A Review of Passivity-Based Task Space Control of Hybrid Rigid-Soft (HyRiSo) Robots with Parametric Uncertainty

Harsh Senjaliya
UID: 120215575
Universiy of Maryland

Dhairya Shah
UID: 120235146
University of Maryland

***Abstract*—This paper review presents the introduction of a novel robotic system known as the hybrid rigid-soft (HyRiSo) robot. This robot showcases a distinctive configuration consisting of interconnected stiff and soft links in a serial manner. The primary objective of this innovative design approach is to use the beneficial attributes of a soft manipulator's dexterity while incorporating the inherent structural support capabilities of a conventional stiff arm. The incorporation of revolute joints with diverse modes of actuation, along with the requirement for flexible bending in the soft links, poses an intricate design obstacle when developing an integrated controller for robots belonging to this category. The main objective of this study is to tackle the above-described difficulty by showcasing the effectiveness of well-recognized passivity-based adaptive and resilient controllers. The utilization of these controllers is strategically implemented to achieve efficient tracking of task space, even when confronted with uncertainties pertaining to mass, stiffness, damping, and actuation. The effectiveness of the suggested control procedures is demonstrated in this study through experimental scenarios involving task space tracking in the face of complex impediments and within predefined joint limitations, utilizing a HyRiSo robot. In order to offer a thorough comprehension of the proposed methodology, the paper showcases numerical illustrations employing a particular arrangement of the HyRiSo robot, specifically a 2-rigid-2-soft system. The performance of the suggested task space controllers is thoroughly examined in these examples, providing valuable insights into the robustness and adaptability of the control frameworks in effectively handling the inherent complexities associated with the various modes of actuation and flexible structure of the HyRiSo robot. In addition, this article makes a valuable contribution to the wider domain of robotic systems by demonstrating the effectiveness of passivity-based adaptive and resilient controllers in tackling the difficulties associated with incorporating both soft and rigid components in a serial design. The findings provided in this study not only demonstrate theoretical breakthroughs but also underscore the practical applicability of the proposed HyRiSo robot in real-world situations. In its entirety, this study makes a significant scholarly contribution to the expanding realm of robotics, providing valuable perspectives on the intricacies of designing and controlling hybrid robotic systems.**

***Index Terms*—Soft Manipulator, Robust control, Adaptive control, Bending, Manipulators, Task Space Analysis**

I. Introduction

The field of soft robotics has witnessed substantial growth in recent years, driven by its inspiration from natural systems [1], [2]. The rapid growth in this phenomenon can be attributed to the potential for enhanced manipulation capabilities in intricate and unorganized scenarios, hence overcoming the limitations of traditional rigid robotic systems. The challenges associated with achieving precise motion control in complex multi-link soft robots operating within expansive workspaces have become evident. The challenge is further intensified by the well-documented phenomenon in which the warping of workspace is induced by longer soft link lengths, resulting in an escalation in control complexity [3].

The emergence of integrated hybrid rigid-soft (HyRiSo) robots can be attributed to the limitations of existing rigid robots, which are unable to perform dexterous task space operations in crowded environments [4]. These robotic systems integrate the accuracy and load-carrying capability of a rigid robot with the dexterity and delicacy of a soft manipulator in a cohesive manner.

HyRiSo robots exhibit a distinctive characteristic by employing a diverse range of actuation modes for its soft links and revolute joints. In the context of this integrated control system, it is imperative to address the unexpected actuator mapping that involves the conversion of pressure/current variables into actual torques. There are two notable advantages associated with the adoption of a model-based integrated dynamic controller. Firstly, it facilitates rapid autonomous operation. Secondly, it fosters a comprehensive understanding of the behavior of the HyRiSo robot, hence ensuring stability and optimal performance.

The design requirement involves the development of controllers that can effectively handle uncertainties in various aspects, such as actuator mapping, stiffness, damping terms of soft connections, and dynamic characteristics. These factors are widely acknowledged as potential sources of uncertainty.

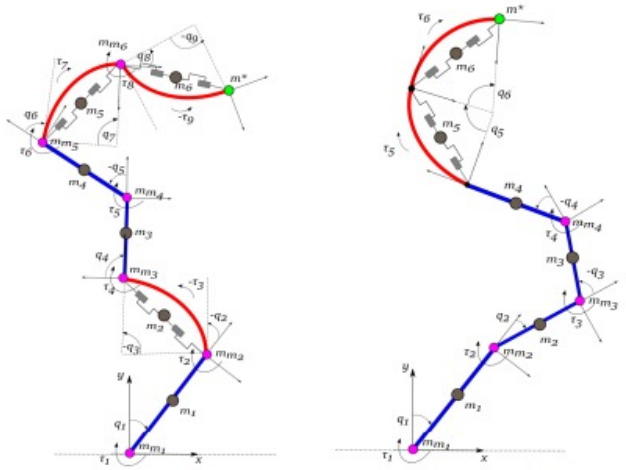
The present study investigates a comprehensive control system developed specifically for HyRiSo robots, which builds upon prior research that demonstrated satisfactory although sluggish performance due to the independent handling of stiff and soft links [5]. In contrast to existing inflexible approaches to robot control that only address parametric uncertainty, the proposed controllers aim to address the intricate challenges posed by model and actuation uncertainties that are unique to the HyRiSo class.

