Trajectory Tracking Nonlinear Model Predictive Control for Autonomous Surface Craft

B. J. Guerreiro, C. Silvestre, R. Cunha, A. Pascoal

***Abstract*— This paper describes a solution to the problem of trajectory tracking control of autonomous surface craft (ASC) in the presence of constant ocean currents. The solution proposed is rooted in nonlinear model predictive control (MPC) techniques and addresses explicitly state and input constraints. Whereas state saturation constraints are added to the underlying optimization cost functional as penalties, input saturation constraints are made intrinsic to the nonlinear model used in the optimization problem, therefore reducing the computational burden of the resulting MPC algorithm. Simulation results obtained with a nonlinear dynamic model of a prototype ASC show that the MPC strategy adopted yields good performance in the presence of constant currents and holds good potential for real-time implementation.**

I. INTRODUCTION

This paper addresses the problem of trajectory tracking control of Autonomous Surface Craft (ASC) under the effect of constant ocean currents. The increasing demand by marine scientists of adequate technological tools to sample the ocean at the appropriate temporal and spatial scales motivates the use of ASCs capable of automatically acquiring and transmitting large data sets to one or more support units installed on shore. In the future, this practical setup will enable scientists to control the execution of sea missions from the security and comfort of their laboratories.

The problem of trajectory tracking control is particularly relevant when performing time-critical ASC missions. Trajectory tracking controllers are traditionally based on a two-step design methodology: a fast inner loop that stabilizes the vehicle's attitude and, using a time-scale separation criterion, a slower outer loop that relies on the kinematic equations of the vehicle and converts the tracking errors into inner loop commands. An integrated approach to the design of inner-outer loop control structures for autonomous vehicles moving in 3D space was proposed in [1], [2]. The methodology adopted relied on linearization techniques for linear controller design about trimming trajectories, together with gain scheduling techniques to switch among the linear controllers. The interested reader is referred to [3], [4] for a discussion of topics related to this circle of ideas. See also [5] for the application of similar techniques to the control of the DELFIMx ASC. It is also worth pointing out that some authors use nonlinear or adaptive control techniques to tackle the ASC control problem [6], [7].

The methodology proposed in this paper is formulated in the scope of model-based predictive control (MPC) [8], [9] and is in line with the methodology developed in [10], to tackle the trajectory tracking control problem under the effect of constant currents within the physical limitations of the vehicle. The control law adopted here is obtained by solving on-line, at each sampling instant, a finite horizon open-loop optimal control problem and using the actual state of the plant as the initial state. The optimization yields an optimal control sequence and the first element of this sequence is applied to the plant. Since the optimal control problem is solved online, it is straightforward to add state and control saturation constraints as penalty functions to the cost functional. The vehicle model constraint can also be readily incorporated using lagrange multipliers. The resulting optimization problem is solved numerically using the quasi-Newton method to compute search directions and resorting to the Wolfe conditions in a line search algorithm to solve a step size optimization subproblem [11]. The vehicle model adopted in this work is a catamaran ASC nonlinear dynamic model derived from first physics principles [12], [13]. The model captures the influence of ocean currents to better evaluate the performance of the controller under realistic scenarios. The standard MPC approach to deal with model constraints is to incorporate basic saturation constraints into the optimization problem, so that the resulting control sequence is always admissible. In an effort to use a self contained nonlinear model of the ASC, the intrinsic physical limitations of the actuators are incorporated in the model using smooth saturation functions. This improved model allows for the optimization algorithm to provide valid control actions, even without using constraints in the cost functional, thus decreasing the computational effort.

The key contribution of the paper is the use of simple and well established optimization techniques to solve the trajectory tracking MPC problem for a full nonlinear model of an ASC under constant disturbances. The methodology adopted yields a controller structure that lends itself to real-time implementation with reduced computational power. Furthermore, the usage of intrinsic saturation described in this paper simplifies the controller design process, since there is no need for input constraints in the optimization problem.

The paper is organized as follows. Section II presents a brief summary of the ASC dynamic model, including the error dynamics and the intrinsic actuation saturation func-

The authors are with the Institute for Systems and Robotics, Instituto Superior Técnico, Av. Rovisco Pais 1, 1049-001 Lisboa, Portugal {bguerreiro,cjs,rita,pascoal}@isr.ist.utl.pt .

This work was partially supported by Fundação para a Ciência e a Tecnologia (ISR/IST pluriannual funding) through the POS Conhecimento Program that includes FEDER funds, project GREX / CEC-IST (Contract No. 035223) and the FREESUBNET RTN of the CEC. The work of B. J. Guerreiro was supported by PhD Student Grant SFRH/BD/21781/2005 from the Portuguese FCT POCTI programme.

tions. Section III formulates the MPC problem, describing the control problem, the saturation and model constraints, and the optimization algorithm. Implementation issues and simulation results are presented in Section IV. Finally Section V contains the main conclusions and discusses issues that warrant further research work.

II. Catamaran Model

This section describes the dynamic model of an ASC in the horizontal plane. The prototype vehicle considered has two hulls, two propellers driven by electrical motors and a torpedo-shaped sensor container, attached to the vehicle by a central wing-shaped link (see [12] for an in-depth presentation of this model and a description of the DELFIMx surface craft).

