

# Optimisation de la trajectoire d'un drone

MTH8408

Jouglet Nicolas, Dawut Esse, Joey Van Melle

Le contrôle de drones quadrotors est un enjeu majeur dans le domaine de la robotique autonome en raison de leur grande maniabilité, mais aussi de leur instabilité naturelle.

Ce projet vise à concevoir un **contrôleur optimal** capable de suivre une trajectoire prédéfinie tout en minimisant la consommation d'énergie.

Nous nous appuyons pour cela sur l'approche proposée par Suicmez et Kutay (2014), qui repose sur la commande optimale discrète en temps.

## Contexte du problème

- Les **quadrotors** sont des drones à 4 moteurs capables de **décoller et atterrir verticalement (VTOL)** et d'exécuter des **manœuvres agiles**.
- Ce sont des systèmes **fortement non linéaires et instables**.
- Leur principale limitation : **une consommation énergétique élevée**.

## Objectif du projet

Concevoir un **contrôleur optimal** permettant à un **drone quadrotor** de :

- suivre avec précision une trajectoire tridimensionnelle prédéfinie  $\mathbf{r}(k)$  ;
- limiter la consommation d'énergie (notamment via la poussée totale  $\mathbf{U}_1$ ) ;
- respecter les contraintes physiques (notamment sur les angles d'Euler) afin de conserver la validité du modèle linéarisé.

## Modélisation du système

Dans un premier temps, le système est modélisé de façon dynamique à partir des équations de Newton :

- **Dynamique translationnelle** : influencée par la poussée verticale  $\mathbf{U}_1$  et les angles  $\phi, \theta$  ;
- **Dynamique rotationnelle** : influencée par les couples de commande  $\mathbf{U}_2, \mathbf{U}_3, \mathbf{U}_4$ .

Le modèle dynamique est constitué de deux parties :

- Équation non linéaire pour le mouvement de translation :

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \text{LEB} \begin{bmatrix} 0 \\ 0 \\ \mathbf{U}_1/m \end{bmatrix} - \left( \frac{\mathbf{K}_t}{m} \right) \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \quad (1)$$

- Équations non linéaires pour le mouvement de rotation :

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{(I_y - I_z)}{I_x} qr \\ \frac{(I_z - I_x)}{I_y} pr \\ \frac{(I_x - I_y)}{I_z} pq \end{bmatrix} + \begin{bmatrix} \frac{u_2 d}{I_x} \\ \frac{u_3 d}{I_y} \\ \frac{u_4}{I_z} \end{bmatrix} \quad (2)$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin(\phi) \tan(\theta) & \cos(\phi) \tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)/\cos(\theta) & \cos(\phi)/\cos(\theta) \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (3)$$

## Simplification du système

L'hypothèse suivante permet ensuite de simplifier les équations :

Si les perturbations par rapport à la condition de vol stationnaire sont faibles, les vitesses angulaires du corps  $[p; q; r]$  et les dérivées des angles d'Euler  $[\dot{\phi}, \dot{\theta}, \dot{\psi}]$  peuvent être considérées comme égales.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad ; \quad \begin{bmatrix} \ddot{p} \\ \ddot{q} \\ \ddot{r} \end{bmatrix} = \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} \quad (4)$$

En utilisant l'équation 4 dans l'équation 2, on obtient :

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{(I_y - I_z)}{I_x} \dot{\theta} \dot{\psi} \\ \frac{(I_z - I_x)}{I_y} \dot{\phi} \dot{\psi} \\ \frac{(I_x - I_y)}{I_z} \dot{\phi} \dot{\theta} \end{bmatrix} + \begin{bmatrix} \frac{u_2 d}{I_x} \\ \frac{u_3 d}{I_y} \\ \frac{u_4}{I_z} \end{bmatrix} \quad (5)$$

## Linéarisation du modèle dynamique simplifié

En posant :

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \end{bmatrix} = \begin{bmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \\ \psi \\ \dot{\psi} \\ x \\ \dot{x} \\ y \\ \dot{y} \\ z \\ \dot{z} \end{bmatrix} \quad ; \quad Y = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \end{bmatrix} \quad ; \quad u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \quad (6)$$

En linéarisant les équations 1 et 5 autour du vol stationnaire grâce aux hypothèses précédentes on obtient :

$$\dot{X} = AX + BU \quad ; \quad Y = CX \quad (7)$$

Où A et B et C valent :

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (A)$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{1}{I_x} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{I_z} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 \end{bmatrix} \quad (B)$$

$$C = I_{12} \quad (C)$$

### Discrétisation du temps :

On discrétise l'équation 7 avec un pas de temps  $ts=0,01s$ , on obtient alors un modèle discret :

$$X(k+1) = A_d X(k) + B_d U(k) \quad ; \quad Y(k) = C_d X(k) \quad (7)$$

La discrétisation des équations continues  $\dot{X} = AX + BU$  avec un pas de temps  $T_s$  selon la méthode d'Euler (ordre 1) donne les équations discrètes suivantes :

$$A_d = I + T_s A \quad ; \quad B_d = T_s B \quad ; \quad C_d = C \quad (8)$$

Le terme  $C_d X(k)$  correspond à la sortie du système observée. Dans notre cas, comme  $C_d = I$ , cela revient à comparer directement les états à la trajectoire de référence.

### Fonction Performance

La fonction performance est construite de la façon suivante :

$$J = \frac{1}{2} \sum_{k=k_0}^{k_f-1} (\|C_d X(k) - r(k)\|_Q^2 + \|U(k)\|_R^2) + \frac{1}{2} \|C_d X(k_f) - r(k_f)\|_F^2 \quad (8)$$

où  $\|x\|_M^2 = x^T M x$  est la norme quadratique pondérée par la matrice  $M$ , et  $F$  est la matrice de coût terminal, souvent choisie égale à  $Q$ , donc  $F = Q$ .

La structure de la fonction  $J$  correspond à un problème d'optimisation quadratique, tel qu'étudié dans le cadre du cours sur l'optimisation sans contrainte. En effet, il s'agit de minimiser une somme pondérée d'erreurs quadratiques entre la trajectoire suivie  $C_d X(k)$  et la trajectoire de référence  $r(k)$ , ainsi que des efforts de commande  $U(k)$ .

Cependant, plutôt que d'utiliser des méthodes générales comme la descente de gradient ou la méthode de Newton, la solution repose ici sur l'*algorithme récursif de Riccati*.

La fonction  $J$  est une fonction quadratique strictement convexe, car les matrices de pondération  $Q$  et  $R$  sont définies positives ( $Q, R \succ 0$ ), ce qui garantit l'existence et l'unicité de la solution optimale.

L'algorithme récursif de Riccati permet de résoudre ce problème efficacement sans avoir à inverser une grande matrice globale. Il procède par rétropropagation à partir de l'instant final  $k_f$ , ce qui est bien adapté aux problèmes de commande optimale sur un horizon fini.

A l'aide de l'équation de Riccati on obtient l'équation permettant de trouver la solutions du problème discrétisé :

$$\begin{aligned}
 P(k) &= A_d^T P(k+1) [I + E P(k+1)]^{-1} A_d + V \\
 V &= C_d^T Q C_d \\
 E &= B_d R^{-1} B_d^T \\
 g(k) &= [A_d^T - A_d^T P(k+1) [I + E P(k+1)]^{-1} E] g(k+1) + C_d^T Q r(k) \\
 \bar{X}(k+1) &= [A_d - B_d L(k)] \bar{X}(k) + B_d L_g(k) g(k+1) \\
 L(k) &= [R + B_d^T P(k+1) B_d]^{-1} B_d^T P(k+1) A_d \\
 L_g(k) &= [R + B_d^T P(k+1) B_d]^{-1} B_d^T \\
 \bar{U}(k) &= -L(k) \bar{X}(k) + L_g(k) g(k+1)
 \end{aligned} \tag{11}$$

Avec comme conditions finales :

$$\begin{aligned}
 P(k_f) &= C_d^T F C_d \\
 g(k_f) &= C_d^T F r(k_f)
 \end{aligned} \tag{13}$$

où  $Q$  et  $R$  sont respectivement les matrices de pondération des erreurs sur les états et des efforts de commande.

$$Q = \text{diag}(100, 50, 10, 5, 0, 0, 100, 1, 100, 1, 1000, 0.1) \quad ; \quad R = \text{diag}(10, 0, 0, 0) \tag{9}$$

De part les hypothèses faites, les angles doivent être contraints près de 0

$$-20^\circ < \phi < 20^\circ \quad ; \quad -20^\circ < \theta < 20^\circ \quad ; \quad -20^\circ < \psi < 20^\circ \tag{10}$$

```

using Pkg
Pkg.activate("projet_env")
Pkg.add("Plots")
Pkg.add("JuMP")
Pkg.add("Ipopt")
Pkg.add("COSMO")
Pkg.add("IterativeSolvers")
Pkg.add("LinearMaps")
Pkg.add("OSQP")
Pkg.add("MathOptInterface")

```

```

Activating new project at `C:\Users\drumj\OneDrive\Documents\Joey\Université\Polymt1\Session 5 - Été
Resolving package versions...

