

Real-time model predictive line following control for underactuated marine vehicles

Nadir Kapetanović * Marco Bibuli ** Nikola Mišković *
Massimo Caccia **

* Faculty of Electrical Engineering and Computing, University of Zagreb,
Zagreb, Croatia (e-mail: nadir.kapetanovic@fer.hr, nikola.miskovic@fer.hr).

** Consiglio Nazionale delle Ricerche, Genoa, Italy, (e-mail:
marco.bibuli@ge.issia.cnr.it, massimo.caccia@ge.issia.cnr.it)

Abstract:

A linear model predictive control (LMPC) based framework is developed for underactuated marine vehicles' kinematic line following while moving at a constant depth in the presence of disturbance. LMPC is used as a high level controller which sets the reference surge and yaw velocities for the low level PID tracking controllers. Used ACADO optimization toolbox solves the optimal problem at hand in real-time. This framework has shown good results both in simulation environment and in on-sea experiments.

© 2017, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

Keywords:

unmanned marine vehicles, marine systems navigation, guidance and control, model predictive control, disturbance rejection

1. INTRODUCTION

Lawnmower pattern is one of the most commonly used solutions for 2D coverage problem. In marine robotics, it is used as a reference path for vehicles (ROVs, AUVs, ASVs, etc.) with applications in sea bottom scanning missions, acoustic scanning, in the fields of archeology, biology, security, ecology, mining, etc. For a marine vehicle to be able to execute these missions, it has to have a motion control module for path following in its control architecture. Path following (PF) problem is defined as a problem of a vehicle converging and staying on a geometrically defined curve. Unlike the trajectory following problem, the curve to be followed in the path following problem is not parametrized by time, Fossen (2011). A comprehensive overview of the path planning approaches (namely backstepping, Lyapunov control functions, adaptive control, disturbance rejection, and line-of-sight approach) is given in Bibuli et al. (2014) and in the references therein.

Our long-term research goal is to develop an adaptive sea floor sonar scanning algorithm, in such a way that the marine vehicle does not necessarily need to traverse the whole length of all the lawnmower's legs. During the mission, this algorithm would steer the vehicle to scan some interesting areas in more detail, while skipping areas which are less interesting. The definition of which areas are interesting and which are not, i.e. some measure

of information gain, depends on the mission itself, but also on the science field for which the marine vehicle is being utilized.

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 Oh and Sun (2010), and Yi et al. (2016). 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 Tahirovic and Magnani (2011), and Tahirovic et al. (2014). MPC has the advantage that it can include a function of the above mentioned informational gain into its cost function, and give such controls which maximize the informational gain. Another advantage of MPC is that, during control optimization, it explicitly takes into account the constraints on system states and controls. The first part of solving this problem is to make MPC framework for basic straight line following, which is presented in this paper. Straight line following problem is a special case of the path following problem, where a parametrized curve is actually a straight line. Mathematical model of line following which is used in this paper is derived from Breivik and Fossen (2004), and Caccia et al. (2008).

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. LMPC is chosen in order to reduce the complexity, i.e. the execution time, of the optimization algorithm which computes the MPC solution. The execution time reduction is done bearing in mind the real-time implementation for the experiments. Simulation environment has been made by integrating ACADO toolbox, Houska et al. (2011), with ROS environment which is omnipresent in today's robotics research field. Linearized

* This research is sponsored by NATO'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).

kinematic model of line following for an underactuated marine vehicle moving at a constant depth with assumed constant disturbance, has been used. Therefore, disturbance rejection had to be added to the standard MPC framework. High level motion control, i.e. reference yaw rate optimization, has been MPC's task, while low level control for reference yaw rate tracking, constant surge speed, and constant depth maintaining, has been delegated to PID controllers. Simulation results presented in this paper show good performance of the MPC-based motion planning framework for line following in a horizontal plane. Results of the sea experiments both on the surface and underwater are given as a validation of the simulation results, and they are consistent with the simulations, with some discrepancies caused by real-world factors.

