Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 53-2 (2020) 9074–9079

Synchronized mobile manipulators for kinematic cooperative tasks: control design and analysis

Jonathan Obregón * **América Morales** *

* *Robotics and Advanced Manufacturing Program CINVESTAV-Saltillo, Ramos Arizpe, 25903, México (e-mail:j.s.obregon90@gmail.com).*

Abstract: In this paper we present a synchronization feedback control scheme for kinematic cooperative mobile manipulator robots performing a global task of manipulation and transportation, where an object is taken to a desired 6D pose. We define the motion of the object which is translated to each end-effector as desired coordinates, that trajectory is generated on-line and is a function of every robot in the scene. The group of robots follow the object's motion where the control of each robot is also a function of every other robot, meanwhile they are constrained to a common object. This leads to a complicated interaction scheme, so we thoroughly analyze the effects of the interactions that alter the behavior of each robot, and determine under what conditions those effects help to increase the performance of the proposed control scheme, such that the motions are coordinated to minimize the total energy and stabilize themselves as the common object reaches its objective. For this we performed a stability analysis and tested our control scheme through simulations.

Copyright © 2020 The Authors. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0)

Keywords: Synchronization control, cooperative robots, hierarchical tasks, constrained motion, stability analysis.

1. INTRODUCTION

Among the robotic tasks that require coordination of robots, cooperative manipulation/transportation of objects are within the most challenging ones. Such tasks requires the robot to come into contact with a common linkage, which occurs when the robots grasp the object, thus everything merges into a single complex robot, where the task becomes a regulation or trajectory tracking one for the object. This is undoubtedly a challenge that involves complex motion, which has been addressed by motion generation algorithms based on the inverse kinematics of the robots, such as Escande et al. (2014), Pek et al. (2016), also based on motion planning Alonso et al. (2015). Moreover, the dynamics of the robots are usually considered in motion planning algorithms Bouyarmane et al. (2019), Saab et al. (2013). Closed-loop controllers have also been proposed with inverse kinematics and dynamics resolution for tasks execution Erhart and Hirche (2015), also within a scheme of synchronization as recently in Sieber and Hirche (2019). Other works that dealt with synchronization are Sira-Ramirez and Castro-Linares (2010) and Sun et al. (2009) but for mobile robots with no manipulation. The inclusion of the dynamics in the closed-loop analysis and motion generation algorithms is of great interest, because of the interaction forces that occur during the cooperation once the robots are in contact. Those forces can be observed and controlled through the analysis of physical contacts Khatib (1995) and grasping modeling Williams and Khatib (1993). Intuitively, undesired interaction forces arise and become greater as errors in motion tasks appear, due to model uncertainties and disturbances (a common issue in dynamic-model-based torque-controls), hence closed-loop robust/adaptive controllers are best suited for this Liu and Arimoto (1998), Sanchez-Sanchez et al. (2017). However, several assumptions, which limit the functionality of those controllers, are considered, such as quasi-static motion, manipulation tasks with inertial robots and with limited motion, such as controlling only end-effector's position through point-to-point trajectories of short distances, whereas special care is given to the orientation control because of the attitude's representation singularities. As a result, some of the works related to complex motion control, rely on kinematic-based solutions, which so far, are enough for certain applications, such as swarms of mobile robots Arechavaleta et al. (2017). From works such as with Antonelli et al. (2008), where the inverse hierarchical kinematics problem is addressed and an stability analysis was provided, and in Jarquin et al. (2013) where inequality tasks are treated as tasks with priority transitions, it can be acknowledged that kinematic approaches are enough to achieve complex coordinated whole-body motion with hierarchal tasks. Moreover, the interaction forces can also be treated with kinematics based controls, provided that the force input can be measured, see Gracia et al. (2018), where the hierarchical-inverse-kinematics (HIK) was solved with quadratic programming using force sensors at the contacts.

It can be seen how for cooperative tasks in mobile manipulators, synchronization schemes for feedback control are not as commonly used as path-planning algorithms and optimization-based controls. Therefore advantages, such as on-line feedback control and faster computation times are lost. In this work the whole-motion of omnidirectional-mobile-manipulators holding an object is controlled by a HIK solver based on Jacobians and null-space projectors, with a linear synchronization control scheme. In other words, we consider the end-effectors as rigid bodies with 6D pose coordinates, constrained by their robots kinematics and by the common object, then we synchronize their motion. To achieve this, we impose the object a motion behavior which is translated to the end-effectors as desired trajectories. To track those trajectories the controller was designed

⋆ CONACyT support acknowledgment.

2405-8963 Copyright © 2020 The Authors. This is an open access article under the CC BY-NC-ND license.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2020.12.2135

by considering how the coupled systems affect each other, for this we performed a stability analysis that acquaints all interconnections. We simulated two mobile manipulators performing a manipulation and transportation task using the proposed controller, and observed how end-effectors' pose errors are minimal when the object performs complicated maneuvers.

