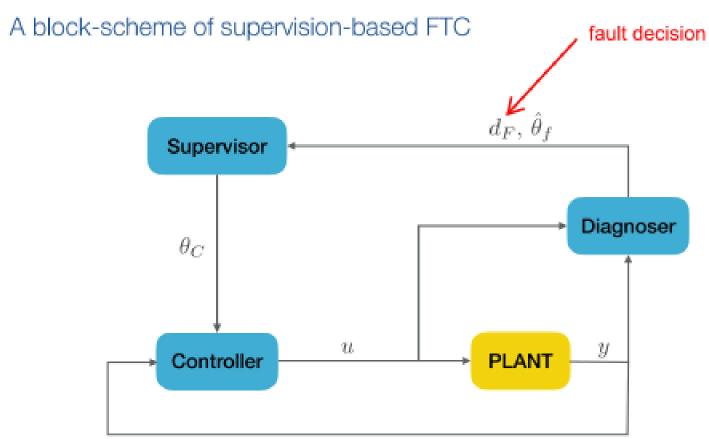
06_02_Active_Fault_Tolerant_Control

1. Structure of Active FTC



- **Supervision Block**
  - **fault diagnosis**
  - an **algorithm** to **alter the control law** based on the diagnosis
    - alteration is described by a **parameters vector** θ_c
    - θ_c can contain coefficients used by controller (**online re-design**), or can select a pre-computed controller (**control switching**)

2. One Active FTC Method: Model Matching Method

Main Idea

Given a **general control law** $u(t) = k\left(y(t), y_{\text{ref}}(t), \theta_c\right)$, produce **new parameters** such that the accommodated faulty closed loop system and the nominal one are "the same"

Pseudo-Inverse Method for Linear Systems

- In general (nonlinear dynamics etc.), there is **no guarantee** that the model-matching fault accommodation approach **has a solution**
- It has, for **linear systems** (if **some geometric conditions are met**)

Linear Systems with State-Feedback

Model:

$$\begin{cases} \dot{\boldsymbol{x}}(t) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) \\ \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t) \end{cases} \qquad \underline{\boldsymbol{u}(t) = -\boldsymbol{K}\boldsymbol{x}(t)}$$

$$\Rightarrow \begin{cases} \dot{\boldsymbol{x}}(t) &= (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K})\,\boldsymbol{x}(t) \\ \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t). \end{cases}$$

CLOSED LOOP DYNAMICS

After a fault the dynamics become:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}_f \boldsymbol{x}(t) + \boldsymbol{B}_f \boldsymbol{u}(t)$$
$$\boldsymbol{y}(t) = \boldsymbol{C}_f \boldsymbol{x}(t)$$

Controller Design

We want to design a **new controller gain** K_f such that

$$\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K} = \boldsymbol{A}_f - \boldsymbol{B}_f \boldsymbol{K}_f$$

- If solution exits, then just calculate
  - Solution exists when B and B_f have **same image(column space),** i.e. there are **redundant actuators**
- If not exists, use a **degraded goal:**

$$\|(\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}) - (\boldsymbol{A}_f - \boldsymbol{B}_f \boldsymbol{K}_f)\|$$

$$\boldsymbol{K}_f = \boldsymbol{B}_f^{+} (\boldsymbol{A}_f - \boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}) = \left(\boldsymbol{B}_f' \boldsymbol{B}_f\right)^{-1} \boldsymbol{B}_f' (\boldsymbol{A}_f - \boldsymbol{A} + \boldsymbol{B}\boldsymbol{K})$$

Output Feedback and Sensor Fault

Model

$$\begin{cases} \dot{\boldsymbol{x}}(t) & = & \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) \\ \boldsymbol{y}(t) & = & \boldsymbol{C}\boldsymbol{x}(t) \end{cases} \qquad \boldsymbol{u}(t) = -\boldsymbol{K}\boldsymbol{y}(t)$$

$$\Rightarrow \begin{cases} \dot{\boldsymbol{x}}(t) & = & (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}\boldsymbol{C})\,\boldsymbol{x}(t) \\ \boldsymbol{y}(t) & = & \boldsymbol{C}\boldsymbol{x}(t). \end{cases}$$

After a fault the dynamics become:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t)$$
$$\boldsymbol{y}(t) = \boldsymbol{C}_f\boldsymbol{x}(t)$$

Controller Design

We want to design **a new controller gain** K_f such that

$$\boldsymbol{K}_f C_f = \boldsymbol{K}\boldsymbol{C}$$

- Condition for exact solution:

$$\mathrm{Kern}\,(\boldsymbol{C}_f) \subseteq \mathrm{Kern}(\boldsymbol{C})$$

  Kern is the orthogonal complement of the row space, means that C_f is providing at least **the same information** as C

  Take the control law into consideration, means, the u space we can reach is at least large as the nominal case.

