Experimental validation of constraint mitigation algorithm in underwater robot depth control.

Juan Luis Rosendo\textsuperscript{1,2}, Benoit Clement\textsuperscript{2} and Fabricio Garelli\textsuperscript{1}

Abstract
This work explores both modeling and control of the experimental Ciscrea autonomous underwater vehicle (AUV). A six degree-of-freedom model is presented and validated for turn and emerge/sink maneuvers. Then, a constraint compensating algorithm is proposed based on quasi sliding mode conditioning ideas, and added to a pre-existing inaccessible PD controller in order to improve the overall closed-loop response. By considering actuator constraints, the employed technique allows path following at greater speed than the original controller for a given error tolerance. Experimental results on the so-called Ciscrea underwater robot are presented.

Keywords
Modeling, AUV, sliding mode control, actuator constraints

Introduction
The study of the marine environment and commercial activities offshore usually has high cost due to the infrastructure necessary, equipment and skilled personnel. The recent campaigns in the Arctic and Antarctica\textsuperscript{1}, the study of seabed\textsuperscript{2}, the research in algal blooms and the analysis of the stock in fisheries surveys\textsuperscript{3}, applications in the oil and gas sector\textsuperscript{4}, among others, prove that the oceans can be successfully explored with robotic probes. The use of autonomous robots for these activities, especially AUV, has powered this kind of research.

During these activities, AUVs are exposed to an unknown environment where tasks like infrastructure inspection, patrolling areas, or carrying element are the most common. These duties share a common goal: to follow a pre-established path, as fast as possible and with the minimum possible error.

As can be seen, a performance trade-off arises. Indeed, if a sharp path is asked by the guidance function of the robot or if a very quick answer is asked for the path completion, the actuators will reach the maximum allowed allocation. The saturation phenomenon will occur and either error or speed performance metrics will degrade (when saturation occurs, the system behavior is like open loop behavior). Thus, physical constraints should be taken into account as long as demanding control objectives are required. Bibliography presents several cases of study about the saturation problem in autonomous systems\textsuperscript{5}.

In particular for marine systems, several related works can be found in the literature and we present here some of them. In the works by Campos et al.\textsuperscript{6} a PD nonlinear control based on saturation functions with varying parameter for depth and yaw set point regulation and trajectory tracking on an underwater vehicle is proposed. Zheng et al.\textsuperscript{7} deal with asymmetric saturation over the actuator of a marine vessel. In their work, a Gaussian error function-based continuous differentiable asymmetric model is employed, for the design of the base control with backstepping technique. Another attempt to AUV application is presented by Steenson et al.\textsuperscript{8} where the saturation of the actuator was considered directly in the controller tuning through a MPC design. Moreover, Sarhadi et al.\textsuperscript{9} take a simpler approach through a model reference adaptive controller with an anti-windup action, which acts over the input signals of an AUV when saturation arises.

These last solutions are valuable and they achieve good results, but in general they require a good model of the

\textsuperscript{1}CONICET and Universidad Nacional de La Plata, Argentina
\textsuperscript{2}Lab-STICC, UMR CNRS 6585, ENSTA Bretagne, France

Corresponding author:
Juan Luis Rosendo, Universidad Nacional de La Plata, C.C.91 (1900), La Plata, Argentina
Email: benoit.clement@ensta-bretagne.fr
system and actuators. In other fields of robotics control techniques based on time-delay estimations were used to overcome these issues, but they do not take into account the constraint problematic over the actuators. In contrast, this work proposes a technique which deals efficiently with the saturation problem in path following even with a poor model for control tuning.

Considering the path following applications, the path to follow is frequently specified as a vector input function which may be parameterized in terms of a motion parameter, as it is proposed in the works of Nenchev and Garelli. Both actuator limits and error tolerances give rise to a tracking speed limit at each point of the path. Traditional control strategies (particularly in commercial robots) have tackled this problem by using a conservative constant tracking speed so that actuators never reach their limits, or they reach them only in some isolated point of the path. A better solution, naturally, requires using variable tracking speed. However, this is in general computationally awkward as the maximum tracking speed compatible with the error tolerance must be computed on-line as the robot advances through the path.

In this work, a simple algorithm for variable speed tracking computation considering actuator limits is addressed and experimentally tested. This is the main contribution of this article, which a preliminary version has already been presented but here a greater detail is provided in the AUV model, experimental results and in the particular mathematical conditions for the proposed algorithm. This is based on quasi sliding mode ideas originally proposed by Garelli et al., for a general cinematic control structure, which was never experimentally tested before.

