Tạp chí Khoa học và Công nghệ Trường Đại học Công nghệ Đông Á

Sử dụng thuật toán QP của bài toán tối ưu toàn phương để xây dựng bộ điều khiển MPC cho hệ phi tuyến

Phan Xuân Minh

Trường Đại học Quốc tế, Đại học Quốc gia Hà Nội

*Email:minh.phanxuan@hust.edu.vn

Tóm tắt

Điều khiển dự báo dựa theo mô hình (Model Predictive Control – MPC) là phương pháp điều khiển được sử dụng rất rộng rãi trong thực tế, mang lại hiệu quả tin cậy. Nó ra đời từ rất sớm, được xuất phát từ bài toán thực tế, dưới tên gọi Thuật toán điều khiển tựa mô hình (Model Algorithm Control – MAC) và dần phát triển thành một công cụ ưa thích để giải quyết những bài toán có ràng buộc cho cả hệ tuyến tính và phi tuyến. Tuy vậy, do là phương pháp dựa vào mô hình, nên việc phân tích, đánh giá chất lượng ổn định hay bền vững của hệ, nhất là hệ phi tuyến, khi mô hình không dừng (non-autonomous), chứa các thành phần bất định, là không hề đơn giản. Có thể nói cho tới tận ngày nay bài toán đánh giá tính ổn định của hệ MPC phi tuyến dạng này vẫn còn là bài toán mở. Lý do chính là vì ở MPC phi tuyến người ta phải tìm nghiệm toàn cục của bài toán tối ưu phi tuyến không lồi. Xuất phát từ lý do đó mà bài báo này đã đưa ra một đề xuất xây dựng bộ điều khiển MPC cho hệ phi tuyến đủ đảm bảo được tính ổn định của hệ. Đề xuất này dựa chủ yếu vào tư tưởng xấp xỉ tuyến tính từng đoạn theo cửa sổ dự báo (receeding linearization) cho hệ phi tuyến rồi sử dụng quy hoạch toàn phương (QP) cũng ở dạng từng đoạn, thay vì trực tiếp tìm nghiệm bài toán tối ưu phi tuyến không lồi. Một ví dụ của bài báo đã xác nhận tính hiệu quả của giải pháp đề xuất trên.

Từ khóa: *Điều khiển MPC; Quy hoạch toàn phương; Tính ổn định của MPC.*

Abstract

Model Predictive Control (MPC) is a widely used control method in practice, providing reliable results. It originated from a real-world problem, under the name Model Algorithm Control (MAC), and gradually developed into a preferred tool for solving constrained problems in both linear and nonlinear systems. However, because it is a model-based method, analyzing and evaluating the stability or robustness of the system, especially nonlinear systems, when the model is non-autonomous and contains uncertain components, is not simple. It can be said that even today, the problem of evaluating the stability of this type of nonlinear MPC system remains an open problem. The main reason is that in nonlinear MPC, one must find a global solution to the non-convex nonlinear optimization problem. Based on this reason, this paper proposes a design for an MPC controller for nonlinear systems that ensures system stability. This proposal is primarily based on the idea of segmented linear approximation using the prediction window (receeding linearization) for nonlinear systems, and then using quadratic programming (QP) also in segmented form, instead of directly finding the solution to the non-convex nonlinear optimization problem. An example in the paper confirms the effectiveness of the proposed solution.

Keywords*: MPC control; Quadratic programming; MPC stability.*

https://doi.org/10.65153/w23q8j15

Journal of Science and Technology of East Asia University of Technology

1. MỞ ĐẦU

Xuất phát từ một bài toán của công nghiệp, năm 1982, mà kỹ thuật MAC (Model Algorithm control) [1] được ra đời. Tiếp theo đó là DMC (dynamic matrix control) [2] và GPC năm 1987 (Generalized Predictive Control) [3]. Tất cả chúng, tuy rằng vẫn tồn tại những ưu nhược điểm riêng biệt [2], song đã tạo tiền đề cho việc hình thành kỹ thuật MPC (Model Predictive Control) [4],[5],[6] với vô vàn ứng dụng thành công trong thực tế. Ưu điểm nổi trội của kỹ thuật này so với các phương pháp điều khiển phi tuyến khác, như thích nghi, bền vững, trượt,… là nó xử lý được các bài toán điều khiển có ràng buộc về trạng thái và tín hiệu điều khiển [7]. Sở dĩ nó làm được điều này là vì bản chất của MPC là từng bước chuyển bài toán điều khiển phi tuyến thông thường về thành bài toán quy hoạch phi tuyến có ràng buộc rồi sử dụng các phương pháp tối ưu hóa để xử lý điều kiện ràng buộc đó [7], cùng với vô vàn công cụ tối ưu hóa đã có sẵn hỗ trợ việc cài đặt chúng lên thiết bị điều khiển số [8]. Tuy nhiên, cũng vì đã chuyển bài toán điều khiển phi tuyến có ràng buộc thành bài toán quy hoạch phi tuyến, tức là bài toán tối ưu có hàm mục tiêu phi tuyến và tập ràng buộc cũng được mô tả bởi các bất phương trình phi tuyến, nên vấn đề khảo sát tính ổn định của hệ thu được gặp khá nhiều khó khăn [9], do còn thiếu những công cụ tìm nghiệm toàn cục cho bài toán quy hoạch phi tuyến không lồi [10].

Nhằm khắc phục nhược điểm trên, bài báo định hướng sẽ cùng với việc dịch cửa sổ dự báo của MPC để chuyển dần từng bước mô hình phi tuyến không dừng thành tuyến tính tham số hằng (hệ LTI) trong cửa sổ dự báo hiện tại. Nhờ đó, thay vì gặp phải bài toán quy hoạch phi tuyến không lồi, ta sẽ có bài toán quy hoạch toàn phương (lồi), giúp ta luôn tìm được nghiệm toàn cục khi tập ràng buộc cũng là một tập lồi nhờ các phương pháp số giải bài toán QP (quadratic programming).

