Hindawi Publishing Corporation
Abstract and Applied Analysis
Volume 2015, Article ID 916864, 11 pages
http://dx.doi.org/10.1155/2015/916864

Research Article

Trajectory Tracking and Stabilization of a Quadrotor Using Model Predictive Control of Laguerre Functions

Mapopa Chipofya,[1] Deok Jin Lee,[2] and Kil To Chong[1]

[1] *Department of Electronic Engineering, Chonbuk National University, Room 421, Engineering Building No. 7, 567 Baekje-daero, Deokjin-gu, Jeonju-si, Jeollabuk-do 561-756, Republic of Korea*
[2] *Department of Mechanical Engineering, Kunsan National University, Republic of Korea*

Correspondence should be addressed to Kil To Chong; kitchong@jbnu.ac.kr

Received 12 September 2014; Accepted 29 October 2014

Academic Editor: Sakthivel Rathinasamy

This paper presents a solution to stability and trajectory tracking of a quadrotor system using a model predictive controller designed using a type of orthonormal functions called Laguerre functions. A linear model of the quadrotor is derived and used. To check the performance of the controller we compare it with a linear quadratic regulator and a more traditional linear state space MPC. Simulations for trajectory tracking and stability are performed in MATLAB and results provided in this paper.

1. Introduction

A quadrotor is a helicopter which has four equally spaced rotors, usually arranged at the corners of a square body. When these rotors spin, they push air downwards and in the process create a thrust force that keeps the quadrotor aloft. Unlike the conventional helicopter (with two rotors) which requires a swashplate mechanism in order to have more degrees of freedom, such mechanism is not needed in quadrotor systems since the two additional rotors provide the same level of control as that with conventional helicopters fixed with swashplate mechanism.

The task to control a quadrotor is a fundamentally difficult and interesting problem. With six degrees of freedom (three translational and three rotational) and only four independent inputs (rotor speeds), quadrotors are severely underactuated. In order to achieve six degrees of freedom, rotational and translational motion are coupled. The resulting dynamics are highly nonlinear, especially after accounting for the complicated aerodynamic effects. Finally, unlike ground vehicles, quadrotors have very little friction to prevent their motion, so they must provide their own damping in order to stop moving and remain stable. Put together, these factors create a very challenging control problem.

2. Literature Review

Amongst the many control techniques used in quadrotor control PID, LQR and recently MPC have been widely used.

While PID is the most popular choice for controlling several types of processes, it turns out that tuning becomes a big challenge especially for MIMO systems like the quadrotor. Several techniques have been used to divide the quadrotor control problem to several SISO systems. While this has worked in some cases it has to be pointed out that this takes away the most natural way of controlling the system and designing many SISO controllers may make maintenance more difficult.

LQR is a relatively modern control method that is very powerful yet limited to applications where linear system models are available. To use this method to control nonlinear systems a linear model has to be obtained from the corresponding nonlinear model. To evaluate the performance of the LQR controller we need to compare it with results obtained from other control techniques. This means designing the same system using another control technique which could be cumbersome. Because of this, LQR is only very popular to use in inherently linear systems or where results of control obtained using other control techniques are available for comparison.

On the other hand, MPC can handle both linear and nonlinear systems [1]. Comparisons between the nonlinear system and its corresponding linear model can easily be made with very little modification. Compared to PID, tuning MPC is easier even for complex MIMO systems.

However, in trying to track the reference trajectory in an optimal way and at the same time obey the constraints imposed MPC makes much more calculations than either PID or LQR. This computation burden makes MPC less suitable for "fast" processes. It requires very powerful processors to be used in hardware implementation. Not surprisingly a lot of effort is being put in to reduce the computations so that MPC can be faster and easily be implemented even on low cost processors.

In [2] Wang proposed a method of designing MPC using orthonormal functions. This method makes less computations than the traditional MPC. With reduced number of computations this technique can be used where rapid system dynamics are required. We will use this method in our task to control a quadrotor using MPC.

Other MPC applications to quadrotor can be found in [3, 4]. In [4] the control structure consists of a controller based on MPC to track the reference trajectory and a second one based on a nonlinear H_∞ control technique to stabilize the rotational movements. Similarly in [3] an MPC controller is designed to control position, and a second feedforward controller is used to perform stabilization of the quadrotor system.

In both [3, 4] the control loop uses two controllers: the controller based on MPC which is used for tracking the position reference trajectory and a second controller (which is designed using other techniques) that is used for stabilization of the rotational movements. In this paper, MPC is used to control both the position and rotational movements. This is good for uniformity and ease of maintenance.

3. Kinematics and Dynamics of a Quadrotor

A quadrotor is completely described by twelve states. The first six states describe translational motion of the quadrotor. These states are x, y, and z which represent position in inertial frame and u, v, and w which denote velocity vectors along the body frame. Similarly, the remaining six variables describe rotational motions. These are ϕ, θ, and ψ commonly known as Euler angles and p, q, and r which denote angular velocities.

A free body diagram of a quadrotor is shown in Figure 1. The force F_* and moment M_* (where $* = f, b, r$, and l) produced by the rotors can be described by their angular speed (ω) as $F_* = k_1\omega^2$ and $M_* = k_2\omega^2$, respectively. Similarly, torques τ_ϕ, τ_θ, and τ_ψ and the thrust force (F) are related to angular speed of rotors as

$$\begin{pmatrix} F \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} = M \begin{pmatrix} \omega_f^2 \\ \omega_r^2 \\ \omega_b^2 \\ \omega_l^2 \end{pmatrix}, \tag{1}$$

where $M = \begin{bmatrix} k_1 & k_1 & k_1 & k_1 \\ 0 & -lk_1 & 0 & lk_1 \\ lk_1 & 0 & -lk_1 & 0 \\ -k_2 & k_2 & -k_2 & k_2 \end{bmatrix}$ and l is the length of arm of the quadrotor.

