# Modélisation LPV pour la commande d'un robot manipulateur parallèle

L. Casagrande Neves, <u>I. Queinnec</u>, L. Douat, V. Mahout

LAAS-CNRS, Université de Toulouse



### Robotique de manipulation (robots de pick-and-place)

- Diminution des temps de cycle
  - Diminution du temps de déplacement
  - Diminution du temps d'arrêt
- Robustesse vis-à-vis des conditions d'utilisation (variation de charge, de cycle, vibrations...)





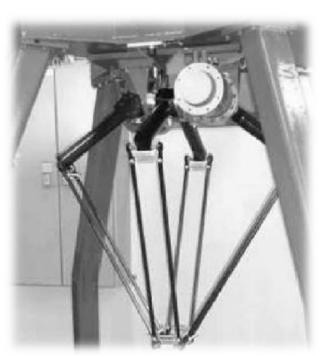
- Architectures mécaniques dynamiquement équilibrées
- Synthèse de contrôleurs présentant des propriétés garanties

Projet Objectif100G: partie « rigide »

# **Objectifs**

- Faire la synthèse de contrôleur pour le suivi de la trajectoire de pick-andplace en utilisant des techniques de commande robuste
  - Modèle rigide du robot

- Utilisation d'un modèle espace d'état incorporant des termes incertains, variant dans le temps et/ou des perturbations additives pour représenter les phénomènes non-linéaires « perdus » lors d'une linéarisation
  - Formalisme espace d'état
  - Variabilité le long de la trajectoire

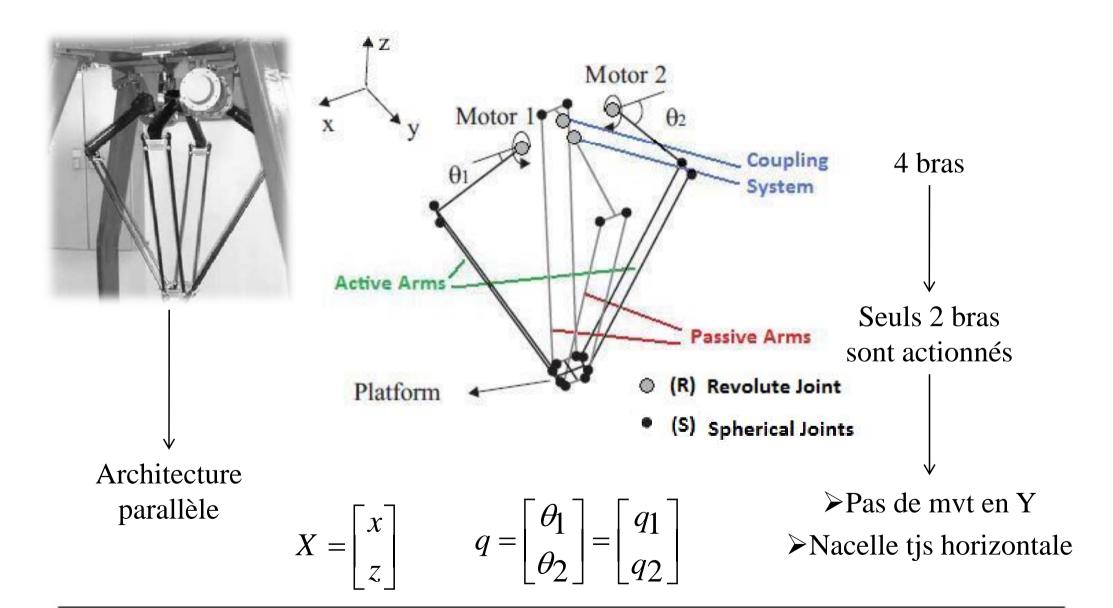


### Plan de la présentation

- I. Modèles « roboticiens »
- II. Passage du modèle dynamique à une forme espace d'état
- III. Modèle espace d'état affine en les paramètres
- IV. Validation des modèles
- V. Résultats préliminaires pour la synthèse de contrôleurs robustes
- VI. Conclusions / perspectives



### I. Les modèles du robot (1/2)



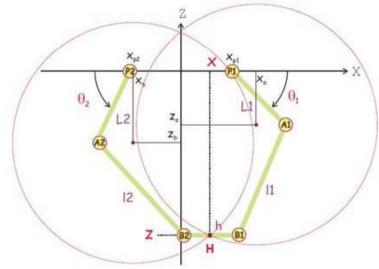
### I. Les modèles du robot (2/2)

- Modèle géométrique
  - Lien entre coordonnées cartésiennes et coordonnées angulaires : X = f(q)

