Computed Torque for Restricted Lagrange-Euler Systems Via DAEs and LMIs

Julián León, David Vázquez, and Miguel Bernal

***Abstract*—This work is motivated by the need of an extension of the well-known computed-torque technique for restricted Lagrange-Euler systems, whose main difficulty lies on the incorporation of physical constraints into the model as well as in the control law. The proposal has two methodological novelties: on the one hand, reaction forces are added to the feedforward term; on the other hand, the feedback term is designed via linear matrix inequalities, which is computationally efficient and systematic. These improvements allow computed torque to deal with trajectory tracking problems in closed kinematic chains. Implementation issues via index-1 differential algebraic equations and recently appeared toolboxes are discussed. A fully reproducible case study on a constrained two-bar system is included to illustrate the effectiveness of the proposal. Moreover, a real-time implementation on a rotary arm system has been included to illustrate that even non-restricted plants are amenable to the novel technique.**

Link to graphical and video abstracts, and to code:
https://latamt.ieeer9.org/index.php/transactions/article/view/10501

***Index Terms*—Differential Algebraic Equations, Linear Matrix Inequalities, Computed Torque, Constrained Euler-Lagrange Systems.**

I. Introduction

THE study of open kinematic chains consisting of n prismatic or revolute joints, all actuated, is the basis of robot analysis and design. Lagrange's equation of movement proved that this sort of systems can be described as a set of 2nd-order nonlinear ordinary differential equations (ODEs) which can be arranged in a variety of models [1]: the Lagrange-Euler form which distinguishes inertia matrix, Coriolis, friction, and gravity vectors [2]; the position-velocity state-space representation which consists of a set of $2n$ 1st-order equations [3]. These systems are typically controlled using the computed-torque technique, which, through an error-tracking mechanism, decomposes the control input into two components: a feedforward term that compensates for the system's nonlinearities (following the principles of feedback linearization [4]), and a linear feedback term that employs position, velocity, and integral error signals to ensure that the tracking error asymptotically converges to zero (based on the well-known proportional-integral-derivative (PID) design [5]).

The associate editor coordinating the review of this manuscript and approving it for publication was Javier Moreno-Valenzuela (*Corresponding author: Miguel Ángel Bernal Reza*).

This work has been supported by the SECIHTI posdoc scholarship for CVU 713931 and SNII 37449, as well as the ITSON PROFAPI 2026-006.

J. Leór, D. Vázquez, and Miguel Ángel Bernal Reza are with the Department of Electrical and Electronics Engineering of the Sonora Institute of Technology, 5 de Febrero 818 Sur, C.P. 85000, Obregon City, Mexico (e-mails: jose.leon228914@potros.itson.edu.mx, david.vazquez@potros.itson.edu.mx, and miguel.bernal@itson.edu.mx).

Closed kinematic chains, on the other hand, resulting from joining the end effector to some fixed subset of the working space, are the basis of parallel robot analysis and design [6]. For this class of systems, several modeling approaches can be adopted, including the augmented Lagrangian method [7], which introduces penalty parameters and therefore departs from a natural physical representation; orthogonal projection techniques [8], which may eliminate or obscure the constraint reaction forces; and the Udwadia–Kalaba formulation [9], whose reliance on pseudoinverse operators complicates linearization and stability analysis. In contrast, the Euler–Lagrange formulation with Lagrange multipliers [10], provides an exact and physically meaningful representation in which constraint forces are explicitly preserved, making it particularly suitable for the systematic design and analysis of computed-torque controllers since position restrictions are incorporated into Lagrange's equation of movement by means of reaction forces, which include Lagrange multipliers; they also produce velocity and acceleration constraints which must be taken into account. Due to the restricted nature of these systems, their mathematical description in the form of differential algebraic equations (DAEs) [11] is only natural: equations involving time derivatives are the dynamical part; equations establishing static relationships are the algebraic part [12].

Problem Statement: Most DAEs arising in physical contexts are reducible to ODEs by means of a process called index reduction, where the DAE index is the number of times algebraic restrictions have to be differentiated for a subset of variables to be solved solely in terms of the differential ones; the algorithm performing this task is the Pantelides algorithm [13]. At least theoretically, once a DAE is index 1, a subset of variables can be solved in terms of the remaining ones to turn the whole representation into an ODE, subject to consistent initialization; in many cases, analytically solving these variables is impossible, but it can be done numerically via recently appeared software such as the DAE tools of the Symbolic Toolbox of MATLAB [14]. This work is concerned with a generalization of the computed-torque technique for restricted Lagrange-Euler representations which, by nature, are index-1 DAEs subject to the referred considerations.

Methodology: In the last decades, formulating control design problems as a set of linear matrix inequalities (LMIs) has been increasingly done because of the systematicness and numerical advantages they present [15]. Indeed, linear and nonlinear systems can be considered by means of convex modeling and control in a single framework; controllers, observers, and other structures can be solved in polynomial time without tuning or depending on the designer's ability

[16]. Well-established LMI solvers can be found implementing interior-point algorithms to efficiently solve them [17]. This work formulates its novel results as LMIs, as done in many contexts involving DAEs [18].

Contribution: This work generalizes the computed-torque technique to the class of restricted Lagrange-Euler systems, formulating its design conditions as LMIs. Details are given concerning simulation and implementation of the proposal as an index-1 DAE.

Organization: The contents of this work are organized as follows: preliminaries are given in Section II concerning the classical computed-torque technique and LMIs; the main result is proved in Section III, effectively extending the computed-torque technique via LMIs to restricted systems in the Lagrange-Euler form; Section IV considers a variety of implementation issues for the technique just developed using the index-1 DAE representation and toolboxes associated to them; a case study is presented in Section V with enough detail to guarantee reproducibility; Section VI provides a simple real-time implementation showing the methodology is not limited to physically restricted systems; finally, Section VII draws some conclusions about this work and discusses related problems to be considered in the future.

Notation: In matrix expressions, $(>)$ stands for positive-definiteness while $(<)$ stands for negative-definiteness. For inline expressions, $(*)$ stands for the transpose of the terms to the left, i.e., $A + (*) = A + A^T$. An identity matrix of adequate size is denoted as I while a zero matrix of adequate size is denoted as 0.

II. Preliminaries

In this section, the basics on computed torque for open kinematic chains and linear matrix inequalities are given as they constitute the basis for our proposal.

A. Computed Torque Technique

Unrestricted robot manipulators consisting in a kinematic chain of n links which may behave as prismatic or revolute joints has the following *Lagrange's equation of motion* [3]:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} + \frac{\partial D}{\partial \dot{q}} = \bar{\tau}, \tag{1}$$

where $q \in \mathbb{R}^n$ is the joint vector, $L \in \mathbb{R}$ is the Lagrangian (which is the difference between kinematic energy K and potential energy P, i.e. $L = K - P$), D is the system dissipation function, and $\bar{\tau} \in \mathbb{R}^n$ is the vector of generalized forces acting on the system.

