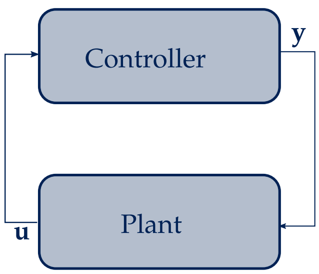
Networked control systems, basic model and timing analysis

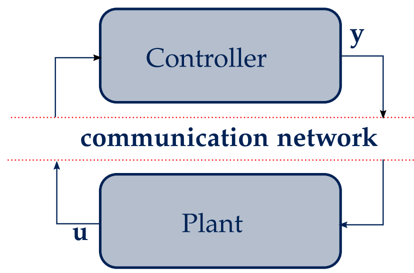
Manel Velasco and Pau Martí

October 2, 2016

1 Basic model for a networked control system

Whenever we work with a networked control system we experience some delays induced by the network. The nature of these delays may be random and unpredictable. This fact adds a lot of difficulty to the analysis of this kind of systems, we propose to use a simplified path to analyze them.





- Analyze the effects of the network in a time line
- Fix the delay to be regular and the sampling period to be constant
- Construct a model that contains these simplified effects

Afterward we will increase the difficulty of the analysis.

2 Time delays induced by the network

Figure (1) sketches a simplified version of what may happen in a network. In this example the control loop is isolated in the network, meaning that nobody else is going to interrupt nor delay the messages on the network.

Under these conditions the sequence of events in the control loop is described as:

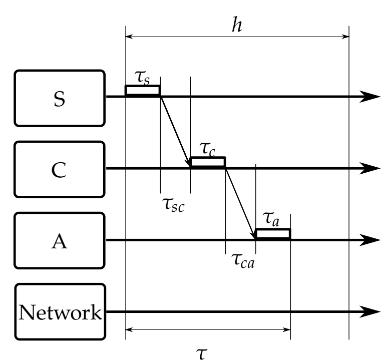


Figure 1: Networked system timeline analysis

- A time interrupt happens in the sensor processor, so the sensor starts the measuring, this instant of time may be labeled as kh. Where h is the sampling interval and this is occurring at the k^{th} sample.
- The sensor takes some time to perform the measure, so a small delay gets into de loop, τ_s.

- The sensor starts retransmitting the message that carries the information of the measurement to the controller, this also takes some time, τ_{sc}.
- The controller gets the message at time $kh + \tau_s + \tau_c a$ and starts the computation of the control action, which also takes some time in the processor, τ_c.
- Once the control action is computed the controller starts the transmission of the message that carries the control action information. This transmission takes some time in the network, namely tau_{ca}
- The actuator node get the message at time $kh + \tau_s + \tau_{sc} + \tau_c + \tau_{ca}$, but also takes some small time to apply it to the plant, and thus it induces a small delay τ_a into the control loop.

This sequence shows that there is an offset between the sampling instant and the actuation instant, which brakes the assumptions we did to derive the discrete time model in previous sections.

To overcome this problem we propose a new model.

3 *Delayed time model*

The proposed model should take into consideration the effects of the network in the control loop, the detailed timing of the model is shown in figure (**??**)

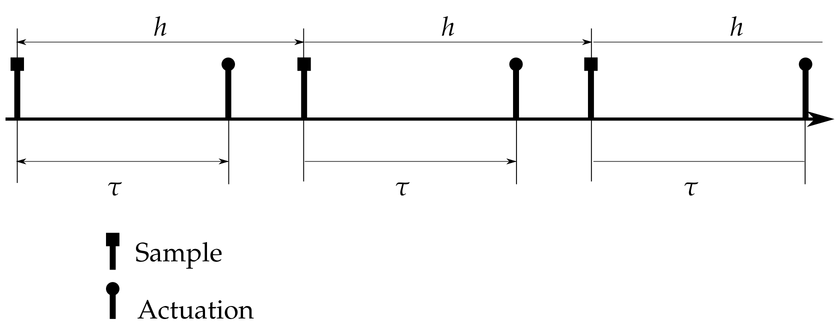


Figure 2: Time analysis of the networked control loop

As it can be seen the actuation and the sampling aren't performed in the same time instant, so we have to clarify how the control action is really aplied to the system. Next figure shows a random control signal as seen from the controller plant point of view.

It can be seen that the control signal changes in the middle of the period, so the assumpltios made in the deduction of the discrete time model are no longer valid. We see how there is a control signal constant from kh up to $kh + \tau$, and then a new interval from $kh + \tau$ up to $(k+1)h$ where the control signal is constant again.