Jacobienne de f

Modèle cinématique

- A partir de  $\dot{X} = J(q, X)\dot{q}$
- Mélange des coordonnées en X et en q
- Modèle dynamique



frottements 
$$\begin{bmatrix} I + J^T M_n J \end{bmatrix} \ddot{q} = \Gamma - f_S \operatorname{sgn}(\dot{q}) - f_V \dot{q} + M_C \cos(q) + J^T M_n G - J^T M_n \dot{J} \ \dot{q}$$
 Terme d'inertie Couples moteurs gravité

### II. Du modèle dynamique à une forme espace d'état (1/5)

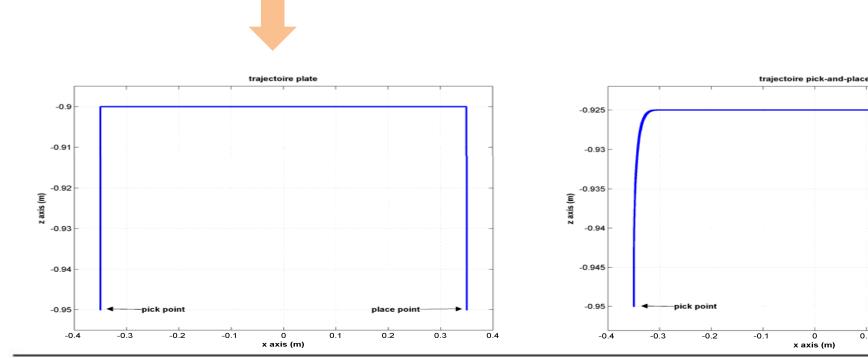
Objectif : Ecriture d'un modèle linéarisé le long d'une trajectoire de référence

→ Génération de la trajectoire de référence

• Trajectoire habituelle de pick-and-place  $(\Gamma_d, q_d, \dot{q}_d, \ddot{q}_d)$ 

• Utilisation des propriétés de platitude pour identifier

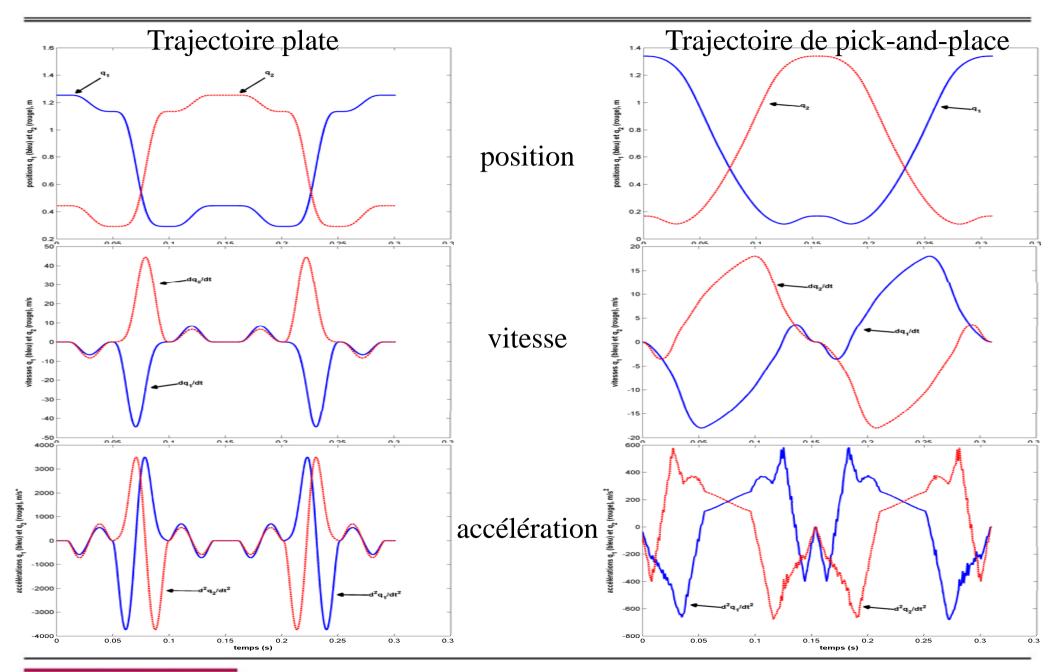
la trajectoire



0.2

0.3

## II. Du modèle dynamique à une forme espace d'état (2/5)





### II. Du modèle dynamique à une forme espace d'état (3/5)

Notation:

$$M(q) \qquad F_{1}(q)$$

$$\begin{bmatrix} I+J^{T}(q)M_{n}J(q) \end{bmatrix} \ddot{q} - \Gamma - \underbrace{\left(M_{C}\cos(q) + J^{T}(q)M_{n}G\right)}_{+f_{S}\operatorname{sgn}(\dot{q}) + f_{V}\dot{q}} + \underbrace{\left(T^{T}(\Theta)M_{n}J(q,\dot{q})\dot{q}\right)}_{F_{3}(q,\dot{q})} = 0$$