```

```

Updating `C:\Users\drumj\OneDrive\Documents\Joey\Université\Polymtl\Session 5 - Été 2025\MTH8404\Pr
[91a5bcdd] + Plots v1.40.16
Updating `C:\Users\drumj\OneDrive\Documents\Joey\Université\Polymtl\Session 5 - Été 2025\MTH8404\Pr
[66dad0bd] + AliasTables v1.1.3
[d1d4a3ce] + BitFlags v0.1.9
[944b1d66] + CodecZlib v0.7.8
[35d6a980] + ColorSchemes v3.29.0
[3da002f7] + ColorTypes v0.12.1
[c3611d14] + ColorVectorSpace v0.11.0
[5ae59095] + Colors v0.13.1
[34da2185] + Compat v4.17.0
[f0e56b4a] + ConcurrentUtilities v2.5.0
[d38c429a] + Contour v0.6.3
[9a962f9c] + DataAPI v1.16.0
[864edb3b] + DataStructures v0.18.22
[8bb1440f] + DelimitedFiles v1.9.1
[ffbed154] + DocStringExtensions v0.9.5
[460bff9d] + ExceptionUnwrapping v0.1.11
[c87230d0] + FFMPEG v0.4.2
[53c48c17] + FixedPointNumbers v0.8.5
[1fa38f19] + Format v1.3.7
[28b8d3ca] + GR v0.73.17
[42e2da0e] + Grisu v1.0.2
[cd3eb016] + HTTP v1.10.17
[92d709cd] + IrrationalConstants v0.2.4
[1019f520] + JLFzf v0.1.11
[692b3bcd] + JLLWrappers v1.7.0
[682c06a0] + JSON v0.21.4
[b964fa9f] + LaTeXStrings v1.4.0
[23fbe1c1] + Latexify v0.16.8
[2ab3a3ac] + LogExpFunctions v0.3.29
[e6f89c97] + LoggingExtras v1.1.0
[1914dd2f] + MacroTools v0.5.16
[739be429] + MbedTLS v1.1.9
[442fdcdd] + Measures v0.3.2
[e1d29d7a] + Missings v1.2.0
[77ba4419] + NaNMath v1.1.3
[4d8831e6] + OpenSSL v1.5.0
[bac558e1] + OrderedCollections v1.8.1
[69de0a69] + Parsers v2.8.3
[ccf2f8ad] + PlotThemes v3.3.0
[995b91a9] + PlotUtils v1.4.3
[91a5bcdd] + Plots v1.40.16
[aea7be01] + PrecompileTools v1.2.1
[21216c6a] + Preferences v1.4.3
[43287f4e] + PtrArrays v1.3.0
[3cdcf5f2] + RecipesBase v1.3.4
[01d81517] + RecipesPipeline v0.6.12
[189a3867] + Reexport v1.2.2
[05181044] + RelocatableFolders v1.0.1
[ae029012] + Requires v1.3.1
[6c6a2e73] + Scratch v1.3.0
[992d4aef] + Showoff v1.0.3
[777ac1f9] + SimpleBufferStream v1.2.0

```

[a2af1166] + SortingAlgorithms v1.2.1  
 [860ef19b] + StableRNGs v1.0.3  
 [82ae8749] + StatsAPI v1.7.1  
 [2913bbd2] + StatsBase v0.34.5  
 [62fd8b95] + TensorCore v0.1.1  
 [3bb67fe8] + TranscodingStreams v0.11.3  
 [5c2747f8] + URIs v1.6.1  
 [1cfade01] + UnicodeFun v0.4.1  
 [1986cc42] + Unitful v1.23.1  
 [45397f5d] + UnitfulLatexify v1.7.0  
 [41fe7b60] + Unzip v0.2.0  
 [6e34b625] + Bzip2\_jll v1.0.9+0  
 [83423d85] + Cairo\_jll v1.18.5+0  
 [ee1fde0b] + Dbus\_jll v1.16.2+0  
 [2702e6a9] + EpollShim\_jll v0.0.20230411+1  
 [2e619515] + Expat\_jll v2.6.5+0  
 [b22a6f82] + FFMPEG\_jll v4.4.4+1  
 [a3f928ae] + Fontconfig\_jll v2.16.0+0  
 [d7e528f0] + FreeType2\_jll v2.13.4+0  
 [559328eb] + FriBidi\_jll v1.0.17+0  
 [0656b61e] + GLFW\_jll v3.4.0+2  
 [d2c73de3] + GR\_jll v0.73.17+0  
 [b0724c58] + GettextRuntime\_jll v0.22.4+0  
 [7746bdde] + Glib\_jll v2.84.3+0  
 [3b182d85] + Graphite2\_jll v1.3.15+0  
 [2e76f6c2] + HarfBuzz\_jll v8.5.1+0  
 [aacddb02] + JpegTurbo\_jll v3.1.1+0  
 [c1c5ebd0] + LAME\_jll v3.100.3+0  
 [88015f11] + LERC\_jll v4.0.1+0  
 [1d63c593] + LLVMOpenMP\_jll v18.1.8+0  
 [dd4b983a] + LZ0\_jll v2.10.3+0  
 [e9f186c6] + Libffi\_jll v3.4.7+0  
 [7e76a0d4] + Libglvnd\_jll v1.7.1+1  
 [94ce4f54] + Libiconv\_jll v1.18.0+0  
 [4b2f31a3] + Libmount\_jll v2.41.0+0  
 [89763e89] + Libtiff\_jll v4.7.1+0  
 [38a345b3] + Libuuid\_jll v2.41.0+0  
 [e7412a2a] + Ogg\_jll v1.3.6+0  
 [458c3c95] + OpenSSL\_jll v3.5.1+0  
 [91d4177d] + Opus\_jll v1.5.2+0  
 [36c8627f] + Pango\_jll v1.56.3+0  
 [30392449] + Pixman\_jll v0.44.2+0  
 [c0090381] + Qt6Base\_jll v6.8.2+1  
 [629bc702] + Qt6Declarative\_jll v6.8.2+1  
 [ce943373] + Qt6ShaderTools\_jll v6.8.2+1  
 [e99dba38] + Qt6Wayland\_jll v6.8.2+1  
 [a44049a8] + Vulkan\_Loader\_jll v1.3.243+0  
 [a2964d1f] + Wayland\_jll v1.23.1+2  
 [02c8fc9c] + XML2\_jll v2.13.6+1  
 [ffd25f8a] + XZ\_jll v5.8.1+0  
 [f67eecfb] + Xorg\_libICE\_jll v1.1.2+0  
 [c834827a] + Xorg\_libSM\_jll v1.2.6+0  
 [4f6342f7] + Xorg\_libX11\_jll v1.8.12+0  
 [0c0b7dd1] + Xorg\_libXau\_jll v1.0.13+0

[935fb764] + Xorg\_libXcursor\_jll v1.2.4+0  
 [a3789734] + Xorg\_libXdmcp\_jll v1.1.6+0  
 [1082639a] + Xorg\_libXext\_jll v1.3.7+0  
 [d091e8ba] + Xorg\_libXfixes\_jll v6.0.1+0  
 [a51aa0fd] + Xorg\_libXi\_jll v1.8.3+0  
 [d1454406] + Xorg\_libXinerama\_jll v1.1.6+0  
 [ec84b674] + Xorg\_libXrandr\_jll v1.5.5+0  
 [ea2f1a96] + Xorg\_libXrender\_jll v0.9.12+0  
 [c7cfdc94] + Xorg\_libxcb\_jll v1.17.1+0  
 [cc61e674] + Xorg\_libxkbfile\_jll v1.1.3+0  
 [e920d4aa] + Xorg\_xcb\_util\_cursor\_jll v0.1.5+0  
 [12413925] + Xorg\_xcb\_util\_image\_jll v0.4.1+0  
 [2def613f] + Xorg\_xcb\_util\_jll v0.4.1+0  
 [975044d2] + Xorg\_xcb\_util\_keysyms\_jll v0.4.1+0  
 [0d47668e] + Xorg\_xcb\_util\_renderutil\_jll v0.3.10+0  
 [c22f9ab0] + Xorg\_xcb\_util\_wm\_jll v0.4.2+0  
 [35661453] + Xorg\_xkbcomp\_jll v1.4.7+0  
 [33bec58e] + Xorg\_xkeyboard\_config\_jll v2.44.0+0  
 [c5fb5394] + Xorg\_xtrans\_jll v1.6.0+0  
 [3161d3a3] + Zstd\_jll v1.5.7+1  
 [35ca27e7] + eudev\_jll v3.2.14+0  
 [214eeab7] + fzf\_jll v0.61.1+0  
 [a4ae2306] + libaom\_jll v3.11.0+0  
 [0ac62f75] + libass\_jll v0.15.2+0  
 [1183f4f0] + libdecor\_jll v0.2.2+0  
 [2db6ffa8] + libevdev\_jll v1.13.4+0  
 [f638f0a6] + libfdk\_aac\_jll v2.0.4+0  
 [36db933b] + libinput\_jll v1.28.1+0  
 [b53b4c65] + libpng\_jll v1.6.50+0  
 [f27f6e37] + libvorbis\_jll v1.3.8+0  
 [009596ad] + mtdev\_jll v1.1.7+0  
 [1270edf5] + x264\_jll v2021.5.5+0  
 [dffa095f] + x265\_jll v3.5.0+0  
 [d8fb68d0] + xkbcommon\_jll v1.9.2+0  
 [0dad84c5] + ArgTools v1.1.1  
 [56f22d72] + Artifacts  
 [2a0f44e3] + Base64  
 [ade2ca70] + Dates  
 [f43a241f] + Downloads v1.6.0  
 [7b1f6079] + FileWatching  
 [b77e0a4c] + InteractiveUtils  
 [b27032c2] + LibCURL v0.6.4  
 [76f85450] + LibGit2  
 [8f399da3] + Libdl  
 [37e2e46d] + LinearAlgebra  
 [56ddb016] + Logging  
 [d6f4376e] + Markdown  
 [a63ad114] + Mmap  
 [ca575930] + NetworkOptions v1.2.0  
 [44cfe95a] + Pkg v1.10.0  
 [de0858da] + Printf  
 [3fa0cd96] + REPL  
 [9a3f8284] + Random  
 [ea8e919c] + SHA v0.7.0

```

[9e88b42a] + Serialization
[6462fe0b] + Sockets
[2f01184e] + SparseArrays v1.10.0
[10745b16] + Statistics v1.10.0
[fa267f1f] + TOML v1.0.3
[a4e569a6] + Tar v1.10.0
[8dfed614] + Test
[cf7118a7] + UUIDs
[4ec0a83e] + Unicode
[e66e0078] + CompilerSupportLibraries_jll v1.1.1+0
[deac9b47] + LibCURL_jll v8.4.0+0
[e37daf67] + LibGit2_jll v1.6.4+0
[29816b5a] + LibSSH2_jll v1.11.0+1
[c8ffd9c3] + MbedTLS_jll v2.28.2+1
[14a3606d] + MozillaCACerts_jll v2023.1.10
[4536629a] + OpenBLAS_jll v0.3.23+4
[05823500] + OpenLibm_jll v0.8.1+4
[efcefd7] + PCRE2_jll v10.42.0+1
[bea87d4a] + SuiteSparse_jll v7.2.1+1
[83775a58] + Zlib_jll v1.2.13+1
[8e850b90] + libblastrampoline_jll v5.11.0+0
[8e850ede] + nghttp2_jll v1.52.0+1
[3f19e933] + p7zip_jll v17.4.0+2

```

Info Packages marked with have new versions available but compatibility constraints restrict the  
Resolving package versions...

```

Updating `C:\Users\drumj\OneDrive\Documents\Joey\Université\Polymtl\Session 5 - Été 2025\MTH8404\Proje
[4076af6c] + JuMP v1.26.0

