
RAPPORT TECHNIQUE

RAPPORT TECHNIQUE SUR LE COMMANDEUR
ET LA SIMULATION

TOULOUSE, 2015–2016

RÉDIGÉ PAR

ZHENGXUAN JIA

ISAE-SUPAERO

[\mathcal{PL}]

2016

SUPAERO

Table des matières

1	Conception simulation pas à pas	1
1.1	Conception du module calcul commande optimale (modification)	1
1.2	Conception du module simulateur (conception)	2
2	Réalisation du contrôleur	3
2.1	Choix de contrôleur	3
2.2	Contrôle à partir du clavier – transmission de consigne	3

Table des figures

1.1	Structure globale de conception	1
1.2	Structure globale de conception	2
2.1	Choix possibles comme contrôleur du drone	3
2.2	Chargeur de Xbox One	4
2.3	Site de SFML	4

1 Conception simulation pas à pas

Avec *ACADO* on peut déjà faire la simulation avec la commande optimale calculée. Maintenant, on veut faire séparément le calcul de commande et la simulation. Ainsi, on peut faire le calcul de commande et la simulation en ligne.

Evidemment, dans la conception, il y a principalement deux modules : calcul de la loi de commande optimale par *ACADO* et simulation avec la commande optimale calculée. La structure globale de cette conception est affichée sur la Figure 1.1. Au début, l'état du drone

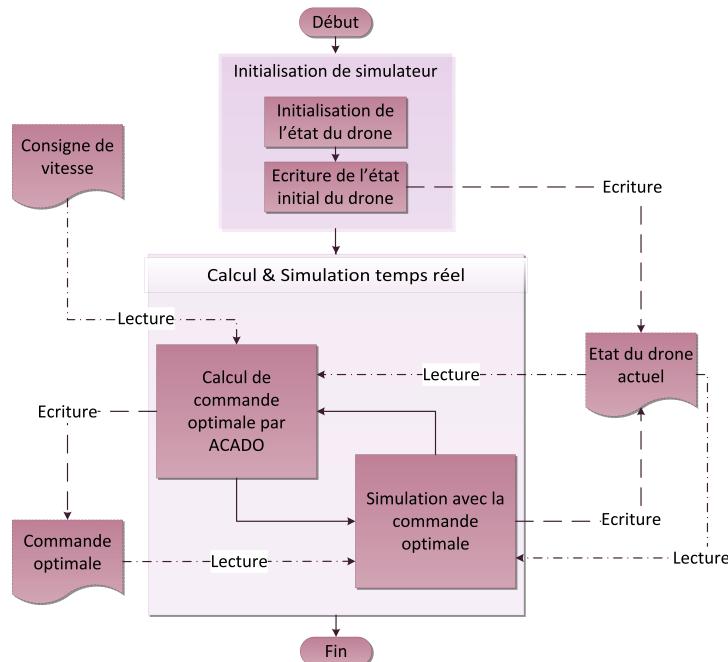


FIGURE 1.1 – Structure globale de conception

est initialisé. Ensuite, un processus alternatif du calcul de commande et de la simulation avec la commande optimale calculée appliquée à la dynamique du drone est activé. A chaque instant, le calculateur de commande optimale lit l'état actuel du drone et la consigne donnée par l'extérieur, et calcule la commande optimale. Cette commande optimale est ainsi enregistrée. Après le simulateur lit cette commande optimale et l'état actuel du drone, et avance l'état du drone un pas plus. Et puis, le calculateur recalcule la commande optimale, et ainsi de suite.

1.1 Conception du module calcul commande optimale (modification)

Ce module est réalisé à l'aide de *ACADO*. Il comprend principalement les étapes suivantes :

Etape 1 Définition complète du problème de contrôle optimal

- Définition de la fonction de coût
- Définition des équations différentielles de la dynamique du système
- Définition des contraintes

Etape 2 Configuration de *RealTimeAlgorithm* par le problème de contrôle optimal et le pas de calcul de *ACADO*

Etape 3 Configuration de contrôleur

Etape 4 Mise en oeuvre des conditions initiales/états initiaux

Etape 5 Calcul de commande optimal pour le pas suivant

Etape 6 Enregistrement de la commande optimale

Cette partie est réalisée basée sur les codes déjà donnés, en faisant des modifications nécessaires.

1.2 Conception du module simulateur (conception)

Pour le simulateur, on utilise dans un premier temps l'algorithme de RungeKutta d'ordre 4 (RK4) dont le principe est donné comme suivant. Considérons le problème

$$\dot{x} = f(t, x) \quad (1.1)$$

$$x(t_0) = x_0 \quad (1.2)$$

La méthode de RK4 donne la variable x au pas suivant par

$$x_{n+1} = x_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (1.3)$$

$$k_1 = f(t_n, x_n) \quad (1.4)$$

$$k_2 = f\left(t_n + \frac{h}{2}, x_n + \frac{h}{2}k_1\right) \quad (1.5)$$

$$k_3 = f\left(t_n + \frac{h}{2}, x_n + \frac{h}{2}k_2\right) \quad (1.6)$$

$$k_4 = f(t_n + h, x_n + hk_3) \quad (1.7)$$

Et le processus de ce module est donné sur la Figure 1.2. D'abord, le simulateur lit l'état actuel

