Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

# Real-time Model Predictive Control for Quadrotors

**Moses Bangura** [*] **Robert Mahony** [*]

[*] *Australian National University, Acton, ACT 0200*
*(e-mail: {Moses.Bangura, Robert.Mahony}@anu.edu.au).*

**Abstract:** This paper presents a solution to on-board trajectory tracking control of quadrotors. The proposed approach combines the standard hierarchical control paradigm that separates the control into low-level motor control, mid-level attitude dynamics control, and a high-level trajectory tracking with a model predictive control strategy. We use dynamic reduction of the attitude dynamics and dynamic extension of the thrust control along with feedback linearisation to obtain a linear system of relative degree three that models force controlled position and trajectory tracking for the quadrotor. Model predictive control is then used on the feedback equivalent system and its control outputs are transformed back into the inputs for the original system. The proposed structure leads to a low complexity model predictive control algorithm that is implemented in real-time on an embedded hardware. Experimental results on different position and trajectory tracking control are presented to illustrate the application of the derived linear system and controllers.

*Keywords:* Aerial Robotics, Nonlinear Control, Model Predictive Control.

## 1. INTRODUCTION

Quadrotors are aerial vehicles with a four motor-rotor assembly for generating lift and controllability. Their light weight, ease of design and simple dynamics have increased their use in aerial robotics research (Mahony et al. (2012)).

Model predictive control (MPC) refers to a set of controllers that use a model to compute inputs from the current time to a future time in order to optimise the behaviour of a model along the input trajectory (Liu et al. (2011)). The predictive nature of the control design makes it ideal for high performance trajectory tracking. A key advantage of MPC is that it offers the ability to design controllers with constraints while solving an optimal control problem along the given trajectory. The most significant issue in using MPC is associated with the large computational costs that is incurred in a general nonlinear MPC problem. This difficulty has tended to limit the real-time application of MPC control for nonlinear systems with non-trivial constraints to the process/chemical industry where the time constants are slow enough that the underlying optimization problem can be solved in real-time.

To address the issues associated with real-time MPC implementation, researchers have proposed the use of a low computational Newton-type method and explicit MPC (Diehl et al. (2005); Bemporad et al. (2002)). The Newton-type method provides approximate solutions to the optimisation problem by using approximate linearisations along optimal trajectories of the nonlinear system without any guarantee of optimality (Diehl et al. (2005)). Explicit MPC on the other hand is based on computing control laws as piecewise affine functions of state offline and then using some search algorithms (usually a KD-tree search) in the online implementation (Bemporad et al. (2002); Zeilinger et al. (2007); Liu et al. (2011)). Even with pre-computed control, the computational load associated with this approach is significant. Alternatively, MPC for a linearised system with no constraints can be solved as an unconstrained Linear Quadratic Regulator (LQR) problem and becomes computationally viable. The translational dynamics of a quadrotor are naturally a linear system and by assuming that the attitude dynamics are fully controlled, linear MPC can be applied directly (Alexis et al. (2011); Mueller and D'Andrea (2013)). The performance of such a system is inherently limited by the bandwidth of the orientation dynamics and the linearisation of the system changes with different flight conditions and the approach must be combined with a gain scheduling algorithm.

Full nonlinear models of quadrotor aerodynamics have been developed over the last fifteen years (Mahony et al. (2012); Bangura and Mahony (2012); Huang et al. (2009); Pounds and Mahony (2004); Hamel et al. (2002)). Recent results demonstrate the fidelity of these models under a wide range of operating conditions (Omari et al. (2013)). Exact feedback linearisation, a problem that is closely related to the question of finding differentially flat outputs for quadrotors is fundamental in generating the minimum jerk trajectory references that provide the feedforward components of high-performance control in most of the more impressive demonstrations of quadrotor acrobatics (Mellinger et al. (2012); Lupashin and D'Andrea (2011)).

In most existing work, the control problem is split into a feedforward control, usually based on a linearisation or near linearisation of the system dynamics, with high gain PID feedback to regulate the desired trajectory tracking. There has been some work on using the linearised model directly for feedback control design. In November 2013, (Achtelik et al. (2013)) applied the concept of dynamic inversion to the nonlinear system and obtained rotation rates outputs from a position controller. The limitation of their approach however is that the system parameters need to be accurately known and no details on other flight states such as hover and height control were provided.

In this paper, we consider the question of implementing Model predictive control (MPC) trajectory tracking control scheme in real-time on an embedded computer for a quadrotor vehicle. Although the computing resources available on embedded platforms is improving extremely quickly, the underlying computational costs of full nonlinear MPC control is prohibitive and a key contribution of this paper is to design a feedback equivalent linear system to which the MPC framework can be applied. However, feedback linearisation of the full system dynamics is highly complex (and potentially not robust) and leads to significant computational costs associated with applying the feedback transformation. To avoid this, we will use the natural high gain nature of the attitude dynamics to avoid considering the full system dynamics in the MPC stage of the design. The full control design consists of four stages: Firstly, we apply a high gain control to the attitude dynamics and assume that under this control, the vehicle angular velocity tracks a desired reference angular velocity accurately, a process known as dynamic reduction. In practice, there is only sufficient gain available to ensure satisfactory stabilisation of an *a-priori* known trajectory and it is necessary to add a feedforward term to the torque control to anticipate the desired motion. This feedforward term is derived from the MPC design loop in the final stage of the control as discussed below. At the same time, applying dynamic reduction of the attitude dynamics, we dynamically extend the thrust command by adding an integrator. This leads to a system of relative degree three which incorporates the translational kinematics and dynamics along with attitude kinematics in the nonlinear system. The second stage of the control design applies feedback linearisation to obtain an equivalent linear system. The state transformation of the feedback linearisation is associated with the natural thrust vectoring dynamics of the vehicle and the resulting virtual inputs are directly mapped to the thrust integrator input and the angular velocity reference for the high gain loop. The third stage of the control design applies unconstrained MPC to the resulting linear time-invariant system. The MPC is undertaken on a receding horizon and is easily run on any autopilot such as the PX4 hardware (PX4 Team (2013)). The resulting control input is associated with an optimal forward trajectory of the system model and this predicted trajectory is then used to compute the optimal predicted angular velocity trajectory using the full nonlinear model of the system. The final stage of the control is to use the angular velocity and acceleration to generate the feedforward torque used in the dynamic reduction of angular velocity and map the MPC control outputs into the reference angular velocities and integrator thrust input to the dynamic reduction. Unlike previous solutions, our solution solves for a general position and/or trajectory tracking problem without separating and linearising the different flight states such as hovering, translational and axial. Experimental results are presented that demonstrate the effectiveness of the concept.

