Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 52-12 (2019) 73–78

Scaling of commanded signals in a cascade control system addressing velocity and acceleration limitations of robotic UAVs

Krzysztof Łakomy [*] Maciej Marcin Michałek [*]
Wojciech Adamski [*]

[*] *Institute of Automation and Robotics, Poznan University of Technology, Piotrowo 3A, 60-965 Poznań, Poland (e-mails: krzysztof.pi.lakomy@doctorate.put.poznan.pl, maciej.michalek@put.poznan.pl, wojciech.adamski@put.poznan.pl)*

Abstract: The problem of commanded velocity scaling in the cascaded control systems designed for a 3-dimensional mobile robots is often neglected, yet it has a substantial influence on the control performance in the transient stage. In this paper, we propose two scaling methods satisfying the velocity and acceleration limitations of the commanded velocities, where the first one is fully analytical and is based on chosen heuristics, while the second method depends on solving the formulated optimization problem. Introduced methods are verified and compared in the simulation study, followed by the experimental validation on the laboratory station equipped with an autonomous airship.

Keywords: velocity scaling, velocity and acceleration limitations, cascade control, robotics

1. INTRODUCTION

In the real-life robotic applications, control signals are practically bounded due to the physical limitations of actuators that have been installed on the hardware. Most of the control algorithms presented in the literature is not equipped with any built-in security procedures preventing the control signal to grow explosively in the transient state of conducted experiments or simulations. Large values of the calculated control signal may be caused by various reasons, such as (i) peaking phenomenon in the high-gain-observer-based controllers, (ii) very fast reference trajectories that do not match physical limitations of control object, (iii) incorrectly tuned controller parameters or (iv) large initial errors transferred to the control signal with the feedback controller. There are numerous papers describing how to deal with peaking phenomenon (e.g. Sussmann (1991)), but the simplest - although effective - method that works well in practice is to keep the controller inactive until the observer will go out of its transient stage (see Huang (2014)). The solution for problems (ii) and (iii) is to design a reference trajectory meeting the physical limitations of the robotic platform and to tune the controller parameters according to the design rules given in the utilized algorithm description, respectively. In the case of a large initial error values, the boundedness of a feedback signal is most commonly achieved by passing the error signal through the bounded function like saturation or hyperbolic tangent function. Although an introduction of such a simple saturation methods works well for linear systems, it may destabilize a dynamic system in the case of a nonlinear control systems. Main contribution of this paper is a proposition of two methods that will handle the problem of large control signal values caused by the reason (iv), and consider not only the velocity-amplitude limitation (see Łakomy (2017)), but also the velocity-rate limitation. Due to the characteristic structure of cascaded control systems (presented e.g. in Michałek (2017) and Chen (2019)), the velocity scaling procedures utilized, e.g., in the trajectory planning (see Biagiotti (2008)) could also be applied in the controller as a signal scaling methods.

The solution presented in this paper is addressed for the torpedo-like mobile robots moving in the 3-dimensional space (e.g. UAVs) with the cascaded control structure performing the trajectory tracking task. A characteristic feature of torpedo-like vehicle (for the definition, see Bechlioulis (2016)) is that there exists a priviliged direction of motion according to the capabilities of force generation by the installed actuator system. The consequence of separating the outer (kinematic-level) and inner (dynamic-level) controllers in the cascade control structure is that the output of the kinematic-level controller, interpreted as the vector of commanded velocities, is usually passed directly to the dynamic-level controller input as a reference. In the case of a large initial errors, the kinematic-level controller usually returns a very large commanded velocity values that may be difficult to follow by the inner controller due to the physical limitations of specific actuators (maximal thrust or not-fast-enough actuator dynamics). To avoid this problem, we introduce the commanded velocity scaling procedures satisfying the amplitude-limitation that guarantee the boundedness of maximal control signal generated by a feedback part of the inner-loop controller, and the rate-limitation preventing the rapid control signal growth that may be impossible to follow. Additional advantage of the the proposed scaling procedures is that it

[⋆] This work was supported by the statutory grant No. 09/93/DSMK/1902.

10.1016/j.ifacol.2019.11.072

will not allow the vehicle to move faster than the initially defined maximal speed in the case of a trajectory that does not meet safety requirements or physical limitations of the vehicle.

In this paper we present the proposition and comparison of two commanded velocity scaling methods. The first one has an analytic form and depends on the preselected heuristics, while the second one is formulated as a quadratic programming optimization problem. In the verification process of the proposed velocity scaling procedures, we have used a cascaded VFO-ADR (*Vector Field Orientation - Active Disturbance Rejection*), trajectory tracking controller proposed by Michałek (2017) with the velocity scaling function applied on the output of the kinematic-level VFO controller.

