

Chapter 11

Optimal Control Strategies for Load Carrying Drones

Alicia Arce Rubio, Alexandre Seuret, Yassine Ariba
and Alessio Mannisi

Abstract This chapter studies control strategies for load carrying drones. Load carrying drones not only have to fly in a cooperative way, but also are mechanically interconnected. Due to these characteristics, the control problem is an interesting and challenging issue to deal with. Throughout this chapter, a dynamic model based on first principle is developed. To that end, it is proposed to model this system as a ball and beam system lifted by two drones. Afterwards, different control techniques are implemented and compared by simulations. Specifically, linear-quadratic regulator (LQR) and model predictive control (MPC) are studied. Both control techniques belong to the optimal control methodology. This comparison is interesting since LQR permits to perform an optimal control law with short execution times, while MPC deals with physical constraints and predictions, being the execution time and the physical constraints important issues to handle in this kind of systems. Finally, simulation results and open issues are discussed.

A.A. Rubio (✉) · A. Seuret · Y. Ariba · A. Mannisi
CNRS, LAAS, 7 avenue du colonel Roche, 31400 Toulouse, France
e-mail: aarce@laas.fr

A. Seuret
e-mail: aseuret@laas.fr

Y. Ariba
e-mail: ariba@icam.fr

A. Mannisi
e-mail: amanissi@laas.fr

A.A. Rubio · A. Seuret · Y. Ariba · A. Mannisi
LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France

Y. Ariba
Icam Toulouse, Toulouse, France

11.1 Introduction

Interest in using drones (that is, unmanned aerial vehicles—UAVs—with the capacity to fly semi- or fully autonomously thanks to an on-board computer and sensors [1]) for scientific investigations dates back to the 1970s. Since then, billions of dollars have been poured into research and development of military and experimental drones. Indeed, during the last years, an increased use of flying drones has been noticed. The invention of light materials, low energy consumption machines, and high performance processing units led to the construction of flexible flying robots. They can be used in a variety of applications such as vehicle tracking, traffic management and fire detection [2, 3]. Within the family of the vertical take-off and landing (VTOL) drones, unmanned quadrotor helicopters [4] that base their operation in the appropriate control of four rotors have received a growing attention, mainly due to their capability to outperform most of other types of helicopters on the issues of maneuver ability, survivability, simplicity of mechanics, and increased payloads [5]. In fact, there are several advantages to quadcopters over comparably-scaled helicopters: the simplicity of their mechanical structure, the use of four small propellers resulting in a more fault-tolerant mechanical design capable of aggressive maneuvers at low altitude, good maneuver ability, and increased payload [6]. Untapping the potential of quadrotors requires, however, advanced control designs so as to achieve precise trajectory tracking combined with effective disturbance attenuation, particularly since quadrotor's model is highly nonlinear and their flight performance can be influenced by sudden wind gusts especially during flights in low altitudes. Moreover, the application studied in this chapter, which is the control of multiple quadrotor robots that cooperatively grasp and transport a payload in two dimensions, adds difficulty to the problem. Although the problem associated to quadrotor control has been addressed by many publications (such as those focused in PID control [7], sliding mode control [8], H_∞ control [9] and bounded control [10]), the novelty of the work presented herein is the application and subsequent comparison of model predictive control (MPC) [11] and linear-quadratic regulator (LQR) control techniques. To the best knowledge of the authors of this article this has not been realized before, a fact which further supports the interest of this work. Such a comparison is valuable since LQR permits to perform an optimal control law with short execution times while MPC deals with physical constraints and predictions. Execution time and physical constraints being important issues to take into account, while facing the control problem discussed in this chapter, the proposed application and comparison of MPC and LQR techniques therefore represents a useful framework aimed to provide researchers in the area with additional control possibilities.

To reach these ends, the chapter is organized as follows: Sect. 11.2 describes the system under study, while Sect. 11.3 presents the dynamic model. In Sect. 11.4, the control problem is motivated and control methodologies are developed. Section 11.5 shows and discusses the simulation results. Finally, the conclusions are exposed in Sect. 11.6.

11.2 System Under Study

The system under study is composed of two drones which aim to carry a load. The main feature of this system is that load carrying drones present mechanic links. These mechanic links depend on the way the drones carry the load. Therefore, it is proposed to describe this kind of system as a ball and beam system lifted by the drones. The mass center of the ball and beam models the load mass center. Figure 11.1 shows a scheme of the proposed system.

As observed in Fig. 11.1, drones are assumed to be quadrotors. The quadrotor comprises four propellers each one. The quadrotor trajectory is regulated by the angular speeds of the propellers resulting in a lift force which is referred to as f_1 for drone 1 and f_2 for drone 2 in Fig. 11.1. The ball and beam systems are lifted by the couple of drones by means of rigid cables with a fixed length equal to h . The beam length is equal to $2L$. In addition, it is supposed that the beam is nondeformable.

