switched I component (1) No subsystems.

4.2 Weirs_struc.tex

MTT command:

```
mtt Weirs struc tex
```

List of inputs for system Weirs			
	Component	System	Repetition
1	f	Weirs__f	1

List of outputs for system Weirs			
	Component	System	Repetition
1	p1	Weirs__p1	1
2	p2	Weirs__p2	1
3	p3	Weirs__p3	1

List of states for system Weirs			
	Component	System	Repetition
1	tank1	Weirs__tank1	1
2	tank2	Weirs__tank2	1
3	tank3	Weirs__tank3	1
4	MTT_SWITCH	Weirs__w1__isw__MTT_SWITCH	1
5	MTT_SWITCH	Weirs__w2__isw__MTT_SWITCH	1

4.3 Weirs_ode.tex

MTT command:

```
mtt Weirs ode tex
```

$$
\begin{aligned}
\dot{x}_1 &= \frac{(\varepsilon_i u_1 - x_4)}{\varepsilon_i} \\
\dot{x}_2 &= \frac{(c_2 x_4 r_l - c_2 x_5 r_l - \varepsilon_i x_2)}{(c_2 \varepsilon_i r_l)} \\
\dot{x}_3 &= \frac{x_5}{\varepsilon_i} \\
\dot{x}_4 &= \frac{(-c_1 c_2 \varepsilon_r x_4 - c_1 \varepsilon_i x_2 + c_2 \varepsilon_i x_1)}{(c_1 c_2 \varepsilon_i)} \\
\dot{x}_5 &= \frac{(-c_2 c_3 \varepsilon_r x_5 - c_2 \varepsilon_i x_3 + c_3 \varepsilon_i x_2)}{(c_2 c_3 \varepsilon_i)}
\end{aligned}
\tag{4.2}
$$

$$
\begin{aligned}
y_1 &= \frac{x_1}{c_1} \\
y_2 &= \frac{x_2}{c_2} \\
y_3 &= \frac{x_3}{c_3}
\end{aligned}
\tag{4.3}
$$

4.4 Weirs_switch.txt

MTT command:

```
mtt Weirs switch txt
```

```
# These are the switches deduced from ISW and CSW components
weirs__w1__isw__mtt_switch 4
weirs__w2__isw__mtt_switch 5
```

4.5 Weirs_simpar.tex

MTT command:

```
mtt Weirs simpar tex
```

```
# Simulation parameters for system Weirs (Weirs_simpar.txt)
# Generated by MTT on Sun Jul 26 13:55:27 BST 1998.
###############################################################
## Version control history
```

```
#################################################################
## $Id: Weirs_simpar.txt,v 1.2 2001/10/05 11:24:35 gawthrop Exp $
## $Log: Weirs_simpar.txt,v $
## Revision 1.2  2001/10/05 11:24:35  gawthrop
## Updated for new mtt features - input.txt etc
##
## Revision 1.1  1999/12/21 09:22:43  peterg
## Initial revision
##
#################################################################


LAST=19.9;
DT=0.01;
STEPFACTOR=1;
```

4.6 Weirs_numpar.tex

MTT command:

```
mtt Weirs numpar tex
```

```
# -*-octave-*- Put Emacs into octave-mode
# Numerical parameter file (Weirs_numpar.txt)
# Generated by MTT at Tue Mar  2 07:55:47 GMT 1999

# %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
# %% Version control history
# %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
# %% $Id: Weirs_numpar.txt,v 1.1 2000/12/28 17:47:43 peterg Exp $
# %% $Log: Weirs_numpar.txt,v $
# %% Revision 1.1  2000/12/28 17:47:43  peterg
# %% To RCS
# %%
# %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

# Parameters
c_1 =  1.0; # Weirs
c_2 =  1.0; # Weirs
c_3 =  1.0; # Weirs
epsilon_i =  0.01; # weir
```

```
epsilon_r =  0.1; # weir
r_l =  5.0; # Weirs
```

4.7 Weirs_input.tex

MTT command:

```
mtt Weirs input tex
```

```
# -*-octave-*- Put Emacs into octave-mode
# Input specification (Weirs_input.txt)
# Generated by MTT at Tue Mar  2 07:48:04 GMT 1999
###############################################################
## Version control history
###############################################################
## $Id: Weirs_input.txt,v 1.3 2003/06/11 15:59:30 gawthrop Exp $
## $Log: Weirs_input.txt,v $
## Revision 1.3  2003/06/11 15:59:30  gawthrop
## Updated examples for latest MTT.
##
## Revision 1.2  2001/10/05 11:24:35  gawthrop
## Updated for new mtt features - input.txt etc
##
## Revision 1.1  2000/05/19 07:12:04  peterg
## Initial revision
##
###############################################################

# Set the inputs
## Removed by MTT on Fri Oct  5 10:47:20 BST 2001: mttu(1) = 1.0*(t<10);



## Removed by MTT on Tue Jun 10 10:53:48 BST 2003: weirs_f
= 1.0*(t<10); # Added by MTT on Fri Oct 05 10:48:01 BST 2001


weirs__f = 1.0*(t<10);
```

