

Minimum-time optimal control for a quadcopter trajectory through gates with arbitrary pose

Oscar E. Ramos

Department of Mechatronics Engineering, Universidad de Ingenieria y Tecnologia - UTEC, Lima, Peru

Abstract—Drone racing has become popular in recent years, both in the manned and unmanned versions. However, currently manned races show more level of acrobatic behaviors through more challenging scenarios. Autonomous quadcopters are still limited to technological proprioceptive and exteroceptive challenges, as well as algorithmic limitations. This paper proposes an approach for unmanned quadcopters to maneuver through gates with arbitrary poses solving a minimum-time optimal control problem. The results show, in a dynamic simulator, that the quadcopter is able to successfully circumvent the nuances of the environment passing through the gates in minimum time satisfying the motion constraints and planning the motion ahead.

Index Terms—Minimum-time optimal control, Quadcopter, Autonomous robot, Trajectory generation

I. INTRODUCTION

The industry of quadcopters has greatly increased in recent years, and nowadays it is common to see drone applications at the industrial, service and personal levels [1]. However, most of the currently used quadcopters are teleoperated, limiting their ability to perform complex trajectories in the air. Moreover, to be able to truly deploy this technology in cohabitation with humans, teleoperation problems should be avoided and a higher level of autonomy is needed [2].

In spite of this development, aggressive and fast trajectories requiring rapid and well planned motions are still a challenging problem. Drone racing competitions have also recently become popular, being the manned versions the ones with more challenging environments but requiring experience of the remote pilot [3]. Unmanned races require autonomy to generate the fastest possible trajectory through a series of waypoints in the track, but is currently hard to achieve. The more autonomy, the fastest and more aggressive the motions. Eventually, an autonomous optimal trajectory in minimum time would be able to beat their manned counterpart [4].

In this paper, a framework for solving the problem of a quadcopter racing through several gates with arbitrary pose is proposed. The main contribution of the paper consists in formulating a minimum-time optimal control approach as an alternative solution to this problem. This paper is organized as follows. Section II develops the background related to the dynamics of the quadcopter that will be used in this work. Section III presents the proposed approach to optimal control as a solution to the minimum-time problem, and Section IV shows the experimental results in a dynamic simulation, as well as their discussion.

II. MODEL OF THE QUADCOPTER

This section summarizes the nonlinear dynamics of the quadcopter, as well as some additional constraints that will be used throughout this paper.

A. Rigid Body Representation of a Quadcopter

Let the Cartesian position of the center of mass of a quadcopter be represented as $\mathbf{x}_p = (x, y, z) \in \mathbb{R}^3$, with respect to an inertial frame, and its orientation as \mathbf{x}_o . Using this notation, the pose of the robot will be given by $\mathbf{x}_{po} = (\mathbf{x}_p, \mathbf{x}_o)$. The orientation can be, in general, represented using any parameterization, but the roll, pitch, yaw angles will be used here, such that $\mathbf{x}_o = (\phi_r, \phi_p, \phi_y) \in SO(3)$. Let the twist of the robot be represented as $\xi = (\mathbf{v}, \boldsymbol{\omega})$, where $\mathbf{v} = (v_x, v_y, v_z)$ is its linear velocity, and $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$ its angular velocity, both in its local frame, also referred to as the body frame. The relation between the rate of change of the orientation and the angular velocity will be given by $\dot{\mathbf{x}}_o = E(\mathbf{x}_o)\boldsymbol{\omega}$, where $E(\mathbf{x}_o) \in \mathbb{R}^{3 \times 3}$ can be derived analyzing the successive rotations that generate the parameterization. Using this expression, the relation between the body frame accelerations and both the angular velocity and angular acceleration of the robot is $\ddot{\mathbf{x}}_o = E(\mathbf{x}_o)^{-1}\dot{\boldsymbol{\omega}} + E(\mathbf{x}_o)^{-1}\boldsymbol{\omega}$.

