# Tracking control of an articulated intervention AUV in 6DOF using the generalized super-twisting algorithm

Ida-Louise G. Borlaug and Kristin Y. Pettersen and Jan Tommy Gravdahl

***Abstract*— The articulated intervention AUV (AIAUV) is an underwater swimming manipulator (USM) with intervention capabilities. Station-keeping and trajectory tracking are essential for the AIAUV to be able to move in confined spaces and to perform intervention tasks. In this paper we propose using the generalized super twisting algorithm, which is an extension of the regular super-twisting algorithm, for the trajectory tracking of the joint angles, position and orientation of the base of the AIAUV in 6DOF. Furthermore, we show the ultimate boundedness of the tracking errors. We also demonstrate the applicability of the proposed control law and compare the performance with the regular super-twisting algorithm with adaptive gains.**

## I. Introduction

The articulated intervention autonomous underwater vehicle (AIAUV) is an underwater vehicle (UV) with multiple joints such as a manipulator arm, and multiple thrusters, i.e. an underwater swimming manipulator (USM). The thrusters give the AIAUV station-keeping capabilities and enable the AIAUV to perform trajectory tracking without using body undulations, which are necessary for underwater snake robots (USRs) [1]. The joints enable the AIAUV to operate as a manipulator arm, thus enabling the AIAUV to perform intervention tasks. These manoeuvring capabilities and its slender body enable the AIAUV to move around in confined spaces in which a remotely operated vehicle (ROV) or AUV would not have access. Moreover, the AIAUV has adopted the high kinematic redundancy of USRs and the fully energy-efficient hydrodynamic properties and tetherless operation of AUVs. These properties enable the AIAUV to exploit the full potential of the inherent kinematic redundancy [2], [3].

Station-keeping and trajectory tracking are essential for the AIAUV to be able to move in confined spaces and to perform intervention tasks. Since the AIAUV is subject to hydrodynamic and hydrostatic parameter uncertainties, uncertain thruster characteristics, unknown disturbances, and unmodeled dynamics, and since the coupling forces caused by joint motion are even larger for the AIAUV than for ROVs because it has no separate vehicle base and a low mass compared to an ROV, it is essential for the control approach to be robust. The design of a robust trajectory tracking controller is therefore the objective of this paper.

Sliding mode control (SMC) is a robust and versatile non-linear control approach that has been used for many different systems and applications, including three-phase power converters [4], Markovian jump systems [5], [6], stochastic systems [7], [8] and microgrid control [9] just to mention some. For UVs SMC has been used for singularity-free control [10] to address partly unknown non-linear thruster characteristics [11], [12] and for trajectory control [13] - [16]. SMC has also been used to handle coupling forces between a manipulator arm and a UV [17]. In [18], SMC is applied to land-based snake robots to achieve robust tracking of a desired gait pattern and underactuated straight-line path following.

In recent years, SMC has been developed into higher-order SMC schemes, that removes the chattering problem. The super-twisting algorithm (STA) with adaptive gains [19] has been tested for the AIAUV in 2DOF and in 6DOF in [20] and [21], respectively, because it is the most powerful second-order continuous SMC algorithm. STA attenuates chattering, and no conservative upper bound on the disturbance gradient has to be considered to maintain sliding because of the adaptive gains.

In [20] tracking control of the centre of mass of the AIAUV in 2D was considered by using STA with adaptive gains [19] and a higher-order sliding mode observer (HOSMO) [22]. It was proven that the tracking errors were ultimately bounded, and the simulation results demonstrated that the proposed control method provided excellent tracking capabilities. The results obtained in [20] were therefore extended to 6DOF in [21]. In [21], the position, orientation and joint angles were considered for the tracking problem, and equally good results were obtained in theory and in simulations. In [20] and [21], a HOSMO had to be used because velocity measurements were unavailable. When using a HOSMO, Euler angles have to be used to represent the system since the observer does not work with a different number of states in position versus velocity.

In this paper the generalized super-twisting algorithm (GSTA) [23], will be used for trajectory tracking in 6DOF. GSTA is an extension of STA that provides finite-time convergence in the presence of time- and state-dependent perturbations, which is essential for robust control of the AIAUV. As in [21], we consider the position, orientation and joint angles for the trajectory tracking, but we assume that the velocity measurements are available. We thus avoid using a HOSMO, which means that we can use quaternions to represent the system. We then avoid singularities in the Jacobian matrix at $\theta = \pm\pi/2$, which is a well-known problem with using Euler angles ($xyz$-convention). Finding and including an observer that works with quaternions will be a task for future work. Furthermore, we show the ultimate

I.-L. G. Borlaug, K. Y. Pettersen and J. T. Gravdahl are with the Centre for Autonomous Marine Operations and Systems, Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway. {Ida.Louise.Borlaug, Kristin.Y.Pettersen, Tommy.Gravdahl}@ntnu.no

boundedness of the tracking error, and we illustrate our theoretical findings with simulation results. Finally, GSTA is compared with STA with adaptive gains.

The contributions can be summarized as follows:

- The trajectory tracking control problem of an AIAUV in 6DOF is solved by using GSTA.
- It is proven that the tracking errors are ultimately bounded.
- The results are compared with those obtained with the STA with adaptive gains.

The remainder of this paper is organized as follows. In Section II, the model and the tracking control problem for the AIAUV are defined mathematically. The control law for tracking the desired trajectory is presented in Section III. In Section IV, we prove the ultimate boundedness of the tracking errors. A description of the simulation model implemented for this paper and the simulation results are presented in Section V. Conclusions and suggestions for future work are given in Section VI.

## II. Modelling and the tracking control problem

In this section, we present the model and the mathematical definition of the tracking control problem for the AIAUV. The AIAUV is composed of $n$ links connected by $n-1$ motorized joints, where each joint is regarded as a one-dimensional Euclidean joint. We consider link 1 to be the base and link $n$ to be the front, where the end-effector is positioned. Furthermore, the AIAUV is equipped with $m$ thrusters. To provide station-keeping capabilities it has tunnel thrusters acting through the links, and to provide forward thrust it has one or more thrusters acting along the body of the AIAUV. For control purposes, the AIAUV is considered to be a floating-base manipulator operating in an underwater environment, subject to added mass forces, dissipative drag forces, and gravity and buoyancy forces. This allows us to model the AIAUV as an underwater vehicle-manipulator system (UVMS), with dynamic equations given in matrix form as [24], [25]

$$M(q)\dot{\zeta} + C(q,\zeta)\zeta + D(q,\zeta)\zeta + g(q,R_B^I) = \tau(q) \quad (1)$$

where $q \in \mathbb{R}^{(n-1)}$ is the vector representing the joint angles, $M(q)$ is the inertia matrix including added mass terms, $C(q,\zeta)$ is the Coriolis-centripetal matrix, $D(q,\zeta)$ is the damping matrix and $g(q,R_B^I)$ is the matrix of gravitational and buoyancy forces. The control input is given by the generalized forces $\tau(q)$:

$$\tau(q) = \begin{bmatrix} T(q) & 0_{6\times(n-1)} \\ 0_{(n-1)\times m} & I_{(n-1)\times(n-1)} \end{bmatrix} \begin{bmatrix} \tau_{thr} \\ \tau_q \end{bmatrix} \quad (2)$$

where $T(q) \in \mathbb{R}^{6\times m}$ is the thruster configuration matrix, $\tau_{thr} \in \mathbb{R}^m$ is the vector of thruster forces and $\tau_q \in \mathbb{R}^{(n-1)}$ represents the joint torques. To implement the control input $\tau(q)$, a thruster allocation scheme as proposed in [26] needs to be implemented to distribute the desired control inputs onto the $m$ thrusters. The vector of body-fixed velocities, $\zeta$, is defined as

$$\zeta = \begin{bmatrix} v^T & \omega^T & \dot{q}^T \end{bmatrix}^T \in \mathbb{R}^{6+(n-1)} \quad (3)$$