Equations describing the kinematics of the quadrotor are given by

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = R(\phi, \theta, \psi) \begin{pmatrix} u \\ v \\ w \end{pmatrix},$$
$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & \frac{s\phi}{c\theta} & \frac{c\phi}{c\theta} \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix}, \tag{2}$$

where $R(\phi, \theta, \psi) = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix}$, c stands for cos, s stand for sin, and t stands for tan. Similarly, dynamic equations can be derived using Newtons laws of motion which are listed in

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} rv - qw \\ pw - rq \\ qu - pv \end{pmatrix} + \begin{pmatrix} -g \sin\theta \\ g \cos\theta \sin\phi \\ g \cos\theta \cos\phi \end{pmatrix} + \frac{1}{m} \begin{pmatrix} 0 \\ 0 \\ -F \end{pmatrix},$$
$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} \frac{J_y - J_z}{J_x} qr \\ \frac{J_z - J_x}{J_y} pr \\ \frac{J_x - J_y}{J_z} pq \end{pmatrix} + \begin{pmatrix} \frac{1}{J_x}\tau_\phi \\ \frac{1}{J_y}\tau_\theta \\ \frac{1}{J_z}\tau_\psi \end{pmatrix}, \tag{3}$$

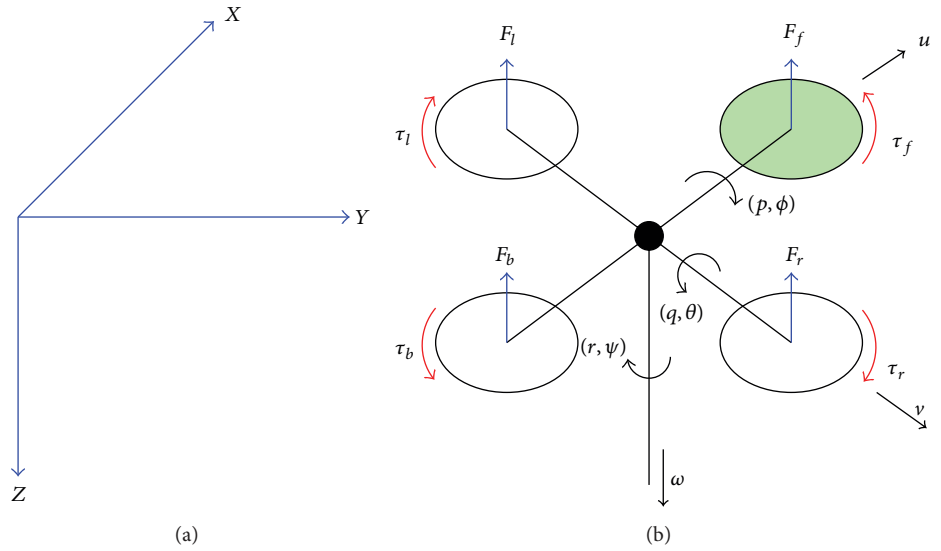
where J_x, J_y, and J_z are the respective moment of inertia about X, Y, and Z axes of the quadrotor. For the simulation of the controller of quadrotor we have considered F, τ_ϕ, τ_θ, and τ_ψ as the inputs to the system.

3.1. Linearization. In order to design a linear controller it is vital that (2) to (3) be simplified to linear ones. To do that roll, pitch, and heading angles are assumed to operate within very small angles [5] which leads to the following:

$$\begin{pmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{pmatrix} = \begin{pmatrix} \frac{1}{J_x}\tau_\phi \\ \frac{1}{J_y}\tau_\theta \\ \frac{1}{J_z}\tau_\psi \end{pmatrix},$$
$$\ddot{z} = g - \frac{F}{m},$$
$$\dot{u} = -g\theta,$$
$$\dot{v} = g\phi. \tag{4}$$

4. Controller Design

In order to control attitude (ϕ, θ, and ψ), altitude (z), and position (x, y), we propose a control architecture given in



(a) (b)

Figure 1: Inertial frame (a) and forces and torques acting on the quadrotor (b).

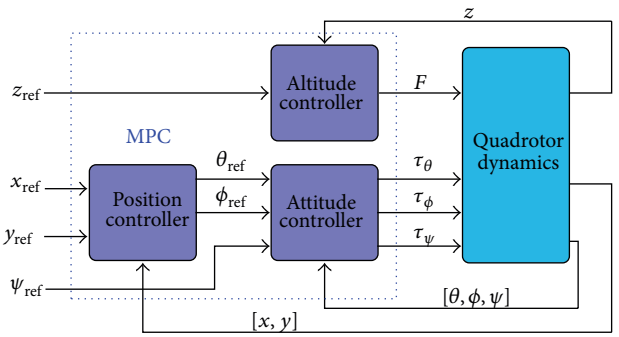


Figure 2: Block diagram showing the structure of the controllers.

Figure 2 where we have utilized MPC to design the three controllers. The attitude controller generates τ_ϕ, τ_θ, and τ_ψ actuator signals whereas the altitude controller generates required thrust for the system. The position controller controls position in the x and y directions and generates control signals θ and ϕ which, when combined with the ψ signal, act as reference signals to the attitude controller.

In the next section we look at the design of each of the three controllers.

4.1. Discretization. In this section, discrete-time state-space equations necessary for altitude, position, and attitude control are derived.

4.1.1. Altitude Controller. The linear equations responsible for altitude control are

$$\begin{aligned} \dot{z} &= w, \\ \dot{w} &= g - \frac{F}{m}. \end{aligned} \tag{5}$$

These two equations can be combined to give the following second order ODE:

$$\ddot{z} = g - \frac{F}{m}. \tag{6}$$

This second order ODE can be simplified to

$$\ddot{z} = G, \tag{7}$$

where $G = g - F/m$ or $F = m(g - G)$. Equation (7) can be written in state-space form as in the following equation:

$$\begin{bmatrix} \dot{z} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} z \\ \dot{z} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} G. \tag{8}$$

Using the forward Euler method and choosing a sampling interval ΔT, we can express (8) in discrete form as in

$$\begin{bmatrix} z(k+1) \\ v_z(k+1) \end{bmatrix} = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} z(k) \\ v_z(k) \end{bmatrix} + \begin{bmatrix} 0 \\ \Delta T \end{bmatrix} G(k). \tag{9}$$

The states $z(k)$ and $v_z(k)$ represent the position and velocity, respectively, and we could choose a proper output matrix C depending on which state we are going to measure. In this paper we are interested in the position and, therefore, we will use the output matrix $C = [1 \quad 0]$.

4.1.2. Position Controller. The equations responsible for position and velocity control in x direction are

$$\begin{aligned} \dot{x} &= u, \\ \dot{u} &= -g\theta. \end{aligned} \tag{10}$$

These two equations can be combined to give the second order ODE:

$$\ddot{x} = -g\theta. \tag{11}$$

Similarly the equations responsible for position and velocity control in y direction are

$$\begin{aligned} \dot{y} &= v, \\ \dot{v} &= g\phi, \end{aligned} \tag{12}$$

which give rise to the second order ODE:

$$\ddot{y} = g\phi. \tag{13}$$

Equations (11) and (13) can be combined and form a state-space equation such as

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{y} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -g & 0 \\ 0 & 0 \\ 0 & g \end{bmatrix} \begin{bmatrix} \theta \\ \phi \end{bmatrix}. \tag{14}$$

As stated before, we can use the forward Euler method to express (14) in discrete form. Consider

$$\begin{aligned} \begin{bmatrix} x(k+1) \\ v_x(k+1) \\ y(k+1) \\ v_y(k+1) \end{bmatrix} &= \begin{bmatrix} 1 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ v_x(k) \\ y(k) \\ v_y(k) \end{bmatrix} \\ &\quad + \begin{bmatrix} 0 & 0 \\ -g\Delta T & 0 \\ 0 & 0 \\ 0 & g\Delta T \end{bmatrix} \begin{bmatrix} \theta(k) \\ \phi(k) \end{bmatrix}. \end{aligned} \tag{15}$$

The states $x(k)$ and $y(k)$ represent the position while $v_x(k)$ and $v_y(k)$ represent velocity. Since we are interested in the position, we will use the output matrix $C = \left[\begin{smallmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{smallmatrix}\right]$ but we could have used a different 2×4 output matrix if we were interested in the velocity. But it is important to note that we can not use a 3×4 or 4×4 as there is no guarantee that we will be able to control each of the measured outputs independently with zero steady state errors. This is generally the case for a system with m inputs, q outputs, and $q > m$ [6].

