



RESEARCH ARTICLE

Newton–Euler modeling and Hamiltonians for robot control in the geometric algebra

Eduardo Bayro-Corrochano* , Jesus Medrano-Hermosillo , Guillermo Osuna-González and Ulises Uriostegui-Legorreta

Centro de Investigaciones y Estudios Avanzados, CINVESTAV, Department of Electrical Engineering and Computer Science, Campus Guadalajara, 1145 Del Bosque Ave., El Bajío, 45019, Zapopan, México

*Corresponding author. E-mail: eduardo.bayro@cinvestav.mx

Received: 19 July 2019; **Revised:** 4 May 2022; **Accepted:** 9 May 2022; **First published online:** 22 July 2022

Keywords: Newton–Euler modeling, motor algebra $G^{+}_{3,0,1}$, dynamic model, Hamiltonians, nonlinear control, SE(3) PD control, sliding mode control, localized control, robot arms, tracking

Abstract

The principal objective of the paper is to show the importance of the Hamiltonian in control theory. Instead of using the Lagrangian formulation of electromechanical or robotic systems, our work is focused on robot dynamics by its Hamiltonian. Using the iterative Newton–Euler, we generate the local Hamiltonians and the derivative of the moments at each joint of the robot manipulator. Thus, we can apply decentralized controllers at each joint. We compare and discuss the efficiency of the controllers. We show that the performance of the sliding modes controller is more robust than that of the PD or Bang–Bang controllers.

1. Introduction

Electromechanical and robotic systems are most currently used in Euler-Lagrange equations. In this formalism, canonical coordinates such as position and velocity are used to develop control. Historically, the Hamiltonian approach has its roots in analytical mechanics and starts from the principle of least action, and proceeds, via the Euler-Lagrange equations and the Legendre transform, towards the Hamiltonian equations of motion. On the other hand, the network approach stems from electrical engineering and constitutes a cornerstone of mathematical systems theory. While most of the analysis of physical systems has been performed within the Lagrangian and Hamiltonian framework, the network point of view is prevailing in the modeling and simulation of (complex) physical engineering systems. Several authors approached the formulation of Hamiltonians for the control of mechanical systems in the phase space using matrix algebra and tensor calculus [1–5].

A first attempt to model electromechanical and robot systems using Hamiltonian equations was just for 2 degree of freedom (DOF) [3, 6]. For a geometric approach, Hestenes [7] and Abou El Dahab [8] and Pappas [9] for the formulation of Hamiltonian mechanics used the geometric algebra framework and show the extension of the invariant formulation for rigid-body mechanics and in Hestenes [10] to a phase space formulation for systems of linked rigid bodies. In the article [11], we present in detail the screw theory, moments and wrenches for Newton–Euler dynamic modeling, and control and quadratic programming-based tracking of robots. The geometric constraints for the quadratic programming are designed using the conformal geometric algebra framework. This paper has a tutorial value for self-study on the Newton–Euler dynamic modeling.

The interest in design stabilization and follow-up controllers for Hamiltonian systems has grown in recent years. Many researchers use passivity as a tool, see Romero et al. [12], Yaghmeaei and Yazdanpanah [13], and Reyes-Baez et al. [14]. The idea to use a structure is to take advantage of its

properties and be able to guarantee better results, as in the case of robotics, see, for example, Kelly et al. [15] or in general Euler–Lagrange systems in Ortega et al. [16].

In the last decades, researchers presented different approaches to solve the problem for systems with port-Hamiltonian structures. Fujimoto et al. [17] proposed to design an error system employing canonical transformations generalized to convert the problem to one on stabilization. However, same as the interconnection and injection of damping (IDA-PBC) (Ortega et al. [18]), the method requires the solution of a partial differential equation (PDE) to perform these transformations. Moreover, it is not an easy task to design a follow-up controller that preserves the port-controlled Hamiltonian structure (PCH) in the closed loop. This is because for the energy function to transform, it must be variant in time, which implies that the passivity property is not fulfilled in general and requires a stabilization technique for Hamiltonian systems through time-variants controllers (Fujimoto and Sugie [19]).

Following the results of Fujimoto et al. [17], Mulero-Martinez [20] presented a strategy of control for robot manipulators represented by the port-Hamiltonian structure utilizing canonical transformations and control based on passivity, proposing a new modification methodology of energy through the introduction of nonhomogeneous virtual fields which allow obtaining standard laws of control for robot manipulators such as the PD controller with gravity compensation (PD+) or the PD controller with pre-compensation (PD with feed-forward).

In Dirksz and Scherpen [21], an adaptive control is combined with canonical transformations to build control, which is then applied to port systems Hamiltonians. A follow-up control that preserves closed-loop PCH structure is shown by Donaire et al. [22], where global asymptotic stability is verified in the error system whose equilibrium point is the origin. To improve robustness in control, the law of control is extended with the addition of the integral action to cancel constants disturbances exploiting the procedure proposed by Donaire and Junco [23]. In a different approach, Romero et al. [12] combined a passivity-based controller with a new observer of immersion and invariance.

Our paper deals with the modeling and control of robotic manipulators, where the robot dynamics are described by employing Hamiltonian mechanics. This way leads to different physical descriptive quantities used in control design. In this paper, the model-oriented Lyapunov-based control is considered. It is introduced in the novel formulation using Hamiltonian mechanics and compared with the use of a conventional controller. The theoretical results, generally applicable to usual articulated industrial robotic manipulators, are demonstrated on a planar 2 DOF robot arm and a spherical pendulum of 4 DOF.

The contribution of this work consists of the following: we compute the Hamiltonians, dynamics, and applied controllers using screw theory in motor algebra. By using the recursive Newton–Euler dynamics, we can compute the Hamiltonians for each joint. There is no paper in the literature where the authors show that they can compute Hamiltonians at each joint for more than 2 DOF. Thanks to the screw theory, it is possible to compute recursively the Hamiltonian and the dynamics for each joint for robot manipulators of n DOF. As a result, we can apply local controllers to each joint. In the experimental analysis, using simulations, we compute via the Newton–Euler dynamics and the Hamiltonians for a planar 2 DOF robot arm and for a spherical pendulum of 4 DOF. We apply and compare four controllers: Bang–Bang, PD, basic sliding modes, and integral sliding modes together with super-twisting. We show that the sliding-mode-based controllers diminish chattering and reject matched perturbations.

The organization of this paper is as follows: Section 2 gives a brief introduction to motor algebra. Section 3 explains the concepts of screw lines, twists, and momenta and the Lie bracket and co-bracket. Section 4 presents the Lagrangian and Hamiltonian formalisms. In Section 5, we explain the dynamics of n -robot links and their local Hamiltonian mechanics on phase space using the geometric algebra framework. In Section 5, we explain the dynamic of n robot links. In Section 6 using screw theory, the iterative Newton–Euler algorithm is introduced to compute locally at each joint the dynamic and its Hamiltonian. Section 7 is devoted to the experimental part comparing the controllers: Bang–Bang, PD, and sliding modes, which are formulated in the phase space. Section 8 presents the conclusion.

2. Motor algebra

For a gentle introduction to geometric algebra see ref. [24]. The word *motor* is an abbreviation of “moment and vector.” Clifford introduced motors with the name bi-quaternions [25]. Motors are isomorph to dual quaternions with the necessary condition $I^2 = 0$, by dual quaternions $\epsilon^2 = 0$. They can be found in the special 4D even subalgebra of $G_{3,0,1}$. This even subalgebra is denoted by $G_{3,0,1}^+$ and is only spanned via a bivector basis, as follows:

$$\underbrace{1}_{\text{scalar}}, \underbrace{e_2e_3, e_3e_1, e_1e_2, e_4e_1, e_4e_2, e_4e_3}_6 \text{ bivectors}, \underbrace{I}_{\text{unit pseudoscalar}}. \tag{1}$$

This kind of basis allows us to represent spinors, which are composed of scalar and bivector terms. A motor represents a special kind of rotor. Because a Euclidean transformation includes both rotation and translation, we will show below that a motor involves both rotation and translation transformations. But first, we must show the relationship between motors and screw motion theory.

Note also that the dual of a scalar is the pseudoscalar I and that the duals of the first three basis bivectors are the following three bivectors $(e_2e_3)^* = e_2e_3I = e_4e_1$.

2.1. Motors, rotors, and translators in $G_{3,0,1}^+$

Since a rigid motion consists of the rotation and translation transformations, it should be possible to split a motor multiplicatively in terms of these two transformations, which we will call a rotor and a *translator*. The equation of a rotor in its Euler representation for a rotation with an angle θ is given by

$$\mathbf{R} = a_0 + a_1e_2e_3 + a_2e_3e_1 + a_3e_1e_2 = a_0 + \mathbf{a} = e^{\frac{\theta}{2}\mathbf{n}} = \cos\left(\frac{\theta}{2}\right) + \sin\left(\frac{\theta}{2}\right)\mathbf{n}, \tag{2}$$

where \mathbf{n} is the unit 3D bivector of the rotation axis spanned by the bivector basis e_2e_3, e_3e_1, e_1e_2 , and $a_c, a_s \in \mathbb{R}$. A 3D translation in motor algebra is defined by $T_c = 1 + I\frac{t_c}{2}$ and called a translator. If we apply a translator from the left to rotor \mathbf{R} , and then apply the translator’s reversion from the right, we get a modified rotor,

$$\mathbf{R}_s = T_c\mathbf{R}\tilde{T}_c = \cos\left(\frac{\theta}{2}\right) + \sin\left(\frac{\theta}{2}\right)(\mathbf{n} + I\mathbf{m}) = \cos\left(\frac{\theta}{2}\right) + \sin\left(\frac{\theta}{2}\right)I. \tag{3}$$

Since a motor is applied from the left and conjugated from the right, we should use the half of t_s in the spinor expression of T_s when we define the motor in its Euler representation

$$\mathbf{M} = T_s\mathbf{R}_s = e^{\frac{(\theta+d)}{2}I} = \cos\left(\frac{\theta}{2} + I\frac{d}{2}\right) + \sin\left(\frac{\theta}{2} + I\frac{d}{2}\right)I. \tag{4}$$

2.2. Properties of motors

A general motor can be expressed as

$$\mathbf{M}_\alpha = \alpha\mathbf{M}, \tag{5}$$

where $\alpha \in \mathbb{R}$ and \mathbf{M} is a unit motor, as explained in the previous sections. In this section, we will employ unit motors. The norm of a motor \mathbf{M} is defined as follows:

$$|\mathbf{M}| = \mathbf{M}\tilde{\mathbf{M}} = T_s\mathbf{R}_s\tilde{T}_s = 1, \tag{6}$$

where $\tilde{\mathbf{M}}$ is the conjugate motor and 1 is the identity of the motor multiplication.

Now we can show that the combination of two rigid motions can be expressed using two consecutive motors. The resultant motor describes the overall displacement, namely,

$$\mathbf{M}_c = \mathbf{M}_a\mathbf{M}_b = (\mathbf{R}_{s_a} + I\mathbf{R}'_{s_a})(\mathbf{R}_{s_b} + I\mathbf{R}'_{s_b}) = \mathbf{R}_{s_c} + I\mathbf{R}'_{s_c}. \tag{7}$$

3. Screws, twists, and momenta

The dynamic motor equation is given by

$$\dot{M} = \frac{1}{2}sM. \tag{8}$$

Assuming that the screw motion of the body is constant through time, the dynamic motor Eq. (8) can be integrated to give

$$M(t) = e^{-\frac{s}{2}t} M(0) = e^{-\frac{\omega + I \left[\dot{t} + \frac{t}{2} \chi \omega \right]}{2}} M(0) = e^{-\frac{\omega + Iv}{2}t} M(0). \tag{9}$$

This represents a motor that rotates with a constant-frequency rotation in the right-hand sense and has a constant linear velocity as well.

Having described finite motion in terms of motors to represent the Lie group of rigid motion $SE(3)$, the logical question of infinitesimal motions or the velocities arises. These are described by elements of the Lie algebra $se(3)$. There are two most useful interpretations to describe the relation between $SE(3)$ and $se(3)$, one as the tangent vectors to the identity element or another as the set of left-invariant vector fields on the group.

3.1. Twists

From Eq. (9), one can derive the spatial velocity or twist s as follows:

$$s = 2\dot{M}\tilde{M} = 2(\dot{T}R + T\dot{R})\tilde{R}\tilde{T} = (\omega + I\dot{t}) = (\omega + Iv), \tag{10}$$

where the angular-velocity bivector ω is the dual of the linear-velocity bivector v .