For sake of simplicity, in this work it is assumed that the drones only move in the XZ plane. Specifically, the x position of the drones is fixed, while the degree of freedom is the altitude z_1 for drone 1 and z_2 for drone 2. Thus, the longitudinal distance between the drones is fixed to the value of the beam length ($2L$). The angles formed by the vertical axis and the rigid cables are denoted as ϕ_1 for drone 1 and ϕ_2 for drone 2 and the angle formed by the beam and the horizontal axis is denoted θ .

It is defined two different coordinate reference systems x_0Oz_0 and $x_BO_Bz_B$. The global coordinate reference system x_0Oz_0 is located in the ground fixed in the x position corresponding to the middle distance between the drones. The local frame $x_BO_Bz_B$ is located in the beam mass center.

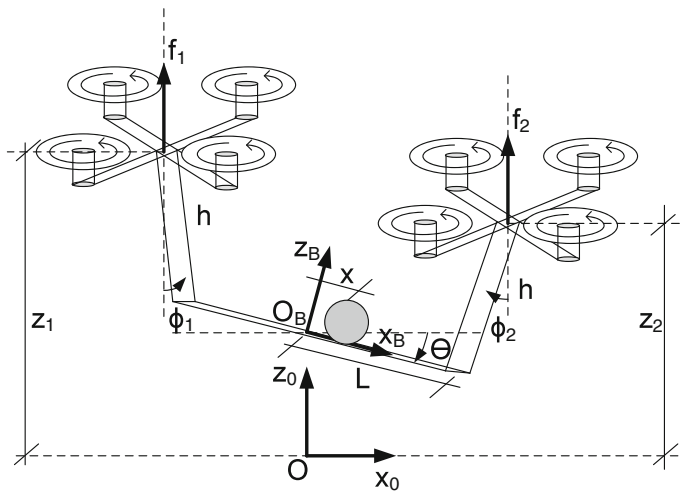


Fig. 11.1 Drone ball and beam system

Table 11.1 System parameters

Parameter	Value
Ball mass	0.1 kg
Beam mass	4 kg
Beam length	2 m
Drone 1	10 kg
Drone 2	10 kg
Rigid cable length	1 m
Damping factor	0.5

Variables related to the gyroscopic effects are not included in this study, since the control is divided in two levels. The control structure based on two control levels has been previously proposed for tracking positioning of quadrotors [9]. In our case, the high-level control calculates the references for the lift forces f_1 and f_2 , while the low-level control is dedicated to the drone stabilization. Herein, the drone stabilization is assumed to be perfectly controlled to be focused on the high-level control.

The main parameters of the system are listed in Table 11.1.

11.3 Dynamic Model

The system under study presented in the previous section is modeled with first principles equations. To that end, the kinematics equations are developed and afterwards the Lagrange–Euler equations are obtained.

11.3.1 Kinematics Equations

As previously mentioned, two different coordinate reference systems are defined. The local frame position, O_B , at the global frame is

$$O_B = \begin{bmatrix} L - h \sin \phi_2 - L \cos \theta \\ z_2 - h \cos \phi_2 - L \sin \theta \end{bmatrix}. \quad (11.1)$$

This matrix represents the transformation from local frame to global frame. The local frame speed, \dot{O}_B , at the global frame is obtained by deriving the position with respect to the time:

$$\dot{O}_B = \begin{bmatrix} -h \dot{\phi}_2 \cos \phi_2 + L \dot{\theta} \sin \theta \\ \dot{z}_2 + h \dot{\phi}_2 \sin \phi_2 - L \dot{\theta} \cos \theta \end{bmatrix}. \quad (11.2)$$

The ball position and speed at local frame are denoted x and \dot{x} and using the transformation matrices Eqs. (11.1) and (11.2), the position and speed at global frame, x_0 and \dot{x}_0 , are:

$$\begin{bmatrix} x_0 \\ z_0 \\ \dot{x}_0 \\ \dot{z}_0 \end{bmatrix} = \begin{bmatrix} L - h\sin\phi_2 - L\cos\theta - x\cos\theta \\ z_2 - h\cos\phi_2 - L\sin\theta - x\sin\theta \\ -h\dot{\phi}_2\cos\phi_2 + (L+x)\dot{\theta}\sin\theta - \dot{x}\cos\theta \\ \dot{z}_2 + h\dot{\phi}_2\sin\phi_2 - (L+x)\dot{\theta}\cos\theta - \dot{x}\sin\theta \end{bmatrix}. \quad (11.3)$$

Given that the longitudinal distance between the drones is fixed, the angle θ can be expressed as a function of angles ϕ_1 and ϕ_2 :

$$\begin{bmatrix} \sin\theta \\ \cos\theta \end{bmatrix} = \frac{1}{2L} \begin{bmatrix} z_2 - z_1 + h\cos\phi_1 - h\cos\phi_2 \\ 2L - h\sin\phi_1 - h\sin\phi_2 \end{bmatrix}. \quad (11.4)$$