The remainder of the paper is organised as follows: the nonlinear model for quadrotor dynamics is presented in Section 2; the dynamic reduction and extension along with the feedback linearisation is covered in Section 3. Section 4 outlines the application of unconstrained MPC to the new system, and finally experimental results are shown in Section 5.

## 2. NONLINEAR QUADROTOR MODEL

In this section, we present the nonlinear dynamic model for quadrotors (see Figure 1). The model accounts for the linear drag term and gyroscopic torques that result from rotating rotor blades. From observing that the dominant drag effects are linear, we derive a lumped parameter model of the drag force which is independent of the attitude of the vehicle. Consequently, it will be incorporated directly into the linearised model and not compensated for in the feedback linearisation presented in Section 3.

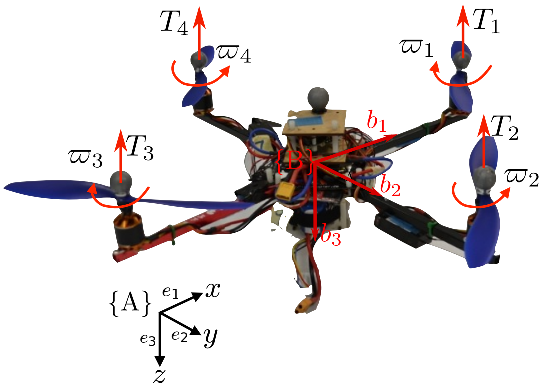

Fig. 1. Schematic of a quadrotor with the different frames of reference.

Let $\zeta = (x,y,z)^\top \in \{A\}$ denote the relative position of the body frame $\{B\}$ to the inertial frame $\{A\}$ and $v = (v_x, v_y, v_z)^\top \in \{A\}$ denote the velocity of $\{B\}$ with respect to $\{A\}$ expressed in $\{A\}$. The angular velocity of $\{B\}$ with respect to $\{A\}$ $\Omega = (\Omega_x, \Omega_y, \Omega_z)^\top \in \{B\}$ represents the rotation rates of the airframe measured in $\{B\}$. Let the orientation of $\{B\}$ with respect to $\{A\}$ be $R = {}^A R_B \in \mathrm{SO}(3)$, then the kinematic equations are given by (Hamel et al. (2002))

$$\dot{\zeta} = v, \tag{1a}$$

$$\dot{R} = R\Omega_\times . \tag{1b}$$

Let $T$ be the combined thrust or heave force generated by the rotors, $D_T$ denote drag which we will discuss in more detail later, and $\tau$ denote the torque generated by the rotors. The dynamics of the quadrotor are given by

$$m\dot{v} = mge_3 - TRe_3 + RD_T, \tag{2a}$$

$$I\dot{\Omega} = -\Omega_\times I\Omega + G_a(\Omega) + \tau, \tag{2b}$$

where $\Omega_\times \in \mathbb{R}^{3\times 3}$ is the skew symmetric matrix derived from $\Omega \in \mathbb{R}^3$ such that $\Omega_\times w = \Omega \times w$ for all $w \in \mathbb{R}^3$. The gyroscopic effect $G_a(\Omega)$ in Equation 2b of the rotors is given by

$$G_a(\Omega) := -\sum_{i=1}^{4} (-1)^i \varpi_i I_r (\Omega \times e_3), \tag{3}$$

where $\varpi_i$ is the angular velocity of the rotors, $I_r$ is the moment of inertia of the rotor blades and the factor $(-1)^i$ accounts for direction of rotation of a given rotor. The total thrust magnitude, $T$ on the vehicle is the sum of the individual thrusts $T_i$ of the rotors

$$T = \sum_{i=1}^{4} T_i.$$

The model used for drag is based on the development provided in (Bangura and Mahony (2012)). The dominant drag effect for a translating rotor vehicle is associated with a combination of rotor blade flapping and induced drag. Both terms depend linearly on the velocity of the vehicle. An effective lumped parameter model is defined by

$$D_T = \sum_{i=1}^{4} D_{r_i} = -TK_r R^\top v \in \{B\},$$

where

$$K_r = \begin{pmatrix} \bar{c} & -\bar{b} & 0 \\ \bar{b} & \bar{c} & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{4}$$

$0 \le \bar{b} << \bar{c}$ are the linear drag coefficients determined from flight tests. In practice, the coupling term $\bar{b}$ can be ignored except in the case of heavy vehicles with very flexible rotors (Omari et al. (2013)). In this case, one may write

$$TRK_rR^\top v = T\bar{c}v - T\bar{c}Re_3V_z,$$

where

$$V_z = e_3^\top R^\top v. \tag{5}$$

If

$$\bar{T} = T + T\bar{c}V_z, \tag{6}$$

using this state transformation leads to the nonlinear system model

$$\dot{\zeta} = v, \tag{7a}$$

$$m\dot{v} = mge_3 - \bar{T}Re_3 - T\bar{c}v, \tag{7b}$$

$$\dot{R} = R\Omega_\times, \tag{7c}$$

$$I\dot{\Omega} = -\Omega_\times I\Omega + G_a(\Omega) + \tau. \tag{7d}$$

Consider Table 1 which shows the sampling frequencies and response times of the different subsystems of a quadrotor. The response time of the motor-rotor subsystem is an order of magnitude faster than the response time of the attitude subsystem, which in turn is at least twice as fast as the linear translational dynamics. Therefore, there is a natural time scale separation between the motor-rotor dynamics and the attitude response and a lesser separation between the attitude dynamics and the position response. Following this analysis, we propose a hierarchical control structure with the motor-rotor thrust control at the low-level, attitude dynamics control at the mid-level and position control at the high-level. The proposed control architecture for the full system is shown in Figure 2. In the sequel, we will show the development of these controllers.

