Journal of Intelligent & Robotic Systems
https://doi.org/10.1007/s10846-019-01128-5

# Sliding Mode Control with Gaussian Process Regression for Underwater Robots

Gabriel S. Lima[1] · Sebastian Trimpe[2] · Wallace M. Bessa[1]

Received: 15 July 2019 / Accepted: 27 November 2019

## Abstract

Sliding mode control is a very effective strategy in dealing not only with parametric uncertainties, but also with unmodeled dynamics, and therefore has been widely applied to robotic agents. However, the adoption of a thin boundary layer neighboring the switching surface to smooth out the control law and to eliminate the undesired chattering effect usually impairs the controller's performance and leads to a residual tracking error. As a matter of fact, underwater robots are very sensitive to this issue due to their highly uncertain plants and unstructured operating environments. In this work, Gaussian process regression is combined with sliding mode control for the dynamic positioning of underwater robotic vehicles. The Gaussian process regressor is embedded within the boundary layer in order to enhance the tracking performance, by predicting unknown hydrodynamic effects and compensating for them. The boundedness and convergence properties of the tracking error are analytically proven. Numerical results confirm the improved performance of the proposed control scheme when compared with the conventional sliding mode approach.

**Keywords** Sliding mode control · Gaussian process regression · Underwater robotic vehicle · Dynamic positioning system

**Mathematics Subject Classification (2010)** 68T40 · 70E60 · 70Q05 · 93E35

## 1 Introduction

Underwater robotic systems play an essential role in ocean monitoring and exploration [1, 2]. In fact, since the advent of low-cost remotely operated vehicles (ROV) in the last decade, there was a significant increase in the application of underwater robots to offshore exploration [3]. In this way, these robots have been substituting for divers in the accomplishment of tasks that could threaten human safety and health.

However, despite their broad range of applications, the problem of designing accurate positioning systems for underwater robotic vehicles still challenges many engineers and researchers interested in this particular branch of engineering science. The main issue is associated with the mathematical representation of the surrounding fluid dynamics and its effects on the robot. A possible way to describe the fluid-body interaction is the adoption of computational fluid dynamics (CFD) [4–8]. CFD methods provide a fairly reliable description of the hydrodynamic effects, but are computationally expensive and usually not feasible for real time controllers [9, 10]. A common alternative to this approach is the application of a lumped-parameters model to estimate hydrodynamic effects [11–15]. Nevertheless, considering that a lumped-parameters system relies on the estimation of empirical coefficients that vary according to the flow conditions, the choice of constant coefficients for all situations represents an oversimplification of the fluid-body interaction [16].

In order to overcome this impasse, computational intelligence may be used to estimate the hydrodynamic effects. Bessa et al. [16], for instance, propose a control scheme for an autonomous diving agent, which combines

✉ Wallace M. Bessa
wmbessa@ct.ufrn.br

Gabriel S. Lima
limagabriel@ufrn.edu.br

Sebastian Trimpe
trimpe@is.mpg.de

[1] RoboTeAM - Robotics & Machine Learning, Federal University of Rio Grande do Norte, Natal, Brazil

[2] Intelligent Control Systems Group at the Max Planck Institute for Intelligent Systems, Stuttgart, Germany

Published online: 13 January 2020

feedback linearization with an adaptive fuzzy inference system. In [17], an adaptive neural network is embedded in a sliding mode controller for the depth regulation of a microdiving robot. Both of these works apply soft computing algorithms to approximate the hydrodynamic effects due to fluid-body interaction and to compensate for them within the control law as well. However, neither of the two mentioned approaches allows for easily obtaining the bounds of the estimation error.

Gaussian process regression (GPR), by contrast, is capable of providing not only a predicted value but also a distribution for the prediction. Thus, GPR represents an appealing approach to deal with uncertainties. A Gaussian process may be understood as an extension of Gaussian random variables to distributions over a suitable function space [18]. GPR can be adopted as a non-parametric model to represent an unknown function and to estimate both structured and unstructured uncertainties related to plant dynamics. For example, GPR has been combined with linear quadratic regulators (LQR) [19, 20] and active disturbance rejection control (ADRC) [21] to effectively find the controllers' open parameters from experimental data. Other examples include model predictive control [22–24] and model reference adaptive control [25, 26].

Because of its robustness against modeling inaccuracies, sliding mode control (SMC) has proven to be a very attractive approach to be employed in underwater vehicles [27–29]. Nevertheless, a well known handicap of conventional sliding mode controllers is the chattering effect. Although a properly designed boundary layer has the capacity to completely eliminate chattering, the adoption of this strategy turns perfect tracking into a tracking with guaranteed precision problem [30]. Therefore, considering the aforementioned benefits of Gaussian process regression, GPR can be also used to overcome the drawbacks of smooth sliding controllers. Aran and Unel [31], for instance, combined GPR with sliding modes for the control scheme of a diesel engine. The algorithm was trained off-line and used to define the feedforward terms of the control law [31].

In this work, we propose the adoption of a Gaussian process regressor within a sliding mode controller for the dynamic positioning of underwater robotic vehicles. GPR is embedded in the boundary layer of a smooth sliding mode controller to predict unknown hydrodynamic effects and compensate for them. Instead of offline supervised training, we use the method of overlapping sliding windows, which allows the regressor to gradually learn by interacting with the environment. This strategy has already been implemented in an underwater vehicle, but only for depth control [32]. In the present paper, this approach is extended to a multivariable control problem with four controllable degrees of freedom. The boundedness and convergence properties of the related tracking error is proven by means of a Lyapunov stability analysis. Numerical simulations are carried out in order to evaluate the control system performance.

## 2 Dynamic Model

The dynamic behavior of an underwater vehicle is governed by its rigid-body dynamics and strongly affected by the interaction with the surrounding fluid. Rigid-body effects are usually described by means of a system of ordinary differential equations, which can be obtained by either Newtonian or Lagrangian approaches. Fluid-body interaction, on the other hand, is mathematically represented by set of partial differential equations, namely Navier-Stokes equations. However, for real-time applications, whether control or simulation problems, a lumped-parameters model is preferable to approximate the hydrodynamic effects, and also the most commonly used approach in the literature [11–15].

On this basis, the equations of motion for underwater vehicles can be presented with respect to an inertial reference frame or with respect to a body-fixed reference frame, Fig. 1.

Equation 1 shows the equations of motion with respect to the body-fixed reference frame [10]:

$$\boldsymbol{\mathcal{M}}\dot{\boldsymbol{v}} + \boldsymbol{k}(\boldsymbol{v}) + \boldsymbol{h}(\boldsymbol{v}) + \boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{\tau}, \tag{1}$$

where $\boldsymbol{v} = [v_x \;\; v_y \;\; v_z \;\; \omega_x \;\; \omega_y \;\; \omega_z]^\top$ stands for both linear and angular velocities in the body-fixed frame, $\boldsymbol{x} = [x \;\; y \;\; z \;\; \alpha \;\; \beta \;\; \gamma]^\top$ represents the position/attitude vector in the inertial frame, $\boldsymbol{\mathcal{M}}$ is the inertia matrix, $\boldsymbol{k}(\boldsymbol{v})$ stands for Coriolis and centrifugal forces, $\boldsymbol{h}(\boldsymbol{v})$ introduces the hydrodynamic damping, $\boldsymbol{g}(\boldsymbol{v})$ represents both gravity and buoyancy forces, and $\boldsymbol{\tau}$ is the vector of control efforts.

Nevertheless, in some kinds of underwater robots, such as remotely operated vehicles, the metacentric height, i.e.

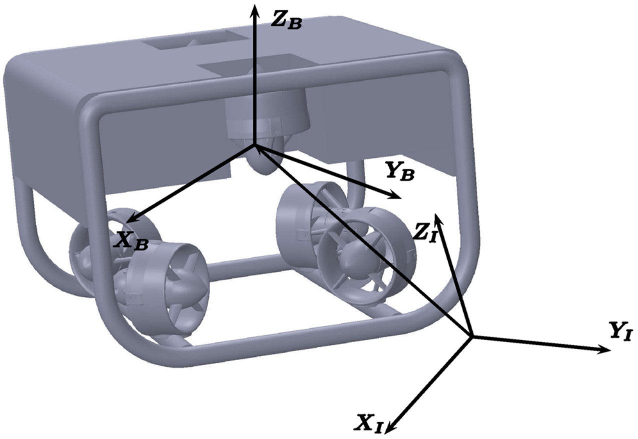


**Fig. 1** Underwater robot with both inertial ($I$) and body-fixed ($B$) reference frames

the distance between buoyancy and gravity centers, is sufficiently large to provide the self-stabilization of roll ($\alpha$) and pitch ($\beta$) angles. This specific constructive feature enables the number of degrees of freedom (DOF) to be reduced to four, $\boldsymbol{\upsilon} = [\upsilon_x \ \upsilon_y \ \upsilon_z \ \omega_z]^\top$ and $\boldsymbol{x} = [x \ y \ z \ \gamma]^\top$. Moreover, it also allows the vertical motion (heave) to be decoupled from the horizontal plane [9, 10, 15, 33–35].

