Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Trajectory tracking of autonomous vessels using model predictive control ⋆

Huarong Zheng * **Rudy R. Negenborn** * **Gabriel Lodewijks** *

* *Department of Maritime & Transport Technology, Delft University of Technology, Delft, The Netherlands (e-mail: {H.Zheng-1, R.R.Negenborn, G.Lodewijks}@tudelft.nl)*

Abstract: Autonomous surface vehicles are with increasing popularity being seen in various applications where automatic control plays an important role. In this paper the problem of two-dimensional trajectory tracking for autonomous marine surface vehicles is addressed using Model Predictive Control (MPC). At each time step, the reference trajectories of a vessel are assumed to be known over a finite time horizon; the MPC controller computes the optimal forces and moment the vessel needs in order to track the trajectory in an optimal way. Based on a horizontal 3 degrees of freedom nonlinear scaled vessel model, we present both nonlinear MPC (NMPC), which solves a constrained multi-variable nonlinear programming problem, and linearized MPC (LMPC), which solves a constrained quadratic programming problem through on-line iterative optimization. In the latter case, the model used in LMPC for prediction is obtained from a successive linearization of the nonlinear vessel model. Comparisons on performance and computational complexity of the two approaches are presented. The effectiveness of the MPC formulations in vessel trajectory tracking, especially the ability of explicitly handling constraints, is demonstrated via simulations.

Keywords: Autonomous vessels; trajectory tracking; Nonlinear model predictive control; Linearized model predictive control.

1. INTRODUCTION

The rising need for transportation services and the demand for a higher safety level have been the driving forces for a general trend towards automatic co-pilots or even autopilots of vehicle systems (Kiencke et al., 2006). The past few years has seen an increasing interest in the design and development of autonomous marine craft, see, e.g., (Moreira and Soares, 2011; Fossen, 2011). Autonomous surface vehicles are being used for different applications ranging from environmental or geographical surveying, weather information acquisition, rescue, military, pure research platforms and transport over water. Such an autonomous system, except for the external hardware (e.g., the hull, actuators, all kinds of sensors, etc.), relies on three dependent software blocks which are known as navigation, guidance and control (NGC) systems, illustrated in Figure 1. In this paper, let us assume that all the system states can be measured or estimated through the sensors or observers within the navigation system and that the guidance system can generate a desired trajectory over a finite future time horizon. The focus then is on the control of the vehicle dynamics via actuator inputs to complete certain missions. Normally these missions have both spatial and temporal constraints on the system states, resulting in a so-called trajectory tracking control problem instead of just following a predefined path independent of time, which is typically referred to as path following.

Due to the complexity and nonlinearity of the vessel dynamics as well as varying environmental disturbances, vessel motion control has been challenging and thus drawn a considerable interest in the field. The first recognized and most widely implemented controller until now is still based on classical PID (Proportional-Integral-Derivative) control theory (Minorski, 1922), because of its simplicity both in theoretical analysis and implementation. PID is sufficient for an automatic steering system which is a typical single-input-single-output case. However, for multiple-input-multiple-output systems, PID shows deficiencies. Cascaded frameworks (Lefeber et al., 2003) may be applicable, but system constraints and disturbances are still difficult to be handled. Other more complicated methods such as Lyapunov's direct can be found in Jiang (2002) where an underactuated vessel model with zero diagonal system matrix terms was employed. Do and Pan (2006) eliminated the assumption of zero diagonal terms and designed a robust and adaptive controller combining backstepping and Lyapunov's direct method both theoretically and experimentally on a model ship outdoor. However, controllers mentioned above dealt with path following problems without temporal considerations which will not be sufficient if specific missions or safety issues (e.g., International Regulations for Preventing Collisions at Sea (COLREGs)) are involved. Constraints such as actuator saturation are also excluded. However, in reality, engines can only provide limited power and mechanical components have maximum deflections or revolutions, which in-

⋆ This research is supported by the China Scholarship Council under Grant 201206950021 and the VENI project "Intelligent multi-agent control for flexible coordination of transport hubs" (project 11210) of the Dutch Technology Foundation STW, a subdivision of the Netherlands Organization for Scientific Research (NWO).

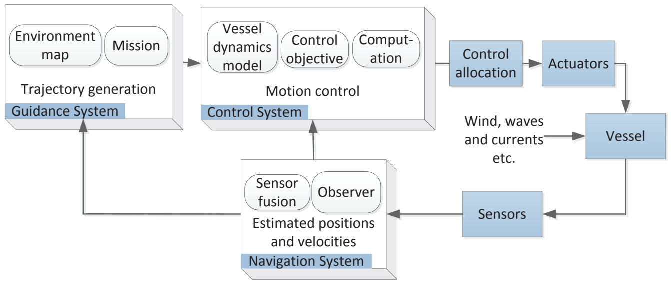


Fig. 1. Diagram of an NGC system for autonomous marine surface vehicles.

troduces constraints to the vessel system; neglecting these constraints in the controller design could lead to poor control performances or even actuator damages in real implementation.

