Super-Twisting Control with Quaternion Output Feedback for a 3-DOF Inertial Stabilization Platform

Matheus F. Reis, João C. Monteiro, Ramon R. Costa, Antonio C. Leite

***Abstract*—**

I. Introduction

II. ISP Modeling

In this section, a procedure for deriving the kinematic and dynamic models of an ISP installed on a moving base.

A. Notations and Conventions

This section presents the notations and conventions used in this work. Let $\mathbb{N} = \{0, 1, 2, \dots\}$ denote the set of *natural numbers* and define $\bar{\mathbb{N}} = \{\bar{0}, \bar{1}, \bar{2}, \dots\}$. Unless otherwise stated, $i, j, k \in \mathbb{N} \cup \bar{\mathbb{N}} \cup \{s, c\}$.

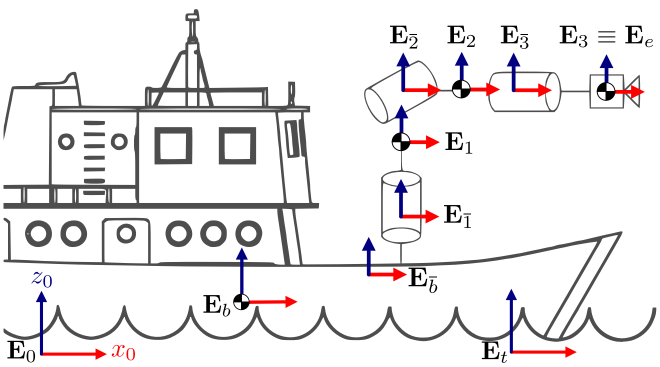

Fig. 1. Frame conventions for a 3-DOF ISP system installed on a vessel.

Considering Fig. 1, define the following:

- $\mathbf{E}_w$: world frame, arbitrarily located;
- $\mathbf{E}_{\bar{i}}$: frame fixed on body i with origin on its center of gravity (CG) ($i \in \mathbb{N}_0$);
- $\mathbf{E}_i$: fixed on body i with origin on joint i axis ($i \in \mathbb{N}_0$);
- $\mathbf{E}_c$: camera frame, fixed on the last link;
- $x_i^k, y_i^k, z_i^k \in \mathbb{R}^3$: $\mathbf{E}_i$ canonical unit vectors, written in $\mathbf{E}_k$;
- $p_{ij}^k \in \mathbb{R}^3$: position vector from the origin of frame $\mathbf{E}_i$ to the origin of $\mathbf{E}_j$, represented in $\mathbf{E}_k$;
- $v_{ij}^k \in \mathbb{R}^3$: linear velocity from $\mathbf{E}_i$ to $\mathbf{E}_j$, written in $\mathbf{E}_k$;
- $\omega_{ij}^k \in \mathbb{R}^3$: angular velocity from $\mathbf{E}_i$ to $\mathbf{E}_j$, written in $\mathbf{E}_k$;
- $h_i^k \in \mathbb{R}^3$: unit vector defining the rotation axis of joint i, represented in $\mathbf{E}_k$ ($i \in \mathbb{N}^*$);
- $m_i \in \mathbb{R}$, $I_i^i \in \mathbb{R}^{3\times 3}$: mass and inertia tensor of body i represented in $\mathbf{E}_i$ ($i \in \mathbb{N}$);
- $\widehat{*}$: $\mathbb{R}^3 \to so(3)$ cross-product operator;
- $\lfloor v \rceil^{\alpha}$: $\mathbb{R}^n \to \mathbb{R}^n$, where its elements are given by $\|v_i\|^{\alpha} \, sgn(v_i)$, with $v_i \in \mathbb{R}$ ($i = 1, ..., n$) being the elements of $v \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$;
- $\mathbf{I}_n \in \mathbb{R}^{n\times n}$: identity matrix of dimension n.

Let body 0 be the vessel and bodies $1, 2, 3$ be the ISP gimbals. Also, if a superscript is omitted, the vector is written in $\mathbf{E}_w$.

B. Quaternion-Based Kinematics

Let $R \in SO(3)$ be a *rotation matrix* describing the rotation from an arbitrary frame to another. Then, R is a diffeomorphism with respect to the projective space $\mathbb{RP}^3 = \left\{ \|v\|^2 \leq \pi \mid v \in \mathbb{R}^3 \right\}$. Therefore, each point $v \in \mathbb{RP}^3$ represent a 4-parameter representation for $SO(3)$ called the *angle-axis*, where the unitary vector on the direction of v represents the rotation axis and $\|v\|$ represents the corresponding rotation angle around that axis.