B. Geometric and Dynamic Parameters

For simplicity, this work will assume that the structure of the aerial robot is both rigid and symmetric, and that its center of mass is located at the geometric center of the robot, where the local robot frame is located. The mass of the robot will be considered to be m . Moreover, it will be assumed that the arms of the robot are aligned with the x and y axes of the local frame. Under this conditions, the inertia matrix \mathcal{I} will be a diagonal matrix with its elements given by I_{xx} , I_{yy} and I_{zz} . Furthermore, due to symmetry, the inertia in x and y will be the same $I_{xx} = I_{yy}$. Another assumption will be that the propellers are rigid, so that the deformation of the propeller blades generated by the material and the high velocities that the rotors generate is negligible.

The rotors will be modeled considering that the lift f_j , generated by the rotation of the j^{th} rotor, is proportional to the square of the respective angular speed ω_j and is given by $f_j = k\omega_j^2$, where k is a constant that depends on the rotor structure and is usually experimentally obtained. Thus, the magnitude of the total thrust will be $f_z = k \sum_{j=1}^4 \omega_j^2$, assuming that the four rotors have the same constant k . Its direction will be parallel to the z axis of the local frame located

on the robot body, and pointing upwards. Also, the relation between the torque of the rotor τ_j and the angular components (velocity and acceleration) will be given by $\mathcal{I}_r \dot{\omega}_j = \tau_j - b\omega_j^2$, where \mathcal{I}_r is the rotor moment of inertia, and b is a constant that considers the drag effect. Provided that $\dot{\omega}_j$ is considered small, its effect can be neglected and the torque due to drag will be equal to the torque about the z axis.

C. Dynamic Model of a Quadcopter

Setting the aforementioned parameters and assumptions, considering that the orientation of the robot with respect to the inertial frame is represented by the rotation matrix ${}^I R_B \in SO(3)$, the translational dynamics of the robot in the inertial frame is

$$\ddot{\mathbf{x}}_p = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + \frac{1}{m} {}^I R_B \begin{bmatrix} 0 \\ 0 \\ f_1 + f_2 + f_3 + f_4 \end{bmatrix} + \frac{1}{m} \boldsymbol{\delta}, \quad (1)$$

where g represents the magnitude of the gravitational acceleration, and $\boldsymbol{\delta}$ accomplishes for other aerodynamic effects such as the drag force that will be further discussed in section II-D. The relation between the rotation matrix ${}^I R_B$ and the roll, pitch, yaw angles used to represent the orientation of the robot can be directly obtained, since the latter one is a well-known parameterization of the former one.

The dynamics of the rotational motion of the quadcopter can be described in the body frame as $\mathcal{I} \dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathcal{I} \boldsymbol{\omega} + \boldsymbol{\tau}_g = \boldsymbol{\tau}$, where $\boldsymbol{\tau}_g$ represents the gyroscopic forces, and $\boldsymbol{\tau}$ represents the effect of the net external force. The gyroscopic forces are $\boldsymbol{\tau}_g = \mathcal{I}_r \boldsymbol{\omega} \times \boldsymbol{\omega}_r$, where $\boldsymbol{\omega}_r = (\omega_1 - \omega_2 + \omega_3 - \omega_4) \mathbf{e}_z = \omega_r \mathbf{e}_z$ represents the contribution of the angular velocities ω_j created by each rotor. Each ω_j is parallel to the local z direction of the robot, and \mathbf{e}_z represents the unit vector in this direction. Using the previous expressions, we get

$$\dot{\boldsymbol{\omega}} = \mathcal{I}^{-1} (-\boldsymbol{\omega} \times \mathcal{I} \boldsymbol{\omega} - \mathcal{I}_r \boldsymbol{\omega} \times \boldsymbol{\omega}_r + \boldsymbol{\tau}), \quad (2)$$

which represents the second order dynamics related to the rotational motion of the aerial robot. Further details of this derivation can be found in [5]. Both (1) and (2) fully describe the dynamics of the quadcopter.

D. Other aerodynamic effects

The so-called aerodynamic effects originate due to the motion of the robot in the air and are usually difficult to model. When the quadcopter is flying, there exists a drag force that is generated by the air resistance. To model these effects, a diagonal matrix can be used to show the effect that the linear velocities have on the generated force as

$$\boldsymbol{\delta} = \text{diag}(D_x, D_y, D_z) \dot{\mathbf{x}}_p, \quad (3)$$

where D_i are the drag force coefficients corresponding to the x , y and z axis. Other effects can be included, such as the effect that the angle of attack, the blade flapping, or the airflow disruptions have on the thrust. A study of these effects, among others can be found in [5], [6]. However, these latter effects will not be considered in the model used in this work.