For the sake of the notational conciseness, trigonometric functions have been written in a shorten form: $\mathrm{s}\alpha \equiv \sin\alpha$, $\mathrm{c}\alpha \equiv \cos\alpha$, $\mathrm{t}\alpha \equiv \tan\alpha$

2. PREREQUISITIES

2.1 Prerequisities

The description of a rigid-body vehicle position and orientation in a 3D space demands an introduction of two coordinate systems - a body-fixed local reference frame $\{B\}$ that has the origin in the center of the vehicle mass C, and the global reference frame $\{G\}$ (see Fig. 1). A configuration vector defined as

$$\boldsymbol{\eta} \triangleq [x\ y\ z\ \phi\ \theta\ \psi]^\top = \begin{bmatrix}\boldsymbol{\eta}_p\\ \boldsymbol{\eta}_o\end{bmatrix} \in \mathbb{R}^4 \times \left(-\frac{\pi}{2}, \frac{\pi}{2}\right) \times \mathbb{R}, \tag{1}$$

corresponds to the location of the body-related frame $\{B\}$ in the global coordinate system $\{G\}$, where $\boldsymbol{\eta}_p = [x\ y\ z]^\top$ refers to the position of the vehicle center of mass, while $\boldsymbol{\eta}_o = [\phi\ \theta\ \psi]^\top$ consist of the values of Euler angles (roll, pitch, and yaw, respectively) that describe a robot attitude. Kinematic equations transforming the velocity vectors of the considered vehicle between introduced coordinate frames are represented by

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}(\boldsymbol{\eta})\boldsymbol{\nu}, \tag{2}$$

where $\boldsymbol{\nu} = [u\ v\ w\ p\ q\ r]^\top$ is the vector of a robotic platform pseudovelocities expressed in the local coordinate frame $\{B\}$, and $\mathbf{J}(\boldsymbol{\eta}) = \mathrm{blkdiag}\{\mathbf{R}(\boldsymbol{\eta}),\ \mathbf{T}(\boldsymbol{\eta})\}$ is a Jacobian matrix that consists of the rotation matrix

$$\mathbf{R}(\boldsymbol{\eta}) = \begin{bmatrix} \mathrm{c}\psi\mathrm{c}\theta & \mathrm{c}\psi\mathrm{s}\phi\mathrm{s}\theta - \mathrm{c}\phi\mathrm{s}\psi & \mathrm{c}\psi\mathrm{c}\phi\mathrm{s}\theta + \mathrm{s}\phi\mathrm{s}\psi \\ \mathrm{s}\psi\mathrm{c}\theta & \mathrm{s}\psi\mathrm{s}\phi\mathrm{s}\theta + \mathrm{c}\phi\mathrm{c}\psi & \mathrm{s}\psi\mathrm{c}\phi\mathrm{s}\theta - \mathrm{s}\phi\mathrm{c}\psi \\ -\mathrm{s}\theta & \mathrm{s}\phi\mathrm{c}\theta & \mathrm{c}\phi\mathrm{c}\theta \end{bmatrix}, \tag{3}$$

and the matrix of angular velocity vector transformation

$$\mathbf{T}(\boldsymbol{\eta}) = \begin{bmatrix} 1 & \mathrm{s}\phi\mathrm{t}\theta & \mathrm{c}\phi\mathrm{t}\theta \\ 0 & \mathrm{c}\phi & -\mathrm{s}\phi \\ 0 & \mathrm{s}\phi/\mathrm{c}\theta & \mathrm{c}\phi/\mathrm{c}\theta \end{bmatrix}. \tag{4}$$

Remark 1. We assume that the configuration vector stays within the domain introduced in (1) guaranteeing that the matrix $\boldsymbol{T}(\boldsymbol{\eta})$ will not reach its singular points ($\theta = \pm\frac{\pi}{2}$) during the vehicle motion.

According to Fossen (1999), the general equations of the robot dynamics in a body-fixed reference frame $\{B\}$ may be represented by

$$\dot{\boldsymbol{\nu}}(t) = \boldsymbol{f}(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}, t) + \boldsymbol{B}\boldsymbol{\tau}(t) \tag{5}$$

where vector $\boldsymbol{f}(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}, t)$ aggregates such phenomena as Coriolis and centripetal accelerations, restoring effects, the influence of gravity, environmental damping, disturbances etc., while $\boldsymbol{B} \in \mathbb{R}^{6\times 6}$ is the input matrix that corresponds to the impact of control signal $\boldsymbol{\tau} = [\tau_u\ \tau_v\ \tau_w\ \tau_p\ \tau_q\ \tau_r]^\top$ on the dynamical system behavior. Due to the existence of environmental damping forces and physical actuator limitations, we know that the robot velocity vector $\dot{\boldsymbol{\eta}}$ is bounded, satisfying the necessary conditions of the ADR controller implementation.

Remark 2. In this paper, we do not impose any assumptions whether the system is fully- or underactuated. In the notation utilized in (5), the information about actuation type is incorporated in a specific form of $\boldsymbol{B}$ matrix (that may be singular in case of underactuated system).

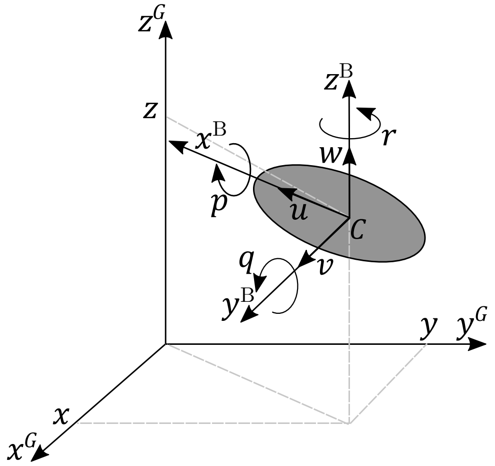

Fig. 1. A representation of a rigid body in the 3D space.