Replacing θ in Eq. (11.3), the ball position and speed at global frame result in:

$$\begin{bmatrix} x_0 \\ z_0 \\ \dot{x}_0 \\ \dot{z}_0 \end{bmatrix} = \begin{bmatrix} x + \frac{hs\phi_1 - hs\phi_2}{2} - \frac{x(hs\phi_1 + hs\phi_2)}{2L} \\ \frac{z_2 + z_1 - hc\phi_1 - hc\phi_2}{2} + \frac{x(z_2 - z_1 + hc\phi_1 - hc\phi_2)}{2L} \\ \left(\frac{(x-L)h\dot{\phi}_1c\phi_1 - (x+L)h\dot{\phi}_2c\phi_2}{2L} - \frac{\dot{x}(2L + hs\phi_1 + hs\phi_2)}{2L} \right) \\ \left(\frac{(L+x)(\dot{z}_2 + h\dot{\phi}_2s\phi_2) + (L-x)(\dot{z}_1 + h\dot{\phi}_1s\phi_1)}{2L} + \frac{\dot{x}(z_2 - z_1 + hc\phi_1 - hc\phi_2)}{2L} \right) \end{bmatrix}, \quad (11.5)$$

where $c\theta$ and $s\theta$ correspond to $\cos\theta$ and $\sin\theta$, respectively.

11.3.2 Lagrange–Euler Equations

The motion equations can be expressed by the Lagrange-Euler formulation based on the kinetic and potential energy concepts:

$$\begin{bmatrix} \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{x}} - \frac{\partial \mathcal{L}}{\partial x} \\ \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{z}_1} - \frac{\partial \mathcal{L}}{\partial z_1} \\ \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{z}_2} - \frac{\partial \mathcal{L}}{\partial z_2} \\ \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{\phi}_1} - \frac{\partial \mathcal{L}}{\partial \phi_1} \\ \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{\phi}_2} - \frac{\partial \mathcal{L}}{\partial \phi_2} \end{bmatrix} = \begin{bmatrix} 0 \\ f_1 \\ f_2 \\ 0 \\ 0 \end{bmatrix}, \quad (11.6)$$

where \mathcal{L} is the Lagrangian of the system. The Lagrangian is calculated as the difference between kinetic and potential energies. The equations are obtained by the software MAXIMA.

The system kinetic and potential energies are the addition of the ball, beam, and drone kinetic and potential energies. The ball kinetic energy, T_b , beam kinetic energy, T_B , and drone kinetic energies, T_{d1} and T_{d2} are:

$$\begin{aligned} T_b &= \frac{1}{2} m_b [x_0, y_0] \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} & T_B &= \frac{1}{2} \rho \int_{-l}^l \|\dot{S}_0(s)\|^2 ds, \\ T_{d1} &= \frac{1}{2} m_{d1} \dot{z}_1^2, & T_{d2} &= \frac{1}{2} m_{d2} \dot{z}_2^2, \end{aligned} \quad (11.7)$$

being m_b the ball mass, l the beam volume, m_{d1} the drone 1 mass, m_{d2} the drone 2 mass and ρ the beam density. The beam density ρ is equal to the ratio between the beam mass, m_B , and the beam volume l . The variable \dot{S}_0 corresponds to the speed of a generic point of the beam at the global coordinate reference system. The ball potential energy, U_b , the beam potential energy, U_B , and the drone potential energies, U_{d1} and U_{d2} , are expressed as

$$\begin{aligned} U_b &= m_b g y_0, & U_B &= m_B g O_{B,z}, \\ U_{d1} &= m_{d1} g z_1, & U_{d2} &= m_{d2} g z_2, \end{aligned} \quad (11.8)$$

where g is the acceleration of gravity and $O_{B,z}$ is the z position of the local frame which is placed at the beam mass center.

11.3.3 State-Space Model

The previous model is linearized at an operating point for control purposes. The operating point corresponds to a ball position equal to $(0, 0)$, drone altitudes z_1, z_2 equal to 0 and null system speeds (ball, beam, and drones). Moreover, it is considered as new variables F_1 and F_2 to represent the lift forces. These variables are equal to zero at equilibrium, and are calculated as

$$\begin{aligned} F_1 &= f_1 - m_{d1} g - \frac{m_b + m_B}{2} g, \\ F_2 &= f_2 - m_{d2} g - \frac{m_b + m_B}{2} g. \end{aligned} \quad (11.9)$$