Table 1. Sampling frequency and response time for the different sublevels of a quadrotor.

| System | Frequency and Response Time |
|---|---|
| Motor/ESC | 1000Hz, 0.05s |
| Attitude | 200Hz, 0.5s |
| Position | 50Hz, ≥1s |

The required thrust for the individual motors is determined as follows (Mahony et al. (2012); Hamel et al. (2002))

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ l & 0 & -l & 0 \\ 0 & -l & 0 & l \\ \kappa & -\kappa & \kappa & -\kappa \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} = \begin{bmatrix} T \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}, \tag{8}$$

where $l$ is the length of each rotor from the centre of $\{B\}$ and $\kappa$ is the thrust to torque ratio of the rotor blades. In controlling thrust, we use the current static model for thrust expressed as a function of the rotor speed to determine the rotor speed setpoint

$$T_i = C_{T_0}\varpi + C_T\varpi^2, \tag{9}$$

where $\varpi$ is the rotor speed, $C_{T_0}$ and $C_T$ are the thrust constants determined from static thrust tests. Rotor speed is controlled using a high-gain proportional ($K_\varpi$) feedback coupled with a feedforward term based on the motor-rotor open-loop voltage response

$$v_a = v_{ff}(\varpi_d) - K_\varpi(\varpi - \varpi_d), \tag{10}$$

where $\varpi_d$ is the desired rotor speed obtained from Equation 9. $v_{ff}(\varpi_d)$ is the feedforward voltage and is a function of $\varpi_d$ based on linear interpolation of the static thrust test results (Bangura and Mahony (2012)). The setpoint voltage $v_a$ is the input to the electronic speed controller (ESC) implemented in practice using Pulse Width Modulation (PWM) of the bus voltage. The local motor-rotor control is implemented at 1kHz on the dedicated ESC hardware.

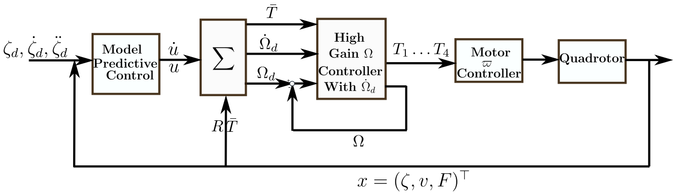

Fig. 2. Proposed Control Scheme. The state $x$ is measured using VICON tracking system, $\Omega$ measurements are provided by rate gyros on the autopilot and $\varpi$ measurements on-board the ESC. Using the $R, v$ and $T$, the $\sum$ block converts the MPC control outputs $u$ and its derivative $\dot{u}$ into a new $T$, and desired $\dot{\Omega}$ and $\Omega$ for the rate controller.

## 3. DECOMPOSING AND SIMPLIFYING THE SYSTEM

Applying model predictive control (MPC) to the full nonlinear dynamics of the quadrotor (Equation 7) is computationally intractable on the embedded hardware typically available on a quadrotor aerial robot. To simplify the system model, we propose to exploit the natural high-gain and passive characteristics of the attitude dynamics to dynamically reduce Equation 7d leaving the remaining dynamic Equations 7a, 7b and 7c to be dealt with using MPC. We will show that the remaining dynamics can be feedback linearised leading to a viable linear MPC (LMPC) problem.

### 3.1 *High Gain Control of* $\Omega$

Let $\Omega_d = \Omega_d(t)$ be a desired reference angular velocity specified at time $t$ by the MPC control along with the nonlinear mapping associated with the feedback linearisation that will be derived in this section. This reference signal is taken as the desired setpoint for the high gain angular velocity control loop. Since $\Omega_d$ is obtained from an MPC algorithm, then it is associated with an optimal control $\Omega_d(\tau)$ on a time interval $[t, t+\mathbf{T})$ into the future. As will be shown in Section 4, the underlying MPC problem is linear and the resulting control is well defined and differentiable in forward time. The mapping from MPC control to $\Omega_d$ is differentiable and its Jacobian is full rank along feasible trajectories of the quadrotor, and it follows that $\dot{\Omega}_d(t)$ is well defined for all time and can be computed. Recalling Equation 7d, we define a reference feedforward torque $\tau_d$ by

$$\tau_d := I\dot{\Omega}_d + (\Omega_d)_\times I\Omega - G_a(\Omega_d), \tag{11}$$

where $\dot{\Omega}_d$ is a feedforward term obtained from the trajectory. Let $\tilde{\Omega} = \Omega - \Omega_d$ and let the feedback torque be

$$\tau := \tau_d - K_\Omega\tilde{\Omega}, \tag{12}$$

where $K_\Omega \in \mathbb{R}^{3\times 3}$ is a positive definite matrix providing the high-gain regulation.

*Lemma 1.* Consider the system Equation 7d with reference signals $\Omega_d$ and $\dot{\Omega}_d$ with control $\tau$ given by Equation 12 along with Equation 11, then $\tilde{\Omega} \to 0$ is globally exponentially stable to zero.

**Proof.** Consider the Lyapunov function

$$V = \frac{1}{2}\tilde{\Omega}^\top I\tilde{\Omega}$$

and note that $V$ is quadratic in the error $\tilde{\Omega}$. The derivative of $V$ is

$$\dot{V} = \tilde{\Omega}^\top I \dot{\tilde{\Omega}}.$$

The error dynamics is given by

$$I\dot{\tilde{\Omega}} = I\dot{\Omega} - I\dot{\Omega}_d. \tag{13}$$

Substituting the system dynamics (Equation 7d) along with the definition for the input torque $\tau$, thus

$$\dot{V} = \tilde{\Omega}^\top \left(-\Omega_\times I\Omega + G_a(\Omega) + \tau_d - K_\Omega\tilde{\Omega} - I\dot{\Omega}_d\right).$$

Substituting for $\tau_d$,

$$\dot{V} = \tilde{\Omega}^\top \left(-\Omega_\times I\Omega + G_a(\Omega) - K_\Omega\tilde{\Omega} + (\Omega_d)_\times I\Omega - G_a(\Omega_d)\right).$$

Hence,

$$\dot{V} = \tilde{\Omega}^\top \left(-\tilde{\Omega}_\times I\Omega + G_a(\tilde{\Omega}) - K_\Omega\tilde{\Omega}\right).$$

Since $\tilde{\Omega}^\top\tilde{\Omega}_\times = -\tilde{\Omega}\times\tilde{\Omega} = 0$, then

$$\dot{V} = -\tilde{\Omega}^\top K_\Omega \tilde{\Omega}.$$

From Theorem 3.8 and Corollary 3.3 of (Khalil (1996)), it follows that the error $\tilde{\Omega}$ is globally asymptotically stable to $\tilde{\Omega} = 0$. Moreover, since the Lyapunov function is quadratic and its derivative is overbound by a negative quadratic i.e. for some $\gamma > 0$, $\dot{V} \leq -\gamma V \ \forall\tilde{\Omega}$, implies that $\tilde{\Omega}$ is globally exponentially stable to $\tilde{\Omega} = 0$ by the control law of Equation 12.