Very often the system (1) is written in the following *Lagrange-Euler form*:

$$M(q)\ddot{q} + C(q,\dot{q}) + G(q) + F(\dot{q}) = \bar{\tau}, \tag{2}$$

where $M(q) \in \mathbb{R}^{n\times n}$ is the inertia matrix which satisfies the property $M(q) = M^T(q) > 0$, $C(q,\dot{q})$ is the Coriolis vector, $G(q)$ is the gravity vector, and $F(\dot{q})$ is the friction force. As it is well known, the following relationships are satisfied:

$$M(q)\ddot{q}+C(q,\dot{q})+G(q)=\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right)-\frac{\partial L}{\partial q},\ F(\dot{q})=\frac{\partial D}{\partial \dot{q}}.$$

Theorem 1. *[3] Let $q_d(t)$ be a $\mathscr{C}^2$ trajectory and $e \equiv q_d - q$ be the corresponding tracking error. The system with Lagrange-Euler model* (2) *(or, equivalently, Lagrange's equation of motion* (1)*), satisfies $\lim_{t\to\infty} e(t) = 0$ if $\bar{\tau} = M(q)(\ddot{q}_d - \bar{u}) + C(q,\dot{q}) + G(q) + F(\dot{q})$ with*

$$\bar{u} = -K_v\dot{e} - K_p e,\ K_v = diag(k_{v_i}),\ K_p = diag(k_{p_i}), \tag{3}$$

with $k_{v_i} > 0$, $k_{p_i} > 0$, $i \in \{1,2,\ldots,n\}$.

Proof. Substituting $\bar{\tau} = M(q)(\ddot{q}_d-\bar{u})+C(q,\dot{q})+G(q)+F(\dot{q})$ into (2) yields

$$M(q)\ddot{q} = M(q)(\ddot{q}_d - \bar{u}) \iff \ddot{e} = \bar{u},$$

which can be arranged into the following linear tracking error models for $i \in \{1,2,\ldots,n\}$:

$$\begin{bmatrix}\dot{e}\\ \ddot{e}\end{bmatrix} = \begin{bmatrix}0 & I\\ 0 & 0\end{bmatrix}\begin{bmatrix}e\\ \dot{e}\end{bmatrix} + \begin{bmatrix}0\\ I\end{bmatrix}\bar{u} \iff \begin{bmatrix}\dot{e}_i\\ \ddot{e}_i\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}e_i\\ \dot{e}_i\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix}\bar{u}_i.$$

Since $\bar{u}$ is chosen as in (3), it means that $\bar{u}_i = -k_{v_i}\dot{e} - k_{p_i}e$, $i \in \{1,2,\ldots,n\}$ is substituted in every linear system above, yielding

$$\begin{bmatrix}\dot{e}_i\\ \ddot{e}_i\end{bmatrix} = \begin{bmatrix}0 & 1\\ -k_{p_i} & -k_{v_i}\end{bmatrix}\begin{bmatrix}e_i\\ \dot{e}_i\end{bmatrix}.$$

Since $k_{v_i} > 0$, $k_{p_i} > 0$, $i \in \{1,2,\ldots,n\}$, the closed-loop matrices are Hurwitz, i.e., $e(t) \to 0$ as $t \to \infty$, thus concluding the proof. □

Remark 1. *Many other results concerning computed torque for Lagrange-Euler systems are available in the literature; the interested reader is referred, for instance, to [2, pg 230], where K_p and K_v above are chosen as positive-definite matrices.*

B. Linear Matrix Inequalities

Many analysis and design problems in control can be cast as LMIs [15]. Indeed, the very Lyapunov equation $PA + A^TP = -I$ can be equivalently formulated as an LMI $PA + A^TP < 0$, where A corresponds to the system matrix in $\dot{x} = Ax$ and P to some positive-definite matrix variable to be found (decision variable), which serves as a Lyapunov matrix in the quadratic function $V(x) = x^TPx$. LMIs belong to the convex optimization field [19], which means they can be solved in polynomial time by numerically effective interior-point algorithms, which are already implemented in commercially available software [17]. Nonlinear problems can also be tackled with these tools thanks to the convex modeling techniques independently developed for Takagi-Sugeno [20] and linear parameter-varying systems [21].

Expressing a control problem as an LMI is not a trivial task. While some results are straightforward, others might require handling matrix expressions via a variety of properties as to find equivalent or at least sufficient LMI formulations. In this work the following are employed:

Property 1. Substitution:*[19] If a matrix inequality includes the product of two decision variables which can be bijectively replaced by a single one, the expression is an LMI, e.g.:*

$$\begin{aligned} &AX + BFX + X^TA^T + X^TF^TB^T < 0\\ &\iff AX + BM + X^TA^T + M^TB^T < 0, \end{aligned}$$

for decision variables X and F on the left and X and M on the right, where $M = FX$.

Property 2. Congruence:*[19] The solution space of a matrix inequality does not change if it is pre- and post-multiplied by an invertible matrix P, e.g.:*

$$PA + A^T P < 0 \iff AX + XA^T < 0,$$

if $X = P^{-1}$ is pre- and post-multiplied as $X(PA+A^T P)X < 0$, provided $P = P^T$.

III. Main Result

Lagrange-Euler systems subject to restrictions have the following Lagrange's equation of motion [10]:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} + \frac{\partial D}{\partial \dot{q}} - \sum_{i=1}^{\kappa} \lambda_i \frac{\partial R_i}{\partial q} = \bar{\tau}, \qquad (4)$$

which, in addition to the terms defined in the previous section, we find Lagrange multipliers $\lambda_i \in \mathbb{R}$ and partial derivatives of the position restrictions $R_i(q) = 0$, $i \in \{1, 2, \ldots, \kappa\}$, where κ is the number of restrictions; the term added due to the presence of these terms corresponds to reaction forces. Obviously, the equation above has an equivalent restricted Lagrange-Euler form:

$$M(q)\ddot{q} + C(q, \dot{q}) + G(q) + F(\dot{q}) - \sum_{i=1}^{\kappa} \lambda_i \frac{\partial R_i}{\partial q} = \bar{\tau}, \qquad (5)$$

where all the terms preserve their original meanings.

Theorem 2. *Let $q_d(t)$ be a sufficiently smooth desired trajectory satisfying $R_i(q_d) = 0$, $i \in \{1, 2, \ldots, \kappa\}$. Let $e \equiv q_d - q$ be the corresponding tracking error. The restricted system with Lagrange-Euler model* (5) *(or, equivalently, Lagrange's equation of motion* (4)*), satisfies* $\lim_{t\to\infty} e(t) = 0$ *if there exist matrices $X \in \mathbb{R}^{2n\times 2n}$ and $N \in \mathbb{R}^{n\times 2n}$ such that*

$$X > 0, \quad \begin{bmatrix} 0_n & I_n \\ 0_n & 0_n \end{bmatrix} X + \begin{bmatrix} 0_n \\ I_n \end{bmatrix} N + (*) < 0. \qquad (6)$$

In such case, the control law is given by

$$\bar{\tau} = M(q)(\ddot{q}_d - \bar{u}) + C(q,\dot{q}) + G(q) + F(\dot{q}) - \sum_{i=1}^{\kappa} \lambda_i \frac{\partial R_i}{\partial q}, \qquad (7)$$

$$\bar{u} = \begin{bmatrix} K_p & K_d \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \end{bmatrix}, \ \begin{bmatrix} K_p & K_d \end{bmatrix} = NX^{-1}. \qquad (8)$$

Proof. Substituting (7) into (5) yields $\ddot{e} = \bar{u}$; substituting (8) in the latter yields the following state-space representation:

$$\begin{bmatrix} \dot{e} \\ \ddot{e} \end{bmatrix} = \left(\begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} \begin{bmatrix} K_p & K_d \end{bmatrix} \right) \begin{bmatrix} e \\ \dot{e} \end{bmatrix}. \qquad (9)$$

