This item was submitted to [Loughborough's Research Repository](#) by the author.
Items in Figshare are protected by copyright, with all rights reserved, unless otherwise indicated.

Piecewise constant model predictive control for autonomous helicopters

PLEASE CITE THE PUBLISHED VERSION

<http://dx.doi.org/10.1016/j.robot.2011.04.004>

PUBLISHER

© Elsevier

VERSION

AM (Accepted Manuscript)

LICENCE

CC BY-NC-ND 4.0

REPOSITORY RECORD

Liu, Cunjia, Wen-Hua Chen, and J.D. Andrews. 2012. "Piecewise Constant Model Predictive Control for Autonomous Helicopters". figshare. <https://hdl.handle.net/2134/11159>.

This item was submitted to Loughborough's Institutional Repository (<https://dspace.lboro.ac.uk/>) by the author and is made available under the following Creative Commons Licence conditions.

For the full text of this licence, please go to:
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

Piecewise Constant Model Predictive Control for Autonomous Helicopters

Cunjia Liu^{a,*}, Wen-Hua Chen^a, John Andrews^b

^a*Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough, LE11 3TU, UK.*

^b*Nottingham Transportation Engineering Centre, University of Nottingham, Nottingham, NG7 2RD, UK.*

Abstract

This paper introduces an optimisation based control framework for autonomous helicopters. The framework contains a high-level Model Predictive Control (MPC) and a low-level linear controller. The proposed MPC works in a piecewise constant fashion to reduce the computation burden and to increase the time available for performing online optimisation. The linear feedback controller responds to fast dynamics of the helicopter and compensates the low bandwidth of the high-level controller. This configuration allows the computationally intensive algorithm applied on systems with fast dynamics. The stability issues of the high-level MPC and the overall control scheme are discussed. Simulations and flight tests on a small-scale helicopter are carried out to verify the proposed control scheme.

Keywords:

Model Predictive Control, helicopter, autonomous flight, stability, flight test

1. Introduction

Autonomous helicopters are increasingly employed in military and civilian applications in the past decade, mainly due to their ability to hover, fly in very low altitudes, and take off and land almost everywhere. However, due to their strong nonlinearities, inherent instabilities and couplings in different channels, control design for autonomous flight of helicopters

*Corresponding author

Email addresses: `c.liu5@lboro.ac.uk` (Cunjia Liu), `w.chen@lboro.ac.uk` (Wen-Hua Chen), `john.andrews@nottingham.ac.uk` (John Andrews)

is a challenge. To this end, many control techniques have been applied to address this problem including the classic cascaded PID control [1], feedback linearisation [2], multivariable adaptive Control [3], neural network adaptive control [4], state-dependent riccati equation (SDRE) control [5], and composite nonlinear feedback control [6].

Recently, model predictive control (MPC) has drawn more and more attentions in the flight control field [7]. MPC is an optimal control strategy that uses a model to predict the future behaviour of a plant over a prediction horizon. Based on these predictions, an objective function defined to penalise tracking errors or state errors is minimised with respect to the sequence of future inputs. Only the first action in the resulting control sequence is applied into the plant, and this procedure is executed repeatedly to continuously generate control signals. Comparing to other control techniques, nonlinear MPC provides a number of unique advantages for autonomous helicopter flight:

- Be able to deal with nonlinear, multiple-input-multiple-output dynamics of helicopters by directly using their mathematical models in the control loop;
- Explicitly take into account states and control constraints to guarantee flight safety and prevent control saturation;
- The kinematics and dynamics of helicopters are considered as an entire system, which results in an integrated guidance and control fashion that enhances flight agility;
- Provide a local path planning function by combining future reference and environment information such as obstacles and collisions.

However, MPC techniques, especially nonlinear MPC, impose challenges in real-time implementation because a computational intensive nonlinear optimisation problem is required to solve at each sampling instant. To this end, lots of existing applications tend to use linear MPC so that the formulated optimisation problem (OP) can be solved by efficient Quadratic Programming [8, 9]. To fit a nonlinear control problem into a linear setting, associated techniques like online linearisation and feedback linearisation are usually required. For nonlinear MPC, although it is more powerful, the computational time restricts the control bandwidth

in a low range. Therefore, the nonlinear MPC is more likely to be seen in the guidance layer to enhance the autonomy of the UAVs rather than in the time-critical flight control layer [10, 11]. Nonlinear MPC also has been applied to the control of helicopters' group formation in [12], but only simulation results are provided.

Although with the development of microprocessor technology online optimisation becomes feasible, there are still difficulties in directly applying nonlinear MPC into plants with fast dynamics like helicopters. Initial trials on helicopters have been reported in [13, 14], but with limited prediction horizon. Nevertheless, the further reduction of computational burden in control algorithms can always benefit the application. The extra computation power can be put on extending the prediction horizon, including a more detailed model, and/or taking into account more information such as obstacles in the flight environment.

This paper proposes a control framework for autonomous helicopters, which explores the advantages of nonlinear MPC while being able to apply to systems with fast dynamics. Instead of attempting to implement a single nonlinear MPC, the proposed framework employs a two-level control structure where the high-level MPC generates baseline control by exploiting the nonlinear helicopter model and environment information, and the low-level linear controller, designed based on linearisation around the state reference provided by the high-level controller, compensates the baseline control in the presence of disturbances and uncertainties. These two level controllers are parallelly operated at different time scales. The high-level MPC strategy runs in a lower sampling rate allowing enough time to perform online optimisation, while the low-level controller is executed in a much higher sampling rate to respond to external disturbances.

One feature of the proposed control framework is that the high-level MPC adopts the piecewise constant control scheme. The modified algorithm allows the online optimisation occurring at scattered sampling instants without losing the prediction accuracy [15]. Moreover, the piecewise constant MPC significantly reduces the number of control variables to be decided (also known as optimisation variables), which helps to ease the workload of the online optimisation.

The stability of the control framework is investigated particularly focusing on the mod-

ified piecewise constant MPC. The design procedure of the terminal region and terminal penalty that guarantee the close-loop stability is discussed in detail and illustrated through an example. In addition, the stability analysis provides a way to construct a feasible initial control sequence for each optimisation, on which the stability is assured even if only the sub-optimal solution can be found during the online optimisation [16].