3.2. Momenta and wrenches

The affine transformation via the matrix N of the twist s yields the momentum which written using 6×1 vectors reads

$$\wp = j + Ip = N(s) \equiv Ns = \begin{bmatrix} j \\ p \end{bmatrix} = \begin{bmatrix} I & m[c]_x \\ m[c]_x^T & mI \end{bmatrix} \begin{bmatrix} \omega \\ v \end{bmatrix} \equiv \begin{bmatrix} I\omega + m[c]_x v \\ m[c]_x^T \omega + mv \end{bmatrix}, \tag{11}$$

where I is the pseudoscalar, I is the inertia, m is the mass, and I is the 3×3 identity matrix. Note that inertia matrix N acts as a dualizing matrix.

The kinetic energy of the body is given by an inner product of the momentum co-twist and the twist

$$\begin{aligned} E_K &= \frac{1}{2}\wp \cdot s = \frac{1}{2}N(s) \cdot s = \frac{1}{2}(j \cdot w + p \cdot v) \\ &\equiv \frac{1}{2}(Ns)^T \cdot s = \frac{1}{2}s^T Ns. \end{aligned} \tag{12}$$

The time derivative of a twist s is the bivector acceleration \dot{s}

$$\dot{s} = \frac{ds}{dt} = \dot{\omega} + Iv \equiv \begin{bmatrix} \frac{d\omega}{dt} \\ \frac{dv}{dt} \end{bmatrix} = \begin{bmatrix} \dot{\omega} \\ \dot{v} \end{bmatrix} \tag{13}$$

and for N time invariant, the time derivative of a momentum \wp is the wrench \mathbf{w} , namely

$$\begin{aligned} \mathbf{w} &= \boldsymbol{\tau} + I\mathbf{f} \\ &= \dot{\wp} = \frac{d}{dt}\mathbf{j} + I\frac{d}{dt}\mathbf{p} = \frac{d}{dt}(I(\boldsymbol{\omega}) + m\mathbf{c}\wedge\mathbf{v}) + I\frac{d}{dt}(m\boldsymbol{\omega}\wedge\mathbf{c} + m\mathbf{v}) \\ &= N(\dot{\mathbf{s}}) = N\left(\frac{d\mathbf{s}}{dt}\right) = N(\dot{\boldsymbol{\omega}} + I\dot{\mathbf{v}}), \end{aligned} \tag{14}$$

where $\boldsymbol{\tau}$ and \mathbf{f} stand for the bivectors torque and force, respectively. This equation of the wrench written using 6×1 vectors reads

$$\mathbf{w} = \begin{bmatrix} \boldsymbol{\tau} \\ \mathbf{f} \end{bmatrix} = N\left(\begin{bmatrix} \frac{d\boldsymbol{\omega}}{dt} \\ \frac{d\mathbf{v}}{dt} \end{bmatrix}\right) = \begin{bmatrix} I & m[\mathbf{c}]_x \\ m[\mathbf{c}]_x^T & mI \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{\omega}} \\ \dot{\mathbf{v}} \end{bmatrix} = \begin{bmatrix} I\dot{\boldsymbol{\omega}} + m[\mathbf{t}]_x\dot{\mathbf{v}} \\ m[\mathbf{t}]_x^T\dot{\boldsymbol{\omega}} + mI\dot{\mathbf{v}} \end{bmatrix}. \tag{15}$$

The inner product between the wrench and the twist is a scalar

$$\mathbf{w} \cdot \mathbf{s} = \boldsymbol{\tau} \cdot \boldsymbol{\omega} + \mathbf{f} \cdot \mathbf{v} \equiv N(\dot{\mathbf{s}}) \cdot \mathbf{s} = \frac{1}{2}\dot{\mathbf{s}}^T N \mathbf{s}. \tag{16}$$

The significance of the wrench is that it can be used to represent the momentum of a rigid body, and similarly, they can describe the action of a wrench on a rigid body, that is, a torque/force bivector.

Finally, the group action on the twist and wrenches using motors are given by

$$\begin{aligned} \mathbf{s}' &= M\mathbf{s}\tilde{M}, \\ \mathbf{w}' &= M\mathbf{w}\tilde{M}. \end{aligned} \tag{17}$$

3.3. Lie bracket

Consider the twist \mathbf{s}_j attached to a j -joint of a moving rigid body; as a function of the time, the equation of an infinitesimal motion in terms of an instantaneous velocity twist \mathbf{s}_i is given by

$$\mathbf{s}_j(t) = M(t)\mathbf{s}_j(0)\tilde{M}(t) = e^{\frac{1}{2}\mathbf{s}_i t}\mathbf{s}_j(0)e^{-\frac{1}{2}\mathbf{s}_i t}, \tag{18}$$

where $\mathbf{s}_i = \boldsymbol{\omega}_i + I\mathbf{v}_i$ and $\mathbf{s}_j = \boldsymbol{\omega}_j + I\mathbf{v}_j$. Differentiating and setting $t = 0$, we get the action of \mathbf{s}_i on \mathbf{s}_j , namely

$$\begin{aligned} \frac{d}{dt}\mathbf{s}_j(t) &= \left(\frac{d}{dt}e^{\frac{1}{2}\mathbf{s}_i t}\mathbf{s}_j(0)e^{-\frac{1}{2}\mathbf{s}_i t}\right)\Big|_{t=0} = \left(\frac{1}{2}\mathbf{s}_i e^{\frac{1}{2}\mathbf{s}_i t}\mathbf{s}_j(0)e^{-\frac{1}{2}\mathbf{s}_i t} - \frac{1}{2}e^{\frac{1}{2}\mathbf{s}_i t}\mathbf{s}_j(0)e^{-\frac{1}{2}\mathbf{s}_i t}\mathbf{s}_i\right)\Big|_{t=0} \\ &= \frac{1}{2}(\mathbf{s}_i\mathbf{s}_j - \mathbf{s}_j\mathbf{s}_i) = [\mathbf{s}_i, \mathbf{s}_j] \end{aligned} \tag{19}$$

where $[\mathbf{s}_i, \mathbf{s}_j]$ is known as the Lie bracket of \mathbf{s}_i and \mathbf{s}_j . Since this result does not depend on where the measurement time begins, in general for two instantaneous twists \mathbf{s}_i and \mathbf{s}_j we can write

$$\frac{d}{dt}\mathbf{s}_j(t) = [\mathbf{s}_i, \mathbf{s}_j] = [(\boldsymbol{\omega}_i + I\mathbf{v}_i), (\boldsymbol{\omega}_j + I\mathbf{v}_j)] \equiv \begin{bmatrix} \boldsymbol{\omega}_i \times \boldsymbol{\omega}_j \\ \mathbf{v}_i \times \mathbf{w}_j + \mathbf{w}_i \times \mathbf{v}_j \end{bmatrix}. \tag{20}$$

3.4. Lie co-bracket

In general for instantaneous twist \mathbf{s}_i and wrench \mathbf{w}_j , we can write

$$\frac{d}{dt}\mathbf{w}_j(t) = \{\mathbf{s}_i, \mathbf{w}_j\} = \{(\boldsymbol{\omega}_i + I\mathbf{v}_i), (\boldsymbol{\tau}_j + I\mathbf{f}_j)\} \equiv \begin{bmatrix} \mathbf{w}_i \times \boldsymbol{\tau}_j + \mathbf{v}_i \times \mathbf{f}_j \\ \boldsymbol{\omega}_i \times \mathbf{f}_j \end{bmatrix}. \tag{21}$$

Note that a co-bracket can be expressed in terms of a Lie bracket as follows, since a wrench is $\mathbf{w}_j = N(\mathbf{s}_j)$ then

$$\{\mathbf{s}_i, \mathbf{w}_j\} = N([\mathbf{s}_i, \mathbf{s}_j]). \tag{22}$$

4. Lagrangian and Hamiltonian formalisms

Hamiltonian mechanics was first formulated by William Rowan Hamilton in 1833, starting from Lagrangian mechanics, a previous reformulation of classical mechanics introduced by Joseph Louis Lagrange in 1788.

The Euler–Lagrange equation of an electromechanical or robot systems is given by [26]

$$M(\theta)\ddot{q} + C(q, \dot{q}) + G(q) = \tau, \quad (23)$$

where $M(\theta)$, $C(q, \dot{q})$, $G(q)$, and τ stand for the inertia matrix, Coriolis matrix, the gravity, and the torque, respectively.

The Lagrange's equation for electromechanical or robot systems is given by

$$\frac{d\partial_{\dot{\theta}}L(\theta, \dot{\theta})}{dt} - \partial_{\theta}L(\theta, \dot{\theta}) = \tau, \quad (24)$$

where L is the Lagrangian of the system that is computed as the difference between the kinetic and potential energies as follows:

$$L(\theta, \dot{\theta}) = K(\theta, \dot{\theta}) - V(\theta). \quad (25)$$

The generalized momentum is defined as

$$p = \partial_{\dot{\theta}}L(\theta, \dot{\theta}). \quad (26)$$

The Hamiltonian of canonical coordinates i is given by

$$H = H(q, p, t) = \sum_i \dot{q}^i p_i - L(q, \dot{q}, t), \quad (27)$$

The partial derivation with respect to p_i gives the first Hamilton equation

$$\frac{\partial \mathcal{H}(q, p, t)}{\partial p_i} = \dot{q}_i. \quad (28)$$

To compute the second equation, it is necessary to take the partial derivation of Eq. (27) with respect to q_i :

$$\frac{\partial \mathcal{H}(q, p, t)}{\partial q_i} = -\frac{\partial \mathcal{L}(q, \dot{q}, t)}{\partial q_i}. \quad (29)$$

Using the Legendre's transformation, the Hamiltonian of canonical coordinates of Eq. (27) is transformed to the Hamiltonian in the phase space

$$H(\theta, p) = p \cdot \dot{\theta} - L(\theta, \dot{\theta}). \quad (30)$$

Taking the derivative of Eq. (30), we get

$$\partial_{\theta}H(\theta, p) = \partial_{\theta}(p \cdot \dot{\theta} - L(\theta, \dot{\theta})) = -\partial_{\theta}L(\theta, \dot{\theta}), \quad (31)$$

henceforth

$$\partial_{\theta}H(\theta, p) = -\partial_{\theta}L(\theta, \dot{\theta}). \quad (32)$$

Using Eqs. (24, 26, 30), we obtain

$$\dot{p} - \partial_{\theta}L(\theta, \dot{\theta}) = \tau. \quad (33)$$

Now taking Eqs. (24, 32, 33), the derivative of the momentum reads

$$\dot{p} = \tau - \partial_{\theta}H(\theta, p). \quad (34)$$

Similarly,

$$\partial_p H(\theta, p) = \partial_p(p \cdot \dot{\theta} - L(\theta, \dot{\theta})) = \dot{\theta}, \quad (35)$$

so

$$\dot{\theta} = \partial_p H(\theta, p). \tag{36}$$

Eqs. (34, 36) are known as the Hamilton’s equations.

In an autonomous system, it is possible to represent the Hamiltonian and the Lagrangian equations as follows:

$$L(\theta, \dot{\theta}) = K(\theta, \dot{\theta}) - V(\theta), \tag{37}$$

$$H(\theta, p) = K(\theta, p) + V(\theta), \tag{38}$$

From Eqs. (34, 38), one obtains

$$\dot{p} = \tau - \partial_{\theta} H(\theta, p) = \tau - \partial_{\theta} K(\theta, p) - \partial_{\theta} V(\theta) = -\tau - \partial_{\theta} K(\theta, p) - G(\theta). \tag{39}$$

So considering Eq. (26) and the Euler–Lagrange Eq. (23), one can derive

$$p = \partial_{\dot{\theta}} L = I(\theta)\dot{\theta}, \tag{40}$$

where the velocity $\dot{\theta}$ is transformed affine to the momentum via the inertia matrix $I(\theta)$, thus the velocity is

$$\dot{\theta} = I(\theta)^{-1}p. \tag{41}$$

The kinetic energy in terms of coordinates reads

$$K(\theta, \dot{\theta}) = \frac{1}{2}\dot{\theta}^T I(\theta)\dot{\theta}, \tag{42}$$

so using Eq. (40), it can be written in terms of the moment

$$K(\theta, p) = \frac{1}{2}p^T I(\theta)^{-1}p. \tag{43}$$

5. Dynamics of n robot links

In this section, first the Newton–Euler algorithm for the computing of dynamics is explained. Next, we show the iterative computing of the local Hamiltonian and the derivative of the momenta.

5.1. Newton–Euler recursive algorithm for dynamics

Selig [27] formulated an iterative algorithm using screw theory but in terms of vectors and matrices. In this work, the Newton–Euler for n links robots is easily formulated as an iterative algorithm using screw theory in the motor algebra framework $G_{3,0,1}^+$.