Choix d'une trajectoire  $(\Gamma_d, q_d, \dot{q}_d, \ddot{q}_d)$  telle que :

$$M(q_d)\ddot{q}_d - \Gamma_d + F_1(q_d) + F_2(\dot{q}_d) + F_3(q_d,\dot{q}_d) = 0$$

=> Ensemble de points discrets choisis le long de la trajectoire ou construction de la trajectoire continue

## II. Du modèle dynamique à une forme espace d'état (4/5)

• Construction du système d'erreur :

$$\begin{cases} q = q_d - \varepsilon \\ \dot{q} = \dot{q}_d - \dot{\varepsilon} \\ \ddot{q} = \ddot{q}_d - \ddot{\varepsilon} \end{cases} \quad \text{avec } \mathcal{E} = \begin{bmatrix} \mathcal{E}_1 \\ \mathcal{E}_2 \end{bmatrix} \quad \text{et les dérivées idem}$$

• On peut alors linéariser sous la forme  $G(\varepsilon, \dot{\varepsilon}, \ddot{\varepsilon}, q_d, \dot{q}_d, \ddot{q}_d, \Gamma, \Gamma_d) = 0$ 

$$\begin{bmatrix} \frac{\partial G}{\partial \ddot{\mathcal{E}}_{1}} & \frac{\partial G}{\partial \ddot{\mathcal{E}}_{2}} \end{bmatrix}_{\varepsilon^{(i)}=0} \ddot{\mathcal{E}} + \begin{bmatrix} \frac{\partial G}{\partial \dot{\mathcal{E}}_{1}} & \frac{\partial G}{\partial \dot{\mathcal{E}}_{2}} \end{bmatrix}_{\varepsilon^{(i)}=0} \dot{\mathcal{E}} + \begin{bmatrix} \frac{\partial G}{\partial \varepsilon_{1}} & \frac{\partial G}{\partial \varepsilon_{2}} \end{bmatrix}_{\varepsilon^{(i)}=0} \mathcal{E} + \begin{bmatrix} \frac{\partial G}{\partial \Gamma_{1}} & \frac{\partial G}{\partial \Gamma_{1}} \end{bmatrix}_{\varepsilon^{(i)}=0} (\Gamma - \Gamma_{d}) = 0$$

$$\Gamma = \Gamma_{d}$$

### II. Du modèle dynamique à une forme espace d'état (5/5)

• On écrit tout « proprement » et on obtient un modèle espace d'état:

$$\begin{cases} \dot{x} = A(q_d, \dot{q}_d, \ddot{q}_d) x + B(q_d, \dot{q}_d, \ddot{q}_d) u \\ y = Cx \end{cases}$$

$$\text{avec} \quad x = \begin{bmatrix} \varepsilon_1 & \varepsilon_2 & \dot{\varepsilon}_1 & \dot{\varepsilon}_2 \end{bmatrix}^T \quad \text{et} \quad u = \Gamma - \Gamma_d$$

mais dans lequel les paramètres de la trajectoire interviennent non linéairement. L'objectif est alors de raffiner encore le modèle pour obtenir une représentation polytopique ou LPV affine, dans laquelle les paramètres du modèle interviennent linéairement

**Remarque** : Pour comparer différentes stratégies, une autre option (qui ne prouve en fait rien) est de prendre une sélection de modèles linéaires à différents points discrets de la trajectoire

### III. Modèle espace d'état affine en les paramètres (1/4)

A partir du modèle

$$\begin{cases} \dot{x} = A(q_d, \dot{q}_d, \ddot{q}_d) x + B(q_d, \dot{q}_d, \ddot{q}_d) u \\ y = Cx \end{cases}$$