where $v$ and $\omega$ are the body-fixed linear and angular velocities of the base of the AIAUV, and $\dot{q}$ is the vector of joint angle velocities. In [21] we used Euler angles to represent the orientation of the AIAUV. Here, we will instead use quaternions to avoid singularities in the Jacobian matrix. Using Euler parameters rather than Euler angles provides us the advantage of a well-defined Jacobian matrix, which is necessary to be able to use the inverse of the Jacobian matrix. However, at the same time, we cannot use the HOSMO from [22], which means that we need velocity measurements to control the system. The complete state vector specifying the position, orientation, and shape of the AIAUV is then represented as

$$\xi = \begin{bmatrix} \eta_1^T & p^T & q^T \end{bmatrix}^T \in \mathbb{R}^{7+(n-1)} \quad (4)$$

where $\eta_1 = \begin{bmatrix} x & y & z \end{bmatrix}^T \in \mathbb{R}^3$ is the position of the base and $p = \begin{bmatrix} \varepsilon^T & \eta \end{bmatrix}^T = \begin{bmatrix} \varepsilon_1 & \varepsilon_2 & \varepsilon_3 & \eta \end{bmatrix}^T \in \mathbb{R}^4$ is the unit quaternion describing the orientation of the base in the inertial frame. The Euler parameters $\eta$ and $\varepsilon$ satisfy

$$\eta^2 + \varepsilon^T\varepsilon = 1. \quad (5)$$

The kinematic differential equation for the unit quaternion can be written as [24]

$$\begin{bmatrix} \dot{\varepsilon} \\ \dot{\eta} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} \eta I_3 + S(\varepsilon) \\ -\varepsilon^T \end{bmatrix}\omega = J_{k,oq}(p)\omega \quad (6)$$

where $I_3$ is the $(3\times 3)$ identity matrix and $S(\cdot)$ is the cross-product operator defined as in [27, Definition 2.2]. To complete the dynamic model, we can write the relationship between the body-fixed velocities and the complete state vector specifying the position, orientation, and shape of the AIAUV as

$$\dot{\xi} = J(p)\zeta = \begin{bmatrix} R_B^I(p) & 0_{3\times 3} & 0_{3\times(n-1)} \\ 0_{4\times 3} & J_{k,oq}(p) & 0_{4\times(n-1)} \\ 0_{(n-1)\times 3} & 0_{(n-1)\times 3} & I_{(n-1)\times(n-1)} \end{bmatrix}\zeta \quad (7)$$

where $R_I^B$ is the rotation matrix expressing the transformation from the inertial frame to the body-fixed frame.

The desired velocities are denoted as

$$\zeta_d = \begin{bmatrix} v_d^T & \omega_d^T & \dot{q}_d^T \end{bmatrix}^T \quad (8)$$

in the body-fixed frame. The desired velocities, $\zeta_d$, are typically given by the inverse kinematics as described in [28]. The desired trajectory, $\begin{bmatrix} \eta_{1,d}^T & p_d^T & q_d^T \end{bmatrix}^T$, can then be reconstructed from the desired velocity using, for instance, a CLIK algorithm [29, Ch. 11]. The desired orientation of the base of the AIAUV with respect to the inertial reference frame is given by the unit quaternion, $p_d = \begin{bmatrix} \varepsilon_d^T & \eta_d \end{bmatrix}^T$, and the corresponding rotation matrix $R(p_d)$. The orientation error can then be specified by the composite rotation

$$R^T(p_d)R(p) = R(\tilde{p}) \quad (9)$$

where

$$\tilde{p} = \begin{bmatrix} \tilde{\varepsilon} \\ \tilde{\eta} \end{bmatrix} = \begin{bmatrix} \eta\varepsilon_d - \eta_d\varepsilon + S(\varepsilon_d)\varepsilon \\ \eta\eta_d + \varepsilon^T\varepsilon_d \end{bmatrix} \quad (10)$$

is the unit quaternion representing the orientation error. For the orientation, the aim is to ensure that $p = \pm p_d$, which corresponds to $\tilde{p} = \begin{bmatrix} 0_{1\times 3} & \pm 1 \end{bmatrix}^T$. The tracking errors then consist of the position error $\tilde{\eta}_1$, the orientation error $\tilde{\varepsilon}$ and the joint angle error $\tilde{q}$, and the tracking error vector can then be written as

$$\tilde{\xi} = \begin{bmatrix} \tilde{\eta}_1 \\ \tilde{\varepsilon} \\ \tilde{q} \end{bmatrix} = \begin{bmatrix} \eta_1 - \eta_{1,d} \\ \eta\varepsilon_d - \eta_d\varepsilon + S(\varepsilon_d)\varepsilon \\ q - q_d \end{bmatrix}. \quad (11)$$

The goal of the tracking problem is to make the error

vector, $\tilde{\xi}$, converge to zero. The tracking control objective is therefore to make $(\tilde{\xi}, \tilde{\zeta}) = (0, 0)$ an asymptotically stable equilibrium point of (1) and (7), which will ensure that the tracking error will converge to zero. Note that $\tilde{\eta}$ is not included as an independent state in (11), since $\tilde{\eta}$ and $\tilde{\varepsilon}$ satisfy (5). When $\tilde{\varepsilon} \to 0$, then $\tilde{p} = \begin{bmatrix} 0_{1\times 3} & \pm 1 \end{bmatrix}^T$.

## III. Sliding Mode Control

In this section, we find the error dynamics for the system and propose a tracking control law for the AIAUV based on the theory of SMC.

### A. Error dynamics

Define $x_1 = \tilde{\xi}$ and

$$\begin{aligned} x_2 &= \begin{bmatrix} R_B^I(\tilde{p}) & 0_{3\times 3} & 0_{3\times(n-1)} \\ 0_{3\times 3} & \frac{1}{2}(\tilde{\eta} I_3 + S(\tilde{\varepsilon})) & 0_{3\times(n-1)} \\ 0_{(n-1)\times 3} & 0_{(n-1)\times 3} & I_{(n-1)\times(n-1)} \end{bmatrix} (\zeta - \zeta_d) \\ &= T(\tilde{p})\tilde{\zeta}. \end{aligned} \tag{12}$$

Note that $T^{-1}(\tilde{p})$ is well defined such that (12) is a globally valid coordinate transformation. The reason why this is well defined will be explained in Sec. IV. The reason for choosing $x_2 = T(\tilde{p})\tilde{\zeta}$ is that this makes $x_2 = \dot{x}_1$, and by using that, we can prove that the error variables asymptotically converge to zero when the sliding surface is equal to zero (see Sec. III-B for the proof), which is a requirement when designing the sliding surface. If $x_2$ was chosen to be equal to $\tilde{\zeta}$, then this would not have been the case. The error dynamics can then be written as

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{d}{dt}(T(\tilde{p}))T^{-1}(\tilde{p})x_2 + M^{-1}(\tilde{q} + q_d)T(\tilde{p}) \\ &(-C(\tilde{q} + q_d, (T^{-1}(\tilde{p})x_2 + \zeta_d))(T^{-1}(\tilde{p})x_2 + \zeta_d) \\ &- D(\tilde{q} + q_d, (T^{-1}(\tilde{p})x_2 + \zeta_d))(T^{-1}(\tilde{p})x_2 + \zeta_d) \\ &- g(\tilde{q} + q_d, R_B^I) + \tau(\tilde{q} + q_d) - M(\tilde{q} + q_d)\dot{\zeta}_d). \end{aligned} \tag{13}$$

To reduce the space used to write the model, we will introduce some new functions, $f_1(\cdot) = \frac{d}{dt}(T(\tilde{p}))T^{-1}(\tilde{p})$ and $f_2(\cdot) = (-C(\cdot)(T^{-1}(\tilde{p})x_2 + \zeta_d) - D(\cdot)(T^{-1}(\tilde{p})x_2 + \zeta_d) - g(\cdot) - M(\cdot)\dot{\zeta}_d$, such that the model can be written as

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= f_1(\cdot)x_2 + M^{-1}(\cdot)T(\cdot)(f_2(\cdot) + \tau(\cdot)) \end{aligned} \tag{14}$$

### B. Sliding surface

To use an SMC approach, we must first design a sliding surface. It should be designed such that when the sliding variable $\sigma$ goes to zero, the error variables asymptotically converge to zero and such that the control input $\tau(q)$ appears in the first derivative of $\sigma$. The sliding surface is chosen as

$$\sigma = x_1 + x_2 \quad \in \mathbb{R}^{6+(n-1)}. \tag{15}$$

If $\sigma = 0$, we will have $x_1 + x_2 = 0$. Since $x_2 = \dot{x}_1$, we can write this as

$$\dot{x}_1 = -x_1 \tag{16}$$

which ensures that $x_1$ globally exponentially converges to zero. Since $x_1 = \tilde{\xi}$, the original state variable $\tilde{\xi}$ will also globally exponentially converge to zero if $\sigma = 0$.