Let us show the equations for $N = 5$, first, we write the equation of motion for each link, and then we can conveniently group them.

$$\begin{aligned} \mathbf{w}_5 + \mathbf{R}_5 + \mathbf{G}_5 &= \mathbf{N}_5(\dot{\mathbf{s}}_5) + \{\mathbf{s}_5, \mathbf{N}_5(\mathbf{s}_5)\}, \\ \mathbf{w}_4 + \mathbf{R}_4 + \mathbf{G}_4 - \mathbf{w}_5 - \mathbf{R}_5 &= \mathbf{N}_4(\dot{\mathbf{s}}_4) + \{\mathbf{s}_4, \mathbf{N}_4(\mathbf{s}_4)\}, \\ \mathbf{w}_3 + \mathbf{R}_3 + \mathbf{G}_3 - \mathbf{w}_4 - \mathbf{R}_4 &= \mathbf{N}_3(\dot{\mathbf{s}}_3) + \{\mathbf{s}_3, \mathbf{N}_3(\mathbf{s}_3)\}, \\ \mathbf{w}_2 + \mathbf{R}_2 + \mathbf{G}_2 - \mathbf{w}_3 - \mathbf{R}_3 &= \mathbf{N}_2(\dot{\mathbf{s}}_2) + \{\mathbf{s}_2, \mathbf{N}_2(\mathbf{s}_2)\}, \\ \mathbf{w}_1 + \mathbf{R}_1 + \mathbf{G}_1 - \mathbf{w}_2 - \mathbf{R}_2 &= \mathbf{N}_1(\dot{\mathbf{s}}_1) + \{\mathbf{s}_1, \mathbf{N}_1(\mathbf{s}_1)\}, \end{aligned} \tag{44}$$

These equations can be rearranged as we did with the two-link robot, namely

$$\mathbf{w}_5 + \mathbf{R}_5 = \mathbf{N}_5(\dot{\mathbf{s}}_5) + \{\mathbf{s}_5, \mathbf{N}_5(\mathbf{s}_5)\} - \mathbf{G}_5, \tag{45}$$

$$\mathbf{w}_4 + \mathbf{R}_4 = \mathbf{N}_4(\dot{\mathbf{s}}_4) + \mathbf{N}_5(\dot{\mathbf{s}}_5) + \{\mathbf{s}_4, \mathbf{N}_4(\mathbf{s}_4)\} + \{\mathbf{s}_5, \mathbf{N}_5(\mathbf{s}_5)\} - \mathbf{G}_5 - \mathbf{G}_4,$$

⋮

$$\mathbf{w}_1 + \mathbf{R}_1 = \sum_{j=i}^5 [\mathbf{N}_j(\dot{\mathbf{s}}_j) + \{\mathbf{s}_j, \mathbf{N}_j(\mathbf{s}_j)\} - \mathbf{G}_j], \tag{46}$$

where the velocity screw or twist \mathbf{s}_j at the $\{1, 2, \dots, 5\}$ -link is computed as follows:

$$\mathbf{s}_1 = \dot{\theta}_1 \mathbf{L}_1, \quad \mathbf{s}_2 = \dot{\theta}_2 \mathbf{L}_2 + \dot{\theta}_1 \mathbf{L}_1, \quad \dots, \quad \mathbf{s}_5 = \sum_{k=1}^5 \dot{\theta}_k \mathbf{L}_k. \tag{47}$$

The accelerations at each link are computed taking the time derivative of the velocities \mathbf{s}_i

$$\begin{aligned} \dot{\mathbf{s}}_1 &= \ddot{\theta}_1 \mathbf{L}_1 + \dot{\theta}_1 \dot{\mathbf{L}}_1, \quad \dot{\mathbf{s}}_2 = \ddot{\theta}_2 \mathbf{L}_2 + \dot{\theta}_2 \dot{\mathbf{L}}_2 + \ddot{\theta}_1 \mathbf{L}_1 + \dot{\theta}_1 \dot{\mathbf{L}}_1, \quad \dots, \\ \dot{\mathbf{s}}_5 &= \sum_{k=1}^5 \left[\ddot{\theta}_k \mathbf{L}_k + \sum_{l=1}^{k-1} \dot{\theta}_l \dot{\theta}_k [\mathbf{L}_l, \mathbf{L}_k] \right]. \end{aligned} \tag{48}$$

The momentum is computed as follows:

$$\mathcal{P}_j = \mathbf{N}_j(\mathbf{s}_j). \tag{49}$$

According to Eq. (46), a i -wrench is computed as follows:

$$\mathbf{w}_i = \sum_{j=i}^5 [\mathbf{N}_j(\dot{\mathbf{s}}_j) \cdot \mathbf{L}_i + \{\mathbf{s}_j, \mathbf{N}_j(\mathbf{s}_j)\} \cdot \mathbf{L}_i - \mathbf{G}_j \cdot \mathbf{L}_i] - \mathbf{R}_i. \tag{50}$$

Pairing each Eq. (46) with its respective screw line \mathbf{L}_i to get rid of the reaction wrenches, we get the five joint torques as follows:

$$\tau_i = \sum_{j=i}^5 [\mathbf{N}_j(\dot{\mathbf{s}}_j) \cdot \mathbf{L}_i + \{\mathbf{s}_j, \mathbf{N}_j(\mathbf{s}_j)\} \cdot \mathbf{L}_i - \mathbf{G}_j \cdot \mathbf{L}_i]. \tag{51}$$

Since the wrench due to the gravity on the j -link is a force acting on the link's center of mass, considering that the z -axis of the coordinate system is parallel to the gravity vector, the gravity wrench can then be written as

$$\mathbf{G}_j = -mg\mathbf{c}_i \wedge \mathbf{e}_{43} - mge_{43} \equiv \begin{bmatrix} -mg\mathbf{c}_i \wedge \mathbf{e}_{43} \\ 0 \\ 0 \\ -mge_{43} \end{bmatrix}. \tag{52}$$

Considering the following screw:

$$\mathbf{G} = -ge_{43} \equiv \mathbf{G} = \begin{bmatrix} \mathbf{0} \\ 0 \\ 0 \\ -g \end{bmatrix}, \tag{53}$$

one can show that the gravity screw \mathbf{G}_j can be expressed in terms of the inertia matrix \mathbf{N} as follows:

$$\mathbf{G}_j = \mathbf{N}_j(\mathbf{G}). \tag{54}$$

Using this result, one can write the equations of motion even in a more compact manner as

$$\tau_i = \sum_{j=i}^5 [N_j(\dot{s}_j - G) \cdot L_i + \{s_j, N_j s_j\} \cdot L_i]. \tag{55}$$

Note that the use of the Newton–Euler iterative algorithm makes it easy to compute the local Hamiltonians of more complex robot systems as n DOF robot arms or n-DOF inverted pendulums.

5.2. Iterative computing of the local Hamiltonian and the derivative of the momenta

The next equations are derived according to Eqs. (24–34). The local Lagrangian L_i of the robot manipulator at the i -joint can be computed as the difference between the kinetic and potential energies as follows:

$$L(\theta, \dot{\theta})_i = K(\theta, \dot{\theta})_i - V(\theta)_i. \tag{56}$$

The momenta p_i is defined as

$$p_i = \partial_{\dot{\theta}_i} L(\theta, \dot{\theta})_i. \tag{57}$$

The Hamiltonian of canonical coordinates at the i -joint is given by

$$H(q_i, p_i)_i = \sum_i \dot{q}_i p_i - L(q_i, \dot{q}^i) \tag{58}$$

Using the Legendre’s transformation, the Hamiltonian of canonical coordinates of Eq. (58) is transformed to the Hamiltonian in the phase space at the i -joint

$$H(\theta_i, p_i)_i = p_i \cdot \dot{\theta}_i - L(\theta_i, \dot{\theta}_i). \tag{59}$$

Taking the derivative of Eq. (59), we get

$$\partial_{\theta} H(\theta_i, p_i) = \partial_{\theta_i} (p_i \cdot \dot{\theta}_i - L(\theta_i, \dot{\theta}_i)) = -\partial_{\theta_i} L(\theta_i, \dot{\theta}_i), \tag{60}$$

henceforth

$$\partial_{\theta_i} H(\theta_i, p_i) = -\partial_{\theta_i} L(\theta_i, \dot{\theta}_i). \tag{61}$$

Using Eqs. (24, 57, 59), we obtain

$$\dot{p}_i - \partial_{\theta_i} L(\theta_i, \dot{\theta}_i) = \tau_i. \tag{62}$$

Now taking Eqs. (33, 56, 61), the time derivative of the i -momentum reads

$$\dot{p}_i = \tau_i - \partial_{\theta_i} H(\theta_i, p_i). \tag{63}$$

Thus using these last equations, we can compute iteratively locally at each joint of the robot manipulator the wrench w_i Eq. (50), torque τ_i Eq. (51), the Hamiltonian $H(\theta_i, p_i)$ Eq. (59), and the time derivative of the i -momentum \dot{p}_i Eq. (63). These equations computed by the Newton–Euler iterative algorithm are presented in the next subsection. Moreover, we will use the computed time derivative of the i -momentum to derive a local decentralized controller in terms of a screw theory.

6. Recursive computing of Hamiltonians for n DOF robot Manipulators

The Lagrangian of the system is defined as the difference between the kinetic energy T and potential energy V .

$$L(\dot{q}_j, \dots, \dot{q}_1, q_j, \dots, q_1) = T(\dot{q}_j, \dots, \dot{q}_1, q_j, \dots, q_1) - V(q_j, \dots, q_1). \tag{64}$$

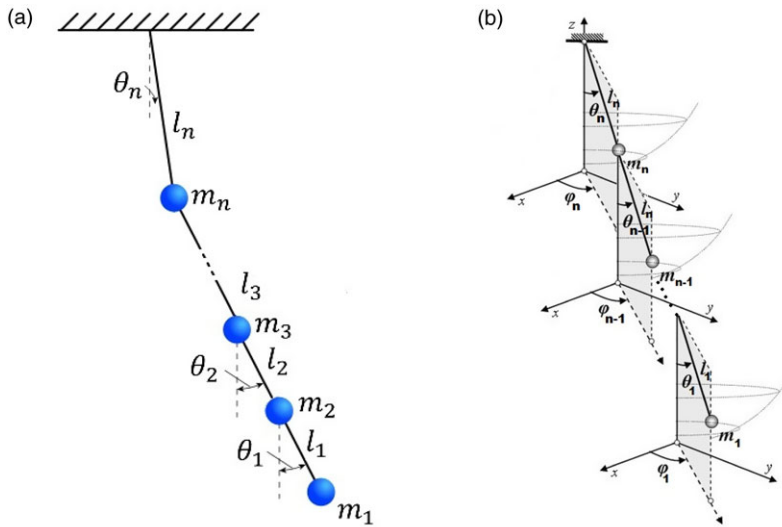


Figure 1. (a) *n*-DOF planar pendulum. (b) *n*-DOF spherical pendulum.

The total Hamiltonian of the system shown in Fig. 1(a) is computed via the Legendre transformation as follows:

$$H = \sum_{j=1}^n p_j \dot{q}_j - L(\dot{q}_j, \dots, \dot{q}_1, q_j, \dots, q_1). \tag{65}$$

For the Bang–Bang control, one requires to compute *n* Hamiltonians and *n* control signals *u_j* for each DOF of the system. The *n* Hamiltonians are computed as follows:

$$H_n = \left(\sum_{j=1}^n p_j \dot{q}_j - L(\dot{q}_j, \dots, \dot{q}_1, q_j, \dots, q_1) \right) - \left(\sum_{j=1}^{n-1} p_j \dot{q}_j - L(\dot{q}_j, \dots, \dot{q}_1, q_j, \dots, q_1) \right), \tag{66}$$

$$H_{n-1} = \left(\sum_{j=1}^{n-1} p_j \dot{q}_j - L(\dot{q}_j, \dots, \dot{q}_1, q_j, \dots, q_1) \right) - \left(\sum_{j=1}^{n-2} p_j \dot{q}_j - L(\dot{q}_j, \dots, \dot{q}_1, q_j, \dots, q_1) \right), \tag{67}$$

⋮

$$H_2 = \left(\sum_{j=1}^2 p_j \dot{q}_j - L(\dot{q}_2, \dot{q}_1, q_2, q_1) \right) - (p_1 \dot{q}_1 - L(\dot{q}_1, q_1)), \tag{68}$$

$$H_1 = p_1 \dot{q}_1 - L(\dot{q}_1, q_1). \tag{69}$$

The sum of the Hamiltonians *H_j* gives the total Hamiltonian of the system.

$$H = \sum_{j=1}^n H_j. \tag{70}$$

The recursive Hamiltonians should fulfill the condition *H_j(0) = 0*. The Hamiltonians are functions of the control *u_j* for each DOF *H_j(p_j, q_j, u_j)*. The computed Hamiltonian depends on the control signal *u_j* and then applying the maximum Pontryagin principle one gets two Hamiltonians a maximum and a minimum. Thus, one obtains 2*n* Hamiltonian, *n* are maximums and *n* are minimums.

$$H_j(p_j, q_j, u_j) = H_j - B_j u_j. \tag{71}$$

For the Bang–Bang control, all the control signals are

$$u_j = \text{if } H(e)_{j_{max}} > H(e)_{j_{min}} \text{ then } -1 \text{ else } 1, \tag{72}$$

where $q_{j_{ref}}$ is the reference for the control related to each DOF q_j , and the error $e = q_j - q_{j_{ref}}$.

