

# Sky – Sailor : Saving energy through control

Projet de semestre

Michaël Devyver

Assistant : Sébastien Gros  
Superviseur : Prof. Dominique Bonvin

EPFL, Lausanne  
Février 2007



# Table des matières

<b>I. INTRODUCTION</b>	<b>4</b>
<b>A. PRÉAMBULE</b>	<b>4</b>
<b>B. L'AVION</b>	<b>4</b>
<b>C. OBJECTIFS DU PROJET DE SEMESTRE</b>	<b>5</b>
<b>II. CONTRÔLE DU SYSTÈME</b>	<b>6</b>
<b>A. STRATÉGIE DE COMMANDE</b>	<b>6</b>
<b>B. REPRÉSENTATION DU SYSTÈME PAR MÉTHODE D'ÉTAT</b>	<b>6</b>
<b>C. DYNAMIQUE DE L'AVION</b>	<b>8</b>
<b>D. FORCES AÉRODYNAMIQUES</b>	<b>9</b>
<b>III. CONTRÔLEUR DE HAUT NIVEAU</b>	<b>10</b>
<b>A. CALCUL DU POINT DE FONCTIONNEMENT</b>	<b>10</b>
<b>B. ADAPTATION DU POINT DE FONCTIONNEMENT</b>	<b>11</b>
1. MESURE DE LA PUISSANCE MOYENNE CONSOMMÉE	11
2. INTERPOLATION DES POINTS	12
3. RECHERCHE DE L'EXTREMUM (MINIMA) DU POLYNÔME	14
<b>C. RÉSULTATS</b>	<b>14</b>
<b>IV. CONTRÔLEUR DE BAS NIVEAU</b>	<b>16</b>
<b>A. LE LQR (LINEAR QUADRATIC REGULATOR)</b>	<b>16</b>
<b>B. RECHERCHE DES MATRICES DE POIDS Q ET R</b>	<b>16</b>
1. FONCTION DE COÛT QUADRATIQUE	17
2. ÉVALUATION DES MATRICES Q ET R AUTOUR DES POINTS DE FONCTIONNEMENT	19
<b>C. RÉSULTATS</b>	<b>20</b>
<b>V. CONCLUSION</b>	<b>21</b>
<b>VI. REMERCIEMENTS</b>	<b>22</b>
<b>VII. BIBLIOGRAPHIE</b>	<b>22</b>
<b>VIII. ANNEXE</b>	<b>23</b>
<b>A. LQR</b>	<b>23</b>
<b>B. CALCUL DU HESSIEN DE LA PUISSANCE</b>	<b>23</b>
<b>C. MATRICE INVERSE DE MOORE-PENROSE</b>	<b>24</b>
<b>D. FMINCON</b>	<b>24</b>

# I. Introduction

## A. Préambule

Le projet « Sky-Sailor » a été lancé début 2004, peu de temps après le « Solar Impulse », par le laboratoire du professeur Roland Siegwart (ETHZ-ASL).

Il a pour but de créer un planeur motorisé ultraléger et sans pilote qui effectuerait, grâce à ses capteurs embarqués, des missions de surveillance, notamment sur la planète Mars mais aussi sur Terre. L'autonomie, tant du point de vue de la navigation que de l'énergie, est une condition importante de ce projet pour atteindre le but fixé, c'est-à-dire naviguer dans les airs pendant des jours, voire des semaines.

## B. L'avion

Un premier prototype (le Sky-Sailor v1.0) a déjà été construit sur la base du modèle réduit de Walter Engel qui détient le record du monde de durée en vol (15h avec une batterie de 1 kg) avec une batterie non-rechargeable.

Sur la base de ce prototype, de nombreuses recherches ont été effectuées dans les domaines du contrôle, de la structure ou du système d'alimentation. Le projet est donc à un stade passablement avancé.

Un modèle du Sky-Sailor est disponible sous Simulink (Matlab) et permet d'effectuer des simulations très réalistes de son comportement en vol.

Quelques chiffres clés :

- Poids total : 2.5 kg
- Poids de la batterie : 1.2 kg
- Envergure de 3.2m
- PC embarqué X-Board 861(avec processeur AMD Geode 266 Mhz)

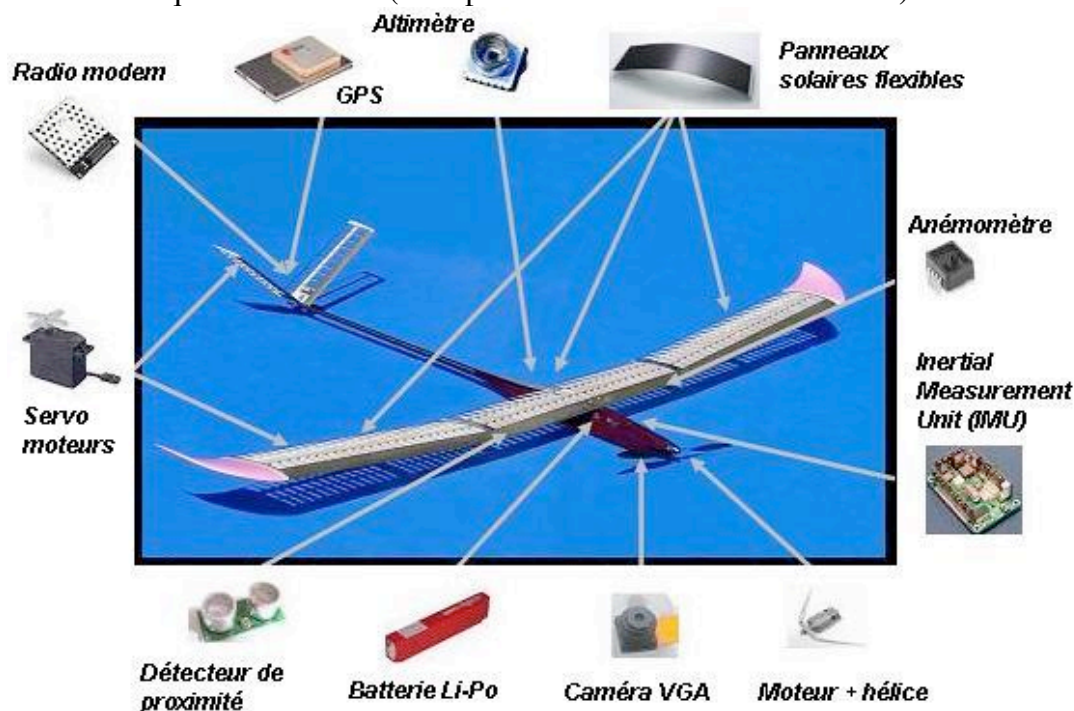


Figure 1 : composants électroniques principaux embarqués sur le Sky-Sailor v1.0

### ***C. Objectifs du projet de semestre***

Un contrôleur convenable a déjà été mis en place, mais des améliorations peuvent être apportées pour optimiser encore plus le comportement de l'avion et faire en sorte qu'il soit le moins gourmand en énergie électrique.

Ce projet de semestre a donc pour but de synthétiser un contrôleur qui soit le plus économe en énergie.

Pour cela, deux objectifs distincts sont à atteindre :

- 1) Développer un contrôleur qui minimise la puissance dépensée pour le contrôle (contrôle de bas niveau)
  
- 2) Trouver puis adapter le point de fonctionnement en temps réel pour pallier les différentes erreurs du modèle et ainsi atteindre la valeur optimale qui minimise l'énergie consommée (contrôle de haut niveau).