Adopting standard notation in the field, let $\{U\}$ denote an inertial coordinate frame and $\{B\}$ the body fixed coordinate frame attached to the vehicle's center of mass. Further consider the position ${}^{U}\mathbf{p}_B = [x\ y]'$ of the origin of $\{B\}$ with respect to $\{U\}$, the linear velocity $\mathbf{v} = [u\ v]'$ of frame $\{B\}$ relative to $\{U\}$ and expressed in $\{B\}$, the heading angle ψ that describes the orientation of frame $\{B\}$ with respect to $\{U\}$, and the angular velocity r of frame $\{B\}$ relative to $\{U\}$, expressed in $\{B\}$. In what follows, generalized variables for the horizontal plane motion be given by $\boldsymbol{\nu} = \begin{bmatrix} u & v & r \end{bmatrix}'$, $\boldsymbol{\eta} = \begin{bmatrix} x & y & \psi \end{bmatrix}'$ and $\boldsymbol{\tau} = \begin{bmatrix} X & Y & N \end{bmatrix}'$, which denote the generalized velocity, position and force vectors, respectively. With this notation, the generalized equation for the vehicle kinematics become

$$\dot{\boldsymbol{\eta}} = J(\boldsymbol{\eta})\boldsymbol{\nu},$$

where $J(\boldsymbol{\eta})$ contains the rotation matrix from $\{B\}$ to $\{U\}$. The generalized equation of motion for the dynamics can be written as

$$M\,\dot{\boldsymbol{\nu}} + C(\boldsymbol{\nu})\,\boldsymbol{\nu} = \boldsymbol{\tau}(\dot{\boldsymbol{\nu}}, \boldsymbol{\nu}, \mathbf{n})\ ,$$

where M is the 2-D rigid body inertia matrix and C the matrix of Coriolis and centripetal terms. The generalized force $\boldsymbol{\tau}$ includes the added mass terms, the hydrodynamic forces and moments acting on the body, and the forces and moments generated by the propellers as functions of the generalized velocity vector $\boldsymbol{\nu}$ and the actuation vector $\mathbf{n} = [n_c\ n_d]'$. The symbols n_c and n_d stand for the common and differential modes of the propellers' speed of rotation, respectively. In this model, the gravitational and wave effects are neglected and the fluid is assumed to be at rest. A simplified model that captures the effect of constant currents on the ASC dynamics can be obtained by rewriting the above equations in terms of the vehicle's velocity relative to the fluid. It is assumed that the generalized velocity results from the sum of two components $\boldsymbol{\nu} = \boldsymbol{\nu}_r + J(\eta)^{-1}\,\boldsymbol{\nu}_f$, where $\boldsymbol{\nu}_r$ is the vehicle's generalized velocity with respect to the fluid expressed in the body frame $\{B\}$ and $\boldsymbol{\nu}_f$ is the water generalized velocity described in the inertial frame $\{U\}$. Notice that the third element of $\boldsymbol{\nu}_f$ is zero.

Using $\boldsymbol{\nu}_r$ instead of $\boldsymbol{\nu}$, assuming that the generalized force $\boldsymbol{\tau}$ is given by $\boldsymbol{\tau}(\dot{\boldsymbol{\nu}}_r, \boldsymbol{\nu}_r, \mathbf{n}) = M_2\dot{\boldsymbol{\nu}}_r + \boldsymbol{\tau}_1(\boldsymbol{\nu}_r, \mathbf{n})$ and that $M_3 = M - M_2$, the kinematic and dynamic equations of motion can be rewritten as

$$\dot{\boldsymbol{\eta}} = J(\boldsymbol{\eta})\boldsymbol{\nu}_r + \boldsymbol{\nu}_f\ , \tag{1}$$

$$\dot{\boldsymbol{\nu}}_r = M_3^{-1}\,\boldsymbol{\tau}_2(\boldsymbol{\nu}_r, \mathbf{n})\ , \tag{2}$$

where $\boldsymbol{\tau}_2(\boldsymbol{\nu}_r, \mathbf{n}) = -C(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \boldsymbol{\tau}_1(\boldsymbol{\nu}_r, \mathbf{n})$.

A. Generalized Error Dynamics

This section introduces the concepts of trimming trajectories for the ASC dynamic model, presents a generalized error space to describe the vehicles's motion about trimming trajectories in the absence of currents, and computes explicitly the dynamics of the vehicle in the new error space.

Consider the equations of motion presented in (1) and (2) without the effect of constant currents (that is $\boldsymbol{\nu}_f = \mathbf{0}$), yielding $\boldsymbol{\nu}_r = \boldsymbol{\nu}$, and let $\boldsymbol{\nu}_c$, $\boldsymbol{\eta}_c$, and $\mathbf{n}_c$ denote the trimming values of the state and input vectors. At trimming, the generalized velocity satisfies $\dot{\boldsymbol{\nu}}_c = \mathbf{0}$, implying that $\dot{\mathbf{n}}_c = \mathbf{0}$. It can be shown that the trimming trajectories are straight lines and circles described by the vehicle at constant speed. These trajectories can be fully described by the parameter vector $\boldsymbol{\xi} = [V_c\ r_c]'$ where $V_c = \|\mathbf{v}_c\|_2 = \sqrt{u^2 + v^2}$. Therefore, $\boldsymbol{\xi}$ fully parameterizes the set of achievable trimming trajectories $\mathcal{E}$.

