Unknown input observer for LPV systems with parameter varying output equation

D. Ichalal[1], B. Marx[2], J. Ragot[2], D. Maquin[2]

[1] Laboratoire d'Informatique, Biologie Intégrative et Systèmes Complexes
IBISC, Evry, France

[2] Centre de Recherche en Automatique de Nancy
CRAN, Nancy, France

Outline of the talk

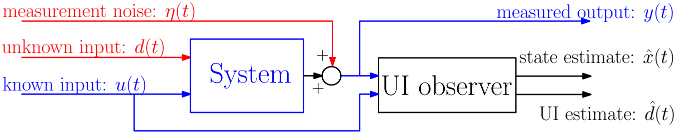
Introduction
- Context: estimation with unknown input
- Background on UIO design
- A motivating example
- Aim of the contribution

Main results: UIO design for LPV systems
- State estimation in the presence of UI
- UI estimation
- State estimation in the presence of UI and measurement noise

Example

Conclusion

Context: estimation with unknown input



- Unknown Input Observer (UIO) design:
  - knowing the system model, the input $u(t)$ and the output measurement $y(t)$
  - ignoring the unknown input $d(t)$ and the measurement noise $\eta(t)$
  - estimate the state variable $\hat{x}(t)$ and the unknown input $\hat{d}(t)$
- Applications in the SAFEPROCESS framework:
  - robust fault estimation
  - robust (actuator) fault diagnosis based on banks of UIOs
  - fault tolerant control based on state and fault estimates

Background on UIO design: the linear case

- For a given system with UI: $\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + Dd(t) \\ y(t) = Cx(t) \end{cases}$

- find the UI Observer gains: $\begin{cases} \dot{z}(t) = Nz(t) + Gu(t) + Ly(t) \\ \hat{x}(t) = z - Ey(t) \end{cases}$

  such that the estimation error: $x(t) - \hat{x}(t) = e(t) \to 0, \ \forall d(t), \ \forall u(t)$

Background on UIO design: the linear case

- For a given system with UI: $\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + Dd(t) \\ y(t) = Cx(t) \end{cases}$

- find the UI Observer gains: $\begin{cases} \dot{z}(t) = Nz(t) + Gu(t) + Ly(t) \\ \hat{x}(t) = z - Ey(t) \end{cases}$

  such that the estimation error: $x(t) - \hat{x}(t) = e(t) \to 0, \ \forall d(t), \ \forall u(t)$

- Decoupling conditions: $rank(CD) = rank(D) \Leftrightarrow \exists E : D = -ECD$

- Observer gain computation:

$$\begin{cases} E = -D(CD)^T(CD(CD)^T)^{-1} \\ G = (I + EC)B \\ \text{find } K \text{ such that } ((I + EC)A - KC) \text{ stable} \\ N = (I + EC)A - KC \\ L = K - NE \end{cases}$$

Background on UIO design: the **LPV** case

- For an UI LPV system: $$\begin{cases} \dot{x}(t) = A(\rho(t))x(t) + B(\rho(t))u(t) + D(\rho(t))d(t) \\ y(t) = Cx(t) \end{cases}$$
- use the poytopic transformation:

$$A(\rho(t)) = \sum_{i=1}^{r} h_i(\rho(t))A_i = A_h, \;\; with\ h_i \geq 0\ and\ \sum_{i=1}^{r} h_i(\rho(t)) = 1$$

- to get a polytopic one: $$\begin{cases} \dot{x}(t) = A_{\mathbf{h}}x(t) + B_{\mathbf{h}}u(t) + D_{\mathbf{h}}d(t) \\ y(t) = Cx(t) \end{cases}$$

Background on UIO design: the **LPV** case

- For an UI LPV system: $\begin{cases} \dot{x}(t) = A(\rho(t))x(t) + B(\rho(t))u(t) + D(\rho(t))d(t) \\ y(t) = Cx(t) \end{cases}$
- use the poytopic transformation:

$$A(\rho(t)) = \sum_{i=1}^{r} h_i(\rho(t))A_i = A_h, \text{ with } h_i \geq 0 \text{ and } \sum_{i=1}^{r} h_i(\rho(t)) = 1$$

- to get a polytopic one: $\begin{cases} \dot{x}(t) = A_{\mathbf{h}}x(t) + B_{\mathbf{h}}u(t) + D_{\mathbf{h}}d(t) \\ y(t) = Cx(t) \end{cases}$
- Design the LPV UIO: $\begin{cases} \dot{z}(t) = N_{\mathbf{h}}z(t) + G_{\mathbf{h}}u(t) + L_{\mathbf{h}}y(t) \\ \hat{x}(t) = z - Ey(t) \end{cases}$
- following the "linear" design except that ...

