Constrained Model Predictive Control with Integral Action for Twin Rotor MIMO Systems

Kelechi U. Ebirim
School of Engineering
University of Leicester
University Road, Leicester
LE1 7RH UK
Email: kue2@leicester.ac.uk

Andrea Lecchini-Visintini *†
School of Electronics and Computer Science
University of Southampton
University Road, Southampton
SO17 1BJ UK
Email: alv1e22@soton.ac.uk

Matteo Rubagotti
Department of Robotics and Mechatronics,
School of Engineering and Digital Sciences,
Nazarbayev University, Astana, 010000
Kazakhstan
Email: matteo.rubagotti@nu.edu.kz

Emmanuel Prempain
School of Engineering
University of Leicester
University Road, Leicester
LE1 7RH UK
Email: ep26@leicester.ac.uk

This paper describes the design and successful implementation of a constrained model predictive controller with integral action for the control of a Twin Rotor MIMO System (TRMS). The integral action guarantees zero steady-state error in set-point tracking with robustness towards perturbations of the nominal system parameters. In addition to saturation constraints on the input variables, hard constraints are imposed on the controlled output variables, i.e. on pitch and yaw angular positions, to avoid collisions with obstacles. The model predictive controller was designed using a high-fidelity nonlinear model of the TRMS developed in previous work. As an intermediate step, exact linearised models of the TRMS are obtained and their closed-form expressions are reported. The controller was tested experimentally, also showing its effectiveness in ensuring actual collision avoidance by the TRMS when physical obstacles were present.

1 INTRODUCTION

The control of the Twin Rotor MIMO System (TRMS) has proven challenging, due to the simultaneous presence of strongly nonlinear and coupled dynamics, and to the fact that actuation torques are obtained through the aerodynamics of rotor blades, rather than directly (as it happens, for instance, in robot manipulators). For these reasons, the control of the TRMS is often employed as test bed for control strategies, see e.g. [1–8]. In this work, we report on the implementation of model predictive control (MPC) for a TRMS.

In MPC [9, 10], a model of the system is used to optimise the trajectories of states and inputs, by solving a constrained optimal control problem over a rolling prediction horizon. The origins of MPC date back to the 1960s, when its first formulations were defined [11]. Thanks to its ability to handle multivariable systems, and later to its ability to deal with operational constraints and nonlinearities, MPC was first employed in the process control industry [12], and later extended to other areas, surveyed for example in [13, 14]. High computational power is a key MPC requirement in order to determine optimal input sequences. As a consequence, the availability of high-performance optimization solvers and faster microprocessors has extended the use of MPC from process control to applications requiring higher sampling frequencies, such as automotive [15], robotics [16], power electronics [17] and building control [18].

From the theoretical point of view, MPC can be designed for both linear and nonlinear systems [19], includ-

[*] Address all correspondence related to ASME style format and figures to this author.

[†] Address all correspondence for other issues to this author.

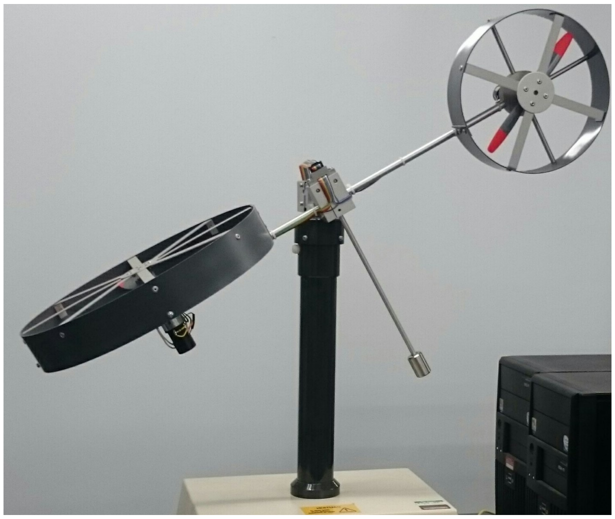

Fig. 1. The TRMS used in this paper's experiments

ing hybrid systems [20]. The presence of model inaccuracies and external disturbances can be accounted for by relying on the inherent robustness of MPC [21], or devising approaches that explicitly account for robust constraint satisfaction, using either a deterministic framework [22] or a stochastic one [23,24]. Specific approaches were also defined to guarantee recursive feasibility and closed-loop stability under inexact numerical optimization [25]. In case a model of the system is not available, data-driven approaches to MPC have been recently proposed [26], following the general data-driven framework used also for other control methods [27–29]. An approach that is in some cases advantageous compared to solving a numerical optimal control problem online is explicit MPC, which has been mainly developed for linear systems, and formulated using both exact [30] and approximate variants [31,32].

The main advantages of MPC compared to other control approaches are the ability to explicitly impose state or output constraints, and the possibility to exploit information on the future evolution of reference variables, when available. These characteristics, together with a relatively simple and intuitive tuning process (similar in most cases to that of linear quadratic regulators) have greatly contributed to the spread of MPC in the above-mentioned application domains. Only a few applications of MPC to a TRMS have been reported in the literature, such as: MPC schemes based on adaptive linearization [33,34], simpler schemes based on a single linearized model [35,36], and decentralized schemes, in which pitch and yaw were regulated by different model predictive controllers [37].

In this work, we present the use of an MPC law with integral action and output constraints for the control of a TRMS. Neither of these two features has ever been shown in the TRMS literature, and thus the formulation and real-time experimental validation of this strategy constitute a novel contribution. More specifically, the main novelties of our approach are:

1. Zero steady-state error tracking of constant set-points is obtained by adding the integral of the tracking error to the MPC cost function. This simple and straightforward approach, whose rationale is illustrated afterwards, is typically used in other state-space control approaches, is shown here to be effective in terms of implementation and achieved performance.
2. Hard constraints are imposed on both inputs (motor voltages) and outputs (pitch and yaw angular positions) and experimental results are shown in which ensuring these constraints prevents a collision of the TRMS with an obstacle in its workspace. To the best of our knowledge, in previous works on MPC applied to the TRMS [33–37], the ability to enforce output constraints, which is one of the main advantages of MPC over other techniques, was never demonstrated. If we consider the possible presence of obstacles in the TRMS workspace, this feature can be important to avoid collisions that might damage the system.

Offset-free MPC is an active field of research, with most approaches based on the use of disturbance observers - see, e.g. [38,39]. On the other hand, the inclusion of an integral action to guarantee zero steady-state error is a simple approach rooted in both frequency-domain and state-space control frameworks. Its use in this work is to show that it is effective for highly nonlinear systems of the TRMS type. It is worth highlighting that only few works could be found in which a simple integral action is used in conjunction with MPC. In [40], MPC with integral action was used for accurate and automatic insulin delivery in patients with Type-1 diabetes, based on continuous glucose monitor readings. It was also used in the field of power electronics and electric power generation to control a surface-mounted permanent-magnet synchronous generator [41], and to control a single phase Z-source inverter [42].

In this work, in order to simplify calculations and obtain a real-time implementation, the high-fidelity nonlinear TRMS model presented in [43] is linearized at some equilibrium points, and the obtained LTI models are used by the model predictive controller. The linearisation of the nonlinear TRMS model [43] is not a trivial task; it has been carried out in a preliminary conference contribution [44] and is reported here for the first time in a journal publication.

The paper is organized as follows: the nonlinear and linearised models of the TRMS are presented in Section 2, the model predictive control problem is defined in Section 3, the tuning strategy of the control law is explained in detail in Section 4, results are presented in Section 5, and conclusions are drawn in Section 6.

Notation: The notation is standard: I_n is the identity matrix of size n. $0_{n,m}$ is the zero matrix of dimensions n and m. $'$ denotes matrix transposition. $\text{diag}(\lambda_1, .., \lambda_n)$ is the diagonal matrix with elements $\lambda_1, .., \lambda_n$. sgn denotes the sign function.

2 CONTROL-ORIENTED MODELING

The TRMS (Fig. 1) is a two-degree-of-freedom mechanical system which consists of a beam mounted on a rotary bearing enabling the beam to rotate in the vertical plane (pitch plane). The horizontal motion (yawing motion) of the beam is permitted thanks to another rotary bearing situated at the top of and aligned with the column that supports the whole assembly. The beam is counterbalanced by a perpendicular rod rigidly attached at its center. The beam is actuated by two mutually perpendicular propellers located at its ends and driven by DC motors. Thrust is governed by and is proportional to the square of propellers' angular velocity. The TRMS is characterized by a highly nonlinear and cross-coupled dynamics, and is a challenging system to control.