One can use the Newton–Euler recursive method of Subsection 5.1 to easily compute the Hamiltonians H_j at each joint of any DOF robot manipulator

$$H(\theta_i, \mathbf{p}_i) = \mathbf{p}_i \cdot \mathbf{s}_i - L(\theta_i, \dot{\theta}_i). \tag{73}$$

This method can be applied for the case of n -D spherical pendulums shown in Fig. 1(b), which have in each joint 2 DOF. Since each joint has 2 DOF, each Hamiltonian H_j is related with two angles ϕ_j and θ_j , thus one split the Hamiltonian H_j in the other two Hamiltonians.

$$H_j = H_{\phi_j} + H_{\theta_j}. \tag{74}$$

In total, one gets for the system $2n$ Hamiltonians.

7. Experimental analysis

In this section, we study the application of localized controllers derived using the local Hamiltonian Eq. (59) and local derivative momenta Eq. (62) for robot systems. The used localized controllers are the PD, slide modes, and Bang–Bang controllers. Since this work focuses on the control of spherical or serial robot manipulators, we cover the most important structures of robot arms. For the comparison, we used three well-known controllers, this is enough to validate the modeling and control using the motor algebra framework.

7.1. Bang–Bang control for a 2 DOF robot manipulator

Using the Newton–Euler iterative algorithm, we can compute the tensors of the Euler–Lagrange state equation with control signals as follows:

$$M \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + C \left(\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}, \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \right) + G(\theta_1, \theta_2) = \begin{bmatrix} b_1 u_1 \\ b_2 u_2 \end{bmatrix}, \tag{75}$$

where

$$M = \begin{bmatrix} (m_1 + m_2)l_1^2 & m_2 l_1 l_2 \cos(\theta_1 - \theta_2) \\ m_2 l_1 l_2 \cos(\theta_1 - \theta_2) & m_2 l_2^2 \end{bmatrix}$$

$$C \left(\begin{bmatrix} \dot{\theta}_2 \\ \dot{\theta}_1 \end{bmatrix}, \begin{bmatrix} \theta_2 \\ \theta_1 \end{bmatrix} \right) = \begin{bmatrix} 0 & m_2 l_1 l_2 \dot{\theta}_2 \sin(\theta_1 - \theta_2) \\ -m_2 l_1 l_2 \dot{\theta}_1 \sin(\theta_1 - \theta_2) & 0 \end{bmatrix}$$

$$G \left(\begin{bmatrix} \theta_2 \\ \theta_1 \end{bmatrix} \right) = \begin{bmatrix} (m_1 + m_2)gl_1 \sin \theta_1 \\ m_2 gl_2 \sin \theta_2 \end{bmatrix}.$$

Furthermore, using the Newton–Euler iterative algorithm of subsection (5.1), we can compute inward for each joint the following local Hamiltonians using generalized coordinates:

$$H_2 = \frac{1}{2} m_2 l_2^2 \dot{\theta}_2^2 - m_2 gl_2 \cos(\theta_2), \tag{76}$$

$$H_1 = \frac{1}{2}(m_1 + m_2)l_1^2\dot{\theta}_1^2 + m_2l_1l_2\dot{\theta}_1\dot{\theta}_2 \cos(\theta_1 - \theta_2) - (m_1 + m_2)gl_1 \cos(\theta_1). \tag{77}$$

Since the $H_1(0) = 0$ and $H_2(0) = 0$, Eq. (76) becomes

$$H_2(\theta_2) = \frac{1}{2}m_2l_2^2\dot{\theta}_2^2 + m_2gl_2(1 - \cos(\theta_2)), \tag{78}$$

$$H_1(\theta_1) = \frac{1}{2}(m_1 + m_2)l_1^2\dot{\theta}_1^2 + m_2l_1l_2\dot{\theta}_1\dot{\theta}_2 \cos(\theta_1 - \theta_2) + (m_1 + m_2)gl_1(1 - \cos(\theta_1)). \tag{79}$$

A control algorithm that checks for the process variable exceeding or falling below the set point is called on/off control. This kind of control is also known as Bang–Bang control because the manipulated variable output of the controller rapidly changes between fully on and fully off involving no intermediate state. It has been shown that Bang–Bang control usually provides imprecise control of the process variable. In this work, we applied control only to revolved joints. A rotation unit is geometrically speaking a dual of a prismatic joint because for the modeling of its rotation one uses rotors for revolved joints and translators for the prismatic joints, thus it is possible to apply Bang–Bang control to prismatic joints.

In this work, Bang–Bang control is applied in a localized manner at each joint using the local Hamiltonians similar to in the state Eq. (75). These control signals $u_i = \pm 1, i = 1, 2$ switch between the scalar functions of H_1 and H_2 .

$$H_2(\theta_2) = \frac{1}{2}m_2l_2^2\dot{\theta}_2^2 + m_2gl_2(1 - \cos(\theta_2)) - b_2\theta_2u_2, \tag{80}$$

$$H_1(\theta_1) = \frac{1}{2}(m_1 + m_2)l_1^2\dot{\theta}_1^2 + m_2l_1l_2\dot{\theta}_1\dot{\theta}_2b \cos(\theta_1 - \theta_2) + (m_1 + m_2)gl_1(1 - \cos(\theta_1)) - b_1\theta_1u_1, \tag{81}$$

Applying in Eqs. (80-82) the maximal principle of Pontryagin and setting the angle references $\theta_{1ref} = 20^\circ$ and $\theta_{2ref} = 30^\circ$ and $u_i = \pm 1, i = 1, 2$ for the switching in the Hamiltonians so that the Hamiltonians reach the zero value, we obtain the following equations:

$$H_{11} = \frac{1}{2}(m_1 + m_2)l_1^2\dot{\theta}_1^2 + m_2l_1l_2\dot{\theta}_1\dot{\theta}_2 \cos((\theta_1 - \theta_{1ref}) - (\theta_2 - \theta_{2ref})) + (m_1 + m_2)gl_1(1 - \cos(\theta_1 - \theta_{1ref})) - b_1(\theta_1 - \theta_{1ref}), \tag{82}$$

$$H_{12} = \frac{1}{2}(m_1 + m_2)l_1^2\dot{\theta}_1^2 + m_2l_1l_2\dot{\theta}_1\dot{\theta}_2 \cos((\theta_1 - \theta_{1ref}) - (\theta_2 - \theta_{2ref})) + (m_1 + m_2)gl_1(1 - \cos(\theta_1 - \theta_{1ref})) + b_1(\theta_1 - \theta_{1ref}), \tag{83}$$

where $u_1 =$ if $H_{12} > H_{11}$ then -1 else 1 and

$$H_{21} = \frac{1}{2}m_2l_2^2\dot{\theta}_2^2 + m_2gl_2(1 - \cos(\theta_2 - \theta_{2ref})) - b_2(\theta_2 - \theta_{2ref}), \tag{84}$$

$$H_{22} = \frac{1}{2}m_2l_2^2\dot{\theta}_2^2 + m_2gl_2(1 - \cos(\theta_2 - \theta_{2ref})) + b_2(\theta_2 - \theta_{2ref}), \tag{85}$$

where $u_2 =$ if $H_{22} > H_{21}$ then -1 else 1. The control signals $u_i = \pm 1, i = 1, 2$ switch between the Hamiltonians H_1 and H_2 surfaces. Figure 3 shows the control signals and the error figures to reach a reference point. Note that even though that the controllers reach the references, they keep switching very rapidly between ± 1 .

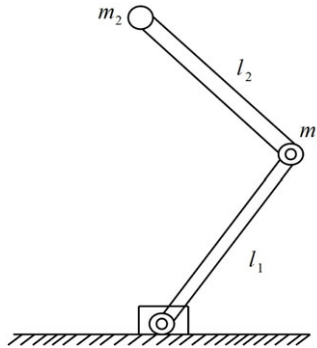


Figure 2. 2 DOF planar robot.

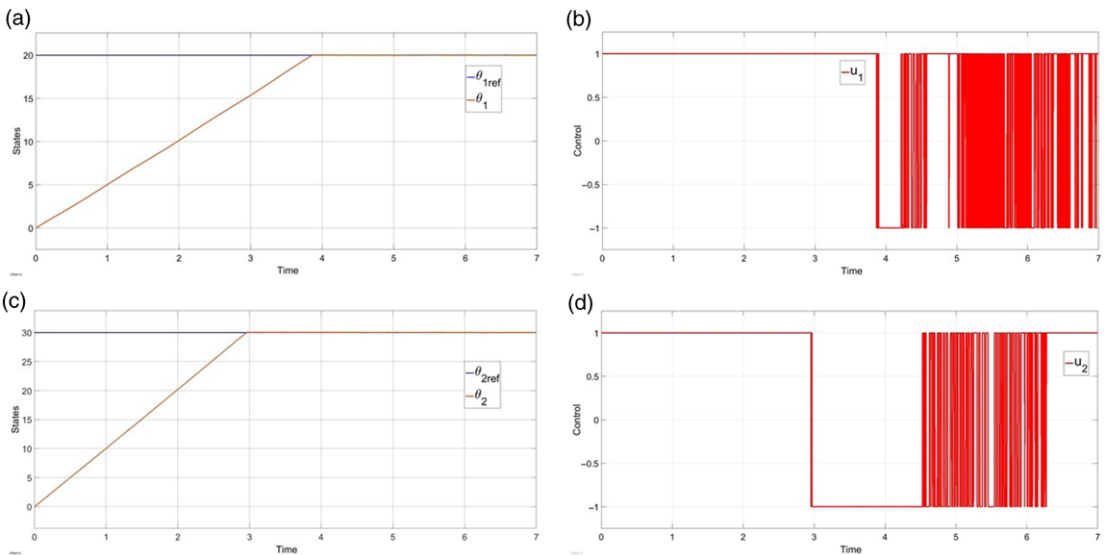


Figure 3. Hamiltonian Bang–Bang controller: (a) state \$\theta_1\$: (b) control \$u_1\$; (c) state \$\theta_2\$: (d) control \$u_2\$.

7.2. PD and sliding mode controller for a 2 DOF robot manipulator

Before constructing the law of control, it is necessary to compute Hamilton’s equations in the screw theory framework. The Hamiltonian on the phase space can be computed using Eqs. (25, 27) as follows:

$$\mathcal{H}(\theta, p, t) = \sum_{j=1}^n \frac{1}{2} \mathbf{P}_j^T N_j^{-1} \mathbf{P}_j + \tilde{g}^T \tilde{c}_j, \tag{86}$$

$$\frac{\partial \mathcal{H}(\theta, p, t)}{\partial p_i} = \frac{1}{\mathbf{S}_i^T N_i \mathbf{S}_i} \left(p_i - \sum_{j=i+1}^n \mathbf{S}_j^T \mathbf{P}_j - \sum_{k=1}^{i-1} \mathbf{S}_i^T \mathbf{P}_{i,k} \right), \tag{87}$$

$$\frac{\partial \mathcal{H}(\theta, p, t)}{\partial \theta_i} = \sum_{j=i}^n \mathbf{P}_j^T [\mathbf{S}_i, N_i^{-1} \mathbf{P}_i] - G_j^T \mathbf{S}_i, \tag{88}$$

where \$\mathbf{S}_i\$ is the twist \$\mathbf{s}_i\$ written as a \$6 \times 1\$ vector and the momentum \$\mathbf{P}_i\$ is a \$6 \times 1\$ vector as well.