  ▹Polytopic stability: find K_i s.t.: $((I + EC)A_h - K_hC)$ is stable.

  ▹Linear decoupling condition: $rank(CD_i) = rank(D_i) \not\Rightarrow D_i = -ECD_i$

  ▹Decoupling condition: $rank(C[D_1 ... D_r]) = rank([D_1 ... D_r]) \Rightarrow D_\rho = -ECD_\rho$

A motivating example

Consider the following UI LPV system:

$$\begin{cases} \dot{x}(t) = \begin{pmatrix} 0 & \rho(t) \\ 1-\rho(t) & -3 \end{pmatrix} x(t) + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u(t) \begin{pmatrix} \rho(t) \\ \rho(t)+1 \end{pmatrix} d(t) \\ y(t) = \begin{pmatrix} 1 & 0 \end{pmatrix} x(t) \end{cases}$$

with the bounded time varying parameter:

$$2 \leq \rho(t) \leq 4$$

Consider the following UI LPV system:

$$\begin{cases} \dot{x}(t) = \begin{pmatrix} 0 & \rho(t) \\ 1-\rho(t) & -3 \end{pmatrix} x(t) + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u(t) \begin{pmatrix} \rho(t) \\ \rho(t)+1 \end{pmatrix} d(t) \\ y(t) = \begin{pmatrix} 1 & 0 \end{pmatrix} x(t) \end{cases}$$

with the bounded time varying parameter:

$$2 \leq \rho(t) \leq 4$$

Using the polytopic transformation, $\rho(t) \in [2\ 4]$, becomes:

$$\rho(t) = \underbrace{\left(\frac{\rho(t)-2}{2}\right)}_{h_1(t)} 4 + \underbrace{\left(\frac{4-\rho(t)}{2}\right)}_{h_2(t)} 2$$

with $h_1(\rho(t)) \geq 0$, $h_2(\rho(t)) \geq 0$ and $h_1(\rho(t)) + h_2(\rho(t)) = 1$

A motivating example

The system becomes polytopic:

$$\begin{cases} \dot{x}(t) = \sum_{i=1}^{2} h_i(\rho(t))(A_i x(t) + B_i u(t) + D_i d(t)) \\ y(t) = \begin{pmatrix} 1 & 0 \end{pmatrix} x(t) \end{cases}$$

with the vertices defined by:

$$A_1 = \begin{pmatrix} 0 & 4 \\ -3 & -3 \end{pmatrix},\ A_2 = \begin{pmatrix} 0 & 2 \\ -1 & -3 \end{pmatrix},\ B_1 = B_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix},\ D_1 = \begin{pmatrix} 4 \\ 5 \end{pmatrix},\ D_2 = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$$

A motivating example

The system becomes polytopic:

$$\begin{cases} \dot{x}(t) = \sum_{i=1}^{2} h_i(\rho(t))(A_i x(t) + B_i u(t) + D_i d(t)) \\ y(t) = \begin{pmatrix} 1 & 0 \end{pmatrix} x(t) \end{cases}$$

with the vertices defined by:

$$A_1 = \begin{pmatrix} 0 & 4 \\ -3 & -3 \end{pmatrix},\ A_2 = \begin{pmatrix} 0 & 2 \\ -1 & -3 \end{pmatrix},\ B_1 = B_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix},\ D_1 = \begin{pmatrix} 4 \\ 5 \end{pmatrix},\ D_2 = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$$

