Programming Intelligent Physical Systems

Lecture 3

Samarjit Chakraborty

Technical University of Munich
Chair of Real-Time Computer Systems (RCS)

04 August, 2019

Programming Intelligent Physical Systems

Integration of computational elements with physical processes.
CPS = Embedded Systems + Control Systems

Goals:

- The system is intelligent
- The system is adaptive
- The system is certifiable
- Example: If a camera in an industrial robot is replaced by a new camera, the system can automatically adjust to this change and exploit the better capabilities of the new camera
- Example: If there is a change in the mechanical sub-system, the control strategy is automatically adapted to fit this new system
- Example: If there is increased wear and tear of certain components (e.g., drill bits) or increased vibration, the production strategy is automatically adapted

Outline of this lecture

In today's lecture we will discuss:

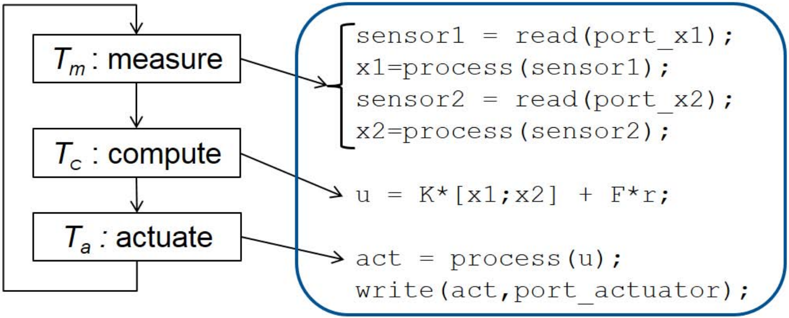
- Controller implementation on an embedded platform
- Controller Design for discrete-time systems with time delay

Control applications

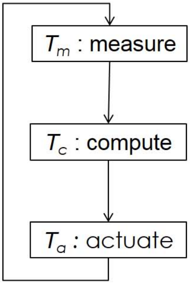
An embedded controller can be implemented using 3 tasks:

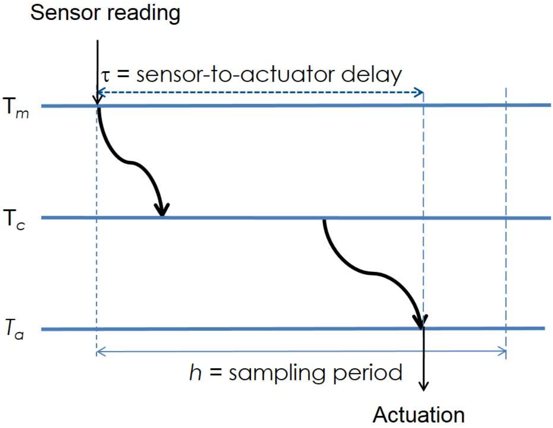
- A sensor task (T_m) reads sensor data and process them to extract state information. Typically, A/D conversion and signal/image processing are performed in this task.
- A controller task (T_c) implements the control law and computes the control input. The execution time of this task depends on the complexity of the control algorithm.
- An actuator task (T_a) writes the control input onto the actuator to be applied to the plant. Typically, D/A conversion and post-processing is done in this task.

Controller implementation



Controller implementation

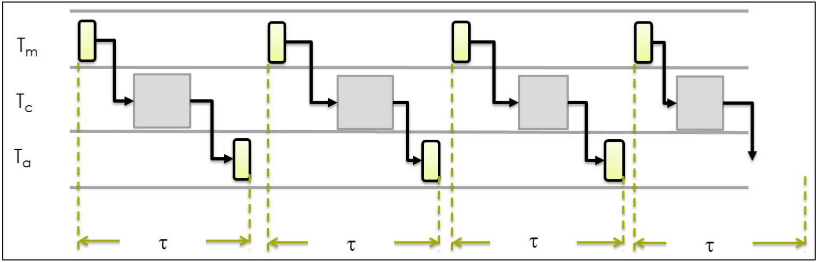




Ideal design assumes $\tau = 0$ or $\tau << h$.

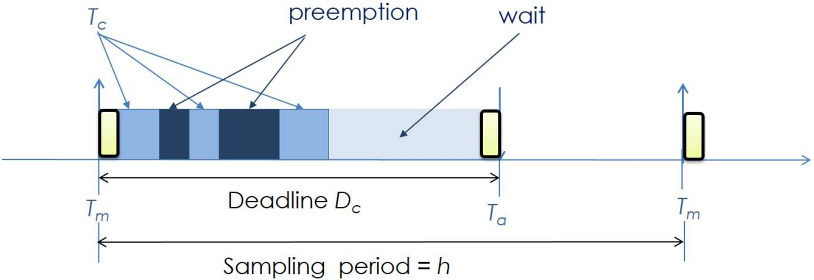
Task triggering

- In general, T_m and T_a consume negligible computational time and are time-triggered.
- T_c needs finite computation time and is event-triggered and preemptive.
- When multiple tasks are running on a processor, T_c can be preempted by a higher priority task.



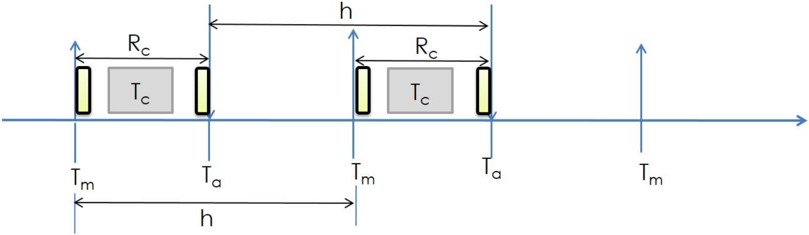
Sensor-to-actuator delay: τ

Control task model – constant delay



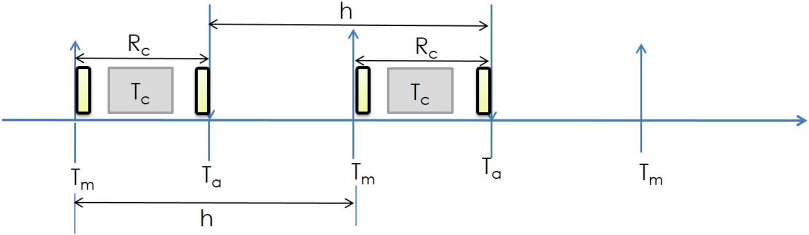
Sensor-to-actuator delay $\tau \approx D_c$

Control task model – constant delay



- T_m is triggered periodically with a period equal to the sampling period h. Schedule for T_m is assumed as $\{0, 0, h\}$, i.e., periodic with zero offset, negligible execution time and period h.
- T_a is also triggered periodically with the same period h. Schedule for T_a is assumed as $\{D_c, 0, h\}$, i.e., periodic with constant offset D_c, negligible execution time and period h.
- T_c is executed in between T_m and T_a.