Model Predictive Control (MPC) naturally offers a solution to the control of constrained systems given its capability of handling system constraints in an explicit way. MPC has been widely implemented in industrial processes (Qin and Badgwell, 2003) ever since a first formulation of it was published in the early 1960s (Propoi, 1963). The first application of MPC for vessels in literature is (Wahl and Gilles, 1998). Recently, MPC has been applied to vessel path following (Li et al., 2009) and heading control (Li and Sun, 2012). In (Oh and Sun, 2010), a guidance algorithm LOS (Line-Of-Sight) is integrated in the MPC design which is based on a linearized model. The incorporation of LOS is proved to have a positive effect on the tracking performance with simulation. Pavlov et al. (2009) also applied MPC to the vessel path following problem with guidance algorithm of LOS, but MPC is used there to update the look ahead distance of LOS and has not actually been applied to nonlinear vessel motion control. Analytic nonlinear MPC was proposed for ship path following control in (Wang, 2009) using Lie derivatives and seems to provide an efficient controller for path following by explicitly giving the control laws. However, same with the Lyapunov-based design techniques, constraints are difficult to be incorporated. Furthermore, the applications of MPC in vessel systems mentioned above are confined to path following problems while trajectory tracking, which has wide applications in various scenarios, sees few utilization of this advanced methodology.

The main contribution of this paper lies in the proposal of two different MPC approaches for the vessel trajectory tracking problem and the comparisons between them in terms of tracking performance and computational complexity. One of the approaches is NMPC which directly employs the nonlinear vessel dynamics model as the prediction model and iteratively solves a nonlinear optimization problem online at each time step; the other approach, LMPC, is based on the idea of successive linearization. At each time step, a linear plant model is constructed by linearizing the nonlinear vessel dynamics model around the current operating point and then linear optimization techniques like quadratic programming can be applied. The successive linearization based MPC has been successfully applied to aircraft control (Keviczky and Balas, 2005) and road vehicle lane change control (Zheng et al., 2013), which are similar to our approach in this paper in converting the nonlinear system to a linear one.

This paper is organized as follows. We first describe the vessel dynamical model used both as the plant process and for controller design in Section 2. Then in Section 3, the control problems are formulated when either a nonlinear model or a linearized model is used. Simulation settings, comparisons and results are given in Section 4. Finally, concluding remarks and future research directions are given in Section 5.

2. VESSEL MODEL

An accurate mathematical vessel model is crucial for the development of a control system. A marine surface vessel experiences motions along 6 DOF which are, for convenience, typically described in two coordinate frames: $\{b\} = (x_b,\ y_b,\ z_b)$ and $\{n\} = (x_n,\ y_n,\ z_n)$. $\{b\}$ is the body-fixed reference frame which is moving with the vessel. $\{n\}$ can be approximately seen as the inertial coordinate system for vessels sailing only in a local area. The motions in the horizontal plane are referred to as *surge* (longitudinal motion), *sway* (sideways motion) and *yaw* (rotation around the vertical axis). The other three DOF are *roll* (rotation about the longitudinal axis), *pitch* (rotation about the transverse axis), and *heave* (vertical motion), see Fig. 2.

For ship maneuvering control, it is common to formulate a 3 DOF ship model as a coupled *surge-sway-yaw* model and thus neglect heave, roll and pitch motions (Fossen, 2011). This section describes the horizontal 3 DOF nonlinear dynamics model of Cybership II (as shown in Figure 3), a 1:70 scale replica of a supply ship, developed in the Marine Cybernetics Laboratory at NTNU. Manoeuver tests have been conducted to identify the physical and hydrodynamical quantities for this ship (Skjetne et al., 2004). It is assumed that the vessel moves in calm water experiencing negligible current, wind and waves such that forces caused by environmental disturbances can be excluded. Another assumption is that the craft has homogeneous mass distribution and xz-plane symmetry such that

$$I_{\mathrm{xy}} = I_{\mathrm{yz}} = 0, \tag{1}$$

where I_{xy} and I_{yz} are the moments of inertia about plane xy and yz, respectively. Following a vectorial model representation in (Fossen, 2011), the kinematics and kinetic model of a marine surface vessel can then be written as:

$$\begin{aligned} \dot{\boldsymbol{\eta}}(t) &= \mathbf{T}(\boldsymbol{\eta}(t))\boldsymbol{v}(t) \\ \mathbf{M}\dot{\boldsymbol{v}}(t) + \mathbf{C}(\boldsymbol{v}(t))\boldsymbol{v}(t) + \mathbf{D}\boldsymbol{v}(t) &= \boldsymbol{\tau}(t), \end{aligned} \tag{2}$$

where $\boldsymbol{\eta}(t)$ is the pose (position and orientation) vector in the inertial frame, $\boldsymbol{v}(t)$ is the body-fixed velocity vector and $\boldsymbol{\tau}(t)$ is the control vector. These vectors are given by:

$$\boldsymbol{\eta} := \begin{bmatrix} x \\ y \\ \psi \end{bmatrix}, \quad \boldsymbol{v} := \begin{bmatrix} u \\ v \\ r \end{bmatrix}, \quad \boldsymbol{\tau} := \begin{bmatrix} f_{\mathrm{u}} \\ f_{\mathrm{v}} \\ t_{\mathrm{r}} \end{bmatrix}, \tag{3}$$

where x (m), y (m) are the positions along axis x_n, y_n, respectively, and ψ (rad) is vessel's orientation (heading angle) in the inertial frame; u (m/s), v (m/s) and r (rad/s) are the surge, sway velocities and yaw rate in the body-fixed frame, respectively; f_{u} (N), f_{v} (N) and t_{r} (Nm) are the surge, sway forces and yaw moment produced by the vessel actuators (propellers, thrusters, and rudders). The

