MP-282
Dynamic Modeling and Control of Multirotor Aerial Vehicles
Chapter 9: MAV Guidance

Prof. Dr. Davi Antônio dos Santos
Instituto Tecnológico de Aeronáutica
www.professordavisantos.com

São José dos Campos - SP
2020

Contents

Introduction . . .

Introduction

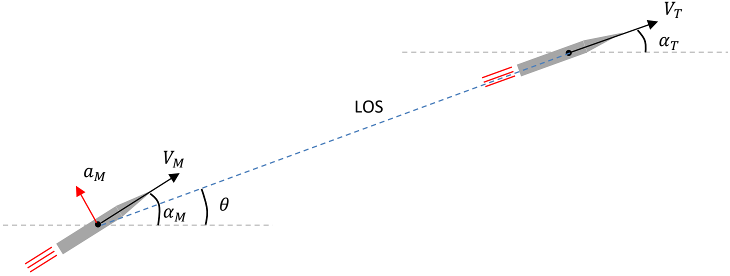
What is guidance?

- Cambridge Dictionary: "The process of directing the flight of a missile or rocket".
- Oxford Dictionary: "The directing of the motion or position of something, especially an aircraft, spacecraft or missile".

Introduction

Missile Guidance

Consider the illustrated scenario:



Proportional Navigation is a classical method of missile guidance. It provides the following lateral acceleration (command):

$$a_M = \kappa \|V_M\| \dot{\theta} \tag{1}$$

where κ is a proportionality factor.

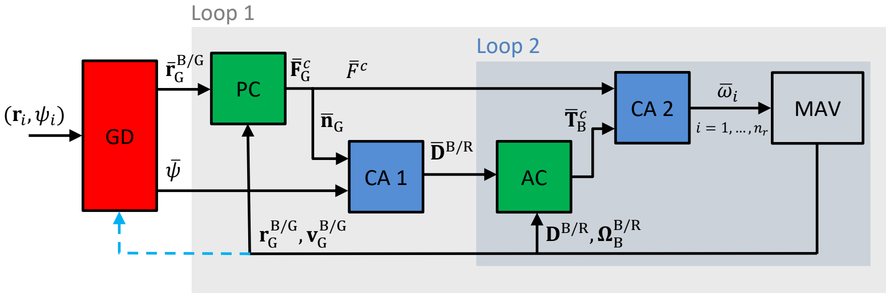
...Missile Guidance

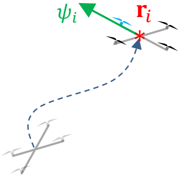
Therefore, we see that the afore-presented dictionary definitions are very consistent with the classical notion of missile or rocket guidance!

What about MAV guidance?!

MAV Guidance

Considering a fixed-rotor MAV, the implementation idea is illustrated below.





Definitions . . .

Definitions

Waypoint

A waypoint is a vector $\mathbf{w}_i \triangleq (\mathbf{r}_i, \psi_i) \in \mathbb{R}^4$ composed by a reference position $\mathbf{r}_i \in \mathbb{R}^3$ and a reference heading $\psi_i \in \mathbb{R}$, both w.r.t. $\mathcal{S}_{\mathrm{G}}$. The vector $\mathbf{r}_i$ is an $\mathcal{S}_{\mathrm{G}}$ representation.

Wayset

A wayset $\mathcal{W}_i$ associated with $\mathbf{w}_i$ is a neighborhood of $\mathbf{w}_i$, *i.e.*,

$$\mathcal{W}_i \triangleq \left\{ \mathbf{w} = (\mathbf{r}, \psi) \in \mathbb{R}^3 \times \mathbb{R} : \|\mathbf{r} - \mathbf{r}_i\| \leq \rho_r, |\psi - \psi_i| \leq \rho_\psi \right\}, \quad (2)$$

where $\rho_r \in \mathbb{R}_{>0}$ and $\rho_\psi \in \mathbb{R}_{>0}$ are given parameters.

Definitions

MAV Guidance

It is the process of commanding the MAV flight control system with a time-varying command

$$\mathbf{c} : \mathbb{R} \to \mathbb{R}^4 \tag{3}$$

$$t \mapsto \mathbf{c}(t) \tag{4}$$

$$\mathbf{c}(t) \triangleq \left(\bar{\mathbf{r}}_{\mathrm{G}}^{\mathrm{B/G}}(t), \bar{\psi}(t)\right) \tag{5}$$

so as to make it visit each wayset of a given sequence $\{\mathcal{W}_i, i = 1, \ldots, f\}$ [1].