Since $X > 0$, let $V = [e^T \ \dot{e}^T] P [e^T \ \dot{e}^T]^T$, $P = X^{-1} > 0$ be a Lyapunov function candidate of (9); its time derivative is given by

$$\dot{V} = 2 \begin{bmatrix} e \\ \dot{e} \end{bmatrix}^T P \begin{bmatrix} \dot{e} \\ \ddot{e} \end{bmatrix} = \begin{bmatrix} e \\ \dot{e} \end{bmatrix}^T P \left(\begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} \begin{bmatrix} K_p & K_d \end{bmatrix} \right) \begin{bmatrix} e \\ \dot{e} \end{bmatrix} + (*).$$

This means that $\dot{V} < 0$ if

$$P \left(\begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} \begin{bmatrix} K_p & K_d \end{bmatrix} \right) + (*) < 0,$$

which is guaranteed by (6) because it is the same inequality: indeed, it suffices to apply the congruence property 2 with P and then use the substitution property 1 with the definitions in (8). Therefore, V is a Lyapunov function for the linear system establishing global asymptotic stability of $e = 0$, thus concluding the proof. □

Remark 2. *The proposal in Theorem 2 assumes perfect knowledge of the model and availability of all positions and velocities. However, robust techniques as described in [3, Section 7.3] can be straightforwardly applied. Moreover, in contrast with the standard computed-torque technique in Theorem 1, Theorem 2 uses full-block gain matrices K_p and K_d, not only diagonal forms. Moreover, the LMI formulation allows for gains to be better planned as many performance specifications can be added in the LMI context, e.g., decay rate, input and output constraint, pole placement within a region, among others [15].*

Remark 3. *Computed torque in open kinematic chains usually requires the plant to be fully actuated as $\bar{\tau}$ contains expressions based on the model that must be fed back into the plant; see Theorem 1. This is not necessarily the case in closed kinematic chains; see Theorem 2. Indeed, the number of degrees of freedom (DOF) in restricted systems is reduced when compared with their open counterparts; therefore, a less number of actuators is required to perform trajectory tracking. However, redundancy of actuators is not an issue as $\bar{\tau}$ in our proposal solves for them. It is also important to notice that the state of a closed kinematic chain, understood as the minimum information to predict the future behaviour of the system, has less variables than its open counterpart, as all the redundant "states" are solved from the real ones [22].*

IV. Implementation Issues

The developments in the previous section focused on the Lagrange-Euler form (4), which incorporated reaction forces through Lagrange multipliers. However, this model is essentially no different from that of an open kinematic chain in (2): how, therefore, can it hold the position restrictions? The answer is that this model is incomplete and requiring, at least, consistent initialization, i.e., it has to begin within the manifold induced by the algebraic restrictions. The missing part of the model can be better understood using an index-1 DAE model[1], i.e.:

$$\dot{x}_1 = f(x_1, x_2, u), \qquad (10)$$

$$0 = g(x_1, x_2, u), \qquad (11)$$

where $x_1 \in \mathbb{R}^{n_d}$ is known as the *differential state* (or simply state), $x_2 \in \mathbb{R}^{n_a}$ as the *algebraic state* (or false state), $u \in \mathbb{R}^m$ as the control input, and $\mathscr{C}^1$ mappings $f(\cdot,\cdot,\cdot) : \mathbb{R}^{n_d} \times \mathbb{R}^{n_a} \times \mathbb{R}^m \mapsto \mathbb{R}^{n_d}$ and $g(\cdot,\cdot,\cdot) : \mathbb{R}^{n_d} \times \mathbb{R}^{n_a} \times \mathbb{R}^m \mapsto \mathbb{R}^{n_a}$. Notice that, without considering the input signal u, the number of

[1] A DAE with high-order index and general forms such as $f(\dot{x}_1, x_1, x_2, u) = 0$ or $M(x_1, x_2)\dot{x}_1 = f(x_1, x_2, u)$ can be reduced to index 1 through the Pantelides algorithm [13]; index reduction is considered a pre-processing task, which is why it is left out of this work.

variables is the same as the number of scalar equations, i.e., $n_d + n_a$.

Assumption 1. *[23]* $\partial g/\partial[x_2^T\ u^T]^T$ *is full column rank.*

Assumption 2. *[24]* $f(x_1, x_2, u)$ *is locally Lipschitz in* x_1, x_2, *and* u *in a domain* $\mathcal{D} \subset \mathbb{R}^{n_d+n_a+m}$ *with solutions lying in a compact subset of* $\mathcal{D}$.

Assumption 1 implies that x_2 and u can be solved in terms of x_1 from (11), thus reducing (10) to an ODE depending on x_1 and u, subject to consistent initialization. Once this is reduced to an ODE, assumption 2 guarantees existence and uniqueness of solutions.

Now, consider the task of writing a restricted Lagrange-Euler system of the form (4) as an index-1 DAE model of the form (10)-(11). As previously discussed, the number of DOF of the restricted plant is lower than that of its open kinematic chain counterpart, which means that we have to select a minimum set of physical variables to act as differential states, whereas the rest will stand as algebraic variables that can be solved in terms of the former and, possibly, inputs. Clearly, this selection is not unique. A step-by-step algorithm to perform this task is the following:

1) Taking into account the number of DOF, select the physical variables that stand for differential states x_1: a trivial option is to choose the minimum number of positions in q (with their corresponding velocities $\dot{q}$) that allows describing the whole dynamics of the plant, i.e., $n - \kappa$ positions and $n - \kappa$ velocities, for a total of $n_d = 2(n - \kappa)$.
2) Define x_2 as the rest of variables: in the trivial option this means κ positions, κ velocities, and n accelerations in q, $\dot{q}$, and $\ddot{q}$, respectively, including κ Lagrange multipliers λ_i, for a total of $n_a = 3\kappa + n$.
3) Define u as a vector gathering all the inputs acting on the system. Note that $\bar{\tau}$ is a vector of generalized torques which might depend on a variety of forces, i.e., they might have different sizes. See the case study in Section V.
4) Define $f(x_1, x_2, u)$ using the differential relationship between $\dot{x}_1$ and the corresponding differential/algebraic variables x_1, x_2. In the trivial option, these dynamics are usually a set of 1st-order linear equations.
5) Define $g(x_1, x_2, u) = 0$ from the κ position, κ velocity, and κ acceleration restrictions, plus the n Lagrange-Euler 2nd-order nonlinear differential equations in (4). Notice that, due to the definition of x_2, no equation in $g(x_1, x_2, u) = 0$ includes time derivatives: they are purely algebraic.

Example 1. *Consider the fully actuated* Δ *parallel robot configuration in Fig. 1, which is a restricted Lagrange-Euler plant consisting of* 3 *2-link RR arms whose end effectors are at the same horizontal plane and at a fixed distance of each other, resulting in* $\kappa = 2$ *position restrictions for a joint vector* $q = [\theta_1\ \theta_2\ \theta_3\ \theta_4\ \theta_5\ \theta_6]^T$ *(*$n = 6$*); therefore, it is characterized by* 3 *pairs of* 2*nd-order scalar equations in the Lagrange form* (4) *or Lagrange-Euler form* (5).