Focus on the decoupling conditions: find E satisfying $D_i = -ECD_i$

- $rank(D_i) = rank(CD_i)$ is not sufficient
- $rank([D_1 \ D_2]) = rank(C[D_1 \ D_2]) \Rightarrow D_i = -ECD_i$, is not satisfied
- The known UIO design for the LPV cannot be applied
- postpone the use of the conservative polytopic transformation and ensure $D(\rho(t)) = -E(?)CD(\rho(t))$ instead of $D_i = -ECD_i$

Aim and assumption of the contribution

Aims:

- design an UIO for LPV with LPV output equation and measurement noise:

$$\begin{cases} \dot{x}(t) = A(\rho(t))x(t) + B(\rho(t))u(t) + D(\rho(t))d(t) \\ y(t) = C(\rho(t))x(t) + \eta(t) \end{cases}$$

- relax the former decoupling conditions (given for constant C)

$$rank\left(\begin{bmatrix} D_1 & D_2 & \dots & D_r \end{bmatrix}\right) = rank\left(C\begin{bmatrix} D_1 & D_2 & \dots & D_r \end{bmatrix}\right)$$

Aim and assumption of the contribution

Aims:

- design an UIO for LPV with LPV output equation and measurement noise:

$$\begin{cases} \dot{x}(t) = A(\rho(t))x(t) + B(\rho(t))u(t) + D(\rho(t))d(t) \\ y(t) = C(\rho(t))x(t) + \eta(t) \end{cases}$$

- relax the former decoupling conditions (given for constant C)

$$rank\left(\begin{bmatrix} D_1 & D_2 & \dots & D_r \end{bmatrix}\right) = rank\left(C\begin{bmatrix} D_1 & D_2 & \dots & D_r \end{bmatrix}\right)$$

Assumptions:

- the matrices are affine in $\rho(t)$: $X(\rho(t)) = X_0 + \rho_1(t)X_1 + \cdots + \rho_{n_\rho}(t)X_{n_\rho}$
- the parameter $\rho(t)$ is real time accessible
- the parameter $\rho(t)$ and its derivative $\dot{\rho}(t)$ are bounded
- the measurement noise $\eta(t)$ and its derivative $\dot{\eta}(t)$ are bounded
- decoupling condition: $rank\left(C(\rho(t))D(\rho(t))\right) = rank\left(D(\rho(t))\right)$

State estimation in the presence of UI

- UI LPV System: $\begin{cases} \dot{x}(t) = A(\rho(t))x(t) + B(\rho(t))u(t) + D(\rho(t))d(t) \\ y(t) = C(\rho(t))x(t) \end{cases}$
- UI LPV Observer: $\begin{cases} \dot{z}(t) = N(\rho(t))z(t) + G(\rho(t))u(t) + L(\rho(t))y(t) \\ \hat{x}(t) = z(t) - E(\rho(t))y(t) \end{cases}$
- Notations:
  - LPV matrices: $X_\rho = X(\rho(t))$
  - $P_\rho = I + E_\rho C_\rho$
  - state estimation error: $e(t) = x(t) - \hat{x}(t)$

State estimation in the presence of UI

- UI LPV System: $$\begin{cases} \dot{x}(t) = A(\rho(t))x(t) + B(\rho(t))u(t) + D(\rho(t))d(t) \\ y(t) = C(\rho(t))x(t) \end{cases}$$
- UI LPV Observer: $$\begin{cases} \dot{z}(t) = N(\rho(t))z(t) + G(\rho(t))u(t) + L(\rho(t))y(t) \\ \hat{x}(t) = z(t) - E(\rho(t))y(t) \end{cases}$$
- Notations:
  - LPV matrices: $X_\rho = X(\rho(t))$
  - $P_\rho = I + E_\rho C_\rho$
  - state estimation error: $e(t) = x(t) - \hat{x}(t)$
- State estimation error dynamics:

$$\dot{e}(t) = N_\rho e(t) + (P_\rho B_\rho - G_\rho)u(t) + (P_\rho D_\rho)d(t) + (\dot{P}_\rho + P_\rho A_\rho - L_\rho C_\rho - N_\rho P_\rho)x(t)$$

