Development of model predictive guidance for underactuated marine vehicles: From simulations to experiments

Nadir Kapetanović, Nikola Mišković, Đula Nađ

Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia.
nadir.kapetanovic@fer.hr, nikola.miskovic@fer.hr, dula.nad@fer.hr

Marco Bibuli, Massimo Caccia, Gabriele Bruzzone

Consiglio Nazionale delle Ricerche, Genoa, Italy.
marco.bibuli@ge.issia.cnr.it, massimo.caccia@ge.issia.cnr.it,
gabriele.bruzzone@ge.issia.cnr.it

Abstract—**A modular open source real-time simulation environment for linear model predictive control (LMPC) of line following at a constant depth is developed for underactuated marine vehicles. Two-stage tuning method of MPC's parameters is implemented as an extension of this simulator. MPC is used for high level control, i.e. setting the yaw rate reference, which is then tracked by the low level PID controllers. Simulation and experimental results show good performance of our MPC-PID control scheme when compared to ordinary PID controllers using Lyapunov-based guidance law.**

I. Introduction

In marine robotics, line following motion control is very important in missions such as sea floor sonar scanning. Lawnmower is one of the most commonly used solutions for 2D coverage problem, and is used as a reference path to be followed by marine vehicles. A comprehensive overview of the path following approaches (e.g. backstepping, Lyapunov control functions, adaptive control, disturbance rejection, and line-of-sight approach, etc.) is given in [1] and in the references therein.

Our ongoing research addresses the concept of an adaptive sea floor sonar scanning algorithm. The adaptation is reflected in steering the vehicle to scan some interesting areas in more detail, while skipping areas which are less interesting, thus maximizing some informational gain measure. This means that not all lawnmower legs need to be traversed along their whole length.

With this in mind, we plan to use model predictive control (MPC) as a control methodology. MPC has been used for path following problems in [2]. It has also been used for autonomous ground vehicles (AGVs) in problems formulated in a similar fashion as our above formulated research goal, namely in [3]. The first part of solving this problem is to make MPC framework for basic straight line following, which is presented here.

In this paper we present a real-time linear MPC (LMPC)-based motion planning environment for underactuated marine vehicles' (e.g. rudder-based vehicles) line following at a constant depth with assumed constant disturbance. LMPC is chosen in order to reduce the execution time of the used optimization algorithm, bearing in mind the real-time implementation for the experiments. Simulation environment was made by integrating ACADO toolbox, [4], with ROS environment. The developed environment has an MPC tuning module, which runs simulations with the given varied MPC parameter values, and based on the metrics presented in [5] and extended in this paper, assesses which parameter set performs the best.

Simulation results show the success of this tuning methodology, and also good performance of the resultant MPC-based motion planning framework. Results of the experiments both on the surface and underwater are given as a validation of the simulation results. They are are consistent with the simulation results, with some discrepancies caused by real-world factors.

This paper is organized as follows: kinematic model for line following problem in the presence of an unknown disturbance is given in Section II. In Section III, MPC framework for solving this problem is described. Simulation results are presented in Section IV. Results of

This research is sponsored by NATO[Pleaseinsert\PrerenderUnicode{âĂŹ}intopreamble]s Emerging Security Challenges Division in the framework of the Science for Peace and Security Programme as Multi Year Project under G. A. number 984807, named Unmanned system for maritime security and environmental monitoring - MORUS. Joint results were obtained as a result of staff exchange within H2020-TWINNING project "EXCELLABUST - Excelling LABUST in Marine Robotics" (Grant Agreement No. 691980).

on-sea surface and underwater experiments are given in Section V.

II. Kinematic model of line following with constant disturbance

Let us assume that we need to use some underactuated marine vehicle for a sea floor sonar scanning mission. Also, let us assume that for accomplishing that mission, the vehicle should move only in a constant depth plane. In this context, we define underactuatedness of the vehicle as its inability to directly control pitch, roll, and sway. Heave motion is neglected since we are assuming that the vehicle is already at the desired depth. Moving in the marine environments necessarily means dealing with different kinds of disturbances, i.e. wind and waves for missions on the surface, sea current for underwater missions. These disturbances can also represent unmodeled system dynamics. Let us furthermore assume that these disturbances are constant, but not observable, and denote them as ν. Since we are assuming a sonar scanning mission, it is preferable for the vehicle to maintain a constant surge speed w.r.t. water $U_r > 0, U_r > \nu, U_r = const.$, while the change in heading is controlled by yaw rate r.