Remark 1. *Note that $\mathbb{RP}^3$ covers $SO(3)$ twice, since any point on it actually represents the same rotation than the opposite point of the sphere.*

This representation can be expressed by $v = \{\theta, n\}$, where $\theta \in \mathbb{R}$ is the angle of rotation around the unit axis vector $n \in \mathbb{R}^3$, $\|n\| = 1$. Another non-minimal representation is the *unit quaternion*. The set of *quaternions* $\mathbb{H}$ is defined by:

$$\begin{aligned} &\mathbb{H} := \{\eta + i\epsilon_1 + j\epsilon_2 + k\epsilon_3 \mid \eta, \epsilon_1, \epsilon_2, \epsilon_3 \in \mathbb{R}\} , \\ &i^2 = j^2 = k^2 = ijk = -1 . \end{aligned} \tag{1}$$

A quaternion $Q \in \mathbb{H}$ can also be represented as the pair $Q := \{\eta, \epsilon\}$, where $\eta = \mathsf{Re}(Q) \in \mathbb{R}$ represents the *real* part of the quaternion and $\epsilon = \mathsf{Im}(Q) = [\, \epsilon_1 \;\; \epsilon_2 \;\; \epsilon_3 \,]^{\mathsf{T}} \in \mathbb{R}^3$ represents the vector part. The quaternion *conjugate* is given by $Q^* = \{\eta, -\epsilon\}$. One can also represent the quaternion in fully vector form by the notation $\bar{Q} = [\, \eta \;\; \epsilon_1 \;\; \epsilon_2 \;\; \epsilon_3 \,]^{\mathsf{T}} \in \mathbb{R}^4$.

Quaternions also form an algebraic *group* with respect to *multiplication*. Given two quaternions $Q_1 = \{\eta_1, \epsilon_1\}$ and $Q_2 = \{\eta_2, \epsilon_2\}$, their multiplication follow the rules established by definition (1), which results in:

$$Q_1 \circ Q_2 = \{\eta_1\eta_2 - \epsilon_1^{\mathsf{T}}\epsilon_2, \, \eta_1\epsilon_2 + \eta_2\epsilon_1 + \epsilon_1 \times \epsilon_2\} . \tag{2}$$

The authors are with the Electrical Engineering Department of COPPE/UFRJ and PUC-Rio, Rio de Janeiro, Brazil matheus.ferreira.reis@gmail.com, {jcmonteiro, ramon}@coep.ufrj.br, antonio@ele.puc-rio.br.

Quaternion multiplication can also be performed as a linear transformation in $\mathbb{R}^4$, by:

$$\overline{Q_1 \circ Q_2} = \mathbf{H}_+(Q_1)\,\overline{Q_2}\,, \tag{3}$$
$$= \mathbf{H}_-(Q_2)\,\overline{Q_1}\,, \tag{4}$$

where $\mathbf{H}_+$, $\mathbf{H}_-$ are *Hamilton operators* defined by:

$$\mathbf{H}_+(Q) = \begin{bmatrix} \eta & -\epsilon^\mathsf{T} \\ \epsilon & \eta\,\mathbf{I}_3 + \widehat{\epsilon} \end{bmatrix}, \tag{5}$$
$$\mathbf{H}_-(Q) = \begin{bmatrix} \eta & -\epsilon^\mathsf{T} \\ \epsilon & \eta\,\mathbf{I}_3 - \widehat{\epsilon} \end{bmatrix}. \tag{6}$$

The square of the quaternion *norm* is defined as the *scalar*:

$$\|Q\|^2 = Q \circ Q^* = \{\eta^2 + \epsilon^\mathsf{T}\epsilon, 0\}\,, \tag{7}$$

and its *inverse* is the quaternion Q^{-1} such that $Q \circ Q^{-1} = \{1, 0\}$, the *unitary* quaternion.