As a result, the linear model in matrix form is

$$\underbrace{\begin{bmatrix} \ddot{x}(t) \\ \ddot{z}_1(t) \\ \ddot{z}_2(t) \\ \ddot{\phi}_1(t) \\ \ddot{\phi}_2(t) \end{bmatrix}}_{\dot{r}} = \mathbf{A} \underbrace{\begin{bmatrix} x(t) \\ z_1(t) \\ z_2(t) \\ \phi_1(t) \\ \phi_2(t) \end{bmatrix}}_{r} + \mathbf{B} \underbrace{\begin{bmatrix} F_1(t) \\ F_2(t) \end{bmatrix}}_{u}, \quad (11.10)$$

where matrices \mathbf{A} and \mathbf{B} are given by

$$\mathbf{A} = \begin{bmatrix} 0 & \frac{g(2m_B + m_b)}{4Lm_B} & -\frac{g(2m_B + m_b)}{4Lm_B} & \frac{g(m_B + m_b)}{4m_B} & -\frac{g(m_B + m_b)}{4m_B} \\ a_1 & 0 & 0 & 0 & 0 \\ -a_2 & 0 & 0 & 0 & 0 \\ 0 & -\frac{gm_b}{4Lhm_B} & \frac{gm_b}{4Lhm_B} & -\frac{g(m_B + m_b)}{hm_B} & \frac{g(m_B + m_b)}{hm_B} \\ 0 & \frac{gm_b}{4Lhm_B} & -\frac{gm_b}{4Lhm_B} & \frac{g(m_B + m_b)}{hm_B} & -\frac{g(m_B + m_b)}{hm_B} \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 \\ \frac{8m_B + 3m_b + 12m_{d2}}{4m_B + 3m_b} & \frac{4m_B + 3m_b}{4m_B + 3m_b} \\ -\frac{4m_B + 3m_b}{p} & \frac{8m_B + 3m_b + 12m_{d1}}{p} \\ 0 & 0 \\ 0 & 0 \end{bmatrix},$$

and where

$$a_1 = \frac{3gm_b(2m_B + m_b + 2m_{d2})}{p}, \quad a_2 = \frac{3gm_b(2m_B + m_b + 2m_{d1})}{p},$$

$$p = L(2m_B m_b + (8m_B + 3m_b)(m_{d1} + m_{d2}) + 12m_{d1}m_{d2} + 4m_B^2),$$

From Eq. (11.10), the following state-space model is obtained:

$$\underbrace{\begin{bmatrix} \ddot{\Gamma} \\ \dot{\Gamma} \end{bmatrix}}_{\dot{\mathbf{X}}} = \underbrace{\begin{bmatrix} \mathbf{M} & \mathbf{A}^T \\ \mathbf{I} & \mathbf{0} \end{bmatrix}}_{\mathbf{A}_{\text{sys}}} \underbrace{\begin{bmatrix} \dot{\Gamma} \\ \Gamma \end{bmatrix}}_{\mathbf{X}} + \underbrace{\begin{bmatrix} \mathbf{B}^T \\ \mathbf{0} \end{bmatrix}}_{\mathbf{B}_{\text{sys}}} \underbrace{\begin{bmatrix} F_1 \\ F_2 \end{bmatrix}}_{\mathbf{U}}. \quad (11.11)$$

Vector \mathbf{X} is the state vector which contains the speeds and positions of the ball and drones and the system is represented by matrices \mathbf{A}_{sys} and \mathbf{B}_{sys} . Matrix \mathbf{M} includes damping factors which affects the angular movement of the rigid cables modeled by variables ϕ_1 and ϕ_2 . The damping factors avoid infinitive bouncing associated to ideal pendulum problem. Thus, this matrix is

$$\mathbf{M} = \begin{bmatrix} 0 & 0 \\ 0 & \begin{bmatrix} -\mu & 0 \\ 0 & -\mu \end{bmatrix} \end{bmatrix}. \quad (11.12)$$

11.4 Control Problem

The control objective of this system is to maintain the ball in the equilibrium point on the beam by means of regulating the drone altitudes. As previously mentioned, the system control is divided in two levels. The high-level control calculates the lift force to track an altitude reference that depends on the ball position and the altitude of the other drone, while the low-level control is dedicated to the drone stabilization. The low-level control is integrated in each drone and calculates the angular speeds of the four rotors to obtain a total lift force equal to the reference provided by the high-level control. The control problem is schemed in Fig. 11.2.

As observed in Fig. 11.2, it is considered that the stabilization controllers are feedback control strategies. The angular speeds, ω_i , of the four rotors are measured and the lift force, f_1 or f_2 , is estimated. Then the control loop is closed by obtaining the error between the reference of the lift force and the estimated lift force. This work focuses on the high-level control and it is assumed that the drone stabilization is perfectly controlled. In addition, the closed-loop scheme of the high-level control receives a reference vector, \mathbf{Y}_{ref} , to be tracked. The system output vector, \mathbf{Y} , comprises all the states that are measured. We assumed that all the states included in vector \mathbf{X} are measured, that is, the ball position, x , and speed, \dot{x} , the drone altitudes, z_1 and z_2 , and speeds, \dot{z}_1 and \dot{z}_2 , and the angles ϕ_1 and ϕ_2 and its angular speeds $\dot{\phi}_1$ and $\dot{\phi}_2$.