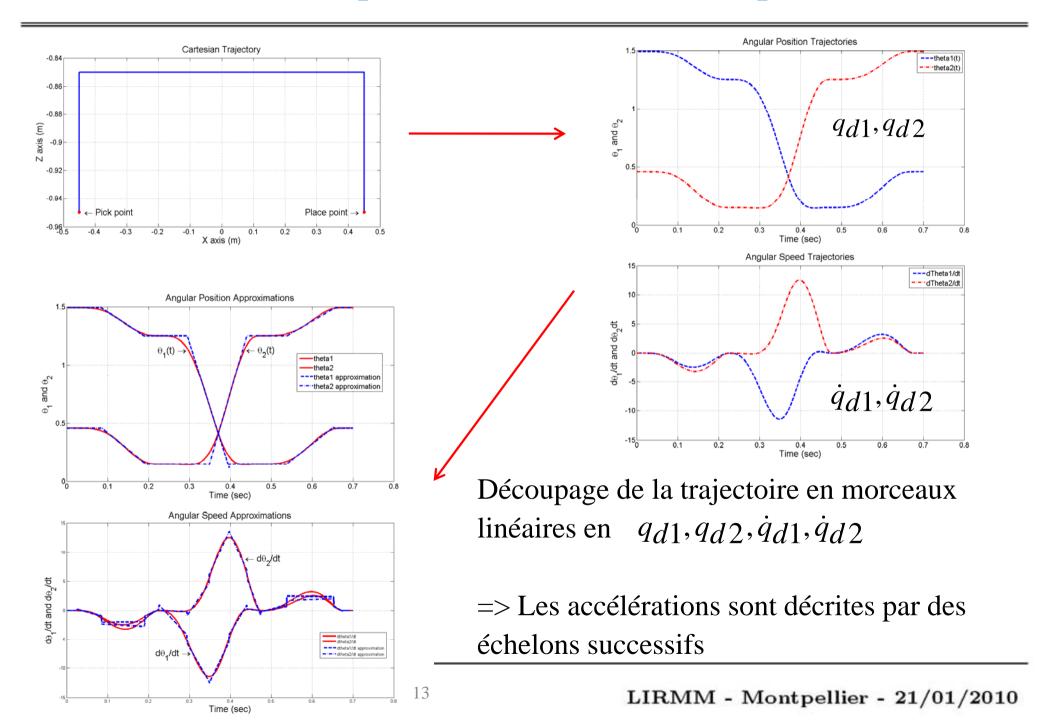
• Expression des éléments  $a_{ij}$  et  $b_{ij}$  des matrices A et B sous forme linéaire dans les paramètres

$$a_{ij} = h_{1,ij}q_{d1} + h_{2,ij}q_{d2} + h_{3,ij}\dot{q}_{d1} + h_{4,ij}\dot{q}_{d2} + h_{5,ij}\ddot{q}_{d1} + h_{6,ij}\ddot{q}_{d2}$$

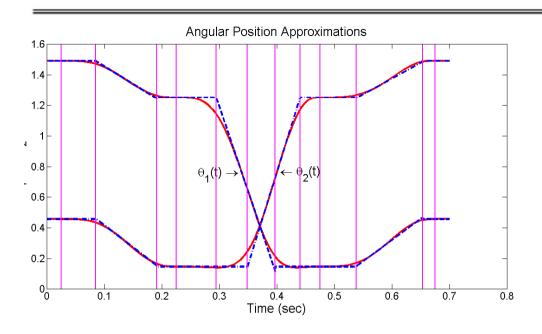
- Pas possible de trouver une forme unique sur tout le long de la trajectoire,
   compte tenu de la variation des paramètres
  - ⇒ Approximation par morceaux
  - ⇒ A partir d'une trajectoire de pick-and-place pré-choisie, on obtient les trajectoires en position et vitesse angulaires



## III. Modèle espace d'état affine en les paramètres (2/4)

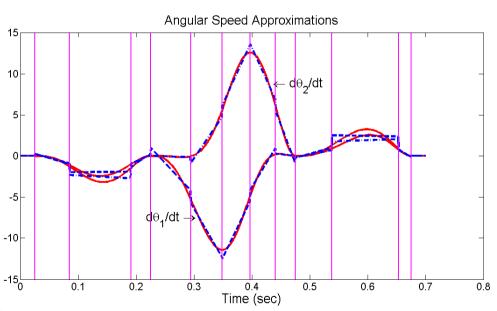


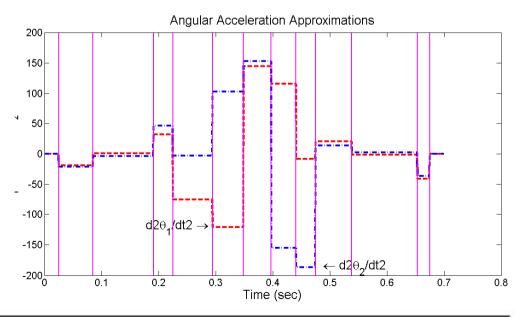
# III. Modèle espace d'état affine en les paramètres (3/4)



13 modèles successifs permettent de décrire la trajectoire, pour lesquels les accélérations sont constantes

$$=> a_{ij} = h_{1,ij}q_{d1} + h_{2,ij}q_{d2} + h_{3,ij}\dot{q}_{d1} + h_{4,ij}\dot{q}_{d2} + h_{7,ij}$$