We assume that the priviliged direction of the torpedo-like motion is induced by the application of main propulsion along x^B (surge motion) and around y^B, and z^B axes (referring to the pitch and yaw angle changes). The forces generated along y^B (sway) and z^B (heave) and around x^B (roll) axes have only a corrective function, if they can be generated at all.

In this article we do not focus on any specific controller design method, thus we will use the VFO-ADR trajectory tracking cascaded controller that was initially described in Michałek (2017).

The outer-loop kinematic-level controller is based on the VFO concept, utilized for the torpedo-like vehicles in Michałek (2017) and Łakomy (2017). The output vector returned by the VFO controller, i.e.,

$$\boldsymbol{\nu}_c \triangleq [u_c\ w_c\ v_c\ p_c\ q_c\ r_c]^\top \in \mathbb{R}^6. \tag{6}$$

will be called the vector of commanded velocities. The longitudinal torpedo-like motion should be performed along x^B axis implying

$$w_c \triangleq 0 \text{ and } v_c \triangleq 0. \tag{7}$$

The existance of a priviliged propulsion direction implies the selection of orienting strategy in a way to set the x^B axis orientation tangentially to the reference trajectory. As a result of definitions (7), we can introduce a reduced vector of commanded velocities $\bar{\boldsymbol{\nu}}_c = [\nu_{1c}\ \nu_{2c}\ \nu_{3c}\ \nu_{4c}]^\top \triangleq [u_c\ p_c\ q_c\ r_c]^\top \in \mathbb{R}^4$ to prevent excessively long notation in the latter parts of this article.

The inner-loop controller is designed according to the error-based ADR method presented in Michałek (2016), that was developped upon the original ADR methodology described e.g. in Han (2009). The dynamic-level controller is responsible for calculation of a vector of the desired generalized forces $\boldsymbol{\tau}(t)$, that depends only on the vehicle configuration $\boldsymbol{\eta}$ defined in (1), and the scaled commanded velocity

$$\boldsymbol{\nu}_s \triangleq [u_s \; 0 \; 0 \; p_s \; q_s \; r_s]^\top \in \mathbb{R}^6. \tag{8}$$

The $\boldsymbol{\nu}_s$ vector corresponds to the commanded velocity vector from (6) transformed in order to satisfy the selected limitations described in Section 2.2. Analogously to $\boldsymbol{\nu}_c$, it will be used in the following in its reduced version, i.e., $\bar{\boldsymbol{\nu}}_s = [\nu_{1s} \; \nu_{2s} \; \nu_{3s} \; \nu_{4s}]^\top \triangleq [u_s \; p_s \; q_s \; r_s]^\top \in \mathbb{R}^4$. In order to assure a specific, limited values of the vehicle velocity vector $\boldsymbol{\nu}_s$, the original VFO-ADR control structure is extended with the velocity scaling block, as can be seen in Fig. 2.

2.2 Formulation of the scaling task

The scaled commanded velocity vector $\bar{\boldsymbol{\nu}}_s$, should fulfill the amplitude limitations

$$\forall t \geq 0 \; |\nu_{is}(t)| \leq \nu_{\text{im}}, \; \nu_{\text{im}} > 0, \; i \in \{1, ..., 4\} \tag{9}$$

to prevent large values of the velocity errors in the initial transient stage. Also, to prevent the explosive growth of the scaled commanded velocity components, we introduce the velocity-rate limitations

$$\forall t \geq 0 \; \left| \frac{d\nu_{is}(t)}{dt} \right| \leq a_{\text{im}}, \; a_{\text{im}}, \; i \in \{1, ..., 4\}. \tag{10}$$

The constraints (9) and (10) imposed on the commanded velocity vector are going to be satisfied by applying the appropriate scaling methods described in the following section.

3. SCALING METHODS

Let us assume, that there are no mutual relations between the components of commanded velocity vector. Then according to (9), we can define a hipercuboid subset of the admissible velocities $\mathcal{D}_\nu \triangleq [-\nu_{1m}, \nu_{1m}] \times [-\nu_{2m}, \nu_{2m}] \times [-\nu_{3m}, \nu_{3m}] \times [-\nu_{4m}, \nu_{4m}]$.

By integrating the velocity-rate limitation (10) within some time interval $(t_2 - t_1)$, we obtain the inequality

$$\int_{\nu_{is}(t_1)}^{\nu_{is}(t_2)} |d\nu_{is}| \leq \int_{t_1}^{t_2} a_{\text{im}} dt, \; \forall (t_2 - t_1) > 0 \tag{11}$$

which leads us to the relation

$$\forall t \in [t_1, t_2] \; |\nu_{is}(t_2) - \nu_{is}(t_1)| \leq a_{\text{im}}(t_2 - t_1). \tag{12}$$

From now on, we will be referring to $\tau_s(t) \triangleq (t_2 - t_1)$ as to the delay time interval, and to

$$\tilde{\nu}_{is}(t) \triangleq a_{\text{im}} \tau_s(t) \tag{13}$$

as to the instantenous maximal admissible change of particular commanded velocity. In general the delay time interval can be varying, but in special case (when we deal with the discrete-time control implementation), it will have a constant value equal to the sampling time.

Remark 3. Relation (12) refers to the mean velocity rate limitation within the delay time interval $\tau_s(t)$. In this paper, we assume that the interval $\tau_s(t)$ is much shorter than any dynamical mode of the control plant, so the velocity change between time instants t_2 and t_1 could be well approximated by a constant or linear function. Under these assumptions, inequality (12) can be treated as an instantenous velocity rate limitation.

