Apprentissage d'une loi de commande optimale d'un petit quadrotor pour le vol dans des tuyaux cylindriques

Vladislav Tempez

Directeurs de thèse : Jean-Baptiste MOURET,
Franck RUFFIER

Loria, INRIA

27 Juin 2022

Les environnements souterrains

Où et quoi ?

- Bâtiments et tunnels
- Conduits d'aération
- Pyramides !

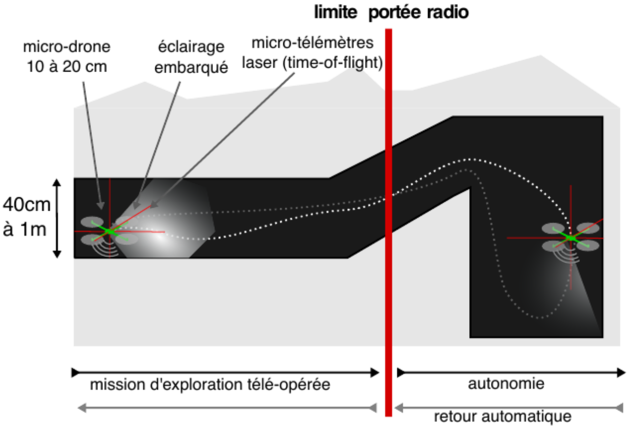
Pourquoi avec des robots ?

- Trop étroit pour des humains
- Dangereux, difficile d'accès, instable, mal aéré

Pourquoi avec des robots volants ?

- Évite un sol encombré (objets, liquide, etc)
- Gère les portions verticales ou pentues
- Parcours rapide de longs réseaux

Les environnements souterrains, exemple d'usage



Les environnements souterrains: difficultés pour les drones

Difficultés

- Se perturbe lui-même (flux d'air en intérieur)
- Étroit (peu de marge de manœuvre, risque de collision)
- Instable (collision=crash)

Photo: nicolasdohr.com

Vidéo drone perturbé (tuyau de 65cm)

Drone volant dans un tuyau de 65cm de diamètre

Le Crazyflie

Caractéristiques du Crazyflie

- 27g
- 4 rotor à 0.16N de poussée maximal (0.64N au total)
- 10cm de diagonale
- Firmware open source
- Système de modules pour ajouter ou retirer des fonctionnalités
- Plusieurs versions à partir de la même carte (moteurs brushless, variante plus grande)

Le quadrotor Crazyflie avec le deck multiranger (télémètres laser)

Contributions

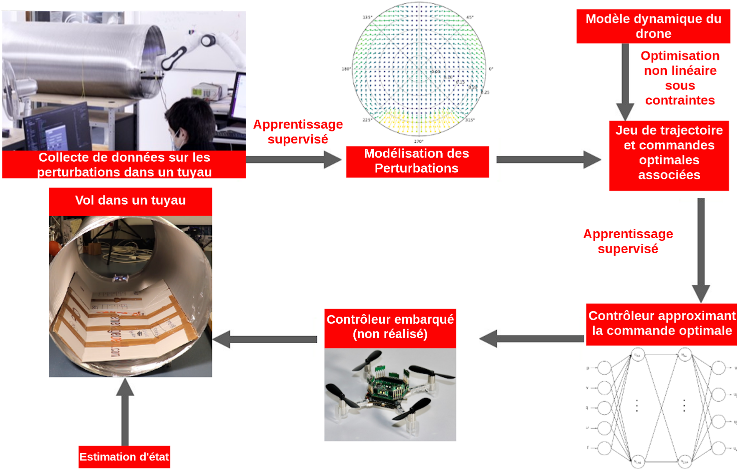
Mesure et Modélisation

1. Dispositif et protocole de mesure des perturbations statique dans un tuyau
2. Collecte de données de perturbations dans des tuyaux de diamètre différent
3. Apprentissage d'un modèle pour interpoler ces perturbations en tout point
4. Deux méthodes d'estimation de la position dans un tuyau.

Commande

1. Formulation et implémentation d'un contrôleur optimal pour quadrotor pour la stabilisation et le suivi de trajectoires
2. Intégration du modèle des perturbations au contrôleur optimal
3. Apprentissage supervisé d'un réseau de neurones imitant le contrôleur optimal

Contributions



Plan

Objectifs

- Vérifier l'existence des perturbations supposées
- Quantifier et cartographier ces perturbations
- Modéliser ces perturbations

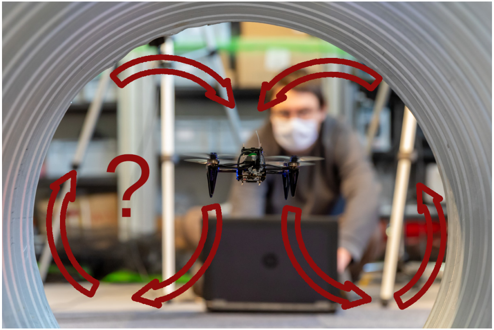


Photo: nicolasdohr.com

État de l'art

Effet de sol

Répulsion proche du sol, vu sur les hélicoptères depuis des décennies[1]

Effet de plafond

Effet d'aspiration à proximité d'un plafond[2]

Effet d'interaction entre rotors

Par des techniques de Vélocimétrie par Image de Particule (PIV): analyser l'interaction de flux de plusieurs rotors adjacents[3]

Dans un tuyau

Rien

[1] Cheeseman et Bennett, Aeronautical Research Council Reports and Memoranda 3021, 1955

[2] Powers et al., 13th International Symposium on Experimental Robotics, 2013

[3] Shukla et Komerath, Drones 2.4, 2018

Plan

Mesure des perturbations dans un tuyau

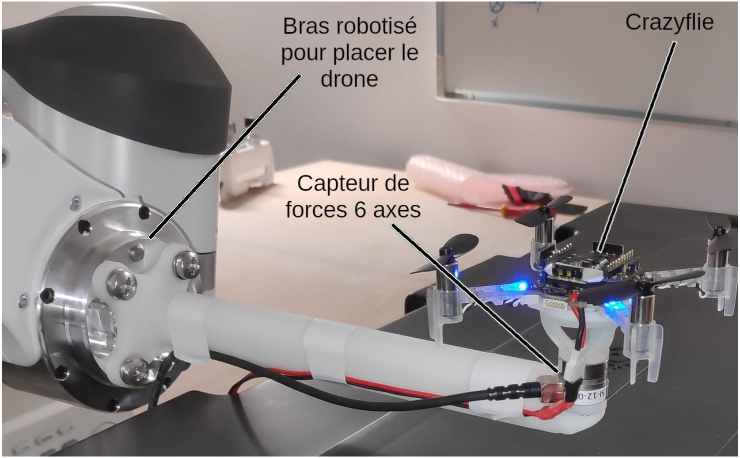
Mesure en vol libre

- Mesure en un point précis difficile
- Autonomie faible, nécessite une intervention régulière
- Robot instable, difficile à automatiser (crash fréquent)
- Mesure de la position difficile dans un tuyau

Mesures en statique (capteur de force 6 axes)

- Fixé donc pas de crash
- Alimenté donc autonome
- Mesure directe des perturbations

Dispositif de mesure



Protocole de mesure et traitement

- Mesure en extérieur (valeur de référence, 5s, moteurs allumés)
- Placement en intérieur (bras robot)
- Mesure en intérieur (différence à la référence, 10s)
- Filtrage à 1Hz de la mesure intérieure
- Valeur moyenne = perturbation statique
- Incertitude contenue dans la variance du signal

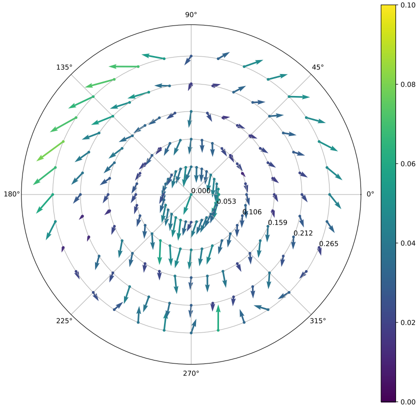
Séquence vidéo de collecte de données (accéléré x10)

Vidéo de la collecte de données (accéléré x10)

Perturbations mesurées (65cm)

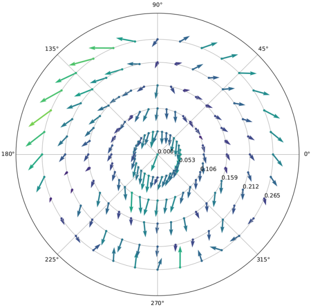
- $\simeq$ 100 points par tuyau (20s/point)
- Mesures dans 3 tuyaux (40cm, 50cm, 65cm)
- Forces non négligeables (0.1N)
- Effet de sol
- Aspiration par les parois
- Pas d'incertitude pour ne pas surcharger

1 flèche = 1 mesure (en N)

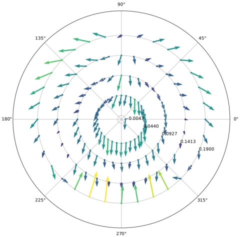


Sur les autres tuyaux

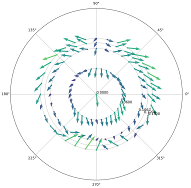
65cm



50cm



40cm



Plan

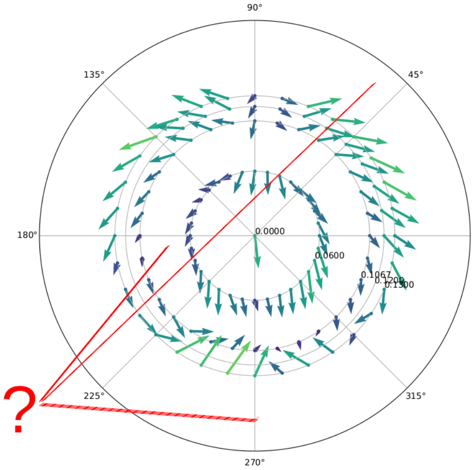
Pourquoi ?

- Pour l'utiliser dans un futur contrôleur (besoin d'estimer en des points intermédiaires)

Fonction de prédiction des forces apprise:

$$f : (x, z) \mapsto (f_x, f_z)$$

Mesure dans un tuyau de 40cm



Processus gaussien[1] (GP) ou réseau de neurones[2] (NN)?