## II. Contrôle du système

### A. Stratégie de commande

Une structure par commande optimale avait déjà été choisie dans le but de synthétiser un régulateur avec les deux exigences suivantes:

- Robuste et capable de rejeter les perturbations comme le vent ou les thermiques
- Implémentable sur un PC embarqué avec une faible puissance de calculs

Un premier régulateur a été synthétisé sur deux niveaux comme ceci (Figure 2):

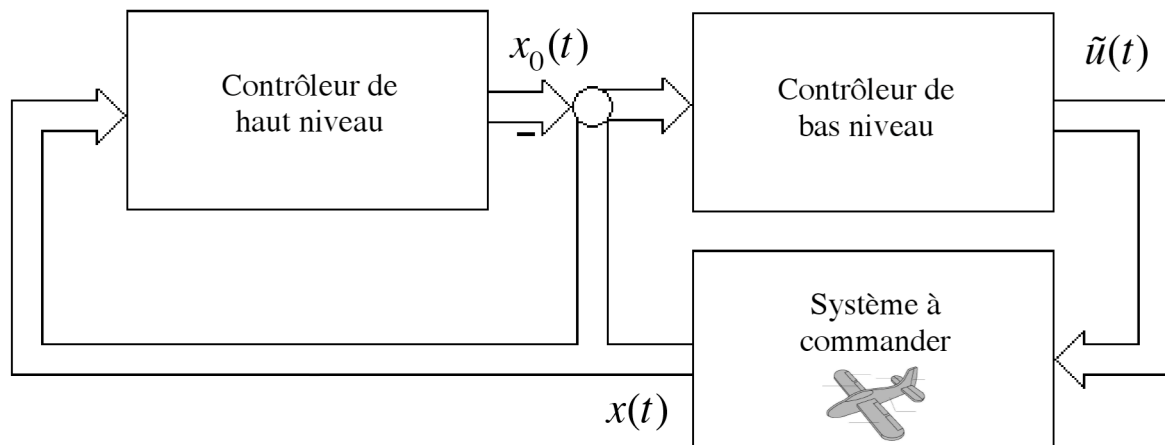


Figure 2 : stratégie de commande

Les objectifs du contrôleur de haut niveau sont :

- Générer la trajectoire
- Suivre la trajectoire aussi en présence de vent
- Donner les consignes au LLC (contrôleur de bas niveau)
- Réduire l'énergie consommée

Les objectifs du contrôleur de bas niveau sont :

- Stabiliser le système (BIBO)
- Suivre les consignes données par le HLC (contrôleur de haut niveau)
- Réduire l'énergie consommée
- Minimiser les sollicitations sur les servomoteurs.

### B. Représentation du système par méthode d'état

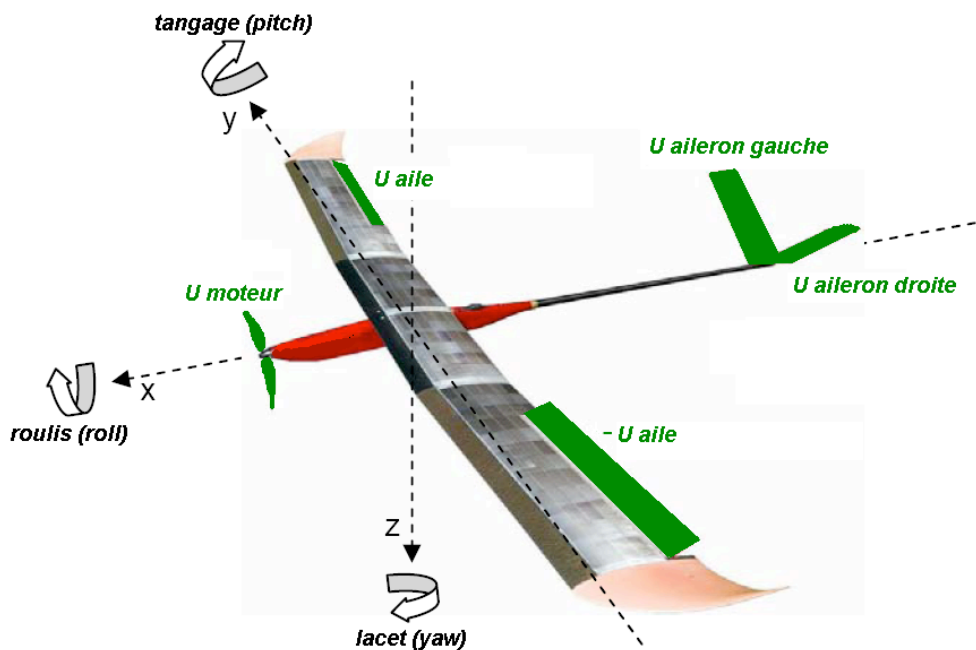
Pour pouvoir synthétiser un régulateur avec la plus grande précision possible, il est nécessaire de connaître le modèle du système à contrôler. Nous sommes en présence d'un système à plusieurs entrées et plusieurs sorties (MIMO).

Il est composé de 12 variables d'états (6 degrés de liberté ou coordonnées généralisées et leurs dérivées) et de 4 entrées:

$$\bar{x}(t) = \begin{pmatrix} x \\ y \\ z \\ \varphi \\ \theta \\ \psi \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} \text{position } x \text{ [m]} \\ \text{position } y \text{ [m]} \\ \text{altitude [m]} \\ \text{roulis [rad]} \\ \text{tangage [rad]} \\ \text{lacet [rad]} \\ \text{vitesse } x \text{ [m/s]} \\ \text{glissement [m/s]} \\ \text{vitesse verticale [m/s]} \\ \text{vitesse angulaire (roulis) [rad/s]} \\ \text{vitesse angulaire (tangage) [rad/s]} \\ \text{vitesse angulaire (lacet) [rad/s]} \end{pmatrix} \quad (\text{II.1})$$

$$\bar{u}(t) = \begin{pmatrix} u_{\text{moteur}} \\ u_{\text{aile}} \\ u_{\text{aileron}_g} \\ u_{\text{aileron}_d} \end{pmatrix} \quad (\text{II.2})$$

Ces variables et ces entrées sont représentées sur l'avion (Figure 3) :



**Figure 3: états et entrées du Sky-Sailor**

Les gouvernes sur les ailes (U aile) sont actionnés avec la même amplitude mais dans des directions opposées. Elles agissent sur le roulis.



Les positions des ailerons en V agissent sur le tangage s'ils pivotent symétriquement (selon la même consigne en tension) ou sinon sur le lacet.



$x, y, z$  sont les positions dans le référentiel de la Terre et  $\varphi, \theta, \psi$  sont les angles d'Euler (roulis, tangage, lacet) dans le même référentiel.

Pour simplifier les notations, le vecteur d'état  $\vec{x}(t)$  et le vecteur d'entrée  $\vec{u}(t)$  seront remplacés respectivement par  $x$  et  $u$  dans la suite de ce rapport. Et pour ne pas confondre la vitesse  $\dot{x}$ , avec la dérivée du vecteur d'état, elle sera remplacée par  $v$ .

### C. Dynamique de l'avion

Les équations dynamiques qui régissent le comportement de l'avion sont les suivantes :

$$\begin{aligned} \dot{x} &= \vec{f}(x, u) \\ y &= \vec{g}(x, u) \end{aligned} \quad (\text{II.3})$$

Après linéarisation, elles sont décrites par le modèle d'état analogique linéarisé ( $\tilde{x} = x - x_0$  et  $\tilde{u} = u - u_0$ ):

$$\begin{aligned} \dot{\tilde{x}} &= A\tilde{x} + B\tilde{u} \\ \tilde{y} &= C\tilde{x} + D\tilde{u} \end{aligned} \quad (\text{II.4})$$

$$\begin{aligned} A_{12 \times 12} &= \left. \frac{\partial f(x, u)}{\partial x} \right|_{x_0, u_0} & B_{4 \times 4} &= \left. \frac{\partial f(x, u)}{\partial u} \right|_{x_0, u_0} \\ C_{12 \times 12} &= \left. \frac{\partial g(x, u)}{\partial x} \right|_{x_0, u_0} & D_{4 \times 4} &= \left. \frac{\partial g(x, u)}{\partial u} \right|_{x_0, u_0} \end{aligned}$$

Les matrices de linéarisation stationnaires  $A, B, C$  et  $D$  (Jacobiens) ont auparavant déjà été calculées à partir des équations dynamiques régissant le système (voir annexe).

L'avantage de la linéarisation est de pouvoir se passer de l'étude difficile de systèmes non linéaires et de se contenter d'une approximation du modèle original, beaucoup plus facile à analyser mais valable uniquement autour de trajectoires nominales (environ 15% autour de la valeur).

Le point de fonctionnement autour duquel le modèle est linéarisé sera calculés au chapitre III.A.