The set of *unit quaternions* $\mathbb{H}^* = \{Q \in \mathbb{R} : \|Q\| = 1\}$ can be used as a parametrization for orientation in the following way. For an element $p = \{\theta, n\} \in \mathbb{RP}$, define:

$$Q = \left\{ cos\left(\frac{\theta}{2}\right), sin\left(\frac{\theta}{2}\right) n \right\} \in \mathbb{H}^*\,. \tag{8}$$

The inverse of an unit quaternion is given by $Q^{-1} = Q^*$, which according to (8), clearly corresponds to the opposite rotation due to negative direction of the rotation axis n.

Let $r_0, r_1, ..., r_n \in \mathbb{H}^*$ be the n absolute rotations between frames $\mathbf{E}_0, \mathbf{E}_1, ..., \mathbf{E}_n$ and the world frame $\mathbf{E}_w$, and $r^i_{i+1} \in \mathbb{H}^*$ $(i = 1, 2, ..., n-1)$ represent the rotations from frame $\mathbf{E}_i$ to $\mathbf{E}_{i+1}$. Then, since the unit quaternions form a group with respect to multiplication, then:

$$r_n = r_1 \circ r^1_2 \circ ... \circ r^{n-1}_n \in \mathbb{H}^*\,. \tag{9}$$

Now, define the set of *pure* quaternions $\mathbb{H}_p = \{v \in \mathbb{H} : \mathsf{Re}(v) = 0\}$. Note that any vector from $\mathbb{R}_3$ can be represented as the vector part of a corresponding element $v \in \mathbb{H}_p$. Besides, note that the following holds for $v, w \in \mathbb{H}_p$:

$$v \circ w = \{-\mathsf{Im}(v)^\mathsf{T}\mathsf{Im}(w),\ \mathsf{Im}(v) \times \mathsf{Im}(w)\}\,. \tag{10}$$

Let v^i and $v^j \in \mathbb{H}_p$ be representations for a vector $\vec{v}$ in frames $\mathbf{E}_i$ and $\mathbf{E}_j$, respectively, and r^i_j represents the rotation from $\mathbf{E}_i$ to $\mathbf{E}_j$, with unitary axis $n^i_j \in \mathbb{R}^3$ and rotation angle θ_{ij}. Then, the following relation is valid:

$$v^i = (r^i_j) \circ v^j \circ (r^i_j)^* = Ad_{r^i_j}\left[v^j\right]\,, \tag{11}$$

where $Ad_{r^i_j}[*]$ is the adjoint *operator*. Note that, in vector algebra, $Ad_{r^i_j}$ represents the corresponding rotation matrix $R_{ij} \in SO(3)$ associated to the unit quaternion $r^i_j \in \mathbb{H}^*$. In terms of the components of r^i_j, this matrix is given by:

$$R_{ij} = n^i_j (n^i_j)^\mathsf{T} + s_{ij}\,\mathbf{S}(n^i_j) + c_{ij}\,(\mathbf{I}_3 - n^i_j (n^i_j)^\mathsf{T})\,, \tag{12}$$

where s_{ij} and c_{ij} are the sine and cosine functions of θ_{ij}. The rotation matrix corresponding to an absolute rotation $r_i \in \mathbb{H}^*$ is written with only one subscript, as $R_i \in SO(3)$.

Algorithm 1 (Kinematic Propagation)**.** *The algorithm is initialized with the vessel configuration $p_{00} \in \mathbb{R}^3$ and $r_0 \in \mathbb{H}^*$. Then, varying index i from 0 to $n-1$, the configuration of each frame with respect to the vessel frame $\mathbf{E}_0$ can be computed by:*

$$p_{0,i+1} = p_{0,i} + R_i\,p^i_{i,i+1}\,, \tag{13}$$
$$\overline{r}_{i+1} = \mathbf{H}_+(r_i)\,\overline{r}^i_{i+1}\,. \tag{14}$$

with R_i computed from r_i in (12)*. The camera pose can be computed as $p_{0c} = p_{0n} + R_c\,p^n_{nc}$, $r_c = \mathbf{H}_+(r_n)\,\overline{r}^n_c$.*

Let $\vec{v}_i$ and $\vec{\omega}_i$ be the physical linear and angular velocities of $\mathbf{E}_i$. They are represented by $v^i_i \in \mathbb{R}^3$ and $\omega^i_i \in \mathbb{R}^3$ when written in its own body frame. Let $r_i = \{\eta_i, \epsilon_i\} \in \mathbb{H}^*$ be the absolute rotation of $\mathbf{E}_i$. The time-derivative of r_i can be related to ω^i_i by:

$$\dot{\overline{r}}_i = \begin{bmatrix} \dot{\eta}_i \\ \dot{\epsilon}_i \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -\epsilon^\mathsf{T}_i \\ \eta_i\,\mathbf{I}_3 + \widehat{\epsilon}_i \end{bmatrix} \omega^i_i\,, \tag{15}$$

which is known as the *quaternion propagation* formula.