In order to develop the simulator that will be used to validate the proposed control law, a lumped-parameters approach is used to represent the hydrodynamic effects. Since, in the case of underwater vehicles, the typical range of velocities usually does not exceed 2 m/s, Morison equation can be used to approximate fluid-body interaction [36]:

$$F_h = \frac{1}{2} C_D A \, \rho \, \upsilon |\upsilon| + C_M \rho \nabla \dot{\upsilon} \,, \tag{2}$$

where $F_h$ stands for the hydrodynamic force, $A$ is a reference area, $\rho$ is the fluid density, $\nabla$ is the fluid's displaced volume, $C_D$ and $C_M$ are hydrodynamic coefficients.

The first term in Eq. 2 is the nonlinear hydrodynamic damping and its effects $\boldsymbol{h}(\boldsymbol{\upsilon})$ over the vehicle can be described in the body-fixed reference frame by:

$$\boldsymbol{h}(\boldsymbol{\upsilon}) = \frac{1}{2} \rho [C_{D_x} \upsilon_x |\upsilon_x| C_{D_y} \upsilon_y |\upsilon_y| C_{D_z} \upsilon_z |\upsilon_z| C_{D_\gamma} \omega_z |\omega_z|]^\top , \tag{3}$$

where the parameters $C_{D_x}$, $C_{D_y}$, $C_{D_z}$ and $C_{D_\gamma}$ depend not only on the shape of the vehicle but also on flow conditions.

The second term in Eq. 2 stands for the hydrodynamic added mass and, at low speeds, may be incorporated into Eq. 1 by means of a diagonally dominant matrix [10]:

$$\mathcal{M}_A = \mathrm{diag}\{C_{\mathcal{M}_x} \rho \nabla, \ C_{\mathcal{M}_y} \rho \nabla, \ C_{\mathcal{M}_z} \rho \nabla, \ C_{\mathcal{M}_\gamma} \rho \nabla\}, \tag{4}$$

with parameters $C_{\mathcal{M}_x}$, $C_{\mathcal{M}_y}$, $C_{\mathcal{M}_z}$ and $C_{\mathcal{M}_\gamma}$ also relying on vehicle's shape and flow conditions. The matrix $\boldsymbol{\mathcal{M}}_A$ may be combined with the rigid-body inertia matrix to obtain the matrix $\boldsymbol{\mathcal{M}}$ in Eq. 1.

Considering that Morison equation depends on the estimation of empirical coefficients that vary according to the flow conditions, we shall use it only for the simulator. For the development of the control law, we propose a different approach. Only the effects related to rigid-body dynamics, which can be accurately estimated, are incorporated into the controller in a straightforward manner. The hydrodynamic effects, however, are predicted from data by means of a Gaussian process regressor, which will be incorporated in the control scheme.

## 3 Sliding Mode Control with Gaussian Process Regression

In this section, we present the sliding mode controller that is augmented with Gaussian process regression for estimating model uncertainty, and its boundedness and convergence analysis as well.

### 3.1 Sliding Mode Control

For control purposes, the equation of motion (1) may be rewritten with respect to the inertial reference frame. Recalling that

$$\dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{x}) \boldsymbol{\upsilon} \,, \tag{5}$$

with $\boldsymbol{J}(\boldsymbol{x}) \in \mathbb{R}^{4 \times 4}$ being the Jacobian transformation matrix, it can be easily verified that $\boldsymbol{\upsilon} = \boldsymbol{J}^{-1} \dot{\boldsymbol{x}}$ and $\dot{\boldsymbol{\upsilon}} = \dot{\boldsymbol{J}}^{-1} \dot{\boldsymbol{x}} + \boldsymbol{J}^{-1} \ddot{\boldsymbol{x}}$.

Thus, considering that the restoring forces could be passively compensated, the equation of motion of the underwater robot with respect to the inertial frame, becomes

$$\boldsymbol{M} \ddot{\boldsymbol{x}} = \boldsymbol{f} + \boldsymbol{u} \,, \tag{6}$$

with $\boldsymbol{M} = \boldsymbol{J}^{-\top} \boldsymbol{\mathcal{M}} \boldsymbol{J}^{-1}$, $\boldsymbol{u} = \boldsymbol{J}^{-\top} \boldsymbol{\tau}$, and $\boldsymbol{f} = -\boldsymbol{J}^{-\top} (\boldsymbol{\mathcal{M}} \dot{\boldsymbol{J}}^{-1} \dot{\boldsymbol{x}} + \boldsymbol{k} + \boldsymbol{h})$.

By taking possible uncertainties in the mathematical model into account, Eq. 6 is equivalently written as

$$\hat{\boldsymbol{M}} \ddot{\boldsymbol{x}} = \hat{\boldsymbol{f}} + \boldsymbol{u} + \boldsymbol{d} \,, \tag{7}$$

with $\hat{\boldsymbol{M}}$ and $\hat{\boldsymbol{f}}$ being estimates of $\boldsymbol{M}$ and $\boldsymbol{f}$, respectively, and $\boldsymbol{d} = \Delta \boldsymbol{M} \ddot{\boldsymbol{x}} - \Delta \boldsymbol{f}$ representing the total uncertainty with respect to the dynamic model, $\Delta \boldsymbol{M} = \hat{\boldsymbol{M}} - \boldsymbol{M}$, and $\Delta \boldsymbol{f} = \hat{\boldsymbol{f}} - \boldsymbol{f}$.

Now, let the dynamic positioning system be designed according to the sliding mode method [37, 38]. First, a sliding surface is defined in the state space by the equation

$$\boldsymbol{s}(\dot{\tilde{\boldsymbol{x}}}, \tilde{\boldsymbol{x}}) = \dot{\tilde{\boldsymbol{x}}} + \boldsymbol{\Lambda} \tilde{\boldsymbol{x}} \,, \tag{8}$$

with $\tilde{\boldsymbol{x}} = \boldsymbol{x} - \boldsymbol{x}_\mathrm{d}$ being the tracking error and $\boldsymbol{\Lambda} \in \mathbb{R}^{4 \times 4}$ a diagonal matrix with strictly positive entries $\lambda_i$. It should be highlighted that the sliding mode method assumes full state feedback, which in practice can be obtained using state estimators [37].

In order to avoid chattering, a thin boundary layer neighboring the sliding surface is designed by replacing the standard relay term $\mathrm{sgn}(\cdot)$ by the saturation function [37, 38]:

$$\mathrm{sat}(x) = \begin{cases} x & \text{if} \quad |x| \leq 1, \\ \mathrm{sgn}(x) & \text{if} \quad |x| > 1. \end{cases} \tag{9}$$

Therefore, the smooth sliding mode controller is designed to ensure the attractiveness of the boundary layer:

$$\boldsymbol{u} = -\hat{\boldsymbol{f}} - \hat{\boldsymbol{d}} + \hat{\boldsymbol{M}} \big[ \ddot{\boldsymbol{x}}_\mathrm{d} - \boldsymbol{\Lambda} \dot{\tilde{\boldsymbol{x}}} - \kappa \, \mathrm{sat}(\boldsymbol{\varphi}^{-1} \boldsymbol{s}) \big], \tag{10}$$

with $\hat{\boldsymbol{d}}$ standing for an estimate of $\boldsymbol{d}$, $\kappa$ representing the control gain, $\boldsymbol{\varphi} \in \mathbb{R}^{4 \times 4}$ being a diagonal matrix with strictly positive entries $\phi_i$, and $\mathrm{sat}(\boldsymbol{\varphi}^{-1} \boldsymbol{s}) = [\mathrm{sat}(s_x/\phi_x) \ \mathrm{sat}(s_y/\phi_y) \ \mathrm{sat}(s_z/\phi_z) \ \mathrm{sat}(s_\gamma/\phi_\gamma)]^\top$. The parameters $\phi_i$

represent the width of the boundary layer related to the $i^{\text{th}}$ degree of freedom and may be properly chosen in order to minimize the chattering effects.

With the purpose of investigating the dynamics of $\boldsymbol{s}$, Eq. 8 may be differentiated with respect to time:

$$\begin{aligned}\dot{\boldsymbol{s}} &= \ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{\Lambda}\,\dot{\tilde{\boldsymbol{x}}} = \ddot{\boldsymbol{x}} - \ddot{\boldsymbol{x}}_{\mathrm{d}} + \boldsymbol{\Lambda}\,\dot{\tilde{\boldsymbol{x}}} \\ &= \hat{\boldsymbol{M}}^{-1}(\hat{\boldsymbol{f}} + \boldsymbol{u} + \boldsymbol{d}) - \ddot{\boldsymbol{x}}_{\mathrm{d}} + \boldsymbol{\Lambda}\,\dot{\tilde{\boldsymbol{x}}}\,.\end{aligned}$$

Then, applying the control law (10) to $\dot{\boldsymbol{s}}$, we get:

$$\dot{\boldsymbol{s}} = \hat{\boldsymbol{M}}^{-1}(\hat{\boldsymbol{d}} - \boldsymbol{d}) - \kappa\,\mathrm{sat}(\boldsymbol{\varphi}^{-1}\boldsymbol{s})\,. \tag{11}$$

From Eq. 9, it can be verified that, inside the boundary layer, i.e. when $|s_i| \leq \phi_i$ for $i = x, y, z, \gamma$, the flow of the vector field becomes

$$\dot{\boldsymbol{s}} + \kappa\boldsymbol{\varphi}^{-1}\boldsymbol{s} = \hat{\boldsymbol{M}}^{-1}(\hat{\boldsymbol{d}} - \boldsymbol{d})\,, \tag{12}$$

which, in fact, shows that the trajectories of $\boldsymbol{s}$ are driven by $\hat{\boldsymbol{M}}^{-1}(\hat{\boldsymbol{d}} - \boldsymbol{d})$.