For illustrative purposes, the method is evaluated in the experimental AUV Ciscrea robot under a closed-controller restriction, i.e. the original AUV controller is fixed and inaccessible due to software and safety constraints. The Ciscrea robot is shown in Fig. 1, and its main characteristics can be seen in Table 1. Due to its difficult hydrodynamic modeling and identification, this robot has been used to test different control laws explained in. This work is structured as follows. Section “Underwater robot model” proposes an AUV control-oriented model and its validation on the measurable variables. Then, section “Quasi sliding-mode conditioning” presents the details of the constraint compensation technique, while section “Experimental application to dynamic AUV control” is devoted to experimental results. Finally, the conclusions of these experiments are given at the end of the paper with future works to involve the results.

Underwater robot model

Model description

In this section a dynamic and kinematic control-oriented model for the AUV under study will be developed following the ideas proposed in the books of Fossen and declined with a Ciscrea AUV that is available at ENSTA Bretagne.

Two coordinate systems are commonly employed for localization, as can be seen in Fig. 1:

- the classical earth frame, called NED-frame due to the main directions that are North East and Down;
- the frame linked to the robot, called B-frame due to the Body fixed reference

All the data are given in the international units: distances are in meters, angles in radians and positive clockwise. The position, velocity and force are denoted $\eta$, $\nu$ and $\tau$. They are defined as follows:

$$\eta = [x, y, z, \phi, \theta, \psi]^T$$ (position)

$$\nu = [u, v, w, p, q, r]^T$$ (velocity)

$$\tau = [X, Y, Z, K, M, N]^T$$ (force and torque)

while the rigid-body dynamics is given by:

$$M_{RB} \ddot{\nu} + C_{RB}(\nu) \nu = \tau_{env} + \tau_{hydro} + \tau_{pro}$$ (2)

and the hydrodynamic formulation (hydrostatics included) is:

$$\tau_{hydro} = -M_A \dot{\nu} - C_A(\nu) \nu - D(|\nu|) \nu - g(\eta)$$ (3)
The corresponding parameters are given in Table 2.

Table 2. Nomenclature of the underwater vehicle model

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_{RB}$</td>
<td>Mass and inertia matrix for the rigid model of the robot</td>
</tr>
<tr>
<td>$M_{A}$</td>
<td>Added mass matrix used for marine vehicles</td>
</tr>
<tr>
<td>$C_{RB}$</td>
<td>Rigid-body matrix induced by Coriolis phenomenon</td>
</tr>
<tr>
<td>$C_{A}$</td>
<td>Added mass matrix induced by Coriolis phenomenon</td>
</tr>
<tr>
<td>$D(\nu)$</td>
<td>Damping matrix due to mechanical frictions</td>
</tr>
<tr>
<td>$g(\eta)$</td>
<td>Restoring forces and moments vector</td>
</tr>
<tr>
<td>$\tau_{env}$</td>
<td>Disturbances from environment (wind, waves and currents, even if when deep the wind end waves can be neglected)</td>
</tr>
<tr>
<td>$\tau_{hydro}$</td>
<td>Vector of the hydrodynamic forces and moments</td>
</tr>
<tr>
<td>$\tau_{pro}$</td>
<td>Propeller forces and moments vector</td>
</tr>
</tbody>
</table>

In the present application, as the Ciscrea vehicle speed remains low, the parameters $C_{RB}$ and $C_{A}$ can be neglected, while the parameters $M_{RB}, M_{A},$ and the damping matrix are obtained from Yang et al. works\textsuperscript{16, 17}.

The vector $g(\eta)$ that represents the forces and the moments produced by the weight and the buoyancy forces applied to the rigid-body is given by the following formulation:

\[
g(\eta) = \begin{bmatrix} - (m - \rho \text{vol})g \sin \theta \\ (m - \rho \text{vol})g \cos \theta \sin \phi \\ (m - \rho \text{vol})g \cos \theta \cos \phi \\ -BG_xmg \cos \theta \cos \phi + BG_ymg \cos \theta \sin \phi \\ -BG_xmg \sin \theta + BG_ymg \cos \theta \sin \phi \\ -BG_xmg \cos \theta \sin \phi - BG_ymg \sin \theta \end{bmatrix}
\]  \hspace{1cm} (4)

where:

- the vector $BG = [BG_x, BG_y, BG_z]^T$ represents the vector from the center of gravity, denoted CG, to the buoyancy center, denoted CB,
- the scalar $\rho$ is the fluid density which can vary according to the sea or lake and temperature environment,
- the scalar $\text{vol}$ is the displaced fluid volume which determine the Archimede force,
- the scalar $g$ is the gravity acceleration module,
- $m$ is the AUV mass.