2. SYNCHRONIZATION CONTROL DESIGN

2.1 Hierarchical task space control

The forward kinematics defines the end-effector pose $x(q) = f(q) \in \mathbb{R}^m$, which is composed by position and orientation coordinates $x(q) = (p(q), o(q))$, where the position $p \in \mathbb{R}^3$, and the orientation is parametrized with unit quaternions $o \in \mathbb{S}^3$. Thus $x \in \mathbb{R}^3 \times \mathbb{S}^3$ for a fully constrictive task, or $x(q) \in \mathbb{R}^m$, where m is the dimension of the constraint, $f(q)$ represents the direct-kinematics of the robot, which is a function of its joint position $q \in \mathbb{R}^n$, where n defines the degrees-of-freedom of the robot.
The first order differential kinematics defines the behavior of the end-effector pose over time:

$$\dot{x} = J\dot{q} \tag{1}$$

where $J = \frac{\partial x}{\partial q} \in \mathbb{R}^{m\times n}$ is the analytic Jacobian of the end-effector, and $\dot{q} \in \mathbb{R}^n$ is the generalized velocity or joint velocity. A task i is defined by the desired end-effector pose $x_d \in \mathbb{R}^m$. The task error becomes:

$$e_i = x_i(q) - x_{d_i} \tag{2}$$

The error behavior is imposed by:

$$\dot{e}_i = -\lambda_i e_i + \dot{x}_{d_i} \tag{3}$$

where $\lambda > 0 \in \mathbb{R}^{m\times m}$, $\dot{e}$ is the operational velocity control input, which is mapped to the joint-space through:

$$\dot{q}_i = J_i^+(\dot{e}_i + \dot{x}_d) \tag{4}$$

where $\dot{q} \triangleq u$ is the control in generalized coordinates and J^+ is the generalized inverse of the Jacobian matrix. If the robot is redundant i.e., $m < n$, secondary tasks can be included through the use of a null-space projector of the Jacobian of the main task:

$$N_i = I - J_i^+ J_i \tag{5}$$

Therefore the control input of secondary tasks are projected on the null-space projectors of higher priority tasks:

$$u_i = (J_i N_{i-1})^+(\dot{e}_i - J_i u_{i-1}) \tag{6}$$

Assuming that $J_i \in \mathbb{R}^{m\times n}$ is no singular, design $\dot{e}_i$ such that: $e_i = 0$, $\dot{q}_i = 0$, as $t \to \infty$. It is assumed that there are r redundant robots to perform any of the p tasks, such that $m < n$ holds for all robots.

2.2 Virtual constraint and tasks design

The virtual constraint is the envelop of the object to be manipulated and transported, in case of geometrically simple objects, a sphere can represent this constraint:

$$\varphi_j = (x_j - x_c)^2 + (y_j - y_c)^2 + (z_j - z_c)^2 - r^2 = 0 \tag{7}$$

where $\varphi_j \in \mathbb{R}$, $p_j = (x_j, y_j, z_j)$ are position coordinates of the robot j, $p_c = (x_c, y_c, z_c)$ are position coordinates of the sphere's center and r is its radius. Let us define the constraint Jacobian as $J_{c_j} = \frac{\partial \varphi_j}{\partial x_j}\frac{\partial x_j}{\partial q_j} = J_{\varphi_j} J_j$, where $J_{\varphi_j} \in \mathbb{R}^m$ is the operational constraint Jacobian, and J_j is the analytic Jacobian of the end-effector j. The orientation of the sphere is given by the unit quaternion $\epsilon_c = [\epsilon_{0_c}\ \epsilon_{v_c}]$, with the scalar real part ϵ_{0_c} and the vectorial imaginary part ϵ_{v_c}. Also the orientation of the end-effector j is the quaternion $\epsilon_j = [\epsilon_{0_j}\ \epsilon_{v_j}]$. Since the goal is to bring the object to a desired pose, we can define the global task in function of the sphere's center desired and current pose: $e_c = f(x_c(q), x_{c_d})$. Partitioning e_c into position error e_{p_c} and orientation error e_{ϵ_c}, we compute each as:

$$e_{p_c} = p_c - p_{c_d}, \quad e_{\epsilon_c} = \epsilon_c \otimes \epsilon'_{c_d} \tag{8}$$

where $\epsilon' = [\epsilon_0\ \ -\epsilon_v]$ is the conjugate of ϵ, and $\otimes$ defines the quaternion Hamiltonian-product that represents rotation of quaternions, which as error, it can be interpreted as the rotation needed to reach $\epsilon_0 = 1$ and $\epsilon_v = [0,\ 0,\ 0]$. Controlling ϵ_v is enough for the orientation task.

Now let us define the local tasks for each individual robot, with one defined in (7). For the end-effectors to reach a specified point on the sphere surface we define the desired end-effectors' pose coordinates by the following transformation:

$$p_{j_d} = p_c + R(\epsilon_c)d_j \tag{9}$$

where $R(\epsilon_c)$ is the quaternion-derived rotation matrix, $d_j = l_j r$ is the distance between the sphere's center and its surface's desired point for the end-effector j, where $l_j \in \mathbb{R}^3$ is a unit vector of the distance direction in the sphere center's frame. The rotation of the end-effectors must correspond to the sphere's rotation. However, a desired initial rotation of end-effectors can be performed by an offset quaternion ϵ_{j_o}, such that end-effectors' local z-axis aligns with d_j.