It is proposed herein to compare different optimal control techniques to evaluate the difficulties associated to this system. In particular, linear-quadratic regulator (LQR) and model predictive control (MPC) are developed. LQR allows to solve online optimization control problems with fast execution time and low computational effort, while MPC deals with physical constraints and predictions. Both control methodologies present interesting features for this control problem. For aerial application, short execution times with low computation effort are demanding but at the same time, handling physical constraints is required to avoid collisions and instable scenarios caused by disturbances. These controllers are detailed in the next subsections.

Moreover, controllers are implemented in CPUs on-board, and therefore discrete control laws are studied. The sampling time is a design parameter which has to be appropriately chosen. It is important to remark that in this control scenario composed

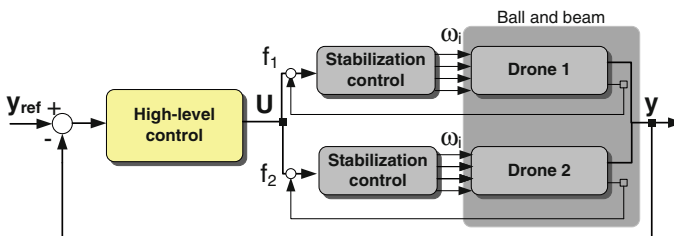


Fig. 11.2 Control scheme

by two control levels, each control level may present different sampling time. Specifically, the low-level control is performed faster than the high-level control for this application. Accounting for possible hardware limitations, a sampling time of 200 ms is set for this study. Note that the optimal controllers require the system model for design. The dynamic model presented in Eq. (11.11) is in continuous time and it has to be discretized for a 200-ms sampling time resulting in

$$\mathbf{X}(k+1) = \mathbf{A}_d \mathbf{X}(k) + \mathbf{B}_d \mathbf{U}(k), \quad (11.13)$$

$$\mathbf{Y}(k) = \mathbf{C}_d \mathbf{X}(k) + \mathbf{D}_d \mathbf{U}(k), \quad (11.14)$$

11.4.1 Linear-Quadratic Regulator

LQR is an optimal and feedback control law which minimizes every sampling time the following objective function J :

$$J = \sum_{k=0}^{\infty} (\mathbf{X}(k)^T \mathbf{Q} \mathbf{X}(k) + \mathbf{U}(k)^T \mathbf{R} \mathbf{U}(k)), \quad (11.15)$$

where $\mathbf{X}(k)$ and $\mathbf{U}(k)$ are the state and input vectors at instant k and matrices \mathbf{Q} and \mathbf{R} are weighting matrices. Given that the system is modeled by the discrete-time state-space model presented in Eqs. (11.13), (11.14), the analytical optimal control sequence results in

$$\mathbf{U}(k) = -\mathbf{F} (\mathbf{X}(k) - \mathbf{X}_{\text{ref}}(k)), \quad (11.16)$$

where

$$\mathbf{F} = (\mathbf{R} + \mathbf{B}_d^T \mathbf{P} \mathbf{B}_d)^{-1} \mathbf{B}_d^T \mathbf{P} \mathbf{A}_d, \quad (11.17)$$

being \mathbf{P} the unique positive definite solution to the discrete-time algebraic Riccati equation (DARE):

$$\mathbf{P} = \mathbf{Q} + \mathbf{A}_d^T (\mathbf{P} - \mathbf{P} \mathbf{B}_d (\mathbf{R} + \mathbf{B}_d^T \mathbf{P} \mathbf{B}_d)^{-1} \mathbf{B}_d^T \mathbf{P}) \mathbf{A}_d. \quad (11.18)$$

11.4.2 Model Predictive Control

Model predictive control has been successfully applied to many industrial processes [11] and drone applications such as [9]. The controller calculates the optimal control action taking future predictions and constraints into account. The optimization is

repeated each sampling time with a moving horizon. MPC is generally formulated with state-space models as

$$\min_{\varepsilon(k+m+1), \mathbf{U}(k+m)} \sum_{m=1}^{N-1} \varepsilon(k+m+1)^T \mathbf{Q} \varepsilon(k+m+1) + \mathbf{U}(k+m)^T \mathbf{R} \mathbf{U}(k+m), \quad (11.19)$$

where

$$\varepsilon(k+m+1) = \mathbf{X}(k+m+1) - \mathbf{X}_{\text{ref}}(k+m+1), \quad (11.20)$$

being N the prediction horizon, \mathbf{Q} and \mathbf{R} the weighting matrices. The error ε is defined as the difference between the state vector \mathbf{X} and the reference \mathbf{X}_{ref} since the output vector \mathbf{Y} is equal to the state vector. The optimization problem is subject to the system dynamic model presented in Eq. (11.14) and the following system constraints:

$$\underline{\mathbf{X}} \leq \mathbf{X}(k) \leq \overline{\mathbf{X}}, \quad \underline{\mathbf{U}} \leq \mathbf{U}(k) \leq \overline{\mathbf{U}}, \quad \underline{\mathbf{Y}} \leq \mathbf{Y}(k) \leq \overline{\mathbf{Y}}. \quad (11.21)$$

The variable vectors denoted with an over line contain the upper limits, while the variable vectors with an under line contains the lower limits. Particularly, the constraints included in this problem are the beam length that limits the ball position, x , and maximum and minimum values for the lift forces, f_1 and f_2 , imposed by rotor physical constraints. Values of 30 and -35 N are chosen for the upper and lower lift force bounds, respectively.