Then a solution is:

$$\boldsymbol{u}(t) = -\boldsymbol{K}\boldsymbol{P}\boldsymbol{y}(t) \quad \boldsymbol{P} = \boldsymbol{C}\boldsymbol{C}_f^+ = \boldsymbol{C}\boldsymbol{C}_f'\left(\boldsymbol{C}_f\boldsymbol{C}_f'\right)^{-1}$$

Output Feedback and Actuator Fault

Model

$$\begin{cases} \dot{\boldsymbol{x}}(t) & = & \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) \\ \boldsymbol{y}(t) & = & \boldsymbol{C}\boldsymbol{x}(t) \end{cases} \qquad \underline{\boldsymbol{u}(t) = -\boldsymbol{K}\boldsymbol{y}(t)}$$

$$\Rightarrow \begin{cases} \dot{\boldsymbol{x}}(t) & = & (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}\boldsymbol{C})\,\boldsymbol{x}(t) \\ \boldsymbol{y}(t) & = & \boldsymbol{C}\boldsymbol{x}(t). \end{cases}$$

After an **actuator-only fault** the dynamics become

$$\dot{\boldsymbol{x}}(t) = (\boldsymbol{A} - \boldsymbol{B}_f \boldsymbol{K}\boldsymbol{C})\,\boldsymbol{x}(t)$$
$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t).$$

Controller Design

We want to design **a new controller gain** K_f such that

$$\boldsymbol{B}_f \boldsymbol{K}_f = \boldsymbol{B}\boldsymbol{K}$$

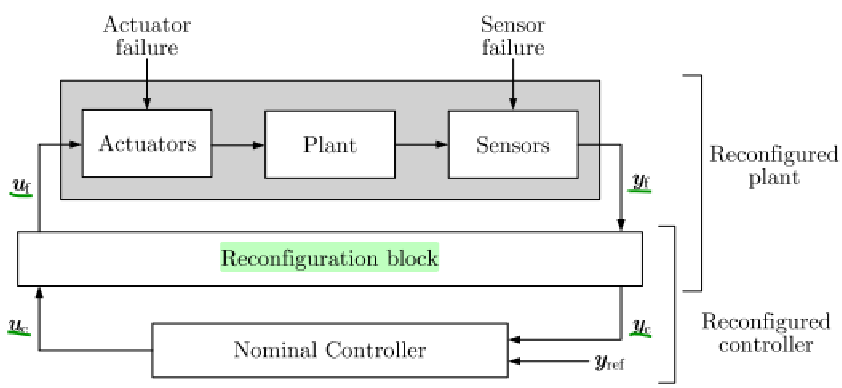
- A **exact solution** exits when:

$$\operatorname{Im}(\boldsymbol{B}_f) \supseteq \operatorname{Im}(\boldsymbol{B})$$

Then a solution is:

$$\boldsymbol{u}(t) = -\boldsymbol{N}\boldsymbol{K}\boldsymbol{y}(t) \quad \boldsymbol{N} = \boldsymbol{B}_f^{+}\boldsymbol{B} = \left(\boldsymbol{B}_f'\boldsymbol{B}_f\right)^{-1}\boldsymbol{B}_f'\boldsymbol{B}$$

3. Virtual Sensors and Virtual Actuators



Virtual Sensors

Motivation

- Model Matching Limitations:
  - controller must be re-designed
  - strong geometric assumptions
- Virtual Sensors Advantages:
  - does not require controller re-design
  - has less stringent assumptions (at least **detectability and stabilisability**)