State estimation in the presence of UI

- UI LPV System: $\begin{cases} \dot{x}(t) = A(\rho(t))x(t) + B(\rho(t))u(t) + D(\rho(t))d(t) \\ y(t) = C(\rho(t))x(t) \end{cases}$
- UI LPV Observer: $\begin{cases} \dot{z}(t) = N(\rho(t))z(t) + G(\rho(t))u(t) + L(\rho(t))y(t) \\ \hat{x}(t) = z(t) - E(\rho(t))y(t) \end{cases}$
- Notations:
  - LPV matrices: $X_\rho = X(\rho(t))$
  - $P_\rho = I + E_\rho C_\rho$
  - state estimation error: $e(t) = x(t) - \hat{x}(t)$
- State estimation error dynamics:

$$\dot{e}(t) = N_\rho e(t) + (P_\rho B_\rho - G_\rho)u(t) + (P_\rho D_\rho)d(t) + (\dot{P}_\rho + P_\rho A_\rho - L_\rho C_\rho - N_\rho P_\rho)x(t)$$

$\dot{e}(t) = N_\rho e(t)$: should be stable; $(P_\rho B_\rho - G_\rho)$: should be null; $(P_\rho D_\rho)$: should be null; $(\dot{P}_\rho + P_\rho A_\rho - L_\rho C_\rho - N_\rho P_\rho)$: should be null

LPV UIO design algorithm

[Step1] Define E_ρ by:

$$E_\rho = -D_\rho((C_\rho D_\rho)^T C_\rho D_\rho)^{-1}(C_\rho D_\rho)^T \quad \Rightarrow \dot{e} \text{ independent of } d$$

[Step1] Define E_ρ by:

$$E_\rho = -D_\rho((C_\rho D_\rho)^T C_\rho D_\rho)^{-1}(C_\rho D_\rho)^T \qquad \Rightarrow \dot{e} \textit{ independent of } d$$

[Step2] Use the polytopic transformation to write C_ρ and $\dot{P}_\rho + P_\rho A_\rho$ as:

$$\begin{cases} \dot{P}_\rho + P_\rho A_\rho = & \Sigma_{i=1}^{r} h_i(\rho,\dot{\rho})\mathcal{A}_i \\ C_\rho = & \Sigma_{i=1}^{r} h_i(\rho,\dot{\rho})\mathcal{C}_i \end{cases}$$

[Step3] Find $\overline{K}_i$ and $X = X^T > 0$ to solve the LMI problem:

$$\begin{cases} 0 > (X\mathcal{A}_i - \overline{K}_i\mathcal{C}_i) + (\ldots)^T \\ 0 > \left(\frac{r-1}{2}(X\mathcal{A}_i - \overline{K}_i\mathcal{C}_i) + X\mathcal{A}_i - \overline{K}_i\mathcal{C}_j + X\mathcal{A}_j - \overline{K}_j\mathcal{C}_i\right) + (\ldots)^T \end{cases}$$

and define: $K_i = \overline{K}_i X^{-1}$ $\qquad \Rightarrow (\mathcal{A}_h - K_h\mathcal{C}_h)$ *stable*

[Step1] Define E_ρ by:

$$E_\rho = -D_\rho((C_\rho D_\rho)^T C_\rho D_\rho)^{-1}(C_\rho D_\rho)^T \qquad \Rightarrow \dot{e} \textit{ independent of } d$$

[Step2] Use the polytopic transformation to write C_ρ and $\dot{P}_\rho + P_\rho A_\rho$ as:

$$\begin{cases} \dot{P}_\rho + P_\rho A_\rho = \Sigma_{i=1}^{r} h_i(\rho, \dot{\rho})\mathcal{A}_i \\ C_\rho = \Sigma_{i=1}^{r} h_i(\rho, \dot{\rho})\mathcal{C}_i \end{cases}$$

[Step3] Find $\overline{K}_i$ and $X = X^T > 0$ to solve the LMI problem:

$$\begin{cases} 0 > (X\mathcal{A}_i - \overline{K}_i\mathcal{C}_i) + (\dots)^T \\ 0 > \left(\frac{r-1}{2}(X\mathcal{A}_i - \overline{K}_i\mathcal{C}_i) + X\mathcal{A}_i - \overline{K}_i\mathcal{C}_j + X\mathcal{A}_j - \overline{K}_j\mathcal{C}_i\right) + (\dots)^T \end{cases}$$

and define: $K_i = \overline{K}_i X^{-1}$ $\qquad \Rightarrow (\mathcal{A}_h - K_h\mathcal{C}_h)$ *stable*