```

```

Updating `C:\Users\drumj\OneDrive\Documents\Joey\Université\Polymtl\Session 5 - Été 2025\MTH8404\Proje
[6e4b80f9] + BenchmarkTools v1.6.0
[523fee87] + CodecBzip2 v0.8.5
[bbf7d656] + CommonSubexpressions v0.3.1
[163ba53b] + DiffResults v1.1.0
[b552c78f] + DiffRules v1.15.1
[f6369f11] + ForwardDiff v1.0.1
[0f8b85d8] + JSON3 v1.14.3
[4076af6c] + JuMP v1.26.0
[b8f27783] + MathOptInterface v1.41.0
[d8a4904e] + MutableArithmetics v1.6.4
[276daf66] + SpecialFunctions v2.5.1
[1e83bf80] + StaticArraysCore v1.4.3
[856f2bd8] + StructTypes v1.11.0
[efe28fd5] + OpenSpecFun_jll v0.5.6+0
[9abbd945] + Profile

```

Resolving package versions...

```

Updating `C:\Users\drumj\OneDrive\Documents\Joey\Université\Polymtl\Session 5 - Été 2025\MTH8404\Proje
[b6b21f68] + Ipopt v1.10.6

```

```

Updating `C:\Users\drumj\OneDrive\Documents\Joey\Université\Polymtl\Session 5 - Été 2025\MTH8404\Proje
[b6b21f68] + Ipopt v1.10.6
[ae81ac8f] + ASL_jll v0.1.3+0
[e33a78d0] + Hwloc_jll v2.12.1+0
[9cc047cb] + Ipopt_jll v300.1400.1701+0
[d00139f3] + METIS_jll v5.1.3+0
[d7ed1dd3] + MUMPS_seq_jll v500.800.0+0
[656ef2d0] + OpenBLAS32_jll v0.3.24+0

```



```

[319450e9] + SPRAL_jll v2025.5.20+0
Info Packages marked with have new versions available but compatibility constraints restrict the
Resolving package versions...
Updating `C:\Users\drumj\OneDrive\Documents\Joey\Université\Polymtl\Session 5 - Été 2025\MTH8404\Pr
[1e616198] + COSMO v0.8.9
Updating `C:\Users\drumj\OneDrive\Documents\Joey\Université\Polymtl\Session 5 - Été 2025\MTH8404\Pr
[14f7f29c] + AMD v0.5.3
[1e616198] + COSMO v0.8.9
[bdd8fffe] + COSMOAccelerators v0.1.0
[c8e1da08] + IterTools v1.10.0
[bfc457fd] + QDLDL v0.4.1
[c4a57d5a] + UnsafeArrays v1.0.8
[4607b0f0] + SuiteSparse
Resolving package versions...
Updating `C:\Users\drumj\OneDrive\Documents\Joey\Université\Polymtl\Session 5 - Été 2025\MTH8404\Pr
[42fd0dbc] + IterativeSolvers v0.9.4
Updating `C:\Users\drumj\OneDrive\Documents\Joey\Université\Polymtl\Session 5 - Été 2025\MTH8404\Pr
[42fd0dbc] + IterativeSolvers v0.9.4
Resolving package versions...
Updating `C:\Users\drumj\OneDrive\Documents\Joey\Université\Polymtl\Session 5 - Été 2025\MTH8404\Pr
[7a12625a] + LinearMaps v3.11.4
Updating `C:\Users\drumj\OneDrive\Documents\Joey\Université\Polymtl\Session 5 - Été 2025\MTH8404\Pr
[7a12625a] + LinearMaps v3.11.4
Resolving package versions...
Updating `C:\Users\drumj\OneDrive\Documents\Joey\Université\Polymtl\Session 5 - Été 2025\MTH8404\Pr
[ab2f91bb] + OSQP v0.8.1
Updating `C:\Users\drumj\OneDrive\Documents\Joey\Université\Polymtl\Session 5 - Été 2025\MTH8404\Pr
[ab2f91bb] + OSQP v0.8.1
[9c4f68bf] + OSQP_jll v0.600.200+0
Resolving package versions...
Updating `C:\Users\drumj\OneDrive\Documents\Joey\Université\Polymtl\Session 5 - Été 2025\MTH8404\Pr
[b8f27783] + MathOptInterface v1.41.0
No Changes to `C:\Users\drumj\OneDrive\Documents\Joey\Université\Polymtl\Session 5 - Été 2025\MTH8404

```

## Partie 2 : reproduction des résultats

Fonctionnement de l'algorithme :

L'algorithme commence à la date finale  $k_f$ , où les conditions terminales sont imposées, puis il effectue une récursion arrière (backward) sur les matrices  $P(k)$  et  $g(k)$  jusqu'à l'instant initial  $k_0$ .

Une fois ces matrices calculées, une simulation en avant (forward) permet d'estimer la trajectoire optimale  $\bar{X}(k)$  et les commandes optimales  $\bar{U}(k)$  en appliquant les gains  $L(k)$  et  $L_g(k)$  à chaque pas de temps.

```

using LinearAlgebra
using Plots

gr() # Choix du backend GR pour les graphiques (fiable pour l'export PDF)
g = 9.81 # gravité

# Dimensions système
n = 12 # nombre d'états du drone
m = 4 # nombre d'entrées de contrôle
T = 6000 # durée de la simulation (en pas de temps)
Ts = 0.01 # pas de temps (10 ms)

```

```

# Paramètres physiques du drone
mass = 0.5
Ix, Iy, Iz = 0.0023, 0.0023, 0.004 # moments d'inertie autour des axes

# Initialisation des matrices A et B pour le modèle linéarisé
A = zeros(n, n)
B = zeros(n, m)

# Mise en place de la dynamique : intégrateurs entre position/vitesse/accélération
for i in 1:2:11
    A[i, i+1] = 1.0 # lien entre position et vitesse (ex: x, vx)
end

# Couplage entre angles et accélérations selon la dynamique du drone
A[8, 3] = g # acc x dépend de
A[10, 1] = -g # acc y dépend de

# Entrées de commande (forces / moments)
B[2,2] = 1/Ix # moment autour de x ()
B[4,3] = 1/Iy # moment autour de y ()
B[6,4] = 1/Iz # moment autour de z ()
B[12,1] = 1/mass # poussée verticale (z)

# Discrétisation d'Euler
Ad = I + Ts * A
Bd = Ts * B

# Matrices de coût pour le LQR
Q = Diagonal([100.0, 50.0, 10.0, 5.0, 0.0, 0.0, 100.0, 1.0, 100.0, 1.0, 1000.0, 0.1])
R = Diagonal([10.0, 1e-3, 1e-3, 1e-3]) # pondération des efforts de contrôle
F = Q # coût terminal

# =====
# Définition des trajectoires désirées (référence)
# =====
r = zeros(n, T+1)

# x : montée puis descente linéaire
for k in 1:T÷2
    r[7, k] = 40.0 * (k / (T÷2))
end
for k in T÷2+1:T+1
    r[7, k] = 40.0 * (1 - (k - T÷2) / (T÷2))
end

# y : reste à 0, saute brusquement à 10 après la moitié du temps
for k in T÷2+1:T+1
    r[9, k] = 10.0
end

# z : montée douce de 0 à 20 entre 15s et 30s, puis redescende
dt = Ts
t15 = Int(15 / dt)

```

```

t30 = Int(30 / dt)

for k in 1:t15
    r[11, k] = 0.0
end
for k in t15+1:t30
    r[11, k] = 20.0 * ((k - t15) / (t30 - t15))
end
for k in t30+1:T+1
    r[11, k] = 20.0 * (1 - (k - t30) / (T + 1 - t30))
end

# =====
# Calcul backward des gains LQR (formulation récursive de Riccati)
# =====

# Allocation des vecteurs/matrices pour les gains
P = Vector{Matrix{Float64}}(undef, T+1) # matrices de Riccati
G = Vector{Vector{Float64}}(undef, T+1) # terme d'offset (trajectoire r)
L = Vector{Matrix{Float64}}(undef, T)   # gain de feedback état
Lg = Vector{Matrix{Float64}}(undef, T)  # gain associé à la référence

# Conditions terminales
C = I(n) # matrice d'observation (identité ici)
P[T+1] = C' * Q * C
G[T+1] = C' * Q * r[:, T+1]

# Recursion backward Riccati
for k in T:-1:1
    E = Bd * inv(R) * Bd'
    V = C' * Q * C
    invIplusE = inv(I + E * P[k+1])

    P[k] = Ad' * P[k+1] * invIplusE * Ad + V
    G[k] = (Ad' - Ad' * P[k+1] * invIplusE * E) * G[k+1] + C' * Q * r[:, k]
end

# =====
# Simulation forward du système contrôlé
# =====

X = zeros(n, T+1) # états
U = zeros(m, T)   # commandes

X[:, 1] = zeros(n) # état initial

for k in 1:T
    # Calcul des gains à l'instant k
    L[k] = inv(R + Bd' * P[k+1] * Bd) * Bd' * P[k+1] * Ad
    Lg[k] = inv(R + Bd' * P[k+1] * Bd) * Bd'

    # Calcul de la commande optimale
    U[:, k] = -L[k] * X[:, k] + Lg[k] * G[k+1]
end

```

```

# Propagation de l'état
X[:,k+1] = Ad * X[:,k] + Bd * U[:,k]

# Contraintes physiques : angles d'attitude max (20°)
phi_max = deg2rad(20)
theta_max = deg2rad(20)
psi_max = deg2rad(20)

# contraintes des angles
X[1,k+1] = clamp(X[1,k+1], -phi_max, phi_max)
X[3,k+1] = clamp(X[3,k+1], -theta_max, theta_max)
X[5,k+1] = clamp(X[5,k+1], -psi_max, psi_max)
end

# =====
# Visualisation de la trajectoire et de l'erreur
# =====

z = X[11, :] # position verticale
x = X[7, :] # position x
y = X[9, :] # position y

# Erreurs de suivi
e_z = z .- r[11, :]
e_x = x .- r[7, :]
e_y = y .- r[9, :]
z = X[11, :] # position verticale
x = X[7, :] # position x
y = X[9, :] # position y

# Erreurs de suivi
e_z = z .- r[11, :]
e_x = x .- r[7, :]
e_y = y .- r[9, :]

# Références
r_z = r[11, :]
r_x = r[7, :]
r_y = r[9, :]

t = Ts .* (0:T) # temps réel en secondes

# Création du layout 3 lignes x 2 colonnes
plt = plot(layout = (3, 2), size=(1000, 800))

# Trajectoire x
plot!(plt[1], t, x, lw=2, label="x (suivie)", color=:blue)
plot!(plt[1], t, r_x, lw=2, label="x (référence)", linestyle=:dash, color=:red)
plot!(plt[1], title="Trajectoire en x", xlabel="Temps (s)", ylabel="x (m)", legend=:bottomright, grid=true)

# Erreur x
plot!(plt[2], t, e_x, lw=2, label="Erreur x", color=:purple)
plot!(plt[2], title="Erreur de suivi (x)", xlabel="Temps (s)", ylabel="Erreur (m)", legend=:topright, grid=true)