This leaves us with a task to control yaw rate in such a way that the vehicle follows the straight lines of the desired lawnmower pattern, i.e. minimizing the distance to the current line (cross-track error) d. Also, heading error $\beta = \psi - \psi_L$, i.e. the difference between the vehicle's heading ψ, and current line's heading ψ_L, should be minimized.

With all the above assumptions taken into account, kinematic model of line following in a horizontal plane is given as

$$\dot{d} = U_r \sin\beta + \nu \simeq U_r \beta + \nu \tag{1}$$

$$\dot{\beta} = r \tag{2}$$

$$\dot{d}_{int} = d \tag{3}$$

where the symbol $\simeq$ denotes linear approximation for small values of β [6].

The linearized kinematic model of line following given by (1) and (2), is thus extended with an additional state given by (3), which represents the integral ofthe cross-track error. This model extension is a common practice in robust MPC schemes with disturbance rejection, as in [7], which enables stabilization of the system around the setpoint even in the presence of an external disturbance.

Furthermore, vehicle's position and orientation $[x\, y\, \psi]$ in the earth-fixed frame $\langle e \rangle$ are expressed as

$$\dot{x} = U_r \cos\psi + \nu_x \tag{4}$$

$$\dot{y} = U_r \sin\psi + \nu_y \tag{5}$$

$$\dot{\psi} = r \tag{6}$$

where ν_x, ν_y are x and y components of the current speed, respectively [6].

III. Linear model predictive control framework for line following

In order to minimize the computation complexity, and be able to implement our control framework on a real marine vehicle, in our approach we used linear MPC as a high level yaw rate controller. Low level PID controllers are delegated to control constant surge speed throughout the whole mission, but also to track the reference yaw rate set by the high level MPC controller.

In order to solve control optimization in a MPC fashion, we used ACADO toolbox [4], as the control optimization tool. Optimization was even more sped up by the use of RealTimeAlgorithm class, which enabled us to make MPC controller ready to be used in real-time. MPC framework which has been used is given here. Cost function J is expressed as

$$J = \int_{t_i}^{t_i+T_p} \left(K_d d^2(\tau) + K_\beta \beta^2(\tau) + K_{d_{int}} {d_{int}}^2(\tau) \right) d\tau \tag{7}$$

subject to

$$-\pi \leq \beta(\tau) \leq \pi, \qquad \forall\tau \in [t_i, t_i + T_p] \tag{8}$$

$$-20^\circ/s \leq r(\tau) \leq 20^\circ/s, \qquad \forall\tau \in [t_i, t_i + T_p] \tag{9}$$

where $t_i = kT_s, k \in \mathbb{N}_0$ is initial time of the prediction horizon which lasts for $T_p[s]$, with sampling time $T_s = 125ms$. Another MPC design parameter in ACADO was N_{steps} parameter. This parameter implicitly sets the duration of the control horizon, since $T_c = N_{steps} T_s[s]$ holds. In the rest of this paper, terms N_{steps} and T_c will be used interchangeably.

IV. Simulation results

Tuning of MPC controller is often done ad hoc, choosing its parameters which give just good enough results. Our idea was to vary MPC controller's parameters (T_p, N_{steps}, K_d, K_β, and $K_{d_{int}}$) for the particular model and environment setting in some relatively wide, but sensible range, simulate the system in the closed-loop, and asses the quality of the system performance numerically. Problem space for tuning all the parameters at the same time was too big, so we divided it into two stages of parameter variation. In the first simulation stage, T_p and N_{steps} were varied, and in the second stage K_d, K_β, and $K_{d_{int}}$ were varied.

It is important to note here that the set value of the surge speed for the vehicle in the simulations was $U_r = 0.5m/s$, and that the disturbance was simulated as $\nu = \nu_x = 0.2m/s$, $\nu_y = 0$. Steady-state ϵ-zone is defined as $\epsilon = 0.1m$.