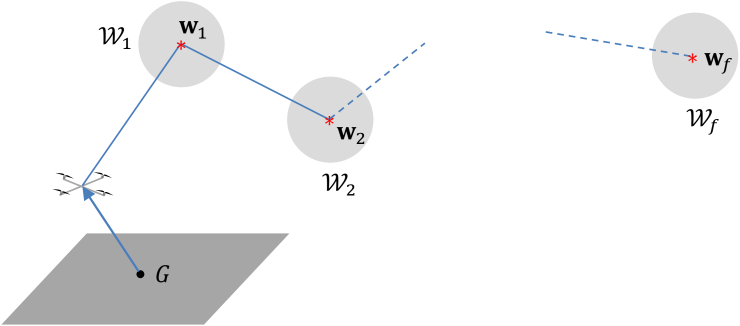
It seems to agree with the Oxford definition, except that here we have included the heading command and the wayset concept.

[1] Later, in Model Based Methods, our guidance problem will be also concerned with command feasibility.

Definitions

... MAV Guidance

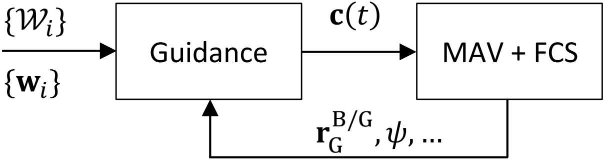
Here we have an illustration:



Basic Methods . . .

Overview

Here we have a simplified block diagram (compared with the one in slide 7) representing the interaction between the flight control system and the guidance module.



Note however that, different from slide 7, now we are not particularly concerned with fixed-rotor MAVs.

Idea

In the brute force method, the command $\mathbf{c}(t)$ is set equal to the current reference waypoint, *i.e.*, the guidance law is simply $\mathbf{c}(t) = \mathbf{w}_i$.

Algorithm 1:

Data: $\{\mathcal{W}_i\}$, $\{\mathbf{w}_i\}$

$i \leftarrow 1$, $f \leftarrow \mathrm{card}\left(\{\mathbf{w}_i\}\right)$
for $k = 1 : end$ **do**
  $\mathbf{c}(k) \leftarrow \mathbf{w}_i$
  write($\mathbf{c}(k)$,`toFCS`)
  read($\mathbf{r}_{\mathrm{G}}^{\mathrm{B/G}}$, ψ,`fromFCS`)
  if $(\mathbf{r}_{\mathrm{G}}^{\mathrm{B/G}}, \psi) \in \mathcal{W}_i$ *and* $i < f$ **then**
    $i \leftarrow i + 1$
  end
end

Remark: The above guidance algorithm will probably give rise to a very aggressive motion and there is nothing there to prevent saturations or constraint violations.

Idea

Now the current waypoint $\mathbf{w}_i$ is not directly applied to the command $\mathbf{c}$. Instead, it is filtered by the following (discrete-time) low-pass filter:

$$\mathbf{c}(k+1) = (\mathbf{I}_4 - \boldsymbol{\Lambda})\mathbf{c}(k) + \boldsymbol{\Lambda}\mathbf{w}_i \tag{6}$$

where

$$\begin{aligned} \mathbf{c}(1) &\equiv \left(\mathbf{r}_{\mathrm{G}}^{\mathrm{B/G}}, \psi\right)(t) \\ \boldsymbol{\Lambda} &\triangleq \operatorname{diag} \boldsymbol{\lambda} \\ \boldsymbol{\lambda} &\triangleq (\lambda_1, \lambda_2, \lambda_3, \lambda_4) \end{aligned}$$

Considering (6) as an Euler time discretization of a 1st order LPF, we have $\lambda_j = T/\tau_j$, where T is the sampling time and τ_j is a time constant.