Control task model – constant delay



- T_c is preemptive.
- Response time of T_c is R_c.
- The time difference between T_m and T_a is the deadline D_c of T_c.
- Sensor-to-actuator delay is $\tau = D_c$ in all samples and the task should be scheduled such that $R_c < D_c$.
- The control task is characterized by $T_c \sim \{h, D_c, e_c\}$ where (i) h is the sampling period of the control application, (ii) D_c is

System stability and control performance

- Deadline D_c for a control task T_c are often firm rather than hard.
  - Okay to miss a few deadlines, but not too many in a row.
  - And it depends on what happens if the deadline is missed.
    - Task is allowed to complete late.
    - Task is aborted at the deadline.

Controller design for delayed discrete-time systems

Controller design steps for Case A: $D_c < h$

Continuous-time model

$\Downarrow$ ZOH sampling with period h and constant sensor-to-actuator delay D_c

Step I — New discrete-time model : Sampled-data model

$\Downarrow$

Step II — Controller design based on the sampled-data model

$\Downarrow$

Objectives

(i) Place system poles

(ii) Achieve $y \rightarrow r$ as $t \rightarrow \infty$

$$\dot{x} = Ax + Bu$$
$$y = Cx$$

$$x[k+1] = f_1(x[k], u[k])$$

$$y[k] = f_2(x[k])$$

$$u[k] = f(x[.])$$

Recall: Ideal discrete-time case

Continuous-time model

$\Downarrow$ ZOH sampling with period h and $D_c = 0$

Discrete-time model

$\Downarrow$

Controller design based on the discrete-time model

$\Downarrow$

Objectives
(i) Place system poles
(ii) Achieve $y \rightarrow r$ as $t \rightarrow \infty$
(iii) Design K and F

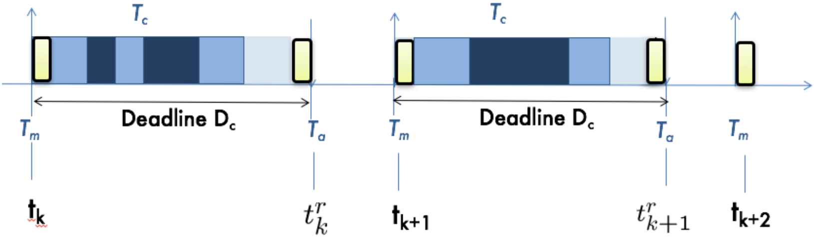
$$\dot{x} = Ax + Bu$$
$$y = Cx$$

$$x[k+1] = \phi x[k] + \Gamma u[k]$$
$$y[k] = Cx[k]$$

$$u[k] = Kx[k] + Fr$$

Step I:
Derivation of sampled-data model with constant sensor-to-actuator delay Dc

Timing properties



Constant sensor-to-actuator delay

$$t_k^r = t_k + D_c$$
$$t_{k+1}^r = t_{k+1} + D_c$$
$$\ldots$$

Sampling period h

$$t_{k+1} = t_k + h$$
$$t_{k+2} = t_{k+1} + h$$
$$\ldots$$

Signals

- Measurement is done in every sampling instant. Therefore, it is essentially assumed that the states are constants between two consecutive measurements , i.e.,

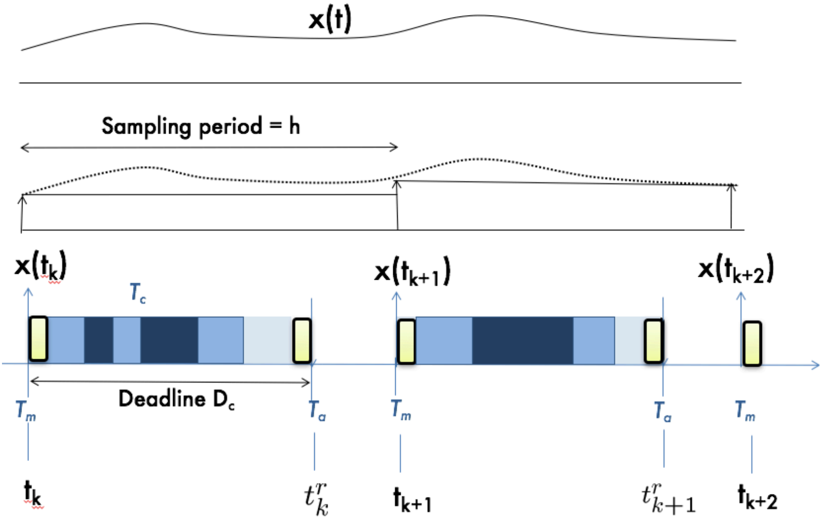
$$x(t) = x(t_k) = x[k] \quad \text{for} \quad t_k \leq t \leq t_{k+1}$$

- The input signal is hold constant for one sampling interval

$$u(t) = u(t_k) = u[k] \quad \text{for} \quad t_k^r \leq t \leq t_{k+1}^r$$
$$u(t) = u(t_{k+1}) = u[k+1] \quad \text{for} \quad t_{k+1}^r \leq t \leq t_{k+2}^r$$
$$\ldots$$

- A control input is updated once in every sampling interval because,

$$t_{k+1}^r - t_k^r = h$$


x(t)
Sampling period = h
x(t_k)
x(t_k+1)
x(t_k+2)
T_c
Deadline D_c
T_m
T_a
T_m
T_a
T_m
t_k
t_k^r
t_k+1
t_{k+1}^r
t_k+2