The vector $V^i_i = [\,(v^i_i)^\mathsf{T}\ (\omega^i_i)^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^6$ is the *body velocity twist* associated to $\mathbf{E}_i$. Two body velocity twists associated to different frames $\mathbf{E}_i$, $\mathbf{E}_j$ located in the *same rigid-body* are related through the constant adjoint map $Ad_{g_{ij}} \in \mathbb{R}^{6\times 6}$:

$$V^i_i = Ad_{g_{ij}} V^j_j\,, \quad Ad_{g_{ij}} = \begin{bmatrix} R_{ij} & \hat{p}^i_{ij}\,R_{ij} \\ 0 & R_{ij} \end{bmatrix}. \tag{16}$$

which has the property $Ad_{g_{ji}} = Ad^{-1}_{g_{ij}}$.

Now, given the body twists V^0_0 and $\dot{V}^0_0$ of frame $\mathbf{E}_0$ in the ship, it is possible to compute all velocities V^i_i and accelerations $\dot{V}^i_i$ associated to each link $(i = 1, 2, 3)$ by means of an iterative algorithm described below. It consists in propagating the body velocity/acceleration twists of each link frame $\mathbf{E}_i$ through the system, obtaining V^i_i, $\dot{V}^i_i$, $i \in \{1, 2, 3\}$.

Algorithm 2 (Propagation of Velocities and Accelerations)**.** *The algorithm is initialized with given V^0_0, $\dot{V}^0_0$. Then, the velocities and accelerations are propagated upwards the kinematic chain from $i = 0$ until $i = n = 3$:*

$$V^i_i = \Omega^\mathsf{T}_{i-1,i}\,(\Phi_{i,i-1}\,V^{i-1}_{i-1} + H_i\,\dot{q}_i)\,, \tag{17}$$
$$\dot{V}^i_i = \Omega^\mathsf{T}_{i-1,i}\,(\Phi_{i,i-1}\,\dot{V}^{i-1}_{i-1} + H_i\,\ddot{q}_i + A_i\,\dot{q}_i)\,. \tag{18}$$

The velocity/acceleration twists of the camera are computed by $V^c_c = Ad_{g_{cn}}\,V^n_n$, $\dot{V}^c_c = Ad_{g_{cn}}\,\dot{V}^n_n$, with a constant transformation g_{cn}. The matrices in (17)*,* (18) *are given by:*

$$\Phi_{i+1,i} = \begin{bmatrix} \mathbf{I}_3 & -\hat{p}^i_{i,i+1} \\ 0 & \mathbf{I}_3 \end{bmatrix}, \qquad \Phi^{-1}_{i+1,i} = \begin{bmatrix} \mathbf{I}_3 & \hat{p}^i_{i,i+1} \\ 0 & \mathbf{I}_3 \end{bmatrix},$$
$$H^\mathbf{T}_{i+1} = \left[0^\mathbf{T}\,(h^i_{i+1})^\mathbf{T}\right], \qquad \Omega_{i,i+1} = \begin{bmatrix} R_{i,i+1} & 0 \\ 0 & R_{i,i+1} \end{bmatrix},$$
$$A_{i+1} = \begin{bmatrix} (\hat{v}^i_{0i} + \widehat{\hat{\omega}^i_{0i}\,p^i_{i,i+1}})\,h^i_{i+1} \\ \hat{\omega}^i_{0i}\,h^i_{i+1} \end{bmatrix},$$

where the rotation matrices $R_{i,i+1}$ are computed from $r_{i,i+1} = \left\{cos\left(\frac{1}{2}q_{i+1}\right), h^i_{i+1}\,sin\left(\frac{1}{2}q_{i+1}\right)\right\} \in \mathbb{H}^*$ using (12)*.*