Avantages (NN)

- Classique pour l'apprentissage supervisé
- Peu coûteux à évaluer

Limites (NN)

- Apprentissage coûteux
- Peu efficace en régime de données faible
- Sur-apprentissage

Avantages (GP)

- Efficace de régime de données faible
- Incertitude intégrée à la prédiction

Limites (GP)

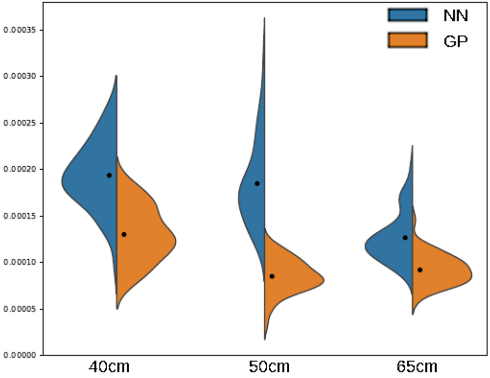
- Coût d'évaluation cubique en la quantité de données
- Difficile à mettre en place en régime de données important

[1] Rasmussen et Williams, MIT Press, 2006
[2] Haykin, Prentice Hall 1999

Choix de l'architecture

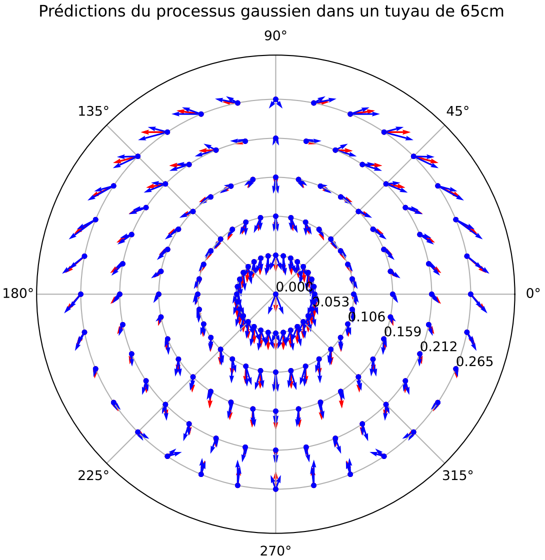
- Séparation des points en deux ensembles (90-10)
- Appris sur 90% des points
- Testé sur 10% restants

- Erreur plus faible pour GP
- Erreur moins dispersée pour GP
- GP utilisé pour la suite



Comparaison des erreurs sur l'ensemble de test

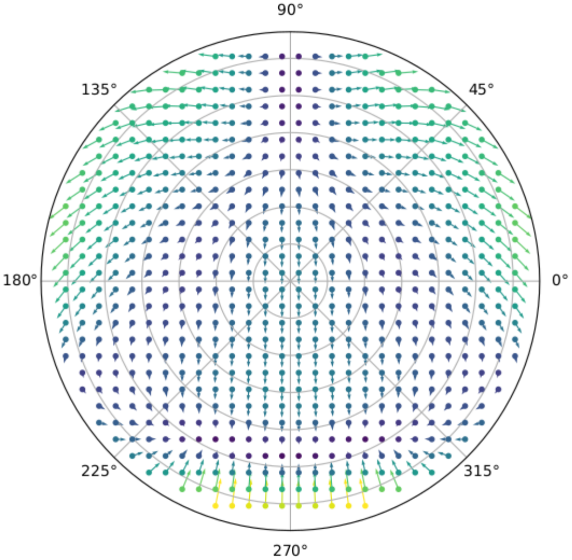
Visualisation de l'erreur de prédiction



Mesures (symetrisées) - Prédictions

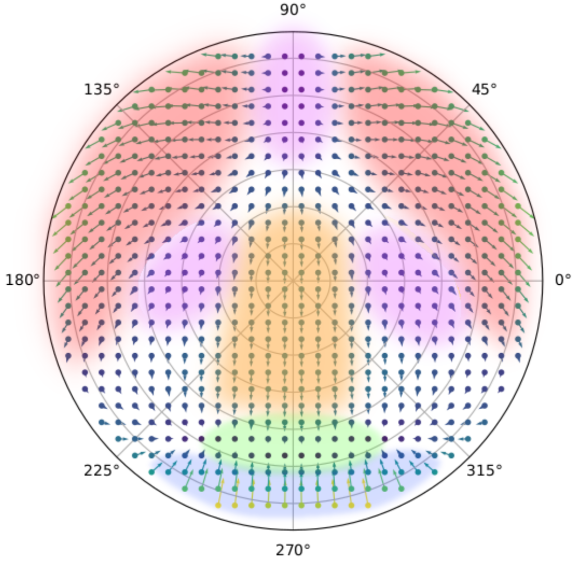
Résultat et analyse des perturbations (65cm - GP)

- Schéma symétrique
- Assez régulier (conséquence modélisation par GP)



Résultat et analyse des perturbations (65cm - GP)

- Effet de sol
- Aspiration latérale
- Faible intensité mais instables
- Constantes vers le bas
- Stable et de faible intensité



Bilan

Observations

- Perturbations statiques non négligeables
- Effet de sol, mais aussi une aspiration par les parois
- Le modèle d'interpolation permet d'observer des zones stables et instables

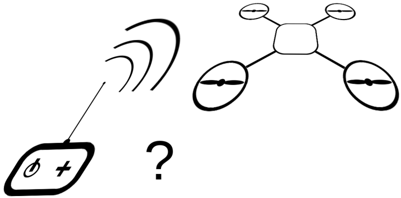
Limites

- Composantes statiques uniquement
- Probable biais causé par le dispositif de mesure

Plan

Objectifs

- Commander un robot à la dynamique non linéaire
- Prévenir, rejeter ou limiter les perturbations dues au tuyau
- Exercer un contrôle précis pour éviter les collisions



État de l'art

Backstepping control

Approche standard: organisation hiérarchique et récursive des états du robot à commander[1].

Apprentissage par renforcement

Recherche d'un contrôleur uniquement avec des exemples d'interaction (environnement simulé) [2].

Commande Optimale et Prédictive

Permet l'intégration d'un modèle des perturbations plus aisée, capacité à planifier et à prévoir[3][4]. S'appuie sur la résolution de problème d'optimisation par des solveurs bien établis.

---

[1] Lee, Leok et McClamroch, 49th IEEE Conference on Decision and Control, 2010

[2] Molchanov et al, IROS, 2019

[3] Carlos et al, ICARCV, 2020

[4] Bansal et al,IEEE 55th Conference on Decision and Control, 2016

Plan

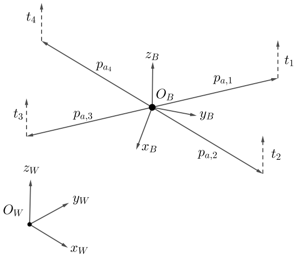
État d'un quadrotor

Représentation de l'état

$$
\begin{aligned}
p &= \overrightarrow{O_W O_B} \\
v &= \frac{dp}{dt} \\
R_B^W &= \left([x_B]_W \middle| [y_B]_W \middle| [z_B]_W \right) \\
\omega_{WB} &= \text{vitesse angulaire} \\
f_i, &\quad i \in [|0, 4|]
\end{aligned}
$$

force exercée
par le rotor i

Schéma d'un quadrotor



Dynamique d'un quadrotor

Équation de la dynamique

dérivée de l'état

masse

Intensité de la force produite par le rotor i

le long de l'axe du rotor

$$
\begin{aligned}
\dot{p} &= v \\
\dot{v} &= \frac{1}{m}\sum_i f_i t_i \\
\dot{R}_B^W &= R_B^W [\omega_{WB}]_\wedge \\
\dot{\omega}_{WB} &= J^{-1}\Big(\sum_i f_i(p_{a_i} \wedge t_i + c_i t_i) \\
&\qquad - \omega_{WB} \wedge (J\omega_{WB})\Big) \\
\dot{f}_i &= u_i
\end{aligned}
$$

opérateur produit vectoriel

couples des rotors

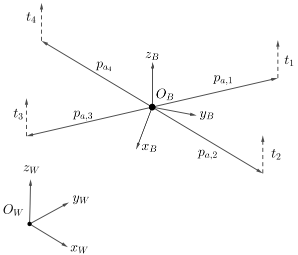
couple de trainée

matrice d'inertie

terme de conservation du moment angulaire

commande

Schéma d'un quadrotor



Plan

Formulation

Objectif

Trouver la commande $\mathfrak{u}$ telle que l'état du robot x minimise un coût C préalablement spécifié[1].

Notations

- $x(t)$ l'état du système
- $C(x(t), \mathfrak{u}(t) \mapsto C(t)$ la fonction de coût
- $f(x(t), \mathfrak{u}(t)) \mapsto \dot{x}$ la dynamique de l'état
- x_0, t_0 l'état et l'instant initiaux

Problème de contrôle optimal

$$\mathfrak{u}^* = \underset{\mathfrak{u} \in \mathcal{F}(\mathcal{T}, \mathcal{U})}{\arg\min} \int_{t_0}^{t_f} C(x(t), \mathfrak{u}(t)) dt$$

$$x(t_0) = x_0$$

$$\dot{x}(t) = f(x(t), \mathfrak{u}(t))$$

[1] Betts, Advances in Design and Control, 2010

Formulation

Objectif

Trouver la commande $\mathfrak{u}$ telle que l'état du robot x minimise un coût C préalablement spécifié[1].

Notations

- $x(t)$ l'état du système
- $C(x(t), \mathfrak{u}(t) \mapsto C(t)$ la fonction de coût
- $f(x(t), \mathfrak{u}(t)) \mapsto \dot{x}$ la dynamique de l'état
- x_0, t_0 l'état et l'instant initiaux

Problème de contrôle optimal

Commande Optimale

$$\mathfrak{u}^* = \underset{\mathfrak{u} \in \mathcal{F}(\mathcal{T}, \mathcal{U})}{\arg\min} \int_{t_0}^{t_f} C(x(t), \mathfrak{u}(t)) dt$$

$$x(t_0) = x_0$$

$$\dot{x}(t) = f(x(t), \mathfrak{u}(t))$$

[1] Betts, Advances in Design and Control, 2010

Formulation

Objectif

Trouver la commande $\mathfrak{u}$ telle que l'état du robot x minimise un coût C préalablement spécifié[1].

Notations

- $x(t)$ l'état du système
- $C(x(t), \mathfrak{u}(t) \mapsto C(t)$ la fonction de coût
- $f(x(t), \mathfrak{u}(t)) \mapsto \dot{x}$ la dynamique de l'état
- x_0, t_0 l'état et l'instant initiaux

Problème de contrôle optimal

Commande Optimale

$$\mathfrak{u}^* = \underset{\mathfrak{u} \in \mathcal{F}(\mathcal{T}, \mathcal{U})}{\arg\min} \int_{t_0}^{t_f} C(x(t), \mathfrak{u}(t))dt$$

Coût de l'état courant

$$x(t_0) = x_0$$

$$\dot{x}(t) = f(x(t), \mathfrak{u}(t))$$

[1] Betts, Advances in Design and Control, 2010

Formulation

Objectif

Trouver la commande $\mathfrak{u}$ telle que l'état du robot x minimise un coût C préalablement spécifié[1].

Notations

- $x(t)$ l'état du système
- $C(x(t), \mathfrak{u}(t) \mapsto C(t)$ la fonction de coût
- $f(x(t), \mathfrak{u}(t)) \mapsto \dot{x}$ la dynamique de l'état
- x_0, t_0 l'état et l'instant initiaux

Problème de contrôle optimal

Commande Optimale

$$\mathfrak{u}^* = \underset{\mathfrak{u} \in \mathcal{F}(\mathcal{T}, \mathcal{U})}{\arg\min} \int_{t_0}^{t_f} C(x(t), \mathfrak{u}(t))dt$$

$$x(t_0) = x_0$$

$$\dot{x}(t) = f(x(t), \mathfrak{u}(t))$$

Coût de l'état courant

Accumulation du coût sur la durée de la trajectoire

[1] Betts, Advances in Design and Control, 2010

Formulation

Objectif

Trouver la commande u telle que l'état du robot x minimise un coût C préalablement spécifié[1].

