Supervisory Control & Monitoring

Topic - Basics of robustness issues in control systems
Teacher - Prof. Elio USAI

elio.usai@unica.it
Dipartimento di Ingegneria Elettrica ed Elettronica
Università di Cagliari

References

Katsuhito OGATA
Modern Control Engineering, 5th edition
Pearson, 2010

John J. D'AZZO, Constantine H. HOUPIS,
Linear control systems analysis & design – Conventional and modern, 3rd edition
McGraw-Hill, 1988

Silvio SIMANI, Cesare FANTUZZI, Ron J. PATTON
Model-Based fault diagnosis in dynamical systems using identification techniques
Springer-Verlag, 2002

Francesco BULLO
Lectures on Network Systems
Createspace, 2018

Summary

- Systems
- System Stability
- SISO control schemes
- Robustness of SISO control schemes
- MIMO systems
- Observers for model-based FDI
- Networked control
- Multi agent systems
- Distributed control

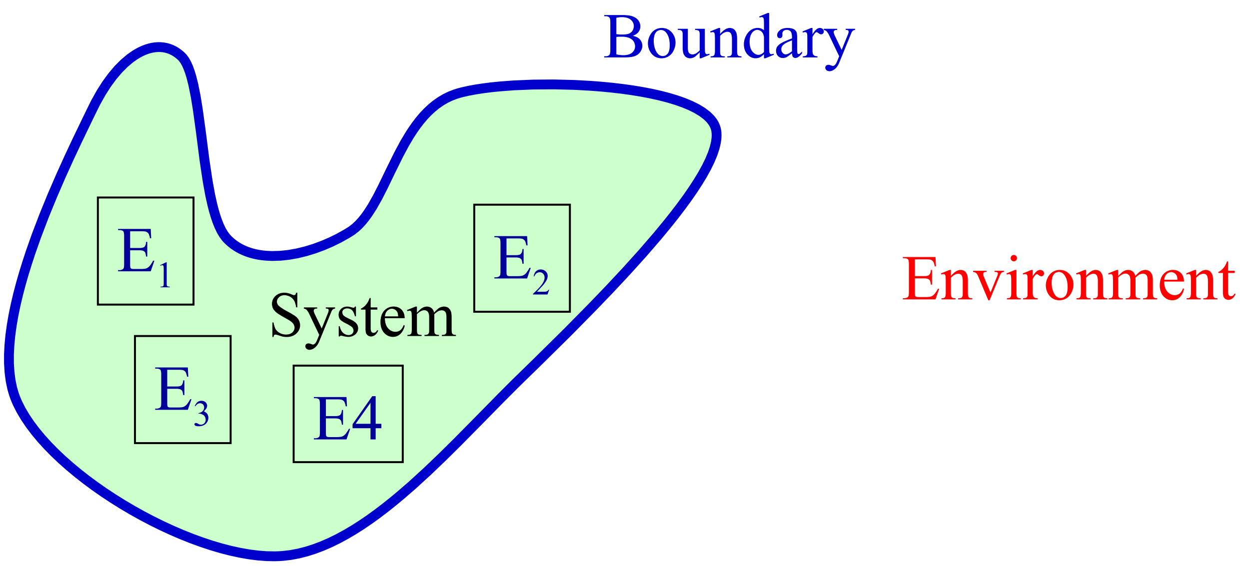
Systems

A system can be defined as:

- a set of things working together as parts of a mechanism or an interconnecting network; a complex whole.
- a set of principles or procedures according to which something is done; an organized scheme or method.
- ……………
- A system is a group of interacting or interrelated entities that form a unified whole.

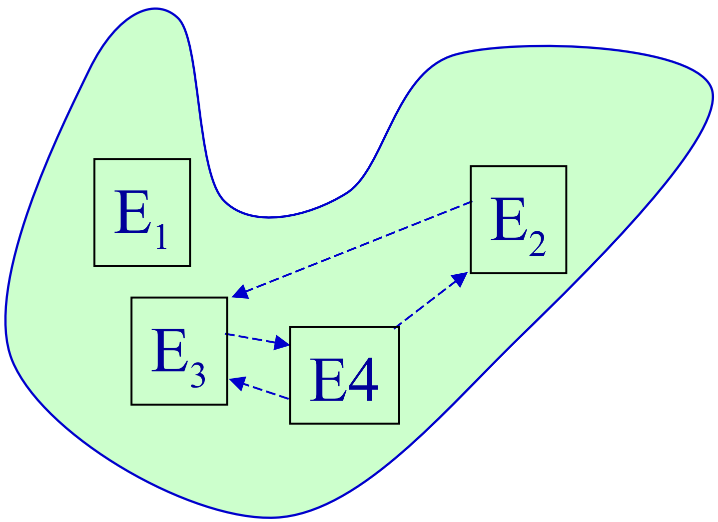
Systems

A system is composed by a number of entities separated from the environment by a boundary



Systems

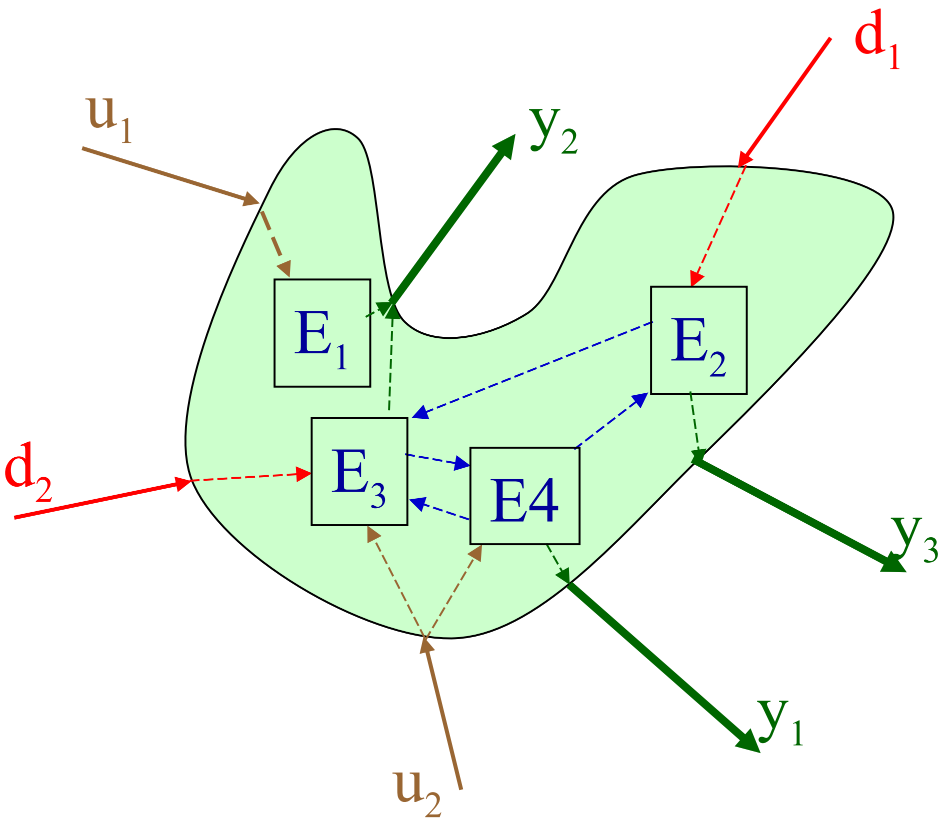
A system is composed by a number of entities separated from the environment by a boundary



It has a structure defined by the relationships among the entities

Systems

A system is composed by a number of entities separated from the environment by a boundary

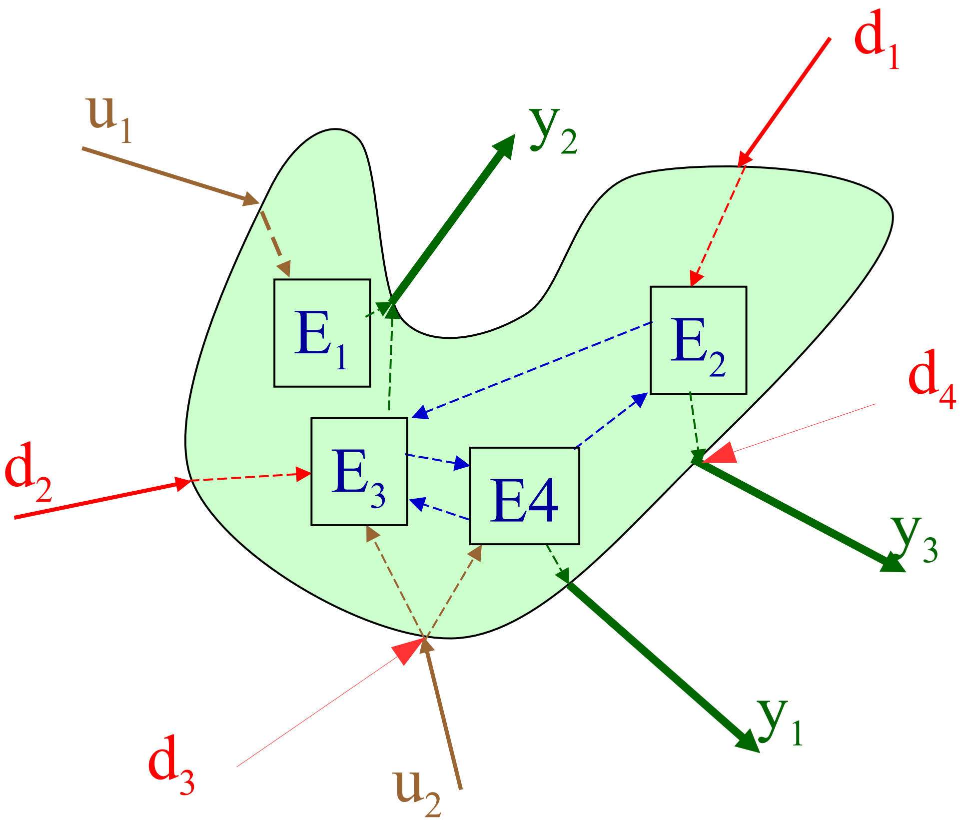


It has a structure defined by the relationships among the entities

It reacts to the environment inputs (**u**, **d**) and acts on it (**y**) to exploit its purpose

Systems

A system is composed by a number of entities separated from the environment by a boundary



The inputs could be *certified* or manipulable, i.e., properly named inputs **u**

The inputs could be *malicious* or not manipulable, i.e., properly named disturbances **d**

Systems

The concept of **system** is very general and include:

- Natural systems
- Human made systems
- Social systems
- Cultural systems
- Economic systems
- Physical systems

Any entity of a systems can be a system itself: **Sub-system**

Systems

In order to exploit its purpose in spite of malicious inputs from the environment, or even internal modifications, the system should have some properties:

- **Robustness**

  Property of a system to stay healthy in perturbed conditions, i.e., the structural ability of a system to resist to changes in parameters/structure and to external perturbations maintaining its steady-state performance

- **Resilience**

  The capacity of a system to recover its behaviour and performance from changes and external stresses. *It is an extension of the property definition for mechanical bodies*.

Systems

The behaviour of systems can be represented by models in the time domain

Differential equations

$$\frac{d\,\boldsymbol{y}(t)}{d\,t} = F\left(\boldsymbol{y}(t), \boldsymbol{u}(t), \boldsymbol{d}(t), t\right)$$

$$H\left(\frac{\partial\,\boldsymbol{y}(x,t)}{\partial\,t}, \frac{\partial\,\boldsymbol{y}(x,t)}{\partial\,x}\right) = F\left(\boldsymbol{y}(x,t), \boldsymbol{u}(x,t), \boldsymbol{d}(x,t), t\right)$$

Difference equations

$$\Delta\, y(t) = F\left(\boldsymbol{y}(t)\right)$$

Systems

The behaviour of systems can be represented by models in the frequency domain *(only linear systems)*

Differential equations

Laplace-transform → $Y(s) = F(s) \cdot U(s)$

Fourier-transform → $Y(j\omega) = F(j\omega) \cdot U(j\omega)$

Transfer function/matrix

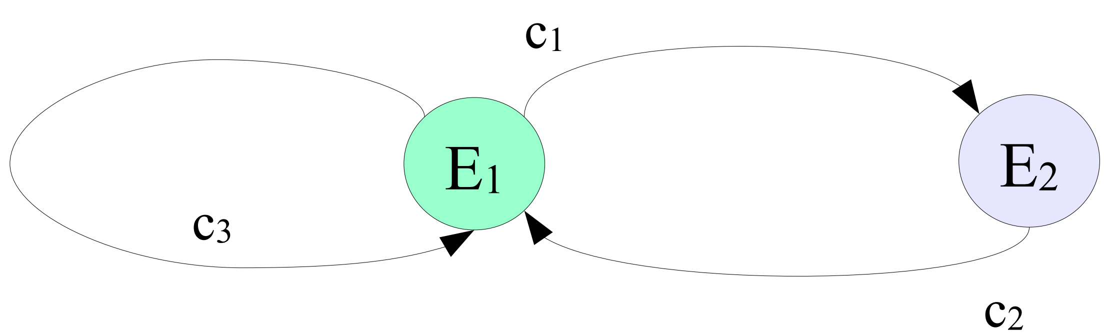
Difference equations

z-transform → $Y(z) = F(z) \cdot U(z)$

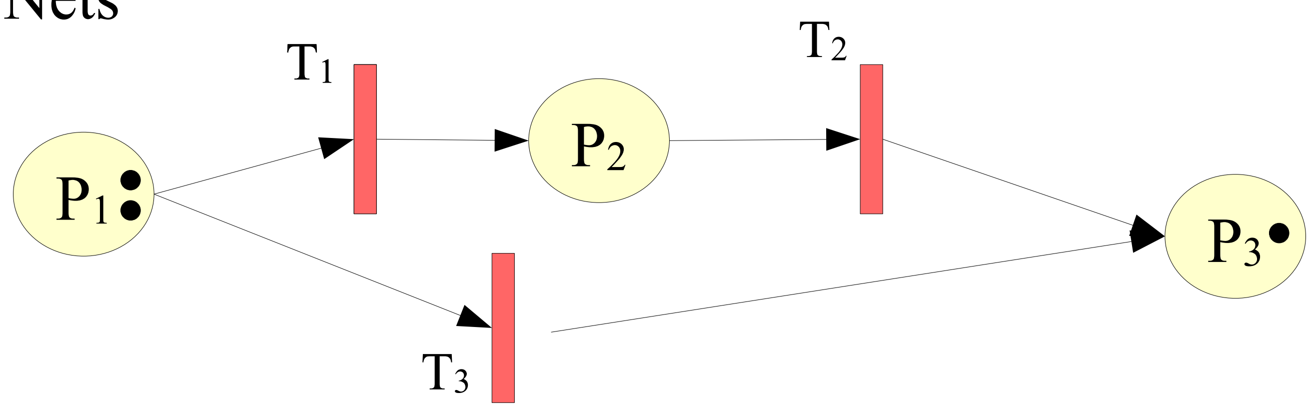
Systems

The behaviour of systems can be represented by finite-state models

Automata

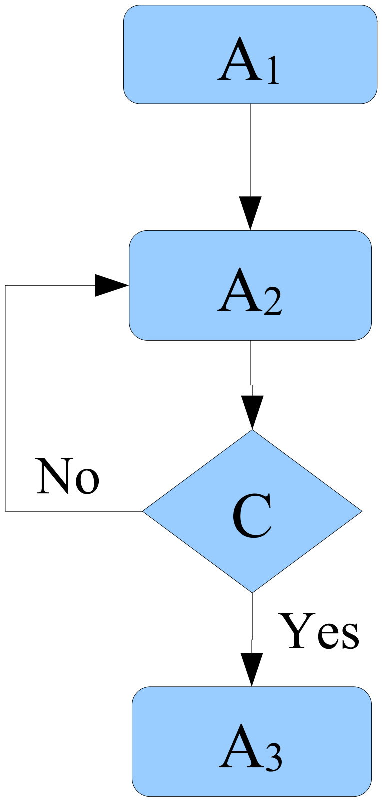


Petri Nets



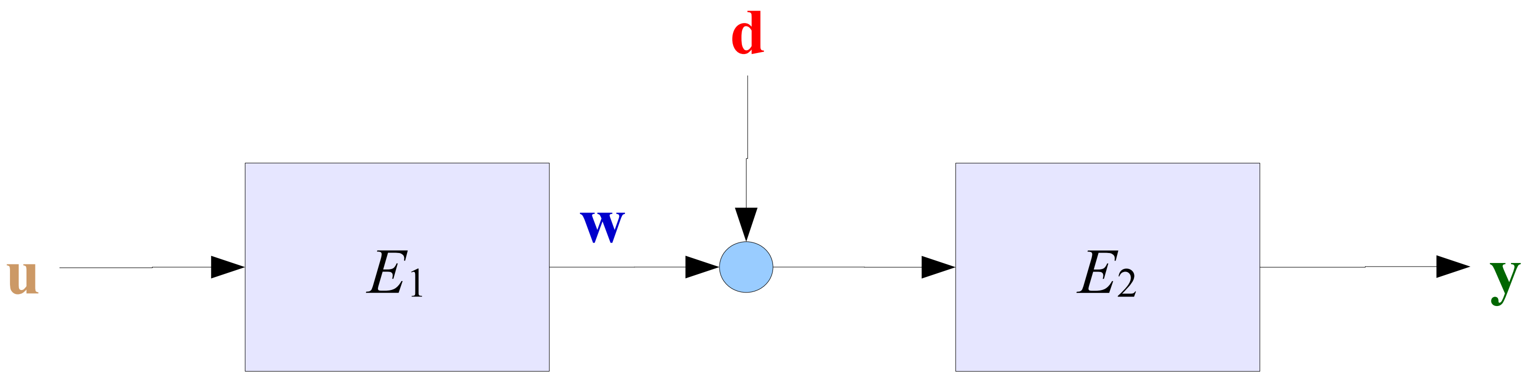
Systems

The behaviour of systems can be represented by flow charts



Systems

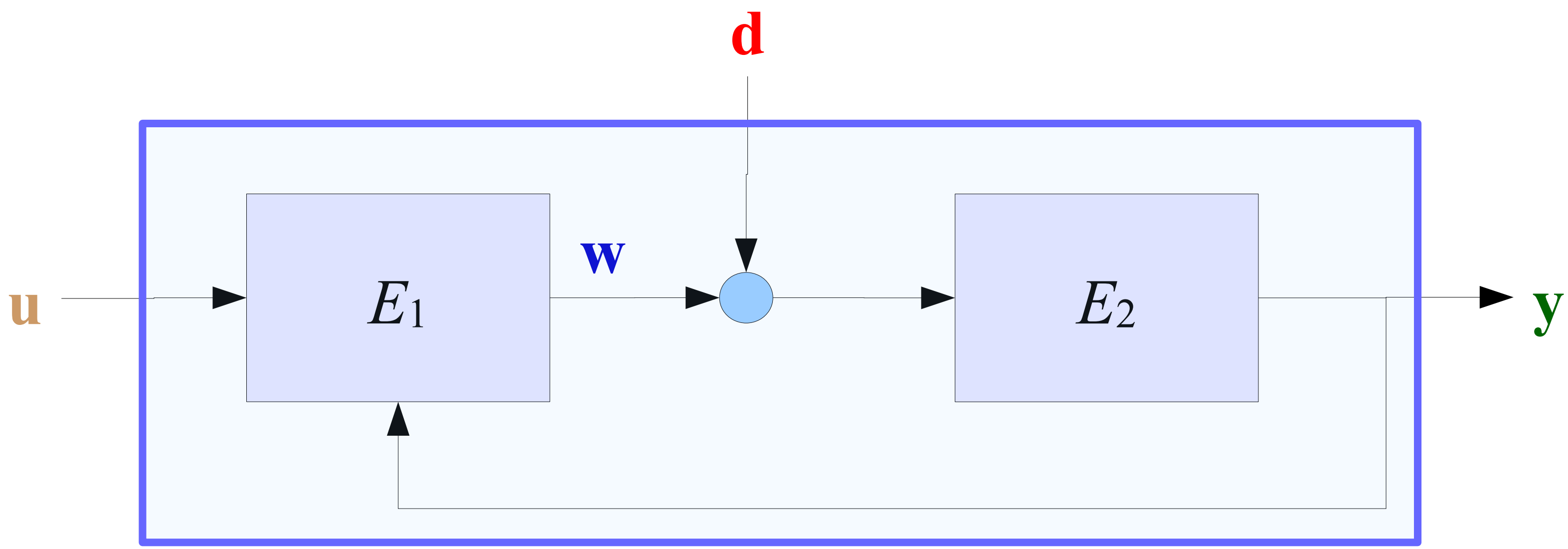
Open-loop systems has very poor robustness and resilience properties



The subsystem E_1 generates the variable **w** taking into account the input **u** only and, by E_2, the output **y** depends on both **w** and **d**

Systems

Closed-loop systems has to be implemented to have **good** robustness and resilience properties



The subsystem E_1 generates the variable **w** taking into account the input **u** and the output **y**, such that **w** compensate for or limit, at least, the influence of **d** on E_2

System Stability

Stability is the property of a system to stay in a vicinity, or return back, to an equilibrium/working point after a perturbation

Equilibrium point: a system state that does not change in the absence of inputs (*autonomous system*)

Working point: a system state and output that does not change in the presence of constant inputs

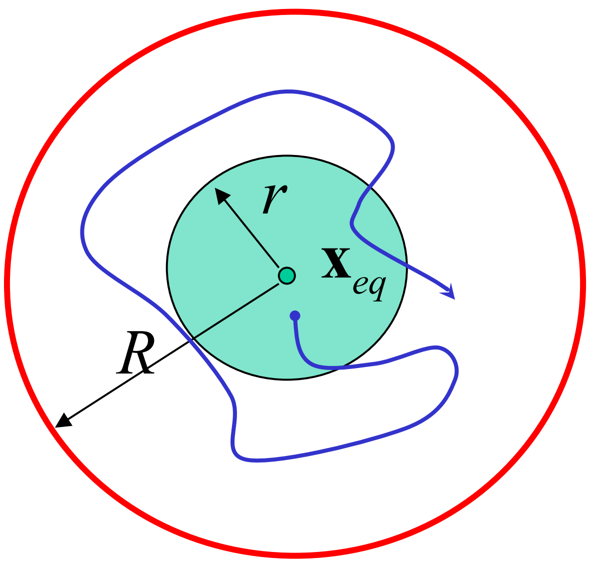
Any working point can be considered as an equilibrium point by means of a state shift

System Stability

Stability is the property of a system to stay in a vicinity, or return back, to an equilibrium/working point after a perturbation

An equilibrium point $\boldsymbol{x}_{eq}$ is **stable** if:
For all $R > 0$, there exist an $r(R) > 0$ such that

$$\|\boldsymbol{x}(t_0)-\boldsymbol{x}_{eq}\| < r \;\Rightarrow\; \|\boldsymbol{x}(t)-\boldsymbol{x}_{eq}\| < R; \quad t \geq t_0$$



R can be chosen arbitrarily small.

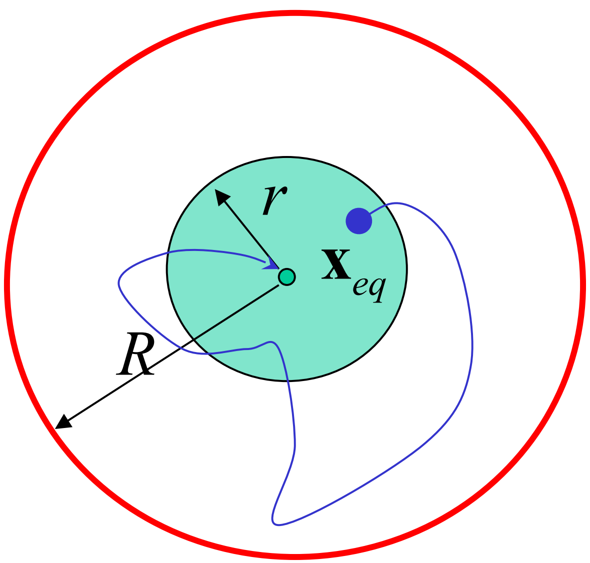
It is not sufficient that the perturbed trajectory remains limited within *R*

System Stability

Stability is the property of a system to stay in a vicinity, or return back, to an equilibrium/working point after a perturbation

An equilibrium point $\boldsymbol{x}_{\mathrm{eq}}$ is **asymptotically stable** if:

It is stable AND $\lim_{t \to \infty} \left\| \mathbf{x}(t) - \mathbf{x}_{eq} \right\| = 0$



R can be chosen arbitrarily small.

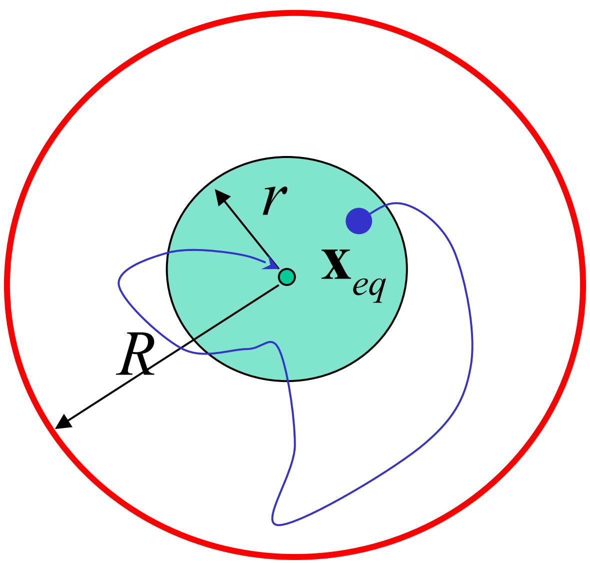
The perturbed trajectory never leaves *R* and goes back to the equilibrium point

System Stability

Stability is the property of a system to stay in a vicinity, or return back, to an equilibrium/working point after a perturbation

An equilibrium point $\boldsymbol{x}_{eq}$ is **finite-time stable** if:

It is stable AND exists $T > t_0$ such that $\|\boldsymbol{x}(t)-\boldsymbol{x}_{eq}\| = 0;\ \ t \geq T(\boldsymbol{x}_0)$



R can be chosen arbitrarily small.

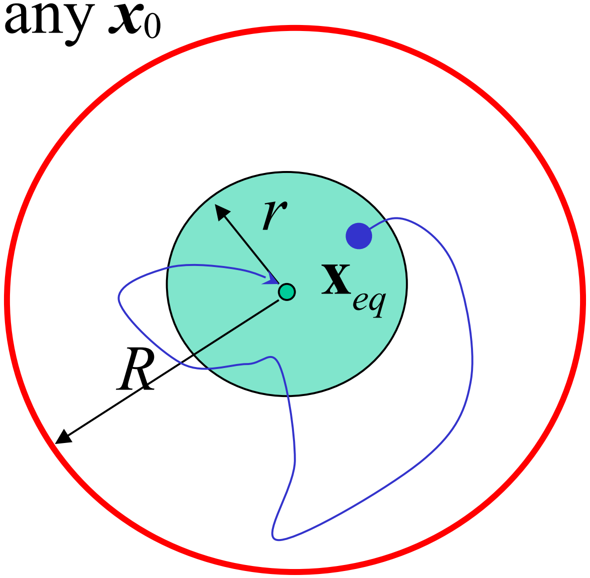
The perturbed trajectory never leave *R* and goes back to the equilibrium point ina finite time *depending* on the initial conditions

System Stability

Stability is the property of a system to stay in a vicinity, or return back, to an equilibrium/working point after a perturbation

An equilibrium point $\boldsymbol{x}_{eq}$ is **fixed-time stable** if:

It is stable AND exists $T > t_0$ such that $\|\boldsymbol{x}(t)-\boldsymbol{x}_{eq}\| = 0;\ \ t \geq T$ for any $\boldsymbol{x}_0$

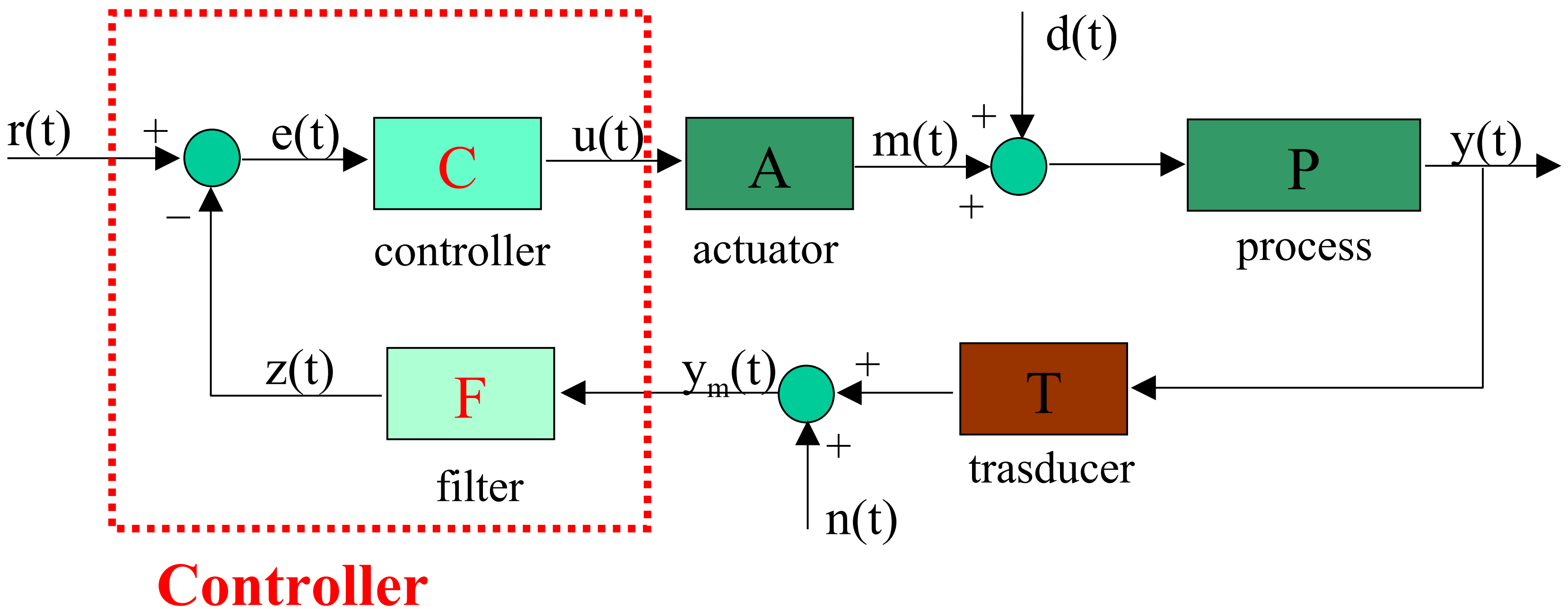


R can be chosen arbitrarily small.

The perturbed trajectory never leave *R* and goes back to the equilibrium point in a finite time *independent* from the initial conditions

Single-loop output feedback

The control action depends on the mismatching between the expected behaviour (set-point) and the actual one as measured.