Previous studies have put forth adaptive control laws in the context of stiff robotic control [6], [7], [8], [9], [10].

Additionally, robust controllers that exhibit resistance to uncertainty and unmodeled disturbances have also been proposed [10], [11]. In a similar vein, the domain of soft robotics has exhibited notable advancements in addressing the challenge of controlling uncertainty [12], [13], [14], [15], [16], [17]. However, an area that remains largely unexplored in the field is the advancement of a comprehensive control framework for HyRiSo robots that effectively handles both actuation and model uncertainty concurrently.

The primary findings of this research encompass the development of resilient and flexible controllers grounded on the concept of passivity, capable of accommodating errors in both actuator mapping and system dynamics. Additionally, the study introduces the HyRiSo robot, which enables skillful manipulation tasks. In addition, the practical utility of the HyRiSo robot is underscored by its demonstrated effectiveness in managing task domains, leveraging system redundancy to navigate complex barriers, overcome joint restrictions, and handle unmodeled disturbances.

The present study builds upon the foundational work presented in reference [5] and contributes to the current discourse surrounding integrated control frameworks for advanced robotic systems within the broader context of existing research. This article presents the HyRiSo robot, a new robotic platform that shows promise in addressing the need for flexible and adaptable solutions in complex real-world scenarios.



(a) RSRRSS HyRiSo robot (b) RRRRSS HyRiSo robot

Fig. 1. This illustration showcases two instances of HyRiSo robots, which are comprised of rigid (R) linkages shown in blue and CC soft (S) links represented in red. These robots have been modeled using the enhanced formulation as described in reference [18]. The masses of the links, denoted as m_i (brown), the masses of the motors, denoted as m_{mi} (pink), and the mass of the end effector, denoted as m^* (green), are also depicted in the illustration. The RSRRSS design is based on a HyRiSo robot that consists of three R links and three S links, all of which have actuated revolute joints. The design denoted as RRRRSS (b) pertains to a HyRiSo robot using four R links and two S links, wherein the R-S and S-S links are connected by fixed joints.

II. The HyRiSo robotic system

A. Robot Dynamics

Let us examine a robotic system with n_r rigid links (R) and n_s soft links (S) that are interconnected serially, resulting in an open chain consisting of $n_s + n_r$ links. In the broader context, it is commonly accepted that all joints, including R-R, R-S, and S-S, are classified as revolute joints. The assumption is made that the soft linkages possess a non-extensible property and maintain a constant curvature, as stated in reference [19].

In a broad sense, the hybrid robotic system will possess a total of $\alpha = n_r + 2n_s$ configuration variables. These variables encompass the revolute joint angles, which account for all $n_r + n_s$ links, as well as the degree of curvature shown by the n_s soft links. The variables in question are represented by the symbols q_i, where i takes on values from 1 to α. These variables are organized into a vector q, which can be expressed as $q = [q_1, q_2, ..., q_\alpha]^T$ and belongs to the set of real numbers $\mathbb{R}^\alpha$.

Some HyRiSo robot designs may use permanent joints between the R-S or S-S links. 1 displays two exemplar HyRiSo robots. The assumption is made that the rigid connections are homogeneous in nature, and the masses m_i are concentrated at the centroid of each link, where i is a member of the set of indices representing the rigid links. The masses of the motors at the revolute joints, denoted as m_{mi} where i is a member of the set of revolute joint indices, are considered to be concentrated at the revolute joints. The mass of the end effector, denoted as m^*, is assumed to be concentrated at the tip of the robot. Furthermore, given the assumption that the soft linkages are Creative Commons (CC) licensed, an enhanced formulation is employed. The increased formulation is described as follows: The resolution of the infinite dimensionality of the soft robot configuration in the context of the Piecewise Constant Curvature (PCC) model involves conceptualizing the robot's shape as a composition of a finite number of segments with constant curvature (CC). These segments are merged together in a manner that ensures the resulting curve is differentiable at all points.

Now, let's consider a Pneumatically Controlled Compliant (PCC) soft robot comprising n Continuum-Continuum (CC) segments. Each segment has an associated reference frame, denoted as $S_1, ..., S_n$, and there is an additional base frame S_0.

The kinematics of the robot are characterized by n homogeneous transformations, $T_0^1, ..., T_{n-1}^n$, mapping each reference system to the succeeding one. This analysis focuses on the planar scenario for brevity, with comprehensive details on the 3D case available in [19]. Figure 3 depicts the kinematics of a single CC segment. Under the non-extensibility hypothesis, the configuration of each segment can be adequately described by a single variable q_i, where q is a vector in $\mathbb{R}^n$ collecting all q_i values. This variable relates to the radius of curvature ρ_i through the linear equation $q_i \rho_i = L_i$, where L_i represents the fixed length of the segment, and q_i is termed the degree of curvature.