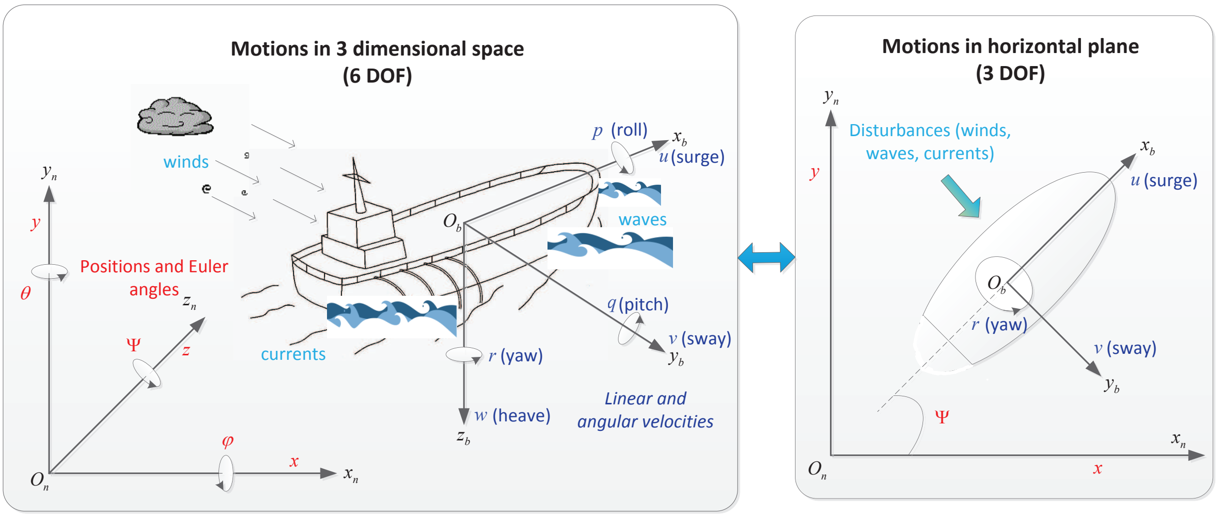


Fig. 2. Vessel motions in 6 DOF and 3 DOF.

Fig. 3. CyberShip II in Marine Cybernetics Laboratory at NTNU (Fossen, 2008).

system matrices $\mathbf{M}$, $\mathbf{C}(\boldsymbol{v})$, and $\mathbf{D}$ are the inertial mass matrix (invertible), Coriolis and centrifugal matrix, and damping matrix, respectively, which are given by:

$$\mathbf{M} := \begin{bmatrix} m_{11} & 0 & 0 \\ 0 & m_{22} & m_{23} \\ 0 & m_{32} & m_{33} \end{bmatrix}, \quad \mathbf{D} := \begin{bmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & d_{23} \\ 0 & d_{32} & d_{33} \end{bmatrix}, \\ \mathbf{C} := \begin{bmatrix} 0 & 0 & -c_{31} \\ 0 & 0 & c_{23} \\ c_{31} & -c_{23} & 0 \end{bmatrix}, \tag{4}$$

where $c_{23} = m_{11}u$ and $c_{31} = m_{22}v + \frac{1}{2}(m_{23} + m_{32})r$. The Jacobian matrix $\mathbf{T}(\boldsymbol{\eta})$ transforms the body-fixed velocities $\boldsymbol{v}$ into the inertial velocities $\dot{\boldsymbol{\eta}}$ and is given by

$$\mathbf{T}(\boldsymbol{\eta}) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5}$$

For modeling as well as experimental details of Cybership II, readers are referred to (Skjetne et al., 2004).

3. MODEL PREDICTIVE CONTROL

MPC, also referred to as model based predictive control or receding horizon control, determines a control action by computing a future control sequence over a defined prediction horizon in such a way that the prediction of the system output is driven close to the reference (Negenborn, 2007). A constrained optimization problem is formulated at each step based on the predictions over a future horizon. A sequence of optimal control inputs can be obtained but only the first element is applied to the system. This procedure is repeated when the time is shifted one step forward, forming the so-called receding horizon approach. Advantages of MPC are that in principle it can take into account all information available and that it can therefore anticipate undesirable situations in the future at an early stage (Maciejowski, 2002). This could be very useful for vessel trajectory tracking since the ship control system, typically strongly nonlinear, susceptible to uncertain parameters and environmental disturbances, does not have a good enough manoeuvrability to respond timely when an emergency happens. Furthermore, for physical dynamics boundaries as well as safe, economical and environmental reasons, there can be various constraints on inputs, states and outputs; MPC can handle those constraints in an explicit way. The characteristics of the ship trajectory tracking control problem make the MPC approach a natural choice.