2.1 Equations of motion

The TRMS equations of motion were derived in [43] and consist of the following four coupled, second-order, nonlinear equations:

$$\begin{aligned} I_\psi \ddot{\psi} - H_Z(\psi)\ddot{\phi} + I_{t1}\dot{\omega}_t &= -\frac{1}{2} I_c \dot{\phi}^2 \sin(2\psi) \\ &- I_{m1}\omega_m \dot{\phi} \sin\psi - G_Y(\psi) + C_{Tm}\omega_m \left|\omega_m\right| l_m \\ &- C_{Rt}\omega_t \left|\omega_t\right| - f_{v\psi}\dot{\psi} - f_{c\psi}\,\text{sgn}\,\dot{\psi} \end{aligned} \tag{1}$$

$$\begin{aligned} &-H_Z(\psi)\ddot{\psi} + \left(I_\phi + I_c\cos^2\psi\right)\ddot{\phi} + I_{m1}\dot{\omega}_m\cos\psi \\ &= H_Y(\psi)\dot{\psi}^2 + I_c\dot{\phi}\dot{\psi}\sin(2\psi) + I_{m1}\omega_m\dot{\psi}\sin\psi \\ &+ C_{Tt}\omega_t\left|\omega_t\right| l_t\cos\psi - C_{Rm}\omega_m\left|\omega_m\right|\cos\psi \\ &- f_{v\phi}\dot{\phi} - f_{c\phi}\,\text{sgn}\,\dot{\phi} - C_c\left(\phi - \phi_0\right) \end{aligned} \tag{2}$$

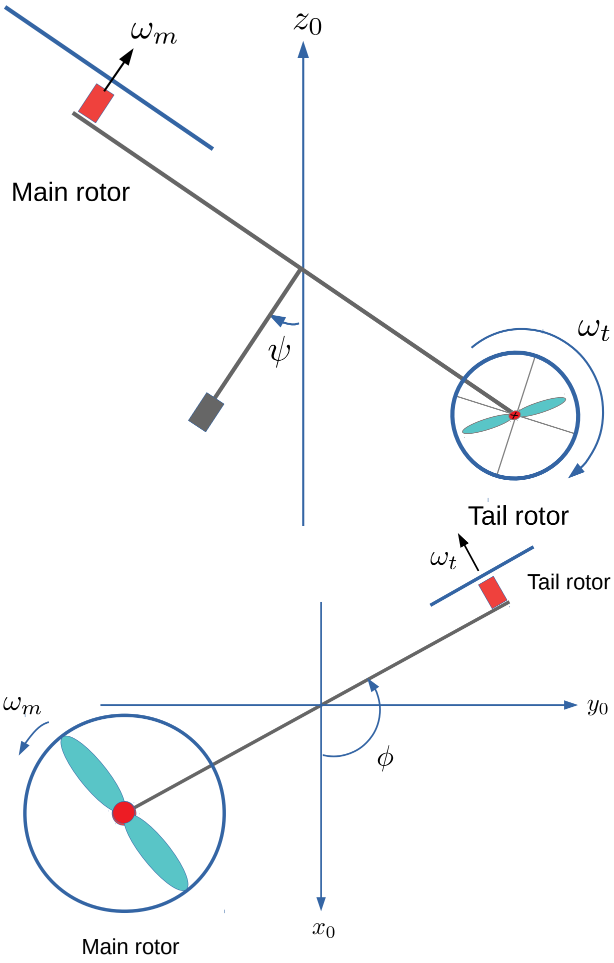

Fig. 2. TRMS pitch (top) and yaw (bottom) angles definitions.

$$\begin{aligned} I_{m1}\dot{\omega}_m + I_{m1}\ddot{\phi}\cos(\psi) &= I_{m1}\dot{\phi}\dot{\psi}\sin\psi + \frac{k_{tm}k_{um}}{R_m}u_m \\ &- \left(C_{Rm}\left|\omega_m\right| + f_{vm} + \frac{k_{vm}k_{tm}}{R_m}\right)\omega_m \end{aligned} \tag{3}$$

$$\begin{aligned} I_{t1}\dot{\omega}_t + I_{t1}\ddot{\psi} &= \frac{k_{ut}k_{tt}}{R_t}u_t \\ &- \left(C_{Rt}\left|\omega_t\right| + f_{vt} + \frac{k_{vt}k_{tt}}{R_t}\right)\omega_t \end{aligned} \tag{4}$$

where

$$\begin{bmatrix} H_Y(\psi) \\ H_Z(\psi) \end{bmatrix} \triangleq \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} H_y \\ H_z \end{bmatrix} \tag{5}$$

In these equations, the subscript "m" or "t" refer to the main or the tail rotor, respectively. The pitch angle is denoted as ψ while ϕ denotes the yaw angle, as indicated in Fig. 2. The main and tail rotor angular velocities are referred to as ω_m and ω_t, respectively. The parameters of the equations of motion are given in Table 1 where l_m and l_t denote the lengths of the main and tail beams, R_m and R_t are the electric resistances of the motors, k_{vm} and k_{vt} are the motors EMF constants. k_{tm} and k_{tt} are the motors torque constants and k_{um} and k_{ut} are the main and tail motors input gains. The other entries in Table 1 are the mass moments of inertia (initial I), friction coefficients (initial f) and main and tail rotors aerodynamics thrust and drag coefficients with initial C where the subscripts T (respectively R) refer to thrust (respectively drag) and the superscripts "$+$" (respectively "$-$") refer to a clockwise (respectively an anticlockwise) rotation of the rotor. Finally, ϕ_0 denotes the yaw angle position for which the restoring moment (due to the cables) is zero and ψ_0 denotes the pitch position at rest (i.e. when the main rotor is idle). See [43] for a complete description and estimation of the TRMS parameters.

In the sequel, the TRMS state vector is defined as:

$$x \triangleq \left[\psi\ \phi\ \dot{\psi}\ \dot{\phi}\ \omega_m\ \omega_t\right]' \tag{6}$$

and the control input as:

$$u \triangleq \left[u_m\ u_t\right]' \tag{7}$$

which consists of voltages in the interval $[-2.5,\ 2.5\,]$V that are then linearly amplified in order to map to the appropriate DC motors ranges.

2.2 Solutions at equilibrium

The equilibrium states of a nonlinear model can, in principle, be found numerically using e.g. quadratic sequential programming or Newton-Raphson algorithms. One drawback is that these algorithms require an initial guess and the convergence to an equilibrium state is not guaranteed. Another issue, due to the ill-conditioned nature of the TRMS model, is that standard numerical tools such as the *"trim"* and *"linmod"* MATLAB routines do not provide reliable results for this model. This subsection and the next provide *exact*, closed-form solutions to the equilibrium states along with explicit expressions for the state-space matrices of the linearized TRMS model. The results of this section were preliminary introduced in [44].

Table 1. Parameters of the equations of motion [43]

parameter	value	unit
l_m	0.254	m
l_t	0.275	m
R_m R_t	8	Ω
k_{vm} k_{vt}	0.0202	$\mathrm{N\,m\,A^{-1}}$
k_{tm} k_{tt}	0.0202	$\mathrm{V\,rad^{-1}\,s}$
k_{um}	8.5	No units
k_{ut}	6.5	No units
I_{m1}	1.7249×10^{-4}	$\mathrm{kg\,m^2}$
I_{t1}	3.2170×10^{-5}	$\mathrm{kg\,m^2}$
I_ψ	6.1644×10^{-2}	$\mathrm{kg\,m^2}$
I_ϕ	2.1117×10^{-2}	$\mathrm{kg\,m^2}$
I_c	8.3286×10^{-2}	$\mathrm{kg\,m^2}$
$f_{v\psi}$	4.2150×10^{-4}	$\mathrm{N\,m\,rad^{-1}\,s}$
$f_{c\psi}$	4.0548×10^{-4}	$\mathrm{N\,m}$
$f_{v\phi}$	1.1243×10^{-4}	$\mathrm{N\,m\,rad^{-1}\,s}$
$f_{c\phi}$	1.1758×10^{-5}	$\mathrm{N\,m}$
f_{vm}	4.1993×10^{-6}	$\mathrm{N\,m\,rad^{-1}\,s}$
f_{vt}	3.2498×10^{-6}	$\mathrm{N\,m\,rad^{-1}\,s}$
C_{Tm}^{+}	1.4971×10^{-5}	$\mathrm{N\,s^2\,rad^{-2}}$
C_{Tm}^{-}	8.9974×10^{-6}	$\mathrm{N\,s^2\,rad^{-2}}$
C_{Tt}^{+}	3.7755×10^{-6}	$\mathrm{N\,s^2\,rad^{-2}}$
C_{Tt}^{-}	2.3046×10^{-6}	$\mathrm{N\,s^2\,rad^{-2}}$
C_{Rm}^{+}	5.0582×10^{-7}	$\mathrm{N\,m\,s^2\,rad^{-2}}$
C_{Rm}^{-}	4.5047×10^{-7}	$\mathrm{N\,m\,s^2\,rad^{-2}}$
C_{Rt}^{+}	9.7031×10^{-8}	$\mathrm{N\,m\,s^2\,rad^{-2}}$
C_{Rt}^{-}	9.9176×10^{-8}	$\mathrm{N\,m\,s^2\,rad^{-2}}$
C_c	9.8664×10^{-3}	$\mathrm{N\,m\,rad^{-1}}$
ψ_0	-0.5369	rad
ϕ_0	0.2500	rad
G_z	$-0.252\,54$	$\mathrm{N\,m}$
G_y	$0.150\,32$	$\mathrm{N\,m}$
H_z	-1.5446×10^{-3}	$\mathrm{kg\,m^2}$
H_y	6.6120×10^{-3}	$\mathrm{kg\,m^2}$