2. NỘI DUNG CHÍNH

2.1. Phát biểu bài toán và các giả thiết kèm theo

Nội dung bài báo liên quan tới nhiệm vụ thiết kế bộ điều khiển bám quỹ đạo có ràng buộc, cho hệ phi tuyến không dừng, mô tả bởi mô hình rời rạc dạng affine

$$\begin{cases} \underline{x}_{k+1} = \underline{f}(\underline{x}_k, k) + B(\underline{x}_k, k)\underline{u}_k \\ \underline{y}_k = C(\underline{x}_k, k)\underline{x}_k \end{cases} \tag{1}$$

trong đó $\underline{x}_k \in \mathbb{R}^n$ là vector của n biến trạng thái ở thời điểm rời rạc k, $\underline{u}_k \in \mathbb{R}^m$ là vector của m các tín hiệu điều khiển ở cùng thời điểm k, $\underline{y}_k \in \mathbb{R}^l$ là vector của các tín hiệu đầu ra,

$B(\underline{x}_k,k)$ là ma trận điều khiển, có kiểu $n\times m$ và $C(\underline{x}_k,k)$ là ma trận đầu ra, có kiểu $l\times n$. Như vậy hệ (1) này có bậc n và MIMO (nhiều đầu vào nhiều đầu ra) với m đầu vào và l đầu ra.

Bộ điều khiển phải thiết kế cần thỏa mãn:

- Làm cho đầu ra $\underline{y}_k$ bám theo được quỹ đạo mẫu $\underline{r}_k$ có thể chỉ cần cho trước từng đoạn $\left[k,k+N_r(k)\right]$ với $N_r(k)\in\mathbb{N}$ cũng có thể thay đổi theo k, và thỏa mãn $\underline{y}_k-\underline{r}_k\to\underline{0}_l$ khi $k\to\infty$, trong đó $\underline{0}_l$ là ký hiệu của vector có tất cả các phần tử bằng 0, và tất nhiên cũng phải có số chiều tương ứng là l.

- Vector các tín hiệu điều khiển $\underline{u}_k$ luôn thuộc U đóng và giới nội (tập compact) cho trước.

Để bài toán ở trên có nghiệm, ta giả thiết:

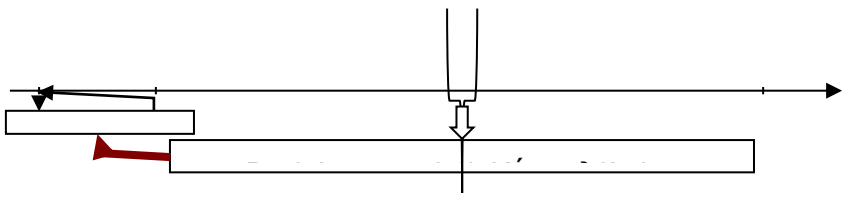
- $B(\underline{x}_k,k)\neq\mathbf{0}_{n\times m},\ \forall\underline{x}_k,k$ để hệ là điều khiển được tại $\underline{x}_k$ ở thời điểm rời rạc k, trong đó ký hiệu $\mathbf{0}_{n\times m}$ là chỉ ma trận kiểu $n\times m$ có tất cả các phần tử bằng 0.

- $C(\underline{x}_k,k)\neq\mathbf{0}_{l\times n},\ \forall\underline{x}_k,k$ để tránh trường hợp hệ rơi vào chế độ động học không tại $\underline{x}_k$ ở thời điểm rời rạc k.

- Vector hàm $\underline{f}(\underline{x}_k,k)$ thỏa mãn
$$\frac{\left\|\underline{f}(\underline{x}_k,k)\right\|}{\left\|\underline{x}_k\right\|}<\infty,\ \forall\underline{x}_k,k \qquad (2)$$
tức là $\underline{f}(\underline{x}_k,k)$ biến đổi không nhanh hơn $\underline{x}_k$ với mọi k. Ta có thể thấy tất cả những vector hàm $\underline{f}(\underline{x}_k,k)$ Lipschitz toàn cục đều thỏa mãn điều kiện (2) này.

2.2. Phương pháp điều khiển MPC truyền thống

Hình 1 ở dưới mô tả nguyên tắc xây dựng bộ điều khiển MPC truyền thống.



Hình 1: Nguyên tắc hoạt động của bộ điều khiển dự báo

Một cách chi tiết thì nó gồm các bước sau:

1. Chọn trước các khoảng dự báo N_u cho tín hiệu điều khiển $\underline{u}_k$ và N_x cho trạng thái $\underline{x}_k$, trong đó $N_u \geq N_x$. Thông thường người ta hay chọn $N_u = N_x = N$ với N không được lớn hơn khoảng $N_r(k)$ mà quỹ đạo mẫu $\underline{r}_k$ sẽ được xác định trước từng đoạn.
2. Xây dựng mô hình dự báo cho tín hiệu đầu ra $\underline{y}_i,\ i = k+1,\ \ldots\ ,k+N$ tính từ thời điểm rời rạc k hiện tại. Thông thường người ta có thể sử dụng ngay mô hình rời rạc (1) của hệ để làm mô hình dự báo.
3. Thực hiện lần lượt cho $k = 0,1,\ \ldots$ các bước sau:

   a) Đo trạng thái $\underline{x}_k$.

   b) Sử dụng mô hình dự báo để xác định các tín hiệu đầu ra tương lai $\underline{y}_i$ trong cửa sổ dự báo hiện tại $i = k+1,\ \ldots\ ,k+N$. Những tín hiệu đầu ra tương lai này sẽ là các vector hàm phụ thuộc $\underline{u}_j,\ j = k,k+1,\ \ldots\ ,k+N-1$. Để đơn giản ta sẽ sử dụng ký hiệu $\underline{y}_i(\underline{\mathbf{u}}_k)$ với $\underline{\mathbf{u}}_k = \mathrm{col}(\underline{u}_k,\underline{u}_{k+1},\ \ldots\ ,\underline{u}_{k+N-1})$ để chỉ sự phụ thuộc đó. Lúc này $\underline{\mathbf{u}}_k$ sẽ là vector trong không gian mN chiều.