According to the inequality (12), we can define the time-varying subset $\mathcal{D}_a(t) \triangleq [-\tilde{\nu}_{1s}(t), \tilde{\nu}_{1s}(t)] \times [-\tilde{\nu}_{2s}(t), \tilde{\nu}_{2s}(t)] \times [-\tilde{\nu}_{3s}(t), \tilde{\nu}_{3s}(t)] \times [-\tilde{\nu}_{4s}(t), \tilde{\nu}_{4s}(t)]$ representing the admissible region for the commanded velocity change at time t.

3.1 Velocity scaling method I

A preliminary concept limiting the vector of commanded velocities to the subset $\mathcal{D}_\nu$, i.e. satisfying (9), was proposed in Łakomy (2017). In this paper, we are extending this idea by introducing additional scaling factor that will preserve the velocity-rate limitations introduced in (10).

Let us firstly define a velocity-amplitude scaling factor

$$s_v(t) \triangleq \left[\max \left\{ 1, \frac{|\nu_{1c}|}{\nu_{1m}}, \frac{|\nu_{2c}|}{\nu_{2m}}, \frac{|\nu_{3c}|}{\nu_{3m}}, \frac{|\nu_{4c}|}{\nu_{4m}} \right\} \right]^{-1} \in (0, 1] \tag{14}$$

that scales the commanded velocity vector $\bar{\boldsymbol{\nu}}_c(t)$, guaranteeing preservation of the commanded velocity magnitudes from (9). The resulting prescaled vector of commanded velocities is in the form

$$\begin{aligned} \bar{\boldsymbol{\nu}}_c^*(t) &= [\nu_{1c}^*(t) \; \nu_{2c}^*(t) \; \nu_{3c}^*(t) \; \nu_{4c}^*(t)]^\top \\ &\triangleq s_v(t) \bar{\boldsymbol{\nu}}_c(t) \in \mathcal{D}_\nu. \end{aligned} \tag{15}$$

To satisfy the velocity-rate limitations introduced in (10), we need to define a change of the prescaled commanded velocity vector, i.e.,

$$\Delta \bar{\boldsymbol{\nu}}_c^*(t) \triangleq \bar{\boldsymbol{\nu}}_c^*(t) - \bar{\boldsymbol{\nu}}_s(t - \tau_s) \tag{16}$$

and scale it with the second scaling factor

$$\begin{aligned} s_a(t) \triangleq \Bigg[\max \Bigg\{ & 1, \frac{|\nu_{1c}^*(t) - \nu_{1s}(t - \tau_s)|}{\tilde{\nu}_{1s}}, \\ & \frac{|\nu_{2c}^*(t) - \nu_{2s}(t - \tau_s)|}{\tilde{\nu}_{2s}}, \\ & \frac{|\nu_{3c}^*(t) - \nu_{3s}(t - \tau_s)|}{\tilde{\nu}_{3s}}, \\ & \frac{|\nu_{4c}^*(t) - \nu_{4s}(t - \tau_s)|}{\tilde{\nu}_{4s}} \Bigg\} \Bigg]^{-1} \in (0, 1] \end{aligned} \tag{17}$$

that will bring the scaled commanded velocity change

$$\begin{aligned} \Delta \bar{\boldsymbol{\nu}}_s(t) &\triangleq s_a(t) \Delta \bar{\boldsymbol{\nu}}_c^*(t) = s_a(t) \left[\bar{\boldsymbol{\nu}}_c^*(t) - \bar{\boldsymbol{\nu}}_s(t - \tau_s) \right] \\ &= s_a(t) \left[s_v(t) \bar{\boldsymbol{\nu}}_c(t) - \bar{\boldsymbol{\nu}}_s(t - \tau_s) \right] \end{aligned} \tag{18}$$

to the admissible subset $\mathcal{D}_a$.

According to the scaling heuristics proposed in (15) and (18), the final form of the scaled velocity vector can be rewritten as the convex combination:

$$\begin{aligned} \bar{\boldsymbol{\nu}}_s(t) &= \bar{\boldsymbol{\nu}}_s(t - \tau_s) + \Delta \bar{\boldsymbol{\nu}}_s(t) \\ &\overset{(18)}{=} s_a(t) s_v(t) \bar{\boldsymbol{\nu}}_c(t) + (1 - s_a(t)) \bar{\boldsymbol{\nu}}_s(t - \tau_s) \in \mathcal{D}_\nu. \end{aligned} \tag{19}$$

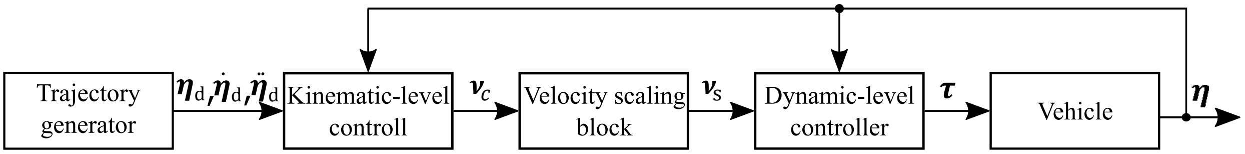

Fig. 2. Cascaded control structure; the kinematic-level controller is based on the VFO algorithm, while the dynamic-level controller is designed upon the ADRC methodology.