Disturbances can appear anywhere outside the controller

SISO control schemes

Single-loop output feedback

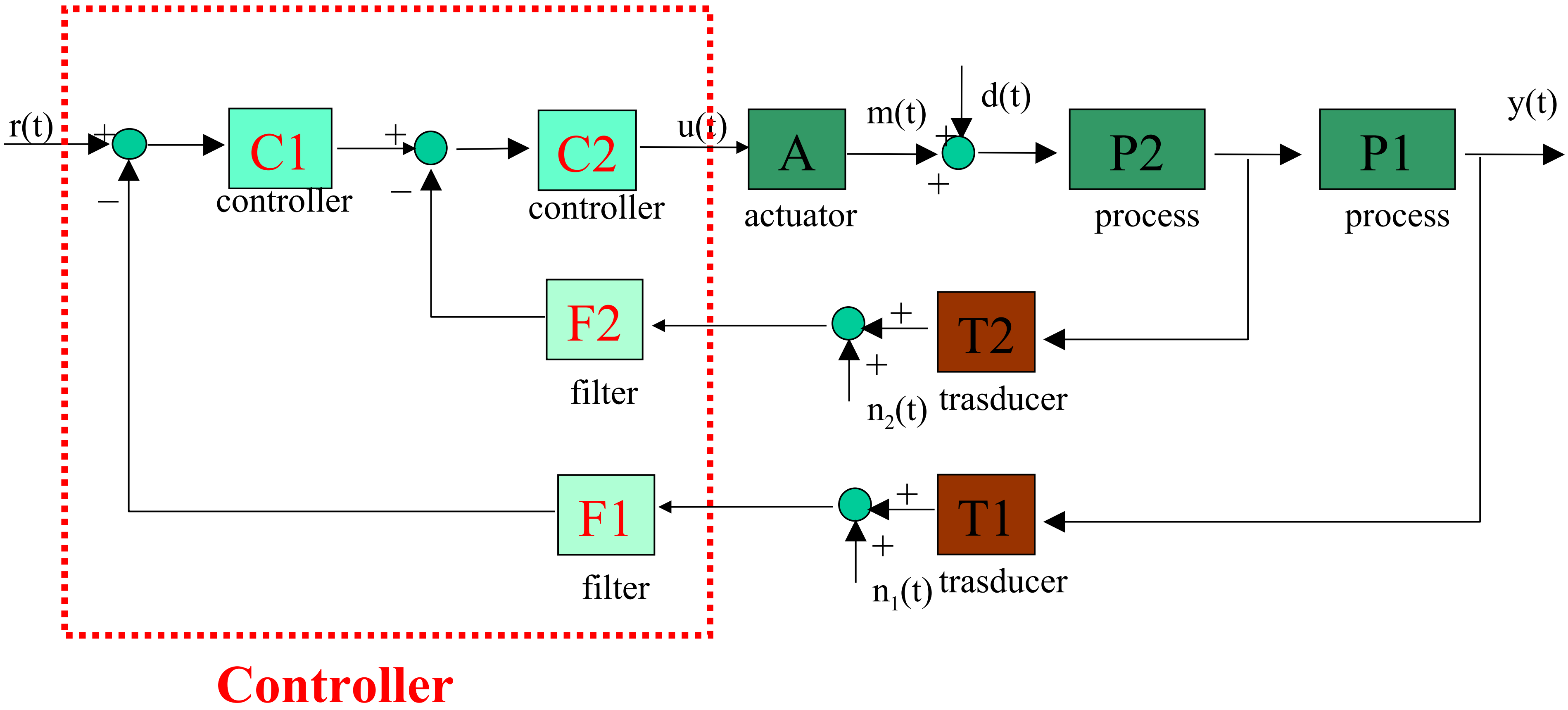
$$W_r(j\omega) = \frac{C(j\omega)A(j\omega)P(j\omega)}{1 + C(j\omega)A(j\omega)P(j\omega)T(j\omega)F(j\omega)}$$

$$W_d(j\omega) = \frac{P(j\omega)}{1 + C(j\omega)A(j\omega)P(j\omega)T(j\omega)F(j\omega)}$$

$$W_n(j\omega) = -\frac{F(j\omega)C(j\omega)A(j\omega)P(j\omega)}{1 + C(j\omega)A(j\omega)P(j\omega)T(j\omega)F(j\omega)}$$

Feedback cascade control

The control action depends on a couple of “nested” control loops.



SISO control schemes

Feedback cascade control

$$W_r = \frac{C_1 C_2 A P_2 P_1}{1 + C_2 A P_2 T_2 F_2 + C_1 C_2 A P_1 P_2 T_1 F_1}$$

$$W_d = \frac{P_2 P_1}{1 + C_2 A P_2 T_2 F_2 + C_1 C_2 A P_1 P_2 T_1 F_1}$$

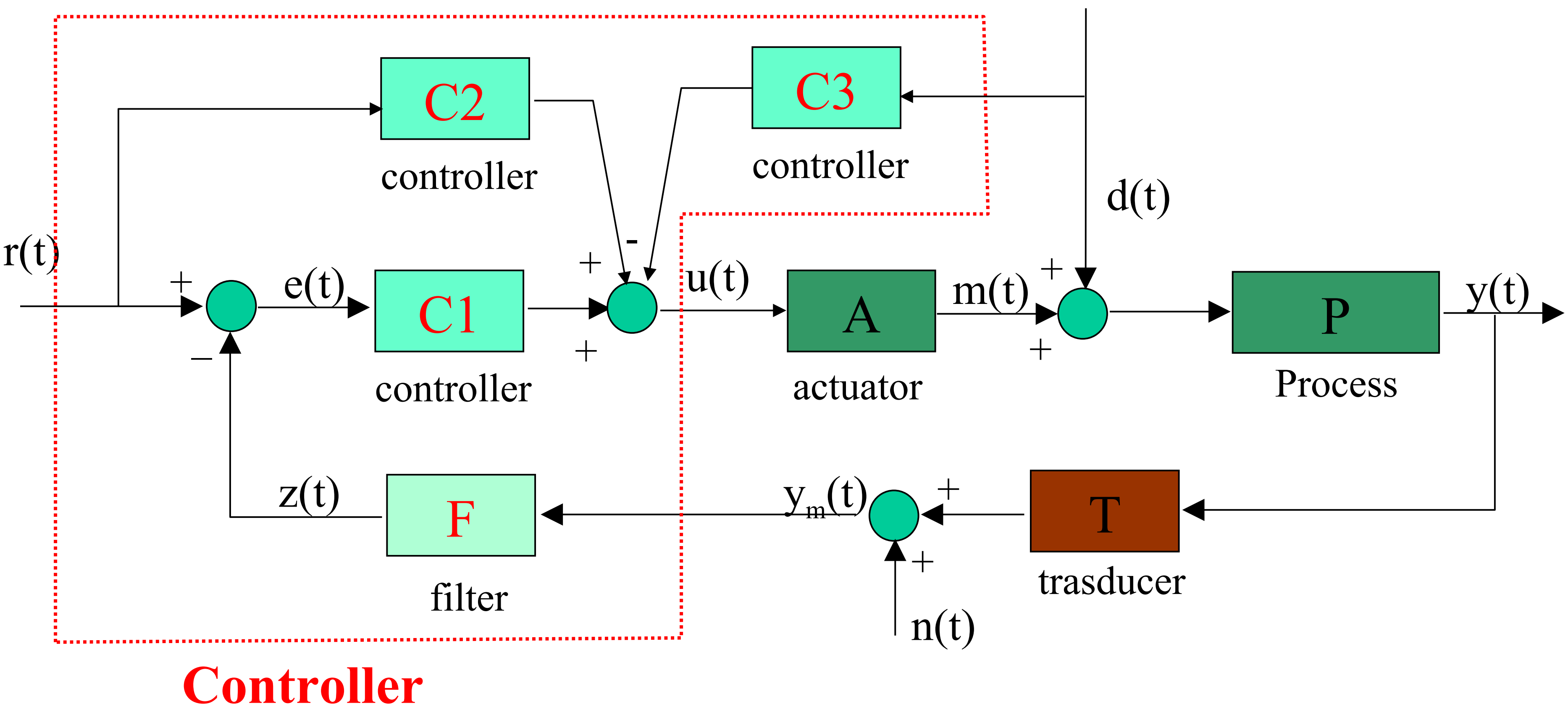
$$W_{n_2} = -\frac{F_2 C_2 A P_2 P_1}{1 + C_2 A P_2 T_2 F_2 + C_1 C_2 A P_1 P_2 T_1 F_1}$$

$$W_{n_1} = -\frac{F_1 C_1 C_2 A P_2 P_1}{1 + C_2 A P_2 T_2 F_2 + C_1 C_2 A P_1 P_2 T_1 F_1}$$

SISO control schemes

Feed-Forward control

The control action is implemented by means of two components: one predictive (feedforward) and one corrective (feedback).



SISO control schemes

Feed-Forward control

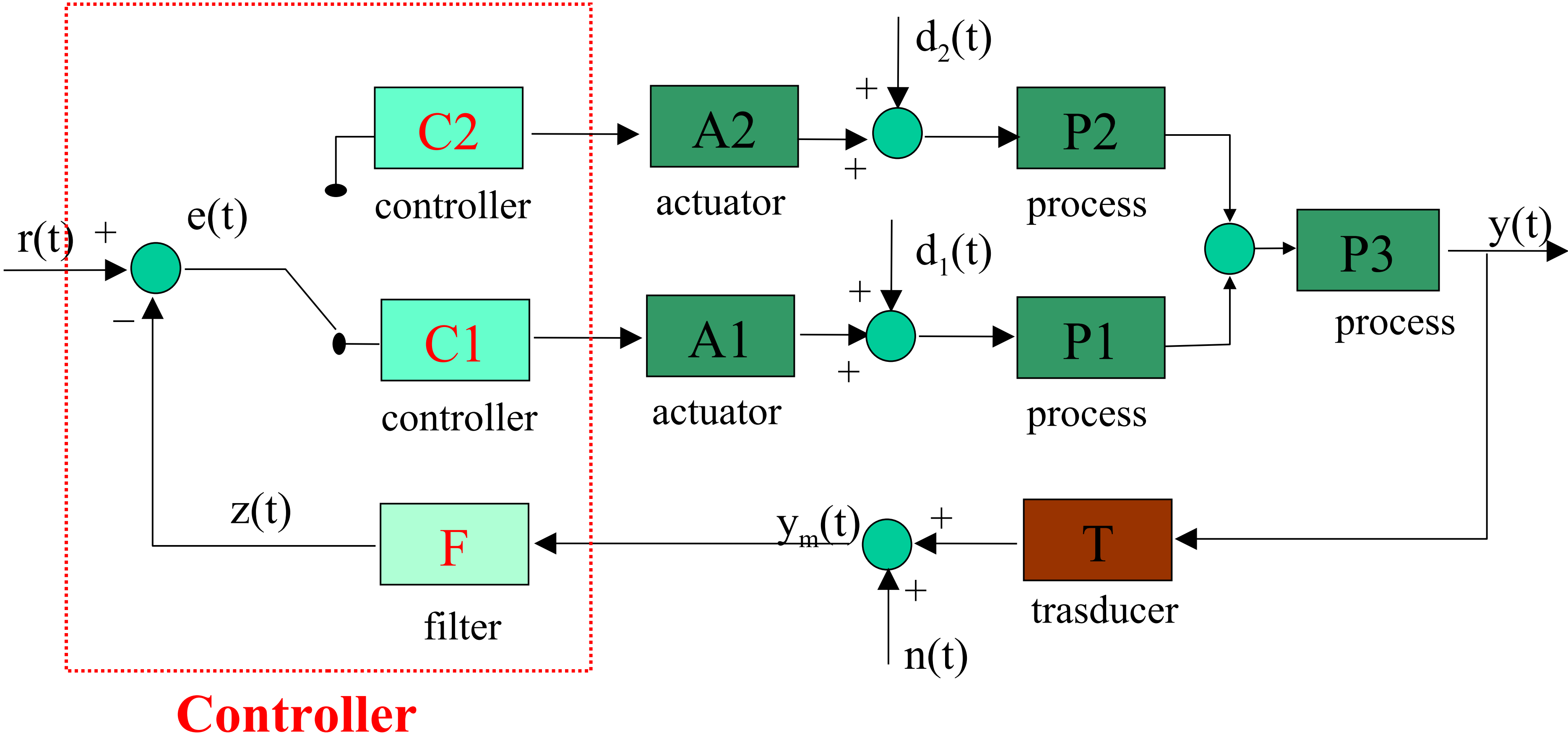
$$W_r = \frac{(C_1 + C_2)AP}{1 + C_1 APTF} \qquad C_2 = \frac{1}{AP} \Rightarrow \quad W_r = \frac{1 + C_1 AP}{1 + C_1 APTF} \approx 1$$

$$W_d = \frac{(1 - C_3 A)P}{1 + C_1 APTF} \qquad C_3 = \frac{1}{A} \Rightarrow \quad W_d \equiv 0$$

$$W_n = -\frac{FC_1 AP}{1 + C_1 APTF}$$

Split-range control

Two controllers alternatively act two actuators affecting the same process.



SISO control schemes

Split-range control

$$W_r = \begin{cases} \dfrac{C_1 A_1 P_1 P_3}{1 + C_1 A_1 P_1 P_3 TF} & e > 0 \\ \dfrac{C_2 A_2 P_2 P_3}{1 + C_2 A_2 P_2 P_3 TF} & e \le 0 \end{cases}$$

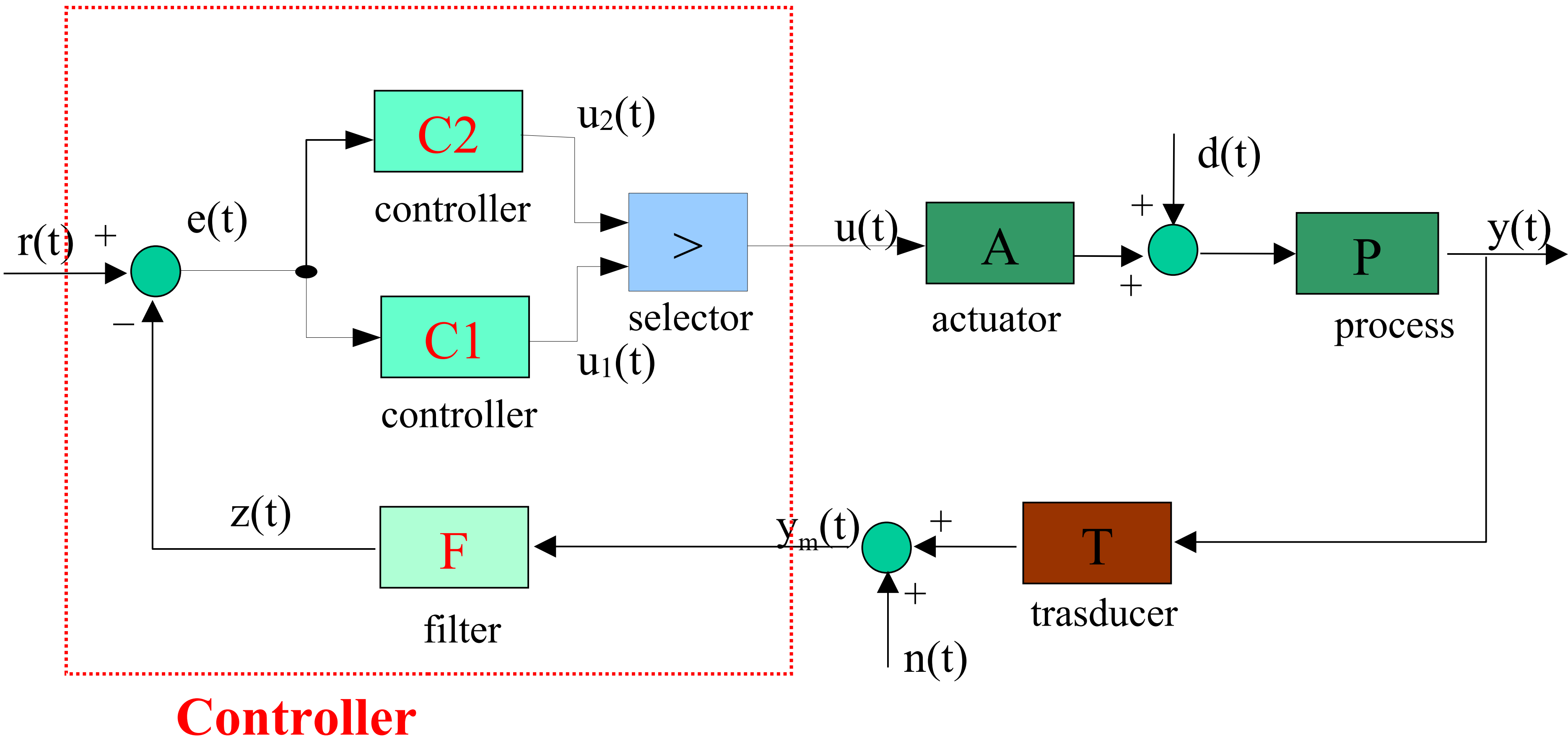
$$W_{d_1} = \begin{cases} \dfrac{P_1 P_3}{1 + C_1 A_1 P_1 P_3 TF} & e > 0 \\ \dfrac{P_1 P_3}{1 + C_2 A_2 P_2 P_3 TF} & e \le 0 \end{cases}$$

$$W_n = \begin{cases} -\dfrac{F C_1 A_1 P_1 P_3}{1 + C_1 A_1 P_1 P_3 TF} & e > 0 \\ \dfrac{F C_2 A_2 P_2 P_3}{1 + C_2 A_2 P_2 P_3 TF} & e \le 0 \end{cases}$$

SISO control schemes

Override control

Two controllers can act on the same actuator with some priority



Override control

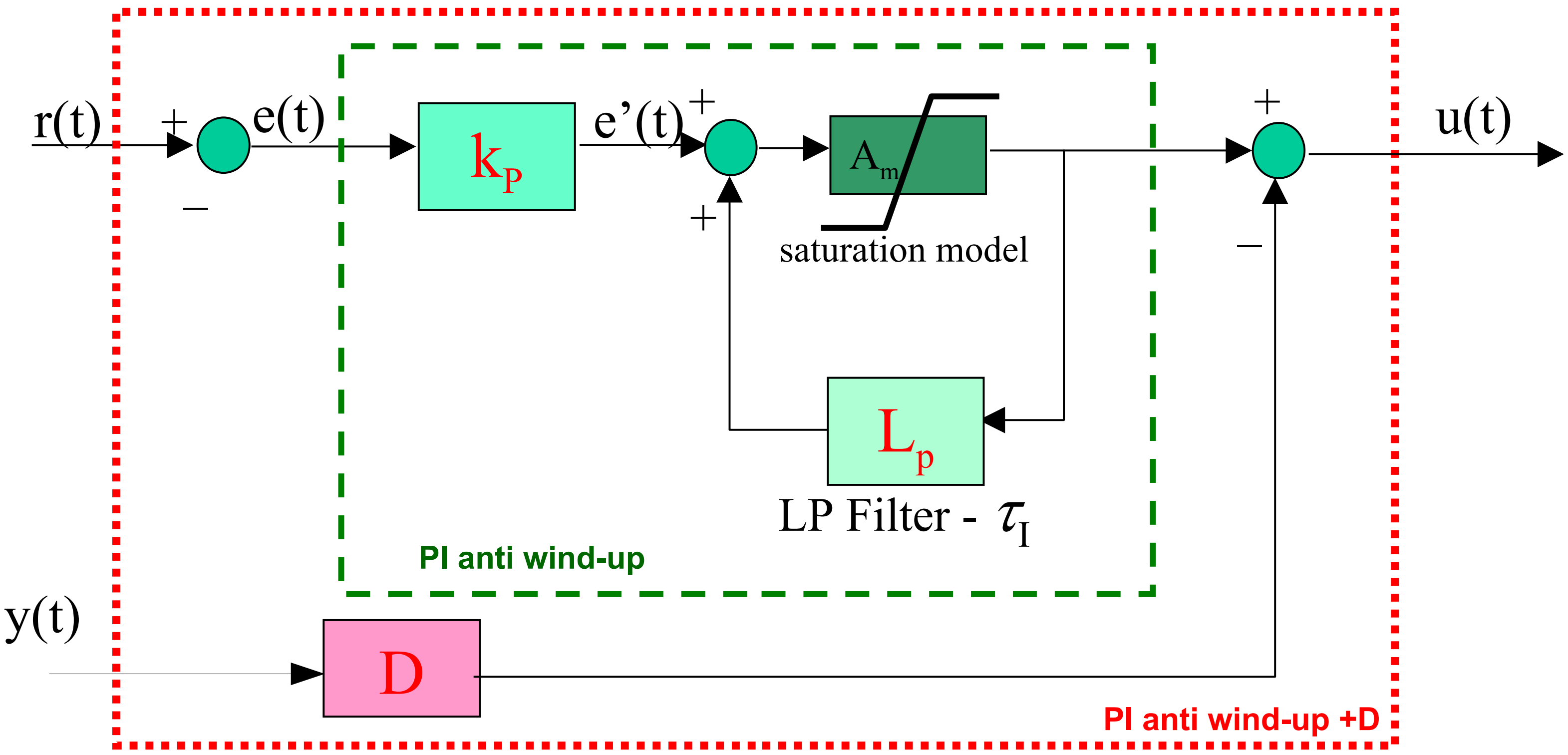
$$W_r = \begin{cases} \dfrac{C_2 A P}{1 + C_2 A P T F} & u_2 > u_1 \\ \dfrac{C_1 A P}{1 + C_1 A P T F} & u_2 < u_1 \end{cases}$$

$$W_d = \begin{cases} \dfrac{P}{1 + C_2 A P T F} & u_2 > u_1 \\ \dfrac{P}{1 + C_1 A P T F} & u_2 < u_1 \end{cases}$$

$$W_n = \begin{cases} \dfrac{-F C_2 A P}{1 + C_2 A P T F} & u_2 > u_1 \\ \dfrac{-F C_1 A P}{1 + C_1 A P T F} & u_2 < u_1 \end{cases}$$

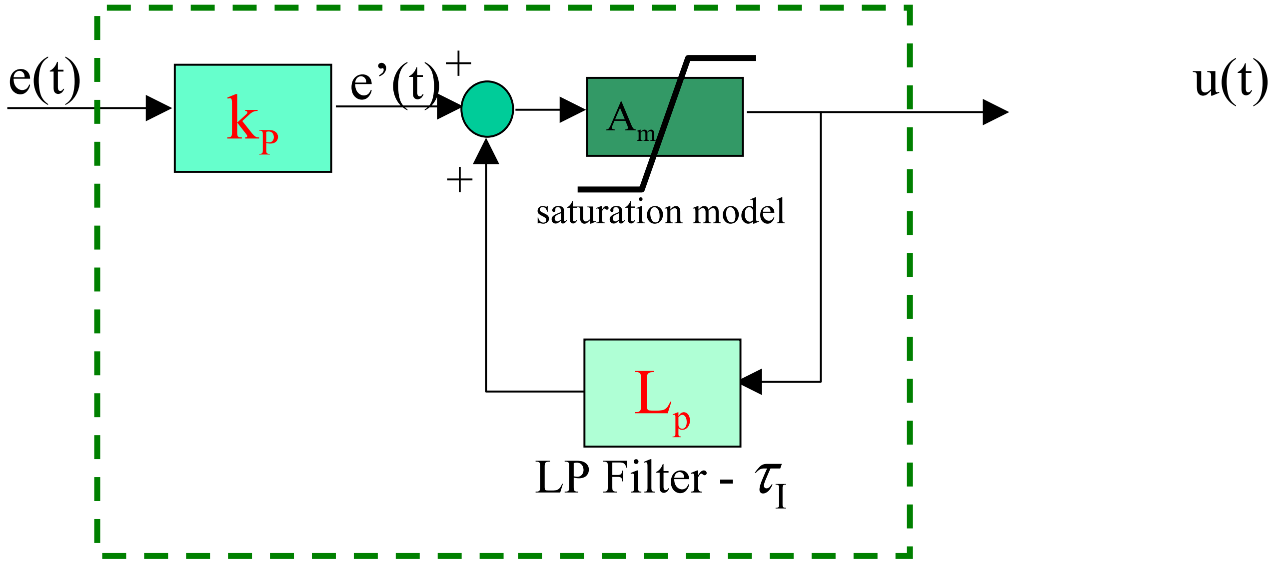
Anti-Wind-up controller

The model of the actuator saturation is embedded into the controller to exit the saturation as soon as the error changes its sign



Anti-Wind-up controller

The model of the actuator saturation is embedded into the controller to exit the saturation as soon as the error changes its sign

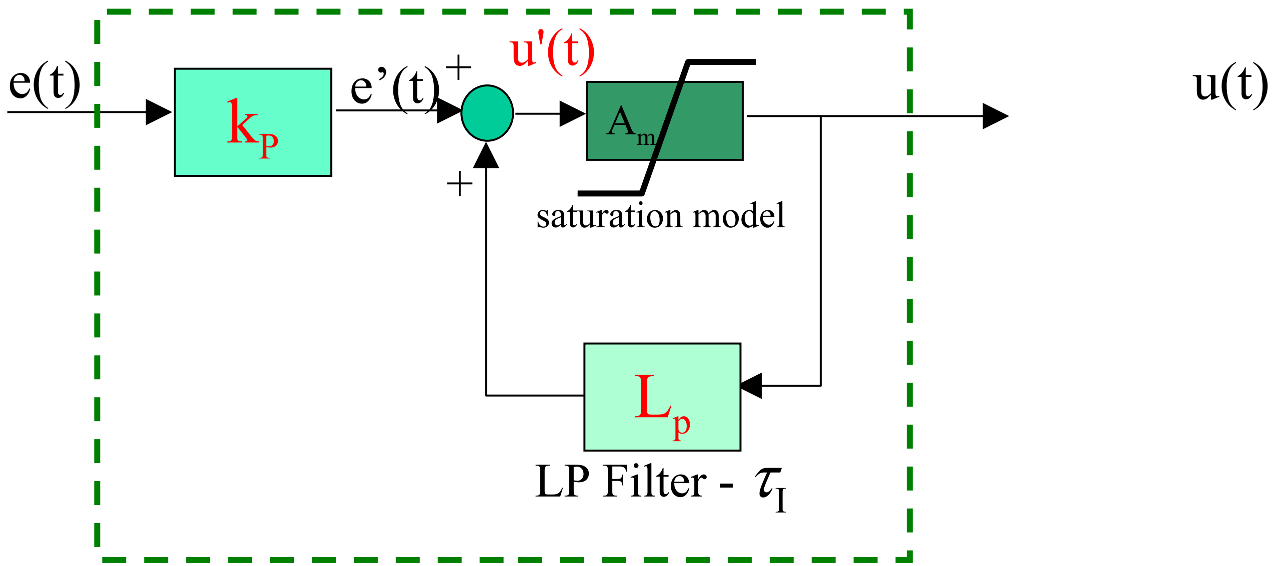


If saturation is not engaged

$$U(s)=k_p \frac{1}{1-\frac{1}{1+\tau s}} E(s)=k_p \frac{1+\tau s}{\tau s} E(s)$$

Anti-Wind-up controller

The model of the actuator saturation is embedded into the controller to exit the saturation as soon as the error changes its sign

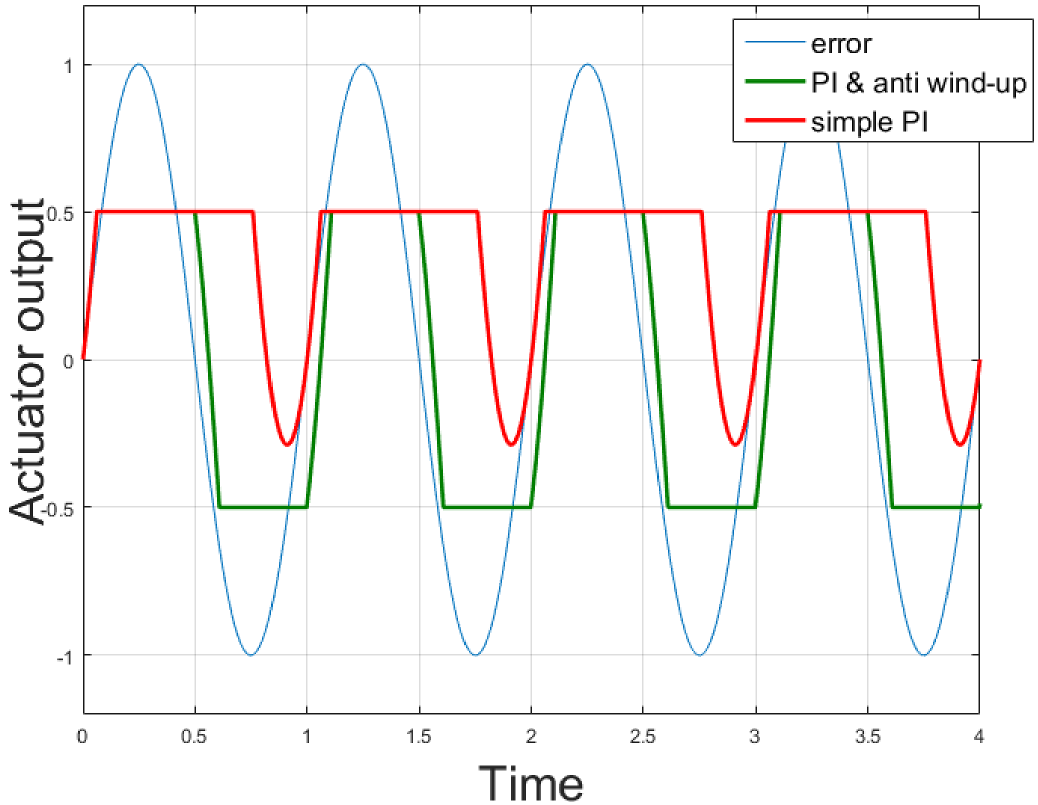


If saturation is engaged $u'(t) = e'(t) + U_M \, sign(u(t))$

SISO control schemes

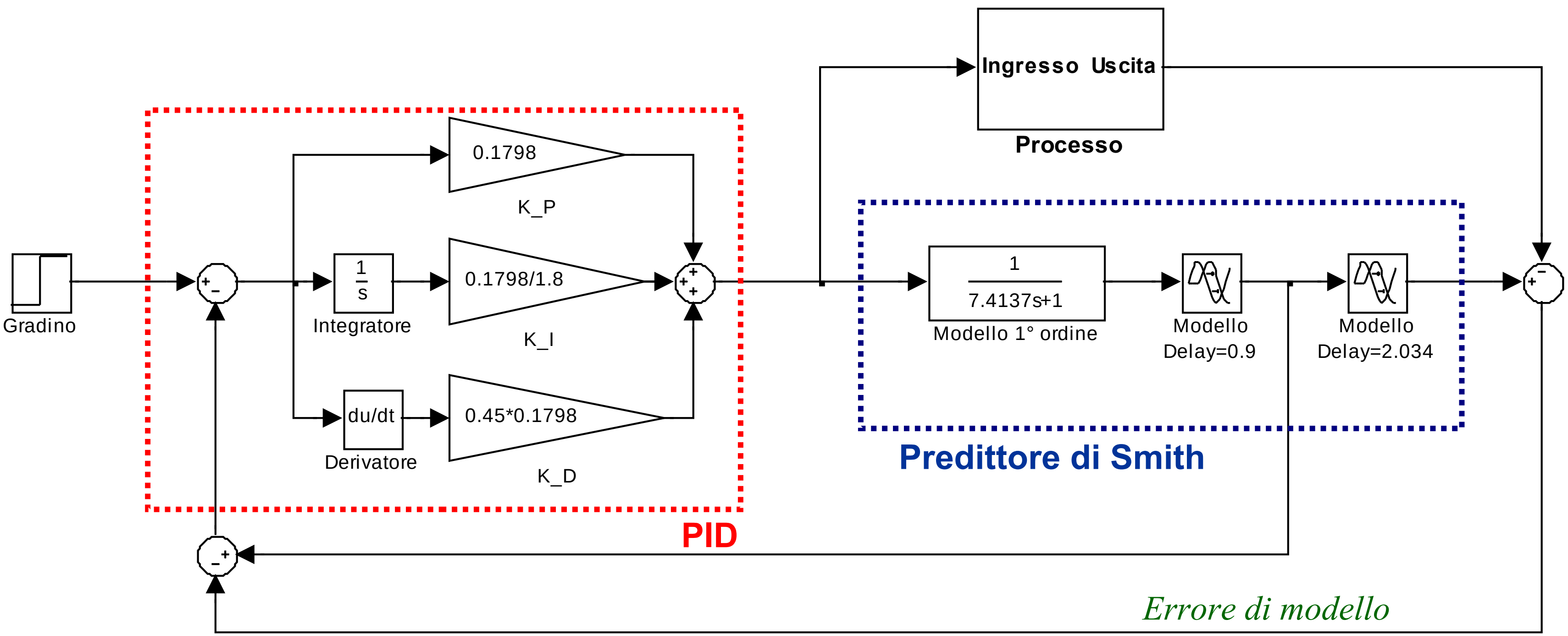
Anti-Wind-up controller

The model of the actuator saturation is embedded into the controller to exit the saturation as soon as the error changes its sign



