Asociación Argentina
de Mecánica Computacional

Mecánica Computacional Vol XXXVII, págs. 1861-1870 (artículo completo)
A. Cardona, L. Garelli, J.M. Gimenez, P.A. Kler, S. Márquez Damián, M.A. Storti (Eds.)
Santa Fe, 5-7 Noviembre 2019

TOWARDS AUTONOMY IN UNMANNED VEHICLES USING RECEDING HORIZON STRATEGIES

Marina H. Murillo, Guido M. Sánchez, Nestor N. Deniz, Lucas M. Genzelis and Leonardo L. Giovanini

Research Institute for Signals, Systems and Computational Intelligence, sinc(i), UNL-CONICET, Ciudad Universitaria UNL, 4to piso FICH, S3000 Santa Fe, Argentina, mmurillo@sinc.unl.edu.ar, http://sinc.unl.edu.ar/

Keywords: Model predictive control, Moving horizon estimation, Receding horizon techniques, Unmanned vehicle.

Abstract. In this article we propose to use receding horizon strategies, like model predictive control (MPC) and moving horizon estimation (MHE), to design guidance, navigation and path-planning tasks, which play an essential role in autonomy of unmanned vehicles. As we propose to design these tasks using MPC and MHE, the physical and dynamical constraints can be included at the design stage, thus leading to optimal and feasible results. In order to evaluate the performance of the proposed framework, we have used Gazebo simulator in order to drive a Jackal unmanned ground vehicle (UGV) along a desired path computed by the path-planning module. The results we have obtained are successful as the estimation and guidance errors are small and the Jackal UGV is able to follow the desired path satisfactorily and it is also capable to avoid the obstacles which are in its way.

1 INTRODUCTION

Autonomous unmanned vehicles (UVs) are capable to perform a set of predefined tasks without human interaction. In order to do this, these vehicles should be able of sensing their environment to measure their positions, velocities and detect if there are obstacles in their proximity so as to navigate through the environment and achieve the targeted positions to perform the tasks. The execution of these tasks involves the adquisition and processing of a wide variety of sensors, as well as the solution of coupled optimization problems at different time scales. In this context, these activities can be organized in three interrelated tasks (See Fig. 1): i) **Guidance**, ii) **Navigation**, and iii) **Path-planning**.

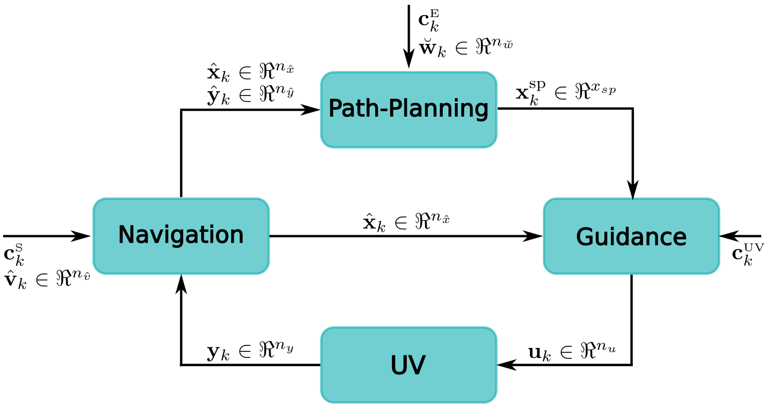

$\mathbf{c}_k^{\mathrm{E}}$
$\breve{\mathbf{w}}_k \in \Re^{n_{\breve{w}}}$
$\hat{\mathbf{x}}_k \in \Re^{n_{\hat{x}}}$
$\hat{\mathbf{y}}_k \in \Re^{n_{\hat{y}}}$
Path-Planning
$\mathbf{x}_k^{\mathrm{sp}} \in \Re^{x_{sp}}$
Navigation
$\hat{\mathbf{x}}_k \in \Re^{n_{\hat{x}}}$
Guidance
$\mathbf{c}_k^{\mathrm{S}}$
$\hat{\mathbf{v}}_k \in \Re^{n_{\hat{v}}}$
$\mathbf{c}_k^{\mathrm{UV}}$
$\mathbf{y}_k \in \Re^{n_y}$
UV
$\mathbf{u}_k \in \Re^{n_u}$

Figure 1: Tasks interaction scheme

The guidance task refers to the control of the position, attitude and velocity of a vehicle along a pre-defined path. To do this, a suitable control algorithm should be used in order to compute which actuators' deflections and/or motors' speeds the vehicle should have so as to accomplish a certain mission. A wide range of control techniques have been tested, ranging from classical to modern ones. Proportional-integral-derivative (PID) control is indeed the most popular control technique used to control UVs. This is mainly due to its simplicity and because its parameters are easy to adjust. Several works dealing with PID control of UV have been found in the specialized literature (Zhao et al., 2012). However, when using a PID controller a decoupled version of the mathematical model of the UV is used. This may lead to unexpected results in the presence of disturbances and even limit its performance due to unmodelled dynamics. Another technique that has been successfully applied to control UVs is the linear quadratic regulator (LQR) control algorithm (Khamseh and Janabi-Sharifi, 2017). Within optimal control techniques model predictive control (MPC) can be found. Unlike LQR, the MPC algorithm allows to include constraints in the optimization problem. This is very useful as physical and dynamical characteristics of the vehicle and different types of obstacles can be taken into account just by the inclusion of proper constraints in the minimization stage (Hang et al., 2017).