The convention for the signed angles is the one shown in Fig. 1.

It is important to notice that the center of buoyancy, CB, and the center of gravity, CG, were experimentally adjusted by moving, adding or removing payload and floats. Both are actually close and they can be consider in the same location. Furthermore, due to the symmetric form of the AUV, CB and CG coincide with the geometrical center of the robot.

The marine disturbances are mainly due to wind, waves and current. They contribute to $\tau_{env}$ term. However, as the vehicle is most of the time under the water surface, waves and wind are not strong enough to have a real effect on the robot. Then only marine currents are considered during underwater operations.

Concerning to the hydrodynamics parameters, it is worth mentioning:

- the term $M_{A} \in \mathbb{R}^{6 \times 6}$ is the added mass. This a classical virtual concept used in marine mechanics for representing the hydrodynamic forces and moments. Indeed any object in a fluid will encounter this $M_{A}$ as soon as he will have an acceleration. This is due to the important inertia of the fluid (in the air, the low density make the phenomenon negligible according to the other forces).
- the term $D(\nu) \in \mathbb{R}^{6 \times 6}$ represents the so-called fluid damping. It can be decomposed in four parts:
  - potential damping,
  - wave drift damping,
  - skin friction,
  - vortex shedding damping.

As explained by Yang R.\textsuperscript{16}, the first two could be dismissed in this kind of application, and the two others are given by approximations. In order to be precise enough, a second order approximation has been chosen and then the term $D(\nu)$ could be given by linear and quadratic matrices, $D_{L}$ and $D_{N}$ respectively, as shown in the following equation:

\[
D(\nu) = D_{L} + D_{N} |\nu|
\]  \hspace{1cm} (5)

Knowing the kinematic relation of velocity vector $\nu$ (in B-frame) and the position vector $\eta$ (in NED-frame):

\[
\dot{\eta} = J(\Theta)\nu
\]  \hspace{1cm} (6)

where $J(\Theta) \in \mathbb{R}^{6 \times 6}$ is a transformation matrix between B-frame and NED-frame defined in Equations 7, 8 and 9 with $\Theta = [\phi, \theta, \psi]^T$.

\[
J(\Theta) = \begin{bmatrix} R(\Theta) & 0_{3\times3} \\ 0_{3\times3} & T(\Theta) \end{bmatrix}
\]  \hspace{1cm} (7)
From B-frame model to NED-frame model, the transformation is possible applying $J(\Theta)$ to Eq. 2 and 3 in order to obtain the differential equation which model the robot behavior:

$$M^*\ddot{\eta} + D^*(|\nu|)(\dot{\eta}) + g^*(\eta) = \tau_{pro} + \tau_{env}$$

with the following new notations:

- $M^* = J^{-T}(\Theta) (M_{RB} + M_A) J^{-1}(\Theta)$, equivalent to the mass,
- $D^*(|\nu|) = J^{-T}(\Theta) D(|\nu|) J^{-1}(\Theta)$, equivalent to the damping term,
- $g^*(\eta) = J^{-T}(\Theta) g(\eta)$, equivalent to forces and moments.

According to all of these equations, the control-oriented model for simulation can be represented with the Eq. 11.

$$T_a = \begin{cases} 
4.7 & \text{if } T_a \geq 127 \\
3.2 \max \left( \frac{T_a}{127}, \frac{T_a-30.3781}{67.56} \right) & \text{if } 0 < T_a < 127 \\
-4.3 \max \left( \frac{T_a}{127}, \frac{T_a-30.3781}{67.56} \right) & \text{if } -127 < T_a < 0 \\
-6.32 & \text{if } T_a \leq -127 
\end{cases}$$

**Model validation**

While mechanical parameter values were identified from laboratory measures on the experimental robot and from data provided by the manufacturer, the hydrodynamic parameter values of the model were taken from Yang et al.\textsuperscript{16}.

Two additional effects were considered for a realistic description of the robot:

- A depth sensor delay that was estimated by experiments to be 0.5 s.
- The non-linear relation between the commanded digital torque signal $T_d$ (-127 to 127) to the real torque $T_a$ in each motor (Fig. 2). The conversion function was synthesized using linear regression from measures realized over the robot. The final expression of the conversion can be expressed as Eq. 12 and a graphic representation is made in Fig. 3.

