

Forcing function and time delay in qualitative simulation

M. Tétreault,^a B. Marcos^b & J. Lapointe^c

^a *Department of Automated Production Engineering, École de Technologie Supérieure, Montréal, Québec, Canada, H2T 1R0*

^b *Department of Chemical Engineering, University of Sherbrooke, Sherbrooke, Québec, Canada, J1K 2R1*

^c *Department of Mechanical Engineering, University of Sherbrooke*

ABSTRACT

Many potential uses of qualitative simulation (QS), such as control systems, require adding functions with explicit time to describe actions taken by agents. Also, many control systems have time delay. Therefore, the modelling of the physical world must consider the effects of actions and time delay.

This paper shows how to extend qualitative simulation to include actions and functions with time delay. The effects of actions are some forced variables (or parameters) added to the description of the system. Our algorithm treats these new parameters like state variables (i.e. system parameters) but their possible state transitions are reduced. In our algorithm, a function with a pure time delay is an additional variable corresponding to the same variable without delay. This function is similar to a forcing function but our algorithm builds its definition during the simulation. The representation of the forcing functions and the functions with time delay are purely qualitative or semiquantitative.

The integration of these functions to QSIM permits its application to control systems. The example shows that qualitative time delay, forcing function and feedback system can be simulated.

INTRODUCTION

Applications of qualitative simulation (QS) are usually in domains where complete models do not exist, but relationships between variables are often known to some degree. Qualitative simulation captures the underlying principles and enables reasoning from incomplete knowledge. Many potential uses of QS, such as control systems, require adding functions with explicit time. For example, a plant operator shuts off a valve within six seconds. Also, there are many control systems that have a time delay within the closed loop. Then, the modelling of the physical world must consider the effects of actions and time delay.

This paper shows how to extend qualitative simulation to include actions and functions with time delay. The effects of actions taken by agents are some forced variables (in a forcing function) of the system. In our algorithm, a function with a pure time delay (without attenuation) is an additional variable $f(t-t_d)$ corresponding to the same variable $f(t)$ without delay. The representation of the forcing functions and the functions with time delay are purely qualitative or semiquantitative.

This paper includes five additional sections. The next section presents a literature survey. The third section provides some definitions. The fourth and fifth sections present the addition of the forcing functions and the functions with time delay in QS, more specifically in QSIM. In the sixth section, the integration of forcing functions and functions with time delay to qualitative physics permits the application of QS to a control system.

LITERATURE SURVEY

The three QS approaches most widely used are the confluences of de Kleer and Brown¹, the Qualitative Process Theory (QPT) of Forbus² and the Qualitative Simulation (QSIM) of Kuipers³. Starting with the variable initial conditions and a set of qualitative constraints, QS generates a tree of behaviours. The generation of multiple behaviours is partly due to the inability of QS methods to represent quantitative information and temporal knowledge. In QS methods, the system state can be described more precisely by adding a quantitative range of numerical values to the landmarks, the length of intervals between them, the monotonic functions $M^\pm$ and order of magnitude (Kuipers and Berleant⁴, Simmons⁵, Wiegand and Leitch⁶, Travé-Massuyès and Piera⁷, Mavrovouniotis and Stephanopoulos⁸, and Tétreault et al.⁹). In Tétreault et al.⁹, combining quantitative and qualitative

knowledge increases temporal information that takes the form of temporal durations.

Generally, existing algorithms cannot represent explicit time dependence processes such as forcing functions and functions with time delay. Forbus¹⁰ extends his QPT algorithm to include the effects of actions, resulting in action-augmented envisionments (AE). He makes two restrictions. First, at most one action can be taken at a time. Second, his method requires that actions occur quickly relative to dynamical changes and that system parameters stay constant during an action. This restricts the expressibility of AE and continuity cannot always be satisfied. Our approach does not impose these restrictions. It is written in Prolog following the formalism of QSIM³.