4.1.3. Attitude Controller. In attitude control we are interested in controlling the roll ϕ, pitch θ, and yaw (heading) ψ such that we generate appropriate torque signals responsible for steering the quadrotor in the desired direction with the required attitude. The equations responsible for roll control are

$$\begin{aligned} \dot{\phi} &= p, \\ \dot{p} &= \frac{1}{J_x}\tau_\phi. \end{aligned} \tag{16}$$

These two equations can be combined to give the second order ODE:

$$\ddot{\phi} = \frac{1}{J_x}\tau_\phi. \tag{17}$$

Similarly the equations responsible for pitch are given by

$$\begin{aligned} \dot{\theta} &= q, \\ \dot{q} &= \frac{1}{J_y}\tau_\theta, \end{aligned} \tag{18}$$

which give rise to the second order ODE:

$$\ddot{\theta} = \frac{1}{J_y}\tau_\theta. \tag{19}$$

Also the equations responsible for heading are expressed as

$$\begin{aligned} \dot{\psi} &= r, \\ \dot{r} &= \frac{1}{J_z}\tau_\psi, \end{aligned} \tag{20}$$

which, like in roll and pitch, give rise to the second order ODE:

$$\ddot{\psi} = \frac{1}{J_z}\tau_\theta. \tag{21}$$

Equations (17), (19), and (21) can be combined and written in state-space form as in

$$\begin{bmatrix} \dot{\phi} \\ \ddot{\phi} \\ \dot{\theta} \\ \ddot{\theta} \\ \dot{\psi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \\ \psi \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ J_x^{-1} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & J_y^{-1} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & J_z^{-1} \end{bmatrix} \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix}, \tag{22}$$

where τ_ϕ, τ_θ, and τ_ψ are the torques required to give the required roll, pitch, and yaw, respectively.

Again, using the forward Euler method we express (22) in discrete form as

$$\begin{aligned} \begin{bmatrix} \phi(k+1) \\ v_\phi(k+1) \\ \theta(k+1) \\ v_\theta(k+1) \\ \psi(k+1) \\ v_\psi(k+1) \end{bmatrix} &= \begin{bmatrix} 1 & \Delta T & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \Delta T & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \phi(k) \\ v_\phi(k) \\ \theta(k) \\ v_\theta(k) \\ \psi(k) \\ v_\psi(k) \end{bmatrix} \\ &\quad + \begin{bmatrix} 0 & 0 & 0 \\ \Delta T J_x^{-1} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \Delta T J_y^{-1} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \Delta T J_z^{-1} \end{bmatrix} \begin{bmatrix} \tau_\phi(k) \\ \tau_\theta(k) \\ \tau_\psi(k) \end{bmatrix}. \end{aligned} \tag{23}$$

Since we are interested in measuring $\phi(k)$, $\theta(k)$, and $\psi(k)$, we will use the output matrix $C = \left[\begin{smallmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{smallmatrix}\right]$.

So far we have formulated the necessary equations used to build the three controllers. Figure 2 shows, in block diagram form, what we have done so far. But that is not the whole story. In the next section we will look at exactly how model predictive control is applied to the quadrotor system using the discrete state-space equations formulated in this section.

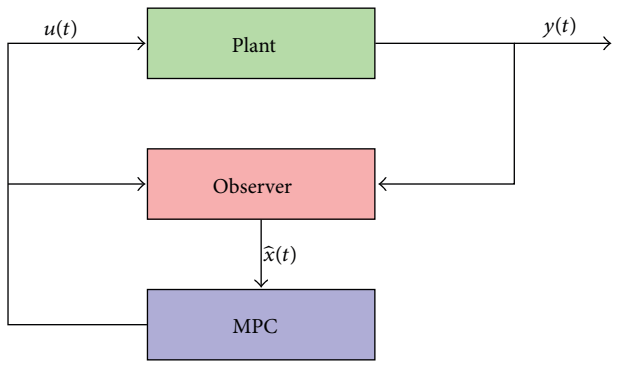


Figure 3: Block diagram of SS-MPC.

5. Model Predictive Control

One of the many tenets deployed in control of complex systems like the quadrotor using MPC is to use models that reduce the computation burden which is a necessity when it comes to online implementation. It is also agreed upon that linear models perform better in this respect than the nonlinear models they represent. In this and subsequent sections the performance of a qudrotor controller designed using Laguerre-based MPC (LMPC) is compared with that designed using the more common linear state-space approach presented in [7]. We will call the linear state-space method SS-MPC.

5.1. Linear State-Space MPC (SS-MPC). Linear state-space MPC is modeled using a linear state-space relation and plant constraints are modeled using linear equalities and inequalities. When combined with convex quadratic cost function this implies that the desired control action can be obtained, at each sample interval, via the solution of a corresponding quadratic program. This is attractive because the quadratic programs can be solved efficiently online [7]. The general block diagram representing SSMPC is shown in Figure 3.

5.1.1. Plant. The plant is modeled by using the following state-space system:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k, \\ y_k &= Cx_k + Du_k + v_k, \end{aligned} \tag{24}$$

where w_k is the state noise and v_k is the measurement noise. Both w_k and v_k are assumed to be Gaussian distributed with zero mean, respective covariances of W and V, and cross covariance Z. This is represented mathematically as

$$\begin{bmatrix} w_k \\ v_k \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} W & Z \\ Z^T & V \end{bmatrix} \right). \tag{25}$$

5.1.2. Observer. Using Gaussian assumptions stated above it is possible to make optimal predictions of state and output using the Kalman filter as

$$\begin{aligned} \widehat{x}_{k+1|k} &= A\widehat{x}_{k|k-1} + Bu + K\left(y_k - \widehat{y}_{k|k-1}\right), \\ \widehat{y}_{k|k-1} &= C\widehat{x}_{k|k-1} + Du_k. \end{aligned} \tag{26}$$