**Figure 2.** Ciscrea’s Thruster

**Figure 3.** Link between the digital control and the applied thrust

For validation of the proposed model, a comparison between simulations and experimental tests in a pool has been performed. The comparison has been done with the logged open loop time responses. In Fig 4 and Fig. 5 it is possible to appreciate the comparison of the yaw direction in both senses between the simulator and angle’s measures in a real maneuvers. In both cases a command torque signal is first sent to the robot’s motors and then the free response of the system is observed. In the same way Fig. 6 and Fig. 7 compare the depth direction for emerge and sink.
Table 3. Thruster's parameters

<table>
<thead>
<tr>
<th>Thruster, i</th>
<th>$x_i$ [m]</th>
<th>$y_i$ [m]</th>
<th>$z_i$ [m]</th>
<th>$\psi_i$ [rad]</th>
<th>$\theta_i$ [rad]</th>
<th>$\phi_i$ [rad]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizontal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.165</td>
<td>0.145</td>
<td>-0.05</td>
<td>-0.5281</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0.165</td>
<td>-0.145</td>
<td>-0.05</td>
<td>0.5281</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>-0.165</td>
<td>0.145</td>
<td>-0.05</td>
<td>3.6697</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>-0.165</td>
<td>-0.145</td>
<td>-0.05</td>
<td>2.6135</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Vertical</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>0</td>
<td>-0.14</td>
<td>-0.05</td>
<td>$\pi/2$</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>0</td>
<td>0.14</td>
<td>-0.05</td>
<td>$\pi/2$</td>
<td>0</td>
</tr>
</tbody>
</table>

To complete the evaluation of the model given, Table 4 presents different error calculus of the presented comparisons. Through the classical analysis of the Root-Mean-Square Error (RMSE), it is possible to show that the model is more accurate in the heave direction than in yaw. Nevertheless, considering the Normalized Mean Absolute Error (NMAE) is possible to see the weight of the errors in consideration in all the cases is smaller than 10%. To conclude, the Bias is obtained, where it can be noted that the model is able to predict the robot dynamics in an acceptable way for a control orientated simulator.

Table 4. Model error calculations

<table>
<thead>
<tr>
<th>Maneuver</th>
<th>RMSE</th>
<th>NMAE</th>
<th>Bias</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turning right</td>
<td>0.2503</td>
<td>0.0247</td>
<td>0.1920</td>
</tr>
<tr>
<td>Turning left</td>
<td>0.1539</td>
<td>0.0802</td>
<td>0.0852</td>
</tr>
<tr>
<td>Sink</td>
<td>0.0805</td>
<td>0.0382</td>
<td>-0.0537</td>
</tr>
<tr>
<td>Emerge</td>
<td>0.0587</td>
<td>0.0238</td>
<td>0.0545</td>
</tr>
</tbody>
</table>

Figure 4. Turning left maneuvering comparison

Quasi sliding-mode conditioning

In this part of the article, the basics of Sliding Mode (SM) control are revisited in order to recall main ideas. A quasi sliding-mode algorithm it then presented to deal with constraints. The objective of this point is to apply this method to path following (applied to the underwater robot presented before).
is possible to impose a robust closed-loop dynamics by applying a discontinuous control action.

In order to sum it up, a switching function is defined and according to the sign, the control signal can take one of two different values defined by a discontinuous control law with an associated manifold on the state-space (also called the sliding surface). This approach allows to enforce the system have a response in two steps:

1. the first action is to reach a specific area of the state space which is the so-called sliding surface
2. once the sliding surface is reached, the objective is to slide on it through a very fast switching control.

Once this particular mode of operation is established, known as sliding mode (SM), the prescribed manifold imposes the new and desired system dynamics.

In order to illustrate this behavior, let consider a nonlinear system which is linear according to the input $u$:

$$\begin{align*}
\dot{x} &= f(x) + g(x)u \\
y &= h(x)
\end{align*}$$

(13)

with $x \in \mathbb{R}^n$ the state vector, $u$ the control action, $y$ the output and $f(x)$, $g(x)$, $h(x)$ vector fields. A discontinuous control law

$$u = \begin{cases} u^- & \text{if } \sigma(x) < 0 \\ u^+ & \text{if } \sigma(x) > 0 \end{cases}$$

(14)

is defined according to the sign of a switching function $\sigma(x)$. The sliding surface $S$ is defined as the manifold where the switching function vanishes. If the switching law in Eq. 14 makes the reaching condition

$$\begin{align*}
\dot{\sigma}(x) < 0 & \text{ if } \sigma(x) > 0 \\
\dot{\sigma}(x) > 0 & \text{ if } \sigma(x) < 0
\end{align*}$$

(15)

hold locally near the surface (on both sides of it). The control is then switching at high frequency in order to constrain the trajectory of the state $x$ to slide on the surface $S$.