FORMALISM

Before presenting the modifications made to the QSIM algorithm, this section provides some definitions. There are two kinds of variables : state variables and forced variables. A *state variable* represents a degree of freedom of the system. A forced variable (in a forcing function) represents an input variable or a triggered alarm. A *variable* v_m has a value for its primitive and its K derivatives. $K+1$ parameters then describe the variable. $^k p$ is a parameter representing the k^{th} derivative of the variable. The *qualitative state of a parameter* is a pair $\langle \text{value, derivative} \rangle : \text{QS}(^k p_n) = \langle ^k p_n, ^{k+1} p_n \rangle$. The *qualitative state of a variable* is the qualitative state of its $K+1$ parameters. The *qualitative state of the system* is the states of its variables. An *event* is a tuple formed by a system state and its associated temporal duration in which the parameters keep their values. A *point event* or instant happens at a distinguished time-point whereas we have an *interval event* between two distinguished time-points.

During the simulation, the parameters have real values included in the set of interval $I(\mathbb{R})$ defined as : $I(\mathbb{R}) = \{X|X=\{x_1, x_2\}, x_1 \leq x_2, \forall x_i \in \mathbb{R}\}$ where $\{$ and $\}$ represent open or closed intervals and $\pm\infty$ are considered excluded. Each parameter $^k p$ has a set of ordered *landmarks* $\mathbb{L}$ that represent some characteristics (extremes, alarms, etc.) :

$$\mathbb{L}(^k p) = \{l_1, l_2, \dots, l_j, \dots, l_J\} \quad \text{where } l_j \in I(\mathbb{R})$$

A parameter has a value given by a landmark or a value between two consecutive landmarks.

In interval arithmetic, the traditional equality sign may have two significations : an attribution ($Y \leftarrow X$) and a qualitative equality relation ($Y \cap X$). A *qualitative attribution* gives a value to a variable.

A *qualitative equality* relation is verified if the interval intersection between X and Y is not empty. Such a relation is said to be consistent. The values of X and Y are evaluated by applying operations in the set $I(\mathbb{R})$. The basic *operations* $(\oplus, \ominus, \otimes, \oslash)$ group the quantity influence in a relation. They are defined in reference 9.

In Tétreault et al.⁹, we modify the framework of QSIM to add quantitative and temporal knowledge. *Reasoning* starts by taking an active state. Forcing functions and removal of landmarks modify the active state. The triggering of a forcing function adds a forced variable and thus forced parameters to the active state whereas the end of a forcing function removes it. This occurs only when the active state is a point event. Then, the algorithm generates the possible transitions by using the transition rules (The tables of P-transitions and I-transitions of QSIM are located in reference 3). The descriptions of the forcing functions constrain the possible transitions of forced parameters. The application of a behavioural consistency step and a temporal consistency step eliminates behaviours that do not respect the relations.

FORCING FUNCTIONS

The extension of qualitative simulation to include actions requires the addition of their knowledge to the qualitative physics, this knowledge being when the actions become effective and how the actions affect the system behaviour.

Table 1 - A forcing function.

Qualitative		Semiquantitative	
local time	flow	local time	flow
		min	L/min
t'_0	$\langle Q_0; 0 \rangle$	$[0; 0]$	$\langle 10; 0 \rangle$
$(t'_0; t'_1)$	$\langle (0; Q_0); - \rangle$	$(0; 60)$	$\langle (0, 10); - \rangle$
t'_1	$\langle 0; 0 \rangle$	$[60; 60]$	$\langle 0; 0 \rangle$
$(t'_1; \infty)$	$\langle 0; 0 \rangle$	$(60; \infty)$	$\langle 0; 0 \rangle$

Forced variables model the effects of actions. Usually, an action modifies a constant quantity, so that it becomes a variable. Table 1 shows a forcing function corresponding to the closing of a flow control valve. The forcing function has a qualitative and a semiquantitative state description at a local time-point or at an interval between time-



points. It becomes active at time-point t'_0 . The apostrophe used in representing local time avoids mix-up with system reference time. The forcing function contains $I+1$ ($t'_0; t'_1; \dots; t'_I$) distinguished time-points t'_i where "I" is greater than two.