Algorithm 2:

Data: $\{\mathcal{W}_i\}$, $\{\mathbf{w}_i\}$, $\boldsymbol{\Lambda}$

$i \leftarrow 1$, $f \leftarrow \mathrm{card}\left(\{\mathbf{w}_i\}\right)$

read($\mathbf{r}_{\mathrm{G}}^{\mathrm{B/G}}$, ψ,`fromFCS`)

$\mathbf{c}(0) \leftarrow \left(\mathbf{r}_{\mathrm{G}}^{\mathrm{B/G}}, \psi\right)$

for $k = 1 : end$ **do**

    $\mathbf{c}(k) \leftarrow (\mathbf{I}_4 - \boldsymbol{\Lambda})\mathbf{c}(k-1) + \boldsymbol{\Lambda}\mathbf{w}_i$

    write($\mathbf{c}(k)$,`toFCS`)

    read($\mathbf{r}_{\mathrm{G}}^{\mathrm{B/G}}$, ψ,`fromFCS`)

    if $(\mathbf{r}_{\mathrm{G}}^{\mathrm{B/G}}, \psi) \in \mathcal{W}_i$ *and* $i < f$ **then**

        $i \leftarrow i + 1$

    end

end

Remark:

Note that if we set $\lambda_i = 0$, $\forall i$, the MAV will remain at the same pose, independent of the value of $\mathbf{w}_i$. This is the least aggressive pose command. On the other hand, if $\lambda_i = 1$, $\forall i$, then **Algorithm 2** coincides with the brute force method, which provides the most aggressive command.

Scenario Modeling . . .

Scenario Modeling

Dynamic Model

In case the position control law is like the one presented in Chapter 6, the closed-loop flight control system can be described by

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{c}(k) \quad (7)$$

where $\mathbf{x} \triangleq \left(\mathbf{r}_{\mathrm{G}}^{\mathrm{B/G}}, \mathbf{v}_{\mathrm{G}}^{\mathrm{B/G}}, \psi, \dot{\psi}\right) \in \mathbb{R}^8$, and $\mathbf{A} \in \mathbb{R}^{8\times8}$ and $\mathbf{B} \in \mathbb{R}^{8\times4}$ are known matrices. We can alternatively express the dynamics of each DOF by

$$\mathbf{x}_j(k+1) = \mathbf{A}_j\mathbf{x}_j(k) + \mathbf{B}_j c_j(k),\ j = 1, ..., 4 \quad (8)$$

where $\mathbf{x}_j \triangleq \left(\mathbf{e}_j^{\mathrm{T}}\mathbf{r}_{\mathrm{G}}^{\mathrm{B/G}}, \mathbf{e}_j^{\mathrm{T}}\mathbf{v}_{\mathrm{G}}^{\mathrm{B/G}}\right) \in \mathbb{R}^2$, for $j = 1, \ldots, 3$, $\mathbf{x}_4 \triangleq \left(\psi, \dot{\psi}\right) \in \mathbb{R}^2$, and $\mathbf{A}_j \in \mathbb{R}^{2\times2}$ and $\mathbf{B}_j \in \mathbb{R}^{2\times1}$ are known matrices.

Scenario Modeling

Flight Space

It can be represented by

$$\mathbf{S}\mathbf{x}(k) \in \mathcal{S}, \ \forall k \tag{9}$$

where $\mathcal{S} \subset \mathbb{R}^4$ is a known compact set and

$$\mathbf{S} \triangleq \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3\times 3} & 0 & 0 \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & 1 & 0 \end{bmatrix} \in \mathbb{R}^{4\times 8} \tag{10}$$

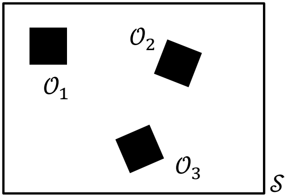
Scenario Modeling

If there exist obstacles, they can be represented by

$$\mathbf{S}_r \mathbf{x}(k) \notin \mathcal{O} \triangleq \bigcup_{l=1}^{n_o} \mathcal{O}_l, \ \forall k \tag{11}$$

where each $\mathcal{O}_l \subset \mathbb{R}^3$ is a compact set representing one particular obstacle and

$$\mathbf{S}_r \triangleq \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3\times 3} & 0 & 0 \end{bmatrix} \in \mathbb{R}^{3\times 8} \tag{12}$$



Scenario Modeling

Velocity Bounds

It can be represented by

$$\mathbf{V}\mathbf{x}(k) \in \mathcal{V},\ \forall k \tag{13}$$

where $\mathcal{V} \subset \mathbb{R}^4$ is a known compact set (usually symmetric with respect to the origin) and

$$\mathbf{V} \triangleq \begin{bmatrix} \mathbf{0}_{3\times 3} & \mathbf{I}_3 & 0 & 0 \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & 0 & 1 \end{bmatrix} \in \mathbb{R}^{4\times 8} \tag{14}$$

Scenario Modeling

Terminal Set

An important requirement to the guidance law is that after a number $N \in \mathbb{Z}_{>0}$ of discrete-time steps, the output $\mathbf{y} \triangleq \left(\mathbf{r}_{\mathrm{G}}^{\mathrm{B/G}}, \psi\right) \in \mathbb{R}^4$ must be inside the currently active wayset $\mathcal{W}_i$, *i.e.*,

$$\mathbf{Sx}(N) \in \mathcal{W}_i, \tag{15}$$

where $\mathbf{S}$ is defined in equation (10).