   c) Chọn hàm mục tiêu đánh giá sai lệch bám

$$J_k(\underline{\mathbf{u}}_k) = h_k(\underline{\mathbf{u}}_k) + \sum_{j=1}^{N} g_k\left(\underline{y}_{k+j}(\underline{\mathbf{u}}_k),\underline{r}_{k+j}\right) \tag{3}$$

trong đó $h_k(\underline{\mathbf{u}}_k)$ phải là hàm xác định dương và $g_k\left(\underline{y}_{k+j}(\underline{\mathbf{u}}_k),\underline{r}_{k+j}\right)$ phải ít nhất là hàm bán xác định dương của các sai lệch $\underline{y}_{k+j}(\underline{\mathbf{u}}_k) - \underline{r}_{k+j}$. Thông thường người ta hay chọn hàm mục tiêu dạng toàn phương như sau:

$$J_k(\underline{\mathbf{u}}_k) = \underline{\mathbf{u}}_k^T \mathcal{R}_k \underline{\mathbf{u}}_k + \sum_{j=1}^{N}\left[\left(\underline{y}_{k+j}(\underline{\mathbf{u}}_k) - \underline{r}_{k+j}\right)^T Q_k \left(\underline{y}_{k+j}(\underline{\mathbf{u}}_k) - \underline{r}_{k+j}\right)\right] \tag{4}$$

thay cho (3), với $\mathcal{R}_k, Q_k$ là các ma trận đối xứng xác định dương chọn trước. Ở đây ta đã sử dụng ký hiệu $\mathcal{R}_k$ thay vì R_k như vẫn làm với ma trận có các phần tử là số thực, vì $\mathcal{R}_k$ thực chất là ma trận khối (các phần tử của nó cũng là ma trận).

Như vậy, nếu ta sử dụng ký hiệu gộp (vector và ma trận)

$$\begin{aligned}\underline{\mathbf{e}}_k &= \mathrm{col}(\underline{e}_{k+1},\ \ldots\ ,\underline{e}_{k+N}) \\ &= \mathrm{col}\left(\underline{y}_{k+1}(\underline{\mathbf{u}}_k) - \underline{r}_{k+1},\ \ldots\ ,\underline{y}_{k+N}(\underline{\mathbf{u}}_k) - \underline{r}_{k+N}\right)\end{aligned} \tag{5}$$

$$\mathcal{Q}_k = \mathrm{diag}(Q_k,\ \ldots\ ,Q_k) = \begin{pmatrix} Q_k & \cdots & \mathbf{0}_{l\times l} \\ \vdots & \ddots & \vdots \\ \mathbf{0}_{l\times l} & \cdots & Q_k \end{pmatrix}$$

trong đó $\mathbf{0}_{l\times l}$ là ma trận kiểu $l\times l$ có tất cả các phần tử bằng 0, thì hàm mục tiêu (4) sẽ trở thành dạng gần giống toàn phương chính tắc

$$J_k\left(\underline{\mathbf{u}}_k\right)=\underline{\mathbf{u}}_k^T\mathcal{R}_k\underline{\mathbf{u}}_k+\underline{\mathbf{e}}_k^T\mathcal{Q}_k\underline{\mathbf{e}}_k. \tag{6}$$

Ở đây ta nói rằng (6) gần giống toàn phương, chứ chưa phải toàn phương theo biến $\underline{\mathbf{u}}_k$, vì còn quan hệ $\underline{y}_{k+i}(\underline{\mathbf{u}}_k)-\underline{r}_{k+i}$ của các phần tử trong $\underline{\mathbf{e}}_k$ không ở dạng tuyến tính theo $\underline{\mathbf{u}}_k$.

d) Tìm nghiệm bài toán tối ưu hóa có ràng buộc

$$\underline{\mathbf{u}}_k^*=\arg\min_{\underline{u}_k\in U}J_k\left(\underline{\mathbf{u}}_k\right) \tag{7}$$

e) Đưa $\underline{u}_k=\left(I_m,\mathbf{0}_{m\times m},\ \dots\ ,\mathbf{0}_{m\times m}\right)\underline{\mathbf{u}}_k^*$ vào điều khiển hệ (1), trong đó I_m là ma trận đơn vị kiểu $m\times m$.

Như vậy, vòng lặp 3 với $k=0,1,\ \dots$ chỉ kết thúc khi quá trình điều khiển kết thúc. Về phương pháp MPC truyền thống này ta có một vài ý kiến đánh giá chất lượng mang lại của nó như sau:

- Khi hệ là phi tuyến dạng (1), để có được đầu ra $\underline{y}_{k+i}$ trong cửa sổ dự báo hiện tại, tức là với $i=1,\ \dots\ ,N$, ta cần dự báo được trạng thái $\underline{x}_{k+i}$ và chúng là những vector hàm phụ thuộc $\underline{x}_k,\underline{u}_k,\ \dots\ ,\underline{u}_{k+N_u-1}$. Sự phụ thuộc này mang tính phi tuyến khá cao. Cửa số dự báo N được chọn càng lớn, tính phi tuyến càng cao. Điều này ta có thể thấy thông qua phép khai triển vector trạng thái $\underline{x}_{k+i}$ trong của số dự báo hiện tại $\left[k+1,k+N\right]$

$$\begin{aligned}
\underline{x}_{k+1}&=\underline{f}(\underline{x}_k,k)+B(\underline{x}_k,k)\underline{u}_k\\
\underline{x}_{k+2}&=\underline{f}(\underline{x}_{k+1},k+1)+B(\underline{x}_{k+1},k+1)\underline{u}_{k+1}\\
&=\underline{f}\left(\underline{f}(\underline{x}_k,k)+B(\underline{x}_k,k)\underline{u}_k,k+1\right)+B\left(\underline{f}(\underline{x}_k,k)+B(\underline{x}_k,k)\underline{u}_k,k+1\right)\underline{u}_{k+1}\\
&\hat{=}\underline{\varphi}_2\left(k,k+1,\underline{x}_k,\underline{u}_k,\underline{u}_{k+1}\right)\\
\underline{x}_{k+3}&=\underline{f}(\underline{x}_{k+2},k+2)+B(\underline{x}_{k+2},k+2)\underline{u}_{k+2}\\
&=\underline{f}\left(\underline{\varphi}_2(k,k+1,\underline{x}_k,\underline{u}_k,\underline{u}_{k+1}),k+2\right)+\\
&\qquad+B\left(\underline{\varphi}_2(k,k+1,\underline{x}_k,\underline{u}_k,\underline{u}_{k+1}),k+2\right)\underline{u}_{k+2}\\
&\hat{=}\underline{\varphi}_3\left(k,k+1,k+2,\underline{x}_k,\underline{u}_k,\underline{u}_{k+1},\underline{u}_{k+2}\right)\\
&\ \vdots\\
\underline{x}_{k+i}&\hat{=}\underline{\varphi}_i\left(k,k+1,k+2,\ \dots\ ,k+i-1\underline{x}_k,\underline{u}_k,\underline{u}_{k+1},\underline{u}_{k+2},\ \dots\ ,\underline{u}_{k+i-1}\right)\\
&\ \vdots
\end{aligned}$$

trong đó, dấu $\hat{=}$ chỉ phép gán, tức là ký hiệu của một quy ước. Tuy nhiên, từ dạng hàm mục tiêu cho ở công thức (4) cũng dễ thấy rằng khi N được chọn càng nhỏ, khả năng bám tín hiệu đặt $\underline{r}_k$ của bộ điều khiển càng thấp.