Although lacking an analytical proof of system's stability, simulation and also experimental results show that, with a common sense reasoning behind the choice of control optimization algorithms' parameters, it is possible to achieve good performance regarding the line following.

This paper is organized as follows: a short description of kinematic model for line following problem is given in Section 2. In Section 3, MPC framework for solving this problem is described, alongside the parameters important for it to work properly. Simulation results are analyzed in Section 4, and experimental results validating the simulation results are presented in Section 5.

2. KINEMATIC MODEL OF LINE FOLLOWING

Assuming that the marine vehicle is moving in the horizontal plane (at a constant depth), pitch, roll, heave, and sway motions can be neglected. With sonar scanning in mind, it is preferable for the marine vehicle to maintain a constant surge speed with respect to water $U_r > 0, U_r = const.$, in order to get sonar measurements in equidistant points along the followed lawnmower lines. The task of the control algorithm is to steer the vehicle to follow the current straight line of the lawnmower pattern, thus reducing the distance to the line, denoted as d , to zero. Heading error is defined as $\beta = \psi - \psi_L$, and it represents the difference between the vehicle's heading $\psi \in [-\pi, \pi)$ and the reference line orientation ψ_L .

Kinematic model of line following in a horizontal plane is given as

$$\dot{d} = U_r \sin \beta + \nu \simeq U_r \beta + \nu \quad (1)$$

$$\dot{\beta} = r \quad (2)$$

where the symbol \simeq denotes linear approximation for small values of β . The effects of disturbance are denoted by ν . The disturbance is of course assumed to be unknown, and it includes the effects of the sea current, waves, wind (for surface missions), but also unmodeled system dynamics, Caccia et al. (2008).

With all the above mentioned model simplifications due to vehicle's movement in the horizontal plane, vehicle's position and orientation $[x y \psi]$ in the earth-fixed frame $\langle e \rangle$ are expressed as

$$\dot{x} = U_r \cos \psi + \nu_x \quad (3)$$

$$\dot{y} = U_r \sin \psi + \nu_y \quad (4)$$

$$\dot{\psi} = r \quad (5)$$

where ν_x, ν_y are x and y components of the current speed, respectively, and r is yaw rate, Caccia et al. (2008).

3. LINEAR MPC FRAMEWORK FOR LINE FOLLOWING

Control framework which is proposed in this paper consists of two levels. High level control is implemented as a linear MPC controller, which sets the reference yaw rate. Low level control is implemented as PID control for surge speed reference tracking (which is set to a constant value during the whole mission), as well as yaw rate reference tracking. MPC controller uses the linearized kinematic model of line following given by (1) and (2), extended with an additional state given by

$$\dot{d}_{int} = d \quad (6)$$

which represents the integral of distance to the line. This model extension is a common practice in robust MPC schemes with disturbance rejection, as in Muske and Badgwell (2002), and Pannocchia (2003), which enables stabilization of the system around the setpoint even in the presence of an external disturbance. Otherwise, the system would become unstable, meaning that the marine vehicle would drift away from the desired line.

This integral state should not be confused with the direct integration block which is added to the system in the PID control schemes for line following in order to reject the disturbance, as in Caccia et al. (2008). State d_{int} is merely implicitly, through its presence in the cost function, causing MPC controller to take into account the effects of the disturbance, thus optimizing the control which rejects it.

MPC controller is internally using the linearized kinematic model of line following for the purpose of control optimization, while the outputs of the real system, i.e. position and orientation vector $[x y \psi]$ must be mapped into state vector $[d \beta d_{int}]$ in the feedback.