$$\epsilon_{j_d} = \epsilon_c \otimes \epsilon'_{j_o} \tag{10}$$

The desired linear velocity for the end-effectors is obtained by differentiating (9), which gives

$$\dot{p}_{j_d} = \dot{p}_c + \omega_c \times d_j \tag{11}$$

where $[\omega_c\times] \in so(3)$ is angular velocity of the sphere's local coordinates frame. In contrast, the quaternion orientation rate is expressed as $\dot{\epsilon}_c$, where $\omega_c \neq \dot{\epsilon}_c$. However, the relationship between them is given via $\omega_c = J_\epsilon \dot{\epsilon}_c$ and $\dot{\epsilon}_c = J_\epsilon^+ \omega_c$, with

$$J_\epsilon = 2\left[-\epsilon_v\ \ \epsilon_0 I + [\epsilon_v\times]\right], \quad J_\epsilon^+ = \frac{1}{2}\begin{bmatrix} -\epsilon_v^T \\ \epsilon_0 I - [\epsilon_v\times] \end{bmatrix} \tag{12}$$

Therefore (11) can alternatively be computed as

$$\dot{p}_{j_d} = \dot{p}_c + (J_\epsilon \dot{\epsilon}_c) \times d_j \tag{13}$$

The end-effectors' desired rate of the orientation expressed with quaternions are computed as

$$\dot{\epsilon}_{j_d} = \dot{\epsilon}_c \tag{14}$$

The position and orientation errors can then be computed as

$$e_{p_j} = p_j - p_{j_d} \tag{15}$$

$$e_{\epsilon_j} = \epsilon_j \otimes \epsilon'_{j_d} \tag{16}$$

3. OPERATIONAL-SPACE SYNCHRONIZATION CONTROL

The control of the sphere's pose can be decomposed in two parts, with the first one defining the sphere's behavior as exponentially diminishing motion

$$\begin{bmatrix}\dot{e}_{p_{c_1}}\\ \dot{e}_{\epsilon_{c_1}}\end{bmatrix} = -\begin{bmatrix}K_{a_p} & 0\\ 0 & K_{a_\epsilon}\end{bmatrix}\begin{bmatrix}e_{p_c}\\ e_{\epsilon_c}\end{bmatrix}$$
$$\dot{e}_{c_1} = -K_a e_c \tag{17}$$

and the second part including the coupling errors of r end-effectors

$$\dot{e}_{p_{c_2}} = -K_{b_p}\left(re_{p_c} - \sum_{j=1}^{r} e_{p_j}\right) \tag{18}$$

$$\dot{e}_{\epsilon_{c_2}} = -\sum_{j=1}^{r} K_{b_\epsilon}\left(e_{\epsilon_c}\otimes e'_{\epsilon_j}\right) \tag{19}$$

where $K_a, K_b \in \mathbb{R}^{m\times m}$ are positive diagonal matrices. Let $\dot{e}_{c_2} = \begin{bmatrix}\dot{e}_{p_{c_2}} & \dot{e}_{\epsilon_{c_2}}\end{bmatrix}^T$. Then we have

$$\dot{e}_c = \dot{e}_{c_1} + \dot{e}_{c_2} \tag{20}$$

On the other hand, the controller that keeps the j-th end-effector in the constraint is

$$\dot{\varphi}_j = -C\varphi_j + J_{\varphi_j}\dot{p}_c \tag{21}$$

where $j = 1, 2\ldots r$, $C \in \mathbb{R}$ is the constraint stabilizer, which is a positive constant, $J_{\varphi_j}\dot{p}_c \triangleq \dot{\varphi}_{j_d}$ tracks the changes in sphere's position. The operational control of the j-th end-effector's position is computed as

$$\dot{e}_{p_j} = -G_{a_p}e_{p_j} - G_{b_p}\left((r-1)e_{p_j} - \sum_{k\neq j}^{r-1} e_{p_k}\right) + \dot{p}_{j_d} \tag{22}$$

and the orientation control of the j-th end-effector is computed as

$$\dot{e}_{\epsilon_j} = -G_{a_\epsilon}e_{\epsilon_j} - \sum_{j=1}^{r-1} G_{b_\epsilon}\left(e_{\epsilon_c}\otimes e'_{\epsilon_j}\right) + \dot{\epsilon}_{j_d} \tag{23}$$

where $G_a, G_b \in \mathbb{R}^{m\times m}$ are positive diagonal matrices. The terms $\dot{p}_{j_d}$ and $\dot{\epsilon}_{j_d}$ from (13) and (14) respectively, track the changes in sphere's position and orientation, thus become the desired linear and angular velocity for the end-effectors. Similarly to (5) and (6) we map the operational space control into the generalized-coordinates space of each end-effector j

$$\begin{aligned}\dot{q}_{c_j} &= J_{c_j}^{+}\dot{\varphi}_j\\ \dot{q}_{x_j} &= \left[J_j N_{c_j}\right]^{+}\left(\dot{e}_{x_j} - J_j\dot{q}_{c_j}\right)\\ \dot{q}_j &= \dot{q}_{c_j} + \dot{q}_{x_j}\end{aligned} \tag{24}$$

