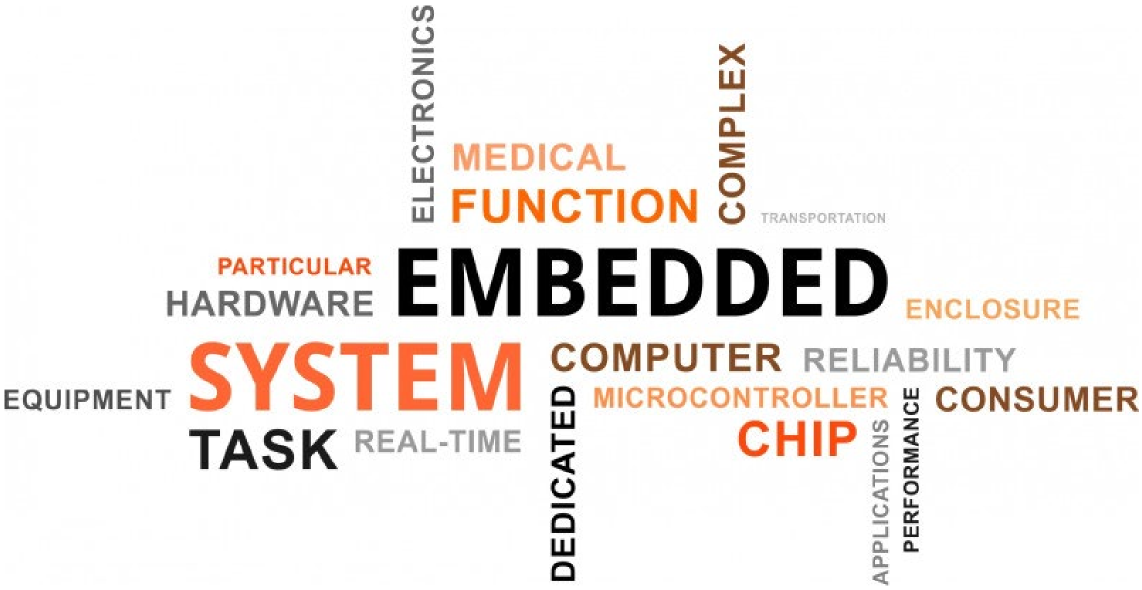
Practice 1: Multitasking - Four Tanks control example .



Manual

Mario Pereira Martin

15 de junio de 2022

Index

1. Multitasking in xenomai - Study Case **2**
1.1. Practice objectives 2
1.2. Xenomai basics concepts on Multitasking 2
1.3. Four Tank Model 2

2. Homework implementation - Using xenomai RT Linux **5**
2.1. Creating a Matlab model - Provided by the teacher 5
2.2. RT Process Schedule - Must be done by students 5
2.3. Delivering 6

1 Multitasking in xenomai - Study Case

1.1 Practice objectives

The objective of this first practice is to use the concepts of multitasking related to Real Time Linux (xenomai) and apply them to a realistic case. As a study case, a well-known and commonly used plant at the academic level, the 4-tank plant, will be used. Our fundamental objective will be to create a real-time control system and apply it to a model of this plant.

1.2 Xenomai basics concepts on Multitasking

During practice, ways to create multiple individual or periodic tasks will be reviewed and studied in order to create a process composed of different tasks. Review the proposed exercises in Moodle ('Multitasking01'). Subsequently, a Report will be delivered on the work carried out with the models of the four tank plant, in addition to the corresponding codes.

1.3 Four Tank Model

We are facing an assembly of 4 interconnected tanks so that when trying to introduce flow in one of the lower tanks, flow is also introduced in the adjacent upper one, which in turn discharges on the corresponding lower tank. This fact means that when trying to maintain a fixed height in one of the lower tanks, we alter the level of the adjoining tank. These disturbances make their control a little more complex.

We are facing a system with 4 states that correspond to the heights of each tank $x = (h_1, h_2, h_3, h_4)$. As system inputs there will be two adjustable flows $Q = (q_a, q_b)$. These two flows distribute water to the tanks through 4 manually adjustable valves that will have a fixed configuration $\gamma = (\gamma_a, \gamma_b)$. For example, the flow rate q_a distributes water to tanks 1 and 4 through the three-way valve γ_a so that the flow rate $q_1 = (\gamma_a q_a)$ and $q_4 = (1 - \gamma_a)q_a$.