E. State-space Model

Let the control vector be $\mathbf{u} = (f_t, \tau_r, \tau_p, \tau_y) \in \mathbb{R}^4$, where $f_t = f_1 + f_2 + f_3 + f_4$ is the resulting thrust of all the rotors and τ_i is the difference of thrust between the rotors on the axis associated with i . That is, τ_r considers the x axis, τ_p considers the y axis, and τ_y the vertical z axis. More concretely, the relation between this input vector and the angular velocity generated by each of the rotors is

$$f_t = k(\omega_1 + \omega_2 + \omega_3 + \omega_4) \quad (4)$$

$$\tau_r = lk(\omega_4^2 - \omega_2^2) \quad (5)$$

$$\tau_p = lk(\omega_3^2 - \omega_1^2) \quad (6)$$

$$\tau_y = b(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2), \quad (7)$$

where l is the distance from the center of the robot to each of the rotors, since symmetry of the quadcopter has been assumed (section II-B). Let the state vector of the robot be $\mathbf{x} = (\mathbf{x}_p, \dot{\mathbf{x}}_p, \mathbf{x}_o, \dot{\mathbf{x}}_o) \in \mathbb{R}^{12}$, such that the nonlinear dynamics can be represented as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}). \quad (8)$$

It is possible that (8) depends on t , but since the dynamics of the aerial robot does not have a direct dependency on t , this variable is not considered. Expanding the terms shown in (1) and (2), it can be shown that the terms in (8) are

$$\begin{aligned} \ddot{x} &= \frac{f_t}{m} (s_1 s_3 + c_1 c_3 s_2) - \frac{D_x}{m} \dot{x} \\ \ddot{y} &= \frac{f_t}{m} (c_1 s_3 s_2 - c_3 s_1) - \frac{D_y}{m} \dot{y} \\ \ddot{z} &= -g + \frac{f_t}{m} c_1 c_2 - \frac{D_z}{m} \dot{z} \\ \dot{\phi}_r &= \dot{\phi}_r + \dot{\phi}_y c_1 t_2 + \dot{\phi}_p s_1 t_2 \\ \dot{\phi}_p &= \dot{\phi}_p c_1 - \dot{\phi}_y s_1 \\ \dot{\phi}_y &= \dot{\phi}_y \frac{c_1}{c_2} + \dot{\phi}_p \frac{s_1}{c_2} \\ \ddot{\phi}_r &= \frac{I_{yy} - I_{zz}}{I_{xx}} \dot{\phi}_y \dot{\phi}_p - \frac{I_r}{I_{xx}} \dot{\phi}_p \omega_r + \frac{\tau_r}{I_{xx}} \\ \ddot{\phi}_p &= \frac{I_{zz} - I_{xx}}{I_{yy}} \dot{\phi}_r \dot{\phi}_y + \frac{I_r}{I_{yy}} \dot{\phi}_r \omega_r + \frac{\tau_p}{I_{yy}} \\ \ddot{\phi}_y &= \frac{I_{xx} - I_{yy}}{I_{zz}} \dot{\phi}_r \dot{\phi}_p + \frac{\tau_y}{I_{zz}} \end{aligned}$$

where $c_i = \cos(\phi_i)$, $s_i = \sin(\phi_i)$, $t_i = \tan(\phi_i)$, $i = 1$ refers to ϕ_r , $i = 2$ refers to ϕ_p , and $i = 3$ refers to ϕ_y .

III. TRAJECTORY GENERATION FOR THE QUADCOPTER

The problem that needs to be solved consists in allowing the robot to pass through different waypoints in minimum time. Each waypoint is a ‘gate’ that resembles an ellipse with additional geometric constraints, which impose bounds in the robot position and orientation. Each of the trajectories between a waypoint will be called a ‘phase’ of the motion and, thus, the motion will be decomposed into P phases.