The navigation task aims to solve the problem of determining the position, velocity and orientation of a vehicle in space using different sources of information (inertial measurement units, GPS, among others). Traditionally, the Extended Kalman Filter (EKF) (Roumeliotis and Bekey, 1999), Unscented Kalman Filter (UKF) (Rhudy et al., 2013) or the Particle Filter (PF) (Cheng and Crassidis, 2004) are used to solve the navigation problem. Recently, the use of non-linear observers such as MHE have been proposed as an alternative to the different types of Kalman filters and statistical methods (Grip et al., 2012; Vandersteen et al., 2013). Both EKF and MHE are based on the solution of a least-squares problem. While EKF uses recursive updates to obtain the estimates and the error covariance matrix, MHE uses a finite horizon window and solve a constrained optimization problem to find the estimates. In this way, the physical limits of the system states and parameters can be modeled through the optimization problem's constraints. The omission of this information can degrade the estimation algorithm performance (Haseltine and Rawlings, 2005).

The path-planning task deals with searching a feasible path between the present location and the desired target while taking into consideration the geometry of the vehicle and its surroundings, its kinematic constraints and other factors that may affect the feasible path. Different methodologies are used to find feasible paths (LaValle, 2006; Saska et al., 2015; Xue et al., 2014), however, most of them do not consider the dynamics of the UV that should follow the path. In their review article, Yang et al. (2016) have surveyed different path-planning algorithms. The authors discuss the fundamentals of the most successful robot 3D path-planning algorithms that have been developed in recent years. They mainly analyze algorithms that can be implemented in aerial robots, ground robots and underwater robots. They classify the different algorithms into five categories: i) sampling based algorithms, ii) node based algorithms, iii) mathematical model based algorithms (which include optimal control and receding horizon strategies), iv) bioinspired algorithms, and v) multifusion based algorithms. From these, only mathematical model based algorithms are able to incorporate in a simple way both the environment (kinematic constraints) and the vehicle dynamics in the path-planning process.

As it can be seen, there are a wide range of techniques that can be used to implement the guidance, navigation and path-planning tasks. In this article we propose to use a unified framework based on receding horizon techniques to design the three aforementioned tasks. To design the path-planning and guidance modules we propose the use of the MPC algorithm (Murillo et al., 2016), and to solve the navigation task we propose the use of the MHE algorithm (Sánchez et al., 2017b). In this way, physical and dynamical constraints can be considered in the path-planning, guidance and estimation stages. The advantages of using the proposed framework are: i) the obtained path is guaranteed to be optimal and feasible, ii) the estimates of states and parameters of the system are improved as they are guaranteed to satisfy physical limits, iii) the position of actuators and motors' speeds are computed in an optimal fashion satisfying their physical limits, and iv) the three modules are in state-space form which is very useful when working with multiple-input multiple-output (MIMO) systems.

This article is organized as follows. In Section 2 we provide a general overview of receding horizon techniques. In Section 3 the guidance problem is formulated. In Section 4 the way in which we estimate states and parameters is presented. The path-planning problem is described in Section 5. Simulation results are presented in Section 6. Finally, conclusions are stated in Section 7.

2 RECEDING HORIZON PRINCIPLE

The main objective of receding horizon techniques is to solve an explicit inverse problem that allows the incorporation, at the design stage, of different types of constraints to obtain the best feasible solution. For both control and estimation, the inverse problem to solve is the minimization of a cost function that quantifies the performance of the system (for control case) and how well we estimate unknown states and parameters (for estimation case). This constrained minimization process is done over a fixed-time horizon window of a certain length. This is shown in Fig. 2 for time k, being the cyan dashed-dot line the control window of length N_c and the red dashed line the estimation window of length N_e. The arrival cost provides a mean to incorporate information from previous measurements to the current estimates and the

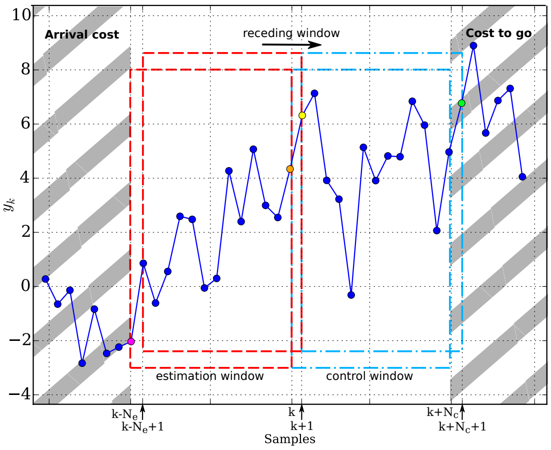


Figure 2: Scheme of the receding horizon principle

cost to go includes the missing information due to the use of a finite horizon approach. At the next sampling instant $k+1$, new information is included and old one is discarded by shifting both windows one step in time and the constrained minimization process is restarted at the next sampling instant. This is also shown in Fig. 2 for time $k+1$.