```

```

# Trajectoire y
plot!(plt[3], t, y, lw=2, label="y (suivie)", color=:blue)
plot!(plt[3], t, r_y, lw=2, label="y (référence)", linestyle=:dash, color=:red)
plot!(plt[3], title="Trajectoire en y", xlabel="Temps (s)", ylabel="y (m)", legend=:bottomright, grid=true)

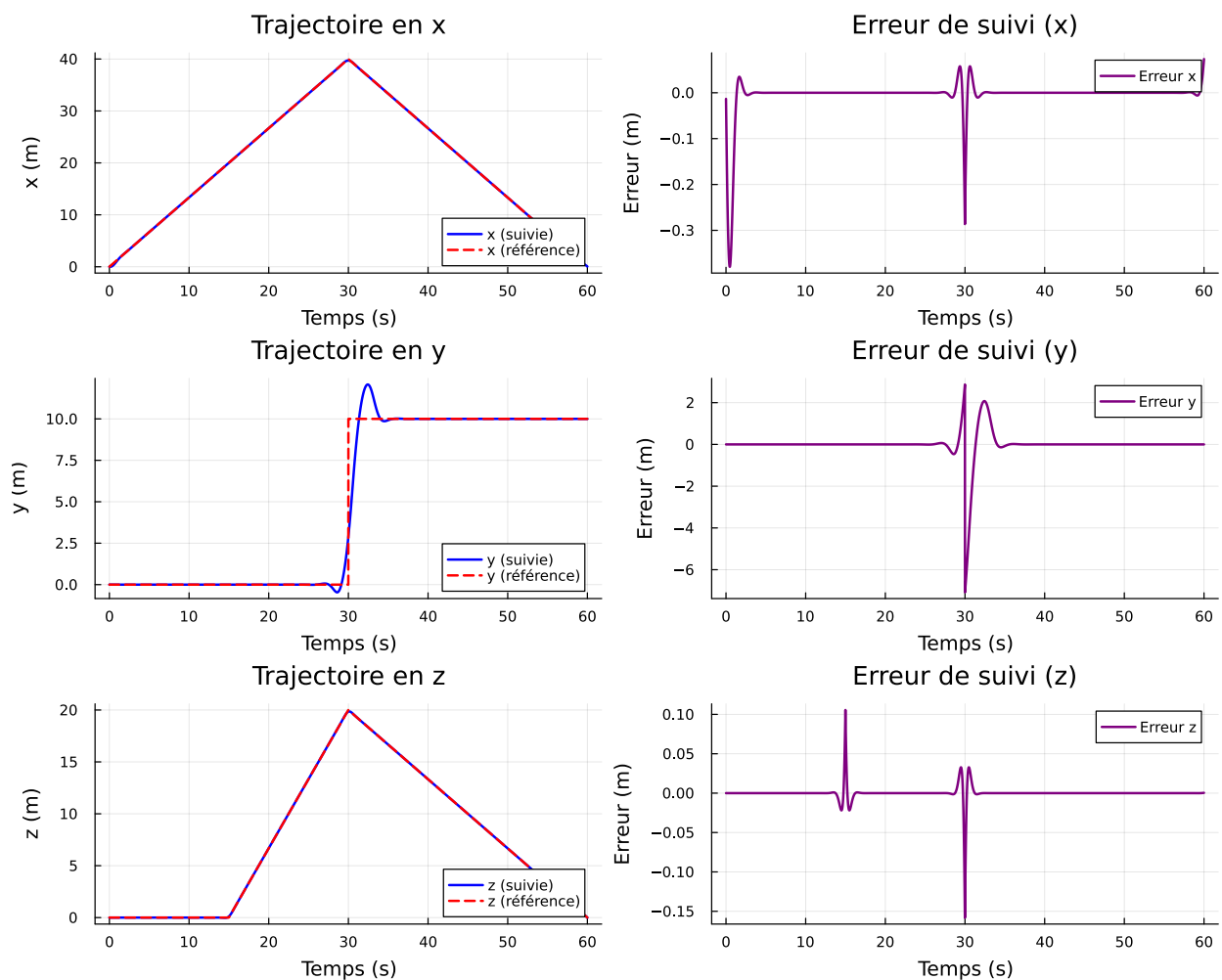
# Erreur y
plot!(plt[4], t, e_y, lw=2, label="Erreur y", color=:purple)
plot!(plt[4], title="Erreur de suivi (y)", xlabel="Temps (s)", ylabel="Erreur (m)", legend=:topright, grid=true)

# Trajectoire z
plot!(plt[5], t, z, lw=2, label="z (suivie)", color=:blue)
plot!(plt[5], t, r_z, lw=2, label="z (référence)", linestyle=:dash, color=:red)
plot!(plt[5], title="Trajectoire en z", xlabel="Temps (s)", ylabel="z (m)", legend=:bottomright, grid=true)

# Erreur z
plot!(plt[6], t, e_z, lw=2, label="Erreur z", color=:purple)
plot!(plt[6], title="Erreur de suivi (z)", xlabel="Temps (s)", ylabel="Erreur (m)", legend=:topright, grid=true)

# Affichage
display(plt)

```



### Partie 3

Le but de cette section est de comparer les résultats obtenus dans la Partie 2 avec des résultats venant de stratégies plus générique comme par exemple IPOPT. Le modèle est construit à partir de la librairie JuMP.

```
using JuMP, Ipopt

x0 = zeros(n)
x0[11] = 1.0          # altitude 1 m en hover

# Modèle JuMP « all-at-once »
model = Model(Ipopt.Optimizer)
set_optimizer_attribute(model, "print_level", 0)

@variable(model, X[1:n, 0:T])      # états
@variable(model, U[1:m, 0:T-1])    # commandes

# - dynamique linéaire
for k in 0:T-1
    @constraint(model, X[:, k+1] .== Ad * X[:, k] + Bd * U[:, k])
end

# - CONTRAINTE d'état initial (= hover)
@constraint(model, X[:, 0] .== x0)

# - bornes (angles  $\pm 20^\circ$ , poussée 0-2 mg)
deg20 = deg2rad(20.0)
@constraint(model, -deg20 .<= X[1, :] .<= deg20) #
@constraint(model, -deg20 .<= X[3, :] .<= deg20) #
@constraint(model, -deg20 .<= X[5, :] .<= deg20) #
@constraint(model, 0 .<= U[1, :] .<= 2 * mass * g)

# - coût
@expression(model, running_cost,
    sum( (X[:, k] - r[:, k+1])' * Q * (X[:, k] - r[:, k+1]) +
        U[:, k]' * R * U[:, k] for k in 0:T-1) )

@expression(model, terminal_cost,
    (X[:, T] - r[:, T+1])' * F * (X[:, T] - r[:, T+1]) )

@objective(model, Min, 0.5 * running_cost + 0.5 * terminal_cost)

# - point initial
set_start_value.(X[:, 0], x0)
set_start_value.(U, 0.0)

optimize!(model)
println("Status : ", termination_status(model))
println("Objective value : ", objective_value(model))

# =====
# Visualisation de la trajectoire et de l'erreur
# =====
```

```

solutionX = Array(value.(X))

z = solutionX[11, :] # position verticale
x = solutionX[7, :]  # position x
y = solutionX[9, :]  # position y

# Erreurs de suivi
e_z = z .- r[11, :]
e_x = x .- r[7, :]
e_y = y .- r[9, :]
z = solutionX[11, :] # position verticale
x = solutionX[7, :]  # position x
y = solutionX[9, :]  # position y

# Erreurs de suivi
e_z = z .- r[11, :]
e_x = x .- r[7, :]
e_y = y .- r[9, :]

# Références
r_z = r[11, :]
r_x = r[7, :]
r_y = r[9, :]

t = Ts .* (0:T) # temps réel en secondes

# Création du layout 3 lignes × 2 colonnes
plt2 = plot(layout = (3, 2), size=(1000, 800))

# Trajectoire x
plot!(plt2[1], t, x, lw=2, label="x (suivie)", color=:blue)
plot!(plt2[1], t, r_x, lw=2, label="x (référence)", linestyle=:dash, color=:red)
plot!(plt2[1], title="Trajectoire en x", xlabel="Temps (s)", ylabel="x (m)", legend=:bottomright, grid=true)

# Erreur x
plot!(plt2[2], t, e_x, lw=2, label="Erreur x", color=:purple)
plot!(plt2[2], title="Erreur de suivi (x)", xlabel="Temps (s)", ylabel="Erreur (m)", legend=:topright, grid=true)

# Trajectoire y
plot!(plt2[3], t, y, lw=2, label="y (suivie)", color=:blue)
plot!(plt2[3], t, r_y, lw=2, label="y (référence)", linestyle=:dash, color=:red)
plot!(plt2[3], title="Trajectoire en y", xlabel="Temps (s)", ylabel="y (m)", legend=:bottomright, grid=true)

# Erreur y
plot!(plt2[4], t, e_y, lw=2, label="Erreur y", color=:purple)
plot!(plt2[4], title="Erreur de suivi (y)", xlabel="Temps (s)", ylabel="Erreur (m)", legend=:topright, grid=true)

# Trajectoire z
plot!(plt2[5], t, z, lw=2, label="z (suivie)", color=:blue)
plot!(plt2[5], t, r_z, lw=2, label="z (référence)", linestyle=:dash, color=:red)
plot!(plt2[5], title="Trajectoire en z", xlabel="Temps (s)", ylabel="z (m)", legend=:bottomright, grid=true)

# Erreur z

