Teaching Control Valve Friction Using Modeling and Simulation

Joseph J. Yamé, Lhoussain El-Bahir

Department of Control Engineering and Systems Analysis, CP165/55
Université Libre de Bruxelles, 50 Avenue F.D. Roosevelt
Brussels-1050, Belgium
Email : joseph.yame@ulb.ac.be, lhoussain.elbahir@ulb.ac.be

1. Introduction

Real-life control engineering requires extensive knowledge in control system design, sensors and actuators, and process dynamics. The topics of control system design and process dynamics are usually taught in greater detail at universities. However, many practical topics, which are critical to the successful operation of real-life control systems, are left out in most textbooks and hence seldom introduced to university students. An example of such topics is the control valve which is a critical and ubiquitous component in most industrial process control loops. Indeed, control valves are the most found final control elements used in the process industries to regulate flow rates, see e.g. figure 1 below for a simple level control through flow rate modulation.

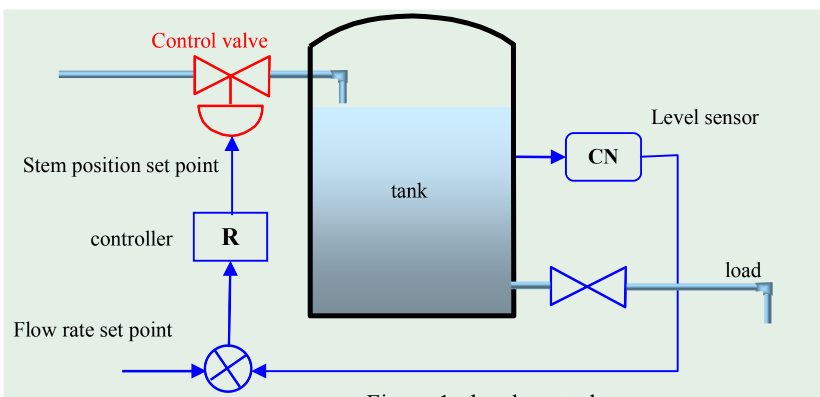

Figure 1 : level control

These regulations or variations in flow rates affect any controlled variables that is related to it such as temperature, level, pressure, PH, the flow itself, etc…When a valve is used for controlling a flow in a pipe, a commonly encountered problem is that the valve may experience friction, causing poor control performance. Friction usually occurs between any two contact parts in mechanical devices. It is a complicated phenomenon which depends on many factors, including contact dynamics, surface topology and surface chemistry, lubrication, …Friction in valves gives rise to nonlinear phenomena and it is well known from control engineering practitioners that such nonlinearities in control valves are the largest sources of process variability and oscillations in process control as well as severe stability problems. The performance degradation caused by valve friction, often manifested as oscillations, are usually compensated for by detuning the controller instead of undertaking valve maintenance. Note that since oscillations in control loops might have their roots in several possible causes, it is not always easy to get rid of oscillations. In order to be able to remedy in a suitable way to such oscillations and performance degradation in process control loop, it is important to recognize the root of the oscillations and particularly when these are induced by control valve friction.

The practical expert knowledge in control valves problems are rarely treated in control engineering courses and are mostly learned on the job and through possibly expensive mistakes. This paper aims at bridging the gap between control theory and practice by encompassing an introductory knowledge on control valve friction using modeling and simulation in the MATLAB/SIMULINK environment. We have developed a control valve simulation model which can be used for the analysis of significant valve effects on process control loop performance. The model incorporates phenomena such as dead time, dead zone, hysteresis, stiction , saturation and dynamics. The simulation model can be parameterized in terms of variables describing the above phenomena and can be used together with process loop models. The model has been used as an instructional tool in the control engineering laboratory for the design of a friction compensator (controller). It is hoped that the incorporation of these key concepts and useful knowledge in the curricula will strengthen the practical base of control engineering students and accelerate the acquisition of some expertise in handling such common nonlinearity in process control.