According to the sliding mode method, control performance relies on the convergence of the tracking error to the sliding surface. Thus, in order to improve trajectory tracking inside the boundary layer, it is essential to bring down the value of $\boldsymbol{s}$ by providing a good estimate $\hat{\boldsymbol{d}}$. Moreover, the value of $\boldsymbol{s}$ can also be understood as a reasonable metric for the success of the adopted approximation scheme.

### 3.2 Gaussian Process Regression

Considering that Gaussian process regressor can be used as a non-parametric model to describe a distribution over functions, GPR is adopted here to compute $\hat{\boldsymbol{d}}$ as a non-parametric estimate of $\boldsymbol{d}$. The estimate $\hat{\boldsymbol{d}}$ is used in the sliding mode controller, Eq. 10, to enhance the tracking performance inside the boundary layer. Because of the association between the disturbance vector and the sliding variables, as stated in Eq. 12, we propose the adoption of $\boldsymbol{s}$ in the regression scheme. Despite $\hat{\boldsymbol{d}}$ being coupled to $\boldsymbol{s}$ by means of $\hat{\boldsymbol{M}}$, the inertia matrix is typically diagonally dominant in underwater robots [10]. Thus, although multi-output GP schemes have already been proposed, in this work, we model each component of $\hat{\boldsymbol{d}}$ using an independent regressor and the corresponding component of $\boldsymbol{s}$. Following [18] and assuming noisy observations, one has

$$\bar{d} = d(s) + \varepsilon\,, \quad \varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2), \tag{13}$$

with $s$ being the sliding variable related to the corresponding component of $\boldsymbol{d}$, and the noise term $\varepsilon$ following a normal distribution with mean 0 and variance $\sigma_\varepsilon^2$. The index $i$ related to each degree of freedom is omitted in this subsection for the sake of clarity.

A Gaussian process (GP) can be understood as a probability distribution over the space of functions $d$ whose restriction to any finite number of function values is jointly Gaussian [18]. Thus, a GP is specified through its prior mean function $\mu$ and covariance function $k$:

$$d(s) \sim \mathcal{GP}\big(\mu(s), k(s, s')\big), \tag{14}$$

with $\mu(s) = \mathbb{E}[d(s)]$ being the expectation of $d(s)$ and $k(s, s') = \mathbb{V}[d(s), d(s')]$ denoting the covariance of $d(s)$ and $d(s')$.

In GPR, learning a function amounts to predicting the (normal) distribution of function values $d(s^*)$ at arbitrary inputs $s^*$ based on previous evaluations. Given $N$ data points $\mathcal{D}_N = \{s_n, d_n\}_{n=1}^N$, the posterior mean and variance of $d^*$ can be stated as

$$\mathbb{E}[d(s^*)|\mathcal{D}_N] = \mu(s^*) + k_N^\top(s^*)(K_N + \sigma_\varepsilon^2 I)^{-1}\tilde{d}_N, \tag{15}$$

$$\mathbb{V}[d(s^*)|\mathcal{D}_N] = k(s^*, s^*) - k_N^\top(s^*)(K_N + \sigma_\varepsilon^2 I)^{-1}k_N(s^*), \tag{16}$$

with $k_N(s^*) := [k(s^*, s_1)\ \ldots\ k(s^*, s_N)]^\top$, $\tilde{d}_N := [d_1 - \mu(s_1)\ \ldots\ d_N - \mu(s_N)]^\top$, and the Gramian matrix $K_N$ being defined by

$$K_N = \begin{bmatrix} k(s_1, s_1') & k(s_1, s_2') & \ldots & k(s_1, s_N') \\ k(s_2, s_1') & k(s_2, s_2') & \ldots & k(s_2, s_N') \\ \vdots & \vdots & \ddots & \vdots \\ k(s_N, s_1') & k(s_N, s_2') & \ldots & k(s_N, s_N') \end{bmatrix}. \tag{17}$$

In order to define the data set $\mathcal{D}_N$, the values of $d_n$ and $s_n$ are assessed by means of Eqs. 7 and 8, respectively. Regarding the estimates of $d_n$ using Eq. 7, it should be highlighted that, in this case, not only the state variables but also the accelerations must be available to compute $\boldsymbol{d} = \hat{\boldsymbol{M}}\ddot{\boldsymbol{x}} - \hat{\boldsymbol{f}} - \boldsymbol{u}$. Considering that underwater robots are usually equipped with inertial measurement units (IMUs), an estimate of the vector $\ddot{\boldsymbol{x}}$ is quite often at hand.

Furthermore, rolling regression is employed to allow GPR to gradually learn by interacting with the environment. Thus, a fixed sliding window with a constant size $r$ slides along subsets of the data as time progresses. With the aim of providing a smooth procedure, only the oldest data entry is updated with the arrival of new values of $d_n$ and $s_n$, i.e. old entries are discarded, as the window slides, the new values assimilated, and the window overlaps with the rest of the data.

At this point, we propose the adoption of the predictive mean (15) to compute the components of $\hat{\boldsymbol{d}}$, i.e. $\hat{d}_i = \mathbb{E}[d(s_i^*)|\mathcal{D}_N]$. Moreover, we can also take advantage of the predictive standard deviation to estimate the bounds of the disturbance term, i.e. $\hat{\boldsymbol{d}} - \vartheta\,\boldsymbol{\sigma} \leq \boldsymbol{d} \leq \hat{\boldsymbol{d}} + \vartheta\,\boldsymbol{\sigma}$, with $\vartheta$ being used to define a proper confidence interval, $\boldsymbol{\sigma} = [\sigma_x\ \sigma_y\ \sigma_z\ \sigma_\gamma]^\top$, and $\sigma_i^2 = \mathbb{V}[d(s_i^*)|\mathcal{D}_N]$, for $i = x, y, z, \gamma$. These bounds play an essential role in the tuning of the control gain, as can be seen in the stability analysis below.

### 3.3 Boundedness and Convergence Analysis

In order to demonstrate the boundedness and convergence properties of the proposed control scheme, we first have to prove the attractiveness of the boundary layer.

From Eq. 11, it is noted that the convergence of $\boldsymbol{s}$ can be investigated in terms of its components. Thus, let a positive-definite Lyapunov function candidate be defined for each degree of freedom

$$V_i(t) = \frac{1}{2}\bar{s}_i^2\,, \tag{18}$$

with $i = x, y, z, \gamma$. By defining $\bar{s}_i$ according to

$$\bar{s}_i(\dot{\tilde{x}}_i, \tilde{x}_i) = s_i - \phi_i\,\mathrm{sat}(s_i/\phi_i)\,, \tag{19}$$

its absolute value may be understood as the distance between each $s_i$ and the corresponding boundary layer.

From Eq. 11 and noting that $\dot{\bar{s}}_i = \dot{s}_i$ outside the boundary layer, then the time derivative of $V_i$ becomes:

$$\dot{V}_i(t) = \bar{s}_i\dot{s}_i = \bar{s}_i\left[\left(\sum_{j=1}^{4}\hat{M}_{ij}^{-1}(d_j - \hat{d}_j)\right) - \kappa\,\mathrm{sat}(s_i/\phi_i)\right],$$

with $d_j$ and $\hat{d}_j$ representing the $j^{\text{th}}$ components of $\boldsymbol{d}$ and $\hat{\boldsymbol{d}}$, respectively, and $\hat{M}_{ij}^{-1}$ being the element in the $i^{\text{th}}$ row and $j^{\text{th}}$ column of the matrix $\hat{\boldsymbol{M}}^{-1}$.

Moreover, since $\bar{s}_i = 0$ in the boundary layer, it can be verified that $\dot{V}_i(t) = 0$ in that region.

Now, by observing that $\mathrm{sat}(s_i/\phi_i) = \mathrm{sgn}(\bar{s}_i)$ outside the boundary layer, we obtain

$$\dot{V}_i(t) = -\bar{s}_i\left[\left(\sum_{j=1}^{4}\hat{M}_{ij}^{-1}(\hat{d}_j - d_j)\right) + \kappa\,\mathrm{sgn}(\bar{s}_i)\right].$$

Considering that $|\sum_{j=1}^{4}\hat{M}_{ij}^{-1}(\hat{d}_j - d_j)| \leq \|\hat{\boldsymbol{M}}^{-1}(\hat{\boldsymbol{d}} - \boldsymbol{d})\|_\infty \leq \|\hat{\boldsymbol{M}}^{-1}\|_\infty\|(\hat{\boldsymbol{d}} - \boldsymbol{d})\|_\infty \leq \vartheta\,\|\hat{\boldsymbol{M}}^{-1}\|_\infty\|\boldsymbol{\sigma}\|_\infty$ and defining the control gain according to $\kappa > \eta + \vartheta\,\|\hat{\boldsymbol{M}}^{-1}\|_\infty\|\boldsymbol{\sigma}\|_\infty$, with $\eta$ being a strictly positive parameter, we get

$$\dot{V}_i(t) \leq -\eta|\bar{s}_i|\,.$$

This implies that $V_i(t) \leq V_i(0)$ and that any initial state will be attracted to the boundary layer. Moreover, recalling that $V_i$ is positive definite, its time derivative is equal to zero, $\dot{V}_i(t) = 0$, inside the boundary layer, and $\dot{V}_i(t) < 0$ outside of it, we can assure that the states remain in the boundary layer as $t \to \infty$.