```

```

plot!(plt2[6], t, e_z, lw=2, label="Erreur z", color=:purple)
plot!(plt2[6], title="Erreur de suivi (z)", xlabel="Temps (s)", label="Erreur (m)", legend=:topright, g

# Affichage
display(plt2)

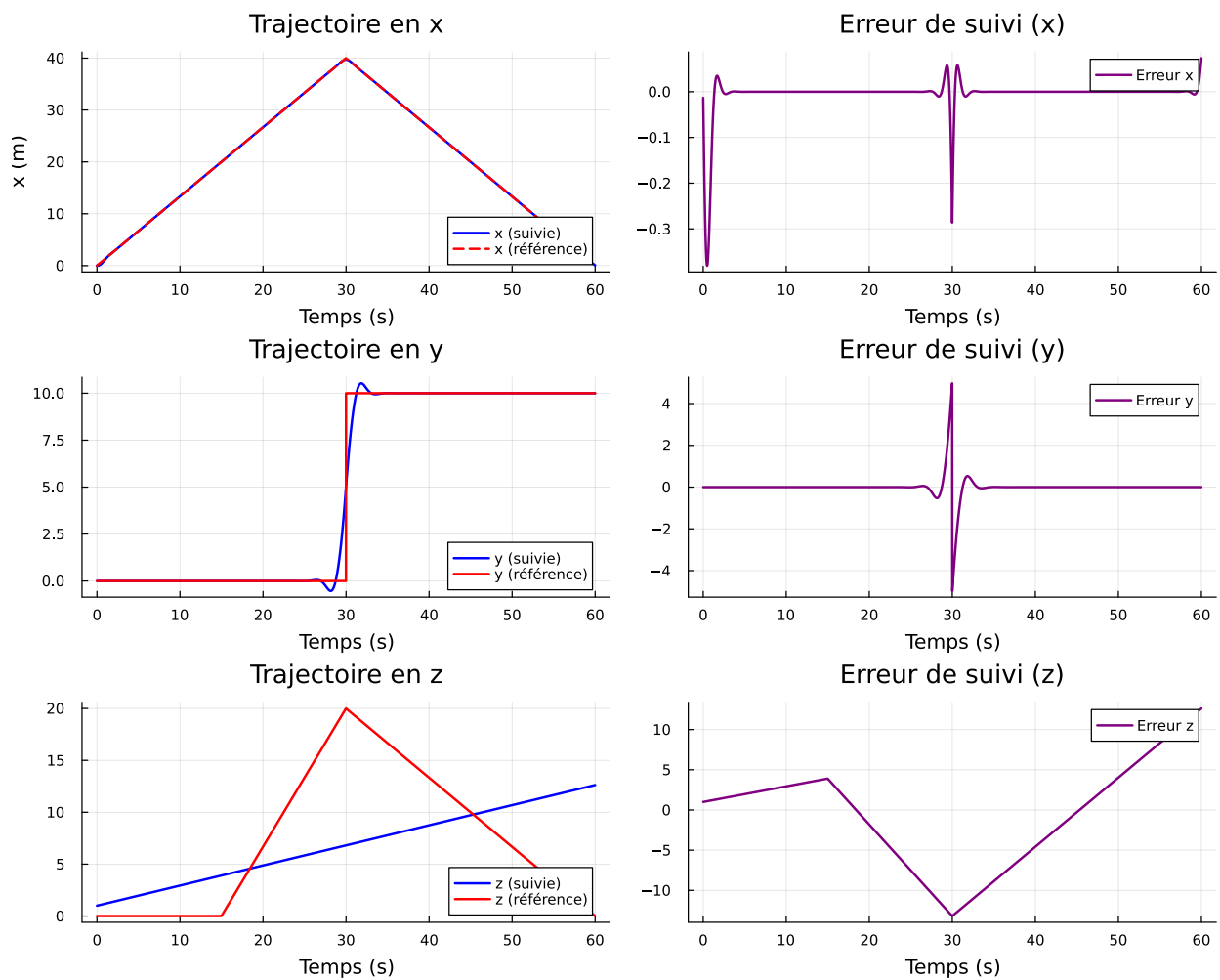
```

\*\*\*\*\*

This program contains Ipopt, a library for large-scale nonlinear optimization.  
 Ipopt is released as open source code under the Eclipse Public License (EPL).  
 For more information visit <https://github.com/coin-or/Ipopt>

\*\*\*\*\*

Status : LOCALLY\_SOLVED  
 Objective value : 1.230511402197558e8



## Analyse des résultats

La méthode IPOPT converge belle et bien pour ce problème et on obtient une valeur de l'objectif de 1.230511402197558e8. Il est possible de faire mieux avec des méthodes utilisant les propriétés du problème. Deux exemples de ces méthodes seront visités dans la prochaine section. Notons que la convergence a été



atteinte en 4000 itérations pour un temps de 1.63e+001 secondes. On observe sur les graphiques que la solution proposée est efficace, en ce sens que le trajet à suivre est assez bien respectés. Notons que les erreurs sont plus grandes autour des points où il y a un changement dans la dérivée. Notre approximation étant une fonction lisse, cela n'est pas étonnant. L'erreur est aussi grande au temps final, ce qui peut être un problème dépendamment des applications. Un grand déplacement programmé comme une somme de petites trajectoires pourrait facilement accumuler des erreurs.

## Partie 4

### Résolution avec le solveur COSMO

Le solveur COSMO implémente le **Conic operator splitting method**, qui est particulièrement adaptée pour résoudre de larges problèmes d'optimisation convexe et conique dont l'objectif est donné par une fonction quadratique. Il s'agit donc d'une méthode adaptée au problème monolithique creux. Le modèle est encore une fois construit à l'aide de la librairie JuMP.

```
using COSMO, IterativeSolvers, LinearMaps, OSQP, MathOptInterface

model3 = Model(COSMO.Optimizer)

@variable(model3, X[1:n, 0:T])          # états
@variable(model3, U[1:m, 0:T-1])        # commandes

# - dynamique linéaire
for k in 0:T-1
    @constraint(model3, X[:, k+1] .== Ad * X[:, k] + Bd * U[:, k])
end

# - CONTRAINTE d'état initial (= hover)
@constraint(model3, X[:, 0] .== x0)

# - bornes (angles ±20°, poussée 0-2 mg)
deg20 = deg2rad(20.0)
@constraint(model3, -deg20 .<= X[1, :] .<= deg20) #
@constraint(model3, -deg20 .<= X[3, :] .<= deg20) #
@constraint(model3, -deg20 .<= X[5, :] .<= deg20) #
@constraint(model3, 0 .<= U[1, :] .<= 2 * mass * g)

# - coût
@expression(model3, running_cost,
    sum( (X[:, k] - r[:, k+1])' * Q * (X[:, k] - r[:, k+1]) +
        U[:, k]' * R * U[:, k] for k in 0:T-1) )

@expression(model3, terminal_cost,
    (X[:, T] - r[:, T+1])' * F * (X[:, T] - r[:, T+1]) )

@objective(model3, Min, 0.5 * running_cost + 0.5 * terminal_cost)

# - point initial
set_start_value.(X[:, 0], x0)
set_start_value.(U, 0.0)

optimize!(model3)
println("Status : ", termination_status(model3))
```

```

println("Objective value : ", objective_value(model3))

# =====
# Visualisation de la trajectoire et de l'erreur
# =====

solutionX = Array(value.(X))

z = solutionX[11, :] # position verticale
x = solutionX[7, :]  # position x
y = solutionX[9, :]  # position y

# Erreurs de suivi
e_z = z .- r[11, :]
e_x = x .- r[7, :]
e_y = y .- r[9, :]
z = solutionX[11, :] # position verticale
x = solutionX[7, :]  # position x
y = solutionX[9, :]  # position y

# Erreurs de suivi
e_z = z .- r[11, :]
e_x = x .- r[7, :]
e_y = y .- r[9, :]

# Références
r_z = r[11, :]
r_x = r[7, :]
r_y = r[9, :]

t = Ts .* (0:T) # temps réel en secondes

# Création du layout 3 lignes × 2 colonnes
plt3 = plot(layout = (3, 2), size=(1000, 800))

# Trajectoire x
plot!(plt3[1], t, x, lw=2, label="x (suivie)", color=:blue)
plot!(plt3[1], t, r_x, lw=2, label="x (référence)", linestyle=:dash, color=:red)
plot!(plt3[1], title="Trajectoire en x", xlabel="Temps (s)", ylabel="x (m)", legend=:bottomright, grid=:true)

# Erreur x
plot!(plt3[2], t, e_x, lw=2, label="Erreur x", color=:purple)
plot!(plt3[2], title="Erreur de suivi (x)", xlabel="Temps (s)", ylabel="Erreur (m)", legend=:topright, grid=:true)

# Trajectoire y
plot!(plt3[3], t, y, lw=2, label="y (suivie)", color=:blue)
plot!(plt3[3], t, r_y, lw=2, label="y (référence)", linestyle=:dash, color=:red)
plot!(plt3[3], title="Trajectoire en y", xlabel="Temps (s)", ylabel="y (m)", legend=:bottomright, grid=:true)

# Erreur y
plot!(plt3[4], t, e_y, lw=2, label="Erreur y", color=:purple)
plot!(plt3[4], title="Erreur de suivi (y)", xlabel="Temps (s)", ylabel="Erreur (m)", legend=:topright, grid=:true)

```

```

# Trajectoire z
plot!(plt3[5], t, z, lw=2, label="z (suivie)", color=:blue)
plot!(plt3[5], t, r_z, lw=2, label="z (référence)", inestyle=:dash, color=:red)
plot!(plt3[5], title="Trajectoire en z", xlabel="Temps (s)", label="z (m)", legend=:bottomright, grid=true)

# Erreur z
plot!(plt3[6], t, e_z, lw=2, label="Erreur z", color=:purple)
plot!(plt3[6], title="Erreur de suivi (z)", xlabel="Temps (s)", label="Erreur (m)", legend=:topright, grid=true)

# Affichage
display(plt3)

```

-----

COSMO v0.8.9 - A Quadratic Objective Conic Solver  
Michael Garstka  
University of Oxford, 2017 - 2022

-----

```

Problem:  x  R^{96012},
          constraints: A  R^{96015x96012} (240015 nnz),
          matrix size to factor: 192027x192027,
          Floating-point precision: Float64
Sets:     ZeroSet of dim: 72012
          Box of dim: 24003
Settings:  _abs = 1.0e-05, _rel = 1.0e-05,
          _prim_inf = 1.0e-04, _dual_inf = 1.0e-04,
          = 0.1, = 1e-06, = 1.6,
          max_iter = 5000,
          scaling iter = 10 (on),
          check termination every 25 iter,
          check infeasibility every 40 iter,
          KKT system solver: COSMO.QdldlKKTsolver
Acc:       Anderson Type2{QRDecomp},
          Memory size = 15, RestartedMemory,
          Safeguarded: true, tol: 2.0
Setup Time: 261706.53ms