According to Eq. (63), the derivative of the momenta at i -joint is

$$\dot{p}_i = \tau_i + \sum_{j=i}^n G_j^T S_i - P_j^T [S_i, N_i^{-1} P_i]. \tag{89}$$

A serial robotic system will track a desired smooth function applying the following decentralized PD law of control:

$$\tau_i = K_{p_i} \tilde{\theta}_i + K_{v_i} \tilde{p}_i + \dot{p}_{d_i} - \sum_{j=i}^n P_j^T [N_i^{-1} P_i, S_i] - G_j^T S_i, \tag{90}$$

where $\tilde{p}_i = p_{d_i} - p_i$ is the error between the desired momentum and the measured momentum and $\tilde{\theta}_i = \theta_{d_i} - \theta_i$ is the error between the desired joint position and the measured joint position and $K_{p_i}, K_{v_i} \in \mathbb{R}^+$.

A robotic system will reach the home position by applying the following decentralized sliding modes law of control:

$$\tau_i = K_i \text{sign}(S_i) + K_{S_i} S_i + \tilde{\theta}_i - \sum_{j=i}^n S_j^T \{S_i, N_j L_i\} - G_j \cdot L_i, \tag{91}$$

where $\{S_i, N_j L_i\} = N_j [S_i, L_i]$ and in each joint, $S_i = -\sum_{j=i}^n L_i^T N_j S_j + \tilde{\theta}_i$ is the sliding surface and $K_i, K_{S_i} \in \mathbb{R}^+$.

Consider the two-link manipulator with revolute joints illustrated in Fig. 2. In addition, the manipulator link lengths are l_1 and l_2 , and the link masses are m_1 and m_2 , where these masses are concentrated at the end of each link.

The dynamic of the proposed robot is computed using (55). Hence, the dynamic of each joint can be written as the following:

$$\tau_1 = \dot{V}_1^T N_1 L_1 + V_1^T N_1 [L_1, V_1] - G_1 \cdot L_1 + \dot{V}_2^T N_2 L_1 + V_2^T N_2 [L_1, V_2] - G_2 \cdot L_1, \tag{92}$$

$$\tau_2 = \dot{V}_2^T N_2 L_2 + V_2^T N_2 [L_2, V_2] - G_2 \cdot L_2. \tag{93}$$

Considering $V_1 = L_1 \dot{\theta}_1$ and $V_2 = L_1 \dot{\theta}_1 + L_2 \dot{\theta}_2$. Thus, the accelerations are constructed by

$$\dot{V}_1 = \frac{d}{dt} (L_1 \dot{\theta}_1) = L_1 \ddot{\theta}_1, \tag{94}$$

$$\dot{V}_2 = \frac{d}{dt} (L_1 \dot{\theta}_1 + L_2 \dot{\theta}_2) = L_1 \ddot{\theta}_1 + L_2 \ddot{\theta}_2 + ad(V_1) L_2 \dot{\theta}_2 = L_1 \ddot{\theta}_1 + L_2 \ddot{\theta}_2 + [L_1, L_2] \dot{\theta}_1 \dot{\theta}_2. \tag{95}$$

Taking the previous accelerations (\dot{V}_1, \dot{V}_2) and velocities (V_1, V_2) into (92) and (93), the dynamic of the robot can be written as follows:

$$\begin{aligned} \tau_1 &= L_1^T N_1 L_1 \ddot{\theta}_1 + L_1^T N_2 L_1 \ddot{\theta}_1 + L_1^T N_2 L_2 \ddot{\theta}_2 + 2L_1^T N_2 [L_1, L_2] \dot{\theta}_1 \dot{\theta}_2 \\ &\quad + L_2^T N_2 [L_1, L_2] \dot{\theta}_2^2 - G_1^T L_1 - G_2^T L_1, \end{aligned} \tag{96}$$

$$\tau_2 = L_1^T N_2 L_2 \ddot{\theta}_1 + L_2^T N_2 L_2 \ddot{\theta}_2 - L_1^T N_2 [L_1, L_2] \dot{\theta}_1^2 - G_2^T L_2. \tag{97}$$

7.2.1. PD controller

Given the state variables $x_1, x_2 \in \mathbb{R}^2$, where $x_1 = (\theta_1, \theta_2)^T$ and $x_2 = (\dot{\theta}_1, \dot{\theta}_2)^T$, the closed-loop system can be represented using Eq. (90) as follows:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ - \left(\begin{matrix} L_1^T N_1 L_1 + L_1^T N_2 L_1 & L_1^T N_2 L_2 \\ L_1^T N_2 L_2 & L_2^T N_2 L_2 \end{matrix} \right)^{-1} \left[\begin{pmatrix} K_{p1} & 0 \\ 0 & K_{p2} \end{pmatrix} x_1 + \begin{pmatrix} K_{v1} & 0 \\ 0 & K_{v2} \end{pmatrix} x_2 - B(\dot{\theta}_1, \dot{\theta}_2) \right] \end{bmatrix} \tag{98}$$

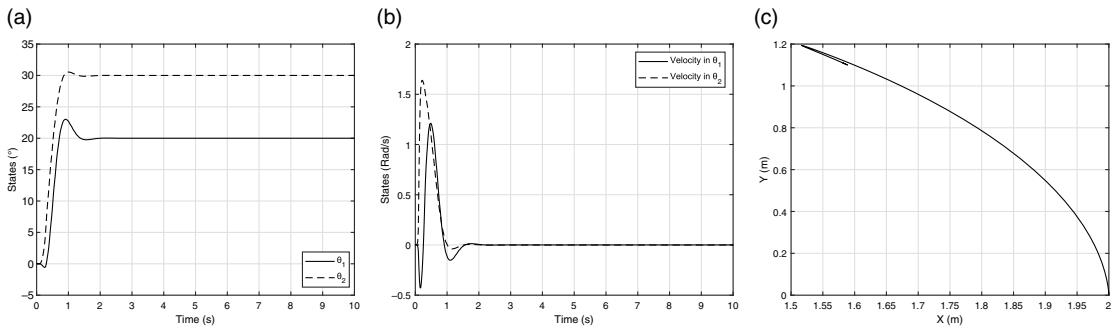


Figure 4. PD controller: (a) states: angles x_1 and angular velocities x_2 . (b) Robot motion.

where

$$B(\dot{\theta}_1, \dot{\theta}_2) = \begin{bmatrix} 2L_1^T N_2 [L_1, L_2] \dot{\theta}_1 \dot{\theta}_2 + L_2^T N_2 [L_1, L_2] \dot{\theta}_2^2 \\ -L_1^T N_2 [L_1, L_2] \dot{\theta}_1^2 \end{bmatrix} \tag{99}$$

To simulate the results of the example, it is necessary to implement the parameters in Table I with $K_{p1}, K_{p2} = 150$ and $K_{v1}, K_{v2} = 50$. The simulations were conducted using the Euler integration method, with a step size of 0.001s. The initial conditions were selected as $x_1(0) = x_2(0) = 0$ and the desired values as $\theta_{d1} = 20^\circ, \theta_{d2} = 30^\circ$. Hence, the results of the example are presented in Fig. 4.

7.2.2. Sliding mode controller

Suppose $x_1, x_2 \in \mathbb{R}^2$, where $x_1 = (\theta_1, \theta_2)^T$ and $x_2 = (\dot{\theta}_1, \dot{\theta}_2)^T$. Then, the closed-loop system using Eq. (91) can be represented by

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ \left(\begin{matrix} L_1^T N_1 L_1 + L_1^T N_2 L_1 & L_1^T N_2 L_2 \\ L_1^T N_2 L_2 & L_2^T N_2 L_2 \end{matrix} \right)^{-1} \left[\begin{matrix} -K_1 \text{sign} [S_1] - K_{S1} \\ -K_2 \text{sign} (S_2) - K_{S2} \end{matrix} \right] - x_2 - B(x_2) \end{bmatrix} \tag{100}$$

where

$$\begin{aligned} S_1 &= L_1^T N_1 L_1 \dot{\theta}_1 + L_1^T N_2 L_1 \dot{\theta}_1 + L_1^T N_2 L_2 \dot{\theta}_2 + \theta_1 \\ S_2 &= L_1^T N_1 L_2 \dot{\theta}_1 + L_2^T N_2 L_2 \dot{\theta}_2 + \theta_2 \\ B(x_2) &= \begin{bmatrix} 2L_1^T N_2 [L_1, L_2] \dot{\theta}_1 \dot{\theta}_2 + L_2^T N_2 [L_1, L_2] \dot{\theta}_2^2 \\ L_2^T N_2 [L_1, L_2] \dot{\theta}_1 \dot{\theta}_2 \end{bmatrix} \end{aligned} \tag{101}$$

To simulate the results of the example, it is necessary to implement the parameters in Table I with $K_1, K_2 = 1$ and $K_{S1}, K_{S2} = 50$. The simulations were conducted using the Euler integration method, with a step size of 0.001s. The initial conditions were selected as $x_1(0) = [45^\circ, 10^\circ]^T$ and $x_2(0) = [0, 0]^T$. Using the sliding mode controller [28], we carry out a tracking experiment. Moreover, it is necessary to take care with the adjustable gains because a small variable cannot easily converge to the position for the case of different robots. Therefore, it is important to adjust the gain depending on the application.

Figure 5(a) and (b) presents the evolution of the angles and angular velocities of the two joints during the tracking. These experiments show that our sliding mode controller performs very well in both stability and tracking. Figure 5 (c) and (d) shows how well the sliding mode controller manages the 2 DOF robot arm to follow a nonlinear curve.

Table I. Parameters of the robot.

Parameter	Value	Unit
m_1	0.25	kg
m_2	0.25	kg
l_1	1.2	m
l_2	1.2	m

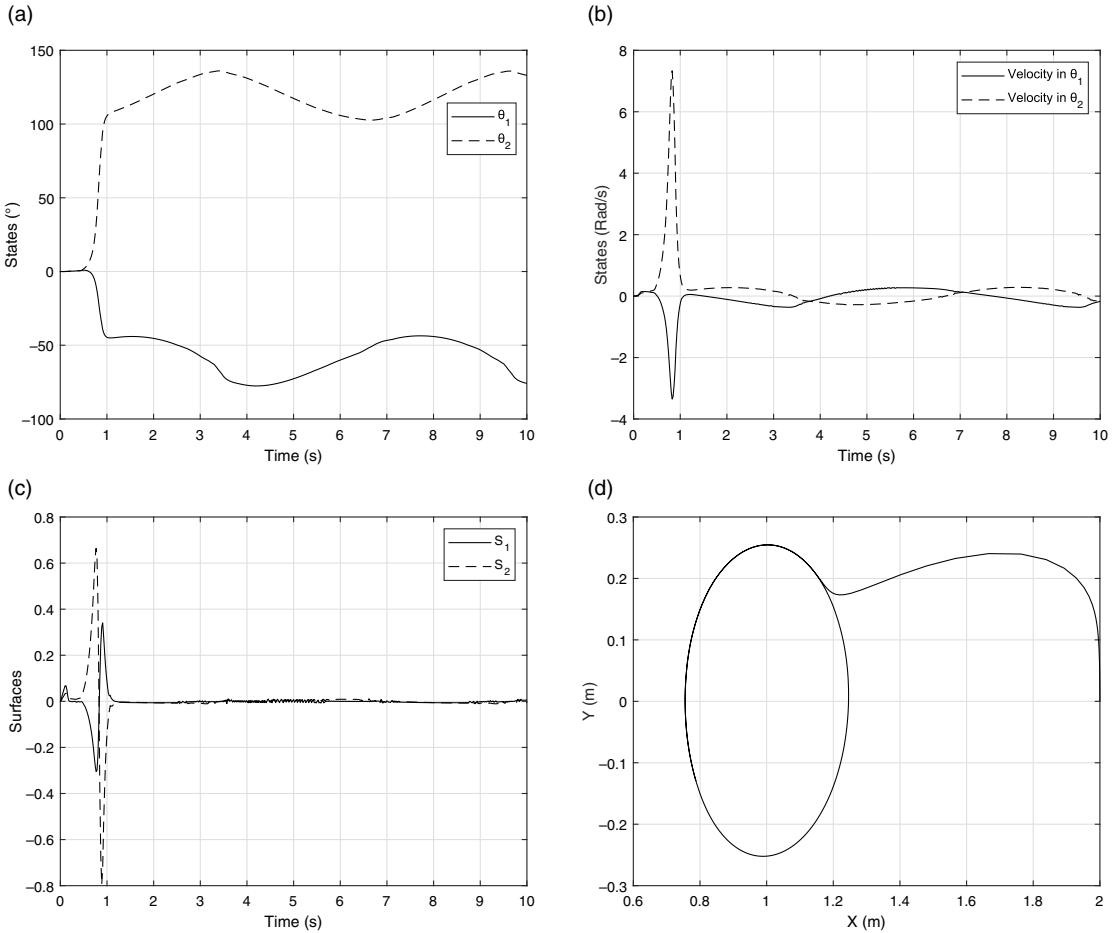


Figure 5. Sliding modes controller: (a) states: angles x_1 ; (b) angular velocities x_2 ; (c) sliding modes surfaces; and (d) robot motion.