The angular velocities of the rotors, ω_{me} and ω_{te}, at equilibrium, are obtained by solving the nonlinear equations of motion (1)-(2) in which $\dot{\psi}$, $\dot{\phi}$, $\ddot{\psi}$ and $\ddot{\phi}$ are set to zero, that is:

$$\begin{bmatrix} C_{Tm} l_m & -C_{Rt} \\ -C_{Rm}\cos\psi_e & C_{Tt} l_t \cos\psi_e \end{bmatrix} \begin{bmatrix} \omega_{me}|\omega_{me}| \\ \omega_{te}|\omega_{te}| \end{bmatrix} = \begin{bmatrix} G_y \cos\psi_e - G_z \sin\psi_e \\ C_c(\phi_e - \phi_0) \end{bmatrix}$$

where ψ_e and ϕ_e are the equilibrium values for pitch and yaw angles, respectively. Solving for $\omega_{me}|\omega_{me}|$ and $\omega_{te}|\omega_{te}|$ yields:

$$\begin{aligned} \omega_{me}|\omega_{me}| &= \frac{1}{\Delta}((G_y \cos\psi_e - G_z \sin\psi_e) C_{Tt} l_t \cos\psi_e \\ &\quad + C_{Rt} C_c(\phi_e - \phi_0)) \\ \omega_{te}|\omega_{te}| &= \frac{1}{\Delta}((G_y \cos\psi_e - G_z \sin\psi_e) C_{Rm} \cos\psi_e \\ &\quad + C_{Tm} l_m C_c(\phi_e - \phi_0)) \end{aligned}$$

where

$$\Delta = (C_{Tm} C_{Tt} l_m l_t - C_{Rm} C_{Rt}) \cos\psi_e.$$

These equations can be rearranged as:

$$\begin{aligned} \omega_{me}|\omega_{me}| &= \frac{C_{Tt} l_t}{2\Delta}\left(\sqrt{G_y^2 + G_z^2}\cos(2\psi_e - \theta) + G_y\right) \\ &\quad + C_{Rt}\frac{C_c}{\Delta}(\phi_e - \phi_0) \qquad (8) \\ \omega_{te}|\omega_{te}| &= \frac{C_{Rm}}{2\Delta}\left(\sqrt{G_y^2 + G_z^2}\cos(2\psi_e - \theta) + G_y\right) \\ &\quad + C_{Tm} l_m \frac{C_c}{\Delta}(\phi_e - \phi_0) \qquad (9) \end{aligned}$$

where $\theta := \tan^{-1}(-G_z/G_y)$.

The propeller thrust and drag coefficients such as C_{Tm} and C_{Rm} (for the main rotor) depend on the sign of the rotor velocity, that is, these can take the values C_{Tm}^+, C_{Rm}^+ or C_{Tm}^-, C_{Rm}^- depending on the direction of rotation of the rotor. Thus, when solving (8)-(9), since there are two rotors, four cases must be tested for each given equilibrium position pair (ψ_e, ϕ_e). The velocities to retain are those which have their signs matching the correct propeller drag and thrust coefficients which were a priori assigned to the right-hand sides of (8) and (9).

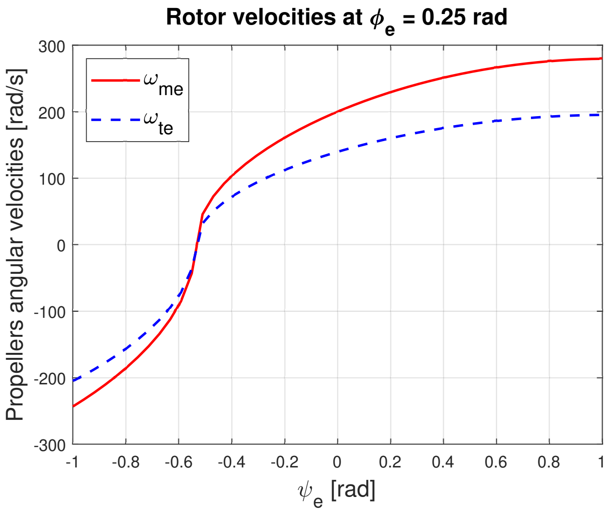

Fig. 3. Propellers' velocities at equilibrium versus pitch positions for the fixed rest yaw position: $\phi_e = \phi_0$.

The solutions are plotted in figures 3 and 4. Fig. 3 shows the equilibrium main and tail rotors angular velocities against various pitch angle equilibrium values when the yaw angle is constant and equals to its rest value ϕ_0. Equations (8)-(9) show that ω_{me} and ω_{te} simultaneously vanish and change signs at the rest positions (ψ_0, ϕ_0) where

$$\begin{aligned} \psi_0 &= \left(-\theta - \cos^{-1} G_y/\sqrt{G_y^2 + G_z^2}\right)/2 \\ &\approx -0.5369\,\text{rad} \qquad (10) \end{aligned}$$

Note that, because thrust and drag coefficients depend on rotor velocity directions, the equilibrium velocities curves are not symmetric with respect to the point $(\psi_0,\, 0)$ as seen in Fig. 3.

In the most general situation, when $\phi_e \neq \phi_0$, equations (8)-(9) show that the equilibrium yaw position will not be affected by the main rotor velocity significantly because $C_{Tt} l_t$ is much greater than $C_{Rt} C_c$. However, because the yaw axis is subject to a restoring torque due to an electric connection cable which acts like a weakly non-linear torsional spring, the equilibrium tail rotor velocity, ω_{te}, will depend heavily on both equilibrium pitch and yaw values as observed on the contour map of Fig. 4. The equilibrium tail rotor velocity also depends on the main rotor position which, due to its size, generates stronger vortices and so stronger aerodynamics couplings.

2.3 Linearization of the equations of motion

The equations of motion are linearized by perturbing the system state, $x(t)$, and the control input, $u(t)$, about their equilibrium points x_e and u_e, that is, by introducing $x(t) = x_e + \delta x(t)$ and $u(t) = u_e + \delta u(t)$ into the equations of motion (1)-(4). Neglecting the Coulomb friction parameters $f_{c\psi}$ and $f_{c\phi}$ and the terms of order higher than one in (1)-(4) provides directly the linear, time-invariant approximated dynamics in descriptor form:

$$E\delta\dot{x} = A_E\delta x + B_E\delta u \tag{11}$$

where the descriptor matrix E is given by:

$$E = \begin{bmatrix} I_2 & 0 \\ 0 & E_2 \end{bmatrix}$$

and where E_2 is the symmetric, invertible matrix given by:

$$E_2 = \begin{bmatrix} I_\psi & -H_Z(\psi_e) & 0 & I_{t1} \\ -H_Z(\psi_e) & I_\phi + I_C\cos^2\psi_e & I_{m1}\cos\psi_e & 0 \\ 0 & I_{m1}\cos\psi_e & I_{m1} & 0 \\ I_{t1} & 0 & 0 & I_{t1} \end{bmatrix}$$

whose entries consist of the various inertia terms and are functions of the pitch equilibrium position ψ_e.

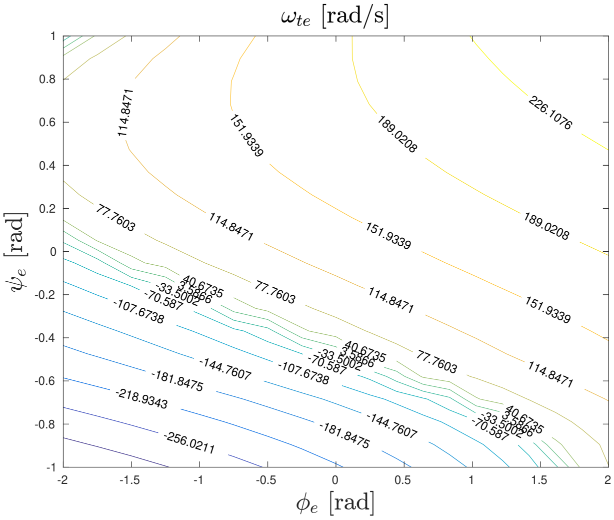

Fig. 4. Equilibrium tail rotor angular velocity contour lines for equilibrium values of pitch angles in ± 1 rad and yaw angles in ± 2 rad.