The generalized error vector between the vehicle state and the desired trajectory can be defined using the nonlinear transformation

$$\mathbf{x}_e = \begin{bmatrix} \boldsymbol{\nu}_e \\ \boldsymbol{\eta}_e \\ \mathbf{x}_i \end{bmatrix} = \begin{bmatrix} \boldsymbol{\nu} - \boldsymbol{\nu}_c \\ J^{-1}(\boldsymbol{\eta})\,(\boldsymbol{\eta} - \boldsymbol{\eta}_c) \\ \int_0^t \Pi\,\boldsymbol{\eta}_e dt \end{bmatrix},$$

where Π denotes the projection matrix $\Pi = \begin{bmatrix} \mathbf{I}_{2\times 2} & \mathbf{0}_{2\times 1} \end{bmatrix}$. In the new coordinates, the error dynamics can be expressed as

$$\left\{ \begin{array}{rcl} \dot{\boldsymbol{\nu}}_e &=& \dot{\boldsymbol{\nu}} \\ \dot{\boldsymbol{\eta}}_e &=& \boldsymbol{\nu} - J^{-1}(\boldsymbol{\eta}_e)\,\boldsymbol{\nu}_c - Q(\boldsymbol{\nu})\,\boldsymbol{\eta}_e \\ \dot{\mathbf{x}}_i &=& \Pi\boldsymbol{\eta}_e \end{array} \right. , \tag{3}$$

where $Q(\boldsymbol{\nu}_r) = S\left([0\ 0\ r]'\right)$ and $S(\mathbf{a})$ stands for the skew-symmetric matrix that verifies $S(\mathbf{a})\mathbf{b} = \mathbf{a} \times \mathbf{b}$. Using (3), it is straightforward to show that the linearization of the error dynamics about $\boldsymbol{\nu}_e = 0$ and $\mathbf{n} = \mathbf{n}_c$ is time invariant. As explained later in the paper, the implementation of this error space uses the velocity relative to the fluid $\boldsymbol{\nu}_r$ instead of the velocity relative to the inertial frame $\boldsymbol{\nu}$, that is, $\boldsymbol{\nu}_e \simeq \boldsymbol{\nu}_r - \boldsymbol{\nu}_c$. This approximation is a consequence of assuming that measurements for $\boldsymbol{\nu}$ are not available.

B. Intrinsic Input Saturation

There are complex physical constraints that can be easily incorporated in the nonlinear model of the vehicle, instead of defining them as a nonlinear constraint function in the optimization problem. This is the case for the common and differential inputs of the catamaran model.

Let the new inputs $\bar{\mathbf{n}} = [\bar{n}_c\ \ \bar{n}_d]'$ be defined as smoothly saturated functions of the regular inputs $\mathbf{n} = [n_c \quad n_d]'$, so that the dynamic equation is now given by $\dot{\boldsymbol{\nu}}_r =$

$M_3^{-1}\,\boldsymbol{\tau}_2(\boldsymbol{\nu}_r, \bar{\mathbf{n}}(\mathbf{n}))$. The saturation functions are derived from the basic function $\bar{a}(a) = \frac{a}{1+|a|}$, applying translations and scaling both to the function and its derivative, such that inside the bounds $\bar{a} = a$ and outside the bounds $\bar{a}$ tends smoothly to the maximum value a_{max} or the minimum value a_{min}. Since the motors used in the type of vehicle considered in this work do not invert their thrust in normal operation, it is necessary to impose a minimum value for the common mode input $\mathbf{n}_c$, and also bounds for the differential input. The saturation function of the common mode input is defined as

$$\bar{n}_c(n_c) = \begin{cases} n_c & , \ \underline{\epsilon}_c \leq n_c \leq \overline{\epsilon}_c \\ \overline{\epsilon}_c + \frac{n_c - \overline{\epsilon}_c}{1 + \frac{n_c - \overline{\epsilon}_c}{\epsilon}} & , \ n_c > \overline{\epsilon}_c \\ \underline{\epsilon}_c + \frac{n_c - \underline{\epsilon}_c}{1 - \frac{n_c - \underline{\epsilon}_c}{\epsilon}} & , \ n_c < \underline{\epsilon}_c \end{cases} ,$$

with $\overline{\epsilon}_c = n_{c_{max}} - \epsilon$ and $\underline{\epsilon}_c = n_{c_{min}} + \epsilon$, where $0 < \epsilon < 1$ is a constant (typically 0.01), that defines the length of the smooth transition. The saturation of the differential input is given by the function $\bar{n}_d(\bar{n}_c, n_d)$, obtained using the same approach presented above, noting that it also depends on the saturated common mode input $\bar{n}_c$.

In brief, considering that $\mathbf{n} \in \mathcal{N} \subset \mathbb{R}^{n_n}$, the procedure described above defines the new saturated input vector as $\bar{\mathbf{n}} \in \mathbb{R}^{n_n}$, simplifying the optimization problem formulation.