[Step4] UIO gain computation:

$$\begin{cases} K_\rho = \Sigma_{i=1}^{r} h_i(\rho, \dot{\rho})K_i \\ N_\rho = (\dot{P}_\rho + P_\rho A_\rho) - K_\rho C_\rho & \Rightarrow N_\rho \textit{ stable} \\ L_\rho = K_\rho - N_\rho E_\rho & \Rightarrow \dot{e} \textit{ independent of } x \\ G_\rho = P_\rho B_\rho & \Rightarrow \dot{e} \textit{ independent of } u \end{cases}$$

UI estimation

- Differentiating the output $y(t) = C_\rho x(t)$:

$$\dot{y}(t) = \dot{C}_\rho x(t) + C_\rho A_\rho x(t) + C_\rho B_\rho u(t) + C_\rho D_\rho d(t)$$

- Since $(C_\rho D_\rho)$ is full column rank:

$$d(t) = ((C_\rho D_\rho)^T C_\rho D_\rho)^{-1} (C_\rho D_\rho)^T \left(\dot{y}(t) - (\dot{C}_\rho + C_\rho A_\rho) x(t) - C_\rho B_\rho u(t) \right)$$

UI estimation

- Differentiating the output $y(t) = C_\rho x(t)$:
$$\dot{y}(t) = \dot{C}_\rho x(t) + C_\rho A_\rho x(t) + C_\rho B_\rho u(t) + C_\rho D_\rho d(t)$$
- Since $(C_\rho D_\rho)$ is full column rank:
$$d(t) = ((C_\rho D_\rho)^T C_\rho D_\rho)^{-1} (C_\rho D_\rho)^T \left(\dot{y}(t) - (\dot{C}_\rho + C_\rho A_\rho) x(t) - C_\rho B_\rho u(t) \right)$$
- **State and UI estimator:**
$$\begin{cases} \dot{z}(t) = N_\rho z(t) + G_\rho u(t) + L_\rho y(t) \\ \hat{x}(t) = z(t) - E_\rho y(t) \\ \hat{d}(t) = ((C_\rho D_\rho)^T C_\rho D_\rho)^{-1} (C_\rho D_\rho)^T \left(\dot{y}(t) - (\dot{C}_\rho + C_\rho A_\rho) \hat{x}(t) - C_\rho B_\rho u(t) \right) \end{cases}$$

UI estimation

- Differentiating the output $y(t) = C_\rho x(t)$:

$$\dot{y}(t) = \dot{C}_\rho x(t) + C_\rho A_\rho x(t) + C_\rho B_\rho u(t) + C_\rho D_\rho d(t)$$

- Since $(C_\rho D_\rho)$ is full column rank:

$$d(t) = ((C_\rho D_\rho)^T C_\rho D_\rho)^{-1}(C_\rho D_\rho)^T \left(\dot{y}(t) - (\dot{C}_\rho + C_\rho A_\rho)x(t) - C_\rho B_\rho u(t)\right)$$

- **State and UI estimator:**

$$\begin{cases} \dot{z}(t) = N_\rho z(t) + G_\rho u(t) + L_\rho y(t) \\ \hat{x}(t) = z(t) - E_\rho y(t) \\ \hat{d}(t) = ((C_\rho D_\rho)^T C_\rho D_\rho)^{-1}(C_\rho D_\rho)^T \left(\dot{y}(t) - (\dot{C}_\rho + C_\rho A_\rho)\hat{x}(t) - C_\rho B_\rho u(t)\right) \end{cases}$$

- The UI estimation error $e_d(t) = d(t) - \hat{d}(t)$ satisfies:

$$e_d(t) = -((C_\rho D_\rho)^T C_\rho D_\rho)^{-1}(C_\rho D_\rho)^T \left(\dot{C}_\rho + C_\rho A_\rho\right) e(t)$$

since $e(t) \to 0, \forall u, \forall d$, thus $\quad \Rightarrow \quad e_d(t) \to 0, \forall u, \forall d$