This controller forms the inner loop of the architecture shown in Figure 2.

### *3.2 Feedback linearisation*

This subsection shows how feedback linearisation can be applied to the nonlinear system to obtain a linear system for MPC in Section 4.

Following on from Lemma 1, we will use the assumption $\Omega = \Omega_d$ in the following derivation and treat $\Omega$ as an input. Consequently only Equations 7a, 7b and 7c need to be considered. However, the new input $\Omega_d$ and the thrust input $T$ do not have the same relative degree with respect to the position of the vehicle. To address this, we dynamically extend the thrust $T$ by assigning a new input

$$\dot{T} := W. \tag{14}$$

The variable $T$ becomes an internal variable in the vehicle control algorithm.

Following these simplifications, we consider the following system

$$\dot{\zeta} = v, \tag{15a}$$
$$m\dot{v} = mge_3 - \bar{T}Re_3 - T\bar{c}v, \tag{15b}$$
$$\dot{R} = R\Omega_\times, \tag{15c}$$
$$\dot{T} = W \tag{15d}$$

with inputs $(\Omega, W)$. This is similar to work presented in (Achtelik et al. (2013)). The difference is that they considered the roll and pitch attitude angles as non-controlled states. It will be shown that including attitude enables us to do hovering flights and prevents the vehicle from entering an infinite number of flips or rotations when the distance to the goal is very large.

Note that Equation 15a is already linear. Consider Equation 15b which can be rewritten as

$$m\dot{v} = mge_3 - \bar{T}Re_3 - (T\bar{c} - mg\bar{c})v - mg\bar{c}v. \tag{16}$$

Let the exogenous force applied to the vehicle be defined by

$$F := mge_3 - \bar{T}Re_3 - (T\bar{c} - mg\bar{c})v. \tag{17}$$

Hence Equation 16 can be rewritten as

$$m\dot{v} = F - mg\bar{c}v. \tag{18}$$

These dynamics are linear and can be seen as a force-mass system with velocity damping. To continue the feedback linearisation process, we propose the following dynamics for $F$

$$\dot{F} := -\beta F + u, \tag{19}$$

where $\beta$ can be seen as a low-pass time constant on the evolution of $F$. $u$ is the "new" control input signal obtained from the solution to the high-level model predictive controller (MPC) and is explained in Section 4. The low-pass $\beta$ is introduced to model the limitations of the attitude response system in a natural manner so that the MPC control will be well conditioned.

The choice of dynamics, Equation 19, along with the input $u$ that will be provided by the MPC control define the physical control inputs $(\Omega, W)$ to the dynamics (Equation 15). Equation 17 is differentiated explicitly and set equal to the definition of $\dot{F}$ in Equation19. Thus

$$-\bar{T}R\Omega_\times e_3 - \dot{\bar{T}}Re_3 - \dot{T}\bar{c}v - (T\bar{c} - mg\bar{c})\dot{v} = -\beta F + u. \tag{20}$$

From Equation 6,

$$\dot{\bar{T}} = \dot{T} + \dot{T}\bar{c}V_z + T\bar{c}\dot{V}_z. \tag{21}$$

Substituting for $\dot{v}, \bar{T}$ and $\dot{\bar{T}}$ in Equation 20 yields

$$\begin{aligned} -\left[\dot{T}\left(1 + \bar{c}V_z\right) + T\bar{c}\dot{V}_z\right]Re_3 - \bar{T}R\Omega_\times e_3 - \dot{T}\bar{c}v - \\ \frac{(T\bar{c} - mg\bar{c})}{m}(F - mg\bar{c}v) = -\beta F + u. \end{aligned} \tag{22}$$

Expanding and collecting like terms, one obtains

$$\begin{aligned} -\dot{T}\left(Re_3 + \bar{c}V_zRe_3 + \bar{c}v\right) - T\bar{c}\dot{V}_zRe_3 - \left(T + T\bar{c}V_z\right)R\Omega_\times e_3 \\ - \frac{(T\bar{c} - mg\bar{c})}{m}(F - mg\bar{c}v) = -\beta F + u. \end{aligned} \tag{23}$$

Noting that $\dot{R}^\top = -\Omega_\times R^\top$, then

$$\dot{V}_z = -e_3^\top\Omega_\times R^\top v + e_3^\top R^\top\dot{v},$$

and therefore,

$$T\bar{c}\dot{V}_zRe_3 = T\bar{c}Re_3\left(-e_3^\top\Omega_\times R^\top v + e_3^\top R^\top \dot{v}\right).$$

Thus expanding Equation 23

$$\begin{aligned} -\dot{T}\left(Re_3 + \bar{c}V_zRe_3 + \bar{c}v\right) - T\bar{c}Re_3\left(-e_3^\top\Omega_\times R^\top v + e_3^\top R^\top\dot{v}\right) \\ - \left(T + T\bar{c}V_z\right)R\Omega_\times e_3 - \frac{(T\bar{c} - mg\bar{c})}{m}(F - mg\bar{c}v) = -\beta F + u. \end{aligned} \tag{24}$$

Also

$$e_3^\top\Omega_\times R^\top v = -v^\top R\Omega_\times e_3.$$

Hence,

$$\begin{aligned} -\dot{T}\left(Re_3 + \bar{c}V_zRe_3 + \bar{c}v\right) - T\bar{c}Re_3\left(v^\top R\Omega_\times e_3 + e_3^\top R^\top\dot{v}\right) \\ - \left(T + T\bar{c}V_z\right)R\Omega_\times e_3 - \frac{(T\bar{c} - mg\bar{c})}{m}(F - mg\bar{c}v) \\ = -\beta F + u. \end{aligned} \tag{25}$$