- Tính phi tuyến cao của kết quả dự báo $\underline{y}_{k+i}$ dẫn tới chất lượng rất xấu của hàm mục tiêu (3) hoặc (4), làm cho nó có tính lõm mạnh với nhiều điểm cực tiểu địa phương, trong khi tất cả các phương pháp tối ưu hóa hiện có đều không có khả năng xác định được ít nhất một điểm cực tiểu toàn cục của bài toán tối ưu lõm (non-convex).

- Hiện nay, bài toán đánh giá chất lượng bám $\underline{e}_k = \underline{y}_k - \underline{r}_k$ khi $k \to \infty$ của MPC phi tuyến vẫn đang mở (chưa có kết quả rõ ràng). Để cải thiện chất lượng bám một cách định tính, người ta thường sử dụng các giải pháp trung gian như:

- Lựa chọn hợp lý các hàm $h_k(\underline{\mathbf{u}}_k)$ và $g_k(\underline{y}_{k+j}, \underline{r}_{k+j})$ cho hàm mục tiêu (3) hoặc bổ sung thêm hàm phạt cho $J_k\left(\underline{\mathbf{u}}_k\right)$ để tạo được dãy $\left\|\underline{e}_k\right\|$ đơn điệu giảm. Tuy nhiên công việc này chưa tổng quát và chỉ mang tính đơn lẻ với từng hệ cụ thể. Ngoài ra, việc có được tính đơn điệu giảm của $\left\|\underline{e}_k\right\|$ cũng chưa đủ để khẳng định sẽ có $\left\|\underline{e}_k\right\| \to 0$.

- Xây dựng được một mô hình dự báo phi tuyến phù hợp, vừa đảm bảo là nó chứa đựng được đầy đủ đặc tính động học của hệ (1), vừa có khả năng điều khiển ổn định điệm cận được, ít nhất là với bộ điều khiển MPC. Hiện tại đây cũng là một bài toán mở và chưa có một gợi ý tổng quát nào cho việc xác định mô hình dự báo đó.

Khắc phục các nhược điểm cơ bản trên của MPC phi tuyến truyền thống cũng chính là mục đích của bài báo này

2.2. Phương pháp điều khiển MPC đề xuất cho hệ phi tuyến (1)

Bản chất phương pháp đề xuất cho MPC phi tuyến (1) của bài báo được xuất phát từ kết quả điều khiển MPC đã có cho hệ tuyến tính tham số hằng (hệ LTI), rằng khi hàm mục tiêu có dạng toàn phương, điều kiện ràng buộc U là tập lồi, thì bài toán tối ưu hóa (7) sẽ là bài toán tối ưu lồi, nên nghiệm toàn cục của nó dễ dàng tìm được nhờ các phương pháp tối ưu hóa hiện có, cũng như khi hệ LTI có chứa m thành phần tích phân thì mọi bộ điều khiển làm nó ổn định cũng sẽ là nó bám theo được tín hiệu đặt khả vi m lần.

Xuất phát từ mô hình (1) thì cùng với giả thiết (2), nó sẽ viết lại được thành

$$\begin{cases} \underline{x}_{k+1} = A(\underline{x}_k, k)\underline{x}_k + B(\underline{x}_k, k)\underline{u}_k \\ \underline{y}_k = C(\underline{x}_k, k)\underline{x}_k \end{cases} \tag{8}$$

trong đó

$$A(\underline{x}_k,k)=\frac{\underline{f}(\underline{x}_k,k)}{\|\underline{x}_k\|^2}\underline{x}_k^T$$

là ma trận kiểu $n\times n$ có tất cả các phần tử là hàm bị chặn theo $\underline{x}_k$ và k. Như vậy, tại thời điểm k hiện tại, và do $\underline{x}_k$ là đã biết do đo được, nên (8) trở thành tuyến tính không dừng

$$\begin{cases}\underline{x}_{k+1}=A_k\underline{x}_k+B_k\underline{u}_k\\ \underline{y}_k=C_k\underline{x}_k\end{cases} \tag{9}$$

với

$$A_k=A(\underline{x}_k,k),\ B_k=B(\underline{x}_k,k),\ C_k=C(\underline{x}_k,k) \tag{10}$$

là những ma trận rõ (certain). Nhược điểm duy nhất của mô hình (9) này, so với hệ LTI, là các ma trận tham số A_k, B_k, C_k phụ thuộc thời gian.

Ta sẽ sử dụng (9) để xây dựng mô hình dự báo với m khâu tích phân bổ sung. Ký hiệu $\underline{v}_k=\underline{u}_k-\underline{u}_{k-1}$ thì do tại thời điểm k hiện tại đã có $\underline{u}_{k-1}$, nên để tìm $\underline{u}_k$ ta chỉ cần tìm $\underline{v}_k$, tức là $\underline{v}_k$ bây giờ sẽ là vector tín hiệu điều khiển của mô hình dự báo. Với ký hiệu này, thì tại thời điểm k mô hình (9) trở thành

$$\begin{cases}\begin{pmatrix}\underline{x}_{k+1}\\ \underline{u}_k\end{pmatrix}=\begin{pmatrix}A_k & B_k\\ \mathbf{0}_{n\times n} & I_{m\times m}\end{pmatrix}\begin{pmatrix}\underline{x}_k\\ \underline{u}_{k-1}\end{pmatrix}+\begin{pmatrix}B_k\\ I_{m\times m}\end{pmatrix}\underline{v}_k\\ \underline{y}_k=\begin{pmatrix}C_k & , & \mathbf{0}_{l\times m}\end{pmatrix}\begin{pmatrix}\underline{x}_k\\ \underline{u}_{k-1}\end{pmatrix}\end{cases}$$

hay

$$\begin{cases}\underline{z}_{k+1}=\mathcal{A}_k\underline{z}_k+\mathcal{B}_k\underline{v}_k\\ \underline{y}_k=\mathcal{C}_k\underline{z}_k\end{cases} \tag{11}$$

trong đó

$$\underline{z}_k=\begin{pmatrix}\underline{x}_k\\ \underline{u}_{k-1}\end{pmatrix},\ \mathcal{A}_k=\begin{pmatrix}A_k & B_k\\ \mathbf{0}_{n\times n} & I_{m\times m}\end{pmatrix},\ \mathcal{B}_k=\begin{pmatrix}B_k\\ I_{m\times m}\end{pmatrix},\ \mathcal{C}_k=\begin{pmatrix}C_k & , & \mathbf{0}_{l\times m}\end{pmatrix} \tag{12}$$