State estimation in the presence of UI and measurement noise

- Disturbed UI LPV System: $\begin{cases} \dot{x}(t) = A_\rho x(t) + B_\rho u(t) + D_\rho d(t) \\ y(t) = C_\rho x(t) + \eta(t) \end{cases}$
- UI LPV Observer: $\begin{cases} \dot{z}(t) = N_\rho z(t) + G_\rho u(t) + L_\rho y(t) \\ \hat{x}(t) = z(t) - E_\rho y(t) \end{cases}$
- State estimation error dynamics:

$$\dot{e}(t) = N_\rho e(t) + (P_\rho B_\rho - G_\rho) u(t) + (P_\rho D_\rho) d(t) + (\dot{P}_\rho + P_\rho A_\rho - L_\rho C_\rho - N_\rho P_\rho) x(t) + E_\rho \dot{\eta}(t) + \dot{E}_\rho \eta(t)$$

should be stable | should be null | should be null | should be null | should be minimized

State estimation in the presence of UI and measurement noise

- Disturbed UI LPV System: $\begin{cases} \dot{x}(t) = A_\rho x(t) + B_\rho u(t) + D_\rho d(t) \\ y(t) = C_\rho x(t) + \eta(t) \end{cases}$
- UI LPV Observer: $\begin{cases} \dot{z}(t) = N_\rho z(t) + G_\rho u(t) + L_\rho y(t) \\ \hat{x}(t) = z(t) - E_\rho y(t) \end{cases}$
- State estimation error dynamics:

$$\dot{e}(t) = N_\rho e(t) + (P_\rho B_\rho - G_\rho) u(t) + (P_\rho D_\rho) d(t) + (\dot{P}_\rho + P_\rho A_\rho - L_\rho C_\rho - N_\rho P_\rho) x(t) + E_\rho \dot{\eta}(t) + \dot{E}_\rho \eta(t)$$

should be stable — should be null — should be null — should be null — should be minimized

- Aim:
  - because of $\eta(t)$ global asymptotic stability of $e(t)$ cannot be ensured
  - using input-to-state stability (ISS), the boundedness of $e(t)$ is ensured

LPV UIO design algorithm with measurement noise

[Step1] Define E_ρ by: $E_\rho = -D_\rho((C_\rho D_\rho)^T C_\rho D_\rho)^{-1}(C_\rho D_\rho)^T$ $\Rightarrow$ *$\dot{e}$ indpt of d*

[Step2] Use the polytopic transformation to write :

$$\begin{cases} \dot{P}_\rho + P_\rho A_\rho = \Sigma_{i=1}^r h_i(\rho, \dot{\rho})\mathcal{A}_i \\ C_\rho = \Sigma_{i=1}^r h_i(\rho, \dot{\rho})\mathcal{C}_i \\ \begin{bmatrix} E_\rho & \dot{E}_\rho \end{bmatrix} = \Sigma_{i=1}^r h_i(\rho, \dot{\rho})\overline{E}_i \end{cases}$$

LPV UIO design algorithm with measurement noise

[Step1] Define E_ρ by: $E_\rho = -D_\rho((C_\rho D_\rho)^T C_\rho D_\rho)^{-1}(C_\rho D_\rho)^T$ $\Rightarrow$ *$\dot{e}$ indpt of d*

[Step2] Use the polytopic transformation to write :

$$\begin{cases} \dot{P}_\rho + P_\rho A_\rho = & \Sigma_{i=1}^r h_i(\rho, \dot{\rho})\mathcal{A}_i \\ C_\rho = & \Sigma_{i=1}^r h_i(\rho, \dot{\rho})\mathcal{C}_i \\ \begin{bmatrix} E_\rho & \dot{E}_\rho \end{bmatrix} = & \Sigma_{i=1}^r h_i(\rho, \dot{\rho})\overline{E}_i \end{cases}$$