A. Indices for line following performance evaluation

Since in the parameter variation simulations many responses get generated, it is of our interest to grade these responses in some objective way. This was done using the methodology proposed in [5]. It enabled us to analyze and quantify performance of different parts of the vehicle's path towards convergence to the desired line, i.e. turn phase, approach phase, and settling and steady-state phase. Graphical explanation of all the parameters in all three phases is given in Fig. 1.

As defined in [5], line following performance during the turn phase is parametrized by $H_{||}[m]$, $A_1[m^2]$, and $H_\perp[m]$, see Fig. 1.

The second phase is the so-called path approach phase, and it is parametrized by $A_2[m^2]$, $\overline{\chi}[m/s]$, and $\overline{\chi}_{max}[m/s]$, see Fig. 1. Here $\overline{\chi}[m/s]$ denotes the mean of cross-track error rate normalized by the duration of the approach phase, while $\overline{\chi}_{max}[m/s]$ is the maximum value of cross-track error rate during the approach phase.

The third and the last phase is the settling and steady-state phase, in which cross-track error is within some ϵ-zone around the followed path. This phase is parametrized by four parameters, namely $H_2[m]$, $A_3^*[m]$, $t_s[s]$, $\overline{\Delta r}[\deg/s]$, see Fig. 1. Settling time t_s has been added as an additional analysis parameter in this paper. Also, instead of computing the rudder stress R as in [5], in this paper we computed yaw rate stress index $\overline{\Delta r}$, and thus implicitly the energy consumption.

In addition to using the quantitative analysis of line following from [5], in this paper all of the indices quantifying the response quality were scaled relative to the minimum/maximum value of that index throughout all simulations. The scaling was done so that we could easily compare system response qualities from our simulations, and make some generalized weighted cumulative performance function denoted as Σ. The value of this performance measure can tell us for which MPC parameter n-tuple we get the best control performance.

Scaling by the minimum value of the performance index was done due to the fact that smaller values of $H_{||}, A_1, H_\perp, A_2, H_2, A_3^*, t_s, \overline{\Delta r}$ mean better system performance. Conversely, scaling by the maximum value was done due to the fact that greater values of performance indices $\overline{\chi}$ and $\overline{\chi}_{max}$, and also of the overall performance measure Σ, mean better system performance. If we generalize parameter p as $p \in \{H_{||}, A_1, H_\perp, A_2, \overline{\chi}, \overline{\chi}_{max}, H_2, A_3^*, t_s, \overline{\Delta r}, \Sigma\}$, then the scaling by the minimun value is given by

$$p_{ij,s} = \frac{min\{p_{1j}, p_{2j}, \cdots, p_{N_{sims}j}\}}{p_{ij}} 100[\%] \tag{10}$$

and scaling by the maximum value of the performance index is given by

$$p_{ij,s} = \frac{p_{ij}}{max\{p_{1j}, p_{2j}, \cdots, p_{N_{sims}j}\}} 100[\%] \tag{11}$$

where $i = \overline{1, N_{sims}}$ denotes the index of the simulation, $j = \overline{1, N_{params}}$ denotes the index of the performance index, N_{sims} denotes the overall number of conducted simulations, i.e. the number of specific parameters' combinations, and N_{params} denotes the number of performance indices evaluated. The overall performance score Σ_i of one set of varied MPC parameters is calculated as a weighted sum of all scaled performance indexes for i-th simulation, given by

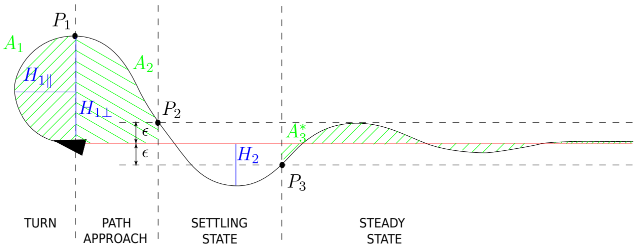


Fig. 1: Performance indices of path following [5].

$$\Sigma_i = \sum_{j=1}^{N_{params}} w_j p_{ij,scaled} \tag{12}$$

where w_j denotes the weight, i.e. the importance of j-th parameter. In this paper we set all weights as $w_j = 1, \forall j = \overline{1, N_{params}}$. After this, all the performance scores are scaled by the maximum principle given by (11).