The closed loop gain K is found by solving the discrete-time algebraic Riccati equation (DARE):

$$\begin{aligned} K &= \left(APC^T + Z\right)\left(CPC^T + V\right)^{-1}, \\ P &= W + APA^T \\ &\quad - \left(APC^T + Z\right)\left(CPC^T + V\right)^{-1}\left(Z^T + CPA^T\right). \end{aligned} \tag{27}$$

5.1.3. Optimal Estimation of State and Output. The general equation representing the optimal estimate of state at instant j and written in terms of the initial state $\widehat{x}_{k+1|k}$ and future control inputs $u_{k+i|k}$ is given by

$$\widehat{x}_{k+j|k} = A^{j-1}\widehat{x}_{k+1|k} + \sum_{n=1}^{j-1} A^{j-n-1}Bu_{k+n|k}. \tag{28}$$

Similarly the output equation is given by

$$\widehat{y}_{k+j|k} = CA^{j-1}\widehat{x}_{k+1|k} + C\left(\sum_{n=1}^{j-1} A^{j-n-1}Bu_{k+n|k}\right) + Du_{k+j|k}. \tag{29}$$

The vector containing all output vectors can be written as

$$Y_k = \begin{bmatrix} y_{k+1|k} \\ \vdots \\ y_{k+N|k} \end{bmatrix}, \tag{30}$$

while that containing all control actions can be written as

$$U_k = \begin{bmatrix} u_{k+1|k} \\ \vdots \\ u_{k+N|k} \end{bmatrix}. \tag{31}$$

Therefore, we can write Y_k as

$$Y_k = F\widehat{x}_{k+1|k} + \Phi U_k, \tag{32}$$

where

$$\begin{aligned} F &= \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{N-1} \end{bmatrix}, \\ \Phi &= \begin{bmatrix} D & & & & \\ CB & D & & & \\ CAB & CB & D & & \\ \vdots & & & & \\ CA^{N-2}B & \cdots & \cdots & CB & D \end{bmatrix}. \end{aligned} \tag{33}$$

5.1.4. Cost Function. The prime goal in MPC is to reject disturbances whilst tracking a reference signal. But at each instant it has to ensure that the control signal is within reasonable range that is practical for actuation. An objective function that combines these two tasks is

$$J\left(\widehat{x}_{k+1|k}, U_k\right) = \frac{1}{2}\sum_{n=1}^{N}\left\|\widehat{y}_{k+n|k} - r_{k+n}\right\|_Q^2 + \left\|u_{k+n|k} - u_{k+n-1|k}\right\|_S^2, \tag{34}$$

where both Q and S are assumed to be symmetric and positive definite.

5.2. Laguerre-Based MPC (LMPC). Model predictive control designed using Laguerre functions is developed and summarized in [2]. The author presented this design technique for both single-variable and multivariable systems. Fortunately, the derivations for the single-variable case and the multivariable case are very similar and knowing one would give enough insight into the derivation of the other. For brevity only the single-variable case is presented and it is hoped that this is enough to set forth the LMPC design technique.

Consider a plant with p inputs, q outputs, and n states as described by (35) where the subscript m stands for model

$$\begin{aligned} x_m(k+1) &= A_m x_m(k) + B_m u(k) + B_d \omega(k), \\ y(k) &= C_m x(k), \end{aligned} \tag{35}$$

where $u(k)$ is the input variable, $y(k)$ is the process output, and $x_m(k)$ is the state vector. $\omega(k)$ is the input disturbance and is assumed to be a sequence of integrated white noise.

The plant described by (35) can be expressed in augmented state-space form [2, 6] as

$$\begin{aligned} x(k+1) &= Ax(k) + B\Delta u(k) + B_\epsilon \epsilon(k), \\ y(k) &= Cx(k), \end{aligned} \tag{36}$$

where $A = \begin{bmatrix} A_m & 0_{n\times q} \\ C_m A_m & I_{q\times q} \end{bmatrix}$, $B = \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix}$, $B_\epsilon = \begin{bmatrix} B_d \\ C_m B_d \end{bmatrix} C = \begin{bmatrix} 0_{q\times n} & I_{q\times q} \end{bmatrix}$, and the state $x(k) = \begin{bmatrix} \Delta x_m(k) & y(k) \end{bmatrix}^T$. $\epsilon(k)$ is a zero mean white noise sequence related to the disturbance by the difference equation $\epsilon(k) = \omega(k) - \omega(k-1)$ [6].

5.2.1. Design Framework. In designing MPC using Laguerre functions the control trajectory ΔU is expressed using a set of orthonormal functions called *Laguerre functions*. Since the state and output vectors can also be described in terms of ΔU, it follows that they too can be expressed using Laguerre functions.

At sampling instant k_i, the state variable $x(k_i)$ is available through measurement. The future control trajectory is given by

$$\Delta U = \left[\Delta u(k_i), \Delta u(k_i+1), \ldots, \Delta u(k_i+N_c-1)\right], \tag{37}$$

where N_c is the control horizon.

Laguerre functions can approximate the incremental terms contained in ΔU. The z-transfroms of the discrete-time Laguerre functions are written as

$$\begin{aligned} \Gamma_1(z) &= \frac{\sqrt{1-a^2}}{1-az^{-1}}, \\ \Gamma_2(z) &= \frac{\sqrt{1-a^2}}{1-az^{-1}}\left(\frac{z^{-1}-a}{1-az^{-1}}\right), \\ &\vdots \\ \Gamma_N(z) &= \frac{\sqrt{1-a^2}}{1-az^{-1}}\left(\frac{z^{-1}-a}{1-az^{-1}}\right)^{N-1}. \end{aligned} \tag{38}$$

In the discrete-time Laguerre network of (38), a is the pole of the discrete-time Laguerre network and $0 \le a < 1$ for stability of the network. The parameter a is called the scaling factor and $N = 1, 2, 3, \ldots$ is the number of Laguerre terms used in the network. Note that

$$\Gamma_k(z) = \Gamma_{k-1}(z)\left(\frac{z^{-1}-a}{1-az^{-1}}\right), \tag{39}$$

with the first term as $\Gamma_1(z) = \sqrt{1-a^2}/(1-az^{-1})$.

Let $l_j(k)$ be the inverse z-transform of the jth term in the discrete Laguerre network and $L(k)$ the vector containing all inverse z-transform terms. Then taking advantage of (39), successive inverse z-transform vectors are obtained through the difference equation:

$$L(k+1) = A_l L(k). \tag{40}$$

The matrix A_l has size $N \times N$ and is a function of parameters a and $\beta = 1 - a^2$. The initial condition is given by

$$L(0)^\top = \sqrt{\beta}\begin{bmatrix} 1 & -a & a^2 & \cdots & (-1)^{N-1}a^{N-1} \end{bmatrix}. \tag{41}$$

In the case where $N = 5$, matrix A_l and the initial condition $L(0)$ are

$$\begin{bmatrix} a & 0 & 0 & 0 & 0 \\ \beta & a & 0 & 0 & 0 \\ -a\beta & \beta & a & 0 & 0 \\ a^2\beta & -a\beta & \beta & a & 0 \\ -a^3\beta & a^2\beta & -a\beta & \beta & a \end{bmatrix}, \tag{42}$$

$$L(0) = \sqrt{\beta}\begin{bmatrix} 1 \\ -a \\ a^2 \\ -a^3 \\ a^4 \end{bmatrix}, \tag{43}$$

respectively.