```

```

Iter:   Objective:  Primal Res:  Dual Res:   Rho:
1      -3.0419e+08  5.9254e-01  1.1996e+04  1.0000e-01
25     -4.7490e+08  7.0087e-02  8.1091e+01  1.0000e-01
50     -4.7488e+08  2.8388e-02  3.7890e+01  1.0000e-01
75     -4.7486e+08  5.9594e-02  7.4153e+01  1.0000e-01
100    -4.7485e+08  3.0238e-02  2.7728e+01  1.0000e-01
125    -4.7483e+08  2.2684e-02  1.8061e+01  1.0000e-01
150    -4.7481e+08  4.1983e-02  4.3684e+01  1.0000e-01
175    -4.7479e+08  2.0286e-02  1.2869e+01  1.0000e-01
200    -4.7478e+08  1.9465e-02  1.0567e+01  1.0000e-01
225    -4.7476e+08  1.8697e-02  2.8641e+01  1.0000e-01
250    -4.7474e+08  1.8072e-02  1.1652e+01  1.0000e-01
275    -4.7472e+08  1.7513e-02  8.5183e+00  1.0000e-01
300    -4.7471e+08  2.0644e-02  1.9313e+01  1.0000e-01
325    -4.7469e+08  1.6534e-02  1.1598e+01  1.0000e-01
350    -4.7467e+08  1.6142e-02  6.4713e+00  1.0000e-01

```

375	-4.7466e+08	2.2601e-02	2.6797e+01	1.0000e-01
400	-4.7464e+08	1.6153e-02	1.3472e+01	1.0000e-01
425	-4.7462e+08	1.5080e-02	4.2927e+00	1.0000e-01
450	-4.7461e+08	1.5490e-02	1.8723e+01	1.0000e-01
475	-4.7459e+08	2.6300e-02	2.5836e+01	1.0000e-01
500	-4.7457e+08	1.4261e-02	6.3697e+00	1.0000e-01
525	-4.7456e+08	1.4032e-02	1.4372e+01	1.0000e-01
550	-4.7454e+08	1.3809e-02	6.7002e+00	1.0000e-01
575	-4.7452e+08	1.3582e-02	3.7849e+00	1.0000e-01
600	-4.7451e+08	2.1539e-02	2.1634e+01	1.0000e-01
625	-4.7450e+08	1.8609e-02	1.8401e+01	1.0000e-01
650	-4.7448e+08	1.3037e-02	5.1720e+00	1.0000e-01
675	-4.7446e+08	1.2869e-02	2.8510e+00	1.0000e-01
700	-4.7445e+08	1.2732e-02	2.9929e+00	1.0000e-01
725	-4.7443e+08	1.2563e-02	4.6681e+00	1.0000e-01
750	-4.7442e+08	2.1020e-02	2.2164e+01	1.0000e-01
775	-4.7440e+08	1.2257e-02	1.0756e+01	1.0000e-01
800	-4.7438e+08	1.2120e-02	4.2650e+00	1.0000e-01
825	-4.7437e+08	1.2008e-02	5.4189e+00	1.0000e-01
850	-4.7435e+08	1.1882e-02	2.9296e+00	1.0000e-01
875	-4.7433e+08	1.1751e-02	3.6104e+00	1.0000e-01
900	-4.7431e+08	1.1637e-02	7.0025e+00	1.0000e-01
925	-4.7430e+08	1.1546e-02	6.2231e+00	1.0000e-01
950	-4.7428e+08	1.1397e-02	4.2443e+00	1.0000e-01
975	-4.7427e+08	1.1317e-02	9.1416e+00	1.0000e-01
1000	-4.7425e+08	1.1200e-02	2.4566e+00	1.0000e-01
1025	-4.7423e+08	1.1102e-02	3.6596e+00	1.0000e-01
1050	-4.7422e+08	1.1450e-02	1.5382e+01	1.0000e-01
1075	-4.7420e+08	1.0921e-02	8.8709e+00	1.0000e-01
1100	-4.7418e+08	1.0834e-02	2.7993e+00	1.0000e-01
1125	-4.7417e+08	1.0751e-02	7.5572e+00	1.0000e-01
1150	-4.7415e+08	1.0671e-02	7.2435e+00	1.0000e-01
1175	-4.7413e+08	1.0585e-02	2.8292e+00	1.0000e-01
1200	-4.7411e+08	1.6458e-02	1.7484e+01	1.0000e-01
1225	-4.7409e+08	1.0404e-02	1.0475e+01	1.0000e-01
1250	-4.7407e+08	1.0307e-02	3.9028e+00	1.0000e-01
1275	-4.7406e+08	4.1333e-02	4.5486e+01	1.0000e-01
1300	-4.7405e+08	1.0195e-02	1.9533e+00	1.0000e-01
1325	-4.7403e+08	1.0131e-02	2.2815e+00	1.0000e-01
1350	-4.7401e+08	1.0064e-02	1.0019e+01	1.0000e-01
1375	-4.7400e+08	1.0005e-02	9.9549e+00	1.0000e-01
1400	-4.7399e+08	9.9518e-03	1.8921e+00	1.0000e-01
1425	-4.7396e+08	9.8590e-03	7.2180e+00	5.2864e-01
1450	-4.7390e+08	9.6373e-03	3.0922e+01	5.2864e-01
1475	-4.7383e+08	1.3864e-02	1.9704e+01	5.2864e-01
1500	-4.7378e+08	9.2494e-03	2.1819e+01	5.2864e-01
1525	-4.7373e+08	9.1326e-03	1.8383e+01	5.2864e-01
1550	-4.7368e+08	4.0465e-02	2.4515e+01	5.2864e-01
1575	-4.7363e+08	8.8730e-03	2.5034e+01	5.2864e-01
1600	-4.7358e+08	8.7522e-03	2.7509e+01	5.2864e-01
1625	-4.7353e+08	1.3568e-02	2.1250e+01	5.2864e-01
1650	-4.7347e+08	1.5499e-02	7.8815e+01	5.2864e-01
1675	-4.7342e+08	8.4098e-03	3.3835e+01	5.2864e-01
1700	-4.7337e+08	1.0430e-01	2.9334e+01	5.2864e-01

1725	-4.7332e+08	3.5574e-02	2.2246e+02	5.2864e-01
1750	-4.7328e+08	8.1571e-03	2.6313e+01	5.2864e-01
1775	-4.7321e+08	5.7008e-02	3.4483e+01	5.2864e-01
1800	-4.7316e+08	4.0829e-02	2.4934e+02	5.2864e-01
1825	-4.7312e+08	2.6429e-02	1.2909e+02	5.2864e-01
1850	-4.7306e+08	3.3035e-02	3.0015e+01	5.2864e-01
1875	-4.7302e+08	7.7855e-03	2.8107e+01	5.2864e-01
1900	-4.7297e+08	7.7209e-03	3.2275e+01	5.2864e-01
1925	-4.7291e+08	7.0269e-02	3.3782e+01	5.2864e-01
1950	-4.7287e+08	1.6950e-02	1.0702e+02	5.2864e-01
1975	-4.7281e+08	7.5353e-03	2.9040e+01	5.2864e-01
2000	-4.7276e+08	2.6158e-02	3.2266e+01	5.2864e-01
2025	-4.7270e+08	7.4108e-03	3.7005e+01	5.2864e-01
2050	-4.7265e+08	7.5872e-03	3.5083e+01	5.2864e-01
2075	-4.7260e+08	1.6493e-02	2.7491e+01	5.2864e-01
2100	-4.7254e+08	1.3955e-02	8.7199e+01	5.2864e-01
2125	-4.7249e+08	7.2135e-03	3.1672e+01	5.2864e-01
2150	-4.7245e+08	3.2857e-02	2.9427e+01	5.2864e-01
2175	-4.7240e+08	2.0192e-02	1.2562e+02	5.2864e-01
2200	-4.7235e+08	7.0855e-03	2.6315e+01	5.2864e-01
2225	-4.7228e+08	5.8013e-02	3.1517e+01	5.2864e-01
2250	-4.7223e+08	7.4819e-03	5.1631e+01	5.2864e-01
2275	-4.7216e+08	6.9347e-03	2.5580e+01	5.2864e-01
2300	-4.7211e+08	7.1441e-02	3.1276e+01	5.2864e-01
2325	-4.7206e+08	2.4731e-02	1.5301e+02	5.2864e-01
2350	-4.7199e+08	5.9096e-02	3.2453e+02	5.2864e-01
2375	-4.7192e+08	8.4939e-03	2.6450e+01	5.2864e-01
2400	-4.7185e+08	4.4111e-02	2.6832e+02	5.2864e-01
2425	-4.7180e+08	6.6668e-03	2.6128e+01	5.2864e-01
2450	-4.7174e+08	3.6798e-02	2.6985e+01	5.2864e-01
2475	-4.7168e+08	6.5874e-03	2.3544e+01	5.2864e-01
2500	-4.7162e+08	6.5495e-03	2.8012e+01	5.2864e-01
2525	-4.7156e+08	5.2279e-02	2.5671e+01	5.2864e-01
2550	-4.7148e+08	1.3396e-01	7.7213e+02	5.2864e-01
2575	-4.7145e+08	6.4451e-03	2.2037e+01	5.2864e-01
2600	-4.7138e+08	2.7381e-02	2.5883e+01	5.2864e-01
2625	-4.7132e+08	6.3664e-03	2.8229e+01	5.2864e-01
2650	-4.7126e+08	6.3311e-03	2.6513e+01	5.2864e-01
2675	-4.7115e+08	1.5654e-01	3.4059e+01	5.2864e-01
2700	-4.7111e+08	6.2476e-03	2.9055e+01	5.2864e-01
2725	-4.7104e+08	6.2110e-03	2.2504e+01	5.2864e-01
2750	-4.7094e+08	1.4015e-01	3.1044e+01	5.2864e-01
2775	-4.7089e+08	8.8751e-02	5.2919e+02	5.2864e-01
2800	-4.7086e+08	6.1154e-03	2.0105e+01	5.2864e-01
2825	-4.7080e+08	2.5806e-02	2.3165e+01	5.2864e-01
2850	-4.7073e+08	6.0480e-03	2.0863e+01	5.2864e-01
2875	-4.7067e+08	6.0207e-03	2.0198e+01	5.2864e-01
2900	-4.7057e+08	1.1305e-01	2.9824e+01	5.2864e-01
2925	-4.7051e+08	5.2111e-02	3.1440e+02	5.2864e-01
2950	-4.7045e+08	5.9166e-03	2.7635e+01	5.2864e-01
2975	-4.7037e+08	8.6940e-03	1.9439e+01	5.2864e-01
3000	-4.7028e+08	5.8426e-03	2.4123e+01	5.2864e-01
3025	-4.7022e+08	5.8161e-03	2.3014e+01	5.2864e-01
3050	-4.7014e+08	1.4760e-01	2.5157e+01	5.2864e-01