Model 1: (Healthy Condition)

$$
\begin{cases}
\dot{\boldsymbol{x}}(t) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{E}\boldsymbol{d}(t)\,, \quad \boldsymbol{x}(0) = \boldsymbol{x}_0 \\
\boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t). \\
\\
\boldsymbol{u}_c(t) = -\boldsymbol{K}\boldsymbol{y}_c(t) + \boldsymbol{V}\boldsymbol{y}_{\text{ref}}(t)
\end{cases}
$$

The closed-loop dynamics:

$$
\begin{cases}
\dot{\boldsymbol{x}}(t) &= (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}\boldsymbol{C})\,\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{V}\boldsymbol{y}_{\text{ref}}(t) + \boldsymbol{E}\boldsymbol{d}(t), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0 \\
\boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t).
\end{cases}
$$

Model 2: Fault Formulation

A fault is assumed to **completely disable a sensor:** (assume C_f is known by us)

$$C \to C_f$$

$$\boldsymbol{y} \to \boldsymbol{y}_f$$

Goal

- Strong Goal:

$$\boldsymbol{y}_f(t) = \boldsymbol{y}(t)$$

- Weak Goal:
  - static: for constant d and y_{ref}

$$\boldsymbol{y}_f(t) \to \boldsymbol{y}(t) \text{ for } t \to \infty$$

  - dynamic: **transfer function** of reconfigured closed loop should be “**approximately the same**” as the one of the healthy system
  - In practice: dominant poles and zeros the same

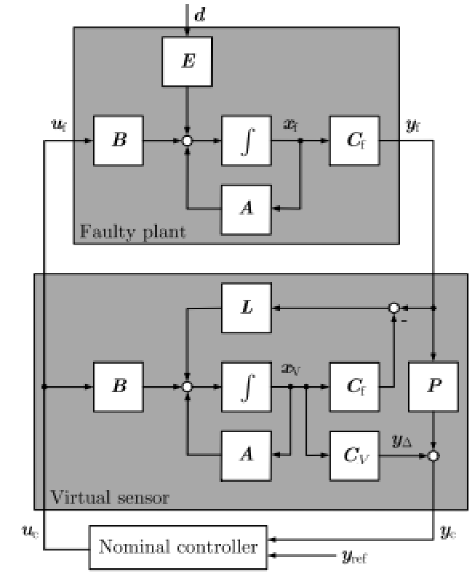
Virtual Sensor Design

The virtual sensor is implemented as a Luenberger observer

$$
\begin{cases}
\dot{\boldsymbol{x}}_V(t) &= \boldsymbol{A}_V\boldsymbol{x}_V(t) + \boldsymbol{B}_V\boldsymbol{u}_c(t) + \boldsymbol{L}\boldsymbol{y}_f(t), \qquad \boldsymbol{x}_V(0) = \boldsymbol{x}_{V0} \\
\boldsymbol{u}_f(t) &= \boldsymbol{u}_c(t) \\
\boldsymbol{y}_c(t) &= \underbrace{\boldsymbol{C}_V\boldsymbol{x}_V(t)} + \boldsymbol{P}\boldsymbol{y}_f(t)
\end{cases}
$$

where P and L are design parameters and

$$
\begin{aligned}
A_V &= A - LC_f \\
B_V &= B \\
C_V &= C - PC_f
\end{aligned}
$$



$$
\begin{aligned}
\begin{pmatrix} \dot{\boldsymbol{x}}_f(t) \\ \dot{\boldsymbol{x}}_V(t) \end{pmatrix} &= \begin{pmatrix} \boldsymbol{A} & \boldsymbol{O} \\ \boldsymbol{LC}_f & \boldsymbol{A} - \boldsymbol{LC}_f \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_f(t) \\ \boldsymbol{x}_V(t) \end{pmatrix} \\
&\quad + \begin{pmatrix} \boldsymbol{B} \\ \boldsymbol{B} \end{pmatrix} \boldsymbol{u}_c(t) + \begin{pmatrix} \boldsymbol{E} \\ \boldsymbol{O} \end{pmatrix} \boldsymbol{d}(t)
\end{aligned}
$$

$$
\boldsymbol{y}_c(t) = \begin{pmatrix} \boldsymbol{PC}_f & \boldsymbol{C}_V \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_f(t) \\ \boldsymbol{x}_V(t) \end{pmatrix}
$$