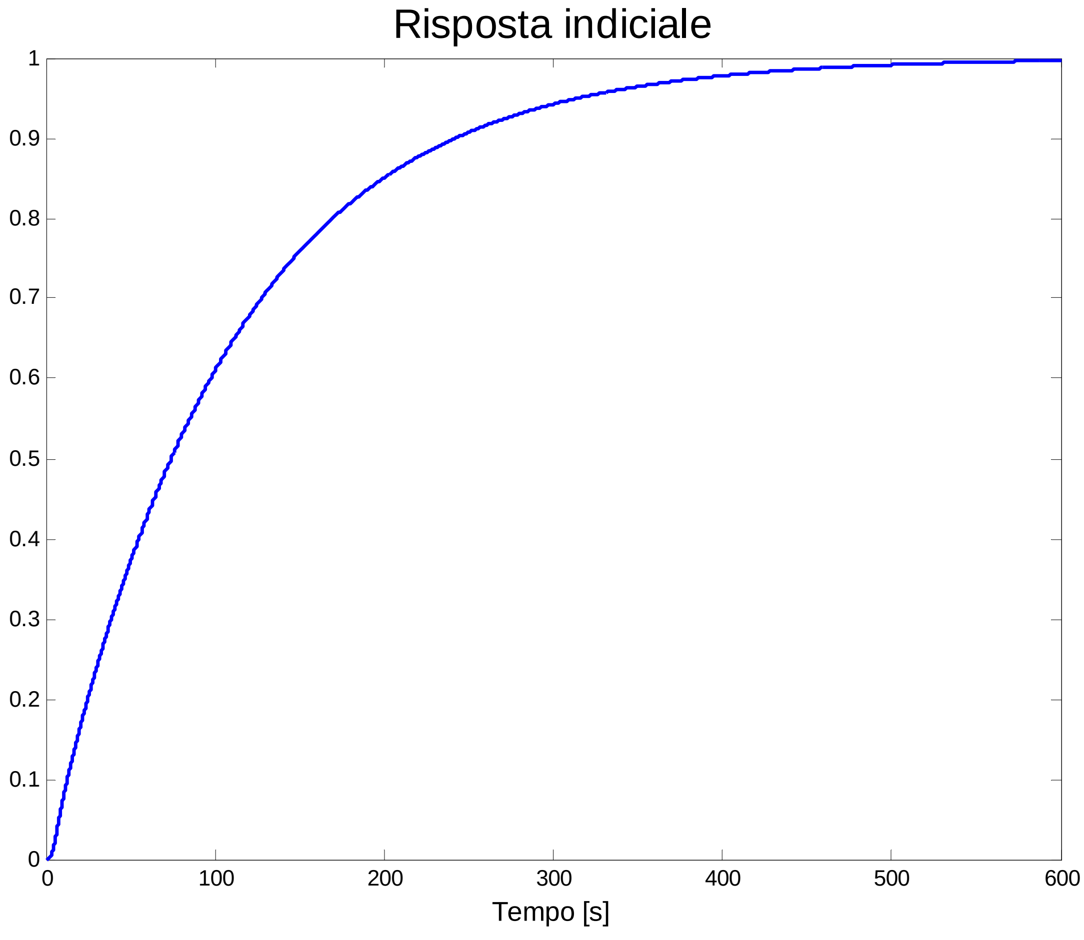
Smith predictor

The plant model without the delay is embedded into the controller to estimate the “real-time” output and a feedback loop is included to compensate for model mismatching



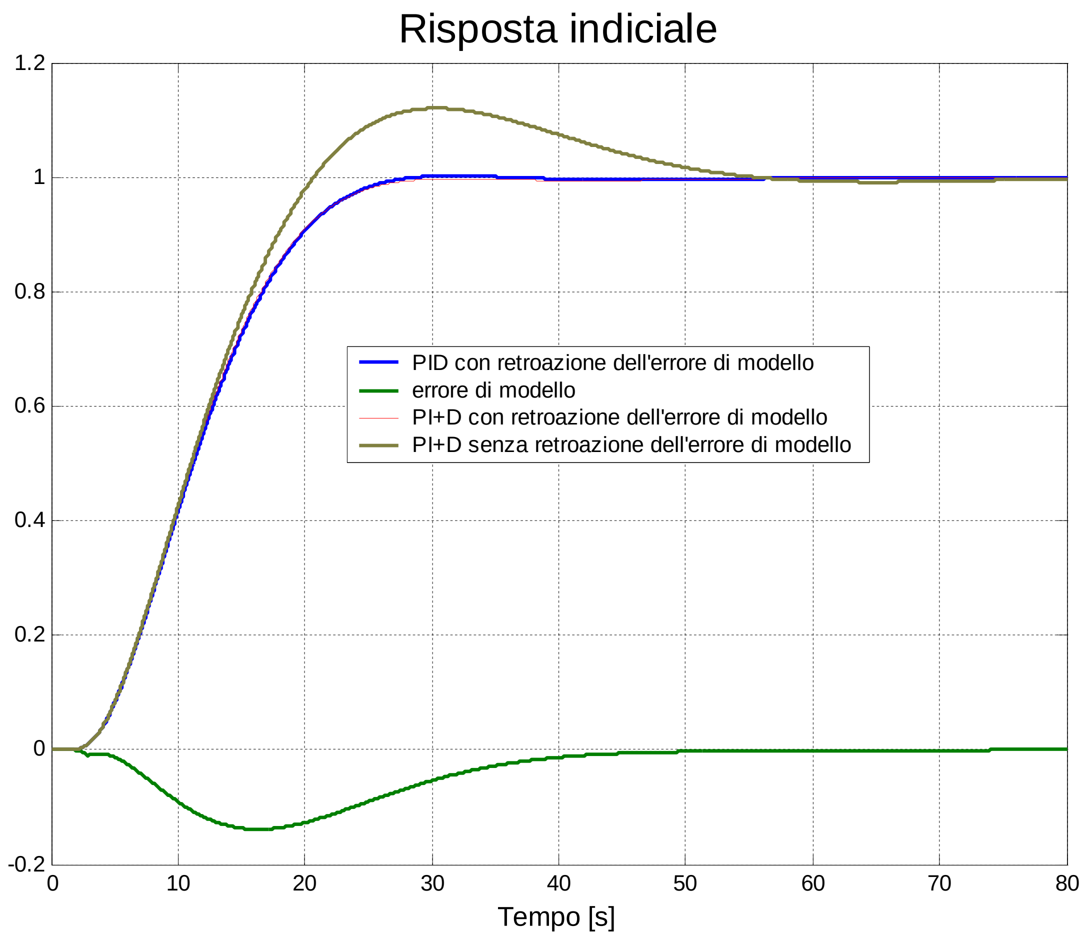
Smith predictor

The step response of the controlled plant without Smith predictor



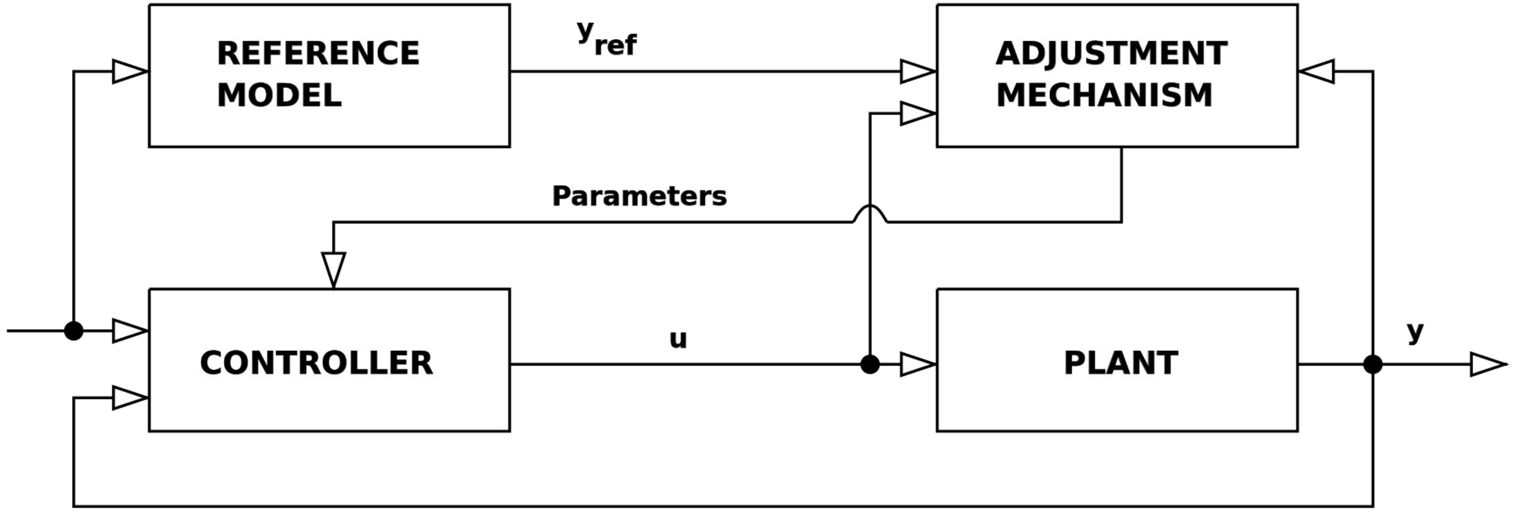
Smith predictor

The step response of the controlled plant with Smith predictor



Adaptive control

The parametric model of the plant is used and its parameters are idenfied to design the proper controller



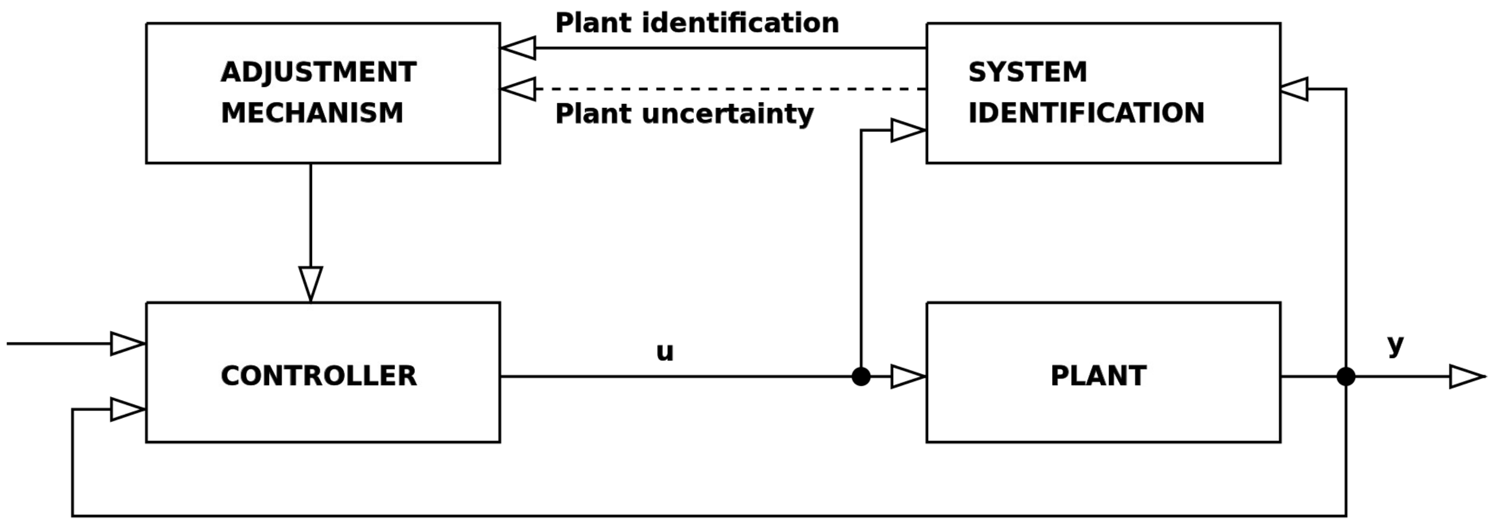
MODEL REFERENCE ADAPTIVE CONTROL (MRAC)

The controller parameters are adjusted such that the behaviour of the plant follows that of the reference model

SISO control schemes

Adaptive control

The parametric model of the plant is used and its parameters are idenfied to design the proper controller



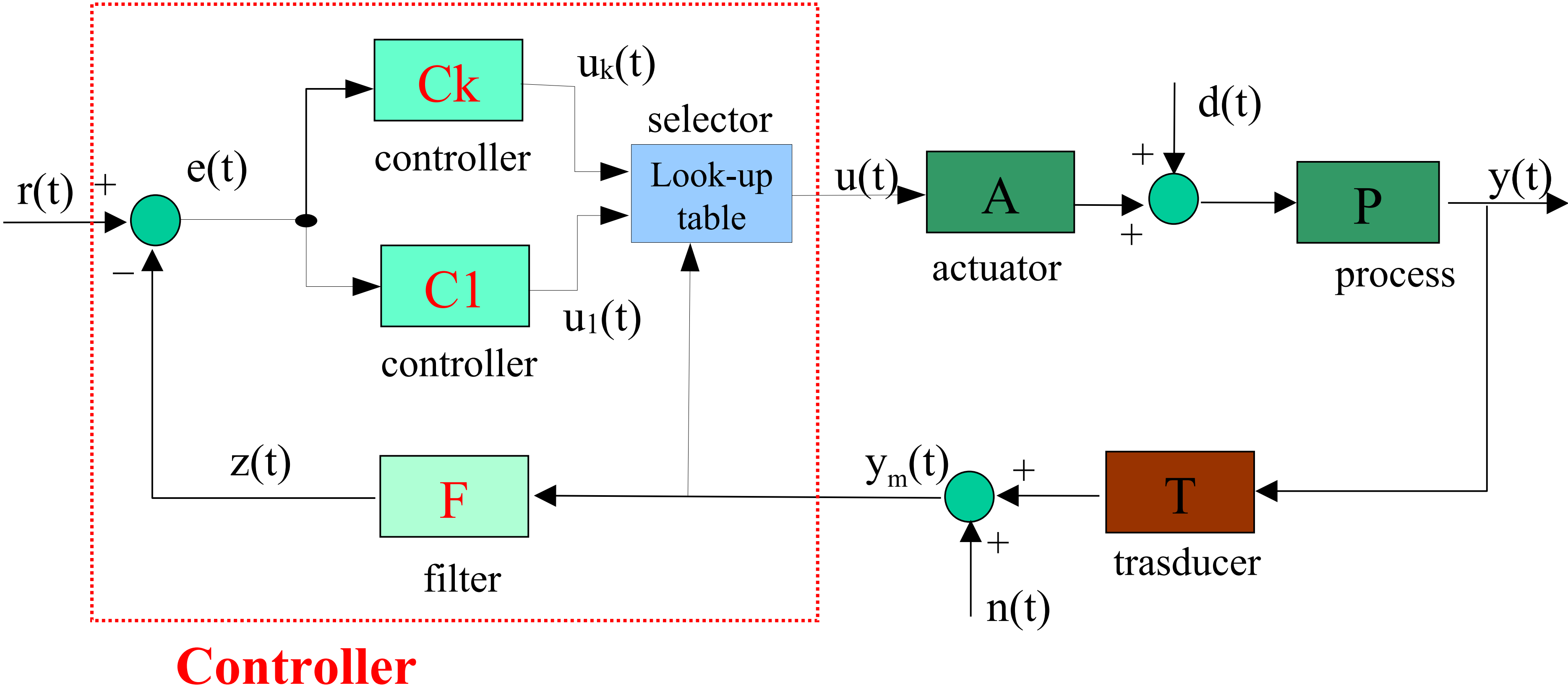
MODEL IDENTIFICATION ADAPTIVE CONTROL (MIAC)

The controller parameters are adjusted taking into account the actual/current parameters of the plant model

SISO control schemes

Gain scheduling control

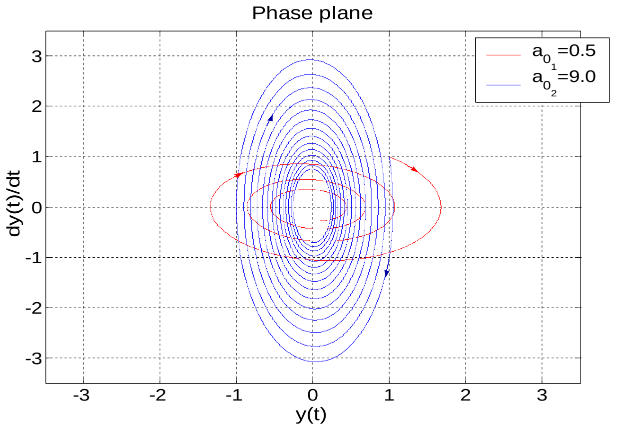
The controller is chosen among a set of controllers designed on the basis of linear models around different working points of a nonlinear plant



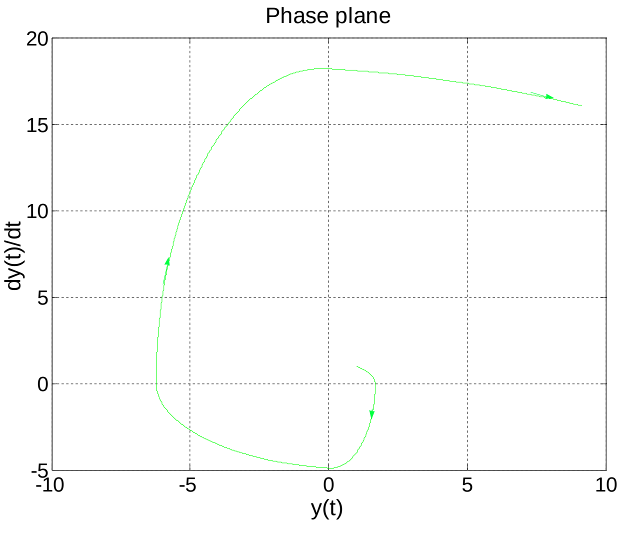
SISO control schemes

Gain scheduling control

No stability feature can be derived just considering the stability property of each feedback system



Switched unstable dynamics



$$\ddot{y} + a_1\dot{y} + a_0 y - \Delta a_0 |y| \operatorname{sgn}(\dot{y}) = 0$$

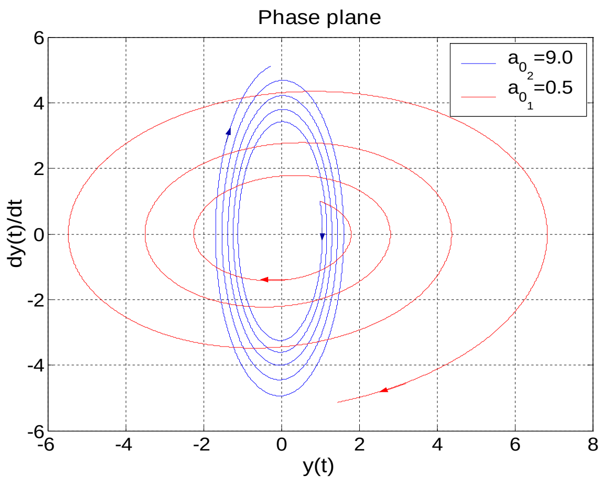
$$a_1 = 0.1 \quad a_0 = 0.7 \quad \Delta a_0 = 0.2$$

Both dynamics are asymptotically stable

SISO control schemes

Gain scheduling control

No stability feature can be derived just considering the stability property of each feedback system

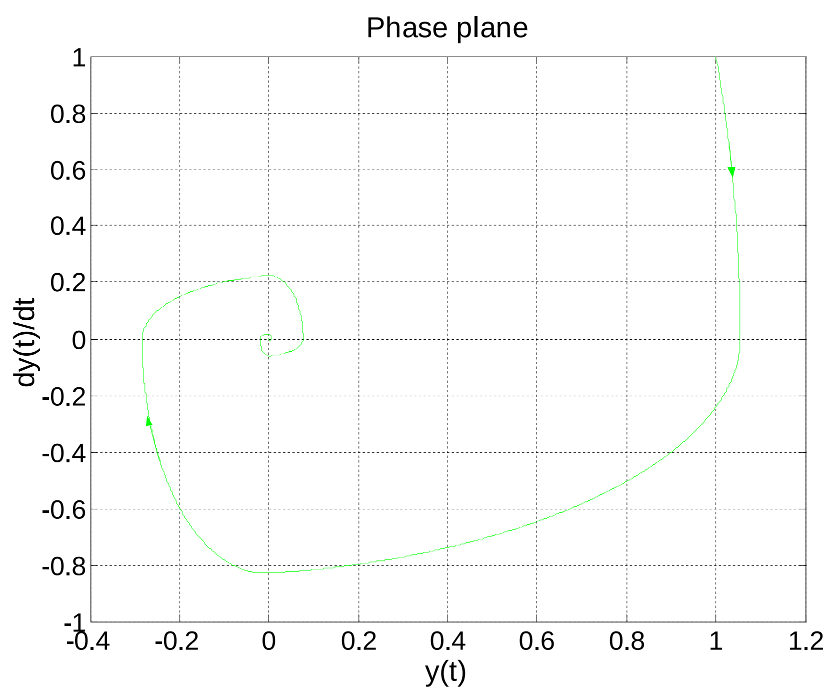


$$\ddot{y} + a_1 \dot{y} + a_0 y + \Delta a_0 \left| y \right| \operatorname{sgn}(\dot{y}) = 0$$

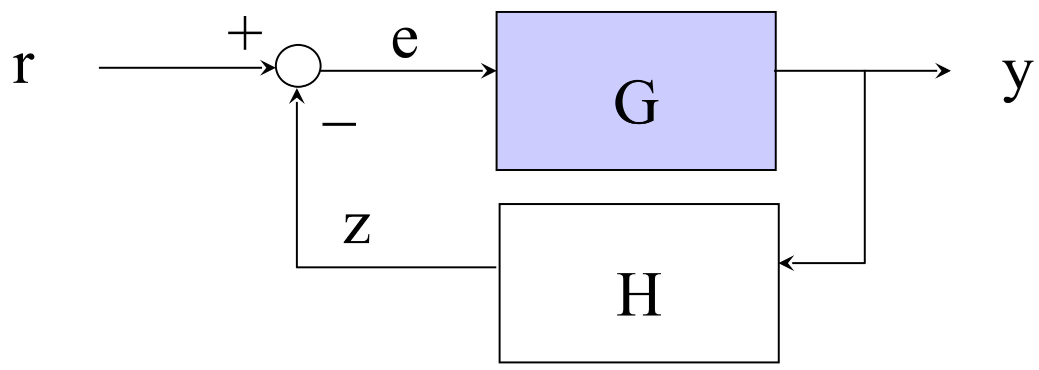
$$a_1 = -0.1 \quad a_0 = 0.7 \quad \Delta a_0 = 0.2$$

Both dynamics are unstable

Switched asymptotically stable dynamics



Robustness of SISO control schemes



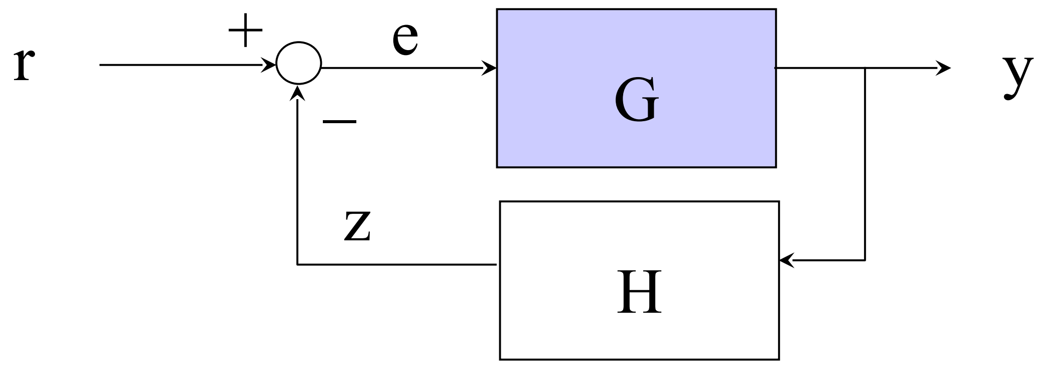
Robustness to variations of the system parameters.

The plant G can be affected by aging and wear such that its dynamics change with time.

Changes of the plant dynamics can be induced by external actions.

The control system should be able to limit the effect of changes in the subsystems dynamics on the output y

Robustness of SISO control schemes



$$W = \frac{G}{1+GH}$$

$$S_G^W = \frac{dW}{W} \bigg/ \frac{dG}{G} = \frac{dW}{dG}\frac{G}{W} = \frac{1}{1+GH}$$

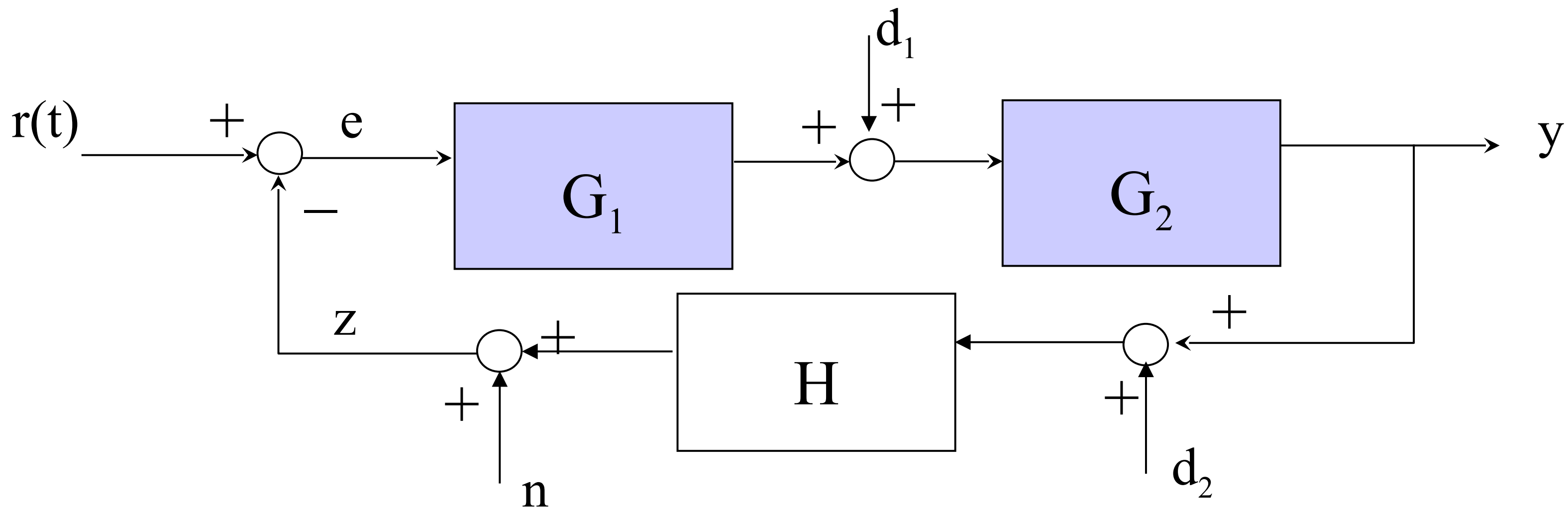
$$S_H^W = \frac{dW}{W} \bigg/ \frac{dH}{H} = \frac{dW}{dH}\frac{H}{W} = -\frac{GH}{1+GH}$$

The effect of the changes on G can be attenuated by its high gain.

The changes on H cannot be attenuated: sensors should be protected and reliable

Robustness of SISO control schemes

Robustness to disturbance: the output should be not sensitive to external uncontrolled inputs



$$W_r = \frac{G_1 G_2}{1 + G_1 G_2 H} \qquad W_{d_1} = \frac{G_2}{1 + G_1 G_2 H}$$

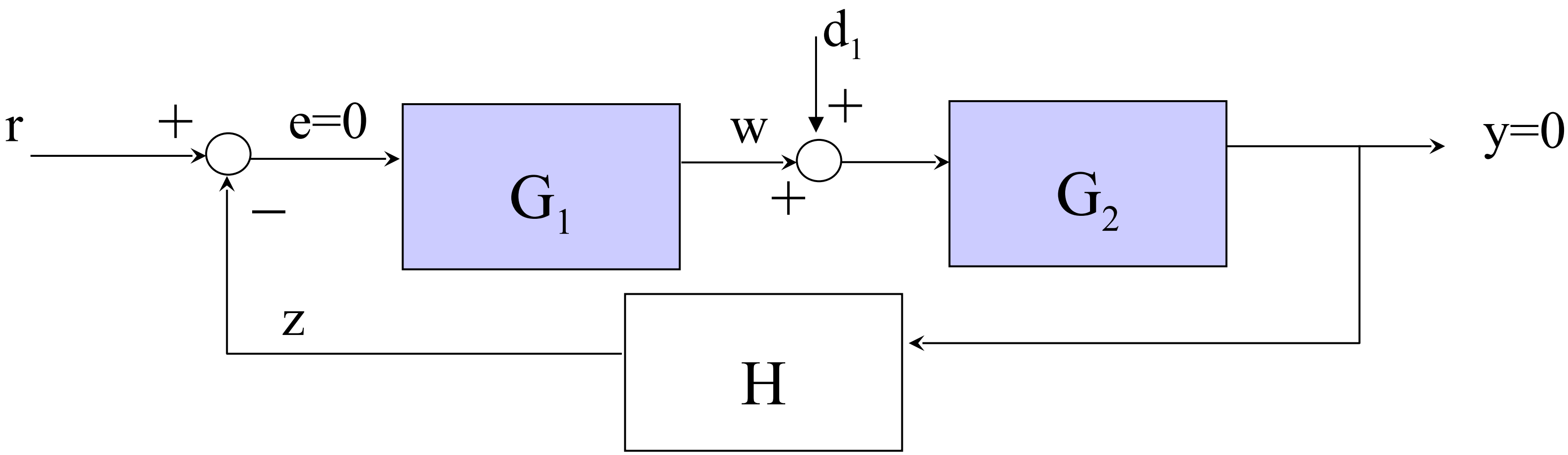
$$W_n = \frac{-G_1 G_2}{1 + G_1 G_2 H} \qquad W_{d_2} = \frac{-G_1 G_2 H}{1 + G_1 G_2 H}$$

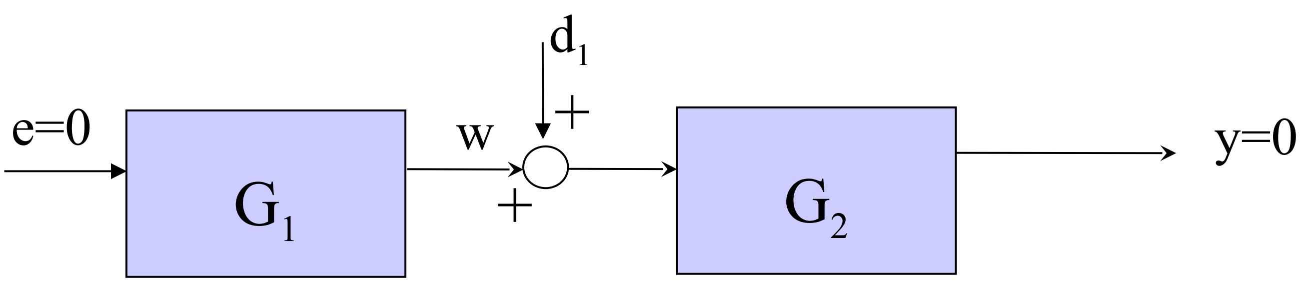
Having G_1 with a high modulus allows for attenuating the effect of disturbances acting on the direct path

Disturbances on the feedback can be hardly attenuated

Robustness of SISO control schemes

Robustness to disturbance: the output should be not sensitive to external uncontrolled inputs

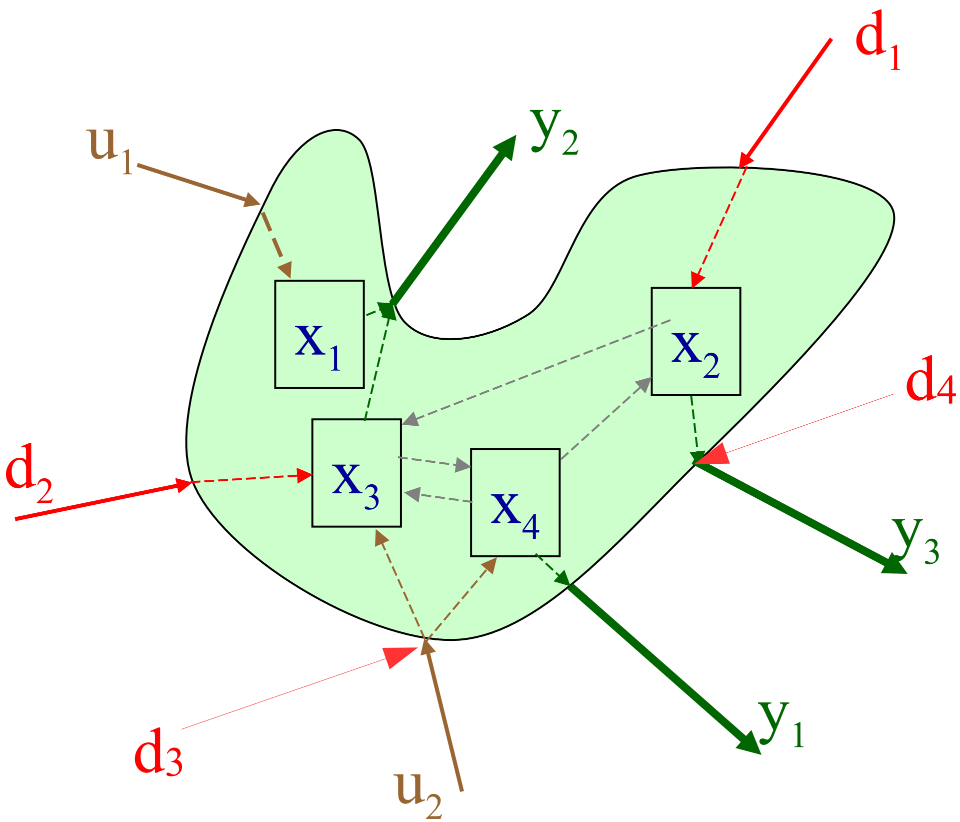




$w = -d_1$ → **G_1 (upstream) should "know" the structure of the disturbance for its complete rejection**