### III. Modèle espace d'état affine en les paramètres (4/4)

- Utilisation de lsquurvefit de matlab pour estimer les approximations linéaires entre les paramètres  $a_{ii}$ ,  $b_{ii}$  et les paramètres de la trajectoire.
- Réduction du problème en utilisant seulement 2 paramètres de la trajectoire pour reconstruire tous les éléments des matrices A et B

$$a_{ij} = h_{1,ij}q_{d1} + h_{4,ij}\dot{q}_{d2} + h_{7,ij}$$

• On obtient au final un multi-modèle LPV (affine en les paramètres du vecteur  $\rho$ ) de la forme:

$$\begin{cases} \dot{x}(t) = \sum_{i=1}^{M} \mu_i(t) (A_i(\rho) x(t) + B_i(\rho) u(t)) \\ y(t) = Cx(t) \end{cases}$$

•  $\mu_i(t)$  traduit le switch entre un sous-modèle est le suivant.  $\mu_i(t)$  vaut 1 quand le modèle est actif, 0 lorsque c'est un autre qui est actif, 1/2 sur la frontière entre 2 modèles successifs

# Du modèle géométrique au modèle espace d'état



Modèle géométrique X = f(q)

Modèle cinématique  $\dot{X} = J(q, X)\dot{q}$ 

Modèle dynamique

$$M(q_d)\ddot{q}_d - \Gamma_d + F_1(q_d) + F_2(\dot{q}_d) + F_3(q_d,\dot{q}_d) = 0$$

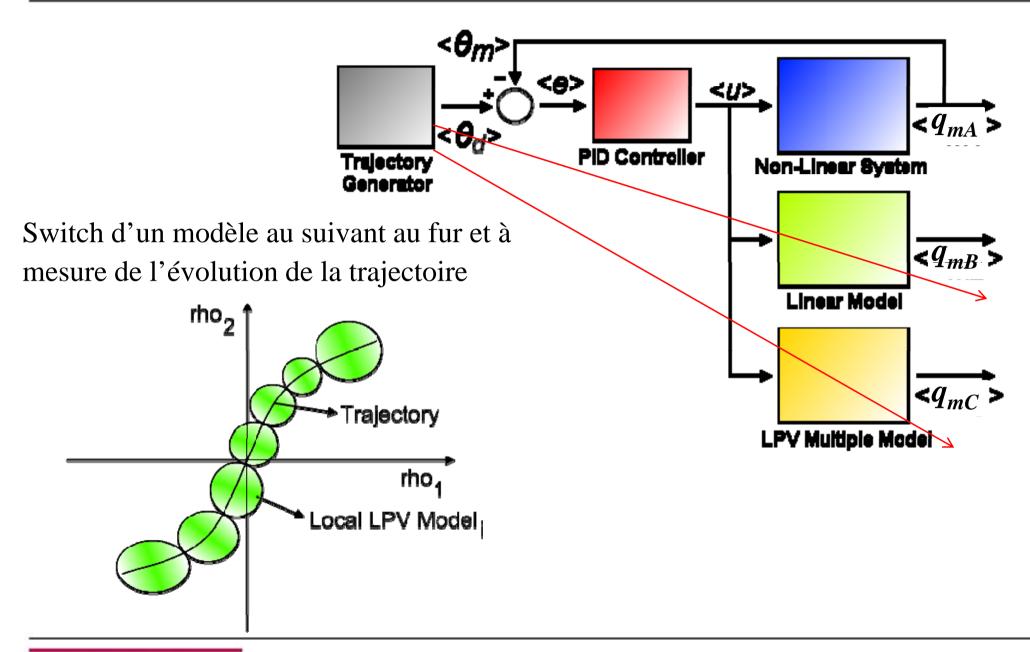
Modèle LPV non affine

$$\begin{cases} \dot{x} = A(q_d, \dot{q}_d, \ddot{q}_d) x + B(q_d, \dot{q}_d, \ddot{q}_d) u \\ y = Cx \end{cases}$$