7.3. Bang–Bang control for a 4 DOF double spherical pendulum

The Lagrangian associated with the double spherical pendulum is given by

$$L = T - V, \tag{102}$$

where

$$\begin{aligned}
 T = & \frac{1}{2} (m_1 + m_2) l_2^2 (\dot{\theta}_2^2 + \sin^2 \theta_2 \dot{\phi}_2^2) + \frac{1}{2} m_1 l_1^2 (\dot{\theta}_1^2 + \sin^2 \theta_1 \dot{\phi}_1^2) \\
 & + m_1 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin \theta_1 \sin \theta_2 + m_1 l_1 l_2 \cos (\phi_2 - \phi_1) (\cos \theta_1 \cos \theta_2 \dot{\theta}_1 \dot{\theta}_2 + \sin \phi_1 \sin \phi_2 \dot{\phi}_1 \dot{\phi}_2) \\
 & + m_1 l_1 l_2 \sin (\phi_2 - \phi_1) (\cos \theta_2 \sin \theta_1 \dot{\phi}_1 \dot{\phi}_2 - \sin \theta_2 \cos \theta_1 \dot{\phi}_2 \dot{\theta}_1), \tag{103}
 \end{aligned}$$

$$V = - (m_1 + m_2) gl_2 \cos \theta_2 - m_1 gl_1 \cos \theta_1. \tag{104}$$

The motion equations can be obtained from the Lagrangian of the double spherical pendulum

$$M(\theta_j, \phi_j) \begin{pmatrix} \ddot{\theta}_j \\ \ddot{\phi}_j \end{pmatrix} + C(\theta_j, \dot{\theta}_j, \phi_j, \dot{\phi}_j) \begin{pmatrix} \dot{\theta}_j \\ \dot{\phi}_j \end{pmatrix} + G(\theta_j, \phi_j) = Bu. \tag{105}$$

As we showed in the previous example of the double pendulum by applying the V.M. Yepez theorem, it is extremely difficult to compute the Hamiltonian for manipulators of more than 2 DOF. This shows that it is better to compute the Hamiltonian of the system via the Legendre transformation and in a local manner apply the V.M. Yepez theorem.

$$H(q_j, p_j, t) = \sum p_j \dot{q}_j - L(q_j, \dot{q}_j, t). \tag{106}$$

For the double spherical pendulum, the Hamiltonian of the system is given by

$$H = T + V. \tag{107}$$

Based on the Hamiltonian of the 4 DOF of the system, we can derive four Hamiltonians for each DOF which are obtained using the Newton–Euler recursive algorithm explained in Subsection 5.1, so the computed Hamiltonian H_j at the j -join is

$$H_j(\theta_j, \phi_j, \mathbf{p}_j) = \mathbf{p}_j \cdot \mathbf{s}_j - L(\theta_j, \phi_j, \dot{\theta}_j, \dot{\phi}_j). \tag{108}$$

The computed four Hamiltonians are

$$H_4 = \frac{1}{2} (m_1 + m_2) l_2^2 \dot{\theta}_2^2 + m_1 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin \theta_1 \sin \theta_2, \tag{109}$$

$$+ m_1 l_1 l_2 \cos (\phi_2 - \phi_1) (\cos \theta_1 \cos \theta_2 \dot{\theta}_1 \dot{\theta}_2 + \sin \phi_1 \sin \phi_2 \dot{\phi}_1 \dot{\phi}_2) \\ + m_1 l_1 l_2 \sin (\phi_2 - \phi_1) (\cos \theta_2 \sin \theta_1 \dot{\phi}_1 \dot{\phi}_2 - \sin \theta_2 \cos \theta_1 \dot{\phi}_2 \dot{\phi}_1) - (m_1 + m_2) gl_2 \cos \theta_2$$

$$H_3 = \frac{1}{2} (m_1 + m_2) l_2^2 (\sin^2 \theta_2 \dot{\phi}_2^2), \tag{110}$$

$$H_2 = \frac{1}{2} m_1 l_1^2 \dot{\theta}_1^2 - m_1 gl_1 \cos \theta_1, \tag{111}$$

$$H_1 = \frac{1}{2} m_1 l_1^2 (\sin^2 \theta_1 \dot{\phi}_1^2). \tag{112}$$

One applies the maximum principle of Pontryagin to H_1, H_2, H_3 y H_4 to obtain two Hamiltonians a maximum and a minimum for each function, resulting then in eight Hamiltonians, where one has to impose the condition $H_j(0) = 0$. The motion equations are obtained from the system Hamiltonians of the double spherical pendulum including the Bang–Bang control.

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_1} \right) - \frac{\partial L}{\partial \theta_1} = b_1 u_1, \tag{113}$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_2} \right) - \frac{\partial L}{\partial \theta_2} = b_2 u_2, \tag{114}$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\phi}_1} \right) - \frac{\partial L}{\partial \phi_1} = b_3 u_3, \tag{115}$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\phi}_2} \right) - \frac{\partial L}{\partial \phi_2} = b_4 u_4. \tag{116}$$

Figure 7 shows the control signals and the error figures to reach a reference point. Note that even though the controllers reach the references, they keep switching very rapidly between ± 1 .

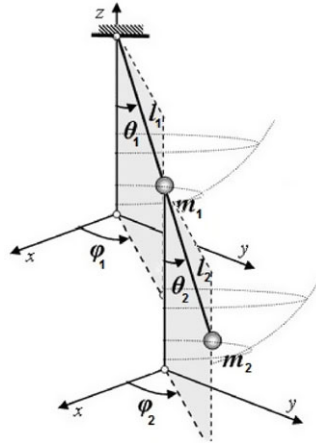


Figure 6. 4 DOF spherical pendulum.

7.4. Sliding mode controller for a 4 DOF double spherical pendulum

Consider the two-link manipulator with four revolute joints as illustrated in Fig. 6. The manipulator link lengths are l_1 and l_2 , and the link masses are m_1 and m_2 , where these masses are concentrated at the end of each link.

The Newton–Euler iterative algorithm is based on Eq. (55), accordingly the resulting wrench at each is given by

$$\tau_i = \sum_{j=i}^5 [N_j(\dot{s}_j - G) \cdot L_i + \{s_j, N_j s_j\} \cdot L_i]. \tag{117}$$

To formulate the sliding-mode-based control, first, we require to represent in state variables the dynamics of the robot arm. This procedure is an extension as presented for the 2 DOF robot arm of Subsection 7.2. We use Eq. (55) and extend Eqs. (100)–(101) of the 2 DOF manipulator to the 4 DOF double spherical pendulum. To simplify the equation, we gather all the terms in matrices as follows:

$$M(x)\ddot{x} + C(\dot{x}, x)\dot{x} + G(x) = \tau, \tag{118}$$

where M is the inertia matrix, C is the Coriolis and centripetal matrix, $G(x)$ is the gravity vector, τ is the actuating torque vector, x is the vector state corresponding with the angular displacement of each joint, and \dot{x} and \ddot{x} are the first and second derivatives of the state vector corresponding to the velocity and acceleration, respectively. The relation between the Hamiltonian and the Euler–Lagrange equations can be established by using the Legendre’s transformation as mentioned in Subsection 4.

Using the previous Eq. (118), the space state model for the angular displacements and their velocities of the joints is defined as follows:

$$\dot{x}_1 = x_2, \tag{119}$$

$$\dot{x}_2 = f(x_1, x_2) + B(x_1, x_2)u, \tag{120}$$

where $x_1 = [\phi_1, \theta_1, \phi_2, \theta_2]^T$ and $x_2 = [\dot{\phi}_1, \dot{\theta}_1, \dot{\phi}_2, \dot{\theta}_2]^T$ are the state variables (angular displacements and velocities, respectively), $f(x_1, x_2) = -M^{-1}(x_1) (C(x_1, x_2)x_2 + G(x_1))$, $B(x_1, x_2) = M^{-1}(x_1)$, $u = \tau$ the signal control laws.

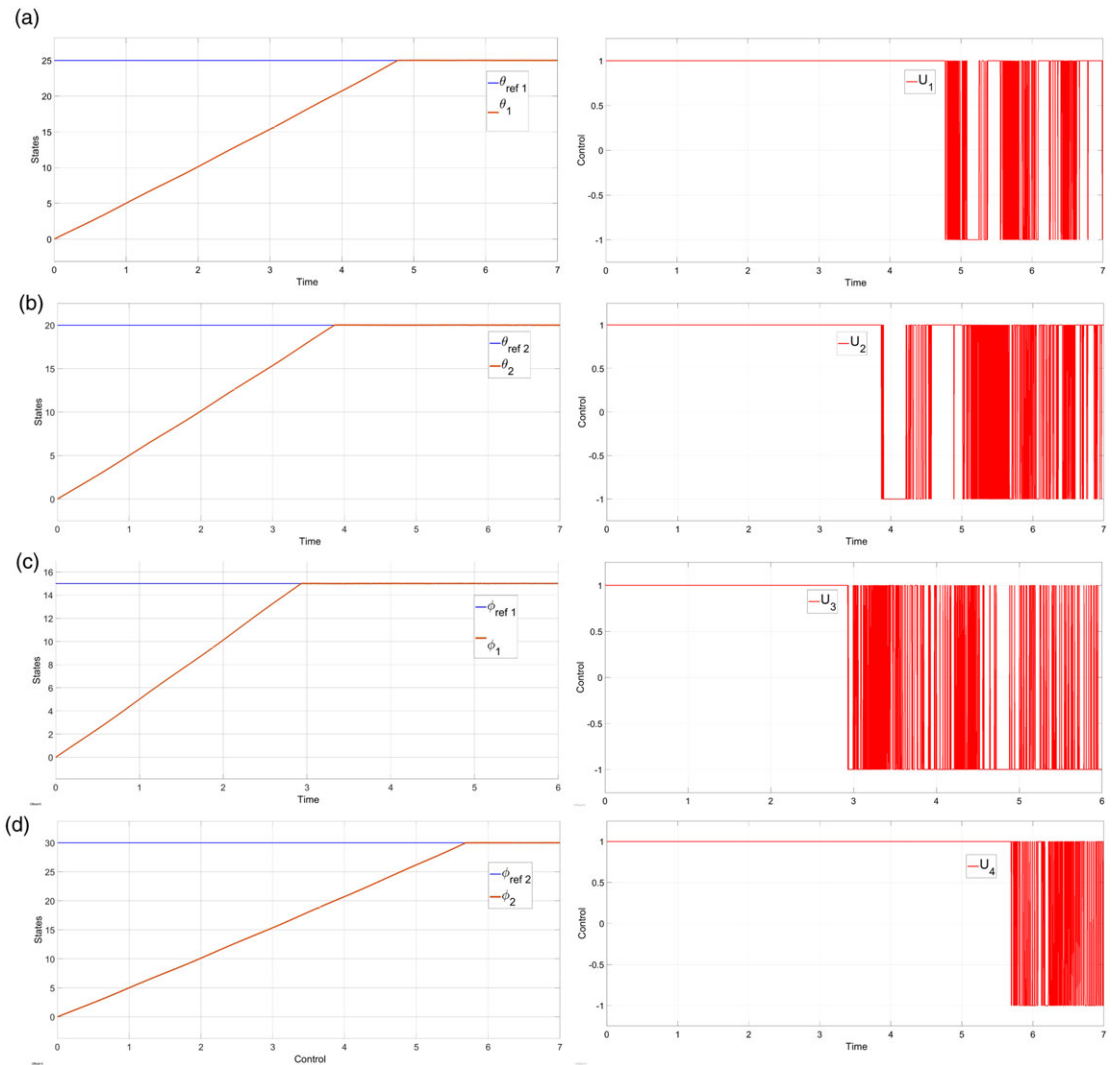


Figure 7. (a) Angle state θ_1 and control u_1 . (b) Angle state θ_2 and control u_2 . (c) Angle state ϕ_1 and control u_3 . (d) Angle state ϕ_2 and control u_4 .