We have used ACADO toolbox, Houska et al. (2011), as the control optimization tool. More precisely, Real-TimeAlgorithm class has been used for control optimization in order that this controller could be used in real-time control applications. MPC framework which has been used is given here. Cost function J is expressed as

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

subject to

$$-\pi \leq \beta(\tau) \leq \pi, \quad \forall \tau \in [t_i, t_i + T_p] \quad (8)$$

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

where $t_i = kT_s, k \in \mathbb{N}_0$ is initial time of the prediction horizon which lasts for $T_p = 30s$, with sampling time $T_s = 125ms$. Weight coefficients in the cost function which have been used are: $K_d = 1, K_\beta = 0.001, K_{d_{int}} = 0.01$.

Another parameter of MPC, which has an impact on system's performance, is the duration of control horizon $T_c \leq T_p$. This parameter can be changed implicitly in

ACADO, by setting the number of control signal step changes per one control horizon as $N_{steps} = T_c/T_s$.

Too few N_{steps} can mean that the system will be unable to stabilize. Too many N_{steps} can make the optimization problem too complex for the algorithm to solve in real-time. This can also make the control signal jittery, which can have a bad effect when transferred to the actuators and their electronics. Trade-off between the above mentioned criteria for choosing the value of N_{steps} resulted in setting $N_{steps} = 10$.

Simulations with state vector initial conditions and algorithm parameters variation have been conducted in order to get the above given values for parameters T_p , N_{steps} , K_d , K_β , and $K_{d_{int}}$ such that the performance of the system is satisfying, i.e. the system stabilizes around the given line, its settling time is as short as possible, the first overshoot is as small as possible, the optimized yaw rate reference signal is not jittery, and the computation time per one MPC iteration is still real-time.

The terminal cost in the cost function J has not been added simply because the chosen prediction horizon duration T_p has been set to a value large enough for the system to stabilize. Adding terminal cost, if designed properly, can guarantee stability of the system, but it also adds up to the control optimization algorithm. This additional optimization time could endanger real-time execution requirement, so this was another reason why it has not been used in our approach.

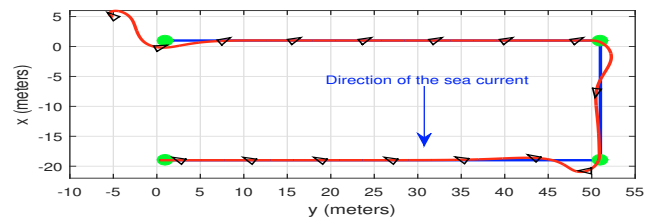
4. SIMULATION RESULTS

During the research, we have developed a stand-alone simulator which integrates ACADO toolbox with ROS environment. The goal was to make a modular code structure, which would allow us to easily expand the functionality of our simulator, as well as interface it with other simulators and software on-board marine vehicles.

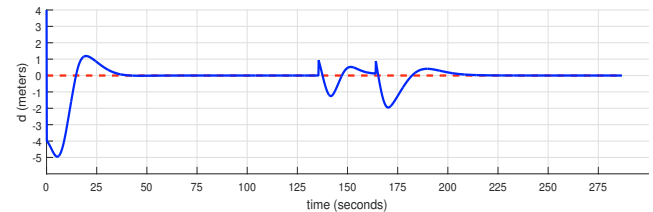
Based on the linearized line following model with modeled disturbance given in Section 2, MPC framework described in Section 3 and implemented in the above mentioned simulator, we obtained the following simulation results. Fig. 1a shows the plot of vehicle's path in the plane vs. the reference lawnmower lines. Surge speed has been set to $U_r = 0.5m/s$. Sea current speed has been set to be $v_x = v = -0.25m/s$, $v_y = 0.0m/s$. It can be noted that, for the given initial position and orientation of the vehicle, it converges to the line in around $7.5m$ of the line's length, which is a portion of 15% of line's $50m$ length. Also, the first overshoot is only $0.75m$, and the distance to the line completely converges to zero in about $50s$, see Fig. 1b.

It is interesting to note the spikes around $t = 135.5s$ and $164s$ in Fig. 1b, 1c, and 1d. These are the consequences of switching the line of the lawnmower which is being actively followed. As soon as the vehicle approaches within $R_{victory} = 1m$ of the current waypoint towards which it is moving, the next line is switched to be followed.