When we refer to a control receding strategy we think in model predictive control. For control case, we generally solve at each sampling instant, a constrained minimization problem of the form

$$\min_{\mathbf{U}_{k|k}} \mathcal{J}_{\text{MPC}}$$
$$\text{st.} \begin{cases} \mathbf{x}_{k+i+1|k} = f(\mathbf{x}_{k+i|k}, \mathbf{u}_{k+i|k}), \\ \mathbf{x}_{k|k} = \mathbf{x}(k), \\ \mathbf{u}_{k+i|k} \in \mathcal{U}, \quad \mathbf{x}_{k+i|k} \in \mathcal{X}, \end{cases} \tag{1}$$

where $\mathcal{J}_{\text{MPC}}$ denotes the cost function to be minimized, $i \in [0, 1, \cdots, N_c-1]$, $\mathbf{x}_{k+i|k} \in \mathcal{X} \subseteq \Re^{n_x}$ is the state vector, $\mathbf{u}_{k+i|k} \in \mathcal{U} \subseteq \Re^{n_u}$ is the control input vector, $\mathcal{X}$ and $\mathcal{U}$ are the state and input constraint sets, respectively, $\mathbf{U}_{k|k} = \left[\mathbf{u}_{k|k}, \cdots, \mathbf{u}_{k+N_c-1|k}\right]^T$ is the control input sequence and $f(\cdot)$ is a continuous and differentiable vector function that describes the dynamics of the system. The solution of Eq. (1) is an optimal control input sequence (denoted here with an asterisk) $\mathbf{U}^*_{k|k} = \left[\mathbf{u}^*_{k|k}, \cdots, \mathbf{u}^*_{k+N_c-1|k}\right]^T$, but only the first control input of this sequence is applied to the system, i.e. $\mathbf{u}_k = \mathbf{u}^*_{k|k}$. Then, the horizon is shifted forward to the next sampling instant ($k \leftarrow k+1$) in a receding horizon fashion, discarding old information but including the new one, thus compensating for unmeasured disturbances and/or unmodelled dynamics. As the reader can see, the cost function $\mathcal{J}_{\text{MPC}}$ in Eq. (1) plays a key role in obtaining the optimal control sequence and it should be carefully designed in order to fulfill the goals of the system. The different cost functions used to design the guidance and path-planning tasks will be properly defined in Section 3 and 5, respectively.

To perform states and/or parameters estimation in a receding horizon fashion we use moving horizon estimation. In this case, we also solve a constrained minimization problem at each sampling instant but the solution obtained is different from the previous one as it includes estimates of states, parameters and noises. The MHE problem has the following form

$$\min_{\hat{\mathbf{x}}_{k-N_e|k},\,\hat{\mathbf{W}}_{k|k}} \mathcal{J}_{\text{MHE}}$$
$$\text{s.t.} \begin{cases} \hat{\mathbf{x}}_{k+i+1|k} = \bar{f}(\hat{\mathbf{x}}_{k+i|k}, \hat{\mathbf{z}}_{k+i|k}, \hat{\mathbf{w}}_{k+i|k}), \\ \hat{\mathbf{y}}_{k+i|k} = \bar{h}(\hat{\mathbf{x}}_{k+i|k}, \hat{\mathbf{v}}_{k+i|k}), \\ \hat{\mathbf{x}}_{k+i|k} \in \bar{\mathcal{X}}, \quad \hat{\mathbf{z}}_{k+i|k} \in \bar{\mathcal{Z}}, \\ \hat{\mathbf{w}}_{k+i|k} \in \bar{\mathcal{W}}, \quad \hat{\mathbf{v}}_{k+i|k} \in \bar{\mathcal{V}}, \end{cases} \tag{2}$$

where $\mathcal{J}_{\text{MHE}}$ denotes the cost function to be minimized, the hat $(\hat{\cdot})$ denotes *estimated* value, $i \in [-N_e, \cdots, -1]$, $\hat{\mathbf{x}}_{k+i|k} \in \bar{\mathcal{X}} \subseteq \Re^{n_{\hat{x}}}$, $\hat{\mathbf{z}}_{k+i|k} \in \bar{\mathcal{Z}} \subseteq \Re^{n_{\hat{z}}}$ stand for the estimated state and algebraic state vectors, respectively, $\hat{\mathbf{w}}_{k+i|k} \in \bar{\mathcal{W}} \subseteq \Re^{n_{\hat{w}}}$ and $\hat{\mathbf{v}}_{k+i|k} \in \bar{\mathcal{V}} \subseteq \Re^{n_{\hat{v}}}$ are the estimated process and measurement noises vectors, respectively, $\hat{\mathbf{y}}_{k+i|k} \in \bar{\mathcal{Y}} \subseteq \Re^{n_{\hat{y}}}$ is the measurement vector, $\bar{f}$ and $\bar{h}$ are differentiable vector functions that define the system dynamics and the measurement equation, respectively, $\bar{\mathcal{X}}$, $\bar{\mathcal{Z}}$, $\bar{\mathcal{W}}$, $\bar{\mathcal{V}}$ and $\bar{\mathcal{Y}}$ are the state, algebraic state, process noise, measurement noise and measurement vector constraints sets, respectively. The solution of problem of Eq. (2) are the optimal sequences $\hat{\mathbf{X}}^*_{k|k} = \left[\hat{\mathbf{x}}^*_{k-N_e+1|k}, \cdots, \hat{\mathbf{x}}^*_{k|k}\right]^T$ and $\hat{\mathbf{W}}^*_{k|k} = \left[\hat{\mathbf{w}}^*_{k-N_e|k}, \cdots, \hat{\mathbf{w}}^*_{k|k}\right]^T$, from where we extract the current estimates, i.e. $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}^*_{k|k}$ and $\hat{\mathbf{w}}_k = \hat{\mathbf{w}}^*_{k|k}$. Then, the estimation window is shifted forward to the next sampling instant ($k \leftarrow k+1$) in order to drop the oldest measurement and to include the current one, and the minimization process is restarted. In this case, the cost function $\mathcal{J}_{\text{MHE}}$ also plays a key role in the behaviour of the MHE algorithm, and it should be carefully chosen in order to obtain the best estimates as possible. The cost function used in the navigation task will be properly described in Section 4.