A. Minimum time optimal control problem

A multiple-phase optimal control problem will be formulated to generate the motion of the quadcopter. Since the system is composed of P phases, let each phase be represented as $p \in \{1, \dots, P\}$. For a given p phase, the problem consists in finding the state $\mathbf{x}^{(p)}(t)$, control, $\mathbf{u}^{(p)}(t)$, initial time,

$t_0^{(p)}$, and final time $t_f^{(p)}$. Provided that the goal of the robot motion consists in allowing the robot to pass through different waypoints in a given order as fast as possible, the nominal optimal control problem will be formulated as

$$\min \sum_{p=1}^P \int_{t_0^{(p)}}^{t_f^{(p)}} dt \quad (9)$$

s.t.

$$\dot{\mathbf{x}}^{(p)} = \mathbf{f}^{(p)}(\mathbf{x}^{(p)}, \mathbf{u}^{(p)}), \quad p = 1, \dots, P$$

where $\mathbf{f}^{(p)}$ represents the nonlinear dynamic constraints (8) that the robot imposes in each phase. Expanding the sum and the integral of the cost function, it can be shown that they simplify to the final time t_f ; that is, the problem has a fixed initial time but the final time t_f is free. If energy was also to be minimized, an approach similar to [7] could be used.

B. Constraints on the control and the states

The system in (9) needs to be bounded to withstand the control and state limits of the quadcopter, and to generate feasible motion. The bounds for the control can be stated as

$$\underline{u}_i \leq u_i \leq \bar{u}_i, \quad (10)$$

where \underline{u}_i and \bar{u}_i are the lower and upper bounds of the i^{th} component of the control vector, respectively. The lower bound on $u_1 = f_t$ in (4), which is the thrust, is zero, provided that we consider the robot to have fixed rotors and propellers. Its upper bound is approximated as $4k\omega_{max}^2$, considering that ω_{max} is the maximum speed that each rotor can generate under normal circumstances. The upper and lower bounds for $u_2 = \tau_r$ and $u_3 = \tau_p$, in (5) and (6) respectively, are $\pm 4kl\omega_{max}^2$. The last control input $u_4 = \tau_y$, given by (7), is lower and upper bounded using $\pm 2b\omega_{max}^2$.

The bounds on the states are mathematically formulated in a similar way, leading to

$$\underline{x}_i \leq x_i \leq \bar{x}_i, \quad (11)$$

where \underline{x}_i and \bar{x}_i are the lower and upper bounds of the i^{th} component of the state, respectively. These bounds are experimentally obtained and may vary according to the intended motion for the robot. The initial conditions for the aerial robot force the robot to be at rest at the very beginning of the first phase; that is, there will be no linear or angular velocities, and the robot will lie at the start point of the first gate. The rest of the phases consider that the position of the quadcopter starts at the center of the respective gate, with a small tolerance of ε m, and that its orientation is set to comply with the geometric constraints of the respective gate. There are no constraints on the linear or angular velocities, so that the optimizer is free to set them according to the needs.

C. Conic velocity constraints at the gates

The optimal control problem shown in (9), considering the constraints imposed by (10) and (11), allows the robot to reach each gate with a given pose in minimum time. However, this

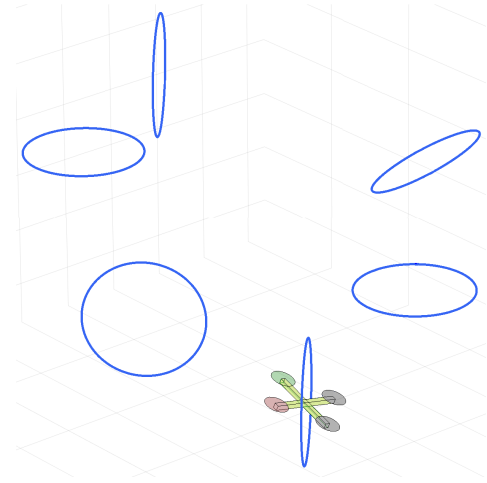


Fig. 1. Environment of the robot showing the gates that it should traverse in a given direction. The simulated robot is shown at the bottom.

problem does not impose any restriction that forces the robot to actually pass through the gate. That is, once the robot reaches the desired pose at the gate, it might detour from it, if that motion locally minimizes time.