4. STABILITY ANALYSIS

Notice that (17)-(19) define the desired trajectories for all end-effectors. For the position task it can be seen that $\dot{e}_{p_{c_1}}$ alone is stable as long as K_{a_p} is positive. However considering that all robots with the sphere are within an unidirectional communication structure, we get $\dot{e}_{p_c} = \dot{e}_{p_{c_1}} + \dot{e}_{p_{c_2}}$, which is analyzed as follows

$$\dot{e}_{p_c} = -(K_a + rK_b)e_{p_c} + K_b\sum_{j=1}^{r} e_{p_j} \tag{25}$$

with K_a, $K_b > 0$. The Lyapunov function is $V_{p_c} = \frac{1}{2}e_{p_c}^T e_{p_c}$ and its derivative becomes

$$\dot{V}_{p_c} = -e_{p_c}^T(K_a + rK_b)e_{p_c} + e_{p_c}^T K_b\sum_{j=1}^{r} e_{p_j} \tag{26}$$

then we need to fulfill

$$\|K_a + rK_b\|\|e_{p_c}\| > \|K_b\|\|e_{p_c}^T\sum_{j=1}^{r} e_{p_j}\| \tag{27}$$

Therefore we establish the following bounding condition

$$e_{p_c}^T K_b\sum_{j=1}^{r} e_{p_j} \leq \lambda_p \leq e_{p_c}^T(K_a + rK_b)e_{p_c} \tag{28}$$

which leads to

$$\dot{V}_{p_c} \leq -e_{p_c}^T(K_a + rK_b)e_{p_c} + \lambda_p \tag{29}$$

For this to hold, we depend on λ_p, which contains all e_{p_j}. Therefore (29) is not guaranteed even if $K_a > rK_b$, most notably when $e_{p_c} \approx 0$ and $re_{p_c}^T e_{p_c} < e_{p_c}^T\sum_{j=1}^{r} e_{p_j}$. Technically this situation means that the sphere has already converged and slowed down, but some e_{p_j} do not converge yet, so the sphere moves ($\dot{V}_{c_p}$ increases), to adjust p_{j_d} such that $e_{p_j} \to 0$, but if such motion disturb some e_{p_k}, then $\dot{V}_{c_p}$ increases again and so on. This occurs given the dependencies between the systems. Therefore, if we ensure that $e_{p_j} \to 0$ faster than $e_{p_c} \to 0$, then (29) always holds. We ensure this later on.

The orientation control variables are parametrized with unit quaternions, such that we avoid representation singularities. Similarly, from (23), the orientation of the sphere with the coupling terms is:

$$\dot{e}_{\epsilon_c} = -K_a e_{\epsilon_c} - K_b(e_{\epsilon_c}\otimes e'_{\epsilon_1})\ldots - K_b(e_{\epsilon_c}\otimes e'_{\epsilon_r}) \tag{30}$$

with $e_{\epsilon_c} = \epsilon_c \otimes \epsilon'_{c_d}$ and $e_{\epsilon_j} = \epsilon_j \otimes \epsilon'_{j_d}$. A quaternion product $\epsilon_c \otimes \epsilon'_r$ can be written as

$$\epsilon_c \otimes \epsilon'_r = Q(\epsilon_c)\epsilon_r \tag{31}$$

where

$$Q(\epsilon) = \begin{bmatrix}\epsilon_0 & -\epsilon_v^T\\ \epsilon_v & \epsilon_0 I - [\epsilon_v\times]\end{bmatrix} \in \mathbb{R}^{4\times 4} \tag{32}$$

thus (30) becomes

$$\dot{e}_{\epsilon_c} = -K_a e_{\epsilon_c} - K_b Q(e_{\epsilon_c})(e_{\epsilon_1} + e_{\epsilon_2}\cdots + e_{\epsilon_r}) \tag{33}$$

Then the Lyapunov's function is $V_{\epsilon_c} = \frac{1}{2}e_{\epsilon_c}^T e_{\epsilon_c}$ and its derivative becomes

$$\dot{V}_{\epsilon_c} = e_{\epsilon_c}^T\dot{e}_{\epsilon_c} = -e_{\epsilon_c}^T K_a e_{\epsilon_c} - e_{\epsilon_c}^T K_b Q(e_{\epsilon_c})\sum_{j=1}^{r} e_{\epsilon_j} \tag{34}$$

where K_a, $K_b > \mathbb{R}^{m\times m}$ are diagonal positive matrices. The operator $Q(\epsilon)$ has useful properties: if $\|\epsilon\| = 1$ then $Q(\epsilon)Q(\epsilon)^T = 1$, also $\det(Q(\epsilon)) = 1$. Also for two different unit quaternions i, j, it holds that $\epsilon_i^T \epsilon_j < 1$. The coupling error terms in (34) include the orientation errors of the sphere (8) and end-effectors' (16), which are also quaternion products with same properties. Now for the first term of (34) we can establish that

$$\lambda_{min_{(K_a)}}\|e_{\epsilon_c}\| \leq e_{\epsilon_c}^T K_a e_{\epsilon_c} \leq \lambda_{max_{(K_a)}}\|e_{\epsilon_c}\| \tag{35}$$