Now, recall that ω_i^i can be written as the sum $\omega_i^i = \omega_0^i + \omega_{0,i}^i$ and can be expressed in terms of $q\,, \dot{q} \in \mathbb{R}^3$ by means of the *angular body link Jacobian* $J_{0i}^i(q, \Pi_g) \in \mathbb{R}^{3\times 3}$ as $\omega_{0,i}^i = J_{0i}^i(q, \Pi_g)\,\dot{q}$:

$$\omega_i^i = J_{0i}^i(q, \Pi_g)\,\dot{q} + \omega_0^i\,, \tag{19}$$

where Π_g is the vector of *geometric* parameters of the ISP, containing combinations of components of the axes and distance vectors of each link frame. Note that the body link Jacobian $J_{0i}^i(q, \Pi_g)$ can be computed numerically from (17), and also that the last $3-i$ columns of J_{0i}^i are null, since the velocity of link i only depends on the previous joints.

These kinematic relations can be used to describe the dependance among vehicle, ISP and camera motion by applying the group operation of $\mathbb{H}^*$, equation (19) and its time-derivative, with $\mathbf{E}_i = \mathbf{E}_c$:

$$\begin{aligned} r_c &= r_0 \circ r_c^0(q, \Pi_g)\,, & (20)\\ \omega_c^c &= J_{0c}^c(q, \Pi_g)\,\dot{q} + \omega_0^c\,, & (21)\\ \dot{\omega}_c^c &= J_{0c}^c(q, \Pi_g)\,\dot{q} + \dot{J}_{0c}^c(q, \Pi_g)\,\dot{q} + \dot{\omega}_0^c\,. & (22) \end{aligned}$$

C. Dynamic Equations for Vehicle Manipulator Systems

In [1], it is shown that the equations of motion for a VMS with respect to the vehicle CG frame $\mathbf{E}_b$ can be written as:

$$M_{qq}\,\ddot{q} + C_{qq}\,\dot{q} + G_q + M_{qV}\,\dot{V}_0^0 + C_{qV}^b\,V_0^0 = \tau_q\,, \tag{23}$$

where $\tau_q \in \mathbb{R}^n$ is the vector of generalized forces acting on the robot joints, collocated with $\dot{q}$. Matrices $M_{qq}(q, \Pi_g, \Pi_d) \in \mathbb{R}^{3\times 3}$ and $M_{qV}(q, \Pi_g, \Pi_d) \in \mathbb{R}^{3\times 6}$ are mass matrices, $C_{qq}(q, \dot{q}, V_0^0, \Pi_g, \Pi_d) \in \mathbb{R}^{3\times 3}$ and $C_{qV}(q, \dot{q}, V_0^0, \Pi_g, \Pi_d) \in \mathbb{R}^{3\times 6}$ are Coriolis matrices and $G_q(q, r_0, \Pi_g, \Pi_d) \in \mathbb{R}^3$ is the gravity vector.

It is well known that the *Newton Euler* method is a computationally efficient algorithm that can be used to numerically solve the *inverse dynamics* problem for (23). Given q, $\dot{q}$, $\ddot{q}$, r_0, V_0^0, $\dot{V}_0^0$ and $g \in \mathbb{R}$, the *Newton-Euler* algorithm for the inverse dynamics is expressed by:

$$\tau_q = \; NE(q, \dot{q}, \ddot{q}, r_0, V_0^0, \dot{V}_0^0, g, \Pi)\,, \tag{24}$$

where $\Pi^\mathsf{T} = \begin{bmatrix} \Pi_g^\mathsf{T} & \Pi_d^\mathsf{T} \end{bmatrix}$ contains combinations of the geometric and dynamic parameters (masses and inertias).

The Newton-Euler algorithm is composed of two steps. The first one is the *propagation of velocities and accelerations* upwards the kinematic chain, summarized in Algorithm 2. The second one consists in solving the dynamic equations of motion for each rigid body in the system, starting from the n-th link and ending up on $\mathbf{E}_b$.