The paper is organized as follows: in the next section, we recall the basic structure of a control valve system. In section 3, a simple mathematical model describing the main phenomena is derived and specifies a control valve as a nonlinear dynamical system. Based on this mathematical model, a SIMULINK model is constructed in section 4 using standard and S-functions blocks of the MATLAB/SIMULINK software. In section 5, we illustrate the effects of control valve friction on a control loop and exhibits the typical oscillations waveform induced by such nonlinearity. The source files of the S-functions are given in the appendix.

2. The control valve

A control valve is a power-operated device which modifies the fluid flow rate in a process control system. It consists of three main parts: the valve body, the actuator that forces the valve stem to move, and the positioner that controls the valve stem position so that it corresponds to the control signal. A schematic diagram of a control valve is depicted in figure 2.

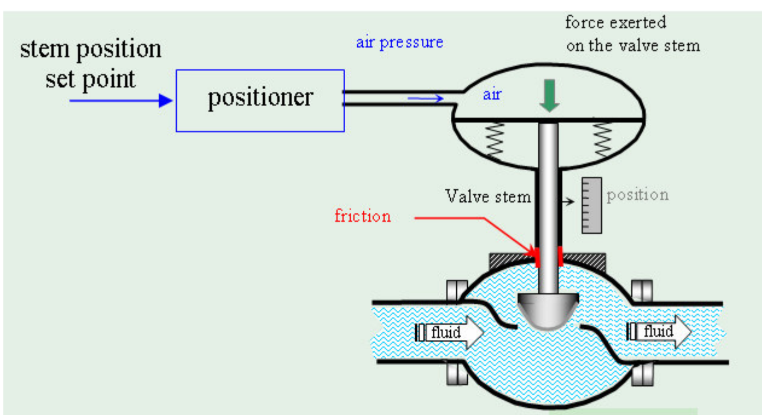

Figure 2 : Control valve

There are many types of valve bodies, actuactors and positioners in process control, however the principle described in figure 2 corresponds to almost all control valves.

3. A simple nonlinear model for control valves

As noted in the introduction, nonlinear phenomenon in control valves are one of the main sources for control performance degradation in process control and this nonlinearities result essentially from friction phenomena. This friction is evidenced by stiction or stick-slip motion, dead band, hysteresis (backlash), time-delay and actuator control saturation. In all sorts of valve, friction appears in the packing boxes around the valve stem, especially when these are tightened hard. Note that in most control problems, such valve nonlinearity is overlooked and valves are simply modelled as first-order linear system with a time constant although time constants of control valves are not as significant as dead band or time-delay. A realistic valve model should take into account the above mentioned friction nonlinearity, that is:

- the delay or dead-time
- the dead band
- the hysteresis
- the stiction
- the dynamics

An empirical model suggested by [1], based on experimental investigations, with the above most critical nonlinearities is given below. It is based on a first-order linear dynamical model into which the nonlinearities are added.

Friction and delay

The time-delay cannot be represented as it is done classically because it depends on the position control error. If we denote the position error (in the position control) by

$$e = i - h \tag{1}$$

where i is the normalized input signal ($0 \le i \le 1$) and h is the normalized valve travel ($0 \le h \le 1$), then the following equation describes the time-delay

$$\frac{dz}{dt} = \begin{cases} \max(e,0) & for \quad z < -z_{\max} \\ \min(e,0) & for \quad z > z_{\max} \\ e & for \quad |z| \le z_{\max} \end{cases} \tag{2}$$

where the variable z can be viewed as a relative pressure (it is the time-delay state variable). When the absolute value of this variable z is below a certain limit value z_{max},, the valve stem does not move (see equation (4) below). Time-delay of the valve model effects only when changing the signal direction. This models the static friction of the valve. The value of z_{max},, can be used to simulate the intensity of the friction forces.