## D. Forces aérodynamiques

Le paramètre qu'il faudra optimiser par la suite pour atteindre le but fixé sera soit la puissance consommée par le moteur soit la puissance de traînée.

Comme nous le verrons au chapitre III.A, cela revient au même pour le système de minimiser l'une ou l'autre puissance.

En vol, les forces et moments qui s'appliquent sur les ailes sont les suivantes :

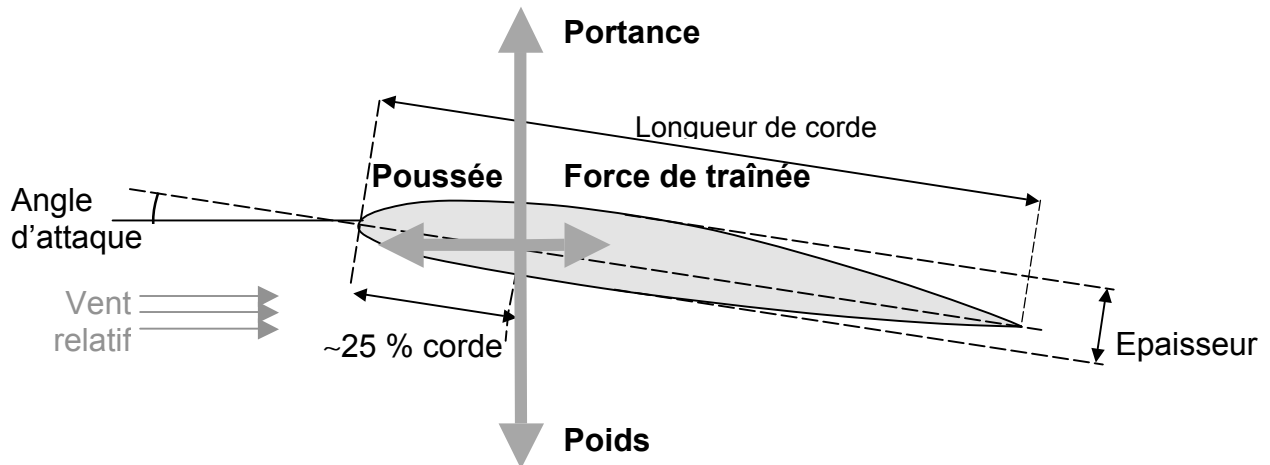


Figure 3 : forces aérodynamiques s'exerçant sur l'aile

Les puissances instantanées sont calculées à partir des forces comme il suit :

$$P_{\text{traînée}}(t) = F_{\text{traînée}}(t) \cdot v(t) \text{ [W]}$$

$$P_{\text{moteur}}(t) = F_{\text{propulsion}}(t) \cdot v(t) \text{ [W]}$$

$$F_{\text{portance}} = C_{\text{portance}} \frac{\rho}{2} S v^2$$

$$F_{\text{traînée}} = C_{\text{traînée}} \frac{\rho}{2} S v^2 \quad (\text{II.5}) \quad \text{où}$$

$$M = C_m \frac{\rho}{2} S v^2 \cdot \text{corde}$$

$\rho$  : densité du fluide (air)

S : surface de l'aile

v : vitesse relative de l'avion par rapport au fluide

$C_{\text{portance}}$  : coefficient de portance

$C_{\text{traînée}}$  : coefficient de traînée

$C_m$  : coefficient de moment

### III. Contrôleur de haut niveau

#### A. Calcul du point de fonctionnement

Le calcul du point de fonctionnement consiste à trouver les valeurs des états  $x$  et des entrées  $u$  qui annulent les forces dynamiques et ne gardent que les forces statiques. C'est-à-dire qu'elles n'évoluent pas au cours du temps et qu'elles mettent le système au repos. Les matrices A,B,C,D seront linéarisées autour de ces points de référence.

Pour trouver les points de fonctionnement  $x_0, u_0$  deux conditions sont à satisfaire:

$$\begin{cases} \bar{f}(x_0, u_0) = \vec{0} \\ \min_{x,u} P_{\text{traînée}}(x, u) \end{cases} \quad (\text{III.1})$$

La première condition stipule que les dynamiques doivent être annulées. C'est la condition principale d'obtention d'un point de fonctionnement : il faut annuler les forces dynamiques pour ne garder que les forces statiques.

La seconde condition ajoute une contrainte à ce point de fonctionnement : il doit rendre le système le plus économe en ce qui concerne la puissance perdue dans la force de traînée.

Les hypothèses suivantes ont été émises:

- Le planeur vole en ligne droite ( $y = \varphi = \dot{\varphi} = u_{\text{ail}} = 0$ ) face au vent ( $\psi = \dot{\psi} = 0$ ).
- Les positions ( $x, y, z$ ) n'influencent pas la dynamique du système.
- Les vitesses verticales et latérales sont nulles ( $\dot{y} = \dot{z} = 0$ ).
- Les consignes appliquées sur les ailerons en V sont symétriques ( $u_{\text{aileron}_g} = u_{\text{aileron}_d}$ ) pour qu'ils fassent pivoter le planeur uniquement sur  $\theta$  et non sur  $\psi$ .

Une fois ces paramètres fixés à zéro, les variables libres restantes sont les suivants :  $v, \theta, u_{\text{motor}}, u_{\text{aileron}}$ . L'algorithme a pour but de les faire varier pour satisfaire au mieux les conditions (IV.1).

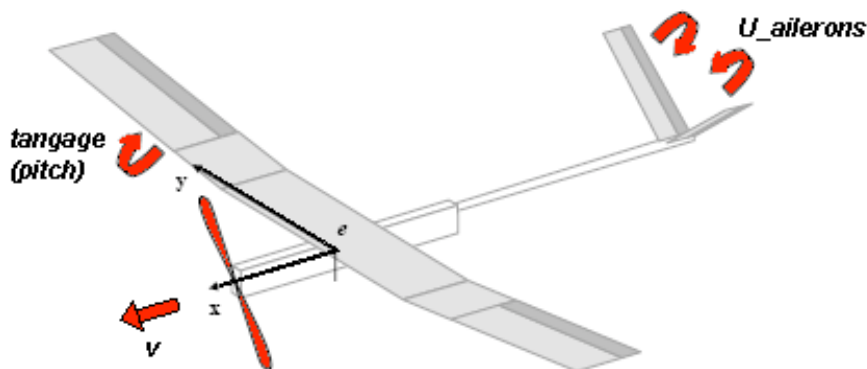


Figure 4 : paramètres libérés pour la minimisation de la puissance

Les résultats obtenus (grâce à la fonction Matlab `fmincon`) sont les suivants :

$$\begin{aligned}\theta_0 &= -0,65^\circ \\ v_0 &= 8.1177 \text{ [m/s]} \\ u_{\_motor_0} &= 19,95 \text{ [V]} \\ u_{\_aileronV_0} &= -0,68^\circ\end{aligned}\quad (\text{III.2})$$

Ces valeurs sont proches de la réalité : des tests effectués avec le prototype ont donné des mesures approximatives très proches de celles-ci ( $v = 8.2$  [V],  $u_{\_motor} = 19$  [V]).

Aussi, en remplaçant la puissance de traînée par la puissance consommée par le moteur, des résultats très similaires à ( III.2 ) sont obtenus :

$$\begin{aligned}\theta_0 &= -0,64^\circ \\ v_0 &= 8.1123 \text{ [m/s]} \\ u_{\_motor_0} &= 19,95 \text{ [V]} \\ u_{\_aileronV_0} &= -0,69^\circ\end{aligned}\quad (\text{III.3})$$

Ces points fixes ( III.3 ) seront utilisés comme références pour la suite, tant pour linéariser les matrices en ( II.4 ) que pour  $x_0$  (Figure 5).

## ***B. Adaptation du point de fonctionnement***

Un point d'équilibre a été calculé théoriquement avec le modèle. Bien que le système soit modélisé très précisément, en réalité, de petites fluctuations dans les constantes du modèle pourraient modifier les équations dynamiques. Par exemple les coefficients (de portance et de traînée) caractérisant la géométrie de l'aile (airfoil) peuvent légèrement varier en cours de vol, lorsque de la poussière s'y accumule. S'ils changent au cours du temps, les équations de force de portée et de traînée ( II.5 ) auront aussi changé et par conséquent, le modèle utilisé lors des simulations ne sera plus valable.