Rõ ràng, tại thời điểm k hiện tại, mô hình dự báo (11) trên mang đầy đủ đặc tính động học của hệ (1) ban đầu. Hơn nữa nó còn có m khâu tích phân trong đó, vì

$$\det\left(\lambda I_{(n+m)\times(n+m)}-\mathcal{A}_k\right)=\det\left(\lambda I_{n\times n}-A_k\right)(\lambda-1)^m=0$$

có nghiệm $\lambda=1$ bội m.

Sử dụng mô hình dự báo (11) này, trong đó các ma trận $\mathcal{A}_k$, $\mathcal{B}_k$, $\mathcal{C}_k$ được giữ cố định trong toàn khoảng dự báo $[k+1,k+N]$ hiện tại, tức là chúng không còn phụ thuộc $k+i$ trong cửa sổ dự báo đó, ta được

$$\begin{aligned}
\underline{z}_{k+1} &= \mathcal{A}_k \underline{z}_k + \mathcal{B}_k \underline{v}_k \\
\underline{z}_{k+2} &= \mathcal{A}_k \left(\mathcal{A}_k \underline{z}_k + \mathcal{B}_k \underline{v}_k\right) + \mathcal{B}_k \underline{v}_{k+1} = \mathcal{A}_k^2 \underline{z}_k + \mathcal{A}_k \mathcal{B}_k \underline{v}_k + \mathcal{B}_k \underline{v}_{k+1} \\
&\vdots \\
\underline{z}_{k+i} &= \mathcal{A}_k^i \underline{z}_k + \mathcal{A}_k^{i-1} \mathcal{B}_k \underline{v}_k + \cdots + \mathcal{A}_k \mathcal{B}_k \underline{v}_{k+i-2} + \mathcal{B}_k \underline{v}_{k+i-1} \\
&\vdots \\
\underline{z}_{k+N} &= \mathcal{A}_k^N \underline{z}_k + \mathcal{A}_k^{N-1} \mathcal{B}_k \underline{v}_k + \cdots + \mathcal{A}_k \mathcal{B}_k \underline{v}_{k+N-2} + \mathcal{B}_k \underline{v}_{k+N-1}
\end{aligned}$$

Vậy, từ (11) ta có vector tín hiệu đầu ra cho mọi $i = 1, \ldots, N$

$$\underline{y}_{k+i} = \mathcal{C}_k \mathcal{A}_k^i \underline{z}_k + \mathcal{C}_k \mathcal{A}_k^{i-1} \mathcal{B}_k \underline{v}_k + \cdots + \mathcal{C}_k \mathcal{A}_k \mathcal{B}_k \underline{v}_{k+i-2} + \mathcal{C}_k \mathcal{B}_k \underline{v}_{k+i-1}. \tag{13}$$

Từ đây, khi viết chung lại (13) cho tất cả các chỉ số $i = 1,2, \ldots, N$ được

$$\begin{pmatrix} \underline{y}_{k+1} \\ \vdots \\ \underline{y}_{k+N} \end{pmatrix} = \begin{pmatrix} \mathcal{C}_k \mathcal{A}_k \\ \vdots \\ \mathcal{C}_k \mathcal{A}_k^N \end{pmatrix} \underline{z}_k + \begin{pmatrix} \mathcal{C}_k \mathcal{B}_k & \mathbf{0}_{l\times m} & \cdots & \mathbf{0}_{l\times m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{C}_k \mathcal{A}_k^{N-1} \mathcal{B}_k & \mathcal{C}_k \mathcal{A}_k^{N-2} \mathcal{B}_k & \cdots & \mathcal{C}_k \mathcal{B}_k \end{pmatrix} \begin{pmatrix} \underline{v}_k \\ \vdots \\ \underline{v}_{k+N-1} \end{pmatrix}$$

Do đó, với các ký hiệu

$$\begin{aligned}
&\underline{\boldsymbol{y}} = \mathrm{col}\left(\underline{y}_{k+1}, \ldots, \underline{y}_{k+N}\right),\ \underline{\boldsymbol{v}} = \mathrm{col}\left(\underline{v}_k, \ldots, \underline{v}_{k+N-1}\right), \\
&\underline{\boldsymbol{r}} = \mathrm{col}\left(\underline{r}_{k+1}, \ldots, \underline{r}_{k+N}\right), \\
&\mathcal{Q} = \begin{pmatrix} Q_k & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & Q_k \end{pmatrix},\ \mathcal{R} = \begin{pmatrix} R_k & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & R_k \end{pmatrix},\ \mathcal{D} = \begin{pmatrix} \mathcal{C}_k \mathcal{A}_k \\ \vdots \\ \mathcal{C}_k \mathcal{A}_k^N \end{pmatrix}, \\
&\mathcal{F} = \begin{pmatrix} \mathcal{C}_k \mathcal{B}_k & \mathbf{0}_{l\times m} & \cdots & \mathbf{0}_{l\times m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{C}_k \mathcal{A}_k^{N-1} \mathcal{B}_k & \mathcal{C}_k \mathcal{A}_k^{N-2} \mathcal{B}_k & \cdots & \mathcal{C}_k \mathcal{B}_k \end{pmatrix}
\end{aligned} \tag{14}$$

thì hàm mục tiêu (4) trở thành

$$J_k(\underline{\boldsymbol{v}}) = \underline{\boldsymbol{v}}^T \left(\mathcal{F}^T \mathcal{Q} \mathcal{F} + \mathcal{R}\right) \underline{\boldsymbol{v}} - 2\left((\underline{\boldsymbol{r}} - \mathcal{D}\underline{z}_k)^T \mathcal{Q} \mathcal{F}\right) \underline{\boldsymbol{v}} \tag{15}$$

và là hàm toàn phương. Từ đó, nếu ràng buộc U còn là tập lồi, điều thường gặp trong thực tế khi các tín hiệu điều khiển luôn thuộc một khoảng đóng xác định cho trước, chẳng hạn như