In this section, we propose two different MPC formulations for the marine surface vessel trajectory tracking problem, namely NMPC and LMPC, based on the vessel dynamical model introduced in the previous section. Basically, the control objective is to get optimal control commands ($\boldsymbol{\tau}(t)$) for actuators so that the difference between the outputs generated by system (2) and the time varying desired trajectory are minimized while all the system constraints are satisfied.

3.1 Trajectory tracking using NMPC

In terms of controller design, it is convenient to rewrite model (2) into a state space format from a control perspective. For NMPC, the state space model is denoted as follows:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= f(\mathbf{x}(t), \mathbf{u}(t)) \\ \mathbf{y}(t) &= g(\mathbf{x}(t)), \end{aligned} \tag{6}$$

where $\mathbf{x} = \begin{bmatrix} \boldsymbol{\eta}^{\mathrm{T}} & \boldsymbol{v}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$, $\mathbf{u} = \boldsymbol{\tau}$ and $f : \mathbb{R}^6 \times \mathbb{R}^3 \to \mathbb{R}^6$ is a nonlinear smooth function, which can be further given as:

$$\begin{bmatrix} \mathbf{T}(\mathbf{p}_1\mathbf{x}(t))(\mathbf{p}_2\mathbf{x}(t)) \\ \mathbf{M}^{-1}(-\mathbf{C}(\mathbf{p}_2\mathbf{x}(t))(\mathbf{p}_2\mathbf{x}(t)) - \mathbf{D}(\mathbf{p}_2\mathbf{x}(t)) + \mathbf{u}(t)) \end{bmatrix}, \tag{7}$$

where $\mathbf{p}_1 = [0\ 0\ 1\ 0\ 0\ 0]$ and $\mathbf{p}_2 = \begin{bmatrix} 0\ 0\ 0\ 1\ 0\ 0 \\ 0\ 0\ 0\ 0\ 1\ 0 \\ 0\ 0\ 0\ 0\ 0\ 1 \end{bmatrix}$. In this paper, for the problem of trajectory tracking, the objective is to minimize the deviations of the vessel's real positions from the reference positions. Therefore the output map $g(\mathbf{x}(t))$ is defined as:

$$g(\mathbf{x}(t)) = \begin{bmatrix} 1\ 0\ 0\ 0\ 0\ 0 \\ 0\ 1\ 0\ 0\ 0\ 0 \end{bmatrix} \mathbf{x}(t) \tag{8}$$

In numerical simulations, the continuous differential model (6) still cannot be used directly to predict open loop state values along the prediction horizon in MPC controllers. Proper sampling is needed to obtain discrete-time dynamics for prediction. Zero order hold in which each element of the control sequence is assumed to be constant during the time interval of $[(k-1)T_\mathrm{s}, kT_\mathrm{s}]$ (k=1, 2,...,N, where N is the prediction horizon and T_s is the sampling time) gives approximate solutions to the ordinary differential equations (ODE) of (6):

$$\begin{aligned} \mathbf{x}(n+1) &= f_\mathrm{d}(\mathbf{x}(n), \mathbf{u}(n)) \\ \mathbf{y}(n) &= g_\mathrm{d}(\mathbf{x}(n)), \end{aligned} \tag{9}$$

where f_d and g_d stand for discretized maps. Available ODE solvers such as ODE45 (Senan, 2012) are able to provide solutions within certain error tolerances given a proper sampling time. Therefore, based on the current measured or estimated states, future predictions can be calculated interactively as the functions of control sequences $\mathbf{u}(k+n)$ (n=0, 2, ..., N-1). The tracking problem is then represented by solving a constrained finite horizon nonlinear programming problem. A minimal trajectory tracking error as well as a minimal control effort at each simulation step k can be realized, reflected by the first and second term in the following cost function, respectively:

$$\begin{aligned} &\min_{\tilde{\mathbf{u}}(k)} J(\tilde{\mathbf{x}}(k),\ \tilde{\mathbf{u}}(k)) = \\ &\sum_{n=1}^{N} (\hat{\mathbf{y}}(k+n) - \mathbf{y}_\mathrm{ref}(k+n))^\mathrm{T} \mathbf{Q}((\hat{\mathbf{y}}(k+n) - \mathbf{y}_\mathrm{ref}(k+n))) \\ &+ \sum_{n=0}^{N-1} \mathbf{u}(k+n)^\mathrm{T} \mathbf{R} \mathbf{u}(k+n) \end{aligned} \tag{10}$$

subject to

$$\mathbf{x}(k) = \mathbf{x}_\mathrm{measure}(k) \tag{11}$$
$$\mathbf{x}(k+n+1) = f_\mathrm{d}(\mathbf{x}(k+n), \mathbf{u}(k+n)) \tag{12}$$
$$\hat{\mathbf{y}}(k+n) = g(\mathbf{x}(k+n)) \tag{13}$$
$$\mathbf{y}_\mathrm{min} \le \hat{\mathbf{y}}(k+n) \le \mathbf{y}_\mathrm{max}, \text{for } n = 1, 2, ..., N \tag{14}$$
$$\mathbf{u}_\mathrm{min} \le \mathbf{u}(k+n) \le \mathbf{u}_\mathrm{max}, \text{for } n = 0, 1, ..., N-1, \tag{15}$$