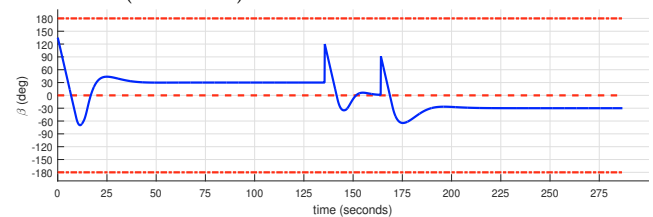
Also, in Fig. 1c the heading error β converges to some constant nonzero value when the vehicle converges to the line, because the vehicle starts moving sideways with



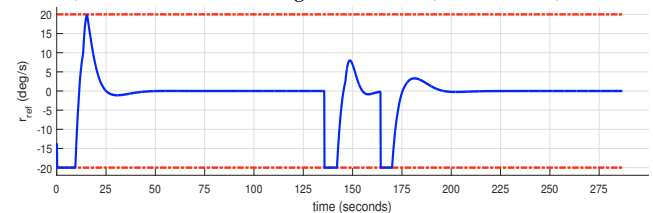
(a) Reference lawnmower lines at a constant depth (blue), defined by given waypoints (green markers). Vehicle's path while following the lines of the set lawnmower pattern (red). Vehicle's heading (black triangles).



(b) Distance to the given lawnmower line(s) (blue). Zero reference for the distance (red dashed).



(c) Vehicle's heading error w.r.t. the orientation of the lawnmower line(s) being followed (blue). Zero reference for the heading error (red dashed). Bounds of the heading error values (red dash-dot).



(d) Optimized control - reference yaw rate (blue). Bounds of the yaw rate values (red dash-dot).

Fig. 1. Simulation results

a constant surge speed and slip angle in order to compensate for the sea current disturbance, see Fig. 1a. This constant slip angle means zero yaw rate, which can be seen in Fig. 1d in time intervals $40 - 135.5s$ and $208 - 287s$.

Yaw rate (see Fig. 1d) is piece-wise smooth or constant, except in the line switching time instants at $135.5s$ and $164s$. It is important that it is not jittery, having in mind that in real system implementations this jitter is transferred to the vehicles electronics and actuators, and it can cause them to last shorter due to wear-off.

For the simulations we used a machine with i5-6200U processor with 2 cores and clock frequency of $2.3 - 2.8GHz$, and with $16GB$ of RAM. It is important to emphasize that the execution times of the real-time control optimization algorithm per one MPC iteration were ranging from a couple of milliseconds to about $20ms$, which was more than fast enough having in mind that the sampling period of the system was $T_s = 125ms$.

5. EXPERIMENTAL RESULTS

5.1 Experiment setup

During the beginning of October 2016 we have conducted series of on-sea trials in Biograd na Moru to validate simulation results presented in Section 4. The hybrid AUV/ROV robotic platform e-URoPe (e-Underwater Robotic Pet, shown in Fig. 2) developed at CNR-ISSIA (Genoa, Italy) has been used for the experiments. Since e-URoPe, which has been used in ROV mode, is an inherently fully actuated vehicle, sway, pitch, and roll controllers had to be shut down. This way, it has been artificially transformed into an under-actuated vehicle, controlling only heave speed, surge speed, and yaw rate. The existing heave controller, already developed for e-URoPe, has been used to keep a constant depth. Experimental results validating the simulation results are given in Subsections 5.2 and 5.3 for surface and underwater experiments, respectively.

Firstly, the ACADO-ROS stand-alone simulator which was already developed needed to be integrated with the ROS environment which was developed by CNR for the e-URoPe ROV. The idea was to use MPC controller as yaw rate reference generator, while the e-URoPe simulator would be used for tracking of constant surge speed reference, and the MPC-generated yaw rate reference. Controllers of surge speed and yaw rate are PID controllers which have already been proven to have very good performance. Secondly, this integrated dual simulator structure needed to be connected with the e-URoPe ROV itself for the experiments to take place.