At sampling instant k_i, the control trajectory is described using Laguerre functions as in

$$\Delta u(k_i + k) = \sum_{j=1}^{N} c_j(k_i)\, l_j(k_i), \tag{44}$$

where $k = 0, 1, 2, \ldots, N_p$ with N_p as the prediction horizon, N terms used in the expansion and $l_j(k_i)$ is the inverse z-transform of the jth term in the discrete Laguerre network.

Equation (44) can be rewritten as

$$\Delta u\left(k_i + k\right) = L\left(k\right)^{\top} \eta, \tag{45}$$

where $L(k) = \left[l_1(k) \;\; l_2(k) \;\; \cdots \;\; l_N(k)\right]^T$, $\eta = \left[c_1 \;\; c_2 \;\; \cdots \;\; c_N\right]^T$, and the coefficients $c_1, c_2, \ldots, c_N$ are obtained from system data.

At an arbitrary future instant m, the state is described using Laguerre functions as

$$x\left(k_i + m \mid k_i\right) = A^m x\left(k_i\right) + \sum_{i=0}^{m-1} A^{m-i-1} BL\left(i\right)^T \eta. \tag{46}$$

Similarly, the output is described as

$$y\left(k_i + m \mid k_i\right) = CA^m x\left(k_i\right) + \sum_{i=0}^{m-1} CA^{m-i-1} BL\left(i\right)^T \eta. \tag{47}$$

5.2.2. Cost Function. The cost function is used to choose the optimal control trajectory ΔU to bring the predicted output as close as possible to the set-point. The cost function

$$J = \sum_{m=1}^{N_p} x\left(k_i + m \mid k_i\right)^T Qx\left(k_i + m \mid k_i\right) + \eta^T R_L \eta \tag{48}$$

minimizes the error between the set-point signal and the output in the shortest possible time by carefully tuning the weighting matrices $Q \geq 0$ and $R_L > 0$.

5.2.3. Cost Function Minimization. The state variable in (46) can be rewritten as

$$x\left(k_i + m \mid k_i\right) = A^m x\left(k_i\right) + \phi\left(m\right)^T \eta, \tag{49}$$

where $\phi(m)^T = \sum_{i=0}^{m-1} A^{m-i-1} BL(i)^T$.

By substituting (49) into (48) and performing the partial derivative $\partial J/\partial \eta = 0$, the Laguerre coefficients vector is found to be

$$\eta = -\Omega^{-1} \Psi x\left(k_i\right), \tag{50}$$

with $\Omega = \sum_{m=1}^{N_p} \phi(m) Q \phi(m)^T + R_L$ and $\Psi = \phi(m) Q A^m$.

5.2.4. Receding Horizon Control. In receding horizon control (RHC) only the first term in ΔU, that is, $\Delta u(k_i)$, is implemented at instant k_i. The rest of the sequence is ignored. Only the most recent measurement is taken to form the state vector for calculation of the control signal. This procedure is repeated in real-time to give the receding horizon control law [6, 8].

5.2.5. Stability. In stability analysis we make use of the technique of exponential data weighting originated by Anderson and Moore [9] and applied to MPC in [6]. More specifically we will concentrate on the discrete exponential factor $e^{\lambda \Delta t}$ where Δt is the sampling interval and the discrete weights form a geometric sequence $\{\alpha^j, j = 0, 1, 2, \ldots\}$.

The proposed cost function is similar to the one used in linear quadratic regulator (LQR) systems but with the inclusion of discrete weights

$$\begin{aligned} J_w = &\sum_{j=1}^{N_p} \alpha^{-2j} x\left(k_i + j \mid k_i\right)^T Qx\left(k_i + j \mid k_i\right) \\ &+ \sum_{j=0}^{N_p} \alpha^{-2j} \Delta u\left(k_i + j\right)^T R \Delta u\left(k_i + j\right). \end{aligned} \tag{51}$$

For $\alpha > 1$, the exponential weights α^{-2j}, $j = 1, 2, 3, \ldots, N_p$, put more emphasis on the current state $x(k_i + j \mid k_i)$ and less emphasis on subsequent future states.

The exponentially weighted cost function can be expressed more compactly as

$$\begin{aligned} J_w = &\sum_{j=1}^{N_p} \widehat{x}\left(k_i + j \mid k_i\right)^T Q \widehat{x}\left(k_i + j \mid k_i\right) \\ &+ \sum_{j=0}^{N_p} \Delta \widehat{u}\left(k_i + j\right)^T R \Delta \widehat{u}\left(k_i + j\right), \end{aligned} \tag{52}$$

with state equation

$$\widehat{x}\left(k + 1\right) = A_\alpha \widehat{x}\left(k\right) + B_\alpha \Delta \widehat{u}\left(k\right), \tag{53}$$

and $A_\alpha = A/\alpha$ and $B_\alpha = B/\alpha$.

With $Q \geq 0$, $R > 0$, and $N_p \to \infty$, minimizing the cost function J_w is equivalent to the DLQR problem which is solved using algebraic Riccati equation (54). The pair (A_α, B_α) is assumed to be controllable and (A_α, C) observable with $Q = C^T C$. Then there is a stabilizing state feedback control gain matrix K_w,

$$\begin{aligned} K_w &= \left(R + \alpha^{-2} B^T P_w B\right)^{-1} \alpha^{-2} B^T P_w A, \\ \frac{A^T}{\alpha} &\left[P_w - P_w \frac{B}{\alpha} \left(R + \frac{B^T}{\alpha} P_w \frac{B}{\alpha}\right)^{-1} \frac{B^T}{\alpha} P_w\right] \frac{A}{\alpha} + Q - P_w = 0 \end{aligned} \tag{54}$$

that gives a stable closed loop system with all its eigenvalues inside the unit circle and the the closed loop system being described by

$$\widehat{x}\left(k_i + j + 1 \mid k_i\right) = \alpha^{-1} \left(A - BK_w\right) \widehat{x}\left(k_i + j \mid k_i\right). \tag{55}$$

From (55), the transformed system has all its eigenvalues inside the unit circle by taking $N_p \to \infty$. So

$$\alpha^{-1} \left|\lambda_{\max}\left(A - BK_w\right)\right| < 1, \tag{56}$$

which gives

$$\left|\lambda_{\max}\left(A - BK_w\right)\right| < \alpha. \tag{57}$$

Thus by choosing $\alpha > 1$ there is a great chance that the system will be stable. Several simulations on the quadrotor system indicate that a choice of α slightly greater than unity makes the system stable almost all the time.