Following the methodology just described to express the mathematical model as a DAE, it is clear that there are $n_d = 2(6-2) = 8$ *differential states in* x_1, *which must include only position and velocities of a minimum set of variables determining the rest of them (due to the algebraic restrictions); a possible choice is the following:*

$$x_1 = [\theta_1\ \dot{\theta}_1\ \theta_3\ \dot{\theta}_3\ \theta_5\ \dot{\theta}_5\ \theta_6\ \dot{\theta}_6]^T;$$

consequently, there are $n_a = 3(2) + 6 = 12$ *algebraic variables in* x_2, *i.e., the remaining positions and velocities, all the accelerations, and all the Lagrange multipliers:*

$$x_2 = [\theta_2\ \dot{\theta}_2\ \theta_4\ \dot{\theta}_4\ \ddot{\theta}_1\ \ddot{\theta}_2\ \ddot{\theta}_3\ \ddot{\theta}_4\ \ddot{\theta}_5\ \ddot{\theta}_6\ \lambda_1\ \lambda_2]^T;$$

and, finally, $m = 6$ *inputs in* u:

$$u = [\bar{\tau}_1\ \bar{\tau}_2\ \bar{\tau}_3\ \bar{\tau}_4\ \bar{\tau}_5\ \bar{\tau}_6]^T,$$

corresponding to the generalized forces, i.e., $u = \bar{\tau}$.

Following the notation $x_{1,i}$, $i \in \{1, 2, \ldots, 8\}$ *for the* 8 *differential states and* $x_{2,i}$, $i \in \{1, 2, \ldots, 12\}$ *for the* 12 *algebraic variables, we have that the dynamical part is given by*

$$\begin{bmatrix} \dot{x}_{1,1} \\ \dot{x}_{1,2} \\ \dot{x}_{1,3} \\ \dot{x}_{1,4} \\ \dot{x}_{1,5} \\ \dot{x}_{1,6} \\ \dot{x}_{1,7} \\ \dot{x}_{1,8} \end{bmatrix} = \begin{bmatrix} x_{1,2} \\ x_{2,5} \\ x_{1,4} \\ x_{2,7} \\ x_{1,6} \\ x_{2,9} \\ x_{1,8} \\ x_{2,10} \end{bmatrix} \equiv f(x_1, x_2, u).$$

We do not detail the 12 *algebraic equations, but recall that they gather* 2 *position restrictions, its* 2 1*st-order time derivatives (velocity restrictions), its* 2 2*nd-order time derivatives (acceleration restrictions), and* 3 *pairs of* 2*nd-order Lagrange-Euler equations, for a total of* 12 *algebraic equations. Therefore, the total number of equations (*$8 + 12 = 20$*) coincides with the number of variables in* x_1 *and* x_2 *(*$n_d + n_a = 20$*). None of the equations in* $g(x_1, x_2, u) = 0$ *has a time derivative: even the Lagrange-Euler equations have their accelerations replaced by their state equivalents, i.e.,* $x_{2,i+4} = \ddot{\theta}_i$, $i \in \{1, 2, \ldots, 6\}$.

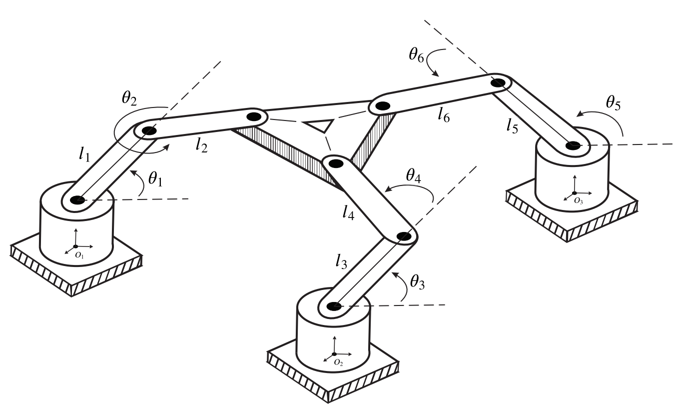


Fig. 1. Fully actuated Δ parallel robot with horizontal platform sustained by 3 2-link RR arms in Example 1.

V. Case Study

Consider the scheme in Fig. 2, which corresponds to a 2-link planar RR arm subject to a position constraint, i.e., the bar on the right slides along the inclined track above. This plant has 3 possible actuators: 2 of them induce torques τ_1 at the leftmost side of the link labelled l_1 and τ_2 at the joint between the two bars; a third one induces a linear force f on the rightmost side of link l_2. In this case study we examine the effectiveness of our proposal and implementation issues for trajectory tracking under 3 different situations: (a) all the actuators acting at the same time; (b) only τ_1 and f acting on the system; (c) only τ_1 serving as actuator.

Considering $q = [\theta_1\ \theta_2]^T$ as the joint vector and assuming link masses are concentrated at the ends of the links, the left-hand side of its Lagrange's equation of motion (4) has the following part associated to the Lagrangian:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} = \begin{bmatrix} \left((m_1+m_2)l_1^2+m_2l_2^2+2m_2l_1l_2\cos\theta_2\right)\ddot{\theta}_1 \\ +\left(m_2l_2^2+m_2l_1l_2\cos\theta_2\right)\ddot{\theta}_2-m_2l_1l_2 \\ \times\left(2\dot{\theta}_1\dot{\theta}_2+\dot{\theta}_2^2\right)\sin\theta_2+(m_1+m_2)gl_1\cos\theta_1 \\ +m_2gl_2\cos(\theta_1+\theta_2) \\ \hline \left(m_2l_2^2+m_2l_1l_2\cos\theta_2\right)\ddot{\theta}_1+m_2l_2^2\ddot{\theta}_2 \\ +m_2l_1l_2\dot{\theta}_1^2\sin\theta_2+m_2gl_2\cos\left(\theta_1+\theta_2\right) \end{bmatrix};$$

the following friction terms

$$\frac{\partial D}{\partial \dot{q}} = \frac{\partial}{\partial \dot{q}}\left(\frac{1}{2}\begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\end{bmatrix}^T \begin{bmatrix}k_1 & 0\\ 0 & k_2\end{bmatrix}\begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\end{bmatrix}\right) = \begin{bmatrix}k_1\dot{\theta}_1\\ k_2\dot{\theta}_2\end{bmatrix};$$

and reaction forces arising from the single position restriction

$$\begin{aligned} R_1(q) =& l_1\sin\theta_1 + l_2\sin(\theta_1+\theta_2) - y_r \\ &- (l_1\cos\theta_1 + l_2\cos(\theta_1+\theta_2))\tan\theta_r = 0, \end{aligned}$$

given by

$$\sum_{i=1}^{\kappa}\lambda_i\frac{\partial R_i}{\partial q} = \lambda_1\begin{bmatrix} l_1\cos\theta_1+l_2\cos(\theta_1+\theta_2)+l_1\sin\theta_1\tan\theta_r \\ +l_2\sin(\theta_1+\theta_2)\tan\theta_r \\ \hline l_2\cos(\theta_1+\theta_2)+l_2\sin(\theta_1+\theta_2)\tan\theta_r \end{bmatrix},$$

with $\kappa = 1$;