where $\tilde{\mathbf{x}}(k)$ and $\tilde{\mathbf{u}}(k)$ are the state and control input matrices over the prediction horizon (future N steps) at the kth step; $\mathbf{Q}$ and $\mathbf{R}$ are weighting matrices of appropriate dimensions; $\hat{\mathbf{y}}(k+n)$ stands for the nth future prediction output vector at prediction step k. Although we obtain a sequence of optimal results ($\mathbf{u}(k)$, $\mathbf{u}(k+1)$, ..., $\mathbf{u}(k+N-1)$) by solving the nonlinear optimization problem (10)–(15), only the first element of the sequence, i.e., $\mathbf{u}(k)$, is applied to the process plant. At each decision step, we go through Algorithm 1 in case of the nonlinear MPC scheme.

Algorithm 1 Nonlinear MPC algorithm

1: Measure current states $\boldsymbol{x}(k)$ of the system and compute future predicted outputs $\hat{\mathbf{y}}(k+n)$, $n = 1, ..., N$ as functions of future control inputs $\mathbf{u}(k+n)$, $n = 1, ..., N-1$
2: Solve the nonlinear programming problem (10) (using MATLAB function $fmincon$)and get the optimal control sequence $\mathbf{u}(k+n)$, $n = 1, ..., N-1$
3: Apply the first element of the above optimal solution, i.e., $\mathbf{u}(k)$ to system with an initial state measured at kth simulation step measure(k) during the time interval $[kT_\mathrm{s}, (k+1)T_\mathrm{s}]$
4: Save system data and shift both the control and reference sequences one step forward
5: At next simulation step $k+1$, go to step 1

Note that time varying references are sampled with the same sampling time T_s as well. At each step, a sequence of reference signals with a length of the prediction horizon is provided to the cost function in (10) and at the next simulation step, the reference sequence will be shifted one step forward in Algorithm 1 step 4.

3.2 Trajectory tracking using LMPC

NMPC directly uses the nonlinear vessel trajectory tracking model which can reflect the system dynamics better. However, the computational burden can be a main obstacle, since a constrained nonlinear optimization problem has to be solved over a finite prediction horizon at each step. Our second approach overcomes this problem by using a linearized system model in the scheme of MPC. This linearized model is obtained at the beginning of each decision step through successive online linearization of the nonlinear system model around the current operating point. In this way, the prediction model, though linear and approximate, can represent the latest plant conditions as accurately as possible. The repeated linearization allows the controller to adapt as plant conditions change.

By linearizing and discreterizing the system model (6) at the latest operating point ($\mathbf{x}_\mathrm{d}(n)$, $\mathbf{u}_\mathrm{d}(n)$), we get a linear discrete state space model at the beginning of each simulation step:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \tag{16}$$
$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k), \tag{17}$$

where $\mathbf{A} \in \mathbb{R}^{6\times6}$, $\mathbf{B} \in \mathbb{R}^{6\times3}$, $\mathbf{C} \in \mathbb{R}^{2\times6}$ are the state, input and output matrices of the linearized state space model, respectively. A similar trajectory tracking problem can be formulated as (10) while linear system dynamics (16) and (17) are employed as the prediction model in linear MPC. Cost function J in the linear case can thus be easily translated into a Quadratic Programming (QP) problem by iterating and stacking system matrices into a compact QP matrix form, which gives:

$$\min_{\tilde{\mathbf{u}}}(k)\ J(\tilde{\mathbf{x}}(k),\ \tilde{\mathbf{u}}(k)) = \mathbf{u}(k)^\mathrm{T}\mathbf{H}\mathbf{u}(k) + \mathbf{h}^\mathrm{T}\mathbf{u}(k) \tag{18}$$

subject to

$$\mathbf{x}(k) = \mathbf{x}_{\mathrm{measure}}(k) \quad (19)$$
$$\mathbf{x}(k+n+1) = \mathbf{A}\mathbf{x}(k+n) + \mathbf{B}\mathbf{u}(k) \quad (20)$$
$$\hat{\mathbf{y}}(k+n) = g(\mathbf{x}(k+n)) \quad (21)$$
$$\mathbf{y}_{\min} \leq \hat{\mathbf{y}}(k+n) \ \leq \mathbf{y}_{\max}, \text{for } n = 1, 2, ..., N \quad (22)$$
$$\mathbf{u}_{\min} \leq \mathbf{u}(k+n) \ \leq \mathbf{u}_{\max}, \text{for } n = 0, 1, ..., N-1, \quad (23)$$

Similarly, the time varying reference trajectory is sampled, stacked and provided with a length of the prediction horizon to the cost function in the QP problem and shifted one step forward at each simulation loop.

Since the successively linearized model is an approximation of the original nonlinear system, performance of LMPC could be worse than the nonlinear counterpart, though working more efficiently. Especially when the state and input trajectories deviate from the current operating points, the model mismatch could increase, which could generate large open loop prediction errors resulting in an instability of the closed-loop system. In the next section, we investigate this further.

4. SIMULATION EXPERIMENTS

Numerical simulations based on the 3 DOF horizontal model of CyberShip II as introduced in Section 2 and the two different controller designs presented in Section 3 are carried out in this section to demonstrate the potential of vessel trajectory tracking using MPC.