4.8 Weirs_logic.tex

MTT command:

```
mtt Weirs logic tex
```

```
# -*-octave-*- Put Emacs into octave-mode
# Simulation parameters for system Weirs (Weirs_logic.txt)
# Generated by MTT on Thu May 18 12:29:15 BST 2000.
###############################################################
## Version control history
###############################################################
## $Id: Weirs_logic.txt,v 1.4 2003/06/11 15:59:47 gawthrop Exp $
## $Log: Weirs_logic.txt,v $
## Revision 1.4  2003/06/11 15:59:47  gawthrop
## Updated examples for latest MTT.
##
## Revision 1.3  2002/09/29 14:23:16  geraint
## Updated switch names.
##
## Revision 1.2  2001/10/05 11:24:35  gawthrop
## Updated for new mtt features - input.txt etc
##
## Revision 1.1  2000/05/19 07:13:29  peterg
## Initial revision
##
## Revision 1.1  2000/05/18 11:29:58  peterg
## Initial revision
##
###############################################################


# Set the switches
# First weir - height 1
# Second weir - height 2
## Removed by MTT on Tue Jun 10 10:54:35 BST 2003: weirs_w1_isw_mt
= ((weirs_tank1 > 1) | (weirs_tank2 > 1));
## Removed by MTT on Tue Jun 10 10:54:35 BST 2003: weirs_w2_isw_mt
= ((weirs_tank2 > 2) | (weirs_tank3 > 2));
weirs__w1__isw__mtt_switch_logic = ((weirs__tank1 > 1) | (weirs__t
weirs__w2__isw__mtt_switch_logic = ((weirs__tank2 > 2) | (weirs__t
```

4.9 Weirs_odeso.ps

MTT command:

```
mtt Weirs odeso ps
```

This representation is given as Figure 4.4 (on page 73).

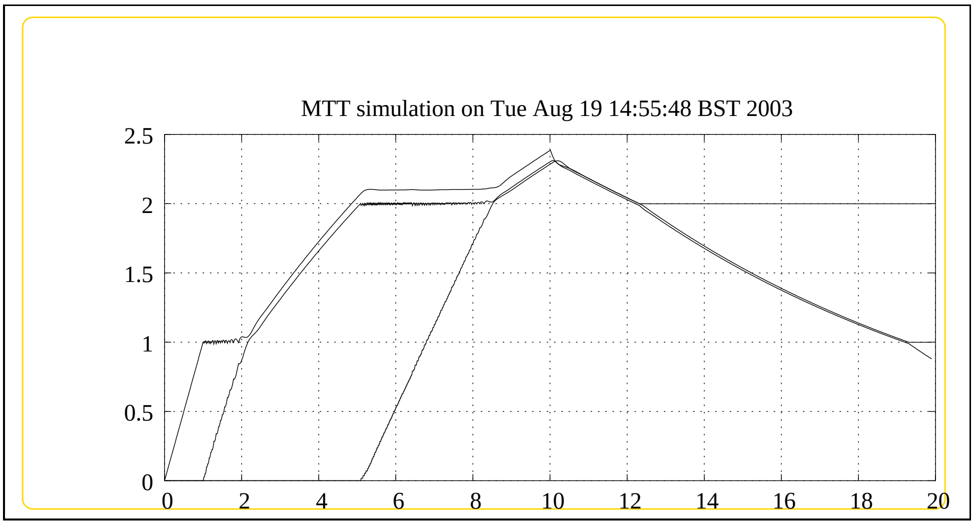

Figure 4.4: System **Weirs**, representation odeso (-noargs)

4.10 Weirs_rep.txt

MTT command:

```
mtt Weirs rep txt
```

```
## -*-octave-*- Put Emacs into octave-mode
## Outline report file for system Weirs (Weirs_rep.txt)
## Generated by MTT on" Tue Mar  2 22:02:59 GMT 1999.

###############################################################
## Version control history
###############################################################
## $Id: Weirs_rep.txt,v 1.6 2001/10/05 11:24:35 gawthrop Exp $
## $Log: Weirs_rep.txt,v $
## Revision 1.6  2001/10/05 11:24:35  gawthrop
```

```
## Updated for new mtt features - input.txt etc
##
## Revision 1.5  2000/05/19 07:15:04  peterg
## Added logic
##
## Revision 1.4  1999/12/21 09:24:26  peterg
## Euler integration - compiled
##
## Revision 1.3  1999/03/02 22:19:22  peterg
## Added switch rep
##
## Revision 1.2  1999/03/02 22:09:49  peterg
## Addded switch rep
##
## Revision 1.1  1999/03/02 22:04:11  peterg
## Initial revision
##
###############################################################

mtt Weirs abg tex # The system description
## mtt Weirs cbg ps          # The causal bond graph
## Uncomment the following lines or add others
mtt Weirs struc tex         # The system structure
## mtt Weirs dae tex         # The system dae
mtt Weirs ode tex         # The system ode
mtt Weirs switch txt         # The system switches
## mtt Weirs sspar tex # Steady-state parameters
## mtt Weirs ss tex  # Steady state
## mtt Weirs dm tex # Descriptor matrices (of linearised system)
## mtt Weirs sm tex # State matrices (of linearised system)
## mtt Weirs tf tex # Transfer function (of linearised system)
## mtt Weirs lmfr ps # log modulus of frequency response (of linea
mtt Weirs simpar tex # Simulation parameters
mtt Weirs numpar tex # Numerical simulation parameters
mtt Weirs input tex # Simulation input
mtt Weirs logic tex # Switch logic
mtt Weirs odeso ps # Simulation output

mtt Weirs rep txt
```

Index