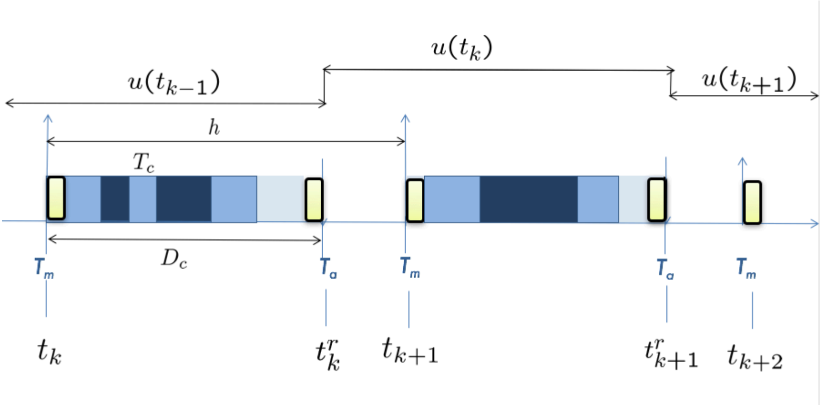
Zero-order hold operation

- The $u(t_k)$ is computed based on the latest measurement $x(t_k)$

$$u(t_k) = f(x(t_k))$$

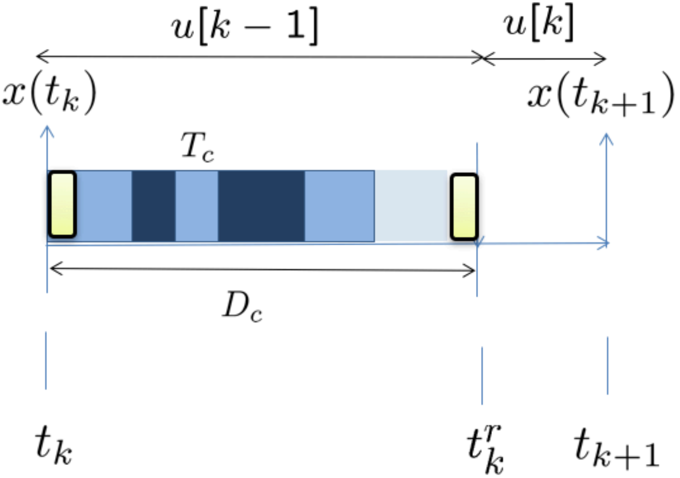
- $u(t_k)$ is applied at $t = (t_k + D_c) = t_k^r$
- In ideal implementation, $u(t_k)$ is applied at $t = t_k^r$
- Due to finite sensor-to-actuator delay, the input value is updated after D_c time
- Between $t_{k-1}^r \leq t \leq t_k^r$, the previous control input is hold,

$$u(t) = u(t_{k-1}) = u[k-1]$$



$u(t_k)$
$u(t_{k-1})$
$u(t_{k+1})$
h
T_c
D_c
T_m
T_a
T_m
T_a
T_m
t_k
t_k^r
t_{k+1}
t_{k+1}^r
t_{k+2}

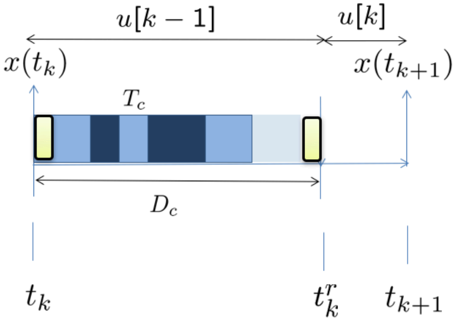
What is happening within one sampling period



Recall

$$\dot{x} = Ax + Bu$$
$$y = Cx$$
$$\Downarrow$$
$$x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau$$
$$y(t) = Cx(t)$$

Model derivation



$$x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau$$
$$y(t) = Cx(t)$$

$$\implies$$

$$x(0) = x(t_k)$$
$$x(t) = x(t_{k+1})$$

$$x(t_{k+1}) = e^{A(t_{k+1}-t_k)}x(t_k) + \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)}Bu(\tau)d\tau$$

$$\Downarrow$$

$$u(\tau) = u[k-1] \quad \text{for} \quad t_k \leq t \leq t_k^r$$
$$u(\tau) = u[k] \quad \text{for} \quad t_k^r \leq t \leq t_{k+1}$$
$$t_{k+1} - t_k = h$$
$$x(t_{k+1}) = x[k+1]$$
$$x(t_k) = x[k]$$

$$\Downarrow$$

$$x[k+1] = e^{Ah}x[k] + \int_{t_k}^{t_k^r} e^{A(t_{k+1}-\tau)}Bd\tau.u[k-1] +$$
$$+ \int_{t_k^r}^{t_{k+1}} e^{A(t_{k+1}-\tau)}Bd\tau.u[k]$$

$$x[k+1] = e^{Ah}x[k] + \int_{t_k}^{t_k^r} e^{A(t_{k+1}-\tau)}Bd\tau.u[k-1]+$$
$$+\int_{t_k^r}^{t_{k+1}} e^{A(t_{k+1}-\tau)}Bd\tau.u[k]$$

$$\Downarrow$$

$$x[k+1] = e^{Ah}x[k] + \int_{h-D_c}^{h} e^{As}Bds.u[k-1]+$$
$$+\int_{0}^{h-D_c} e^{As}Bds.u[k]$$

$$\Downarrow$$

$$x[k+1] = \phi x[k] + \Gamma_1(D_c)u[k-1] + \Gamma_0(D_c)u[k]$$

$$\phi = e^{A}h$$
$$\Gamma_1(D_c) = \int_{h-D_c}^{h} e^{As}Bds$$
$$\Gamma_0(D_c) = \int_{0}^{h-D_c} e^{As}Bds$$

Sampled-data model

$$\dot{x} = Ax + Bu$$
$$y = Cx$$

$\Longrightarrow$ $\boxed{\text{Continuous-time model}}$

$\Big\Downarrow$

ZOH sampling with period h and constant sensor to actuator delay D_c

$$x[k+1] = \phi x[k] + \Gamma_1(D_c)u[k-1] + \Gamma_0(D_c)u[k]$$
$$y[k] = Cx[k]$$

$\Longrightarrow$ $\boxed{\text{Sampled-data model}}$

where,

$$\phi = e^{A}h$$
$$\Gamma_1(D_c) = \int_{h-D_c}^{h} e^{As} B ds$$
$$\Gamma_0(D_c) = \int_{0}^{h-D_c} e^{As} B ds$$

End of Step 1

Example 1

Consider the following continuous-time system -

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 37 & 7.5 \end{bmatrix} x + \begin{bmatrix} 0 \\ 6450 \end{bmatrix} u, \qquad y = \begin{bmatrix} 1 & 0 \end{bmatrix} x$$

(a) Compute the system model considering ZOH sampling with sampling period $h = 0.01$ sec and a constant sensor-to-actuator delay $D_c = 0.005$sec.

$$\phi = e^{Ah} \approx I + Ah = \begin{bmatrix} 1 & 0.01 \\ 0.37 & 1.075 \end{bmatrix}$$

$$\Gamma_1(D_c) = \int_{h-D_c}^{h} e^{As} B ds = A^{-1}(e^{Ah} - e^{A(h-D_c)})B$$

$$\approx A^{-1}(I + Ah - I - A(h - D_c))B$$

$$= D_c B = \begin{bmatrix} 0 \\ 32.25 \end{bmatrix}$$

$$\Gamma_0(D_c) = (h - D_c)B = \begin{bmatrix} 0 \\ 32.25 \end{bmatrix}$$

Step II:

Controller design based on sampled-data model with constant sensor-to-actuator delay D_c

System model with $D_c < h$

$$x[k+1] = \phi x[k] + \mathit{\Gamma}_1(D_c)u[k-1] + \mathit{\Gamma}_0(D_c)u[k]$$
$$y[k] = Cx[k]$$

$$\phi = e^A h$$

$$\mathit{\Gamma}_1(D_c) = \int_{h-D_c}^{h} e^{As} B ds$$

$$\mathit{\Gamma}_0(D_c) = \int_{0}^{h-D_c} e^{As} B ds$$

Augmented system

- We define new system states:

$$z[k] = \begin{bmatrix} x[k] \\ u[k-1] \end{bmatrix}$$

- With the new definition of states, the state-space becomes

$$\begin{aligned} z[k+1] &= \phi_{aug} z[k] + \Gamma_{aug} u[k] \\ y[k] &= C_{aug} z[k] \end{aligned}$$

where the augmented matrices are defined as follows

$$\phi_{aug} = \begin{bmatrix} \phi & \Gamma_1(D_c) \\ 0 & 0 \end{bmatrix}, \quad \Gamma_{aug} = \begin{bmatrix} \Gamma_0(D_c) \\ I \end{bmatrix}$$

$$C_{avg} = \begin{bmatrix} C & 0 \end{bmatrix}$$

Example 2

Consider the continuous-time system (voltage stabilizer)

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

(a) Compute the system model considering ZOH sampling with sampling period $h = 0.01$ sec and a constant sensor-to-actuator delay $D_c = 0.005$sec.

$$\phi = e^{Ah} \approx \mathbf{I} + Ah = \begin{bmatrix} 1 & 0.001 \\ -0.001 & 0.999 \end{bmatrix}$$

$$\Gamma_1(D_c) \approx D_c B = \begin{bmatrix} 0 \\ 0.0005 \end{bmatrix}$$

$$\Gamma_0(D_c) \approx (h - D_c) B = \begin{bmatrix} 0 \\ 0.0005 \end{bmatrix}$$

- Augmented system has the following

$$\phi_{aug} = \begin{bmatrix} \phi & \Gamma_1(D_c) \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0.001 & 0 \\ -0.001 & 0.999 & 0.0005 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\Gamma_{aug} = \begin{bmatrix} \Gamma_0(D_c) \\ I \end{bmatrix} = \begin{bmatrix} 0 \\ 0.0005 \\ 1 \end{bmatrix}$$

$$C_{aug} = \begin{bmatrix} C & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

- We could see that the augmented system has a higher dimension compared to the original system

Controller design for $D_c < h$

- Given system:
  $z[k+1] = \phi_{aug} z[k] + \Gamma_{aug} u[k]$
  $y[k] = C_{aug} z[k]$
- Control law: $u[k] = Kz[k] + Fr$

Objectives:
(i) Place system poles
(ii) Design K and F
(iii) Achieve $y \rightarrow r$ as $t \rightarrow \infty$

$$\Downarrow$$

1. Check controllability of the augmented system $(\phi_{aug}, \Gamma_{aug})$. To be controllable, γ_{aug} must be invertible where

$$\gamma_{aug} = \begin{bmatrix} \Gamma_{aug} & \phi_{aug}\Gamma_{aug} & \phi_{aug}^2\Gamma_{aug} & \cdots & \phi_{aug}^{n-1}\Gamma_{aug} \end{bmatrix}$$

2. Apply Ackermann's formula $K = -\begin{bmatrix} 0 & 0 & \cdots & 1 \end{bmatrix} \gamma_{aug}^{-1} H(\phi_{aug})$ where
   $H(\phi_{aug}) = (\phi_{aug} - \alpha_1 \mathbf{I})(\phi_{aug} - \alpha_2 \mathbf{I}) \cdots (\phi_{aug} - \alpha_n \mathbf{I})$
   and $\alpha_1, \alpha_2, \cdots, \alpha_n$ are the poles of the augmented system.
3. Feedforward gain $F = \frac{1}{C_{aug}(\mathbf{I} - \phi_{aug} - \Gamma_{aug} K)^{-1} \Gamma_{aug}}$

Summary: Overall design for $D_c < h$

Continuous-time model

$$\dot{x} = Ax + Bu$$
$$y = Cx$$

$$\Downarrow$$

Sampled-data model

$$x[k+1] = \phi x[k] + \Gamma_1(D_c)u[k-1] + \Gamma_0(D_c)u[k]$$
$$y[k] = Cx[k]$$

$$\Downarrow$$

Augmented system

$$z[k+1] = \phi_{aug} z[k] + \Gamma_{aug} u[k]$$
$$y[k] = C_{aug} z[k]$$

$$\Downarrow$$

Controller gains

$$u[k] = Kz[k] + Fr$$
$$K = -\begin{bmatrix} 0 & 0 & \cdots & 1 \end{bmatrix} \gamma_{aug}^{-1} H(\phi_{aug})$$
$$F = \frac{1}{C_{aug}(\mathbf{I} - \phi_{aug} - \Gamma_{aug} K)^{-\mathbf{1}} \Gamma_{aug}}$$

Example 3

Consider the following continuous-time system:

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

(a) Compute the system model considering ZOH with sampling period $h = 0.001$s and a constant sensor-to-actuator delay $D_c = 0.0005$s.

Solution: The sampled-data model is given by

$$x[k+1] = \phi x[k] + \Gamma_1(D_c)u[k-1] + \Gamma_0(D_c)u[k]$$

where,

$$\phi = e^{Ah} \approx \mathbf{I} + Ah = \begin{bmatrix} 1 & 0.001 \\ -0.001 & 0.999 \end{bmatrix}$$

$$\Gamma_1(D_c) \approx D_c B = \begin{bmatrix} 0 \\ 0.0005 \end{bmatrix}$$

$$\Gamma_0(D_c) \approx (h - D_c)B = \begin{bmatrix} 0 \\ 0.0005 \end{bmatrix}$$

Example 3

(b) Using $x_1(0) = 45$ and $x_2(0) = 0$, design u such that $y \to 90$ as $t \to \infty$.