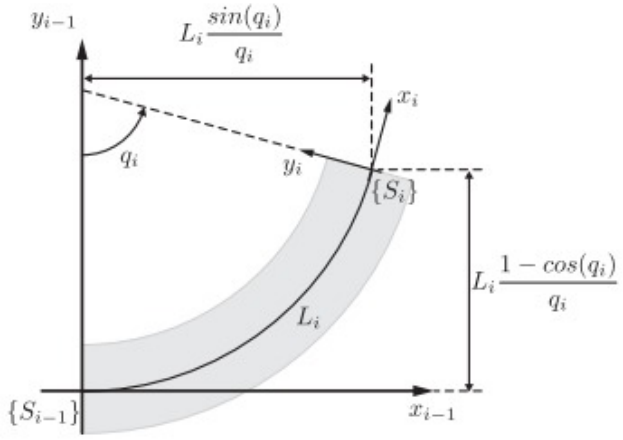


Fig. 2. The kinematic representation of the i-th planar constant curvature section. Two local frames are positioned at the two endpoints of the segment, denoted as $\{S_{i-1}\}$ and $\{S_i\}$ accordingly. The magnitude of the segment is denoted as L_i, whereas q_i represents the curvature degree.

The i-th homogeneous transformation can be derived using geometrical considerations as

$$T_i^{i-1}(q_i) = \begin{bmatrix} \cos(q_i) & -\sin(q_i) & L_i \\ \sin(q_i) & \cos(q_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

Equation(1) can be reformulated using Denavit-Hartenberg transformations, implicitly establishing equivalence between the soft and rigid robots.

From a kinematic perspective, any representation ensuring the alignment of CC segment endpoints with corresponding reference points of the rigid robot is considered equivalent. However, when addressing dynamics, an additional constraint arises: the inertial properties of the augmented and soft robots must match. This is achieved by aligning the centers of mass using an equivalent point mass attached to the rigid robot structure. The configurations are linked through a continuously differentiable map $m : \mathbb{R}^n \to \mathbb{R}^{nh}$:

$$\xi = m(q)$$

This map ensures that the nonlinear constraint $\xi = m(q)$ guarantees alignment of CC segment endpoints and point masses with the rigid robot.

For instance, considering the midpoint of the chord as a mass distribution approximation, the DH parametrization in Table 1 is provided. The corresponding map is:

$$m_i(q_i) = \begin{bmatrix} q_i/2 \\ Li\sin(q_i/2) \\ q_i \\ Li\sin(q_i/2) \\ q_i \end{bmatrix}$$

Figure3 illustrates a CC segment and its rigid counterpart for various postures. A continuous soft robot represented by the PCC model can be effectively matched with a dynamically consistent rigid robot constructed from these elements. Figure 3 illustrates an example with four distinct segments, connected by the map $m(q) = m_1(q_1)^T ... m_n(q_n)^T$.

Table 1. Description of the stiff robot equivalent to a single CC segment. Parameters θ, d, a, α correspond to the classical DH parametrization, while μ represents mass.

Link	θ	d	a	α	μ
1	$q_i/2$	0	0	$\pi/2$	0
2	0	$Li\sin(q_i/2)$	q_i	0	0
3	0	$Li\sin(q_i/2)$	q_i	$-\pi/2$	0
4	$q_i/2$	0	0	0	0

The dynamics of the augmented rigid robot are governed by the equation:

$$B\xi(\xi)\ddot{\xi} + C\xi(\xi,\dot{\xi})\dot{\xi} + G\xi(\xi) = J_T\xi(\xi) f_{\text{ext}}$$

Here, $\xi, \dot{\xi}, \ddot{\xi} \in \mathbb{R}^{nh}$ represent the robot configuration and its derivatives, $B\xi(\xi) \in \mathbb{R}^{nh\times nh}$ is the inertia matrix, $C\xi(\xi,\dot{\xi})\dot{\xi} \in \mathbb{R}^{nh}$ includes Coriolis and centrifugal terms, and $G\xi(\xi) \in \mathbb{R}^{nh}$ accounts for gravity effects.

The robot is subject to external wrenches f_{ext}, transmitted through the Jacobian J_ξ. The dynamics are described on the sub-manifold defined by the map $\xi = m(q)$, encompassing scleronomic and holonomic nonlinear constraints.

To integrate these constraints into the system's dynamics, we evaluate the augmented configuration derivatives $\xi, \dot{\xi}, \ddot{\xi}$ with respect to $q, \dot{q}, \ddot{q}$:

$$\begin{cases} \xi = m(q) \\ \dot{\xi} = Jm(q)\dot{q} \\ \ddot{\xi} = J\dot{m}(q,\dot{q})\dot{q} + Jm(q)\ddot{q} \end{cases}$$