The generalized torque $\bar{\tau}$ on the right-hand side of (4) should include τ_1, which acts at the origin of link l_1, τ_2, which acts at the joint of the links, and torque $\bar{\tau}_f$ induced by f at the endpoint of link l_2; such endpoint has the following coordinates:

$$\vec{x} = \begin{bmatrix} l_1\cos\theta_1 + l_2\cos(\theta_1+\theta_2) \\ l_1\sin\theta_1 + l_2\sin(\theta_1+\theta_2) \end{bmatrix},$$

which are used to obtain $\bar{\tau}_f$ as

$$\bar{\tau}_f = \left(\frac{\partial \vec{x}}{\partial q}\right)^T \vec{f}, \text{ with } \vec{f} = \begin{bmatrix} f\cos\theta_r \\ f\sin\theta_r \end{bmatrix},$$

which yields the generalized torque

$$\bar{\tau} = \begin{bmatrix} \tau_1-(l_1\sin\theta_1+l_2\sin(\theta_1+\theta_2))f\cos\theta_r \\ +(l_1\cos\theta_1+l_2\cos(\theta_1+\theta_2))f\sin\theta_r \\ \hline \tau_2-l_2\sin(\theta_1+\theta_2)f\cos\theta_r+l_2\cos(\theta_1+\theta_2)f\sin\theta_r \end{bmatrix}. \quad (12)$$

A DAE model of the above system can be obtained by recalling that it has only 1 degree of freedom; therefore, it suffices to describe position θ_1 and velocity $\dot{\theta}_1$ to fully describe the future behaviour of the system. This means that the true states (i.e., dynamical variables) are $x_{1,1} = \theta_1$ and $x_{1,2} = \dot{\theta}_1$, with $x_1 = [x_{1,1}\ \ x_{1,2}]^T$; additional variables are defined as $x_{2,1} = \theta_2$, $x_{2,2} = \dot{\theta}_2$, $x_{2,3} = \ddot{\theta}_1$, $x_{2,4} = \ddot{\theta}_2$ and $x_{2,5} = \lambda$, with $x_2 = [x_{2,1}\ \ x_{2,2}\ \ x_{2,3}\ \ x_{2,4}\ \ x_{2,5}]^T$ (i.e., algebraic variables). Inputs τ_1, τ_2, and f can be renamed as u_1, u_2, and u_3, respectively, gathered as $u = [u_1\ u_2\ u_3]^T$.

Therefore, the dynamic equations can be written as:

$$\dot{x}_1 = \begin{bmatrix}\dot{x}_{1,1}\\ \dot{x}_{1,2}\end{bmatrix} = \begin{bmatrix}x_{1,2}\\ x_{2,3}\end{bmatrix}, \quad (13)$$

subject to the following algebraic constraints:

1) Position constraint:

$$\begin{aligned} &l_1\sin x_{1,1}+l_2\sin(x_{1,1}+x_{2,1})-y_r \\ &-(l_1\cos x_{1,1}+l_2\cos(x_{1,1}+x_{2,1}))\tan\theta_r = 0. \end{aligned}$$

2) Velocity constraint:

$$\begin{aligned} &\tan\theta_r(l_1x_{1,2}\sin x_{1,1}+l_2\ \sin(x_{1,1}+x_{2,1})(x_{1,2}+x_{2,2})) \\ &+l_1x_{1,2}\cos x_{1,1}+l_2\cos(x_{1,1}+x_{2,1})(x_{1,2}+x_{2,2}) = 0. \end{aligned}$$

3) Acceleration constraint:

$$\begin{aligned} &\tan\theta_r(l_1x_{2,3}\sin x_{1,1}+l_1x_{1,2}^2\cos x_{1,1}+l_2\sin(x_{1,1}+x_{2,1}) \\ &\times(x_{2,3}+x_{2,4})+l_2\cos(x_{1,1}+x_{2,1})(x_{1,2}+x_{2,2})^2) \\ &+l_1x_{2,3}\cos x_{1,1}-l_2\sin(x_{1,1}+x_{2,1})(x_{1,2}+x_{2,2})^2 \\ &-l_1x_{1,2}^2\sin x_{1,1}+l_2\cos(x_{1,1}+x_{2,1})(x_{2,3}+x_{2,4}) = 0. \end{aligned}$$

4) First Lagrange-Euler equation:

$$\begin{aligned} &((m_1+m_2)l_1^2+m_2l_2^2+2m_2l_1l_2\cos x_{2,1})x_{2,3} \\ &+(m_2l_2^2+m_2l_1l_2\cos x_{2,1})x_{2,4} \\ &-m_2l_1l_2(2x_{1,2}x_{2,2}+x_{2,2}^2)\sin x_{2,1} \\ &+(m_1+m_2)gl_1\cos x_{1,1}+m_2gl_2\cos(x_{1,1}+x_{2,1}) \\ &+k_1x_{1,2}-x_{2,5}(l_1\cos x_{1,1}+l_2\cos(x_{1,1}+x_{2,1}) \\ &+l_1\sin x_{1,1}\tan\theta_r+l_2\sin(x_{1,1}+x_{2,1})\tan\theta_r) \\ &-(u_1-(l_1\sin x_{1,1}+l_2\sin(x_{1,1}+x_{2,1}))u_3\cos\theta_r \\ &+(l_1\cos x_{1,1}+l_2\cos(x_{1,1}+x_{2,1}))u_3\sin\theta_r) = 0. \end{aligned}$$

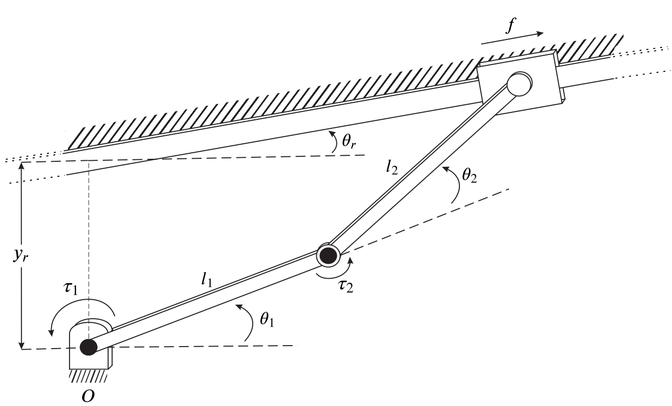


Fig. 2. Constrained two-bar system under 1 linear inclined and 2 rotational forces in Section V.

5) Second Lagrange-Euler equation:

$$(m_2l_2^2+m_2l_1l_2\cos x_{2,1})x_{2,3}+m_2l_2^2x_{2,4}$$
$$+m_2l_1l_2x_{1,2}^2\sin x_{2,1}+m_2gl_2\cos(x_{1,1}+x_{2,1})+k_2x_{2,2}$$
$$-x_{2,5}(l_2\cos(x_{1,1}+x_{2,1})+l_2\sin(x_{1,1}+x_{2,1})\tan\theta_r)$$
$$-(u_2-l_2\sin(x_{1,1}+x_{2,1})u_3\cos\theta_r$$
$$+l_2\cos(x_{1,1}+x_{2,1})u_3\sin\theta_r)=0.$$

Note that the total number of variables (7 in total: 2 in x_1 and 5 in x_2) match the total number of equations (7 in total: 2 differential and 5 algebraic). It should also be noted that, in this case, a state-space ODE representation does not naturally precede the DAE representation; rather, one would need to solve for the variables in x_2 from the DAE in order to describe the dynamics of the 2 equations in x_1, which govern the entire system, with τ_1 and f given.