Solution: We choose the new augmented states

$$z[k] = \begin{bmatrix} x[k] \\ u[k-1] \end{bmatrix}$$

The augmented system with new system states is

$$z[k+1] = \phi_{aug} z[k] + \Gamma_{aug} u[k], \qquad y[k] = C_{aug} z[k].$$

where,

$$\phi_{aug} = \begin{bmatrix} \phi & \Gamma_1(D_c) \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0.001 & 0 \\ -0.001 & 0.999 & 0.0005 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\Gamma_{aug} = \begin{bmatrix} \Gamma_0(D_c) \\ \mathbf{I} \end{bmatrix} = \begin{bmatrix} 0 \\ 0.0005 \\ 1 \end{bmatrix}$$

$$C_{aug} = \begin{bmatrix} C & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

Example 3

The controllability matrix of the augmented system is given by

$$\gamma_{aug} = \begin{bmatrix} \Gamma_{aug} & \phi_{aug}\Gamma_{aug} & \phi_{aug}^2\Gamma_{aug} \end{bmatrix}$$

$$= \begin{bmatrix} 0.000000124979167 & 0.000000999458333 & 0.000001997958334 \\ 0.000499875000003 & 0.0009990 & 0.000998000001416 \\ 1 & 0 & 0 \end{bmatrix}$$

And since $\det(\gamma_{aug} \neq 0)$, the augmented system is controllable.

Example 3

We apply a control input

$$u[k] = Kz[k] + Fr$$

The feedback gain is designed using Ackermann's formula. Towards this, we first choose the closed loop system poles

$$\begin{bmatrix} 0.9 & 0.9 & 0.9 \end{bmatrix}.$$

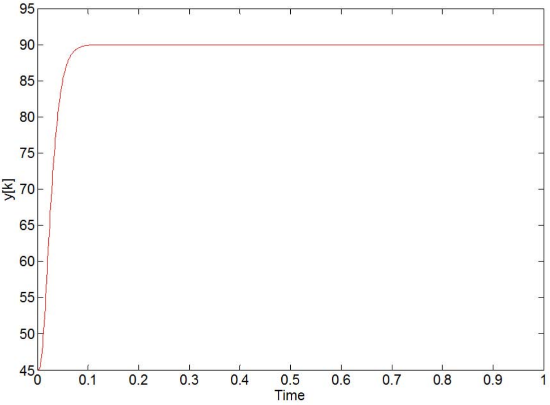
$$H(\phi_{aug}) = (\phi_{aug} - 0.9\mathbf{I})^3 = \begin{bmatrix} 0.0010 & 0.0000 & -0.0000 \\ -0.0000 & 0.0010 & 0.0004 \\ 0 & 0 & -0.7290 \end{bmatrix}$$

$$K = -\begin{bmatrix} 0 & 0 & \cdots & 1 \end{bmatrix} \gamma_{aug}^{-1} H(\phi_{aug}) = \begin{bmatrix} -1000.2 & -28.7 & 0.7 \end{bmatrix}$$

$$F = \frac{1}{C_{aug}(\mathbf{I} - \phi_{aug} - \Gamma_{aug} K)^{-1} \Gamma_{aug}} = 1000.5$$

Example 3

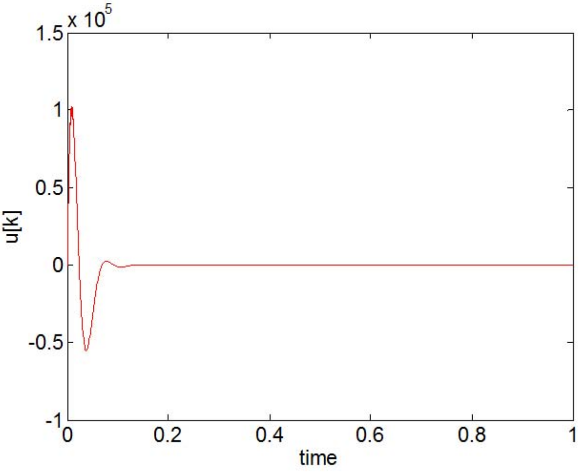
We apply the above designed feedback and feedforward gains and obtain the following response



Settling time is arround 0.1 seconds.

Example 3

Plot of input signal



The input signal requirement is given by $\max u[k] = 102380$.

Example 3

(c) Repeat part (b) assuming that the designer does not know about D_c and assumes $D_c = 0$. Plot the system response.

Solution: The discrete-time system is given by

$$x[k+1] = \phi x[k] + \Gamma u[k], \qquad y[k] = Cx[k].$$

where,

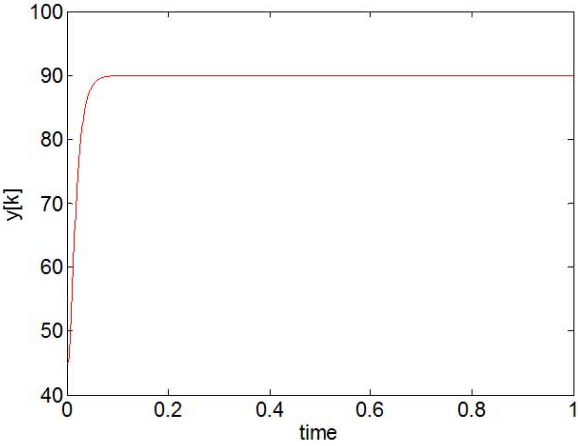
$$\phi = \begin{bmatrix} 1.0 & 0.001 \\ -0.001 & 0.999 \end{bmatrix}, \qquad \Gamma = \begin{bmatrix} 0 \\ 0.001 \end{bmatrix}$$

$$\text{For, } u[k] = Kx[k] + Fr \text{ and } \alpha = \begin{bmatrix} 0.9 & 0.9 \end{bmatrix}$$

$$\text{we get, } K = \begin{bmatrix} -10004 & -194 \end{bmatrix}, \quad F = 10005.$$

Example 3

We apply the above designed feedback and feedforward gains and obtain the following response



Example 3

(d) Redesign the controller assuming a period $h = 0.5$s and a constant sensor-to-actuator delay $D_c = 0.4$s. And plot the system response.