Here, $Jm(q) : \mathbb{R}^n \to \mathbb{R}^{nh\times n}$ is the Jacobian of $m(\cdot)$. For $m_i(q_i)$ defined as in (3), $Jm_i = \frac{1}{2}Lc_iLc_i^T$, where $Lc_i = Li\frac{\cos(q_i/2)-2\sin(q_i/2)}{2q_i^2}$. Substituting (6) into (5) results in:

$$\begin{aligned} B\xi(m(q))(&J\dot{m}(q,\dot{q})\dot{q} \\ &+ Jm(q)\ddot{q}) + C\xi(m(q), Jm(q)\dot{q})Jm(q)\dot{q} \\ &+ G\xi(m(q)) = J_T\xi(m(q)) f_{\text{ext}} \end{aligned} \tag{2}$$

The soft links masses m_i, with $i \in \{\text{soft link indices}\}$ are at the center of the main chord connecting the ends of the soft link. The dynamics of HyRiSo in the form is then written as,

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + D\dot{q} + Kq + G(q) = \tau \tag{3}$$

B. Skew Symmetric Property

The Skew Symmetry property refers to an important relationship between the inertia matrix $D(q)$ and the matrix $C(q,\dot{q})$ that will be of fundamental importance for the problem of manipulator control considered in later chapters.

Let $D(q)$ be the inertia matrix for an n-link robot, and define $C(q,\dot{q})$ in terms of the elements of $D(q)$ according to Equation [20]. Then the matrix $N(q,\dot{q}) = \dot{D}(q) - 2C(q,\dot{q})$ is skew-symmetric, i.e., the components n_{jk} of N satisfy $n_{jk} = -n_{kj}$.

Proof:

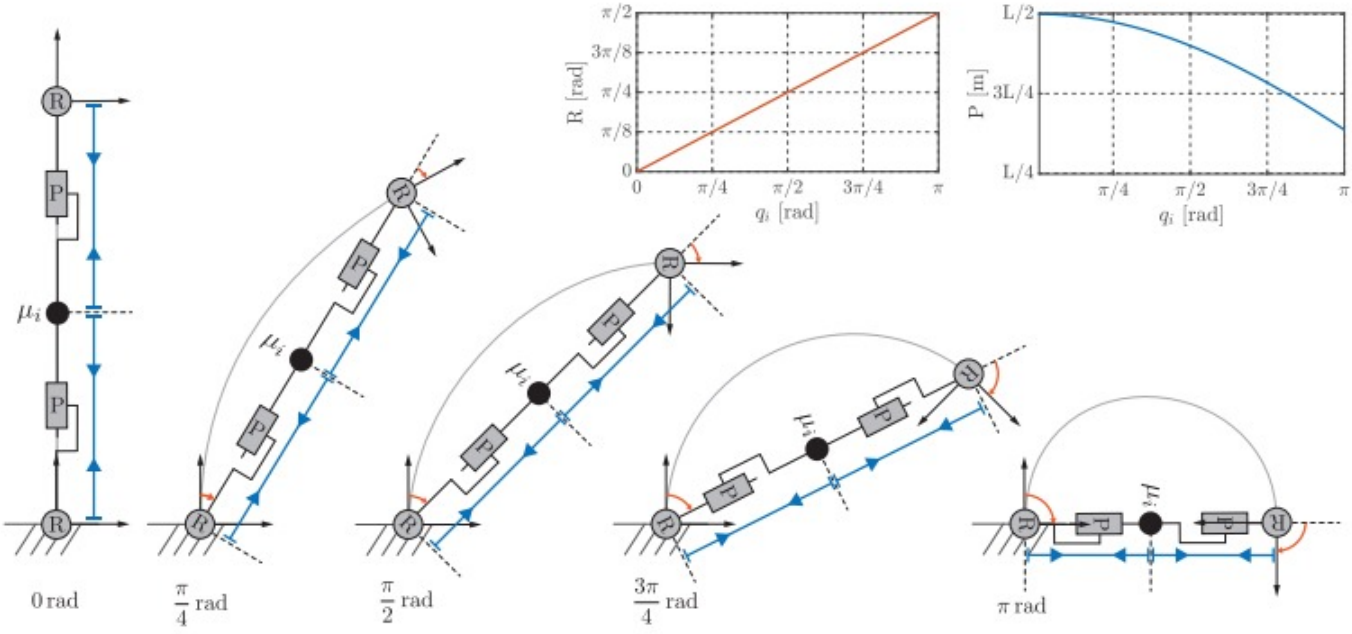


Fig. 3. This image presents instances of a dynamically consistent augmented robot (RPPR) successfully matching a singular section. We provide a series of five distinct configurations that correlate to varying degrees of curvature, ranging from $q_i = 0$ rad (representing a straight configuration) to $q_i = \pi$ rad (representing a half circle).The map $\xi_i = m_i(q_i)$ in equation (3) enforces restrictions that ensure the accurate placement of the equivalent mass at the midpoint of the chord, as well as the alignment of the reference frames associated with the tip of the segment and the end of the rigid robot. The charts depict the variations in the angle of the revolute joints R and the length of the prismatic joints P when q_i undergoes modifications.