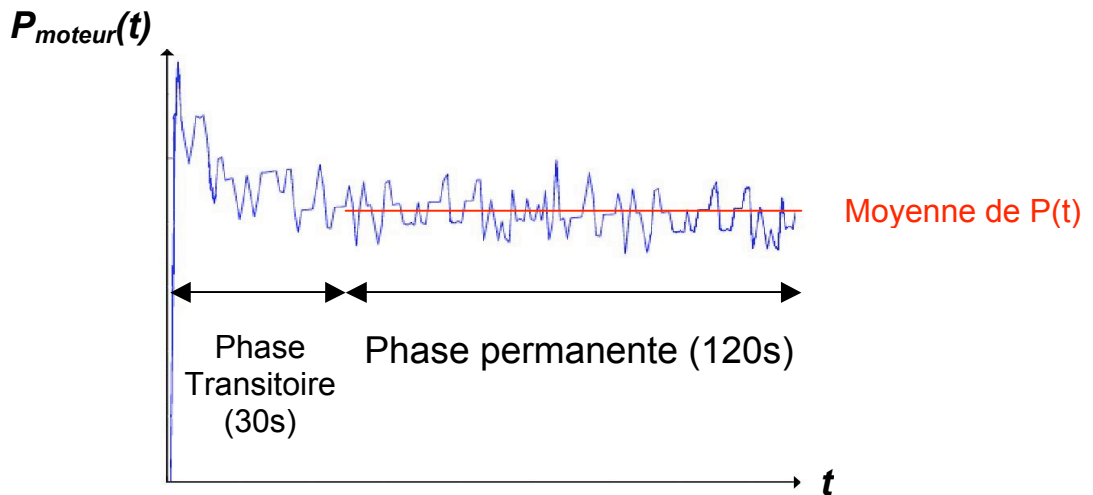
Ainsi les points de fonctionnement calculés avec le modèle non valable ne seront plus utilisables.

Il serait donc judicieux d'adapter le comportement de l'avion aux conditions externes en cherchant les valeurs des consignes nominales qui minimisent la puissance consommée.

Nous choisissons à l'adapter à la vitesse  $v_x$  en temps réel et de trouver sa valeur qui minimiserait la fonction de coût énergétique.

### **1. Mesure de la puissance moyenne consommée**

Pour connaître précisément la consommation du moteur pour une certaine consigne en vitesse imposée, la moyenne est calculée sur une période de 120s avec une fréquence d'échantillonnage de 1Hz.

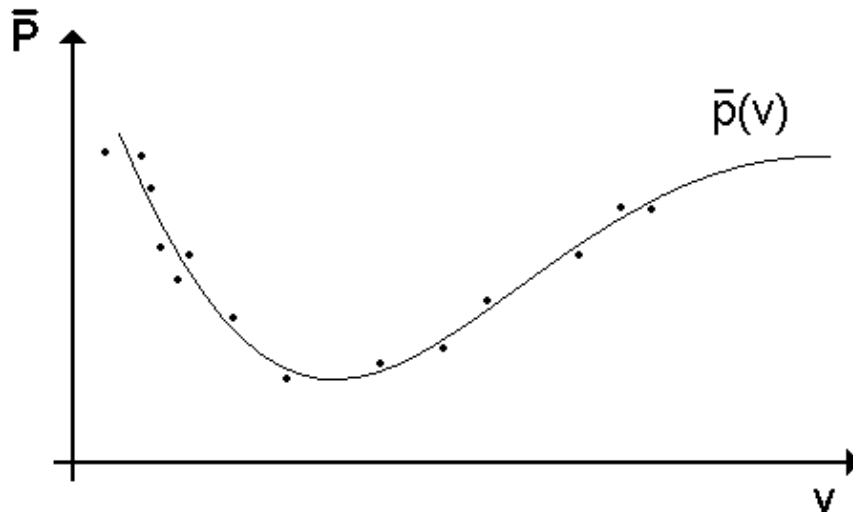


**Figure 6 : mesure de la moyenne de P pour une certaine consigne de vitesse**

La phase transitoire ne doit pas être prise en compte dans le calcul de la moyenne puisqu'elle ne dépend pas uniquement de la consigne posée.

## 2. Interpolation des points

Une fois un nombre suffisant de points mesurés (puissances moyennes en fonction des vitesses) atteint, il suffit de les interpoler pour obtenir le polynôme caractéristique.



**Figure 7 : principe d'interpolation**

Soit  $\bar{p} = \sum_{i=0}^n \alpha_i v^i$  l'équation algébrique du polynôme ( $n$  est le degré du polynôme,  $\alpha_i$  sont les coefficients).

Avec un nombre  $r$  de points, en l'occurrence les puissances moyennes  $\bar{p}_i$  et les vitesses  $v_i$ , les coefficients  $\alpha$  du polynôme se déterminent comme ceci :

$$\begin{pmatrix} \bar{p}_1 \\ \bar{p}_2 \\ \vdots \\ \bar{p}_r \end{pmatrix} = \begin{pmatrix} 1 & v_1^1 & v_1^2 & \cdots & v_1^n \\ 1 & v_2^1 & v_2^2 & \cdots & v_2^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & v_r^1 & v_r^2 & \cdots & v_r^n \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix} \Leftrightarrow \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix} = \begin{pmatrix} 1 & v_1^1 & v_1^2 & \cdots & v_1^n \\ 1 & v_2^1 & v_2^2 & \cdots & v_2^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & v_r^1 & v_r^2 & \cdots & v_r^n \end{pmatrix}^{-1} \begin{pmatrix} \bar{p}_1 \\ \bar{p}_2 \\ \vdots \\ \bar{p}_r \end{pmatrix}$$

La matrice des vitesses n'est pas une matrice carrée. Théoriquement, elle n'est donc pas inversible. Cependant, il existe une méthode permettant de calculer l'unique « pseudo inverse »  $X$  d'une matrice non-carrée  $A$ , c'est la méthode de Moore-Penrose (voir Annexe):

$$\text{Si } r < n : X = A^T (AA^T)^{-1}$$

$$\text{Si } r > n : X = (A^T A)^{-1} A^T$$

### Considérations pratiques :

- Le degré  $n$  du polynôme est choisi de telle sorte à ce que le polynôme ne soit pas une droite ( $n=1$ ) ni qu'il ait trop d'oscillations donc de minimums locaux. Le choix se porte sur un polynôme du second degré :  $n=2$  (qui n'a qu'un seul extremum).
- Pour ne pas prendre en compte tout le passé et ainsi ne pas garder tous les anciens points, seulement une dizaine sont conservées :  $r = 10$ .
- La matrice des vitesses et le vecteur des puissances moyennes sont des matrices FIFO (*first in, first out*) : les plus anciens échantillons sont remplacés par les nouveaux
- Les équations matricielles sont assez simples pour pouvoir être embarqués sur le microcontrôleur et assurer des temps de calculs raisonnables (il n'y a que l'inversion de matrice qui est coûteuse). D'autant qu'elles seront exécutées une fois toutes les 150 secondes.

### 3. Recherche de l'extremum (minima) du polynôme

Une fois le polynôme trouvé, il reste à calculer son minimum pour déterminer à quelle vitesse optimale le moteur électrique consomme le moins d'énergie.