Solution: We first obtain the augmented system dynamics as follows:

$$z[k+1] = \phi_{aug} z[k] + \Gamma_{aug} u[k], \qquad y[k] = C_{aug} z[k].$$

Subsequently, we follow the design similar to part (b).
For, $\alpha = \begin{bmatrix} 0.2 & 0.2 & 0.2 \end{bmatrix}$

$$H(\phi_{aug}) = (\phi_{aug} - 0.2\mathbf{I})^3 = \begin{bmatrix} 0.0932 & 0.2506 & 0.1108 \\ -0.2506 & -0.1574 & -0.0489 \\ 0 & 0 & -0.0080 \end{bmatrix}$$

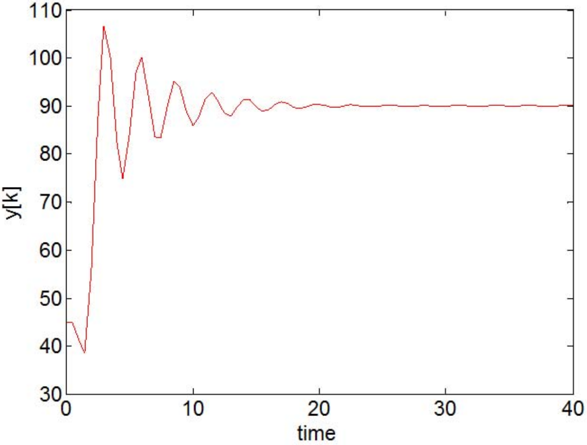
$$K = -\begin{bmatrix} 0 & 0 & \cdots & 1 \end{bmatrix} \gamma_{aug}^{-1} H(\phi_{aug}) = \begin{bmatrix} -0.9993 & -1.5905 & -0.6579 \end{bmatrix}$$

$$F = \frac{1}{C_{aug}(\mathbf{I} - \phi_a ug - \Gamma_{aug} K)^{-1} \Gamma_{aug}} = 2.65$$

$$u[k] = Kz[k] + Fr$$

Example 3

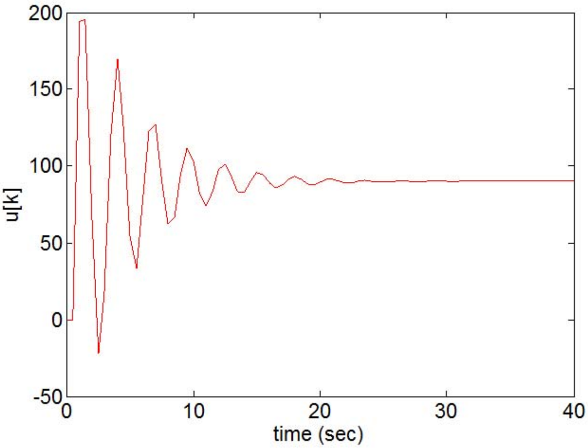
We obtain the following system response



Settling time is around 20s.

Example 3

Plot of input signal



$$\text{Maximum input signal } \max u[k] = 195.60.$$

Example 3

(e) Repeat part (d) assuming that the designer does not know about D_c and assumes $D_c = 0$. Plot the system response.

Solution: The discrete-time system can be obtained as

$$x[k+1] = \phi x[k] + \Gamma u[k], \qquad y[k] = Cx[k].$$

where,

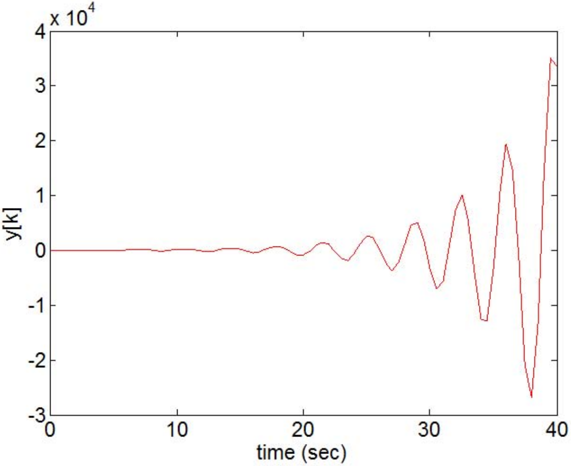
$$\phi = \begin{bmatrix} 0.8956 & 0.3773 \\ -0.3773 & 0.5182 \end{bmatrix}, \qquad \Gamma = \begin{bmatrix} 0.1044 \\ 0.3773 \end{bmatrix}$$

$$\text{For, } u[k] = Kx[k] + Fr \text{ and } \alpha = \begin{bmatrix} 0.2 & 0.2 \end{bmatrix}$$

$$\text{we get, } K = \begin{bmatrix} -2.3215 & -2.0445 \end{bmatrix}, \quad F = 3.3215.$$

Example 3

We obtain the following system response by ignoring the effect of delay



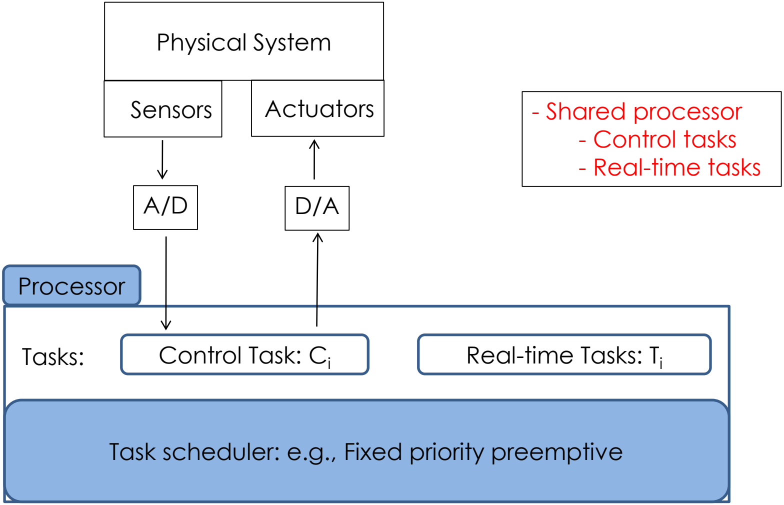
Clearly, the system is unstable if the design ignores the effect of delay.

Example 3: Conclusion

- The effect of sensor-to-actuator delay is prominent when the sampling period is longer. Since the sampling period is very short in part (a) and (b), part (c) shows that the effect of sensor-to-actuator delay can be ignored. However, since the sampling period is longer in part (d) and (e), the system gets unstable when the effect of delay is ignored.
- The important design parameters are the following
  - Sampling period (h)
  - System poles (α)
  - Maximum input signal requirement $\max u[k]$
  - System settling time
  - Sensor-to-actuator delay (D_c)

. . . coming back to implementation onto single-processor platform. . .