To enforce the robot pass through the gate, a velocity cone is imposed on the motion once it reaches the desired pose at the respective gate. This cone is generated using $\mathbf{c}_i = \hat{\mathbf{v}}_i \cdot \mathbf{n}_i$, where $\hat{\mathbf{v}}_i$ is the normalized velocity that the robot has when reaching gate i , and \mathbf{n}_i is the normal to the gate pointing in the desired direction of motion. Then, the resulting \mathbf{c}_i is bounded to some limits to generate the cone that will enforce the velocity to lie within this limits, allowing the robot to locally have the desired direction. This constraint is imposed only when the robot arrives to the gate.

IV. EXPERIMENTAL SETUP

For the experiments, the nonlinear dynamics of the quadcopter have been used within an environment created in Python. In this environment, all the gates lie at approximately the same distance, but their orientations are arbitrary, as shown in Fig. 1. The robot has to then pass through all the gates, without collisions, and in minimum time, since the environment simulates the so-called drone races. The desired position of the quadcopter is assumed to be at the center of each gate, and the initial control input allows the robot to hover. The constraints that are imposed on the control inputs depend on the chosen parameters for the model and in this case they are $u_1 \in [0, 45]$, $u_2 = u_3 \in [-6, 6]$ and $u_4 \in [-2.5, 2.5]$. The states are bounded so that each element in \mathbf{x}_p and $\dot{\mathbf{x}}_p$ is in the interval $[-15, 15]$. The roll and pitch angles are bounded by $\pm 90^\circ$, the yaw angle by $\pm 180^\circ$, and the angular rate $\dot{\mathbf{x}}_o$ is bounded by $\pm 0.9^\circ/\text{s}$.

The minimum-time optimal control problem formulated in (9), considering the constraints given by (10) and (11), was solved using a numeric approach in CasADi [8] within Python. Additional constraints were imposed on the velocities of the robot when it reached the desired gate, as stated in

section III-C, using 0.9 as an experimental threshold for the cone. The optimization problem was solved in around three minutes considering the 6 gates shown in Fig. 1, each of which adds a geometric constraint to the robot motion as a new target position and orientation. Due to the computational complexity, this planner cannot be used in real time; the whole trajectory of the robot to be planned ahead.

The generated motion obtained using the proposed approach is depicted in Fig. 2, which shows the positions x_p of the robot as well as the linear velocities \dot{x}_p for four seconds, which is the duration of the motion. Each color shows a phase of the motion and the point where the color change occurs depicts the presence of a gate. As shown, the position changes are smooth, as well as the linear velocity in x and y . There are some fast changes in z , which could be further constrained in the formulation of the optimal control problem. However, all the velocities lie within the restricted limits.

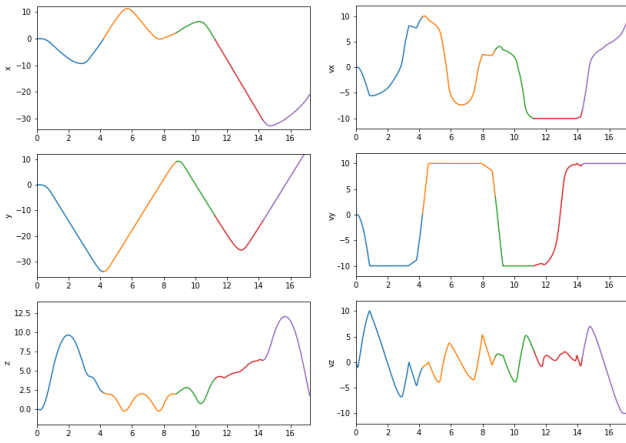


Fig. 2. Generated cartesian positions (in meters) and linear velocities (in m/s) of the quadcopter. The horizontal axis shows the time in seconds.

Fig. 3 shows the generated control inputs, as specified by (4), (5), (6), (7). Some of the components have abrupt changes due to the fast velocity and acceleration profile that is needed to allow for the maneuvers of the quadcopter satisfying the dynamics of the robot. The color of the segment indicates the phase of the motion. Fig. 4 shows the generated motion as well as the velocity constraining cones at the gates. The robot passes through the gates minimizing the total time.

