Generalized Predictive Control Design with Colored Noise: a Simulated Case Study for Spacecraft Emulation

1st Antonio Silveira
Lab. of Control and Systems
Federal University of Pará (UFPA)
Belém, Brazil
asilveira@ufpa.br

2nd Marco Sagliano
Dept. of Navigation and Control Systems
German Aerospace Center (DLR)
Bremen, Germany
marco.sagliano@dlr.de

3rd Rodrigo Trentini
Dept. of Electrical Engineering
Federal Institute of Santa Catarina (IFSC)
Jaraguá do Sul, Brazil
rodrigo.trentini@ifsc.edu.br

4th David Seelbinder
Dept. of Navigation and Control Systems
German Aerospace Center (DLR)
Bremen, Germany
david.seelbinder@dlr.de

5th Stephan Theil
Dept. of Navigation and Control Systems
German Aerospace Center (DLR)
Bremen, Germany
stephan.theil@dlr.de

***Abstract*—In this work the Generalized Predictive Control (GPC) is revisited in order to assess its capabilities in handling colored noise disturbances using a novel design procedure. The proposed method is investigated in a simulated case study of spacecraft emulation. Such emulation is proposed by combining a network-controlled quadcopter and a set of computer-based control algorithms that impose approximated spacecraft dynamics to the aerial system. The corresponding model was built upon real registered flight data. The contribution of this paper is twofold: first, we propose a novel GPC design in the colored noise case. Second, we assess the use of quadcopters to investigate spacecraft guidance and control algorithms. Simulation results confirmed the proposed methodology as a possible low-budget spacecraft-emulation alternative for a future experimental setup. One of the main findings of the current investigation is that no significant enhancements coming from the use of GPC in minimizing the chattering of the control signal under the colored noise disturbance case could be observed. An important discussion on this matter and possible solutions are presented for future investigations.**

***Index Terms*—Generalized Predictive Control, Spacecraft Guidance and Control, Stochastic Control, Colored Noise Disturbance, Quadcopter.**

I. Introduction

In this work, two contributions are given: a theoretical one, related to a novel design method for Generalized Predictive Control (GPC) in the colored noise disturbance case, and the prospects for a future experimental setup, by proposing a 3-Degrees-of-Freedom (3-DOF) spacecraft cyber-physical emulation system that combines a network-controlled quadcopter and a set of computer-based control algorithms to impose approximated spacecraft dynamics to the quadcopter in order to work as a low budget alternative to assess spacecraft guidance and control algorithms.

The GPC was first proposed in 1987 as a stochastic control technique [1]. Its stochasticity was inherited from previous early stage stochastic model predictive control techniques, such as minimum variance control, where modeling uncertainties and measurement noise were associated with a stochastic process affecting the controlled system dynamics. However, over the years GPC gained popularity not for its stochastic control properties, but rather for the simplicity of tuning. Then, despite its formal presentation had assumed the colored noise disturbance case, it is seldom used in this way.

GPC stochastic properties comes from the integrated moving average part of an auto-regressive model (ARIMAX) that is used to design the minimum variance predictor. However, in order to avoid a more complex design procedure, only auto-regressive models integrated with exogenous inputs (ARIX) are usually implemented, as can be observed even in the state of the art related to GPC. For example, in [2], a ARIX-based GPC was used for active stability augmentation and vibration reduction of an aeroelastic tiltrotor model with simulated random external disturbances, such as wind gusts and measurement noise in order to assess actuator chatter. In [3], the GPC was used to optimize the performance of a turbofan engine and despite the theoretical background presented assumed the colored noise case, for simplicity, the authors adopted the ARIX model.

In [4], GPC was applied to control switching systems under unknown switching sequences, where the authors have remarked the importance of the stochastic approach but stated that the ARIX approach was the "sole focus of the current paper, with the intention of elaborating the general case in a upcoming research". And many more similar occurrences of such simplification in GPC can be found in the state of the art, as in [5]–[8].

This design procedure gap for the GPC can be traced back to its introduction in 1987 and just a few papers have been reported using the GPC based on the colored noise case but without long range prediction, as remarked in [9]. Even though, the GPC has proven to be an efficient and reliable control method using ARIX. In this sense, in this paper we propose a systematic procedure to design the complete GPC based on ARIMAX and compare its results to the ARIX-based case in a spacecraft emulation with

Manuscript 73 submitted to 2023 15th IEEE International Conference on Industry Applications (INDUSCON). Received April 19, 2023.

simulated random disturbances in the colored process noise case, representing stochastic modeling uncertainties which are inherent in aerospace control applications.

Autonomous rendezvous and docking of spacecrafts and on-orbit or in-space servicing, assembly and manufacturing (OSAM/ISAM) [10] operations require safe and reliable control systems, preferably with validation studies using experimental testbeds. However, guidance and control experiments with spacecrafts in zero gravity are, of course, very expensive. So, on Earth 1-g alternatives are employed in order to assess the reliability of softwares, processors, sensors and actuators, such as in the Test Environment for Applications of Multiple Spacecrafts, TEAMS [11], of the German Aerospace Center. TEAMS is designed to experiment space technologies using 3-DoF and 5-DoF spacecrafts using robotic-platform systems floating over a flat granite surface by means of air bearings.

In this work we investigate by simulations the prospects of emulating 3-DoF spacecrafts by using hardware and software technologies of quadcopters readily available for implementing interesting applications, such as the augmented reality game in [12], developed by the European Space Agency, that used the cyber-physical combination of a quadcopter and a software to simulate the docking procedure to the International Space Station, so to study autonomous rendezvous and docking methods.

In our proposed spacecraft emulation, however, we changed the flight dynamics of the quadcopter in a simulated study, approximating its translational dynamics to a double-integrator system while assuming the validity of such model for short-time and short-distance experiments for spacecrafts in free flight [13]. A model-following control approach was adopted (cf. Figure 1), allowing to embed the double integrator dynamics and to emulate the spacecraft's mass. However, later on, we present that the control signal chattering due to noise feedback is a major issue, since it increases the power loss and jeopardizes the quadcopter's structural and electrical components. Thus, we investigate the ARIMAX-based GPC to reduce the control chattering, expecting to exploit the GPC's prediction ability so to enhance the model-following control performance in order to propose a future experimental on-Earth testbed for 3-DoF spacecraft emulation.

This paper is organized in the following form: in Sec. II the proposed GPC design in the colored noise case is addressed, focusing solely in the aimed contribution, so assuming the single-input, single-output (SISO) with no saturation constraints. Then, the spacecraft emulation solution is described in Sec. III, followed by the quadcopter system identification and model assessment by simulations with the approximated spacecraft simulated dynamics, which is the subject of Sec. IV. ARIMAX versus ARIX-based GPC simulations are assessed. The results obtained are then discussed in Sec. V, and followed by some concluding remarks in Sec. VI.