Now, it can be proved that, once inside the boundary layer, the tracking error exponentially converges to a small bounded region in the vicinity of $(\dot{\tilde{x}}_i, \tilde{x}_i) = (0, 0)$. From Eq. 8 and considering that $-\phi_i \leq s_i(\dot{\tilde{x}}_i, \tilde{x}_i) \leq \phi_i$ inside the boundary layer, we have for each degree of freedom:

$$-\phi_i \leq \dot{\tilde{x}}_i + \lambda_i\tilde{x}_i \leq \phi_i\,. \tag{20}$$

Thus, multiplying (20) by $e^{\lambda_i t}$ and integrating between 0 and $t$:

$$-\phi_i e^{\lambda_i t} \leq (\dot{\tilde{x}}_i + \lambda_i\tilde{x}_i)e^{\lambda_i t} \leq \phi_i e^{\lambda_i t}\,,$$

$$-\phi_i e^{\lambda_i t} \leq \frac{d}{dt}(\tilde{x}_i e^{\lambda_i t}) \leq \phi_i e^{\lambda_i t}\,,$$

$$-\phi_i\int_0^t e^{\lambda_i\xi}\,d\xi \leq \int_0^t \frac{d}{d\xi}(\tilde{x}_i e^{\lambda_i\xi})\,d\xi \leq \phi_i\int_0^t e^{\lambda_i\xi}\,d\xi\,,$$

$$-\frac{\phi_i}{\lambda_i}e^{\lambda_i t} + \frac{\phi_i}{\lambda_i} \leq \tilde{x}_i(t)e^{\lambda_i t} - \tilde{x}_i(0) \leq \frac{\phi_i}{\lambda_i}e^{\lambda_i t} - \frac{\phi_i}{\lambda_i}\,,$$

$$-\frac{\phi_i}{\lambda_i} - \left[|\tilde{x}_i(0)| + \frac{\phi_i}{\lambda_i}\right]e^{-\lambda_i t} \leq \tilde{x}_i(t) \leq \frac{\phi_i}{\lambda_i} + \left[|\tilde{x}_i(0)| + \frac{\phi_i}{\lambda_i}\right]e^{-\lambda_i t}\,.$$

Furthermore, for $t \to \infty$:

$$-\frac{\phi_i}{\lambda_i} \leq \tilde{x}_i \leq \frac{\phi_i}{\lambda_i}\,. \tag{21}$$

By applying Eqs. 21 to 20, it can be easily verified that:

$$-2\phi_i \leq \dot{\tilde{x}}_i \leq 2\phi_i\,. \tag{22}$$

Hence, it follows that the tracking error vector will exponentially converge to the closed region $\Phi = \{(\dot{\tilde{\boldsymbol{x}}}, \tilde{\boldsymbol{x}}) \in \mathbb{R}^8 : |\dot{\tilde{x}}_i| \leq 2\phi_i \text{ and } |\tilde{x}_i| \leq \phi_i/\lambda_i\,;\ i = x, y, z, \gamma\}$.

It should be noted that, in theory, since Gaussian distribution has infinite support, it will always be possible for a value of $\boldsymbol{d}$ to arise outside the bounds of confidence. However, as a matter of fact, in real world applications, unlimited disturbances do not usually occur. In addition, parameters $\eta$ and $\vartheta$ may be chosen appropriately to capture the robustness that is desired.

Finally, the proposed control scheme is summarized in Algorithm 1.

**Algorithm 1** Dynamic positioning with SMC and GPR.

```
Define control parameters
Specify GP prior (mean $\mu$, kernel $k$)
Define initial states: $\dot{\boldsymbol{x}}_0,\ \boldsymbol{x}_0$
Initialize data set: $\mathcal{D}_0$
loop
    Evaluate desired trajectory: $\ddot{\boldsymbol{x}}_\mathrm{d},\ \dot{\boldsymbol{x}}_\mathrm{d},\ \boldsymbol{x}_\mathrm{d}$
    Compute tracking error: $\dot{\tilde{\boldsymbol{x}}},\ \tilde{\boldsymbol{x}}$
    $\boldsymbol{s} \leftarrow \dot{\tilde{\boldsymbol{x}}} + \boldsymbol{\Lambda}\tilde{\boldsymbol{x}}$
    $\hat{d}_i \leftarrow \mathbb{E}[d(s_i^*)|\mathcal{D}_{n-1}]\,;\ i = x, y, z, \gamma$
    $\sigma_i^2 \leftarrow \mathbb{V}[d(s_i^*)|\mathcal{D}_{n-1}]\,;\ i = x, y, z, \gamma$
    $\kappa \leftarrow \eta + \vartheta\,\|\hat{\boldsymbol{M}}^{-1}\|_\infty\|\boldsymbol{\sigma}\|_\infty$
    $\boldsymbol{u} \leftarrow -\hat{\boldsymbol{f}} - \hat{\boldsymbol{d}} + \hat{\boldsymbol{M}}[\ddot{\boldsymbol{x}}_\mathrm{d} - \boldsymbol{\Lambda}\dot{\tilde{\boldsymbol{x}}} - \kappa\,\mathrm{sat}(\boldsymbol{\varphi}^{-1}\boldsymbol{s})]$
    Apply $\boldsymbol{u}$ to the dynamic model
    Update states: $\dot{\boldsymbol{x}},\ \boldsymbol{x}$
    Update GP posterior: $\mathbb{E}[d(s_i^*)|\mathcal{D}_n],\ \mathbb{V}[d(s_i^*)|\mathcal{D}_n]$
end loop
```

**Table 1** Influence of the hyperparameters on the overall control performance

| Case | $\sigma_\varepsilon$ | $\sigma_f$ | $\int \lvert\kappa\rvert \mathrm{d}t$ | $\int \lvert u_z\rvert \mathrm{d}t$ | $\int \lvert\tilde{z}\rvert \mathrm{d}t$ |
|---|---|---|---|---|---|
| Sim. 1 | — | — | 0.6188 | 98.5227 | 0.2979 |
| Sim. $2_a$ | 1.2 | 0.1 | 0.5854 | 98.4066 | 0.2757 |
| Sim. $2_b$ | 1.2 | 0.5 | 1.3398 | 98.1947 | 0.0351 |
| Sim. $2_c$ | 1.2 | 1.0 | 1.4598 | 98.0554 | 0.0102 |
| Sim. $3_a$ | 0.8 | 0.1 | 0.5515 | 98.2922 | 0.2531 |
| Sim. $3_b$ | 0.8 | 0.5 | 0.9802 | 98.1171 | 0.0244 |
| Sim. $3_c$ | 0.8 | 1.0 | 1.0207 | 98.0369 | 0.0071 |
| Sim. $4_a$ | 0.4 | 0.1 | 0.4424 | 97.9713 | 0.1805 |
| Sim. $4_b$ | 0.4 | 0.5 | 0.5553 | 98.0450 | 0.0121 |
| Sim. $4_c$ | 0.4 | 1.0 | 0.5608 | 98.0331 | 0.0049 |
| Sim. $5_a$ | 0.1 | 0.1 | 0.2053 | 97.8546 | 0.0363 |
| Sim. $5_b$ | 0.1 | 0.5 | 0.2079 | 98.0535 | 0.0069 |
| Sim. $5_c$ | 0.1 | 1.0 | 0.2080 | 98.0601 | 0.0067 |

## 4 Simulation Results

The proposed control scheme is now evaluated by means of numerical simulations considering a computer implementation of the dynamic model in C++. The fourth order Runge-Kutta method is employed and sampling rates of 500 Hz for the controller and 1 kHz for system dynamics are chosen. In the dynamic model, the inertia matrix and the hydrodynamic quadratic damping are assumed to be, respectively, $\boldsymbol{\mathcal{M}} = \operatorname{diag}\{80\,\mathrm{kg}, 80\,\mathrm{kg}, 100\,\mathrm{kg}, 8\,\mathrm{kgm}^2\}$ and $\boldsymbol{h} = [125\,\upsilon_x|\upsilon_x|\,\mathrm{N}\;175\,\upsilon_y|\upsilon_y|\,\mathrm{N}\;250\,\upsilon_z|\upsilon_z|\,\mathrm{N}\;12.5\,\omega_z|\omega_z|\,\mathrm{Nm}]^\top$. By numerically integrating the dynamic model, state variables become available to the control scheme at each time step. In real world applications, on the other hand, state estimation is usually performed using an acoustic positioning system, with a network of transponders and receivers, and inertial measurement units [3]. Moreover, with the view to simulate an actual underwater robot, the control scheme is evaluated in the presence of actuator saturation.