MIMO control schemes

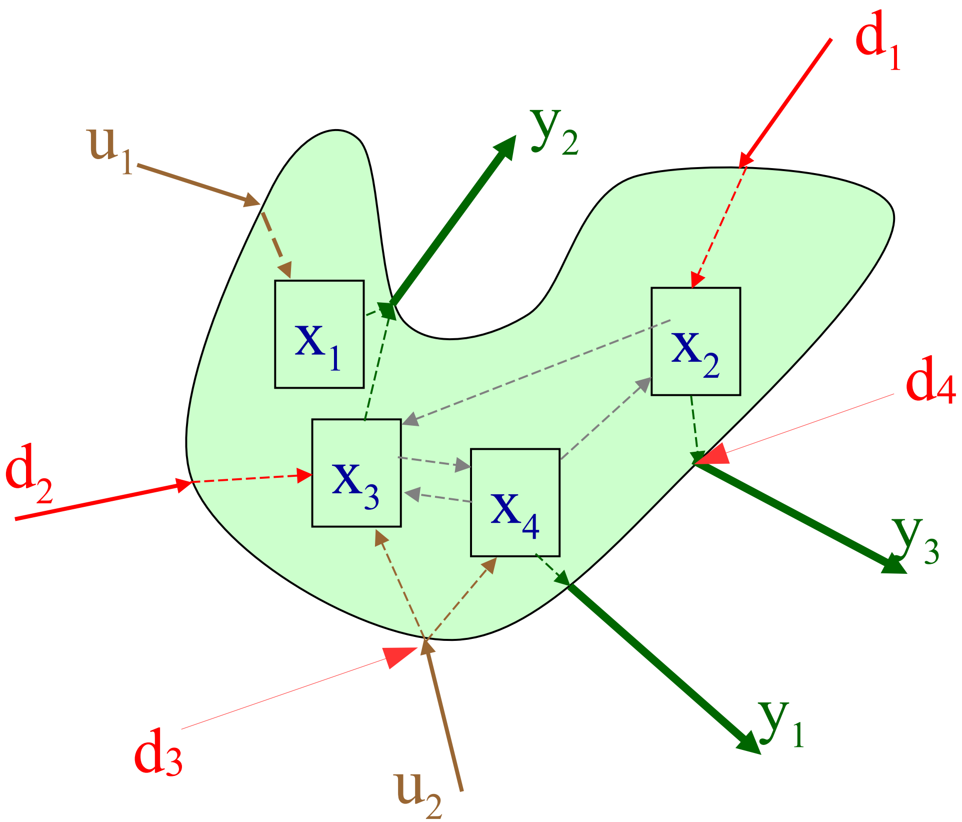
MIMO control systems are characterise by having a number of manipulated inputs and a number of measured variables (outputs)



u: input
d: disturbance
y: output
x: state

MIMO control schemes

Linear time-invariant MIMO systems can be represented in the state-space form by constant matrices



$$\dot{\boldsymbol{x}}(t) = A\,\boldsymbol{x}(t) + B\,\boldsymbol{u}(t) + E\,\boldsymbol{d}(t)$$
$$\boldsymbol{y}(t) = C\,\boldsymbol{x}(t) + D\,\boldsymbol{u}(t) + F\,\boldsymbol{d}(t)$$

$\boldsymbol{x} \in R^n$ $\quad \boldsymbol{u} \in R^q$ $\quad \boldsymbol{y} \in R^p$

$\boldsymbol{d} \in R^m$

MIMO control schemes

Linear time-invariant MIMO systems can be represented in the state-space form by constant matrices

$$\dot{\boldsymbol{x}}(t) = A\,\boldsymbol{x}(t) + B\boldsymbol{u}(t) + E\,\boldsymbol{d}(t)$$
$$\boldsymbol{y}(t) = C\,\boldsymbol{x}(t) + D\boldsymbol{u}(t) + F\,\boldsymbol{d}(t)$$

$\boldsymbol{x} \in R^n$ $\quad \boldsymbol{u} \in R^q$ $\quad \boldsymbol{y} \in R^p$ $\quad \boldsymbol{d} \in R^m$

A represents the connections among the systems components

B represents how the inputs act on the system

E represents how the disturbances act on the system

C represents how the measurements depend on the system's internal energy

D represents the direct/instantaneous influence of the inputs to the measurements

F represents the effect of the interferences on the measurements

MIMO control schemes

Linear time-invariant MIMO systems can be represented in the state-space form by constant matrices

$$\dot{\boldsymbol{x}}(t) = A\boldsymbol{x}(t) + B\boldsymbol{u}(t) + E\boldsymbol{d}(t)$$
$$\boldsymbol{y}(t) = C\boldsymbol{x}(t) + D\boldsymbol{u}(t) + F\boldsymbol{d}(t)$$

$\boldsymbol{x} \in R^n$ $\boldsymbol{u} \in R^q$ $\boldsymbol{y} \in R^p$ $\boldsymbol{d} \in R^m$

Laplace transform

$$s\boldsymbol{X}(s) - \boldsymbol{x}(0) = A\boldsymbol{X}(s) + B\boldsymbol{U}(s) + E\boldsymbol{D}(s)$$
$$\boldsymbol{Y}(t) = C\boldsymbol{X}(s) + D\boldsymbol{U}(s) + F\boldsymbol{D}(s)$$

$$\boldsymbol{Y}(s) = C(sI - A)^{-1}\boldsymbol{x}(0) + \left[C(sI - A)^{-1}B + D\right]\boldsymbol{U}(s) + \left[C(sI - A)^{-1}E + F\right]\boldsymbol{D}(s)$$

MIMO control schemes

Linear time-invariant MIMO systems can be represented in the Input-Output form by means of the Transfer Matrix whose elements are Tranfer Functions

$$\boldsymbol{Y_f}(s) = G(s)\,\boldsymbol{U}(s) + H(s)\,\boldsymbol{D}(s)$$

$$G(s) = \begin{bmatrix} G(s)_{11} & \dots & G(s)_{1q} \\ \vdots & \ddots & \vdots \\ G(s)_{p1} & \dots & G(s)_{pq} \end{bmatrix} \qquad H(s) = \begin{bmatrix} H(s)_{11} & \dots & H(s)_{1m} \\ \vdots & \ddots & \vdots \\ H(s)_{p1} & \dots & H(s)_{pm} \end{bmatrix}$$

$$Y_{f_i}(t) = \sum_{j=1}^{q} G_{ij}(s)\,U_j(s) + \sum_{h=1}^{m} H_{ij}(s)\,D_h(s)$$

Multivariabile control

The control actions on the plant are implemented by single-loop schemes somehow coordinated and interactions are considered similar to disturbances

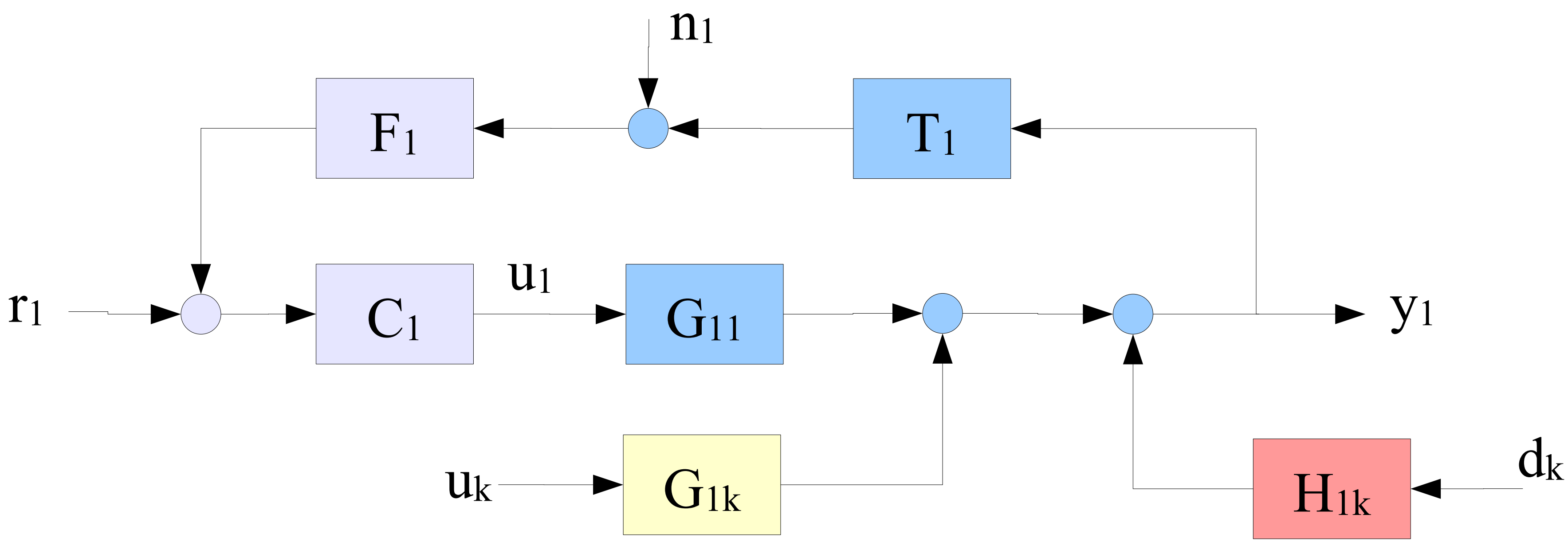
$$\boldsymbol{Y}_{\boldsymbol{f}}(s) = G(s)\,\boldsymbol{U}(s) + H(s)\,\boldsymbol{D}(s)$$

$$G(s) = \begin{bmatrix} G(s)_{11} & \dots & G(s)_{1q} \\ \vdots & \ddots & \vdots \\ G(s)_{p1} & \dots & G(s)_{pq} \end{bmatrix} \qquad H(s) = \begin{bmatrix} H(s)_{11} & \dots & H(s)_{1m} \\ \vdots & \ddots & \vdots \\ H(s)_{p1} & \dots & H(s)_{pm} \end{bmatrix}$$

$$Y_{f_i}(t) = \sum_{j=1}^{q} L^{-1}\left\{G_{ij}(s)\,U_j(s)\right\} + \sum_{h=1}^{m} L^{-1}\left\{H_{ij}(s)\,D_h(s)\right\}$$

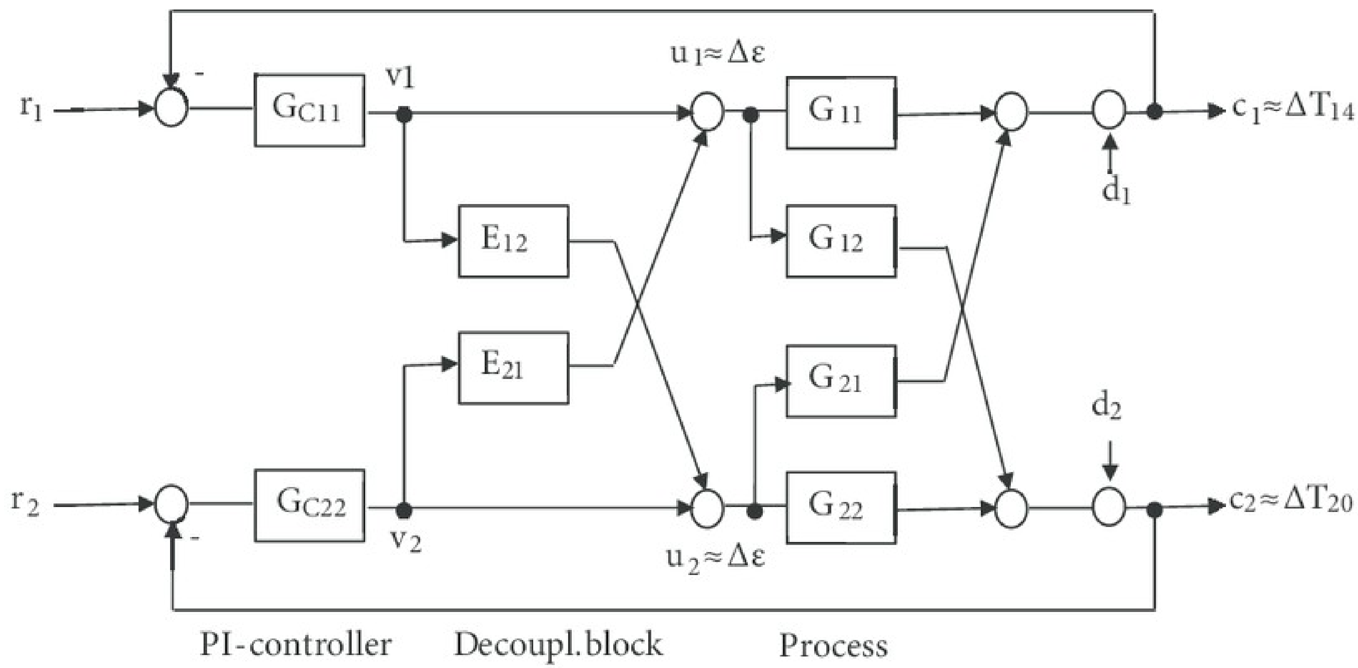
Multivariabile control

The control actions on the plant are implemented by single-loop schemes somehow coordinated and interactions are considered similar to disturbances



Multivariabile control

The control actions on the plant are implemented by single-loop schemes somehow coordinated and interactions are considered similar to disturbances



LQR control

The control actions on the plant are implemented by a state feedback scheme with the feedback gains (*direct disturbances on the output are classified as noise*)

$$\dot{\boldsymbol{x}}(t) = A\,\boldsymbol{x}(t) + B\boldsymbol{u}(t) + E\,\boldsymbol{d}(t)$$
$$\boldsymbol{y}(t) = C\,\boldsymbol{x}(t) + D\boldsymbol{u}(t) + F\,\boldsymbol{n}(t)$$

$$\boldsymbol{u}(t) = K\,\boldsymbol{x}(t)$$

$$\dot{\boldsymbol{x}}(t) = (A + BK)\,\boldsymbol{x}(t) + E\,\boldsymbol{d}(t)$$
$$\boldsymbol{y}(t) = (C + DK)\,\boldsymbol{x}(t) + F\,\boldsymbol{n}(t)$$

The control is chosen such that a performance index is minimized

$$J(\boldsymbol{u}) = \int_{t=0}^{\infty} \left[\boldsymbol{x}^{T}(t)\,Q\,\boldsymbol{x}(t) + \boldsymbol{u}^{T}(t)\,R\,\boldsymbol{u}(t) \right] dt$$

MIMO control schemes

LQG control

The state feedback control actions on the plant are implemented using the state extimates from an optimal observer that is less sensitive from disturbance and noise

$$\dot{\boldsymbol{x}}(t)=A\,\boldsymbol{x}(t)+B\boldsymbol{u}(t)+E\,\boldsymbol{d}(t)$$
$$\boldsymbol{y}(t)=C\,\boldsymbol{x}(t)+D\boldsymbol{u}(t)+F\,\boldsymbol{n}(t)$$

$$\boldsymbol{u}(t)=K\,\hat{\boldsymbol{x}}(t)$$

$$\dot{\hat{\boldsymbol{x}}}(t)=A\,\hat{\boldsymbol{x}}(t)+B\,\boldsymbol{u}(t)+L\left[\hat{\boldsymbol{y}}(t)-\boldsymbol{y}(t)\right]$$
$$\hat{\boldsymbol{y}}(t)=C\,\hat{\boldsymbol{x}}(t)+D\boldsymbol{u}(t)$$

Kalman filter

The gain matrices *K* and *L* are chosen such that a performance indexes are independently minimized

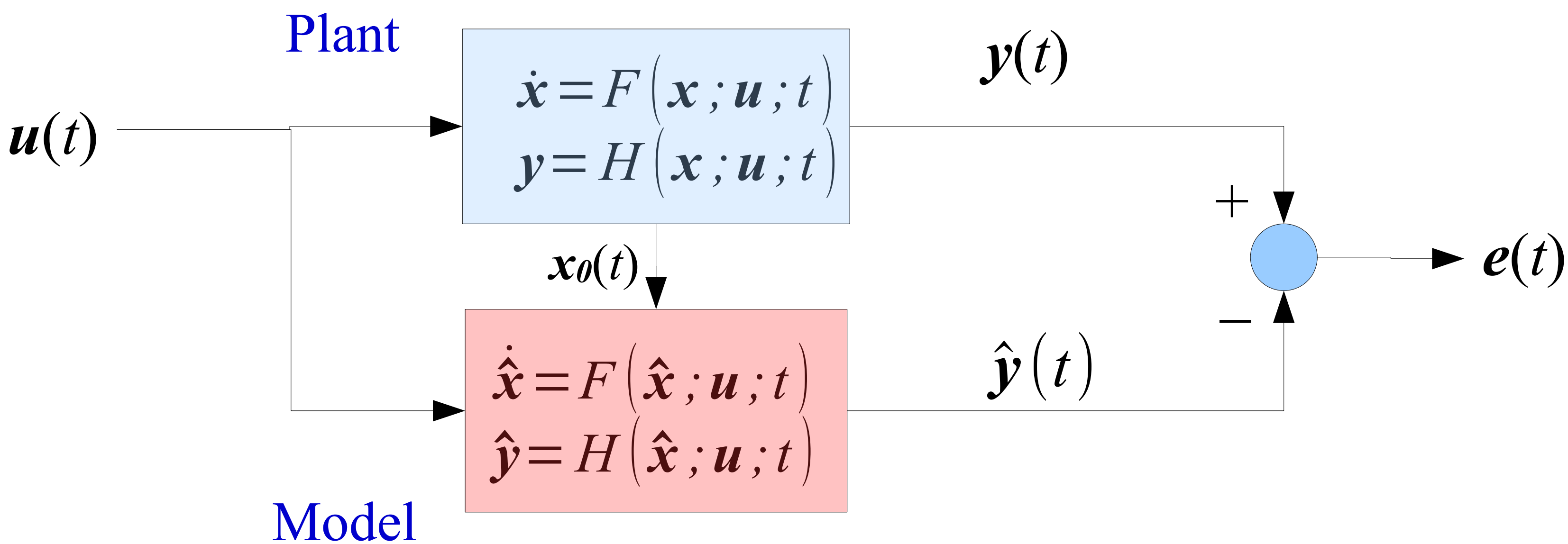
$$J_{contr}=\int_{t=0}^{\infty}\left[\boldsymbol{x}^{T}(t)\,Q\,\boldsymbol{x}(t)+\boldsymbol{u}^{T}(t)\,R\,\boldsymbol{u}(t)\right]dt$$

$$J_{obs}=\int_{t=0}^{\infty}\left[\boldsymbol{x}^{T}(t)\,\tilde{Q}_{d}\,\boldsymbol{x}(t)+\boldsymbol{y}^{T}(t)\,\tilde{R}_{n}\,\boldsymbol{y}(t)\right]dt$$

Observers for model-based FDI

The observer

It is a copy of the process model which has an additional input that takes into account the difference between the estimated and the actual output.



If the error is not zero, possibly a fault is present. (*parity check*)

Observers for model-based FDI

The observer

It is a copy of the process model which has an additional input that takes into account the difference between the estimated and the actual output.

System dynamics
$$\dot{\boldsymbol{x}} = F\left(\boldsymbol{x}\,;\boldsymbol{u}\,;t\right)$$
$$\boldsymbol{y} = H\left(\boldsymbol{x}\,;\boldsymbol{u}\,;t\right)$$

State observer
$$\dot{\hat{\boldsymbol{x}}} = F\left(\hat{\boldsymbol{x}}\,;\boldsymbol{u}\,;t\right) + G\left(\hat{\boldsymbol{y}} - \boldsymbol{y}\,;\right)$$
$$\hat{\boldsymbol{y}} = H\left(\hat{\boldsymbol{x}}\,;\boldsymbol{u}\,;t\right)$$

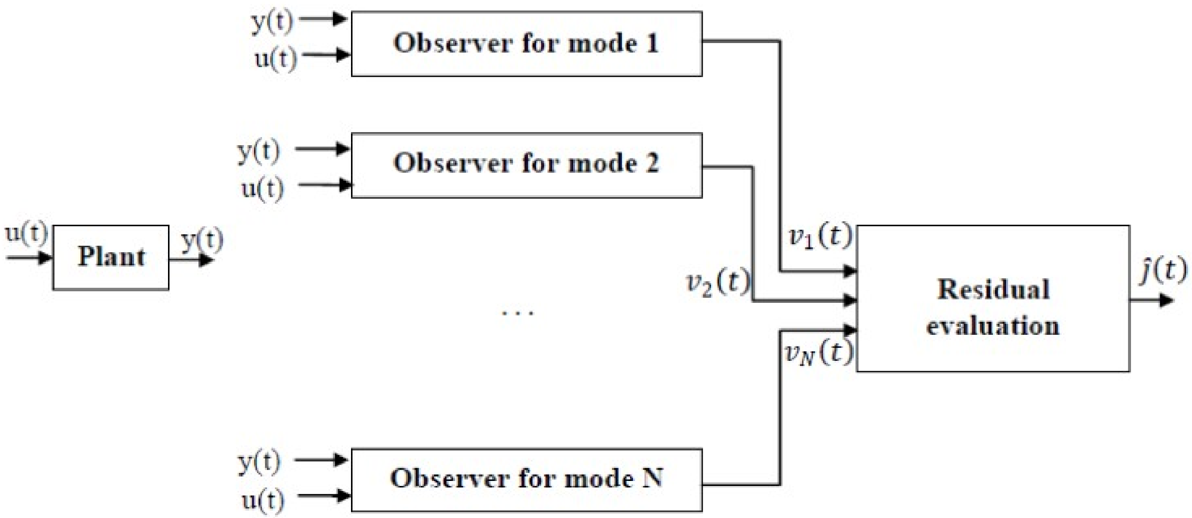
Output injection

The output injection is designed to drive the estimation error to zero.
If it is not, possibly a fault is present.

Observers for model-based FDI

The observer

A stack of observers can be used to detect faults or different operating conditions

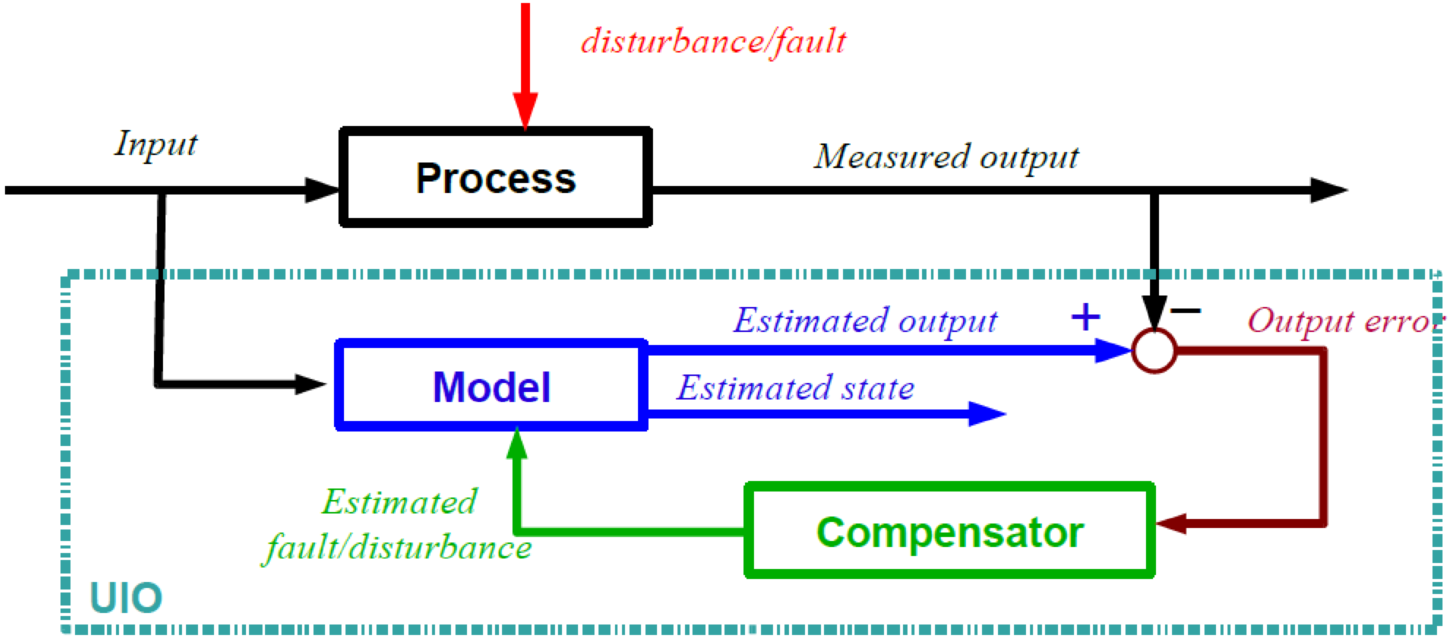


$$f(t) = \arg\left(\min_i\left(\int_{t-T}^{t} |v_i(\tau)|\, d\tau\right)\right)$$

The output injection will be zero only for the curent operating or faulty condition.

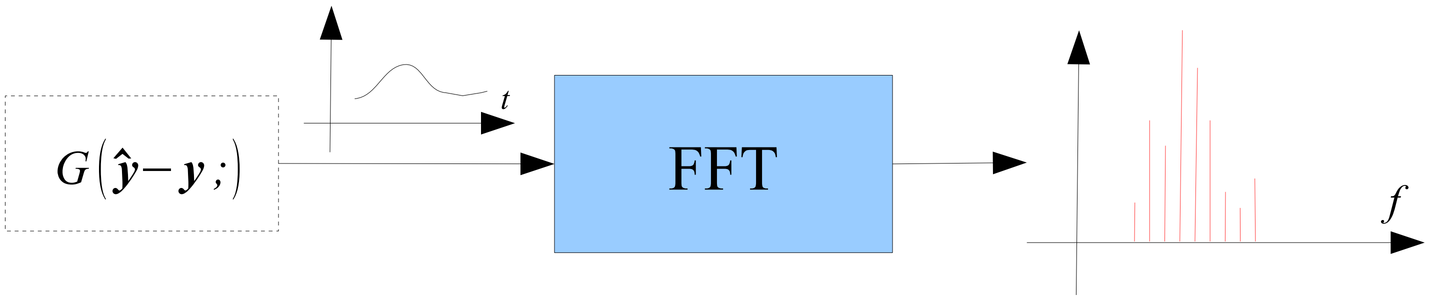
The Unknown Input Observer

It is designed as an observer but exploits a structural feature of the plant such that the output error is zero if and only if the state estimation error is zero.



Combining UIO and data analysis

The output injection of a UIO contains informations on the fault. By analysing the time serie of the output injection signal some characteristics of the fault can be derived.



Combining UIO and data analysis

The application to rotor broken bar diagnosis.

System model

$$\begin{cases} \dot{x}_1 = a_1\left(x_3x_4 - x_2x_5\right) - a_2x_1 + a_3T_L \\ \dot{x}_2 = b_1x_4 - b_2x_2 + b_3x_1x_3 + b_4u_{s_\alpha} \\ \dot{x}_3 = b_1x_5 - b_2x_3 - b_3x_1x_2 + b_4u_{s_\beta} \\ \dot{x}_4 = c_1x_2 - c_2x_4 - n_px_1x_5 \\ \dot{x}_5 = c_1x_3 - c_2x_5 + n_px_1x_4 \end{cases}$$

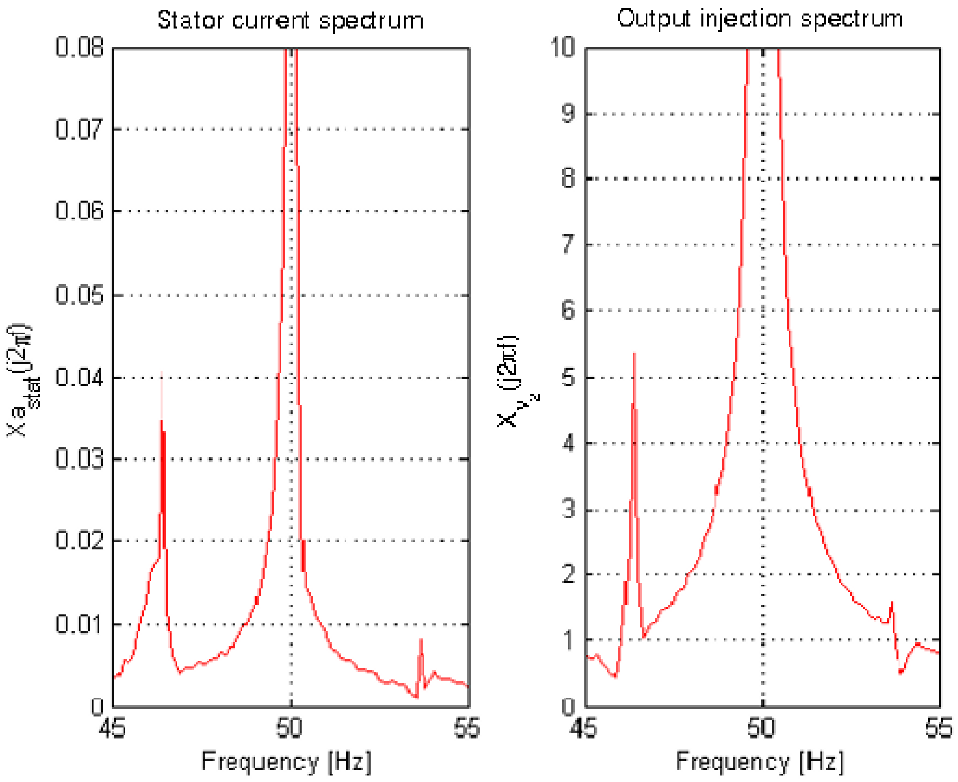
UIO

$$\begin{cases} \dot{\hat{x}}_1 = a_1\left(x_3\hat{x}_4 - x_2\hat{x}_5\right) - a_2x_1 + a_3\nu_1 \\ \dot{\hat{x}}_2 = b_1\hat{x}_4 - b_2x_2 + b_3x_1\hat{x}_5 + b_4(u_{s\alpha} + \nu_2) \\ \dot{\hat{x}}_3 = b_1\hat{x}_5 - b_2x_3 - b_3x_1\hat{x}_4 + b_4(u_{s\beta} + \nu_3) \\ \dot{\hat{x}}_4 = c_1x_2 - c_2\hat{x}_4 - n_px_1\hat{x}_5 \\ \dot{\hat{x}}_5 = c_1x_3 - c_2\hat{x}_5 + n_px_1\hat{x}_4 \end{cases}$$

Output injections

Combining UIO and data analysis

The application to rotor broken bar diagnosis.

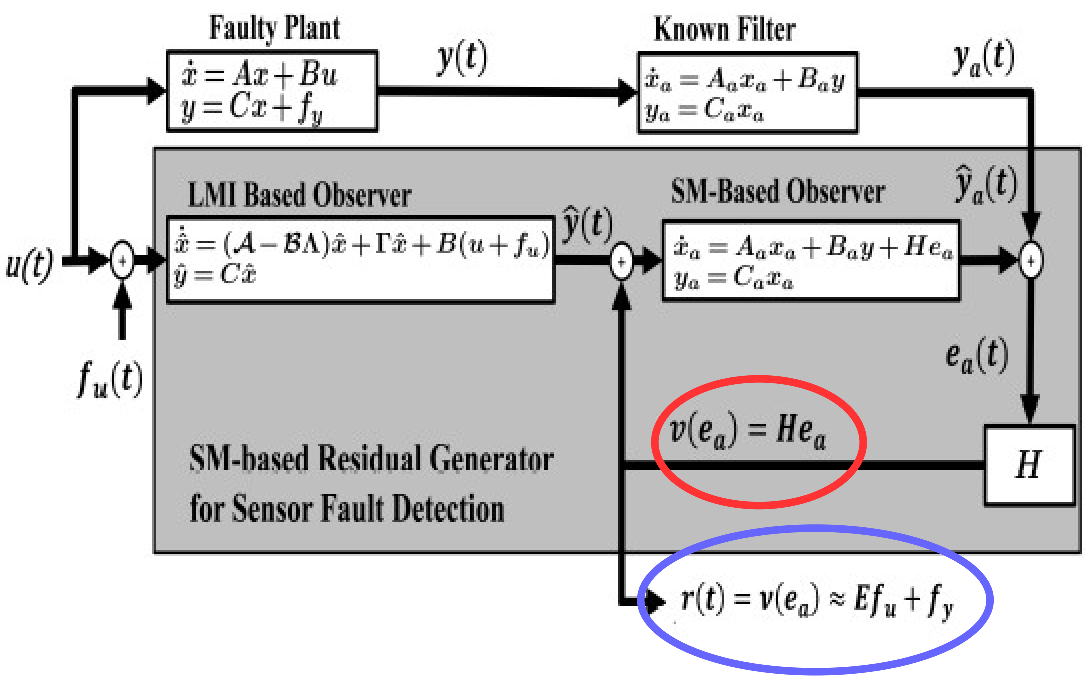


Combining UIO and data analysis

The output injection of a UIO contains informations on the fault. By analysing the time serie of the output injection signal some characteristics of the fault can be derived.