II. ARIMAX-Based GPC Design

The GPC came after the Generalized Minimum Variance Control (GMVC) by incorporating trajectories of output and control predictions into a finite-time optimal control problem, maintaining GMVC's Minimum Variance Predictor based on the ARIMAX model, given by:

$$A_k(q^{-1})y(k) = B_k(q^{-1})u(k-d) + C_k(q^{-1})\frac{w(k)}{\Delta}, \quad (1)$$

$$\begin{aligned} A_k(q^{-1}) &= 1 + a_{k,1}q^{-1} + \cdots + a_{k,n_a}q^{-n_a}, \\ B_k(q^{-1}) &= b_{k,0} + b_{k,1}q^{-1} + \cdots + b_{k,n_b}q^{-n_b}, \\ C_k(q^{-1}) &= 1 + c_{k,1}q^{-1} + \cdots + c_{k,n_c}q^{-n_c}. \end{aligned} \quad (2)$$

In Eq. (1), $y(k)$, $u(k-d)$, $w(k)$, defined in the discrete time domain $k \in \mathbb{Z}^{0+}$, are, respectively, the output, the d samples delayed input and a zero mean Gaussian noise disturbance of variance σ_w^2. The system is assumed to be linear parameter varying with its polynomials in Eq. (2) defined in the backward-shift operator domain, q^{-1}, and $\Delta = 1 - q^{-1}$ is the discrete-difference operator, such that $\Delta u(k) = u(k) - u(k-1)$. Henceforth, the sub-index k in the polynomials are going to be omitted for the sake of compactness.

The colored noise disturbance is characterized by the presence of the term $C(q^{-1})\frac{w(k)}{\Delta}$ in the model. This leads to the following stochastic distrubance process $\bar{w}(k)$:

$$\bar{w}(k) = \bar{w}(k-1) + C(q^{-1})w(k), \quad (3)$$

which is incremental and having the white noise $w(k)$ being modified by the $C(q^{-1})$ polynomial. Thus, the participation of this polynomial when its order is $n_c \geq 1$ establishes the colored noise disturbance case.

The GPC considers the following optimization problem:

$$\min_{\underrightarrow{\mathbf{\Delta_u}}} \; J = \mathbb{E}\left\{ \sum_{j=d}^{N_y} \left[y(k+j) - r(k+j)\right]^2 + \lambda \sum_{j=1}^{N_u} \left[\Delta u(k+j-1)\right]^2 \right\}, \quad (4)$$

$$\underrightarrow{\mathbf{\Delta_u}} = \begin{bmatrix} \Delta u(k) & \Delta u(k+1) & \cdots & \Delta u(k+N_u-1) \end{bmatrix}^T, \quad (5)$$

subject to the dynamical constraints of the system in Eq. (1).

In Eq. (4), $\mathbb{E}[.]$ denotes the mathematical expectation operator, $N_y \in \mathbb{Z}^+$ is the output prediction horizon, $N_u \in \mathbb{Z}^+ \leq N_y$ is the control prediction horizon and $\lambda \in \mathbb{R}^+$ is a control effort weighting factor. The reference sequence $r(k+j)$ for $j = d, \ldots, N_y$ is assumed to be known *a priori*.

The solution to the problem in Eq. (4) is to determine the sequence $\underrightarrow{\mathbf{\Delta_u}}$ in Eq. (5) that minimizes J, which depends on the knowledge of $y(k+j)$ for $j = d, \ldots, N_y$, as in

$$y(k+j) = \frac{B(q^{-1})}{\Delta A(q^{-1})}\Delta u(k+j-d) + \frac{C(q^{-1})}{\Delta A(q^{-1})}w(k+j), \quad (6)$$

where the future values of the stochastic disturbances are unknown. In the GPC, the solution comes from the Minimum Variance Predictor, by separating this unknown future information into past-to-present and future, as follows:

$$\frac{C(q^{-1})}{\Delta A(q^{-1})}w(k+j) = \underbrace{\frac{F_j(q^{-1})}{\Delta A(q^{-1})}w(k)}_{\text{Past - to - Present}} + \underbrace{E_j(q^{-1})w(k+j)}_{\text{Future}} \quad (7)$$

By doing so, the future part is neglected and assumed as a prediction error of the predicted output based on data available up to time k, i.e.,

$$\hat{y}(k+j|k) = \frac{B(q^{-1})}{\Delta A(q^{-1})}\Delta u(k+j-d) + \frac{F_j(q^{-1})}{\Delta A(q^{-1})}w(k). \tag{8}$$

The prediction error is then associated with the neglected future term, given by:

$$E_j(q^{-1})w(k+j) = y(k+j) - \hat{y}(k+j|k), \tag{9}$$

$$w(k) = \frac{y(k) - \hat{y}(k|k)}{E_j(q^{-1})}. \tag{10}$$

Substituting Eq. (10) into Eq. (8), the Minimum Variance Predictor is obtained as follows:

$$\hat{y}(k+j|k) = \frac{E_j(q^{-1})B(q^{-1})\Delta u(k+j-d) + F_j(q^{-1})y(k)}{q^{-j}F_j(q^{-1}) + \Delta A(q^{-1})E_j(q^{-1})}. \tag{11}$$

To determine these $E_j(q^{-1})$ and $F_j(q^{-1})$ polynomials, for $j = d, \ldots, N_y$, it is required the solution of the Diophantine equation obtained from Eq. (7), that is:

$$C(q^{-1}) = q^{-j}F_j(q^{-1}) + \Delta A(q^{-1})E_j(q^{-1}). \tag{12}$$

The $E_j(q^{-1})$ and $F_j(q^{-1})$ polynomials have the following structure:

$$E_j(q^{-1}) = 1 + e_{j,1}q^{-1} + \cdots + e_{j,(j-1)}q^{-(j-1)} \tag{13}$$

$$F_j(q^{-1}) = f_{j,0} + f_{j,1}q^{-1} + \cdots + f_{j,n_a}q^{-n_a}, \tag{14}$$

so that their orders are $n_{e_j} = j-1$ and $n_{f_j} = n_a$, $\forall j \in \mathbb{N}$.

By comparing the Diophantine equation in Eq. (12) and the characteristic polynomial of the predictor in Eq. (11), the Minimum Variance Predictor can be redefined as

$$\hat{y}(k+j|k) = \frac{E_j(q^{-1})B(q^{-1})\Delta u(k+j-d) + F_j(q^{-1})y(k)}{C(q^{-1})}. \tag{15}$$

So, the colored noise case can deeply affect predictions since the roots of $C(q^{-1})$ are the poles of j predictors.