Notations

- $x(t)$ l'état du système
- $C(x(t), u(t)) \mapsto C(t)$ la fonction de coût
- $f(x(t), u(t)) \mapsto \dot{x}$ la dynamique de l'état
- x_0, t_0 l'état et l'instant initiaux

Problème de contrôle optimal

Commande Optimale

Espace des commandes

$$u^* = \underset{u \in \mathcal{F}(\mathcal{T}, \mathcal{U})}{\arg\min} \int_{t_0}^{t_f} C(x(t), u(t))dt$$

Coût de l'état courant

$$x(t_0) = x_0$$

$$\dot{x}(t) = f(x(t), u(t))$$

Accumulation du coût sur la durée de la trajectoire

[1] Betts, Advances in Design and Control, 2010

Formulation

Objectif

Trouver la commande $\mathfrak{u}$ telle que l'état du robot x minimise un coût C préalablement spécifié[1].

Notations

- $x(t)$ l'état du système
- $C(x(t), \mathfrak{u}(t)) \mapsto C(t)$ la fonction de coût
- $f(x(t), \mathfrak{u}(t)) \mapsto \dot{x}$ la dynamique de l'état
- x_0, t_0 l'état et l'instant initiaux

Problème de contrôle optimal

Commande Optimale

Espace des commandes

$$\mathfrak{u}^* = \underset{\mathfrak{u} \in \mathcal{F}(\mathcal{T},\mathcal{U})}{\arg\min} \int_{t_0}^{t_f} C(x(t), \mathfrak{u}(t))dt$$

Coût de l'état courant

Condition initiale $x(t_0) = x_0$

Accumulation du coût sur la durée de la trajectoire

$$\dot{x}(t) = f(x(t), \mathfrak{u}(t))$$

[1] Betts, Advances in Design and Control, 2010

Formulation

Objectif

Trouver la commande $\mathfrak{u}$ telle que l'état du robot x minimise un coût C préalablement spécifié[1].

Notations

- $x(t)$ l'état du système
- $C(x(t), \mathfrak{u}(t)) \mapsto C(t)$ la fonction de coût
- $f(x(t), \mathfrak{u}(t)) \mapsto \dot{x}$ la dynamique de l'état
- x_0, t_0 l'état et l'instant initiaux

Problème de contrôle optimal

Commande Optimale

Espace des commandes

$$\mathfrak{u}^* = \underset{\mathfrak{u} \in \mathcal{F}(\mathcal{T}, \mathcal{U})}{\arg\min} \int_{t_0}^{t_f} C(x(t), \mathfrak{u}(t))dt$$

Coût de l'état courant

Accumulation du coût sur la durée de la trajectoire

Condition initiale $\quad x(t_0) = x_0$

Dynamique de l'état $\quad \dot{x}(t) = f(x(t), \mathfrak{u}(t))$

[1] Betts, Advances in Design and Control, 2010

Formulation modifiée

Résolution du problème d'optimisation non linéaire sous contraintes suivant:

$$
\begin{aligned}
(u^*, x^*) = \underset{u \in \mathcal{U}^k,\; x \in \mathcal{X}^k,}{\operatorname{argmin}} \quad & \sum_{i=0}^{k+1} C(x_i, u_i) \\
x_0 &= x(t_0), \\
\forall i &\in [0, k], \\
x_{i+1} &= f(x_i, u_i, dt)
\end{aligned}
$$

Via une formulation symbolique des équations (CasADi [1]) et l'utilisation du solveur utilisant la méthode du point intérieur IPOpt[2].

[1] Andersson et al, Mathematical Programming Computation 11.1, 2019

[2] Wächter et Biegler, Mathematical Programming 106.1, 2006

Formulation modifiée

Résolution du problème d'optimisation non linéaire sous contraintes suivant:

$$
\begin{aligned}
(u^*, x^*) = \ & \underset{u \in \mathcal{U}^k,\, x \in \mathcal{X}^k,}{\operatorname{argmin}} \sum_{i=0}^{k+1} C(x_i, u_i) \\
& x_0 = x(t_0), \\
& \forall i \in [0, k], \\
& x_{i+1} = f(x_i, u_i, dt)
\end{aligned}
$$

Séquence de commandes optimales et d'états correspondants

Via une formulation symbolique des équations (CasADi [1]) et l'utilisation du solveur utilisant la méthode du point intérieur IPOpt[2].

[1] Andersson et al, Mathematical Programming Computation 11.1, 2019
[2] Wächter et Biegler, Mathematical Programming 106.1, 2006

Formulation modifiée

Résolution du problème d'optimisation non linéaire sous contraintes suivant:

$$
\begin{aligned}
(u^*, x^*) = \underset{u \in \mathcal{U}^k,\ x \in \mathcal{X}^k,}{\operatorname{argmin}} \quad & \sum_{i=0}^{k+1} C(x_i, u_i) \\
x_0 &= x(t_0), \\
\forall i &\in [0, k], \\
x_{i+1} &= f(x_i, u_i, dt)
\end{aligned}
$$

Séquence de commandes optimales et d'états correspondants

Coût de l'étape i

Via une formulation symbolique des équations (CasADi [1]) et l'utilisation du solveur utilisant la méthode du point intérieur IPOpt[2].

[1] Andersson et al, Mathematical Programming Computation 11.1, 2019

[2] Wächter et Biegler, Mathematical Programming 106.1, 2006

Formulation modifiée

Résolution du problème d'optimisation non linéaire sous contraintes suivant:

$$(u^*, x^*) = \underset{\substack{u \in \mathcal{U}^k,\ x \in \mathcal{X}^k, \\ x_0 = x(t_0), \\ \forall i \in [0, k], \\ x_{i+1} = f(x_i, u_i, dt)}}{\operatorname{argmin}} \sum_{i=0}^{k+1} C(x_i, u_i)$$

Séquence de commandes optimales et d'états correspondants

Coût de l'étape i

Accumulation durant la trajectoire (discrète)

Via une formulation symbolique des équations (CasADi [1]) et l'utilisation du solveur utilisant la méthode du point intérieur IPOpt[2].

---

[1] Andersson et al, Mathematical Programming Computation 11.1, 2019
[2] Wächter et Biegler, Mathematical Programming 106.1, 2006

Formulation modifiée

Résolution du problème d'optimisation non linéaire sous contraintes suivant:

Espace de recherche
(dimension finie)

$$
\begin{aligned}
(u^*, x^*) = \quad & \underset{u \in \mathcal{U}^k,\, x \in \mathcal{X}^k,}{\operatorname{argmin}} \quad \sum_{i=0}^{k+1} C(x_i, u_i) \\
& x_0 = x(t_0), \\
& \forall i \in [0, k], \\
& x_{i+1} = f(x_i, u_i, dt)
\end{aligned}
$$

Séquence de
commandes
optimales et d'états
correspondants

Coût de l'étape i

Accumulation
durant la
trajectoire
(discrète)

Via une formulation symbolique des équations (CasADi [1]) et l'utilisation du solveur utilisant la méthode du point intérieur IPOpt[2].

[1] Andersson et al, Mathematical Programming Computation 11.1, 2019

[2] Wächter et Biegler, Mathematical Programming 106.1, 2006

Formulation modifiée

Résolution du problème d'optimisation non linéaire sous contraintes suivant:

Espace de recherche (dimension finie)

$$(u^*, x^*) = \underset{u \in \mathcal{U}^k,\ x \in \mathcal{X}^k,}{\operatorname{arg\,min}} \sum_{i=0}^{k+1} C(x_i, u_i)$$

Séquence de commandes optimales et d'états correspondants

Coût de l'étape i

État Initial $x_0 = x(t_0),$

$$\forall i \in [0, k],$$

$$x_{i+1} = f(x_i, u_i, dt)$$

Accumulation durant la trajectoire (discrète)

Via une formulation symbolique des équations (CasADi [1]) et l'utilisation du solveur utilisant la méthode du point intérieur IPOpt[2].

---

[1] Andersson et al, Mathematical Programming Computation 11.1, 2019

[2] Wächter et Biegler, Mathematical Programming 106.1, 2006

Formulation modifiée

Résolution du problème d'optimisation non linéaire sous contraintes suivant:

Espace de recherche (dimension finie)

$$
\begin{aligned}
(u^*, x^*) = \underset{u \in \mathcal{U}^k,\ x \in \mathcal{X}^k,}{\operatorname{argmin}} \quad & \sum_{i=0}^{k+1} C(x_i, u_i) \\
& x_0 = x(t_0), \\
& \forall i \in [0, k], \\
& x_{i+1} = f(x_i, u_i, dt)
\end{aligned}
$$

Séquence de commandes optimales et d'états correspondants

État Initial

Contrainte de dynamique

Coût de l'étape i

Accumulation durant la trajectoire (discrète)

Via une formulation symbolique des équations (CasADi [1]) et l'utilisation du solveur utilisant la méthode du point intérieur IPOpt[2].

[1] Andersson et al, Mathematical Programming Computation 11.1, 2019

[2] Wächter et Biegler, Mathematical Programming 106.1, 2006

Principe et apport de la commande prédictive

Principe de la commande prédictive[1]

- Calculer une “bonne” trajectoire en boucle ouverte sur une durée finie
- Exécuter la première commande
- Planifier à nouveau à partir de l’état actuel

Avantages

- Referme la boucle en utilisant un retour d’état à chaque instant
- Tire parti des capacités de planification et du modèle utilisé
- Agnostique vis à vis de l’algorithme de calcul de trajectoire en boucle ouverte (ici résolution du problème de contrôle optimal)

[1] Rawlings, Mayne, Diehl, Nob Hill Publishing Madison, 2017

Suivi de trajectoire avec intégration du modèle appris des perturbations au contrôle optimal

Simulateur

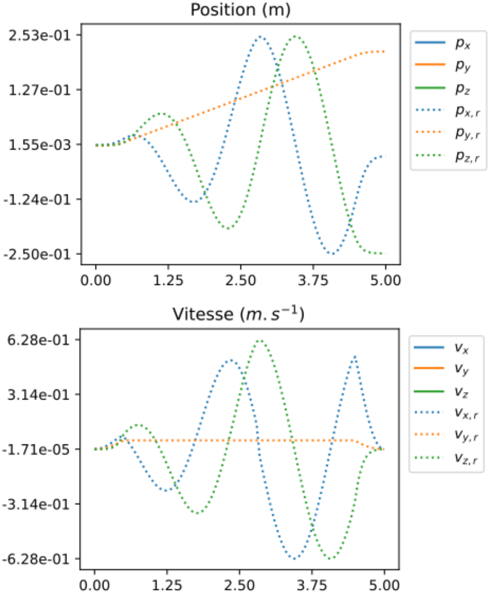
- Dynamique du drone
- Modèle GP appris des perturbations statiques

Contrôle optimal

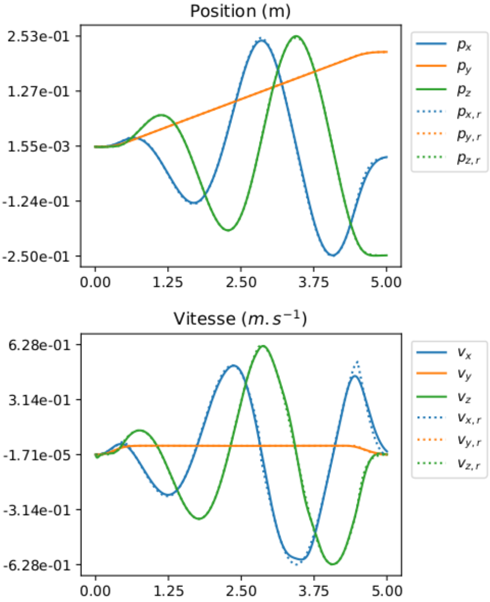
- Dynamique du drone
- Modèle GP appris des perturbations statiques