B. Choosing prediction and control horizon

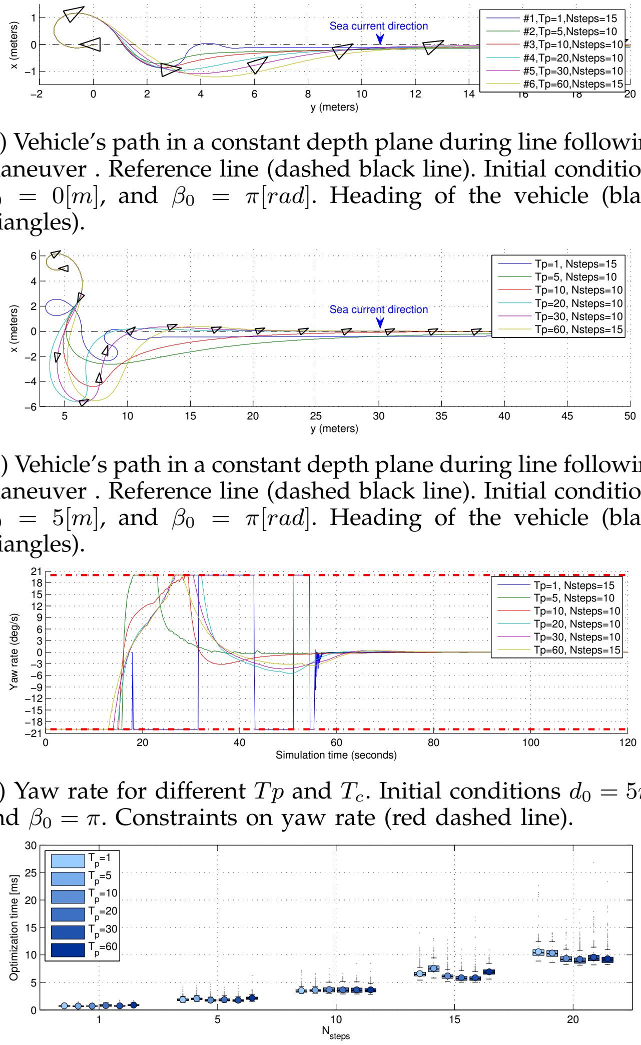
The first stage of parameter variation procedure was to vary parameters T_p and N_{steps} (thus implicitly varying the control horizon T_c). Prediction horizon duration T_p has a very important role in the overall stability of the controlled system in closed-loop. Apart from varying T_p and N_{steps}, we also varied the initial distance of the vehicle w.r.t. the line to follow, in order to choose T_p such that the system stabilizes in both cases.

On the other hand, control horizon duration T_c can cause the system to destabilize if T_c is chosen too short. If T_c is chosen too long, it causes faster response in most cases, but also an additional control optimization burden. Large value of T_c can also cause the controls to change too rapidly, which can lead to shortened life of actuators in the long term.

The first set of simulations was conducted with varying $T_p \in \{1, 5, 10, 20, 30, 60\}[s]$ and $N_{steps} \in \{1, 5, 10, 15, 20\}$. The initial conditions of controller internal system model were $d_0 = 0, \beta_0 = \pi, d_{int_0} = 0$. For these experiments cost function parameters were set as $K_d = 1$, $K_\beta = 0.001$, and $K_{d_{int}} = 0.01$. For each T_p the best response with corresponding T_c has been shown in Fig. 2a. The second set of initial conditions was $d_0 = 5, \beta_0 = \pi, d_{int_0} = 0$. The corresponding best system responses for each T_p are shown in Fig. 2b.

It can be noted that longer T_p cause the response to have a larger overshoot (see also H_2 column of Table I). Also, shorter T_p caused the system to have a longer settling time, which is apparent in t_s column of Table I.