Dead band and hysteresis

Dead band is caused by mechanical backlash and static friction. Backlash is lost motion due to slop in valve linkages. This is evidenced by hysteresis in the control valve and it is modelled by a relative deviation from the ideal set-point. The control error can be computed as

$$e = \begin{cases} e - e_{dead}\, sign(e) & for \quad |e| > e_{dead} \\ 0 & for \quad |e| \le e_{dead} \end{cases} \tag{3}$$

where e_{dead} is a half from the measured dead band.

Dynamics

The dynamics can be described by a first-order system with a velocity limitation as

$$\frac{dh}{dt} = \begin{cases} \min\left(\left|\frac{e}{\tau}\right|, \nu_{\max}\right) sign(e) & quand \quad |z| > z_{\max} \\ 0 & quand \quad |z| \le z_{\max} \end{cases} \tag{4}$$

where h is the valve stem position, τ is the time constant, and $\nu_{\max}$ is the maximum relative speed. Note that the velocity is set to zero when the time-delay state variable z is inside the "delay-time zone".

4. The Simulink Model

The above valve model equations have been programmed in the MATLAB/SIMULINK environment using S-functions and standard blocks for laboratory courses. The valve model is represented as a main Simulink model with the stem set-point signal as the valve input and the actual stem position as the valve output

Figure 3. Simulink Control Valve Model

The sub-systems composing this Simulink model can be viewed by looking under the mask and this is given in the figure below

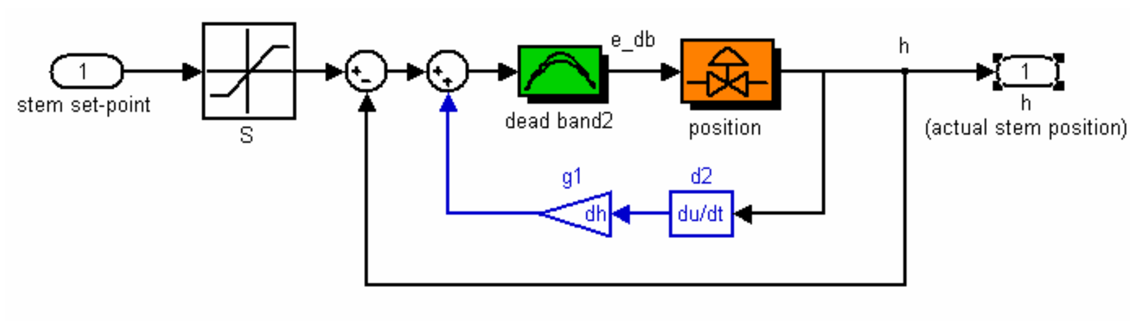

Figure 4. Blocks inside the Control Valve Model

From the nonlinear control model derived in section 3, the parameters and the tuning of the control valve model are done through four (4) parameters which are:

- τ the time constant
- v_{max} the maximum relative speed
- e_{dead} half from the measured dead band
- z_{max} time-delay parameter (friction parameter)

The following graphical interface is used to set the model parameters in Simulink where *dh* is the stick-slip parameter