Another contribution in this work is to provide an experimental solution to realise the proposed control algorithm. Flight tests on a small-scale helicopter are carried out by utilising the indoor test facility where a small helicopter can perform manoeuvres under the control of a ground station. The ground station is constructed with the capabilities of observing the helicopter states and integrating the OP solver with the real-time controller.

The remaining part of this paper is organised as follows: Section 2 presents the general mathematical model of small-scale helicopters; in Section 3, the structure of the control framework is introduced, including the formulation of high-level piecewise constant MPC and the design of the low-level linear controller; Section 4 discusses the stability of the proposed MPC algorithm and a practical design procedure is presented; Section 5 devotes to the introduction of the simulation and flight test setting and flight test results, followed by a conclusion in section 6.

2. Helicopter model

The dynamics of general small-scale helicopters can be described by a six-degrees-of-freedom rigid-body model augmented with a simplified rotor dynamic model [17, 18]. The external forces and moments exerting on the helicopter are primarily generated by main and tail rotors thrusts, and fins and fuselage drags, which means that they are dependent on the rotor and helicopter states and the control inputs. In the control design the external forces and moments can be approximated by the linear combination of states and control inputs using stability and control derivatives, but the cross coupling among different channels

remains nonlinear relationship. The model structure is represented in (1).

$$\begin{aligned}
\begin{bmatrix} \dot{x} & \dot{y} & \dot{z} \end{bmatrix}' &= \mathbf{R}_b^i(\phi, \theta, \psi) \begin{bmatrix} u & v & w \end{bmatrix}' \\
\dot{u} &= vr - wq - g \sin \theta + X_u u + X_a a \\
\dot{v} &= wp - ur + g \cos \theta \sin \phi + Y_v v + Y_b b \\
\dot{w} &= uq - vp + g \cos \theta \cos \phi + Z_w w + Z_{col} \delta_{col} - g \\
\dot{p} &= -qr(I_{yy} - I_{zz})/I_{xx} + L_a a + L_b b \\
\dot{q} &= -pr(I_{zz} - I_{xx})/I_{yy} + M_a a + M_b b \\
\dot{r} &= -pq(I_{xx} - I_{yy})/I_{zz} + N_r r + N_{col} \delta_{col} + N_{ped} \delta_{ped} \\
\dot{\phi} &= p + q \sin \phi \tan \theta + r \cos \phi \tan \theta \\
\dot{\theta} &= q \cos \phi - r \sin \phi \\
\dot{\psi} &= q \sin \phi \sec \theta + r \cos \phi \sec \theta \\
\dot{a} &= -q - \frac{a}{\tau} + \frac{A_{lat}}{\tau} \delta_{lat} + \frac{A_{lon}}{\tau} \delta_{lon} \\
\dot{b} &= -p - \frac{b}{\tau} + \frac{B_{lat}}{\tau} \delta_{lat} + \frac{B_{lon}}{\tau} \delta_{lon}
\end{aligned} \tag{1}$$

where $\mathbf{x} = [x \ y \ z \ u \ v \ w \ p \ q \ r \ \phi \ \theta \ \psi]'$ is the state of the rigid-body of the helicopter consisting of inertial position, local velocity, angular rate and attitude, respectively; $\mathbf{R}_b^i$ is a transformation matrix from body to inertial coordinates given in (2) with short notation c for cosine and s for sine; $\mathbf{u} = [\delta_{lat} \ \delta_{lon} \ \delta_{ped} \ \delta_{col}]'$ is the control inputs including lateral and longitudinal cyclic, pedal and collective pitch respectively; the dynamics of the main rotor is described by the flapping angles $[a \ b]'$ with the effective time constant τ ; the other parameters in the model are the stability and control derivatives, whose values are obtained using system identification.

$$\mathbf{R}_b^i(\phi, \theta, \psi) = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \tag{2}$$

In this model, the rotor flapping states a and b cannot be directly measured, which usually relies on a state observer. In order to reduce the complexity and focus on MPC

design, we use steady state approximation as a measurement of the flapping angles [5]:

$$\begin{aligned} a &= -\tau q + A_{lat}\delta_{lat} + A_{lon}\delta_{lon} \\ b &= -\tau p + B_{lat}\delta_{lat} + B_{lon}\delta_{lon} \end{aligned} \tag{3}$$

By substituting (3) into (1), we can represent the helicopter model into a compact form:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \tag{4}$$

Note that a more detailed model can be used to describe helicopter dynamics without affecting the controller design. However, to facilitate the flight test we adopt the presented model, so that their parameters for the test helicopter can be estimated through system identification.

3. MPC based control framework

3.1. Piecewise constant MPC

The traditional MPC is either developed based on a continuous system model or a discrete counterpart. A continuous-time model is much more natural and accurate in terms of describing the behaviour of a system, but the corresponding MPC algorithm involves continuously solving OPs, which is difficult to implement as a computational time is required for online optimisation. In contrast, the discrete MPC uses the discrete representation of the system and makes online implementation feasible by solving OPs only at each sampling instant. The computational demand reduces when the discretization sampling time increases, but in turn the accuracy of approximated discrete representation degrades. Moreover, system states and constraints can only be evaluated at sampling instants leaving those within sampling intervals being ignored.

In this section, we introduce a modified MPC strategy that uses piecewise constant controls to drive a continuous system or an accurate discrete approximation. The piecewise constant control suggests that the control signals keep constant values (i.e. zero-order holding) for several discretization intervals, which makes the proposed algorithm different

from normal sampled data MPC or discrete time MPC that has been investigated by many researchers [19].

By trading-off between the prediction accuracy and computational burden, a discrete model approximated from a continuous model (4) with a high sampling frequency is chosen as the prediction model. The discretization sampling time is defined as T_d , which is also the integration step used in prediction. The error between the discrete model and continuous model increases monotonically with T_d . The MPC designed on a discrete model can stabilise the original continuous model if T_d is small enough [20]. However, the small T_d increases the computational burden, as there are more variables to be decided with respect to the same prediction length. To avoid this problem, another important parameter is introduced, namely the MPC sampling time T_s , defined as the interval of the MPC updating system states and generating a new control sequence. In conventional discrete MPC, $T_d = T_s$. However, in this study, there is a *control holding horizon* N with respect to T_d and T_s , such that $T_s = N \cdot T_d$. This also implies that the control inputs remains constant values over N integration steps T_d .