3 THE GUIDANCE PROBLEM

To design the guidance task, we propose to solve Eq. (1) using the following cost function:

$$\mathcal{J}_g = \sum_{j=0}^{N_g-1} \left(\|\mathbf{x}_{k+j|k} - \mathbf{x}^{\text{sp}}_{k+j|k}\|^2_{\mathbf{Q}} + \|\Delta\mathbf{u}_{k+j|k}\|^2_{\mathbf{R}} \right) + \|\mathbf{x}_{k+N_g|k} - \mathbf{x}^{\text{sp}}_{k+N_g|k}\|^2_{\mathbf{P}}, \tag{3}$$

where N_g is the guidance prediction horizon, $\mathbf{Q}$, $\mathbf{R}$ and $\mathbf{P}$ are positive definite matrices. $\|(\cdot)\|^2_\alpha$ stands for the $alpha$-weighted 2-norm, $\Delta\mathbf{u}_{k+j|k} = \mathbf{u}_{k+j|k} - \mathbf{u}_{k+j-1|k}$ and $\mathbf{x}^{\text{sp}}_{k+j|k}$ is the desired path which is computed by the path-planning task.

To test the guidance task, we propose to use a Jackal[1] unmmaned ground vehicle (UGV), which is a small, fast, entry-level field robotics research platform. We need to emphasize that we do not know the exact mathematical model of the Jackal UGV, however, we do know that it is a complex model simulated by Gazebo simulator[2]. We propose to model Jackal UGV with the mathematical model of a differential drive UGV. This last model is very simple, but for us is the best model at hand and, as it will be shown in the simulation example, it will allow us to control accurately the Jackal UGV along the pre-defined path. The mathematical model of the differential drive UGV can be written as follows:

$$\dot{\tilde{\mathbf{x}}}(t) = \tilde{f}(\tilde{\mathbf{x}}(t), \tilde{\mathbf{u}}(t)) = \begin{bmatrix} \tilde{v}(t)\cos\tilde{\psi}(t) \\ \tilde{v}(t)\sin\tilde{\psi}(t) \\ \tilde{\omega}(t) \end{bmatrix}, \tag{4}$$

[1] https://clearpathrobotics.com/jackal-small-unmanned-ground-vehicle/

[2] http://gazebosim.org/

where $\tilde{\mathbf{x}}(t) = [\tilde{x}(t),\, \tilde{y}(t),\, \tilde{\psi}(t)]^T$ and $\tilde{\mathbf{u}}(t) = [\tilde{v}(t),\, \tilde{\omega}(t)]^T$ are the UGV state and contol input vectors, respectively, $\tilde{x}(t)$ and $\tilde{y}(t)$ denote the UGV *xy*-position, $\tilde{\psi}(t)$ denotes its yaw angle, $\tilde{v}(t)$ and $\tilde{\omega}(t)$ stands for the UGV linear and angular velocities, respectively. As it can be seen, Eq. (4) is expressed in its continuous-time form, but in order to be used in Eq. (1) it should be in its discrete-time form. We have solved this issue using a discretization method called *collocation* since it provides great accuracy at a relatively low computational cost (Sánchez et al., 2017a).

4 THE NAVIGATION PROBLEM

In order to design the navigation task, we propose to solve Eq. (2) with the following cost function

$$\begin{aligned}\mathcal{J}_n =& P_0(1 - ||\hat{q}_b^n(k - N_e)||_2^2) + ||\hat{\mathbf{x}}_{k-N_e|k} - \bar{\mathbf{x}}_{k-N_e|k}||^2_{\mathbf{P}_1} + \\ & ||\hat{\mathbf{z}}_{k-N_e|k} - \bar{\mathbf{z}}_{k-N_e|k}||^2_{\mathbf{P}_2} + \sum_{j=-N_e}^{0} \left(||\hat{\mathbf{w}}_{k+j|k}||^2_{\mathbf{Q}_w} + ||\hat{\mathbf{v}}_{k+j|k}||^2_{\mathbf{R}_v}\right)\end{aligned} \tag{5}$$

where N_e is navigation prediction horizon, $\mathbf{Q}_w$ and $\mathbf{R}_v$ are symmetric positive definite matrices that penalize the estimated noise vectors $\hat{\mathbf{w}}_{k+j|k}$ and $\hat{\mathbf{v}}_{k+j|k}$, respectively, $\bar{\mathbf{x}}_{k-N_e|k}$ and $\bar{\mathbf{z}}_{k-N_e|k}$ are the current knowledge of the initial states and algebraic states estimates, respectively, P_0 is a positive constant which penalizes the deviations of the quaternion $\hat{q}_b^n$ which must have unit norm, and $\mathbf{P}_1$ and $\mathbf{P}_2$ are symmetric positive semi-definite weighting matrices.