$$\underline{\alpha}_k \le \underline{u}_k \le \underline{\beta}_k$$

với $\underline{\alpha}_k, \underline{\beta}_k$ là hai vector và phép so sánh vector được thực hiện cho các phần tử của chúng, thì bài toán tối ưu phi tuyến (7) bây giờ trở thành bài toán tối ưu lồi, có hàm mục tiêu là toàn phương, tập ràng buộc được mô tả bởi các bất đẳng thức tuyến tính (bài toán QP)

$$\underline{\boldsymbol{v}}^* = \arg\min_{\underline{v}\in V_k} J_k(\underline{\boldsymbol{v}}) \tag{16}$$

trong đó tập ràng buộc V_k được sinh ra từ U bằng cách dịch nó một khoảng $-\underline{u}_{k-1}$ trong không gian điều khiển $\mathbb{R}^m$, cũng là một tập lồi, mô tả bởi các bất đẳng thức tuyến tính. Cuối cùng, ta suy ra được từ $\underline{\boldsymbol{v}}^*$

$$\underline{u}_k = \underline{u}_{k-1} + \left(I_m, \mathbf{0}_{m\times m}, \ \ldots \ , \mathbf{0}_{m\times m}\right)\underline{\boldsymbol{v}}^*. \tag{17}$$

và đó là tín hiệu điều khiển cần tìm ở thời điểm k hiện tại (hình 1).

2.3. Thuật toán điều khiển

Để tiện cho việc cài đặt phương pháp đề xuất, sau đây ta sẽ đúc kết lại nội dung của nó đã trình bày ở trên dưới dạng thuật toán gồm các bước sau:

1. Khởi tạo: Chọn trước khoảng dự báo N. Khoảng dự báo được chọn này không được nhỏ hơn khoảng biết trước của tín hiệu đặt $\underline{r}_k$, nếu nó cũng chỉ được cho trước từng đoạn. Ở trường hợp đã có $\underline{r}_k$ với mọi $k \geq 0$ thì $N \geq 2$ là tùy chọn.

   Gán $k = 0$. Tùy gán $\underline{u}_{-1}$.

2. Thực hiện với $k = 0,1, \ \ldots$ các bước sau:

   • Đo trạng thái $\underline{x}_k$. Xác định $\underline{z}_k$, $\mathcal{A}_k$, $\mathcal{B}_k$, $\mathcal{C}_k$ theo (12).

   • Chọn hai ma trận đối xứng xác định dương R_k, Q_k rồi thiết lập $\mathcal{R}, \mathcal{Q}$ theo (14). Trong trường hợp chúng được chọn không phụ thuộc k thì bước này sẽ được chuyển vào phần khởi tạo.

   • Xây dựng các ma trận $\mathcal{D}, \mathcal{F}$ và vector $\underline{\boldsymbol{r}}$ theo (14). Xác định tập ràng buộc V_k từ U và $\underline{u}_{k-1}$.

   • Tìm nghiệm tối ưu $\underline{\boldsymbol{v}}^*$ của bài toán tối ưu toàn phương (16) với ràng buộc lồi, trong đó $J_k(\underline{\boldsymbol{v}})$ cho tại (15). Có thể thấy, nếu

$$\underline{\boldsymbol{v}}^* = \left(\mathcal{F}^T\mathcal{Q}\mathcal{F} + \mathcal{R}\right)^{-1}\mathcal{F}^T\mathcal{Q}(\underline{\boldsymbol{r}} - \mathcal{D}\underline{z}_k) \in V_k \tag{18}$$

thì nó là nghiệm toàn cục của bài toán tối ưu (16), ngược lại $\underline{v}^*$ chỉ có thể nằm trên biên của tập lồi V_k và tìm được nhờ thuật toán truyền thống của Karush-Kuhn-Tucker (KKT) [11]. Thuật toán này cũng đã được tích hợp trong MatLab dưới tên lệnh `fmincon` rất tiện sử dụng [12],[13].

• Tính $\underline{u}_k$ theo (17) và đưa vào điều khiển hệ (1).

Có thể thấy là hoàn toàn không khó khăn gì khi ta mở rộng thuật toán trên cho cả những hệ (1) affine nhưng có dạng tổng quát hơn ở đầu ra như sau:

$$\begin{cases} \underline{x}_{k+1} = \underline{f}(\underline{x}_k, k) + B(\underline{x}_k, k)\underline{u}_k \\ \underline{y}_k = \underline{g}(\underline{x}_k, k) \end{cases} \tag{19}$$

bằng cách đưa nó về dạng (1) với

$$C(\underline{x}_k, k) = \frac{\underline{g}(\underline{x}_k, k)}{\|\underline{x}_k\|^2} \underline{x}_k^T .$$

3. MINH HỌA BẰNG VÍ DỤ SỐ

Để minh họa chất lượng của bộ điều khiển đề xuất, có so sánh với bộ điều khiển MPC phi tuyến truyền thống, sau đây ta sẽ cài đặt cả hai phương pháp để đưa đầu ra y_k của hệ

$$\begin{aligned} \underline{x}_{k+1} &= \begin{pmatrix} x_{k_2} - x_{k_3} + u_k \cos x_{k_1} \\ 2x_{k_3} - x_{k_2} \\ x_{k_1} \sin x_{k_2} + x_{k_3} + u_k \end{pmatrix} \\ y_k &= \left| x_{k_1} - x_{k_2} \right| x_{k_1} + 2x_{k_2} \end{aligned} \tag{20}$$