La méthode de Newton est choisie. Elle stipule qu'il y a un extremum au point où la tangente de la fonction est horizontale (pente nulle).

$$\Rightarrow \bar{P}(v_i + \Delta v) = \bar{P}(v_i) + \bar{P}'(v_i) \cdot \Delta v$$

$$\Leftrightarrow \frac{d\bar{P}}{dv} = 0 = \left. \frac{d\bar{P}}{dv} \right|_{v_i} + \left. \frac{d^2\bar{P}}{dv^2} \right|_{v_i} \Delta v$$

$$\Leftrightarrow \Delta v = - \left( \left. \frac{d^2\bar{P}}{dv^2} \right|_{v_i} \right)^{-1} \left. \frac{d\bar{P}}{dv} \right|_{v_i}$$

$\Delta v$  est l'écart entre la vitesse actuelle et l'extremum

$$\Rightarrow v_{i+1} = v_i + \Delta v$$

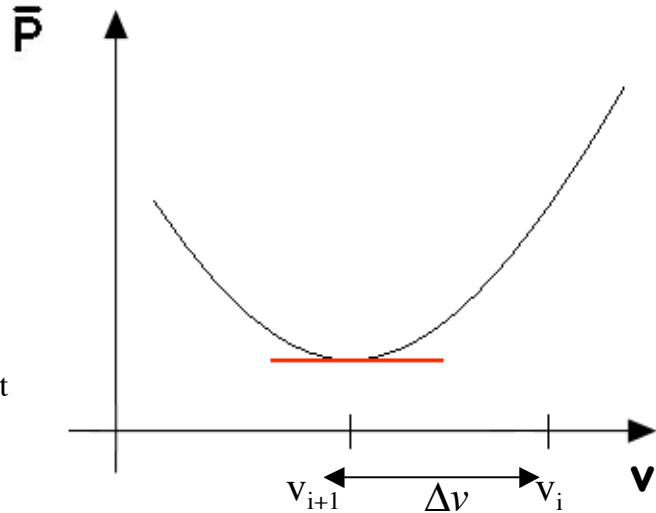


Figure 8 : écart entre le point actuel et l'extremum

Il faut veiller après calculs que l'algorithme ne fournisse pas un maximum mais bien un minimum.

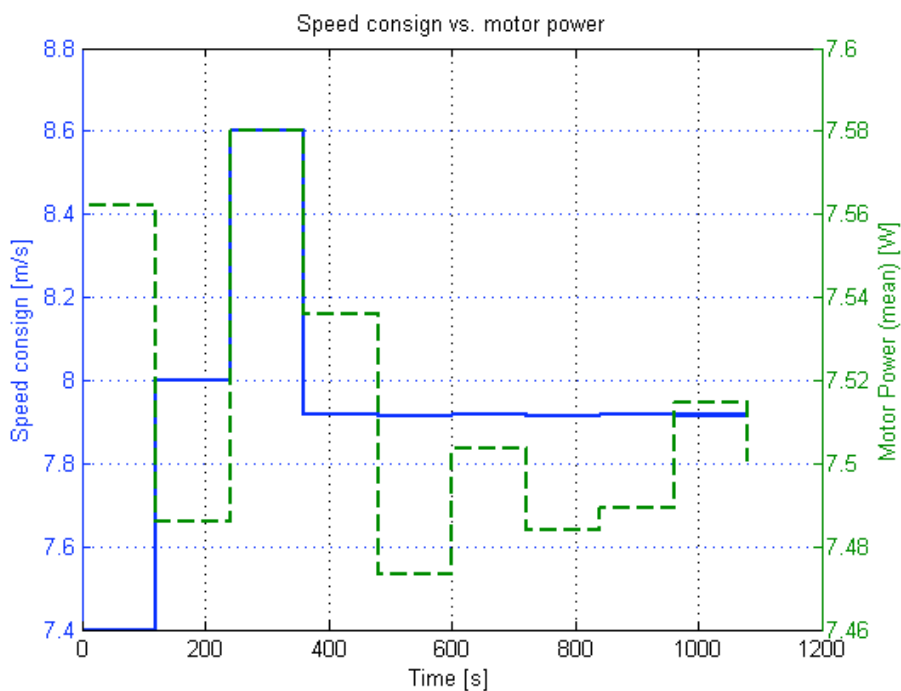
Pour être vraiment rigoureux, il aurait fallu réévaluer en temps réel les matrices de linéarisation A et B ainsi que les matrices Q, R et N autour du nouveau point de fonctionnement. Cependant, vu le temps de calculs nécessaire avec un processeur classique (Pentium IV), il est impossible de le faire avec le processeur embarqué (de plus petite fréquence) qui est à notre disposition sur le Sky-Sailor.

Aussi, le modèle reste linéarisé reste valable pour des valeurs pas trop éloignées (~10 - 20%) du point de fonctionnement.

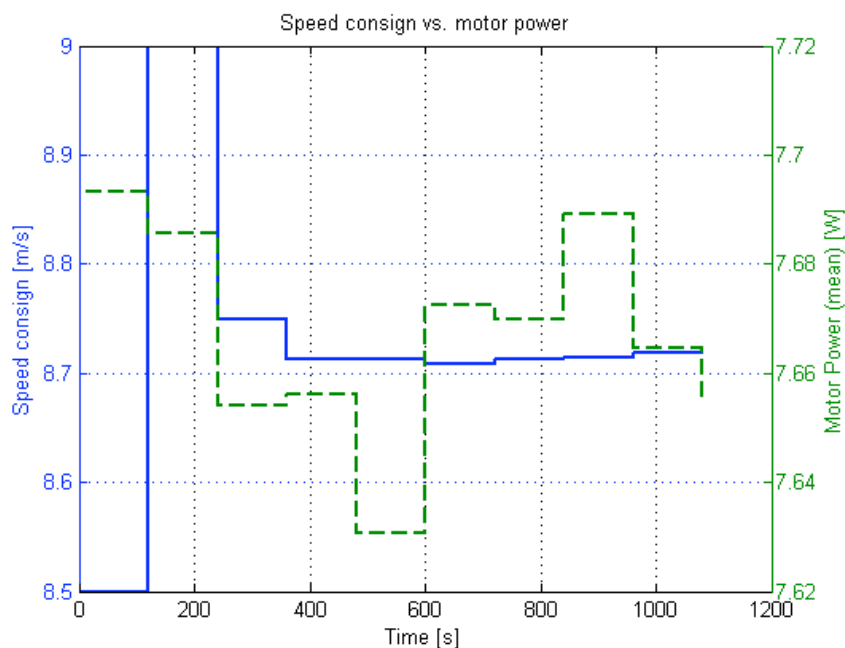
### C. Résultats

La Figure 9 et la Figure 10 montrent que la vitesse converge bien vers le minimum d'énergie consommée.

Comme nous pouvons le constater, les vitesses initiales sont importantes et doivent, pour donner de bons résultats, couvrir la plus grande gamme de vitesses possibles (Figure 9) et pas seulement une région restreinte (Figure 10), sous peine de se retrouver dans un minima local ou sur un point-selle.



**Figure 9 : consigne de vitesse (bleu) contre puissance moyenne consommée (vert). La vitesse converge aux alentours de 7.9 m/s**



**Figure 10 : consigne de vitesse (bleu) contre puissance moyenne consommée (vert). La vitesse converge aux alentours de 8.73 m/s**

## IV. Contrôleur de bas niveau

### A. Le LQR (Linear Quadratic Regulator)

Une contre-réaction de type LQR a pour but de trouver la matrice de gain  $K$  de telle sorte à ce que le système minimise une fonction de coût quadratique. Ici en l'occurrence, nous aimerions minimiser l'énergie consommée par l'avion.

Un LQR a pour but de minimiser la fonction de coût suivante (de type quadratique) :

$$J = \int_0^{\infty} (\tilde{x}^T Q \tilde{x} + \tilde{u}^T R \tilde{u} + 2 \tilde{x}^T N \tilde{u}) dt \quad (\text{IV.1})$$

Avec  $Q = Q^T \geq 0$  et  $R = R^T \geq 0$

La matrice de gain de contre-réaction est ainsi donnée par  $K = R^{-1} B^T P$  (IV.2)

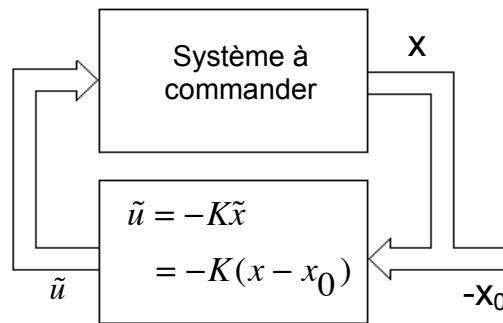


Figure 10 : système contre-réactionné

Où  $P$  est solution de l'équation de Riccati algébrique:

$$PA + A^T P - PBR^{-1}B^T P + Q = 0 \quad (\text{voir annexe pour plus de détails}) \quad (\text{IV.3})$$

### B. Recherche des matrices de poids $Q$ et $R$

À ce jour, il n'existe aucune méthode théorique permettant de dimensionner les matrices de poids  $Q$  et  $R$  d'une commande optimale LQR. Généralement, on assigne manuellement des poids sur les diagonales de ces matrices afin de simplifier le problème. Ces poids sont choisis en fonction de comment on aimerait « pénaliser » les variables d'états par rapport aux entrées lorsqu'elles s'écartent de la consigne, ou inversement.