Then, let us set the following bound:

$$e_{\epsilon_c}^T Q(e_{\epsilon_c}) \sum_{j=1}^{r} e_{\epsilon_j} \leq \|e_{\epsilon_c}^T\| \|Q(e_{\epsilon_c}) \sum_{j=1}^{r} e_{\epsilon_j}\| \triangleq \xi \tag{36}$$

where since $\|Q(\epsilon)\| = 1$, then $\|Q(e_{\epsilon_c}) \sum_{j=1}^{r} e_{\epsilon_j}\| = \| \sum_{j=1}^{r} e_{\epsilon_j}\|$. Therefore we get

$$\|e_{\epsilon_c}^T\| \| \sum_{j=1}^{r} e_{\epsilon_j}\| \triangleq \xi \tag{37}$$

Note that if e_{ϵ_c} and e_{ϵ_j} are unitary we directly get $r \triangleq \xi$, with this we can establish the bounds as

$$-\xi\lambda_{max_{(K_b)}} \leq e_{\epsilon_c}^T K_b Q(e_{\epsilon_c}) \sum_{j=1}^{r} e_{\epsilon_j} \leq \xi\lambda_{max_{(K_b)}} \tag{38}$$

Note that the lower bound exists since we can have negative quaternions, however they have the same bound ξ. Finally, the derivative of Lyapunov's function becomes

$$\dot{V}_{\epsilon_c} \leq -\lambda_{max_{(K_a)}}\|e_{\epsilon_c}\| + \xi\lambda_{max_{(K_b)}} \tag{39}$$

which means that

$$\lambda_{max_{(K_a)}}\|e_{\epsilon_c}\| > \xi\lambda_{max_{(K_b)}} \tag{40}$$

therefore $K_a > \xi K_b$ or $K_a > rK_b$.

On the other hand, the control term for the end-effectors position to track the trajectories described by the sphere's motion is (22), which follows an all-to-all communication structure:

$$\dot{e}_p = -Ge_p + \dot{p}_d \tag{41}$$

with $\dot{e}_p = [\dot{e}_{p_1} \ \dots \ \dot{e}_{p_r}] \in \mathbb{R}^{rm}$, $e_p \in \mathbb{R}^{rm}$, $\dot{p}_d \in \mathbb{R}^{rm}$ and

$$G = -\begin{bmatrix} G_a + (r-1)G_b & \cdots & -G_b \\ \vdots & \ddots & \vdots \\ -G_b & \cdots & G_a + (r-1)G_b \end{bmatrix}, \tag{42}$$

where $G \in \mathbb{R}^{r(m\times m)}$ is the coupling matrix, whose r eigenvalues are negative, i.e. is Hurwitz if

$$G_a > rG_b \tag{43}$$

where G_a and G_b are positive diagonal matrices. Then, the solution of (42) is

$$\dot{e}_p = e_{p_0}\exp(Gt) \tag{44}$$

where e_p, $\dot{e}_p \in \mathbb{R}^{rm}$ are stacked vectors of errors, $e_{p_0} \in \mathbb{R}^{rm}$ are the initial conditions of the r robots for the task i. The Lyapunov function is $V_p = \frac{1}{2}e_p^T e_p$ and its derivative becomes

$$\dot{V}_p = e_p^T \dot{e}_p = -e_p^T G e_p + e_p^T \dot{p}_d \tag{45}$$

where (13) describe the desired velocities: $\dot{p}_d = \dot{p}_c + (J_\epsilon \dot{\epsilon}_c) \times d_j$, from which we already verified that $\dot{p}_c$, $\dot{\epsilon}_c \to 0$ as $t \to \infty$. However, notice that $\dot{p}_d$ may become larger enough such that (45) does not hold. To prevent this let see how $\dot{p}_d$ from (22) affects the Lyapunov's function of the j-th robot, which we rewrite as follows:

$$\dot{V}_{p_j} = -\beta_{j_1} - \beta_{j_2} + \beta_{j_3} + \beta_{j_4} \tag{46}$$

with

$$\beta_{j_1} \triangleq e_{p_j}^T G_{a_p} e_{p_j} > 0 \tag{47}$$

$$\beta_{j_2} \triangleq e_{p_j}^T G_{b_p}(r-1) e_{p_j} > 0 \tag{48}$$

$$\beta_{j_3} \triangleq e_{p_j}^T G_{b_p} \sum_{k\neq j}^{r-1} e_{p_k} \tag{49}$$

$$\beta_{j_4} \triangleq e_{p_j}^T \dot{p}_{j_d} \tag{50}$$

it is interesting to compare β_{j_2} and β_{j_3}, since in both terms the errors are scaled by the same constant matrix and depend on the same number of robots. From (46) we can only guarantee that $\beta_{j_1} > \beta_{j_3} - \beta_{j_2}$ by setting $G_a > rG_b$, which is consistent with the Hurwitz condition (43), then $\beta_{j_1} + \beta_{j_2} - \beta_{j_3} > 0$. On the other hand, β_{j_4} is a vanishing term dependent on the desired trajectory, it must be verified that $\beta_{j_1} + \beta_{j_2} - \beta_{j_3} > \beta_{j_4}$. Then let $\beta_{j_1} + \beta_{j_2} - \beta_{j_3} \triangleq \eta_j$, such that

$$\eta_j > e_{p_j}^T \dot{p}_{j_d} = \eta_j > e_{p_j}^T (\dot{p}_c + (J_\epsilon \dot{\epsilon}_c) \times d_j)$$

substituting $\dot{p}_c$ with (25)

$$\eta_j > e_{p_j}^T \left(-(K_a + rK_b)e_{p_c} + K_b \sum_{j=1}^{r} e_{p_j} + (J_\epsilon \dot{\epsilon}_c) \times d_j \right)$$