Therefore Equation 25 becomes

$$
\begin{aligned}
&-\dot{T}\left((1+\bar{c}e_3^\top R^\top v)Re_3+\bar{c}v\right)\\
&-T\left(\bar{c}v^\top Re_3+(1+\bar{c}e_3^\top R^\top v)\right)R\Omega_\times e_3\\
&-\frac{(T\bar{c}-mg\bar{c})}{m}(F-mg\bar{c}v)\\
&-T\bar{c}\left[e_3^\top R^\top\left(\frac{F-mg\bar{c}v}{m}\right)\right]Re_3=-\beta F+u.
\end{aligned}
\tag{26}
$$

From Equation 5, one has

$$v=V_zRe_3, v^\top=V_ze_3^\top R^\top,$$

and this implies that

$$
\begin{aligned}
&-W\left((1+\bar{c}e_3^\top R^\top v)Re_3+\bar{c}v\right)\\
&-T\left(\bar{c}v^\top Re_3+(1+\bar{c}e_3^\top R^\top v)\right)R\Omega_\times e_3\\
&-\frac{(T\bar{c}-mg\bar{c})}{m}(F-mg\bar{c}v)\quad -T\bar{c}\left[e_3^\top R^\top\left(\frac{F-mg\bar{c}v}{m}\right)\right]Re_3\\
&=-\beta F+u.
\end{aligned}
\tag{27}
$$

We define the following variables ($\gamma,\lambda\in\mathbb{R}^3,\eta\in\mathbb{R}$)

$$
\begin{aligned}
\gamma &:= (1+\bar{c}e_3^\top R^\top v)Re_3+\bar{c}v,\\
\eta &:= -\bar{c}v^\top Re_3+(1+\bar{c}e_3^\top R^\top v),\\
\lambda &:= -\frac{(T\bar{c}-mg\bar{c})}{m}(F-mg\bar{c}v)-T\bar{c}\left[e_3^\top R^\top\left(\frac{F-mg\bar{c}v}{m}\right)\right]Re_3,
\end{aligned}
$$

then Equation 27 becomes

$$-W\gamma-T\eta R\Omega_\times e_3=-\beta F+u-\lambda. \tag{28}$$

It should be noted that Equation 28 has solutions for all $v$ and $R$ except when

$$
\begin{aligned}
T&=0,\\
\bar{c}v&=-(1+\bar{c}e_3^\top R^\top v)Re_3,\\
\bar{c}v^\top Re_3&=-(1+\bar{c}e_3^\top R^\top v).
\end{aligned}
$$

which is associated with the situation where the quadrotor is in free fall or executing extreme acrobatic manoeuvres that can never be approached in normal flight conditions. However, acrobatic manouvres can be chosen to avoid such a configuration as it corresponds to a fundamental loss of controllability of the vehicle.

With the state of the vehicle known, Equation 28 is solved to obtain $W,\Omega_x$ and $\Omega_y$. The final degree of freedom $\Omega_z$ of the rotation rate is set in order to match the evolution of the free degree of freedom of the rotation around the $e_3$ axis. These are used as inputs ($T=T+\int_0^\tau W\mathrm{d}\tau$ and $\Omega_d$) to the mid-level controller proposed in Section 3.1.

Combining Equation 17 with Equations 1a, 18 and 19, the new system is shown in Equation 29

$$\dot{\zeta}=v, \tag{29a}$$

$$m\dot{v}=F-mg\bar{c}v, \tag{29b}$$

$$\dot{F}=-\beta F+u, \tag{29c}$$

$$\dot{T}=W. \tag{29d}$$

The resulting system is linear and will be used in Section 4 for the MPC design.

It remains to show how to generate the $\dot{\Omega}_d$ signal that is used in the feedforward term for the angular velocity control. We assume that from the MPC control in Section 4, we have both the control $u$ and its derivative $\dot{u}$. This is reasonable since the MPC problem is linear and the resulting control input is forward differentiable. To do this, we must differentiate Equation 28 to see the dependence on $\dot{\Omega}$. It is straightforward to see that one obtains

$$-\dot{W}\gamma-T\eta R\dot{\Omega}_\times e_3=f(\gamma,\dot{\gamma},\lambda,\dot{\lambda},\eta,\dot{\eta},F,\dot{F},W,T,R,\dot{R},u,\dot{u}). \tag{30}$$

Space constraints prevent us providing a full expression for the function $f$ in Equation 30. However, it is stratightforward to verify that it depends on known quantitites. The resulting values for $\dot{W}$, $\dot{\Omega}_x$ and $\dot{\Omega}_y$ are then obtained by solving Equation 30 and only the values for $\dot{\Omega}_x$ and $\dot{\Omega}_y$ are used. The value for $\dot{\Omega}_z$ is assigned as desired. Typically either by setting $\dot{\Omega}_z=0$ or by using a derivative of the actual trajectory from previous time plus some desired goal for future time.

## 4. LINEAR MODEL PREDICTIVE CONTROL

To execute the high-level trajectory tracking with optimal control and online optimisation suggests the use of model predictive control (MPC). This has the added advantage of providing a set of lookahead points within a finite time horizon. Other advantages of MPC are the ability to handle inputs/outputs constraints, use of the plant model, robustness to modelling errors and easy adaptation to changes in system dynamics.

Rearranging Equations 29a to 29c into the standard state space format ($\dot{x}=Ax+Bu$), Equation 31 is obtained.

$$
\begin{bmatrix}\dot{\zeta}\\ \dot{v}\\ \dot{F}\end{bmatrix}=\begin{bmatrix}0 & 1 & 0\\ 0 & -g\bar{c} & \frac{1}{m}\\ 0 & 0 & -\beta\end{bmatrix}\begin{bmatrix}\zeta\\ v\\ F\end{bmatrix}+\begin{bmatrix}0\\ 0\\ 1\end{bmatrix}u. \tag{31}
$$

If one has the time derivative of the trajectory ($\dot{\zeta},\dot{v}$), then $F_d$ in the desired state ($\zeta_d,v_d,F_d$) can be determined using Equation 18.

In Section 5, it will be shown that the states $\left(x=(\zeta,v,F)^\top\right)$ can be measured or estimated and therefore future states of the vehicle are predictable.