4.1 Model parameters

The following specifications of Cybership II was obtained in (Skjetne et al., 2004) and are used in our simulations:

- System matrices:

$$\mathbf{M} = \begin{bmatrix} 25.8 & 0 & 0 \\ 0 & 33.8 & 1 \\ 0 & 1 & 2.8 \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} 0.72 & 0 & 0 \\ 0 & 0.89 & 0.03 \\ 0 & 0.03 & 1.9 \end{bmatrix}. \quad (24)$$

- Maximum nominal speed: 0.2 m/s (which corresponds to 1.7 m/s or 3.3 knots of the corresponding full scale vessel).
- Maximum actuator forces (f_{u} and f_{v}) and yaw moment (t_{r}): 2 N, 2 N and 1.5 Nm, respectively (which corresponds to 686 kN, 686 kN and 36015 kNm of the corresponding full scale vessel).

4.2 Controller parameters

For both of the controllers (NMPC and LMPC), in order to compare their trajectory tracking performances and computational complexities under different prediction horizons, different groups of parameters (N_p, T_s) have been shown in Table 1. For all the simulations, the following time varying reference trajectory $(x_{\mathrm{d}}(t),\ y_{\mathrm{d}}(t))$ is used:

$$\begin{aligned} x_{\mathrm{ref}}(t) &= 4\sin(0.02t) \\ y_{\mathrm{ref}}(t) &= 2.5(1-\cos(0.02t)), \end{aligned} \quad (25)$$

which is an ellipse with varying trajectory curvature. The following weighting matrices $(\mathbf{Q}, \mathbf{R})$ and simulation time are used: $\mathbf{Q}=\begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}, \mathbf{R}=\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, t_{\mathrm{end}} = 400$ s. Constraints on control inputs are given by:

Table 1. Parameters for different groups of simulations

Parameters	N (step)	Sampling time (s)
Scenario 1	3	2
Scenario 2	5	1
Scenario 3	10	0.5

$$\begin{bmatrix} -2 \\ -2 \\ -1.5 \end{bmatrix} \leq \mathbf{u}(k) \leq \begin{bmatrix} 2 \\ 2 \\ 1.5 \end{bmatrix}. \quad (26)$$

4.3 Simulation results and analysis

All the simulations are done in MATLAB 2011b and the nonlinear 3 DOF vessel model for successive linearization is constructed in Simulink. Simulations are first run under the settings of $N = 10$, $T_{\mathrm{s}} = 0.5$ with an initial position at $(x_0,\ y_0) = (0.5,\ 0.5)$.

The trajectory tracking errors and performances of NMPC and LMPC are shown in Figure 4 and Figure 5, respectively. Control inputs produced by two controllers are shown in Figure 6.

Figure 4 shows the absolute distance error between the measured and desired positions, which is calculated as $d_{\mathrm{e}}(k) = \sqrt{(x_{\mathrm{measure}}(k) - x_{\mathrm{r}}(k))^2 + (y_{\mathrm{measure}}(k) - y_{\mathrm{r}}(k))^2}$. It can be seen that NMPC yields smaller tracking errors while LMPC sees a few big fluctuations. But all errors are bounded within the range of 1 m. Trajectory tracking performances are shown in Figure 5. It can be seen that both NMPC and LMPC show the ability to track the reference trajectory. However, NMPC generates more smooth trajectories with higher tracking accuracy while LMPC goes through some deviations which are acceptable if tight tracking is not required. Minimal tracking errors, and thus better tracking performances, is the result of the first term in cost function of the optimization problem (10). The second term in (10) is included to minimize the control effort which is desirable from an economical perspective. The control inputs shown in Figure 6(a) and 6(b) are all bounded within the actuator saturation limits due to MPC's ability in handling constraints. NMPC shows relatively large changes in control inputs at the beginning and then remains small fluctuations while LMPC has input changes all the time. But both of them succeed to keep the input values around 0, and thus be able to minimize costs.

Although NMPC is better than LMPC in tracking accuracy, it is noteworthy that it takes much longer time for NMPC to run the simulation under same settings and a too heavy computational burden can bring about implementing issues. Therefore, different metrics are specified to compare these two approaches. Tracking performance metrics including the maximum, average and standard deviation of the absolute tracking error are considered; in addition, computational times were also saved as another metric. When evaluating the computational burden of successive linearization based MPC, we consider that the time for linearizing the nonlinear model to get a discrete linear state space model ((16) and (17)) and further translating it into a standard QP problem in each iteration loop is necessary to be included in addition to the time required

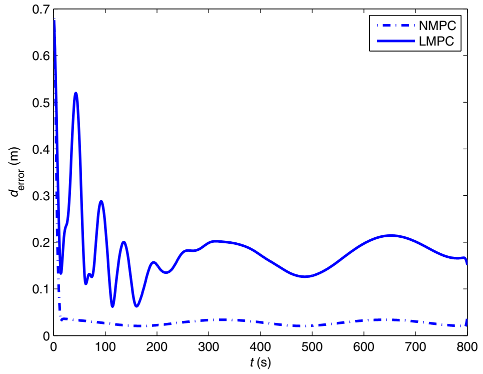


Fig. 4. Comparison of tracking error of NMPC and LMPC.