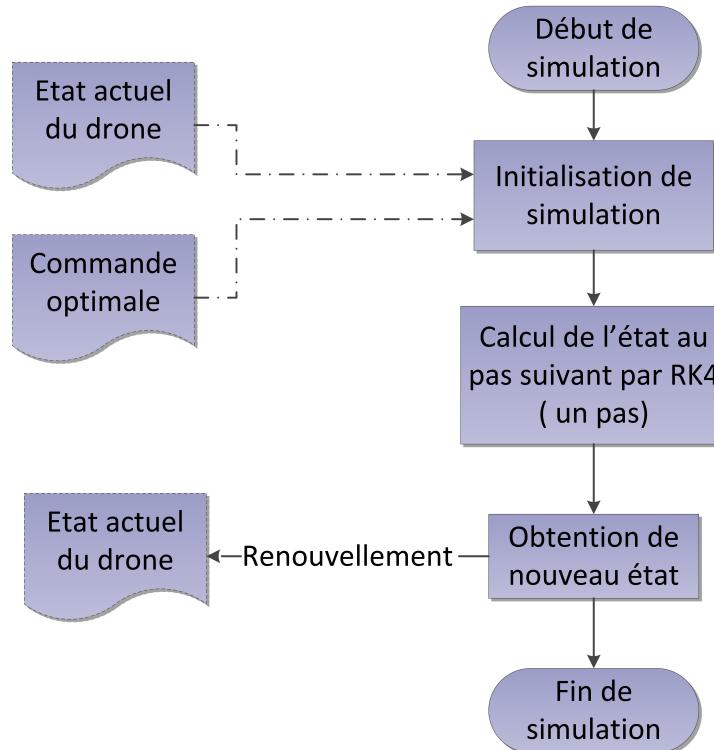


FIGURE 1.2 – Structure globale de conception

et la commande optimale. Ensuite, l'état au pas suivant est obtenu par le calcul de RK4. Et cet état calculé est renouvelé dans l'état actuel du drone.

La communication entre les deux modules sont réalisée de deux manières différentes. Soit on enregistre les données dans les fichiers, et la communication est basée sur ces fichiers ; soit on définit une classe et on crée un objet dont un attribut est utilisé pour l'enregistrement des données.

2 Réalisation du contrôleur

2.1 Choix de contrôleur

Le contrôleur d'un drone peut être les suivants :

- PC
- SmartPhone
- Joystick

Pour le contrôler **plus facilement** et avec **un coût plus bas**, on choisit de le contrôler avec un Joystick. Pour ce contrôleur, il y a déjà des [projets disponibles sur Internet](#). Dans ce projet, il y a déjà des application pour iOS, Android et Windows 8 avec les quadrotors de Parrot. Ce projet est aussi applicable pour contrôler un drone avec un Xbox Joystick comme montré sur la Figure 2.1 (a). Si on veut utiliser un Joystick sans fil, celui sur la Figure 2.1 (b). Ce Joystick



(a) Joystick Xbox

(b) Joystick Xbox One sans fil

FIGURE 2.1 – Choix possibles comme contrôleur du drone

qui marche aussi avec Windows 10 PCs et Tablettes utilise un chargeur comme montré sur la Figure. Cette chargeur peut être chargée à plein dans 4h. Tout ce qui est plus haut qui concerne l'utilisation d'un Joystick pour contrôler un drone est déjà testé par l'auteur du blog : [Flying a Drone with Windows 10 and an Xbox Controller](#)

2.2 Contrôle à partir du clavier – transmission de consigne

Même qu'il est possible de contrôler le drone depuis un Joystick, comme dans ce projet nous sommes demandés principalement de concentrer à la simulation au lieu du manœuvre d'un vrai drone, alors on adopte le contrôle en simulation depuis le clavier comme solution. Ainsi, on s'intéresse, vis-à-vis du contrôle, à la réalisation de contrôler depuis clavier.

Pour cette réalisation, on choisit d'utiliser SFML(Simple and Fast Multimedia Library), comme affichée sur la Figure 2.3. SFML offre une interface simple vers les différents com-

2 RÉALISATION DU CONTRÔLEUR



FIGURE 2.2 – Chargeur de Xbox One



Simple and Fast Multimedia Library

SFML is multi-media

SFML provides a simple interface to the various components of your PC, to ease the development of games and multimedia applications. It is composed of five modules: system, window, graphics, audio and network.

Discover their features more in detail in the tutorials and the API documentation.



SFML is multi-platform

With SFML, your application can compile and run out of the box on the most common operating systems: Windows, Linux, Mac OS X and soon Android & iOS.

Pre-compiled SDKs for your favorite OS are available on the download page.



SFML is multi-language

SFML has official bindings for the C and C# languages. And thanks to its active community, it is also available in many other languages such as Java, Ruby, Python, Go, and more.

Learn more about them on the bindings page.

[Learn more about them on the bindings page.](#)

FIGURE 2.3 – Site de SFML

posants de votre PC, afin de faciliter le développement de jeux ou d'applications multimedia. Elle se compose de cinq modules : système, fenêtrage, graphisme, audio et réseau.

Dans notre cas, on ne s'intéresse qu'à l'utilisation du module 'Window' qui comprend la classe sf::Keyboard. Cette classe nous permet de détecter l'appui des touches depuis le clavier sous C++ et de réagir à l'action de l'utilisateur. A l'aide de cette classe, on peut réaliser le contrôle.

Dans ce projet, dans simulation, on ne fait que 6 mouvements différents :

- avancer
- reculer
- déplacer vers la gauche
- déplacer vers la droite
- monter
- descendre

Pour chacun de ces 6 mouvements, on donne simplement une consigne de vitesse norme constante vers les 6 directions différentes. Pour détecter la touche appuyée, on utilise principalement la méthode 'sf::Keyboard::isKeyPressed' qui a un paramètre de type 'sf::Keyboard::Key'. Dès la détection, on crée une consigne de vitesse correspondante et l'envoie au solveur MPC.