Scenario Modeling

Force Command Bounds

Suppose that the flight control system respects the time-scale separation assumption, it has no active saturation, and its position control law is given by (see Chapter 6)

$$\bar{\mathbf{F}}_{\mathrm{G}}^{c} = m\left(\mathbf{K}_1\left(\bar{\mathbf{r}}_{\mathrm{G}}^{\mathrm{B/G}} - \mathbf{r}_{\mathrm{G}}^{\mathrm{B/G}}\right) - \mathbf{K}_2\mathbf{v}_{\mathrm{G}}^{\mathrm{B/G}} + g\mathbf{e}_3\right) \tag{16}$$

On the other hand, consider that $\bar{\mathbf{F}}_{\mathrm{G}}^{c}$ is required to satisfy $\bar{\mathbf{F}}_{\mathrm{G}}^{c} \in \mathcal{F}$, where $\mathcal{F} \subset \mathbb{R}^3$ is a given compact set. Therefore, we obtain the constraint

$$\mathbf{F}_1\mathbf{x}(k) + \mathbf{F}_2\mathbf{c}(k) \in \mathcal{F} \ominus mg\mathbf{e}_3, \ \forall k \tag{17}$$

where

$$\mathbf{F}_1 \triangleq -\begin{bmatrix} m\mathbf{K}_1 & m\mathbf{K}_2 & \mathbf{0}_{3\times 2} \end{bmatrix} \tag{18}$$

$$\mathbf{F}_2 \triangleq \begin{bmatrix} m\mathbf{K}_1 & \mathbf{0}_{3\times 1} \end{bmatrix} \tag{19}$$

Model Based Methods . . .

The optimal low-pass filter method can be formulated as

$$\begin{aligned}
\max_{\lambda_j \in [0,1], j=1,\dots,4} \quad & \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 \\
s.t., \quad & \text{for } k = 1, \dots, N, \\
& \mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{c}(k),\ \mathbf{x}(1) \equiv \check{\mathbf{x}} \\
& \mathbf{c}(k+1) = (\mathbf{I}_4 - \boldsymbol{\Lambda})\mathbf{c}(k) + \boldsymbol{\Lambda}\mathbf{w}_i,\ \mathbf{c}(1) \equiv \mathbf{S}\check{\mathbf{x}} \\
& \mathbf{S}\mathbf{x}(k) \in \mathcal{S},\ \mathbf{S}_r\mathbf{x}(k) \notin \mathcal{O} \\
& \mathbf{V}\mathbf{x}(k) \in \mathcal{V} \\
& \mathbf{F}_1\mathbf{x}(k) + \mathbf{F}_2\mathbf{c}(k) \in \mathcal{F} \ominus mg\mathbf{e}_3 \\
& \mathbf{S}\mathbf{x}(N) \in \mathcal{W}_i
\end{aligned}$$

Remark: Assume that the sets $\mathcal{S}, \mathcal{V}, \mathcal{F}, \mathcal{W}_i$ are boxes with the sides parallel to the coordinate axes. For simplicity, consider the absence of obstacles. In this scenario, the above problem can be replaced by the four scalar optimizations presented in the sequel.

Model Based Methods

For $j = 1, 2, 3$, consider the following scalar optimization:

$$\begin{aligned}
&\max_{\lambda_j \in [0,1]} \lambda_j \\
&\quad s.t., \quad \text{for } k = 1, ..., N, \\
&\qquad \mathbf{x}_j(k+1) = \mathbf{A}_j \mathbf{x}_j(k) + \mathbf{B}_j c_j(k), \ \mathbf{x}_j(1) \equiv \check{\mathbf{x}}_j \\
&\qquad c_j(k+1) = (1-\lambda_j) c_j(k) + \lambda_j w_{ij}, \ c_j(1) = \mathbf{e}_1^{\mathrm{T}} \check{\mathbf{x}}_j \\
&\qquad \mathbf{e}_1^{\mathrm{T}} \mathbf{x}_j(k) \in \mathcal{S}_j \\
&\qquad \mathbf{e}_2^{\mathrm{T}} \mathbf{x}_j(k) \in \mathcal{V}_j \\
&\qquad \mathbf{e}_1^{\mathrm{T}} \left(\mathbf{F}_1 \mathbf{x}(k) + \mathbf{F}_2 \mathbf{c}(k) \right) \in \mathcal{F}_j \ominus \mathbb{I}_{j=3} mg \\
&\qquad \mathbf{e}_1^{\mathrm{T}} \mathbf{x}_j(N) \in \mathcal{W}_{ij}
\end{aligned}$$

where $\mathbb{I}_{j=3}$ is an indicator function, w_{ij} represents the jth component of $\mathbf{w}_i$, and $\mathcal{W}_{ij}$, $\mathcal{S}_j$, $\mathcal{V}_j$, and $\mathcal{F}_j$ are the projections of $\mathcal{W}_i$, $\mathcal{S}$, $\mathcal{V}$, and $\mathcal{F}$ on appropriate subspaces.

On the other hand, for $j = 4$, consider the following scalar optimization:

$$
\begin{aligned}
&\max_{\lambda_j \in [0,1]} \lambda_j \\
&\quad s.t., \quad \text{for } k = 1, \ldots, N, \\
&\qquad \mathbf{x}_j(k+1) = \mathbf{A}_j \mathbf{x}_j(k) + \mathbf{B}_j c_j(k), \;\; \mathbf{x}_j(1) \equiv \check{\mathbf{x}}_j \\
&\qquad c_j(k+1) = (1-\lambda_j) c_j(k) + \lambda_j w_{ij}, \;\; c_j(1) = \mathbf{e}_1^{\mathrm{T}} \check{\mathbf{x}}_j \\
&\qquad \mathbf{e}_1^{\mathrm{T}} \mathbf{x}_j(k) \in \mathcal{S}_j \\
&\qquad \mathbf{e}_2^{\mathrm{T}} \mathbf{x}_j(k) \in \mathcal{V}_j \\
&\qquad \mathbf{e}_1^{\mathrm{T}} \mathbf{x}_j(N) \in \mathcal{W}_{ij}
\end{aligned}
$$

Remark: The above scalar optimization problems can be simply solved by the bisection method.

Algorithm 3:

Data: $\{\mathcal{W}_i\}$, $\{\mathbf{w}_i\}$, N, etc.

$i \leftarrow 1$, $f \leftarrow \mathrm{card}\left(\{\mathbf{w}_i\}\right)$, $read(\check{\mathbf{x}},\texttt{fromFCS})$

$\mathbf{c}(0) \leftarrow \mathbf{S}\check{\mathbf{x}}$, $\boldsymbol{\Lambda}_i^* \leftarrow optimalLPF(N, i, \check{\mathbf{x}})$

for $k = 1 : end$ **do**

- $\mathbf{c}(k) \leftarrow (\mathbf{I}_4 - \boldsymbol{\Lambda}_i^*)\mathbf{c}(k-1) + \boldsymbol{\Lambda}_i^* \mathbf{w}_i$
- $write(\mathbf{c}(k),\texttt{toFCS})$
- $read(\check{\mathbf{x}},\texttt{fromFCS})$
- **if** $\mathbf{S}\check{\mathbf{x}} \in \mathcal{W}_i$ *and* $i < f$ **then**
  - $i \leftarrow i + 1$
  - $\boldsymbol{\Lambda}_i^* \leftarrow optimalLPF(N, i, \check{\mathbf{x}})$
  - $\mathbf{c}(k) \leftarrow \mathbf{S}\check{\mathbf{x}}$
- **end**

end

Model Based Methods

Direct Optimization

Idea

In the direct optimization method, an optimal command sequence $\left\{\mathbf{c}^*(k)\right\}$ can be directly obtained by solving the problem below.