This system can be modeled mathematically through the following differential equation:

$$\dot{h}_1 = -\frac{a_1}{A}\sqrt{2gh_1} + \frac{a_3}{A}\sqrt{2gh_3} + \frac{\gamma_a}{3600 \cdot A} \cdot q_a \tag{1}$$

$$\dot{h}_2 = -\frac{a_2}{A}\sqrt{2gh_2} + \frac{a_4}{A}\sqrt{2gh_4} + \frac{\gamma_b}{3600 \cdot A} \cdot q_b \tag{2}$$

$$\dot{h}_3 = -\frac{a_3}{A}\sqrt{2gh_3} + \frac{1-\gamma_b}{3600 \cdot A} q_a \tag{3}$$

$$\dot{h}_4 = -\frac{a_4}{A}\sqrt{2gh_4} + \frac{1-\gamma_a}{3600 \cdot A} q_b \tag{4}$$

This model will be subject to the following upper and lower constraints:

$$Q_j^{min} \leq q_j \leq Q_j^{max} \quad j = a, b \tag{5}$$

$$H_i^{min} \leq h_i \leq H_i^{max} \quad i = 1, 2, 3, 4 \tag{6}$$

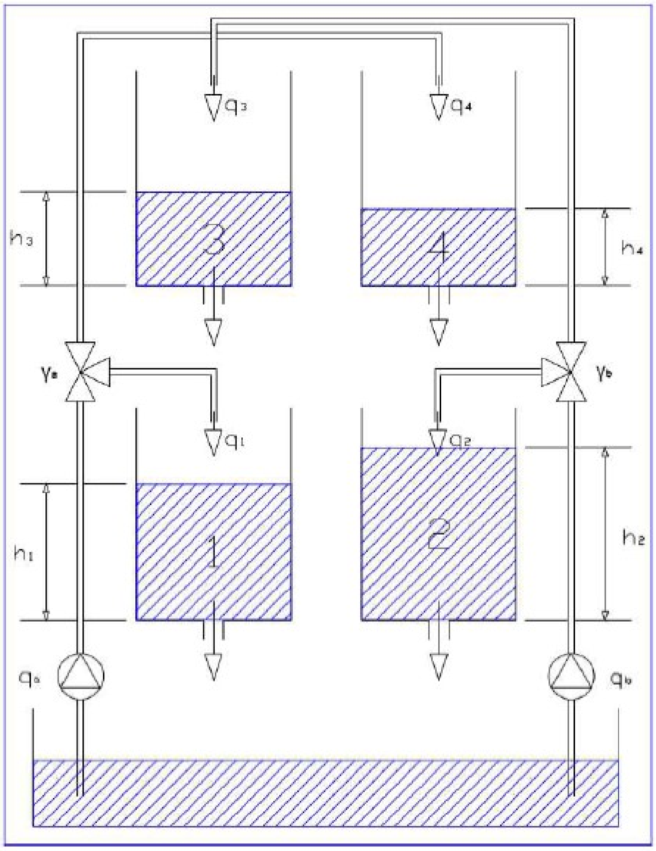


Figure 1: Four tank scheme

The values of the parameters of this model can be found in the figure 2.

This multivariable and continuous model cannot be used directly without software to carry out the corresponding integration. Therefore, the next step to be done is a discretization and linearization of the model. The sampling time for this discretization of the model would be 5 seconds. After the corresponding linearization and discretization, the resulting model would be the following:

$$h(k+1) = Ah(k) + Bq(k) \tag{7}$$
$$y(k+1) = Ch(k) + Dq(k) \tag{8}$$
$$\tag{9}$$

$$A = \begin{pmatrix} 0.9419 & 0 & 0.0401 & 0 \\ 0 & 0.9334 & 0 & 0.0380 \\ 0 & 0 & 0.9587 & 0 \\ 0 & 0 & 0 & 0.9607 \end{pmatrix}, \quad B = \begin{pmatrix} 0.0083 & 0.0003 \\ 0.0004 & 0.0111 \\ 0.0000 & 0.0168 \\ 0.0196 & 0.0000 \end{pmatrix} \tag{10}$$

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \quad D = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \tag{11}$$

	Value	Unit	Description
H_1^{max}	1.20	m	Maximum level of the tank 1
H_2^{max}	1.20	m	Maximum level of the tank 2
H_3^{max}	1.20	m	Maximum level of the tank 3
H_4^{max}	1.20	m	Maximum level of the tank 4
H_i^{min}	0.2	m	Minimum level of the tanks
Q_a^{max}	2.50	m^3/h	Maximal flow
Q_b^{max}	2.50	m^3/h	Maximal flow
Q_i^{min}	0	m^3/h	Minimal flow
a_1	$1.341e^{-4}$	m^2	Discharge constant of the tank 1
a_2	$1.533e^{-4}$	m^2	Discharge constant of the tank 2
a_3	$9.322e^{-5}$	m^2	Discharge constant of the tank 3
a_4	$9.061e^{-5}$	m^2	Discharge constant of the tank 4
A	0.03	m^2	Cross-section of all tanks
γ_a	0.3	n.u.	Parameter of the 3-ways valve
γ_b	0.4	n.u.	Parameter of the 3-ways valve
h_1^o	0.627	m	Equilibrium level tank 1
h_2^o	0.636	m	Equilibrium level tank 2
h_3^o	0.652	m	Equilibrium level tank 3
h_4^o	0.633	m	Equilibrium level tank 4
Q_a^o	1.6429	m^3/h	Equilibrium flow a
Q_b^o	2.0000	m^3/h	Equilibrium flow b

Figure 2: Parameters

The equilibrium point where the non-linear model of the tanks has been linearized would be $h^o = [h_1^o, h_2^o, h_3^o, h_4^o]$ and $Q^o = [Q_a^o, Q_b^o]$.