### C. Generalized super-twisting algorithm

In this section, the equations describing GSTA are presented in detail. The GSTA proposed in [23] can be written as

$$\begin{aligned} u_{\mathrm{GSTA}} &= -k_1\phi_1(\sigma) + z \quad \in \mathbb{R}^{6+(n-1)} \\ \dot{z} &= -k_2\phi_2(\sigma) \end{aligned} \tag{17}$$

where

$$\begin{aligned} \phi_1(\sigma) &= \lceil \sigma \rfloor^{\frac{1}{2}} + \beta_{\mathrm{GSTA}}\sigma \\ \phi_2(\sigma) &= \frac{1}{2}\lceil \sigma \rfloor^0 + \frac{3}{2}\beta_{\mathrm{GSTA}}\lceil \sigma \rfloor^{\frac{1}{2}} + \beta_{\mathrm{GSTA}}^2\sigma \end{aligned} \tag{18}$$

where $\lceil a \rfloor^b = |a|^b \operatorname{sgn}(a)$, and $k_1 \in \mathbb{R}^{6+(n-1)}$, $k_2 \in \mathbb{R}^{6+(n-1)}$ and $\beta_{\mathrm{GSTA}} \in \mathbb{R}^{6+(n-1)}$ are controller gains. With the extra linear term, compared to STA, three degrees of freedom are obtained in the design of GSTA gains: $k_1$, $k_2$ and $\beta_{\mathrm{GSTA}}$. The linear growth term $\beta_{\mathrm{GSTA}}\sigma$ in $\phi_1$ helps to counteract the effects of state-dependent perturbations, which can exponentially increase in time. By choosing the gains as described in [23], the algorithm is proven to make $\sigma$ go to zero, globally and in finite time in the presence of state- and time-dependent uncertain control coefficients and perturbations. Note that the gains when chosen as described in [23], are defined based on bounds on the perturbations and control coefficients.

### D. Super-twisting algorithm

We want to compare the GSTA with an algorithm that has previously been used and proved more efficient than a PD controller for the AIAUV; the STA with adaptive gains, [20]. In this section for completeness, we briefly present the equations describing the STA with adaptive gains, previously presented in [20] and [21]. The difference between [20], [21] and the algorithm presented here, is that the sliding surface $\sigma$ is chosen differently. The STA with adaptive gains proposed in [19] can be written by the update law

$$\begin{aligned} u_{\mathrm{STA}} &= -\alpha|\sigma|^{1/2}\operatorname{sgn}(\sigma) + v \quad \in \mathbb{R}^{6+(n-1)} \\ \dot{v} &= -\beta_{\mathrm{STA}}\operatorname{sgn}(\sigma) \end{aligned} \tag{19}$$

where the adaptive gains are defined as

$$\dot{\alpha} = \begin{cases} \omega_1\sqrt{\frac{\gamma_1}{2}} & \text{if } \sigma \neq 0 \\ 0 & \text{if } \sigma = 0 \end{cases} \tag{20}$$

and

$$\beta_{\mathrm{STA}} = 2\varepsilon_1\alpha + \lambda_1 + 4\varepsilon_1^2 \tag{21}$$

where $\varepsilon_1 \in \mathbb{R}^{6+(n-1)}$, $\lambda_1 \in \mathbb{R}^{6+(n-1)}$, $\gamma_1 \in \mathbb{R}^{6+(n-1)}$ and $\omega_1 \in \mathbb{R}^{6+(n-1)}$ are positive constants and $\sigma$ is the sliding surface. For implementation purposes, a small boundary is put on the sliding surface such that the adaptive gains can be expressed as

$$\begin{aligned} \dot{\alpha} &= \begin{cases} \omega_1\sqrt{\frac{\gamma_1}{2}} & \text{if } |\sigma| > \alpha_m \\ 0 & \text{if } |\sigma| \leq \alpha_m \end{cases} \\ \beta_{\mathrm{STA}} &= 2\varepsilon_1\alpha_x + \lambda_1 + 4\varepsilon_1^2 \end{aligned} \tag{22}$$

where the design parameter $\alpha_m$ is a small positive constant chosen empirically. The STA with adaptive gains makes $\sigma$ and $\dot{\sigma}$ go to zero in finite-time, [19].

## IV. Stability analysis

In this section, we will analyse the closed-loop system, and we show that the tracking error converges asymptotically

to zero. In the proof of Theorem 1 under the analysis of subsystem 2, we state the results obtained in [23] before we use the Lyapunov function obtained in [23] to prove uniformity, which has not been shown previously.

### A. Overall closed-loop dynamics

By using the fact that $\dot{x}_1 = x_2$ from (13), (15) can be written as

$$\dot{x}_1 = \sigma - x_1. \tag{23}$$

By differentiating (15), we obtain

$$\dot{\sigma} = \dot{x}_1 + \dot{x}_2 = x_2 + f_1(\cdot)x_2 + M^{-1}(\cdot)T(\cdot)(f_2(\cdot)+\tau(\cdot)) \tag{24}$$

and by using that $x_2 = \sigma + x_1$, we obtain

$$\dot{\sigma} = \sigma + x_1 + f_1(\cdot)(\sigma + x_1) + M^{-1}(\cdot)T(\cdot)(f_2(\cdot)+\tau(\cdot)). \tag{25}$$

Now, by introducing $\varphi(\sigma, x_1, t) = \varphi_1(\sigma, x_1, t) + \varphi_2(\sigma, x_1, t)$, where $\varphi_1(0, x_1, t) = 0$, $\gamma(\cdot) = M^{-1}(\cdot)$, and by setting

$$\tau(\cdot) = T^{-1}(\cdot)u_{\mathrm{GSTA}} \tag{26}$$

we obtain

$$\dot{\sigma} = -k_1\gamma(\cdot)\phi_1(\sigma) + \varphi_1(\sigma, x_1, t) + \gamma(\cdot)(z + \frac{\varphi_2(\sigma, x_1, t)}{\gamma(\cdot)}) \tag{27}$$

where $\varphi_1(\sigma, x_1, t) = \sigma + f_1(\cdot)\sigma + \gamma(\cdot)(-C(\cdot)\sigma - D(\cdot)\sigma)$ and $\varphi_2(\sigma, x_1, t) = x_1 + f_1(\cdot)x_1 + \gamma(\cdot)(-C(\cdot)(x_1 + T(\cdot)\zeta_d) - D(\cdot)(x_1 + T(\cdot)\zeta_d) - T(\cdot)g(\cdot) - T(\cdot)M(\cdot)\dot{\zeta}_d)$. Now by setting $\sigma_1 = \sigma$ and $\sigma_2 = z + \varphi_2(\sigma, x_1, t)/\gamma(\cdot)$, we can write the overall closed-loop dynamics as

$$\sum_1 \begin{cases} \dot{x}_1 = \sigma_1 - x_1 \end{cases}$$
$$\sum_2 \begin{cases} \dot{\sigma}_1 = -k_1\gamma(\cdot)\phi_1(\sigma_1) + \varphi_1(\sigma_1, x_1, t) + \gamma(\cdot)\sigma_2 \\ \dot{\sigma}_2 = -k_2\phi_2(\sigma_1) + \frac{d}{dt}(\frac{\varphi_2(\sigma_1, x_1, t)}{\gamma(\cdot)}) \end{cases} \tag{28}$$

*Theorem 1:* Consider the error dynamics given by (13) and let the sliding surface $\sigma$ be defined by (15). Let the control input be given by (26). Then, the closed-loop dynamics is described by (28), and the origin of this cascade system is uniformly globally asymptotically stable (UGAS), which ensures the asymptotic convergence of the tracking error when $0 < k_m \leq \gamma(\cdot) \leq k_M$, $|\varphi_1(\cdot)| \leq \alpha|\phi_1(\sigma)|$ and $|\frac{d}{dt}(\frac{\varphi_2(\cdot)}{\gamma(\cdot)})| \leq \Delta$, where $k_m$, $k_M$, $\alpha$ and $\Delta$ are positive constants.