A necessary condition for satisfying Eq. 15 is that $\dot{\sigma}(x)$ explicitly depends on $u$, which is known as transversality condition $^{19}$.

**SMRC (Sliding Mode Reference Conditioning)**

The so-called Sliding Mode Reference Conditioning (SMRC) technique has been developed by one of the authors of this work to cope with general constrained control systems, and it takes advantage from the high frequency switching of sliding regimes $^{20, 21}$.
Differing from conventional SM, SMRC typically acts on one side of the surface, but it does nothing on the other side. So, it can be seen as a one-way SM. Also, as SMRC only becomes active when a constraint is reached (or about to be reached), but it turns off when that risk is over, it usually gives rise to transient quasi-SM operation on the limit surfaces (differing from conventional SM, in which after a reaching mode the desired operation is on the sliding surface).

Let us consider a constrained subsystem of the closed-loop composed by the system in Eq. 13 and a given controller, with the following state-space description:

\[
S_c : \begin{cases} \dot{x}_s = f(x_s) + g(x_s)u \\ v = h_v(x_s) \end{cases} \tag{16}
\]

where the vector \(x_s\) is the state vector and the vector \(v\) the constrained variable (which might represent the plant input, an internal state or even the controlled variable). In order to specify the bounds on the variable \(v\), the set \(\Sigma(x_s)\) is defined as follows:

\[
\Sigma(x_s) = \{x_s | \sigma(v) \leq 0\} \tag{17}
\]

We aim at generating a control input \(u\) which make the system remain within \(\Sigma\). So, the right part of the first equation in Eq. 16 must be oriented to the interior of \(\Sigma\) at all points on the border \(\partial \Sigma = \{x_s | x_s \wedge \sigma(v) = 0\}\), which is achieved if \(12\):

\[
u = \begin{cases} \leq u^\sigma : x_s \in \partial \Sigma \land L_g \sigma > 0 \\ \geq u^\sigma : x_s \in \partial \Sigma \land L_g \sigma < 0 \\ \exists : x_s \in \partial \Sigma \land L_g \sigma = 0 \\ \text{free : } x_s \in \Sigma \setminus \partial \Sigma \end{cases} \tag{18}
\]

with \(u^\sigma\) a scalar magnitude defined as:

\[
u^\sigma = -L_f \sigma / L_g \sigma \tag{19}
\]

The generic operator \(L_fh(x) : \mathbb{R}^n \rightarrow \mathbb{R}\) denotes the directional or Lie derivative with the classical definition:

\[
L_fh(x) = \frac{\partial h}{\partial x} f(x). \tag{20}
\]

Note that the control action \(u^\sigma\) is required to keep the system just in the neighbourhood of the invariant region border, \(L_g \sigma \neq 0\) is the necessary transversality condition for SM to exist, and that \(L_f \sigma > 0\) was assumed without loss of generality.

From Eq. 18 it can also be seen that \(u\) might be freely chosen inside the region \(\Sigma\). Taking \(u = 0\), which does not affect the original control system when constraints are not reached, we can thus make \(\Sigma\) invariant by implementing an auxiliary loop like the one of Fig. 8.

![Block diagram of SMRC technique.](image)

In this scheme, an augmented constrained system \(S'_c\) is considered which is composed by the system \(S_c\) and a filter \(F(s)\), while the signal \(w\) is now the discontinuous signal generated by the SMRC. Additionally, a generic additive disturbance \(d\) at the input of the constrained system can be considered, such as \(u = rf + d\). Assuming \(S_c\) is a biproper dynamical system (i.e. with relative degree equal zero), the following switching law is there implemented:

\[
w = \begin{cases} w^- & \text{if } \sigma < 0 \\ w^+ & \text{if } \sigma > 0 \\ 0 & \text{if } \sigma = 0 \end{cases} \tag{21}
\]

with the trivial switching function

\[
\sigma(v) = v - \bar{v} \tag{22}
\]

and where \(\bar{v}\) represents both the upper (\(\bar{v}\)) and lower (\(\bar{v}\)) limit of \(v\). From Eq. 22 either the upper \(\bar{v}\) or lower \(\bar{v}\) switching function may result (see Fig. 8).