The matrix A_E depends on the pitch equilibrium, ψ_e, and on the propellers angular velocities, ω_{me} and ω_{te}, and is given by:

$$A_E = \left[\begin{array}{c|c} 0_{2,2}\ I_2 & 0_{2,2} \\ A_{E21} & A_{E22} \end{array}\right]$$

where

$$A_{E21} = \begin{bmatrix} G_Z(\psi_e) & 0 & -f_{v\psi} & -I_{m1}\omega_{me}\sin\psi_e \\ C_{Rm}\omega_{me}^2\sin\psi_e & -C_c & I_{m1}\omega_{me}\sin\psi_e & -f_{v\phi} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

and

$$A_{E22} = \begin{bmatrix} 2C_{Tm}l_m\omega_{me}\,\mathrm{sgn}\,\omega_{me} & -2C_{Rt}\omega_{te}\,\mathrm{sgn}\,\omega_{te} \\ -2C_{Rm}\cos\psi_e\omega_{me}\,\mathrm{sgn}\,\omega_{me} & 2C_{Tt}l_t\cos\psi_e\omega_{te}\,\mathrm{sgn}\,\omega_{te} \\ -2C_{Rm}\omega_{me}\,\mathrm{sgn}\,\omega_{me} & 0 \\ 0 & -2C_{Rt}\omega_{te}\,\mathrm{sgn}\,\omega_{te} \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ f_{vm} + \frac{k_{vm}k_{tm}}{R_m} & 0 \\ 0 & f_{vt} + \frac{k_{vt}k_{tt}}{R_t} \end{bmatrix}$$

Finally, the control matrix is given by:

$$B_E = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{k_{tm}k_{um}}{R_m} & 0 \\ 0 & \frac{k_{tt}k_{ut}}{R_t} \end{bmatrix}$$

Because E is non-singular for any ψ_e, the linearized TRMS dynamics reduce to the continuous-time, linear state-space dynamics:

$$\begin{aligned} \delta\dot{x} &= E^{-1}A_E\delta x + E^{-1}B_E\delta u \\ &= A_c(\psi_e, \omega_{me}, \omega_{te})\delta x + B_c(\psi_e)\delta u \end{aligned} \tag{12}$$

which are valid about the equilibrium state:

$$x_e = \begin{bmatrix} \psi_e\ \phi_e\ 0\ 0\ \omega_{me}\ \omega_{te} \end{bmatrix}' \tag{13}$$

and the equilibrium control input voltages, deduced from (3) and (4), given by:

$$u_{me} = \frac{R_m}{k_{tm}k_{um}}\left(C_{Rm}\left|\omega_{me}\right| + f_{vm}\right)\omega_{me} + \frac{k_{vm}}{k_{um}}\omega_{me} \tag{14}$$

$$u_{te} = \frac{R_t}{k_{tt}k_{ut}}\left(C_{Rt}\left|\omega_{te}\right| + f_{vt}\right)\omega_{te} + \frac{k_{vt}}{k_{ut}}\omega_{te} \tag{15}$$

where ω_{me} and ω_{te} — which are both functions of the pitch and yaw equilibrium positions — satisfy equations (8) and (9). The interested reader is referred to [44] for more details.

3 THE MODEL PREDICTIVE CONTROL LAW

3.1 Formulation

As the model predictive control law is formulated in discrete time, it is necessary to discretize the linearized system dynamics given in (12). These are obtained with the zero-order-hold method with sampling period T_s as:

$$\delta x_{k+1} = A\delta x_k + B\delta u_k \tag{16}$$

with

$$A = e^{A_c T_s}, \quad B = \int_0^{T_s} e^{A_c s} B_c \, ds \tag{17}$$

and with state and control input deviations from equilibrium defined as:

$$\delta x_k = x_k - x_e, \quad \delta u_k = u_k - u_e \tag{18}$$

in which x_e and u_e refer to the equilibrium state and equilibrium control input about which the continuous-time nonlinear TRMS model (1)-(4) is linearized. For the sampling periods typically used for the TRMS, that is, ranging from 1 to 25 ms, the pair (A, B) is found to be stabilizable.

The control objective is to steer asymptotically to zero the tracking error:

$$e_k = r_k - y_k = r_k - C(x_e + \delta x_k) \tag{19}$$

where r_k is the set-point signal which specifies the desired pitch and yaw trajectories and $y_k = [\psi_k, \phi_k]' = Cx_k$ is the TRMS output to be controlled where

$$C = \begin{bmatrix} 1\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0 \end{bmatrix} \tag{20}$$

The set-point introduced above specifies absolute trajectories. Depending on the tracking problem, a relative set-point can be specified with $e_k = r_k - C\delta x_k$.

In order to achieve a zero steady state tracking error in response to a constant set-point, the tracking error $e_k = r_k - y_k$ is integrated as follows:

$$\sigma_{k+1} = \sigma_k + T_s e_k \tag{21}$$

where σ denotes the integrator state vector. The discrete-time plant model (16) together with the integrator dynamics given in (21) is thus governed by:

$$\xi_{k+1} = f_a(\xi_k, \delta u_k, \rho_k) = A_a\xi_k + B_a\delta u_k + \rho_k \tag{22}$$

where

$$\xi_k = \begin{bmatrix} \delta x_k \\ \sigma_k \end{bmatrix} \tag{23}$$

is the state-vector of the augmented system and where

$$A_a = \begin{bmatrix} A & 0 \\ -T_s C & I \end{bmatrix}, \; B_a = \begin{bmatrix} B \\ 0 \end{bmatrix}, \; \rho_k = \begin{bmatrix} 0 \\ T_s(r_k - y_e) \end{bmatrix}$$

are the corresponding state space matrices where $y_e = Cx_e$.

It can be shown that the pair (A_a, B_a) is stabilizable if (A, B) is stabilizable and if $\begin{bmatrix} A - I & B \\ C & 0 \end{bmatrix}$ has full row-rank. The latter condition ensures that the system does not contain transmission zeros at the zero frequency. For the TRMS, it is easy to verify that the rank condition always holds with the C matrix given in (20). In general, if the above conditions are satisfied and if the control horizon is long enough, an unconstrained model predictive controller for the augmented system will be able to asymptotically steer the error to zero.

Throughout, we will assume that the specified set-points are feasible, that is, asymptotic tracking can be achieved without violating the physical constraints. Hence, in the context of integral control, the MPC tracking problem of a constant set-point reduces to a standard MPC regulation problem.

The model predictive controller is implemented by solving a finite-horizon optimal control problem at each sampling instant. In the finite-horizon optimal control problem formulation, the discrete-time index k refers to the time evolution of the actual system dynamics, while $k+i|k$ refers to values predicted at time $k+i$ given the initial condition at time k. The total length of the prediction in terms of discrete-time steps (namely, the *prediction horizon*) is indicated as N, and is chosen for our application as $N=20$ together with a sampling period $T_s=20$ ms: this implies a total prediction time of $NT_s=0.4$ s. We also define the sequences

$$\delta U_k = \left\{\delta u_{k|k}, \delta u_{k+1|k}, \ldots, \delta u_{k+N-1|k}\right\} \tag{24}$$

$$\Xi_k = \left\{\xi_{k|k}, \xi_{k+1|k}, \ldots, \xi_{k+N|k}\right\} \tag{25}$$

which allow us to introduce the cost function

$$\begin{aligned} J(\Xi_k, \delta U_k) & \\ &= \sum_{i=0}^{N-1} \ell(\xi_{k+i|k}, \delta u_{k+i|k}) + V_f(\xi_{k+N|k}) \end{aligned} \tag{26}$$

In (26), the so-called *stage cost* is defined as

$$\begin{aligned} \ell(\xi_{k+i|k}, \delta u_{k+i|k}) & \\ &= \xi'_{k+i|k} Q \xi_{k+i|k} + \delta u'_{k+i|k} R \delta u_{k+i|k} \end{aligned} \tag{27}$$

where Q and R are diagonal matrices with strictly positive entries. The *terminal cost* is given by

$$V_f(\xi_{k+N|k}) = \xi'_{k+N|k} P \xi_{k+N|k} \tag{28}$$

where P is the unique symmetric, positive semi-definite matrix, solution to the discrete-time algebraic Riccati equation for A_a, B_a, Q and R. The existence of P is guaranteed thanks to the stabilizability of the pair (A_a, B_a).

The constraints in the finite-horizon optimal control problem impose upper and lower bounds on the control inputs, as $u = u_e + \delta u \in [u_{\min}, u_{\max}]$, where

$$u_{\min} = \begin{bmatrix} -2.5 \\ -2.5 \end{bmatrix}, \; u_{\max} = \begin{bmatrix} 2.5 \\ 2.5 \end{bmatrix} \tag{29}$$

are defined to satisfy the physical limits on input voltages already mentioned in Section 2. The constraints on the output are defined by $y = y_e + \delta y \in [y_{\min}, y_{\max}]$, where $\delta y = C\delta x$ and $y_{\min}, y_{\max}$ are limits which depend on the specific obstacles present in the TRMS workspace.