introduce **observation error** $x_\Delta = x_f - x_V$

$$
\begin{aligned}
\begin{pmatrix} \dot{\boldsymbol{x}}_f(t) \\ \dot{\boldsymbol{x}}_\Delta(t) \end{pmatrix} &= \begin{pmatrix} \boldsymbol{A} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{A} - \boldsymbol{LC}_f \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_f(t) \\ \boldsymbol{x}_\Delta(t) \end{pmatrix} \\
&\quad + \begin{pmatrix} \boldsymbol{B} \\ \boldsymbol{O} \end{pmatrix} \boldsymbol{u}_c(t) + \begin{pmatrix} \boldsymbol{E} \\ -\boldsymbol{E} \end{pmatrix} \boldsymbol{d}(t) \\
\boldsymbol{y}_c(t) &= \begin{pmatrix} \boldsymbol{C} & \boldsymbol{C}_V \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_f(t) \\ \boldsymbol{x}_\Delta(t) \end{pmatrix} \\
\begin{pmatrix} \boldsymbol{x}_f(0) \\ \boldsymbol{x}_\Delta(0) \end{pmatrix} &= \begin{pmatrix} \boldsymbol{x}_{f0} \\ \boldsymbol{x}_{V0} - \boldsymbol{x}_{f0} \end{pmatrix} .
\end{aligned}
$$

Then the **closed-loop dynamics** are

$$
\begin{aligned}
\begin{pmatrix} \dot{\boldsymbol{x}}_f(t) \\ \dot{\boldsymbol{x}}_\Delta(t) \end{pmatrix} &= \begin{pmatrix} \boldsymbol{A}-\boldsymbol{BKC} & -\boldsymbol{BKC}_V \\ \boldsymbol{O} & \boldsymbol{A}-\boldsymbol{LC}_f \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_f(t) \\ \boldsymbol{x}_\Delta(t) \end{pmatrix} + \\
&\quad + \begin{pmatrix} \boldsymbol{E} \\ -\boldsymbol{E} \end{pmatrix} \boldsymbol{d}(t) + \begin{pmatrix} \boldsymbol{BV} \\ \boldsymbol{O} \end{pmatrix} \boldsymbol{y}_{\mathrm{ref}}(t) \\
\boldsymbol{y}_f(t) &= \begin{pmatrix} \boldsymbol{C}_f & \boldsymbol{O} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_f(t) \\ \boldsymbol{x}_\Delta(t) \end{pmatrix}.
\end{aligned}
$$

Autonomous Behavior Case

Assume **autonomous behavior ($y_{ref} = 0$ and $d = 0$)**

Faulty syst. dynamics

cross-coupling $C_v = C - PC_f$

$$
\begin{aligned}
\begin{pmatrix} \dot{\boldsymbol{x}}_f(t) \\ \dot{\boldsymbol{x}}_\Delta(t) \end{pmatrix} &= \begin{pmatrix} \boxed{\boldsymbol{A}-\boldsymbol{BKC}} & \boxed{-\boldsymbol{BKC}_V} \\ \boldsymbol{O} & \boxed{\boldsymbol{A}-\boldsymbol{LC}_f} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_f(t) \\ \boldsymbol{x}_\Delta(t) \end{pmatrix} \\
\boldsymbol{y}_f(t) &= \begin{pmatrix} \boldsymbol{C}_f & \boldsymbol{O} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_f(t) \\ \boldsymbol{x}_\Delta(t) \end{pmatrix} \\
\begin{pmatrix} \boldsymbol{x}_f(0) \\ \boldsymbol{x}_\Delta(0) \end{pmatrix} &= \begin{pmatrix} \boldsymbol{x}_{f0} \\ \boldsymbol{x}_{V0} - \boldsymbol{x}_{f0} \end{pmatrix}.
\end{aligned}
$$

Virtual sensor

- By using **Separation principle**
  - The eigenvalues of the reconfigured closed loop are the union of
    - the eigenvalues of $A - BKC$
    - the eigenvalues of $A - LC_f$
  - If the healthy closed loop is stable, the first ones are stable
  - the second ones can be made stable by choosing L
    - always possible assuming the pair (A, C_f) is **observable**