Recall: single processor setting



Recall: controller design

Continuous-time model	$$\dot{x} = Ax + Bu$$ $$y = Cx$$
	$$\Downarrow$$
Sampled-data model	$$x[k+1] = \phi x[k] + \Gamma_1(D_c)u[k-1] + \Gamma_0(D_c)u[k]$$ $$y[k] = Cx[k]$$
	$$\Downarrow$$
Augmented system	$$z[k+1] = \phi_{aug} z[k] + \Gamma_{aug} u[k]$$ $$y[k] = C_{aug} z[k]$$
	$$\Downarrow$$
Controller gains	$$u[k] = Kz[k] + Fr$$ $$K = -\begin{bmatrix} 0 & 0 & \cdots & 1 \end{bmatrix} \gamma_{aug}^{-1} H(\phi_{aug})$$ $$F = \frac{1}{C_{aug}(\mathbf{I} - \phi_{aug} - \Gamma_{aug} K)^{-1} \Gamma_{aug}}$$

Recall: schedulability analysis for single processor

- Response time with fixed priority preemptive scheduling for the given task set is given by:

$$R_i = e_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{R_i}{p_j} \right\rceil e_j$$

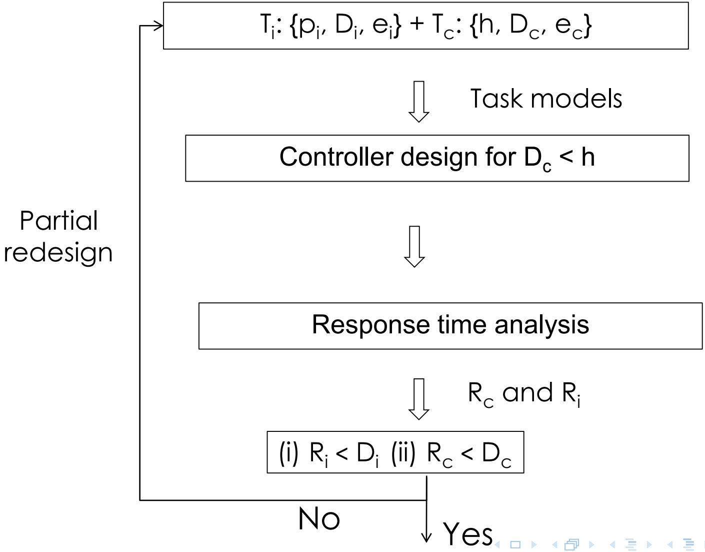
- Recurrence relation can be solved iteratively

$$R_i^{n+1} = e_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{R_i^n}{p_j} \right\rceil e_j$$

starting with $R_i^0 = 0$

- Schedulability test implies worst-case response time must be smaller than the deadline, i.e., $R_i = D_i$

Overall design steps



Illustrative design example from automotive: Implementation of cruise control system onto an ECU

Problem Description

- Consider the following dynamics of cruise control system

$$\dot{v}(t) = Av(t) + Bu(t), y(t) = Cv(t),$$

$$v(t) = \begin{bmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \end{bmatrix}, A = \begin{bmatrix} 0 & 1.0 & 0 \\ 0 & 0 & 1.0 \\ -6.05 & -5.29 & -0.24 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 2.48 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

- It receives the reference or the commanded vehicle's speed from the driver and regulates the speed following the driver's command. Based on the reference speed and the feedback signals, the cruise control system regulates the vehicle's speed by adjusting the engine throttle angle to increase or decrease the engine drive force.
- The state $v_1(t)$ captures the speed of the vehicle and $u(t)$ is the engine throttle angle. The objective is to choose $u(t)$ such that

- The cruise controller has to be implemented on an Electronic Control Unit (ECU) where a number of other period real-time tasks are also running. The real-time tasks are characterized as follows:

Tasks	p_i(ms)	D_i(ms)	e_i(ms)	Remark
T_1	10	10	3	Real-time Task
T_2	15	15	4	Real-time Task
T_3	25	25	4	Real-time Task

- Due to thermal constraint, the maximum processor utilization is U_{max} $= 0.8$.
- The sensor-to-actuator delay of the control application must be constant and must not exceed 50% of the chosen sampling period.
- Assume that the measurement operation by the sensor task takes negligible time. Also, the actuation takes negligible time.
- The controller task of the control application has a WCET $e_c = 2$ms.

Design

1. sampling period h of the controller such that the utilization limit is not violated
2. the scheduling policy on the control and real-time tasks such that real-time tasks meet theirs deadline and controller task meets its sensor-to-actuator delay constraint
3. the controller such that the cruise controller is able to track the speed

Choice of sampling period

The utilization by all the real-time tasks is given by

$$U_{RT} = \frac{3}{10} + \frac{4}{15} + \frac{4}{25} = 0.7267$$

The utilization available for the control application

$$U_C = U_{max} - U_{RT} = 0.8 - 0.7267 = 0.0732$$

With the sampling period h,

$$U_C \geq \frac{e_c}{h} \rightarrow h \geq 27.32\text{ms}$$

We choose $h = 30$ms. Since the sensor-to-actuator delay must not exceed 50% of the length of sampling period h, the deadline D_c for the control task T_c is 15ms. Therefore, $h = 30$ms and $D_c = 15$ms.

The resulting task set including the control task becomes:

Tasks	p_i(ms)	D_i(ms)	e_i(ms)	Remark
T_1	10	10	3	Real-time Task
T_2	15	15	4	Real-time Task
T_3	25	25	4	Real-time Task
T_c	30	15	2	Control Task

Scheduling policy

First, we try fixed priority with rate-monotonic scheme. The task priorities as follows

Tasks	p_i(ms)	D_i(ms)	e_i(ms)	priority	Remark
T_1	10	10	3	1	Real-time Task
T_2	15	15	4	2	Real-time Task
T_3	25	25	4	3	Real-time Task
T_c	$h = 30$	$D_c = 15$	2	4	Control Task

With rate-monotonic scheme, we obtain the response times

$$R_1 = 3\text{ms}, R_2 = 7\text{ms}, R_3 = 14\text{ms}, R_c = 20\text{ms}$$

Clearly, $R_c > D_c$ is violating the timing requirement. Therefore, timing requirements are not met with rate-monotonic scheme.

Next, we try fixed priority with deadline-monotonic scheme. The task priorities as follows

Tasks	p_i(ms)	D_i(ms)	e_i(ms)	priority	Remark
T_1	10	10	3	1	Real-time Task
T_2	15	15	4	2	Real-time Task
T_3	25	25	4	4	Real-time Task
T_c	$h = 30$	$D_c = 15$	2	3	Control Task

With deadline-monotonic scheme, we obtain the response times

$$R_1 = 3\text{ms}, R_2 = 7\text{ms}, R_3 = 20\text{ms}, R_c = 9\text{ms}$$

Clearly, the timing requirements are met. The deadline monotonic scheme meets the timing requirements and we assign priorities as per deadline monotonic scheme.