Algorithm 3 (Backward Propagation of Wrenches)**.** *Solving the Newton-Euler equations for the contact body wrenches $F_i^i \in \mathbb{R}^6$ between the VMS bodies (links and vehicle), yields:*

$$\begin{aligned} F_i^i &= \Phi_{i+1,i}^\mathsf{T}\,\Omega_{i,i+1}\,F_{i+1}^{i+1} + M_i\,\dot{V}_{0i}^i + B_i\,, \qquad (25)\\ M_i &= \begin{bmatrix} m_i\,\mathbf{I}_3 & -m_i\,\hat{p}_{i\bar{i}}^i \\ m_i\,\hat{p}_{i\bar{i}}^i & I_i^i \end{bmatrix},\\ B_i &= \begin{bmatrix} m_i\,\hat{\omega}_i^i\,(\hat{\omega}_i^i\,p_{i\bar{i}}^i + v_i^i) \\ m_i\,\hat{p}_{i\bar{i}}^i\,\hat{\omega}_i^i\,v_i^i + \hat{\omega}_i^i\,I_i^i\,\omega_i^i \end{bmatrix}, \end{aligned}$$

where the parameters $p_{i\bar{i}}^i$, m_i and I_i^i compose Π_d in (24)*.*

These equations must be solved from $i = n = 3$ to $i = 1$, using the velocity and acceleration twists V_i^i and $\dot{V}_i^i$ previously computed in Algorithm 2. Also, we set here $n+1 = c$ and we do not consider external wrenches acting on the camera frame $\mathbf{E}_c$, so that $F_{n+1}^{n+1} = 0$.

Finally, the joint torques can be computed projecting the wrenches acting on frames $\mathbf{E}_i$ into their rotation axis by:

$$\tau_{q_i} = H_i^\mathsf{T}\,\Omega_{i-1,i}\,F_i^i\,. \qquad i = 1, ..., n\,. \tag{26}$$

Note that (25) does not take into account the gravity forces acting on the links. The effect of gravity (in $-z_0$ direction) is introduced by modifying $\dot{V}_i^i$ in (25) for each i-th link with:

$$\dot{V}_i^i \leftarrow \dot{V}_i^i - g \begin{bmatrix} R_i^\mathsf{T}\,z_0 \\ 0 \end{bmatrix}. \tag{27}$$

This algorithm can be used to compute the terms and some matrices of (23) separately: the mass matrices M_{qq}, M_{qV}, the gravity vector G_q and the Coriolis term $C_{qq}\,\dot{q} + C_{qV}\,V_0^0$.

III. Super-Twisting Control

A. Super-Twisting Control with Full State Feedback

In this section, a novel second-order sliding mode (SOSM) controller based on super-twisting algorithm (STC) will be developed for the stabilization and tracking of the LOS.

Consider (23), rewritten here as:

$$M_{qq}\,\ddot{q} + \tau_d = \tau_q\,, \tag{28}$$

where $\tau_q \in \mathbb{R}^3$ is the vector of generalized forces acting on the robot joints, collocated with $\dot{q}$ and $\tau_d = C_{qq}\,\dot{q} + G_q + M_{qV}\,\dot{V}_{0b}^b + C_{qV}\,V_{0b}^b \in \mathbb{R}^3$ is a disturbance vector. The dynamic model (28) can be rewritten as:

$$\begin{aligned} \dot{x}_1 &= x_2\,,\\ \dot{x}_2 &= M_{qq}^{-1}(x_1, \Pi)\,\tau_q + x_3(x_1, x_2, \Pi, t)\,, \end{aligned} \tag{29}$$

where the states $x_1 = q$, $x_2 = \dot{q}$ are the ISP joint angles and velocities and $x_3 = -M_{qq}^{-1}(x_1)\,\tau_d$ is a state-dependent disturbance.

Remark 2. *Note that, under assumption of torque control $u(t) = \tau_q$, state-space model* (29) *is a double-integrator with a nonlinear high-frequency gain and a matched disturbance $x_3 \in \mathbb{R}^3$.*

First, define the following *sliding surface*:

$$s_x = x_2 - \int_0^t w(\tau)\, d\tau\,. \tag{30}$$

where $w(\tau) \in \mathbb{R}^3$ is an arbitrary signal. Due to (29), the dynamics of the sliding variable s_x is given by:

$$\dot{s}_x = \dot{x}_2 - w(t) = M_{qq}^{-1}\, u + x_3 - w(t)\,. \tag{31}$$

The *super-twisting control law* is defined as:

$$S_t(s, A, B) = A\lfloor s\rceil^{1/2} + B\int_0^t sgn(s) d\tau\,, \tag{32}$$

where matrices $A, B > 0$. Using the inner control law

$$u(t) = M_{qq}(x_1, \Pi)\; [w(t) - S_t(s_x, \Lambda_1, \Lambda_2)]\;, \tag{33}$$