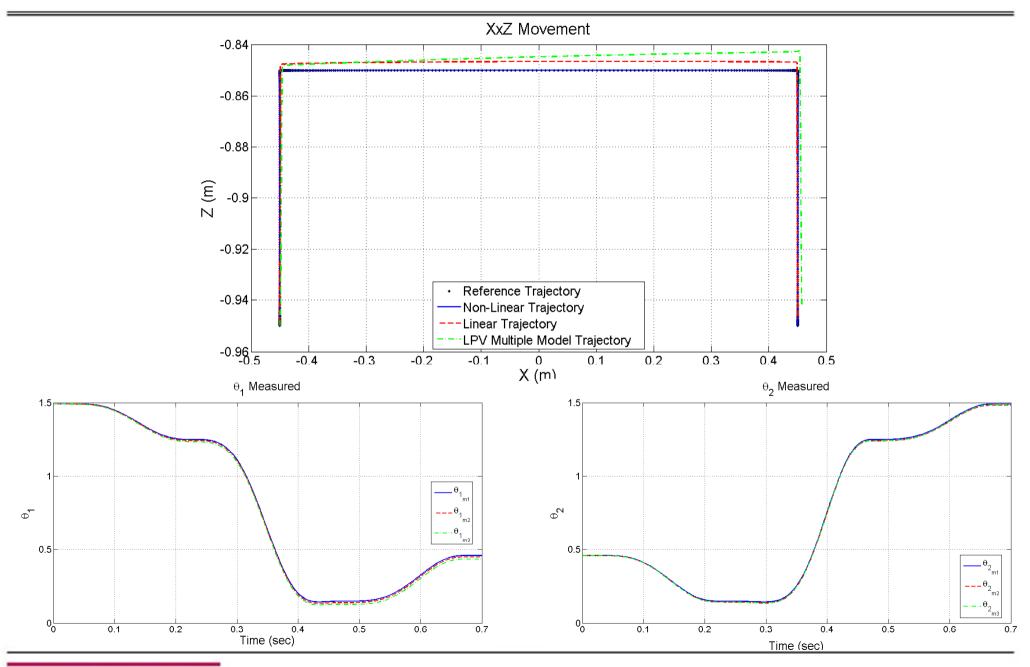
Multi-modèle LPV affine

$$\begin{cases} \dot{x}(t) = \sum_{i=1}^{M} \mu_i(t) (A_i(\rho) x(t) + B_i(\rho) u(t)) \\ y(t) = Cx(t) \end{cases}$$

### IV. Validation des modèles (1/2)



### IV. Validation des modèles (2/2)

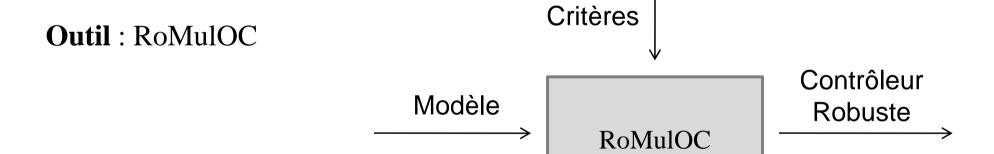




### V. synthèse de contrôleurs (1/7)

**Objectif** : synthèse d'un contrôleur *K* de retour d'état unique qui garantit la stabilité de tous les sous-modèles le long de la trajectoire

Version la plus simple (la plus conservative aussi) : problème de stabilité quadratique (P et K unique sur l'ensemble des sommets polytopiques définis le long de la trajectoire)



#### Stratégie de commande:

- On fixe une trajectoire désirée (pt de départ, d'arrivée, temps de parcours, ...)
- On génère l'ensemble des modèles pour cette trajectoire
- On résout le problème de synthèse robuste



### V. synthèse de contrôleurs (2/7)

#### Contrôleur « LPV »

**Objectif** : synthèse d'un contrôleur espace d'état unique, garantissant la stabilité du système tout au long de la trajectoire.

$$\begin{cases} \dot{x}(t) = \sum_{i=1}^{M} \mu_i(t) (A_i(\rho) x(t) + B_i(\rho) u(t)) \\ y(t) = Cx(t) \end{cases}$$

Chaque  $A_i(\rho)$ ,  $B_i(\rho)$  est décrit par un modèle polytopique :  $2^2$  sommets correspondant aux variations de tous les éléments du vecteur  $\rho$  entre une valeur min et une valeur max ( $\rho = \begin{bmatrix} q_{d1} & \dot{q}_{d2} \end{bmatrix}^T$ ) liées au tronçon de trajectoire considéré

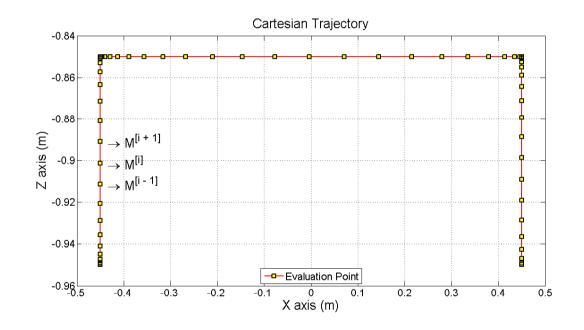
**Remarque**: L'ensemble des modèles  $A_i(\rho)$ ,  $B_i(\rho)$  peuvent aussi être décrit dans un formalisme polytopique sachant que  $\mu_i = 0$  ou  $1 \Rightarrow 0 \le \mu_i \le 1$ ,  $\sum_{i=1}^{M} \mu_i = 1$ 