To clearly explain the time setting, an example is illustrated in Fig 1. The control holding horizon N is set to 4 steps, the same with the prediction horizon H . Within the period of T_s the control variables are set to constant, while the integration of system equations follows the discrete sampling time T_d . This setting maintains the accuracy of the prediction but significantly reduces the number of optimisation variables covering the same length of prediction. For example, when $N = 4$ and $H = 4$, in the conventional MPC there are $N \times H = 16$ variables to be optimised, but in the control holding scheme, only 4 variables need to be optimised.

Under the piecewise constant setting, a discrete MPC form is employed for the flight control. By defining the reference trajectory as $\mathbf{x}_r$ and the tracking error $\mathbf{x}_e = \mathbf{x}_r - \mathbf{x}$, the

Figure 1: Time setting example

performance index to be minimised can be formulated as:

$$J(k) = F(\mathbf{x}(k + HN)) + \sum_{i=0}^{H-1} \sum_{j=0}^{N-1} L(\mathbf{x}(k + iN + j), \mathbf{u}(k + iN + j)) \quad (5)$$

$$L(\mathbf{x}(k), \mathbf{u}(k)) = \mathbf{x}_e(k)' Q \mathbf{x}_e(k) + \mathbf{u}(k)' R \mathbf{u}(k)$$

$$F(\mathbf{x}(k + HN)) = \mathbf{x}_e(k + HN)' P \mathbf{x}_e(k + HN)$$

where, $L(\mathbf{x}(k), \mathbf{u}(k))$ is the penalty term for each integration step, $F(\mathbf{x}(k + HN))$ is the terminal penalty, H is the prediction horizon, N is the control holding horizon, and P , Q and R are the positive definite weighting matrices.

Remark 1. The control holding horizon N plays an important role in the modified MPC formulation. If $N = 1$ the modified MPC reverts to the conventional discrete MPC. For the given prediction length, increase of N can reduce the number of variables to be optimised, hence significantly reduce the computational burden. On the other hand, for a given number of optimisation variables that the online solver can handle, increase of N can extend the prediction length.

Remark 2. With the modified time setting the time allowed for OP solving is increased to T_s , while maintaining the resolution of the integration to T_d in the prediction. Note

that other MPC formulations may also have different sampling time for discretization and optimisation [10]. This is due to the heavy computational load rather than actively using the control holding mechanism to reduce the computational load.

The nonlinear OP that minimises the performance index (5) and subjects to helicopter dynamics and various constraints can be stated as:

$$\mathbf{x}_m, \mathbf{u}_m = \arg \min_{\hat{\mathbf{x}}, \hat{\mathbf{u}}} J(k) \quad (6)$$

subject to:

$$\begin{aligned} \hat{\mathbf{x}}(k+j+1) &= f(\hat{\mathbf{x}}(k+j), \hat{\mathbf{u}}(k+j)) \\ \hat{\mathbf{x}}(k+j) &\in \mathbf{X} \\ \hat{\mathbf{u}}(k+j) &\in \mathbf{U} \\ \hat{\mathbf{x}}(k+HN) &\in \mathbf{\Omega} \\ j &= 0, 1, \dots, N-1 \\ \hat{\mathbf{x}}(k) &= \mathbf{x}(k) \end{aligned}$$

where $\mathbf{x}(k+1) = f(\mathbf{x}(k), \mathbf{u}(k))$ is the discrete form of the helicopter dynamics with the discretization time T_d ; $\mathbf{X}$, $\mathbf{U}$ and $\mathbf{\Omega}$ are control constraints, state constraints and terminal region, respectively. The hat symbol is used to indicate the variables in the prediction distinguishing from the real variables. This optimisation problem is solved at each sampling instant, producing the state reference $\mathbf{x}_m$ and the baseline control sequence $\mathbf{u}_m$, in which the first element is applied to control the helicopter.

3.2. Two-level control framework

The proposed MPC scheme eases the computational burden by (i) increasing the computational interval to give more time for optimisation and (ii) reducing the number of variables to be optimised. However, the MPC strategy becomes an open-loop optimal control within the interval T_s . Unfortunately, due to the mismatch between the mathematical model and the real helicopter, the noises and disturbances in the process, this kind of optimal control would not perform as being designed. Within the interval T_s , the MPC cannot suppress any

model can be linearised around the nominal reference and input as:

$$\begin{aligned}\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \approx f(\mathbf{x}_m, \mathbf{u}_m) &+ \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\mathbf{x}_m, \mathbf{u}_m} (\mathbf{x} - \mathbf{x}_m) \\ &+ \left. \frac{\partial f}{\partial \mathbf{u}} \right|_{\mathbf{x}_m, \mathbf{u}_m} (\mathbf{u} - \mathbf{u}_m)\end{aligned}\quad (7)$$

By defining the error state $\mathbf{x}_\Delta = \mathbf{x} - \mathbf{x}_m$ and control compensation $\mathbf{u}_\Delta = \mathbf{u} - \mathbf{u}_m$. The system (7) can be stated as a parameter dependent system (8).

$$\dot{\mathbf{x}}_\Delta = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\mathbf{x}_m, \mathbf{u}_m} \mathbf{x}_\Delta + \left. \frac{\partial f}{\partial \mathbf{u}} \right|_{\mathbf{x}_m, \mathbf{u}_m} \mathbf{u}_\Delta = A(\mathbf{x}_m, \mathbf{u}_m) \mathbf{x}_\Delta + B(\mathbf{x}_m, \mathbf{u}_m) \mathbf{u}_\Delta \quad (8)$$

Considering a static output feedback K , the close-loop system can further be expressed as:

$$\dot{\mathbf{x}}_\Delta = (A(\cdot) - B(\cdot)K) \mathbf{x}_\Delta = A_{cl}(\cdot) \mathbf{x}_\Delta \quad (9)$$

The parameters in $A_{cl}(\cdot)$ are dependent on $\mathbf{x}_m$ and $\mathbf{u}_m$, which are bounded in the high-level optimisation. Hence, the system (9) can be converted into a polytopic system with its vertices computed by the uncertainty parameters with defined boundary values, and the robust stability of such a system can be guaranteed by using the parameter dependent Lyapunov function technique [21, 22].