Since the purpose of the controller is to track trajectories, if we define the tracking error as $e_1 = x_d - x_1$ and using (119-120), the error system is described as

$$\begin{aligned} \dot{e}_1 &= \dot{x}_d - \dot{x}_2 = e_2 \\ \dot{e}_2 &= \ddot{x}_d - f(x_1, x_2) - B(x_1, x_2)u. \end{aligned} \tag{121}$$

Because the system can contain uncertainties attributed to parametric variations, unmodeled dynamics, and external disturbances, it is necessary to add a term that contains these uncertainties, so our error system is redefined as

$$\begin{aligned} \dot{e}_1 &= e_2 \\ \dot{e}_2 &= \ddot{x}_d - f(x_1, x_2) - B(x_1, x_2)u_0 - B(x_1, x_2)u_1 + h(x_1, x_2, t), \end{aligned} \tag{122}$$

where $h(x_1, x_2, t)$ satisfies the matching condition [28]. Besides, $h(x_1, x_2, t)$ must be bounded by known positive defined scalar functions such that

$$\|h(x_1, x_2, t)\| \leq h_i^+(x_1, x_2, t), \quad i = 1, \dots, n.$$

Once the error system model of our robotic system is obtained, we proceed to design the law of control. The law of control is defined as

$$u = u_0 + u_1, \quad (123)$$

where u_0 is computed using the inverse dynamic algorithm [29] for the nominal system and u_1 is an integral sliding mode control algorithm that deals with external disturbances and parametric variations.

First, we describe u_0 as follows:

$$u_0 = M(x_1)a_d + C(x_1, x_2)x_2 + G(x_1), \quad (124)$$

where a_d is the desired dynamics that will be designed later.

Substituting the control law u_0 (124) in the second equation of the error system (122), we have the following:

$$\begin{aligned} \dot{e}_2 &= \ddot{x}_d - f(x_1, x_2) + B(x_1, x_2)u_0 + h(x_1, x_2, t) \\ &= \ddot{x}_d - M^{-1}(x_1)(C(x_1, x_2)x_2 + G(x_1)) + M^{-1}(x_1)u_0 + h(x_1, x_2, t) \\ &= \ddot{x}_d - M^{-1}(x_1)(u_0 - C(x_1, x_2)x_2 - G(x_1)) + h(x_1, x_2, t) \\ &= \ddot{x}_d - a_d + h(x_1, x_2, t), \end{aligned} \quad (125)$$

which cancels the internal dynamics. To ensure the tracking of the references, we then define a_d as follows:

$$\begin{aligned} a_d &= \ddot{x}_d - K_p(x_d - x_1) - K_d(\dot{x}_d - \dot{x}_2) \\ &= \ddot{x}_d - K_p e_1 - K_d e_2, \end{aligned} \quad (126)$$

where x_d , \dot{x}_d and \ddot{x}_d are the position, velocity, and acceleration references, and K_p and K_d are 4×4 matrices. Then, the error system with u_0 is proposed as follows:

$$\dot{e}_1 = e_2, \quad (127)$$

$$\dot{e}_2 = K_p e_1 + K_d e_2 + h(x_1, x_2, t). \quad (128)$$

A simple way to design the gain matrices K_p and K_d is

$$\begin{aligned} K_p &= \text{diag}(\omega_1^2, \omega_2^2, \dots, \omega_n^2) \\ K_d &= \text{diag}(2\omega_1, 2\omega_2, \dots, 2\omega_n), \end{aligned} \quad (129)$$

which, leaving aside the term $h(x_1, x_2, t)$, results in a decoupled closed-loop system, where each response of each joint is equal to the response of a critically damped second-order system with natural frequency ω_i , which determines the velocity response of the joints, or equivalently, the decay rate of the tracking error.

It is noteworthy that to compute u_0 , the inverse dynamics can be calculated using the Newton–Euler algorithm described in Section 5. To deal with the term $h(x_1, x_2, t)$, which includes parametric variations and disturbances of the system, the control law u_1 is designed by using the integral sliding mode algorithm as mentioned above. This algorithm presupposes the existence of a previously designed controller for the nominal plant (without disturbances), u_0 , adding to this existing controller a discontinuous term to force the occurrence of sliding modes and thus obtaining a response as if the plant was the nominal system with its closed-loop controller, despite external disturbances and parametric variations of the system.

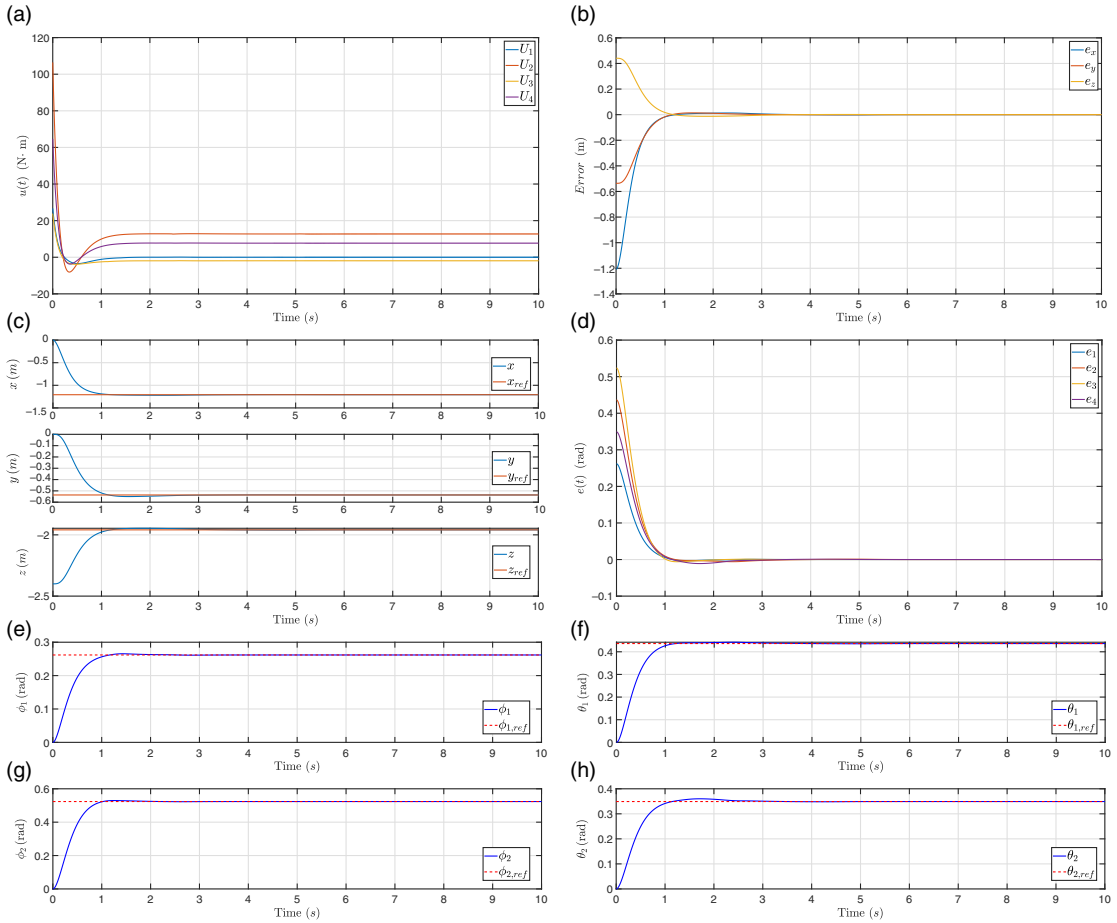


Figure 8. Sliding mode-based control: (a) control signals U_1, U_2, U_3, U_4 . (b) Space x, y, z errors. (c) Motion in space (x, y, z) . (d) State angle errors. (e–h) States angles $\phi_1, \phi_2, \theta_1, \theta_2$.

For u_1 , we define the sliding surface

$$s = \sigma(x) + z, \tag{130}$$

where $\sigma(x) = Ce_1 + e_2$ and z is the integral term that will be determined later. Then, if we derive (130) and substitute (122) and (123) in the result, we have

$$\dot{s} = Ce_2 + \ddot{x}_d - f(x_1, x_2) - B(x_1, x_2)u_0 - B(x_1, x_2)u_1 - h(x_1, x_2, t) + \dot{z}, \tag{131}$$

and defining

$$\dot{z} = Ce_2 + f(x_1, x_2) + B(x_1, x_2)u_0 - \ddot{x}_d, \quad z(0) = -Ce_1(0) - e_2(0), \tag{132}$$

Eq. (131) reduces to

$$\dot{s} = -B(x_1, x_2)u_1 - h(x_1, x_2, t). \tag{133}$$

Now it is necessary to propose the control law u_1 , which will be designed using the *super-twisting* algorithm

$$\begin{aligned} u_1 &= \alpha_1 |s|^{1/2} \text{sign}(s) + v \\ \dot{v} &= \alpha_2 \text{sign}(s). \end{aligned} \tag{134}$$

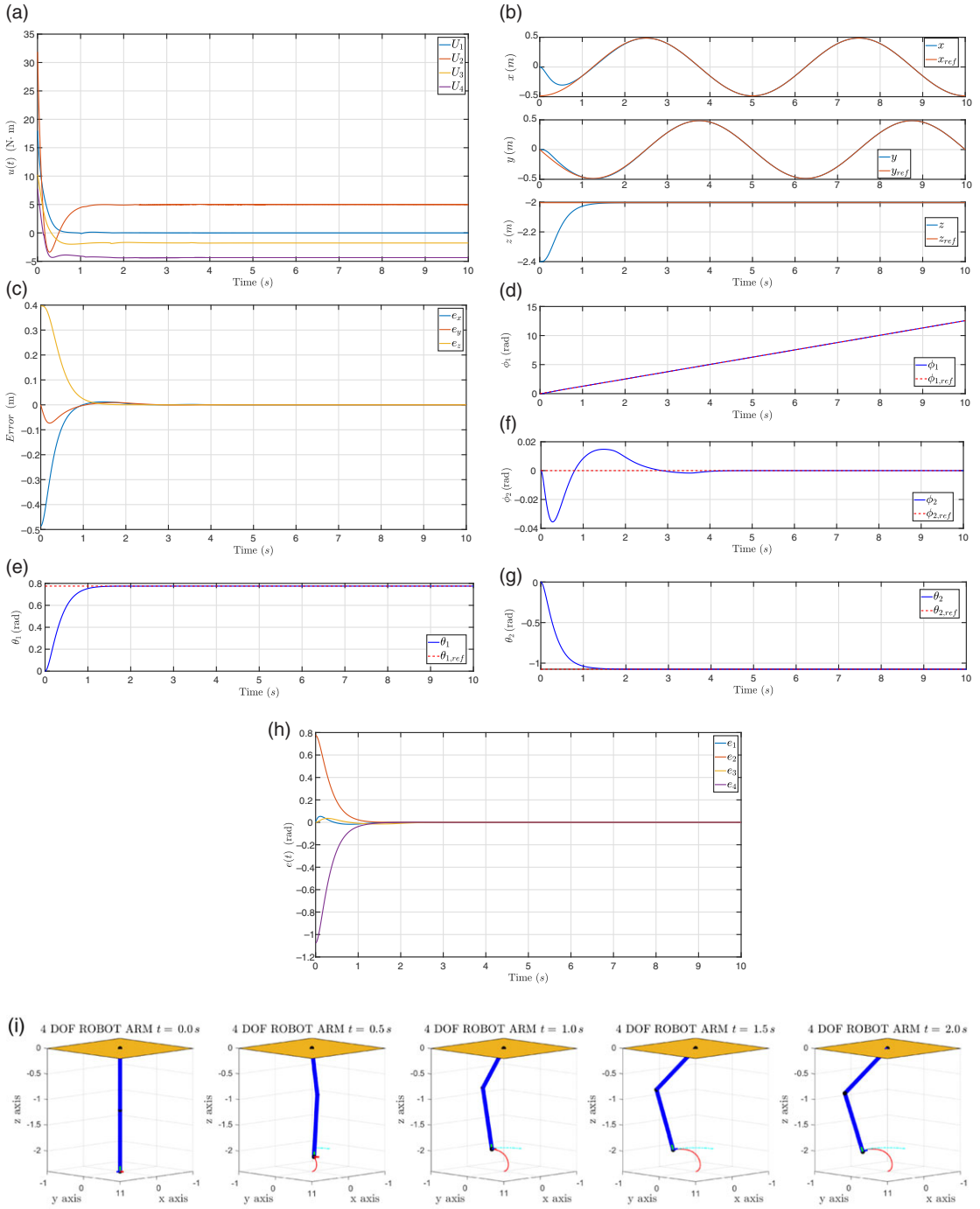


Figure 9. Sliding mode-based control: (a) control signals U_i . (b) Space (x,y,z) tracking. (c) Space x, y, z errors. (d–g) States angles $\phi_1, \phi_2, \theta_1, \theta_2$. (h) Errors of state angle $\phi_1, \phi_2, \theta_1, \theta_2$. (i) Sequence of tracking.