In Fig. 3 we can see a graphical interpretation of the proposed method, simplified (due to the presentation limitation) to the 2D case where two velocities $\nu_{3c} = \nu_{4c} = 0$. The first step of velocity scaling, desribed in (15), assure us that the prescaled commanded velocity vector $\bar{\boldsymbol{\nu}}_c^*$ lays within the domain $\mathcal{D}_\nu$. The second step of scaling procedure, represented by (19), approximates the prescaled velocity vector to assure that the velocity change vector $\Delta\bar{\boldsymbol{\nu}}_s \in \mathcal{D}_a$.

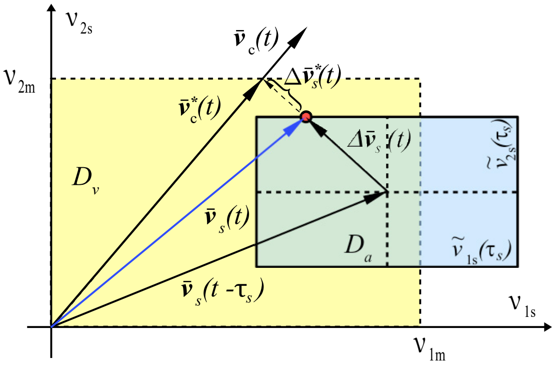
ν_{2s}
$\bar{\boldsymbol{\nu}}_c(t)$
ν_{2m}
$\Delta\bar{\boldsymbol{\nu}}_s^*(t)$
$\bar{\boldsymbol{\nu}}_c^*(t)$
$\Delta\bar{\boldsymbol{\nu}}_s(t)$
$\tilde{\nu}_{2s}(\tau_s)$
D_ν
$\bar{\boldsymbol{\nu}}_s(t)$
D_a
$\tilde{\nu}_{1s}(\tau_s)$
$\bar{\boldsymbol{\nu}}_s(t-\tau_s)$
ν_{1s}
ν_{1m}

Fig. 3. Graphical interpretation of the velocity scaling method I.

3.2 Velocity scaling method II

The first step of the second scaling method is exactly the same as in the first one, i.e., we are obtaining the prescaled commanded velocity vector as in (15) to set the commanded velocity directly within $\mathcal{D}_\nu$. In the case when the difference

$$\Delta\bar{\boldsymbol{\nu}}_s^*(t) \triangleq \bar{\boldsymbol{\nu}}_c^*(t) - \bar{\boldsymbol{\nu}}_s(t-\tau_s) \tag{20}$$

lay outside $\mathcal{D}_a$ (see Fig. 3), we propose to calculate the scaled velocity vector $\bar{\boldsymbol{\nu}}_s(t)$ by solving the quadratic optimization problem

$$\min_{\Delta\bar{\boldsymbol{\nu}}_s^*(t)} \frac{1}{2}\Delta\bar{\boldsymbol{\nu}}_s^{*\top}(t)\Delta\bar{\boldsymbol{\nu}}_s^*(t), \tag{21}$$

subject to the constraints satisfying limitations (9), (10):

$$\begin{bmatrix}1\\-1\end{bmatrix}\bar{\nu}_{is}(t) \leq \begin{bmatrix}\nu_{ic}^*(t-\tau_s)+\tilde{\nu}_{is}(t)\\ -\nu_{ic}^*(t-\tau_s)+\tilde{\nu}_{is}(t)\end{bmatrix} \tag{22}$$

for $i \in \{1,...,4\}$.

Remark 4. The cost function presented in (21) formulated in the proposed way allows us to define the scaling method as a solution to a convex quadratic optimization problem that can be solved relatively quickly. The scaling task could be also achieved using another cost functions, e.g., the one minimizing the angle between $\bar{\boldsymbol{\nu}}_s(t)$ and $\bar{\boldsymbol{\nu}}_c^*(t)$ vectors, but it would make the optimization problem nonlinear, more complex, and presumably longer to solve.

4. SIMULATION EXAMPLE

In the simulation, we have used a model of the 6 degree-of-freedom rigid body described in Section 2. The reference trajectory was defined as follows

$$\boldsymbol{\eta}_d(t) = \begin{bmatrix}\boldsymbol{\eta}_{pd}(t)\\ \boldsymbol{\eta}_{od}(t)\end{bmatrix} = \begin{bmatrix}[x_d(t)\ y_d(t)\ z_d(t)]^\top\\ [\phi_d(t)\ \theta_d(t)\ \psi_d(t)]^\top\end{bmatrix}, \tag{23}$$

where the desired position vector had the specific form of

$$\boldsymbol{\eta}_{pd}(t) = \begin{bmatrix}x_d(t)\\ y_d(t)\\ z_d(t)\end{bmatrix} = \begin{bmatrix}2\cos\frac{\pi}{25}t\\ 4\sin\frac{\pi}{25}t\\ \sin\frac{2\pi}{25}t\end{bmatrix}. \tag{24}$$