It is interesting to note that the vehicle's heading in the steady-state phase is not aligned with the followed



(a) Vehicle's path in a constant depth plane during line following maneuver . Reference line (dashed black line). Initial conditions $d_0 = 0[m]$, and $\beta_0 = \pi[rad]$. Heading of the vehicle (black triangles).

(b) Vehicle's path in a constant depth plane during line following maneuver . Reference line (dashed black line). Initial conditions $d_0 = 5[m]$, and $\beta_0 = \pi[rad]$. Heading of the vehicle (black triangles).

(c) Yaw rate for different Tp and T_c. Initial conditions $d_0 = 5m$, and $\beta_0 = \pi$. Constraints on yaw rate (red dashed line).

(d) Statistical analysis of control optimization execution times per one sampling period.

Fig. 2: Simulation results for for various Tp and N_{steps} values.

TABLE I: Prediction and control horizon variation. Quantitative analysis of response quality of the approach and steady-state phase, control stress, and final score. Initial conditions: $d_0 = 0$, $\beta_0 = \pi$, $d_{int_0} = 0$. For # referencing to T_p and N_{steps} values see Fig. 2a legend.

#	A_2	$\overline{\chi}$	$\overline{\chi}_{max}$	H_2	A_3^*	t_s	$\overline{\Delta r}$	$\Sigma_{i,s}$
1	2.984	0.039	0.069	0.88	0.105	56.5	4.54	77.8
2	2.986	0.038	0.065	0.72	0.048	50.8	1.31	86.9
3	2.988	0.038	0.064	0.86	0.026	39.8	1.19	92.9
4	3.016	0.037	0.059	0.96	0.018	37.0	1.12	97.3
5	3.105	0.036	0.053	1.09	0.016	34.0	0.86	100
6	3.283	0.033	0.048	1.21	0.019	38.6	0.72	94.5

line's heading, see Fig. 2a and 2b. This is due to the fact that the vehicle orients itself at an angle to compensate against the current at the assumed constant surge speed.

For the sake of brevity, we did not include the turn phase indices values here. We found that all three turn phase performance indices have the same values $H_{||} = 1.46m$, $A_1 = 8.49m^2$, and $H_{\perp} = 1.17m$, invariant to the chosen stabilizing MPC parameters. The vehicle tries to turn as fast as possible in this phase, so these indices depend only on the surge speed and yaw rate bounds.

Table I shows the unscaled performance indices of the approach, and steady-state phase, but also the overall scaled score of each $T_p - N_{steps}$ combination. It can be noted that the score does not grow with the increase of T_p, but is some nonlinear function of T_p. This leads to the conclusion that longer prediction horizon does not always guarantee better MPC controller performance.

Also, it can be noted that as the fraction T_c/T_p increases, the vehicle approaches the line faster. This means that, as T_p gets shorter, the system cannot predict its future trajectories far in time, so as T_c takes up a bigger part of T_p, MPC generates more aggressive controls (see $\overline{\Delta r}$ column of Table I and Fig. 2c) which in turn cause the vehicle to approach the line faster. Based on Table I, we chose $T_p = 30s$, and $N_{steps} = 10$.

Another thing which we were trying to asses empirically, was the estimation of the optimization duration in each sampling period. It is important to note here that the machine which was used for simulations had an Intel® Core^{TM} i5-3210M processor with clock frequency of $2.5 - 3.1GHz$, and $6GB$ of RAM. In Fig. 2d it can be seen that the distributions of execution times of simulations for each pair (N_{steps}, T_p) have been repeated 10 times to gain some statistical credibility. It is evident that the complexity, i.e. execution time is dependent only on the number of control steps being optimized at each MPC iteration For $N_{steps} = 10$, execution times for any tested T_p are under $5ms$, which is still far below the value of sampling time $T_s = 125ms$. Even with the transition to real system's lower performance on-board computer, this way we could make sure that the execution time of OCP solver will be real-time.

C. Choosing cost function parameters

After choosing prediction horizon T_p and control horizon T_c, the next step was to choose the parameters of cost function J given by (7). Again, this was done through parameter variation analysis for $K_d \in \{0.01, 0.1, 1, 2, 5\}$, $K_\beta \in \{0.001, 0.01, 0.1, 2, 5\}$, and $K_{d_{int}} \in \{0.001, 0.01, 0.1, 1, 2\}$. The best 6 resulting responses are chosen, and shown in Fig. 3.