The line following model used in the experiments consisted of only states d , and β , given by (1) and (2), respectively. The reason for excluding (6) from the model for the real system implementation was the increased oscillatory behavior of the vehicle when it gets close to the line due to integral action. This was the consequence of GPS and USBL measurement noise switching the vehicle's position from one side of the line to another. In order to minimize the d_{int} state, optimization algorithm tended to change the sign of the state d , resulting in unwanted oscillations of the vehicle around the lines of the lawnmower pattern. This simplified approach gave much better control performance in the experimental results compared to the previously used approach. One of the reasons for this is that in the simulations, the perfect knowledge of vehicle's position, surge speed, and yaw rate was assumed, so this case of measurement error conditioning the control performance could not have happened.

Since the d_{int} state was not used any more, we have increased the K_β coefficient to $K_\beta = 0.1$ in order for the control optimization algorithm to put more effort into aligning the vehicle's heading with the orientation of the line being followed. Victory radius was increased to $R_{victory} = 2m$. Surge speed reference was set to a value $U_r = 0.2m/s$, since the vehicle is more maneuverable when moving at a slower speeds. Also the bounds for the yaw rate were lowered to $|r| \leq 12^\circ/s$. Larger values of yaw rate bounds in the experiments have caused the controller to generate more aggressive control values near the bounds due to the position estimation errors, which in



Fig. 2. e-URoPe hybrid AUV/ROV robotic platform used in the experiments.

turn resulted in the unwanted oscillations of the vehicle around the reference line.

Experiments testing the performance of the developed MPC-PID framework have been conducted both on the sea surface and underwater (at constant depth of $1.4m$). Low level yaw rate controller was a P controller with $K_p = 0.4$.

Also, both surface and underwater experiments were conducted with a guidance scheme from Bruzzone et al. (2016). It is composed of a Lyapunov-based virtual-target path-following algorithm that generates a heading reference, combined with a PID heading controller which directly generated a yaw torque command. We used this method to compare the performance of the proposed MPC-PID framework with it.

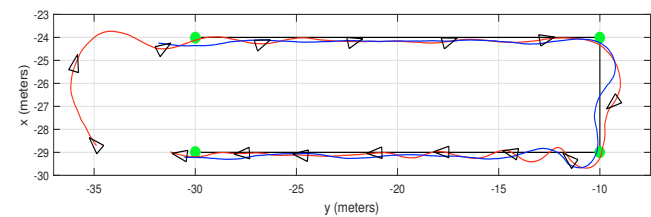
5.2 Surface experiments

The first set of experiments has been conducted with e-URoPe ROV on the sea surface, using GPS for localization. In Fig. 3a path of the ROV is shown when MPC-PID (red), and PID controller (blue) have been used. In both cases, the ROV shows oscillatory behavior when moving in the close vicinity of the given lines. MPC-PID controlled ROV converges to the line in a matter of 50s and keeps on moving really close to the line (see Fig. 3b).

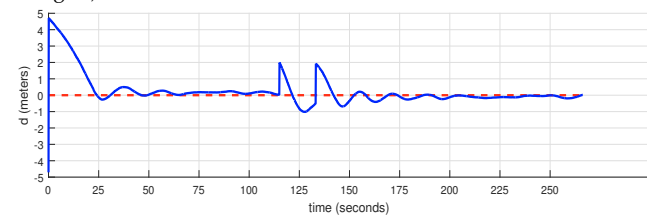
However, its heading error w.r.t. the desired line does not converge to a constant value, see Fig. 3c. This value should be zero in case no disturbance is present, and a nonzero value in case disturbance is present and rejected. Afore mentioned oscillations have an amplitude of less than $0.5m$, which is a small value of the same order of magnitude as the GPS sensor precision class. But in case that the ROV or any other autonomous marine vehicle with this control algorithm on-board are deployed for sea floor sonar scanning, the resulting interlaced sonar scans would be of greatly deteriorated quality.