Using the equations that describe the rigid body dynamics (for a detailed description, see Polóni et al. (2015)), the mathematical model used in the navigation task can be written in Earth-Centered Earth-Fixed (ECEF) coordinates as follows:

$$\dot{\mathbf{x}}(t) \;=\; \bar{f}(\mathbf{x}(t), \mathbf{z}(t), \mathbf{w}(t)) \;=\; \begin{bmatrix} v^e(t) \\ -2S(\omega^e_{ie}(t))v^e(t) + a^e(t) + g^e(p^e(t)) \\ \frac{1}{2}q^e_b(t)\cdot\tilde{\omega}^b_{ib}(t) - \frac{1}{2}\tilde{\omega}^e_{ie}(t)\cdot q^e_b(t) \\ 0 \\ 0 \end{bmatrix}, \tag{6}$$

where $\mathbf{x}(t) = [p^n(t),\, v^n(t),\, q_b^n(t),\, \alpha(t),\, \beta(t)]^T$ and $\mathbf{z}(t) = [\omega^b(t),\, a^n(t)]^T$ are the state vector and the algebraic state vector, respectively, $p^n(t) \in \Re^3$ is the position, $v^n(t) \in \Re^3$ is the linear velocity, the quaternion $q_b^n(t) \in \Re^4$ determines the orientation of the rigid body and $\alpha(t) \in \Re^3$ and $\beta(t) \in \Re^3$ are the gyroscope and accelerometer bias, respectively. $\omega^b(t) \in \Re^3$ and $a^n(t) \in \Re^3$ are the angular velocity and linear acceleration vectors. The superscript n, b and e denote the East-North-Up (ENU) reference frame, the body reference frame and the ECEF reference frame, respectively.

The measurement equations with measurement noise $\mathbf{v}(t) = [\mathrm{v}_\omega(t),\, \mathrm{v}_a(t),\, \mathrm{v}_m(t),\, \mathrm{v}_p(t),\, \mathrm{v}_v(t)]^T$, with $\mathrm{v}_{(\cdot)}(t) \in \Re^3$, are given by:

$$\mathbf{y}(t) = \bar{h}(\mathbf{x}(t), \mathbf{v}(t)) = \begin{bmatrix} \omega^b(t) + \alpha(t) + \mathrm{v}_\omega(t) \\ \mathbf{R}(q^e_b)^T a^e(t) + \beta(t) + \mathrm{v}_a(t) \\ \mathbf{R}(q^e_b)^T m^e(t) + \mathrm{v}_m(t) \\ p^e(t) + \mathrm{v}_p(t) \\ v^e(t) + \mathrm{v}_v(t) \end{bmatrix}, \tag{7}$$

where $m^e(t)$ contains the values of the magnitude of the terrestrial magnetic field given our current latitude and longitude[3], $\omega^b(t)$ and $a^e(t)$ are the angular velocity and linear acceleration vectors, respectively. The matrix $\mathbf{R}(q_b^e(t))$ is the rotation matrix associated with the current orientation quaternion. As before, Eqs. (6) and (7) are discretized using *collocation* method.

5 THE PATH-PLANNING PROBLEM

In order to design the path-planning task, we propose to solve Eq. (1) with the following cost function:

$$\mathcal{J}_p = \sum_{j=0}^{N_p-1} \left(\|\mathbf{x}_{k+j|k} - \breve{\mathbf{w}}_{k+j|k}\|^2_{\breve{\mathbf{Q}}} + \|\Delta \mathbf{u}_{k+j|k}\|^2_{\breve{\mathbf{R}}} \right) + \|\mathbf{x}_{k+N_p|k} - \breve{\mathbf{w}}_{k+N_p|k}\|^2_{\breve{\mathbf{P}}}, \tag{8}$$

where N_p is the path-planning horizon length, $\breve{\mathbf{w}}_{k+j|k}$ defines each consecutive target waypoint, $\breve{\mathbf{Q}}$, $\breve{\mathbf{R}}$ and $\breve{\mathbf{P}}$ are positive definite matrices.

In this article we propose to use an MPC based path-planning algorithm which uses a virtual particle vehicle (PV) model to compute the paths (Murillo et al. (2018)). The mathematical model that describes the movement of the PV can be written as follows:

$$\dot{\breve{\mathbf{x}}}(t) \;=\; \breve{f}(\breve{\mathbf{x}}(t), \breve{\mathbf{u}}(t)) \;=\; \begin{bmatrix} \breve{v}(t) \cos \breve{\psi}(t) \\ \breve{v}(t) \sin \breve{\psi}(t) \\ -\tau \breve{v}(t) + \kappa \breve{\mathcal{T}}(t) \end{bmatrix}, \tag{9}$$

where $\breve{\mathbf{x}}(t) = [\breve{x}(t),\, \breve{y}(t),\, \breve{v}(t)]^T$ and $\breve{\mathbf{u}}(t) = [\breve{\psi}(t),\, \breve{\mathcal{T}}(t)]^T$ are the PV state and control input vectors, respectively, $\breve{x}(t)$ and $\breve{y}(t)$ denote the PV *xy*-position of the PV and $\breve{v}(t)$ is the modulus of the PV velocity vector, $\breve{\psi}(t)$ and $\breve{\mathcal{T}}(t)$ denote the yaw angle and the thrust force of the PV, respectively, τ is a damping constant that determines the rate of change of the PV velocity and κ is a constant proportional to the thrust force $\breve{\mathcal{T}}$. Equation (9) is also discretized using collocation method.