Table 1

Parameter	Value
Mass, m	0.7 kg
Length, l	0.275 m
J_x	0.1063 kg·m^2
J_y	0.1063 kg·m^2
J_z	0.2122 kg·m^2

Table 2

Control horizon N_c	N	Required a
10	5	0.6065
15	5	0.7165
20	5	0.7788
25	5	0.8187
50	5	0.9048

6. Simulation Parameters and Results

In order to illustrate the controller's stability we will check if the eigenvalues of each of the three individual controllers appear inside a unit circle. And we will use a trajectory in a 3D plane to check the overall controller's ability to track a desired trajectory.

Table 1 shows the parameters of the quadrotor used in simulation.

6.1. Stability Analysis. In this section we look at the location of eigenvalues of LMPC and check if they fall within the unit circle for stability of the system. We also compare them to those obtained from the optimal DLQR system. To ensure stability as explained in Section 5.2.5 we use $\alpha = 1.2$.

By looking at Figure 4 we note that all the eigenvalues appear inside the unit circle as required. Also the eigenvalues obtained from LMPC closely match with those of the optimal DLQR. This not only shows the system is stable but also shows the controllers perform optimally.

6.2. Trajectory Tracking. In this section we look at the performance of LMPC when used to track a 3D trajectory. Also LMPC is compared to SS-MPC and individual state trajectories are observed. The starting point of the trajectory is $(x, y, z) = (0, 0, 0)$ and final point is $(x, y, z) = (5, 5, 10)$. Parameter $a = 0.8187$ while $N = 5$. See Figure 5.

Clearly both LMPC and SS-MPC track the desired trajectory very well. See Figures 5, 6 and 7. However, LMPC performs better than SS-MPC in that it requires only five parameters ($N = 5$) to capture the control trajectory compared to SS-MPC's minimum of 25 ($N_c = 25$) parameters (see Table 2 for relationship of N and N_c with $a = 0.8187$).

6.3. Control Horizon in Relation to a and N. The control horizon (N_c) is related to the parameters a and N [2] by

$$a \approx e^{-N/N_c}. \tag{58}$$

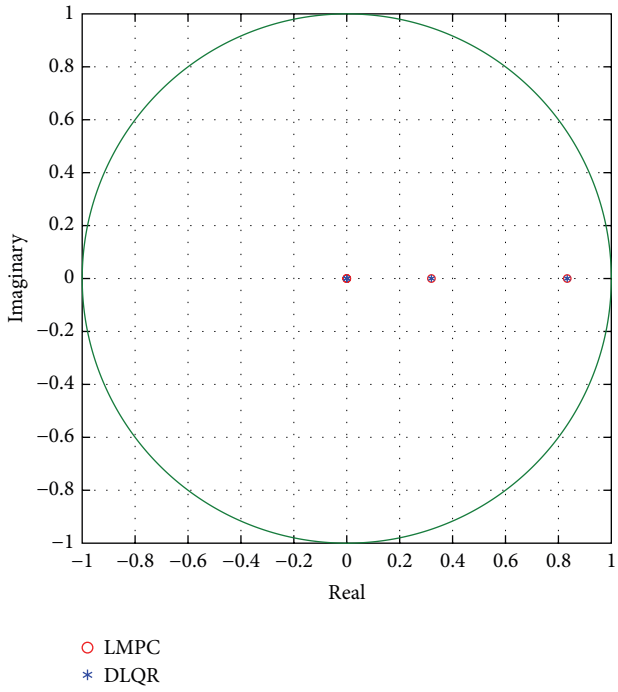


Figure 4: Eigenvalues for DLQR and LMPC in altitude control.

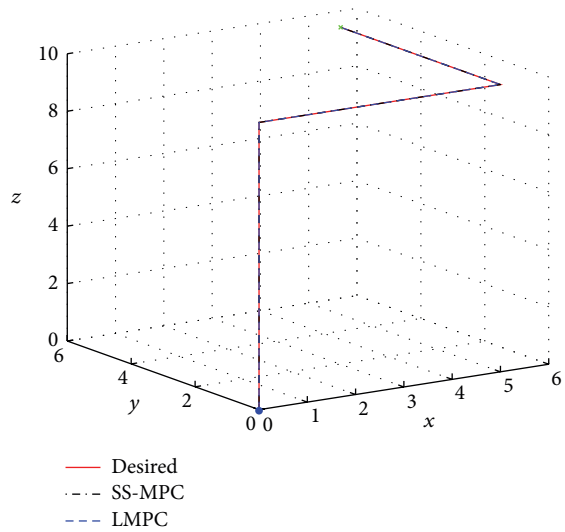


Figure 5: Tracking a 3D trajectory.

6.3.1. Constant N. Using (58) and choosing a constant parameter $N = 5$ we get Table 2.

Thus we can increase the control horizon by increasing the parameter a without necessarily changing the order N.

6.3.2. Constant Control Horizon N_c. Assuming we want to achieve a control horizon of 30, we choose one parameter usually N because it is an integer and directly defines the order of the discrete Laguerre network. Therefater, we simply use (58) to find the corresponding scaling factor a as shown in Table 3.

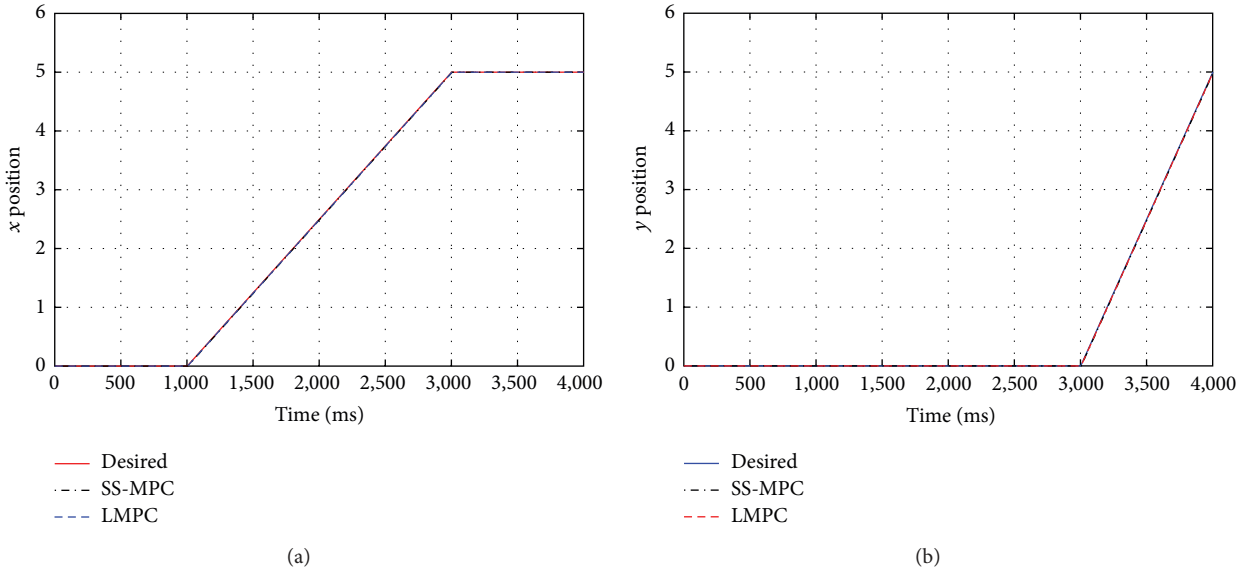


Figure 6: Tracking x (a) and y in (b).