Par exemple, si l'on ne veut pas que l'état  $\theta$  ne s'écarte trop de son point de fonctionnement (il pourrait induire une trop grande force de traînée), il faudrait alors mettre du poids sur le 5<sup>ème</sup> élément de la diagonale de  $Q$ .



## 1. Fonction de coût quadratique

La fonction de coût à minimiser, c'est-à-dire la puissance de traînée (qui s'oppose au mouvement) est la suivante :  $\min_{x,u} E_{\text{traînée}}(x,u) = \min_{x,u} \int P_{\text{traînée}}(x,u) dt$  (IV.4)

En décomposant  $P_{\text{traînée}}$  (disponible algébriquement dans le modèle) en série de Taylor, nous obtenons :

$$\begin{aligned} P_{\text{traînée}}(\tilde{x}, \tilde{u}) &= P(x_0, u_0) + \nabla P(x, u)|_{x_0, u_0} \begin{pmatrix} x - x_0 \\ u - u_0 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} x^T - x_0^T & u^T - u_0^T \end{pmatrix} \nabla^2 P(x, u)|_{x_0, u_0} \begin{pmatrix} x - x_0 \\ u - u_0 \end{pmatrix} + \dots \\ &= P(x_0, u_0) + \nabla P(x, u)|_{x_0, u_0} \begin{pmatrix} \tilde{x} \\ \tilde{u} \end{pmatrix} + \frac{1}{2} \begin{pmatrix} \tilde{x}^T & \tilde{u}^T \end{pmatrix} \nabla^2 P(x, u)|_{x_0, u_0} \begin{pmatrix} \tilde{x} \\ \tilde{u} \end{pmatrix} + \dots \end{aligned} \quad (\text{IV.5})$$

$$\text{où } \nabla P(x, u)|_{x_0, u_0} = \left( \frac{\partial P}{\partial x_1}, \frac{\partial P}{\partial x_2}, \dots, \frac{\partial P}{\partial x_{12}}, \frac{\partial P}{\partial u_1}, \frac{\partial P}{\partial u_2}, \dots, \frac{\partial P}{\partial u_4} \right) \Big|_{x_0, u_0} = \left( \frac{\partial P}{\partial \xi_1}, \frac{\partial P}{\partial \xi_2}, \dots, \frac{\partial P}{\partial \xi_{16}} \right) \Big|_{x_0, u_0} \quad (\text{IV.6})$$

est le gradient de la puissance P par rapport aux états et aux entrées, évalué en  $x_0, u_0$

avec  $\xi_{1 \times 16} = \begin{pmatrix} x^T & u^T \end{pmatrix}$

$$\text{et } \nabla^2 P(x, u)|_{x_0, u_0} = \begin{pmatrix} \frac{\partial^2 P}{\partial \xi_1^2} & \frac{\partial^2 P}{\partial \xi_1 \partial \xi_2} & \dots & \frac{\partial^2 P}{\partial \xi_1 \partial \xi_{16}} \\ \frac{\partial^2 P}{\partial \xi_2 \partial \xi_1} & \frac{\partial^2 P}{\partial \xi_2^2} & \dots & \frac{\partial^2 P}{\partial \xi_2 \partial \xi_{16}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 P}{\partial \xi_{16} \partial \xi_1} & \frac{\partial^2 P}{\partial \xi_{16} \partial \xi_2} & \dots & \frac{\partial^2 P}{\partial \xi_{16}^2} \end{pmatrix} \Big|_{\substack{x=x_0 \\ u=u_0}} \quad (\text{IV.7})$$

est le Hessian de la puissance P par rapport aux états et aux entrées, évalué en  $x_0, u_0$

Par conséquent, en combinant (IV.4) et (IV.5) :

$$\begin{aligned} &\min_{\tilde{x}, \tilde{u}} E_{\text{traînée}}(\tilde{x}, \tilde{u}) \\ &= \min_{\tilde{x}, \tilde{u}} \left( \underbrace{\int P_0 dt}_{\text{cst}} + \underbrace{\int \nabla P(x, u)|_{x_0, u_0} \begin{pmatrix} \tilde{x} \\ \tilde{u} \end{pmatrix} dt}_{\approx 0} + \underbrace{\frac{1}{2} \int \begin{pmatrix} \tilde{x} & \tilde{u} \end{pmatrix} \nabla^2 P(x, u)|_{x_0, u_0} \begin{pmatrix} \tilde{x} \\ \tilde{u} \end{pmatrix} dt}_{\text{terme quadratique}} + \dots \right) \quad (\text{IV.8}) \end{aligned}$$

Le gradient de la puissance  $\nabla P(x, u)$  s'annule au point de fonctionnement (minimum de la fonction P) et les termes d'ordre supérieur au terme de second degré sont négligeables. Par conséquent, le seul terme qui peut être minimisé est le terme quadratique. Il correspond en fait à la mesure de performance, minimisée par le régulateur linéaire quadratique (LQR).

$$\begin{aligned}
\text{Ainsi, } & \frac{1}{2} \int_0^{\infty} \begin{pmatrix} \tilde{x}^T & \tilde{u}^T \end{pmatrix} \nabla^2 P(x, u) \Big|_{x_0, u_0} \begin{pmatrix} \tilde{x} \\ \tilde{u} \end{pmatrix} dt \quad (\text{terme quadratique}) & \quad (\text{IV.9}) \\
& = \frac{1}{2} \int_0^{\infty} \begin{pmatrix} \tilde{x}^T & \tilde{u}^T \end{pmatrix} \begin{pmatrix} Q & N \\ N^T & R \end{pmatrix} \begin{pmatrix} \tilde{x} \\ \tilde{u} \end{pmatrix} dt \\
& = \frac{1}{2} \int_0^{\infty} (\tilde{x}^T Q \tilde{x} + \tilde{u}^T R \tilde{u} + 2 \tilde{x}^T N \tilde{u}) dt \quad (\text{fonction de coût}) \\
& = J
\end{aligned}$$

Connaissant le Hessien (IV.7) de la puissance P, nous pouvons en déduire les matrices Q(12x12), R(4x4) et N(12x4) par identification.

$$\nabla^2 P(x, u) = \begin{pmatrix} Q & N \\ N^T & R \end{pmatrix}$$

$$Q = \begin{pmatrix} \frac{\partial^2 P(x, u)}{\partial x_1^2} & \frac{\partial^2 P(x, u)}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 P(x, u)}{\partial x_1 \partial x_{12}} \\ \frac{\partial^2 P(x, u)}{\partial x_2 \partial x_1} & \frac{\partial^2 P(x, u)}{\partial x_2^2} & \dots & \frac{\partial^2 P(x, u)}{\partial x_2 \partial x_{12}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 P(x, u)}{\partial x_{12} \partial x_1} & \frac{\partial^2 P(x, u)}{\partial x_{12} \partial x_2} & \dots & \frac{\partial^2 P(x, u)}{\partial x_{12}^2} \end{pmatrix} \Bigg|_{\substack{x=x_0 \\ u=u_0}}$$

$$N = \begin{pmatrix} \frac{\partial^2 P(x, u)}{\partial x_1 \partial u_1} & \frac{\partial^2 P(x, u)}{\partial x_1 \partial u_2} & \frac{\partial^2 P(x, u)}{\partial x_1 \partial u_3} & \frac{\partial^2 P(x, u)}{\partial x_1 \partial u_4} \\ \frac{\partial^2 P(x, u)}{\partial x_2 \partial u_1} & \frac{\partial^2 P(x, u)}{\partial x_2 \partial u_2} & \frac{\partial^2 P(x, u)}{\partial x_2 \partial u_3} & \frac{\partial^2 P(x, u)}{\partial x_2 \partial u_4} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial^2 P(x, u)}{\partial x_{12} \partial u_1} & \frac{\partial^2 P(x, u)}{\partial x_{12} \partial u_2} & \frac{\partial^2 P(x, u)}{\partial x_{12} \partial u_3} & \frac{\partial^2 P(x, u)}{\partial x_{12} \partial u_4} \end{pmatrix} \Bigg|_{\substack{x=x_0 \\ u=u_0}}$$