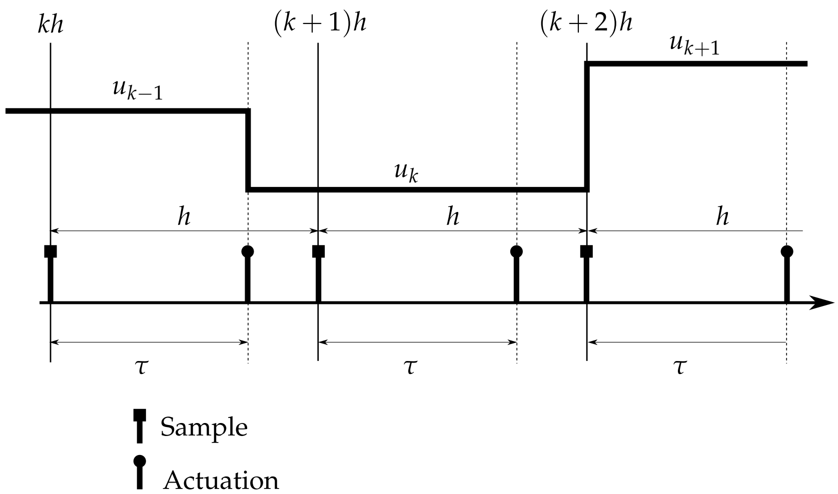


Figure 3: Control signal change instants due to the delays induced by the network

Applying the idea of chain of initial condition we can write these equations:

- From kh up to $kh+\tau$ we can write

$$x_{kh+\tau} = e^{A\tau}x_{kh} + \int_0^{\tau} e^{As}Bds\, u_{(k-1)h} \tag{1}$$

- From $kh+\tau$ up to $(k+1)h$ the exact solution is

$$x_{(k+1)h} = e^{A(h-\tau)}x_{kh+\tau} + \int_0^{h-\tau} e^{As}Bds\, u_k \tag{2}$$

If we substitute equation (1) into equation (2) we get:

$$x_{(k+1)h} = e^{Ah}x_{kh} + e^{A(h-\tau)}\int_0^{\tau} e^{As}Bds\, u_{(k-1)h} + \int_0^{h-\tau} e^{As}Bds\, u_k \tag{3}$$

And, using a more convenient notation we usually write[1] it in the form

$$x_{k+1} = \Phi(h)x_k + \Phi(h-\tau)\Gamma(\tau)u_{k-1} + \Gamma(h-\tau)u_k \tag{4}$$

As it can be seen that is a two step model. If we propose a controller with the form $u_k = Kx_k$ we get the equation

$$x_{k+1} = \Phi(h)x_k + \Phi(h-\tau)\Gamma(\tau)Kx_{k-1} + \Gamma(h-\tau)Kx_k \tag{5}$$

Where is impossible to get a closed loop expression, due to the fact that we cannot take common factor x_k. Going back to equation (7) we see that it may be rearranged as[2]

[1] We use the property $\Phi(h-\tau)\Phi(\tau) = \Phi(h)$. This property mey be only applied if the matrices e^{Ah} and $e^{A(h-\tau)}$ commute. In this case those matrices commute, try to prove that

[2] Just perform the matrix multiplications to get two equations, the first one is (5) and the second one just states that $u_k = u_k$

$$\begin{bmatrix} x_{k+1} \\ u_k \end{bmatrix} = \begin{bmatrix} \Phi(h) & \Phi(h-\tau)\Gamma(\tau) \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ u_{k-1} \end{bmatrix} + \begin{bmatrix} \Gamma(h-\tau) \\ 1 \end{bmatrix} u_k \qquad (6)$$

This lifted[3] system collects the system dynamics and the dynamics of the control signal. It has an extra pole, so some care must be taken when designing the controller. The standard way to design controllers works flawlessly with these lifted systems.

[3] A matrix of matrices operates as this example. Suppose R and T to be

$$R = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$T = \begin{bmatrix} e \\ f \end{bmatrix}$$

Then the matrix of matrices

$$S = \begin{bmatrix} R & T \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} a & b & e \\ c & d & f \\ 0 & 0 & 0 \end{bmatrix}$$

Example 3.1. Lets use the dobule integrator model

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} x$$

We want to design a controller that sets the discret eigen-values of the system to $z_{1,2} = \pm 0.5$ with a sampling period of $h = 0.01$s, we know that there is a small delay induced by the network of $\tau = 0.005$s

Solution:

Taking advantage of our knowledge of Matlab we can construct directly the delayed system and construct the controller

---

Matlab code:

```
A=[0 1;0 0];                  %system matrix
B=[0;1];                      %Input matrix
h=0.01;                       %sampling period
tau=0.005
[phi_h,gamma_h]=c2d(A,B,h); %Discrete model
[phi_t,gamma_t]=c2d(A,B,h); %Discrete model
[phi_ht,gamma_ht]=c2d(A,B,h);%Discrete model
Phi_extended=[phi_h phi_ht*gamma_t;
             0  0      0      ];
Gamma_extended=[gamma_ht;1];
z=[0.5, -0.5 , 0];            %Discrete poles
k=acker(Phi_extended,Gamma_extended,z) %Controler
```