By assuming that $C(q^{-1})$ roots are supervised and asymptotic stability is guaranteed, the predictor incorporates the colored noise filter as follows:

$$\hat{y}(k+j|k) = E_j(q^{-1})B(q^{-1})\Delta u_f(k+j-d) + F_j(q^{-1})y_f(k). \tag{16}$$

Thus, GPC in the colored noise case utilizes a filtered output feedback and reconstructs the control signal variation after solving $\Delta u_f(k)$, respectively, as follows:

$$y_f(k) = \frac{y(k)}{C(q^{-1})}, \tag{17}$$

$$\Delta u(k) = C(q^{-1})\Delta u_f(k). \tag{18}$$

From this point on there is a gap in the GPC literature in terms of solving $E_j(q^{-1})$ and $F_j(q^{-1})$ in a generalized way and aided by computational methods for the colored noise case. Methods commonly employed are the recursive form [14] or a different approach independent of $E_j(q^{-1})$ and $F_j(q^{-1})$, which is state-space Model Predictive Control [15], based on state-space realizations and being equivalent to ARIX-based GPC. To overcome this gap, we propose the novel approach presented in the next section.

A. GPC with colored noise

Without loss of generality we are going to assume $d = 1$ and $N_u = N_y$, so a complete trajectory for predictions is generated for $j = 1, \ldots, N_y$ and then it can be truncated as required as we are going to present later on.

We start by converting the ARIMAX model into an observable canonical realization in the state-space, given by:

$$\begin{aligned} \mathbf{x}(k) &= \mathbf{A}\mathbf{x}(k-1) + \mathbf{B}\Delta\mathbf{u}(k-1) + \mathbf{\Gamma}\mathbf{w}(k-1), \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{w}(k), \end{aligned} \tag{19}$$

$$\mathbf{A} = \begin{bmatrix} -\bar{a}_1 & & \\ \vdots & \mathbf{I}_{n_a} & \\ -\bar{a}_{n_{\bar{a}}} & \mathbf{0}_{1\times n_a} & \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} b_0 \\ \vdots \\ b_{n_{\bar{a}}} \end{bmatrix}, \tag{20}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} c_1 - \bar{a}_1 \\ \vdots \\ c_{n_{\bar{a}}} - \bar{a}_{n_{\bar{a}}} \end{bmatrix}, \tag{21}$$

where $\mathbf{I}$ and $\mathbf{0}$ are an identity and a null matrix, respectively, and the new barred elements seen in $\mathbf{A}$ and $\mathbf{\Gamma}$ comes from the ARIMAX $n_{\bar{a}} = n_a + 1$ augmented polynomial

$$\bar{A}(q^{-1}) = \Delta A(q^{-1}) = 1 + \bar{a}_1 q^{-1} + \cdots + \bar{a}_{n_{\bar{a}}} q^{-n_{\bar{a}}}. \tag{22}$$

Considering the proofs obtained by [16] for GMVC's Minimum Variance Predictor polynomials, then the same must apply to the GPC as well, as it follows:

$$E_j(q^{-1}) = 1 + \mathbf{C}\mathbf{A}^0\mathbf{\Gamma}q^{-1} + \mathbf{C}\mathbf{A}^1\mathbf{\Gamma}q^{-2} + \cdots + \mathbf{C}\mathbf{A}^{(j-2)}\mathbf{\Gamma}q^{-(j-1)}, \tag{23}$$

$$F_j(q^{-1}) = \begin{bmatrix} q^0 & q^{-1} & \cdots & q^{-n_a} \end{bmatrix}\mathbf{A}^{(j-1)}\mathbf{\Gamma}. \tag{24}$$

Now that the solution for $E_j(q^{-1})$ and $F_j(q^{-1})$ is feasible in the colored noise case, the predictor is further developed as follows:

$$\hat{y}(k+j|k) = G_j(q^{-1})\Delta u_f(k+j-1) + F_j(q^{-1})y_f(k), \tag{25}$$

with $G_j(q^{-1}) = E_j(q^{-1})B(q^{-1})$ so that for $j = 1, \ldots, N_y$ this becomes

$$G_1(q^{-1}) = g_0 + \underbrace{\bar{g}_{1,0}q^{-1} + \cdots + \bar{g}_{1,n_b}q^{-n_b}}_{n_b \text{ terms}}$$

$$G_2(q^{-1}) = g_0 + g_1q^{-1} + \underbrace{\bar{g}_{2,0}q^{-2} + \cdots + \bar{g}_{2,n_b}q^{-(n_b+1)}}_{n_b \text{ terms}}$$

$$\vdots$$

$$G_{N_y}(q^{-1}) = g_0 + g_1 q^{-1} + \cdots + g_{N_y-1} q^{-(N_y-1)} + \underbrace{\bar{g}_{N_y,0} q^{-N_y} + \cdots + \bar{g}_{N_y,n_b} q^{-(n_b+N_y-1)}}_{n_b \text{ terms}}. \tag{26}$$

The last n_b terms of every $G_j(q^{-1})$ will operate on $\Delta u_f(k)$ regressors $\underleftarrow{\mathbf{\Delta_{u_f}}}$, from $\Delta u_f(k-1)$ until $\Delta u_f(k-n_b)$, whilst the first and up to $N_y - 1$ elements of $G_j(q^{-1})$ will operate on the control predictions, $\underrightarrow{\mathbf{\Delta_{u_f}}}$, i.e., $\Delta u_f(k)$ up to $\Delta u_f(k+N_y-1)$. Rewriting the GPC predictor in a matrix-vector form simplifies the notation as follows:

$$\hat{\mathbf{y}} = \mathbf{G} + \bar{\mathbf{G}} \underleftarrow{\mathbf{\Delta_{u_f}}} + \mathbf{F} \underleftarrow{\mathbf{y_f}} = \mathbf{G} + \mathbf{f_r}, \tag{27}$$

$\mathbf{f_r}$ is the Free Response Model,

$$\mathbf{G} = \begin{bmatrix} g_0 & 0 & 0 & \cdots & 0 \\ g_1 & g_0 & 0 & \ddots & \vdots \\ g_2 & g_1 & g_0 & \ddots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ g_{N_y-1} & g_{N_y-2} & g_{N_y-3} & \cdots & g_0 \end{bmatrix}, \tag{28}$$

where $\mathbf{G}$ is the lower triangular matrix of the Toeplitz matrix constructed based on the vector formed by the first $N_y - 1$ elements of $G_{N_y}(q^{-1})$ shown in Eq. (26), while $\bar{\mathbf{G}}$ is formed by the extracted last n_b terms of every $G_j(q^{-1}) = E_j(q^{-1})B(q^{-1})$ for $j = 1, \ldots, N_y$, which leads to

$$\bar{\mathbf{G}} = \begin{bmatrix} \bar{g}_{1,0} & \cdots & \bar{g}_{1,n_b} \\ \bar{g}_{2,0} & \cdots & \bar{g}_{2,n_b} \\ \vdots & \cdots & \vdots \\ \bar{g}_{N_y,0} & \cdots & \bar{g}_{N_y,n_b} \end{bmatrix}. \tag{29}$$