The first order filter is used to smooth out the conditioning signal \(w\):

\[
F(s) : \begin{cases} \dot{x}_f = \lambda_f x_f + w + r \\ r_f = -\lambda_f x_f \end{cases} \tag{23}
\]

with \(r\) being the original reference of the control system. The filter bandwidth should be higher than the one of the constrained system for the system response not to be unnecessarily degraded during unconstrained operation.

Finally, it is worth remarking that if the constrained subsystem \(S_c\) was not biproper, the SMRC could always be applied by considering additional system states \(x_s\) in \(\sigma\), so that the transversality condition (with respect to \(w\)) holds. This case is illustrated in Fig. 8 with the arrow labeled \(x_s\) pointing to the block \(\sigma\).
Experimental SMRC application to dynamic AUV control

In this section an analysis of the SMRC application to auto-regulate the AUV Ciscrea speed reference is presented, along with the corresponding simulations and its experimental validation under inaccessible main controller restriction.

AUV reference conditioning

Here, the SMRC is not used exactly as presented in the theoretical approach. Indeed it has been modified in order to better fit with the path following issue. The modification consist in adapting the speed of the path reference for the AUV Ciscrea when a given controller reaches its propeller actuator limits. For this, we start from the premise that the path is parameterizable and continuous, it means that it is possible to express the path reference \( \eta_{ref} \) and its first derivate as:

\[
\eta_{ref} = f(\lambda) \quad \dot{\eta}_{ref} = \frac{\partial f}{\partial \lambda} \dot{\lambda}
\]  

(24)

where \( \lambda \) is the parameterization, and \( \dot{\lambda} \) can be considered the speed of the path reference. Taking this into account, it is possible to modify the classic schema of SMRC as it is shown in Fig. 9. In this case, for the sake of clarity, only the heave direction is considered from the model proposed in section “Underwater robot model”.

![Figure 9. SMRC constraint mitigation scheme](image)

Differing from the traditional SMRC, \( w \) is smoothed through a low pass filter “\( F(s) \)" to modify the speed reference parameter \( \dot{\lambda} \) instead of the reference itself, resulting in a motion parameter \( \dot{\lambda} \):

\[
\dot{\lambda} = \lambda_d w_f
\]

(26)

Once the reference \( \eta_{ref} \) is generated through the integration block and the path generation block “\( f(\lambda) \)”, it is compared with the actual robot position \( z \) in order to generate the error signal for the controller block. The latter will generate the control signal for the robot.

In few words, the operation could be resume as follows: when actuators are in their linear region, the SMRC auxiliary loop stays inactive and the reference speed is \( \lambda_d \). When saturation limits are reached, \( w \) switches to zero as fast as necessary to decrease the reference speed and avoid the controller surpassing those limits. Finally, it means that if the condition is over, then the SMRC loop turns into the inactive condition.

The following parameters should be considered for the control tuning. They are listed below and a guide for tuning is given:

- \( \lambda_d \): this parameter is the reference speed during the inactive condition of the auxiliary loop. Differing from more conservative strategies, here it can be chosen large enough to force the saturation of actuators once in the path to be followed.
- \( F(s) \): the filter cut-off frequency can be taken high enough. This kind of choice allows fast stops of the reference, but sufficiently low to smooth the discontinuous signal \( w \), in order not to produce chattering effect on the path reference.
- \( \sigma \): signal must have relative degree equal to one with respect to the discontinuous signal \( w \). This is the necessary condition for the SM establishment. In this case as long as we use a controller with a derivative component this condition is guaranteed (if this were not the case, extra states should be considered in the switching function). Considering a classical PD controller with the form:

\[
v = K_p \dot{e} + K_d \ddot{e}
\]

(27)

it is possible to get an expression of \( \dot{\sigma} \) as:

\[
\dot{\sigma} = \beta(\ddot{e}, \dot{z}, \lambda_f) - K_d \lambda_d \dot{\lambda}_f \frac{\partial f(\lambda)}{\partial \lambda} w
\]

(28)
On the right side of this expression two terms appear, one depending on \( w \) guaranteeing the necessary condition provided \( K_d \) is different from zero, and the other term \( \beta \) is a function of the error derivate \( \dot{e} \), the robot acceleration \( \ddot{z} \), and the bandwidth of the low pass filter \( \lambda_f \). As the sufficient condition for the SM is achieved if the term containing \( w \) can change the sign of \( \dot{\sigma} \) (recall Eq. 15), this means that both \( \dot{e} \) and \( \ddot{z} \) must be bounded for SM to be established, which is always true in practice.