C. Discretization and Delay Modeling

The ASC error model described in (3) can be rewritten as

$$\dot{\mathbf{x}}_e(t) = \mathbf{f}_e\left(\mathbf{x}_e(t), \mathbf{n}_e(t)\right) \ ,$$

where $\mathbf{n}_e = \mathbf{n} - \mathbf{n}_c$. Since the control problem is formulated as a discrete-time open loop optimal control problem, the equations of motion of the vehicle are described as a difference equation, which can be obtained using forward Euler discretization, yielding

$$\begin{aligned} \mathbf{x}_e((k+1)T_s) &\approx \mathbf{x}_e(kT_s) + T_s \mathbf{f}_e\left(\mathbf{x}_e(kT_s), \mathbf{n}_e(kT_s)\right) \\ &= \mathbf{f}_d\left(\mathbf{x}_e(k\,T_s), \mathbf{n}_e(k\,T_s)\right) \ , \end{aligned}$$

where T_s is the sample time. The previous equation can be rewritten in compact form as

$$\mathbf{x}_{e_{k+1}} \approx \mathbf{f}_d\left(\mathbf{x}_{e_k}, \mathbf{n}_{e_k}\right) \ .$$

To model the delay between the instants the state variables $\mathbf{x}_{e_k}$ are measured and the instants a new control action $\mathbf{n}_{e_{k+1}}$ is made available, the model is augmented with an extra delay state. Considering the new state vector $\mathbf{x}_k = [\mathbf{x}'_{e_k} \ \mathbf{x}'_{n_k}]'$ and the input vector $\mathbf{u}_k = \mathbf{n}_{e_k}$, the discrete-time model takes the form

$$\mathbf{x}_{k+1} = \mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right) \ , \tag{4}$$

where $\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right) = \left[\ \mathbf{f}_d\left(\mathbf{x}_{e_k}, \mathbf{x}_{n_k}\right)' \quad \mathbf{n}'_{e_k} \ \right]'$.

III. Model Predictive Control Problem

In this section the Model Predictive Control problem is formulated as a discrete-time open-loop optimal control problem with finite horizon, subject to the discrete nonlinear model equations and the state and input saturation constraints. Recalling (4), the vehicle dynamics can be modeled as a discrete-time state-space equation with state $\mathbf{x}_k \in \mathcal{X}$ and input $\mathbf{u}_k \in \mathcal{U}$, where $\mathcal{X} \subset \mathbb{R}^{n_x}$ and $\mathcal{U} \subset \mathbb{R}^{n_u}$ denote the sets of admissible state and control vectors, respectively. Let N be the prediction horizon of the control problem, $U_k = \{\mathbf{u}_k, \ldots, \mathbf{u}_{k+N-1}\}$ the sequence of control inputs at iteration k, and $X_k = \{\mathbf{x}_k, \ldots, \mathbf{x}_{k+N}\}$ the sequence of state vectors generated by that control sequence. The saturation constraints for the state and input sequences are defined by the conditions $X_k \in \mathfrak{X}_N$ and $U_k \in \mathfrak{U}_N$, where $\mathfrak{X}_N = \{X_k : \mathbf{x}_i \in \mathcal{X}, \forall_{i=k,\ldots,k+N}\}$ and $\mathfrak{U}_N = \{U_k : \mathbf{u}_i \in \mathcal{U}, \forall_{i=k,\ldots,k+N-1}\}$. Using (4), the model constraint for an horizon of N steps ahead can be written as

$$F_M(X_k, U_k) = \begin{bmatrix} \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) - \mathbf{x}_{k+1} \\ \vdots \\ \mathbf{f}(\mathbf{x}_{k+N-1}, \mathbf{u}_{k+N-1}) - \mathbf{x}_{k+N} \end{bmatrix} = \mathbf{0} \ .$$

Given these constraints, the MPC problem can be defined as

$$U_k^* = \arg\min_{U_k} J_k \tag{5}$$

$$s.t. \quad X_k \in \mathfrak{X}_N \ , \ U_k \in \mathfrak{U}_N \tag{6}$$

$$F_M(X_k, U_k) = 0 \tag{7}$$

where $J_k = F_{k+N} + \sum_{i=k}^{k+N-1} L_i$, $F_i = \frac{1}{2}\,\mathbf{x}'_i\,P\,\mathbf{x}_i$, $L_i = \frac{1}{2}\,[\mathbf{x}'_i\,Q\,\mathbf{x}_i + \mathbf{u}'_i\,R\,\mathbf{u}_i]$ and P, Q, and R are symmetric positive definite matrices. In brief, the MPC objective is to find, at each iteration k, the optimal control sequence U_k^* with horizon N, such that the resulting state sequence X_k^* together with U_k^* minimize the cost functional J_k without violating the state and input constraints imposed by (6).

Following a by now standard approach, the constrained optimization problem presented above can be solved by reformulating it as an unconstrained optimization problem and using gradient methods to estimate the optimal solution. While constraint (7) is eliminated using lagrange multipliers, constraint (6) is incorporated into the cost functional by resorting to penalty methods.

A. State and Input Saturation Constraint

The saturation constraints defined in (6) are included in the optimization problem to complement the intrinsic input constraints described in Section II-B and thus enable the definition of mission specific bounds for both the state and input vectors. These constraints can be incorporated in the cost functional as a penalty function $\mathbf{f}_R\left(\mathbf{x}, \mathbf{u}\right)$, which is zero-valued for $\mathbf{x} \in \mathcal{X}$ and $\mathbf{u} \in \mathcal{U}$ and behaves as a quadratic function outside these sets. Assuming that the feasibility sets for state and input vectors are given by $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^{n_x} : x^{(j)}_{\min} \leq x^{(j)} \leq x^{(j)}_{\max} \ \forall_{j=1,\ldots,n_x}\}$ and $\mathcal{U} = \{\mathbf{u} \in \mathbb{R}^{n_u} : u^{(l)}_{\min} \leq u^{(l)} \leq u^{(l)}_{\max} \ \forall_{l=1,\ldots,n_u}\}$, respectively, the penalty function can be defined as