3075	-4.7010e+08	5.7640e-03	2.7885e+01	5.2864e-01
3100	-4.7003e+08	1.6946e-02	1.0271e+02	5.2864e-01
3125	-4.6994e+08	4.7193e-02	2.5746e+01	5.2864e-01
3150	-4.6989e+08	5.6739e-03	1.9034e+01	5.2864e-01
3175	-4.6979e+08	5.6368e-03	2.4193e+01	5.2864e-01
3200	-4.6974e+08	6.6384e-03	1.6504e+01	5.2864e-01
3225	-4.6966e+08	5.5858e-03	2.6036e+01	5.2864e-01
3250	-4.6958e+08	9.2238e-03	5.4887e+01	5.2864e-01
3275	-4.6951e+08	1.1395e-01	2.3282e+01	5.2864e-01
3300	-4.6945e+08	5.5068e-03	1.7254e+01	5.2864e-01
3325	-4.6938e+08	5.4786e-03	1.7123e+01	5.2864e-01
3350	-4.6930e+08	4.5705e-02	2.1161e+01	5.2864e-01
3375	-4.6923e+08	2.0284e-02	1.2165e+02	5.2864e-01
3400	-4.6916e+08	5.4029e-03	1.8642e+01	5.2864e-01
3425	-4.6904e+08	1.0359e-01	3.0038e+01	5.2864e-01
3450	-4.6899e+08	1.1571e-02	7.3901e+01	5.2864e-01
3475	-4.6895e+08	5.3289e-03	1.8406e+01	5.2864e-01
3500	-4.6888e+08	6.1170e-02	2.2534e+01	5.2864e-01
3525	-4.6880e+08	5.7227e-03	3.7546e+01	5.2864e-01
3550	-4.6875e+08	5.2632e-03	1.7892e+01	5.2864e-01
3575	-4.6866e+08	1.2936e-02	1.7240e+01	5.2864e-01
3600	-4.6856e+08	5.2042e-03	2.3501e+01	5.2864e-01
3625	-4.6850e+08	5.1848e-03	2.4271e+01	5.2864e-01
3650	-4.6840e+08	2.2120e-01	3.0045e+01	5.2864e-01
3675	-4.6837e+08	1.3199e-02	7.7768e+01	5.2864e-01
3700	-4.6829e+08	5.1229e-03	1.6757e+01	5.2864e-01
3725	-4.6821e+08	3.2678e-02	1.9328e+01	5.2864e-01
3750	-4.6814e+08	2.5504e-02	1.5406e+02	5.2864e-01
3775	-4.6809e+08	5.0633e-03	1.8747e+01	5.2864e-01
3800	-4.6798e+08	9.9240e-02	2.5892e+01	5.2864e-01
3825	-4.6791e+08	1.2549e-01	7.4640e+02	5.2864e-01
3850	-4.6787e+08	5.0003e-03	1.7186e+01	5.2864e-01
3875	-4.6779e+08	5.0628e-02	2.1352e+01	5.2864e-01
3900	-4.6773e+08	7.1531e-03	5.3525e+01	5.2864e-01
3925	-4.6768e+08	4.9479e-03	1.6005e+01	5.2864e-01
3950	-4.6761e+08	4.9308e-03	1.6706e+01	5.2864e-01
3975	-4.6753e+08	1.2294e-02	7.3496e+01	5.2864e-01
4000	-4.6746e+08	4.8904e-03	2.5046e+01	5.2864e-01
4025	-4.6737e+08	4.7614e-02	2.2347e+01	5.2864e-01
4050	-4.6730e+08	6.0400e-03	3.0906e+01	5.2864e-01
4075	-4.6724e+08	6.2412e-03	1.8803e+01	5.2864e-01
4100	-4.6715e+08	5.0434e-02	2.0857e+01	5.2864e-01
4125	-4.6710e+08	1.4032e-02	8.5322e+01	5.2864e-01
4150	-4.6704e+08	1.2671e-02	5.7333e+01	5.2864e-01
4175	-4.6698e+08	7.5228e-03	1.6513e+01	5.2864e-01
4200	-4.6693e+08	4.7578e-03	1.6643e+01	5.2864e-01
4225	-4.6687e+08	2.2721e-02	1.3573e+02	5.2864e-01
4250	-4.6677e+08	4.9560e-02	2.7362e+01	5.2864e-01
4275	-4.6672e+08	1.2808e-02	7.8946e+01	5.2864e-01
4300	-4.6669e+08	4.7001e-03	2.1170e+01	5.2864e-01
4325	-4.6662e+08	8.4881e-03	2.0322e+01	5.2864e-01
4350	-4.6655e+08	2.6158e-02	1.5687e+02	5.2864e-01
4375	-4.6646e+08	4.6489e-03	2.0922e+01	5.2864e-01
4400	-4.6640e+08	2.9306e-02	2.3687e+01	5.2864e-01

4425	-4.6634e+08	3.8369e-02	2.2915e+02	5.2864e-01
4450	-4.6629e+08	4.6110e-03	2.6147e+01	5.2864e-01
4475	-4.6622e+08	1.3233e-02	1.9024e+01	5.2864e-01
4500	-4.6616e+08	4.5808e-03	1.7267e+01	5.2864e-01
4525	-4.6610e+08	4.5683e-03	2.0522e+01	5.2864e-01
4550	-4.6602e+08	3.7109e-02	2.6812e+01	5.2864e-01
4575	-4.6598e+08	4.5418e-03	1.9288e+01	5.2864e-01
4600	-4.6589e+08	5.3785e-02	3.2298e+02	5.2864e-01
4625	-4.6582e+08	3.1368e-02	2.5506e+01	5.2864e-01
4650	-4.6577e+08	4.4992e-03	1.8433e+01	5.2864e-01
4675	-4.6571e+08	4.4852e-03	2.1801e+01	5.2864e-01
4700	-4.6566e+08	5.9432e-02	2.2866e+01	5.2864e-01
4725	-4.6560e+08	4.4627e-03	2.0600e+01	5.2864e-01
4750	-4.6553e+08	4.4480e-03	2.2566e+01	5.2864e-01
4775	-4.6545e+08	6.8758e-02	2.4829e+01	5.2864e-01
4800	-4.6541e+08	4.4252e-03	2.0865e+01	5.2864e-01
4825	-4.6535e+08	4.4136e-03	2.0653e+01	5.2864e-01
4850	-4.6529e+08	4.0147e-02	2.3007e+01	5.2864e-01
4875	-4.6524e+08	4.3918e-03	1.8218e+01	5.2864e-01
4900	-4.6514e+08	5.6427e-03	2.8861e+01	5.2864e-01
4925	-4.6505e+08	1.6695e-01	2.8630e+01	5.2864e-01
4950	-4.6501e+08	1.7318e-02	1.0488e+02	5.2864e-01
4975	-4.6495e+08	3.0637e-02	1.8187e+02	5.2864e-01

```

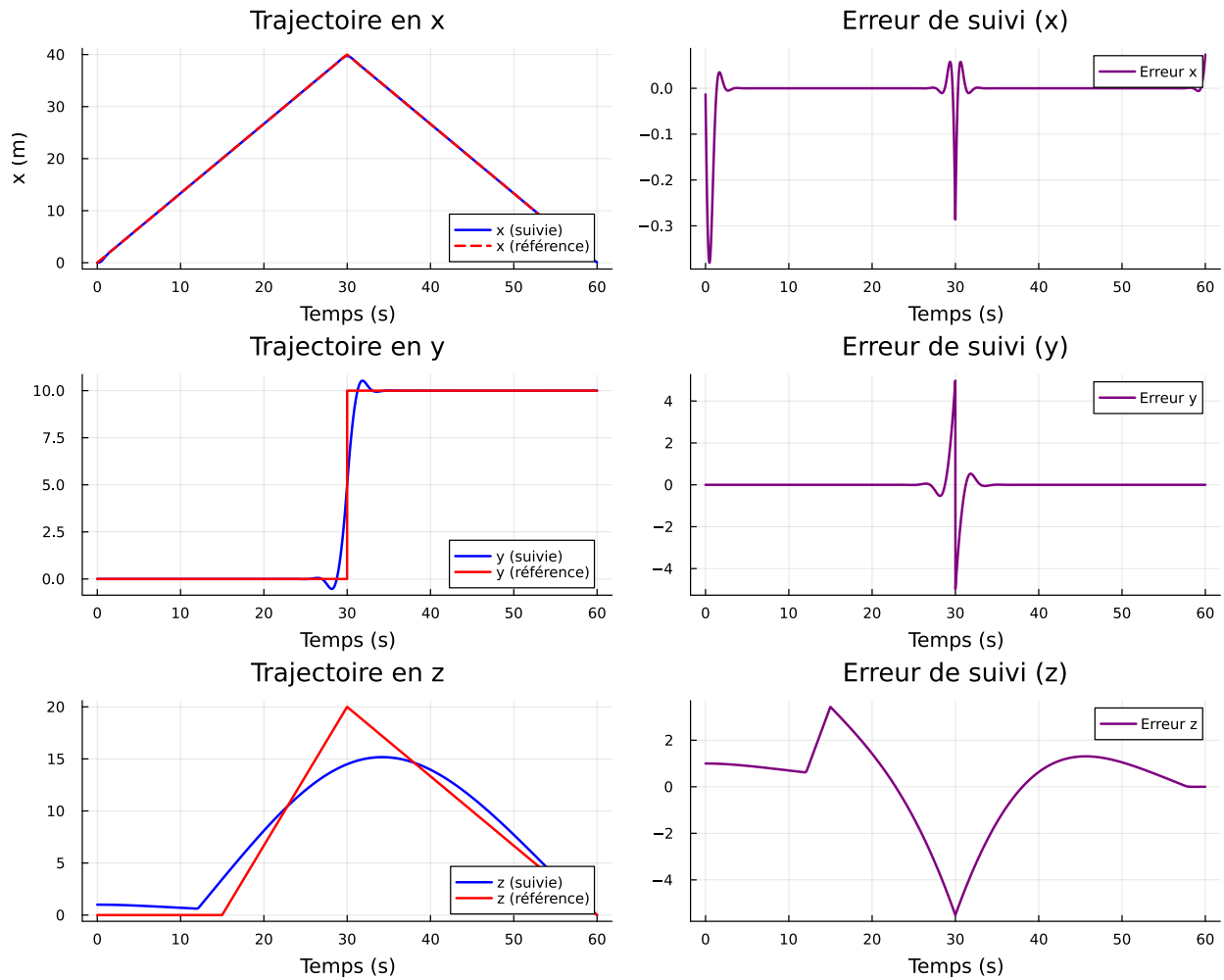
-----
>>> Results
Status: Max_iter_reached
Iterations: 5000 (incl. 10 safeguarding iter)
Optimal objective: -4.65e+08
Runtime: 325.211s (325211.0ms)

```

```

Status          : ITERATION_LIMIT
Objective value  : 1.0119104251354754e7

```



## Analyse des résultats

La méthode ne parvient pas à converger en moins de 5000 itérations. Toutefois, on constate une amélioration de la valeur de l'objectif qui est 10x plus petite que celle obtenue via Ipopt. Le temps de calcul est toutefois beaucoup plus élevé et on en déduit que cette méthode a une plus grande complexité que l'algorithme précédent. Comme le nombre de variable est différent que dans le problème formulé et présenté dans l'article, il est difficile de comparer la complexité des deux algorithmes. Lorsque le temps de calcul est limité, il est toutefois préférable d'utiliser la méthode présentée dans l'article. Les graphes présentes une erreur qui est encore une fois espectral. On observe par ailleurs que la trajectoire en y est bien meilleure avec COSMO qu'avec Ipopt. Encore une fois, on observe de plus grandes erreurs là où il y a des changements dans la valeur des dérivées de la trajectoire à suivre. Bien que la valeur optimale obtenue via COSMO soit bonne, il est possible de faire mieux avec le solveur OSQP.