11.5 Simulation Results

This section is dedicated to compare and discuss the simulation results for LQR and MPC. First, preliminary simulations are performed in order to tune the weighting matrices, \mathbf{Q} and \mathbf{R} , for the controllers with the objective to achieve fast performance with nonaggressive control actions. The best values for both matrices and both controllers correspond to

$$\begin{aligned} \mathbf{Q} &= \mathbf{I}_{n_x \times n_x}, \\ \mathbf{R} &= 0.004 \mathbf{I}_{n_u \times n_u}. \end{aligned} \quad (11.22)$$

Furthermore, for the MPC the prediction horizon, N , is another design parameter. The prediction horizon directly influences on the computational demand, that is, the higher the prediction horizon the higher the computational demand is. Otherwise, MPC and LQR have the same performance for a sufficiently higher N . Then it is achieved a tradeoff with a prediction horizon set at 12 samplings.

Figure 11.3 compares LQR and constrained MPC simulated for initial drones altitudes equal to 4.5 and 3.5 m for drone 1 and drone 2.

Figure 11.3a shows the performance of the ball position for LQR in red, for MPC in blue, and bounds in magenta dash lines. Note that both controllers are able to

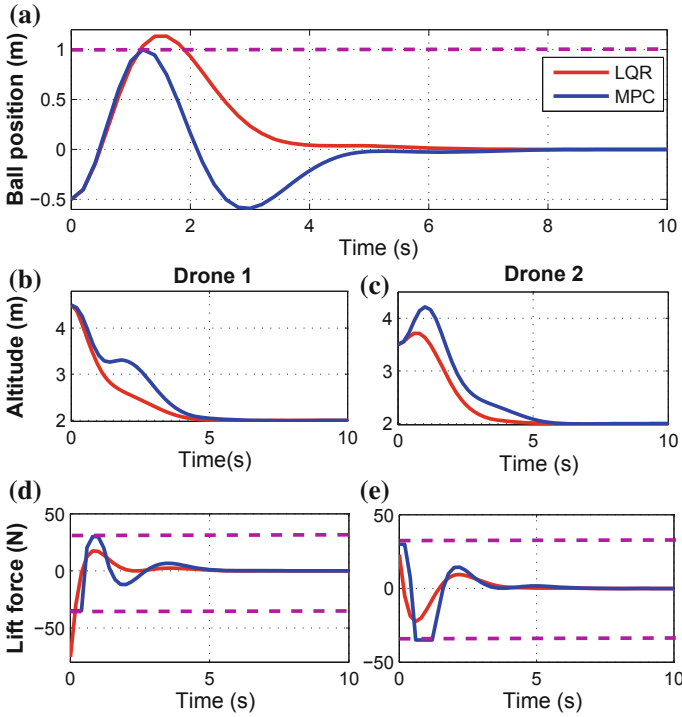


Fig. 11.3 Comparison of LQR and MPC controllers on simulations

regulated the ball position and stabilized the ball in the equilibrium point set at $x = 0$. However, LQR violates the constraints imposed by the beam length and at time 1.75 s the ball falls from the beam. Figure 11.3b, c show the drone altitudes, z_1 and z_2 . The altitude references for both altitudes are maintained constant at 2 m during all the simulation. LQR and MPC appropriately regulate the altitudes. As seen in the figures, MPC is slightly slower due to the constraints. Figure 11.3d, e present the simulation results for the drone lift forces. MPC performs inside the bounded region for all the simulation, while LQR violates the constraints at the beginning of the simulation imposed to the drone 1 lift force.

In order to test in more detail the LQR, several simulations are performed and presented in Fig. 11.4. The weighting matrices are modified and the lift forces are saturated to the maximum and minimum values only for the case of the LQR. In Fig. 11.4a, the ball position performances are presented where red, magenta, and green dash lines correspond to the LQR performance with a weighting matrix \mathbf{R} equal to $0.008 \mathbf{I}_{n_u \times n_u}$, $0.004 \mathbf{I}_{n_u \times n_u}$, and $0.001 \mathbf{I}_{n_u \times n_u}$, respectively. Matrix \mathbf{Q} is kept constant and equal to the identity matrix. Moreover, red, magenta, and green solid lines are the LQR performances for the previous weighting matrices including saturations on the maximum and minimum values of the lift forces. As observed in the figure, only