For quadrotors, there exists a maximum thrust $T_{i_{\max}}$ produced by each rotor, thus there is a maximum velocity $v_{\max}$ and $\Omega_{\max}$. From $v_{\max}$, $F_{\max}$ can be determined. The constraint on the rate of the thrust i.e. $W_{\max}$ implies that $\dot{F}_{\max}$ exists and is a function of the vehicle attitude ($R$) linked by Equation 17. Hence through Equation 19, the instantaneous constraints on the control input $u$ can be obtained. Therefore, the state and input constraints are dependent on $R,T$ and $v$ of the vehicle. To the best knowledge of the authors, there is no published work on the solution to the state and input constraints MPC problem that are state dependent. However, a solution to the linearisation of nonlinear MPC with state dependent input constraints has been proposed in (Simon et al. (2013); Deng et al. (2009)). This paper uses unconstrained inputs and states in order to have real-time solution. In addition, these input constraints are dependent on the maximum voltage of the battery through Equation 10 that is two levels down in the hierarchy. This enables the complete removal of the constraints on the MPC control input $u$ and state $x$ thereby transforming the problem into an unconstrained MPC problem.

*Remark 1.* If the time between two samples is $\Delta t$, then the discretised state and input matrices can be approximated using Starred Transform and the result is $A_d=\mathrm{I}_{n\times n}+A\Delta t\in\mathbb{R}^{n\times n}$, $B_d=B\Delta t\in\mathbb{R}^{n\times m}$.

In the sequel, $A \in \mathbb{R}^{n\times n}$ refers to $A_d \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ to $B_d \in \mathbb{R}^{n\times m}$. $n = 9$ is the total number of states and $m = 3$ is the total number of control inputs.

The model predictive control algorithm is outlined in the following texts

(1) At time $t = k$, measure $x$ and $R$ and estimate $v$.
(2) With current thrust input $T$ and attitude $R$, determine $F$, hence the reduced states $\left(x(k) = (\zeta, v, F)^\top \in \mathbb{X}\right)$.
(3) Use this as the initial value $x(k|k) \doteq x(k)$ to solve the unconstrained optimal control problem (OCP).

$$V^*(x_{k|k}) = \min_{u^*(.)\in\mathbb{U}} J_N(x(k|k),u(.)) \doteq \sum_{k=0}^{N-1} l\left(x_u(k+i|k),u(k)\right),$$

with soultion $u^*(.) \in \mathbb{U}^N(x_0)$ where $N$ is the optimisation horizon.
(4) Using Equation 28, determine $W$ (hence $T$) and $\Omega_d$ and $\dot{\Omega}_d$ from Equation 30 to be regulated by the controller shown in Section 3.1.
(5) Using Equation 8, determine the thrust and use it as a setpoint for the motor controller in Equation 10.
(6) Restart the entire process with $x(k) = x(k|k+1)$.

If the desired state is $x_d$, we propose the following incremental cost

$$l\left(x_u(k,x_0),u(k)\right) = x_e^\top Q x_e + u^\top P u. \tag{32}$$

For which $x_e = x - x_d$, $Q \in \mathbb{R}^{9\times 9}$ and $P \in \mathbb{R}^{3\times 3}$ are positive definite weighting matrices.

To solve the linear model predictive control (LMPC) OCP, consider finding the optimal control $u^*(k)$ at time $t = k$ and rewriting the incremental cost as

$$J(k) = \sum_{i=0}^{N-1}\left[x_e(k+i|k)^\top Q x_e(k+i|k) + u(k+i|k)^\top P u(k+i|k)\right]. \tag{33}$$

With the discretised state equation, for a given set of control inputs, one can generate the set of states $x_e(k)$

$$\begin{aligned}
x_e(k|k) &= x_e(k)\\
x_e(k+1|k) &= Ax_e(k) + Bu(k|k)\\
x_e(k+2|k) &= A^2x_e(k) + ABu(k|k) + Bu(k+1|k)\\
x_e(k+3|k) &= A^3x_e(k) + A^2Bu(k|k) + ABu(k+1|k) +\\
&\quad Bu(k+2|k)\\
&\vdots\\
x_e(k+N-1|k) &= A^{N-1}x_e(k) + A^{N-2}Bu(k|k) + \ldots +\\
&\quad Bu(k+N-2|k).
\end{aligned}$$

Which can be written in compact form as

$$x_e(k+i|k) = A^i x_e(k) + \mathscr{C}_i u(k), i = 0,\ldots,N-1$$

or

$$x_e(k|k) = \mathscr{M}x_e(k) + \mathscr{C}u(k),$$

$$\text{with } \mathscr{M} = \begin{bmatrix} 1\\ A\\ A^2\\ \vdots\\ A^{N-1}\end{bmatrix} \textit{ and } \mathscr{C} = \begin{bmatrix} 0 & 0 & 0\\ B & 0 & \ldots\ 0\\ AB & B & \ldots\ 0\\ \vdots & \vdots & \vdots\\ A^{N-2}B & A^{N-3}B & \ldots\ B\end{bmatrix}.$$

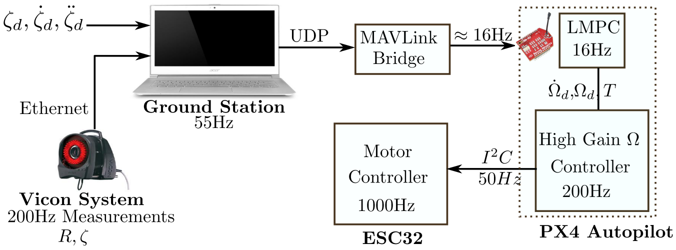
$\zeta_d, \dot{\zeta}_d, \ddot{\zeta}_d$
UDP
MAVLink Bridge
≈ 16Hz
LMPC 16Hz
$\dot{\Omega}_d, \Omega_d$ $T$
Ethernet
Ground Station 55Hz
Motor Controller 1000Hz
ESC32
$I^2C$ 50Hz
High Gain Ω Controller 200Hz
PX4 Autopilot
Vicon System 200Hz Measurements $R, \zeta$

Fig. 3. Layout of the experimental setup showing timescale separation of the various components of the control architecture. Due to thread priority scheduling and data packet losses in the wireless communication link, the position data arrives in the MPC controller at 16Hz which dictates the frequency at which the MPC is run.