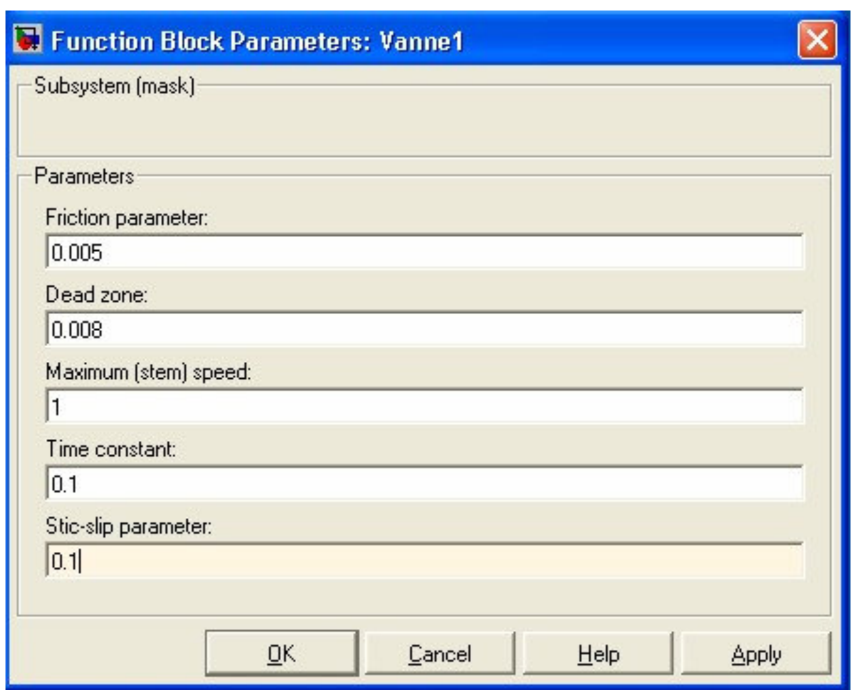

Figure 5. Graphical User Interface for Control Valve Parameters Setting

These parameters might be fixed such that the model behaviour meets the control valves specifications as established in [3]. The students can simulate the step response and the response to a periodic signal in the form of a ramp for the control valve in healthy and friction situations and compare the obtained valve behaviours. The healthy control valve behaviour described by the model is illustrated by figures 6. The backlash is near 1%, the time constant is fixed at 0.1s and the stiction is represented by time-delay parameter fixed at 0.005

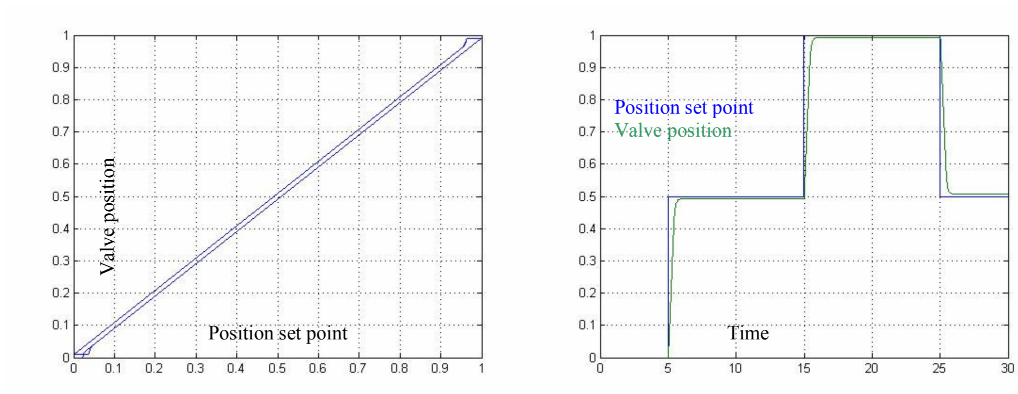

Figure 6. healthy valve behaviour.
Left: plot of position setpoint versus valve position.
Right: multi-step valve response.

Figure 7 below shows a case when the valve suffers from stiction and dead band.

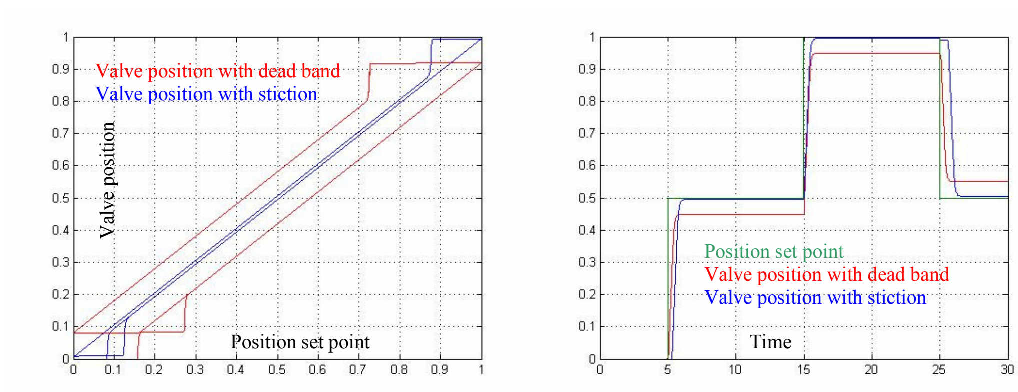

Figure 7. Valve with stiction and dead band.
Left: plot of position setpoint versus valve position.
Right: multi-step valve response.