Tracking Behavior Case

Assume x_{f_0}, x_{V_0} and d are all 0

$$
\begin{cases}
\dot{\boldsymbol{x}}_f(t) = (\boldsymbol{A}-\boldsymbol{BKC})\boldsymbol{x}_f(t) + \boldsymbol{BV}\boldsymbol{y}_{\mathrm{ref}}(t), & \boldsymbol{x}_f(0) = \boldsymbol{O} \\
\boldsymbol{y}_f(t) = \boldsymbol{C}_f \boldsymbol{x}_f(t), &
\end{cases}
$$

- Identical to the nominal one (**strong goal is reached**)

Virtual Actuators

Model 1: Healthy Condition

$$
\begin{cases}
\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{E}\boldsymbol{d}(t)\,, \quad \boldsymbol{x}(0) = \boldsymbol{x}_0 \\
\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t). \\
\boldsymbol{u}_c(t) = -\boldsymbol{K}\boldsymbol{y}_c(t) + \boldsymbol{V}\boldsymbol{y}_{\mathrm{ref}}(t)
\end{cases}
$$

The Closed-Loop Dynamics:

$$
\begin{cases}
\dot{\boldsymbol{x}}(t) = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}\boldsymbol{C})\,\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{V}\boldsymbol{y}_{\mathrm{ref}}(t) + \boldsymbol{E}\boldsymbol{d}(t), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0 \\
\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t).
\end{cases}
$$

Fault Formulation

A fault is assumed to **completely disable an actuator**

$$
\boldsymbol{B} \rightarrow \boldsymbol{B}_f
$$

$$
\boldsymbol{u} \rightarrow \boldsymbol{u}_f
$$

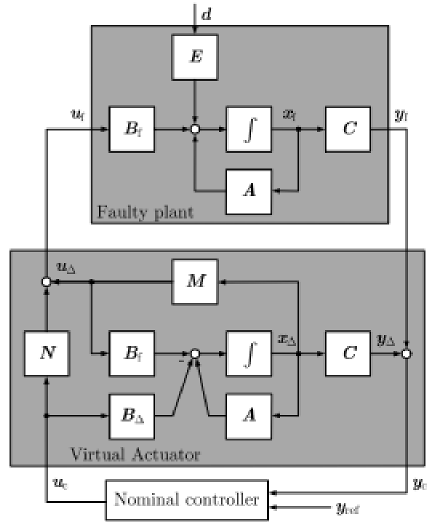
Virtual Actuator Design

The virtual actuator is the dual of the virtual sensor: requires **controllability** in the faulty condition

$$
\begin{cases}
\dot{\boldsymbol{x}}_\Delta(t) = \boldsymbol{A}_\Delta \boldsymbol{x}_\Delta(t) + \boldsymbol{B}_\Delta \boldsymbol{u}_c(t), \qquad \boldsymbol{x}_\Delta(0) = \boldsymbol{x}_{\Delta 0} \\
\boldsymbol{u}_f(t) = \boldsymbol{C}_\Delta \boldsymbol{x}_\Delta(t) + \boldsymbol{D}_\Delta \boldsymbol{u}_c(t) \\
\boldsymbol{y}_c(t) = \boldsymbol{C}\boldsymbol{x}_\Delta(t) + \boldsymbol{y}_f(t)
\end{cases}
$$

where M and N are design parameters

$$
\begin{aligned}
\boldsymbol{A}_\Delta &= \boldsymbol{A} - \boldsymbol{B}_f \boldsymbol{M} \\
\boldsymbol{B}_\Delta &= \boldsymbol{B} - \boldsymbol{B}_f \boldsymbol{N} \\
\boldsymbol{C}_\Delta &= \boldsymbol{M} \\
\boldsymbol{D}_\Delta &= \boldsymbol{N}.
\end{aligned}
$$



$$
\begin{pmatrix} \dot{\boldsymbol{x}}_f(t) \\ \dot{\boldsymbol{x}}_\Delta(t) \end{pmatrix} = \begin{pmatrix} \boldsymbol{A} & \boldsymbol{B}_f \boldsymbol{M} \\ \boldsymbol{O} & \boldsymbol{A} - \boldsymbol{B}_f \boldsymbol{M} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_f(t) \\ \boldsymbol{x}_\Delta(t) \end{pmatrix} + \begin{pmatrix} \boldsymbol{B}_f \boldsymbol{N} \\ \boldsymbol{B} - \boldsymbol{B}_f \boldsymbol{N} \end{pmatrix} \boldsymbol{u}_c(t) + \begin{pmatrix} \boldsymbol{E} \\ \boldsymbol{O} \end{pmatrix} \boldsymbol{d}(t)
$$

$$
\boldsymbol{y}_c(t) = \begin{pmatrix} \boldsymbol{C} & \boldsymbol{C} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_f(t) \\ \boldsymbol{x}_\Delta(t) \end{pmatrix}.
$$