V. CONCLUSION

This paper has presented a solution to the trajectory generation problem of a quadcopter moving through arbitrary gates using a minimum-time optimal control approach. The optimization takes into account the limits of the quadcopter so as to generate feasible motion. In order to enforce the robot motion through the constraining gates, the proposed approach introduces cones that limit the velocity vector as inequality constraints. The results show that the quadcopter has been able to properly move through the constrained environment. Future work involves reducing the computation time as well as adding other constraints to the motion.

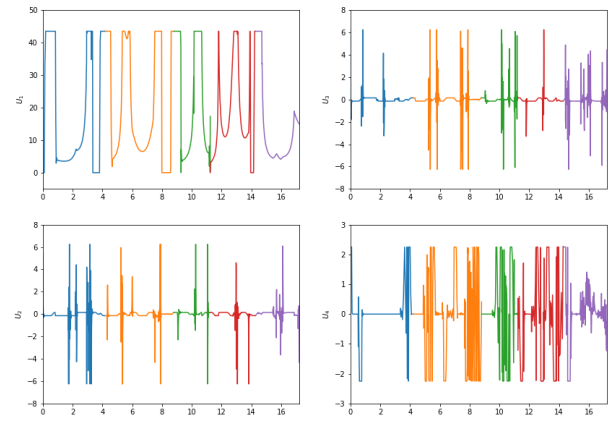


Fig. 3. Control outputs applied to the quadcopter. The horizontal axis shows the time in seconds

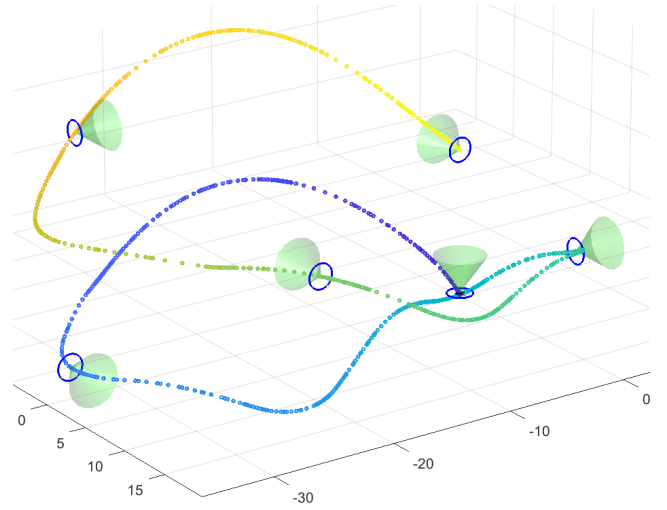


Fig. 4. Generated trajectory for the quadcopter. The velocity constraining cones are shown for each of the gates.

REFERENCES

- [1] J. Shahmoradi, E. Talebi, P. Roghanchi, and M. Hassanalani, "A comprehensive review of applications of drone technology in the mining industry," *Drones*, vol. 4, no. 3, p. 34, 2020.
- [2] S. Watkins, J. Burry, A. Mohamed, M. Marino, S. Prudden, A. Fisher, N. Kloet, T. Jakobi, and R. Clothier, "Ten questions concerning the use of drones in urban environments," *Building and Environment*, 2020.
- [3] A. Loquercio, E. Kaufmann, R. Ranftl, A. Dosovitskiy, V. Koltun, and D. Scaramuzza, "Deep drone racing: From simulation to reality with domain randomization," *IEEE Transactions on Robotics*, pp. 1–14, 2019.
- [4] J. Delmerico, T. Cieslewski, H. Rebecq, M. Faessler, and D. Scaramuzza, "Are we ready for autonomous drone racing? the uzh-fpv drone racing dataset," in *2019 Int. Conf. on Robotics and Automation*. IEEE, 2019.
- [5] G. Hoffmann, H. Huang, S. Waslander, and C. Tomlin, "Quadrotor helicopter flight dynamics and control: Theory and experiment," in *AIAA guidance, navigation and control conference and exhibit*, 2007, p. 6461.
- [6] H. Huang, G. M. Hoffmann, S. L. Waslander, and C. J. Tomlin, "Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering," in *IEEE Int. Conf. on Robotics and Automation*, 2009.
- [7] O. E. Ramos, "Optimal control for time and energy minimization in the trajectory generation of a mobile robot," in *2019 IEEE XXVI Int. Conf. on Electronics, Electrical Engineering and Computing*, 2019.
- [8] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, 2019.