Table II shows the unscaled performance indices of approach and steady-state phase, but also the overall score of each $T_p - N_{steps}$ combination. Based on Table II, we chose $K_d = 1.0$, $K_\beta = 5.0$, and $K_{d_{int}} = 0.001$. Indeed, the response with these parameters converges to the steady-state much faster than most of the other considered responses, see Fig. 3. Also, its overshoot over

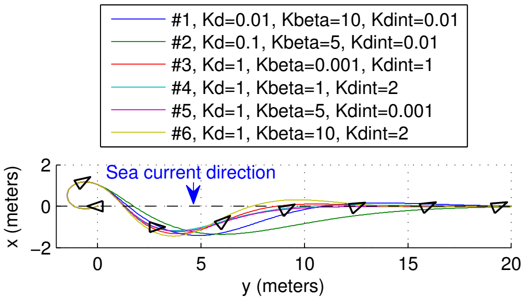


Fig. 3: Cost function paratemers variation for $T_p = 30s$ and $N_{steps} = 10$. Initial conditions $d_0 = 0[m]$, and $\beta_0 = \pi[rad]$. Reference line (dashed black line). Heading of the vehicle (black triangles).

TABLE II: Cost function parameters variation. Quantitative analysis of response quality of the approach and steady-state phase, control stress, and final score. Initial conditions: $d_0 = 0$, $\beta_0 = \pi$, $d_{int_0} = 0$. For # referencing to K_d, K_β, and $K_{d_{int}}$ value see Fig. 3 legend.

#	A_2	$\overline{\chi}$	$\overline{\chi}_{max}$	H_2	A_3^*	t_s	$\overline{\Delta r}$	$\Sigma_{i,s}$
1	3.21	0.0334	0.048	1.42	0.01	46.0	0.7	94.5
2	3.47	0.0302	0.041	1.36	0.0094	47.3	0.6	94.9
3	3.07	0.0362	0.056	1.33	0.009	34.8	0.9	99.8
4	3.08	0.0358	0.054	1.19	0.0133	30.1	0.9	95.5
5	3.09	0.0355	0.054	1.22	0.0097	31.5	0.8	100
6	3.04	0.0365	0.058	1.45	0.0089	38.6	1.1	98.1

the line is second smallest among all other responses taken into account.

V. Experimental results

A. Experiment setup

Validation of the simulation results was conducted in a series of on-sea experiments in October 2016 in Biograd na Moru. We used Hybrid AUV/ROV robotic platform e-URoPe (e-Underwater Robotic Pet), developed at CNR-ISSIA (Genoa, Italy) in ROV mode for the experiments. E-URoPe was used in an underactuated mode. Heave and surge motions were controlled by its existing controllers, while its low level controllers were used for tracking yaw rate reference which MPC controller sets.

Since the experiments were conducted in shallow water, both on surface and underwater, sea current effects were negligible, so we had to exclude d_{int} state from the MPC controller's system model. This has also greatly reduced the oscillations of the vehicle around the desired lines it followed. Due to the mentioned simplification of the model, parameter K_β was set to $K_\beta = 0.1$ in order to get better line following. Constant surge speed was set to $U_r = 0.2m/s$, and yaw rate was limited to $|r| \leq 12\deg/s$ in order to improve the maneuverability of the ROV. Low level yaw rate controller for MPC's yaw rate reference tracking was a P controller with $K_p = 0.4$.

During both surface and underwater experiments our approach was compared to a guidance scheme presented in [8]. This scheme is based on a Lyapunov-based virtual-target path-following algorithm for setting the heading reference, combined with a PID heading controller which controls yaw torque.

B. Surface experiments

The first set of experiments was conducted on sea surface, during which e-URoPe ROV used GPS for localization. Fig. 4a shows the path of the e-URoPe while following the desired lawnmower pattern, controlled by MPC-PID approach (red), and the above mentioned LCF-PID approach (blue). It is interesting to note that in both cases the vehicle oscillates around the desired lines in the steady-state phase. These oscillations have an amplitude smaller than $0.5m$, which is a value of the same order of magnitude as the GPS precision class.