And the $\mathbf{F}$ matrix is obtained as follows:

$$\mathbf{F} = \begin{bmatrix} \left(\mathbf{A}^0\mathbf{\Gamma}\right)^{\mathbf{T}} \\ \left(\mathbf{A}^1\mathbf{\Gamma}\right)^{\mathbf{T}} \\ \vdots \\ \left(\mathbf{A}^{N_y-1}\mathbf{\Gamma}\right)^{\mathbf{T}} \end{bmatrix} = \begin{bmatrix} f_{1,0} & \cdots & f_{1,n_a} \\ f_{2,0} & \cdots & f_{2,n_a} \\ \vdots & \cdots & \vdots \\ f_{N_y,0} & \cdots & f_{N_y,n_a} \end{bmatrix}. \tag{30}$$

Vectors for predictions and regressors of the filtered output and control, in the colored noise case, are defined as:

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{y}(k+1|k) \\ \vdots \\ \hat{y}(k+N_y|k) \end{bmatrix}, \ \underleftarrow{\mathbf{y_f}} = \begin{bmatrix} y_f(k-1) \\ \vdots \\ y_f(k-n_a) \end{bmatrix}, \tag{31}$$

$$\underrightarrow{\mathbf{\Delta_{u_f}}} = \begin{bmatrix} \Delta u_f(k) \\ \vdots \\ \Delta u_f(k+N_y-1) \end{bmatrix}, \underleftarrow{\mathbf{\Delta_{u_f}}} = \begin{bmatrix} \Delta u_f(k-1) \\ \vdots \\ \Delta u_f(k-n_b) \end{bmatrix} \tag{32}$$

From this generalized structure for the preditor, based on $j = 1, \ldots, N_y$, the developed matrices can be truncated in order to comply with systems with time delays, $j = d, \ldots, N_y$, and when $N_u < N_y$ is desired. In this case, the first $d-1$ lines of $\hat{\mathbf{y}}$, $\mathbf{G}$, $\bar{\mathbf{G}}$, $\mathbf{F}$ are eliminated, only the first N_u columns of $\mathbf{G}$ are utilized and $\underrightarrow{\mathbf{\Delta_{u_f}}}$ goes until $\Delta u_f(k+N_u-1)$.

B. GPC control law in the colored noise case

By using the Minimum Variance Predictor, developed previously, the GPC optimization problem can be rewritten in the following vector-matrix form:

$$\min_{\underrightarrow{\mathbf{\Delta_{u_f}}}} \ J = [\hat{\mathbf{y}} - \mathbf{r}]^{\mathbf{T}} [\hat{\mathbf{y}} - \mathbf{r}] + \lambda \underrightarrow{\mathbf{\Delta_{u_f}^T}} \underrightarrow{\mathbf{\Delta_{u_f}}}, \tag{33}$$

in which the trajectory of future references is defined as

$$\mathbf{r^T} = \begin{bmatrix} r(k+1) & r(k+2) & \cdots & r(k+N_y) \end{bmatrix}. \tag{34}$$

Substituting the GPC predictor in Eq. (27) into J,

$$J = \left[\mathbf{G}\underrightarrow{\mathbf{\Delta_{u_f}}} + \mathbf{f_r} - \mathbf{r}\right]^{\mathbf{T}} \left[\mathbf{G}\underrightarrow{\mathbf{\Delta_{u_f}}} + \mathbf{f_r} - \mathbf{r}\right] + \lambda \underrightarrow{\mathbf{\Delta_{u_f}^T}} \underrightarrow{\mathbf{\Delta_{u_f}}}, \tag{35}$$

$$J = \underrightarrow{\mathbf{\Delta_{u_f}^T}} \left[\mathbf{G^T G} + \lambda \mathrm{I}\right] \underrightarrow{\mathbf{\Delta_{u_f}}} - 2\underrightarrow{\mathbf{\Delta_{u_f}^T}} \mathbf{G^T} [\mathbf{r} - \mathbf{f_r}] + [\mathbf{r} - \mathbf{f_r}]^{\mathbf{T}} [\mathbf{r} - \mathbf{f_r}]. \tag{36}$$

The control sequence $\underrightarrow{\mathbf{\Delta_{u_f}}}$ that minimize J is obtained by solving

$$\min_{\underrightarrow{\mathbf{\Delta_{u_f}}}} \ J = \frac{\partial J}{\partial \underrightarrow{\mathbf{\Delta_{u_f}}}} = 0, \tag{37}$$

that results in

$$\underrightarrow{\mathbf{\Delta_{u_f}}} = \left[\mathbf{G^T G} + \lambda \mathbf{I}\right]^{-1} \mathbf{G^T} [\mathbf{r} - \mathbf{f_r}] = \mathbf{K} [\mathbf{r} - \mathbf{f_r}]. \tag{38}$$

The matrix gain of the GPC is then defined as

$$\mathbf{K} = \left[\mathbf{G^T G} + \lambda \mathrm{I}\right]^{-\mathbf{1}} \mathbf{G^T} = \begin{bmatrix} \mathbf{k_1} & \cdots & \mathbf{k_{N_y}} \end{bmatrix}^{\mathbf{T}}. \tag{39}$$

Then, the receding horizon control is employed and just the first term of the control sequence is applied, and

$$\Delta u_f(k) = \mathbf{k_1} \left[\begin{pmatrix} r(k+d) \\ \vdots \\ r(k+N_y) \end{pmatrix} - \bar{\mathbf{G}} \begin{pmatrix} \Delta u_f(k-1) \\ \vdots \\ \Delta u_f(k-n_b) \end{pmatrix} - \mathbf{F} \begin{pmatrix} y_f(k-1) \\ \vdots \\ y_f(k-n_a) \end{pmatrix} \right]. \tag{40}$$

In order to implement the control increment $\Delta u(k)$ based on $\Delta u_f(k)$, the relation shown in Eq. (18) is used:

$$\Delta u(k) = \Delta u_f(k) + c_1 \Delta u_f(k-1) + \cdots + c_{n_c} \Delta u_f(k-n_c), \tag{41}$$

and, finally, the control signal is built as follows.

$$u(k) = u(k-1) + \Delta u(k) \tag{42}$$

III. 3-DoF Spacecraft Emulation

The proposed spacecraft emulation is illustrated in Figure 1. In a real setup, the quadcopter would be the physical part, while the reference model and the controller would be part of a computer software. Specifically to our future experimental setup, the AR.Drone 2.0 quadcopter model, from Parrot Drones SAS, is considered. The software part runs within Mathwork's Matlab/Simulink and communicates with the quadcopter by means of a WiFi TCP/IP connection with User Datagram Protocol, operating at the sampling period of $T_s = 65$ ms or sampling frequency $f_s \approx 15.3846$ Hz, which limits our operational range up to the Nyquist frequency at 7.6923 Hz.