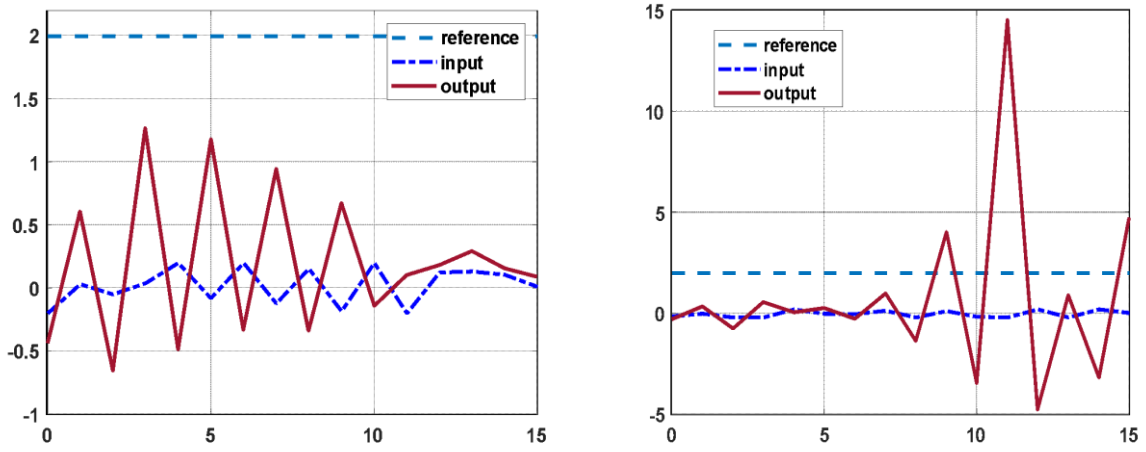
với vector trạng thái $\underline{x}_k = \left(x_{k_1}, x_{k_2}, x_{k_3}\right)^T$, tín hiệu đầu vào (tín hiệu điều khiển) u_k, bám theo được tín hiệu đặt $r_k = 2,\ \forall k$ và u_k thỏa mãn điều kiện ràng buộc dạng khoảng $\left|u_k\right| \le 0.2$. Hàm mục tiêu (3) tương ứng với chất lượng bám đặt ra này là

$$J_k\left(\underline{\mathbf{u}}_k\right) = \left\|\underline{\mathbf{u}}_k\right\|^2 + q \sum_{j=1}^{N} \left(\underline{y}_{k+j}(\underline{\mathbf{u}}_k) - \underline{r}_{k+j} \right)^2$$

trong đó $N \ge 1$ là độ rộng của sổ dự báo tùy chọn, $\underline{\mathbf{u}}_k = \left(u_k, u_{k+1}, \ \ldots\ , u_{k+N-1}\right)^T$ và hệ số $q = 5$ cho trước.

Trước tiên ta cài đặt thuật toán truyền thống như mô tả ở mục 2.2 thành chương trình `pxm1.m` viết trên MatLab, có mã nguồn như dưới đây, sẽ thu được kết quả như ở hình 2, cho

hai trường hợp $N=10$ và $N=20$ khác nhau. Có thể thấy ở cả hai trường hợp này đầu ra của hệ không bám theo được tín hiệu đặt cho trước. Thậm chí là khi N được chọn càng lớn, chất lượng càng xấu.:



a) Khi $N=10$ b) Khi $N=20$

Hình 2: Kết quả điều khiển với phương pháp truyền thống

```
% pxm1.m: Conventional method
global N x r q
op=optimoptions('fmincon','algorithm','sqp');
M=16; N=10; w=2; r=w*ones(N,1); q=5*ones(N,1);
ub=0.2*ones(N,1); lb=-ub; u=0; x=[0;0;0]; up=[]; yp=[]; t=[];
for i=1:M
    t=[t;i-1]; p0=u*ones(N,1);
    p=fmincon(@myobj,p0,[],[],[],[],lb,ub,[],op);
    u=p(1); [x,y]=mysys(u,x); up=[up;u]; yp=[yp;y]; % for
printing
end
plot(t,w*ones(M,1),t,up,t,yp);
legend('reference','input','output');
function J=myobj(u)    % objective function
    global N x r q
    J=0; xk=x;
    for i=1:N
        [xk,yk]=mysys(u(i),xk); J=J+q(i)*(r(i)-yk)^2+u(i)^2;
    end
end
function [x,y]=mysys(u,x)    % system dynamic
    x=[x(2)-x(3)+u*cos(x(1));2*x(3)-
x(2);x(1)*sin(x(2))+x(3)+u];
    y=abs(x(1)-x(2))*x(1)+2*x(1);
end
```

Tiếp theo, ta sẽ sử dụng phương pháp đề xuất mà ở đó hệ (20) đã được viết lại thành dạng tương đương

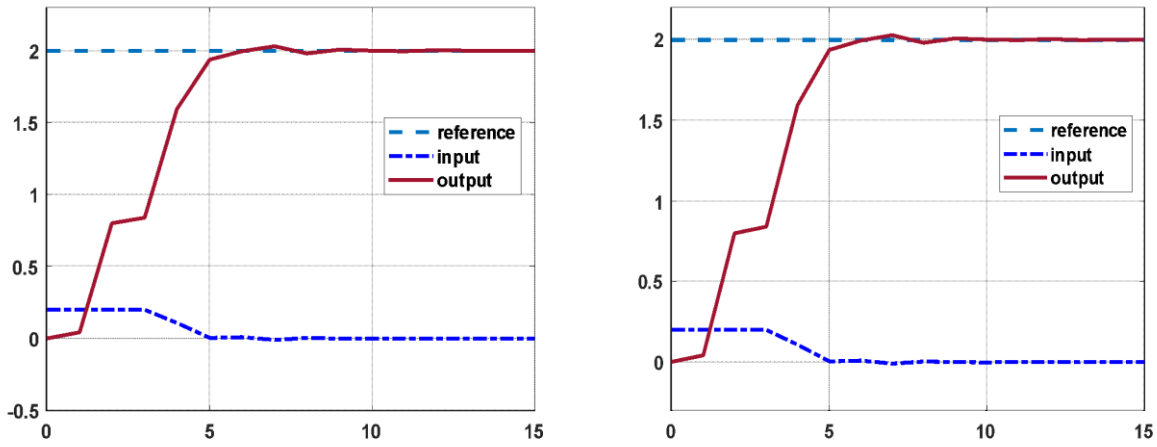
$$\underline{x}_{k+1} = \begin{pmatrix} x_{k_2} - x_{k_3} + u_k \cos x_{k_1} \\ 2x_{k_3} - x_{k_2} \\ x_{k_1} \sin x_{k_2} + x_{k_3} + u_k \end{pmatrix} = \begin{pmatrix} 0 & 1 & -1 \\ 0 & -1 & 2 \\ \sin x_{k_2} & 0 & 1 \end{pmatrix} \underline{x}_k + \begin{pmatrix} \cos x_{k_1} \\ 0 \\ 1 \end{pmatrix} u_k$$

$$y_k = \left| x_{k_1} - x_{k_2} \right| x_{k_1} + 2x_{k_2} = \left(\left| x_{k_1} - x_{k_2} \right| \ , \ 2 \ , \ 0 \right) \underline{x}_k$$

tức là nó có

$$A_k = \begin{pmatrix} 0 & 1 & -1 \\ 0 & -1 & 2 \\ \sin x_{k_2} & 0 & 1 \end{pmatrix}, \ B_k = \begin{pmatrix} \cos x_{k_1} \\ 0 \\ 1 \end{pmatrix}, \ C_k = \left(\left| x_{k_1} - x_{k_2} \right| \ , \ 2 \ , \ 0 \right)$$