4. Stability analysis

4.1. Stability of the piecewise constant MPC

The stability of proposed piecewise constant MPC is investigated by using Lyapunov technique inspired by [15]. We consider the helicopter performing hovering flight which is a typical flight mode for helicopters. Since all the states in the hovering are zeros, the error state $\mathbf{x}_e$ in (5) can be replaced by $\mathbf{x}$. Next, we need to define a terminal region Ω and an associated terminal controller $k_f(\cdot)$ satisfying a number of assumptions:

Assumption 1. The terminal region is a neighbourhood of the origin where state constraints are satisfied in this region. $0 \in \Omega$, and $\Omega \in \mathbf{X}$ is closed.

Assumption 2. There exists a terminal controller $k_f(\cdot)$ such that control constraints are satisfied for all the states in the terminal region. $k_f(x) \in \mathbf{U}, \forall x \in \mathbf{\Omega}$.

Assumption 3. N step evolution of the system under the terminal control $k_f(\cdot)$ stays in the terminal region. $\varphi_N(x, k_f(x)) \in \mathbf{\Omega}, \forall x \in \mathbf{\Omega}$, where $\varphi_i(x, u)$ denotes the states of the system at i step from an initial state x under the constant control signal $u = k_f(\cdot)$.

Theorem 1. *Suppose that the Assumptions 1-3 are satisfied, and the optimisation problem (6) has a solution at beginning, then there is a feasible control sequence at time instant $k+N$*

$$\mathbf{u}_{k+N:k+HN} = \{u_{k+N}, u_{k+2N}, \dots, u_{k+(H-1)N}, k_f(x_{k+HN})\} \quad (10)$$

where, $u_{k+i \cdot N}, i = 0, \dots, H-1$, and x_{k+HN} are from the previous control sequence and terminal state at time instant k , respectively. Moreover, the closed-loop system is stable if Eq.(10) is used as an initial solution in online optimisation and the following stability condition is satisfied:

$$\|x_{k+(H+1)N}\|_P^2 - \|x_{k+HN}\|_P^2 + \sum_{i=0}^{N-1} \|x_{k+H \cdot N+i}\|_Q^2 + N \cdot \|u_{k+H \cdot N}\|_R^2 < 0 \quad (11)$$

Proof. Considering the evolution of the system along time line, at sampling instant k the optimised performance index is denoted as:

$$\bar{V}_m(x_k) = \|\bar{x}_{k+HN}\|_P^2 + \sum_{j=0}^{H-1} \sum_{i=0}^{N-1} \|\bar{x}_{k+j \cdot N+i}\|_Q^2 + \sum_{j=0}^{H-1} N \cdot \|\bar{u}_{k+j \cdot N}\|_R^2 \quad (12)$$

where $\bar{x}_{k+j \cdot N+i}$ denotes the optimised states evolving from the time instant k , and $\bar{u}_{k+j \cdot N}$ denotes the control sequence that is generated in a zero-holding fashion. Note that the bar symbol is used to indicate a variable with the optimised value.

Following Assumptions 1-3, it can be noted that $\bar{x}_{k+HN} \in \mathbf{\Omega}$, $k_f(\bar{x}_{k+HN}) \in \mathbf{U}$ and $x_{k+HN+N} = \varphi_N \in \mathbf{\Omega}$. Then, at the next MPC sampling instant $k+N$ the initial control defined by Eq.(10) is feasible. Thus, the corresponding performance index at time instant

$k + N$ is:

$$\begin{aligned}
V_m(x_{k+N}) &= \|x_{k+HN+N}\|_P^2 + \sum_{j=0}^{H-1} \sum_{i=0}^{N-1} \|x_{k+(j+1) \cdot N+i}\|_Q^2 + \sum_{j=0}^{H-1} N \cdot \|u_{k+(j+1) \cdot N}\|_R^2 \\
&= \|x_{k+(H+1)N}\|_P^2 + \sum_{j=1}^H \sum_{i=0}^{N-1} \|x_{k+j \cdot N+i}\|_Q^2 + \sum_{j=1}^H N \cdot \|u_{k+j \cdot N}\|_R^2
\end{aligned} \tag{13}$$

When there is no error between the model and the real plant and in the absence of disturbances, the measured state x_{k+N} at instant $k + N$ should equal to the state $\bar{x}_{k+N}$ predicted at instant k . Therefore, we can inspect the following relationship.

$$\begin{aligned}
&V_m(x_{k+N}) - \bar{V}_m(x_k) \\
&= \|x_{k+(H+1)N}\|_P^2 - \|x_{k+HN}\|_P^2 + \sum_{j=1}^H \sum_{i=0}^{N-1} \|x_{k+j \cdot N+i}\|_Q^2 - \sum_{j=0}^{H-1} \sum_{i=0}^{N-1} \|x_{k+j \cdot N+i}\|_Q^2 \\
&\quad + \sum_{j=1}^H N \cdot \|u_{k+j \cdot N}\|_R^2 - \sum_{j=0}^{H-1} N \cdot \|u_{k+j \cdot N}\|_R^2 \\
&= \|x_{k+(H+1)N}\|_P^2 - \|x_{k+HN}\|_P^2 + \sum_{i=0}^{N-1} \|x_{k+H \cdot N+i}\|_Q^2 + N \cdot \|u_{k+H \cdot N}\|_R^2 \\
&\quad - \sum_{i=0}^{N-1} \|x_{k+i}\|_Q^2 - N \cdot \|u_k\|_R^2
\end{aligned} \tag{14}$$

Note that at time instant $k + N$, the online optimisation will be performed to minimise the cost function initialised by (13). Thereby the optimised performance index implies:

$$\bar{V}_m(x_{k+N}) \leq V_m(x_{k+N}) \tag{15}$$

Next, $V_m(x)$ is adopted as the Lyapunov function for the proposed piecewise constant MPC. The stability condition based on Lyapunov theory requires:

$$\bar{V}_m(x_{k+N}) \leq \bar{V}_m(x_k) \tag{16}$$

which means the Lyapunov function is non-increasing. To achieve this, we can impose a condition on (14) following Eq.(15), such that:

$$V_m(x_{k+N}) - \bar{V}_m(x_k) < 0 \tag{17}$$

Furthermore, by recalling Eq.(14) the stability condition (11) is derived. $\square$

Remark 3. The terminal controller will never apply to the real system, but can be used to construct the initial solution of the OP through Eq(10). This feasible solution always exists as long as the initial feasible solution exists, and it speeds up the convergence of the optimisation during the implementation.