- Tâche de suivi de trajectoire
- Proximité des parois croissante (zone instable)

Suivi de trajectoire avec intégration du modèle appris des perturbations au contrôle optimal (Référence)



Suivi de trajectoire avec intégration du modèle appris des perturbations au contrôle optimal (Trajectoire)



Suivi de trajectoire SANS intégration du modèle appris des perturbations au contrôle optimal

Simulateur

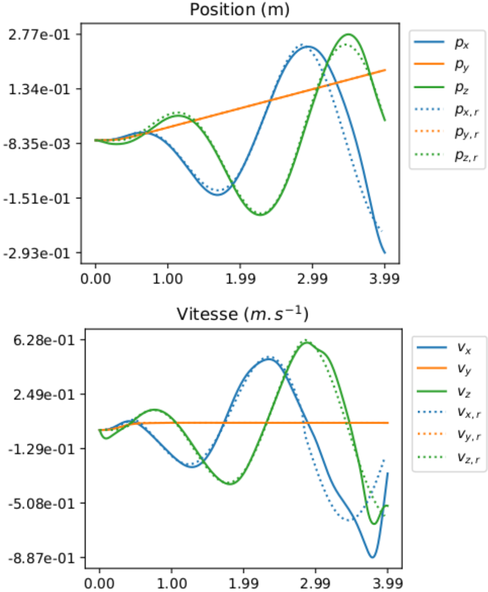
- Dynamique du drone
- Modèle GP appris des perturbations statiques

Contrôle optimal

- Dynamique du drone
-

- Tâche de suivi de trajectoire
- Proximité des parois croissante (zone instable)

Suivi de trajectoire SANS intégration du modèle appris des perturbations au contrôle optimal



Bilan commande prédictive basée contrôle optimal (MPC)

Avantages

- Intégration générique d'éléments à la dynamique
- Compatibilité avec des solveurs externes et bien établis
- Capable de prévoir les perturbations pour voler proche des parois malgré l'aspiration

Inconvénients

- Coût en calcul important pour la résolution de nombreux problèmes de contrôle optimal (plusieurs dizaines de fois le temps réel)
- Assez dépendant de la qualité du modèle
- Nécessité d'adapter les paramètres au robot (pas de temps, horizon)

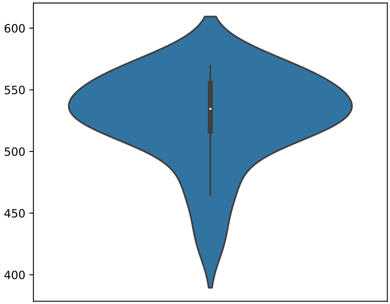
Plan

Pourquoi apprendre ?

- MPC: trop long
- Capacité de calcul faible
- Peu d'espoir d'améliorer assez le MPC pour tourner en temps réel

- Un NN à 200Hz tourne sur le Crazyflie
- Approximation du MPC par un NN appris (imitation, supervisé)

Temps de calcul (s) pour 1s de trajectoire par le MPC, distribution pour 20 trajectoires



État de l'art

Apprentissage par imitation

- À l'aide de démonstrations d'un pilote d'hélicoptère pour des acrobaties[1]

Démonstration guidant l'apprentissage

- Démonstrations choisies proches de ce que peut faire le contrôleur en cours d'apprentissage[2]

Démonstration pour initialiser un contrôleur

- Imitation préalable à un apprentissage par renforcement[3]

---

[1] Abbeel, Coates et Ng, The International Journal of Robotics Research 29.13, 2010

[2] Levine et Koltun, ICML, 2013

[3] Lin et al, IROS, 2019

Plan

Formulation

Objectifs

- Trouver les paramètres θ d'un contrôleur $\mathcal{C}_\theta$ qui imite au mieux un contrôleur expert $\mathcal{C}_E$

Équation

$$\theta^* = \arg\min_{\theta} \mathop{\mathbb{E}}_{x \sim P_\theta(x)} \left[||\mathcal{C}_E(x) - \mathcal{C}_\theta(x)||^2 \right]$$

Formulation

Objectifs

- Trouver les paramètres θ d'un contrôleur $\mathcal{C}_\theta$ qui imite au mieux un contrôleur expert $\mathcal{C}_E$

Équation

$$\underbrace{\theta^*}_{\text{Paramètres optimaux du contrôleur}} = \arg\min_{\theta} \mathop{\mathbb{E}}_{x \sim P_\theta(x)} \left[||\mathcal{C}_E(x) - \mathcal{C}_\theta(x)||^2 \right]$$

Formulation

Objectifs

- Trouver les paramètres θ d'un contrôleur $\mathcal{C}_\theta$ qui imite au mieux un contrôleur expert $\mathcal{C}_E$

Équation

Commande experte

Commande apprise

$$\theta^* = \arg\min_{\theta} \mathop{\mathbb{E}}_{x \sim P_\theta(x)} \left[||\mathcal{C}_E(x) - \mathcal{C}_\theta(x)||^2 \right]$$

Paramètres optimaux du contrôleur

Formulation

Objectifs

- Trouver les paramètres θ d'un contrôleur $\mathcal{C}_\theta$ qui imite au mieux un contrôleur expert $\mathcal{C}_E$

Équation

Accumulation sur l'ensemble des états

Commande experte

Commande apprise

$$\theta^* = \arg\min_{\theta} \mathop{\mathbb{E}}_{x\sim P_\theta(x)} \left[||\mathcal{C}_E(x) - \mathcal{C}_\theta(x)||^2 \right]$$

Paramètres optimaux du contrôleur

Formulation

Objectifs

- Trouver les paramètres θ d'un contrôleur $\mathcal{C}_\theta$ qui imite au mieux un contrôleur expert $\mathcal{C}_E$

Équation

Accumulation sur l'ensemble des états

Commande experte

Commande apprise

$$\theta^* = \arg\min_\theta \mathop{\mathbb{E}}_{x \sim P_\theta(x)} \left[||\mathcal{C}_E(x) - \mathcal{C}_\theta(x)||^2 \right]$$

Paramètres optimaux du contrôleur

Distribution des états (dépendant de θ)

Résolution (DAGGER)

Principe de DAGGER[1]

- Approche la distribution des états rencontré par échantillonnage itéré
  - À partir d'un ensemble initialement vide
  - Générer une trajectoire avec $\mathcal{C}_{\theta_{\mathbf{0}}}$
  - Demander à l'expert les commandes optimales pour les états de la trajectoire générée
  - Mettre à jour θ pour imiter l'expert
  - Recommencer
- Fait coïncider la distribution des états rencontré avec ceux utilisés pour l'apprentissage
- Expert = contrôle optimal
- Ajout à l'ensemble d'apprentissage de *toute* la trajectoire calculée lors de la résolution du problème de contrôle optimal

[1] Ross, Gordon et Bagnell, Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, 2011

Illustration DAGGER

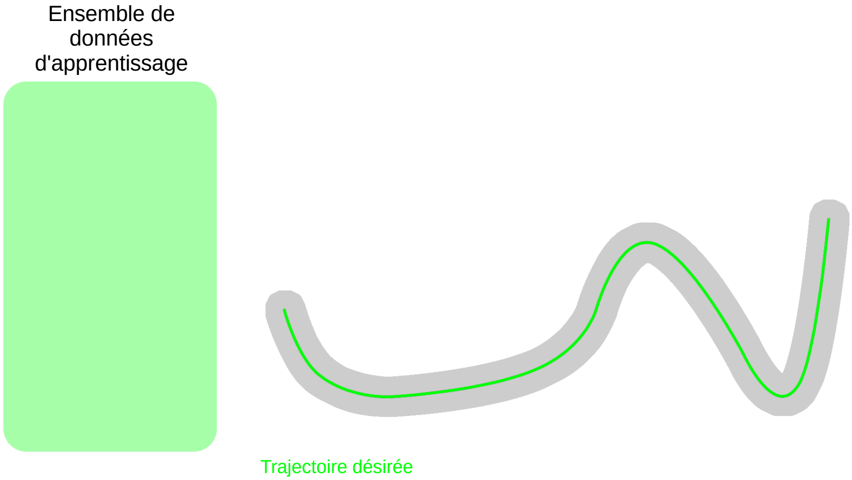


Illustration DAGGER

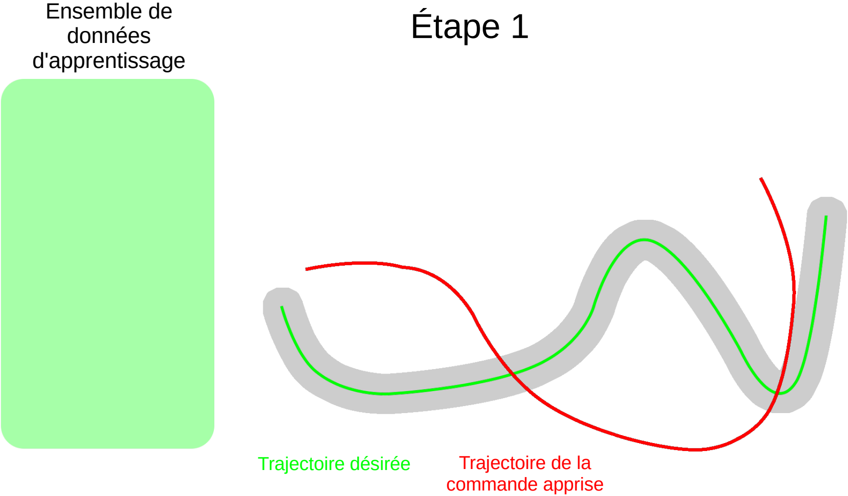


Illustration DAGGER

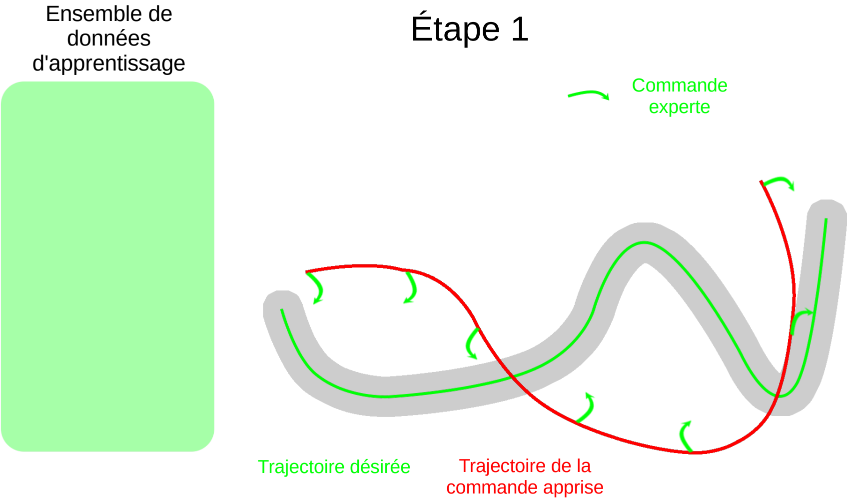


Illustration DAGGER

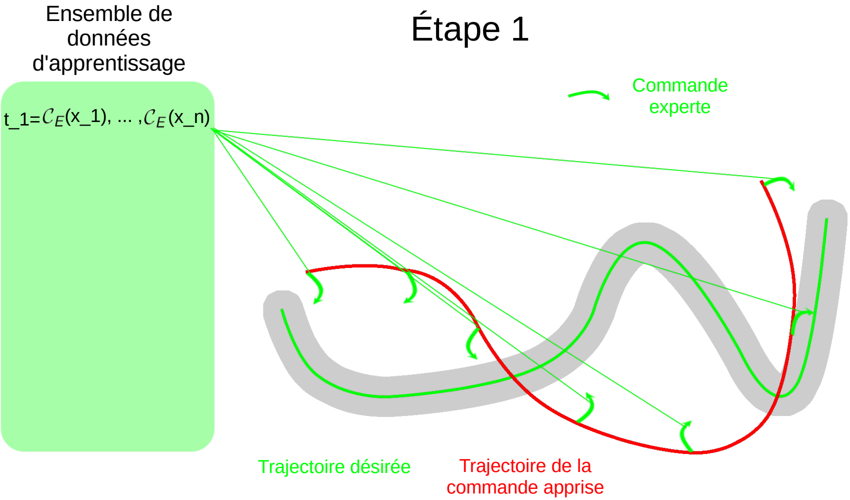


Illustration DAGGER

Ensemble de données d'apprentissage

t_1=$\mathcal{C}_E$(x_1), ... ,$\mathcal{C}_E$(x_n)