In the Fig. 3d it can be noted that the used low level yaw rate controller tracks the reference yaw rate values with perhaps only a small delay which is tolerable. This means that the subpar performance of the yaw rate tracking controller can be ruled out as a possible cause of the vehicle's oscillations around the given lines.

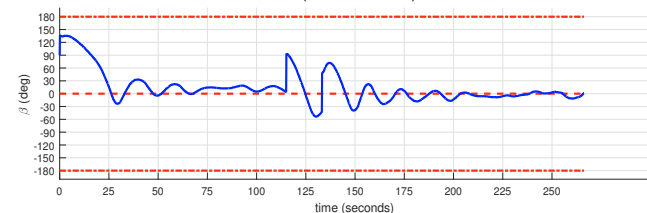
Oscillatory behavior could have been the consequence of the disturbances such as: waves hitting the GPS sensor



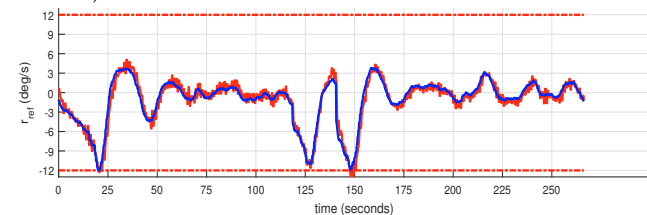
(a) Reference lawnmower lines at a constant depth (black), defined by given waypoints (green markers). Vehicle's path while following the lines of the set lawnmower pattern: MPC controller (red), PID controller (blue). Heading of the vehicle controlled by MPC (black triangles).



(b) MPC controller. Distance to the given lawnmower line(s) (blue). Zero reference for the distance (red dashed).



(c) MPC controller. Vehicle's heading error w.r.t. the orientation of the lawnmower line(s) being followed (blue). Zero reference for the heading error (red dashed). Bounds of the heading error values (red dash-dot).



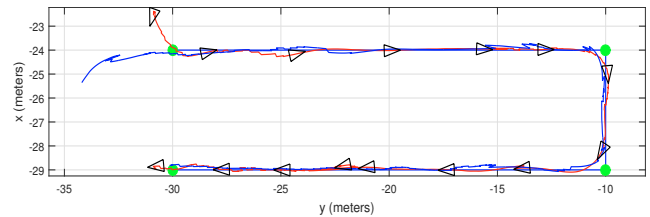
(d) MPC controller. Estimated tracking (red) of the reference yaw rate (blue). Bounds of the yaw rate values (red dash-dot).

Fig. 3. Surface experiments.

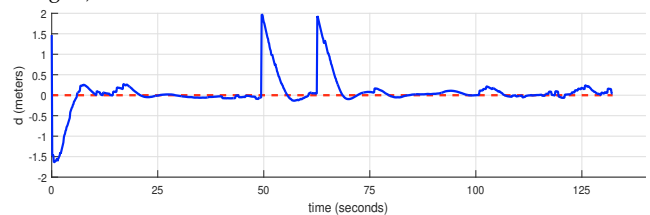
causing measurement error (and also swaying the vehicle from the desired line), sea current, or unoptimized parameters of Kalman filter for position estimation. Also, it could be the case that the flat square-shaped front side of the ROV was causing it problems with movement on the sea surface due to the hydrodynamic effects, increased drag, etc.