To complete the stability analysis, the next step is to find a terminal penalty P , a suitable terminal region and an associated terminal control $k_f(\cdot)$ such that $\forall x_{k+HN} \in \Omega$ condition (11) and Assumptions A1-A3 are fulfilled.

4.2. Terminal region and controller

In this subsection, the remaining problems in the previous subsection are solved. Assuming the linearised discrete model in the hovering mode is as follows (note that the notation with the index in the round bracket is used to represent the states of the linearised model):

$$x(k+1) = \left. \frac{\partial f}{\partial x} \right|_{0,0} x(k) + \left. \frac{\partial f}{\partial u} \right|_{0,0} u(k) = Ax(k) + Bu(k) \quad (18)$$

Because the control inputs remain constant during the MPC sampling time $T_s = N \cdot T_d$ as discussed before, we can obtain that:

$$\begin{aligned} x(k+2) &= A^2x(k) + ABu(k) + Bu(k) \\ &\vdots \\ x(k+N) &= A^Nx(k) + (A^{N-1}B + \dots + AB + B)u(k) \end{aligned} \quad (19)$$

By defining two matrices $A_i = A^i$ and $B_i = A^{i-1}B + \dots + AB + B$, a compact form of (19) is found:

$$x(k+i) = A_ix(k) + B_iu(k) \quad (20)$$

Moreover, if one can find a linear terminal control

$$u(k) = -k_fx(k), \quad \forall x(k) \in \Omega \quad (21)$$

the equation (20) can be further written as:

$$x(k+i) = A_ix(k) + B_ik_fx(k) = A_i^{cl}x(k) \quad (22)$$

where A_i^{cl} is the close-loop system matrix. When designing the terminal controller k_f , the linear discrete control theory can be used. The resulting control has to guarantee that the eigenvalues of A_N^{cl} stay in the unit disc.

Then, from a terminal state $x_k = x(k)$ to i step of the evolution under the terminal control, the difference between the nonlinear model states x_{k+i} and linearised model states $x(k+i)$ can be described by:

$$\Phi_i(x_k) = \varphi_i(x_k, k_f x_k) - A_i^{cl} x_k \quad (23)$$

By invoking (23), one can derive the following relationship from Eq(11):

$$\begin{aligned} & \|x_{k+(H+1)N}\|_P^2 - \|x_{k+HN}\|_P^2 + \sum_{i=0}^{N-1} \|x_{k+H \cdot N+i}\|_Q^2 + N \cdot \|u_{k+H \cdot N}\|_R^2 \\ &= \|\Phi_N(x_{k+HN}) + A_N^{cl} x_{k+HN}\|_P^2 - \|x_{k+HN}\|_P^2 \\ &+ \sum_{i=0}^{N-1} \|\Phi_i(x_{k+HN}) + A_i^{cl} x_{k+HN}\|_Q^2 + N \cdot \|u_{k+H \cdot N}\|_R^2 \\ &= \|\Phi_N(x_{k+HN})\|_P^2 + 2 \cdot \Phi_N(x_{k+HN})' P A_N^{cl} x_{k+HN} + \|A_N^{cl} x_{k+HN}\|_P^2 - \|x_{k+HN}\|_P^2 \\ &+ \sum_{i=0}^{N-1} \|\Phi_i(x_{k+HN})\|_Q^2 + \sum_{i=0}^{N-1} 2 \cdot \Phi_i(x_{k+HN})' Q A_i^{cl} x_{k+HN} \\ &+ \sum_{i=0}^{N-1} \|A_i^{cl} x_{k+HN}\|_Q^2 + N \cdot \|u_{k+H \cdot N}\|_R^2 \end{aligned} \quad (24)$$

Notice that in the optimisation problem (6), the terminal state x_{k+HN} is forced in the terminal region Ω , which can be specified as the neighbourhood of the origin with the radius α :

$$\Omega(k_f, N) = \{x \in \mathbf{R}^n : \|x\|_P^2 < \alpha\} \quad (25)$$

It follows from the definition of $\Phi_i(x_k)$ that it depends on a high order of x , so that $\|\Phi_i(x_k)\|$ is approaching zero faster than $\|x_k\|$ when the radius of the terminal region α approaching zero. Therefore, for a small enough α , a positive scalar γ and a positive definite matrix $\tilde{Q}$, where eigenvalues $\lambda(\tilde{Q}) > \lambda(Q)$, can be found, such that:

$$\|\Phi_N(x_{k+HN})\|_P^2 + 2 \cdot \Phi_N(x_{k+HN})' P A_N^{cl} x_{k+HN} \leq \gamma \|x_{k+HN}\|^2 \quad (26)$$

and

$$\|\Phi_i(x_{k+HN})\|_Q^2 + 2 \cdot \Phi_i(x_{k+HN})' P A_i^{cl} x_{k+HN} + \|A_i^{cl} x_{k+HN}\|_Q^2 \leq \|A_i^{cl} x_{k+HN}\|_{\tilde{Q}}^2 \quad (27)$$

By substituting (26), (27) and terminal control (21) into the stability condition (24), one can derive:

$$\begin{aligned} & \|x_{k+(H+1)N}\|_P^2 - \|x_{k+HN}\|_P^2 + \sum_{i=0}^{N-1} \|x_{k+H \cdot N+i}\|_Q^2 + N \cdot \|u_{k+H \cdot N}\|_R^2 \\ & \leq \|A_N^{cl} x_{k+HN}\|_P^2 - \|x_{k+HN}\|_P^2 + x_{k+HN}' \left(\sum_{i=0}^{N-1} \|A_i^{cl}\|_{\tilde{Q}}^2 \right) x_{k+HN} \\ & \quad + N \cdot \|k_f x_{k+HN}\|_R^2 + \gamma \|x_{k+HN}\|^2 \\ & = x_{k+HN}' \left(A_N^{cl'} P A_N^{cl} - P + \sum_{i=0}^{N-1} \|A_i^{cl}\|_{\tilde{Q}}^2 + N \cdot k_f' R k_f + \gamma I_n \right) x_{k+HN} \end{aligned} \quad (28)$$

Solving the discrete Lyapunov equation

$$A_N^{cl'} P A_N^{cl} - P + \sum_{i=0}^{N-1} \|A_i^{cl}\|_{\tilde{Q}}^2 + N \cdot k_f' R k_f + \gamma I_n = 0 \quad (29)$$

yields the terminal penalty weighting matrix P , which is also used to defined the terminal region in (25).