2 Homework implementation - Using xenomai RT Linux

2.1 Creating a Matlab model - Provided by the teacher

The creation in Matlab/simulink of a mathematical model in differential equations is recommended to see the operation of the non-linear plant and its linearized model. This operation will save problems or doubts when trying to create it in xenomai (RT Linux).

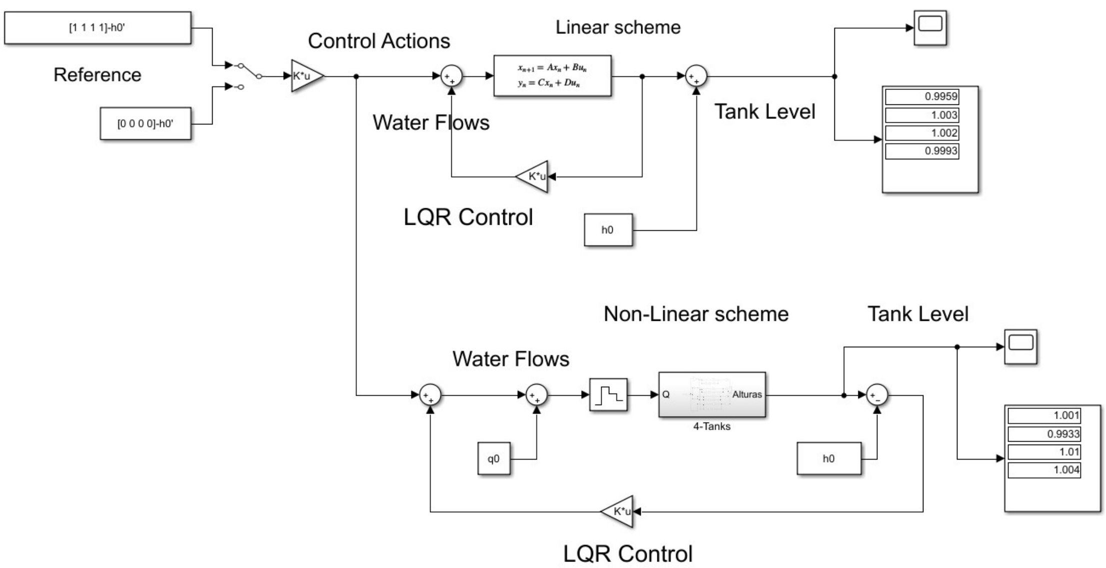


Figure 3: Matlab scheme.

Two files will be delivered in moodle:

- `Tank.m` : Matlab script file with all the parameters needed to run the simulink file.
- `Tanque1.slx` : Matlab/Simulink file with the four tank nonlinear/linear model.

Remember that the behavior of the approach in xenomai RT Linux should provide similar results to the one simulated in Matlab.

2.2 RT Process Schedule - Must be done by students

The practice will consist of the creation of an approach similar to the one done in Matlab/simulink. A real-time process will be created exclusively using the framework learned in class (xenomai RT Linux and the fork or exec commands can be used if the students need it) that will consist of the following periodic tasks:

- Task #1 (**4 Tanks Model**): This is a periodic task which must be executed every 5 seconds. It must do the following operation: $h(k+1) = Ah(k) + Bu(k)$. You must create the functions to multiply and sum matrices. See exercises of the Multithreading lesson to know the commands you could need.

- Task #2 (**controller**) : This is a periodic task that must calculate the control actions (water flows), taking into account the reference (desired tank levels) and the current output (Tank levels). The period of this task is 5 seconds but if the current tank levels are the same than at the previous sample time, the task doesn't change the control actions and the corresponding operations are not done.
- Task #3 (**Results**): This is a periodic task focused on print on screen the current value of the Tank levels and control actions, and store the values on a file. The file must be opened in main function before starting every tasks.

NOTE:

- Control actions and Tank levels must be global variables. In addition, every variable you need to use between task must be a global variable.

2.3 Delivering

The delivery of the practice will consist of the corresponding files `.c` encapsulated in a zip file together with a report of this practice. This Report will consist of the following parts(it will be delivered through Moodle):

- Description of the files `.c` exposing the pseudocode proposed in each of them.
- Results obtained from the execution of the assembled application and comparison with the simulink model.
- Final conclusions where the difficulties obtained during the realization of the practice are exposed.

Alterations depending on the DNI:

- Last DNI number between 0 and 5: Students will make the controller in the same way to the simulink model.
- Last DNI number between 6 and 9: Students will carry out the assembly with the model in open loop. The controller will enter the equilibrium water flows value for 5 sampling times and then enter a 5 % step over the previous value.