Consider the task of joint vector $q(t) = [\theta_1(t)\ \theta_2(t)]^T$ tracking a desired trajectory $q_d(t) = [\theta_{d1}(t)\ \theta_{d2}(t)]^T$, where $\theta_{d1}(t) = -1.6+0.1\sin(0.1t)+0.07\cos(0.5t)+0.05\sin t$ and $\theta_{d2}(t)$ is induced by the same position restriction as the plant, i.e., $\theta_{d2}(t)$ is implicitly defined by

$$l_1\sin\theta_{d1}(t)+l_2\sin(\theta_{d1}(t)+\theta_{d2}(t))-y_r$$
$$-(l_1\cos\theta_{d1}(t)+l_2\cos(\theta_{d1}(t)+\theta_{d2}(t)))\tan\theta_r = 0. \quad (14)$$

The control law in (7) (also known as inner loop) requires the 2nd-order time derivative of $q_d(t)$, which can be explicitly known for $\ddot{\theta}_{d1}(t)$ and implicitly known for $\ddot{\theta}_{d2}(t)$ if the position restriction (14) is differentiated two times. Moreover, since the outer loop (8) requires $e(t) = q_d(t) - q(t)$ and $\dot{e}(t) = \dot{q}_d(t) - \dot{q}(t)$, where $\theta_{d1}(t)$ and $\dot{\theta}_{d1}(t)$ are calculated explicitly whereas $\theta_{d2}(t)$ and $\dot{\theta}_{d2}(t)$ are obtained implicitly from (14) and its first time derivative.

For simulation purposes, the MATLAB Symbolic Math Toolbox 25.2 is employed, specifically the equation solving tools for DAEs. These tools require specifying equations (both differential and algebraic) and variables (which must match the number of equations). Besides variables in x_1 and x_2 and equations in $f(x_1,x_2,u)$ and $g(x_1,x_2,u)$ we have to add variables θ_{d2}, $\dot{\theta}_{d2}$, and $\ddot{\theta}_{d2}$, along with equations (14) and its 1st and 2nd time derivatives (obviously, θ_{d1} and its time derivatives are explicitly given, which means they do not require to be declared as variables). Once equations are provided, the MATLAB simulation routine implies checking that the differentiability index is reduced (which is the case in our proposal), finding consistent initial conditions, fixing those which are critical for the application, and calling a robust solver which keeps the algebraic restrictions characterizing the DAE[2].

Case with 3 *actuators:* If all inputs are actuated, they must be solved from (12), where $\bar{\tau}_1$ and $\bar{\tau}_2$ are already known according to (7). Clearly, there are infinite solutions to this system (3 variables, 2 equations); a possible choice is considering $\tau_2 = 1/2\bar{\tau}_2$ along with (12). Theorem 2 is invoked to obtain $\bar{\tau}$; LMIs (6) yield the following gains for the outer control loop $\bar{u}$:

$$K_p = \begin{bmatrix} -0.7972 & -0.2207 \\ -0.1625 & -0.9656 \end{bmatrix}, K_d = \begin{bmatrix} -1.0045 & -0.8645 \\ -0.8894 & -1.5385 \end{bmatrix}.$$

In Fig. 3 system positions $\theta_1(t) = x_{1,1}(t)$ and $\theta_2(t) = x_{2,1}(t)$ are displayed along with the desired trajectories $\theta_{d1}(t)$ and $\theta_{d2}(t)$, respectively. Over the initial absolute error, the reference is reached within 1% of the true value in 3.6s. For completeness, control signals are shown in Fig. 4. The energy (integral of the squared signal) spent over 80s by τ_1, τ_2, and f is given by 10.2108, 0.008326, and 0.0168, respectively.

Case with 2 *actuators:* If τ_1 and f are available with $\tau_2 = 0$, they can be uniquely solved from (7). Obviously, the outer loop remains the same. As before, Fig. 5 and Fig. 6 show trajectory tracking and control inputs along the time, confirming the fact that 2 actuators are still enough to perform the task. Over the initial absolute error, the reference is reached within 1% of the true value in 3.6s, while the control effort over 80s by τ_1, τ_2, and f is given by 10.0496, 0, and 0.0102, respectively.

Case with 1 *actuator:* If only τ_1 is available for control, i.e., $\tau_2 = f = 0$, system (7) has no solution. However, it

[2]The interested reader is referred to instructions such as `isLowIndexDAE`, `reduceDAEIndex`, `daeFunction`, `decic`, and `ode15i` [25].

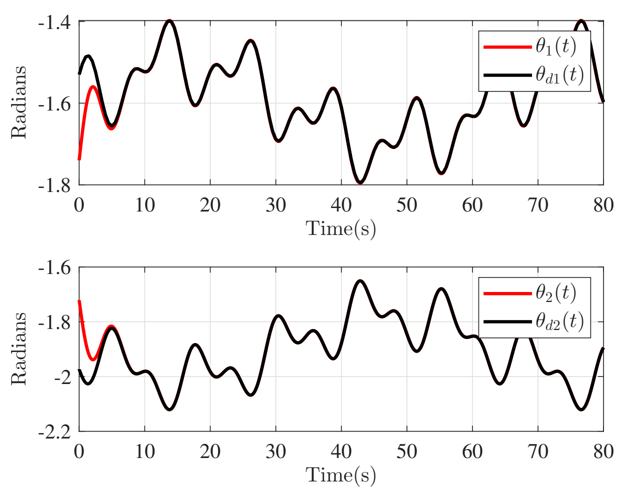


Fig. 3. Trajectory tracking for the case with 3 actuators: base angle $\theta_1(t)$ (top); upper angle $\theta_2(t)$ (bottom).

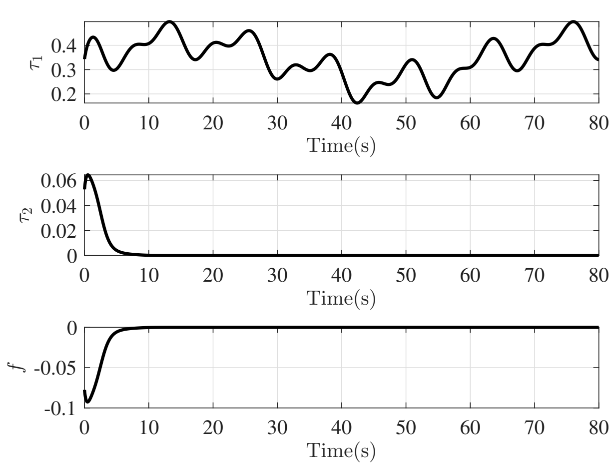


Fig. 4. Control inputs for the case with 3 actuators: base torque $\tau_1(t)$ (top); upper torque $\tau_2(t)$ (middle); linear force (bottom).

is obvious from the physical configuration in Fig 2 that one actuator must be enough to perform trajectory tracking since the whole system has only 1 DOF. Indeed, $\lim_{t\to\infty}\tau_2(t) = \lim_{t\to\infty} f(t) = 0$ in both cases 1 and 2 (Figs. 4 and 6).

In order to apply Theorem 2 to the under-actuated case, some pre-processing is required to reduce the problem to a *single* actuator with a *single* equation, namely, considering only the first equation in (7) to solve τ_1 from $\bar{\tau}_1$. Trajectory tracking follows from τ_1 enforcing the first bar to track $\theta_{d1}(t)$ while the rest is subject to the algebraic restrictions, thus guarantee tracking too.