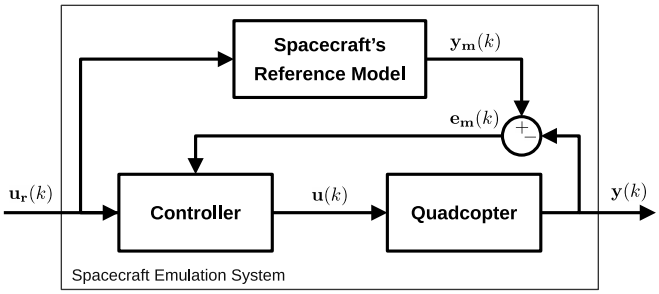

Fig. 1. Model-following closed-loop control to force the quadcopter to behave as the spacecraft's reference model.

The experimental setup will be comprised of the original payload of the AR.Drone 2.0, in which its embedded inertial measurement unit works along with an ultrasound-based altimeter and a camera-based optical flow velocity sensor, measuring the altitude (m), the longitudinal or forward/backward velocity (m/s), and the lateral or right/left velocity (m/s), given in the body-axes reference frame.

The quadcopter is assumed to be an asymptotically stable multi-input multi-output linear time varying system, described by

$$\mathbf{A}_k\mathbf{y}(k) = \mathbf{B}_k\mathbf{u}(k-d) + \mathbf{C}_k\mathbf{v}(k). \tag{43}$$

$\mathbf{A}_k$, $\mathbf{B}_k$ and $\mathbf{C}_k$ are diagonal matrices of polynomial functions of three ARIMAX subsystems, as shown in Eq. (2), related to the output $\mathbf{y}(k)$ which is a column vector comprised of the body-axes translational dynamics defined in the North-East-Down coordinate system [17], so that $y_x(k)$, $y_y(k)$, $y_z(k)$, are, respectively, the longitudional velocity along the x-axis, the lateral velocity along the y-axis and the vertical position along the z-axis. The input $\mathbf{u}(k)$ is also a column vector, comprised of dimensionless actuator thrust commands in the range $[-1; 1] \in \mathbb{R}^3$, for $u_x(k)$, $u_y(k)$, $u_z(k)$. These paired input to output system is assumed to be coupled but uncertain, and possible mutual interference is considered as the stochastic disturbance process $\mathbf{v}(k)$.

The controller of the emulation system is a function of the spacecraft's and the quadcopter's dynamics, whose objective, based on the block diagram shown in Figure 1, is defined as:

$$\mathbf{u}(k) = \mathbf{H}_k(q^{-1})\mathbf{e_m}(k). \tag{44}$$

This controller $\mathbf{H}_k(q^{-1})$ must act by means of this intermediate control vector $\mathbf{u}(k)$ in order to eliminate the model-following error, $\mathbf{e_m}(k) = \mathbf{y_m}(k) - \mathbf{y}(k)$, between the quadcopter closed-loop behavior and the spacecraft's reference model, the latter defined as:

$$\mathbf{A_m}(q^{-1})\mathbf{y_m}(k) = \mathbf{B_m}(q^{-1})\mathbf{u}(k-1). \tag{45}$$

This system must incorporate dual discrete-time integrators in order to emulate the spacecraft's dynamics in free-flight for short-time and short-distance intervals [13].

To simplify notation, henceforth the controller synthesis is presented for the SISO case and can be replicated for the three axis. The spacecraft's reference model with the double discrete integrators and $T_s = 65$ ms is given by:

$$\frac{Y_m(z)}{U_r(z)} = \left(\frac{1}{M}\right)\frac{\left(0.002113 + 0.002112z^{-1}\right)z^{-1}}{(1 - 2z^{-1} + z^{-2})}. \tag{46}$$

These parameters are related to the zero-order-hold transformation of the double integrators system, defined in the continuous s frequency domain as follows:

$$\frac{Y_m(s)}{U_r(s)} = \left(\frac{1}{M}\right)\frac{1}{s^2}, \tag{47}$$

where M is the spacecraft's mass in Kilogram. Note, however, that this is a positional reference model and so it can only be applied to the altitude control problem within this particular application with the AR.Drone. For the velocities along the x and y axes the following velocity reference model is considered:

$$\frac{sY_m(s)}{U_r(s)} = \left(\frac{1}{M}\right)\frac{1}{s}, \tag{48}$$

in which the zero-order-hold equivalent at $T_s = 65$ ms is given as follows:

$$\frac{\Delta Y_m(z)}{U_r(z)} = \left(\frac{1}{M}\right)\frac{0.065z^{-1}}{1 - z^{-1}}. \tag{49}$$

The main control problem is to find a control solution $u(k)$ so that $y(k) \to y_m(k)$ for all $k > 0$, governed by the emulator's input $u_r(k)$. To better illustrate this problem, let us consider a simple example with a proportional control solution of the following form:

$$u(k) = K_p e_m(k) = K_p\left[\frac{B_m(q^{-1})}{A_m(q^{-1})}u_r(k-1) - y(k)\right], \tag{50}$$

where K_p is a proportional gain.

Considering Eq. (50) we can remark that the model-following control performance can be enhanced by boosting the K_p gain. However, if $y(k)$ is noisy, then $u(k)$ chattering is increased and could compromise the quadcopter's mechanical and electrical components.

Differently from proportional control, the GPC is expected to workaround the noise problem in $y(k)$ and can take advantage of the spacecraft model working as a reference signal as shown in Eq. (50), since such a reference can be precisely predicted using Eq. (46). Thus, the problem in Eq. (50) can be optimally solved by the GPC as in Eq. (33), by assuming the vector of future references constructed based on the spacecraft's model response,

$$\mathbf{r^T} = \left[\ y_m(k+1) \quad \cdots \quad y_m(k+N_y)\ \right]. \tag{51}$$

The GPC can also workaround the quadcopter's noisy output, $y(k)$, since such a signal is not directly fed back to the

controller but is in fact filtered by GPC's minimum variance predictor, $\hat{\mathbf{y}}$, shown in Eq. (27). This, of course, generates the dependence on the knowledge of the quadcopter's model, which is covered next.