$$
\begin{aligned}
\min_{\left\{\mathbf{c}(k)\right\},\left\{\mathbf{x}(k)\right\}} & J\left(\mathbf{w}_i; \left\{\mathbf{c}(k)\right\}, \left\{\mathbf{x}(k)\right\}\right) \\
s.t. \quad & \forall k, \\
& \mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{c}(k),\ \mathbf{x}(1) \equiv \check{\mathbf{x}} \\
& \mathbf{S}\mathbf{x}(k) \in \mathcal{S},\ \mathbf{S}_r\mathbf{x}(k) \notin \mathcal{O} \\
& \mathbf{V}\mathbf{x}(k) \in \mathcal{V} \\
& \mathbf{F}_1\mathbf{x}(k) + \mathbf{F}_2\mathbf{c}(k) \in \mathcal{F} \ominus mg\mathbf{e}_3 \\
& \mathbf{S}\mathbf{x}(N) \in \mathcal{W}_i
\end{aligned}
$$

Assuming that there is no obstacle in the scenario and specifying the problem sets as

$$\mathcal{S} \triangleq \left\{ (\mathbf{r}, \psi) \in \mathbb{R}^4 : \mathbf{r}_{\min} \leq \mathbf{r} \leq \mathbf{r}_{\max}, \psi_{\min} \leq \psi \leq \psi_{\max} \right\} \tag{20}$$

$$\mathcal{V} \triangleq \left\{ \boldsymbol{\nu} \in \mathbb{R}^4 : -\boldsymbol{\nu}_{\max} \leq \boldsymbol{\nu} \leq \boldsymbol{\nu}_{\max} \right\} \tag{21}$$

$$\mathcal{F} \triangleq \left\{ \mathbf{f} \in \mathbb{R}^3 : \mathbf{F}_{\min} \leq \mathbf{f} \leq \mathbf{F}_{\max}, \right\} \tag{22}$$

$$\mathcal{W}_i \triangleq \left\{ \mathbf{w} \in \mathbb{R}^4 : \mathbf{w} = \mathbf{w}_i + \tilde{\mathbf{w}}, -\mathbf{w}_{\max} \leq \tilde{\mathbf{w}} \leq \mathbf{w}_{\max} \right\} \tag{23}$$

and the cost function as

$$J\left(\mathbf{w}_i; \left\{\mathbf{c}(k)\right\}, \left\{\mathbf{x}(k)\right\}\right) \triangleq \frac{1}{2} \sum_{k=1}^{N-1} \left(\|\mathbf{S}\mathbf{x}(k) - \mathbf{w}_i\|_{\mathbf{Q}_1}^2 + \|\mathbf{c}(k)\|_{\mathbf{Q}_2}^2 \right) + \frac{1}{2} \|\mathbf{S}\mathbf{x}(N) - \mathbf{w}_i\|_{\mathbf{Q}_3}^2 \tag{24}$$

...

... the above problem can be rewritten in the form:

$$\min_{\mathbf{C}} \frac{1}{2}\mathbf{C}^{\mathrm{T}}\mathbf{Q}\mathbf{C} + \mathbf{f}^{\mathrm{T}}\mathbf{C} \tag{25}$$

$s.t.$

$$\boldsymbol{\Gamma}\mathbf{C} \leq \gamma \tag{26}$$

Remark: The above optimization is a quadratic program. Its solution can be obtained efficiently using active-set or interior-point methods. In MATLAB, one can use the command `quadprog`.

Let us present a constructive proof (on the black board)...

Prediction Model

Define the extended input and state vectors

$$\mathbf{C} \triangleq (\mathbf{c}(1), \mathbf{c}(2), \ldots, \mathbf{c}(N-1)) \tag{27}$$

$$\mathbf{X} \triangleq (\mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(N)) \tag{28}$$

Using equatin (7), we can express

$$\mathbf{X} = \mathcal{A}\check{\mathbf{x}} + \mathcal{B}\mathbf{C} \tag{29}$$

where

$$\mathcal{A} \triangleq \begin{bmatrix} \mathbf{I}_8 \\ \mathbf{A} \\ \vdots \\ \mathbf{A}^{N-1} \end{bmatrix} \in \mathbb{R}^{8N\times 8}, \quad \mathcal{B} \triangleq \begin{bmatrix} \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{B} & \ldots & \mathbf{0} \\ \vdots & \vdots & \vdots \\ \mathbf{A}^{N-2}\mathbf{B} & \ldots & \mathbf{B} \end{bmatrix} \in \mathbb{R}^{8N\times 4(N-1)}$$