Introduce a **new state** $\hat{\boldsymbol{x}}(t) = \boldsymbol{x}_f(t) + \boldsymbol{x}_\Delta(t)$

$$
\begin{aligned}
\frac{d}{dt} \begin{pmatrix} \hat{\boldsymbol{x}}(t) \\ \boldsymbol{x}_\Delta(t) \end{pmatrix} &= \begin{pmatrix} \boldsymbol{A} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{A} - \boldsymbol{B}_f \boldsymbol{M} \end{pmatrix} \begin{pmatrix} \hat{\boldsymbol{x}}(t) \\ \boldsymbol{x}_\Delta(t) \end{pmatrix} \\
&\quad + \begin{pmatrix} \boldsymbol{B} \\ \boldsymbol{B} - \boldsymbol{B}_f \boldsymbol{N} \end{pmatrix} \boldsymbol{u}_c(t) + \begin{pmatrix} \boldsymbol{E} \\ \boldsymbol{O} \end{pmatrix} \boldsymbol{d}(t) \\
\boldsymbol{y}_c(t) &= \begin{pmatrix} \boldsymbol{C} & \boldsymbol{O} \end{pmatrix} \begin{pmatrix} \hat{\boldsymbol{x}}(t) \\ \boldsymbol{x}_\Delta(t) \end{pmatrix} \\
\begin{pmatrix} \hat{\boldsymbol{x}}(0) \\ \boldsymbol{x}_\Delta(0) \end{pmatrix} &= \begin{pmatrix} \boldsymbol{x}_0 + \boldsymbol{x}_{\Delta 0} \\ \boldsymbol{x}_{\Delta 0} \end{pmatrix}.
\end{aligned}
$$

- x_Δ is not observable from y_c, it does not influence the I/O behavior. We can rewrite the dynamics:

$$\begin{cases} \dot{\boldsymbol{x}}(t) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}_c(t), \qquad \boldsymbol{x}(0) = \boldsymbol{x}_0 + \boldsymbol{x}_{\Delta 0} \\ \boldsymbol{y}_c(t) &= \boldsymbol{C}\boldsymbol{x}(t). \end{cases}$$

which is the similar to the healthy system

Autonomous Behavior

Assume **autonomous behavior ($y_{ref} = 0$ and $d = 0$)**

$$\begin{aligned} \frac{d}{dt}\begin{pmatrix} \hat{\boldsymbol{x}}(t) \\ \boldsymbol{x}_\Delta(t) \end{pmatrix} &= \begin{pmatrix} \boldsymbol{A} - \boldsymbol{BKC} & \boldsymbol{O} \\ -\boldsymbol{B}_\Delta \boldsymbol{KC} & \boldsymbol{A} - \boldsymbol{B}_f\boldsymbol{M} \end{pmatrix}\begin{pmatrix} \hat{\boldsymbol{x}}(t) \\ \boldsymbol{x}_\Delta(t) \end{pmatrix} \\ \begin{pmatrix} \hat{\boldsymbol{x}}(0) \\ \boldsymbol{x}_\Delta(0) \end{pmatrix} &= \begin{pmatrix} \boldsymbol{x}_0 + \boldsymbol{x}_{\Delta 0} \\ \boldsymbol{x}_{\Delta 0} \end{pmatrix}. \end{aligned}$$

> The **eigenvalues** of the **reconfigured closed loop** are the **union** of

a. the eigenvalues of $\boldsymbol{A}$-$\boldsymbol{BKC}$

b. the eigenvalues of $\boldsymbol{A}$-$\boldsymbol{B}_f\boldsymbol{M}$

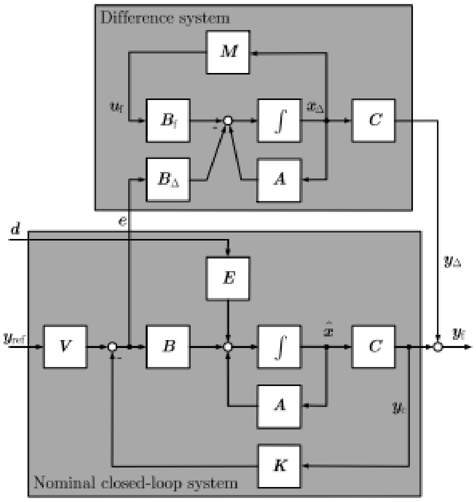
> If the **healthy closed loop** is **stable**, the **first** ones are **stable**