Our algorithm adds forcing parameters to the set of system parameters when a forcing function becomes effective. Our algorithm considers all parameters the same way but a forcing function imposes some restrictions to the possible state transitions of forcing parameters, thus limiting them to those listed in table 2. Local time of forcing function is similar to the state parameter in which the set of landmarks is $(t'_0; t'_1; \dots; t'_I)$ and the derivative of the parameter (time) is always positive. The possible state transitions are then limited to P_4, P_5, I_3 and I_4 . The local time value in table 2 indicates the state parameter given by the forcing function description (ex.: table 1).

Table 2 - Transitions for a forcing function.

P'-transitions	$QS(f; t_i) \longrightarrow QS(f; t_i; t_{i+1})$	
P'_4	$\langle t'_i; + \rangle$	$\langle \langle t'_i; t'_{i+1} \rangle; + \rangle$
P'_5	$\langle \langle t'_i; t'_{i+1} \rangle; + \rangle$	$\langle \langle t'_i; t'_{i+1} \rangle; + \rangle$
I'-transitions	$QS(f; t_i; t_{i+1}) \longrightarrow QS(f; t_{i+1})$	
I'_3	$\langle \langle t'_i; t'_{i+1} \rangle; + \rangle$	$\langle t'_{i+1}; + \rangle$
I'_4	$\langle \langle t'_i; t'_{i+1} \rangle; + \rangle$	$\langle \langle t'_i; t'_{i+1} \rangle; + \rangle$

The next state produced by a P'-transition is always unique. In a P'_4 transition, the next state of the parameter corresponds to the state at the next local time interval in the description of the forcing function. The transition P'_5 produces no state modification. The I'-transitions generate two possible transitions : no state change (I'_4) or the forcing function reaches its next landmark (I'_3).

Modelling of external actions using this method applies to both fully qualitative and semiquantitative simulations.



FUNCTIONS WITH TIME DELAY

There are many control systems that have a time delay within the closed loop of the system. For instance, a change in input variables (reference inputs, disturbances or manipulated variables) is observed in the state variables or in the outputs after some time delay. Qualitative simulation of control systems requires the modelling of time delays.

Our algorithm model functions with pure time delay without attenuation. A function with time delay $f(t-t_d)$ is an additional parameter describing the state of the system and corresponding to a parameter $f(t)$ without time delay. The addition of $f(t-t_d)$ to the simulation is necessary to produce its qualitative state description over time.

A function $f(t-t_d)$ is similar to a forcing function but its definition is carried out during the simulation. The evolution of the parameter $f(t)$ governs the definition of $f(t-t_d)$. When the function $f(t)$ reaches a landmark at time t_i , the function $f(t-t_d)$ will get to the same landmark at time t_i+t_d . Therefore, the function $f(t-t_d)$ is a forcing function activated at time t_i and terminating after a time delay t_d . Table 3 specifies the definition of the forcing function with a time delay. During the simulation, if $f(t)$ reaches a second landmark before the time delay is over, the forcing function corresponding to $f(t-t_d)$ could have several time-points.

Table 3 - Forcing function of a function with time delay.

local time		state of
t'_0	$[0; 0]$	$f(t-t_d)$ at t_i
$(t'_0; t'_1)$	$(0; t_d)$	$f(t-t_d)$ at t_i
t'_1	$[t_d; t_d]$	$f(t_i)$
$(t'_1; \infty)$	$(t_d; \infty)$	$f(t_i; t_{i+1})$

The modelling of function with time delay is applicable to both qualitative and semiquantitative simulations. For purely qualitative simulation, the number of behaviours could be limited by assuming that $f(t)$ will not reach a second landmark during the time delay.

EXAMPLE

In the following example, the liquid level control system of figure 1 contains a disturbance and a time delay represented by forcing functions. The example shows that qualitative simulation of forcing function and time delay is possible in a feedback control system.