[Step3] For $\alpha > 0$, find c, γ and $Q = Q^T > 0$, minimizing γ

$$\begin{cases} 0 \geq c - \alpha\gamma \\ 0 > \mathcal{M}_{ii} \\ 0 > \frac{1}{r-1}\mathcal{M}_{ii} + \mathcal{M}_{ij} + \mathcal{M}_{ji} \end{cases} \quad \textit{with } \mathcal{M}_{ij} = \begin{pmatrix} (Q\mathcal{A}_i - \overline{K}_i\mathcal{C}_j) + (\dots)^T + \alpha Q & Q\overline{E}_i \\ (Q\overline{E}_i)^T & -cI \end{pmatrix}$$

and define: $K_i = Q^{-1}\overline{K}_i$ $\Rightarrow$ *$e(t)$ bounded*

LPV UIO design algorithm with measurement noise

[Step1] Define E_ρ by: $E_\rho = -D_\rho((C_\rho D_\rho)^T C_\rho D_\rho)^{-1}(C_\rho D_\rho)^T$ $\Rightarrow$ *$\dot{e}$ indpt of d*

[Step2] Use the polytopic transformation to write :

$$\begin{cases} \dot{P}_\rho + P_\rho A_\rho = \Sigma_{i=1}^r h_i(\rho,\dot{\rho})\mathcal{A}_i \\ C_\rho = \Sigma_{i=1}^r h_i(\rho,\dot{\rho})\mathcal{C}_i \\ \begin{bmatrix} E_\rho & \dot{E}_\rho \end{bmatrix} = \Sigma_{i=1}^r h_i(\rho,\dot{\rho})\overline{E}_i \end{cases}$$

[Step3] For $\alpha > 0$, find c, γ and $Q = Q^T > 0$, minimizing γ

$$\begin{cases} 0 \geq c - \alpha\gamma \\ 0 > \mathcal{M}_{ii} \\ 0 > \frac{1}{r-1}\mathcal{M}_{ii} + \mathcal{M}_{ij} + \mathcal{M}_{ji} \end{cases} \quad \textit{with } \mathcal{M}_{ij} = \begin{pmatrix} (Q\mathcal{A}_i - \overline{K}_i\mathcal{C}_j) + (\dots)^T + \alpha Q & Q\overline{E}_i \\ (Q\overline{E}_i)^T & -cI \end{pmatrix}$$

and define: $K_i = Q^{-1}\overline{K}_i$ $\Rightarrow$ *$e(t)$ bounded*

[Step4] UIO gain computation:

$$\begin{cases} K_\rho = \Sigma_{i=1}^r h_i(\rho,\dot{\rho})K_i \\ N_\rho = (\dot{P}_\rho + P_\rho A_\rho) - K_\rho C_\rho \\ L_\rho = K_\rho - N_\rho E_\rho \\ G_\rho = P_\rho B_\rho \end{cases}$$

$\Rightarrow$ *$e(t)$ bounded* (for N_ρ)
$\Rightarrow$ *$\dot{e}$ independent of x* (for L_ρ)
$\Rightarrow$ *$\dot{e}$ independent of u* (for G_ρ)

Example 1: back to the motivating example

- Consider the following UI LPV system with LPV output

$$\begin{cases} \dot{x}(t) = \begin{pmatrix} 0 & \rho(t) \\ 1-\rho(t) & -3 \end{pmatrix} x(t) + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u(t) \begin{pmatrix} \rho(t) \\ \rho(t)+1 \end{pmatrix} d(t) \\ y(t) = \begin{pmatrix} \rho(t) & 0 \end{pmatrix} x(t) \end{cases}$$

with the bounded time varying parameter: $2 \leq \rho(t) \leq 4$.

- The classical decoupling condition cannot be checked
  - classical LPV UIO cannot be designed

Example 1: back to the motivating example

- Consider the following UI LPV system with LPV output

$$\begin{cases} \dot{x}(t) = \begin{pmatrix} 0 & \rho(t) \\ 1-\rho(t) & -3 \end{pmatrix} x(t) + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u(t) \begin{pmatrix} \rho(t) \\ \rho(t)+1 \end{pmatrix} d(t) \\ y(t) = \begin{pmatrix} \rho(t) & 0 \end{pmatrix} x(t) \end{cases}$$

with the bounded time varying parameter: $2 \leq \rho(t) \leq 4$.