Cost Function

We can transform (24) into (25) with

$$\mathbf{Q} \triangleq \mathcal{B}^{\mathrm{T}} \mathbf{Q}_1^{+} \mathcal{B} + \mathrm{diag}_{N-1}\, \mathbf{Q}_2 \tag{30}$$

$$\mathbf{f}^{\mathrm{T}} \triangleq \check{\mathbf{x}}^{\mathrm{T}} \mathcal{A}^{\mathrm{T}} \mathbf{Q}_1^{+} \mathcal{B} - \mathbf{f}_1^{\mathrm{T}} \mathcal{B} \tag{31}$$

where [2]

$$\mathbf{Q}_1^{+} \triangleq \mathrm{diag}\left(\mathrm{diag}_{N-1}\, \mathbf{S}^{\mathrm{T}} \mathbf{Q}_1 \mathbf{S}, \mathbf{S}^{\mathrm{T}} \mathbf{Q}_3 \mathbf{S}\right)$$

$$\mathbf{f}_1 \triangleq \begin{bmatrix} \left[\mathbf{S}^{\mathrm{T}} \mathbf{Q}_1 \mathbf{w}_i\right]_{N-1} \\ \mathbf{S}^{\mathrm{T}} \mathbf{Q}_3 \mathbf{w}_i \end{bmatrix}$$

[2] Consider some $\mathbf{a} \in \mathbb{R}^n$ and some $\mathbf{A} \in \mathbb{R}^{n \times n}$. Then, $[\mathbf{a}]_L \in \mathbb{R}^{nL}$ denotes an extended vector constructed with stacked copies of $\mathbf{a}$, while $\mathrm{diag}_L\, \mathbf{A} \in \mathbb{R}^{nL \times nL}$ denotes a block diagonal matrix with L blocks equal to $\mathbf{A}$.

Flight Space

Using (20) and (29), equation (9) can be converted into

$$\mathbf{S}^{++}\mathcal{B}\mathbf{C} \leq \begin{bmatrix} \mathbf{r}_{\max} \\ \psi_{\max} \end{bmatrix}_N - \mathbf{S}^{++}\mathcal{A}\check{\mathbf{x}} \tag{32}$$

$$-\mathbf{S}^{++}\mathcal{B}\mathbf{C} \leq -\begin{bmatrix} \mathbf{r}_{\min} \\ \psi_{\min} \end{bmatrix}_N + \mathbf{S}^{++}\mathcal{A}\check{\mathbf{x}} \tag{33}$$

where

$$\mathbf{S}^{++} \triangleq \operatorname{diag}(\mathbf{S}, \ldots, \mathbf{S}) \in \mathbb{R}^{4N \times 8N} \tag{34}$$

Velocity bounds

Using (21) and (29), equation (13) can be converted into

$$\mathbf{V}^{+}\mathcal{B}\mathbf{C} \leq [\boldsymbol{\nu}_{\max}]_N - \mathbf{V}^{+}\mathcal{A}\check{\mathbf{x}} \tag{35}$$

$$-\mathbf{V}^{+}\mathcal{B}\mathbf{C} \leq [\boldsymbol{\nu}_{\max}]_N + \mathbf{V}^{+}\mathcal{A}\check{\mathbf{x}} \tag{36}$$

where

$$\mathbf{V}^{+} \triangleq \operatorname{diag}(\mathbf{V}, \ldots, \mathbf{V}) \in \mathbb{R}^{4N \times 8N} \tag{37}$$

Force bounds

Using (22) and (29), equation (17) can be converted into

$$\left(\mathbf{F}_1^{+}\mathcal{B}+\mathbf{F}_2^{+}\right)\mathbf{C} \leq \left[\mathbf{F}_{\max}-mg\mathbf{e}_3\right]_N-\mathbf{F}_1^{+}\mathcal{A}\check{\mathbf{x}} \tag{38}$$

$$-\left(\mathbf{F}_1^{+}\mathcal{B}+\mathbf{F}_2^{+}\right)\mathbf{C} \leq -\left[\mathbf{F}_{\min}-mg\mathbf{e}_3\right]_N+\mathbf{F}_1^{+}\mathcal{A}\check{\mathbf{x}} \tag{39}$$

where

$$\mathbf{F}_1^{+} \triangleq \left[\mathrm{diag}\left(\mathbf{F}_1,\ldots,\mathbf{F}_1\right) \quad \mathbf{0}_{3(N-1)\times 8}\right] \in \mathbb{R}^{3(N-1)\times 8N} \tag{40}$$

$$\mathbf{F}_2^{+} \triangleq \mathrm{diag}\left(\mathbf{F}_2,\ldots,\mathbf{F}_2\right) \in \mathbb{R}^{3(N-1)\times 4(N-1)} \tag{41}$$