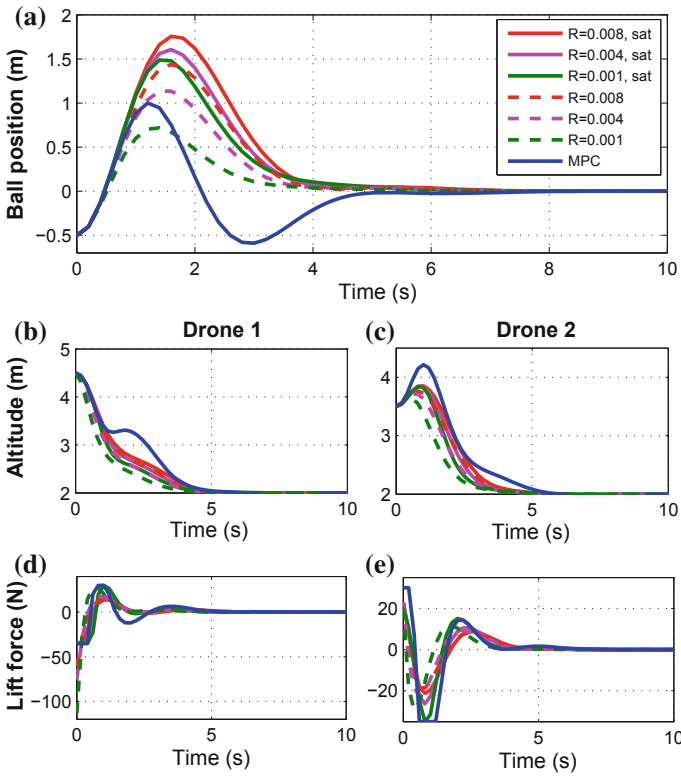


Fig. 11.4 Saturated and nonsaturated LQR with different weighting matrices

the LQR performance for a weighting matrix \mathbf{R} of $0.004 \mathbf{I}_{n_u \times n_u}$ without saturation avoids the ball to fall. However, after studying Fig. 11.4d, e, LQR with a weighting matrix \mathbf{R} of $0.004 \mathbf{I}_{n_u \times n_u}$ requires to implement lift forces for drone 1 out of bounds. Therefore, it is demonstrated that even though LQR is a good candidate to regulate this system due to its fast execution time, the system performance under the bounds is not guaranteed.

11.5.1 Disturbances

LQR and MPC are tested under disturbances. To that end, simulations are performed with disturbances in the ball position and drone altitudes. In addition, changes in the references are also included in the simulation. The results are compared in Fig. 11.5. The initial conditions are the same as presented in the previous simulations. The ball position is modified to a value of x equal to 0.5 m at time 15 s as seen in Fig. 11.5a. The altitude of drone 1 is modified to values of z_1 equal to 1.2 and 2.7 at time 20

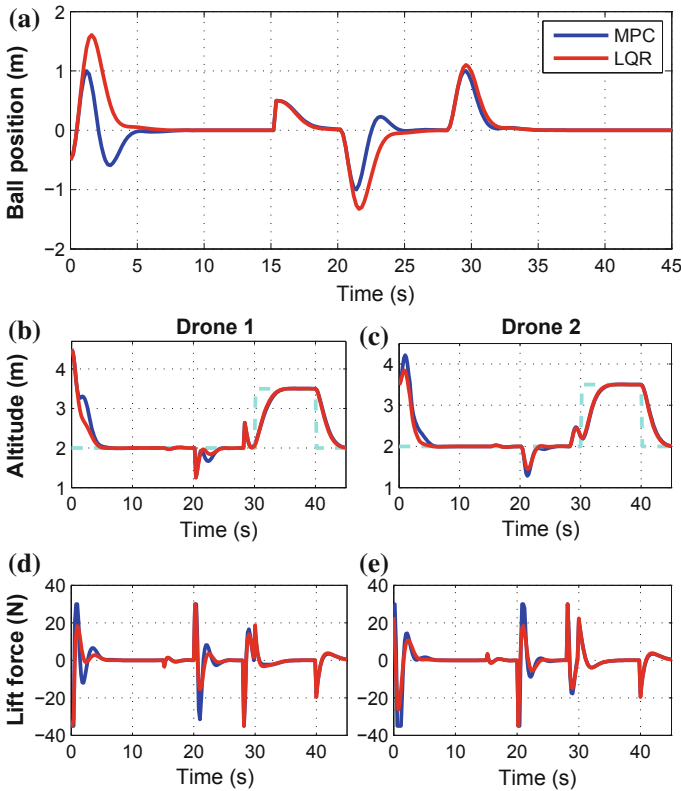


Fig. 11.5 LQR and MPC with disturbances

and 28 s as shown in Fig. 11.5b. The altitude references are at the beginning of the simulation equal to 2 m and simultaneously change to a value of 3.5 m at time 30 s and again to 2 m at time 40 s. The references are shown in Fig. 11.5b, c as cyan dash lines. In Fig. 11.5, blue solid lines correspond to the MPC, while red solid lines correspond to the LQR with lift forces saturated. As mentioned in the previous subsection, LQR is not able to perform under the bounded region and the ball falls from the beam at times 1.75, 21, and 29 s. MPC performs under bounds during all the simulation and faster than the saturated LQR as seen in Fig. 11.5a at time 21 s. For both cases, the altitude references are perfectly tracked with a similar rise time.