At this stage, Assumptions 1-3 and condition (11) are fulfilled by properly choosing control parameters in the MPC design procedure, which is summarized as follows:

- Step 1. Determine the process weighting matrix Q and control weighting matrix R based on performance specification.
- Step 2. Determine control holding horizon N according to computational power and burden, and calculate the corresponding linearised system matrices A_i and B_i .
- Step 3. Design the terminal control gain k_f with respect to A_i and B_i , with eigenvalues of A_N^{cl} staying in the unit disc .
- Step 4. Determine γ and $\tilde{Q}$, and solve the Lyapunov equation (29) to obtain the terminal weighting matrix P .

- Step 5. Determine the terminal region radius α . For any states within the terminal region, check if the resulting terminal control and evolved states $\varphi_i(x, k_fx)$ stay in the corresponding constraints.
- Step 6. Check the maximum value of expression (11) with respect to all the states in the terminal region. If the value is larger than zero, reduced the radius α and repeat this process until its value is smaller than zero.

If Step 5 or Step 6 are failed or the resulting radius of the terminal region is too small, one needs to go back to tune the design parameters (usually by increasing γ or reducing N) to obtain a proper terminal control gain and terminal region that guarantee the stability of the piecewise constant MPC. The process outlined in [23] can be adopted to implement Step 5 and 6. This procedure can be further improved by adopting the method proposed in [24], where terminal region can be maximised using an optimisation algorithm to search the best terminal penalty P and terminal control.

4.3. Behaviour of the two-level control framework

The control framework for the autonomous helicopter consists of two parts: high-level MPC and low-level linear controller. Although the stability of the high-level MPC and the low-level linear controller has been discussed separately, the stability of the overall system needs to be investigated. Since the control signals are eventually produced in a high sampling rate, the discussion remains in the continuous domain.

In the absence of uncertainty and disturbance, the helicopter state $\mathbf{x}(t)$ always follows MPC trajectory $\mathbf{x}_m(t)$. In reality, within a MPC sampling interval $t \in [t_k, t_{k+1})$, the actual helicopter state $\mathbf{x}(t)$ will diverge from $\mathbf{x}_m(t)$ under the constant control $\mathbf{u}_m(t_k)$ due to uncertainties and/or disturbances. However, within a surrounding area of $\mathbf{x}_m(t)$, the error dynamics can be described by a linearised system (8), where the surrounding is defined as $\Omega(\mathbf{x}_m) = \{\mathbf{x} \mid \|\mathbf{x} - \mathbf{x}_m\| \leq \bar{e}\}$, where $\bar{e}$ is small positive scalar.

Theorem 2. *The two-level control ensures that for any perturbed state $\mathbf{x}(t) \in \Omega(\mathbf{x}_m)$, $t \in (t_k, t_{k+1})$, caused by a bounded constant disturbance $\mathbf{d}$, its divergence $\mathbf{x}_\Delta(t)$ is bounded for all $t > 0$.*

Proof. The behaviour of the divergence $\mathbf{x}_\Delta$ can be modified from (9) as:

$$\dot{\mathbf{x}}_\Delta = A_{cl}\mathbf{x}_\Delta + E\mathbf{d} \quad (30)$$

where E is a disturbance input matrix. As at the time $t = t_k$, the high-level MPC reset the error state to zero, i.e. $\mathbf{x}_\Delta = 0$, the state response in the period $t \in (t_k, t_{k+1})$ can be written as:

$$\mathbf{x}_\Delta(t) = A_{cl}^{-1}(e^{A_{cl}(t-t_k)} - I)E\mathbf{d} \quad (31)$$

Therefore, there exists a bound on $\mathbf{x}_\Delta(t)$, which may depend on the magnitude of the disturbance. Furthermore, the high-level MPC resets the error state to zero at $t = t_{k+1}$, i.e. $\mathbf{x}_\Delta(t_{k+1}^+) = 0$. After MPC synchronises the real helicopter state $\mathbf{x}$ and MPC state $\mathbf{x}_m$, the overall stability is guaranteed as MPC runs in an closed-loop fashion (with a longer sampling time), rather than in an open-loop fashion. $\square$

The low-level controller can improve the performance of the high-level MPC in the presence of uncertainties and disturbances, without scarifying the original MPC stability. This phenomenon can be further explained in Fig 3, where the dash lines represent the MPC solutions, the dotted line is the real state of the system and the reference trajectory is plotted as the solid line on the top. At sampling instant $t_k = kT_s$ there is a trajectory $\mathbf{x}_m(t)$, $t_k \leq t \leq t_{k+1}$, yielded by the high-level MPC, towards the reference x_r . If the disturbance occurs or there is model mis-matching, there is discrepancy between the real helicopter and MPC trajectories, i.e. $\mathbf{x}_\Delta \neq 0$. The low-level controller generates compensation controls to eliminate $\mathbf{x}_\Delta \neq 0$. Even if it cannot be reduced to zero within a short time period, at next MPC sampling point $t_{k+1} = (k+1)T_s$ the MPC measures the current state of the system and produces a new trajectory $\mathbf{x}_m(t)$, $t_{k+1} \leq t \leq t_{k+1+H}$ towards x_r , and $\mathbf{x}_\Delta(t_{k+1}^+)$ is reset to zeros automatically.

5. Flight test

5.1. Test facility

The hummingbird helicopter was used as a testbed to verify the proposed MPC based control framework. It is a 300-class indoor helicopter with the rotor diameter of 0.36m.

Figure 3: State trajectory under the disturbance

The flight experiments are carried out on the flight test facility consisting of small-scale helicopters, Vicon motion capture system and ground station (see Fig 4). This test platform adopts a number of pieces of commercial-off-the-shelf equipment and combines them effectively into the Matlab environment, which provides a seamless way from control analysis and design to numerical simulation and experimental validation.

Figure 4: Test environment

The implementation of the two-level control framework in real-time is achieved by integrating Matlab and its xPC target real-time environment. The low-level controller is located on xPC target, and the high-level controller is executed on another computer in the Matlab environment where OPs are solved online. Information exchange between them relies on the local area network (LAN) with UDP protocol (see Fig 5). The synchronisation between the two computers is guaranteed by the real-time xPC target application calling Matlab

program based on its own timer. The xPC target also integrates interfaces to the Vicon tracking system and the radio controller which measures the helicopter states and sends control signals, respectively.