It is apparent that $\bar{\tau}$ is the same *in all cases* while the difference lies on the way it is used to assign values to f, τ_1 and τ_2, depending on whether they are available (actuated) or not.

Remark 4. *Standard PD computed torque relies on tuning for stabilization; if used in this case, it would not be able to guarantee specific decay rate, mean squared error, or input energy, as all these specifications are not design parameters, but trial-and-error goals. On the other hand, LMI-based methodologies, as the one proposed, allow the designer to impose the referred specifications in an LMI form.*

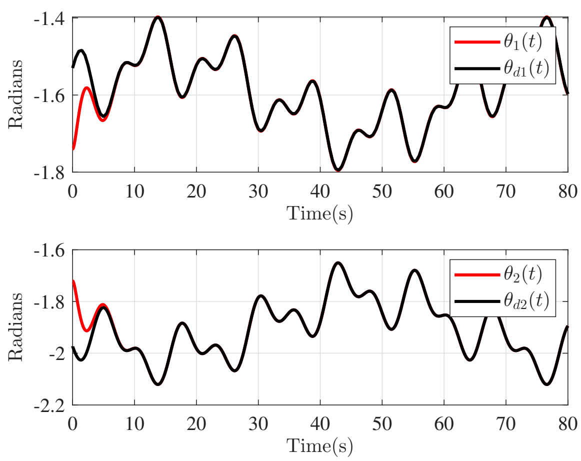


Fig. 5. Trajectory tracking for the case with 2 actuators: base angle $\theta_1(t)$ (top); upper angle $\theta_2(t)$ (bottom).

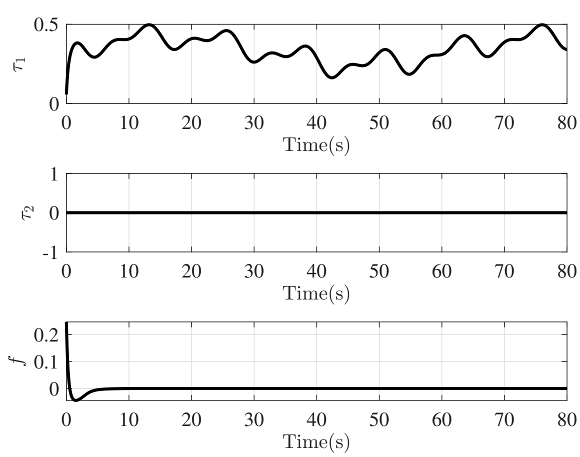


Fig. 6. Control inputs for the case with 2 actuators: base torque $\tau_1(t)$ (top); upper torque $\tau_2(t)$ (middle); linear force (bottom).

VI. Implementation on a Rotary Arm System

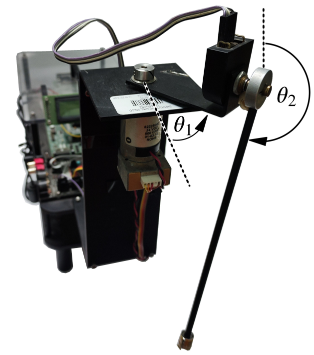


Fig. 7. Rotary arm system characterized as an underactuated setup with only 1 degree of freedom and a second link as a mass disturbance.

Consider the experimental setup in Fig. 7, which does not incorporate explicit mechanical constraints. Rather, the platform can be characterized as an underactuated system in which only the first degree of freedom is actively actuated and controlled, while the second coordinate evolves according to the system's natural dynamics and is effectively treated as a disturbance. As the proposed method is not intended to address general underactuated control problems, the experimental results should be interpreted as a proof-of-concept implementation for unrestricted Lagrange-Euler plants whose model is put as DAEs, rather than as a validation of the method for constrained or underactuated systems.

The mathematical model is the same as the well-known Furuta pendulum, i.e.,

$$\tau = (\beta\sin^2\theta_2 + \alpha)\ddot{\theta}_1 + \delta\cos\theta_2\ddot{\theta}_2 + 2\beta\dot{\theta}_1\dot{\theta}_2\cos\theta_2\sin\theta_2 - \delta\dot{\theta}_2^2\sin\theta_2, \quad (15)$$

$$0 = \delta\cos\theta_2\ddot{\theta}_1 + (\beta+\gamma)\ddot{\theta}_2 - \sigma g\sin\theta_2 - \beta\dot{\theta}_1^2\cos\theta_2\sin\theta_2, \quad (16)$$

where θ_1 is the angular position of the horizontal arm with respect to a given reference, θ_2 is the angle of the free arm with respect to the vertical position, and the system parameters are given by $\alpha = 0.0363$, $\beta = 0.0306$, $\gamma = 0.0356$, $\delta = 0.0260$, $\sigma = 0.3829$, and $g = 9.804$.

According to our methodology, the first equation (15) corresponds to the dynamical part of a DAE while the second one is the algebraic restriction. Therefore, our computed-torque proposal is applied considering $q = \theta_1$ in the form (5), where $M(q) = \beta\sin^2\theta_2 + \alpha$ and $H(q,\dot{q}) = 2\beta\dot{\theta}_1\dot{\theta}_2\cos\theta_2\sin\theta_2$, plus an extra term $O(\cdot) = (\delta\cos\theta_2)\ddot{\theta}_2 - \delta\dot{\theta}_2^2\sin\theta_2$ which does not depend on q but whose highest-order derivative $\ddot{\theta}_2$ can be explicitly computed from the algebraic part (16).

Consider a desired trajectory $q_d(t) = -0.5\sin(4t)$; gains $K_p = -238.1234$ and $K_d = -22.2786$ are obtained via LMIs (8), leading to an inner control loop $\tau = H(q,\dot{q}) + O(\cdot) + M(q)(\ddot{q}_d - u)$, with outer control loop $u = -K_p e - K_d\dot{e}$ where $e = q_d - q$. The result of applying this control law in a real-time setup is shown in Fig. 8: on the top it is shown how $\theta_1(t)$ tracks $\theta_{1d}(t)$; on the bottom, the control signal $\tau(t)$ is displayed.

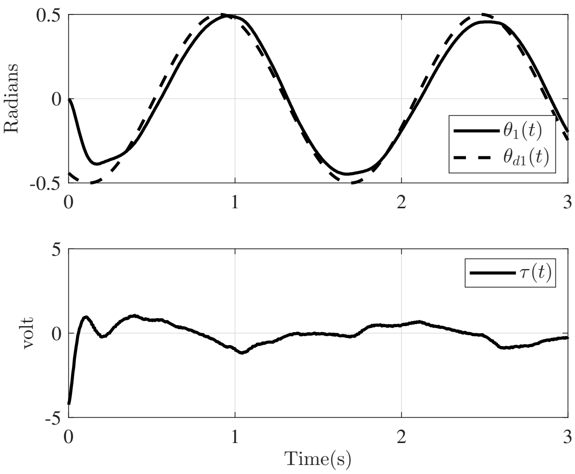


Fig. 8. Real-time trajectory tracking for angle $\theta_1(t)$ (top) and torque control signal $\tau(t)$ (bottom).