Étape i

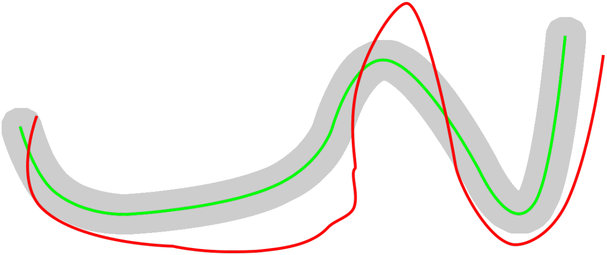


Illustration DAGGER

Ensemble de données d'apprentissage

t_1=$\mathcal{C}_E$(x_1), ... ,$\mathcal{C}_E$(x_n)

Étape i

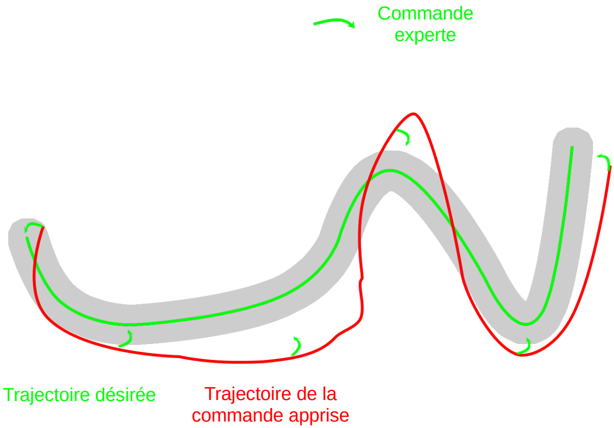


Illustration DAGGER

Ensemble de données d'apprentissage

t_1=$\mathcal{C}_E$(x_1), ... ,$\mathcal{C}_E$(x_n)

$\vdots$

t_i=$\mathcal{C}_E$(x_i1), ... ,$\mathcal{C}_E$(x_in)

Étape i

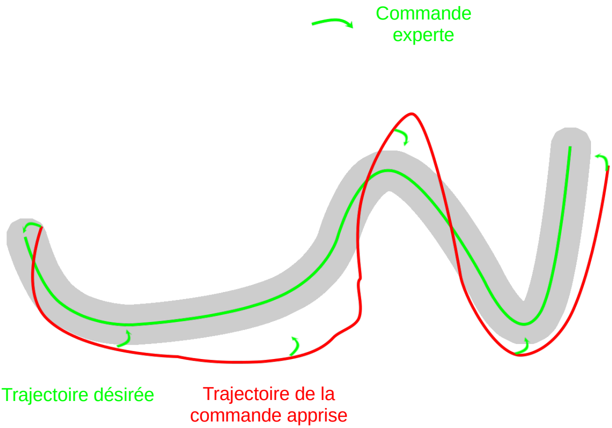


Illustration DAGGER

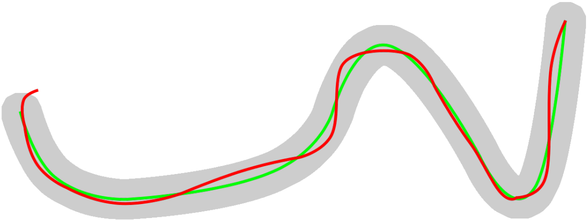
Ensemble de
données
d'apprentissage

t_1=$\mathcal{C}_E$(x_1), ... ,$\mathcal{C}_E$(x_n)

⋮

t_i=$\mathcal{C}_E$(x_i1), ... ,$\mathcal{C}_E$(x_in)

Étape j



Trajectoire désirée

Trajectoire de la
commande apprise

Illustration DAGGER

Ensemble de données d'apprentissage

t_1=$\mathcal{C}_E$(x_1), ... ,$\mathcal{C}_E$(x_n)

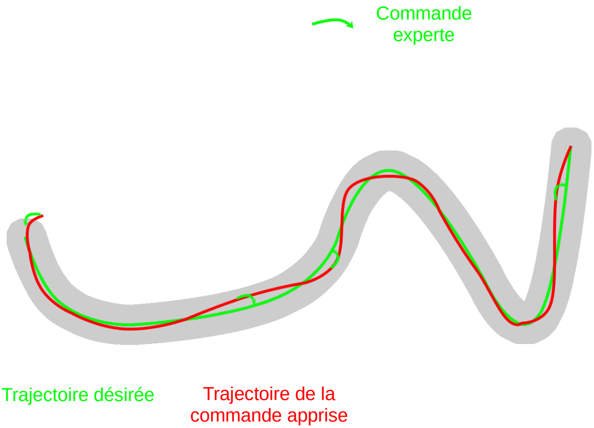
$\vdots$

t_i=$\mathcal{C}_E$(x_i1), ... ,$\mathcal{C}_E$(x_in)

$\vdots$

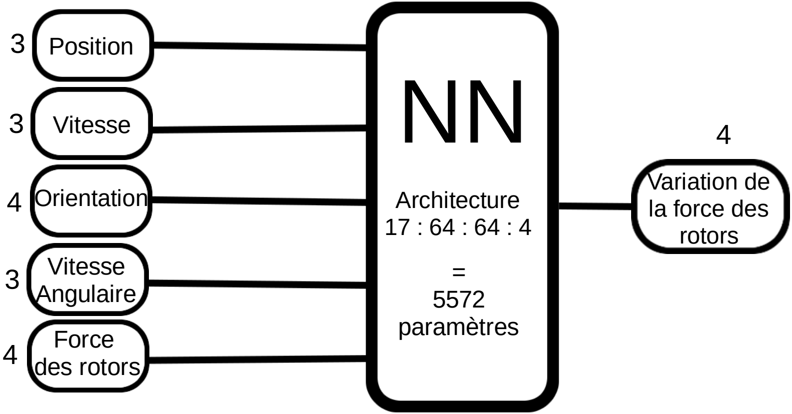
t_j=$\mathcal{C}_E$(x_j1), ... ,$\mathcal{C}_E$(x_jn)

Étape j



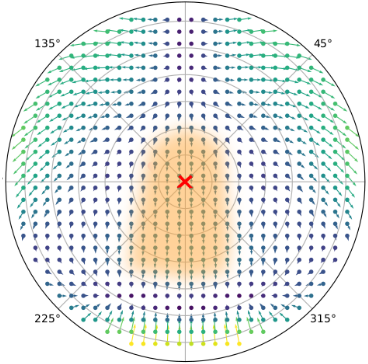
Plan

Architecture du contrôleur appris

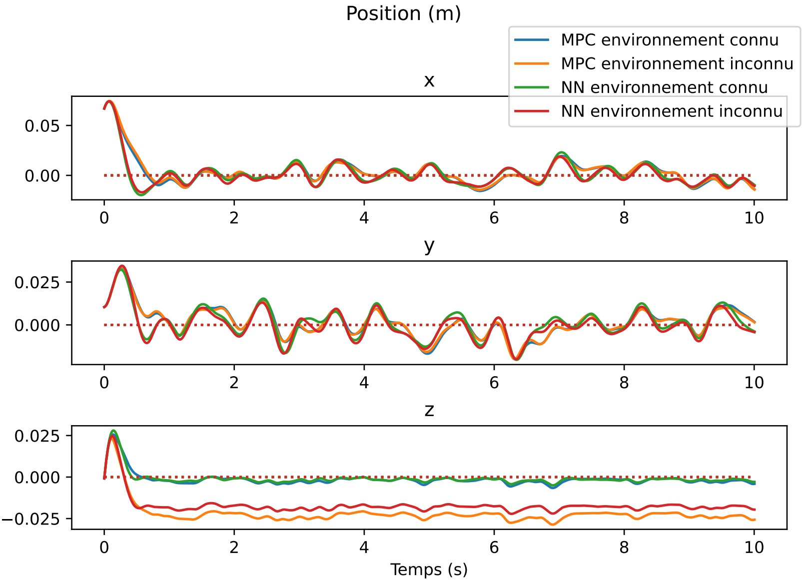


Tâche de stabilisation

- Amener le drone au centre du tuyau à l'horizontale
- État de départ perturbé
  - Pas au centre
  - Vitesse non nulle
  - Orientation non horizontale
- Perturbations plus faciles que pour le suivi

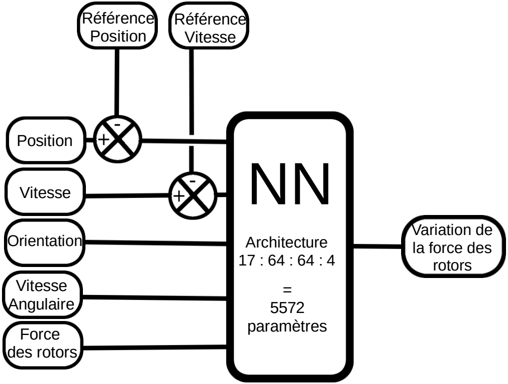


Comparaison des performances de stabilisation en fonction de la connaissance de l'environnement (tuyau 65cm)

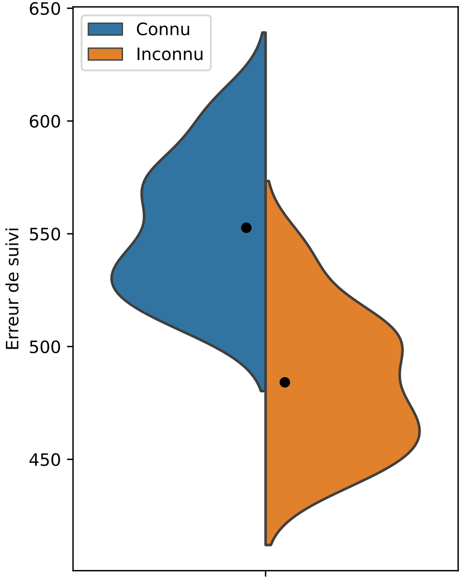


Suivi de trajectoire

- Référence en entrée: quantité de données pour apprendre trop importante
- Solution: différence entre l'état et la référence en entrée
- Exploitation du modèle des perturbations difficile