Now the following terms arise

$$\alpha_{j_1} = e_{p_j}^T (K_a + rK_b) e_{p_c} \tag{51}$$

$$\alpha_{j_2} = e_{p_j}^T K_b \sum_{j=1}^{r} e_{p_j} \tag{52}$$

$$\alpha_{j_3} = e_{p_j}^T (J_\epsilon \dot{\epsilon}_c) \times d_j \tag{53}$$

from (33) and (40) we see that $\alpha_{j_3} < 0$. Let us compare α_{j_2} with terms from η, then it is clear that

$$e_{p_j}^T (G_{a_p} + G_{b_p}(r-1)) e_{p_j} > e_{p_j}^T K_b \sum_{j=1}^{r} e_{p_j} \tag{54}$$

then let us compare it with α_{j_2}

$$e_{p_j}^T (G_{a_p} + G_{b_p}(r-1)) e_{p_j} > e_{p_j}^T (K_a + rK_b) e_{p_c} \tag{55}$$

which is true if

$$(G_{a_p} + G_{b_p}(r-1)) > (K_a + rK_b) \tag{56}$$

this guarantees that $\beta_{j_1} + \beta_{j_2} - \beta_{j_3} > \beta_{j_4}$, so (45) is fulfilled. Note that this condition implies that $\dot{e}_{p_j} \to 0$ faster than $\dot{e}_{p_c} \to 0$, which was the missing condition for (29) to hold. The control term for the end-effectors orientation to track the trajectories

described by the sphere's motion is (23), and follows the same all-to-all communication structure. From (33) we have:

$$\dot{e}_{\epsilon_j} = -G_a e_{\epsilon_j} - G_b Q(e_{\epsilon_j}) \sum_{k \neq j}^{r-1} e_{\epsilon_k} + \dot{\epsilon}_{j_d} \tag{57}$$

Thus the Lyapunov's function is $V_{j_\epsilon} = \frac{1}{2} e_{\epsilon_j}^T e_{\epsilon_j}$ and its derivative becomes

$$\dot{V}_{j_\epsilon} = -e_{\epsilon_j}^T G_a e_{\epsilon_j} - e_{\epsilon_j}^T G_b Q(e_{\epsilon_j}) \sum_{k \neq j}^{r-1} e_{\epsilon_k} + e_{\epsilon_j}^T \dot{\epsilon}_{j_d} \tag{58}$$

where $\dot{\epsilon}_{j_d} = \dot{\epsilon}_c$ as defined in (14), its form is given in (33). Then we have

$$e_{\epsilon_j}^T \dot{\epsilon}_{j_d} = -e_{\epsilon_j}^T K_a e_{\epsilon_c} - e_{\epsilon_j}^T K_b Q(e_{\epsilon_c}) \sum_{j=1}^{r} e_{\epsilon_j} \tag{59}$$

In this case we can directly compare the terms from (58), (59), such that we find a condition that ensures

$$e_{\epsilon_j}^T G_a e_{\epsilon_j} + e_{\epsilon_j}^T G_b Q(e_{\epsilon_j}) \sum_{k \neq j}^{r-1} e_{\epsilon_k} > e_{\epsilon_j}^T \dot{\epsilon}_{j_d} \tag{60}$$

which leads to the conclusion that $G_a > K_a$ and $G_b > K_b$, similar to (56). Now, let us follow the same procedures that get to (39), such that, for each j-th robot, we arrive to a bound for the second term of (58)

$$\dot{V}_{j_\epsilon} \leq -\lambda_{max_{(G_a)}} \|e_{\epsilon_j}\| + \|e_{\epsilon_j}^T\| \| \sum_{k=1}^{r-1} e_{\epsilon_k} \| \lambda_{max_{(G_b)}} + e_{\epsilon_j}^T \dot{\epsilon}_{j_d} \tag{61}$$

Finally we can sum the Lyapunov's functions derivatives:

$$\dot{V} = \dot{V}_{c_p} + \dot{V}_{c_\epsilon} + \dot{V}_p + \sum_{j=1}^{r} \dot{V}_{j_\epsilon} \leq 0 \tag{62}$$

which fulfills that $\dot{V} < 0, \forall\, e_{p_c}, e_{p_j}, e_{\epsilon_c}, e_{\epsilon_j} \in D - \{0\}$