It is worth mentioning that in this approach, differing from the traditional SM developments, the switching signals are restricted to the digital implementation of an auxiliary loop. As a consequence, the proposal can be added to any pre-existing controller, and the commanded signal to the actuators is not a switching one, thus avoiding one of the main drawbacks of traditional SM designs: the chattering phenomenon.

**Simulations**

The simulation objective is to compare the performance achieved by the proposed methodology with a classical pre-designed PD controller considered as a reference. The simulations are performed for the Ciscrea heave direction only but they can be performed in the other axis. The simulation with the PD controller with a constant speed for the path reference is compared to the simulation achieved with the same controller when the quasi-sliding mode variable speed technique explained in the previous section is added.

The chosen path was a sinusoidal reference and one can note that this work is dedicated to the spacial result as we follow a path, not a trajectory. Indeed, it is worth remembering here that a trajectory is a path with a velocity profile to be fulfilled.

As the PD controller in the robot cannot be tuned, the speed of the reference is changed as a tuning parameter. The value is chosen such as the actuators are on the border of saturating the actuators, as can be seen in Fig. 10. For comparative purposes, when applying the SMRC approach, \( \lambda_d \) is chosen to have a similar bounded position error as for the classic PD controller implementation, see Fig. 11. A set of simulations were run and the results are given in Fig. 10 to 13. The following paragraphs describe the outcomes and show how the SMRC approach improves the performance of the control.

Fig. 12 shows the depth time evolution with (red line) and without (blue line) the SMRC, together with their corresponding path references (green and black lines, respectively). It is possible to appreciate that the time evolution of the reference is not the same for both techniques, but spatially it is the same. Actually, we have a fixed speed reference for the PD control, and a reference with variable speed due to the SMRC loop. The SMRC allows accelerating the reference as long as no saturation over the actuator exists, and when saturation arrives, it is slowed down (see times 19 s to 22 s and 56 s to 59 s). In this way the auxiliary loop exploits better the operating range of the actuators. Furthermore, for the same error tolerances the SMRC constraint mitigation algorithm allows completing the path 10.5 seconds faster, which represents a 12.8% improvement in time. Similarly, it could improve path error if the same path following time were set for both cases.

Fig. 13, displays the remaining signals of the SMRC loop. Between 0 seconds and 19 seconds, no saturation phenomenon occurs, so the SMRC mitigation algorithm is turned-off and the the path speed is fixed at \( \lambda_d \). From time 19 s to 22 s the robot enters a closer path section where the speed imposed by \( \lambda_d \) can not be followed. Then, the SMRC makes \( w \) switch to slow down the reference so that the controller does not exceed the saturation limits. This can be verified in Fig. 12 where a *bump* in the SMRC reference can be seen. Also, note in Fig. 10 that the path reference speed generated by the SMRC loop is the maximum one that avoids exceeding the saturation limits. Finally, the SMRC becomes inactive again until time 56 s, when a similar speed adaption happens. The tuning of parameters involved in these simulations are listed in Table 5.

From the partial results of simulations, an improvement in travel time with SMRC is noticed. This result may mask the main advantage of the proposed method. For this, the Table 6 has been drawn up to highlight the benefits of the method. This table compares travel times and errors involved in several simulations. The first two columns (PD&SMRC, PD) show the results of the previously explained simulations. Again, from Fig. 12 it is noticed that the two reference “in time” are not the same, as the PD reference speed is constant and as fast as possible to avoid open-loop operation due to actuator saturation. This could conduct to think what would happen if we choose a reference speed which speeds up the fixed controller PD. This is done in the simulation which results on the data of the third column in the table (PD \( \lambda_d \)), where it is possible to appreciate that in these conditions a greater error is found apart from open-loop operation (time during which the controller output exceeds the actuator limit). Despite not possible on the real robot, the next step in this logic of reasoning would be...
Table 5. Simulation parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed of reference PD technique</td>
<td>$\lambda_d = 0.15$</td>
</tr>
<tr>
<td>Maximum speed reference SMRC technique</td>
<td>$\lambda_{dm} = 0.175$</td>
</tr>
<tr>
<td>Cutoff frequency of the low pass filter</td>
<td>$f_c = 2$ Hz</td>
</tr>
<tr>
<td>Sample time</td>
<td>$\Delta_t = 0.1$ s</td>
</tr>
<tr>
<td>Controller</td>
<td>$K_p = 541.43$, $K_d = 250$</td>
</tr>
</tbody>
</table>