Terminal Set

Using (23) and (29), equation (15) can be converted into

$$\mathbf{M}_1 \mathcal{B} \mathbf{C} \leq \mathbf{w}_i + \mathbf{w}_{\max} - \mathbf{M}_1 \mathcal{A} \check{\mathbf{x}} \tag{42}$$

$$-\mathbf{M}_1 \mathcal{B} \mathbf{C} \leq -\mathbf{w}_i + \mathbf{w}_{\max} + \mathbf{M}_1 \mathcal{A} \check{\mathbf{x}} \tag{43}$$

where

$$\mathbf{M}_1 \triangleq [\mathbf{0}_{4\times 8} \ \ldots \ \mathbf{0}_{4\times 8} \ \mathbf{S}] \in \mathbb{R}^{4\times 8N} \tag{44}$$

Finally, by stacking the inequalities (37),(38),(40),(41),(43),(44),(47),(48), we can immediately obtain (26).

Algorithm 4:

Data: $\{\mathcal{W}_i\}$, $\{\mathbf{w}_i\}$, N, etc.

$i \leftarrow 0$, $f \leftarrow \mathrm{card}\left(\{\mathbf{w}_i\}\right)$, *read*($\check{\mathbf{x}}$,`fromFCS`)

$\left\{\mathbf{c}^*(j)\right\}_{j=1}^{N-1} \leftarrow DO(N, i, \check{\mathbf{x}})$

for $k = 1 : end$ **do**

  write($\mathbf{c}^*(k)$,`toFCS`)

  if $k = i(N-1)$ *and* $i < f$ **then**

    $i \leftarrow i + 1$

    read($\check{\mathbf{x}}$,`fromFCS`)

    $\left\{\mathbf{c}^*(j)\right\}_{j=(i-1)N+1}^{i(N-1)} \leftarrow DO(N, i, \check{\mathbf{x}})$

  end

  if $k \geq fN$ **then**

    $\mathbf{c}(k) = \mathbf{c}(k-1)$

  end

end

Comment:

The *direct optimization* method is based on an open-loop optimization in which only the initial condition $\check{\mathbf{x}}$ is fed back into the guidance module. Note that, during the flight, any disturbance could prevent the system output $\mathbf{y}$ to reach $\mathcal{W}_i$ in N discrete-time steps, making Algorithm 4 inapropriate (why?).

A form of robustifying the above algorithm is to implement it in closed loop by means of the receding horizon strategy: at each time iteration k, read the system state $\check{\mathbf{x}}(k)$, solve the open-loop optimization $\left\{\mathbf{c}^*(j)\right\}_{j=1}^{N-1} \leftarrow DO(N, i, \check{\mathbf{x}}(k))$, get only the first optimal command of the sequence $\mathbf{c}(k) \leftarrow \mathbf{c}^*(1)$ and apply it to the flight control system. The resulting algorithm is an example of model predictive control (MPC) formulation [3].

[3] See reference [1] for a basic introduction to MPC.

Algorithm 5: MPC formulation of reference [2].

Data: $\{\mathcal{W}_i\}$, $\{\mathbf{w}_i\}$, N, etc.

$i \leftarrow 0$, $f \leftarrow \mathrm{card}\left(\{\mathbf{w}_i\}\right)$

for $k = 1 : end$ **do**

  read($\check{\mathbf{x}}(k)$,`fromFCS`)

  $\left\{\mathbf{c}^*(j)\right\}_{j=1}^{N-1} \leftarrow DO(N, i, \check{\mathbf{x}}(k))$

  $\mathbf{c}(k) \leftarrow \mathbf{c}^*(1)$

  write($\mathbf{c}(k)$,`toFCS`)

  if $\check{\mathbf{x}}(k) \in \mathcal{W}_i$ *and* $i < f$ **then**

    $i \leftarrow i + 1$

  end

end

References . . .

References

[1] J. M. Maciejowiski. Predictive Control with Constraints. Prentice Hall, Upper Saddle River, 2002.

[2] Santos, D. A., Afonso, R. J. M. Dynamic Modeling, Flight Control and Trajectory Planning for a Balloon-Hexacopter. In: 24th ABCM International Congress of Mechanical Engineering, 2017, Curitiba.