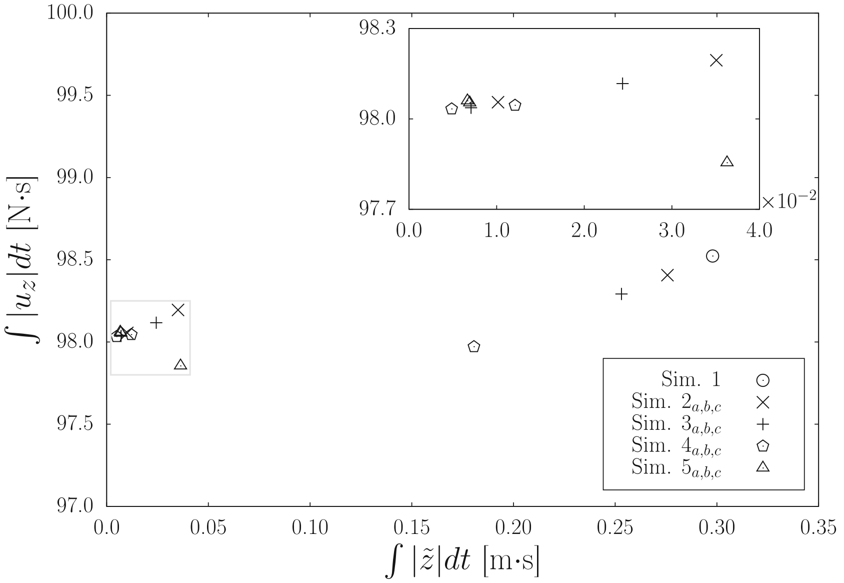
Control parameters are set to $\lambda_i = 0.4$, $\phi_i = 0.05$, $\eta = 0.005$, and $\vartheta = 2$. In addition, considering the previously mentioned issues with respect to the estimation of the fluid-body interaction, added mass and hydrodynamic damping are assumed to be unknown and not taken into account in the control law.

Regarding the adopted Gaussian process regressor, used to estimate the unknown hydrodynamic effects, the squared exponential kernel is chosen as covariance function:

$$k(s_i, s_i') = \sigma_f^2 \exp\left(-\frac{\|s_i - s_i'\|^2}{2\ell^2}\right) \tag{23}$$

with $\ell$ being the length-scale and $\sigma_f$ representing the prior standard deviation of the process. In addition to the

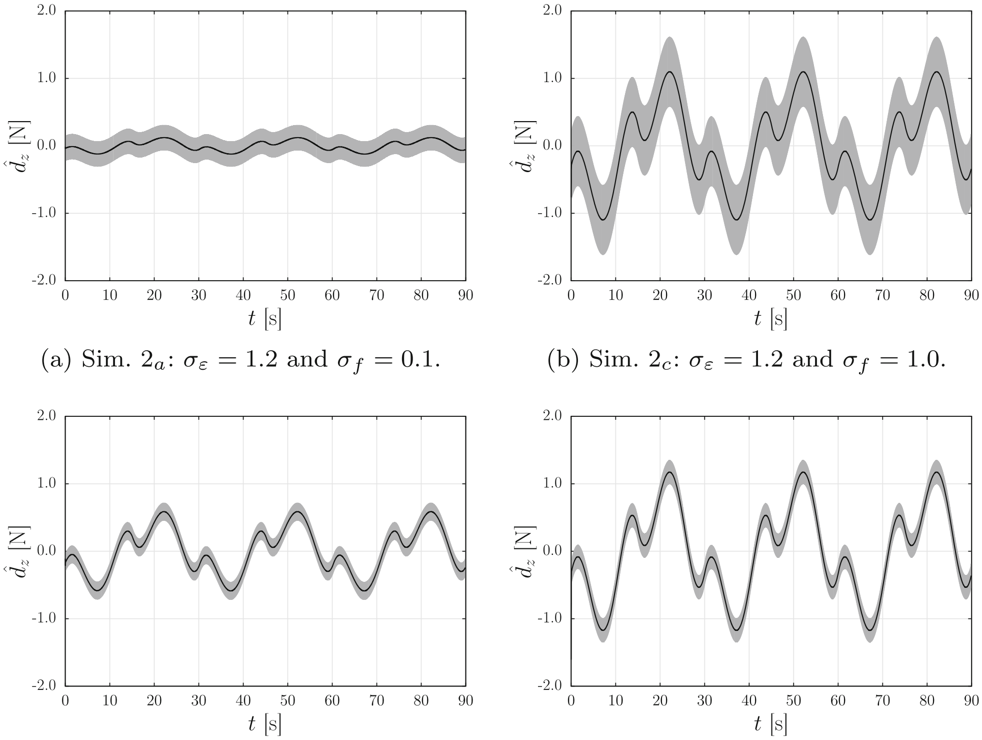
**Fig. 2** Influence of $\sigma_\varepsilon$ and $\sigma_f$ on the total control action and total tracking error



free parameters $\ell$ and $\sigma_f$, the noise standard deviation $\sigma_\varepsilon$ from Eq. 13 represents the third hyperparameter of the GP distribution. In all simulations the prior mean is set to zero and the length-scale is defined as $\ell = 0.05$.

In order to properly tune $\sigma_\varepsilon$ and $\sigma_f$, their effects are assessed by means of numerical simulations. Assuming that the initial state is equal to the initial desired state, i.e. $\tilde{z}(0) = [\dot{\tilde{z}}\ \tilde{z}]^\top = \mathbf{0}$, depth tracking of a sinusoidal trajectory $z_\mathrm{d} = 0.5[1 - \cos(\pi t/15)]$ m is taken into account. Table 1 presents the sum of the absolute values of the gain $\kappa$, of the control signal $u_z$, and of the tracking error $\tilde{z}$ obtained in 13 simulation studies. For comparison purposes, the conventional sliding mode controller is also applied to the same tracking problem, Sim. 1. By combining 4 different values for $\sigma_\varepsilon$ and 3 different values for $\sigma_f$, the influence of these hyperparameters is evaluated in the other 12 cases, Sim. $2_a$–$5_c$. It should be noted that the proposed controller is easily turned into the classical sliding mode scheme by setting $\hat{\boldsymbol{d}} = \mathbf{0}$ and defining a fixed gain $\kappa$. Figure 2 shows the influence of the hyperparameters on the overall control performance, and their effects on the regression output are graphically depicted in Fig. 3.

Both Table 1 and Fig. 2 show that, when compared with the conventional sliding mode controller (SMC), the proposed scheme (SMC+GPR) is able to significantly increase the tracking efficiency, by reducing not only the overall tracking error but also the required control effort. It can be observed in Fig. 2 that all points related to the proposed controller, Sim. $2_a$–$5_c$, are located on the left side of the plot and below the standard SMC, Sim. 1, which demonstrates the controller's efficacy when combined with GPR. This improved performance is due to the regressor's ability to predict and compensate for model uncertainties. Figure 2 also shows the consistency with respect to the control performance, even considering variations in hyperparameters, when compared with the classical SMC. However, Fig. 3 shows that proper choice of hyperparameters can further enhance GPR performance. For instance, it can be observed in Fig. 3 that a higher noise variance $\sigma_\varepsilon^2$ leads to a more scattered estimate $\hat{d}_z$. Furthermore, an increase in the prior variance of the process $\sigma_f^2$ improves the ability of the GPR algorithm to correlate inputs that lie far apart in the data set. As a matter of fact, according to Table 1, it can be verified that the value of $\sigma_f$ plays an essential role, not only on the outcome of the regressor, but also on the tracking performance. For instance, by increasing the prior standard deviation from $\sigma_f = 0.1$ to $\sigma_f = 1.0$, the sum of the tracking error $\int |\tilde{z}| \mathrm{d}t$ drops about 97%, in the first three cases: $\sigma_\varepsilon = 1.2$ (from 0.2757 to 0.0102), $\sigma_\varepsilon = 0.8$ (from 0.2531 to 0.0071), and $\sigma_\varepsilon = 0.4$ (from 0.1805 to 0.0049); and 81% with $\sigma_\varepsilon = 0.1$ (from 0.0363 to 0.0067). Moreover, even considering this strong enhancement of the tracking performance, the total control action $\int |u_z| \mathrm{d}t$ is not increased, remaining