Reference yaw rate generated by the MPC controller (blue) and the estimated yaw tracking by the low level PID controller (red) are given in Fig. 4b. It is obvious that all the reference yaw rate values are within the set bounds, and also that the tracking is very good, with only a small latency that is tolerable.

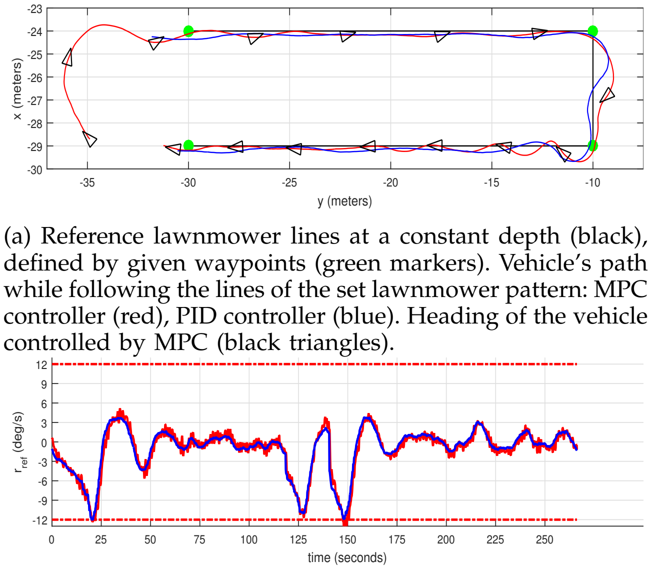
C. Underwater experiments

Underwater experiments were conducted with the same lawnmower pattern at a depth of $1.4m$, where e-URoPe ROV used USBL for localization. Lawnmower line following performance of the MPC-PID approach (red), and LCF-PID approach (blue) is shown in Fig. 5a. Compared to the surface experiments, the performance of line following underwater is much better. There are no oscillations around the lines in the steady-state phase, only a few position estimation outliers.Yaw rate tracking (red) of the yaw rate reference computed by MPC controller (blue) is given in Fig. 5b. The performance of the P-type tracking controller is very fast and precise.

Analyzing the experimental results from Fig. 4a and 5a, and comparing MPC-PID approach with LCF-PID approach from [8], one could say that the difference is negligible. However, if Fig. 4b and 5b are analyzed, it can be noted that both the reference and estimated yaw rate values are within the set bounds. On the other hand, estimated yaw rate values of LCF-PID approach for surface and underwater experiments, shown in Fig. 6, violated the set bounds even by 50% which if significant. Since LCF-PID approach from [8] controls yaw torque, yaw rate could not be directly controlled and thus saturated.

VI. Conclusion and future work

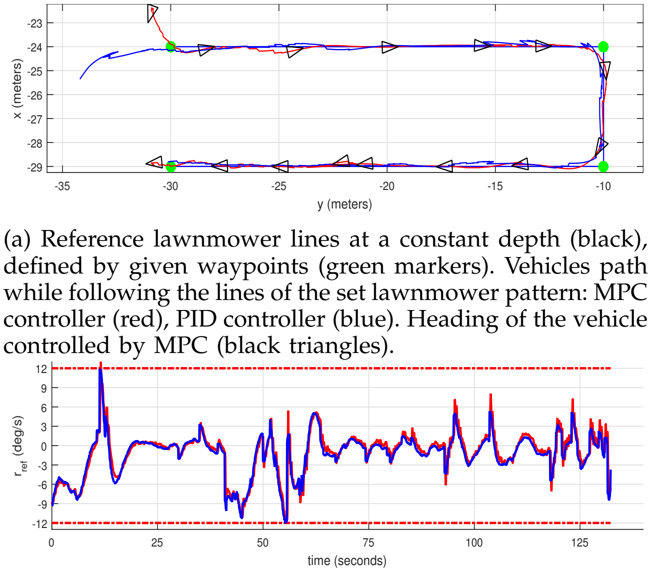
LMPC has shown good performance when applied to line following problem for underactuated marine vehicles in the presence of disturbance. It generates optimized and feasible motions for the vehicle to perform, while taking into account all the constraints on the states and controls. These motions can be kinematically, but also dynamically feasible, depending on the model of th



(a) Reference lawnmower lines at a constant depth (black), defined by given waypoints (green markers). Vehicle's path while following the lines of the set lawnmower pattern: MPC controller (red), PID controller (blue). Heading of the vehicle controlled by MPC (black triangles).