IV. Quadcopter Parametric Estimation

For real-time adaptation and optimization of the GPC we must focus on real-time parametric estimation based on most recent data, whilst for the purpose of simulating the quadcopter for the proper assessment of the ARIMAX-based GPC, we must focus on simulation models best suited to represent the quadcopter system in a more general form, considering a longer data set representing the quadcopter input/output average behavior. Thus, the non-recursive Least-Squares estimator was adopted and is discussed next.

Considering the ARIMAX model shown in Eq. (1), let us assume the following parametric estimation problem based on the following estimator:

$$\hat{\mathbf{h}}(k+1) = \hat{\mathbf{h}}(k) + \mathbf{L}(k)\left[y(k) - \hat{y}(k)\right], \tag{52}$$

$$\hat{y}(k) = \mathbf{C}(k)\hat{\mathbf{h}}(k). \tag{53}$$

The main problem is to optimally estimate the parameter vector $\hat{\mathbf{h}}(k)$ that minimizes the error $v(k) = \frac{w(k)}{\Delta} = y(k) - \hat{y}(k)$, in which $\hat{\mathbf{h}}(k)$ and $\mathbf{C}(k)$ are constructed based on parameters and signal regressors of

$$y(k) = \frac{B_k(q^{-1})}{A_k(q^{-1})}u(k-d) + \frac{C_k(q^{-1})}{A_k(q^{-1})}v(k), \tag{54}$$

so that

$$\begin{aligned} y(k) = &\begin{bmatrix} -y(k-1) & \cdots & -y(k-n_a) \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_{n_a} \end{bmatrix} \\ &+ \begin{bmatrix} u(k-d) & \cdots & u(k-n_b) \end{bmatrix} \begin{bmatrix} b_0 \\ \vdots \\ b_{n_b} \end{bmatrix} \\ &+ \begin{bmatrix} v(k-1) & \cdots & v(k-n_c) \end{bmatrix} \begin{bmatrix} c_1 \\ \vdots \\ c_{n_c} \end{bmatrix} + v(k) \end{aligned} \tag{55}$$

which leads to

$$y(k) = \mathbf{C}(k)\hat{\mathbf{h}}(k) + v(k). \tag{56}$$

The non-recursive Least-Squares parametric estimator, in the ARIMAX case, is solved based on a previously identified ARX model [18], and it works on the data by processing the whole data set of N samples all at once.

By assuming a ARX model with temporary $\bar{A}(q^{-1})$ and $\bar{B}(q^{-1})$ polynomials,

$$y(k) = \frac{\bar{B}(q^{-1})}{\bar{A}(q^{-1})}u(k-d) + v(k), \tag{57}$$

a similar representation as the one shown in Eq. (56) applies, however, not solely for k, but for the whole data set given by $k = 0, \ldots, N$, that is:

$$\underbrace{\begin{bmatrix} y(0) \\ \vdots \\ y(N) \end{bmatrix}}_{\mathbf{Y}_{ls}} = \underbrace{\begin{bmatrix} \mathbf{C}(0) \\ \vdots \\ \mathbf{C}(N) \end{bmatrix}}_{\mathbf{C}_{ls}} \bar{\mathbf{h}} + \underbrace{\begin{bmatrix} v(0) \\ \vdots \\ v(N) \end{bmatrix}}_{\mathbf{v}_{ls}}. \tag{58}$$

The main problem is to estimate this temporary vector $\bar{\mathbf{h}}$, containing the parameters of $\bar{A}(q^{-1})$ and $\bar{B}(q^{-1})$, given by:

$$\bar{\mathbf{h}} = \left(\mathbf{C}_{ls}^{\mathbf{T}}\mathbf{C}_{ls}\right)^{-\mathbf{1}}\mathbf{C}_{ls}^{\mathbf{T}}\mathbf{Y}_{ls}. \tag{59}$$

To estimate the ARIMAX model a second turn of the non-recursive Least-Squares estimator is used, now based on the ARIMAX vector of regressors as shown in Eq. (55), by applying the estimated noise vector $\mathbf{v}_{ls}$ so to determine all three (final) polynomials, $A(q^{-1})$, $B(q^{-1})$, $C(q^{-1})$.

This form of parametric estimation focusing on determining simulation models is experimental and supervised by the designer. Thus, the determination of the model's order and how it fits a certain model-validation data set is a human decision, generally quantified and assisted by some sort of "size of a signal" method, such as the mean of the squared estimation error or the normalized root mean squared error (NRMSE), in which the latter was adopted in this work, given by

$$J_{NRMSE} = 1 - \frac{\sqrt{\sum_{k=0}^{N}\left|y(k) - \hat{y}(k)\right|^2}}{\sqrt{\sum_{k=0}^{N}\left|y(k) - \bar{y}\right|^2}}, \tag{60}$$

with $\bar{y}$ representing the mean average of $y(k)$.

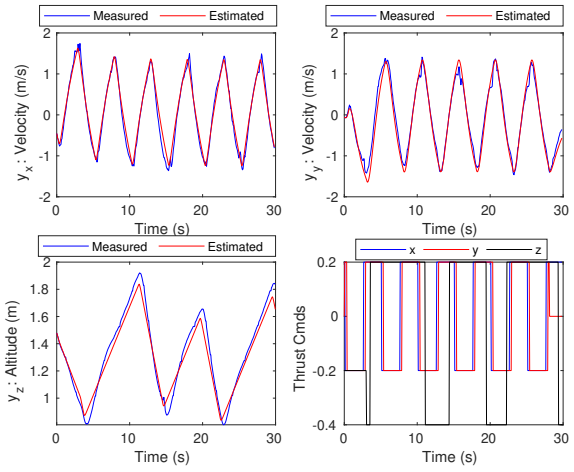

Fig. 2. Longitudinal and lateral velocities, and altitude models validation.

In Figure 2 it is shown the validation results of the ARIMAX simulation models obtained for $y_x(k)$ and $y_y(k)$ velocities, and $y_z(k)$ altitude, in which the NRMSE achieved was 83%, 81% and 78%, respectively, where the closer to 100% the better. These results were based on indoor registered flight data of the quadcopter. Half of the data was

used for the parametric estimation procedure and the other half for validation. The complete data set is available at [19].