#### Contrôleur « linéaire »

Problème de synthèse robuste pour un ensemble de systèmes linéaires

$$\begin{cases} \dot{x}(t) = A_i \ x(t) + B_i \ u(t) \\ y(t) = Cx(t) \end{cases}$$

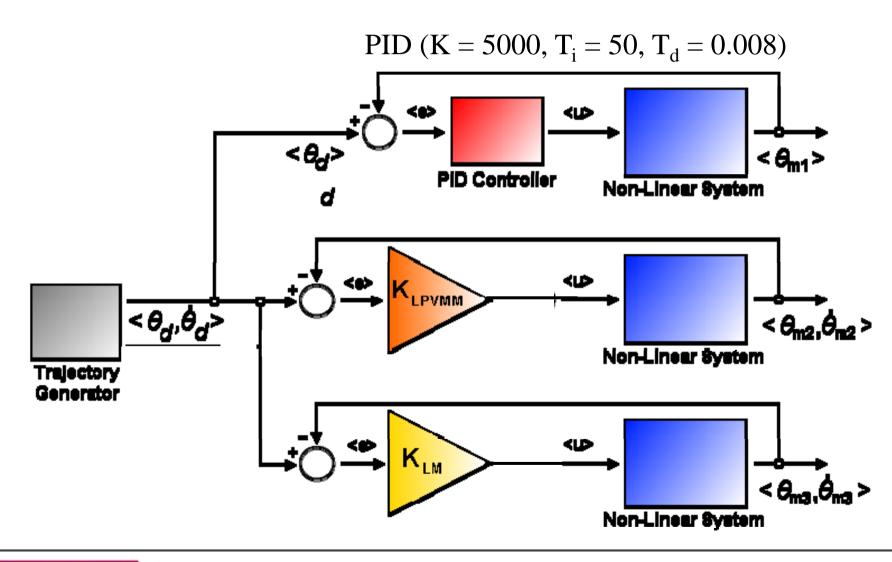
que l'on traite comme un problème de stabilisation quadratique classique



• Ce cas ne prouve rien, juste que le contrôleur est stabilisant de chacun des points... pas de la trajectoire

### V. synthèse de contrôleurs (4/7)

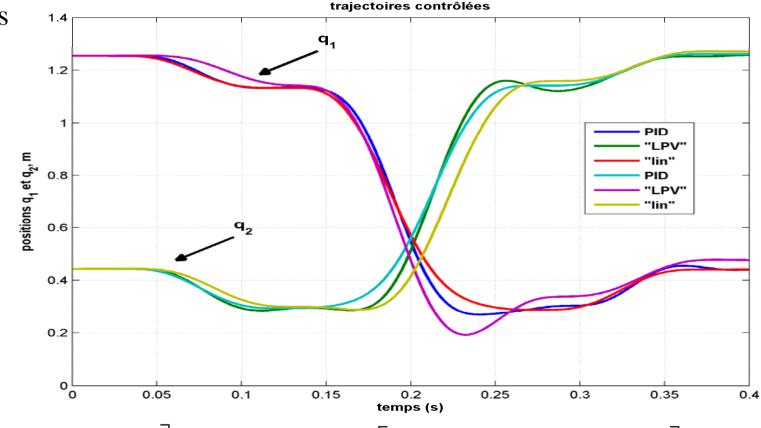
Comparaison des différents contrôleurs appliqués sur le simulateur original



# V. synthèse de contrôleurs (5/7)

### Trajectoire 1:

- z de -0.95 à -0.90, x de -0.35 à 0.35
- temps de parcours sur z = 0.1s, sur x = 0.15s
- temps d'arrêt = 0.05s



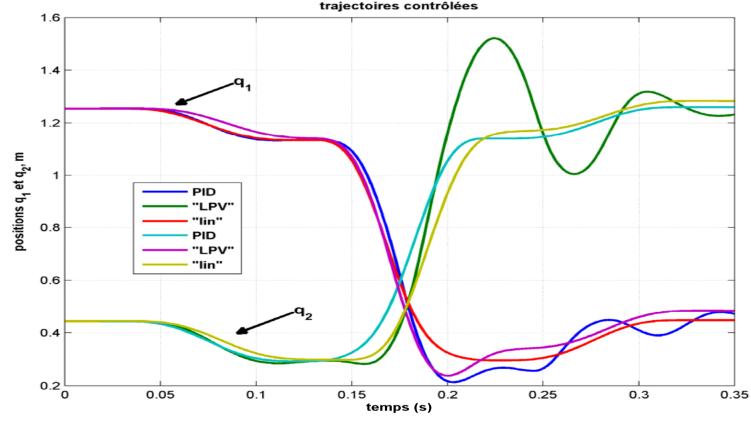
$$K_{lpv} = \begin{bmatrix} -409.9 & -32.7 & -122.6 & -241.0 \\ 451.8 & -135.9 & 220.3 & -5.6 \end{bmatrix}$$