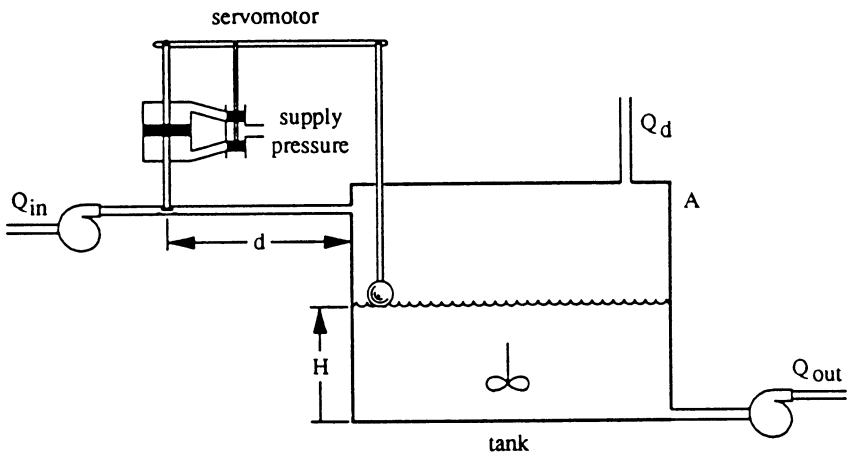


Figure 1 - A liquid level control system.

The feedback system represented in figure 1 maintains the water level constant in the tank. A hydraulic servomotor actuates a flow control valve (Q_{in}) when the disturbance Q_d modifies the water level. A time delay occurs between the actuating signal and its effect on the water level. The delay is due to the pipe length between the valve and the tank. This example uses a time delay of one second. The flow rate Q_{out} is assumed to be constant.

Figure 2 shows the corresponding block diagram. The quantities are defined about their reference point when the system is at a steady state. The reference input h_r is set to zero.

The feedback system can be represented by the following differential equation :

$$a = q_d - 10 \cdot v - h(t-1) \quad (1)$$

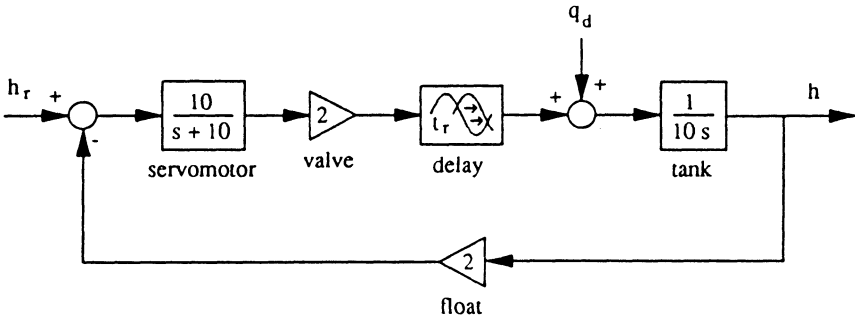


Figure 2 - Block diagram of the control system.

where :

- $h(t-1)$: water level in the tank about the reference value at time t minus one second (meter).
- v : derivative of h (m/s).
- a : second derivative of h (m/s²).
- q_d : disturbance (L/min).

Qualitatively, equation (1) is expressed as a qualitative equality and two relations on the derivatives :

$$\begin{aligned} a \cap q_d \ominus 10 \otimes v \ominus h(t-1) \\ \text{DERIV}(h; v) \\ \text{DERIV}(v; a) \end{aligned}$$

The parameters h and v represent the two degrees of freedom for the system. At the beginning of the simulation, both parameters have the following landmarks :

$$\begin{aligned} L(h) &= \{E1; E2; E3\} \\ &= \{-1; 0; 1\} \text{ m.} \\ L(v) &= \{E1; E2; E3\} \\ &= \{-\infty; 0; \infty\} \text{ m/s.} \end{aligned}$$

The initial conditions on the system are :

$$\begin{aligned} EQ(h) &= \langle E2; 0 \rangle \\ EQ(v) &= \langle E2; 0 \rangle \end{aligned}$$

At time t_0 , a disturbance represented by the forcing function of table 4 is applied to the system. Forcing functions, in our algorithm, have both a qualitative and semiquantitative representation. The flow rate q_d increases by one litre per second in a tenth of a second.