5. SIMULATION RESULTS

We performed a numerical simulation of manipulation and transportation of an object, using two robots controlled through the proposed control scheme. The common object is an sphere with radius $r = 1[m]$. The sphere reached the following sequence desired coordinates (see Figure 1):

t: time [s]	p_d: position [m]	ϵ_d: orientation
$0 \leq t < 2$	$[2\,,0\,,1]$	$[0.7071\,,0\,,0\,,0.7071]$
$2 \leq t < 4$	$[0\,,0\,,3]$	$[1\,,0\,,0\,,0]$
$4 \leq t < 6$	$[0\,,0\,,2]$	$[0.9239\,,0\,,0.3827\,,0]$
$6 \leq t < 8$	$[-0.5\,,2\,,2]$	$[0.6533\,,0.2706\,,0.2706\,,0.6533]$
$t > 8$	$[2\,,0\,,1]$	$[0.7071\,,0\,,0\,,0.7071]$

The initial sphere's and end-effectors' coordinates with respect to the world's frame are:

Robot	p_0: position [m]	ϵ_0: orientation
1	$[-1\,,0\,,4.5]$	$[1\,,0\,,0\,,0]$
2	$[4\,,0\,,4.5]$	$[0\,,0\,,0\,,1]$
Sphere	$[2\,,0\,,1]$	$[0.7071\,,0\,,0\,,0.7071]$

Figure 2 shows the sphere pose errors, which are also trajectories for end-effectors. Note that when a quaternion error converges, it becomes $e_{\epsilon_0} \to 1$ and $e_{\epsilon_v} \to [0\,,0\,,0]$. Notice that the desired coordinates and initial coordinates indicate large displacements in short time, this was done to experiment with a varied succession of desired-coordinates taking a reasonable amount of time. This in turn required larger gains for faster convergence, we used the same gains for the position and orientation control, and also are the same for both robots: We chose $C = 10$, $K_a = 15$, $K_b = 10$, $G_a = 45$, $G_b = 20$. Figure 3 shows the control signals (joint velocities) of the end-effectors with the largest magnitudes at the beginning, because of the large initial position errors. Afterwards, the signals remain small and with smooth changes that occurred as the desired sphere's pose changed. In Figure 4 we see the errors at the sphere's contact surface, which indicates that once the end-effectors reached the sphere, they were fixed at its surface without separating. Figure 5 shows the results of having a trajectory and controller that accounts the interaction within the robots, as the position and orientation errors remain very close to zero even after the abrupt changes in sphere's desired coordinates.

6. CONCLUSIONS

We performed a stability analysis for a proposed control scheme based on HIK and linear synchronization approaches. As a result we deduced the appropriate control values such to achieve a synchronized equilibrium of all interacting systems. Testing the controller on simulations showed good performance when coupling errors are included in the control, since the end-effectors' pose errors where negligible even though the object performed complicated maneuvers. Having small errors at cooperative tasks is of crucial importance, as undesired interaction effects can result in damage to the robots and to the manipulated object. Nonetheless, such errors are difficult to prevent, specially at complicated or aggressive maneuvers. The use of torque-controlled robots and force sensors help to alleviate the effect of such errors, but for kinematic velocity-controlled robots without force sensors, the cooperative possibilities are quite limited, yet approaches like the one presented in this work extends the usefulness of velocity-controlled robots for cooperative tasks.

ACKNOWLEDGEMENTS

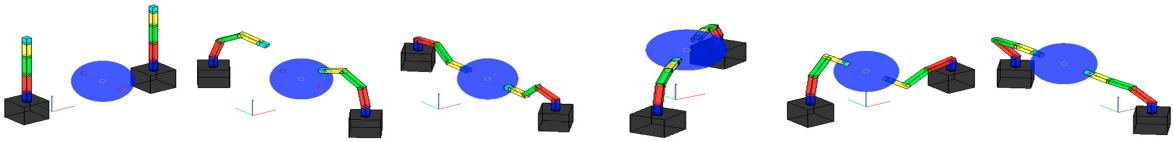
The authors would like to thank CONACyT for the financial support through this work which is called "Análisis, control y sincronización de sistemas complejos con interconexiones dinámicas y acoplamientos flexibles, A1-S-26123". The first author thanks CONACyT for his PhD scholarship.

REFERENCES

Alonso, J., Knepper, R., Siegwart, R., and Rus, D. (2015). Local motion planning for collaborative multi-robot manipulation of deformable objects. *IEEE International Conference on Robotics and Automation*, 5495–5502.

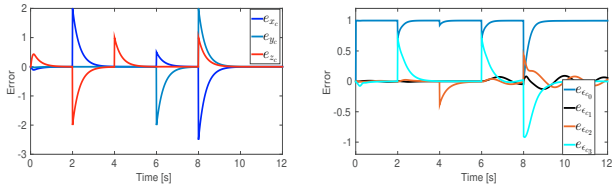
Antonelli, G., Arrichiello, F., and Chiaverini, S. (2008). The null-space-based behavioral control for autonomous robotic systems. *Intelligent Service Robotics*, 1(1), 27–39.

Arechavaleta, G., Morales-Diaz, A., Perez-Villeda, H.M., and Castelan, M. (2017). Hierarchical task-based control of multirobot systems with terminal attractors. *IEEE Transactions on Control Systems Technology*, 25(1), 334–341.