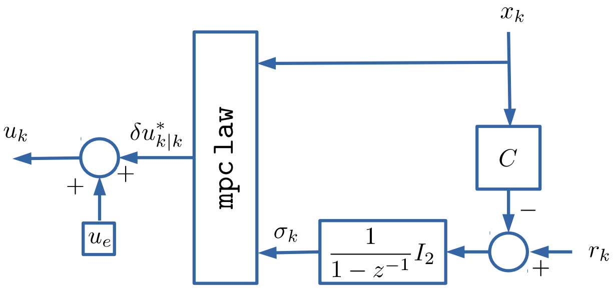

Fig. 5. Model Predictive Controller with integral action (z^{-1} represents the unit time-delay)

After defining as $\mathbb{N}_{[a,b]}$, with a, b integers, the sequence $\{a, a+1, \ldots, b\}$, the finite-horizon optimal control problem can be finally introduced as follows:

$$\min_{\Xi_k, \delta U_k} J(\Xi_k, \delta U_k) \tag{30a}$$

subject to

$$\xi_{k|k} = \xi_k \tag{30b}$$

$$\begin{aligned} \xi_{k+i+1|k} & \\ &= f_a(\xi_{k+i|k}, \delta u_{k+i|k}, \rho_k),\; i \in \mathbb{N}_{[0,N-1]} \end{aligned} \tag{30c}$$

$$u_{\min} \leq \delta u_{k+i|k} + u_e \leq u_{\max},\; i \in \mathbb{N}_{[0,N-1]} \tag{30d}$$

$$y_{\min} \leq \delta y_{k+i|k} + y_e \leq y_{\max},\; i \in \mathbb{N}_{[1,N]} \tag{30e}$$

The first equality constraint (30b) provides the initial state to the prediction model (30c) with the current value ξ_k, whose components are directly obtained from sensor measurements (for δx_k) and from the control law (for σ_k). The second equality constraint (30c) imposes the dynamics (22) of the augmented system along the prediction horizon. Finally, the inequality constraints (30d)-(30e) impose the input and output constraints, respectively. Once the optimal control sequence δU^*_k has been computed only its first entry, namely $\delta u^*_{k|k}$ is used to form the control input applied to the plant:

$$u_k = u_e + \delta u^*_{k|k} \tag{31}$$

The above optimization problem is then repeatedly solved at each sampling instant, according to the *receding-horizon principle* [10]. The overall model predictive control law with integral action is shown in Fig 5. Notice that the integral action, in addition to allowing the achievement of a set-point different from x_e without relinearizing the system dynamics, provides an implicit compensation of the discrepancy between the actual (nonlinear) TRMS dynamics and its linearized version.

If the MPC control law that generates u_k in (31) asymptotically brings system (22) to a steady state, then it can be shown that e_k tends to zero as $k \to \infty$, and thus no steady-state error is present. Indeed, if system (22) tends to a steady state as $k \to \infty$, this holds also for the state of the integral action σ_k, and thus $\lim_{k\to\infty} \sigma_{k+1} - \sigma_k = 0$. By substituting this result in (21), as T_s is a positive constant, we obtain

$$\lim_{k\to\infty} e_k = 0. \tag{32}$$

As e_k is defined in (19) as the error between angles ψ and ϕ and their references, the convergence to zero steady-state error directly holds for the nonlinear system as well. As this result does not rely on any specific values of the system parameters, it also holds when inexact values are used for the parameters in (1)-(4).

3.2 Model predictive controller implementation

As all equality and inequality constraints in (30) are linear and the cost function is convex and quadratic, the optimal control problem is a convex quadratic program. Both input and state variables are employed as decision variables, and following the so-called *simultaneous approach*, the quadratic optimization program is formulated and solved using CVXGEN [45,46]. The high level CVXGEN code for the MPC optimization problem of this paper is given in Fig. 6. For computational efficiency, the sparsity of the augmented system matrices A_a and B_a was taken into account in the 'parameter definition section' (not shown here) of the code, which permitted a prediction horizon of length 20. The code of Fig. 6 was then processed by the online CVXGEN code generator [47] to produce a C source code, which was then compiled and used by a 3.20 GHz laboratory PC for the real-time execution of the control law.

```
dimensions
m = 2 # Dimension of the control input
n = 8 # Dimension of the augmented state
N = 20 # Length of the horizon
end
variables
# Augmented system state-vector array
xi[i] (n), i=1..N
# Control input array
delta_u[i] (m), i=0..N-1
end
minimize
# Cost function
sum[i=0..N-1](quad(xi[i], Q) +
quad(delta_u[i], R)) + quad(xi[N], P)
subject to
# Augmented System Dynamics
xi[i+1] == Aa*xi[i] + Ba*delta_u[i] +
rho, i=0..N-1
# Control Input Constraints
u_min <= (delta_u[i] + u_e) <= u_max,
 i=0..N-1
# Pitch and Yaw Constraints
psi_con <= (xi[i][1] + x_e[1]), i=1..N
phi_con <= (xi[i][2] + x_e[2]), i=1..N
end
```

Fig. 6. High level CVXGEN code of the finite-horizon optimal control problem defined in (30).

4 CONTROLLER TUNING

To summarize, at each sampling instant (with $T_s =$ 20 ms), after obtaining the state measurement, the MPC problem (30) is solved using the embedded MPC controller generated via CVXGEN. As a quadratic cost function is used, the parameters to be tuned consist of the weighting matrices, Q and R in the stage cost (27). These are tuned appropriately to penalise the augmented state (ξ) and the control input (u) respectively. Tuning is aimed at achieving a fast tracking of set commands with a smooth transient response i.e. small overshoot, rise and settling times. Q and R are initially set up using the Bryson's rule [48], and then finetuned so as to yield responses in line with the contributions of this paper. If we define Q and R as follows:

$$Q = \mathrm{diag}(q_1, q_2, ..., q_8); \qquad R = \eta I_2$$

where η is a gain value used in tuning R whose diagonal elements are equally weighted, the finetuning done for Q is based on the experiential knowledge that:

- The higher q_7 and q_8 entries are, the quicker offset-free tracking response is achieved for ψ and ϕ respectively, by reducing settling time and rise time, but this may cause an overshoot or an increase of it.
- The higher q_3 and q_4 entries are, the smoother the transient responses of ψ and ϕ become, but at the cost of increasing the rise time.
- The higher q_1 and q_2 entries are, the lesser the overshoots of ψ and ϕ become, but at the cost of increased control action potentially leading to saturation.
- q_5 and q_6 are left at zero (not tuned) as this would

otherwise present competing objectives against pitch (ψ) and yaw (ϕ) position control.

In tuning the R matrix towards achieving satisfactory tracking, care is taken in finding a tradeoff between not effectively penalising the control inputs on the one hand, and avoiding a continued saturation of the control inputs (u_m, u_t) about u_{max} and u_{min} on the other hand. While the former is usually very evident in simulation, the latter (continued saturation of control inputs) is rather subtle in simulation with hints such as a slight increase in actual time taken for each simulation, and at an extreme, noisy control input signals. As a result, continued saturation of inputs is usually most evident in real-time experimentation, and in the case of the TRMS, causes inordinate and noisy vibrations of the rotor shields and will tend to damage the rotor motors if sustained over longer periods. To avoid continued saturation while getting the best out of penalising the control inputs, R is usually set as an identity matrix multiplied by a *high* gain value, η, usually 0.1% – 0.2% of the maximum gain value in Q. Simulation and experimentation offered no significant merit when the diagonal elements of R are weighted differently.

The differences between the TRMS pitch and yaw positions at rest $y_0 = Cx_0$, the equilibrium output y_e and the reference positions to be tracked r_k (i.e. $y_0 \neq y_e \neq r_k$) present a large error. This phenomenon leads to control actions starting at saturation points, meaning the rotors are forced to start at maximum velocities, and it causes overshoots usually between $0s$ and $5s$ for pitch and yaw. Having different Q matrices for each command in r_k is one good method to counteract the effect of this phenomenon, among its other benefits to tracking set commands but this method presents an added complexity as the commands to be tracked in r_k increases. As an alternative method, σ_0 is carefully chosen to reduce the overshoots caused by this phenomenon.

5 RESULTS AND DISCUSSIONS

Three experimental cases corresponding to Figs. 7 – 9 respectively are presented below, in which simultaneous offset-free output command tracking is shown in the first case, and output constraints are present in the second and third cases. In all cases (see bottom plots of Figs. 7 – 9), the control signal satisfies the voltage constraints given in (29).

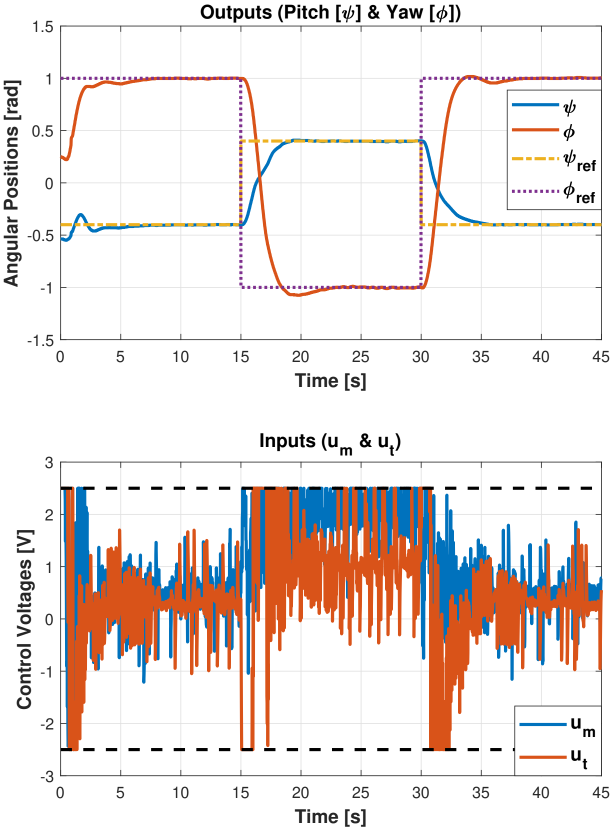

Fig. 7. Top: pitch (ψ) and yaw (ϕ) responses to ± 0.4 rad and ± 1 rad set-points without output constraints. Bottom: main and tail rotors control signals with limits

In the first case, the TRMS pitch and yaw angles are required to follow two square waves applied simultaneously, each of period 30 s and of respective amplitudes 0.4 rad and 1 rad. In this case, the model predictive controller uses the TRMS dynamics linearized at values of pitch and yaw both equal to zero (i.e. $y_e = [0,\ 0]'$), as well as $Q = \mathrm{diag}(6000, 3500, 2000, 1000, 0, 0, 5500, 5000)$, $R = \mathrm{diag}(8, 8)$, and $\sigma_0 = [-0.9,\ 0.3]'$. Well-damped responses with zero steady-state tracking errors are achieved thanks to the integral actions as seen in Fig. 7 (top plots). This is also achieved for the second and third cases, before and after the output constraints are actively imposed.