- The classical decoupling condition cannot be checked
  - classical LPV UIO cannot be designed

- The condition $rank\left(D_\rho\right) = rank\left(C_\rho D_\rho\right)$ holds

  $rank\left(D_\rho\right) = 1$ and $rank\left(C_\rho D_\rho\right) = rank\left(\rho^2(t)\right) = 1$

  - with $E_\rho = \left[-\frac{1}{\rho(t)} \quad -\frac{1+\rho(t)}{\rho^2(t)}\right]^T$
    the proposed LPV UIO can be designed

Example 2: state and UI estimation

- Consider the UI LPV system

$$\begin{cases} \dot{x}(t) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -\rho(t) & -3 & -\rho(t) \end{pmatrix} x(t) + \begin{pmatrix} 2\rho(t) & 0 \\ 0 & 1 \\ 0 & -\rho(t) \end{pmatrix} d(t) \\ y(t) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} x(t) \end{cases}$$

with bounded parameter: $\rho(t) \in [1\ 3]$

Example 2: state and UI estimation

- Consider the UI LPV system

$$\begin{cases} \dot{x}(t) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -\rho(t) & -3 & -\rho(t) \end{pmatrix} x(t) + \begin{pmatrix} 2\rho(t) & 0 \\ 0 & 1 \\ 0 & -\rho(t) \end{pmatrix} d(t) \\ y(t) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} x(t) \end{cases}$$

  with bounded parameter: $\rho(t) \in [1\ 3]$
- decoupling condition for classical UIO design does not hold:

$$rank\,(C\,[D_1\ \ D_2]) \neq rank\,([D_1\ \ D_2])$$

Example 2: state and UI estimation

- Consider the UI LPV system

$$\begin{cases} \dot{x}(t) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -\rho(t) & -3 & -\rho(t) \end{pmatrix} x(t) + \begin{pmatrix} 2\rho(t) & 0 \\ 0 & 1 \\ 0 & -\rho(t) \end{pmatrix} d(t) \\ y(t) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} x(t) \end{cases}$$

  with bounded parameter: $\rho(t) \in [1\ 3]$
- decoupling condition for classical UIO design does not hold:

$$rank\left(C\left[D_1\ D_2\right]\right) \neq rank\left(\left[D_1\ D_2\right]\right)$$

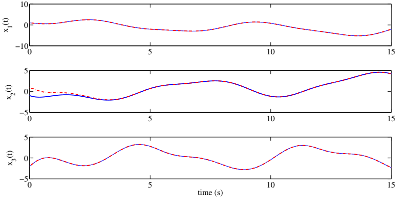
- proposed decoupling condition holds: $rank\left(C_\rho D_\rho\right) = rank\left(D_\rho\right)$

$$E_\rho = -D_\rho((C_\rho D_\rho)^T C_\rho D_\rho)^{-1}(C_\rho D_\rho)^T = \begin{pmatrix} -1 & 0 \\ 0 & 1/\rho(t) \\ 0 & -1 \end{pmatrix}$$

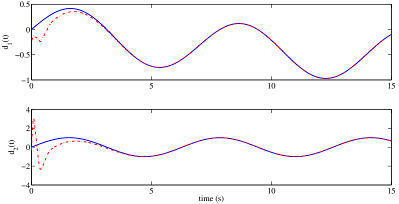
- the UIO design algorithm can be applied

Example2: state and UI estimation

original and estimated state



original and estimated UI



Conclusion

The aims of the contribution are:

- to design UI Observer for LPV systems
- to relax the classical decoupling condition
  by postponing the use of the polytopic transformation
- to adress the case of LPV output equation
- to jointly estimate the state and UI (fault)
- to attenuate measurement noise

Conclusion

The aims of the contribution are:

- to design UI Observer for LPV systems
- to relax the classical decoupling condition
  by postponing the use of the polytopic transformation
- to adress the case of LPV output equation
- to jointly estimate the state and UI (fault)
- to attenuate measurement noise

Main problem:

- what can be done when the parameter $\rho(t)$ is not accessible or known?

Thank you for your attention.

Unknown input observer for LPV systems with parameter varying output equation

D. Ichalal[1], B. Marx[2], J. Ragot[2], D. Maquin[2]

[1] Laboratoire d'Informatique, Biologie Intégrative et Systèmes Complexes
IBISC, Evry, France

[2] Centre de Recherche en Automatique de Nancy
CRAN, Nancy, France