Combining UIO and data analysis

The application to the diagnosis of a steam separator drum.



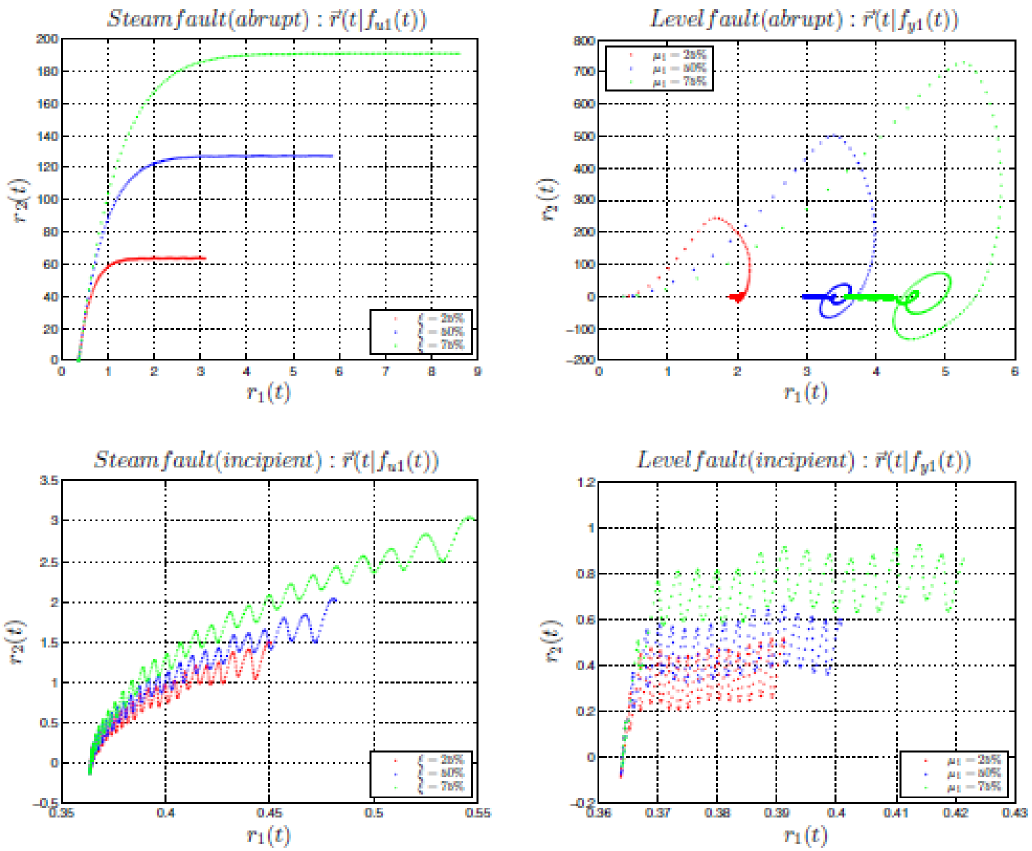
Output injection

residuals

Observers for model-based FDI

Combining UIO and data analysis

The application to the diagnosis of a steam separator drum.

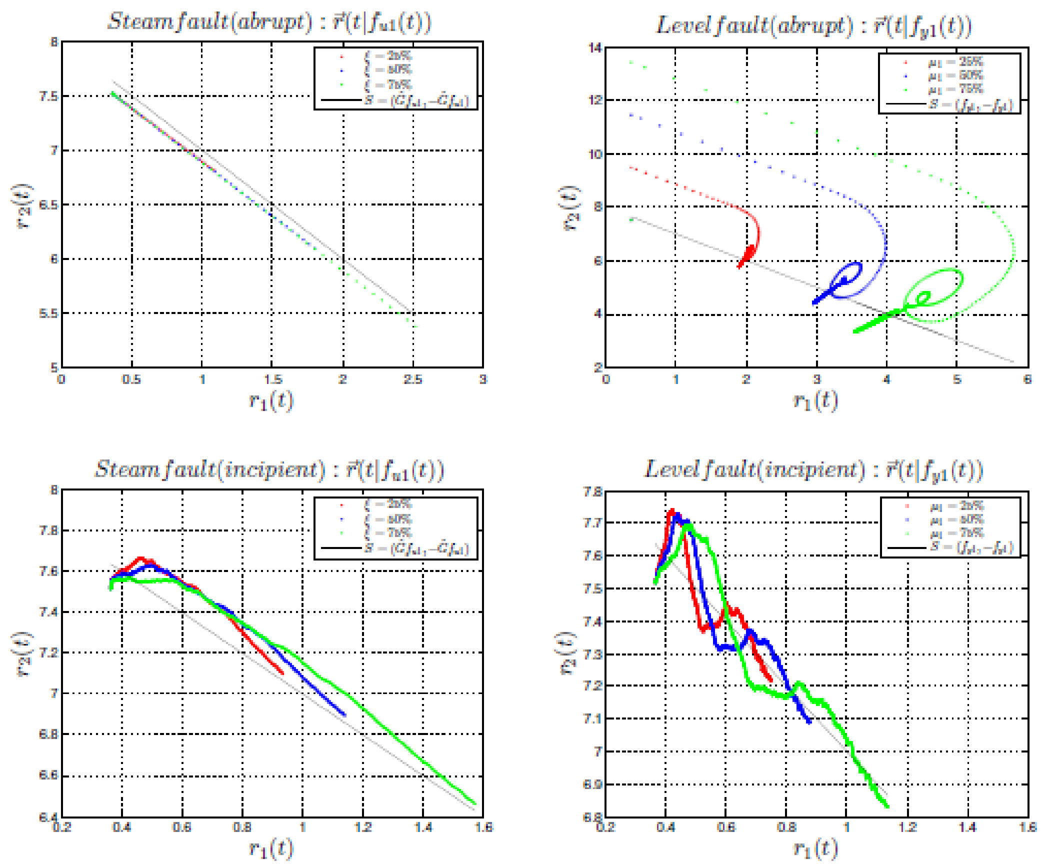


Method 1

Each fault, either matching or on the sensor, has its own shape/signature in the residuals' domain.

Combining UIO and data analysis

The application to the diagnosis of a steam separator drum.

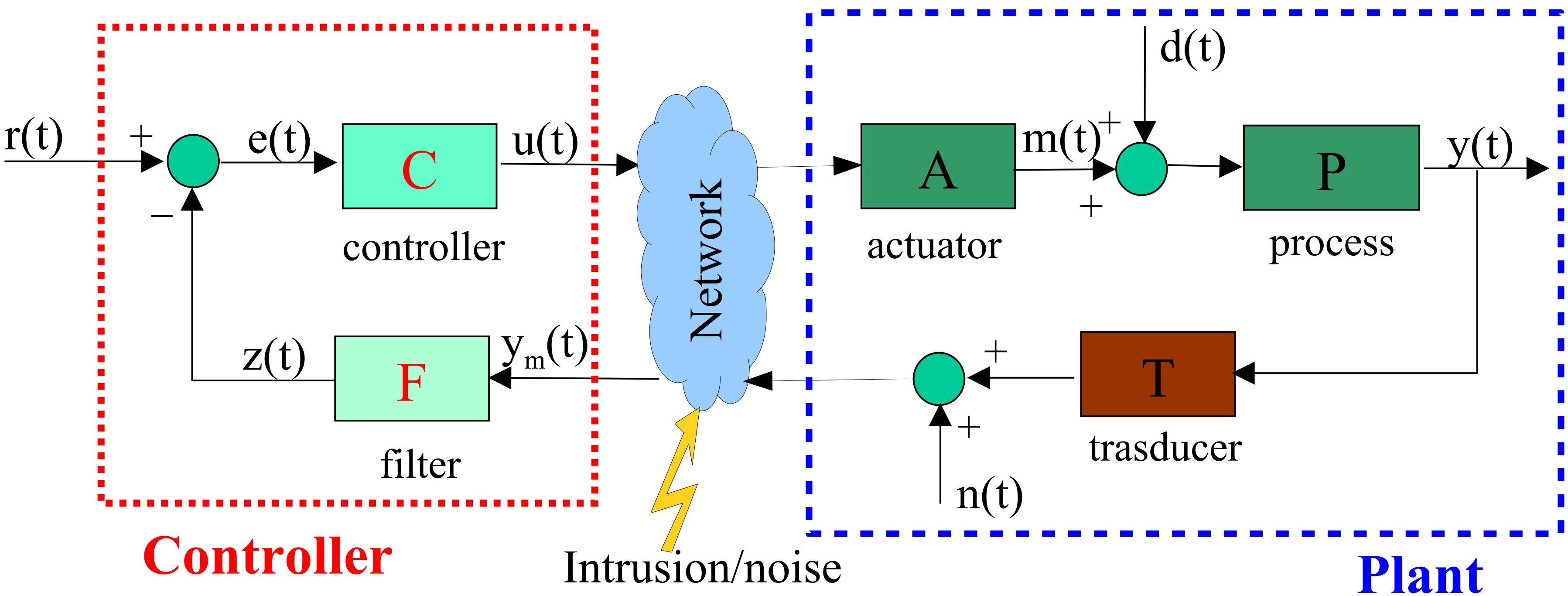


Method 2

Each fault, either matching or on the sensor, has its own shape/signature in the residuals' domain.

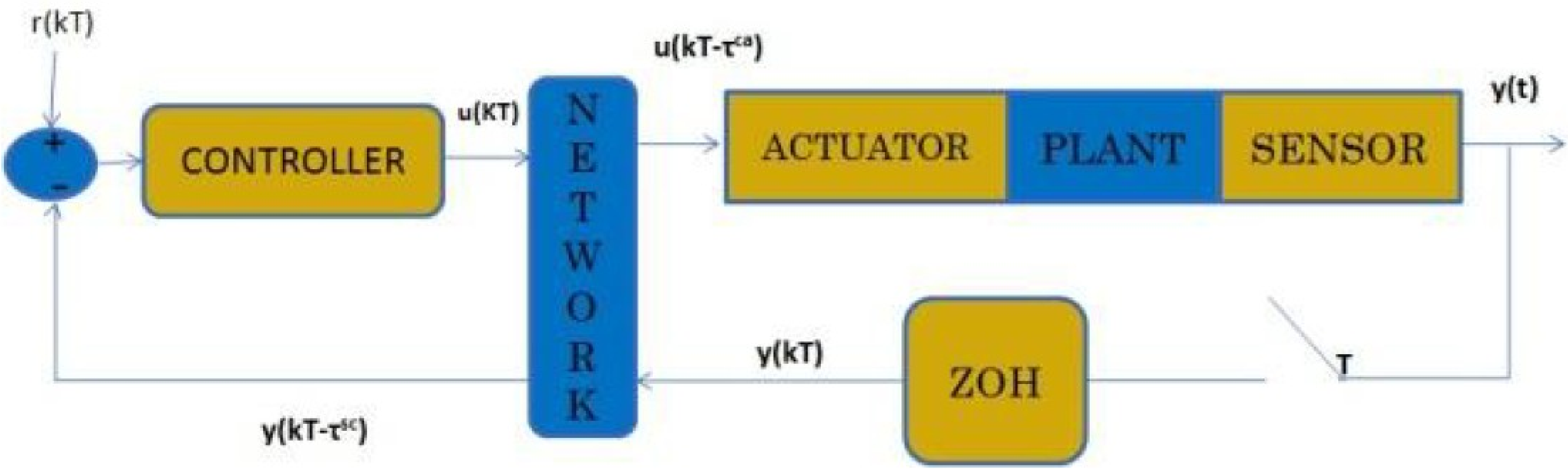
Networked control

The connection between the control and the plant is implemented by means of a network (public via VPN or Fieldbus)



Networked control

The connection between the control and the plant is implemented by means of a network (public via VPN or Fieldbus).



Communication is digitally implemented and delays are usually induced by networks. Even packet drops can occur unless a deterministic network is implemented.

Networked control

Generally, stability problems arise in the presence of delays either due to the process itselt or because of the communication network.

Uncertainty in the delay amount makes the use of delay compensation approaches less, or even not at all, effective.

Packets dropout can be considered as an extreme infinite delay when designing the control law.

Networks are the potential origin of additional disturbances to the system.

Networked control

Packet losses or delays larger than designed can be faced at the actuator side by:

- Hold the previous value of the manipulated variable
- Set the manipulated variable at the nominal value
- Use prediction techniques based on a plant model

Transmitting only the variations of the variables can decrease the loss of packets because of network congestion

Disturbances on the communication network are not taken into account in the above design approaches

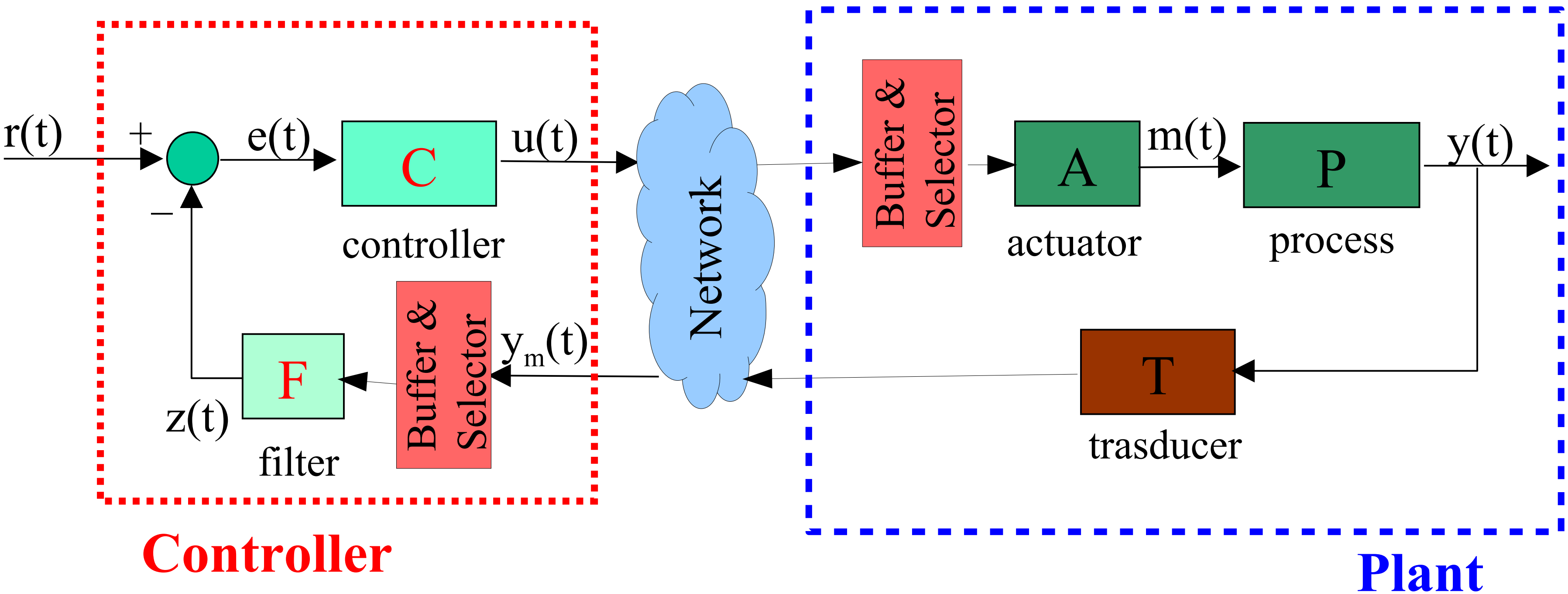
Networked predictive control

A number of future control actions, designed by means of an observer, are sent to the actuator that can use them in the case of missing/delayed command from the controller.

- A microprocessor is needed on the actuator
- Time stamp of the variable is needed
- Sensor should send an array with all data not transmitted yet
- Comparison between the data in the registers and the received/computed values should be effectively implemented

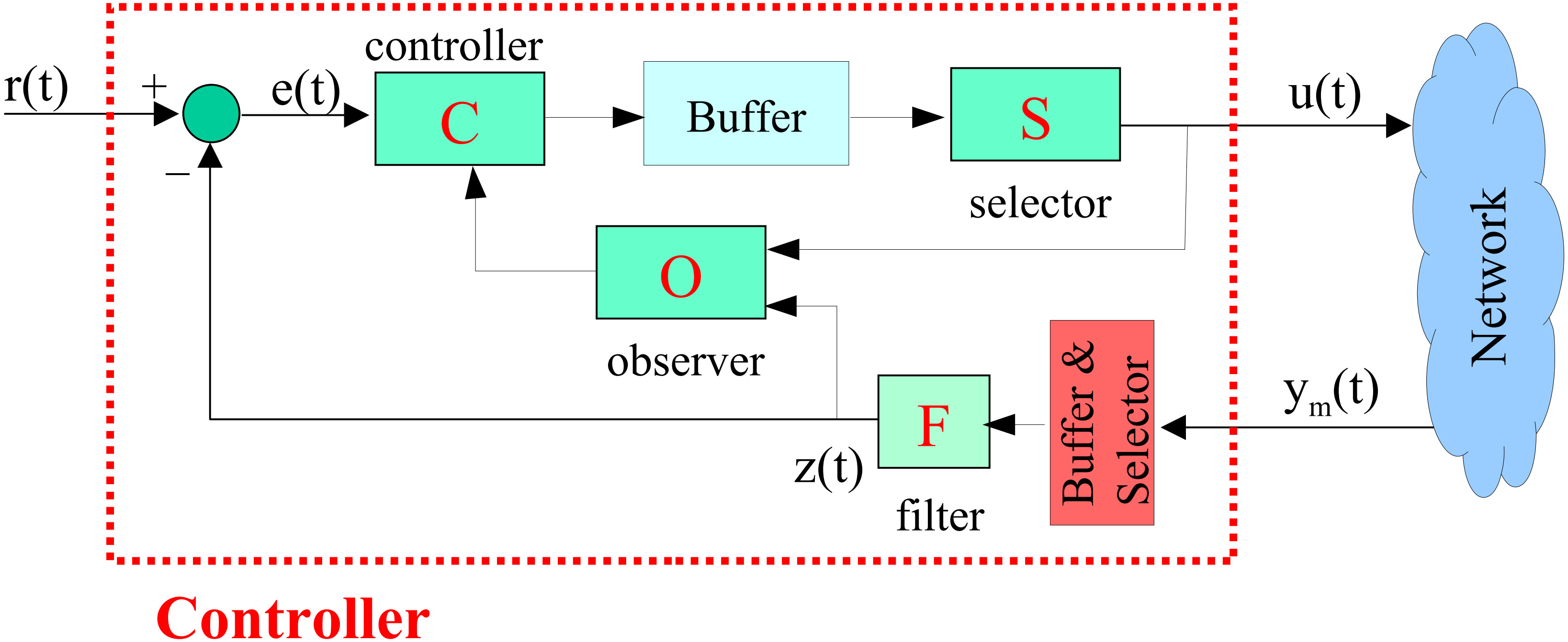
Networked predictive control

A number of future control actions, designed by means of an observer, are sent to the actuator that can use them in the case of missing/delayed command from the controller.



Networked predictive control

A number of future control actions, designed by means of an observer, are sent to the actuator that can use them in the case of missing/delayed command from the controller.



Networked predictive control

The observer

$$\hat{x}_{t-k+1|t-k} = A\hat{x}_{t-k|t-k-1} + Bu_{t-k} + L(y_{t-k} - \hat{y}_{t-k})$$

$$\hat{y}_{t-k} = C\hat{x}_{t-k|t-k-1}$$

The control stack at the actuator over the orizon M

$$\begin{bmatrix} u(t-k-2|t-k-2) \\ u(t-k-1|t-k-2) \\ . \\ . \\ . \\ u(t|t-k-2). \\ . \\ . \\ u(t+M-k-3|t-k-2) \end{bmatrix}, \begin{bmatrix} u(t-k-1|t-k-1) \\ u(t-k|t-k-1) \\ . \\ . \\ . \\ u(t|t-k-1). \\ . \\ . \\ u(t+M-k-2|t-k-1) \end{bmatrix}$$

$$u_t = u(t+\tau|t-k) = K\hat{x}_{t+\tau|t-k}$$

The control law

Networked control

Networked predictive control

The use of the predicted control

Index	t+1	t+2	t+3
Data	k	X	Y

Example of application in ideal case without data loss

Index	t+1	t+2	t+3=loss
Data	K	X	?
Data Predicted	X	Y	?

Example of application of the prediction in case of data loss

Networked predictive control

For the predicted control design some bound for the uncertain delays are needed

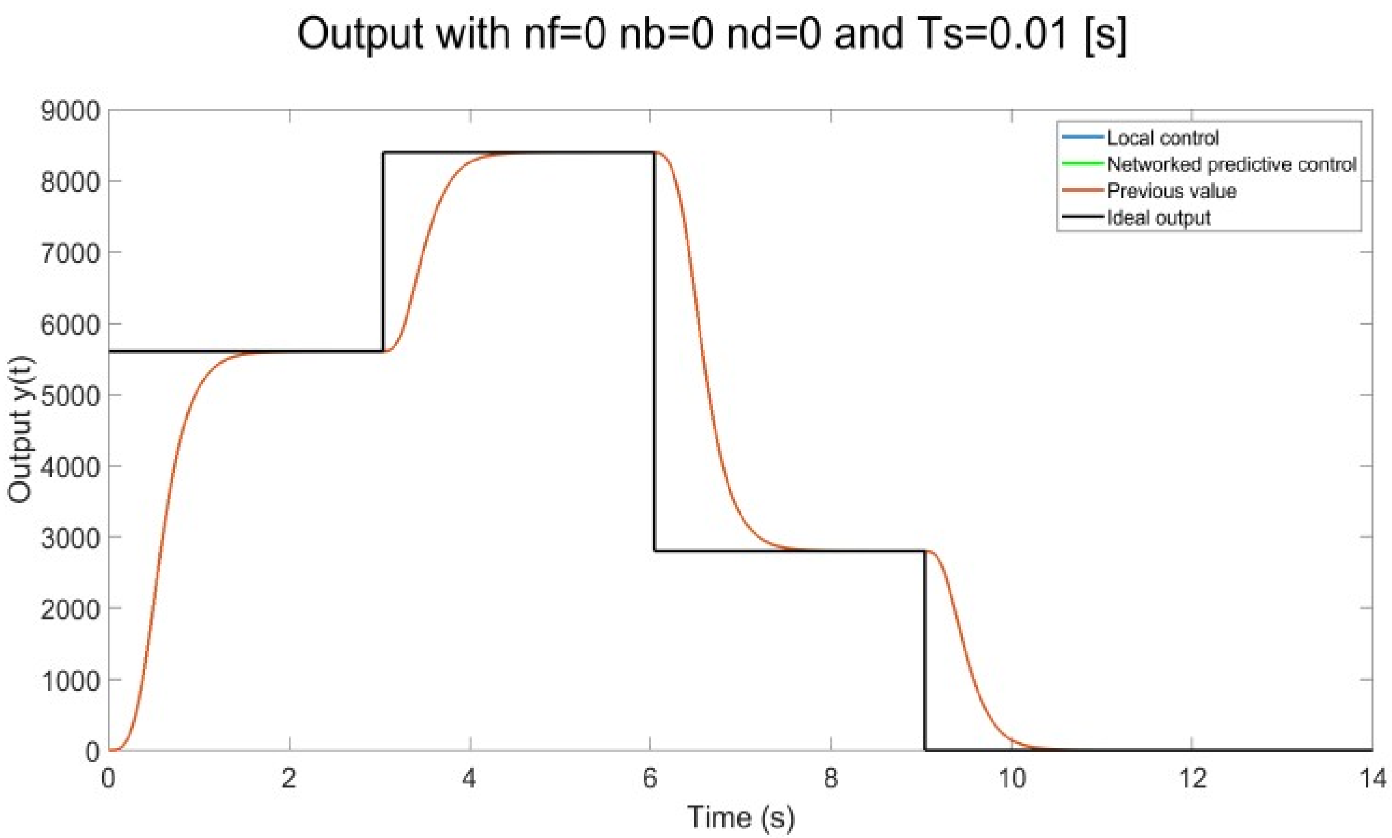
1. The communication delay in the feedback channel(from the sensor to the controller) is bounded by n_b (that is, how many number of samples the data is late)

2. The communication delay in the forward channel(from the controller to the actuator) is bounded by n_f (that is, how many number of samples the data is late)