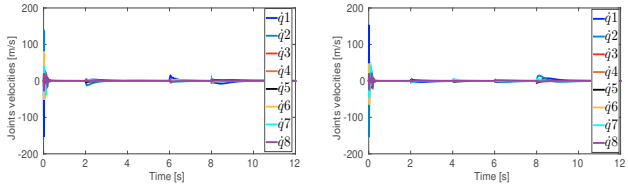
(a) Initial position at $t = 0$ (b) Moving towards target: $0 \leq t < 2$ (c) Robots holding the sphere: $0 \leq t < 2$ (d) Robots turning: $2 \leq t < 4$ (e) Robots tilting the sphere: $4 \leq t < 6$ (f) Robots displacing the sphere: $6 \leq t < 8$

Fig. 1. Two mobile manipulators being synchronized through a common linkage



(a) Sphere's position error (b) Sphere's orientation error

Fig. 2. Position and orientation errors of the sphere



(a) Generalized control of robot 1 (b) Generalized control of robot 2

Fig. 3. Generalized controls (joint-velocities) of both robots

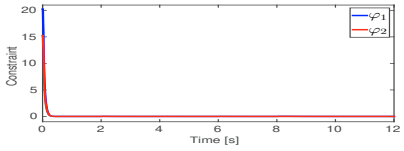
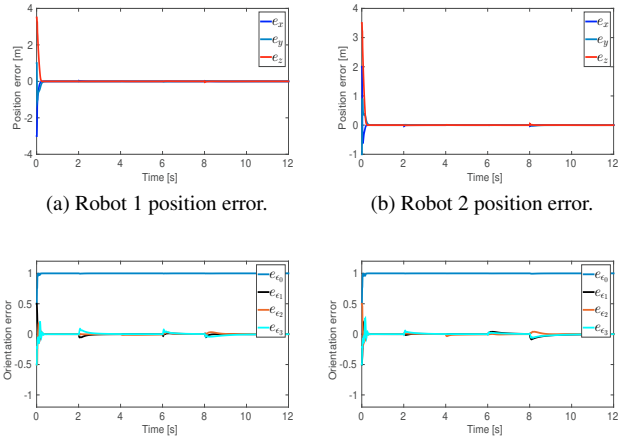


Fig. 4. Constraint errors of robot 1 and robot 2



(a) Robot 1 position error. (b) Robot 2 position error.

(c) Robot 1 orientation error. (d) Robot 2 orientation error.

Fig. 5. Position and orientation errors of both end-effectors when using the proposed synchronization approach within the HIK.

Bouyarmane, K., Chappellet, K., Vaillant, J., and Kheddar, A. (2019). Quadratic programming for multirobot and task-space force control. *IEEE Transactions on Robotics*, 35(1).

Erhart, S. and Hirche, S. (2015). Internal force analysis and load distribution for cooperative multi-robot manipulation. *IEEE Transactions on Robotics*, 31(5), 1238–1243.

Escande, A., Mansard, N., and Wieber, P.B. (2014). Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *The International Journal of Robotics Research*, 33(7), 1006–1028.

Gracia, L., Solanes, J.E., Muoz-Benavent, P., Miro, J.V., Perez-Vidal, C., and Tornero, J. (2018). Adaptive sliding mode control for robotic surface treatment using force feedback. *Mechatronics*, 52, 102 – 118.

Jarquin, G., Escande, A., Arechavaleta, G., Moulard, T., Yoshida, E., and Parra-Vega, V. (2013). Real-time smooth task transitions for hierarchical inverse kinematics. In *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 528–533.

Khatib, O. (1995). Inertial properties in robotic manipulation: An object-level framework. *The international journal of robotics research*, 14(1), 19–36.

Liu, Y.H. and Arimoto, S. (1998). Decentralized adaptive and nonadaptive position/force controllers for redundant manipulators in cooperations. *The International Journal of Robotics Research*, 17(3), 232–247.

Pek, C., Muxfeldt, A., and Kubus, D. (2016). Simplifying synchronization in cooperative robot tasks an enhancement of the manipulation primitive paradigm.

Saab, L., Ramos, O.E., Keith, F., Mansard, N., Soures, P., and Fourquet, J. (2013). Dynamic whole-body motion generation under rigid contacts and other unilateral constraints. *IEEE Transactions on Robotics*, 29(2), 346–362.

Sanchez-Sanchez, P., Pliego-Jimenez, J., and Arteaga-Perez, M. (2017). A centralized hybrid position/force controller for cooperative robots with bounded torque inputs. In *2017 13th IEEE International Conference on Control Automation (ICCA)*, 839–844.

Sieber, D. and Hirche, S. (2019). Human-guided multirobot cooperative manipulation. *IEEE Transactions on Control Systems Technology*, 27(4), 1492–1509.

Sira-Ramirez, H. and Castro-Linares, R. (2010). Trajectory tracking for non-holonomic cars: A linear approach to controlled leader-follower formation. In *49th IEEE Conference on Decision and Control (CDC)*, 546–551.

Sun, D., Wang, C., Shang, W., and Feng, G. (2009). A synchronization approach to trajectory tracking of multiple mobile robots while maintaining time-varying formations. *IEEE Transactions on Robotics*, 25(5), 1074–1086.

Williams, D. and Khatib, O. (1993). The virtual linkage: a model for internal forces in multi-grasp manipulation. In *[1993] Proceedings IEEE International Conference on Robotics and Automation*, 1025–1030 vol.1.