where $\Lambda_1, \Lambda_2 > 0$, (31) becomes:

$$\begin{aligned} \dot{s}_x &= -\Lambda_1\, \lfloor s_x\rceil^{1/2} + w_x\,, \\ \dot{w}_x &= -\Lambda_2\, \lfloor s_x\rceil^{0} + \dot{x}_3\,, \end{aligned} \tag{34}$$

which is identified as the STA and is finite-time stable for $\|\dot{x}_3\| < L_x$. According to [**?**], the STA guarantees SOSM in finite-time, which means that after a time $t > T_1 > 0$, $s_x = \dot{s}_x = 0$. Therefore, $\dot{x}_2 = w(t)\; \forall t > T_1$, even in the presence of the bounded disturbance x_3.

Now, in a similar way than in (29), (15) and (22) can be rewritten as:

$$\dot{y}_1 = \frac{1}{2}\begin{bmatrix} -y_{12}^{\mathsf{T}} \\ y_{11}\,\mathbf{I}_3 + \widehat{y}_{12} \end{bmatrix} y_2\,, \tag{35}$$

$$\dot{y}_2 = J_{0c}^{c}(x_1, \Pi_g)\, \dot{x}_2 + y_3(x_1, x_2, \Pi_g, t)\,. \tag{36}$$

where the state $y_1^{\mathsf{T}} = \bar{r}_c^{\mathsf{T}} = \begin{bmatrix} y_{11} & y_{12}^{\mathsf{T}} \end{bmatrix}$ is the vector representation of the camera orientation $r_c \in \mathbb{H}^*$, with $y_{11} = \eta_c$ and $y_{12} = \epsilon_c$ being the scalar and vector components. State $y_2 = \omega_c^c$ is the camera angular velocity, while $y_3 = \dot{J}_{0c}^{c}\, \dot{q} + \dot{\omega}_0^c$ is another state-dependent disturbance.

Remark 3. *Note that the state-space model* (36) *is a double integrator with a nonlinear high-frequency gain and a matched disturbance y_3 with respect to a control input $\dot{x}_2$.*

This structure strongly suggests the use of a *cascade controller* for both stabilization and tracking.

Given an orientation reference $r_{c_d}(t) \in \mathbb{H}^*$ for the camera, it can be represented in vector form by $\bar{r}_{c_d}^{\mathsf{T}}(t) = y_{1_d}^{\mathsf{T}}(t) = \begin{bmatrix} y_{11_d}(t) & y_{12_d}^{\mathsf{T}}(t) \end{bmatrix}$. The angular velocity of the camera is also given as $\omega_{c_d}^c(t) = y_{2_d}(t)$. The quaternion and angular velocity errors can be defined as:

$$e_c = r_{c_d}(t) \circ r_c^*\,, \tag{37}$$

$$e_\omega = y_{2_d}(t) - y_2\,. \tag{38}$$

Note that when $r_c = r_{c_d}(t)$, the orientation error (37) is zero. Define another *sliding surface* as

$$s_y = e_\omega + K_c\, \mathsf{Im}(e_c)\,, \quad K_c > 0\,, \tag{39}$$

whose dynamics is given by

$$\dot{s}_y = \dot{y}_{2_d} - J_{0c}^{c}(x_1)\, \dot{x}_2 - y_3 + K_c\, \psi(y_1, y_2, r_{c_d})\,, \tag{40}$$

with $\psi(y_1, y_2, r_{c_d}) = y_{11}\, \dot{y}_{12_d} - 0.5\, y_{12}^{\mathsf{T}}\, y_2\, y_{12_d} - \dot{y}_{11_d}\, y_{12} - \dot{\widehat{y}}_{12_d}\, y_{12} - 0.5\, y_{11_d}\, (y_{11}\, \mathbf{I}_3 - \widehat{y}_{12})\, y_2 - 0.5\, \widehat{y}_{12_d}\, (y_{11}\, \mathbf{I}_3 - \widehat{y}_{12})\, y_2$ and the derivatives of the reference are $\dot{y}_{11_d} = -0.5\, y_{12}^{\mathsf{T}}\, y_{2_d}$ and $\dot{y}_{12_d} = 0.5(y_{11}\, \mathbf{I}_3 - \widehat{y}_{12})\, y_{2_d}$.