5.3 Underwater experiments

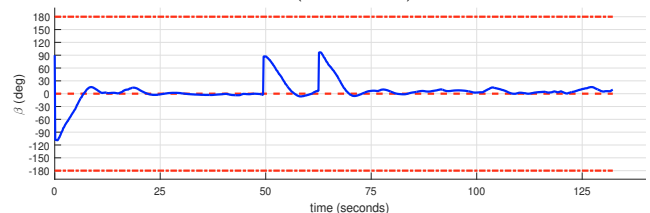
In addition to the experiments on the sea surface with e-URoPe ROV, underwater experiments were also conducted. During these experiments, USBL was used for localization. For the same given lawnmower pattern segment as in the Subsection 5.2, the path of the vehicle is given in Fig. 4a, for both MPC-PID framework (red) and PID controller (blue). Compared to the line following on the surface, these underwater experiment results are



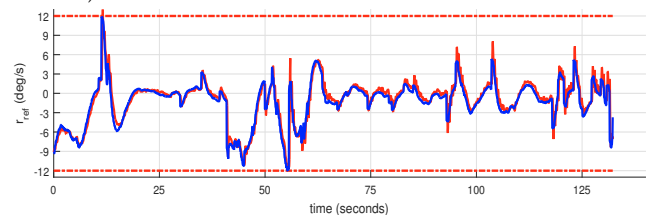
(a) Reference lawnmower lines at a constant depth (black), defined by given waypoints (green markers). Vehicle's path while following the lines of the set lawnmower pattern: MPC controller (red), PID controller (blue). Heading of the vehicle controlled by MPC (black triangles).



(b) MPC controller. Distance to the given lawnmower line(s) (blue). Zero reference for the distance (red dashed).



(c) MPC controller. Vehicle's heading error w.r.t. the orientation of the lawnmower line(s) being followed (blue). Zero reference for the heading error (red dashed). Bounds of the heading error values (red dash-dot).



(d) MPC controller. Estimated tracking (red) of the reference yaw rate (blue). Bounds of the yaw rate values (red dash-dot).

Fig. 4. Underwater experiments.

much better. The vehicle does not oscillate around the line (except for a few position estimation outliers) in the case of both MPC-PID and PID controller. The distance to the desired line for MPC-PID framework (see Fig. 4b) converges to zero in about 25s, and stays close to zero.

Heading error w.r.t. the desired line orientation (see Fig. 4c) also converges to zero as soon as the vehicle steadily converges to the line, around 25s from the start of the experiment. Since the experiments have been conducted in shallow waters, the effect of sea current was negligible, so the vehicle has been able to follow the desired lines, decreasing both the distance and the heading error to zero.

The same P controller mentioned in Subsection 5.2 has been used for yaw rate tracking, whose tracking performance for the underwater experiment is shown in Fig. 4d. It can be noted that its performance is really good, and

that it has a short response time, meaning that the reference is tracked fast and precise. Also, reference yaw rate drops to values closer to zero as soon as the vehicle converges to the line, meaning that the vehicle moves with a constant heading, aligned with the followed line.

Looking only at the Fig. 3a and 4a, and comparing the performance of MPC-PID framework and PID line following controller alone, it can be deduced that there is not a big difference between the control performance of the two approaches. However, if yaw rate tracking graphs in Fig. 3d and 4d are analyzed, it can be noted that both the reference and accomplished values of yaw rate are within the set bounds.

On the other hand, when the guidance scheme from Bruzzone et al. (2016) was used, estimated yaw rate values violated the set bounds, which is shown in Fig. 5. In this case, yaw rate was not controlled directly, so it could not even be saturated. Even if a PID yaw rate controller were used with a saturation block, this kind of control signal cut-off can, generally speaking, endanger the system's stability and deteriorate system's performance. Also, this kind of constraints violation on control signal can make control allocation problem much harder.

6. CONCLUSION AND FUTURE WORK

MPC has been shown to have good performance when used for motion planning of the underactuated autonomous marine vehicle. Its main advantage as a motion planning method is that it generates kinematically (or even dynamically, depending on the model being used) feasible paths, for which it optimizes control signal(s) in such a way that the given constraints on system states and controls are met. This method takes into account the model, cost function, and constraints during the process of control signal optimization, as opposed to other methods such as PID and LCF control, which do not deal with the constraints explicitly, and satisfy them by eventually saturating the generated control signal.