The estimated ARIMAX polynomials were as follows:

- Longitudinal velocity model polynomials:

$$\begin{aligned} A(q^{-1}) &= 1 - 1.5870q^{-1} + 0.5939q^{-2} \\ B(q^{-1}) &= 0.0176 - 0.1625q^{-1} \\ C(q^{-1}) &= 1 - 0.1794q^{-1} + 0.1739q^{-2} \end{aligned} \tag{61}$$

- Lateral velocity model polynomials:

$$\begin{aligned} A(q^{-1}) &= 1 - 1.8328q^{-1} + 0.8387q^{-2} \\ B(q^{-1}) &= -0.0318 + 0.0966q^{-1} \\ C(q^{-1}) &= 1 - 0.5561q^{-1} + 0.5536q^{-2} \end{aligned} \tag{62}$$

- Altitude model polynomials:

$$\begin{aligned} A(q^{-1}) &= 1 - 1.4452q^{-1} + 0.4454q^{-2} \\ B(q^{-1}) &= 0.0087 + 0.0157q^{-1} \\ C(q^{-1}) &= 1 - 0.0065q^{-1} + 0.0063q^{-2} \end{aligned} \tag{63}$$

The discrete time delay observed within the registered flight data was $d = 3$, which is equivalent to a 195 ms delay. Also, the estimated variances for the process noise $v(k)$ were: $\sigma^2_{v_x} = 0.0332$, $\sigma^2_{v_y} = 0.0644$, $\sigma^2_{v_z} = 0.0359$.

In Figure 3 it is shown the validation results but this time including the colored process noise. In total, 1000 Monte Carlo simulations were realized and they depict the possible dispersion that can occur (at least probabilistically) to the longitudinal and lateral velocities, and to the altitude. In Figure 3, every red line tells a possible "story" (i.e., events that follow the influence of a new Gaussian sequence), whilst the blue line in the middle is one of the stories when there is no colored process noise acting on these systems.

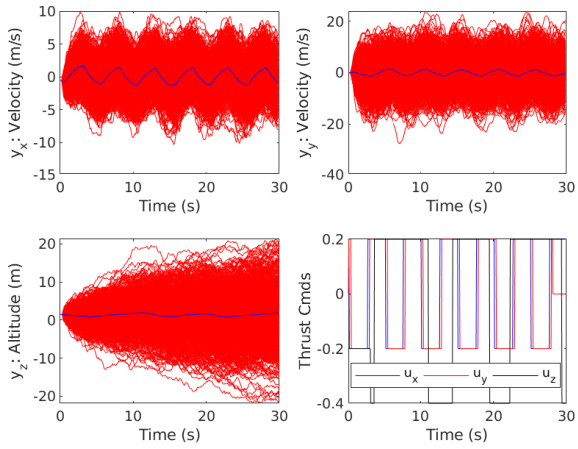

Fig. 3. The dispersion of the longitudinal and lateral velocities, and altitude, when the estimated stochastic uncertainties are applied over 1000 Monte Carlo simulations. Every red line tells a "possible story", and, among these stories, the one in blue that was registered in flight.

Theoretically, the GPC, as proposed in [1], was designed to deal with the colored noise case as depicted in Figure 3, by incorporating the appropriate ARIMAX model into the design and this is investigated next by using the GPC design technique proposed in this work.

V. Simulation Examples

In the simulation examples covered in this section, the system's models are assumed to be known, so that no real-time parametric estimation is adopted and the GPC is optimized both in the ARIX and ARIMAX cases in order to investigate and analyze the (unfortunately small) contribution in the proposed colored noise case design for the modeled quadcopter. The mass of the emulated spacecraft reference model adopted was $M = 10$ kg. The GPC was tuned with $N_y = 12$, $N_u = 3$, $\lambda = 50$ for the longitudinal and lateral velocities control, and $N_y = 12$, $N_u = 3$, $\lambda = 1$ for the altitude control.

It is important to remark that the tuning of the prediction horizons, as well as the control energy weighting factor of GPC, were found by trial and error while observing the output signals, looking for an offset-free model-following performance achieved by a realizable control signal (within the range of the command thrusters, $[-1, 1]$). Thus, despite other possible combination of tuning parameters could be achieved, such decision factor does not affect the presented results, since the tuning will be the same for both ARIX- and ARIMAX-based methods, and does not change their difference in dealing with the colored process noise and their capability to achieve offset-free model-following control.

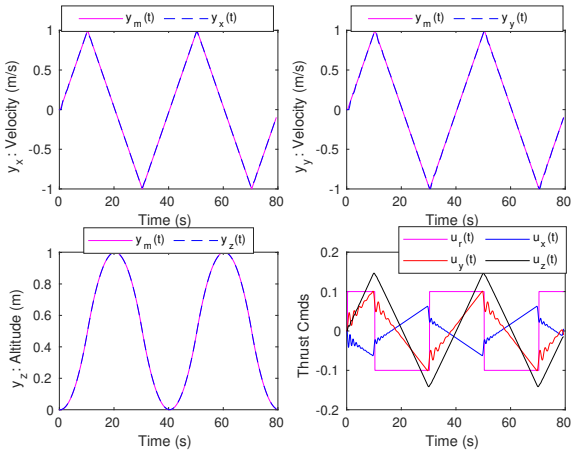

Fig. 4. Longitudinal and lateral velocities, and altitude model-following using GPC. Simulations without the colored process noise. The reference $u_r(t)$ for the altitude control is the one presented, whilst for the velocity cases it was omitted since its amplitude was 10 times higher.

In Figure 4 it is shown the simulation results when there is no process noise, thus ARIX and ARIMAX behave in the same way. In this figure it is possible to observe that the GPC can achieve a good model-following performance since both velocities and the altitude were tracked flawlessly. Also, the required thrust commands were all within the saturation limits. These results were obtained by assuming square waves as the reference model's input $u_r(k)$, in which the one adopted for the velocities control loops was omitted from the plot since its unitary amplitude would have diminished the details of the thrust control signals shown.

In Figure 5 the longitudinal model-following control results for ARIMAX-based versus ARIX-based GPC when the colored process noise is considered is presented. The other results for the y-axis and z-axis were omitted due

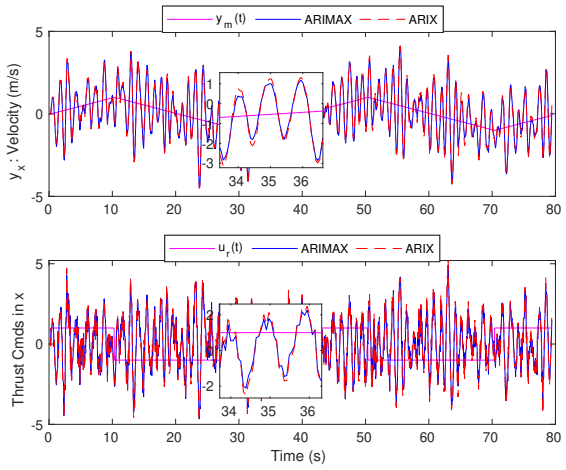

Fig. 5. Longitudinal velocity model-following control simulations with colored process noise in ARIMAX-based versus ARIX-based GPC.

to space limitations but they follow Figure 5 similarly. The control signal chattering is beyond the acceptable threshold since it trespasses the saturation limit and within this particular "story", it is possible to observe a small reduction in the variability of input and output signals in favor of the ARIMAX-based case, also confirmed numerically, as the output and input variances were respectively 4.38% and 10.59% smaller than the ARIX-based case. This also holds after 1000 "stories", as the average variance percentage reduction, for the output and control, was 3.35% and 9.37%, respectively.