Given the inertia matrix $D(q)$, the kj-th component of $\dot{D}(q)$ is given by the chain rule as

$$\dot{d}_{kj} = \sum_{i=1}^{n} \frac{\partial d_{kj}}{\partial q_i} \dot{q}_i$$

Therefore, the kj-th component of $N = \dot{D} - 2C$ is given by

$$n_{kj} = \dot{d}_{kj} - 2c_{kj}$$

$$= \sum_{i=1}^{n} \frac{\partial d_{kj}}{\partial q_i} - \frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k} \dot{q}_i$$

$$= \sum_{i=1}^{n} \frac{\partial d_{ij}}{\partial q_k} - \frac{\partial d_{ki}}{\partial q_j} \dot{q}_i$$

Since the inertia matrix $D(q)$ is symmetric, i.e., $d_{ij} = d_{ji}$, it follows from [20] by interchanging the indices k and j that

$$n_{jk} = -n_{kj}$$

which completes the proof.

It is important to note that, in order for $N = \dot{D} - 2C$ to be skew-symmetric, one must define C according to Equation [20].

C. Linearly Parameterizable Property

The robot equations of motion are defined in terms of certain parameters, such as link masses, moments of inertia, etc., that must be determined for each particular robot in order, for example, to simulate the equations or to tune controllers. The complexity of the dynamic equations makes the determination of these parameters a difficult task. Fortunately, the equations of motion are linear in these inertia parameters in the following sense.

There exists an $n \times n$ function, $Y(q, \dot{q}, \ddot{q})$, which we assume is completely known, and an l-dimensional vector Θ such that the Euler-Lagrange equations can be written as:

$$D(q) + C(q, \dot{q})\dot{q} + g(q) = Y(q, \dot{q}, \ddot{q})\Theta = \tau$$

The function, $Y(q, \dot{q}, \ddot{q})$, is called the Regressor, and Θ is the Parameter vector. The dimension of the parameter space, $\mathbb{R}^l$, i.e., the number of parameters needed to write the dynamics in this way, is not unique. In general, a given rigid body is described by ten parameters, namely, the total mass, the six independent entries of the inertia tensor, and the three coordinates of the center of mass. An n-link robot then has a maximum of $10n$ dynamics parameters. However, since the link motions are constrained and coupled by the joint interconnections, there are actually fewer than $10n$ independent parameters. Finding a minimal set of parameters that can parametrize the dynamic equations is, however, difficult in general.

So using these properties, we can say that in this work, we note a simple way to incorporate the actuator parameters as suggested in [21], [22]. Using the relationship of the actuator mapping for the torque in Equation(3) and defining $M_0 = A^{-1}M$, $C_0 = A^{-1}C$, $D_0 = A^{-1}D$, $K_0 = A^{-1}K$, and $G_0 = A^{-1}G$, we can rewrite the dynamics [2]:

$$M_0(q)\ddot{q} + C_0(q,\dot{q})\dot{q} + D_0\dot{q} + K_0 q + G_0(q) = p. \quad (4)$$

We note that, since M, C, and G are linear in the dynamic parameters, so are M_0, C_0, and G_0 as premultiplication by A^{-1} is a linear operation. Therefore, M_0, C_0, and G_0 are linear in suitably selected parameters which are nonlinearly related to the dynamic parameters and the actuator mapping parameters. Therefore, we can extend the linear in parameters property of the dynamics to the system [2], such that for any differentiable vector $\gamma \in \mathbb{R}^{\alpha}$, there exists a regressor $Y_2(q,\dot{q},\gamma,\dot{\gamma}) \in \mathbb{R}^{\alpha\times\beta_2}$ and a constant parameter vector $\Theta_2 \in \mathbb{R}^{\beta_2}$ such that:

$$M_0(q)\dot{\gamma}+C_0(q,\dot{q})\gamma+D_0(q)\gamma+K_0 q+G_0(q) = Y_2(q,\dot{q},\gamma,\dot{\gamma})\Theta_2$$

III. Passivity-Based Control for Hybrid Robots

The primary objective of the control approach is to minimize the tracking error, defined as $e(t) = X(t) - X_d(t)$, where $X_d(t)$ represents the desired reference task space trajectory. To achieve this, the tracking error is constrained by a control law $s(t)$ defined as:

$$s(t) = \dot{q}(t) + J^{+}(q)\Lambda e(t) - (I_\alpha - J^{+}(q)J(q))\psi s(t) \quad (5)$$

Here, Λ is a positive definite gain matrix, I_α is the identity matrix, and $\psi s(t)$ is the negative gradient of a convex function used for subtask control.

A. Passivity-Based Control Law

The control law incorporates the Jacobian pseudoinverse $J^{+}(q)$ to ensure stability. The pseudo-inverse is given by $J^{+} \equiv J^{\top}(JJ^{\top})^{-1}$ and has the property $JJ^{+} = I_2$. Once the tracking error approaches zero ($s = 0$), the derivative of the error $\dot{e}(t)$ is shown to converge to $-\Lambda e(t)$, guaranteeing error convergence to the origin.

B. Signal Definitions

The paper introduces signals $v = \dot{q} - s$ and $a = \ddot{q} - s\dot{q}$, where q represents the configuration space. Additionally, uncertainty in parameters is considered, denoted by $\hat{(\cdot)}$ for estimated values and $\tilde{(\cdot)}$ for estimation errors.