In the second case, the yaw reference is fixed at $\phi_0 = 0.25$ rad, while the pitch is required to follow a square wave of amplitude 0.4 rad and of period 30 s. As before, the model predictive controller uses the TRMS dynamics linearized at values of pitch and yaw both equal to zero, as well as $\sigma_0 = [-0.9,\ 0.3]'$ and $R = \mathrm{diag}(8, 8)$ but with $Q = \mathrm{diag}(5000, 500, 500, 6000, 0, 0, 5000, 6000)$. The responses in the absence of output constraints and obstacles are shown in Fig. 8 (left plots). A first obstacle is then inserted, consisting of a paper box. Collision avoidance can be guaranteed for the TRMS if the yaw angle is

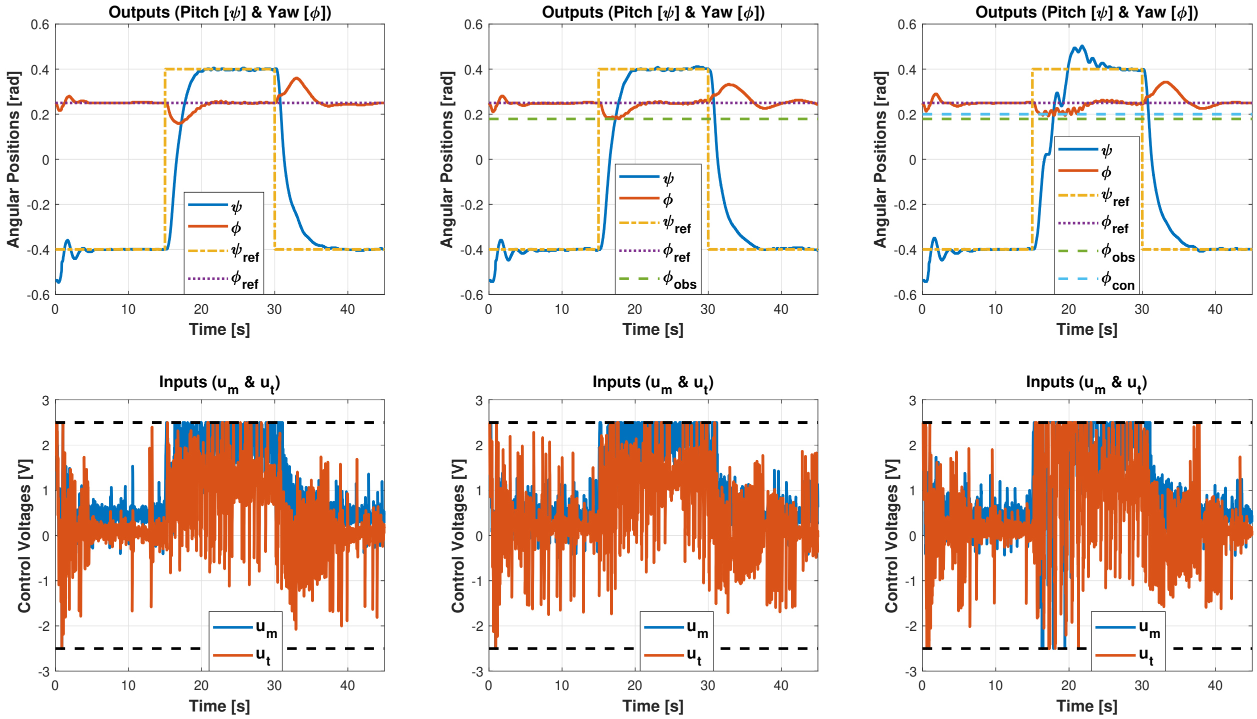

Fig. 8. Top: pitch (ψ) and yaw (ϕ) responses to set-points ± 0.4 rad and $\phi_0 = 0.25$ rad for unconstrained and unobstructed yaw (left), unconstrained but physically obstructed yaw (center), and obstructed yaw with output constraint. The line ϕ_{obs} indicates the yaw values at which the TRMS hits the obstacle and the line ϕ_{con} represents the lower yaw constraint. Bottom: control signals u_m, u_t and limits

kept equal to or above $\phi_{obs} = 0.17$ rad (Fig. 10 - top - and Fig. 8 - middle plots). To prevent collisions with the box, a lower bound equal to $\phi_{con} = 0.20$ rad is enforced for the yaw angle in the MPC law; the value of ϕ_{con} is chosen slightly higher than ϕ_{obs} to safeguard against a possible position error of the box, or to the possibility that slight constraints violation occur due to (i) slight discrepancies between the nonlinear system model and the behaviour of the actual plant, (ii) model uncertainties caused by the linearization procedure and (iii) small computation delays due to the need to solve the optimisation problem (30) online. The plots of Fig. 8 (right) show the corresponding responses. It can be seen that pitch overshoot and coupling into pitch are more pronounced when compared to the unconstrained case i.e. Fig. 8 - middle plots. This is a probable adverse effect of MPC enforcing the yaw constraint at the expense of the introduction of extra dynamic couplings into the pitch axis. One can also notice that a slight violation of the imposed state constraint occurs around 18 s, as the value of ϕ falls below $\phi_{con} = 0.20$ rad. However, as the violation is very small, the TRMS still remains at a safe distance from the box, and no collisions occur.

The third case follows a similar scenario. The yaw axis is now required to track a ± 1 rad square wave of period 30 s and the pitch axis is required to stay at its rest position, $\psi_0 = -0.5369$ rad. Unlike previous cases, the TRMS dynamics is linearized at a pitch value equal to -0.5 rad and a yaw value equal to 0 rad (i.e. $y_e = [-0.5,\ 0]'$) since the TRMS dynamics is highly influenced by the pitch position. This challenge is most evident in this case due to the constant pitch position being tracked (-0.5369 rad) and explains the vast gap in the weights in q_1, q_3, q_7 vs q_2, q_4, q_8 in $Q = \mathrm{diag}(6000, 2000, 6000, 2000, 0, 0, 6000, 1500)$. $R = \mathrm{diag}(8, 8)$, and $\sigma_0 = [-0.45,\ 0]'$ are also used in this case. Fig. 9 - left - shows the responses in the absence of output constraints and obstacles. A paper box is then placed such that the TRMS will hit the box if the pitch angle falls below $\psi_{obs} = -0.69$ rad (Fig. 10 - bottom - and Fig. 9 - middle plots). To prevent collisions with the box, a state constraint is introduced in the MPC law, by requiring the pitch angle to remain above $\psi_{con} = -0.6369$ rad; this value is chosen slightly higher than ψ_{obs} to safeguard against the sources of uncertainty already mentioned above for the yaw angle constraints.