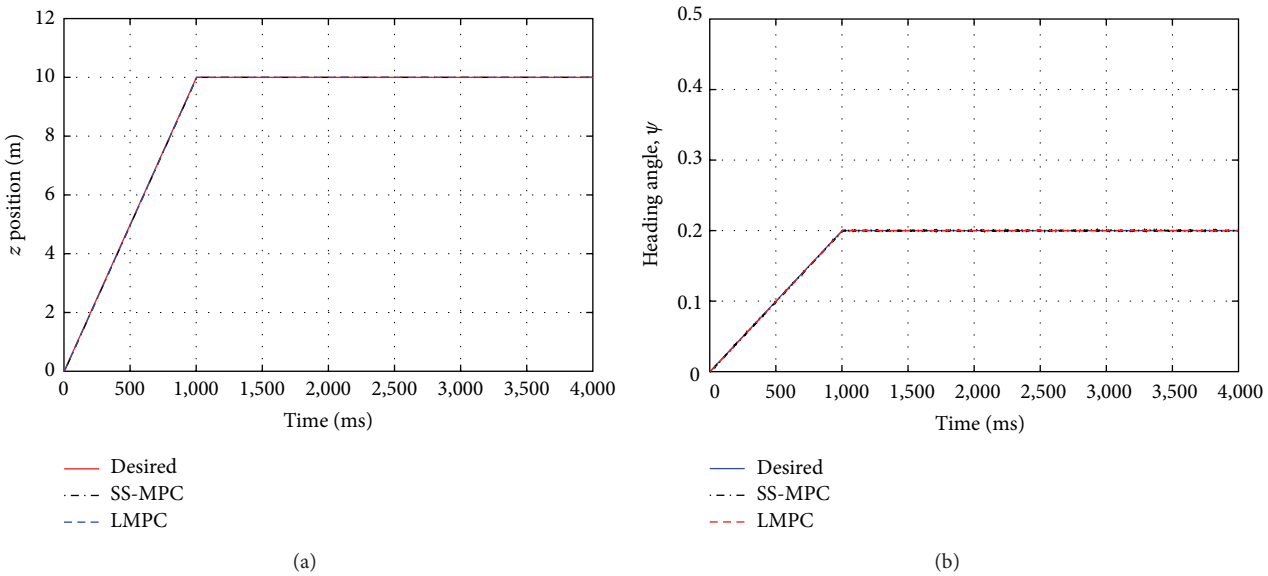


Figure 7: Altitude (z) (a) and heading angle (ψ) (b).

Table 3

Target control horizon N_c	Chosen N	Required a
30	1	0.9672
30	3	0.9048
30	5	0.8465
30	7	0.7919
30	9	0.7408

Figures 8, 9, 10, 11, 12, and 13 compare two LMPC designs both of which yield a control horizon of 30. One design uses $N = 1$ and the other $N = 9$. Thus we can choose a Laguerre network with lower number of terms N (that gives lower computation burden) but with a larger scaling factor a to achieve the same control horizon. The computational cost is lower if a smaller number of parameters are used. For $N = 1$, only one Laguerre term is used to capture the control trajectory while $N = 9$ requires nine Laguerre terms to capture the same control trajectory.

7. Conclusion

This paper has presented a solution to stability and trajectory tracking of quadrotor systems using a model predictive controller designed by using a special type of orthonormal functions called Laguerre functions. A quadratic cost

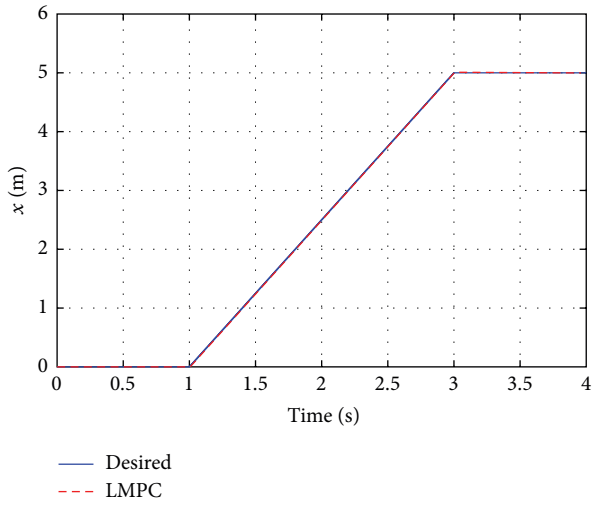


Figure 8: x trajectory $N = 1$.

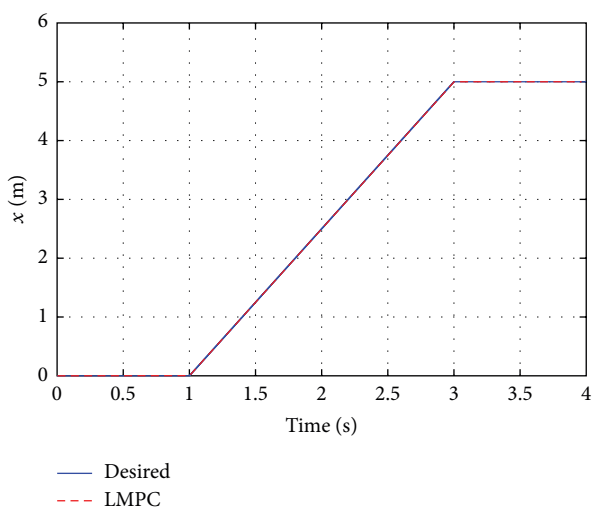


Figure 9: x trajectory $N = 9$.