Controller Design

We have sampling period $h = 30$ms. With deadline monotonic scheme, the worst-case response time is $R_c = 9$ms. Therefore, we design the controller with sensor-to-actuator delay 9ms, i.e., $D_c = 9$ms.

$$
\begin{aligned}
x[k+1] &= \phi x[k] + \gamma_1(D_c)u[k-1] + \gamma_0(D_c)u[k] \\
y[k] &= Cx[k] \\
\phi &= \begin{bmatrix} 1.0000 & 0.0300 & 0.0004 \\ -0.0027 & 0.9976 & 0.0299 \\ -0.1806 & -0.1606 & 0.9905 \end{bmatrix} \\
\gamma_1(D_c) &= \begin{bmatrix} 0.0000 \\ 0.0005 \\ 0.0519 \end{bmatrix}, \gamma_0(D_c) = \begin{bmatrix} 0.0000 \\ 0.0006 \\ 0.0221 \end{bmatrix}
\end{aligned}
$$

We choose new system states: $z[k] = \begin{bmatrix} x[k] \\ u[k-1] \end{bmatrix}$

The resulting augmented system is:

$$z[k+1] = \phi_{aug} z[k] + \gamma_{aug} u[k]$$
$$y[k] = C_{aug} z[k]$$

$$\phi_{aug} = \begin{bmatrix} \phi & \gamma_1(D_c) \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1.0000 & 0.0300 & 0.0004 & 0.0000 \\ -0.0027 & 0.9976 & 0.0299 & 0.0006 \\ -0.1806 & -0.1606 & 0.9905 & 0.0221 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 \end{bmatrix}$$

$$\gamma_{aug} = \begin{bmatrix} \gamma_0(D_c) \\ I \end{bmatrix} = \begin{bmatrix} 0.0000 \\ 0.0005 \\ 0.0519 \\ 1.0000 \end{bmatrix}, C_{aug} = \begin{bmatrix} C & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$

- The controllability matrix

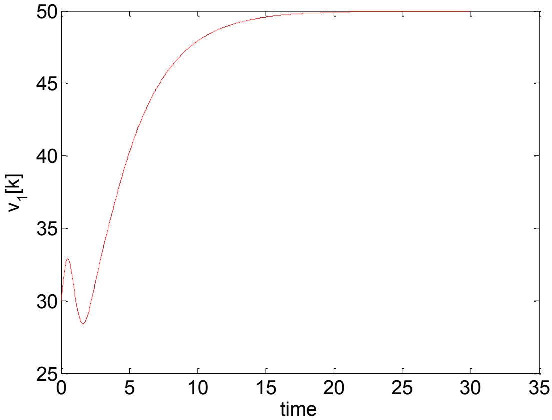
$$\gamma_{aug} = \begin{bmatrix}\gamma_{aug} & \phi_{aug}\gamma_{aug} & \phi_{aug}^2\gamma_{aug} & \phi_{aug}^3\gamma_{aug}\end{bmatrix}$$
$$= \begin{bmatrix} 0.0000 & 0.0001 & 0.0002 & 0.0003 \\ 0.0005 & 0.0027 & 0.0048 & 0.0070 \\ 0.0519 & 0.0734 & 0.0723 & 0.0708 \\ 1.0000 & 0 & 0 & 0 \end{bmatrix}$$

- $det(\gamma_{aug}) \neq 0$ indicates the augmented system with sensor-to-actuator delay D_c is controllable.
- With $\alpha = \begin{bmatrix}0.9 & 0.9 & 0.98 & 0.98\end{bmatrix}$, the feedback gain is:

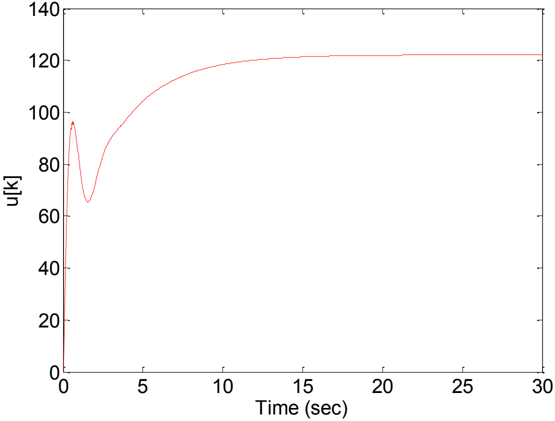
$$K = -\begin{bmatrix}0 & 0 & \dots & 1\end{bmatrix}\gamma_{aug}^{-1}H(\phi_{aug})$$
$$= \begin{bmatrix}0.4773 & 0.3265 & -0.1579 & 0.7799\end{bmatrix}$$

The feedforward gain is $F = 0.0601$

The system response is the following with initial condition $v_1[0] = 30\text{m/s}$, $v_2[0] = 10\text{m/s}^2$, $v_3[0] = 5\text{m/s}^3$. It takes 20 sec to reach a velocity of $v_1[0] = 50\text{m/s}$



The maximum throttle angle is: $u[k] = 122.08°$



Tools for Timing Analysis

There are various commercial timing analysis tools that are used in the industry.

Real-Time Timing

By Ralf Klein, Symtavision GmbH

Seamless integration of SymTA/S with ETAS tools

With its products designated SymTA/S and TraceAnalyzer, Symtavision offers a consistent, AUTOSAR-conformant tool chain for the planning, optimization, and verification of the real-time capability of embedded systems. It seamlessly integrates with the ASCET and ISOLAR-A tools from ETAS. The same is true of the latest releases, i.e., SymTA/S 3.2 and TraceAnalyzer 3.2.

Using SymTA/S, software developers can efficiently analyze the timing behavior of functions developed in ETAS ASCET directly from within the ASCET environment. It is also possible to optimize the schedule and task layout of the ETAS RTA operating system so as to permit the best possible use of available resources.

Working closely with ETAS, Symtavision offers a plain interface to the AUTOSAR tools of the ETAS ISOLAR-A tool chain, which is also based on the open AUTOSAR tool platform (Artop). ISOLAR-A's arxml files are directly imported in SymTA/S, providing developers with the means for the consistent modeling and verification of

of the ten Contributing Members, which also include BMW, Continental, and Peugeot/Citroën.

Dr. Kai Richter, CEO of Symtavision, is enthusiastic about his company's well-earned recognition "Our new role as a Contributing Member of the Artop User Group underscores our contribution to the development of AUTOSAR-conformant systems and ECUs. It also emphasizes our motivation for participating in the work on a shared foundation for AUTOSAR development tools." "Also, the deployment of Artop helps us to reduce the development costs for basic functions of our products, and to keep the development focus on our unique features and customer benefits. At the same time, as Contributing Member, we will be able to integrate our extensions and improvements of Artop in future releases. That way, we can make these benefits available to others as well."

Manabendra N. Gupta – Product Manager ASCET at ETAS

"With ASCET, the implementation of functional or physical behavior in production software is both efficient and safe. However, some aspects that have largely been ignored so far relate to the character and verification of a preemptively planned

Tools for Design Space Exploration

1. We have seen that the architecture and the implementation of the system impacts control performance.
2. There are also several tools for automated architecture synthesis and design space exploration.