(IV.10) et (IV.11)

$$R = \begin{pmatrix} \frac{\partial^2 P(x, u)}{\partial u_1^2} & \frac{\partial^2 P(x, u)}{\partial u_1 \partial u_2} & \frac{\partial^2 P(x, u)}{\partial u_1 \partial u_3} & \frac{\partial^2 P(x, u)}{\partial u_1 \partial u_4} \\ \frac{\partial^2 P(x, u)}{\partial u_2 \partial u_1} & \frac{\partial^2 P(x, u)}{\partial u_2^2} & \frac{\partial^2 P(x, u)}{\partial u_2 \partial u_3} & \frac{\partial^2 P(x, u)}{\partial u_2 \partial u_4} \\ \frac{\partial^2 P(x, u)}{\partial u_3 \partial u_1} & \frac{\partial^2 P(x, u)}{\partial u_3 \partial u_2} & \frac{\partial^2 P(x, u)}{\partial u_3^2} & \frac{\partial^2 P(x, u)}{\partial u_3 \partial u_4} \\ \frac{\partial^2 P(x, u)}{\partial u_4 \partial u_1} & \frac{\partial^2 P(x, u)}{\partial u_4 \partial u_2} & \frac{\partial^2 P(x, u)}{\partial u_4 \partial u_3} & \frac{\partial^2 P(x, u)}{\partial u_4^2} \end{pmatrix} \Bigg|_{\substack{x=x_0 \\ u=u_0}} \quad (\text{IV.12})$$

## 2. Évaluation des matrices Q et R autour des points de fonctionnement

La fonction algébrique de la puissance P de traînée est disponible facilement dans le modèle. Cependant, c'est une équation très longue, comportant d'innombrables variables (variables d'états, variables d'entrées, constantes diverses, etc.) et que Matlab a difficile à évaluer autour des points nominaux.

La méthode directe, qui consiste à calculer algébriquement le Hessien de la puissance puis à l'évaluer, en remplaçant les variables d'état et d'entrées par leurs valeurs numériques, ne fonctionne pas sur Matlab. Le programme affiche le message d'erreur suivant lors de l'évaluation en  $x_0, u_0$  :

??? Out of memory. Type HELP MEMORY for your options.

Le code a été exécuté sur une machine de bureau contenant un mono-processeur Pentium Xeon 3 Ghz et 1GB de mémoire. La même erreur se reproduit sur d'autres machines plus puissantes (bi-processeur 3 Ghz, 16GB de mémoire).

Le même message apparaît si l'on évalue une à une chaque cellule du Hessien.

Une solution possible à ce problème serait d'utiliser la fonction Matlab `fmincon` (elle permet de calculer le minimum d'une fonction sous contrainte).

Il se fait que pour des raisons internes à son fonctionnement, cette dernière a besoin du Hessien du Lagrangien  $L = \phi + \mu^T \vec{f}$  pour converger vers la solution de (IV.13).

$$\begin{cases} \vec{f}(x_0, u_0) = 0 \\ \min_{x, u} \phi \end{cases} \quad (\text{IV.13})$$

`fmincon` va chercher les points de fonctionnement  $x_0, u_0$  qui minimisent (IV.14) (la puissance) avec comme contrainte d'annuler l'équation dynamique.

$$\text{où} \quad \phi = F_{\text{moteur}}(t) \cdot v(t) - \frac{d}{dt}(mv^2 + mgz) = F_{\text{moteur}}(t) \cdot v(t) - \frac{1}{2}mv\dot{v} - mg\dot{z} \quad (\text{IV.14})$$

$\vec{f}(x, u)$  est l'équation dynamique de notre système.

La solution en première approximation [4] est  $\tilde{u} = -K\tilde{x}$  où K est un régulateur quadratique linéaire (LQR) avec les poids :

$$\begin{pmatrix} Q & N \\ N^T & R \end{pmatrix} = \nabla^2 L(x, u) \quad (\text{IV.15})$$

Si la puissance mécanique (dérivée de l'énergie potentielle et de l'énergie cinétique) intervient dans la fonction de coût (IV.14) c'est pour donner la possibilité au système de transformer l'énergie créée par le moteur en énergie potentielle ou cinétique (par exemple un gain en altitude) sans être pénalisé.

### C. Résultats

Il faut veiller à ce que les points de fonctionnement sortis de fmincon soient proches de pour que le Hessien soit évalué correctement. Pour cela, on fait varier cette fois tous les paramètres, mais on force fmincon à annuler les variables qui doivent l'être (mêmes hypothèses qu'en ( III. A )).

Le Hessien obtenu est :

$$\nabla^2 P \Big|_{\xi_0} = \begin{array}{c} \mathbf{Q} \\ \left( \begin{array}{cccccccccccc|cccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0.02 & 0 & 0.02 & -0.06 & 0.18 & -0.33 & 0.23 & 0.14 & 1.53 & 0.02 & 0.01 & 0 \\ 0 & 0 & 0 & 0.02 & 0.15 & 0.01 & 0.06 & -0.19 & 0.6 & -1.07 & 0.72 & 0.45 & 4.96 & 0.03 & 0.02 & 0 \\ 0 & 0 & 0 & 0 & 0.01 & 0.01 & 0.01 & 0.04 & -0.12 & 0.22 & -0.14 & -0.09 & -1.02 & -0.02 & -0.01 & 0 \\ 0 & 0 & 0 & 0.02 & 0.06 & 0.01 & 0.05 & -0.14 & 0.43 & -0.77 & 0.52 & 0.32 & 3.59 & 0.03 & 0.02 & 0 \\ 0 & 0 & 0 & -0.06 & -0.19 & 0.04 & -0.14 & 0.9 & -2.44 & 5.11 & -3.13 & -2.14 & -23.73 & -0.22 & -0.18 & 0 \\ 0 & 0 & 0 & 0.18 & 0.6 & -0.12 & 0.43 & -2.44 & 7.52 & -13.80 & 9.21 & 5.78 & 64.05 & 0.68 & 0.55 & 0 \\ 0 & 0 & 0 & -0.33 & -1.07 & 0.22 & -0.77 & 5.11 & -13.80 & 28.95 & -17.76 & -12.12 & -134.51 & -1.24 & -1.01 & 0 \\ 0 & 0 & 0 & 0.23 & 0.72 & -0.14 & 0.52 & -3.13 & 9.21 & -17.76 & 11.48 & 7.44 & 82.46 & 0.83 & 0.67 & 0 \\ 0 & 0 & 0 & 0.14 & 0.45 & -0.09 & 0.32 & -2.14 & 5.78 & -12.12 & 7.44 & 5.07 & 56.32 & 0.52 & 0.42 & 0 \\ \hline 0 & 0 & 0 & 1.53 & 4.96 & -1.02 & 3.59 & -23.73 & 64.05 & -134.51 & 82.46 & 56.32 & 625.01 & 5.74 & 4.68 & 0 \\ 0 & 0 & 0 & 0.02 & 0.03 & -0.02 & 0.03 & -0.22 & 0.68 & -1.24 & 0.83 & 0.52 & 5.74 & 0.07 & 0.06 & 0 \\ 0 & 0 & 0 & 0.01 & 0.02 & -0.01 & 0.02 & -0.18 & 0.55 & -1.01 & 0.67 & 0.42 & 4.68 & 0.06 & 0.05 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right) \mathbf{N} \\ \mathbf{N}^T \\ \mathbf{R} \end{array}$$

fmincon renvoie les points de fonctionnement suivants, qui sont relativement proches de ( III.3 ) :

$$\begin{aligned} \xi_0 &= (x \ y \ z \ \varphi \ \theta \ \psi \ \dot{x} \ \dot{y} \ \dot{z} \ \dot{\varphi} \ \dot{\theta} \ \dot{\psi} \ u_{ail} \ u_{aileron\_g} \ u_{aileron\_d} \ u_{motor})^T \\ &= (0 \ 0 \ 0 \ 0 \ -0.008 \ 0 \ 7.8575 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ -0.206 \ -0.206 \ 19.8045)^T \end{aligned}$$

L'une des conditions imposée par le LQR sur les matrices Q et R est qu'elles doivent être définies semi-positives ( IV.1 ). Ce n'est pas le cas avec celles trouvées ci-dessus, c'est pourquoi une astuce permet de contourner cette erreur. Cela consiste à additionner des petites valeurs sur la diagonale de chacune des deux matrices de poids jusqu'à ce que celles-ci deviennent définies semi-positives :

$$Q = Q + \varepsilon I$$

$$R = R + \varepsilon I$$

## V. Conclusion

Les points de fonctionnement ont été calculés et sont optimisés pour minimiser la puissance consommée, donc perdue, par le système.