This simulation results helps students to understand that under stiction (which is actually a resistance to the start of motion), the actuactor piston applies increasing pressure in the air cylinder which suddenly gives a rapid moving of valve's stem. Sudden pressure decreasing in the air cylinder causes a temporary valve stem stopping and leads to a jumpy movement as shown in figure 8.

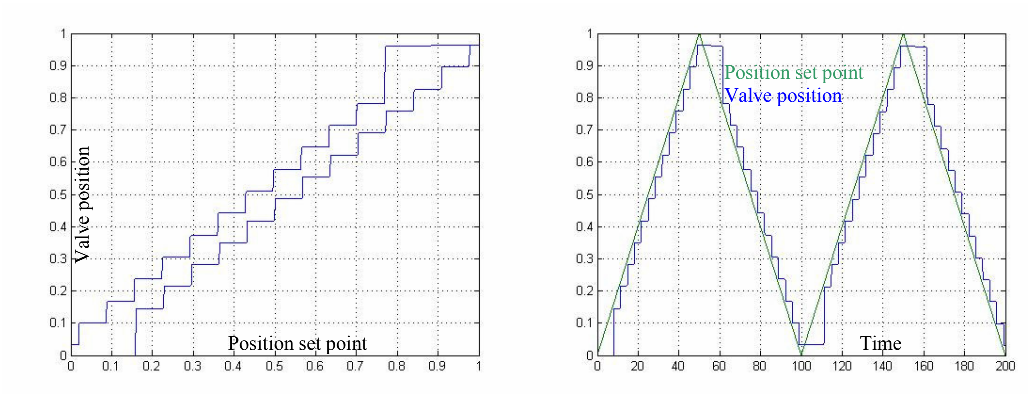

Figure 8. Jumpy valve movement due to stiction.
Left: plot of position setpoint versus valve position.
Right: Multi-ramp valve response

5. Illustration of control valve stiction effects in closed-loop control system

The Simulink control valve model can be used as a block component in a feedback system to illustrate valve friction effects on control performance. As mentioned in the introduction, such effects are evidenced by oscillations in the control loop.