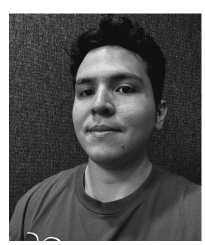
VII. Conclusions

This work has presented an extension of standard computed torque for restricted Lagrange-Euler systems. The inner control loop has been augmented as to include reaction forces while the outer control loop has been solved via linear matrix inequalities allowing full exploitation of the gain entries. A methodology to implement the proposed technique using index-1 differential algebraic equations has been provided. A case study as well as a real-time implementation have been developed to discuss a variety of configurations depending on the number of actuators.

Future work is undergoing to address the shortcomings of the proposal, namely:
(a) handling of underactuated restricted systems;
(b) handling of higher-index DAE models without performing index reduction;
(c) handling the lack of information of certain states via observer-based and output feedback.

References

[1] J. L. Lagrange, *Mécanique analytique*. Mallet-Bachelier, 1853, vol. 1.
[2] R. Kelly, V. S. Davila, and A. Loría, *Control of robot manipulators in joint space*. Springer, 2005, DOI: 10.1007/1-85233-999-3_3.
[3] F. L. Lewis, D. M. Dawson, and C. T. Abdallah, *Robot manipulator control: theory and practice*. CRC Press, 2003, DOI: 10.1201/9780203026953.
[4] A. Isidori, *Nonlinear Control Systems*, 3rd ed. London, England: Springer-Verlag, 1995, DOI: 10.1007/978-1-84628-615-5.
[5] G. Franklin, J. Powell, and A. Emami-Naeini, *Feedback control of dynamic systems*. Englewood Cliffs, USA: Prentice-Hall, 2002.
[6] J.-P. Merlet, *Parallel robots*. Springer, 2006, DOI: 10.1007/1-4020-4133-0.
[7] Jia, Xiaoxi and Kanzow, Christian and Mehlitz, Patrick and Wachsmuth, Gerd, "An augmented Lagrangian method for optimization problems with structured geometric constraints," *Mathematical Programming*, vol. 199, no. 1, pp. 1365–1415, 2023, DOI: 10.1007/s10107-022-01870-z.
[8] Bayo, E and Ledesma, Ragnar, "Augmented Lagrangian and mass-orthogonal projection methods for constrained multibody dynamics," *Nonlinear dynamics*, vol. 9, no. 1, pp. 113–130, 1996, DOI: 10.1007/BF01833296.
[9] Zhao, Xiao-Min and Chen, Ye-Hwa and Zhao, Han and Dong, Fang-Fang, "Udwadia–Kalaba equation for constrained mechanical systems: formulation and applications," *Chinese journal of mechanical engineering*, vol. 31, no. 1, p. 106, 2018, DOI: 10.1186/s10033-018-0310-x.
[10] Spong, Mark W, *Robot Modeling and Control.* John Wiley & Sons, 2012, ISBN:0471649902.
[11] P. J. Rabier and W. C. Rheinboldt, *Theoretical and numerical analysis of differential-algebraic equations*. Elsevier, 2002, DOI: 10.1016/S1570-8659(02)08004-3.
[12] A. Portela, I. Bessa, R. Márquez, and M. Bernal, "A novel LMI-based design of rational control laws for a class of nonlinear systems modeled as differential algebraic equations," *IEEE Control Systems Letters*, 2025, DOI: 10.1109/LCSYS.2025.3605600.
[13] C. C. Pantelides, "The consistent initialization of differential-algebraic systems," *SIAM Journal on scientific and statistical computing*, vol. 9, no. 2, pp. 213–231, 1988, DOI: 10.1137/0909014.
[14] N. S. Nedialkov, J. D. Pryce, and G. Tan, "Algorithm 948: Daesa—a matlab tool for structural analysis of differential-algebraic equations: Software," *ACM Transactions on Mathematical Software (TOMS)*, vol. 41, no. 2, pp. 1–14, 2015, DOI: 10.1145/2700586.
[15] S. Boyd, L. El Ghaoui, E. Féron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. Philadelphia, USA: Studies in Applied Mathematics, 1994, vol. 15, DOI: 10.1137/1.9781611970777.
[16] M. Bernal, A. Sala, Z. Lendek, and T. Guerra, *Analysis and Synthesis of Nonlinear Control Systems: A convex optimsation approach*. Springer, Cham, 2022, DOI: 10.1007/978-3-030-90773-0.
[17] P. Gahinet, A. Nemirovsky, A. Laub, and M. Chilali, *LMI control toolbox*. Natick, USA: Math Works, 1995, URL: 3611303_LMI_control_toolbox_user's_guide.
[18] J. L. Álvarez Urias and B. Castillo-Toledo, "LMI-based impulsive observers for sampled-output linear descriptor systems," in *Memorias del Congreso Nacional de Control Automático (CNCA 2025)*. Cancún, Quintana Roo, México: Asociación de México de Control Automático (AMCA), Oct. 2025.
[19] B. S. and L. Vandenberghe, *Convex Optimization*. New York, USA: Cambridge University Press, 2004, DOI: 10.1017/CBO9780511804441.
[20] K. Tanaka and H. Wang, *Fuzzy control systems design and analysis: A linear matrix inequality approach*. New York, USA: John Wiley and Sons, 2001, DOI: 10.1002/0471224596.
[21] C. Scherer and S. Weiland, *Linear Matrix Inequalities in Control*. Delft, The Netherlands: Dutch Institute for Systems and Control, 2015, URL: LectureNotes.
[22] M. Bernal, A. Sala, and A. González, "LMI-based nonlinear observer design for a class of nonlinear systems modeled with differential algebraic equations," *Kybernetika*, vol. 61, no. 3, pp. 429–446, 2025, DOI: 10.14736/kyb-2025-3-0429.
[23] Sala, Antonio and Bernal, Miguel and González, Antonio, "A Unified LMI-Based Observer Design Framework for Nonlinear Differential Algebraic Equations," *International Journal of Robust and Nonlinear Control*, 2026, DOI: 10.1002/rnc.70437.
[24] H. Khalil, *Nonlinear Control*. New Jersey, USA: Prentice Hall, 2014.
[25] The MathWorks Inc., "Symbolic Math Toolbox," Natick, Massachusetts, United States, 2024. [Online]. Available: https://www.mathworks.com/help/symbolic/index.html



Julián León is a Mechatronics Student with the Dept. of Electrical Engineering at the Sonora Institute of Technology. He is currently working on his final project to receive the degree on Mechatronics Engineering, working on computed torque for restricted Lagrange-Euler systems modelled as differential algebraic equations.

David Vázquez is Posdoc at the Dept. of Electrical Engineering of the Sonora Institute of Technology. He received his Ph.D. degree in Engineering Sciences from the Sonora Institute of Technology, Mexico, in 2023. Since then, he has been a postdoctoral researcher at the Sonora Institute of Technology. He has supervised several Master and Bachelor students. His research on nonlinear control via convex structures, linear matrix inequalities and sliding mode control has resulted in three journal papers.

Miguel Bernal is Full Professor at the Dept. of Electrical Engineering of the Sonora Institute of Technology. He received his Ph.D. degree in automatic control from the Czech Technical University at Prague in 2005. He was a postdoctoral researcher at the University of Valenciennes and Hainaut-Cambresis, France, from 2006 to 2009. He is a member of the National Research System of Mexico since 2007. In 2011 he was appointed a Full Professor at the Sonora Institute of Technology, Mexico. He has led several research projects and Ph.D. theses in Mexico and abroad on nonlinear control via convex structures and linear matrix inequalities, an area where he counts with more than 70 journal papers and over 2500 citations.