Cài đặt thuật toán đề xuất đã giới thiệu ở mục 2.3 thành chương trình `pxm2.m` viết trên MatLab có mã nguồn chi tiết như ở dưới, ta thu được kết quả cho ở hình 3. Kết quả này cho thấy bộ điều khiển đề xuất đã đáp ứng được yêu cầu chất lượng bám đặt ra cho cả hai trường hợp $N = 10$ và $N = 20$. Hơn nữa nó còn có tốc tính toán nhanh hơn nhiều so với phương pháp truyền thống do chỉ cần tìm nghiệm của bài toán QP thay vì của bài toán quy hoạch phi tuyến.



a) Khi $N = 10$ b) Khi $N = 20$

Hình 3: Kết quả điều khiển với phương pháp đề xuất

```
% pxm2.m: Proposed algorithm
N=10; w=2; Q=5*eye(N); R=eye(N); x=[0;0;0]; u=[];y=[]; a=0.2;
op=optimoptions(@fmincon,'algorithm','sqp');
t=[]; r=w*ones(N,1);u0=0; M=16; O=[1 zeros(1,N-1)]; v=0;
%O=[1,0,...,0]
for k=1:M
    A=[0 1 -1;0 -1 2;sin(x(2)) 0 1];B=[cos(x(1));0;1];
    C=[abs(x(1)-x(2)) 2 0]; E=[];F=zeros(N,N);
    Ah=[A B;0 0 0 1]; Bh=[B;1]; Ch=[C 0];
    for i=1:N
        E=[E;Ch*Ah^i];
        for j=1:i F(i,j)=Ch*Ah^(i-j)*Bh; end
```

```
    end
    t(k)=k-1; y(k)=C*x; d=E*[x;u0]; G=F'*Q*F+R; b=F'*Q*(r-d);
    lb=-inf*ones(N,1); ub=-lb; lb(1)=-a-u0; ub(1)=a-u0;
    p0=v*ones(N,1);
    p=fmincon(@(p) p'*G*p-2*b'*p,p0,[],[],[],[],lb,ub,[],op);
    du=O*p; u(k)=u0+du; x=A*x+B*u(k); u0=u(k);
end
plot(t,w*ones(length(t),1),t,u,t,y);
legend('reference','input','output');
```

Một nhận xét thêm ở phương pháp đề xuất này là mặc dù khi N được chọn càng lớn, sai lệch giữa mô hình gốc (1) ban đầu với mô hình tuyến tính (9) càng cao do ở đó, trong cửa sổ dự báo hiện tại, ta đã giữ nguyên các ma trận tham số A_k, B_k, C_k, tức là đã gán $A_k = A_{k+i}$, $B_k = B_{k+i}$, $C_k = C_{k+i}$ với mọi $0 \le i \le N-1$, song ảnh hưởng của sai lệch mô hình này tới chất lượng điều là không đáng kể, vì kết quả u_k thu được cũng chỉ là phần tử đầu tiên trong dãy $\underline{\mathbf{u}}_k = \left(u_k, u_{k+1}, \ \ldots \ , u_{k+N-1}\right)$ của cửa sổ dự báo hiện tại.

4. KẾT LUẬN

Chất lượng bám tín hiệu đặt theo nghĩa $\left\|\underline{e}_k\right\| \to 0$ của phương pháp đề xuất đã được kiểm chứng bằng ví dụ mô phỏng. Chất lượng này có được là do ta đã sử dụng sự dịch chuyển của cửa sổ dự báo để chuyển từng bước bài toán quy hoạch phi tuyến không lồi thành bài toán quy hoạch toàn phương có ràng buộc là tập lồi, rồi tìm nghiệm toàn cục nhờ phương pháp QP trong cửa sổ dự báo đó, cũng như đã bổ sung thêm được m khâu tích phân vào mô hình dự báo. Tuy nhiên còn thiều lời chứng minh lý thuyết một cách chặt chẽ cho chất lượng thu được này và nó sẽ là nội dung nghiên cứu tiếp theo của tác giả.

TÀI LIỆU THAM KHẢO

[1] Ramine Rouhani, Raman K. Mehra (1982). *Model algorithmic control (MAC): Basic theoretical properties*. Automatica Volume 18, Issue 4, 401-414.

[2] P. Lundström; J.H. Lee; M. Morari; S. Skogestad (1995). *Limitations of dynamic matrix control*. Computers & Chemical Engineering, Volume 19, Issue 4, 409-421

[3] D.W. Clarke; C. Mohtadi; P.S. Tuffs (1987). *Generalized predictive control. The basic algorithm*. Automatica Volume 23, Issue 2, 137-148

[4] Rawlings JB (2000). *Tutorial overview of model predictive control*. IEEE Control Sys 20(3):38–52

[5] Camacho; Bordons (2004). *Model predictive control*. Springer Verlag

[6] Morari Manfred (2025). *Model Predictive Control: The Genesis of an Idea - Histories of Control*. IEEE Control Systems. 45 (4): 86–88

[7] Nguyễn Doãn Phước (2016). *Tối ưu hóa và điều khiển tối ưu*, Nhà xuất bản Bách khoa Hà Nội.

[8] Rawlings James B.; Mayne David Q. and Diehl, Moritz M. (2017). *Model Predictive Control: Theory, Computation, and Design (2nd Ed.)*. Nob Hill Publishing

[9] Mayne David Q.; Rawlings James B.; Rao Christopher V.; Scokaert Pierre O. M. (2000). *Constrained model predictive control: Stability and Optimality*. Automatica. 36 (6): pp. 789–814.

[10] A. L. Peressini; F. E. Sullivan and J. J. Uhl, Jr. (1988). *The Mathematics of Nonlinear Programming*. Springer-Verlag, New York.

[11] Nocedal,J. and Wright,S.J. (1996). *Numerical Optimization*. Springer-New York.

[12] S. J. Chapman (2004). *MATLAB Programming for Engineers*. Thomson, 2004

[13] A. Gilat (2004). MATLAB: *An introduction with Applications*. John Wiley and Sons.