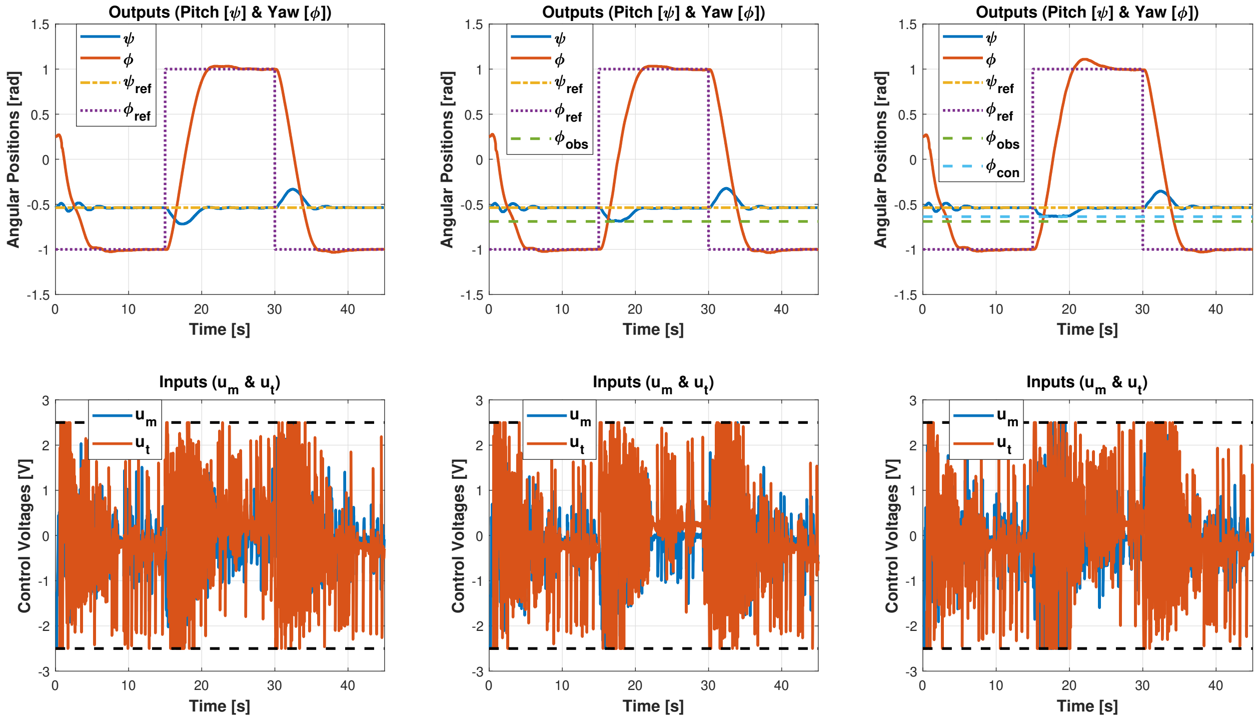

Fig. 9. Top: pitch (ψ) and yaw (ϕ) responses to set-points $\psi_0 = -0.5369$ rad and ± 1 rad for unconstrained and unobstructed yaw (left), unconstrained but physically obstructed pitch (center), and obstructed pitch with output constraint. The line ψ_{obs} represents the pitch values at which the TRMS hits the obstacle and the line ψ_{con} represents the lower pitch constraint. Bottom: control signals u_m, u_t and limits

Similar to the second case, the yaw response presents an increased overshoot (Fig. 9 - right) when the pitch constraint is present. Also, no significant violation of the state constraint can be seen in this case.

The difference in the violations for the constrained yaw and the constrained pitch cases can be partially attributed to the effect of cross-couplings. The effect on the yaw position with pitch constraint imposition is significantly milder (Fig. 9 - right) in comparison with the opposite case in Fig. 8 - right. This is expected since the dynamics of the TRMS depend on the pitch position, making it easier to control in comparison with the yaw.

A potential key limitation to this work is the relatively short prediction horizon ($N = 20$) used in the control law implementation, and this may have also contributed to the negligible output constraint violations discussed above. This is because the online solver generator, CVXGEN, can only generate solvers for medium-size QPs up to 4000 non-zero Karush-Kuhn-Tucker (KKT) matrix entries, thereby limiting the possible prediction horizon length. Different online fast solver generators that can allow longer prediction horizons could be considered as future work. Another possibility for future work is to define distributed MPC schemes that can leverage on a reduced number of decision variables for each controller, thereby providing further chances for increasing the predictions horizons.

6 CONCLUSIONS

This paper has proposed, formulated and experimentally tested model predictive control laws for a TRMS with both input and output constraints. In addition, this paper has proposed a simple yet effective alternative to the offset-free tracking paradigm commonly found in the MPC literature. Here, integral action is employed to provide zero steady-state tracking to constant set-points. The implementation of a model predictive control law in real-time has been demonstrated and the experimental results have shown success in tracking constant references while satisfying all the input and output constraints aimed at avoiding collisions between the TRMS and the obstacles present in its surrounding environment.

Some drawbacks of MPC have also been shown, in particular the satisfaction of the constraints sometime led to nonlinear couplings and/or irregularities in the re-

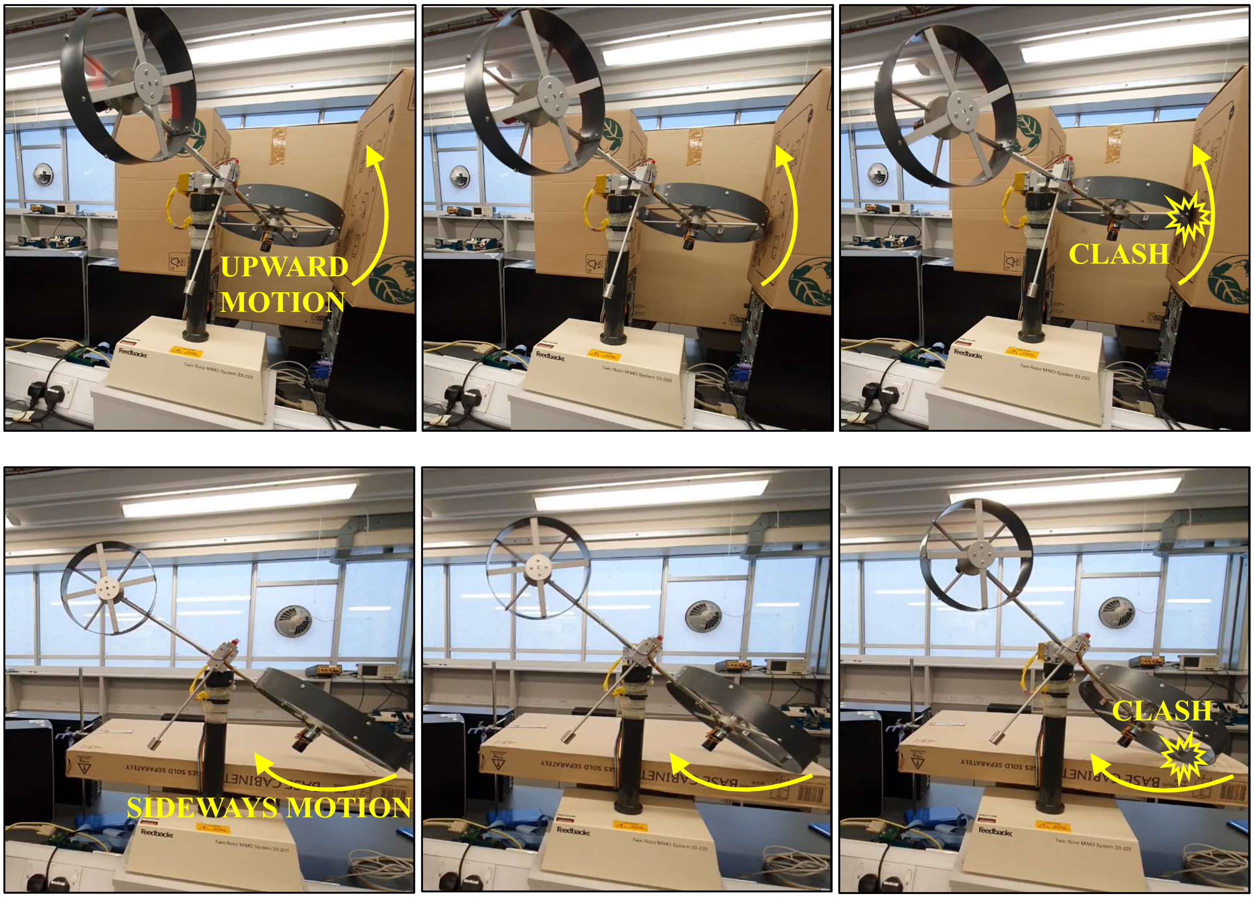

Fig. 10. Obstructed yaw (top) and obstructed pitch (bottom) corresponding to the second and the third cases of Section 5

sponses which were not observed in the absence of output constraints and/or when using a linear state-feedback controller. These issues, which can be critical in some applications and which cannot be easily suppressed by adding extra constraints to the optimization control problem, require future attention.

REFERENCES

[1] López-Martínez, M., Ortega, M. G., Vivas, C., and Rubio, F. R., 2007, "Nonlinear l_2 control of a laboratory helicopter with variable speed rotors," *Automatica,* **43**(4), pp. 655–661.

[2] Wen, P., and Lu, T. W., 2008, "Decoupling control of a twin rotor MIMO system using robust deadbeat control technique," *IET Control theory & applications,* **2**(11), pp. 999–1007.

[3] Mullhaupt, P., Srinivasan, B., Lévine, J., and Bonvin, D., 2008, "Control of the toycopter using a flat approximation," *IEEE Transactions on Control Systems Technology,* **16**(5), pp. 882–896.

[4] Mondal, S., and Mahanta, C., 2012, "Adaptive second-order sliding mode controller for a twin rotor multi-input–multi-output system," *IET Control Theory & Applications,* **6**(14), pp. 2157–2167.

[5] Rotondo, D., Nejjari, F., and Puig, V., 2013, "Quasi–LPV modeling, identification and control of a twin rotor MIMO system," *Control Engineering Practice,* **21**(6), pp. 829 – 846.