We assumed that for the torpedo-like system, the reference velocity along the x^B axis should be nominally tangent to the desired positional trajectory, thus the elements of the desired orientation vector $\boldsymbol{\eta}_{od}(t)$ were computed according to the formulas

$$\phi_d(t) = 0, \tag{25}$$

$$\psi_d(t) = \operatorname{atan2}(\sigma\dot{y}_d(t), \sigma\dot{x}_d(t)), \tag{26}$$

$$\theta_d(t) = \operatorname{atan}\left(\frac{-\dot{z}_d(t)}{\dot{x}_d(t)\mathrm{c}\psi_d(t) + \dot{y}_d(t)\mathrm{s}\psi_d(t)}\right) \tag{27}$$

where $\sigma \in \{-1, 1\}$ determines the forward/backward vehicle motion strategy. The error used to the tracking performance assesment was defined as

$$\boldsymbol{e}(t) = \begin{bmatrix}\boldsymbol{e}_p(t)\\ \boldsymbol{e}_o(t)\end{bmatrix} \triangleq \begin{bmatrix}\boldsymbol{\eta}_{pd}(t)-\boldsymbol{\eta}_p(t)\\ \boldsymbol{\eta}_{od}(t)-\boldsymbol{\eta}_o(t)\end{bmatrix}. \tag{28}$$

Results of the simulation conducted with the scaling I method are presented in Fig. 4. The maximal velocity and acceleration values were set to $\nu_{im} = 1.0$, $a_{im} = 1.0$ (in the appropriate SI units) for $i \in \{1,...,4\}$. The utilized VFO-ADR algorithm guarantees practical stability, implying convergence of the error vector to some neighbourhood of zero. Thus, the ultimate non-zero error values visible on the second plot in Fig. 4 are caused by the controller limitations and not due to the improper definition of a scaling procedure. To prevent the peaking phenomenon of the observer implemented in the ADR dynamic controller, we held the dynamic controller inactive for the first $T_{\mathrm{start}} = 5$s of simulation allowing the observation errors decrease to acceptably small values.

Due to the almost none visual difference in the plotted results obtained with both scaling functions, to present a quantitative comparison between presented methods, we have computed a control quality criteria

$$J_p^{T_1,T_2} = \frac{1}{T_2-T_1}\int_{T_1}^{T_2} \|\boldsymbol{e}_p(t)\|dt, \text{ and} \tag{29}$$

$$J_o^{T_1,T_2} = \frac{1}{T_2-T_1}\int_{T_1}^{T_2} \|\boldsymbol{e}_o(t)\|dt, \tag{30}$$

together with the control cost

$$J_{\tau}^{T_1,T_2} = \frac{1}{T_2 - T_1} \int_{T_1}^{T_2} \|\boldsymbol{\tau}(t)\|^2 dt. \tag{31}$$

The values of particular functionals for the presented methods, and for the simple saturation procedure where $\nu_{is} = \nu_{im}\mathrm{sat}(\nu_{ic}/\nu_{im})$, are presented in Table 1. The difference between methods I and II is marginal, although the second method has a slightly smaller mean error values with slightly greater control cost. The saturation-based procedure, used in the last case, results in the smaller mean control error during the transient stage (the errors converge faster due to the lack of acceleration-level constraints). However, the error values in the steady state are similar to the ones obtained with methods I and II. The greater value of control cost obtained in the simulation utilizing simple saturation is caused by the large values of the desired forces and torques in the transient stage (maximal forces reached ≈ 100N, which corresponds to approximately 10 times the value obtained with methods I and II), that may be impossible to generate by the physical actuators.

Table 1. Values of the performance indexes

-	$J_p^{5,50}$	$J_o^{5,50}$	$J_p^{30,50}$	$J_o^{30,50}$	$J_\tau^{5,50}$
Method I	2.1585	0.2221	0.0002	0.0002	112.2583
Method II	2.1201	0.2192	0.0002	0.0002	116.6499
Saturation	1.6702	0.1043	0.0002	0.0002	171.7289

In Table 2, we can see the average simulation time t_{avg} and the standard deviation σ_s of a single scaling function call compared for the two scaling methods. The results were calculated by running the scaling functions 10 000 times using the previously prepared data samples. The values were obtained on the PC equipped with i7-4720HQ CPU @ 2.60GHz processor and 8 GB DDR3 RAM, while the optimization was done with the *quadprog* Matlab function. Execution speed of the method I significantly exceeds the speed obtained for method II, although the ratio of the average execution times calculated for both methods could probably be reduced by utilizing more efficient optimization tool.

Table 2. Mean values and standard deviations of scaling function calculation time

-	t_{avg} [s]	σ_s [s]
Method I	0.0003	0.0002
Method II	0.0052	0.0024

5. EXPERIMENTAL VERIFICATION

The experimental study was conducted on the laboratory station equipped with an underactuated autonomous airship, unable to generate force along y^B axis. The block diagram of utilized setup is presented in Fig. 5. The reference positional trajectory was defined by

$$\boldsymbol{\eta}_{pd}(t) \triangleq \begin{bmatrix} 0.8 + 0.1t \\ 0.15\tanh(0.15t - 3) - 0.15\tanh(-3) \\ 0.5 - 0.05\sin(\frac{\pi}{40}t - \frac{\pi}{2}) + 0.05\sin(-\frac{\pi}{2}) \end{bmatrix}, \tag{32}$$

while the desired angles were calculated according to (25)-(27). The values of velocity-amplitude and velocity-rate limitations were set to $\nu_{1m} = 0.25$ m/s, $a_{1m} = 0.1$ m/s^2, $\nu_{im} = 0.2$ rad/s and $a_{im} = 0.4$ rad/s^2 for $i \in \{2, 3, 4\}$.