3. The number of consecutive data drop in feedback and forward channel is bounded by n_d

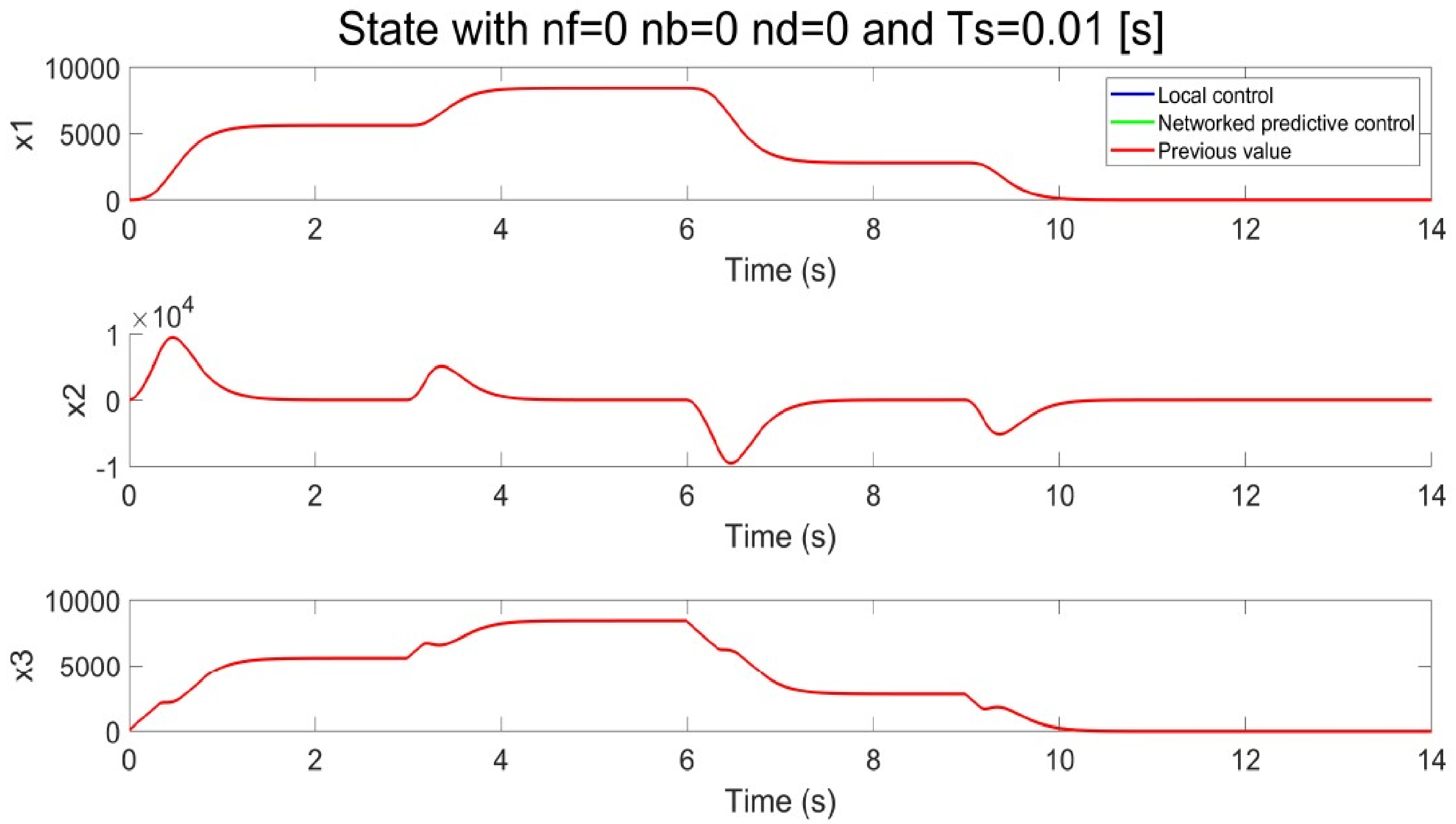
Networked predictive control

Example of application in ideal case without data loss



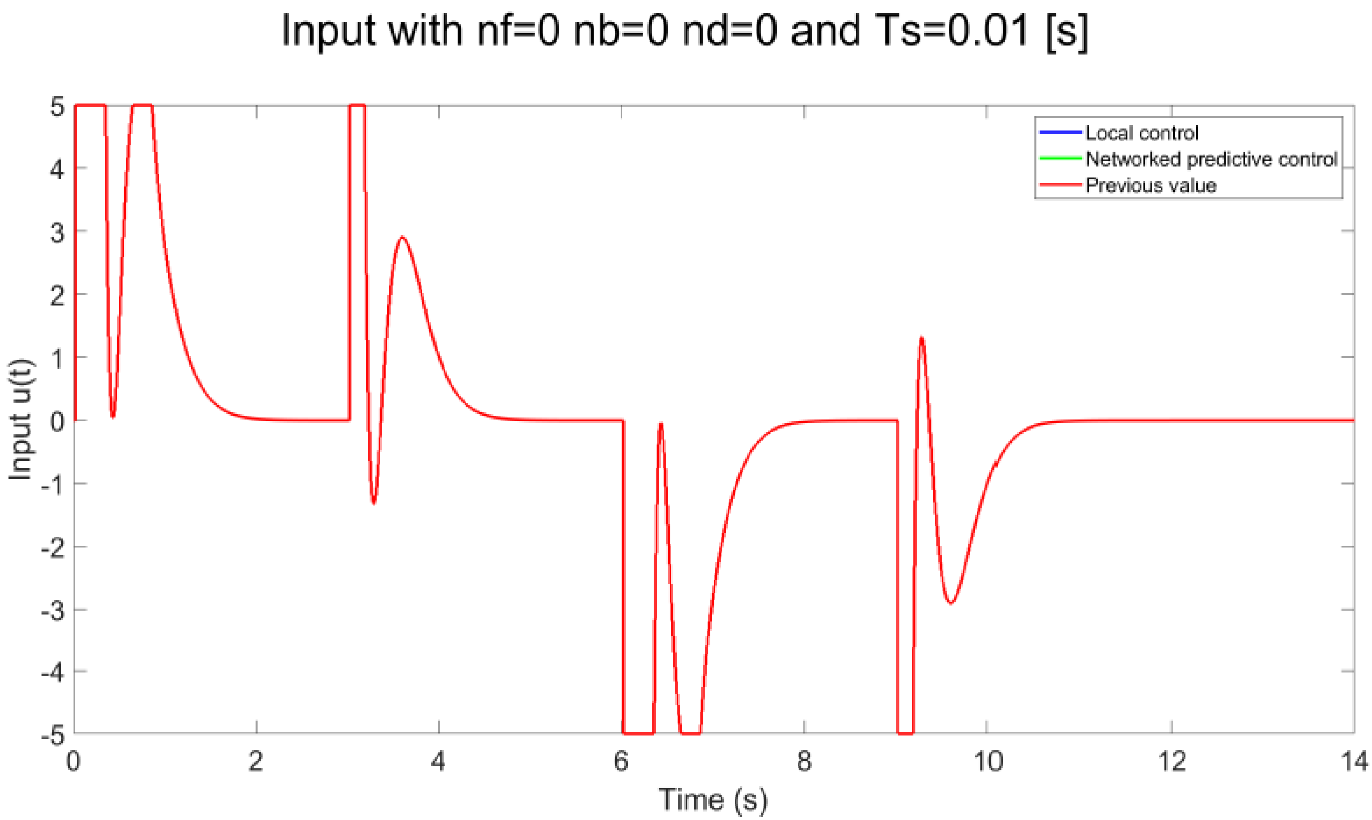
Networked predictive control

Example of application in ideal case without data loss



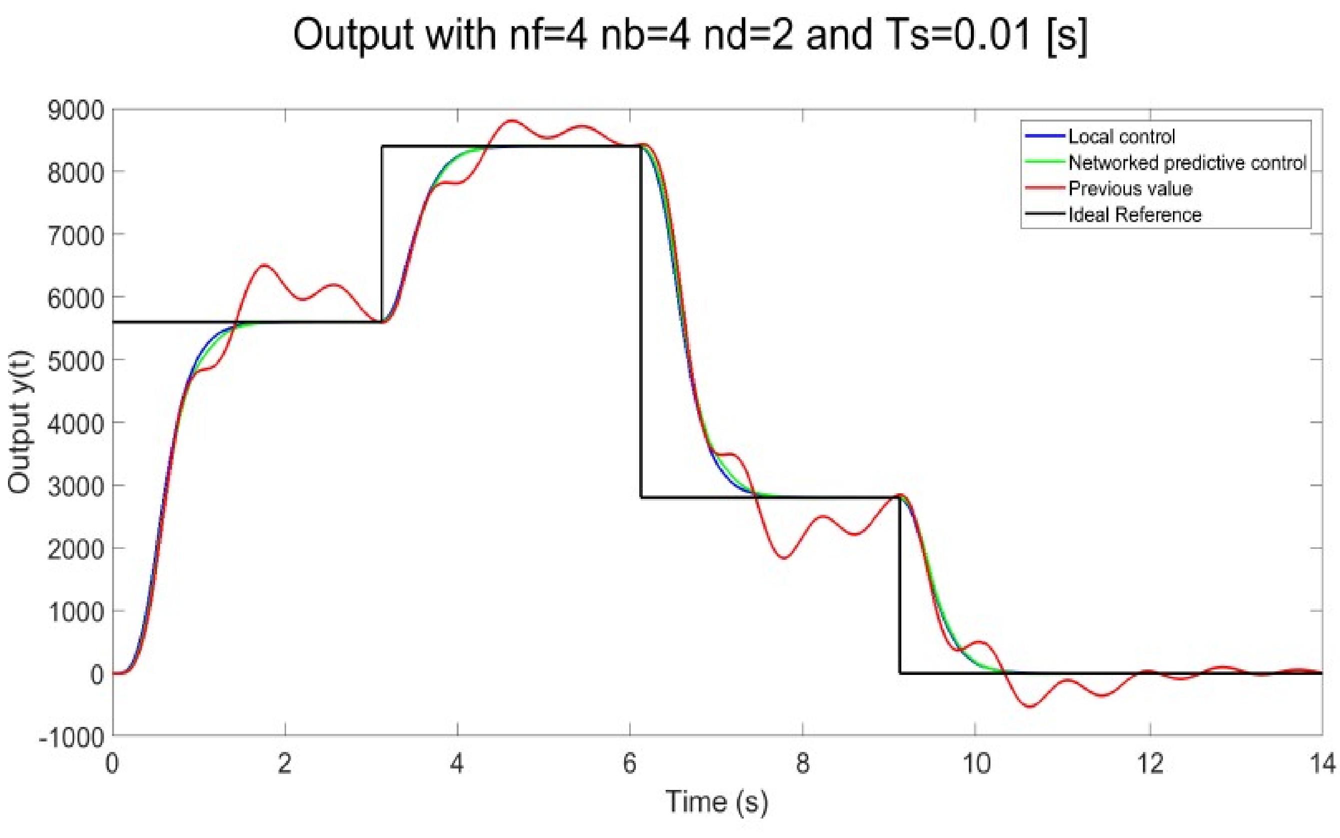
Networked predictive control

Example of application in ideal case without data loss



Networked predictive control

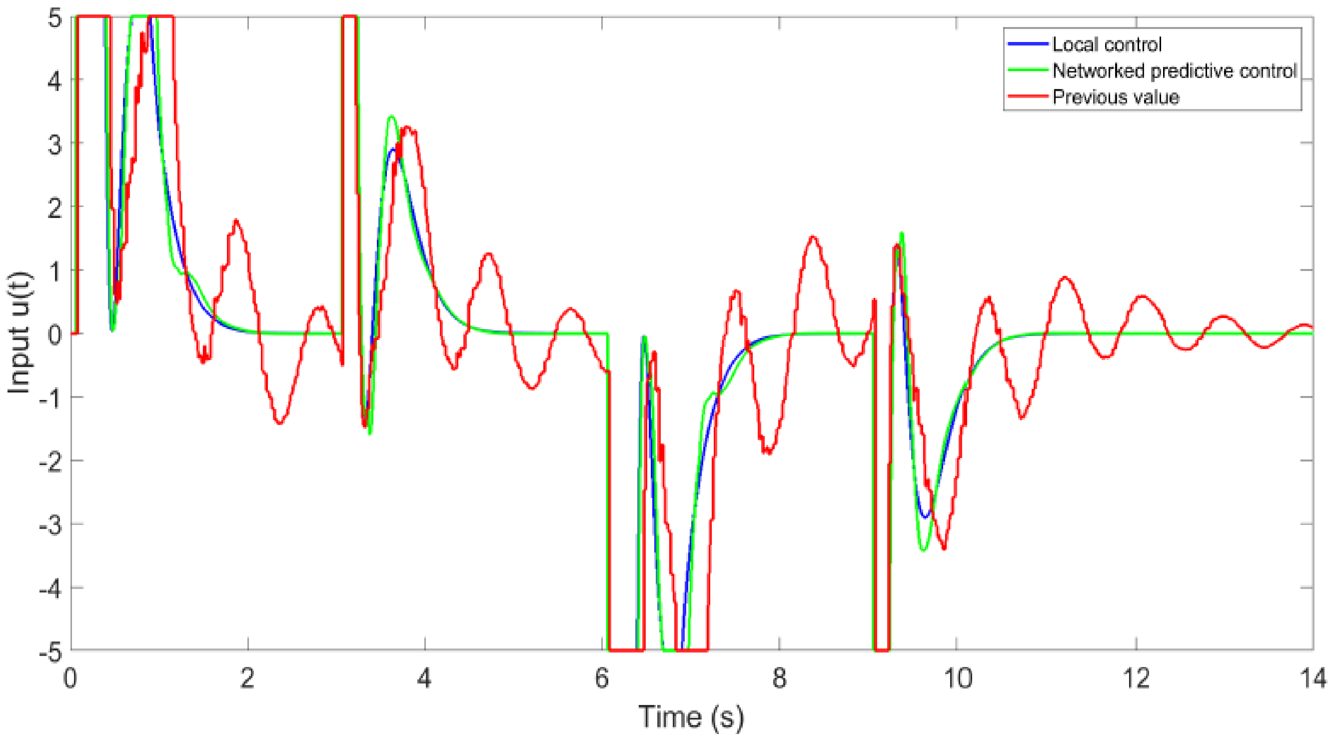
Example of application in ideal case without data loss



Networked predictive control

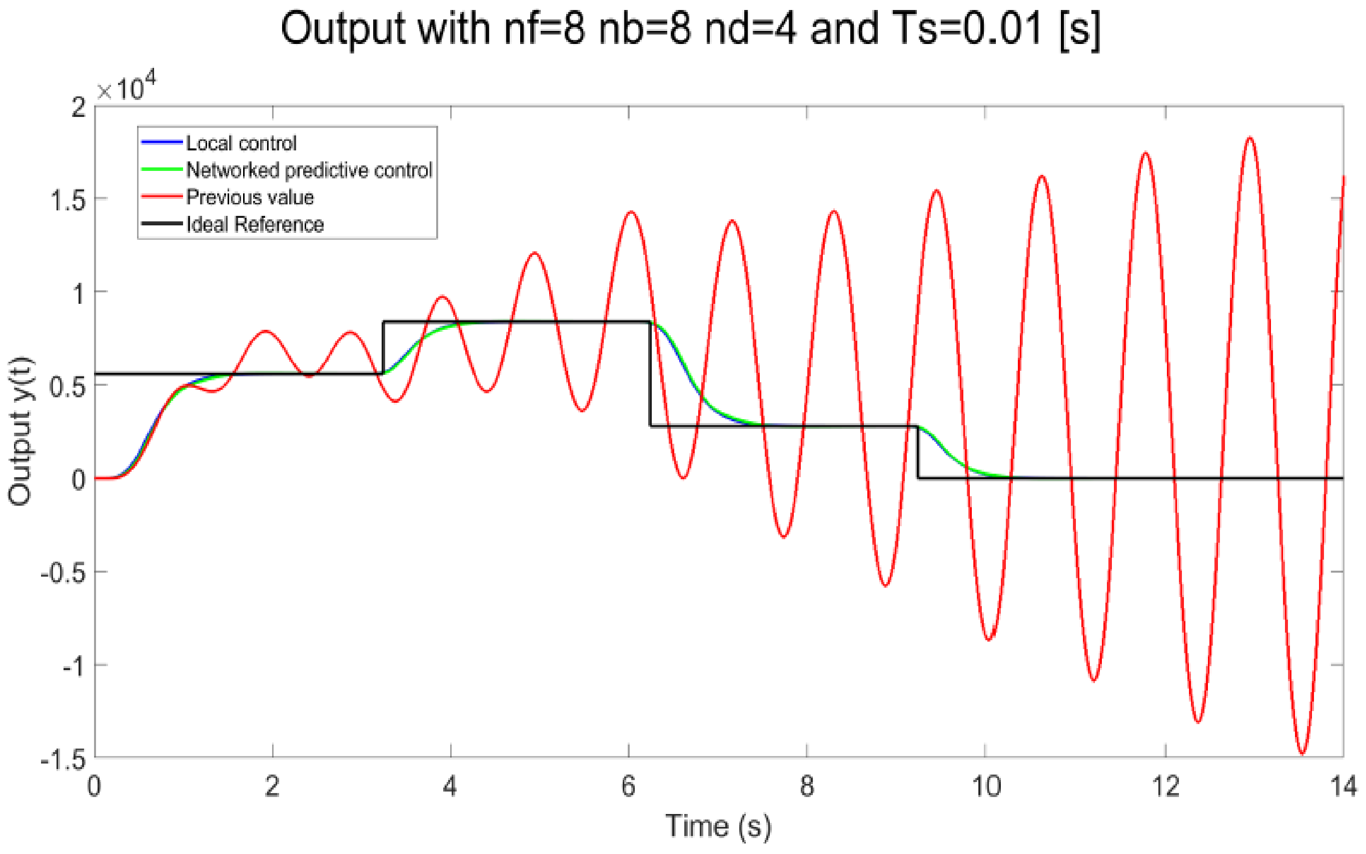
Example of application in ideal case without data loss

Input with nf=4 nb=4 nd=2 and Ts=0.01 [s]



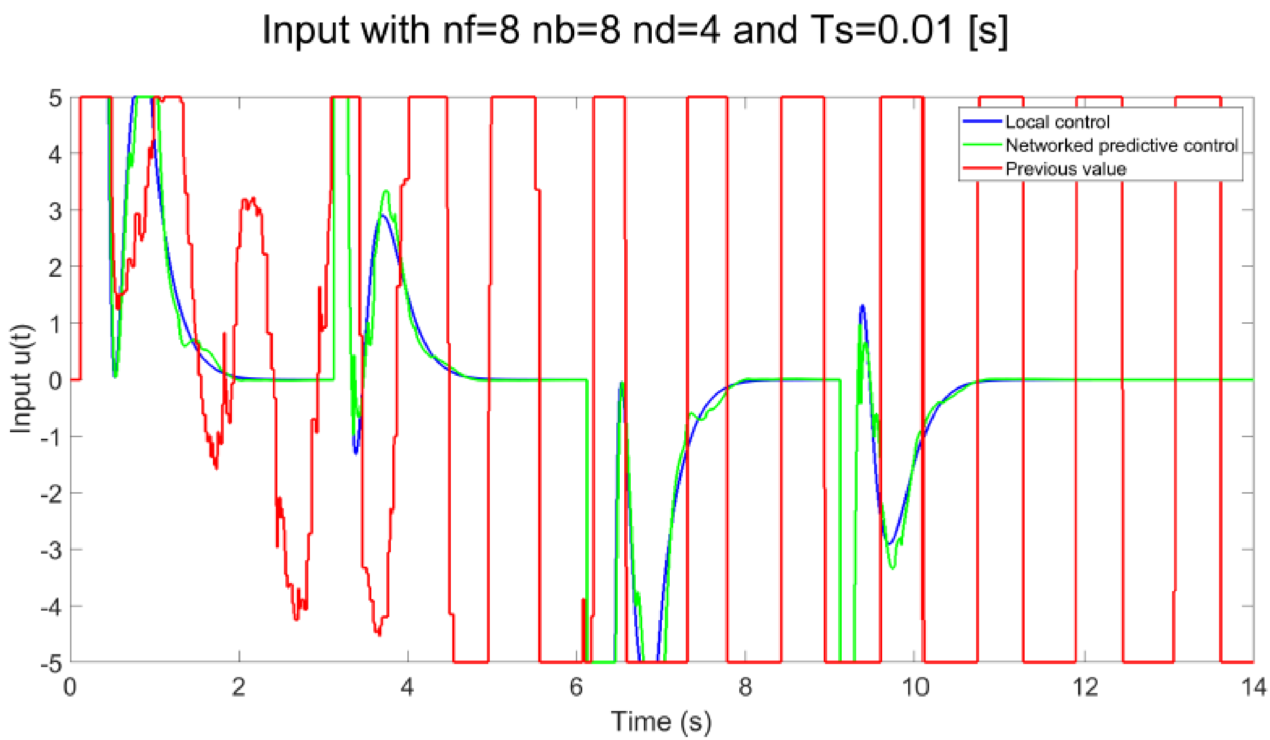
Networked predictive control

Example of application in ideal case without data loss



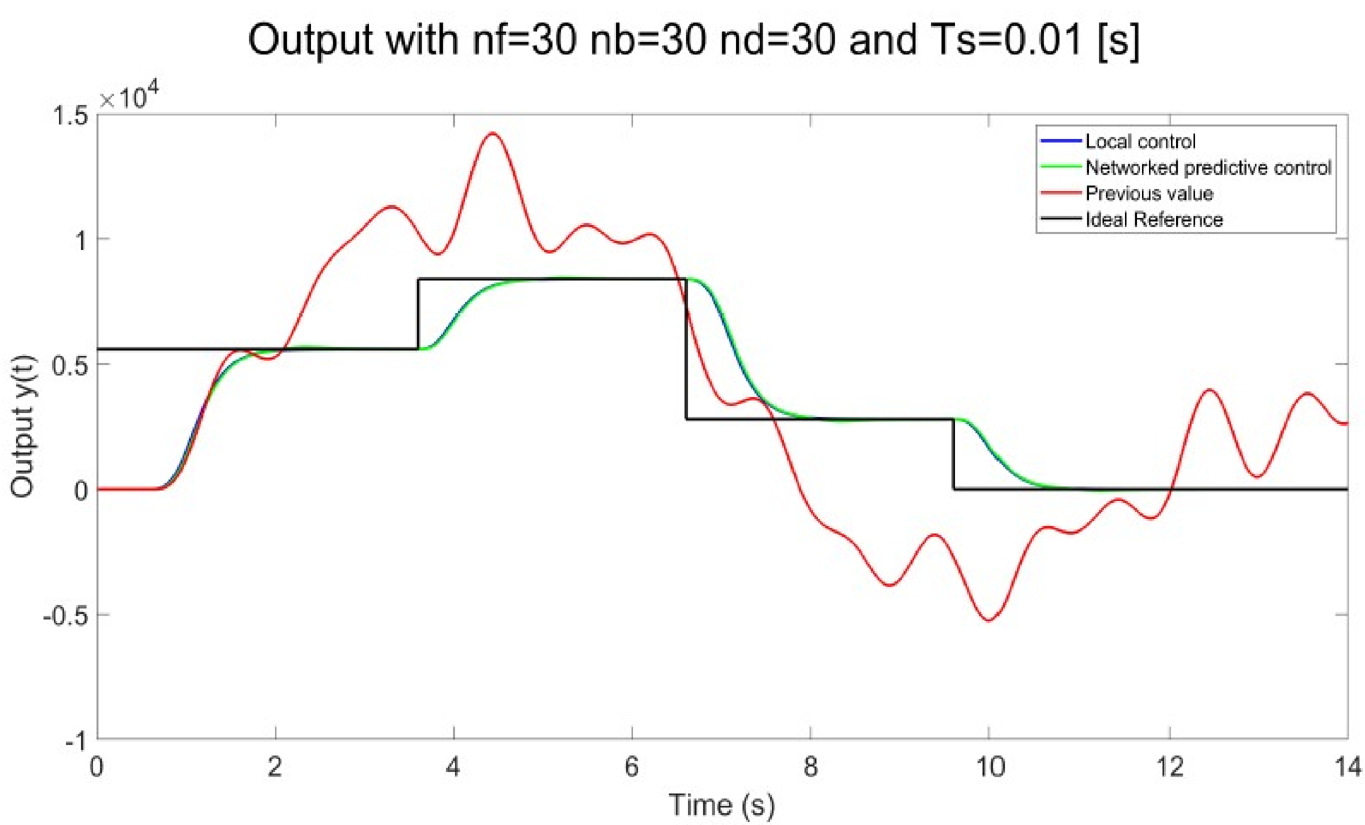
Networked predictive control

Example of application in ideal case without data loss



Networked predictive control

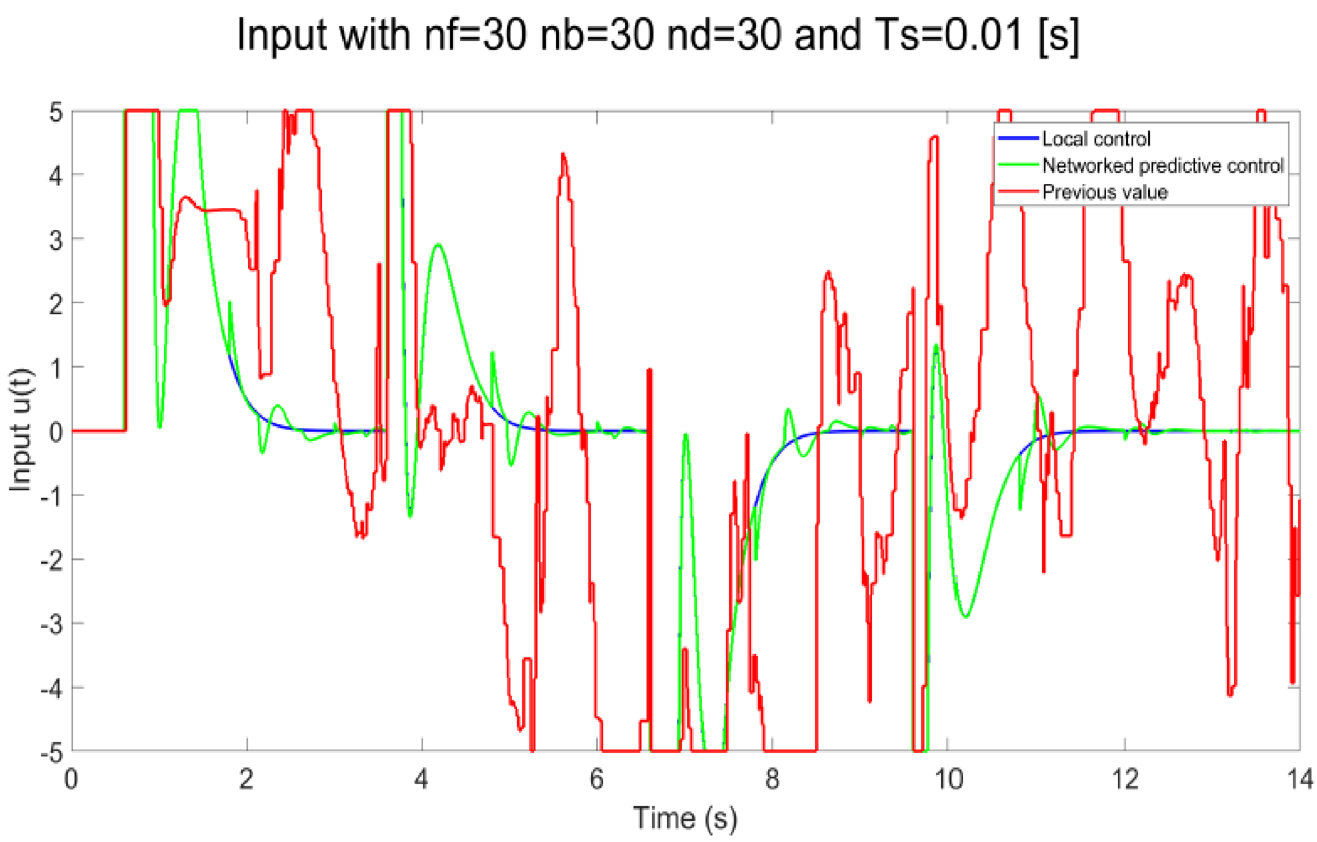
Example of application in ideal case without data loss



Networked control

Networked predictive control

Example of application in ideal case without data loss



Multi-agent systems

Agent

A system which is able to make an activity taking into account its own dynamics and evolution rules as well as its possible connections with other agents

Continuous time dynamical systems

Lumped parameters

$$\dot{\mathbf{x}} = \mathbf{f}\left(\mathbf{x}, \mathbf{u}(t), t\right)$$
$$\mathbf{y} = \mathbf{h}\left(\mathbf{x}, \mathbf{u}(t), t\right)$$

$\mathbf{x} \in R^n \quad \mathbf{y} \in R^p \quad \mathbf{u} \in R^q$

Distributed parameters

$$\frac{\partial y}{\partial t} = \alpha(\vartheta)\frac{\partial^2 y}{\partial x^2}; \qquad y(0,t) = u_1(t); \qquad y(L,t) = u_2(t)$$

Multi-agent systems

Agent

A system which is able to make an activity taking into account its own dynamics and evolution rules as well as its possible connections with other agents

Discrete time dynamical systems

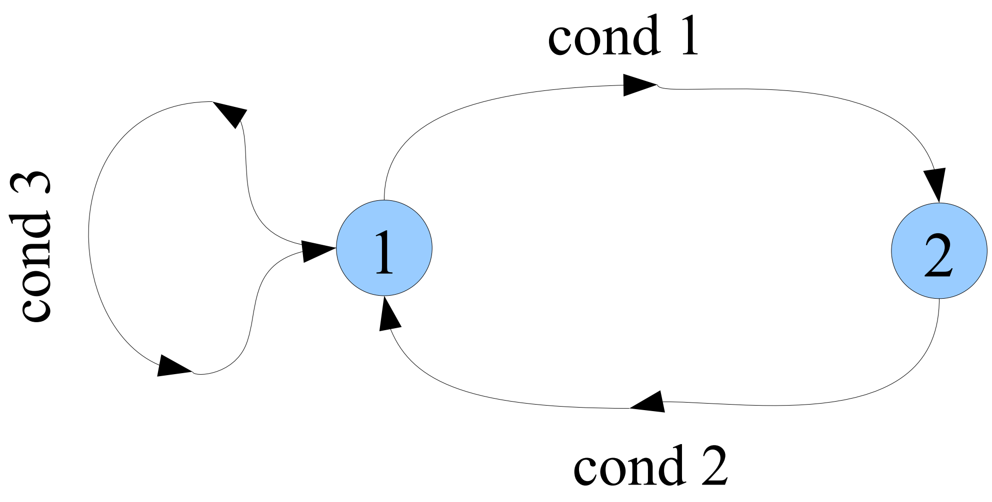
$$\boldsymbol{x}_{k+1} = \boldsymbol{f}\left(\boldsymbol{x}_k ; \boldsymbol{u}_k ; k\right) \qquad \boldsymbol{x} \in R^n ; \boldsymbol{u} \in R^q ; \boldsymbol{y} \in R^p$$

$$\boldsymbol{y}_k = \boldsymbol{h}\left(\boldsymbol{x}_k ; \boldsymbol{u}_k ; k\right) \qquad k = 0, 1, 2, \ldots$$

Agent

A system which is able to make an activity taking into account its own dynamics and evolution rules as well as its possible connections with other agents

Discrete event dynamical systems



Multi Agents System

A complex system composed by a number of interconnected agents that exchange information and/or materials

$\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{A})$

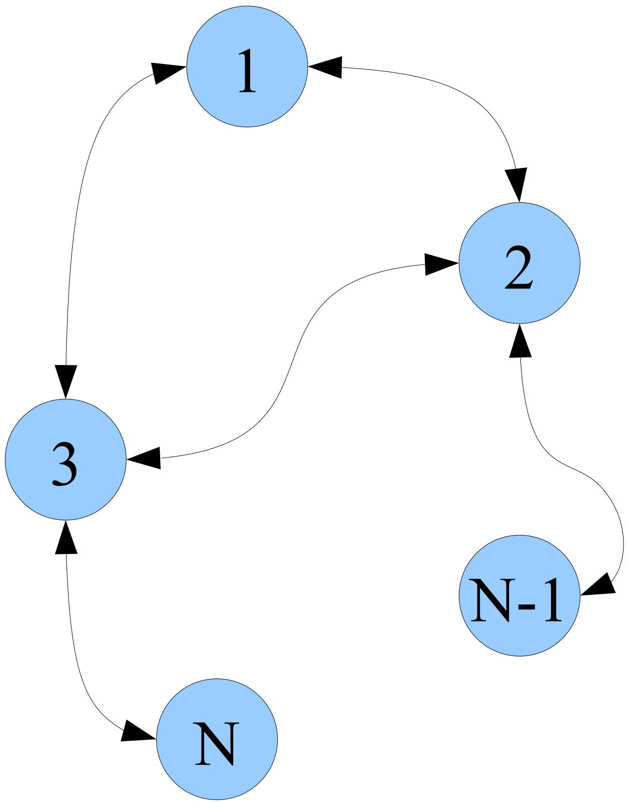
$\mathcal{V} = \{1, 2, \dots, N\}$ the vertex set

$\mathcal{E} \subseteq \{\mathcal{V} \times \mathcal{V}\}$ the edge set

$\mathcal{A} \in \mathbb{R}_{\geq 0}^{N \times N}$ the adjacency matrix

$\mathcal{L} = [\ell_{ij}] \in \mathbb{R}^{N \times N}$ the Laplacian matrix

$\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$



Multi Agents System

A complex system composed by a number of interconnected agents that exchange information and/or materials

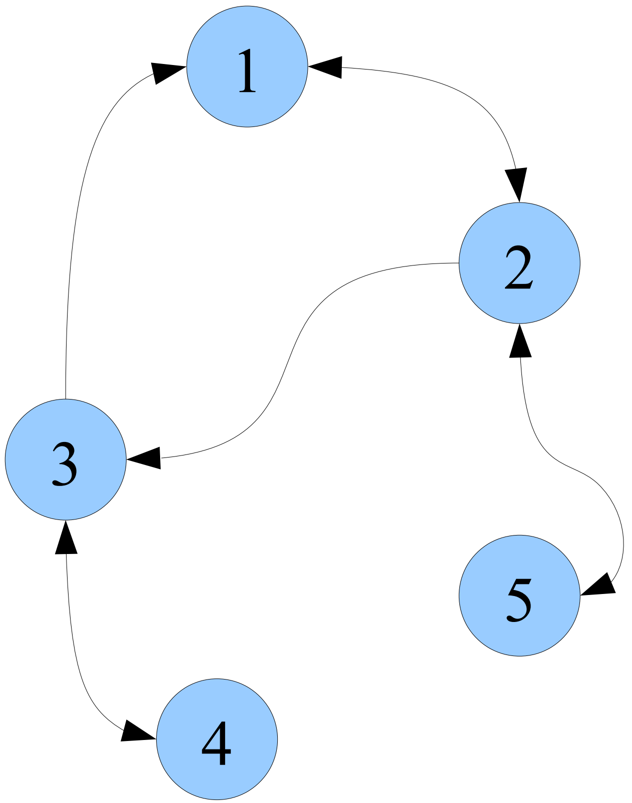
$$\mathcal{G}(\mathcal{V},\mathcal{E},\mathcal{A})$$

the vertex set

$$\mathcal{V}=\{1,2,3,4,5\}$$

the edge set

$$\varepsilon=\left\{\begin{matrix}(1,2),(2,1),(2,3),(2,5),\\(3,1),(3,4),(4,3),(5,2)\end{matrix}\right\}$$



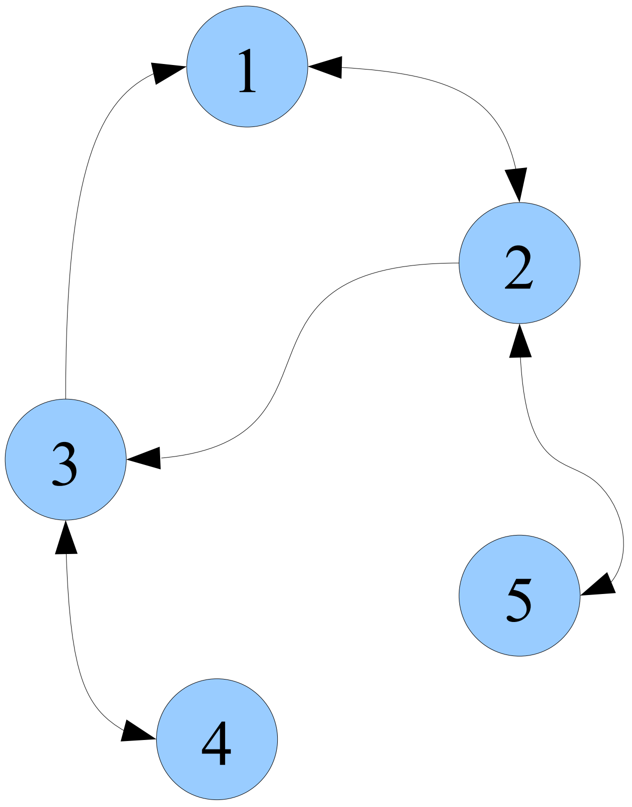
Multi Agents System

A complex system composed by a number of interconnected agents that exchange information and/or materials

$$\mathcal{G}(\mathcal{V},\mathcal{E},\mathcal{A})$$

the adjacency matrix

$$A=\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$



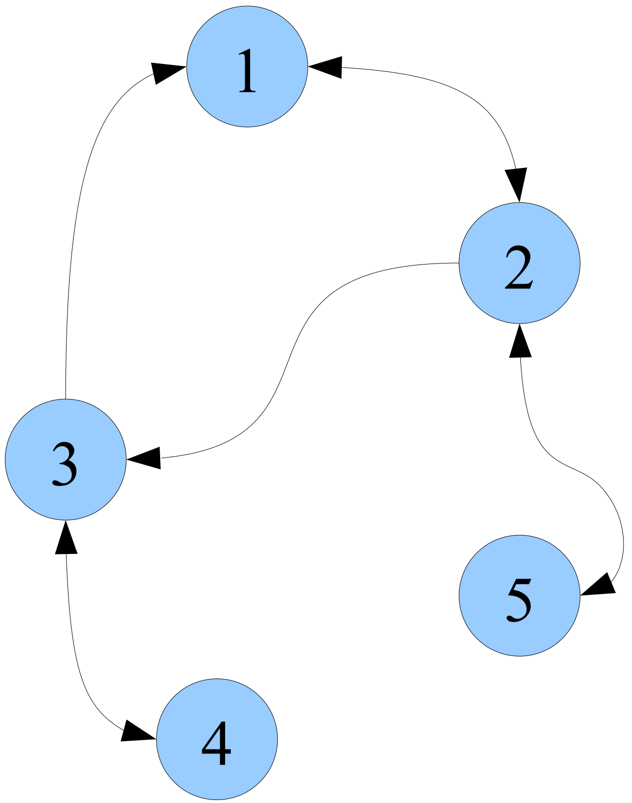
Multi Agents System

A complex system composed by a number of interconnected agents that exchange information and/or materials

$$\mathcal{G}(\mathcal{V},\mathcal{E},\mathcal{A})$$

the Degree-out matrix

$$D_{out} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$



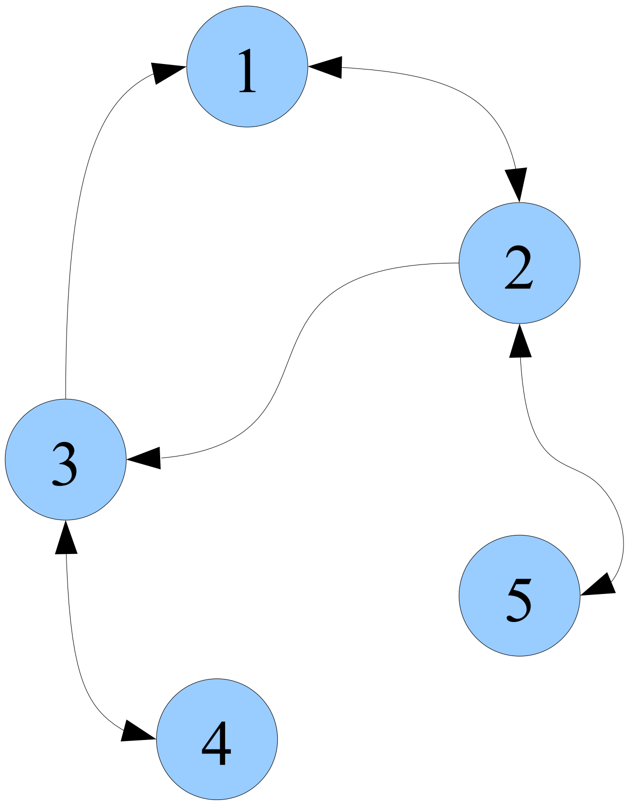
Multi Agents System

A complex system composed by a number of interconnected agents that exchange information and/or materials

$$\mathcal{G}(\mathcal{V},\mathcal{E},\mathcal{A})$$

the Laplacian matrix

$$L = D_{out} - A = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & -1 \\ -1 & 0 & 2 & -1 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 0 & 1 \end{bmatrix}$$