11.5.2 Execution Times

Finally, executions times are obtained during simulation to test the suitability of the real-time implementation of MPC for this application. Note that the controllers

Table 11.2 Execution times

Controller	Average (ms)	Maximum (ms)	Minimum (ms)
LQR	0.01225	0.0338	0.00798
MPC	5.8	6.8	4.9

are simulated using MATLAB in PC with i7-260M CPU under 64 bits Windows platform. Table 11.2 lists the average, maximum and minimum execution times for LQR and MPC. The average time for LQR is 0.01225 ms, while MPC is 5.8 ms. Both average times are far from 200 ms which is the sampling time, and thus a robust online implementation is feasible. Note that the hardware on-board with a real-time software may execute the control laws faster. Due to the fast execution times, the sampling time could be reduced in order to obtain better performance in terms of response speed. Then the prediction horizon for MPC needs to be recalculated for the new sampling time. The prediction horizon depends on the system rise time. Therefore, if the sampling time is reduced, the prediction horizon increases. Also, the MPC execute time increases with the prediction horizon. The reduction of the sampling time needs to be carefully studied.

11.6 Conclusion

In the present chapter we have presented, modeled, and analyzed a novel application of multiagent drone system dealing with load carrying. The main problem related to this application comes from the mechanical links between the load and the drones which carry it. This system was proposed to be modeled as a ball and beam lifted by to drones and a mathematical model based on first principles was developed. Under the assumption that all the states are measured, LQR and MPC controllers have been analyzed by simulations. LQR presented very fast execution times but it is not guaranteed a performance in the bounded area, that is, that the ball does not fall from the beam as seen in different simulations. On the other hand, MPC deals with constraints and regulates the ball position without violating the constraints imposed in the problem. Moreover, execution times are short enough for this application to guarantee the online implementation.

This work has presented only a preliminary study with centralized controllers which have the knowledge of all the states variables. In future work, we aim at considering a more general setup with a higher number of drones and where only part of the states are locally measured on each drone. In order to reduce the execution time of the MPC and to make the problem scalable in an easy way, distributed MPC strategies should be studied. In addition, a uncertainties and aerodynamic disturbances need to be included in future studies.

References

1. P. McBride, B. Orwell, The application of unmanned aircraft systems in domestic surveillance operations. *J. Air Law Comm.* Summer **74**(3), 627–628 (2009)
2. D. Zorbas, T. Razafindralambo, D.P. Luigi, F. Guerriero, Energy efficient mobile target tracking using flying drones. *Proc. Comput. Sci.* **19**, 80–87 (2013)
3. H. Chen, X. Wang, Y. Li, A survey of autonomous control for uav, in *International Conference on Artificial Intelligence and Computational Intelligence*, vol. 2, 2009, pp. 267–271
4. G. Hoffmann, H. Huang, S. Waslander, C. Tomlin, Quadrotor helicopter flight dynamics and control: theory and experiment, in *The AIAA Guidance, Navigation, and Control Conference*, 2007
5. S. Bouabdallah, M. Becker, R. Siegwart, Autonomous miniature flying robots: coming soon—research, development, and results. *IEEE Robot. Autom. Mag.* **14**(3), 88–98 (2007)
6. K. Alexis, G. Nikolakopoulos, A. Tzes, Switching model predictive attitude control for a quadrotor helicopter subject to atmospheric disturbances. *Control Eng. Pract.* **19**, 1195–1207 (2011)
7. S. Bouabdallah, A. Noth, R. Siegwart, PID-LQ control techniques applied to an indoor micro quadrotor, in *The IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004, pp. 2451–2456
8. S. Waslander, G. Hoffmann, J. Jang, C. Tomlin, Multi-agent quadrotor test bed control design: integral sliding mode vs. reinforcement learning, in *The IEEE/RSJ International Conference on Intelligent Robotics and Systems*, 2005, pp. 468–473
9. G.V. Raffo, M.G. Ortega, F.R. Rubio, An integral predictive/nonlinear \mathcal{H}_∞ control structure for a quadrotor helicopter. *Automatica* **46**, 29–39 (2010)
10. S. Bouabdallah, Design and control of quadrotors with application to autonomous flying. Ph.D. dissertation, EPFL STI School of Engineering, Lausanne, Switzerland, 2007
11. E. Camacho, C. Bordons, *Model Predictive Control* (Springer, London, 2004)