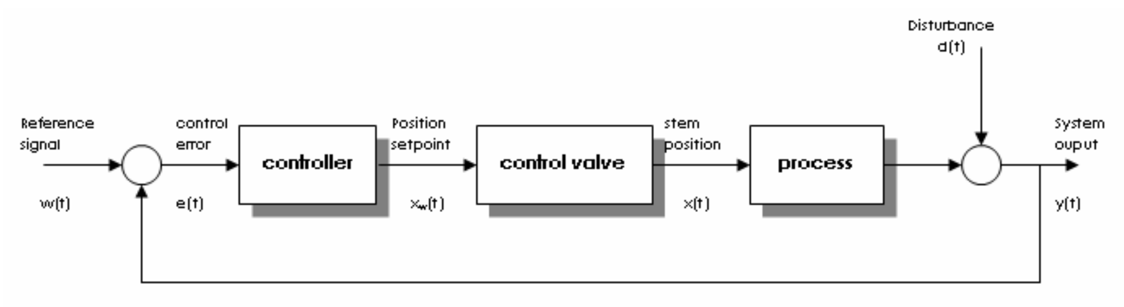

Figure 9. Closed-loop System with control valve

Consider a closed-loop control system as depicted in figure 9 and assume that the valve undergoes a stuck in a certain position due to high static friction. When the controller has an integrating mode, it increases the position set-point until the static friction can be overcome. As a consequence, the valve breaks off and moves to a new position which is usually on the other side of the desired set-point. The valve sticks again in the new position and the same process starts in the opposite direction. Typical stick-slip behavior results in a rectangular or triangular process output and triangular control signal. The period of oscillation depends on the process dynamics, controller dynamics, and valve characteristics (friction, dead band, etc…). The simulation in figure 10 below obtained with the SIMULINK valve model shows that a high valve friction leads to a high frequency and high magnitude oscillation and vice versa. This simulation clearly exhibits the typical rectangular waveforms of the oscillations of the system output and the triangular waveforms of the control signals induced by valve friction.

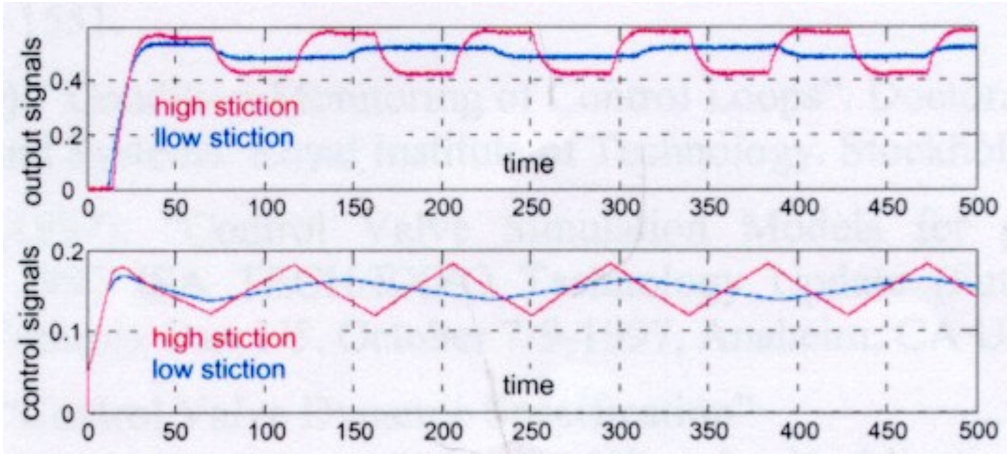

Figure 10. Closed-loop response: Effects of valve friction (oscillations)

Using the valve model in control loops, students can simulate different control systems with different combinations of controllers and processes types and plot system responses to step signal set-point.

6. Conclusion

In this paper, the need to incorporate practical contents in the curricula of control engineering courses have been highlighted by the use of a nonlinear control valve model to teach friction phenomena and its effects on closed-loop control performance. A simple realistic nonlinear valve model has been constructed and implemented in Simulink. The model offers inexpensive educational possibilities to illustrate practical nonlinear effects in process control loops. Experiments using this model have been performed by students in our control laboratory courses and they have been well received and enhanced the understanding of friction effects on control performance.

References

[1]- Riihilahti J. (1997) – Control Valve Simulation Models for Process Simulators. Proceedings of the 1997 ISA TECH/EXPO technology. Octobr 7-9, 1997, Anaheim, CA, USA

[2]- Hägglund T. (2002)- A friction Compensator for Pneumatic Control Valves. Journal of Process Control, Vol.12, pp. 897-904

[3]- Entech (1998) – Control Valve Dynamic Specifications. http://www.emersonprocess.com/entechcontrol/download/publications/valvsp30.pdf