*Proof:* To analyse the cascade system (28), [30, Lemma 2.1] will be used. Note that the system is actually interconnected, but since subsystem 1 is well behaved as long as $\sigma$ does not explode, i.e. $x_1$ is bounded, the system can be analysed with cascaded theory by analysing along the trajectories with $x_1(t)$ bounded. When analysing the complete system, we will prove that this is indeed the case, i.e. prove that $x_1(t)$ is uniformly globally bounded. We first start by analysing subsystem 1 without perturbations.

*Analysis of subsystem 1 with $\sigma_1 = 0$:* With $\sigma_1 = 0$, subsystem 1 can be written as

$$\sum_1 \begin{cases} \dot{x}_1 = -x_1 \end{cases} \tag{29}$$

This is clearly a globally exponentially stable linear system, but since we will need a Lyapunov function to analyse this system when $\sigma_1 \neq 0$, we use the Lyapunov function candidate $V_1(x_1) = \frac{1}{2}x_1^2$ for the analysis. The derivative of $V_1$ yields

$$\dot{V}_1(x_1) = x_1\dot{x}_1 = x_1(-x_1) = -x_1^2 \leq -||x_1||^2 \tag{30}$$

This means that the Lyapunov function satisfies:

$$\begin{aligned} k_1||x_1||^a \leq V_1(x_1) \leq k_2||x_1||^a \\ \frac{\partial V_1}{\partial x} f(t, x) \leq -k_3||x_1||^a \end{aligned} \tag{31}$$

with $k_1 = k_2 = \frac{1}{2}$, $k_3 = 1$ and $a = 2$. Hence, by virtue of [31, Theorem 4.10], the origin for subsystem $\sum_1$ with $\sigma = 0$ is globally exponentially stable.

*Analysis of subsystem 2:* Subsystem $\sum_2$ has the same structure as the system in [23]. In [23], it is proven that the origin of the system is globally finite-time stable (GFTS) if $0 < k_m \leq \gamma(\cdot) \leq k_M$, $|\varphi_1(\cdot)| \leq \alpha|\phi_1(\sigma)|$ and $|\frac{d}{dt}(\frac{\varphi_2(\cdot)}{\gamma(\cdot)})| \leq \Delta$, where $k_m$, $k_M$, $\alpha$ and $\Delta$ are positive constants. Since the system is GFTS it is also globally asymptotically stable [32, Proposition 3]. To prove that the origin of $\sigma$ is UGAS, [32, Theorem 12] will be used. The function $V = \xi^T P \xi$, where $\xi^T = [\phi_1(\sigma_1) \quad \sigma_2]$ and $P = \begin{bmatrix} p_1 & -1 \\ -1 & p_2 \end{bmatrix}$, $p_1 p_2 > 1$, is the generalized Lyapunov function for subsystem 2; see [23] for details. This function is globally proper and continuous (but not Lipschitz continuous on the line $\sigma_1 = 0$). For $\sigma_1 \neq 0$, this function is differentiable and

$$DV_{F(\sigma_1,\sigma_2)}(\sigma_1, \sigma_2) \leq -\mu_1\sqrt{V(\sigma_1, \sigma_2)} \tag{32}$$

where $\mu_1 > 0$ and

$$\begin{pmatrix} \dot{\sigma}_1 \\ \dot{\sigma}_2 \end{pmatrix} \in F(\sigma_1, \sigma_2) = \begin{pmatrix} -k_1\gamma(\cdot)\phi_1(\sigma_1) + \varphi_1(\sigma_1, x_1, t) + \gamma(\cdot)\sigma_2 \\ -k_2\phi_2(\sigma_1) + \frac{d}{dt}(\frac{\varphi_2(\sigma_1, x_1, t)}{\gamma(\cdot)}) \end{pmatrix} \tag{33}$$

For $\sigma_1 = 0$ and $\sigma_2 \neq 0$ we need to calculate a generalized directional derivative. Thus, consider the limit

$$D_{\{h_n\},\{u_n\}}V(0,\sigma_2) = \lim_{n\to\infty} \frac{V(h_n u_n^{\sigma_1}, \sigma_2 + h_n u_n^{\sigma_2}) - V(0,\sigma_2)}{h_n} \tag{34}$$

where $\{h_n\} \in \mathbb{K}$ ($\mathbb{K}$ is a set of all sequences of real numbers converging to zero, i.e. $\{h_n\} \in \mathbb{K} \Leftrightarrow h_n \to 0, h_n \neq 0$), $u_n = (u_n^{\sigma_1}, u_n^{\sigma_2})^T$, $\{u_n\} \in \mathbb{M}(d)$ ($\mathbb{M}(d)$ is a set of all sequences of real vectors converging to $d \in \mathbb{R}^n$, i.e. $\{v_n\} \in \mathbb{M}(d) \Leftrightarrow v_n \to d, v_n \in \mathbb{R}^n$), and $d \in F(0, \sigma_2)$. In this case $u_n^{\sigma_1} \to \sigma_2$ and $u_n^{\sigma_2} \to q$, where $q \in [-\frac{1}{2}k_2 \pm \Delta, \frac{1}{2}k_2 \pm \Delta]$. Hence,

$$\begin{aligned} D_{\{h_n\},\{u_n\}}V(0, \sigma_2) &= \lim_{n\to\infty} \frac{V(h_n\sigma_2, \sigma_2 + h_n q) - V(0, \sigma_2)}{h_n} \\ &= \lim_{n\to\infty} \big(p_1(|h_n\sigma_2|^{(1/2)}\operatorname{sgn}(h_n\sigma_2) + \beta h_n\sigma_2)^2 \\ &\quad - 2(|h_n\sigma_2|^{(1/2)}\operatorname{sgn}(h_n\sigma_2) + \beta h_n\sigma_2)(\sigma_2 + h_n q) \\ &\quad + p_2(\sigma_2 + h_n q)^2 - p_2\sigma_2^2\big)/h_n = -\infty \end{aligned} \tag{35}$$

Therefore,

$$D_{F(\sigma_1,\sigma_2)}V(0,\sigma_2) = \{-\infty\} \leq -\mu_1\sqrt{V(0,\sigma_2)} \text{ for } \sigma_2 \neq 0 \tag{36}$$

and the origin of subsystem 2 is therefore globally uniformly finite-time stable [32, Theorem 12], and therefore, it is also UGAS. This results implies that $||\sigma(t)|| < \beta \ \forall t \geq 0$.

*Analysis of the complete system:* To analyse the complete system, [30, Lemma 2.1] is used. To check whether the solutions of the complete system are uniformly globally bounded, the boundedness of $x_1$ must be evaluated when $\sigma_1 \neq 0$. The derivative of the Lyapunov function $V_1$ is then as follows:

$$\begin{aligned} V_1(x_1) &= -||x_1||^2 + \sigma x_1 \\ &\leq -||x_1||^2 + \theta||x_1||^2 - \theta||x_1||^2 + \beta||x_1|| \\ &\leq -(1-\theta)||x_1||^2 \quad \forall \quad ||x_1|| \geq \frac{\beta}{\theta} \end{aligned} \tag{37}$$

where $0 < \theta < 1$. The solutions are then UGB because the conditions of [31, Theorem 4.18] are satisfied. Consequently, the conditions of [30, Lemma 2.1] are satisfied, which implies that the origin of the complete system is UGAS. ■

*Remark 1:* One way for the inequalities in Theorem 1 to be satisfied is if the assumptions in Theorem 2 are satisfied. The parameters $\alpha$, $k_m$, $k_M$ and $\Delta$ should then be chosen according to the inequalities (38), (41) and (42) given in the proof, and the procedure in [23] can then be used for choosing the gains $k_1$, $k_2$ and $\beta_{\text{GSTA}}$ in (17) and (18), which will ensure the finite-time convergence.