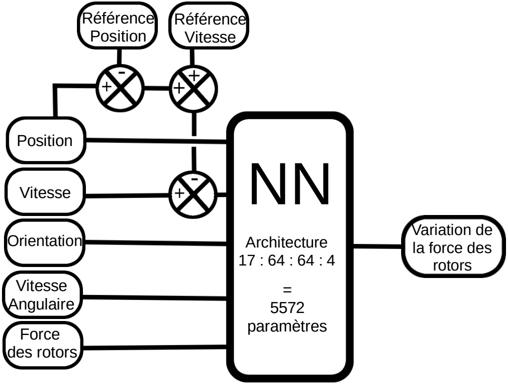


Limite: suivi de trajectoire (tuyau 65 cm simulé, 20 répliques)

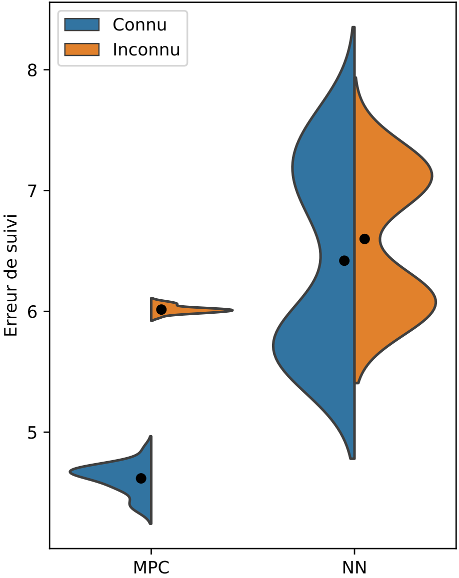


Adaptation

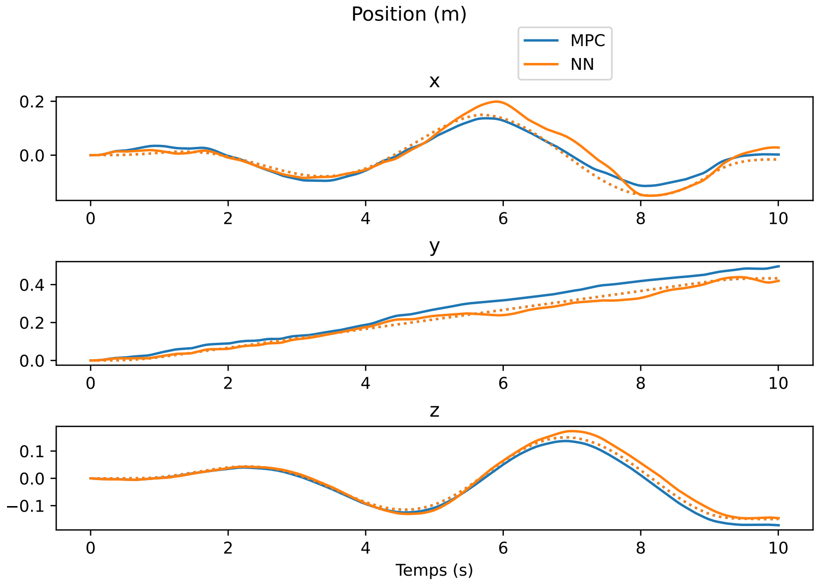
- Apprendre un contrôleur en vitesse (poids nul pour la position dans le coût MPC)
- Modifier la référence en vitesse pour asservir la position



Résultat adaptation: distribution de l'erreur de suivi (tuyau 65cm, 20 répliques)



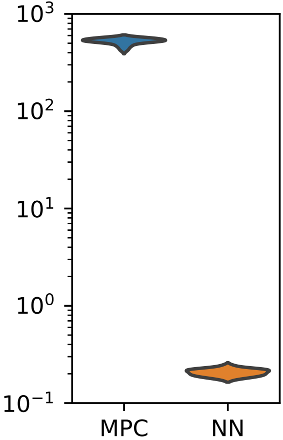
Suivi de trajectoire, comparaison au MPC (tuyau 65cm)



Conclusion

- Contrôleur appris capable de stabiliser et suivre des trajectoires
- Intégration du modèle de l'environnement possible et utile
- Temps de calcul bien plus faible avec le contrôleur NN appris (600Hz vs 0.3Hz max)

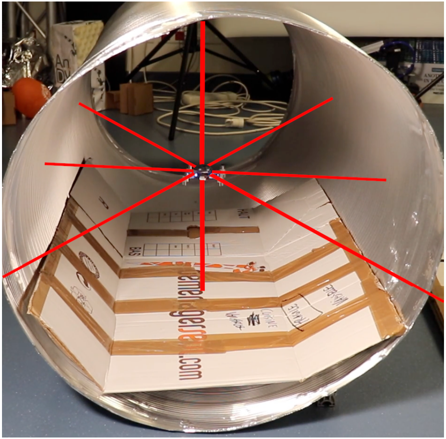
Distribution du temps de calcul de la commande pour 1s de trajectoire



Estimation d'état

- Estimer la position pour un contrôle précis
- Estimation de l'équation des droites portées par les murs : lacet et position longitudinale du drone
- Inversion de l'équation de la distance aux parois : position verticale et longitudinale

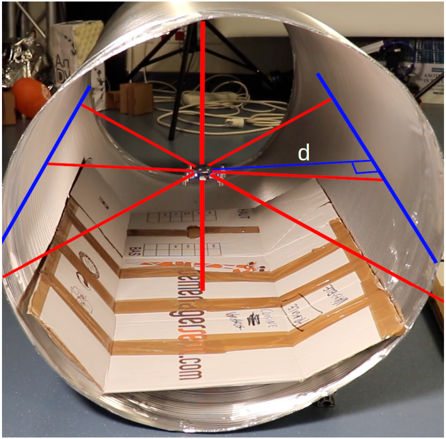
Localisation avec des télémètres



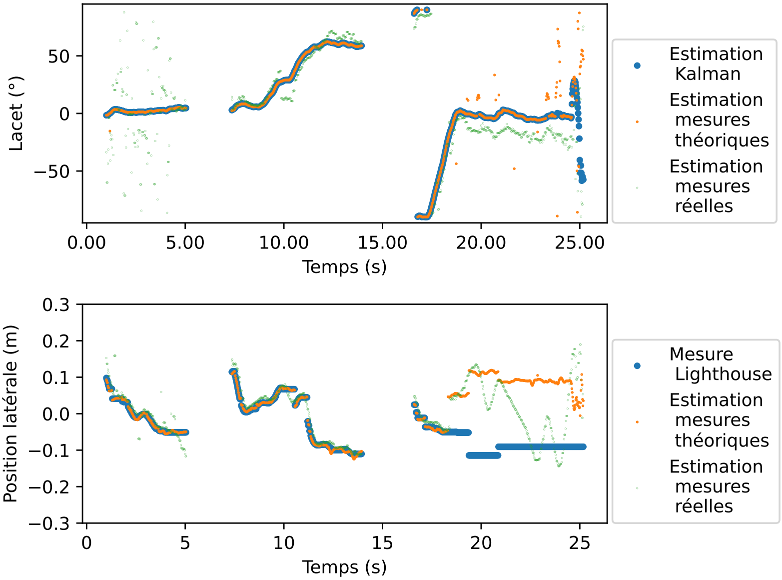
Estimation d'état

- Estimer la position pour un contrôle précis
- Estimation de l'équation des droites portées par les murs : lacet et position longitudinale du drone
- Inversion de l'équation de la distance aux parois : position verticale et longitudinale

Estimation avec équations de droite



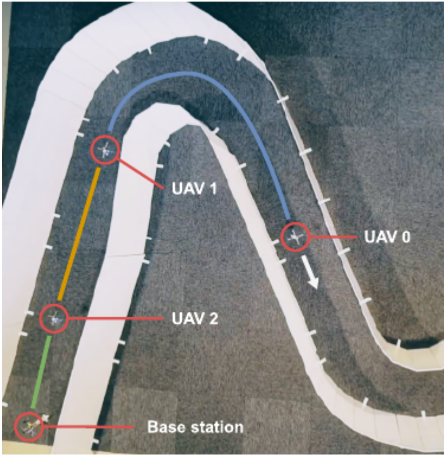
Estimation d'état régression linéaire données



Organisation d'une flotte de drones

Algorithme de positionnement d'une flotte dans un tunnel

- Ajustement automatique et décentralisé de la position pour le maintien d'un contact radio [1]
- En régulant la qualité de signal
- Uniquement pour des environnements en 1D
- Contribution: preuve de l'optimalité et de la correction de l'algorithme



[1] Laclau et al, Frontiers in Robotics and AI 8, 2021

Séquence vidéo de la flotte de drones

Vidéo du contrôle décentralisé d'une flotte de drones

Limites et extensions de mes travaux de thèse

Limites

- Estimation d'état validée en simulation mais mais à affiner pour les données réelles

- Mesure des perturbation statiques uniquement
- Biais dans les mesures (effets au bord, présence du dispositif)
- Uniquement réalisé pour un drone non incliné

Extensions

- Apprentissage d'un contrôleur directement à partir des mesures des capteur

- Utiliser un premier contrôleur avec modèle, récolter des données en vol libre, recommencer

Limites et extensions de mes travaux de thèse

Contrôleur appris par imitation

- Contrôleur appris par imitation pas encore embarqué
- Robustesse du contrôleur appris par imitation probablement limitée (évaluée uniquement avec le "vrai" modèle et en simulation)
- Un contrôleur similaire fait voler le Crazyflie (Corentin Bunel)

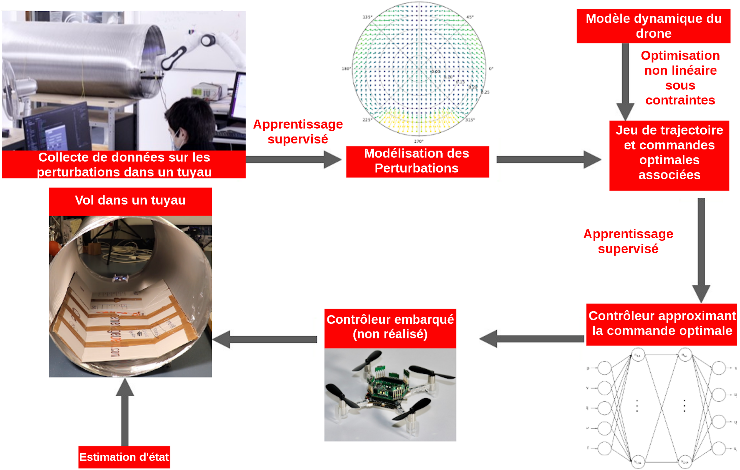
Autres approches indépendantes

- Architecture complètement actionnée[1] (non coaxiale) pour un meilleur rejet des perturbations du tuyau
- Apprentissage d'autre contrôleur par imitation: contrôle à plus haut niveau, apprentissage des gains d'un contrôleur hiérarchique

---

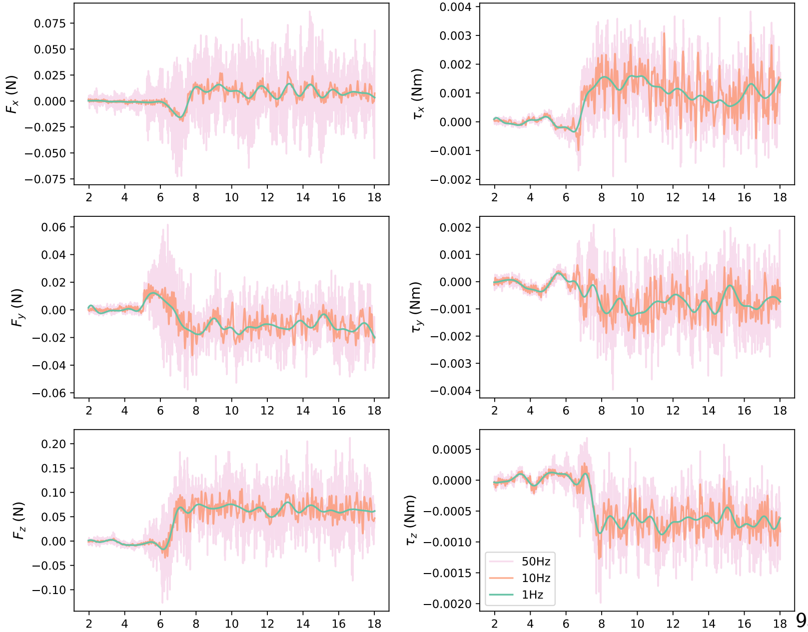
[1] Rashad et al, IEEE Robotics Automation Magazine 27.3, 2020

Questions ?