C. Regressor and Parameter Vectors

The formulation incorporates the extended linear in parameters property of Lagrangian systems, defining a regressor $Y_2(q,\dot{q},v,a)$ and parameter vector Θ_2. The dynamic terms $M\hat{0}(q)a, C\hat{0}(q,\dot{q})v, D\hat{0}v, K\hat{0}q$, and $G\hat{0}(q)$ are expressed as a linear combination using $Y_2(q,\dot{q},v,a)\Theta_2$.

D. Assumptions

The approach assumes time-invariant uncertainties in dynamic terms, stiffness, damping, and actuator parameters, ensuring that the parameter vector Θ_2 remains constant.

Now,

Consider the following:

1) $X(t) = h(q)$ is the end effector position in the task space.
2) $\dot{X}(t) = J(q)\dot{q}$ is the velocity of the end effector, where $J(q) = \frac{\partial h(q)}{\partial q}$ is the Jacobian matrix.
3) $X_d(t)$ is the desired reference task space trajectory.
4) $e(t) = X(t) - X_d(t)$ is the tracking error.
5) $s(t)$ is a variable designed to drive the tracking error to the origin.

The Passivity-Based control law is given by:

$$s(t) = \dot{q}(t) + J^{+}(q)\Lambda e(t) - \left(I - J^{+}(q)J(q)\right)\psi s(t)$$

Where:

1) $\dot{q}(t)$ represents the joint velocity.
2) $J^{+}(q)$ is the pseudoinverse of the Jacobian matrix $J(q)$.
3) Λ is a positive definite matrix used for weighting the tracking error.
4) $e(t)$ is the tracking error.
5) $(I - J^{+}(q)J(q))$ is a term that accounts for the null space of the Jacobian, where I is the identity matrix.
6) ψ is a positive constant.

Also,

1) **Kinematics:**
   - Defining the end effector position as $X(t) = h(q)$.
   - Computing the velocity of the end effector: $\dot{X}(t) = J(q)\dot{q}$.
2) **Tracking Error:**
   - Defining the tracking error as $e(t) = X(t) - X_d(t)$.
3) **Pseudoinverse Term:**
   - Expressing $J^{+}(q)$ as the pseudoinverse of the Jacobian matrix $J(q)$: $J^{+}(q) = J^{T}(JJ^{T})^{-1}$.
4) **Null Space Compensation Term:**
   - The term $(I - J^{+}(q)J(q))$ is designed to account for the null space.

E. Control Law Derivation

1. Defining the Jacobian Pseudoinverse:

$$J^{+} \equiv J^{\top}(JJ^{\top})^{-1}$$

2. Showing that $J^{+}J = I$:

$$J^{+}J = J^{\top}(JJ^{\top})^{-1}J = J^{\top}(J^{\top})^{-1}(J^{-1}J) = J^{\top}J^{\top} = I$$

3. Derivation of the actual the Control Law:

Given control law expression:

$$s(t) = \dot{q}(t) + J^{+}(q)\Lambda e(t) - (I_\alpha - J^{+}(q)J(q))\psi s(t)$$

Joint Velocity Term $\dot{q}(t)$: Represents current joint velocity.
Jacobian Pseudoinverse Term $J^{+}(q)\Lambda e(t)$:

$$J^\top (JJ^\top)^{-1}\Lambda e(t)$$

Damping Term $-J^{+}(q)J(q)\psi s(t)$:

$$\left(I_\alpha + J^\top (JJ^\top)^{-1}J(q)\right)\psi s(t)$$

Thus combining all terms, we get:

$$\begin{aligned} s(t) = \dot{q}(t) &+ J^\top (JJ^\top)^{-1}\Lambda e(t) \\ &- \left(I_\alpha + J^\top (JJ^\top)^{-1}J(q)\right)\psi s(t) \end{aligned} \tag{6}$$

3.1. Expressing $\dot{e}$ in terms of s::

$$\dot{e} = -J^{+}(q)\Lambda e(t) + (I_\alpha - J^{+}(q)J(q))\psi s(t)$$

3.2. Showing that $\dot{e}$ converges to Zero as $s \to 0$::

$$\lim_{s\to 0} \dot{e} = 0$$

The tracking error equation is thus given by:

$$\dot{e} = -J(q) + (q)\Lambda e(t) + (I_\alpha - J(q)J(q))\,\psi_s(t)$$

As $\epsilon \to 0$ (or $s \to 0$), $\epsilon\dot{e}(\epsilon)$ becomes constant. The term $(I_\alpha - J(q)J(q))\,\psi_s(t)$ goes to 0, making $\dot{e} \to -J(q) + (q)\Lambda e(t)$. This shows that as the tracking error approaches zero ($e = 0$, $s = 0$), the derivative of the error converges to zero.