## Résolution avec le solveur OSQP

OSQP (Operator Splitting Quadratic Program) est une méthode servant à résoudre des problèmes quadratiques à contraintes linéaires. C'est exactement les propriétés que nous souhaitons exploiter pour le problème monolithique.

```
model2 = Model(OSQP.Optimizer)
```



```

set_optimizer_attribute(model, "print_level", 1)

@variable(model2, X[1:n, 0:T])          # états
@variable(model2, U[1:m, 0:T-1])        # commandes

# - dynamique linéaire
for k in 0:T-1
    @constraint(model2, X[:, k+1] .== Ad * X[:, k] + Bd * U[:, k])
end

# - CONTRAINTE d'état initial (= hover)
@constraint(model2, X[:, 0] .== x0)

# - bornes (angles ±20°, poussée 0-2 mg)
deg20 = deg2rad(20.0)
@constraint(model2, -deg20 .<= X[1, :] .<= deg20) #
@constraint(model2, -deg20 .<= X[3, :] .<= deg20) #
@constraint(model2, -deg20 .<= X[5, :] .<= deg20) #
@constraint(model2, 0 .<= U[1, :] .<= 2 * mass * g)

# - coût
@expression(model2, running_cost,
    sum( (X[:, k] - r[:, k+1])' * Q * (X[:, k] - r[:, k+1]) +
        U[:, k]' * R * U[:, k] for k in 0:T-1) )

@expression(model2, terminal_cost,
    (X[:, T] - r[:, T+1])' * F * (X[:, T] - r[:, T+1]) )

@objective(model2, Min, 0.5 * running_cost + 0.5 * terminal_cost)

# - point initial
set_start_value.(X[:, 0], x0)
set_start_value.(U, 0.0)

optimize!(model2)
println("Status          : ", termination_status(model2))
println("Objective value : ", objective_value(model2))

# =====
# Visualisation de la trajectoire et de l'erreur
# =====

solutionX = Array{value.(X)}

z = solutionX[11, :] # position verticale
x = solutionX[7, :]  # position x
y = solutionX[9, :]  # position y

# Erreurs de suivi
e_z = z .- r[11, :]
e_x = x .- r[7, :]
e_y = y .- r[9, :]
z = solutionX[11, :] # position verticale

```

```

x = solutionX[7, :] # position x
y = solutionX[9, :] # position y

# Erreurs de suivi
e_z = z .- r[11, :]
e_x = x .- r[7, :]
e_y = y .- r[9, :]

# Références
r_z = r[11, :]
r_x = r[7, :]
r_y = r[9, :]

t = Ts .* (0:T) # temps réel en secondes

# Création du layout 3 lignes × 2 colonnes
plt4 = plot(layout = (3, 2), size=(1000, 800))

# Trajectoire x
plot!(plt4[1], t, x, lw=2, label="x (suivie)", color=:blue)
plot!(plt4[1], t, r_x, lw=2, label="x (référence)", linestyle=:dash, color=:red)
plot!(plt4[1], title="Trajectoire en x", xlabel="Temps (s)", ylabel="x (m)", legend=:bottomright, grid=true)

# Erreur x
plot!(plt4[2], t, e_x, lw=2, label="Erreur x", color=:purple)
plot!(plt4[2], title="Erreur de suivi (x)", xlabel="Temps (s)", ylabel="Erreur (m)", legend=:topright, grid=true)

# Trajectoire y
plot!(plt4[3], t, y, lw=2, label="y (suivie)", color=:blue)
plot!(plt4[3], t, r_y, lw=2, label="y (référence)", linestyle=:dash, color=:red)
plot!(plt4[3], title="Trajectoire en y", xlabel="Temps (s)", ylabel="y (m)", legend=:bottomright, grid=true)

# Erreur y
plot!(plt4[4], t, e_y, lw=2, label="Erreur y", color=:purple)
plot!(plt4[4], title="Erreur de suivi (y)", xlabel="Temps (s)", ylabel="Erreur (m)", legend=:topright, grid=true)

# Trajectoire z
plot!(plt4[5], t, z, lw=2, label="z (suivie)", color=:blue)
plot!(plt4[5], t, r_z, lw=2, label="z (référence)", linestyle=:dash, color=:red)
plot!(plt4[5], title="Trajectoire en z", xlabel="Temps (s)", ylabel="z (m)", legend=:bottomright, grid=true)

# Erreur z
plot!(plt4[6], t, e_z, lw=2, label="Erreur z", color=:purple)
plot!(plt4[6], title="Erreur de suivi (z)", xlabel="Temps (s)", ylabel="Erreur (m)", legend=:topright, grid=true)

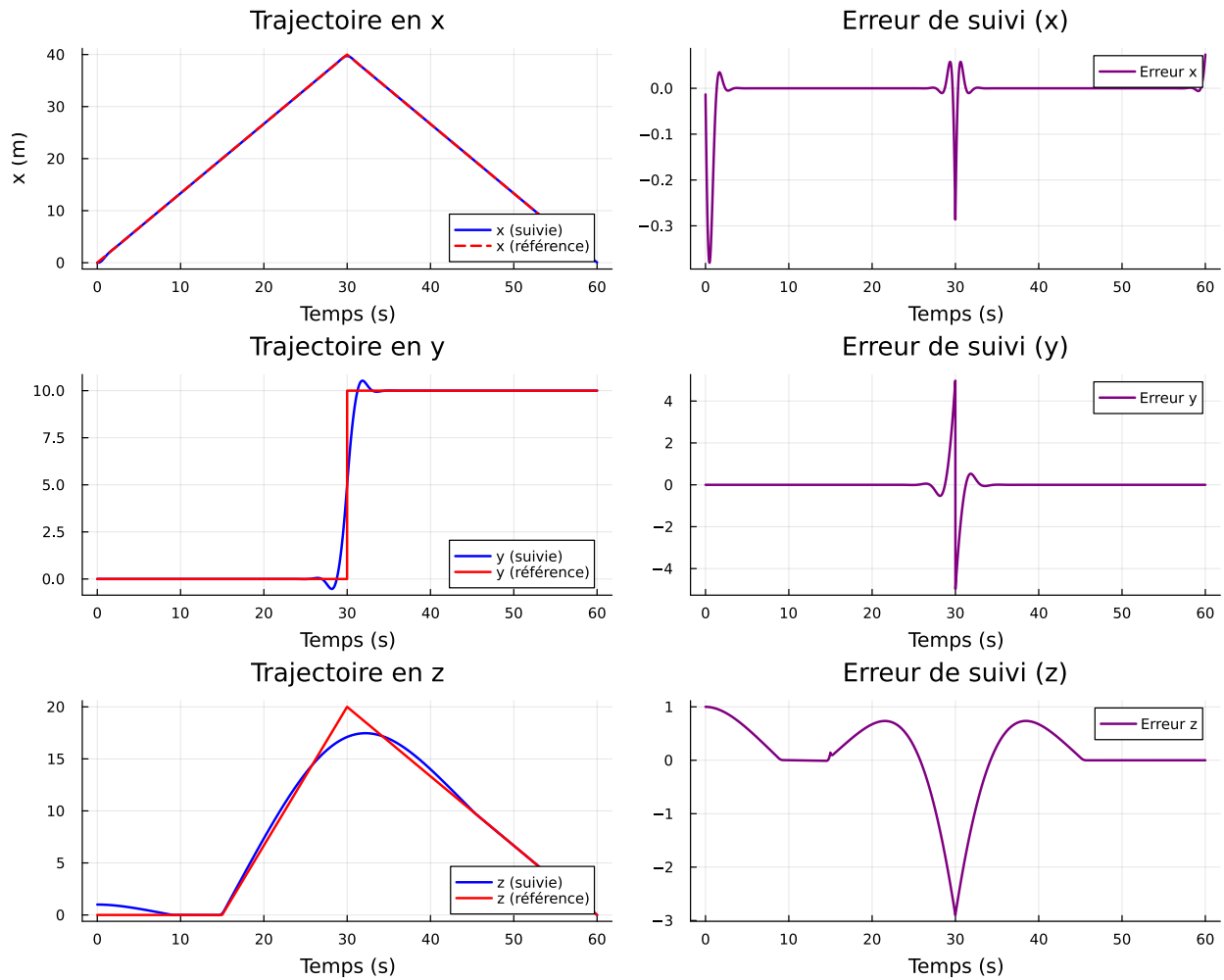
# Affichage
display(plt4)

```

```

Status          : ALMOST_OPTIMAL
Objective value  : 1.548999877283454e6

```



## Analyse des résultats

Notons que cette méthode est bien plus efficace que Ipopt et COSMO. La trajectoire désirée est en général mieux approximée, le temps de calcul est meilleur que COSMO en restant très proche de celui d'Ipopt. De plus, la valeur optimale de l'objectif est dix fois plus petite que pour COSMO et cent fois plus petite qu'Ipopt. Cette méthode est donc à privilégier par rapport aux deux méthodes précédentes.

## Lien Github

[https://github.com/joey-van-melle/MTH8408\\_Optimal-Path-Tracking](https://github.com/joey-van-melle/MTH8408_Optimal-Path-Tracking)