Results of the experiment presented in Fig. 6 were obtained using the velocity scaling method I. During the initial maneuvers, the airship is approaching the desired trajectory and converging to the vicinity of the desired position in a linear manner (see the position and orientation error plot). At the beginning of movement, the airship is accelerating satisfying the velocity-rate limitation (10), what results in a visible ramp at the commanded velocity plot (see the constant value of $\dot{u}_c = a_{1m}$ and the constant offset in the τ_u). After the airship have reached

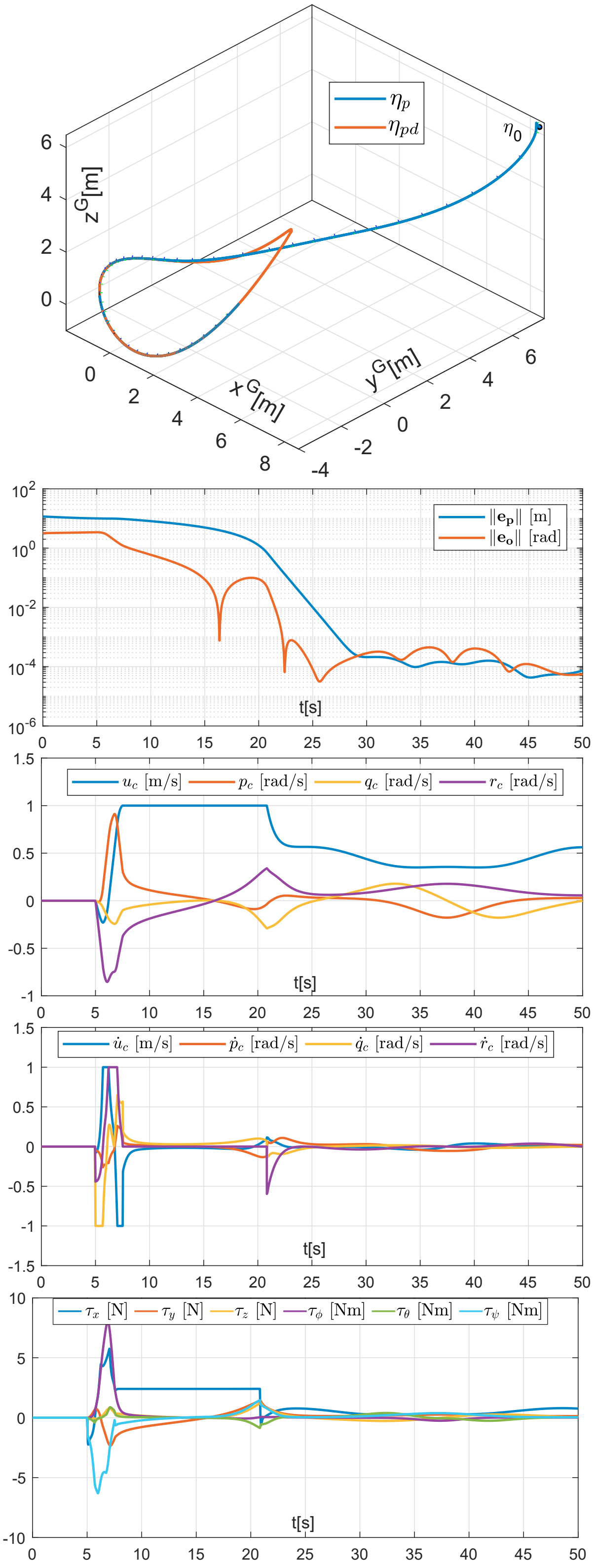

Fig. 4. Simulation results for the scaling method I

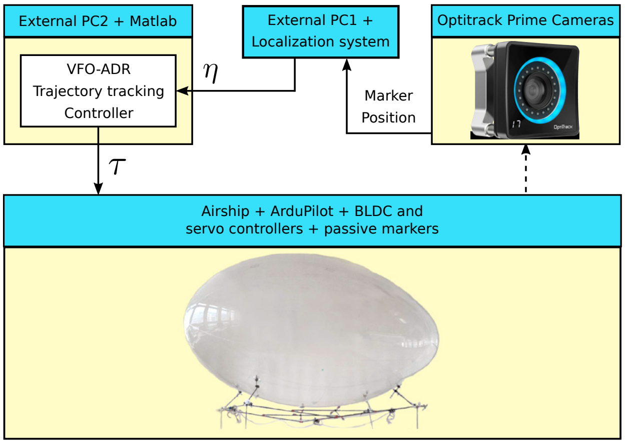
External PC2 + Matlab
VFO-ADR Trajectory tracking Controller
η
External PC1 + Localization system
Optitrack Prime Cameras
Marker Position
τ
Airship + ArduPilot + BLDC and servo controllers + passive markers

Fig. 5. Functional block diagram of the experimental setup.

$u_c = \nu_{1m}$, it keeps the maximal longitudinal commanded velocity until it approaches the close neighbourhood of the desired trajectory. The VFO-ADR algorithm is fragile to the measurement noise due to a high-gain observer used in the ADR part, so the amplified high-frequency noise in the commanded velocity rates was expected. The positive influence of the scaling procedure in this case is that the values of calculated velocity rates were bounded to the values a_{im}, what resulted in the partial reduction of noise level in commanded velocities and control signals.