*Theorem 2:* Consider the closed-loop system in (28). If the following assumptions are satisfied

*Assumption 1:* The AIAUV is neutrally buoyant.

*Assumption 2:* The AIAUV has only revolute joints.

*Assumption 3:* The reference trajectory and its derivatives are continuous and bounded by design.

*Assumption 4:* The matrix $||\frac{d^2}{dt^2}T(\cdot)|| \leq T_M$, where $T(\cdot)$ is defined in (12) is bounded, the Coriolis-centripetal matrix is bounded by $||C(\cdot)|| \leq C_M||x_2||$ and $||\frac{d}{dt}C(\cdot)|| \leq C_m||x_2||$, the damping matrix is bounded by $||D(\cdot)|| \leq D_M||x_2||$ and $||\frac{d}{dt}D(\cdot)|| \leq D_m||x_2||$, and the matrix of gravitational and buoyancy forces is bounded by $||\frac{d}{dt}g(\cdot)|| \leq g_M||x_2||$.

*Assumption 5:* $x_2(t)$ is bounded.

then positive constants $k_m$, $k_M$, $\alpha$ and $\Delta$ exist such that

1) Inequality 1: $0 < k_m \leq \gamma(\cdot) \leq k_M$
2) Inequality 2: $|\varphi_1(\cdot)| \leq \alpha|\phi_1(\sigma)|$
3) Inequality 3: $|\frac{d}{dt}(\frac{\varphi_2(\cdot)}{\gamma(\cdot)})| \leq \Delta$

are satisfied.

*Remark 2:* These assumptions are valid since the AIAUV is a mechanical system.

*Proof:* To be able to prove that the above inequalities are satisfied, we first note some properties that arise from having revolute joints: [24].

1) Property 1: $\lambda_{\min}(M) \leq ||M|| \leq \lambda_{\max}(M)$
2) Property 2: $M = M^T > 0$
3) Property 3: $\dot{M} = C + C^T$ and $\zeta^T(\dot{M} - 2C)\zeta = 0 \ \forall \ \zeta \in \mathbb{R}^{6+(n-1)}$

*Proof of Inequality 1:* $0 < k_m \leq \gamma(\cdot) \leq k_M$

Since $\gamma(\cdot) = M^{-1}(\cdot)$, we need to prove that

$$0 < k_m \leq M^{-1}(\cdot) \leq k_M \tag{38}$$

From Properties 1 and 2, we have that the above is true, and Inequality 1 is therefore satisfied.

*Proof of Inequality 2:* $|\varphi_1(\cdot)| \leq \alpha|\phi_1(\sigma)|$

Since $\varphi_1(\sigma, x_1, t) = \sigma + f_1(\cdot)\sigma + \gamma(\cdot)(-C(\cdot)\sigma - D(\cdot)\sigma)$ with $f_1(\cdot) = \frac{d}{dt}(T(\tilde{p}))T^{-1}(\tilde{p})$, we need to prove that

$$\begin{aligned} &|\sigma + f_1(\cdot)\sigma + \gamma(\cdot)(-C(\cdot)\sigma - D(\cdot)\sigma)| \leq \\ &\alpha|\phi_1(\sigma)| = \alpha|\lceil\sigma\rfloor^{\frac{1}{2}} + \beta_{\text{GSTA}}\sigma|. \end{aligned} \tag{39}$$

By rewriting

$$\begin{aligned} &|1 + f_1(\cdot) + \gamma(\cdot)(-C(\cdot) - D(\cdot))||\sigma| \leq \\ &\alpha|\phi_1(\sigma)| = \alpha|\lceil\sigma\rfloor^{\frac{1}{2}} + \beta_{\text{GSTA}}\sigma| \end{aligned} \tag{40}$$

we find that if

$$|1 + f_1(\cdot) + \gamma(\cdot)(-C(\cdot) - D(\cdot))| \leq \alpha, \tag{41}$$

the inequality holds. Now, $T(\cdot)$ is a matrix that contains the rotation matrix $R_B^I$, the identity matrix and the expression $(1/2)(\tilde{\eta}I_3 + S(\tilde{\varepsilon}))$, which comes from $J_{k,oq}(\tilde{p})$. Since they are all bounded, the matrix $T(\cdot)$ will also be bounded. The matrix $T(\cdot)$ is also well defined since quaternions are used, which means that $T^{-1}(\tilde{p})$ exists and will also be bounded. By taking the derivative of $T(\cdot)$ we find that for $\frac{d}{dt}(T(\tilde{p}))$ to be bounded, $x_2(t)$ needs to be bounded, which it is by assumption. The function $f_1(\cdot)$ is therefore a function of bounded signals and $f_1(\cdot)$ is thus bounded. The function $\gamma(\cdot)$ is found to be bounded in the proof of Inequality 1. The matrices $C(\cdot)$ and $D(\cdot)$ are bounded by assumption as long as $x_2(t)$ is bounded, which is bounded by assumption. The matrices $C(\cdot)$ and $D(\cdot)$ are therefore bounded, and since all the functions on the right-hand side of (41) are bounded, the inequality holds. Eq. (39) is therefore satisfied, and thus Inequality 2 is satisfied.

*Proof of Inequality 3:* $|\frac{d}{dt}(\frac{\varphi_2(\cdot)}{\gamma(\cdot)})| \leq \Delta$

Since $\varphi_2(\sigma, x_1, t) = x_1 + f_1(\cdot)x_1 + \gamma(\cdot)(-C(\cdot)(x_1 + T(\cdot)\zeta_d) - D(\cdot)(x_1 + T(\cdot)\zeta_d) - T(\cdot)g(\cdot) - T(\cdot)M(\cdot)\dot{\zeta}_d)$, we need to prove that

$$\begin{aligned} &|\frac{d}{dt}((x_1 + f_1(\cdot)x_1 + \gamma(\cdot)(-C(\cdot)(x_1 + T(\cdot)\zeta_d) - D(\cdot) \\ &(x_1 + T(\cdot)\zeta_d) - T(\cdot)g(\cdot) - T(\cdot)M(\cdot)\dot{\zeta}_d))/\gamma(\cdot))| \leq \Delta. \end{aligned} \tag{42}$$

By differentiating, we find that for the above to hold, we need that $\frac{d}{dt}f_1(\cdot)$, $\frac{d}{dt}\gamma(\cdot)$, $\frac{d}{dt}C(\cdot)$, $\frac{d}{dt}D(\cdot)$, $g(\cdot)$, $\frac{d}{dt}g(\cdot)$ and $\frac{d}{dt}M(\cdot)$ are bounded since $x_1(t)$, $x_2(t)$, $f_1(\cdot)$, $\gamma(\cdot)$, $T(\cdot)$, $M(\cdot)$, $C(\cdot)$, $D(\cdot)$, $\frac{d}{dt}(T(\tilde{p}))$, $\zeta_d$, $\dot{\zeta}_d$ and $\ddot{\zeta}_d$ have been proven to be bounded before or are bounded by assumption.