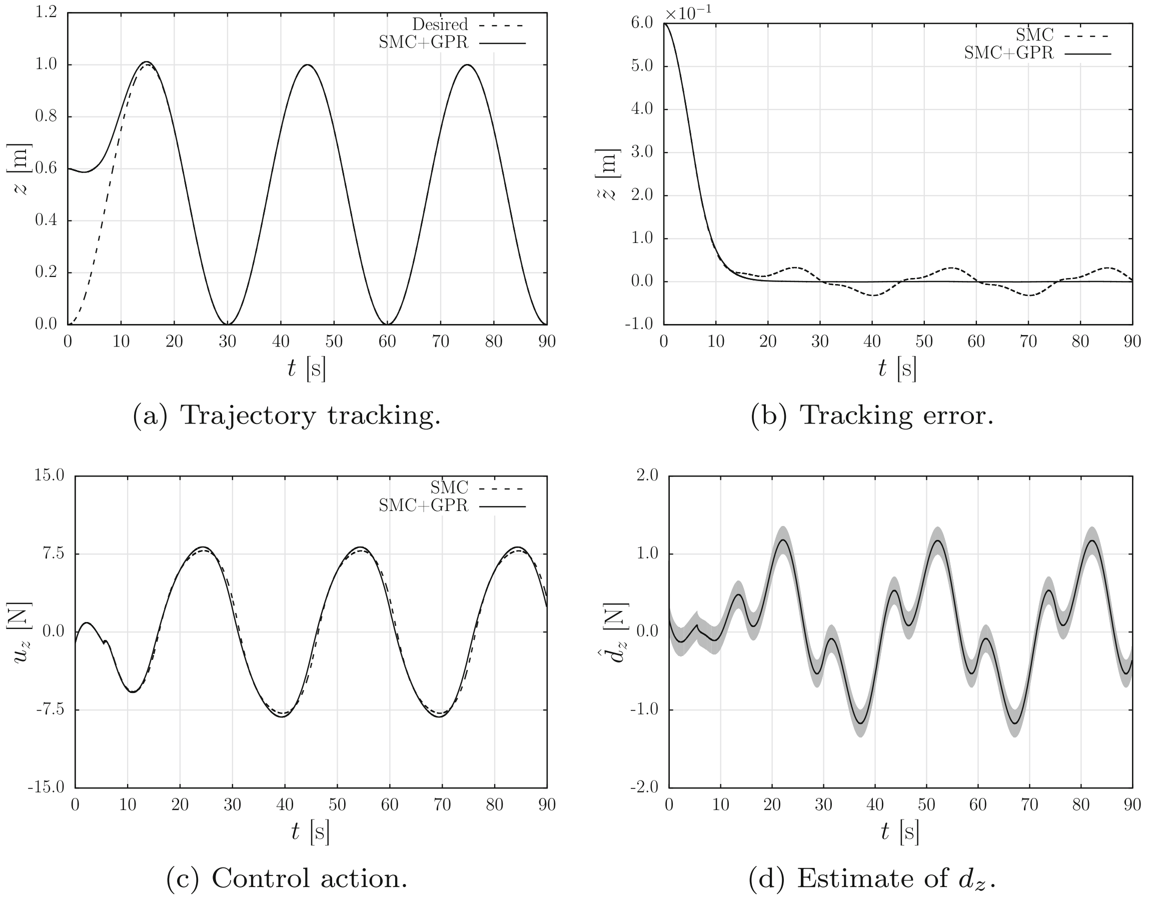
(a) Sim. $2_a$: $\sigma_\varepsilon = 1.2$ and $\sigma_f = 0.1$.

(b) Sim. $2_c$: $\sigma_\varepsilon = 1.2$ and $\sigma_f = 1.0$.

(c) Sim. $4_a$: $\sigma_\varepsilon = 0.4$ and $\sigma_f = 0.1$.

(d) Sim. $4_c$: $\sigma_\varepsilon = 0.4$ and $\sigma_f = 1.0$.

**Fig. 3** Estimate of $d_z$: solid line depicting $\hat{d}_z$ and shaded area representing the confidence interval related to $\pm 2\,\sigma_z$



(a) Trajectory tracking.

(b) Tracking error.

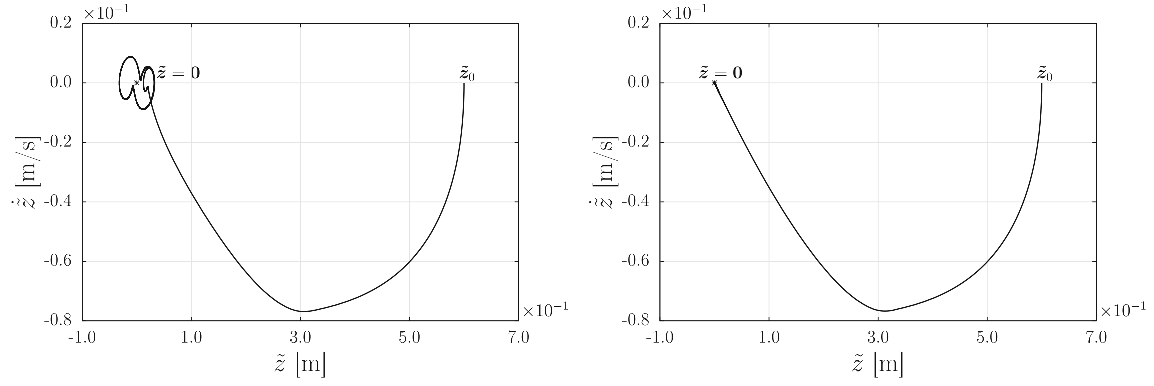
(c) Control action.

(d) Estimate of $d_z$.

**Fig. 4** Depth tracking of $z_\mathrm{d} = 0.5[1 - \cos(\pi t/15)]$ m with $\tilde{z}(0) \neq \mathbf{0}$

almost the same in all four cases. This is mainly due to the gain growth, that is how the proposed controller deals with uncertainties regarding the estimate $\hat{\boldsymbol{d}}$. By setting the control gain according to $\kappa = \eta + \vartheta \, \|\hat{\boldsymbol{M}}^{-1}\|_\infty \|\boldsymbol{\sigma}\|_\infty$, it can be observed that $\kappa$ grows as the GP posterior variance increases. This ensures trajectory tracking even when the regressor is not able to provide a proper estimate $\hat{\boldsymbol{d}}$. However, in this case, with a lower $\hat{\boldsymbol{d}}$ and a higher $\boldsymbol{\sigma}$, the performance of the proposed control scheme becomes similar to that obtained with a conventional sliding mode controller.

It should be noted that in all simulations the size of the sliding window is set to 20 samples, i.e. $r = 20$. Initially, while the window is not fully filled, GPR is computed using



(a) With conventional SMC.

(b) With proposed SMC+GPR.

**Fig. 5** Phase space with the conventional and the proposed schemes

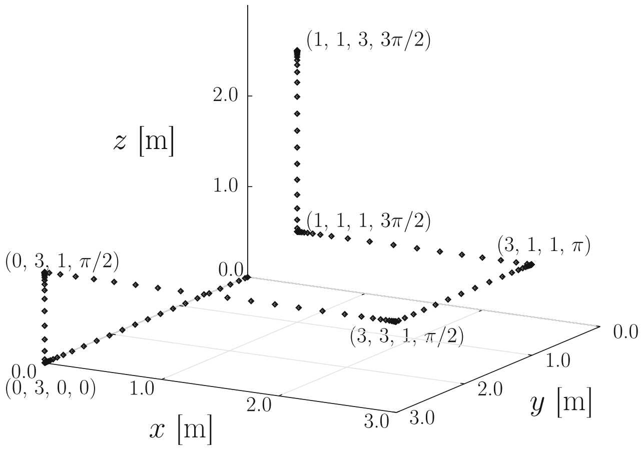


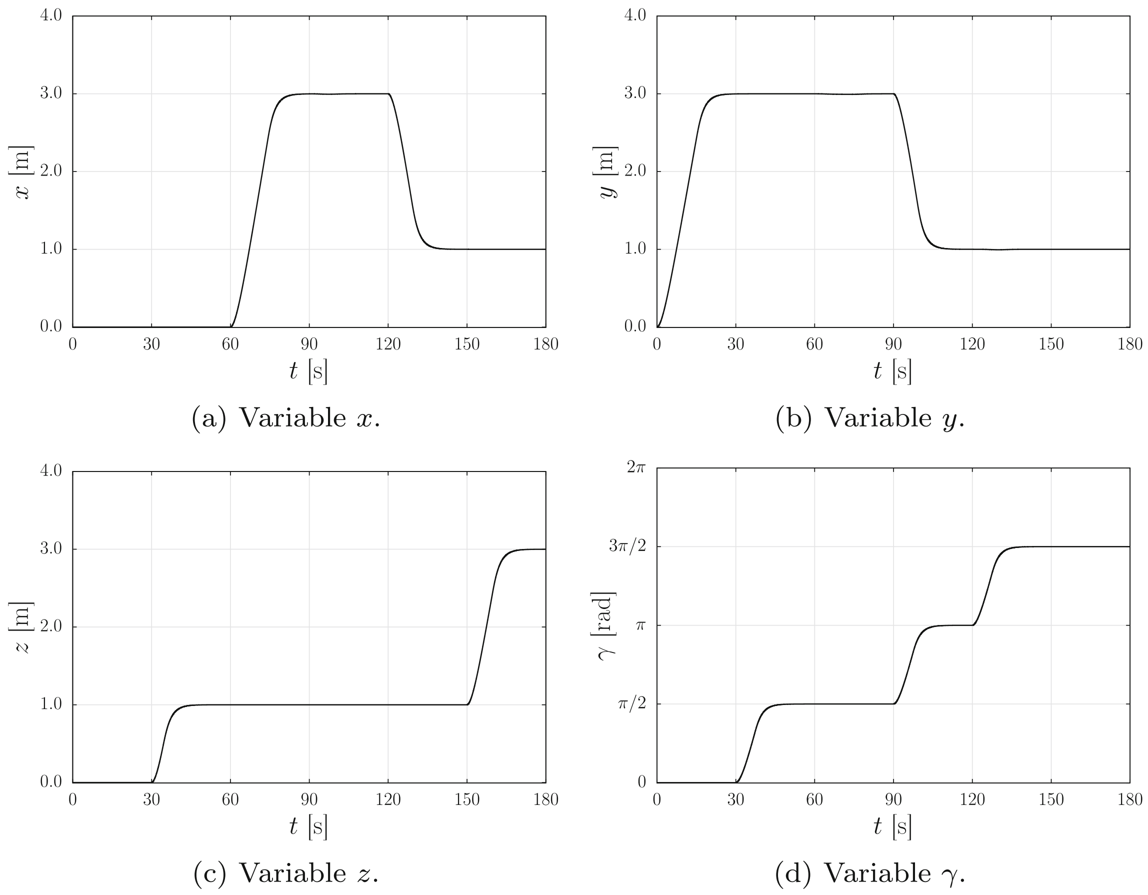
**Fig. 6** Dynamic positioning in $\mathbb{R}^3$

the available data. However, as time progresses and the first 20 samples are gathered, the window starts to slide. Thus, one at a time, the old entries are discarded and the new values are assimilated in their place. The previous window overlaps the next window in the rest of the data.