Unfortunately, the variance reduction observed for the longitudinal velocity control problem could not be observed in the same way in the lateral and altitude cases. In the lateral model-following control, the average variance reduction after 1000 simulations resulted in an increase of 20.31% and a reduction of 9.77%, respectively, with respect to the output and control variances.

For what concerns the altitude model-following, the output and control variances, after 1000 simulations, resulted in a slightly worsening (0.09%) and improvement (0.11%), respectively. Thus, in the lateral control the ARIMAX-based GPC increased the output variance considerably, while in the altitude problem it is possible to assume a draw.

VI. Conclusions

In this work the algorithm to design and implement the ARIMAX-based GPC was presented in detail, filling this gap in the GPC's literature by proposing a systematic way to incorporate the $C(q^{-1})$ polynomial into the design for any long-range prediction horizon requirement. Also, it was presented the prospects for an experimental setup of a 3-Degrees-of-Freedom spacecraft cyber-physical emulator.

Based on the results of the considered case study and despite not achieving a significant contribution in reducing the control signal chattering in the colored noise case, the outcome is valuable in the sense that this could diminish the expectations on GPC, by researchers and practitioners, in terms of how it can handle the colored noise disturbances.

Additionally, we assessed the presented ARIMAX- and ARIX-based cases assuming $v(k)$ as an output noise. In this scenario the ARIMAX-based GPC outperformed the ARIX case. In the longitudinal velocity, the output and control variances were, respectively, 1.92% and 12.78% smaller. The same was observed for the lateral velocity, with 1.80% and 36.11% smaller, and for the altitude, 0.62% and 0.47% smaller. Thus, despite not directly intended for it, the proposed ARIMAX-based design may also allow further investigations in the more conventional output noise case.

References

[1] D. Clarke, C. Mohtadi, and P. Tuffs, "Generalized predictive control: Part i. the basic algorithm," *Automatica*, vol. 23, no. 2, pp. 137 – 148, 1987.

[2] T. G. Ivanco, H. Kang, A. R. Kreshock, R. P. Thornburgh, and B. Newman, "Generalized predictive control for active stability augmentation and vibration reduction on an aeroelastic tiltrotor model," in *AIAA SCITECH 2022 Forum*, 2022.

[3] X. Zhou, J. Huang, F. Lu, J. Liu, and C. Wang, "Hnn-based generalized predictive control for turbofan engine direct performance optimization," *Aerospace Science and Technology*, vol. 112, p. 106602, 2021.

[4] A. Thamallah, A. Sakly, and F. M'Sahli, "Multi-objective generalized predictive control for switching systems under unknown switching sequences," *ISA Transactions*, vol. 126, pp. 144–159, 2022.

[5] R. A. Franco, Álisson A. Cardoso, G. L. Filho, and F. H. Vieira, "Mimo auto-regressive modeling-based generalized predictive control for grid-connected hybrid systems," *Computers & Electrical Engineering*, vol. 97, p. 107636, 2022.

[6] O. Asvadi-Kermani, B. Felegari, and H. Momeni, "Adaptive constrained generalized predictive controller for the pmsm speed servo system to reduce the effect of different load torques," *e-Prime - Advances in Electrical Engineering, Electronics and Energy*, vol. 2, p. 100032, 2022.

[7] Y. Cheng, L. Dai, A. Li, Y. Yuan, and Z. Chen, "Active disturbance rejection generalized predictive control of a quadrotor uav via quantitative feedback theory," *IEEE Access*, vol. 10, pp. 37 912–37 923, 2022.

[8] H. Wang and Z. S. Li, "Multi-area load frequency control in power system integrated with wind farms using fuzzy generalized predictive control method," *IEEE Transactions on Reliability*, pp. 1–11, 2022.

[9] A. Silveira, R. Trentini, A. Coelho, R. Kutzner, and L. Hofmann, "Generalized minimum variance control under long-range prediction horizon setups," *ISA Transactions*, vol. 62, no. 1, pp. 325 – 332, 2016.

[10] H. Chen and K. Ho, "Development of on-orbit servicing, assembly and manufacturing creates new capabilities in spacefaring operations," http://encr.pw/yNgbN, 2022, [Online; accessed 01-Feb-2023].

[11] O. B. Iskender, K. V. Ling, L. Simonini, M. Schlotterer, D. Seelbinder, S. Theil, and J. M. Maciejowski, "Dual quaternion based autonomous rendezvous and docking via model predictive control," in *70th International Astronautical Congress, IAC 2019, of the International Astronautical Federation, IAF*, October 21 - 25, Washington, United States 2019.

[12] E. S. Agency, "Smartphone app turns home drone into spacecraft," http://esa.int/Enabling_Support/Space_Engineering_Technology/Smartphone_app_turns_home_drone_into_spacecraft, 2013, [Online; accessed 10-Dec-2022].

[13] B. H. Kang, F. Y. Hadaegh, and D. P. Scharf, "On the validity of the double integrator approximation in deep space formation flying," 2002. [Online]. Available: https://hdl.handle.net/2014/37196

[14] E. F. Camacho and C. A. Bordons, *Model Predictive Control*, 2nd ed. London: Springer-Verlag, 2007.

[15] L. Wang, *Model Predictive Control System Design and Implementation Using MATLAB*, 1st ed. Springer Publishing Company, Incorporated, 2009.

[16] A. S. Silveira and A. A. R. Coelho, "Generalised minimum variance control state-space design," *Control Theory and Applications, IET*, vol. 5, no. 15, pp. 1709–1715, 13 2011.

[17] B. L. Stevens, F. L. Lewis, and E. N. Johnson, *Aircraft Control and Simulation: dynamics, controls design, and autonomous systems*, 3rd ed. John Wiley & Sons, Inc., 2016.

[18] A. Silveira, A. Silva, A. Coelho, J. Real, and O. Silva, "Design and real-time implementation of a wireless autopilot using multivariable predictive generalized minimum variance control in the state-space," *Aerosp Sci Technol*, vol. 105, 2020.

[19] LACOS-UFPA, "Ar.drone 2.0 data set for system identification," https://lacos.ufpa.br/plantas/ardrone/ardrone.html, [Accessed 05-Mar-2023].