$$\begin{aligned} \mathbf{f}_R\left(\mathbf{x}, \mathbf{u}\right) = & \frac{1}{2}\sum_{j=1}^{n_x} h^2(|x^{(j)} - x^{(j)}_{\text{center}}| - x^{(j)}_{\text{range}})\, w_x^{(j)} \\ & + \frac{1}{2}\sum_{l=1}^{n_u} h^2(|u^{(l)} - u^{(l)}_{\text{center}}| - u^{(l)}_{\text{range}})\, w_u^{(l)}, \end{aligned}$$

where $w_x^{(j)}$ and $w_u^{(l)}$ are positive scalar weights, $a_{\text{center}}^{(j)} = (a_{\max}^{(j)} + a_{\min}^{(j)})/2$, $a_{\text{range}}^{(j)} = a_{\max}^{(j)} - a_{center}^{(j)}$ and

$$h(a) = \begin{cases} a & \text{, if } a > 0 \\ 0 & \text{, otherwise} \end{cases} .$$

B. Unconstrained Optimization Problem

Adding to the optimization cost functional the state and input saturation constraint, the new problem can be written as

$$U_k^* = \arg\min_{U_k} \bar{J}_k \quad (8)$$

$$s.t. \qquad F_M(X_k, U_k) = 0 \quad (9)$$

where $\bar{J}_k = \bar{F}_{k+N} + \sum_{i=k}^{k+N-1} \bar{L}_i$,, $\bar{F}_i = F_i + \mathbf{f}_R(\mathbf{x}_i, 0)$ and $\bar{L}_i = L_i + \mathbf{f}_R(\mathbf{x}_i, \mathbf{u}_i)$. The elimination method using lagrange multipliers is used to solve the model constraint (9). Introducing the lagrange multiplier sequence $\Lambda_k = \{\boldsymbol{\lambda}_{k+1}, \ldots, \boldsymbol{\lambda}_{k+N}\}$ and the Hamiltonian $H_i = H(\mathbf{x}_i, \mathbf{u}_i) = \bar{L}_i + \boldsymbol{\lambda}_{i+1}' \mathbf{f}(\mathbf{x}_i, \mathbf{u}_i)$, the cost functional $\bar{J}$ can be rewritten as

$$\bar{J}_k = \bar{F}_{k+N} - \boldsymbol{\lambda}_{k+N}' \mathbf{x}_{k+N} + \sum_{i=k+1}^{k+N-1} \left[H_i - \boldsymbol{\lambda}_i' \mathbf{x}_i\right] + H_k.$$

For a fixed initial state $\mathbf{x}_k$, the first order conditions of optimality yield

$$\frac{\partial \bar{J}_k}{\partial \mathbf{x}_i} = \frac{\partial H_i}{\partial \mathbf{x}_i} - \boldsymbol{\lambda}_i = 0 \ , \ \forall_{i=k+1,\ldots,k+N-1} \ , \quad (10)$$

$$\frac{\partial \bar{J}_k}{\partial \mathbf{x}_{k+N}} = \frac{\partial \bar{F}_{k+N}}{\partial \mathbf{x}_{k+N}} - \boldsymbol{\lambda}_{k+N} = 0 \, , \quad (11)$$

$$\frac{\partial \bar{J}_k}{\partial \mathbf{u}_i} = \frac{\partial H_i}{\partial \mathbf{u}_i} = 0 \ , \ \forall_{i=k,\ldots,k+N-1} \, , \quad (12)$$

where $\frac{\partial H_i}{\partial \mathbf{u}_i} = \frac{\partial \bar{L}_i}{\partial \mathbf{u}_i} + \frac{\partial \mathbf{f}_i}{\partial \mathbf{u}_i} \boldsymbol{\lambda}_{i+1}$ and $\frac{\partial H_i}{\partial \mathbf{x}_i} = \frac{\partial \bar{L}_i}{\partial \mathbf{x}_i} + \frac{\partial \mathbf{f}_i}{\partial \mathbf{x}_i} \boldsymbol{\lambda}_{i+1}$.

Since the lagrange multipliers sequence multiplies zero value terms in the cost functional, $\boldsymbol{\lambda}_{i+1}(\mathbf{f}_i - \mathbf{x}_{i+1})$, they can be arbitrarily chosen. In particular, by choosing $\boldsymbol{\lambda}_{k+N} = \frac{\partial \bar{F}_{k+N}}{\partial \mathbf{x}_{k+N}}$ and $\boldsymbol{\lambda}_i = \frac{\partial H_i}{\partial \mathbf{x}_i}$, for all $i = k+1, \ldots, k+N-1$, the first order conditions of optimality reduce to (12).

An iterative algorithm based on the first order gradient method can be applied to estimate U_k^*, whereby at each iteration j, the control sequence is updated according to

$$U_k^{(j+1)} = U_k^{(j)} + s^{(j)} \, \Delta_k^{(j)} \ , \quad (13)$$

where the step size is denoted by $s^{(j)}$ and the search direction by $\Delta_k^{(j)}$. The optimization algorithm can be summarized as follows.

Algorithm 1: Minimization algorithm for the MPC unconstrained problem.