Result:

```
Phi_extended =
[1.0000    0.0100    0.0002]
[     0    1.0000    0.0100]
[     0         0         0]
Gamma_extended =
[0.0001]
[0.0100]
[1.0000]
k =
1.0e+03 *
[3.7500    0.0813    0.0010]
```

4 *Induced delay longer than the sampling period*

Now let us consider the case when τ is longer than h. The number of possible cases grows as τ gets longer and longer. To develop the model we are going to consider the simplest case, the one where $h > \tau > 2h$. The associated timing diagram of the sampling and the control actions may be observer in next figure

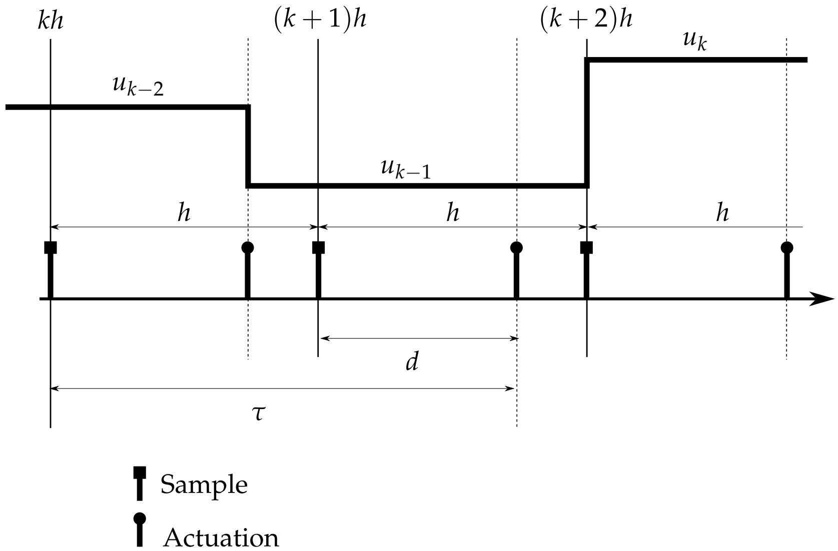


Figure 4: Control signal change instants due to the delays induced by the network. In this case the delay is larger than the sampling period, and thus, $h > \tau > 2h$

We define[4] $d = \tau - h$. As it can be seen in the diagram of figure (**??**), at sampling instant kh the control law acting on the system is u_{k-2}, taking thisinto account and following the philosophy of chain of initial conditions we can state[5] that

$$x_{k+1} = \Phi(h)x_k + \Phi(h-d)\Gamma(d)u_{k-2} + \Gamma(h-d)u_{k-1} \tag{7}$$

Once again the control signal replaced by its expression will result into a strange pattern which can be solved using a lifted system. In this case will be

$$\begin{bmatrix} x_{k+1} \\ u_{k-1} \\ u_k \end{bmatrix} = \begin{bmatrix} \Phi(h) & \Phi(h-d)\Gamma(d) & \Gamma(h-d) \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ u_{k-2} \\ u_{k-1} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_k \tag{8}$$

So new dimensions appear as the delay grows. This can be easily extended to longer delays.

[4] For longer delays this is defined as $d = \tau - nh$, and n is the number of full sampling steps that the delay covers

[5] Try to prove this yourself

Example 4.1. Lets use the dobule integrator model

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$
$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} x$$

We want to design a controller that sets the discret eigen-values of the system to $z_{1,2} = \pm 0.5$ with a sampling period of $h = 0.01$s, we know that there is a small delay induced by the network of $\tau = 0.015$s

Solution:

Taking advantage of our knowledge of Matlab we can construct directly the delayed system and construct the controller, in thid case $n = 1$ and $d = 0.005$

Matlab code:

```
A=[0 1;0 0];                   %system matrix
B=[0;1];                       %Input matrix
h=0.01;                        %sampling period
tau=0.005
[phi_h,gamma_h]=c2d(A,B,h); %Discrete model
[phi_t,gamma_t]=c2d(A,B,h); %Discrete model
[phi_ht,gamma_ht]=c2d(A,B,h);%Discrete model
Phi_extended=[phi_h phi_ht*gamma_t gamma_ht;
              0  0       0          1
              0  0       0          0    ];
Gamma_extended=[0 ; 0 ; 0 ; 1];
z=[0.5, -0.5 , 0 , 0 ];          %Discrete poles
k=acker(Phi_extended,Gamma_extended,z) %Controler
```

Result:

```
Phi_extended =
[1.0000    0.0100    0.0002    0.0001]
[     0    1.0000    0.0100    0.0100]
[     0         0         0    1.0000]
[     0         0         0         0]
Gamma_extended =
[0]
[0]
[0]
[1]
k =
1.0e+03 *
[3.7500    0.1188    0.0014    0.0020]
```