Table 4 - Forcing function of a disturbance.

qualitative		semiquantitative	
t	q_d	t (s)	q_d (L/min)
t'_i	$\langle 0; 0 \rangle$	0	$\langle 0; 0 \rangle$
$(t'_i; t'_{i+1})$	$\langle \langle 0; q_0 \rangle; + \rangle$	$(0; 0,1)$	$\langle \langle 0; 1 \rangle; + \rangle$
t'_{i+1}	$\langle q_0; 0 \rangle$	0,1	$\langle 1; 0 \rangle$
$(t'_{i+1}; \infty)$	$\langle q_0; 0 \rangle$	$(0,1; \infty)$	$\langle 1; 0 \rangle$

To verify our approach, we first perform a quantitative simulation in which an analytical function replaces the qualitative description of the disturbance. We assume that the flow rate q_d increases linearly between initial and final state. Figure 3 shows the quantitative simulation of equation (1). The water level increases by a quarter of a meter in five seconds.

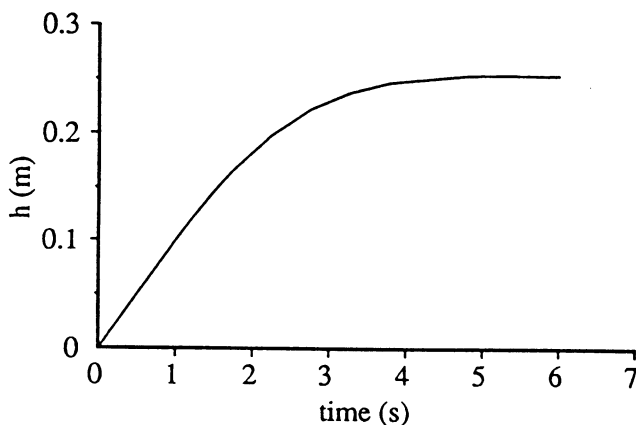


Figure 3 - Quantitative simulation of the control system.

Our semiquantitative simulation gives the results shown in figure 4. At time t_4 , four behaviours are active. The behaviours leading to node 25 correspond to the quantitative simulation and are further described in table 5. The disturbance produces an increase in water level h up to time t_4 while the parameter v increases up to time

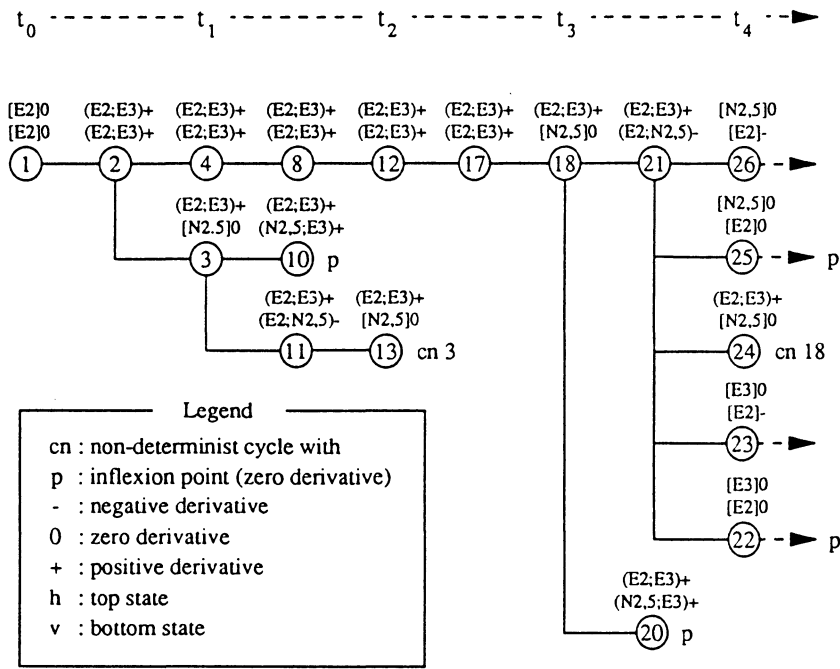


Figure 4 - Qualitative simulation of the control system.