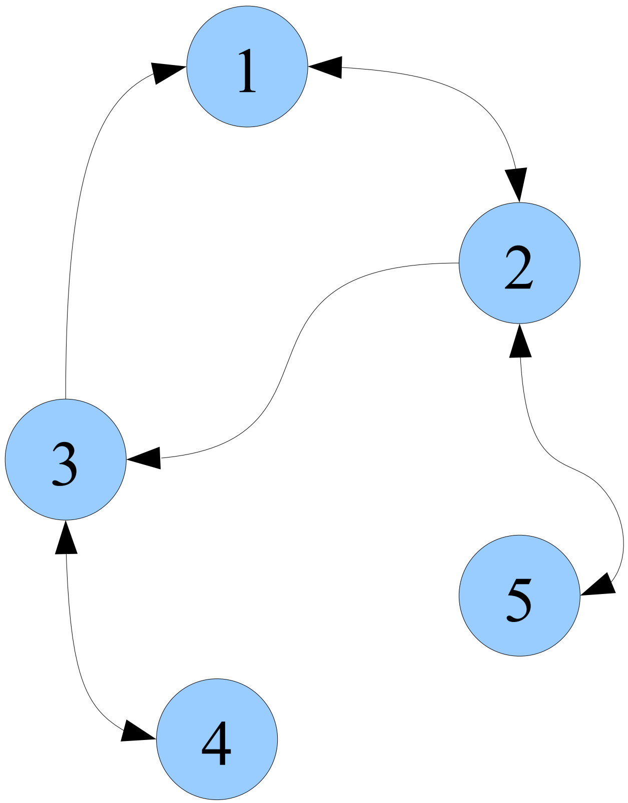
Multi Agents System

A complex system composed by a number of interconnected agents that exchange information and/or materials

$$\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{A})$$

the in-neighbours sets

$$N_{input_1} = \{2, 3\}$$
$$N_{input_2} = \{1, 5\}$$
$$N_{input_3} = \{2, 4\}$$
$$N_{input_4} = \{3\}$$
$$N_{input_5} = \{2\}$$

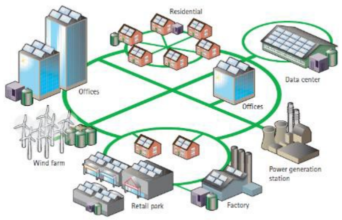


Multi-agent systems

Multi Agents System

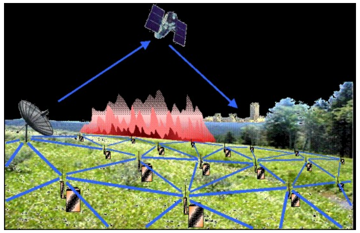
A complex system composed by a number of interconnected agents that exchange information and/or materials

Smart grids



UAV Coordination

Sensing wide areas

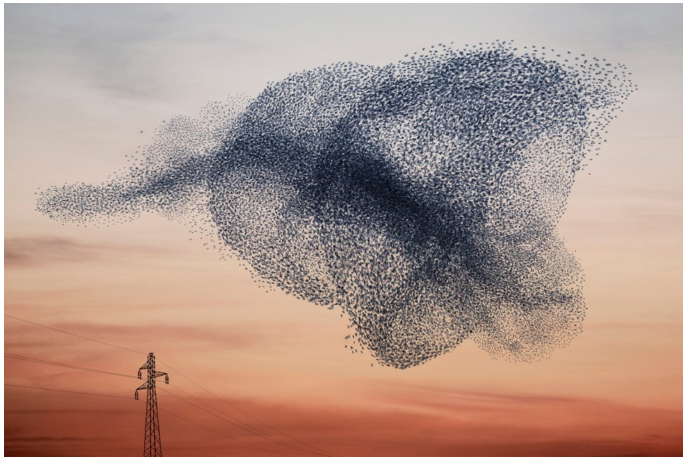


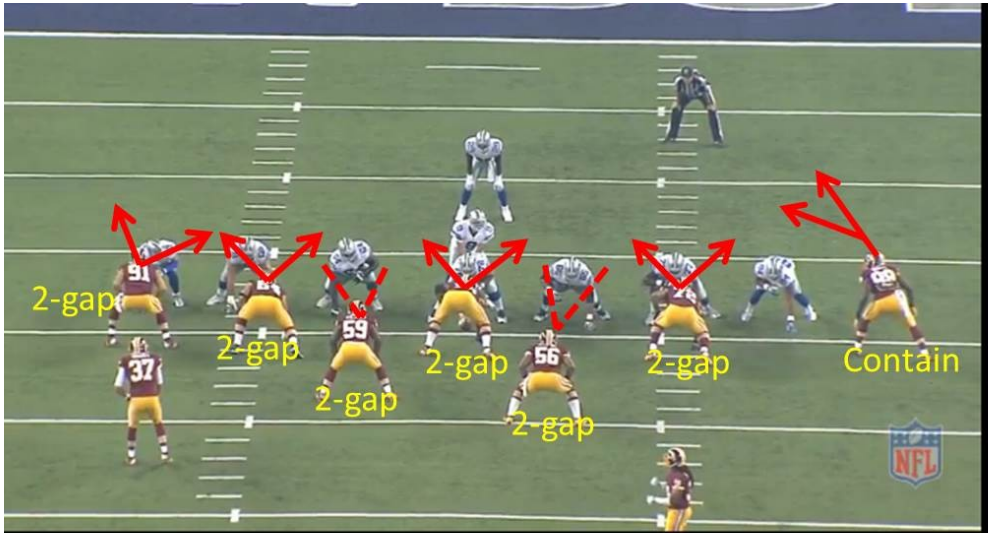
Social networks

Multi Agents System

The behaviour of a Multi agent system is not the simple combination of the behaviour of each system and emerging dynamics can appear

Flocking





Coordinated teams

Multi Agents System

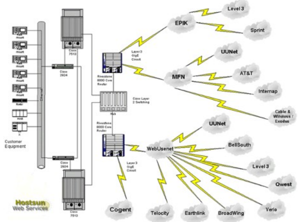
The behaviour of a Multi agent system is not the simple combination of the behaviour of each system and emerging dynamics can appear

Crowd

Not always the behaviour of a multi agent system is predictable by the knoledge of its components

Multi Agents System

The behaviour of a Multi agent system could be heavily dependent on external inputs



Smart grids

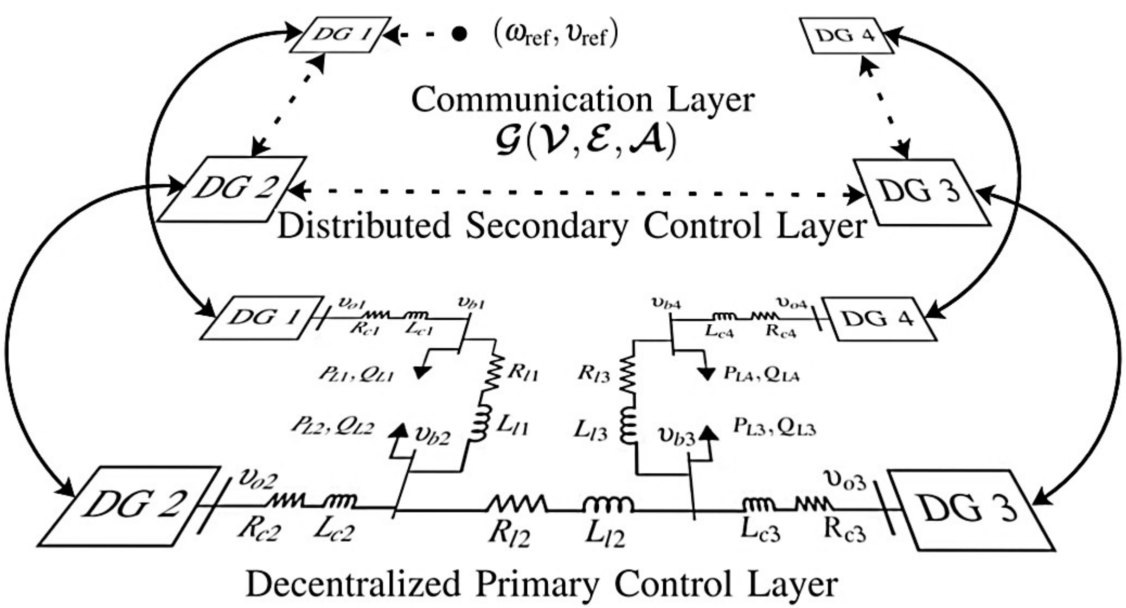
UAV Coordination

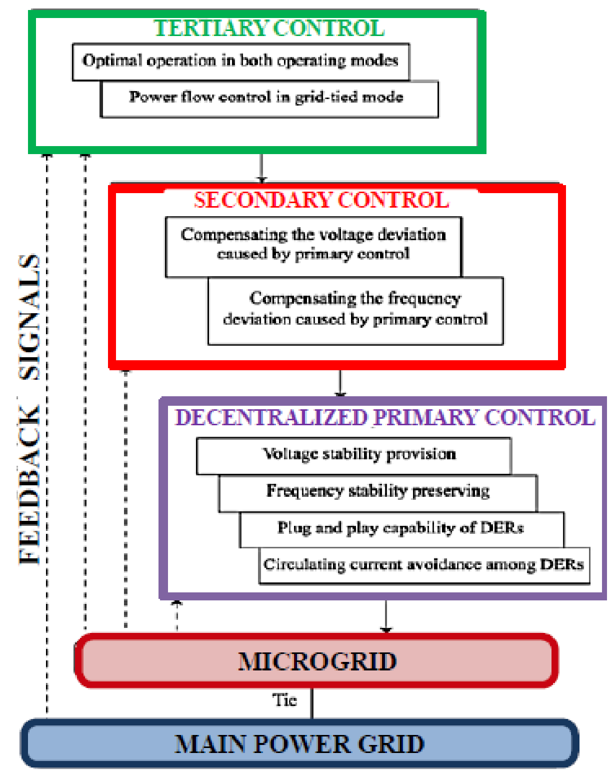
Sensing wide areas

Social networks

Multi Agents System

Multi Agent Systems can appear when implementing jerarchical controls





Multi Agents System

Multi Agent Systems have complex interacting dynamics

$$\dot{x}_i = f_i\left(x_i; x_j; u_i; t\right) \qquad x \in \mathbb{R}^n; u \in \mathbb{R}^q; y \in \mathbb{R}^p$$

$$y_i = h\left(x_i; u_i; t\right) \qquad i = 1,2,,N \qquad x_j \in \mathcal{N}_i$$

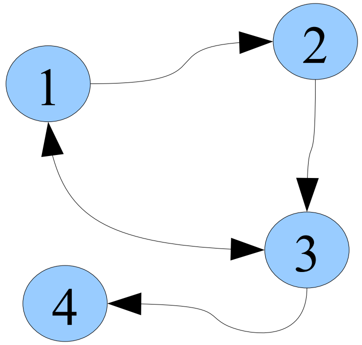
The properties of a Multi Agent System mainly depend on the Laplacian matrix that represents how the agents are connected one each other

$$L = D - A$$

L: laplacian matrix
D: connection degree matrix
A: adiancency matrix

Multi Agents System

The properties of a Multi Agent System mainly depend on the Laplacian matrix that represents how the agents are connected one each other



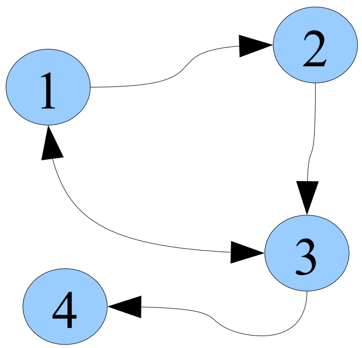
$$D_{out} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$L = D_{out} - A = \begin{bmatrix} 2 & -1 & -1 & 0 \\ 0 & 1 & -1 & 0 \\ -1 & 0 & 2 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Multi Agents System

The control of a Multi Agent System is mainly based on the knowledge of the state of the agent itself and of its neighbors



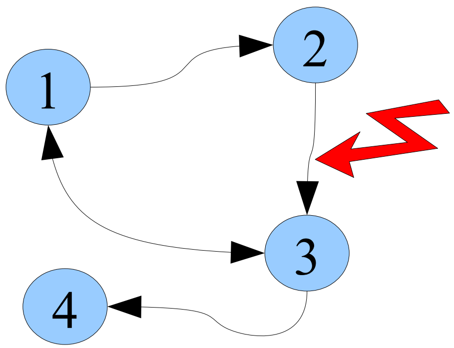
$$L = D_{out} - A = \begin{bmatrix} 2 & -1 & -1 & 0 \\ 0 & 1 & -1 & 0 \\ -1 & 0 & 2 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{u}_i = \boldsymbol{g}_i\left(\boldsymbol{x}_i; \boldsymbol{x}_j; t\right) \qquad \boldsymbol{x}_i \in \mathbb{R}^n; \boldsymbol{u}_i \in \mathbb{R}^q; \boldsymbol{y}_i \in \mathbb{R}^p$$

$$\boldsymbol{x}_j \in \mathcal{N}_i$$

Multi Agents System

The control of a Multi Agent System should be designed in order to be able to not suffer from external undesired inputs



Robustess: the ability of the system to react to perturbations, internal failures, and environmental events by **compensating** the disturbance and/or **limiting their effect on the state and the output**

Resilience: the ability of the system to react to perturbations, internal failures, and environmental events by **absorbing** the disturbance and/or **reorganizing to maintain its functions**

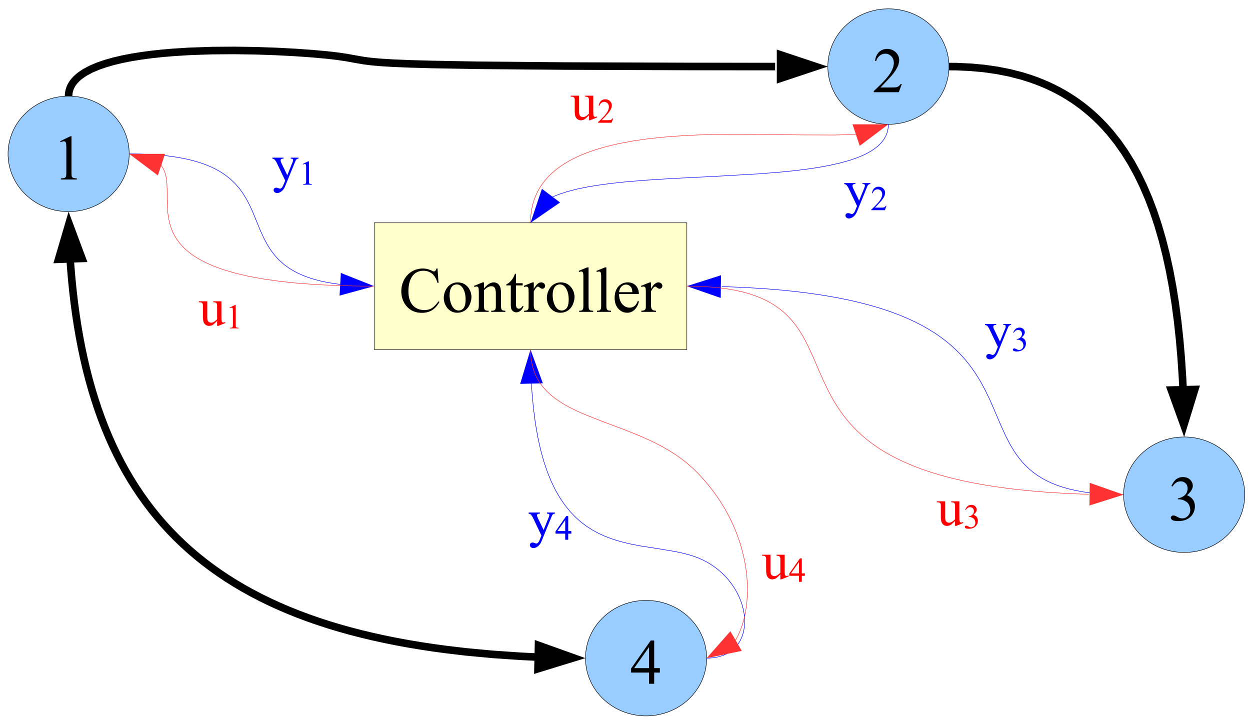
Distributed control

When considering the control problem for multi-agent systems we have to design the structure of the control system:

- Centralized control
- De-centralized control
- Distributed control

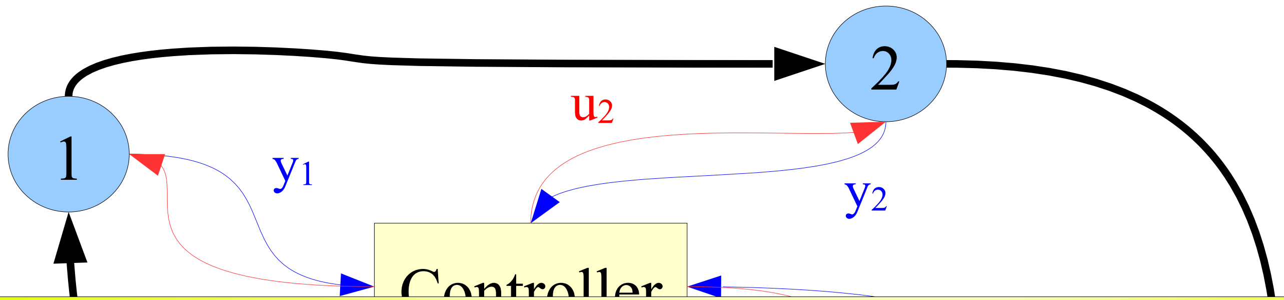
Centralised control

All measurements are collected by a central agent that defines the control law for all of the connected agents taking into account the overall system condition



Centralised control

All measurements are collected by a central agent that defines the control law for all of the connected agents taking into account the overall system condition



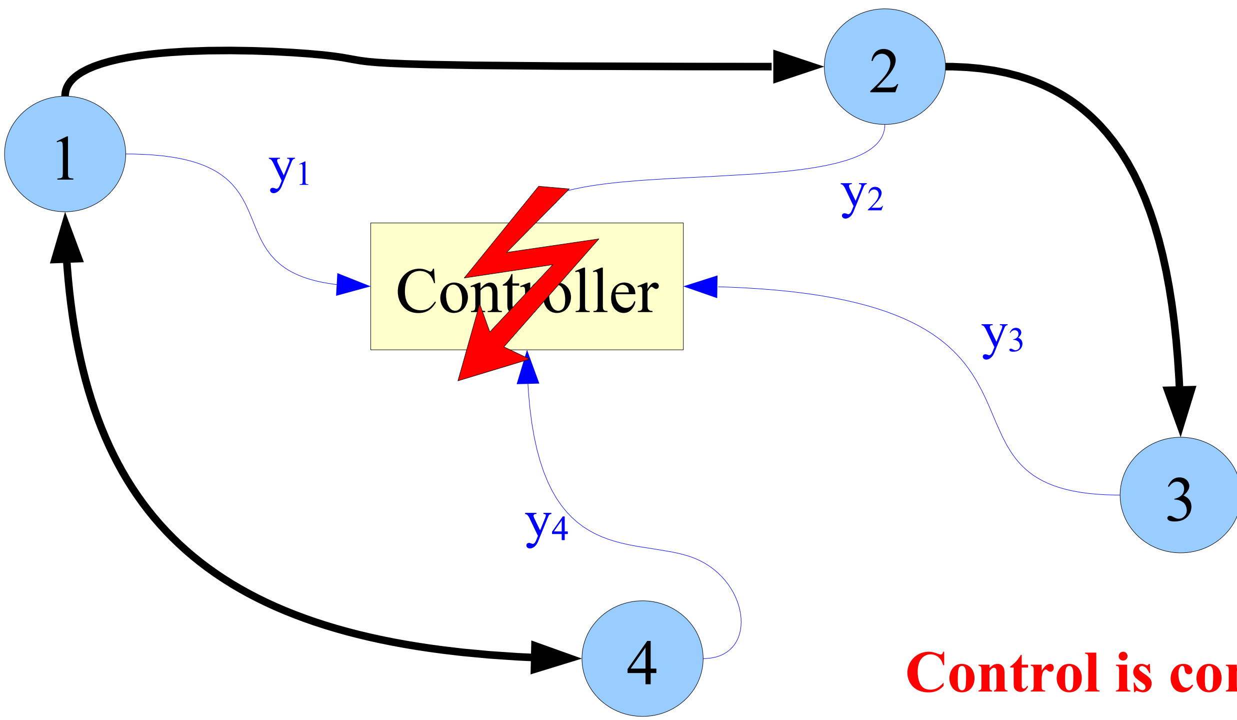
Optimization can be obtained since the controller can consider all the information that are available

$$J(\hat{\boldsymbol{u}}) = \min_{\boldsymbol{u} \in U} \int_{t=t_0}^{t} L(\boldsymbol{y}; \boldsymbol{u})\, dt$$

Distributed control

Centralised control

All measurements are collected by a central agent that defines the control law for all of the connected agents taking into account the overall system condition

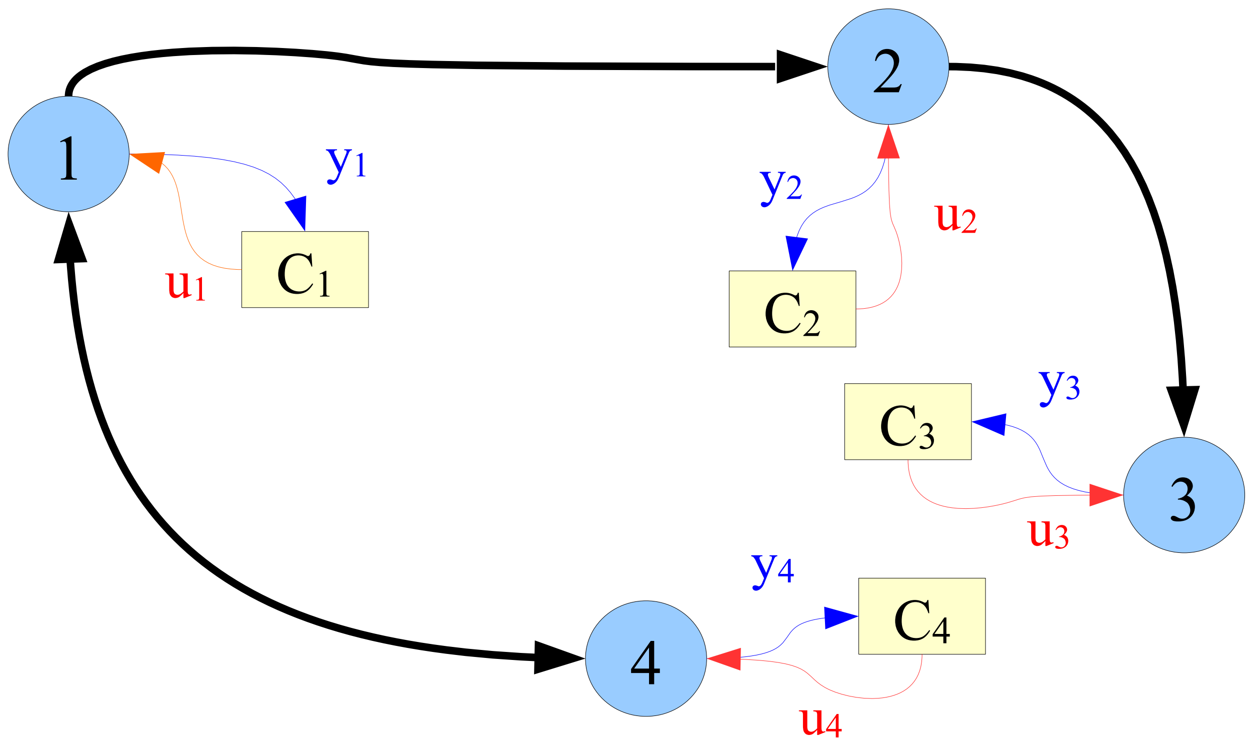


Control is completely lost in the case of controller fault

Distributed control

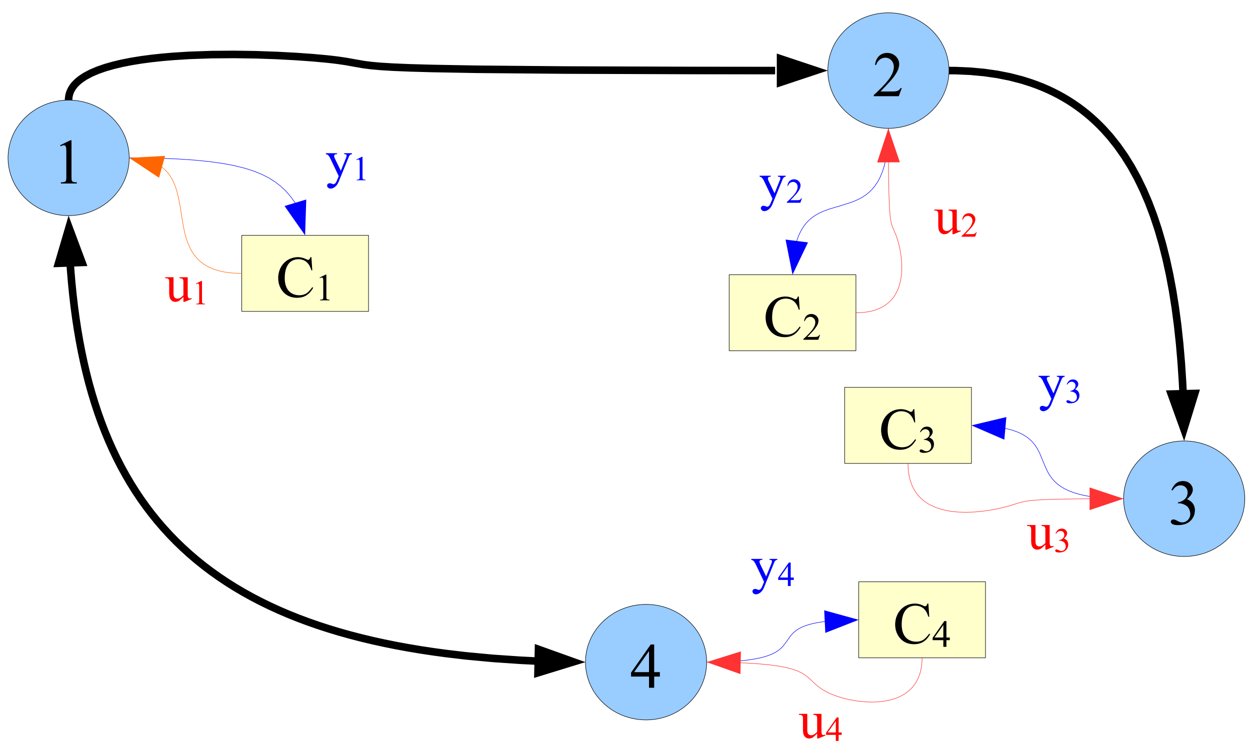
De-Centralised control

Each agent has its own control based on its own mesurement; no information is shared



De-Centralised control

Each agent has its own control based on its own mesurement; no information is shared

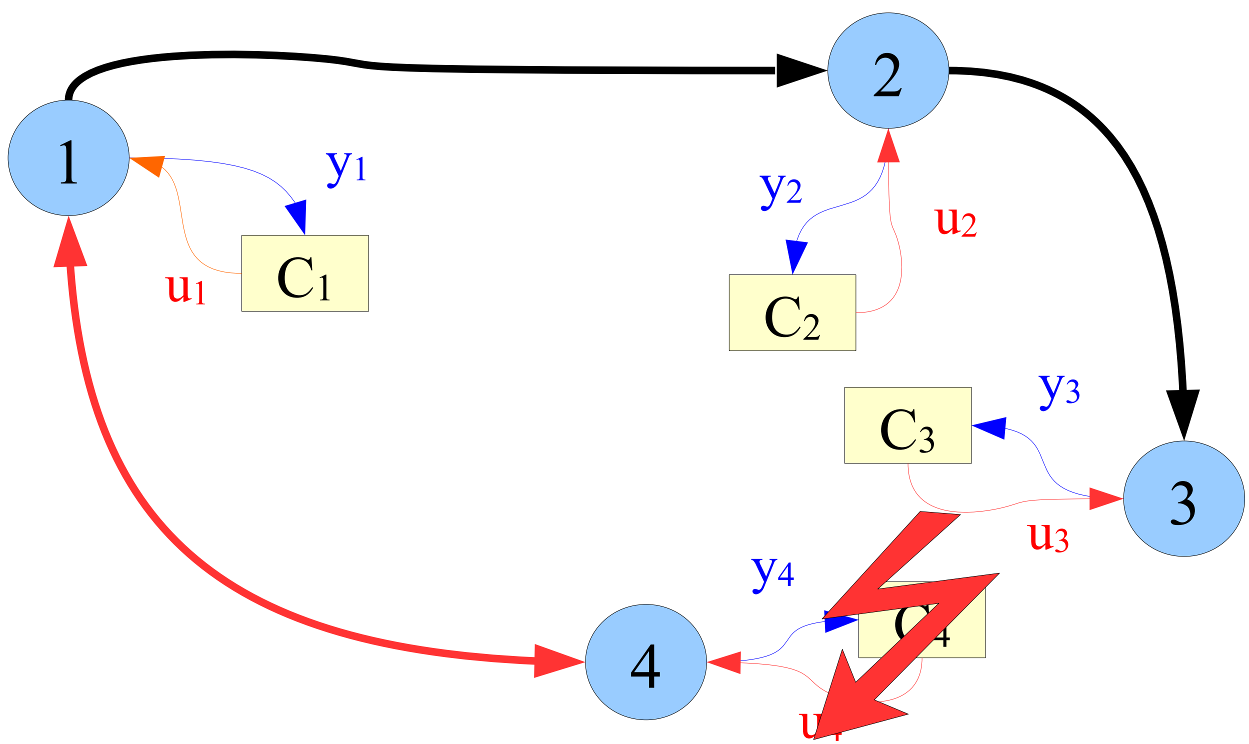


Physical interactions among agents are considered as disturbances to compensate for

Distributed control

De-Centralised control

Each agent has its own control based on its own mesurement; no information is shared

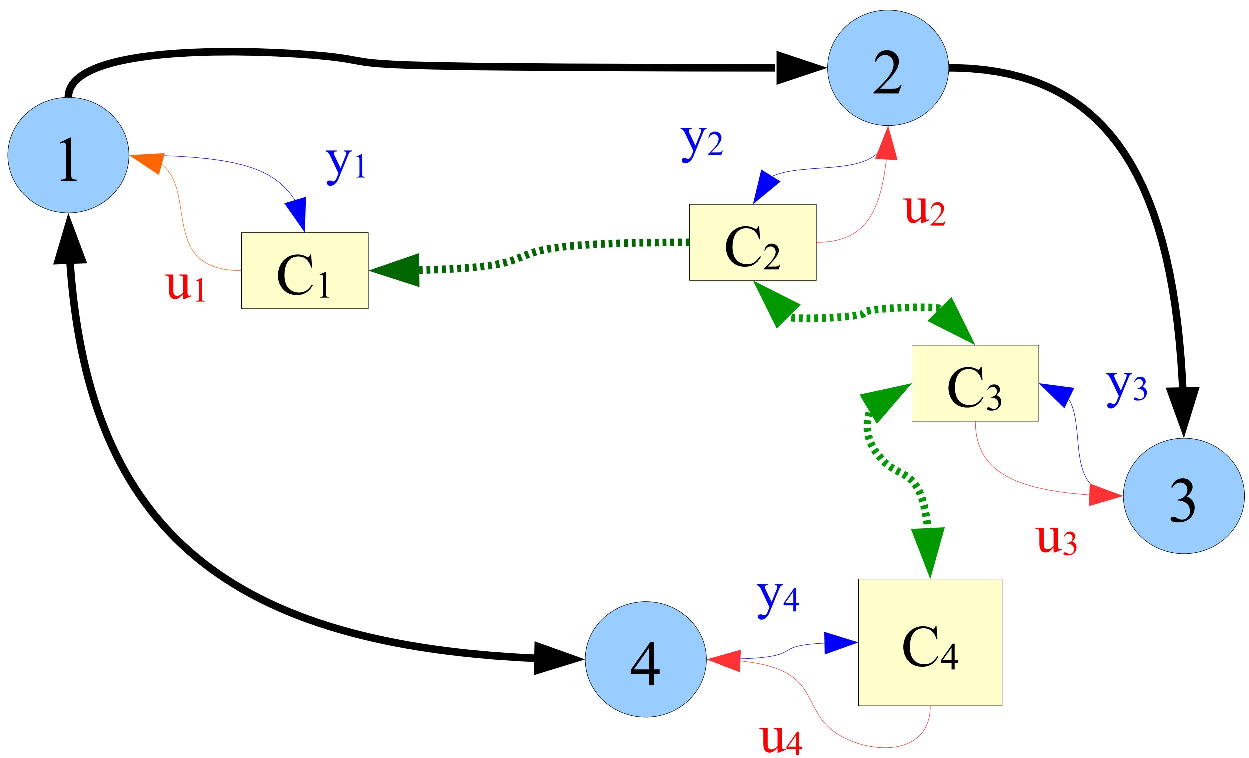


The fault of one local controller does not fully propagate to the all agents and its effect can be mitigated by neighbour controllers