For the functions $\frac{d}{dt}f_1(\cdot)$ to be bounded, we need for the matrix $\frac{d^2}{dt^2}T(\cdot)$ to be bounded, which it is by assumption; thus $\frac{d}{dt}f_1(\cdot)$ is bounded. The time derivative $\frac{d}{dt}\gamma(\cdot)$ is bounded if $\frac{d}{dt}M(\cdot)$ and $M(\cdot)$ are bounded. Since $C(\cdot)$ is bounded, $\frac{d}{dt}M(\cdot)$ is bounded (from Property 3), and $M(\cdot)$ is bounded by Property 1. The function $\frac{d}{dt}\gamma(\cdot)$ is therefore bounded. Furthermore, $\frac{d}{dt}C(\cdot)$ and $\frac{d}{dt}D(\cdot)$ are bounded since $x_2(t)$ is bounded by assumption. The matrix $g(\cdot)$ is bounded since the AIAUV is neutrally buoyant, and $\frac{d}{dt}g(\cdot)$ is bounded by assumption since $x_2(t)$ is bounded. Now, since (42) is satisfied Inequality 3 is satisfied. ■

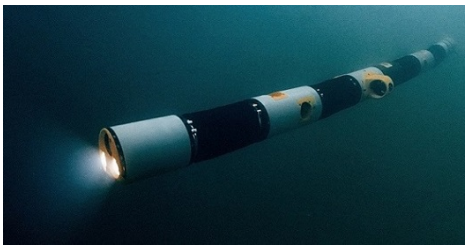

Fig. 1. The Eelume vehicle (Courtesy: Eelume)

## V. Simulation Results

### A. Implementation

The complete model and controllers are implemented in MATLAB Simulink. The model is implemented by the method described in [33].The implemented AIAUV is based on the Eelume robot, Fig. 1. The AIAUV has $n = 9$ links and $m = 7$ thrusters. The properties of each link are presented

in Tab. I. In the thrusters column, "2: Z, Y" means that the links have 2 thrusters, one working in the $z$-direction and one working in the $y$-direction of the link. Since the robot has $n = 9$ links, it has $n - 1 = 8$ joints. All the joints were implemented as revolute. The joint properties are presented in Tab. II. In the simulation we use an inverse kinematic

TABLE I
EELUME LINK PROPERTIES

| Link nr. | Length [m] | Volume [$m^3$] | Thrusters |
|---|---|---|---|
| 1 | 0.62 | 0.0143 | 0 |
| 2, 4, 6, 8 | 0.104 | 0.006 | 0 |
| 3 | 0.584 | 0.0127 | 2: Z, Y |
| 5 | 0.726 | 0.0098 | 3: X, X, Z |
| 7 | 0.584 | 0.0127 | 2: Y, Z |
| 9 | 0.37 | 0.0078 | 0 |

TABLE II
EELUME JOINT PROPERTIES

| Joint nr. | Joint rotation axes |
|---|---|
| 1, 3, 5, 7 | Z |
| 2, 4, 6, 8 | Y |

controller to give us the reference that we want the AIAUV to follow, as proposed in [26]. The thruster allocation matrix is also implemented as proposed in [26].

### B. Simulations

The task that is performed in the simulation is trajectory tracking for the base of the AIAUV. A suitable path for the base to follow is generated by giving set-points to an inverse kinematic controller. The set-points given are for the end-effector of the AIAUV, and the inverse kinematic then generates a reference trajectory for the base and joints, such that the end-effector reaches it target. Three different set-points are given to the inverse kinematic, and they change at 5, 200, 400 seconds. In Fig. 2 and Fig. 3 the reference trajectory for the base (i.e. position and orientation) and joints are presented. For the simulations, a fixed-step solver with a step size of $10^{-4}$ was used. Gains where chosen such that the comparison with the STA with adaptive gains would be as fair as possible. Specifically, the gains were chosen such that the two algorithms use the same maximum thruster force, i.e. the absolute maximum amplitude for the thruster forces are as similar as possible. Since the STA has an adaptive gain $\alpha$, the choice of parameters is not that important for the STA. The choice of gains can impact how fast the adaptive gain reaches its optimal value, but it will always reach that value. The gains for the STA were therefore chosen by tuning them manually. Specifically, the gains in the super-twisting algorithm with adaptive gains were set to $\varepsilon_1 = [0.0001e_{14}]^T$, $\lambda_1 = [0.1e_6 \quad 5e_8]^T$, $\gamma_1 = [e_{14}]^T$, $\omega_1 = [8e_{14}]^T$ and $\alpha_m = [0.005e_{14}]^T$. In Fig. 5 the thruster forces applied when using STA with adaptive gains are presented. The GSTA gains were then tuned to achieve

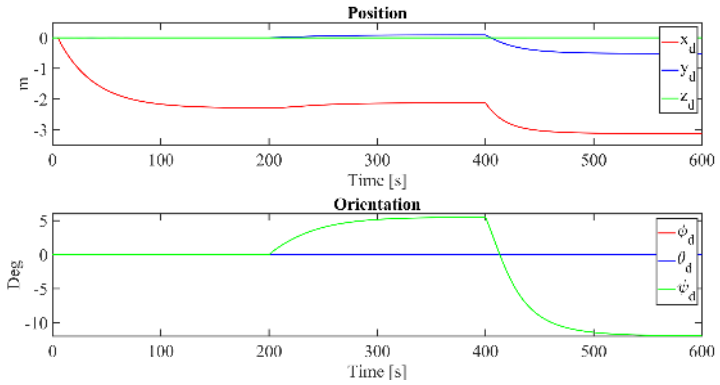


Fig. 2. Reference position and orientation of the base

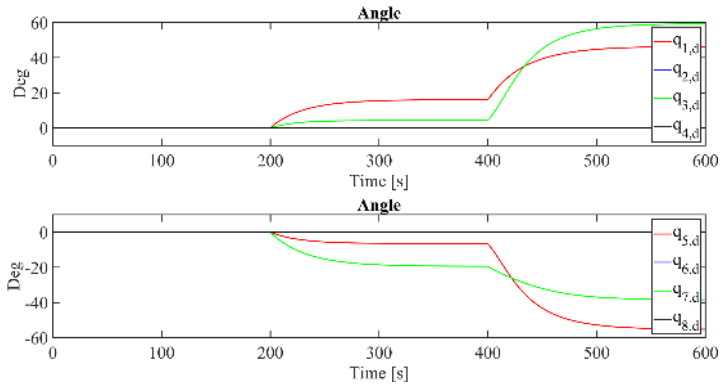


Fig. 3. Reference joint angles

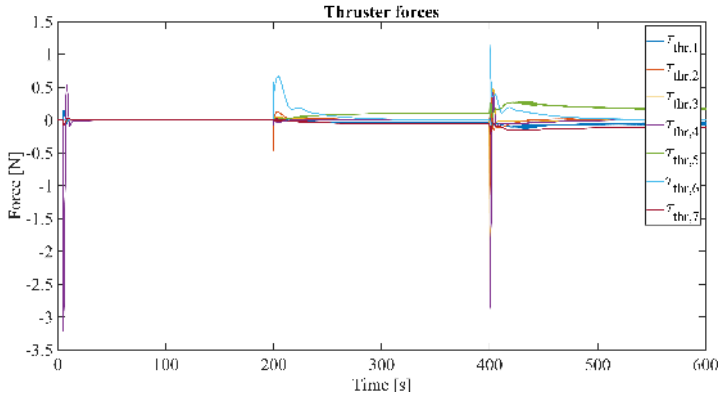


Fig. 4. GSTA: Thruster forces

similar maximum amplitude for the resulting thruster forces. The gains were chosen as $k_1 = [5e_{14}]^T$, $k_2 = [0.0002e_{14}]^T$ and $\beta_{\text{GSTA}} = [15e_{14}]^T$ where $e_i$ is a $1 \times i$ vector of ones. In Fig. 4 the thruster forces applied when using GSTA are presented. Fig. 6 presents the simulation results for the position errors and orientation errors of the base. In Fig. 7, the simulation results for the joint angles errors are presented. Tab. III presents the absolute maximum position error before and after settling for both algorithms.

### C. Discussion

From Figs. 6 and 7 we can see that there are small differences in the tracking performance of the two algorithms.