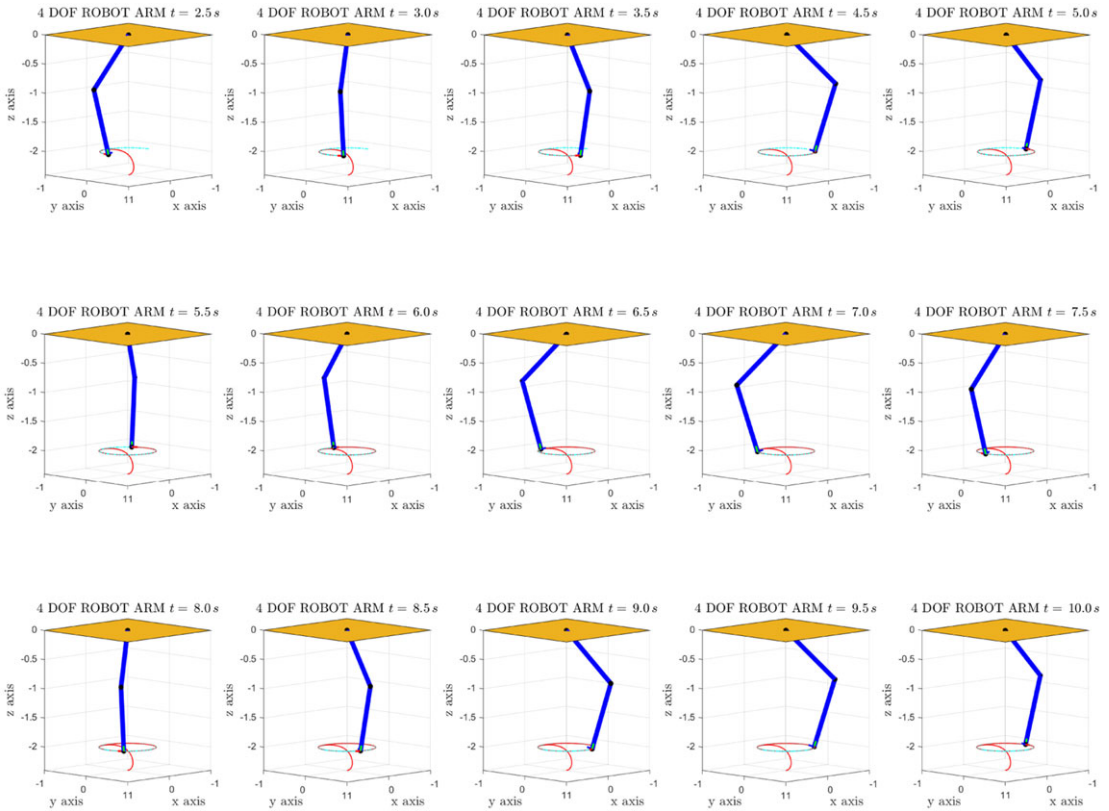


Figure 9. Continued.

Substituting previous Eq. (134) in (133), the dynamics of the sliding surface is described as

$$\dot{s} = -B(x_1, x_2)(\alpha_1 |s|^{1/2} \text{sign}(s) + v) + h(x_1, x_2, t), \tag{135}$$

$$\dot{v} = \alpha_2 \text{sign}(s). \tag{136}$$

It is possible to prove that the dynamics of the sliding surface is globally asymptotically stable [30] with certain conditions of gains α_1 and α_2 . With the explained control above, it will be achieved that the control u_1 rejects the disturbance $h(x_1, x_2, t)$, that is, $B(x_1, x_2)u_1 = -h(x_1, x_2, t)$ and even more, achieve the error system has the form

$$\begin{aligned} \dot{e}_1 &= e_2 \\ \dot{e}_2 &= \ddot{x}_d - f(x_1, x_2) - B(x_1, x_2)u_0, \end{aligned} \tag{137}$$

which is equivalent to the nominal system with control signal u_0 . It should be noted that u_0 was designed such that the nominal system tracks the desired references.

In order to show the performance of the proposed controller, two simulation experiments were developed. The parameters for the two-link manipulator are the same as described in the previous section. The values of the parameters of the controller used are as follows: $\alpha_1 = \text{diag}(3, 3, 1.5, 1.8)$, $\alpha_2 = \text{diag}(0.3, 0.3, 0.3, 0.3)$, $C = \text{diag}1, 1, 1, 1$, $K_p = \text{diag}(25, 25, 25, 25)$, $K_d = \text{diag}(10, 10, 10, 10)$.

For the first simulation experiment, the reference is described as $x_d = \left[\frac{\pi}{12}, \frac{5\pi}{36}, \frac{\pi}{6}, \frac{\pi}{9} \right]^T$, which corresponds to fixed point in the space.

Figure 8 shows the double pendulum reaching a 3D point. It shows the control signals, the states, and angles evolution as their errors. Compared with the Bang–Bang control of the previous subsection, the integral sliding modes with super-twisting diminish chattering and reject matched perturbations.

Figure 9 shows the results for tracking a nonlinear trajectory. It shows the control signals, the states, and angles evolution as their errors. The integral sliding modes with super-twisting diminish chattering and reject matched perturbations.

8. Conclusion

In this work, the authors show the importance of the Hamiltonian in control theory. Instead of using the Lagrangian formulation of electromechanical or robotic systems, our work is focused on robot dynamics by its Hamiltonian. We compute the Hamiltonians, dynamics, and applied controllers using screw theory in the motor algebra. By using the recursive Newton–Euler dynamics, we can compute the Hamiltonians for each joint. Thanks to the screw theory, it is possible to compute recursively the Hamiltonian and the dynamics for each joint for robot manipulators of n DOF. As a result, we can apply local controllers to each joint.

In the experimental analysis, using simulations, we compute via the Newton–Euler dynamics and the Hamiltonians for a planar 2 DOF robot arm and a spherical pendulum of 4 DOF. We apply and compare four controllers: Bang–Bang., PD, basic sliding modes, and integral sliding modes together with super-twisting. We show that the sliding-mode-based controllers diminish chattering and reject matched perturbations. In future work, we will continue to relate geometric algebra and nonlinear control. We will develop observers and new nonlinear controllers using the motor algebra framework and conformal geometric algebra.

Compliance with ethical standards. This study was funded by Mexican Conacyt Grant A1-S-1042.

Conflict of interest. The authors declare that they have no conflict of interest.

Authors participation. All authors have contributed in an equal manner to the elaboration of the article.

References

- [1] C. Aguilar-Ibáñez, J. Moreno-Valenzuela, O. García - Alarcón, M. Martínez-Lopez, J. A. Acosta and M. Suarez-Castanon., “PI-type controllers and Σ - Δ modulation for saturated DC-DC buck power converters,” *IEEE Access* **9**, 20346–20357 (2021).
- [2] J. Aguilar-Ibáñez, E. Orozco, D. Cordova, M. Islas, J. Pacheco, G. Gutierrez, A. Zacarias, L. Soriano, J. Meda-Campaña, D. Mujica-Vargas, “Modified linear technique for the controllability and observability of robotic arms,” *IEEE Access* **10**, 3366–3377 (2022).
- [3] R. Ortega, A. J. van der Schaft, B. Maschke and G. Escobar, “Energy-shaping of port-controlled Hamiltonian systems by interconnection,” *In: Proc. IEEE Conf. on Decision and Control*, vol. **2** (1999) pp. 1646–1651.
- [4] V. Záda and K. Belda, “Robot Control in Terms of Hamiltonian Mechanics,” *In: Proc. of the 22nd International Conference Engineering Mechanics*, Svratka, Czech Republic, 9–12 May (2016) pp. 627–630
- [5] Z. Wanxie, W. Zhigan and T. Shujun. *Theory and Computation of State Space Control* (Science Publishing House, Beijing, 2006).
- [6] R. Ortega, A. van der Schaft, F. Castanos and A. Astolfi, “Control by interconnection and standard passivity-based control of port-Hamiltonian systems,” *IEEE Trans. Autom. Cont.* **53**(11), 2527–2542 (2008).
- [7] D. Hestenes, “Hamiltonian mechanics with geometric calculus,” *In: Spinors, Twistors, Clifford Algebras, and Quantum Deformations* (Z. Oziewicz et al., eds.) (Springer Science+Business Media, B.V., 1993) pp. 203–214.
- [8] E. T. Abou El Dahab, “A formulation of Hamiltonian mechanics using geometric algebra,” *Adv. Appl. Clifford Al.* **10**(2), 217–223 (1972).
- [9] R. Pappas, “A Formulation of Hamiltonian Mechanics Using Geometric Calculus,” *In: Clifford Algebras and Their Applications in Mathematical Physics* (F. Brackx et al., eds.) (Kluwer Academic Publishers, 1993) pp. 251–258.
- [10] D. Hestenes. *New Foundation of Classical Mechanics* (D. Reidel Publishing Co., Dordrecht/Boston, 1986).
- [11] E. Bayro-Corrochano and G. Osuna-González, “Modeling, control and tracking in robotics using screw theory in geometric algebra,” *J. Robot. (to appear)* (2022).

- [12] J. G. Romero, R. Ortega and I. Sarras, “A globally exponentially stable tracking controller for mechanical systems using position feedback,” *IEEE Trans. Automat. Contr.* **60**(3), 818–823 (2015).
- [13] A. Yaghmeaei and M. J. Yazdanpanah, “Trajectory tracking for a class of contractive port hamiltonian systems,” *Automatica* **83**(September), 331–336 (2017).
- [14] R. Reyes-Baez, A. J. Van der Schaft and B. Jayawardhana, “Tracking Control of Fully Actuated Port-Hamiltonian Mechanical Systems via Sliding Manifolds and Contraction Analysis,” **In: IFAC Papers On-Line** (2017) pp. 50–51.
- [15] R. Kelly, V. Santibáñez and A. Loria, *Control of Robot Manipulators in Joint Space* Advanced Textbooks in Control and Signal Processing (Springer, 2005).
- [16] R. Ortega, A. Loria, P. J. Nicklasson and H. Sira-Ramirez, “Passivity based Control of Euler Lagrange Systems: Mechanical, Electrical and Electromechanical Applications,” **In: Communication and Control Systems** (Springer, 1998).
- [17] K. Fujimoto, K. Sakurama and T. Sugie, “Trajectory tracking control of port-controlled hamiltonian systems via generalized canonical transformations,” *Automatica* **39**(12), 2059–2069 (2003).
- [18] R. Ortega, A. Van Der Schaft, B. Maschke and G. Escobar, “Interconnection and damping assignment passivity-based control of port-controlled hamiltonian systems,” *Automatica* **38**(4), 585–596 (2002).
- [19] K. Fujimoto and T. Sugie, “Time-varying stabilization of hamiltonian systems via generalized canonical transformations,” *IFAC Proc. Vol.* **33**(2), 63-68 (2000). D. Hestenes, *New Foundation of Classical Mechanics* (D. Reidel Publishing Co., Dordrecht/Boston, 1986)
- [20] J. I. Mulero-Martinez, “Canonical transformations used to derive robot control laws from a port-controlled Hamiltonian system perspective,” *Automatica* **44**(9), 2435–2440 (2008).
- [21] D. Dirksz and J. M. A. Scherpen, “Structure preserving adaptive control of port-hamiltonian systems,” *IEEE Trans. Automat. Contr.* **57**(11), 2880–2885 (2021).
- [22] A. Donaire, T. Perez and N. Bartlett, “Tracking control of a class of hamiltonian mechanical systems with disturbances,” **In: Proceedings of Australasian Conference on Robotics and Automation. Australian Robotics and Automation Association ARAA** (2014) pp. 1–7.
- [23] A. Donaire and S. Junco, “On the addition of integral action to port-controlled hamiltonian systems,” *Automatica* **45**(8), 1910–1916 (2009).
- [24] E. Bayro-Corrochano, *Geometric Algebra Applications. vol. I*, Computer Vision, Graphics and Neurocomputing (Springer Verlag, Heidelberg, 2018).
- [25] W. K. Clifford, “Preliminary sketch of bi-quaternions,” *Proc. London Math. Soc.* **4**, 381–395 (1873).
- [26] J. Craig, *Introduction to Robotics 2nd edition (Mechanics and Control*, Pearson, 1989).
- [27] J. Selig, *Introductory Robotics* (Prentice-Hall International, Hertfordshire, UK, 1992).
- [28] V. Utkin, J. Guldner and J. Shi. *Sliding Mode Control in Electro-Mechanical Systems*, 2nd edition. Automation and Control Engineering (Taylor & Francis, London, UK, 2009).
- [29] M. W. Spong, S. Hutchinson and M. Vidyasagar. *Robot Modeling and Control* (John Wiley & Sons, 2006).
- [30] J. A. Moreno and M. Osorio, “A Lyapunov approach to second-order sliding mode controllers and observers,” **In: 47th IEEE Conference on Decision and Control** (December 2008) pp. 2856–2861.