Table 6. Comparative simulations

<table>
<thead>
<tr>
<th></th>
<th>PD&amp;SMRC</th>
<th>PD</th>
<th>PD $\lambda_{dt}$</th>
<th>PD+ $\lambda_{dt}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Travel time [s]</td>
<td>74.3</td>
<td>84.8</td>
<td>74.3</td>
<td>74.3</td>
</tr>
<tr>
<td>Maximum absolute error [m]</td>
<td>0.23</td>
<td>0.23</td>
<td>0.30</td>
<td>0.24</td>
</tr>
<tr>
<td>RMSE [m]</td>
<td>0.12</td>
<td>0.11</td>
<td>0.15</td>
<td>0.12</td>
</tr>
<tr>
<td>Percentage of time in open-loop due to saturation</td>
<td>0%</td>
<td>0%</td>
<td>11.64%</td>
<td>9.65%</td>
</tr>
</tbody>
</table>

It is important to recall that for the experimental results we have a closed controller restriction, i.e. that the original PD controller is neither changeable nor tunable due to safety reasons. In any case, the SMRC algorithm would also work with any other biproper controller, and even greater improvements can be expected with PID-like controllers for which windup problems are typical in presence of input saturation.
Table 7. Experimental parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed of reference for PD technique</td>
<td>$\lambda_d = 0.15$</td>
</tr>
<tr>
<td>Maximum speed reference for SMRC technique</td>
<td>$\lambda_{d_{sm}} = 0.2$</td>
</tr>
<tr>
<td>low pass filter (Cutoff frequency)</td>
<td>$f_c = 0.24$ Hz</td>
</tr>
<tr>
<td>Sample time</td>
<td>$\Delta t = 0.1$ s</td>
</tr>
<tr>
<td>Controller</td>
<td>$K_p = 541.43$ and $K_d = 250$</td>
</tr>
</tbody>
</table>

Experiments

To complete the study, experiments in a pool have been performed and this section presents the results of the experimental test. For this case we have similar results with slight differences due to the real dynamics of the robot. The tuning of the parameters involved are listed in Table 7. It is meaningful to note the difference in the cutoff frequency and $\lambda_d$ parameter with respect to the simulation case. This is mainly to compensate for uncertainty and noise present in the real robot, which may affect the SM existence condition (see Eq. 28). The experimental setup used to perform these experiments is shown in Fig. 14.

Figures 15, 16, 17, and 18 show similar results to the previous section (the same colour lines are used). The main difference that can be noticed is that in this case the sliding regime is longer than in the simulation, as it is now established between times $20$ s to $35$ s and $58$ s to $72$ s. This is attributable in part to the lower cutoff frequency in the low-pass filter and the high frequency components neglected in the robot modeling.

The experimental results show that the SMRC technique fulfills the torque constraints (Fig. 18) mitigating their effects on closed-loop performance. Indeed, given the same error tolerance (Fig. 16) the path is effectively completed in a shorter time when the SMRC path reference adaption is added to the original controller (Fig. 15).

Conclusion

In this work, the main SMRC algorithms were firstly applied to a dynamic AUV model in order to show that it can significantly improve the behavior of the robot for a task of path following. It is then implemented with commercial robot to compensate for actuator saturation under closed controller configurations. This implementation is make possible by the fact that the SMRC algorithms are pretty simple and can be executed in a real time environment.

From both simulation and experimental validation it is possible to conclude that SMRC technique applied to the dynamic control of the AUV effectively mitigates the actuator saturation effect on path following task, achieving a performance improvement in the time response. Future research involves dealing not only with physical constraints (as in this case, actuator saturation), but also virtual ones (for example constraints due to energetic criteria) in any point of the system. Finally, experiments into the open sea should be performed to complete the research program.
Figure 17. Experimental SMRC signals

Figure 18. Experimental torque comparison

Acknowledgements

The authors thank the funding by EIFFEL scholarship from Ministre de l’Europe et des Affaires Etrangres de France, CONICET (PIP0837), MINCYT (PICT2394) and UNLP (I216), Argentina.

References


Competitions (ICARSC), 2014