If $H = \mathscr{C}^\top \bar{Q}\mathscr{C} + \bar{P}$, $S = \mathscr{C}^\top \bar{Q}\mathscr{M}$, $G = \mathscr{M}^\top \bar{Q}\mathscr{M} + Q$ with $\bar{Q} = \begin{bmatrix} Q & 0 & \ldots & 0\\ 0 & \ddots & & \vdots\\ \vdots & & Q & 0\\ 0 & \ldots & 0 & Q\end{bmatrix}$ and $\bar{P} = \begin{bmatrix} P & & & 0\\ & P & & \\ & & \ddots & \\ 0 & & & P\end{bmatrix}$, then substituting for $x_e(k+i|k)$ in the incremental cost (Equation 33) and collecting like terms yield (Maciejowski (2002))

$$J(k) = u^\top(k)Hu(k) + 2x_e^\top(k)S^\top u(k) + x_e^\top(k)Gx_e(k) \tag{34}$$

For unconstrained optimisation,

$$\nabla_u J(k) = 0 \Rightarrow \nabla_u J(k) = 2Hu(k) + 2Sx_e(k).$$

From which the optimal control input can be calculated as

$$u^*(k) = -H^{-1}Sx_e(k). \tag{35}$$

This can be solved in real-time with the optimal control being the first row of elements of $u^*(k)$. With this $u$, and differentiating to obtain $\dot{u}$, $\dot{F}$ is obtained and subsequently the desired inputs ($T$, $\Omega$ and $\dot{\Omega}$) to the mid-level controller presented in Section 3.1.

## 5. FLIGHT TESTS AND RESULTS

In this section, we present experimental results for the quadrotor performing autonomous takeoff, hover, position control and a “figure 8” trajectory tracking to illustrate applicability of the proposed linear system and controllers.

The architecture for the experimental system setup is shown in Figure 3. The VICON system provides measurements of the position and attitude of the vehicle at 200Hz (Vicon Team (2013)). The Robot Operating System (ROS) is used on the ground station computer to access and transmit the position, attitude and desired trajectory $(\zeta_d, v_d, F_d)^\top$ to the PX4 autopilot encoded using MAVLink protocol through a UDP connection to the quadrotor (Fernandez (2013); Meier (2013)). The MPC then determines the desired $T, \dot{\Omega}$ and $\Omega$ that are sent to the rotation rates controller. The output of this controller is the desired thrusts from each motor-rotor system. These are then sent to the ESCs using I2C protocol.

To control the free degree of freedom, yaw, we set the desired yaw angle and use a proportional integral (PI) controller to regulate it. The output of the controller is the desired yaw rate. After careful tuning, the following MPC values were used $N = 5$, $\beta = 0.1$ and $\bar{c} = 0.01$, $P = \mathbb{I}_{3\times 3}, Q = \mathbb{I}_{9\times 9}$

$$\begin{aligned}
Q_{1,1} &= 150, Q_{1,4} = Q_{4,1} = 200, Q_{1,7} = Q_{7,1} = 1;\\
Q_{2,2} &= 150, Q_{2,5} = Q_{5,2} = 200, Q_{2,8} = Q_{8,2} = 1;\\
Q_{3,3} &= 1300, Q_{3,6} = Q_{6,3} = 500, Q_{3,9} = Q_{9,3} = 1.
\end{aligned}$$

The experimental results for the quadrotor doing takeoff, hover, translational and tracking of a "figure 8" trajectory are shown in Figure 4. The "figure 8" trajectory is defined by

$$x = 0.6\cos\left(\frac{\pi}{10}t\right), y = 0.6\sin\left(2\frac{\pi}{10}t\right), z = -0.4,$$

$$\dot{x} = -0.06\pi\sin\left(\frac{\pi}{10}t\right), \dot{y} = 0.12\pi\cos\left(2\frac{\pi}{10}t\right), \dot{z} = 0,$$

$$\dot{v}_x = -0.6\left(\frac{\pi}{10}\right)^2\cos\left(\frac{\pi}{10}t\right), \dot{v}_y = -2.4\left(\frac{\pi}{10}\right)^2\sin\left(2\frac{\pi}{10}t\right),$$
$$\dot{v}_z = 0$$

where $t$ is the time since the start of the trajectory.

As can be seen, the quadrotor is able to takeoff, hover, transition to a new position and execute the desired "figure 8" trajectory using the derived linear model and control law of Equation 35. It should be noted that more weighting was placed on position relative to velocity hence a better position tracking compared to velocity.

## 6. CONCLUSIONS

In this paper, we have presented a solution to on-board trajectory tracking for quadrotors that is based on the development of a new linear model. The approach uses a hierarchical control architecture with timescale separation of the different dynamic levels of the quadrotor. From the full state nonlinear system, using dynamic reduction of the attitude dynamics and dynamic extension on the total thrust, feedback linearisation was applied that led to the new linear system with reduced states and of relative degree three. MPC was applied to the ensuing linear system for tracking a trajectory at the high-level. The outputs from the unconstrained MPC are the thrust and rotation rates and angular acceleration that are regulated using a Lyapunov based attitude dynamics controller. The control system along with the new linear model have been implemented on a quadrotor and used in real-time to do position and trajectory tracking.

## ACKNOWLEDGEMENTS

This research was funded by Australian Research Council through Discovery Grant DP120100316.

## REFERENCES

Achtelik, M.W., Lynen, S., Chli, M., and Siegwart, R. (2013). Inversion based direct position control and trajectory following for micro aerial vehicles. *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Alexis, K., Papachristos, C., Nikolakopoulos, G., and Tzes, A. (2011). Model predictive quadrotor indoor position control. *IEEE Mediterranean Conf. on Control and Automation*.

Bangura, M. and Mahony, R. (2012). Nonlinear dynamic modeling for high performance control of a quadrotor. In *Australasian Conference on Robotics and Automation*.

Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E. (2002). *The explicit linear quadratic regulator for constrained systems*.