Collecte de données sur les perturbations dans un tuyau
Apprentissage supervisé
Modélisation des Perturbations
Modèle dynamique du drone
Optimisation non linéaire sous contraintes
Jeu de trajectoire et commandes optimales associées
Apprentissage supervisé
Contrôleur approximant la commande optimale
Contrôleur embarqué (non réalisé)
Vol dans un tuyau
Estimation d'état

Plan

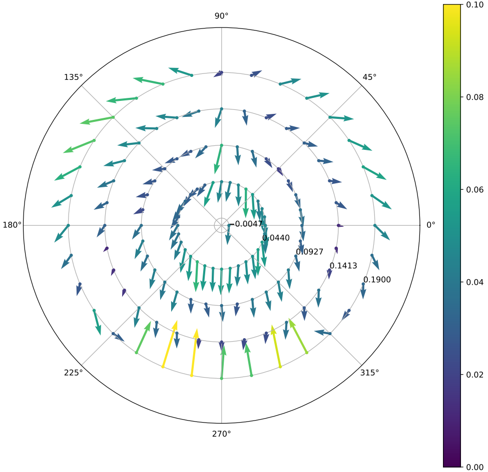
Composante dynamique des perturbations



Tuyau de 50cm de diamètre

1 flèche = 1 mesure (N)

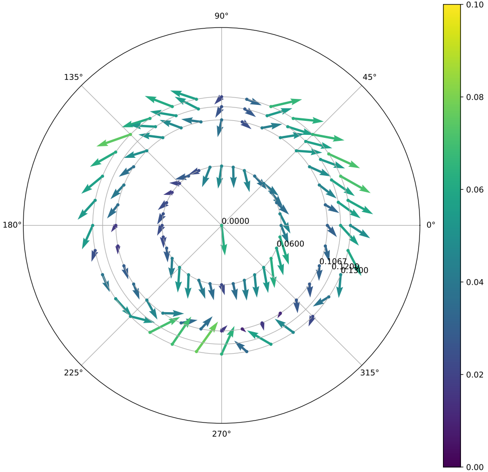
- Effet de sol plus important
- Aspiration par les parois similaire
- Forces similaires (hors effet de sol)



Tuyau de 40cm de diamètre

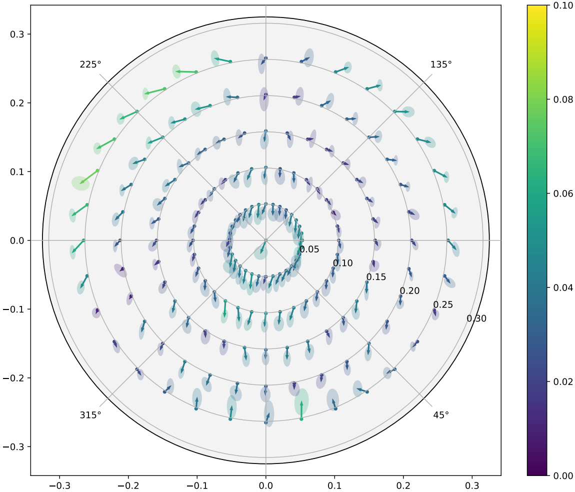
1 flèche = 1 mesure (N)

- Moins de mesures (peu de place)
- Schéma similaire
- Forces similaires



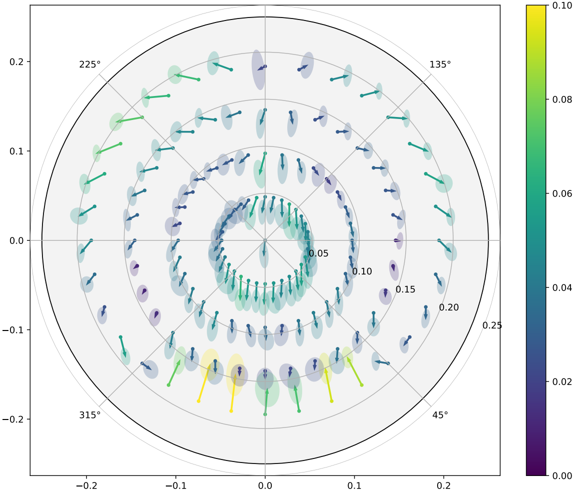
Tuyau de 65cm de diamètre avec covariance

- Ellipse = covariance de la perturbation durant la mesure
- Variance plus faible que les perturbations
- Plus de variance dans la zone "effet de sol"



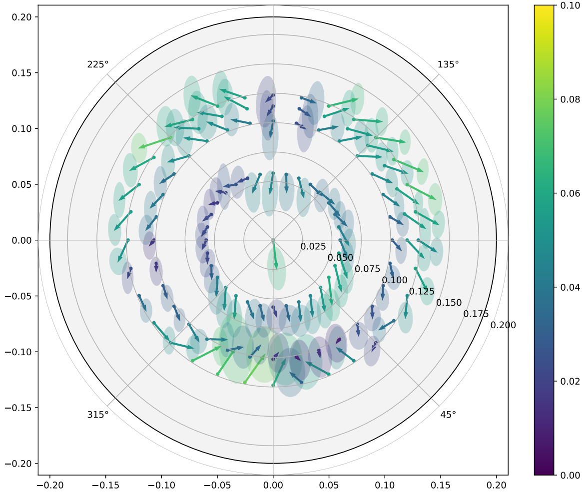
Tuyau de 50cm de diamètre avec covariance

- Ellipse = covariance de la perturbation durant la mesure
- Variance plus importante que dans le tuyau de 65cm
- Plus encore dans la zone "effet de sol"



Tuyau de 40cm de diamètre avec covariance

- Ellipse = covariance de la perturbation durant la mesure
- Variance très importante
- Plus encore dans la zone "effet de sol"



Exploitation des symétries

Coordonnées polaires

- Respect des symétries
- Non continue (angle)

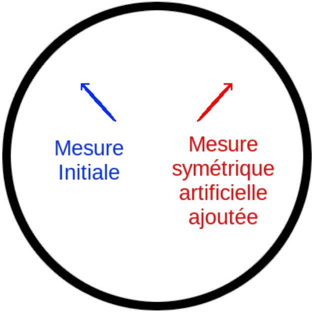
Coordonnées cartésiennes

- Ignore les symétries
- Continue

Coordonnées hybrides

- Coordonnées polaires modifiées
- Remplace l'angle par (sin,cos)
- Continue
- Coordonnés cartésiennes + rayon

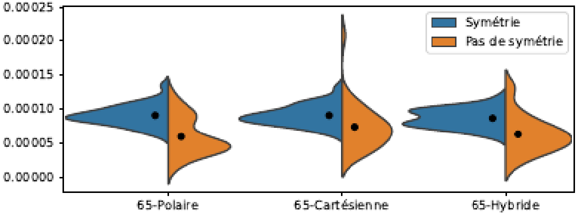
Enrichissement des données par symétrie

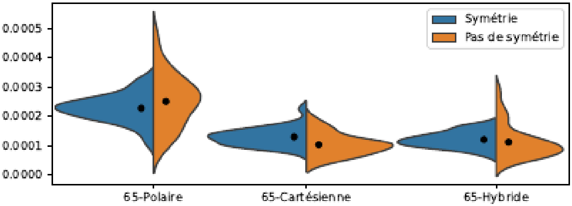


Choix des symétries

- Symétrie = erreur moins dispersée
- Représentation sans effet pour GP
- Représentation polaire moins bien pour NN
- Symétrie = erreur plus grande pour les GP (attendu)
- Choix de la représentation cartésienne et de l'enrichissement par symétrie

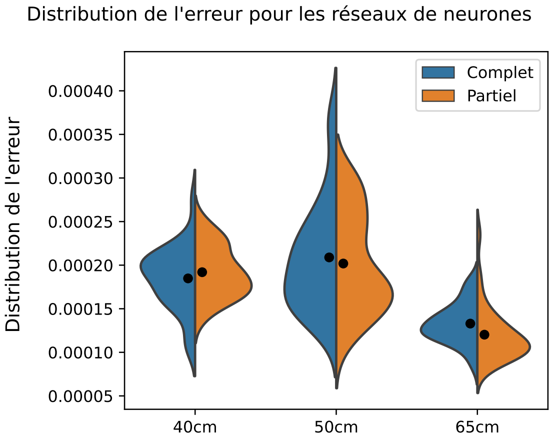
Erreur d'apprentissage (65cm)





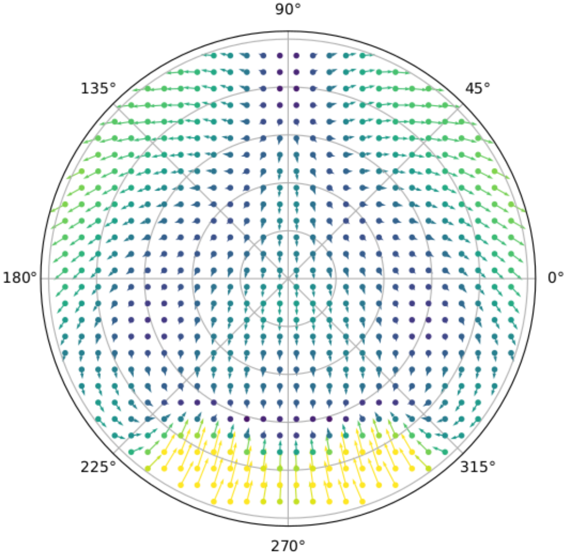
Importance du type de prédiction pour les réseaux de neurones

- Quel impact d'une prédiction de l'ensemble des perturbations $(F_{x,y,z},\ \tau_{x,y,z})$ par rapport à uniquement celles qui nous intéressent $(F_x,\ F_y)$?
- Partiel est un peu mieux

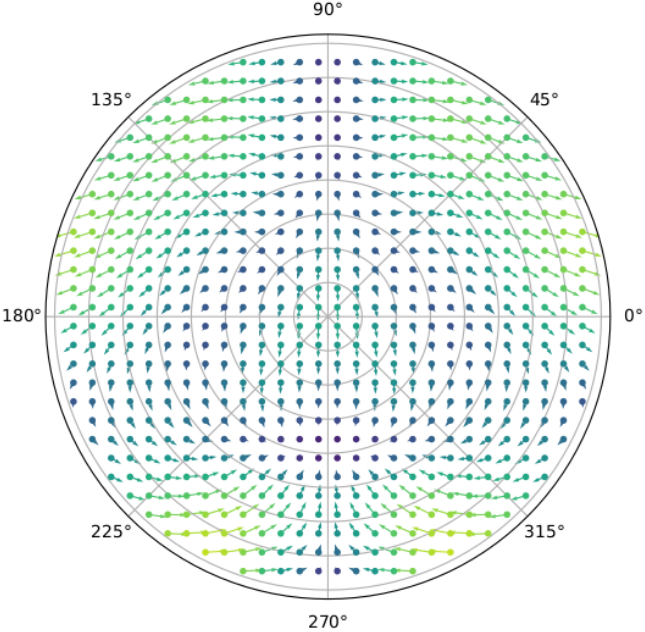


Résultat et analyse des perturbations (50cm GP)

- Disposition similaire des perturbations
- Effet de sol plus fort (déjà présent dans les mesure)
- Zone stable plus éloignée du sol



Interpolation - tuyau de 40cm



Plan

Approche pour la résolution

Approches principales:

dites *indirectes*:
Résolution de l'équation différentielle (détermination de conditions KKT sur u) puis optimisation dans l'espace des solutions (*optimize then discretize*)

dites *directes*:
Discrétisation de l'équation, restriction de x et u à un espace de dimension finie et optimisation sous contrainte dans cet espace (*discretize then optimize*)

Approche choisie:

- *directe*.
- u et x sont constants par morceaux: temps discrétisé.
- Formulation proche de la simulation
- Optimisation non linéaire sur un espace vectoriel.