Now, by keeping the hyperparameters as $\ell = 0.05$, $\sigma_f = 1.0$, and $\sigma_\varepsilon = 0.4$, the performance of the proposed control scheme is evaluated in two new situations.

First, depth tracking of the previous sinusoidal trajectory is assumed, but in this case an initial tracking error is taken into account, $\tilde{z}(0) = [0.0\ 0.6]^\top$. Figures 4 and 5 present the obtained results.

As observed in Fig. 4a, even subject to unmodeled dynamics and actuator saturation, the proposed control scheme is able to track the desired trajectory. Moreover, Fig. 4 shows that the adoption of the Gaussian process regressor within the boundary layer (SMC+GPR) allows a stronger improved performance when compared to the conventional sliding mode controller (SMC). In fact, a much smaller tracking error, Fig. 4b, is obtained without increasing the control effort, Fig. 4c. The phase spaces shown in Fig. 5 highlight the decrease of the tracking error achieved with GPR. It can be noted in Fig. 5a that the conventional sliding mode controller leads to a residual steady-state error around the desired goal, $\tilde{z} = \mathbf{0}$, depicted as $*$ in the plot. The proposed control scheme, on the other hand, is able to provide perfect tracking (see Fig. 5b) despite the presence of modeling inaccuracies. This efficacy is due



(a) Variable $x$.

(b) Variable $y$.

(c) Variable $z$.

(d) Variable $\gamma$.

**Fig. 7** State variables in the time domain for the dynamic positioning in $\mathbb{R}^3$

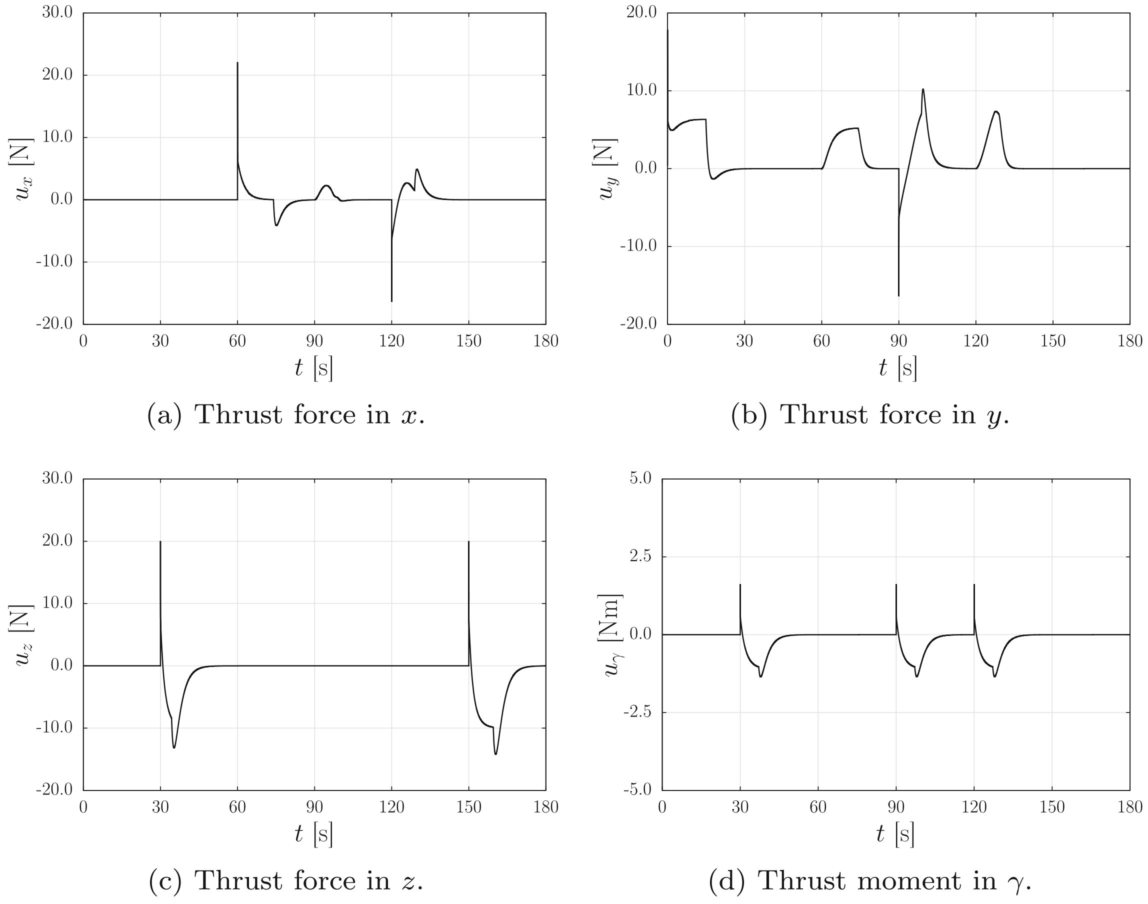
to the ability of the Gaussian process regressor to recognize and previously compensate for unmodeled dynamics.

The second numerical simulation considers the dynamic positioning in $\mathbb{R}^3$. Starting from the initial state $\boldsymbol{x}_0 = [0\,0\,0\,0]^\top$ at rest, the robot moves itself every 30 seconds to a different position/attitude, according to $\boldsymbol{x}_1 = [0\ 3\ 0\ 0]^\top$, $\boldsymbol{x}_2 = [0\ 3\ 1\ \pi/2]^\top$, $\boldsymbol{x}_3 = [3\ 3\ 1\ \pi/2]^\top$, $\boldsymbol{x}_4 = [3\ 1\ 1\ \pi]^\top$, $\boldsymbol{x}_5 = [1\ 1\ 1\ 3\pi/2]^\top$, and $\boldsymbol{x}_6 = [1\ 1\ 3\ 3\pi/2]^\top$. Once a desired state has been reached, the robot must maintain its position/attitude until the next set-point change. The desired attitude $\gamma_\mathrm{d}$ is chosen to keep the robot always pointing in the direction of the horizontal motion. The obtained results are presented in Figs. 6–8.

By observing Fig. 6, it can be verified that the proposed controller successfully drives the robot to the desired states. In addition, Fig. 7 shows that the underwater robot is able to keep its position/attitude as well, even considering actuators' saturation and the fact that hydrodynamic effects are completely unknown during the controller design phase. According to Fig. 8, it can be noted that the thrust forces are assumed to saturate at 25 N. Moreover, Figs. 4c and 8 show that, by embedding GPR within the sliding mode controller, the resulting control approach is able to completely eliminate chattering without compromising the tracking performance.

## 5 Concluding Remarks

The present work addresses the dynamic positioning of underwater robots by combining Gaussian process regression with sliding mode control. Considering that underwater robots are subject to unknown hydrodynamic effects, GPR is embedded within the boundary layer to predict and compensate for these and other possible modeling inaccuracies as well. The boundedness and convergence properties of the closed-loop signals are analytically proven. By means of numerical simulations, the influence of the hyperparameters on both regressor and controller performance is assessed. As expected, a higher noise variance leads to a more scattered estimate, and an increase in the prior variance of the process improves the ability of the GPR algorithm to correlate inputs that lie far apart in the data set. Based on the



(a) Thrust force in $x$.

(b) Thrust force in $y$.

(c) Thrust force in $z$.

(d) Thrust moment in $\gamma$.

**Fig. 8** Control signals in the time domain for the dynamic positioning in $\mathbb{R}^3$

obtained results, it should be emphasized that Gaussian process regression proved to be quite effective as a nonparametric model to compensate for unmodeled dynamics. In fact, when compared with the conventional sliding mode controller, the proposed scheme (SMC+GPR) is able to significantly increase the tracking efficiency, by reducing not only the overall tracking error, but also the required control effort. It can be also verified that the proposed control approach is able to provide accurate trajectory tracking without the undesired chattering effects. The adoption of the predictive variance to estimate the bounds of the disturbance term provides an appealing approach to conveniently tune the gain of the sliding mode controller. Moreover, considering that the method of overlapping sliding windows has been adopted, the computational cost associated with matrix inversion can be conveniently set to the available hardware. By keeping computational complexity low makes the proposed control approach also suitable for embedded devices. Investigating the method's applicability on hardware (including, e.g., appropriate state estimation) and thus evaluating its performance in practice is an important aspect for future work.

**Acknowledgments** This work was supported by the Alexander von Humboldt Foundation [3.2-BRA/1159879 STPCAPES], the Brazilian Coordination for the Improvement of Higher Education Personnel [BEX 8136/14-9], and the Brazilian National Council for Scientific and Technological Development [308429/2017-6].