$$\mathbf{K_{lin}} = \begin{bmatrix} -149.6 & 119.2 & -95.0 & 120.1 \\ -37.3 & -2.8 & 5.9 & -45.6 \end{bmatrix}$$

# V. synthèse de contrôleurs (6/7)

### Trajectoire 2:

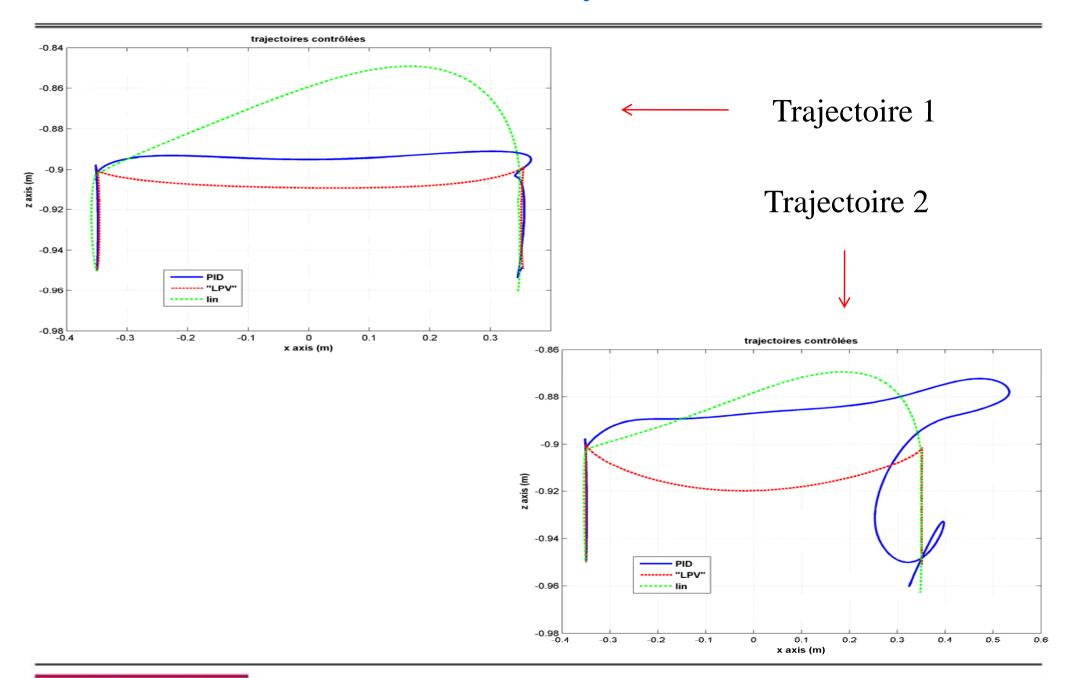
- z de -0.95 à -0.90, x de -0.35 à 0.35
- temps de parcours sur z = 0.1s, sur x = 0.1s
- temps d'arrêt = 0.05s



$$K_{lpv} = \begin{bmatrix} 49.1 & -269.7 & 138.3 & -392.8 \\ 1474.1 & -1084.8 & 1253.8 & -1167.9 \end{bmatrix}$$

$$K_{lin} = \begin{bmatrix} -158.3 & 113.4 - 114.9 & 114.1 \\ 18.7 & -52.1 & 28.9 & -78.7 \end{bmatrix}$$

### V. synthèse de contrôleurs (7/7)





### VI. Conclusions / Perspectives

- S'appuie sur la construction d'une trajectoire calculée en utilisant des notions de platitude
- Construction d'un multi-modèle le long de la trajectoire
  - ⇒ Validation pour diverses trajectoires
  - ⇒ multi-modèle versus modèle polytopique
- Uniquement des résultats préliminaires
  - ⇒ Rajouter des contraintes de performance sur la synthèse des contrôleurs (on a juste traiter le problème de faisabilité en terme de stabilité)
  - ⇒ Passer à du retour de sortie dynamique (typiquement du PID)
  - ⇒ Utiliser des stratégies dépendant des paramètres (au moins pour la fonction de Lyapunov)