Now, the control law can be given by:

$$u(e) = \dot{e}(e) + \epsilon + (q)\Lambda e - (\epsilon\dot{e}(\epsilon) - \epsilon + (q)\epsilon e(\epsilon))\,\epsilon^{-1}$$

1. Substituting $\dot{e} = -J(q) + (q)\Lambda e$ into the control law:

$$\begin{aligned} u(e) = &-J(q) + (q)\Lambda e + \epsilon + (q)\Lambda e \\ &- (\epsilon(-J(q) + (q)\Lambda e) - \epsilon + (q)\epsilon e)\,\epsilon^{-1} \end{aligned} \tag{7}$$

2. On further simplification, We get :

$$u(e) = -J(q) + (q)\Lambda e + \epsilon + (q)\Lambda e + J(q) - (q)\Lambda e - \epsilon + (q)e$$

$$u(e) = J(q) + \epsilon + (q)e$$

Therefore, the control law $u(e) = J(q) + \epsilon + (q)e$ ensures stable and effective control, driving the system towards the desired trajectory with minimized tracking error as $\epsilon \to 0$.

Thus let us define signals, $v = \dot{q} - s$ and $a = \ddot{q} - s\dot{q}$.

By utilizing the property of extended linearity in parameters within the context of Lagrangian systems, we are able to establish the regressor for the given system (2). $(Y_2(q, \dot{q}, v, a\dot{q}))$ and parameter (Θ_2) vector pair for the estimated systems,

$$Y_2(q, \dot{q}, v, a\dot{q})\Theta\hat{2} = M\hat{0}(q)a + C\hat{0}(q, \dot{q})v + D\hat{0}v + K\hat{0}q + G\hat{0}(q) \tag{8}$$

In this case, the actuator parameters, stiffness, damping, and dynamic terms are assumed to be time-invariantly unpredictable. Thus, the parameter vector Θ_2 is a constant.

IV. Passivity-Based Adaptive Control

This section examines the convergence qualities of an adaptive force/position control rule for mechanical manipulators. The control rule won't require prior knowledge of the beginning parameters; it will simply require measurement of $q, \dot{q}$, and F.

In accordance with the adaptive control methodology, the HyRiSo robot's control input is described as,

$$p = Y^2(q, \dot{q}, v, \dot{a})\hat{\Theta}^2 - Ks \tag{9}$$

where Ks is a positive definite diagonal gain matrix.

Now deriving the dynamics of the closed loop system by using the above control

Starting with the equation:

$$M_0(q)\ddot{q} + C_0(q, \dot{q})\dot{q} + D_0\dot{q} + K_0 q + G_0(q) = p$$

and substituting $p = Y_2(q, q, v, \dot{a})\Theta^2 - K_s$:

$$M_0(q)\ddot{q} + C_0(q, \dot{q})\dot{q} + D_0\dot{q} + K_0 q + G_0(q) = Y_2(q, q, v, \dot{a})\Theta^2 - K_s$$

Now, use the given expression for $Y_2(q, q, v, \dot{a})\Theta^2$:

$$Y_2(q, q, v, \dot{a})\Theta^2 = M_0^\wedge(q)a + C_0^\wedge(q, \dot{q})v + D_0^\wedge\dot{a} + K_0^\wedge q + G_0^\wedge(q)$$

Substituting this back into the equation:

$$\begin{aligned} &M_0(q)\ddot{q} + C_0(q, \dot{q})\dot{q} + D_0\dot{q} + K_0 q + G_0(q) \\ &= (M_0^\wedge(q)a + C_0^\wedge(q, \dot{q})v + D_0^\wedge\dot{a} + K_0^\wedge q + G_0^\wedge(q)) - K_s \end{aligned} \tag{10}$$

$$\begin{aligned} &M_0(q)\ddot{q} + C_0(q, \dot{q})\dot{q} + D_0\dot{q} + K_0 q + G_0(q) \\ &= M_0^\wedge(q)a + C_0^\wedge(q, \dot{q})v + D_0^\wedge\dot{a} + K_0^\wedge q + G_0^\wedge(q) - K_s \end{aligned} \tag{11}$$

Moving all terms involving a, v, and $\dot{a}$ to one side:

$$\begin{aligned} &M_0(q)\ddot{q} + C_0(q, \dot{q})\dot{q} + D_0\dot{q} + K_0 q + K_s \\ &\quad = M_0^\wedge(q)a + C_0^\wedge(q, \dot{q})v + D_0^\wedge\dot{a} + K_0^\wedge q + G_0^\wedge(q) \end{aligned} \tag{12}$$

Now, you can express this as a matrix equation:

$$\mathbf{M_0(q)\ddot{q} + C_0(q, \dot{q})\dot{q} + D_0\dot{q} + K_0 q + K_s = Y_2(q, q, v, \dot{a})\Theta^2} \tag{13}$$

where

$$Y_2(q, q, v, \dot{a})\Theta^2 = M_0^\wedge(q)a + C_0^\wedge(q, \dot{q})v + D_0^\wedge\dot{a} + K_0^\wedge q + G_0^\wedge(q) \tag{14}$$

and the terms on the left side correspond to the mass matrix $M_0(q)$, damping matrix $C_0(q, \dot{q})$, velocity matrix D_0, stiffness matrix K_0, and the external force K_s.