6 SIMULATION EXAMPLE

In this section we present the usage of the proposed framework in order to estimate unknown states, compute a feasible path and follow this path with a Jackal UGV using Gazebo simulator, CasADi (Andersson et al., 2019), MPCTools (Risbeck and Rawlings, 2016) and Ipopt (Wächter and Biegler, 2006).

For the guidance task we assume that the initial conditions of model defined in Eq. (4) are $\tilde{\mathbf{x}}_0 = [0,\, 0,\, 0]^T$ and $\tilde{\mathbf{u}}_0 = [0,\, 0]^T$. This model is discretized using collocation method with a sampling rate $T_s = 0.1\,\mathrm{s}$ and the guidance horizon is set to $N_g = 10$. The weight matrices are chosen as: $\mathbf{R} = diag([20,\, 20])$, $\mathbf{Q} = diag([300,\, 300,\, 150])$ and $\mathbf{P} = diag([500,\, 500,\, 400])$. The constraints are configured according to the capabilities of the Jackal UGV: $0 \leq \tilde{v}(t) \leq 1.9\,(\mathrm{m/s})$, $-2 \leq \tilde{\omega}(t) \leq 2\,(\mathrm{rad/s})$, $-0.3 \leq \Delta\tilde{v}(t) \leq 0.3\,(\mathrm{m/s})$, $-30 \leq \Delta\tilde{\omega}(t) \leq 30\,(\mathrm{deg/s})$. $\tilde{\psi}(t)$, $\tilde{x}(t)$ and $\tilde{y}(t)$ are unconstrained.

For the path-planning task we assume that the PV model initial conditions are $\breve{\mathbf{x}}_0 = [0,\, 0,\, 0]^T$ and $\breve{\mathbf{u}}_0 = [0,\, 0]^T$. This model is also discretized using collocation method with a sampling rate $T_s = 0.1\,\mathrm{s}$. The path-planning horizon is set to $N_p = 15$. The weight matrices are chosen as $\breve{\mathbf{R}} = diag([0.01,\, 0.01])$, $\breve{\mathbf{Q}} = diag([100,\, 100,\, 100])$ and $\breve{\mathbf{P}} = diag([500,\, 500,\, 500])$. The

[3] This data is tabulated and can be obtained from https://www.ngdc.noaa.gov/geomag-web

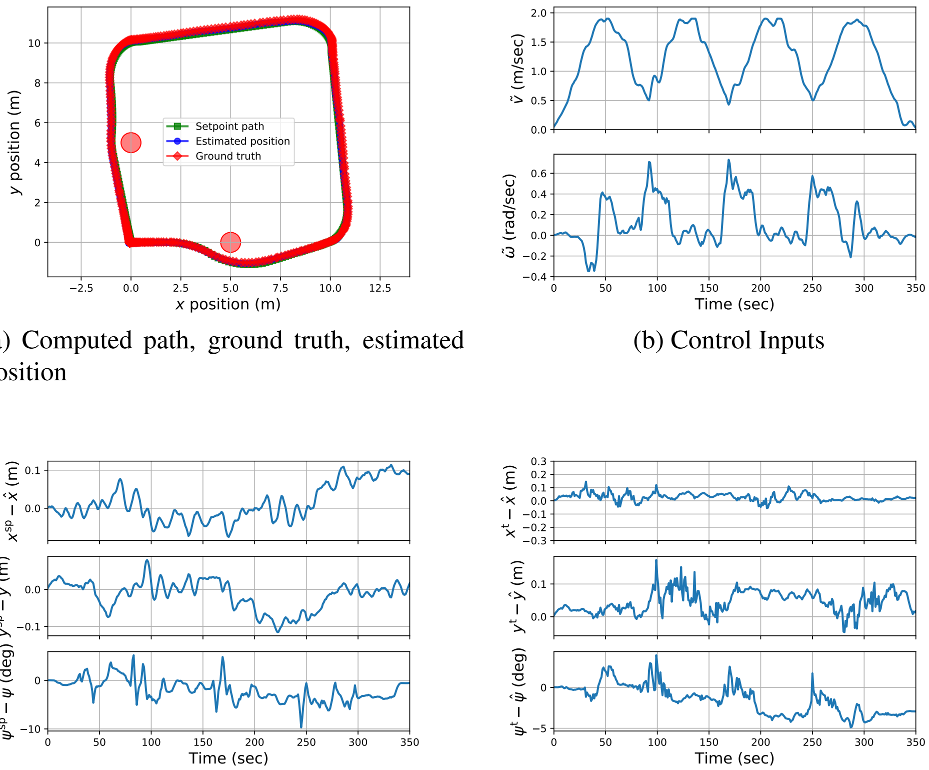
PV constraints are configured as follows: $0 \leq \mathcal{T}(t) \leq 5\,(\mathrm{N})$, $-30 \leq \Delta\breve{\psi}(t) \leq 30\,(\mathrm{deg/s})$, $-0.5 \leq \Delta\mathcal{T}(t) \leq 0.5\,(\mathrm{N})$ and $0 \leq \breve{v}(t) \leq 1.9\,(\mathrm{m/s})$. $\breve{\psi}(t)$, $\breve{x}(t)$ and $\breve{y}(t)$ are unconstrained. Both constants of the PV model are set as $\tau = 2\,(1/\mathrm{s})$ and $\kappa = 2\,(1/\mathrm{kg})$. When computing the path we assume that there are two circular obstacles of radii $r_{o_1} = r_{o_2} = 0.5\,(\mathrm{m})$ and centres $\mathbf{c}_{o_1} = [5,\ 0]^T\,(\mathrm{m})$ and $\mathbf{c}_{o_2} = [0,\ 5]^T\,(\mathrm{m})$. The Jackal UGV should achieve the following consecutive waypoints: $\breve{\mathbf{w}}_1 = [10,\ 0,\ 0.2]^T$, $\breve{\mathbf{w}}_2 = [0,\ 10,\ 0.2]^T$, $\breve{\mathbf{w}}_3 = [10,\ 10,\ 0.2]^T$ and $\breve{\mathbf{w}}_4 = [0,\ 0,\ 0]^T$.