Si malgré toutes les précautions prises, ces derniers ont été déterminés avec un modèle trop éloigné de la réalité, l'algorithme permet de corriger cela en temps réel et d'adapter la consigne de vitesse en tout temps. Les résultats sont convaincants et ils correspondent aux attentes.

En ce qui concerne le LQR, le Hessien a pu être calculé puis évalué correctement grâce à `fmincon`.

Ce projet m'a fait apprendre beaucoup de choses. D'une part j'ai pu travailler sur un sujet lié à l'aéronautique, domaine qui m'intéresse très fort et d'autre part, j'ai appris à dimensionner concrètement un contrôleur optimal, ce que je n'avais jamais fait auparavant.

Lausanne, le jeudi 15 février 2007

Michaël Devyver

## VI. Remerciements

Je tiens à remercier chaleureusement Sébastien Gros pour son assistance et sa disponibilité durant tout ce semestre. Je remercie également le professeur Dominique Bonvin ainsi que tout le laboratoire d'Automatique qui m'a permis de faire ce projet de semestre.

## VII. Bibliographie

- [1] D. Gillet, *Systèmes multivariables I, Méthodes d'état*, sept. 2006, Lausanne
- [2] A. Mattio, *Modelling and Control of the UAV Sky-Sailor*, été 2006, Lausanne
- [3] A. Noth, S. Bouabdallah, S. Michaud, R. Siegwart, W. Engel, *Sky-Sailor : design of an autonomous solar powered martian airplane*, Proceedings of Field and Service Robotics, August 2005, Port Douglas, Australia
- [4] Bryson et Ho, *Applied Optimal Control*, 1975, Washington DC, Hemisphere Publications
- [5] J. W. Burdick, *Course handouts on the pseudo-inverse*  
<http://robotics.caltech.edu/~jwb/courses/ME115/handouts/handouts.html>
- [6] R.M Murray, Lecture notes on LQR control, California Institute of Technology, janvier 2006

### Internet

- [6] <http://sky-sailor.epfl.ch>, Site officiel du Sky-Sailor
- [7] <http://fr.wikipedia.org> , Encyclopédie en ligne Wikipedia

## VIII. Annexe

### A. LQR

Le système du type :  $\begin{aligned}\dot{\tilde{x}} &= A\tilde{x} + B\tilde{u} \\ \tilde{y} &= C\tilde{x} + D\tilde{u}\end{aligned}$

Qui doit minimiser une fonction de coût  $J = \int_0^{\infty} (\tilde{x}^T Q \tilde{x} + \tilde{u}^T R \tilde{u} + 2\tilde{x}^T N \tilde{u}) dt$

Est résolu comme ceci :

$$H = x^T Q x + u^T R u + \lambda^T (A x + B u)$$

$$\dot{x} = \left( \frac{\partial H}{\partial \lambda} \right)^T = A x + B u \quad x(0) = x_0$$

$$-\dot{\lambda} = \left( \frac{\partial H}{\partial x} \right)^T = Q x + A^T \lambda \quad \lambda(T) = P_1 x(T)$$

$$0 = \frac{\partial H}{\partial u} = R u + \lambda^T B \quad \Rightarrow u = -R^{-1} B^T \lambda$$

Il est contrôlable si le rang de  $Q = n$  (dimension du système) où

$$Q = \begin{pmatrix} B & AB & A^2 B & \cdots & A^{n-1} B \end{pmatrix}$$

### B. Calcul du Hessien de la puissance

Code Matlab

```
LB = [-0.001 -0.1745 -0.001 5 -0.001 -0.001 -0.001 -0.001 -0.001 -0.001 -  
0.1745 -0.1745 18];  
UB = [0.001 0.1745 0.001 12 0.001 0.001 0.001 0.001 0.001 0.001 0.001  
0.1745 0.1745 22];  
  
options = optimset;  
options.TolCon=0.00001;  
options.TolFun=0.00001;  
options.TolX=0.00001;  
options.MaxIter = 15000;  
options.MaxFunEvals = 15000;  
  
parameters0 = [0 -0.01 0 8 0 0 0 0 0 0 0.03 0.03 19];  
%phi,theta,psi,dx,dy,dz,dphi,dtheta,dpsi,u_ail,u_vtail_r,u_vtail_l,u  
  
tic  
[x,fval,exitflag,output,lambda,grad,hessian] =  
fmincon('phi',parameters0,[],[],[],[],[],LB,UB,'dynamic_cost_function',options)  
toc  
  
save hessian hessian
```

Phi

```
states = [phi theta psi dx dy dz dphi dtheta dpsi];  
input = [u_ail u_vtail_l u_vtail_r u];  
phi = Propeller_Thrust([u dx]).*dx - m*dz*(A(9,4:12)*states' + B(9,:)*input') - m*9.81*dz;
```

### C. Matrice inverse de Moore-Penrose

La matrice pseudo-inverse de Moore-Penrose est une façon de trouver la solution à l'équation :

$$\vec{b} = A\vec{y} \quad \vec{b} \in \mathbb{R}^m, \vec{y} \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}$$

Moore-Penrose a trouvé qu'il existe une solution à ce problème du type :  $\vec{y} = X\vec{b}$  où X est la matrice pseudo-inverse de A.

X satisfait les quatre propriétés : (1)  $AXA = A$

$$(2) XAX = X$$

$$(3) (AX)^H = AX$$

$$(4) (XA)^H = XA$$

Différents cas de figure ont à envisager.

- Si A est de rang plein (c'est-à-dire que ses colonnes sont linéairement indépendantes) :
  - $m < n$  : La solution est celle qui minimise la quantité  $\|\vec{b} - A\vec{y}\|$ . La pseudo-inverse donne le vecteur  $\vec{y}$  qui rapproche au plus  $\vec{b}$  de  $A\vec{y}$ .

$$X = A^T (AA^T)^{-1}$$

- $m > n$  : La solution minimise la norme de  $\vec{y}$ .

$$X = (A^T A)^{-1} A^T$$

- Si A n'est pas de rang plein, il faut décomposer A en valeurs singulières.

Si  $A = U\Sigma V^T$  (décomposition en valeurs singulières)

Alors  $X = V\Sigma^{-1}U^T$

$$\Sigma = \begin{pmatrix} \sigma_1 & 0 & \dots & 0 & 0 \\ 0 & \sigma_2 & \dots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & \sigma_p & 0 \end{pmatrix} \text{ et } \Sigma^{-1} = \begin{pmatrix} 1/\sigma_1 & 0 & \dots & 0 & 0 \\ 0 & 1/\sigma_2 & \dots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & 1/\sigma_p & 0 \end{pmatrix}$$

### D. Fmincon

Lorsqu'on lui donne des contraintes statiques du type :

$$\begin{cases} \vec{f}(x_0, u_0) = \vec{0} \\ \min_{x,u} \phi(x, u) \end{cases}$$

fmincon calcule le Lagrangien :  $L = \phi + \mu^T \vec{f}$  ( $\mu$  est le multiplicateur de Lagrange) et va faire en sorte d'atteindre ces trois conditions :



$$L_u = 0$$

$$L_x = 0$$

$$F = 0$$

Pour des raisons internes à son fonctionnement, il a besoin de calculer le Hessien du

Lagrangien.  $\nabla^2 L(x, u) = \begin{pmatrix} L_{xx} & L_{xu} \\ L_{ux} & L_{uu} \end{pmatrix}$