1) Initialize $X_k^{(0)}$, $U_k^{(0)}$ and $j = 0$;
2) Compute $\{\boldsymbol{\lambda}_i\}$, $i = k+N, \ldots, k$;
3) Compute $\left\{\frac{\partial H_i}{\partial \mathbf{u}_i}\right\}$, $i = k, \ldots, k+N-1$;
4) Compute the search direction $\Delta_k^{(j)}$;
5) Compute the step size $s^{(j)}$ using Wolfe conditions;
6) Compute $U_k^{(j+1)}$ using (13) and $X_k^{(j+1)} = \{\mathbf{x}_i\}$ using $\mathbf{x}_{i+1} = \mathbf{f}(\mathbf{x}_i, \mathbf{u}_i)$, for $i = k, \ldots, k+N-1$;
7) Test $\|\nabla \bar{J}_k^{(j)}|_{(j)}\| < \varepsilon$: if false set $j = j+1$ and go to step (2), if true let $\hat{U}_k = U_k^{(j+1)}$ be the final estimate, apply first input vector $\hat{\mathbf{u}}_k$ to system and set the next initial solution to $U_{k+1}^{(0)} = \{\hat{\mathbf{u}}_{k+1}, \ldots, \hat{\mathbf{u}}_{k+N-1}\}$.

The search direction is obtained using the quasi-Newton method, $\Delta_k^{(j)} = -D^{(j)} \frac{\partial \mathbf{H}_k^{(j)}}{\partial U_k^{(j)}}$, where $\mathbf{H}_k^{(j)}$ is the sequence of the Hamiltonian functions $H_i^{(j)}$ for $i = k, \ldots, k+N-1$ and $D^{(j)}$ is an estimate of the inverse matrix of the second-order derivative of the Hamiltonian [11].

The line search optimization subproblem is numerically solved using the Wolfe rule. This approach guarantees a decrease of the cost functional, as the well known Armijo rule does, and ensures reasonable progress by ruling out unacceptably short steps [11]. Consider the step size optimization subproblem defined by

$$s^{(j)} = \arg\min_{s \geq 0} \phi(s) \ ,$$

where $\phi(s) = \bar{J}_k^{(j+1)} = \bar{J}\left(X_k^{(j+1)}, U_k^{(j+1)}\right)$, with $U_k^{(j+1)}$ as in (13) and its derivative is given by

$$\phi'(s) = \frac{d\,\phi(s)}{ds} = \frac{d\,\bar{J}_k^{(j+1)}}{ds} \ .$$

Let also $\mu_i = \phi(0) + \sigma\,\phi'(0)\,s_i$ and $\mu_0 = \lambda\,\phi'(0)$, where σ and λ are parameters of the search algorithm. The Wolfe conditions classify a step size according to the sets

$$\mathcal{A} = \{s_i > 0 : \phi(s_i) \leq \mu_i \ \wedge \ \phi'(s_i) \geq \mu_0\}$$
$$\mathcal{D} = \{s_i > 0 : \phi(s_i) > \mu_i\}$$
$$\mathcal{E} = \{s_i > 0 : \phi(s_i) \leq \mu_i \ \wedge \ \phi'(s_i) < \mu_0\}$$

that define the acceptable, the right unacceptable and the left unacceptable step sizes, respectively, and finds an acceptable step size, i.e. an estimate of the optimal step size.

IV. Simulation Results

Real-time implementation of a nonlinear MPC methodology for trajectory tracking has the potential to yield a reliable control system that can meet the physical constraints of the ASC, even in the presence of ocean currents. In this section, the performance of the nonlinear MPC controller introduced above is evaluated in simulation by drawing a comparison with the results obtained with a basic LQR gain switching methodology (see [10] for a similar approach using $\mathcal{H}_2$ synthesis). The main disadvantage of the LQR gain switching methodology, as well as of $\mathcal{H}_2$, $\mathcal{H}_\infty$ and other gain switching/scheduling techniques, is the need for prior computation of the gain matrices for a limited number of operating points (or regions), increasing the sensitivity of the closed-loop system to parametric uncertainties or external disturbances. Given that the actual state measure is used to infer the operating point of the vehicle at each instant of time, the selection of the proper gain assumes a relevant role in the performance of this controller, and failing to select the adequate gain may lead to instability. In the MPC algorithm there is no a priori computation since it relies on the full nonlinear model of the vehicle. Additionally, the

MPC algorithm is able to cope with the physical limitations of the vehicle, such as actuation saturations, and also with additional state or actuation constraints that may be specific to the mission at hand.

In the results presented hereafter, the ASC nonlinear model described in Section II is parameterized for the DelfimX Catamaran and used both in the MPC control algorithm and plant simulation. The simulations where carried out in an Intel Pentium Centrino processor at 1.7 GHz, using Matlab/Simulink with C mex-functions. The reference trajectory was selected to illustrate the behavior of the control algorithms in extreme conditions, which include discontinuities in the reference velocities and a non zero initial position error. The reference trajectory is composed of three different sections: $i)$ a straight line ($\|\mathbf{v}_c\| = 1$ m/s and $r_c = 0$ rad/s), to be tracked between $\boldsymbol{\eta}_c = \mathbf{0}_{3\times 1}$ and $\boldsymbol{\eta}_c = [72\ 0\ 0]'$ with initial condition $\boldsymbol{\nu}_0 = [0.6\ 0\ 0]'$ and $\boldsymbol{\eta}_0 = [0\ -1\ -\pi/4]'$; $ii)$ a quarter of a circle ($\|\mathbf{v}_c\| = 1.5$ m/s and $r_c = -0.0524$ rad/sec); and $iii)$ a complete circle ($\|\mathbf{v}_c\| = 1$ m/s and $r_c = 0.0279$ rad/sec). The sample time is $T_s = 0.2$ s and the horizon is $N = 30$ sample times, or equivalently 6 seconds. The precision of the solution is determined by the algorithm stop conditions, e. g., $|\bar{J}_k^{(j)} - \bar{J}_k^{(j-1)}| < 10^{-2}$.