[6] Belmonte, L. M., Morales-Viviescas, R. M., Fernández-Caballero, A., and Somolinos, J. A., 2016, "Robust decentralized nonlinear control for a twin rotor MIMO system," *Sensors,* **16**(8).

[7] Roman, R. C., Radac, M. B., and Precup, R. E., 2016, "Multi-input-multi-output system experimental validation of model-free control and virtual reference feedback tuning techniques," *IET Control Theory & Applications,* **10**(12).

[8] Pandey, S. K., Dey, J., and Banerjee, S., 2018, "Design of robust proportional-integral-derivative

controller for generalized decoupled twin rotor multi-input-multi-output system with actuator nonlinearity," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering,* **232**(8), pp. 971–982.

[9] Grüne, L., and Pannek, J., 2017, *Nonlinear model predictive control* Springer.

[10] Rawlings, J. B., Mayne, D. Q., and Diehl, M., 2017, *Model predictive control: Theory, computation, and design* Nob Hill Publishing Madison, WI.

[11] Propoi, A. I., 1963, "Application of linear programming methods for the synthesis of automatic sampled-data systems," *Avtomat. i Telemekh,* **24**(7), pp. 912–920.

[12] Richalet, J., Rault, A., Testud, J., and Papon, J., 1978, "Model predictive heuristic control: Applications to industrial processes," *Automatica,* **14**(5), pp. 413–428.

[13] Qin, S. J., and Badgwell, T. A., 2003, "A survey of industrial model predictive control technology," *Control engineering practice,* **11**(7), pp. 733–764.

[14] Lee, J. H., 2011, "Model predictive control: Review of the three decades of development," *International Journal of Control, Automation and Systems,* **9**(3), pp. 415–424.

[15] Di Cairano, S., Bernardini, D., Bemporad, A., and Kolmanovsky, I. V., 2013, "Stochastic MPC with learning for driver-predictive vehicle control and its application to hev energy management," *IEEE Transactions on Control Systems Technology,* **22**(3), pp. 1018–1031.

[16] Oleinikov, A., Kusdavletov, S., Shintemirov, A., and Rubagotti, M., 2021, "Safety-aware nonlinear model predictive control for physical human-robot interaction," *IEEE Robotics and Automation Letters,* **6**(3), pp. 5665–5672.

[17] Bordons, C., and Montero, C., 2015, "Basic principles of MPC for power converters: Bridging the gap between theory and practice," *IEEE Industrial Electronics Magazine,* **9**(3), pp. 31–43.

[18] Oldewurtel, F., Parisio, A., Jones, C. N., Gyalistras, D., Gwerder, M., Stauch, V., Lehmann, B., and Morari, M., 2012, "Use of model predictive control and weather forecasts for energy efficient building climate control," *Energy and Buildings,* **45**, pp. 15–27.

[19] Mayne, D. Q., 2014, "Model predictive control: Recent developments and future promise," *Automatica,* **50**(12), pp. 2967–2986.

[20] Camacho, E. F., Ramírez, D. R., Limón, D., De La Peña, D. M., and Alamo, T., 2010, "Model predictive control techniques for hybrid systems," *Annual reviews in control,* **34**(1), pp. 21–31.

[21] Allan, D. A., Bates, C. N., Risbeck, M. J., and Rawlings, J. B., 2017, "On the inherent robustness of optimal and suboptimal nonlinear MPC," *Systems & Control Letters,* **106**, pp. 68–78.

[22] Mayne, D. Q., Rawlings, J. B., Rao, C. V., and Scokaert, P. O., 2000, "Constrained model predictive control: Stability and optimality," *Automatica,* **36**(6), pp. 789–814.

[23] Kantas, N., Maciejowski, J. M., and Lecchini-Visintini, A., 2009, "Sequential Monte Carlo for model predictive control," In *Nonlinear Model Predictive Control*. Springer, pp. 263–273.

[24] Mesbah, A., 2016, "Stochastic model predictive control: An overview and perspectives for future research," *IEEE Control Systems Magazine,* **36**(6), pp. 30–44.

[25] Rubagotti, M., Patrinos, P., and Bemporad, A., 2014, "Stabilizing linear model predictive control under inexact numerical optimization," *IEEE Transactions on Automatic Control,* **59**(6), pp. 1660–1666.

[26] Berberich, J., Köhler, J., Müller, M. A., and Allgöwer, F., 2020, "Data-driven model predictive control with stability and robustness guarantees," *IEEE Transactions on Automatic Control,* **66**(4), pp. 1702–1717.

[27] Viswanathan, S. P., Sanyal, A. K., and Samiei, E., 2018, "Integrated guidance and feedback control of underactuated robotics system in SE (3)," *Journal of Intelligent & Robotic Systems,* **89**(1), pp. 251–263.

[28] Chen, T., Babanin, A., Muhammad, A., Chapron, B., and Chen, C., 2020, "Modified evolved bat algorithm of fuzzy optimal control for complex nonlinear systems," *Rom. J. Inf. Sci. Technol,* **23**, pp. T28–T40.

[29] Precup, R.-E., Roman, R.-C., and Safaei, A., 2022, *Data-Driven Model-Free Controllers* CRC Press.

[30] Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E. N., 2002, "The explicit linear quadratic regulator for constrained systems," *Automatica,* **38**(1), pp. 3–20.

[31] Oliveri, A., Barcelli, D., Bemporad, A., Genuit, B., Heemels, M., Poggi, T., Rubagotti, M., and Storace, M., 2012, "MOBY-DIC: A MATLAB toolbox for circuit-oriented design of explicit MPC," In Proc. IFAC Conference on Nonlinear Model Predictive Control, pp. 218–225.

[32] Rubagotti, M., Barcelli, D., and Bemporad, A., 2014, "Robust explicit model predictive control via regular piecewise-affine approximation," *International Journal of Control,* **87**(12), pp. 2583–2593.

[33] Rahideh, A., and Shaheed, M. H., 2011, "Stable model predictive control for a nonlinear system," *Journal of the Franklin Institute,* **348**(8), pp. 1983–2004.

[34] Rahideh, A., and Shaheed, M. H., 2012, "Constrained output feedback model predictive control for nonlinear systems," *Control Engineering Practice,* **20**(4), pp. 431–443.

[35] Duţescu, D. A., Radac, M. B., and Precup, R. E., 2017, "Model predictive control of a nonlinear laboratory twin rotor aero-dynamical system," In IEEE International Symposium on Applied Machine Intelligence and Informatics, pp. 37–42.

[36] Raghavan, R., and Thomas, S., 2017, "Practically implementable model predictive controller for a twin rotor multi-input multi-output system," *Journal of Control, Automation and Electrical Systems,* **28**(3), pp. 358–370.

[37] Tahir, F., Ahmed, Q., and Bhatti, A. I., 2013, "Real-time switched model predictive control of a twin rotor system," In IEEE Conference on Decision and Control, pp. 4847–4852.

[38] Pannocchia, G., 2015, "Offset–free tracking MPC: A tutorial review and comparison of different formulations," In European Control Conference, pp. 527–532.

[39] Ferramosca, A., González, A. H., and Limon, D., 2017, "Offset–free multi-model economic model predictive control for changing economic criterion," *Journal of Process Control,* **54**, pp. 1–13.

[40] Incremona, G. P., Messori, M., Toffanin, C., Cobelli, C., and Magni, L., 2018, "Model predictive control with integral action for artificial pancreas," *Control Engineering Practice,* **77**, pp. 86–94.

[41] Abdelrahem, M., Hackl, C. M., Kennel, R., and Rodriguez, J., 2018, "Efficient direct-model predictive control with discrete-time integral action for PMSGs," *IEEE Transactions on Energy Conversion,* **34**(2), pp. 1063–1072.

[42] Ramírez, R. O., Espinoza, J. R., Baier, C. R., Rivera, M., Villarroel, F., Guzman, J. I., and Melín, P. E., 2018, "Finite-state model predictive control with integral action applied to a single-phase z–source inverter," *IEEE Journal of Emerging and Selected Topics in Power Electronics,* **7**(1), pp. 228–239.

[43] Tastemirov, A., Lecchini-Visintini, A., and Morales-Viviescas, R. M., 2017, "Complete dynamic model of the Twin Rotor MIMO System (TRMS) with experimental validation," *Control Engineering Practice,* **66**, pp. 89–98.

[44] Prempain, E., and Lecchini-Visintini, A., 2018, "Dynamic analysis of a twin rotor MIMO system and control design," In UKACC International Conference, pp. 87–92.

[45] Mattingley, J., Wang, Y., and Boyd, S., 2010, "Code generation for receding horizon control," In IEEE International Symposium on Computer-Aided Control System Design, pp. 985–992.

[46] Mattingley, J., and Boyd, S., 2012, "Cvxgen: A code generator for embedded convex optimization," *Optimization and Engineering,* **13**(1), pp. 1–27.

[47] Mattingley, J., and Boyd, S. CVXGEN: Code generation for convex optimization On the WWW URL `https://cvxgen.com/docs/index.html`.

[48] Bryson, A. E., and Ho, Y. C., 1975, "Applied optimal control, revised printing," *Hemisphere, New York.*