Table 5 - Behavior of the control level system.

t	t (s)	h (m)	h(t-1) (m)	v (m/s)	q _d (L/s)
t ₀	[0;0]	<0;0>	<0;0>	<0;0>	<0;0>
(t ₀ ;t ₁)	(0;0,1)	<<(0;0,01);+>	<0;0>	<<(0;0,1);+>	<<(0;1);+>
t ₁	[0,1;0,1]	<<(0;0,01);+>	<0;0>	<<(0;0,1);+>	<1;0>
(t ₁ ;t ₂)	(0,1;1)	<<(0;0,1);+>	<0;0>	<<(0;0,1);+>	<1;0>
t ₂	[1;1]	<<(0;0,1);+>	<0;0>	<<(0;0,1);+>	<1;0>
(t ₂ ;t ₃)	(1;∞)	<<(0;1);+>	<<(0;1);+>	<<(0;0,1);+>	<1;0>
t ₃	(1;∞)	<<(0;1);+>	<<(0;1);+>	<<(0;0,1);0>	<1;0>
(t ₃ ;t ₄)	(1;∞)	<<(0;1);+>	<<(0;1);+>	<<(0;0,1);->	<1;0>
t ₄	(1;∞)	<<(0;1);0>	<<(0;1);+>	<0;0>	<1;0>

t_3 , reaching then a new landmark. The disturbance becomes constant at time t_1 . The delay on parameter h produces the distinguished time-point t_2 . The time interval $(t_0; t_2)$ is one second.

This example shows that it is possible to model qualitatively the feedback system with time delays. The addition of second derivative constraints would reduce the number of behaviours generated by the simulation. This filtering rule would have to be adapted to functions with time delay.

CONCLUSION

Many potential uses of QS, such as control systems, require adding functions with explicit time : forcing functions (actions) and functions with time delay. This paper showed how to extend qualitative simulation to include actions and functions with time delay. The effects of actions described as forced variables are added to the description of the system. In our algorithm, a function with a pure time delay (without attenuation) is an additional variable $f(t-t_d)$ corresponding to the variable $f(t)$ without delay. The representation of the forcing functions and the functions with time delay are purely qualitative or semiquantitative.

REFERENCES

1. de Kleer, J. and Brown, J.S. 'A qualitative physics based on confluences' *Artificial Intelligence*, Vol.24, pp.7-83, 1984.
2. Forbus, K.D. 'Qualitative process theory' *Artificial Intelligence*, Vol.24, pp.85-168, 1984.
3. Kuipers, B. 'Qualitative simulation' *Artificial Intelligence*, Vol.29, pp.289-338, 1986.
4. Kuipers, B.J. and Berleant, D. 'Using incomplete quantitative knowledge in qualitative reasoning' in AAAI/88 (Morgan Kaufmann), pp.324-329, *Proceedings of the National Conference on Artificial Intelligence*, Los Altos, California, 1988.
5. Simmons, R. 'Commonsense arithmetic reasoning' in AAAI/86 (Morgan Kaufmann), pp.118-124, *Proceedings of the AAAI*, Los Altos, 1986.
6. Wiegand, M.E. and Leitch, R.R. 'A predictive engine for the qualitative simulation of dynamic systems' in *Artificial Intelligence in Manufacturing*, pp.141-150, *Proceedings of the Fourth International*



Conference on the Applications of Artificial Intelligence in Engineering, Cambridge, UK, July, 1989.

7. Travé-Massuyès, L. and Piera, N. 'The orders of magnitude models as qualitative algebras' in IJCAI/89, pp.1261-1266, Vol.2, *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, Detroit, MI, USA, August, 1989.

8. Mavrovouniotis, M.L. and Stephanopoulos, G. 'Order-of-magnitude reasoning with $O(M)$ ' *Artificial Intelligence in Engineering*, Vol.4 (1), pp.106-114, 1989.

9. Tétreault, M., Marcos, B. and Lapointe, J. 'Temporal duration reasoning in qualitative simulation' *Artificial Intelligence in Engineering*, Vol.7, pp.185-197, 1992.

10. Forbus, K.D. 'Introducing actions into qualitative simulation' in IJCAI/89, pp.1273-1278, Vol.2, *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, Detroit, MI, USA, August, 1989.