## References

1. Ludvigsen, M., Sørensen, A.J.: Towards integrated autonomous underwater operations for ocean mapping and monitoring. Annu. Rev. Control. **42**, 145–157 (2016)
2. Mindell, D.A.: Our Robots, Ourselves: Robotics and the Myths of Autonomy. Viking, New York (2015)
3. Teague, J., Allen, M.J., Scott, T.S.: The potential of low-cost ROV for use in deep-sea mineral, ore prospecting and monitoring. Ocean. Eng. **147**, 333–339 (2018)
4. Mitra, A., Panda, J.P., Warrior, H.V.: The effects of free stream turbulence on the hydrodynamic characteristics of an AUV hull form. Ocean. Eng. **174**, 148–158 (2019)
5. Go, G., Ahn, H.T.: Hydrodynamic derivative determination based on CFD and motion simulation for a tow-fish. Appl. Ocean. Res. **82**, 191–209 (2019)
6. Chen, C.-W., Jiang, Y., Huang, H.-C., Ji, D.-X., Sun, G.Q., Yu, Z., Chen, Y.: Computational fluid dynamics study of the motion stability of an autonomous underwater helicopter. Ocean. Eng. **143**, 227–239 (2017)
7. Ramírez-Macías, J.A., Brongers, P., Rúa, S., Vásquez, R.E.: Hydrodynamic modelling for the remotely operated vehicle Visor3 using CFD. IFAC Papers Online. **49**(23), 187–192 (2016)
8. Alam, K., Ray, T., Anavatti, S.G.: Design optimization of an unmanned underwater vehicle using low-and high-fidelity models. IEEE Trans. Syst. Man. Cybern: Syst. **47**(11), 2794–2808 (2015)
9. Bessa, W.M., Dutra, M.S., Kreuzer, E.: Depth control of remotely operated underwater vehicles using an adaptive fuzzy sliding-mode controller. Robot. Auton. Syst. (8) **56**, 670–677 (2008)
10. Bessa, W.M., Dutra, M.S., Kreuzer, E.: An adaptive fuzzy sliding-mode controller for remotely operated underwater vehicles. Robot. Auton. Syst. **58**, 16–26 (2010)
11. Peng, Z., Wang, J., Wang, J.: Constrained control of autonomous underwater vehicles based on command optimization and disturbance estimation. IEEE Trans. Ind. Electron. **66**(5), 3627–3635 (2019)
12. Londhe, P.S., Patre, B.M.: Adaptive fuzzy sliding mode control for robust trajectory tracking control of an autonomous underwater vehicle. Intell. Serv. Robot. **12**(1), 87–102 (2019)
13. Ingrosso, R., Palma, D., Indiveri, G., Avanzini, G.: Preliminary results of a dynamic modelling approach for underwater multi-hull vehicles. IFAC Papers Online. **51**(29), 86–91 (2018)
14. Ghavidel, H.F., Kalat, A.A.: Robust control for MIMO hybrid dynamical system of underwater vehicles by composite adaptive fuzzy estimation of uncertainties. Nonlinear Dyn. **89**(4), 2347–2365 (2017)
15. Bessa, W.M., Dutra, M.S., Kreuzer, E.: Dynamic Positioning of Underwater Robotic Vehicles with Thruster Dynamics Compensation. Int. J. Adv. Robot. Syst (9)10 (2013)
16. Bessa, W.M., Kreuzer, E., Lange, J., Pick, M.A., Solowjow, E.: Design and adaptive depth control of a micro diving agent. IEEE Robot. Aut. Lett. **2**(4), 1871–1877 (2017)
17. Bessa, W.M., Brinkmann, G., Duecker, D.-A., Kreuzer, E., Solowjow, E.: A biologically inspired framework for the intelligent control of mechatronic systems and its application to a micro diving agent. Math. Probl. Eng. 2018 Article ID 9648126 (2018)
18. Rasmussen, C.E., Williams, C.K.: Gaussian Processes for Machine Learning. MIT Press, London (2006)
19. Marco, A., Henning, P., Bohg, J., Schaal, S., Trimpe, S.: Automatic LQR tuning based on Gaussian process global optimization. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 270–277 (2016)
20. Marco, A., Henning, P., Schaal, S., Trimpe, S.: On the design of LQR kernels for efficient controller learning. In: Proceedings of the IEEE 56th Annual Conference on Decision and Control, pp. 5193–5200 (2017)
21. Neumann-Brosing, M., Marco, A., Schwarzmann, D., Trimpe, S.: Data-efficient auto-tuning with Bayesian optimization: An industrial control study. IEEE Trans. Control. Syst Technol (2019)
22. Cho, K., Oh, S.: Learning-based model predictive control under signal temporal logic specifications. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 7322–7329 (2018)
23. Klenske, E.D., Zeilinger, M.N., Schölkopf, B., Hennig, P.: Gaussian Process-Based Predictive Control for Periodic Error Correction. IEEE Trans. Control Syst. Technol. **24**(1), 110–121 (2016)
24. Kocijan, J., Murray-Smith, R., Rasmussen, C.E., Girard, A.: Gaussian process model based predictive control. In: Proceedings of the American Control Conference, pp. 2214–2219 (2004)
25. Liu, M., Chowdhary, G., Castro da Silva, B., Liu, S., How, J.P.: Gaussian processes for learning and control: A tutorial with examples. IEEE Control Syst. Mag. **38**(5), 53–86 (2018)
26. Joshi, G., Chowdhary, G.: Adaptive control using gaussian-process with model reference generative network. In: Proceedings of the IEEE Conference on Decision and Control, pp. 237–243 (2018)
27. Healey, A.J., Lienard, D.: Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles. IEEE J. Ocean. Eng. **18**(3), 327–339 (1993)
28. Christi, R., Papoulias, F.A., Healey, A.J.: Adaptive sliding mode control of autonomous underwater vehicles in dive plane. IEEE J. Ocean. Eng. **15**(3), 152–160 (1990)

29. Yoerger, D.R., Slotine, J.J.E.: Robust trajectory control of underwater vehicles. IEEE J. Ocean. Eng. **10**(4), 462–470 (1985)
30. Bessa, W.M.: Some remarks on the boundedness and convergence properties of smooth sliding mode controllers. Int. J. Autom. Comp. **2**(6), 154–158 (2009)
31. Aran, V., Unel, M.: Gaussian process regression feedforward controller for diesel engine airpath. Int. J. Automot. Technol. **4**(19), 635–642 (2018)
32. Lima, G.S., Bessa, W.M., Trimpe, S.: Depth control of underwater robots using sliding modes and gaussian process regression. IEEE LARS (2018)
33. Hsu, L., Costa, R.R., Lizarralde, F., Cunha, J.P.V.S.: Dynamic positioning of remotely operated underwater vehicles. IEEE Robot. Autom. Mag. **7**(3), 21–31 (2000)
34. Zanoli, S.M., Conte, G.: Remotely operated vehicle depth control. Control. Eng. Pract. **11**, 453–459 (2003)
35. Guo, J., Chiu, F.C., Huang, C.C.: Design of a sliding mode fuzzy controller for the guidance and control of an autonomous underwater vehicle. Ocean. Eng. **30**, 2137–2155 (2003)
36. Newman, J.N. Marine Hydrodynamics, 5th edn. MIT Press, Massachusetts (1986)
37. Slotine, J.J.E., Li, W.: Applied Nonlinear Control. Prentice Hall, New Jersey (1991)
38. Khalil, H.K. Nonlinear Systems, 3rd edn. Prentice Hall, New Jersey (2001)

**Gabriel S. Lima** received the B.Sc. degree in science and technology, in 2015, the B.Sc. degree in mechanical engineering, in 2017, and the master's degree in mechanical engineering, in 2019, from the Federal University of Rio Grande do Norte, Natal, Brazil. He is currently a graduate researcher at the Federal University of Rio Grande do Norte and his research interests include intelligent control, nonlinear control, artificial neural networks, Gaussian process, and reinforcement learning.

**Sebastian Trimpe** received the B.Sc. degree in general engineering science and the M.Sc. degree (Dipl.-Ing.) in electrical engineering from Hamburg University of Technology, Hamburg, Germany, in 2005 and 2007, respectively, and the Ph.D. degree (Dr. sc.) in mechanical engineering from ETH Zurich, Zurich, Switzerland, in 2013. He is currently a Max Planck Research Group Leader at the Max Planck Institute for Intelligent Systems, Stuttgart, Germany, where he leads the independent Max Planck Research Group on Intelligent Control Systems. His main research interests are in systems and control theory, machine learning, networked and autonomous systems. Dr. Trimpe is recipient of several awards, among others, the triennial IFAC World Congress Interactive Paper Prize (2011), the Klaus Tschira Award for achievements in public understanding of science (2014), the Best Demo Award of the International Conference on Information Processing in Sensor Networks (2019), and the Best Paper Award of the International Conference on Cyber-Physical Systems (2019).

**Wallace M. Bessa** is Associate Professor in the Department of Mechanical Engineering at the Federal University of Rio Grande do Norte and Research Fellow of the Brazilian National Council for Scientific and Technological Development (CNPq). He also serves as Associate Editor of Journal of the Brazilian Society of Mechanical Sciences and Engineering (Springer) and as Member of the Committee for Dynamics of the Brazilian Society of Mechanical Sciences and Engineering (ABCM). He graduated in mechanical engineering from the State University of Rio de Janeiro, Brazil, in 1997, earned the master's degree in mechanical engineering from the Brazilian Military Institute of Engineering, Rio de Janeiro, Brazil, in 2000, and obtained the doctoral degree in mechanical engineering from the Federal University of Rio de Janeiro, Brazil, in 2005. From 2015 to 2017, he was a CAPES-Humboldt Research Fellow at the Institute of Mechanics and Ocean Engineering of the Hamburg University of Technology. His research interests include control theory, computational intelligence, robotics, and nonlinear dynamics.