It is worth noting, that the control performance of the high-gain observer based control algorithms (including the inner-loop ADR controller) highly depends on the control-loop frequency. Obtained control-loop frequencies f and standard deviations of control-loop sampling time σ_s are presented in the Table 3. Utilizing the method I, the control-loop frequency is almost twice as large as the one obtained with method II, so we can conclude that method I is of better use for the VFO-ADR algorithm.

Table 3. Average control-loop frequency and standard deviation of its sampling time

-	f [Hz]	σ_s [s]
Method I	164.52	0.0003
Method II	84.57	0.0023

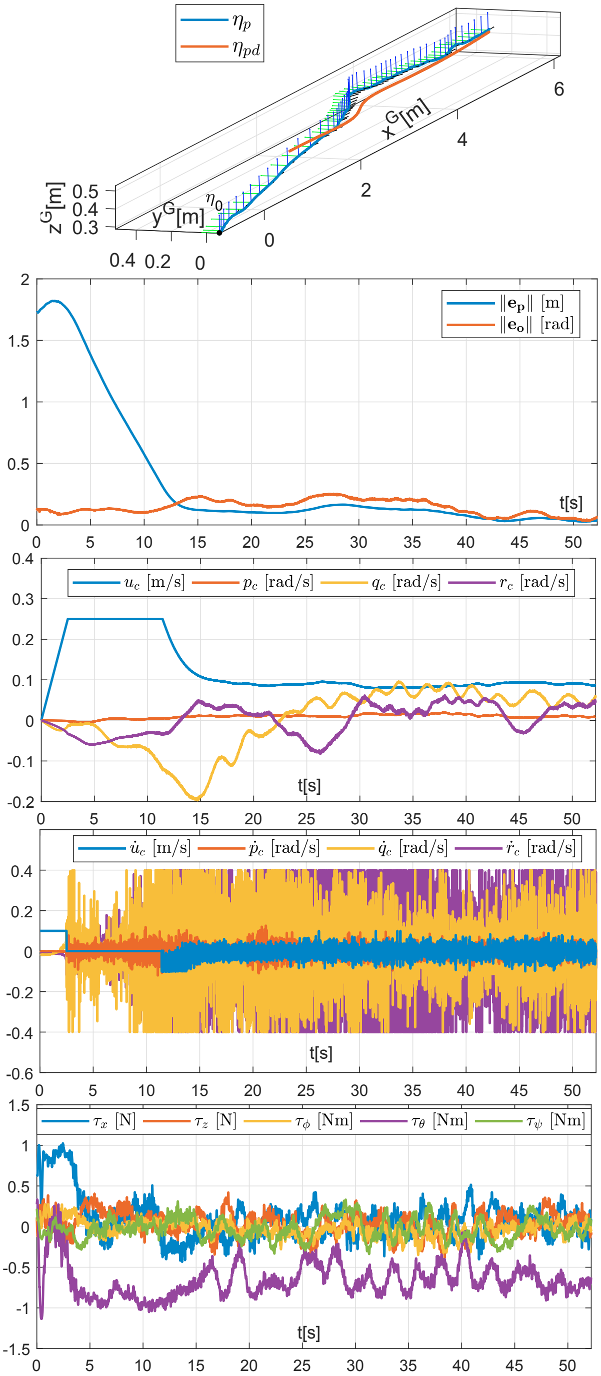

Fig. 6. Results of the experiment conducted with the underactuated airship.

REFERENCES

Bechlioulis C. P., et. al. Trajectory tracking With Prescribed Performance for Underactuated Underwater Vehicles Under Model Uncertainties and External Disturbances *IEEE Trans. Control Syst. Technol.*, 25(2):429–440, 2016.

Biagiotti L., Melchiorri C. Trajectory Planning for Automatic Machines and Robots pp. 209–244, *Springer*, 2008, Berlin, Germany.

Han J. From PID to Active Disturbance Rejection Control *IEEE Trans. Ind. Electron.*, 56(3):900-906, 2009.

Łakomy K., Michałek M. M. The VFO path-following kinematic controller for robotic vehicles moving in a 3D space *RoMoCo 2017*, 264–268, July 3-5, 2017, Poland.

Michałek M. M., Adamski W. Output-feedback cascaded VFO-ADRC tracking controller for autonomous rigid-body vehicles moving in a 3D space *RoMoCo 2017*, 251–256, July 3-5, 2017, Poland.

Michałek M. M. Robust trajectory following without availability of the reference time-derivatives in the control scheme with active disturbance rejection *ACC 2016*, 1536–1541, July 6-8, 2016, Boston, MA, USA.

Sussmann H. J., Kokotovic P. V. The peaking phenomenon and the global stabilization of nonlinear systems *IEEE Trans. Autom. Control*, 36(4):424–440, 1991.

Chen T., Zhu M., Zheng Z. Asymmetric error-constrained path-following control of a stratospheric airship with disturbances and actuator saturation *Mech Syst Signal Process*, 119:501–522, 2019.

Huang Y., Xue W. Active disturbance rejection control: Methodology and theoretical analysis *ISA Transactions*, 53(4):964–976, 2014.

Fossen, T.I. Guidance and control of ocean vehicles *John Wiley & Sons*, 1999, Chichcester, UK.