Real-time MPC framework implementation has been achieved for the line following problem which has been addressed in this paper. Simulation and experimental results show good performance of MPC controller when compared to PID controller.

The main advantage of MPC over PID control in this perspective is its possibility to be used as a general control optimization framework. The goal of our research is to make some kind of adaptive, informational-gain-based motion planner which would be used for sonar sea floor scanning by underactuated (rudder-based) autonomous underwater vehicles. Informational gain interpolated function could be used in the optimization problem cost function, so this is our main motivation for using MPC as a control method.

Also, we will further research the possibility of using the dynamical model(s) of the marine vehicle(s) with real-time implementation on mind. The proof of system's stability when using MPC controller is another research problem which we are trying to solve in order to theoretically justify the chosen prediction horizon duration and the chosen cost function.

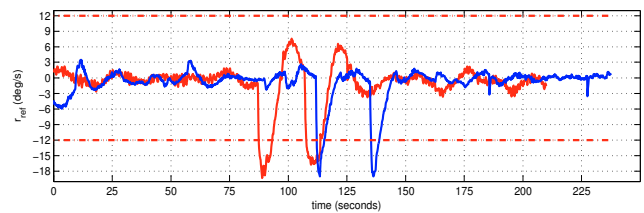


Fig. 5. Estimated yaw rate signals caused by the use of PID controller. Surface experiment (red), underwater experiment (blue). Bounds of the yaw rate values (red dash-dot).

REFERENCES

- Bibuli, M., Caharija, W., Pettersen, K.Y., Bruzzone, G., Caccia, M., and Zereik, E. (2014). ILOS guidance-experiments and tuning. *IFAC Proceedings Volumes*, 47(3), 4209–4214.
- Breivik, M. and Fossen, T.I. (2004). Path following of straight lines and circles for marine surface vessels. *Proceeding of the 6th IFAC CAMS*.
- Bruzzone, G., Bibuli, M., Zereik, E., Ranieri, A., and Caccia, M. (2016). Cooperative adaptive guidance and control paradigm for marine robots in an emergency ship towing scenario. *International Journal of Adaptive Control and Signal Processing*.
- Caccia, M., Bibuli, M., Bono, R., and Bruzzone, G. (2008). Basic navigation, guidance and control of an unmanned surface vehicle. *Autonomous Robots*, 25(4), 349–365.
- Fossen, T.I. (2011). *Handbook of marine craft hydrodynamics and motion control*.
- Houska, B., Ferreau, H., and Diehl, M. (2011). ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization. *Optimal Control Applications and Methods*, 32(3), 298–312.
- Muske, K.R. and Badgwell, T.A. (2002). Disturbance modeling for offset-free linear model predictive control. *Journal of Process Control*, 12(5), 617–632.
- Oh, S.R. and Sun, J. (2010). Path following of underactuated marine surface vessels using line-of-sight based model predictive control. *Ocean Engineering*, 37(2–3), 289 – 295.
- Pannocchia, G. (2003). Robust disturbance modeling for model predictive control with application to multivariable ill-conditioned processes. *Journal of Process Control*, 13(8), 693–701.
- Tahirovic, A. and Magnani, G. (2011). General framework for mobile robot navigation using passivity-based MPC. *IEEE Transactions on Automatic Control*, 56(1), 184–190.
- Tahirovic, A., Brkic, M., Magnani, G., and Bascetta, L. (2014). A planner for all terrain vehicles on unknown rough terrains based on the MPC paradigm and D*-like algorithm. In *IEEE International Conference on Robotics and Automation ICRA 2014. Workshop on Modelling, Estimation, Perception and Control of All Terrain Mobile Robots*. IEEE.
- Yi, B., Qiao, L., and Zhang, W. (2016). Two-time scale path following of underactuated marine surface vessels: Design and stability analysis using singular perturbation methods. *Ocean Engineering*, 124, 287 – 297.