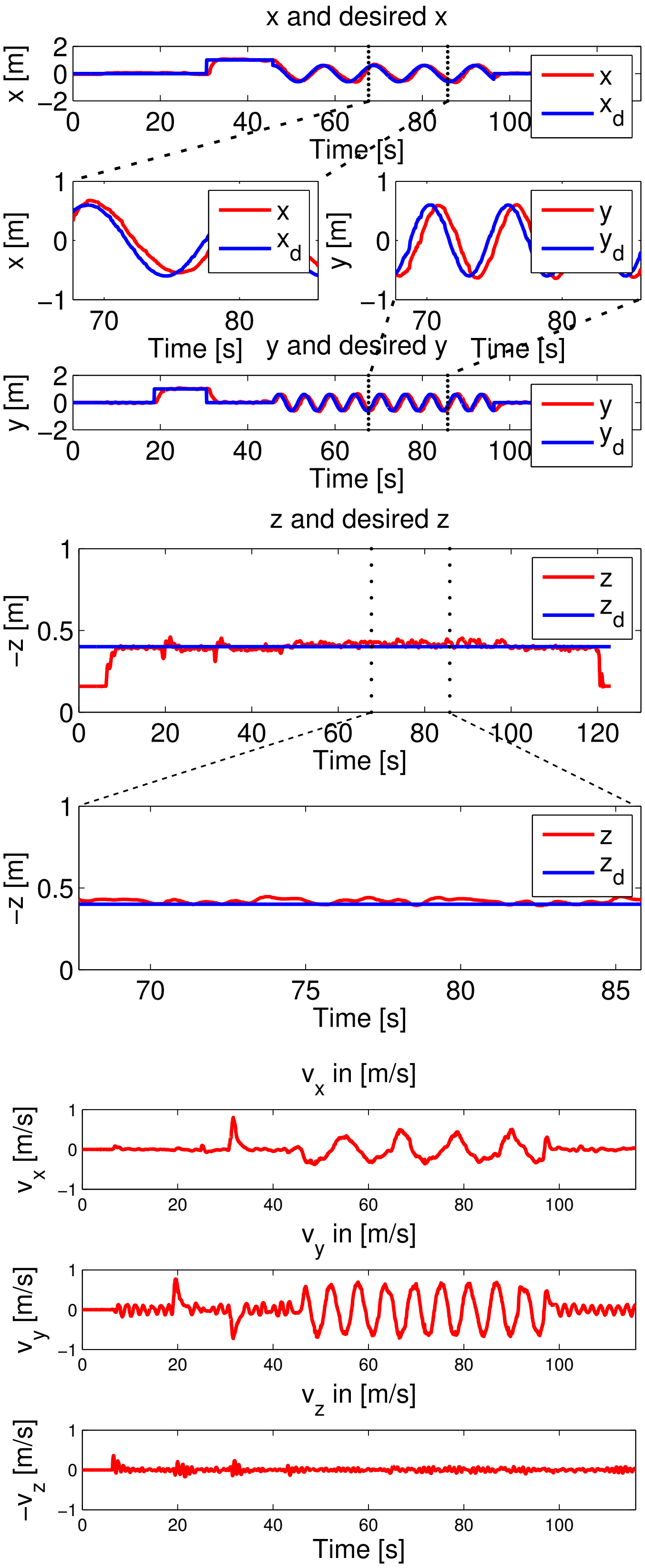

Fig. 4. Results for the quadrotor taking off, translational position control, trajectory tracking and hover recorded onboard the PX4 autopilot.

Deng, J., Becerra, V.M., and Stobart, R. (2009). Input constraints handling in an mpc/feedback liearization scheme. In *Int. Journal of Applied Mathematics*, volume 19, 219–232.

Diehl, M., Bock, H.G., and Schlöder, J.P. (2005). A real-time iteration scheme for nonlinear optimization in optimal feedback control. *Society for Industrial and Applied Mathematics*, 43(5), 1714–1736.

Fernandez, E. (2013). Robot operating system. "`http://wiki.ros.org/`". [Online; accessed 14-October-2013].

Hamel, T., Mahony, R., Lozano, R., and Ostrowski, J. (2002). Dynamic modelling and configuration stabilization for an x4-flyer. *IFAC World Congress*.

Huang, H., Hoffmann, G.M., Waslander, S.L., and Tomlin, C.J. (2009). Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering. In *Robotics and Automation, 2009. ICRA'09. IEEE Int. Conference on*, 3277–3282. IEEE.

Khalil, H.K. (1996). *Nonlinear Systems*. Prentice-Hall, 2nd edition.

Liu, C., Chen, W., and Andrew, J. (2011). Trajectory tracking of small helicopters using explicit nonlinear mpc and dobc. *IFAC World Congress*, 18.

Lupashin, S. and D'Andrea, R. (2011). Adaptive open-loop aerobatic maneuvers for quadrocopters. *IFAC World Congress*.

Maciejowski, J.M. (2002). *Predictive Control: with constraints*. Prentice-Hall.

Mahony, R., Kumar, V., and Corke, P. (2012). Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *Robotics Automation Magazine, IEEE*, 19(3), 20–32. doi: 10.1109/MRA.2012.2206474.

Meier, L. (2013). Mavlink micro air vehicle communication protocol. "`http://qgroundcontrol.org/mavlink/start`". [Online; accessed 14-October-2013].

Mellinger, D., Michael, N., and Kumar, V. (2012). Trajectory generation and control for precise aggressive maneuvers with quadrotors. *The Int. Journal of Robotics Research*, 31(5), 664–674.

Mueller, M.W. and D'Andrea, R. (2013). A model predictive controller for quadrocopter state interception. *IEEE European Control Conference*.

Omari, S., Hua, M.D., Ducard, G., and Hamel, T. (2013). Nonlinear control of vtol uavs incorporating flapping dynamics. In *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, to appear.

Pounds, P. and Mahony, R. (2004). Towards dynamically-favourable quad-rotor aerial robots. In *Australasian Conference on Robotics and Automation*.

PX4 Team (2013). Pixhawk px4 autopilot. "`https://pixhawk.ethz.ch/px4/en/start`". [Online; accessed 14-October-2013].

Simon, D., Lofberg, J., and Glad, T. (2013). Nonlinear model predictive control using feedback linearisation and local inner convex constraints approximations. *IEEE European Control Conference*.

Vicon Team (2013). Vicon motion systems. "`http://www.vicon.com/`". [Online; accessed 14-October-2013].

Zeilinger, M.N., Jones, C.N., and Morari, M. (2007). *Real-Time Model Predictive Control using a combination of Explicit MPC and Online Optimization*. IEEE Conference on Decision and Control.