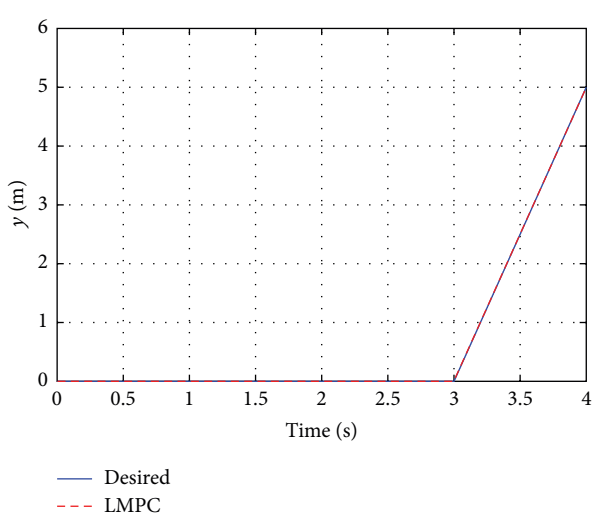


Figure 10: y trajectory $N = 1$.

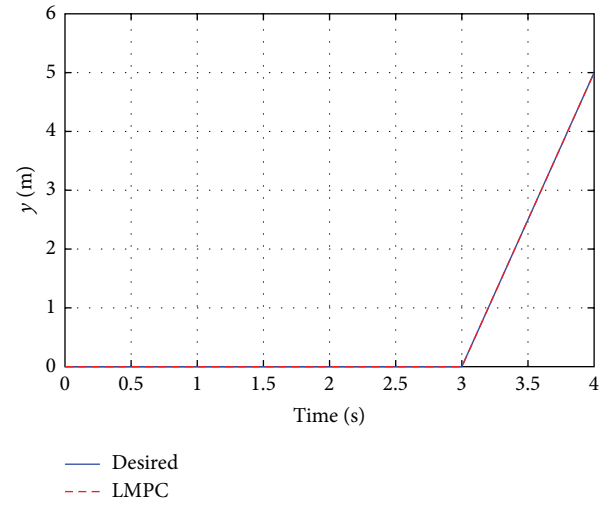


Figure 11: y trajectory $N = 9$.

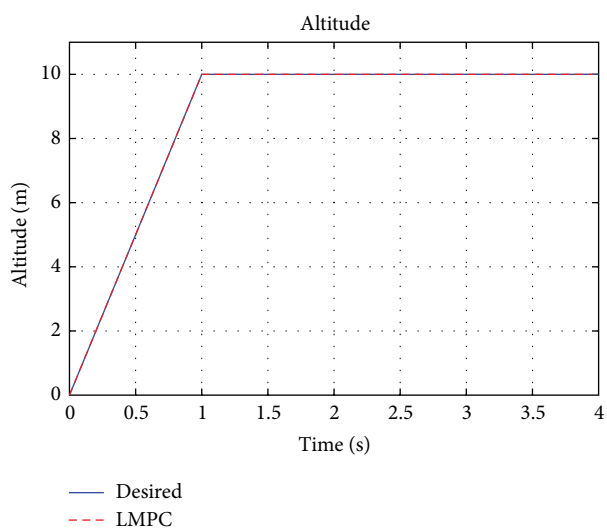


Figure 12: z trajectory $N = 1$.

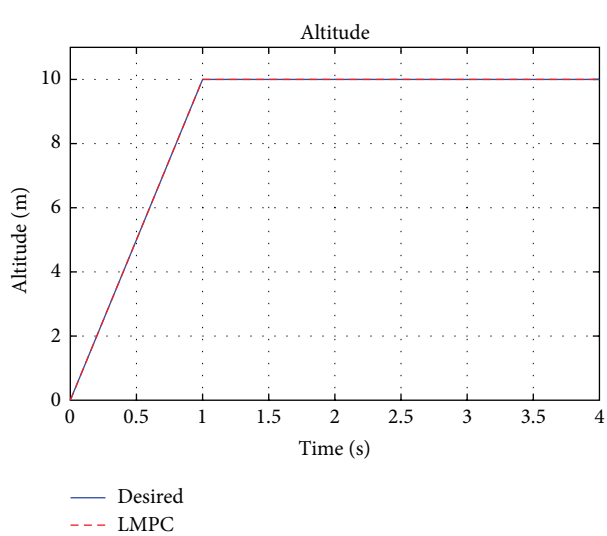


Figure 13: z trajectory $N = 9$.

function similar to that used in linear quadratic regulator (LQR) has been used. In stability analysis LMPC has been compared to the optimal DLQR system whereas in trajectory tracking LMPC has been compared to a popular linear state-space MPC desgin technique in which plant constraints are modeled using linear equalities and inequalities. The results from the simulations indicate that the controller performs very well and is considered feasible. With this perceived feasibility LMPC offers the added advantage that it can handle systems where rapid sampling and more complicated process dynamics are required [5, 10]. By selecting appropriate values of a and N LMPC reduces the number of parameters required for accurate prediction when using the traditional (MPC) approach. This is a big advantage in that, with the reduced number of parameters, online implementation might be possible where the traditional MPC would have failed.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MEST) (no. 2013R1A2A2A01068127 and no. 2013R1A1A2A10009458).

References

[1] J. M. Maciejowski, *Predictive Control with Constraints*, Prentice Hall, New York, NY, USA, 1st edition, 2002.

[2] L. Wang, "Discrete time model predictive control design using Laguerre functions," in *Proceedings of the American Control Conference*, pp. 2430–2435, Arlington, Va, USA, June 2001.

[3] K. Alexis, G. Nikolakopoulos, and A. Tzes, "Experimental model predictive attitude tracking control of a quadrotor helicopter subject to wind-gusts," in *Proceedings of the 18th Mediterranean Conference on Control & Automation (MED '10)*, pp. 1461–1466, IEEE, Marrakech, Morocco, June 2010.

[4] G. V. Raffo, M. G. Ortega, and F. R. Rubio, "MPC with nonlinear H_∞ control for path tracking of a quad-rotor helicopter," in *Proceedings of the 17th World Congress, International Federation of Automatic Control (IFAC '08)*, July 2008.

[5] A. M. Singh, D. J. Lee, D. P. Hong, and K. T. Chong, "Successive loop closure based controller design for an autonomous quadrotor vehicle," *Applied Mechanics and Materials*, vol. 483, pp. 361–367, 2014.

[6] L. Wang, *Model Predictive Control Design and Implementation Using MATLAB*, Advances in Industrial Control, Springer, New York, NY, USA, 1st edition, 2009.

[7] A. G. Wills, "Technical report EE04025-Notes on linear model predictive control," December 2004.

[8] Y. Gao, C. G. Lee, and K. T. Chong, "Receding horizon tracking control for wheeled mobile robots with time-delay," *Journal of Mechanical Science and Technology*, vol. 22, no. 12, pp. 2403–2416, 2008.

[9] B. D. O. Anderson and J. B. Moore, *Linear Optimal Control*, Prentice-Hall, 1971.

[10] J. A. Rossiter and L. Wang, "Exploiting Laguerre functions to improve the feasibility/performance compromise in MPC," in *Proceedings of the 47th IEEE Conference on Decision and Control (CDC '08)*, pp. 4737–4742, December 2008.