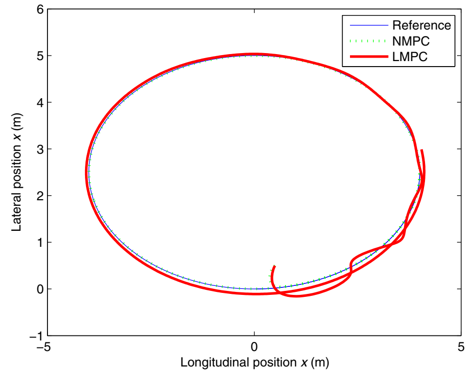
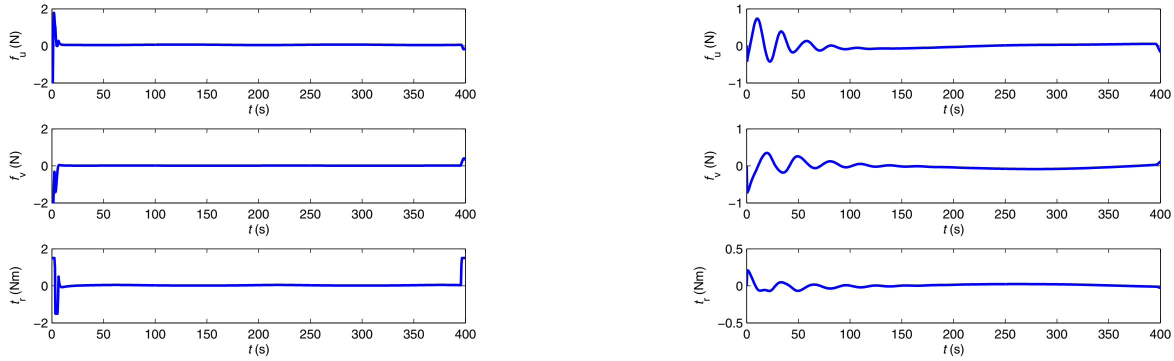


Fig. 5. Comparison of trajectory tracking performance of NMPC and LMPC.



(a) NMPC

(b) LMPC

Fig. 6. Control inputs of NMPC and LMPC.

Table 2. Performance comparison in terms of tracking errors and computational time

		Max. (m)	Mean (m)	SD (m)	Time (s)
	Scenario 1	0.39	0.00	0.03	828
NMPC	Scenario 2	0.60	0.00	0.04	4355
	Scenario 3	0.67	0.01	0.04	37584
	Scenario 1	10.81	2.30	3.45	29
LMPC	Scenario 2	8.70	1.20	1.86	53
	Scenario 3	0.83	0.22	0.14	112

Table 3. Effects of prediction horizon on tracking performance of LMPC

Ts (s)	N (step)	Max. (m)	Mean (m)	SD (m)	Time (s)
	10	0.70	0.01	0.08	709
0.1	20	0.70	0.10	0.09	826
	50	1.51	0.84	0.33	1137
	10	0.83	0.22	0.14	112
0.5	20	0.67	0.35	0.07	177
	50	1.06	0.67	0.15	202

to solve the QP optimization problem. Table 2 shows the comparisons in respect to these metrics under different settings. For both NMPC and LMPC, the computational time increases along with the increase of prediction step, which is intuitive because larger N means larger computation in prediction and more variables in the optimization problem. However, the tracking accuracy has shown different trends in NMPC and LMPC. For NMPC, the differences for the three groups are very small while the general trends shows larger N and smaller sampling time lead to larger errors for all the metrics, maximum, mean and standard deviation tracking error. LMPC, on the other hand, is quite sensible to the parameter settings. A larger N is observed to result in larger errors. This could be due to the fact that in the chosen approach, the model is linearized only around the current operation point at the beginning of each simulation loop. Between too long sampling times, the operation point might have great changes and could thus cause a serious model mismatch, causing a large error over the open loop prediction horizon. When $T_s = 2$s, a maximum error of almost 11m is observed. So, for LMPC, controller parameter tuning is considered to be important to get a satisfying control performance. Large sampling times and prediction steps which will result in large prediction time horizon are not suggested because of serious model mismatch. The effects of prediction horizon parameters (T_s, N) on the tracking performance of successive linearized based MPC is given in Table 3. Increasing prediction step from $N = 10$ to $N = 20$ can improve tracking performance slightly, but further increase in N will on the contrary deteriorate the control results because the prediction time $(T_s N)$ has lead to an unacceptable model mismatch.