> the **second** ones can be **made stable** by **choosing** $\boldsymbol{M}$

> > always possible **assuming** the **pair** $(\boldsymbol{A},\boldsymbol{B}_f)$ is **stabilisable**

Analysis of I/O behavior of reconfigured closed-loop system

$$\begin{aligned} \frac{d}{dt}\begin{pmatrix} \hat{\boldsymbol{x}}(t) \\ \boldsymbol{x}_\Delta(t) \end{pmatrix} &= \begin{pmatrix} \boldsymbol{A} - \boldsymbol{BKC} & \boldsymbol{O} \\ -\boldsymbol{B}_\Delta \boldsymbol{KC} & \boldsymbol{A} - \boldsymbol{B}_f\boldsymbol{M} \end{pmatrix}\begin{pmatrix} \boldsymbol{x}(t) \\ \boldsymbol{x}_\Delta(t) \end{pmatrix} \\ &\quad + \begin{pmatrix} \boldsymbol{BV} \\ \boldsymbol{B}_\Delta \boldsymbol{V} \end{pmatrix}\boldsymbol{y}_{\mathrm{ref}}(t) + \begin{pmatrix} \boldsymbol{E} \\ \boldsymbol{O} \end{pmatrix}\boldsymbol{d}(t) \\ \begin{pmatrix} \hat{\boldsymbol{x}}(0) \\ \boldsymbol{x}_\Delta(0) \end{pmatrix} &= \begin{pmatrix} \boldsymbol{x}_0 + \boldsymbol{x}_{\Delta 0} \\ \boldsymbol{x}_{\Delta 0} \end{pmatrix} \\ \boldsymbol{y}_c(t) &= (\boldsymbol{C} \quad \boldsymbol{O})\begin{pmatrix} \hat{\boldsymbol{x}}(t) \\ \boldsymbol{x}_\Delta(t) \end{pmatrix} \\ \boldsymbol{y}_f(t) &= (\boldsymbol{C} \quad -\boldsymbol{C})\begin{pmatrix} \hat{\boldsymbol{x}}(t) \\ \boldsymbol{x}_\Delta(t) \end{pmatrix}. \end{aligned}$$



$$\boldsymbol{e} = \boldsymbol{V}\boldsymbol{y}_{\mathrm{ref}} - \boldsymbol{K}\boldsymbol{y}_c$$

$$\begin{cases} \dot{\boldsymbol{x}}_\Delta(t) = \underline{(\boldsymbol{A} - \boldsymbol{B}_f\boldsymbol{M})}\boldsymbol{x}_\Delta(t) + \underline{\boldsymbol{B}_\Delta\boldsymbol{e}(t)} \\ \boldsymbol{y}_\Delta(t) = \boldsymbol{C}\boldsymbol{x}_\Delta(t), \end{cases}$$

$$\boldsymbol{x}_\Delta(0) = \boldsymbol{x}_{\Delta 0}$$

- I/O behaviour from y_{ref} and d to y_c is the same as nominal system
- I/O behaviour from y_{ref} and d to y_f is affected by “difference system”
- To have complete reconfiguration, we would like to have

$$B_\Delta = B - B_f N = O$$

Note:

- The graph above is not a direct mapping from the virtual actuators real block graph. The mapping sequence is: VA real block graph → state-space model → I/O graph
- This graph is an I/O graph, that is from y_{ref}, d to y_f

Summary

- Structure of Active FTC:
  - diagnoser
  - supervisor → choose/change controller
- Method 1: Model Matching Method
  - Need model**,** then calculate new K to try to get the same performance or minimize the loss
  - Different scenarios
- Method 2: Virtual Sensors and Virtual Actuators
  - Need model
  - Design using **Separation Principle**