(b) MPC controller. Tracking (red) of the reference yaw rate (blue). Bounds of the yaw rate values (red dash-dot).

Fig. 4: Surface experiments.



(a) Reference lawnmower lines at a constant depth (black), defined by given waypoints (green markers). Vehicles path while following the lines of the set lawnmower pattern: MPC controller (red), PID controller (blue). Heading of the vehicle controlled by MPC (black triangles).

(b) MPC controller. Tracking (red) of the reference yaw rate (blue). Bounds of the yaw rate values (red dash-dot).

Fig. 5: Underwater experiments.

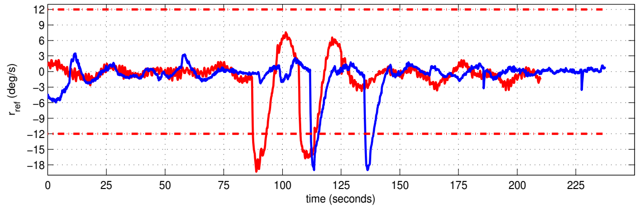


Fig. 6: Estimated yaw rate signals caused by the use of PID controller. Surface experiment (red), underwater experiment (blue). Bounds of the yaw rate values (red dash-dot).

vehicle used. LMPC was used as a high level controller, optimizing the yaw rate reference, while low level PID controllers were used to track the yaw rate reference, and also maintain a constant surge speed and depth.

We made a modular real-time open source simulator by joining ROS framework with ACADO toolbox for MPC. Tuning of MPC controller's parameters is done in an automated and two-stage fashion. Human operator needs only to set the parameter variation values, while all the simulations, as well as the performance assessment of each parameter n-tuple is using a preprogrammed methodology. Simulation and experimental results show good performance of MPC when compared to PID controllers, especially in the case of constraints satisfaction.

References

[1] Marco Bibuli, Walter Caharija, Kristin Y Pettersen, Gabriele Bruzzone, Massimo Caccia, and Enrica Zereik. ILOS guidance-experiments and tuning. *IFAC Proceedings Volumes*, 47(3):4209–4214, 2014.

[2] So-Ryeok Oh and Jing Sun. Path following of underactuated marine surface vessels using line-of-sight based model predictive control. *Ocean Engineering*, 37(2–3):289 – 295, 2010.

[3] A. Tahirovic and G. Magnani. General framework for mobile robot navigation using passivity-based MPC. *IEEE Transactions on Automatic Control*, 56(1):184–190, Jan 2011.

[4] B. Houska, H.J. Ferreau, and M. Diehl. ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization. *Optimal Control Applications and Methods*, 32(3):298–312, 2011.

[5] Eleonora Saggini, Enrica Zereik, Marco Bibuli, Gabriele Bruzzone, Massimo Caccia, and Eva Riccomagno. Performance indices for evaluation and comparison of unmanned marine vehicles' guidance systems. *IFAC Proceedings Volumes*, 47(3):12182–12187, 2014.

[6] Massimo Caccia, Marco Bibuli, Riccardo Bono, and Gabriele Bruzzone. Basic navigation, guidance and control of an unmanned surface vehicle. *Autonomous Robots*, 25(4):349–365, 2008.

[7] Kenneth R Muske and Thomas A Badgwell. Disturbance modeling for offset-free linear model predictive control. *Journal of Process Control*, 12(5):617–632, 2002.

[8] G. Bruzzone, M. Bibuli, E. Zereik, A. Ranieri, and M. Caccia. Cooperative adaptive guidance and control paradigm for marine robots in an emergency ship towing scenario. *International Journal of Adaptive Control and Signal Processing*, 2016.