Figure 5: System architecture

5.2. Controller design

After a number of trials, the MPC parameters used for flight tests presented in this paper are shown in Table 1. The design procedure developed in Section 4 is illustrated by an example where only the lateral and longitudinal channels are considered as they contain the majority of the helicopter nonlinearity and avoid to illustrate the complicated full helicopter states. The auxiliary parameters used to determine the terminal region and control are as follows: $\tilde{Q} = 1.1 \cdot Q$, $\gamma = 0.2$ and $\alpha = 25$. The terminal penalty matrix P is calculated by solving the Lyapunov equation (29), whereas the terminal controller is designed by using

the standard Linear Quadratic Regulator (LQR) algorithm based on Eq.(20).

$$P = \begin{bmatrix} 23.54 & 0.01 & 11.86 & 0.02 & -0.56 & -1.70 & -0.77 & -35.48 \\ 0.01 & 23.55 & 0.01 & 12.71 & 2.30 & -0.55 & 36.91 & -0.95 \\ 11.86 & 0.01 & 33.12 & 0.02 & -1.54 & -4.38 & -2.27 & -95.00 \\ 0.02 & 12.71 & 0.02 & 37.80 & 6.70 & -1.61 & 109.79 & -2.77 \\ -0.56 & 2.30 & -1.54 & 6.70 & 8.39 & -0.22 & 35.26 & 4.18 \\ -1.70 & -0.55 & -4.38 & -1.61 & -0.22 & 8.04 & -5.20 & 28.63 \\ -0.77 & 36.91 & -2.27 & 109.80 & 35.26 & -5.20 & 591.95 & -2.38 \\ -35.48 & -0.95 & -95.00 & -2.77 & 4.18 & 28.63 & -2.38 & 548.86 \end{bmatrix}$$

$$k_f = \begin{bmatrix} 0.0042 & -0.1392 & 0.0116 & -0.4463 & -0.4542 & 0.0926 & -3.3903 & -0.0912 \\ 0.1367 & 0.0038 & 0.3805 & 0.0121 & -0.0529 & -0.4365 & 0.0980 & -3.0768 \end{bmatrix}$$

Table 1: MPC design parameters

Prediction horizon H	10
Control holding horizon N	5
Discretization time T_d	0.02s
MPC sampling time T_s	0.1s
Weighting matrix Q	$diag(0.1 \ 0.1 \ 1 \ 1 \ 2 \ 2 \ 2 \ 2)$
Weighting matrix R	$diag(0.02 \ 0.02)$

The terminal region yielded is in a high dimension space which cannot be illustrated directly. To show that Assumption A1-A3 are fulfilled in the terminal region, we assume the helicopter is in the hovering status and only has the position error. In this case the terminal region reduce to a constraint on the position:

$$\begin{bmatrix} x & y \end{bmatrix} \cdot \begin{bmatrix} 23.54 & 0.01 \\ 0.01 & 23.55 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \leq 25 \quad (32)$$

Then the position phase portrait under the terminal control can be calculated as in Fig 6. It can be seen that position trajectories of the helicopter are driven to zero under the terminal control. During this process the other states are also bounded.

Figure 6: Position phase portrait

5.3. Simulation

Before the real flight test, simulations are first carried out. Since there is no disturbance in numerical simulations, the high-level MPC along is able to control the helicopter. The aim of numerical simulations is to investigate the computational attributes of the proposed MPC scheme, and to compare with the conventional MPC.

One simulation is to track a square trajectory containing sharp 90° turns, which pose extra burdens on the OP solver as it has to replan a smooth trajectory that fits the helicopter dynamics. The time setting for piecewise constant MPC is $T_d = 0.02s$, $H = 10$, and $N = 5$. Therefore, the MPC sampling time is $T_s = 0.1s$ and the prediction length is 1s. On the other hand, in a conventional MPC when $N = 1$, one has to increase H to 50 steps to cover the same prediction length. The model of full helicopter dynamics (1) is used in prediction. From Fig.7, it can be seen that the piecewise constant and conventional MPC gives almost the same tracking performance. However, the computational burdens in two MPC schemes are quite different. Fig.8 compares the computation time spent at each sampling instant

along the simulation time. It is shown that in piecewise constant scheme the calculation time is around 0.05s and the maximal value is below the sampling interval suggesting that it is suitable for online execution. In contrast, the conventional MPC scheme needs more time to solve the OP as more variables need to be handled, which means it has to sacrifice the control bandwidth and prediction horizon in order to be applied in reality.

Figure 7: Square tracking

5.4. Test results

Many flight tests have been carried out in our flight testbed to verify the proposed controller in different scenarios, one of which presented in this section is to execute the same flight pattern used in the simulation that tracks a square trajectory with 2m length. The reference progresses at a constant speed of 1m/s, so the helicopter needs to complete the whole maneuver in 8 seconds. Moreover, this reference requires the helicopter starts from stationary at one corner and finishes in stationary at the next corner and then keeps going in a different direction. **This trajectory is dynamically infeasible for helicopters, but it is deliberately used to demonstrate the prediction feature of MPC, which uses online**

Figure 8: Computational time

optimisation to generate a smooth trajectory allowing the helicopter to fly along the reference as close as possible and keep stable.

The tracking result is shown in Fig.9 in a 3 dimensional plot. In the flight test helicopter was controlled to hover at the start point first and started to track after 40s. During this process, the roll angle and pitch angle are cooperated to increase the translational speed at one direction and decrease at another as shown in Fig.10. Note that a positive roll angle gives a positive lateral acceleration and a positive pitch angle generates a negative longitudinal acceleration, vice versa. The corresponding control signals are provided in Fig.11, where the baseline control from high-level is plotted in solid line, whereas the overall control is given in dash line. It can be observed that the high-level MPC gives the basic trend of the control signal and the low-level controller adds compensations on it to achieve the required control.

The flight test demonstrates an excellent tracking performance under the proposed two-level control scheme. It shall be reminded that control of a small-scale helicopter is even more difficult than a large one as small ones are quite sensitive to any wind gust and turbulence,

Figure 9: Flight tracking result

Figure 10: Attitude angles

Figure 11: Control signals

and any small change in helicopter structure and propulsion systems. To this end, a quite good robustness of the proposed scheme has been clearly demonstrated in the flight tests. The model used in MPC online calculation is simplified and the parameters are estimated through system identification. There are certainly mis-matching between the model and the real helicopter dynamics.