Assumption 1. *Here, suppose that the system* (34) *is already sliding after a finite time $T_1 > 0$. Therefore, $\dot{x}_2 = w(t)$. Under this assumption,* (40) *becomes*

$$\dot{s}_y = \dot{y}_{2_d}(t) - J_{0c}^{c}(x_1, \Pi_g)\, w(t) - y_3 + K_c\, \psi(y_1, y_2, r_{c_d})\,.$$

Under Assumption 1, the *outer* control law:

$$w(t) = J_{0c}^{c}(x_1, \Pi)^{-1}\, [\dot{y}_{2_d}(t) + K_c\, \psi + S_t(s_y, \Lambda_3, \Lambda_4)]\;. \tag{41}$$

can be applied to (40), yielding:

$$\begin{aligned} \dot{s}_y &= -\Lambda_3\, \lfloor s_y\rceil^{1/2} + w_y\,, \\ \dot{w}_y &= -\Lambda_4\, sign(s_y) - \dot{y}_3\,, \end{aligned} \tag{42}$$

once again guaranteeing finite-time stabilization after a time $T_2 > 0$ and under the assumption of $\|\dot{y}_3\| < L_y$. It means that for all $t \geq T_2$, the system is sliding and therefore, it follows the nonlinear dynamics of the sliding variable (39), which is asymptotically stable according to [2].

Theorem 1.

Proof. □

B. HOSMO Observer for STC with Output Feedback

If all system states $q, \dot{q} \in \mathbb{R}^3$ are not available, but only the joint angles q, a second order differentiator can be used to estimate $\dot{q}$. It was introduced by Levant in his seminal work []. This system is a third-order observer based on the theory of HOSMs.

Defining the estimation error $e_{x_1} = x_1 - \hat{x}_1$, the HOSM observer for the second order nonlinear system (29) is:

$$\begin{aligned} \dot{\widehat{x}}_1 &= k_1\, \lfloor e_{x_1}\rceil^{2/3} + \widehat{x}_2\,, \\ \dot{\widehat{x}}_2 &= k_2\, \lfloor e_{x_1}\rceil^{1/3} + \widehat{x}_3 + M_{qq}^{-1}(x_1, \Pi)\, u\,, \\ \dot{\widehat{x}}_3 &= k_3\, sgn(e_{x_1})\,. \end{aligned} \tag{43}$$

Its already proven by literature [**?**] that after a finite time $T_{reach} > 0$, the states of the second-order differentiator (43) uniformly converge to $\widehat{x}_1 = x_1$, $\widehat{x}_2 = x_2$ and $\widehat{x}_3 = x_3$, given that $\|\dot{x}_3\| < L_x$.

Defining the *modified* sliding surface

$$\widehat{s}_x = \widehat{x}_2 - \int_0^t w(\tau)\, d\tau\,, \tag{44}$$

its dynamics is given by:

$$\dot{\widehat{s}}_x = k_2\, \lfloor e_{x_1}\rceil^{1/3} + \widehat{x}_3 + M_{qq}^{-1}(x_1, \Pi)\, u - w(t)\,. \tag{45}$$

Modifying control law (33) to

$$u(t) = M_{qq}(x_1, \Pi)\left[w(t) - k_2 \lfloor e_{x_1} \rceil^{1/3} - S_t(\widehat{s}_x, \Lambda_1, \Lambda_2)\right], \tag{46}$$

the exact same STA than in (34) is obtained, but with s_x replaced by the modified sliding variable $\widehat{s}_x$.

Remark 4. *Another HOSMO could be designed to estimate the camera angular velocity $\omega_c^c(t)$. However, usually the camera orientation $r_c(t)$ is obtained from an Inertial Measurement Unit (IMU), a device that combines measurements of gyroscopes (which measure angular velocity) and magnetometers (which measure magnetic fields), providing an accurate estimate for $r_c(t)$. Therefore, trustworthy direct measurements of $\omega_c^c(t)$ are usually already available.*

Theorem 2.

Proof. □

IV. Simulation Results

V. Conclusion and Future Works

References

[1] P. J. From, J. T. Gravdahl, and K. Y. Pettersen, *Vehicle-manipulator systems*. Springer, 2014.

[2] B. Siciliano, L. Sciavicco, and L. Villani, *Robotics : Modelling, Planning and Control*, ser. Advanced Textbooks in Control and Signal Processing. London: Springer, 2009, 013-81159.