5. CONCLUSIONS AND FUTURE RESEARCH

In this paper, we have proposed two different MPC approaches for addressing the horizontal nonlinear trajectory tracking problem of autonomous marine surface vehicles. The first approach directly adopts the nonlinear vessel dynamics as the prediction model and designs a nonlinear MPC controller which requires solving a constrained nonlinear optimization problem each step; the second approach uses a linearized model which is obtained by iteratively linearizing the nonlinear model online at the beginning of each simulation loop and translates the LMPC formulation into a QP problem. Simulations are run and results have illustrated the feasibility and effectiveness at the utilization of MPC in vessel trajectory tracking. Nonlinear MPC has the advantage of better tracking performance but it is much more computational complex especially under larger prediction horizons, which will lead to difficulties in implementation; LMPC, however, though it does show relatively poor tracking performance, seems to be real-time implementable if the tracking accuracy is not very strict and the initial position does not deviate from the tracking point too much. Note that in this paper we made a comparison using particular quadratic and nonlinear optimization problem solvers. We do expect, however, that the main conclusions drawn from the comparison will also hold with other solvers. We will make test of this hypothesis explicitly in future work. Research here thus starts an interesting but challenging topic, which deserves and requires further exploration.

Future work includes mainly three parts: from the control performance point of view, we will investigate improving the efficiency of NMPC and the accuracy of LMPC either by more efficient optimization techniques or model mismatch reductions. It is noteworthy that environmental disturbances (winds, waves and currents) are excluded in this paper; these disturbances will be considered in future research by introducing either an observer or filter into the system; desired trajectories have been designed to be an ideal ellipse, however, specific applications of autonomous marine surface vehicles such as inter terminal transport would involve multiple constraints in terms of mission requirements or safety reasons (e.g., COLREGs), so the guidance or optimal trajectory generation problem in specific scenarios also needs to be investigated further.

REFERENCES

K. D. Do and J. Pan. Robust path-following of underactuated ships: Theory and experiments on a model ship. *Ocean Engineering*, 33(10):1354–1372, 2006.

T. I. Fossen. Cybership II, April 2008. `http://www.itk.ntnu.no/ansatte/{Fossen_Thor}/GNC/cybership2.htm`.

T. I. Fossen. *Handbook of marine craft hydrodynamics and motion control*. John Wiley and Sons Ltd., West Sussex, U.K., 2011.

Z. Jiang. Global tracking control of underactuated ships by Lyapunov's direct method. *Automatica*, 38(2):301–309, 2002.

T. Keviczky and G. J. Balas. Flight test of a receding horizon controller for autonomous UAV guidance. In *Proceedings of the 2005 American Control Conference*, pages 3518–3523, Portland, Oregon, 2005.

U. Kiencke, L. Nielsen, R. Sutton, K. Schilling, M. Papageorgiou, and H. Asama. The impact of automatic control on recent developments in transportation and vehicle systems. *Annual Reviews in Control*, 30(1):81–89, 2006.

E. Lefeber, K. Y. Pettersen, and H. Nijmeijer. Tracking control of an underactuated ship. *IEEE Transactions on Control Systems Technology*, 11(1):52–61, 2003.

Z. Li and J. Sun. Disturbance compensating model predictive control with application to ship heading control. *IEEE Transactions on Control Systems Technology*, 20 (1):257–265, 2012.

Z. Li, J. Sun, and S. Oh. Path following for marine surface vessels with rudder and roll constraints: an mpc approach. In *Proceedings of American Control Conference*, pages 3611–3616, St. Louis, Missouri, 2009.

J. M. Maciejowski. *Predictive Control with Constraints*. Pearson education, 2002.

N. Minorski. Directional stability of automatically steered bodies. *Journal of American Society of Naval Engineers*, 42(2):280–309, 1922.

L. Moreira and C. G. Soares. Autonomous ship model to perform maneuver test. *Journal of Maritime Research*, VIII(2):29–46, 2011.

R. R. Negenborn. *Multi-Agent Model Predictive Control with Applications to Power Networks*. PhD thesis, Delft University of Technology, Delft, the Netherlands, 2007.

S. Oh and J. Sun. Path following of underactuated marine surface vessels using line-of-sight based model predictive control. *Ocean Engineering*, 37(2):289–295, 2010.

A. Pavlov, H. Nordahl, and M. Breivik. MPC-based optimal path following for underactuated vessels. In *Proceedings of the 8th IFAC International Conference on Manoeuvring and Control of Marine Craft*, pages 340–345, Guaruj, Brazil, 2009.

A.I. Propoi. Use of linear programming methods for the synthesizing of sampled-data automatic systems. *Automation and Remote Control*, 24(7):837–844, 1963.

S. J. Qin and T. A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733–764, 2003.

Nur Adila Faruk Senan. A brief introduction to using ode45 in matlab, 2012.

R. Skjetne, Ø. N. Smogeli, and T. I Fossen. A nonlinear ship manoeuvering model: Identification and adaptive control with experiments for a model ship. *Modeling, Identification and Control*, 25(1):3–27, 2004.

A. Wahl and E. Gilles. Track-keeping on waterways using model predictive control. In *Proceedings of the IFAC Conference on Control Applications in Marine Systems*, pages 149–154, Fukuoka, Japan, 1998.

X. Wang. *Study on Path Following Control for Underactuated Ships by Using Analytic Model Predictive Control*. PhD thesis, Shanghai Jiao Tong University, Shanghai, China, 2009.

H. Zheng, Z. Huang, C. Wu, and R. R. Negenborn. Model predictive control for intelligent vehicle lane change. In *Proceedings of the 2nd Internation Conference on Transportation Information and Safety*, pages 265–276, Wuhan, China, 2013.