For the navigation task we assume that there is no process noise ($\hat{\mathbf{w}}_{k+j|k} = 0$). The Jackal UGV simulated by Gazebo has two sensors: i) IMU, and ii) GNSS. The data given by these sensors is corrupted by Gaussian noise and it is fused in order to estimate the Jackal's *xy*-position and its yaw angle. Here we use model of Eq. (6) in ENU coordinates with initial conditions $\bar{\mathbf{x}}_{-N_e} = [0,\ 0,\ 0.0635,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0]^T$ and $\bar{\mathbf{z}}_{-N_e} = [0,\ 0,\ 0,\ 0,\ 0,\ 0]^T$. This model is also discretized using collocation method with a sampling rate $T_s = 0.1\,\mathrm{s}$ and the estimation horizon is set to $N_e = 6$. The weights defined in Eq. (5) are chosen as $\mathbf{Q}_w = diag([10^{-1},\ 10^{-1},\ 10^{-1},\ 10^{-3},\ 10^{-3},\ 10^{-3},\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1])$, $\mathbf{R}_v = diag([1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 10^{-3},\ 10^{-3},\ 10^{-3}])$, $P_0 = 0.1$, $\mathbf{P}_1 = diag([10^{-1}, 10^{-1}, 10^{-1}, 10^{-3}, 10^{-3}, 10^{-3}, 0.5,\ 0.5,\ 0.5,\ 0.5,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0])$ and $\mathbf{P}_2 = diag([1,\ 1,\ 1,\ 1,\ 1,\ 1])$.

The results obtained can be seen in Figures 3a- 3d. In Fig. 3a, three different paths are shown: i) the setpoint path (squared-green line) which is computed by the path-planning task, ii) the estimated Jackal *xy*-position (rounded-blue line) which is computed by the navigation task and controlled by the guidance task in order to follow the setpoint path, and iii) the ground truth (diamond-red line) which is given by Gazebo. From this figure it can be seen that the interrelated work of the three tasks (path-planning, navigation and guidance) is done in such a satisfactory way that it allows to move the Jackal UGV along the desired path with minor errors, considering that sensors are corrupted by Gaussian noise. In Fig. 3b the Jackal UGV control inputs are shown. Both control inputs are feasible and, as it can be seen from this figure, the linear velocity of the vehicle is limited up to $1.9\,(\mathrm{m/s})$ which matches both constraints defined in the path-planning and guidance tasks. In Fig. 3c the guidance errors are shown, where x^{sp} and y^{sp} are the desired xy-coordinates that define the desired path and ψ^{sp} is the desired yaw angle. Note that we do not only define the Jackal UGV xy-coordinates but also we impose the orientation it should have along the desired path. $\hat{x}$, $\hat{y}$ and $\hat{\psi}$ are the components of the estimated Jackal state vector. As it can be seen form this figure (top and middle), the guidance errors in xy-position are small. However, the guidance error in the yaw angle seems to be a bit bigger than the errors in the position. This is mainly due to the fact that the yaw angle is a control input for the PV model in Eq. (9) and consequently no dynamics is considered. This problem can be solved by adding this angle as a state variable and defining a first order dynamics for it, as we did with the velocity of the PV. Figure 3d depicts the estimation errors, where x^{t}, y^{t} and ψ^{t} denote the truth xy-component and the truth yaw angle, respectively, which can be obtained from Gazebo simulator in order to check if our estimations are close to this truth values. It can be seen that the navigation task is successfully employed as the errors we have made in the estimation of xy-position (between $\pm 0.15\,(\mathrm{m})$) and the yaw angle (between $\pm 5\,(\mathrm{deg})$) of the Jackal UGV are small.

7 CONCLUSIONS

In this article, we have presented a unified receding horizon framework that can be used for the guidance and navigation of any UV along any feasible path. This framework can be split into three interrelated task, all of them designed using a receding horizon principle, which



(a) Computed path, ground truth, estimated position

(b) Control Inputs

(c) Guidance Errors

(d) Estimation Errors

allows us to include physical dynamics and constraints at the design stage. We have used MPC technique for guidance and path-planning tasks and MHE for the navigation task. To evaluate the performance of the proposed framework, we have used Gazebo simulator in order to drive a Jackal UGV model along the path computed by the path-planning task.