Approche pour la résolution II

Single shooting:

- Trajectoire en un seul morceau
- $x(t)$ déduit récursivement $x(0)$, de $\mathfrak{u}(t)$ et $x(t-1)$
- Rend C très non linéaire par rapport à $\mathfrak{u}$

Multiple shooting:

- Trajectoire découpée en sous trajectoires
- $x(t)$ devient optimisable
- Introduction de contraintes forcer $x(t)$ à respecter la dynamique
- Découple partiellement les variables optimisées
- Approche choisie

Résultats boucle ouverte sans déviation au modèle utilisé

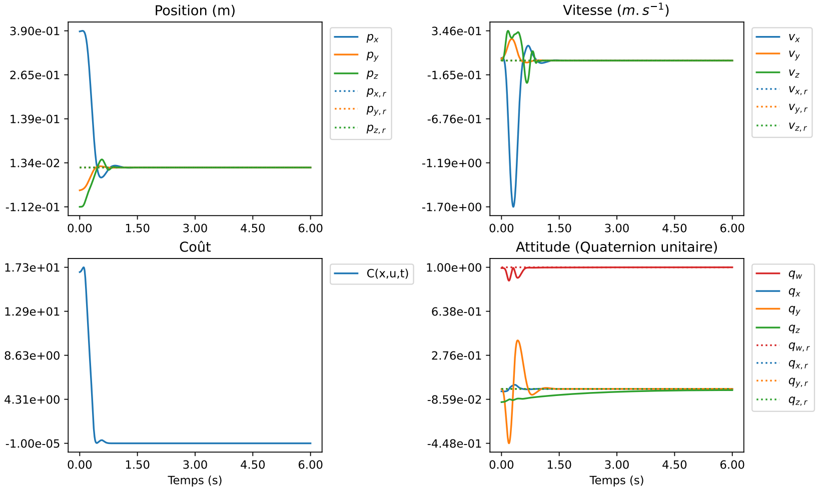


Figure: Tâche de stabilisation en simulation pour un quadrotor aux caractétistiques du Crazyflie par résolution du problème de contrôle optimal en l'absence de perturbations

Résultats boucle ouverte avec déviation au modèle utilisé

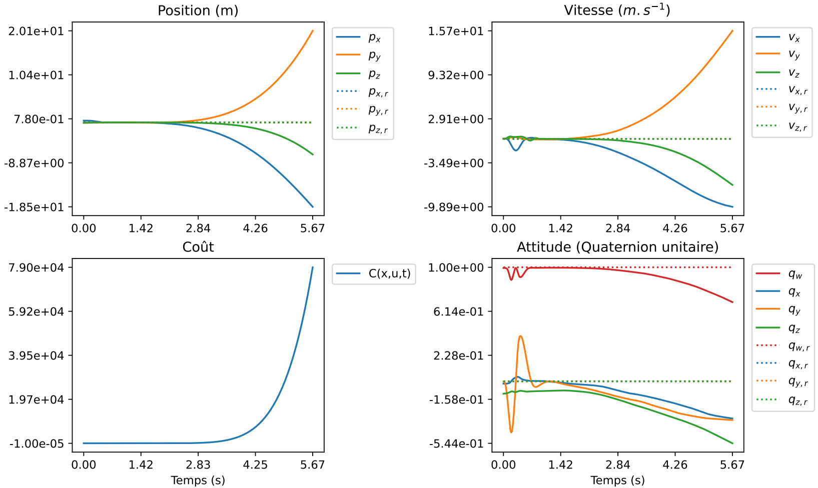
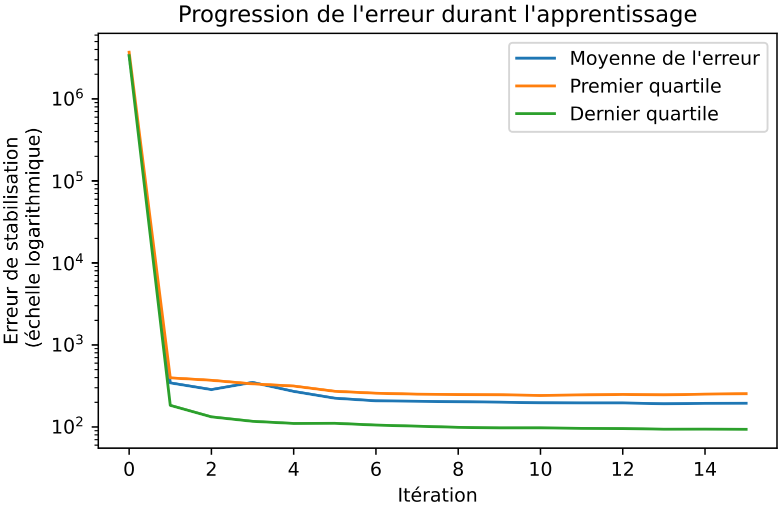


Figure: Tâche de stabilisation en simulation pour un quadrotor aux caractétistiques du Crazyflie par résolution du problème de contrôle optimal en présence de perturbations

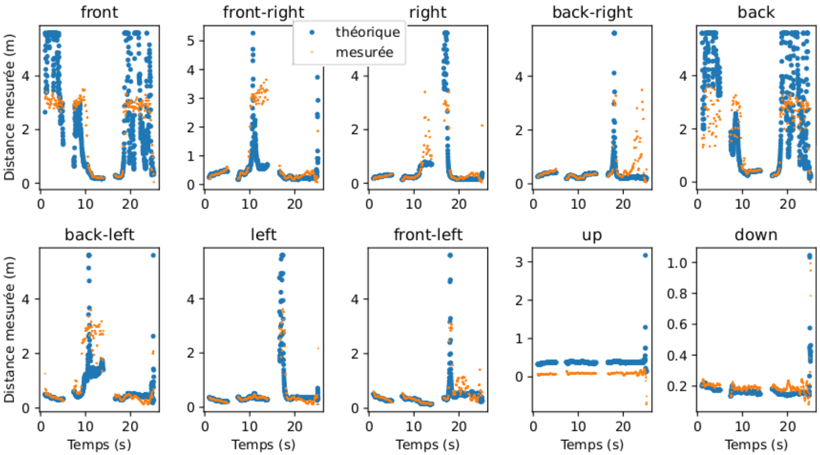
Plan

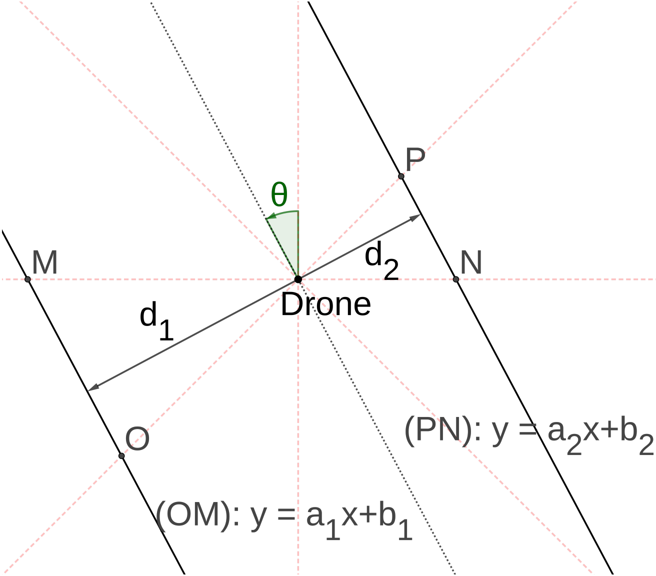
Courbe d'apprentissage



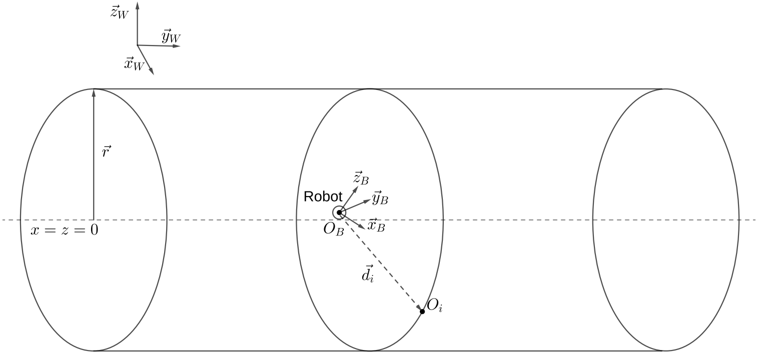
Plan

Mesures des télémètres

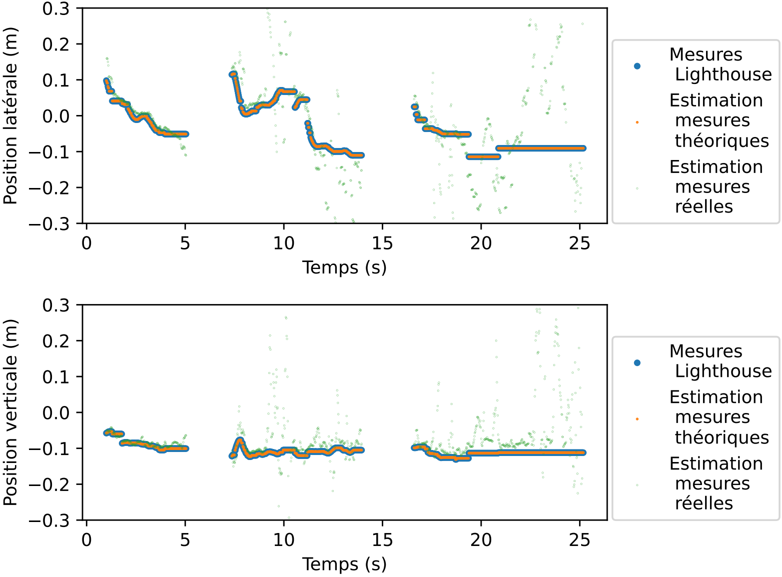



P
θ
M
d2
N
Drone
d1
O
(PN): y = a2x+b2
(OM): y = a1x+b1

Estimation d'état modèle inverse théorie



Estimation d'état modèle inverse données



Plan

Résultats en simulation