Two different situations were simulated in order to highlight the major differences between the LQR and MPC controllers: $i)$ trajectory-tracking without current; and $ii)$ trajectory-tracking with a constant current ($\boldsymbol{\nu}_f = [-0.1\ 0.2\ 0]'$ m/s). The simulation results of these two situations are presented in Figures 1 and 2, respectively, that show the trajectories described by the ASC using the MPC and LQR controllers, as well as the time evolution of the position error and the actuation. In both cases, it was assumed that the velocity measurement available for feedback was relative to the fluid and not to the inertial frame. It can be seen that both control methodologies achieve the tracking objective, with or without constant current. However, the LQR method presents larger excursions in actuation both at the initial stage and during the transitions between sections, which generally translate into larger position errors. Moreover, the control effort demanded by the LQR method is far greater than that of the MPC, and it even violates the conditions for valid operation (by having $|n_d| > n_c$). The major limitation of the MPC method is the computation time needed to determine the next control action, which has to be smaller than the sampling time $T_s = 0.2$ s. For the specified simulations, this threshold was never exceeded, since the average and maximum CPU times obtained in the absence of currents were 0.016 s and 0.14 s, respectively, whereas in the presence of a constant current these values slightly increased to 0.023 s and 0.16 s, respectively. The highest CPU times are obtained in the initial instants and in the transitions between parts of the reference trajectory, where the algorithm is not close to the optimal solution. This fact indicates that the maximum CPU time can be reduced in real mission scenarios by the appropriate choice of initial conditions and the generation of smoother reference signals.

In brief, the results indicate that the proposed MPC control algorithm is adequate for real time implementation, demonstrating the ability to track a demanding trajectory while rejecting constant current disturbances and keeping the actuation signals within their limits of operation.

V. Conclusions

This paper presented a MPC-based strategy for motion control of ASCs under the effect of constant currents.

In contrast to the standard approach in MPC literature, the actuation constraints were incorporated into the model so that every control action provided by the MPC algorithm is always valid and the overall computational time is thereby reduced. In addition, following a standard approach the constrained optimization problem was reformulated as an unconstrained optimization problem using the penalty method to accommodate the state saturation constraints and lagrange multipliers to eliminate the ASC dynamic model constraint. The optimization problem was then solved using an iterative algorithm that relies on the quasi-Newton method to find the search direction and the Wolfe conditions to find an estimate of the optimal step size. The simulation results were obtained using the nonlinear ASC model both in the MPC algorithm and in the real plant simulation. These results show that the presented solution can steer the vehicle along a demanding reference trajectory and in the presence of constant currents. The computation times observed in the simulation also indicate that this methodology is well suited for real time implementation.

As noted in [14], for some vehicles the use of trajectory-tracking may impose performance bounds on the controlled system. Since some of these bounds are not present when using path-following, and bearing in mind mission scenarios with no time-critical requirements, further modifications shall include the formulation of a path-following error space. Additionally, it is important that the model of the vehicle include wave disturbances in order to test the control algorithm performance under more realistic scenarios.

References

[1] I. Kaminer, A. Pascoal, E. Hallberg, and C. Silvestre, "Trajectory tracking for autonomous vehicles: An integrated approach to guidance and control," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 1, pp. 29–38, January 1998.

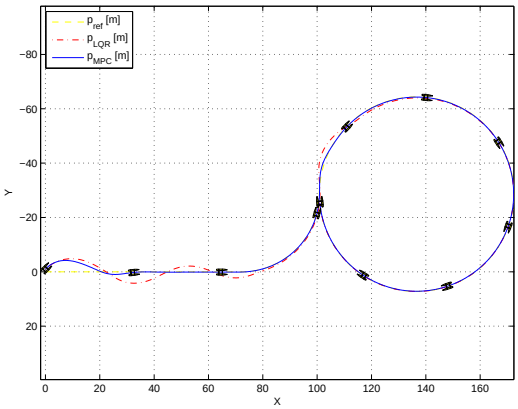
[2] C. Silvestre, A. Pascoal, and I. Kaminer, "On the design of gain-scheduled trajectory tracking contollers," *International Journal of Robust and Nonlinear Control*, vol. 12, no. 9, pp. 797–839, July 2002.

[3] W. J. Rugh and J. S. Shamma, "Research on gain scheduling," *Automatica*, vol. 36, no. 10, pp. 1401–1425, October 2000, survey Paper.

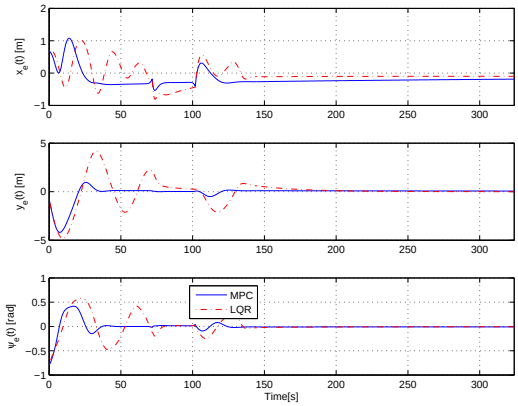
[4] N. Paulino, C. Silvestre, and R. Cunha, "Affine parameter-dependent preview control for rotorcraft terrain following," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 29, no. 6, pp. 1350–1359, 2006.