6. Conclusion

Development of a control system to support autonomous flight of helicopters is very challenging as helicopters are unstable, highly nonlinear and exhibit fast dynamics. This paper proposes a MPC based control framework for autonomous flight of small-scale helicopter. The framework has two levels of controls including a high-level MPC and low-level linear feedback control. The MPC works in a piecewise constant fashion to reduce the computation burden and to increase the time available for real-time optimisation. The linear feedback control responds to fast dynamics of the helicopter in the presence of disturbances and model mis-matching and compensate the low bandwidth of the high-level control due to

the adoption of the piecewise constant control policy. With this configuration, it is possible to implement nonlinear MPC algorithms in system with fast dynamics such as helicopters. The stability issue of the high-level MPC and the overall control scheme are discussed and the design procedure is provided. The overall control framework was successfully tested on a hummingbird helicopter through various flight experiments, and satisfactory performance has been demonstrated.

References

- [1] H. J. Kim, D. H. Shim, A flight control system for aerial robots: algorithms and experiments, *Control Engineering Practice* 11 (12) (2003) 1389 – 1400, award winning applications-2002 IFAC World Congress.
- [2] T. Koo, S. Sastry, Output tracking control design of a helicopter model based on approximate linearization, in: *Decision and Control, 1998. Proceedings of the 37th IEEE Conference on*, Vol. 4, 1998, pp. 3635–3640.
- [3] A. Krupadanam, A. Annaswamy, R. Mangoubi, Multivariable adaptive control design with applications to autonomous helicopters, *Journal of Guidance, Control, and Dynamics* 25 (5) (2002) 843–851.
- [4] E. Johnson, S. Kannan, Adaptive trajectory control for autonomous helicopters, *Journal of Guidance, Control, and Dynamics* 28 (3) (2005) 524–538.
- [5] A. Bogdanov, E. Wan, State-dependent riccati equation control for small autonomous helicopters, *Journal of Guidance, Control, and Dynamics* 30 (1) (2007) 47–60.
- [6] K. Peng, G. Cai, B. M. Chen, M. Dong, K. Y. Lum, T. H. Lee, Design and implementation of an autonomous flight control law for a uav helicopter, *Automatica* 45 (10) (2009) 2333 – 2338.
- [7] A. Ollero, L. Merino, Control and perception techniques for aerial robotics, *Annual Reviews in Control* 28 (2) (2004) 167 – 178.
- [8] C. Castillo, W. Moreno, K. Valavanis, Unmanned helicopter waypoint trajectory tracking using model predictive control, in: *Control Automation, 2007. MED '07. Mediterranean Conference on*, 2007, pp. 1 –8. doi:10.1109/MED.2007.4433726.
- [9] T. Keviczky, G. Balas, Software-Enabled Receding Horizon Control for Autonomous Unmanned Aerial Vehicle Guidance, *Journal of Guidance, Control, and Dynamics* 29 (3) (2006) 680–694.
- [10] J. Hedrick, Y. Kang, Linear tracking for a fixed-wing uav using nonlinear model predictive control, *Control Systems Technology, IEEE Transactions on* 17 (5) (2009) 1202–1210.
- [11] C. Leung, S. Huang, N. Kwok, G. Dissanayake, Planning under uncertainty using model predictive

- control for information gathering, *Robotics and Autonomous Systems* 54 (11) (2006) 898 – 910, planning Under Uncertainty in Robotics. doi:DOI: 10.1016/j.robot.2006.05.008.
- [12] M. Saffarian, F. Fahimi, Non-iterative nonlinear model predictive approach applied to the control of helicopters' group formation, *Robotics and Autonomous Systems* 57 (6-7) (2009) 749 – 757. doi:DOI: 10.1016/j.robot.2008.10.021.
 - [13] H. Kim, D. Shim, S. Sastry, Nonlinear model predictive tracking control for rotorcraft-based unmanned aerial vehicles, in: *American Control Conference, 2002. Proceedings of the 2002*, Vol. 5, 2002, pp. 3576–3581.
 - [14] D. Shim, H. Kim, S. Sastry, Decentralized nonlinear model predictive control of multiple flying robots, in: *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, Vol. 4, 2003, pp. 3621–3626.
 - [15] L. Magni, R. Scattolini, Model predictive control of continuous-time nonlinear systems with piecewise constant control, *Automatic Control, IEEE Transactions on* 49 (6) (2004) 900–906.
 - [16] D. Gu, H. Hu, Receding horizon tracking control of wheeled mobile robots, *Control Systems Technology, IEEE Transactions on* 14 (4) (2006) 743–749. doi:10.1109/TCST.2006.872512.
 - [17] B. Mettler, M. B. Tischler, T. Kanade, System identification modeling of a small-scale unmanned rotorcraft for flight control design, *Journal of the American Helicopter Society* 47 (1) (2002) 50–63.
 - [18] V. Gavrillets, B. Mettler, E. Feron, Dynamic model for a miniature aerobatic helicopter, in: *AIAA Guidance Navigation and Control Conference*, Montreal, Canada, 2001.
 - [19] D. Q. Mayne, J. B. Rawlings, C. V. Rao, P. O. M. Scokaert, Constrained model predictive control: Stability and optimality, *Automatica* 36 (6) (2000) 789 – 814.
 - [20] E. Gyurkovics, A. M. Elaiw, Stabilization of sampled-data nonlinear systems by receding horizon control via discrete-time approximations, *Automatica* 40 (12) (2004) 2017 – 2028.
 - [21] L. Grman, D. RosinovÁ, V. Veselý, A. KovÁ, Robust stability conditions for polytopic systems, *International Journal of Systems Science* 36 (15) (2005) 961–973.
 - [22] J. C. Geromel, P. Colaneri, Robust stability of time varying polytopic systems, *Systems & Control Letters* 55 (1) (2006) 81 – 85. doi:DOI: 10.1016/j.sysconle.2004.11.016.
 - [23] H. Chen, F. Allgwer, A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability, *Automatica* 34 (10) (1998) 1205 – 1217.
 - [24] W. Chen, J. O'Reilly, D. Ballance, On the terminal region of model predictive control for non-linear systems with input/state constraints, *International Journal of Adaptive Control and Signal Processing* 17 (3) (2003) 195–207.