Appendix – Source file of S-functions "dead band" and "position"

```
function [sys,x0,str,ts] = deadband(t,x,u,flag,edead)
A=[0 ];B=[1];C=[1];D=[0];

switch flag,
  % Initialization %
  case 0,
    [sys,x0,str,ts]=mdlInitializeSizes(A,B,C,D);

  % Derivatives %
  case 1,
       sys = 0;
  % Outputs %
  case 3,
    if abs(u)>edead,
      sys=u-edead*sign(u);
      else
      sys=0;
    end

  % Unhandled flags %
  case {2, 4, 9 },
    sys = [];

  % Unexpected flags %
  otherwise
    error(['Unhandled flag = ',num2str(flag)]);

end
%==============================================================================
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-function.
%==============================================================================
%
function [sys,x0,str,ts]=mdlInitializeSizes(A,B,C,D)
sizes = simsizes;
sizes.NumContStates  = 1;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 1;
sizes.NumInputs      = 1;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;

sys = simsizes(sizes);
x0  = zeros(1,1);
str = [];
ts  = [0 0];
```

```
function [sys,x0,str,ts] = position(t,x,u,flag,tau,vmax,zmax)
%========================================================
% This 'S-function' simulates the time-delay (dead-time)
% and the stem position from the mathematical model of the
% control valve
% u=e;
% x(1)=z (deadtime)  x(2)=h (position)
%============================================
A=[0   0; 0   0]; B=[0;0]; C=[0 0]; %D=[0 0];
```

```
switch flag,

  %%%%%%%%%%%%%%%%%%
  % Initialization %
  %%%%%%%%%%%%%%%%%%
  case 0,
    [sys,x0,str,ts]=mdlInitializeSizes(A,B,C);

  %%%%%%%%%%%%%%%
  % Derivatives %
  %%%%%%%%%%%%%%%
  case 1,
    % u=e;
   H=sign(u)*min(abs(u/tau),vmax);
   if x(1) < -zmax
       sys = [max(u,0);H];
       elseif x(1) > zmax
       sys = [min(u,0);H];
   else
       sys = [u;0];
   end

  %%%%%%%%%%%
  % Outputs %
  %%%%%%%%%%%
  case 3,
   %sys = x(2);
   if x(2) < 0,
      sys = 0;
  elseif x(2) >1,
      sys=1;
  else
      sys=x(2);
  end

  %%%%%%%%%%%%%%%%%%%
  % Unhandled flags %
  %%%%%%%%%%%%%%%%%%%
  case { 2, 4, 9 },
    sys = [];

  %%%%%%%%%%%%%%%%%%%%
  % Unexpected flags %
  %%%%%%%%%%%%%%%%%%%%
  otherwise
    error(['Unhandled flag = ',num2str(flag)]);

end
% end csfunc

%
%=============================================================================
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-function.
%=============================================================================
%
function [sys,x0,str,ts]=mdlInitializeSizes(A,B,C)

sizes = simsizes;
sizes.NumContStates  = 2;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 1;
sizes.NumInputs      = 1;
```

```
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1;

sys = simsizes(sizes);
x0  = zeros(2,1);
str = [];
ts  = [0 0];

% end mdlInitializeSizes
```

Biographical information

JOSEPH J. YAMÉ graduated in electrical and mechanical engineering and received the post-graduate degree in control engineering and the degree of Doctor in Applied Sciences all from the Université Libre de Bruxelles (ULB), Brussels, Belgium. In 1987, he was appointed by the National Polytechnic Institute of the Ivory Coast (Côte d'Ivoire) as a lecturer in the Electrical and Electronic Engineering Department. Since November 2000, he has been a researcher in the Control Engineering and Systems Analysis Department of the ULB. Over the past years his educational and research activities have focused on different subjects in control engineering. He has been a consultant in automation and process control for a variety of industries (breweries, oil refineries, power plants, and others) in the Ivory Coast.

LHOUSSAIN EL BAHIR graduated in physics from the Université Abdelmalek Essaâdi, Tétouan, Morocco in 1989. He received his post-graduate degree in control engineering and his Doctor in Applied Sciences from the Université Libre de Bruxelles (ULB) in 1992 and 2000 respectively. He is now a researcher in Control Engineering Department of the ULB. His research interests include identification, predictive control, fault detection and isolation and loop monitoring. Both theoretical aspects and applications to the process industry are considered.