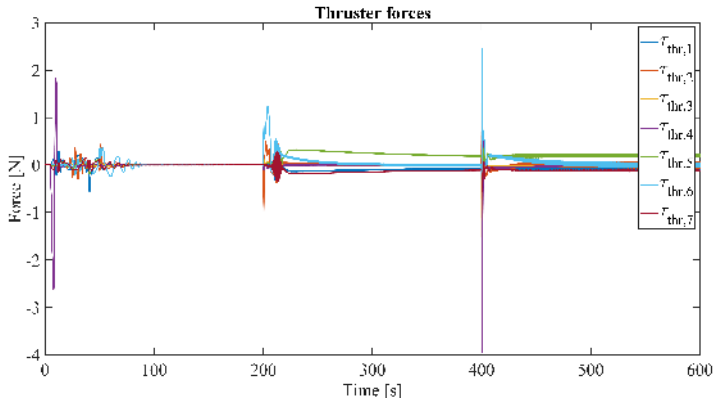


Fig. 5. STA with adaptive gains: Thruster forces

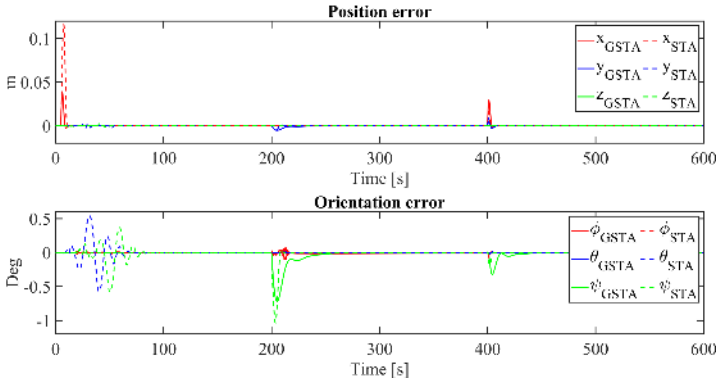


Fig. 6. Position and orientation error

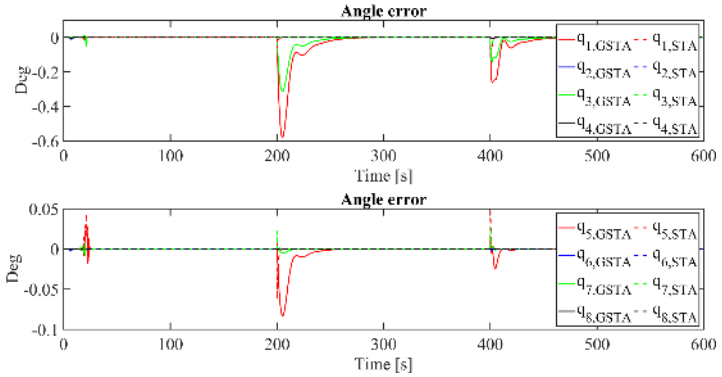


Fig. 7. Joint angles error

In the $x$-direction we can see that the GSTA has a smaller overshoot than the STA before settling, and for $\theta$ and $\psi$ we can see that when the STA is used there are small oscillations at the beginning of the simulation. For $\psi$ we can see that after $200s$ and $400s$, i.e. when the set-points are changed for the end-effector, there is a small overshoot for both algorithms. However, the overshoot at $200s$ is a bit larger for the STA then for the GSTA. For the joint angle errors, shown in Fig. 7, we can see the same tendencies as we did for the orientation errors. The STA produce some oscillations in the beginning that the GSTA does not have, and then there is an overshoot for both algorithms at $200s$ and $400s$. However, for the joint angle errors the overshoot at $200s$ is a bit larger for the GSTA than for the STA. By taking a look at Table III, we see that for the position and orientation (expect for $y$) the GSTA gives better error before settling, while for the joint angle errors the STA gives better error before settling. The STA also gives better error after settling ((11/14) cases). This can perhaps be explained by looking at the thruster use in Fig. 4 and Fig. 5. We can see that the thruster forces used when using STA (Fig. 5) are much more aggressive than when using GSTA (Fig. 4). In the beginning the STA uses longer time to settle, but after $200s$ and $400s$ the GSTA uses longer time to settle. The STA does however have some tendencies of chattering, that the GSTA does not have. So the GSTA has a bit better thruster use, since the thruster forces are smoother. It also has mostly smaller errors before settling, at least for the position and orientation, but it does have larger errors after settling. The STA on the other hand is more aggressive, and therefore produces some larger overshoots in the beginning, but it gives better errors than the GSTA after settling. The thruster use for both algorithms are well within the thruster limit of $50N$, which is the limit of the thrusters on the AIAUV.

One thing worth noticing about the GSTA is that by tuning the gains one can reduce the thruster use noticeably, without losing too much when it comes to tracking capabilities. While for the STA this is not that easy since the gains are adaptive, and therefore will converge to what is appropriate. One can choose the gains such that the STA with adaptive gains is a bit less aggressive, but then it takes longer time for the errors to converge, and the errors are not noticeable changed. Using the GSTA, the simulations show that even though we reduce the thruster use we can achieve an error in the magnitude of $10^{-4}$, and also a low error before settling. Then the question comes down to whether we need better tracking error than $10^{-4}$, and that depends on restrictions on thrusters and positioning systems. For instance, if the positioning system does not give better measurement errors than $10^{-4}$, then we might not need better tracking errors than $10^{-4}$.

TABLE III
ABSOLUTE MAXIMUM VALUE FOR ERRORS

| | **Errors** | | | |
|---|---|---|---|---|
| | **GSTA** | | **STA** | |
| | Before settling | After settling | Before settling | After settling |
| $x$ | 0.0384 | $2.4886 \cdot 10^{-6}$ | 0.1166 | $6.1575 \cdot 10^{-8}$ |
| $y$ | 0.0089 | $2.6793 \cdot 10^{-7}$ | 0.0075 | $3.7481 \cdot 10^{-7}$ |
| $z$ | $1.2834 \cdot 10^{-4}$ | $1.3775 \cdot 10^{-7}$ | 0.0039 | $2.7480 \cdot 10^{-8}$ |
| $\phi$ | $5.2596 \cdot 10^{-4}$ | $4.1004 \cdot 10^{-5}$ | 0.0018 | $3.8510 \cdot 10^{-6}$ |
| $\theta$ | $2.8117 \cdot 10^{-4}$ | $2.3776 \cdot 10^{-7}$ | 0.0101 | $2.8715 \cdot 10^{-7}$ |
| $\psi$ | 0.0126 | $1.2887 \cdot 10^{-5}$ | 0.0181 | $5.4592 \cdot 10^{-7}$ |
| $q_1$ | 0.0101 | $1.8077 \cdot 10^{-5}$ | 0.0026 | $8.2499 \cdot 10^{-7}$ |
| $q_2$ | $1.6885 \cdot 10^{-4}$ | $8.1314 \cdot 10^{-6}$ | $1.5384 \cdot 10^{-5}$ | $3.9801 \cdot 10^{-6}$ |
| $q_3$ | 0.0055 | $5.5463 \cdot 10^{-6}$ | 0.0023 | $6.1911 \cdot 10^{-8}$ |
| $q_4$ | $8.5975 \cdot 10^{-5}$ | $2.6329 \cdot 10^{-6}$ | $7.1403 \cdot 10^{-6}$ | $2.0984 \cdot 10^{-6}$ |
| $q_5$ | 0.0015 | $1.4348 \cdot 10^{-7}$ | 0.0011 | $4.2552 \cdot 10^{-9}$ |
| $q_6$ | $4.5758 \cdot 10^{-5}$ | $2.9402 \cdot 10^{-6}$ | $9.6045 \cdot 10^{-6}$ | $1.3011 \cdot 10^{-6}$ |
| $q_7$ | $1.0849 \cdot 10^{-4}$ | $5.2766 \cdot 10^{-8}$ | $5.0211 \cdot 10^{-4}$ | $3.1874 \cdot 10^{-8}$ |
| $q_8$ | $6.3008 \cdot 10^{-6}$ | $4.5485 \cdot 10^{-8}$ | $4.5491 \cdot 10^{-6}$ | $3.4363 \cdot 10^{-7}$ |

## VI. Conclusions and Future Research

In this paper, we have proposed the generalized super-twisting algorithm for solving the trajectory tracking control problem of the AIAUV. Furthermore, we have proven that the closed-loop error system is uniformly globally asymptotically stable, and have performed a simulation study to verify the applicability of the proposed control law in 6DOF. Specifically, we have performed a comparison study between the generalized super-twisting algorithm and the super-twisting algorithm with adaptive gains. The conclusion of the simulation study is that both algorithms can be used, but which one should be used depends on restrictions on thruster forces and the accuracy of the measurement data. The generalized super-twisting algorithm shows better thruster use than the super-twisting algorithm with adaptive gains.

Future work includes experiments to investigate the performance of the control algorithm in practice and finding and including an observer that work with quaternions.