ACKNOWLEDGMENTS

The authors wish to thank the *Universidad Nacional de Litoral* (with CAI+D 500 201501 00050 LI and CAI+D 504 201501 00098 LI), the *Agencia Nacional de Promoción Científica y Tecnológica* (with PICT 2016-0651 and PICT 2017-0543) and the *Consejo Nacional de Investigaciones Científicas y Técnicas* (CONICET) from Argentina, for their support.

REFERENCES

Andersson J.A., Gillis J., Horn G., Rawlings J.B., and Diehl M. Casadi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2019.

Cheng Y. and Crassidis J. Particle Filtering for Sequential Spacecraft Attitude Estimation. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*. American Institute of Aeronautics and Astronautics, Reston, Virigina, 2004. ISBN 978-1-62410-073-4. doi:10.2514/6.2004-5337.

Grip H.F., Fossero T.I., Johansent T.A., and Saberi A. A nonlinear observer for integration of gnss and imu measurements with gyro bias estimation. In *American Control Conference (ACC), 2012*, pages 4607–4612. IEEE, 2012.

Hang P., Luo F., Fang S., and Chen X. Path tracking control of a four-wheel-independentsteering electric vehicle based on model predictive control. In *Control Conference (CCC), 2017 36th Chinese*, pages 9360–9366. IEEE, 2017.

Haseltine E.L. and Rawlings J.B. Critical evaluation of extended kalman filtering and moving-horizon estimation. *Industrial & engineering chemistry research*, 44(8):2451–2460, 2005.

Khamseh H.B. and Janabi-Sharifi F. Ukf–based lqr control of a manipulating unmanned aerial vehicle. *Unmanned Systems*, 5(03):131–139, 2017.

LaValle S.M. *Planning algorithms*. Cambridge university press, 2006.

Murillo M., Sánchez G., Genzelis L., and Giovanini L. A real-time path-planning algorithm based on receding horizon techniques. *Journal of Intelligent & Robotic Systems*, 91(3-4):445–457, 2018.

Murillo M., Sánchez G., and Giovanini L. Iterated non-linear model predictive control based on tubes and contractive constraints. *ISA Transactions*, 62:120 – 128, 2016. doi:http://dx.doi.org/10.1016/j.isatra.2016.01.008.

Polóni T., Rohal-Ilkiv B., and Arne Johansen T. Moving Horizon Estimation for Integrated Navigation Filtering. *IFAC-PapersOnLine*, 48(23):519–526, 2015. ISSN 24058963. doi:10.1016/j.ifacol.2015.11.331.

Rhudy M., Gu Y., Gross J., Gururajan S., and Napolitano M.R. Sensitivity Analysis of Extended and Unscented Kalman Filters for Attitude Estimation. *Journal of Aerospace Information Systems*, 10(3):131–143, 2013. ISSN 2327-3097. doi:10.2514/1.54899.

Risbeck M. and Rawlings J. Mpctools: Nonlinear model predictive control tools for casadi. 2016.

Roumeliotis S. and Bekey G. 3-D Localization for a Mars Rover Prototype. In *Artificial Intelligence, Robotics and Automation in Space, Proceedings of the Fifth International Symposium, ISAIRAS '99*, pages 441–448. 1999.

Sánchez G., Murillo M., Genzelis L., Deniz N., and Giovanini L. Mpc for nonlinear systems: A comparative review of discretization methods. In *Information Processing and Control (RPIC), 2017 XVII Workshop on*, pages 1–6. IEEE, 2017a.

Sánchez G., Murillo M., and Giovanini L. Adaptive arrival cost update for improving moving horizon estimation performance. *ISA transactions*, 68:54–62, 2017b.

Saska M., Spurný V., and Vonásek V. Predictive control and stabilization of nonholonomic formations with integrated spline-path planning. *Robotics and Autonomous Systems*, 2015. ISSN 0921-8890. doi:http://dx.doi.org/10.1016/j.robot.2015.09.004.

Vandersteen J., Diehl M., Aerts C., and Swevers J. Spacecraft Attitude Estimation and Sensor Calibration Using Moving Horizon Estimation. *Journal of Guidance, Control, and Dynamics*, 36(3):734–742, 2013. ISSN 0731-5090. doi:10.2514/1.58805.

Wächter A. and Biegler L.T. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1):25–57, 2006.

Xue Q., Cheng P., and Cheng N. Offline path planning and online replanning of uavs in complex terrain. In *Proceedings of 2014 IEEE Chinese Guidance, Navigation and Control Conference*, pages 2287–2292. 2014. doi:10.1109/CGNCC.2014.7007525.

Yang L., Qi J., Song D., Xiao J., Han J., and Xia Y. Survey of robot 3d path planning algorithms. *Journal of Control Science and Engineering*, 2016, 2016. doi:http://dx.doi.org/10.1155/2016/7426913.

Zhao P., Chen J., Song Y., Tao X., Xu T., and Mei T. Design of a control system for an autonomous vehicle based on adaptive-pid. *International Journal of Advanced Robotic Systems*, 9(2):44, 2012.