[5] P. Gomes, C. Silvestre, A. Pascoal, and R. Cunha, "A coastline following preview controller for the delfimx vehicle," in *16th International Offshore and Polar Engineering Conference*, Lisbon, Portugal, July 2007.

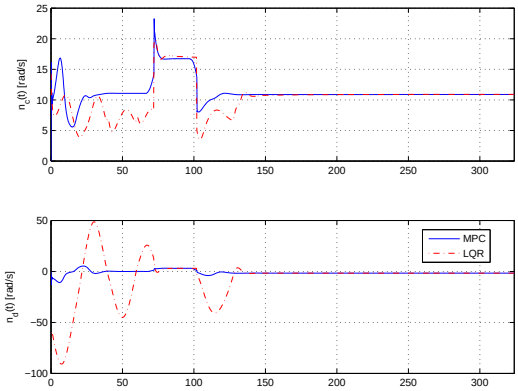
[6] P. Encarnação, A. Pascoal, and M. Arcak, "Path following for autonomous marine craft," in *5th IFAC Conference on Marine Craft Maneuvering and Control*, Aalborg, Denmark, August 2000, pp. 117–122.



(a) Trajectory

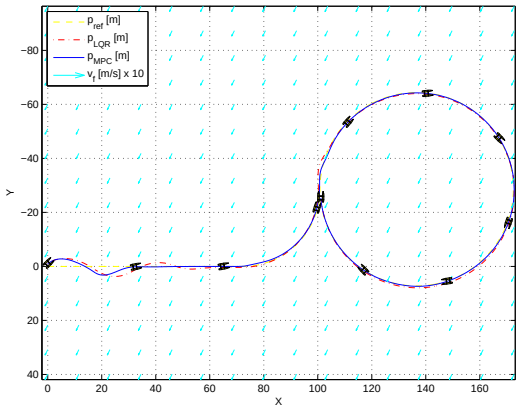


(b) Generalized position error

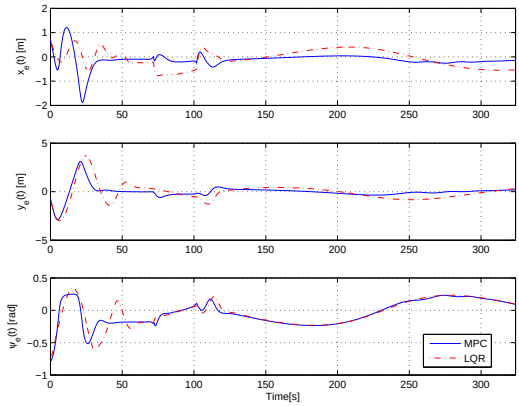


(c) Actuation

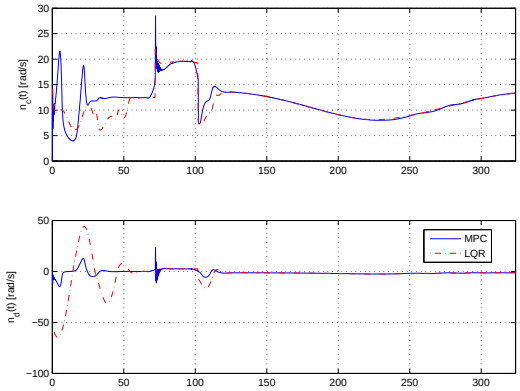
Fig. 1. Simulation without current



(a) Trajectory



(b) Generalized position error



(c) Actuation

Fig. 2. Simulation with current ($\boldsymbol{\nu}_f = [-0.1\ 0.2\ 0]$ m/s)

[7] K. D. Do, Z. P. Jiang, and J. Pan, "Robust adaptive path following of underactuated ships," *Automatica*, vol. 40, no. 6, pp. 929–944, June 2004.

[8] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, pp. 790–814, 2000, survey Paper.

[9] G. Sutton and R. Bitmead, "Computational implementation of nonlinear model predictive control to nonlinear submarine," in *Nonlinear Model Predictive Control*, ser. Progress in Systems and Control Theory, F. Allgöwer and A. Zheng, Eds. Basel-Boston-Berlin: Birkhäuser Verlag, 2000, vol. 26, pp. 461–471.

[10] B. Guerreiro, C. Silvestre, and R. Cunha, "Terrain avoidance model predictive control for autonomous rotorcraft," in *Proceedings of the 17th IFAC World Congress (IFAC'08)*, Seoul, Korea, July 2008, pp. 1076–1081.

[11] J. Nocedal and S. Wright, *Numerical Optimization*, ser. Springer Series in Operation Reasearch. Springer, 1999.

[12] M. Prado, "Modeling and control of an autonomous oceanographic vehicle," Master's thesis, Instituto Superior Técnico, Lisbon, 2002.

[13] T. I. Fossen, *Guidance and Control of Ocean Vehicles*. New York, USA: Wiley, 1994.

[14] A. P. Aguiar, D. B. Dacic, J. P. Hespanha, and P. Kokotovic, "Pathfollowing or reference-tracking? an answer relaxing the limits to performance," in *5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles*, Lisbon, Portugal, July 2004.