Distributed control

Distributed control

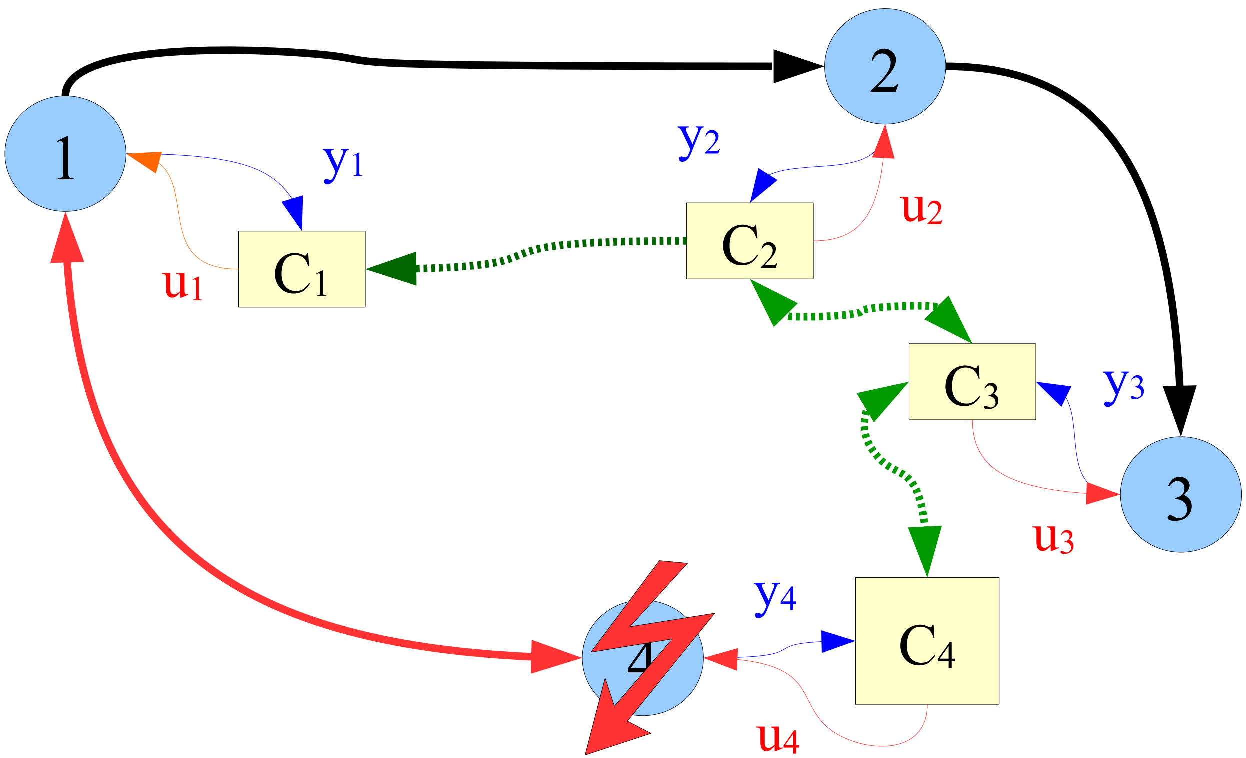
Each agent has its own control based on its own mesurement and some data from its neighbors; same information are shared



The communication graph can differs from the graph representing the physical connections between agents

Distributed control

Each agent has its own control based on its own mesurement and some data from its neighbors; same information are shared

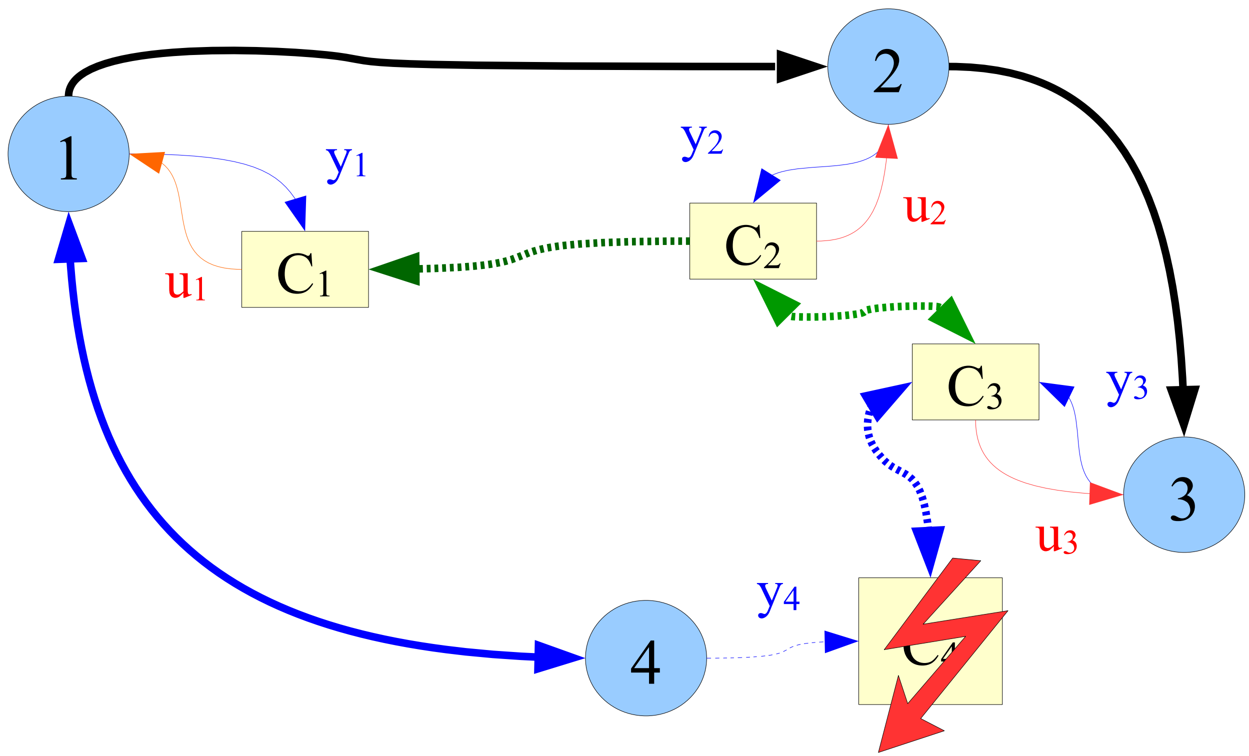


The fault in one agent can be compensated by the coordinated actions of the agent and its neighbors

Distributed control

Distributed control

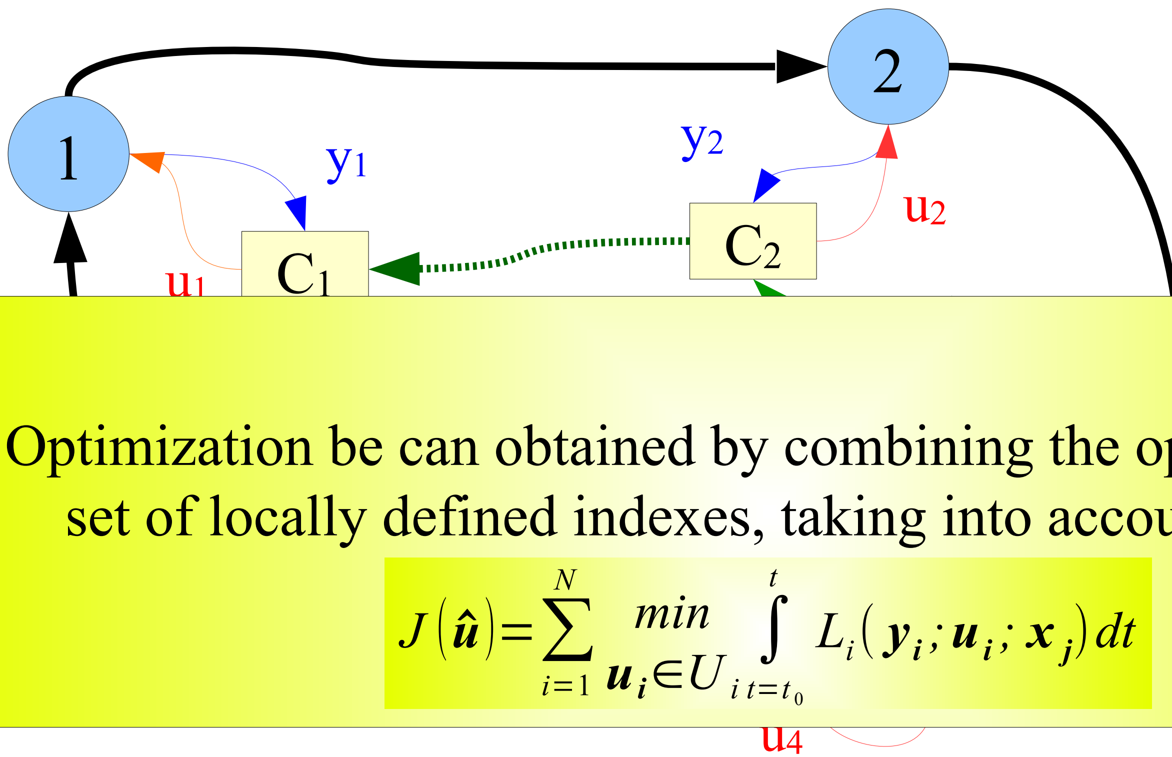
Each agent has its own control based on its own mesurement and some data from its neighbors; same information are shared



The fault in one controller can be mitigated by the coordinated actions of the neighbor's controllers

Distributed control

Each agent has its own control based on its own mesurement and some data from its neighbors; same information are shared



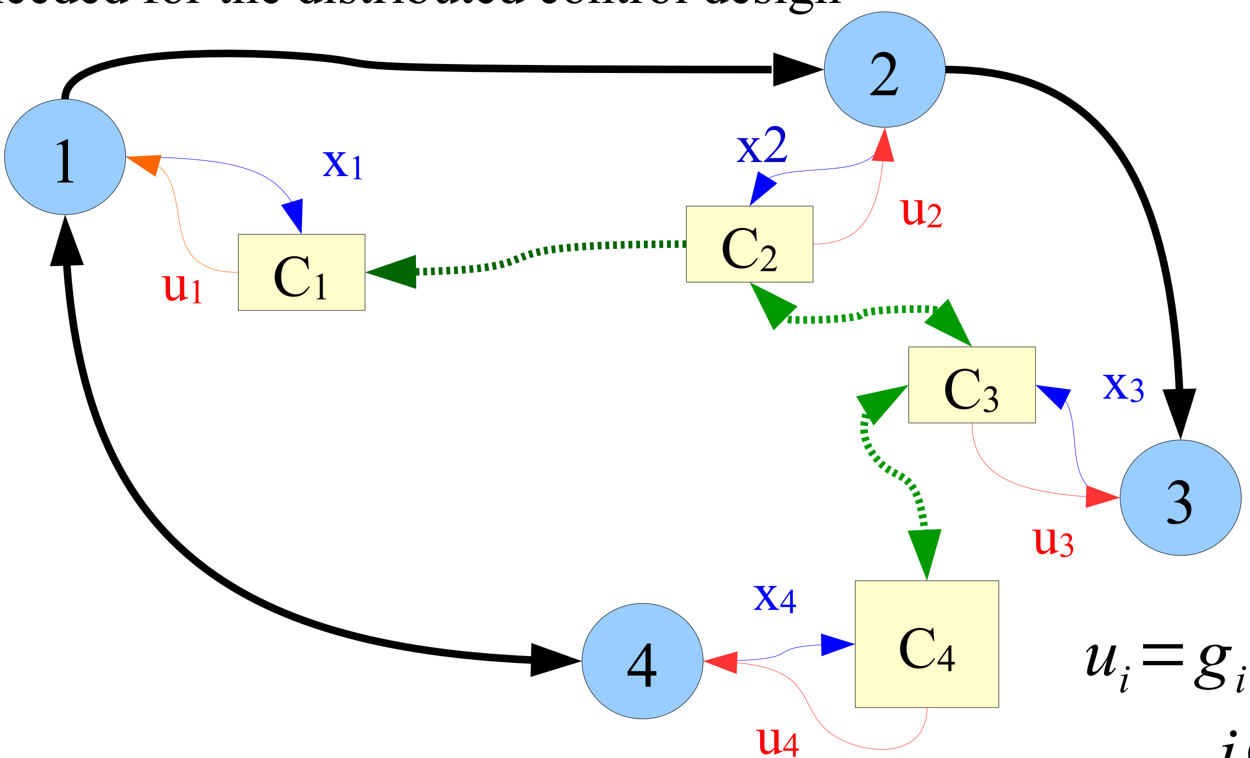
The communication graph can

Optimization be can obtained by combining the optimization of a set of locally defined indexes, taking into account neighbors

$$J(\hat{\boldsymbol{u}}) = \sum_{i=1}^{N} \min_{\boldsymbol{u}_i \in U_i} \int_{t=t_0}^{t} L_i(\boldsymbol{y}_i; \boldsymbol{u}_i; \boldsymbol{x}_j)\, dt$$

Distributed control via *Consensus*

Consensus is achieved when the agents "agree" in the sense that their states tend to the same value or profile. The states of each agent and its neighbours are needed for the distributed control design



$$u_i = g_i\left(x_i ; x_j\right);$$
$$j \in \mathcal{N}_i$$

Distributed control via *Consensus*

Consensus is achieved when the agents "agree" in the sense that their states tend to the same value or profile.

$$\left|\boldsymbol{x}_i - \boldsymbol{x}_j\right| \rightarrow 0\,; \qquad \forall i,\, j\left(i \neq i\right)$$

Consensus can be achieved both because of the physical connections and by a proper control design

The states of each agent and its neighbours are needed for the distributed control design

$$u_i = g_i\left(x_i;\, x_j\right); \quad j \in \mathcal{N}_i$$

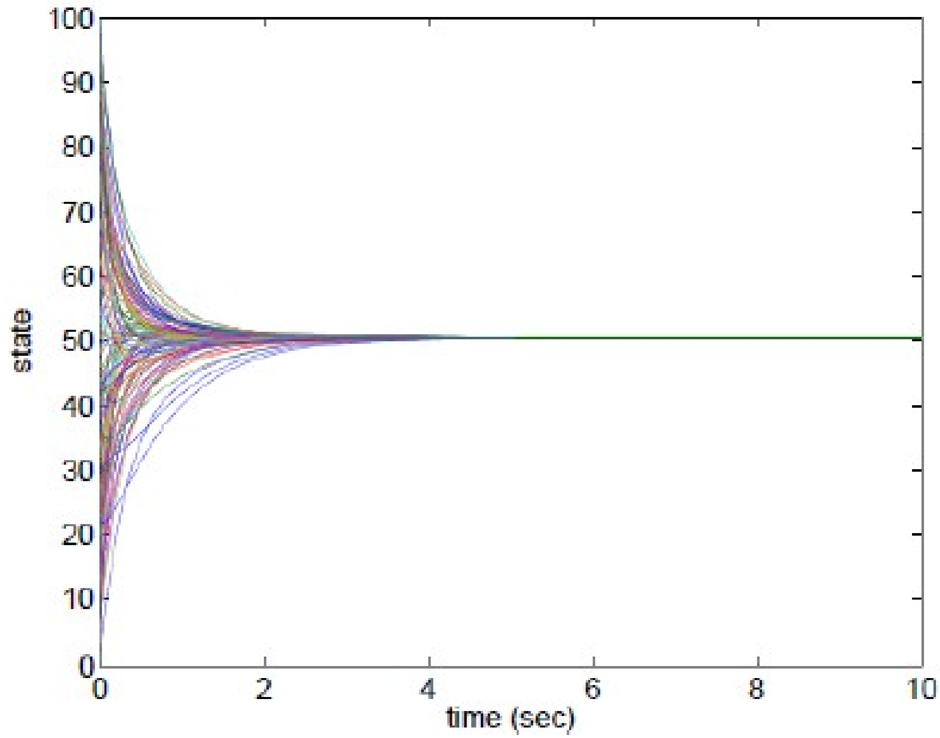
Distributed control via *Consensus*

Consensus can be achieved to the average value of the initial states of each agent (*simple integrator*).

$$x_i \rightarrow \frac{1}{N}\sum_{i=1}^{N} x_i(0)$$

$$\dot{x}_i = -\sum_{i \in \mathcal{N}_i} \left(x_i - x_j\right)$$

$$\forall i \in \mathcal{V}$$



Distributed control via *Consensus*

Consensus can be achieved to the median value of the initial states of each agent (*simple integrator*).

$$x_i \rightarrow m\,;$$

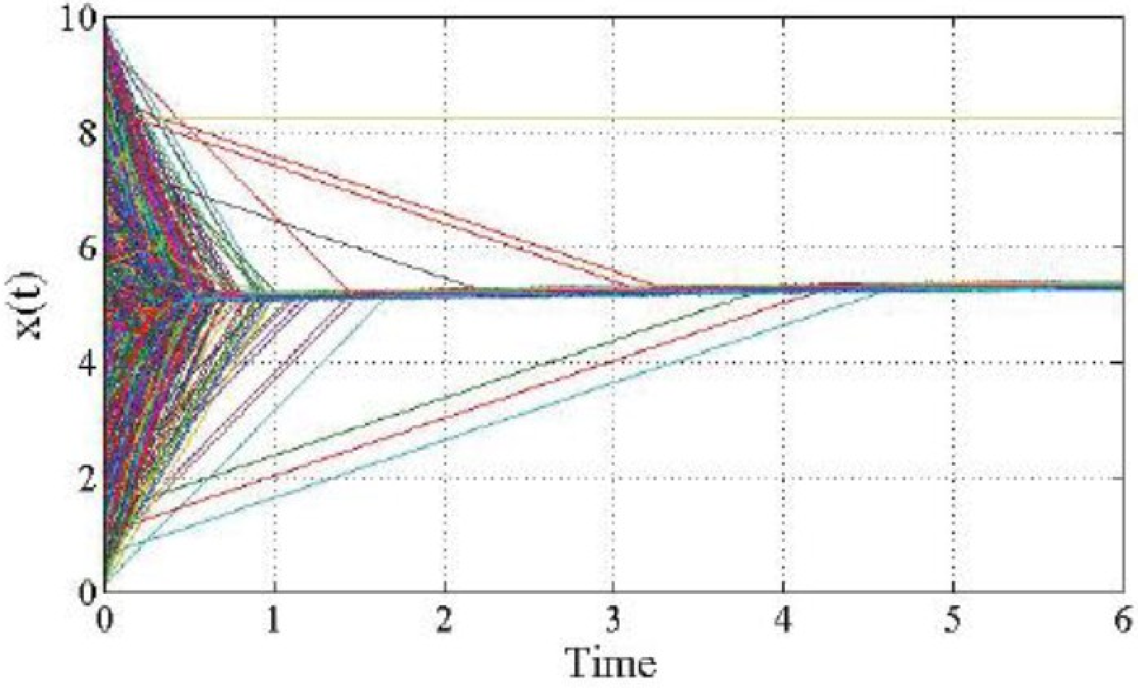
$$m \in \left\{ \begin{array}{ll} \left[x_k(0), x_{k+1}(0)\right] & k = \dfrac{N}{2},\ for\ N\ even \\ x_k(0) & k = \dfrac{N+1}{2},\ for\ N\ odd \end{array} \right\}$$

$$\dot{x}_i = -\alpha^2 sign\left(x_i - x_{i_0}\right) - \lambda^2 \sum_{i \in \mathcal{N}_i} sign\left(x_i - x_j\right)$$

$$\forall i \in \mathcal{V}$$

Distributed control via *Consensus*

Consensus can be achieved to the median value of the initial states of each agent (*simple integrator*).



Consensus to median value is more robust with respect to the presence of outlayers or uncooperative agents

Distributed control via *Consensus*

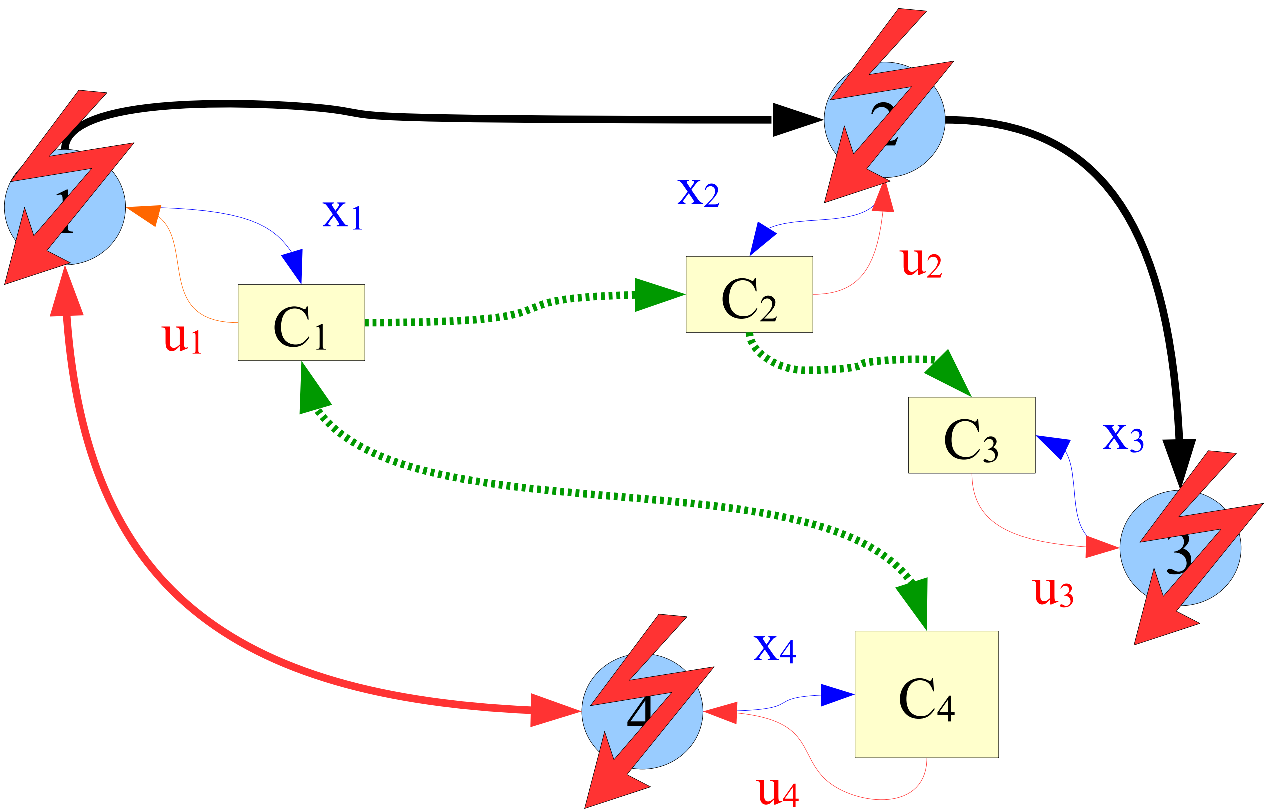
Consensus can be achieved to the median value of the initial states of each agent (*simple integrator*).

Agent	value	Attack 1	Attack 2	Attack 3	Attack 4	Attack 5	Attack 6
1	4	8	4	4	4	4	4
2	0	0	0	0	0	0	0
3	3	3	3	3	3	3	3
4	2	2	2	2	2	2	2
5	-1	-1	-4	-1	-1	-1	-1
6	1	1	1	4	1	1	1
7	1	1	1	1	1	1	1
8	---	---	---	---	5	2	-3
mean	**1,43**	2,00	1,00	1,86	1,88	1,50	0,88
median	**1,00**	**1,00**	**1,00**	2,00	1,50	1,50	**1,00**

Consensus to median value is more robust with respect to the presence of outlayers or uncooperative agents

Distributed control via *Consensus*

Consensus can be robustified with respect to matching faults and uncertainties.



Distributed control via *Consensus*

Consensus can be robustified with respect to matching faults and uncertainties.

$$\dot{x}_i(t) = w_i(t) + u_i(t), \quad i \in \mathcal{V}$$

$$\|w_i(t)\|_\infty \leq \Pi_i \leq \Pi$$

robustifying term

$$u_i(t) = -\alpha \cdot \sum_{j \in \mathcal{N}_i} (x_i(t) - x_j(t)) - \beta \cdot \mathrm{SIGN}(x_i(t) + z_i(t))$$

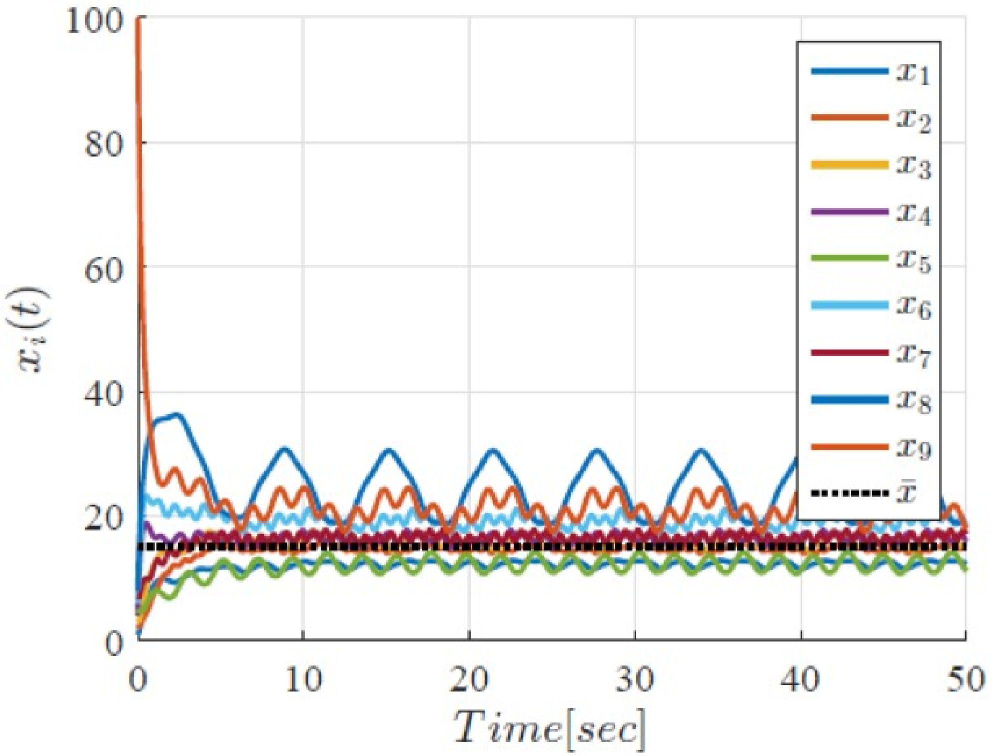
$$\dot{z}_i(t) = \alpha \cdot \sum_{j \in \mathcal{N}_i} (x_i(t) - x_j(t)), \qquad z_i(0) = -x_i(0)$$

$$\alpha > 0 \quad , \quad \beta > \Pi$$

Distributed control via *Consensus*

Consensus can be robustified with respect to matching faults and uncertainties.

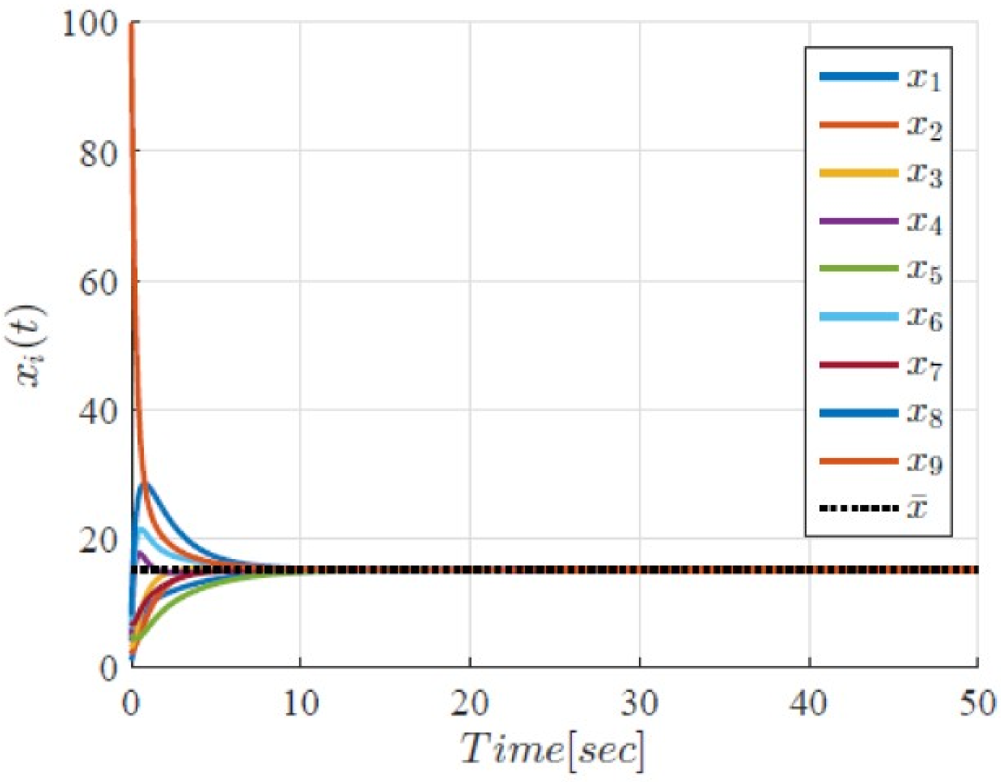
Simple average linear consensus



Distributed control via *Consensus*

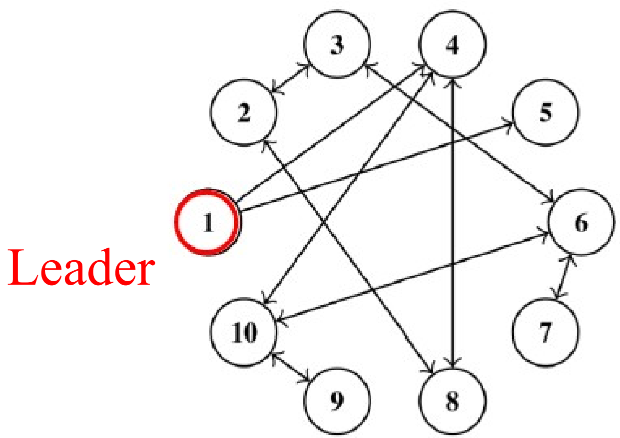
Consensus can be robustified with respect to matching faults and uncertainties.

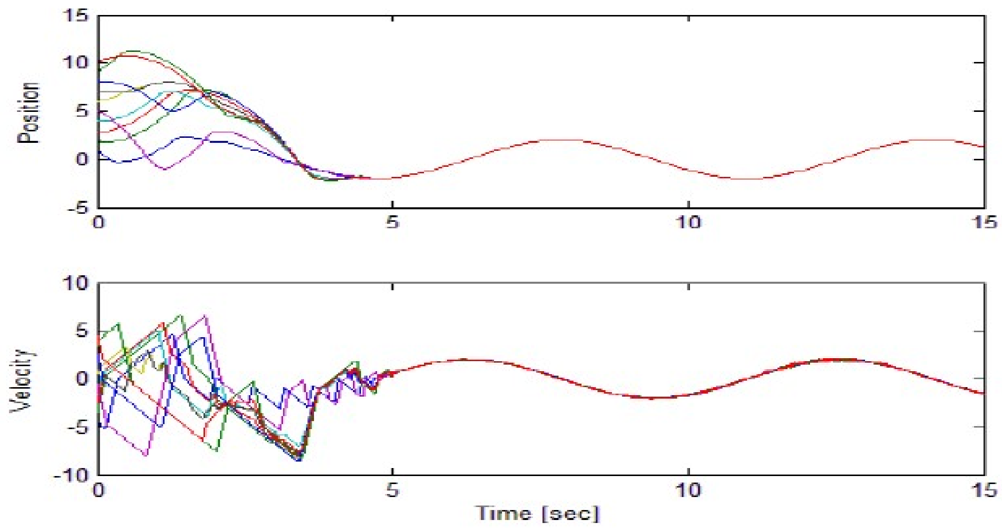
Robust average linear consensus



Distributed control via *Consensus*

Consensus can be used also to follow a leader.





Distributed control via *Consensus*

Consensus can be used also to achieve and keep a formation with respect to a leader.

Leader