## Acknowledgement

This research was funded by the Research Council of Norway through the Centres of Excellence funding scheme, project No. 223254 NTNU AMOS.

## References

[1] E. Kelasidi, K. Y. Pettersen, J. T. Gravdahl, and P. Liljebäck, "Modeling of underwater snake robots," in *Proc. 2014 IEEE International Conference on Robotics and Automation*, Hong Kong, China, May 31-June 7 2014, pp. 4540–4547.

[2] J. Sverdrup-Thygeson, E. Kelasidi, K. Y. Pettersen, and J. T. Gravdahl, "A control framework for biologically inspired underwater swimming manipulators equipped with thrusters," in *Proc. 10th IFAC Conference on Control Applications in Marine Systems*, vol. 49, no. 23, Trondheim, Norway, Sep. 13-16 2016, pp. 89–96.

[3] ——, "The Underwater Swimming Manipulator - A Bio-Inspired AUV," in *Proc. 2016 IEEE OES Autonomous Underwater Vehicles*, Tokyo, Japan, Nov 6-8 2016, pp. 387–395.

[4] J. Liu, S. Vazquez, L. Wu, A. Marquez, H. Gao, and L. G. Franquelo, "Extended State Observer-Based Sliding-Mode Control for Three-Phase Power Converters," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 1, pp. 22–31, 2017.

[5] F. Li, L. Wu, P. Shi, and C. Lim, "State estimation and sliding mode control for semi-Markovian jump systems with mismatched uncertainties," *Automatica*, vol. 51, pp. 385–393, 2015.

[6] B. Chen, Y. Niu, Y. Zou, and B. Chen, "Adaptive sliding mode control for stochastic Markovian jumping systems with actuator degradation," *Automatica*, vol. 49, no. 6, pp. 1748–1754, 2013.

[7] W. Ligang, Z. Wei Xing, and G. Huijun, "Dissipativity-Based Sliding Mode Control of Switched Stochastic Systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 3, pp. 785–791, 2013.

[8] S. Peng, X. Yuanqing, G. P. Liu, and D. Rees, "On designing of sliding-mode control for stochastic jump systems," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 97–103, 2006.

[9] H. Yan, X. Zhou, H. Zhang, F. Yang, and Z. G. Wu, "A novel sliding mode estimation for microgrid control with communication time delays," *IEEE Transactions on Smart Grid*, vol. PP, no. 99, pp. 1–1, 2017.

[10] G. Antonelli and S. Chiaverini, "Singularity-free regulation of underwater vehicle-manipulator systems," in *Proc. American Control Conference*, Philadelphia, Pennsylvania, June 24-26 1998, pp. 399–403.

[11] T. I. Fossen, "Adaptive macro-micro control of nonlinear underwater robotic systems," in *Proc. 5th International Conference on Advanced Robotics*, Pisa, Italy, June 19-22 1991, pp. 1569–1572.

[12] T. I. Fossen and S. Sagatun, "Adaptive control of nonlinear underwater robotic systems," *Modeling, Identification and Control*, vol. 12, no. 2, pp. 95–105, 1991.

[13] S. Soylu, B. J. Buckham, and R. P. Podhorodeski, "A chattering-free sliding-mode controller for underwater vehicles with fault-tolerant infinity-norm thrust allocation," *Ocean Engineering*, vol. 35, no. 16, pp. 1647–1659, 2008.

[14] D. Zhu and B. Sun, "The bio-inspired model based hybrid sliding-mode tracking control for unmanned underwater vehicles," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 10, pp. 2260–2269, 2013.

[15] J. Xu, M. Wang, and L. Qiao, "Dynamical sliding mode control for the trajectory tracking of underactuated unmanned underwater vehicles," *Ocean Engineering*, vol. 105, pp. 54–63, 2015.

[16] S. Liu, Y. Liu, and N. Wang, "Nonlinear disturbance observer-based backstepping finite-time sliding mode tracking control of underwater vehicles with system uncertainties and external disturbances," *Nonlinear Dynamics*, vol. 88, no. 1, pp. 465–476, 2017.

[17] M. W. Dannigan and G. T. Russell, "Evaluation and reduction of the dynamic coupling between a manipulator and an underwater vehicle," *IEEE Journal of Oceanic Engineering*, vol. 23, no. 3, pp. 260–273, 1998.

[18] E. Rezapour, K. Y. Pettersen, P. Liljebäck, and J. T. Gravdahl, "Differential geometric modelling and robust path following control of snake robots using sliding mode techniques," in *Proc. 2014 IEEE International Conference on Robotics and Automation*, Hong Kong, China, May 31-June 7 2014, pp. 4532–4539.

[19] Y. B. Shtessel, J. A. Moreno, F. Plestan, L. M. Fridman, and A. S. Poznyak, "Super-twisting adaptive sliding mode control: A Lyapunov design," in *Proc. 49th IEEE Conference on Decision and Control*, Atlanta, GA, USA, Dec. 15-17 2010, pp. 5109–5113.

[20] I.-L. G. Borlaug, J. T. Gravdahl, J. Sverdrup-Thygeson, K. Y. Pettersen, and A. Loría, "Trajectory tracking for underwater swimming manipulator using a super twisting algorithm," *Asian Journal of Control*, vol. 21, no. 1, pp. 1–16, 2019.

[21] I.-L. G. Borlaug, K. Y. Pettersen, and J. T. Gravdahl, "Trajectory tracking for an articulated intervention AUV using a super-twisting algorithm in 6DOF," *IFAC PapersOnLine*, vol. 51, no. 29, pp. 311–316, Sep. 10-12 2018, Proc. 11th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles, Opatija, Croatia.

[22] K. Kumari, A. Chalanga, and B. Bandyopadhyay, "Implementation of Super-Twisting Control on Higher Order Perturbed Integrator System using Higher Order Sliding Mode Observer," in *Proc. 10th IFAC Symposium on Nonlinear Control Systems*, California, USA, Aug. 23-25 2016, pp. 873–878.

[23] I. Castillo, L. Fridman, and J. A. Moreno, "Super-Twisting Algorithm in presence of time and state dependent perturbations," *International Journal of Control*, pp. 1–14, 2016.

[24] G. Antonelli, *Underwater Robots*, 3rd ed., ser. Springer Tracts in Advanced Robotics. Springer International Publishing, 2014, vol. 96.

[25] P. J. From, J. T. Gravdahl, and K. Y. Pettersen, *Vehicle-Manipulator Systems: Modeling for Simulation, Analysis, and Control*, ser. Advances in Industrial Control. London: Springer London, 2014.

[26] J. Sverdrup-Thygeson, E. Kelasidi, K. Y. Pettersen, and J. T. Gravdahl, "The Underwater Swimming Manipulator - A Bioinspired Solution for Subsea operations," *IEEE Journal of Oceanic Engineering*, vol. 43, no. 2, pp. 1–16, 2018.

[27] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. Chichester, UK: John Wiley & Sons, Ltd, 2011.

[28] J. Sverdrup-Thygeson, S. Moe, K. Y. Pettersen, and J. T. Gravdahl, "Kinematic singularity avoidance for robot manipulators using set-based manipulability tasks," in *Proc. 1st IEEE Conference on Control Technology and Applications*, Kohala Coast, Hawaii, Aug 27-30 2017, pp. 142–149.

[29] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*. Berlin: Springer, 2008.

[30] A. Loría and E. Panteley, "Cascaded nonlinear time-varying systems: Analysis and design," *Lecture Notes in Control and Information Sciences*, vol. 311, pp. 23–64, 2005.

[31] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, N.J: Prentice Hall, 2002.

[32] A. Polyakov and L. Fridman, "Stability notions and lyapunov functions for sliding mode control systems," *Journal of the Franklin Institute*, vol. 351, no. 4, pp. 1831–1865, 2014.

[33] H. M. Schmidt-Didlaukies, A. J. Sørensen, and K. Y. Pettersen, "Modeling of Articulated Underwater Robots for Simulation and Control," in *Proc. 2018 IEEE/OES Autonomous Underwater Vehicles*, Porto, Portugal, Nov. 6-9 2018.