The dynamics of the closed-loop system can be found by substituting the proposed control law [7]

$$\delta = -rye^{\dot{e}}$$

resulting in:

$$\ddot{q} + \dot{q} + q = \dot{D}q + C\dot{q} + G = \dot{e}$$

Therefore,

$$\begin{aligned}
&\dot{q}_0(q)\dot{q} + \dot{q}_0(q,\dot{q})q + \dot{q}_0\dot{e} \\
&= \dot{q}_2(q,\dot{q},\ddot{q},\dddot{q})\Theta_2^{\sim} \\
&= M_0(q)\dot{s} + C_0(q,\dot{q})s + D_0 s + Ks \\
&= Y_2(q,\dot{q},v,\ddot{a})\Theta_2^{\sim}
\end{aligned} \tag{15}$$

where, we can make the declaration that
$\Theta_2^{\sim} = \hat{\Theta_2} - \Theta_2$
Thus, the adaptation law for the parameter estimation be defined as:

$$\dot{\hat{\Theta}}_2 = -\Gamma Y_2^T$$

where Γ is a positive definite symmetric gain matrix that needs to be tuned.

A. Theorem

Consider the closed-loop system (7) with the parameter adaptation law (8) and sliding surface (3). In the absence of any external wrenches, the task space position error (e) and velocity error $(\dot{e})$ asymptotically reach the origin while the parameter estimation error $(\Theta_2^{\sim})$ remains bounded.

Proof:: Consider a Lyapunov-like function for the system defined as,

$$V = \frac{1}{2}s^{\top}M_0 s + \Theta_2^{\sim\top}\Gamma^{-1}\Theta_2^{\sim},$$

where s is a vector, M_0 is a matrix, $\Theta_2^{\sim}$ is the parameter estimation error vector, and Γ^{-1} is the inverse of the matrix Γ.

Differentiating V with respect to time

$$\frac{dV}{dt} = s^T\frac{ds}{dt}M_0 s + s^T M_0\frac{ds}{dt} + \Theta^T\frac{d\Theta}{dt}\Gamma^{-1}\Theta + \Theta^T\Gamma^{-1}\frac{d\Theta}{dt}$$

$V = \frac{1}{2}s^T M_0 s + \frac{1}{2}\Theta^T\Gamma^{-1}\Theta$, the differentiation of V with respect to time t is:

$$\begin{aligned}
\frac{dV}{dt} = \frac{1}{2}\left(\frac{ds}{dt}\right)^T sM_0 s + \frac{1}{2}s^T M_0\frac{ds}{dt} \\
+ \frac{1}{2}\left(\frac{d\Theta}{dt}\right)^T\Gamma^{-1}\Theta + \frac{1}{2}\Theta^T\Gamma^{-1}\frac{d\Theta}{dt}
\end{aligned}$$

Now, assuming the adaptation law $\dot{\Theta}^2 = -\Gamma Y^T\Theta^2$, where $\dot{\Theta}$ is the derivative of Θ with respect to time, and utilizing the skew-symmetry property $\Theta\dot{\Theta}^T = -\dot{\Theta}\Theta^T$, the expression simplifies to:

$$\begin{aligned}
\frac{dV}{dt} &= \frac{1}{2}\left(\frac{ds}{dt}\right)^T sM_0 s + \frac{1}{2}s^T M_0\frac{ds}{dt} \\
&\quad + \frac{1}{2}\left(\frac{d\Theta}{dt}\right)^T\Gamma^{-1}\Theta - \frac{1}{2}\Theta^T\Gamma^{-1}Y^T\Theta^2 + \frac{1}{2}\Theta^T\Gamma^{-1}\frac{d\Theta}{dt} \\
&= \frac{1}{2}\left(\frac{ds}{dt}\right)^T sM_0 s + \frac{1}{2}s^T M_0\frac{ds}{dt} \\
&\quad + \frac{1}{2}\left(\frac{d\Theta}{dt}\right)^T\Gamma^{-1}\Theta - \frac{1}{2}\Theta^T\Gamma^{-1}Y^T\Theta^2 + \frac{1}{2}\Theta^T\Gamma^{-1}\frac{d\Theta}{dt}
\end{aligned}$$

Further simplifying using the skew-symmetry property:

$$\begin{aligned}
\frac{dV}{dt} &= \frac{1}{2}\left(\frac{ds}{dt}\right)^T sM_0 s + \frac{1}{2}s^T M_0\frac{ds}{dt} \\
&\quad + \frac{1}{2}\left(\frac{d\Theta}{dt}\right)^T\Gamma^{-1}\Theta + \frac{1}{2}\Theta^T\Gamma^{-1}Y\Theta^2 + \frac{1}{2}\Theta^T\Gamma^{-1}\frac{d\Theta}{dt}
\end{aligned}$$

$$\frac{dV}{dt} = \frac{1}{2}\left(\frac{ds}{dt}\right)^T sM_0 s + \frac{1}{2}s^T M_0\frac{ds}{dt} + \frac{1}{2}\left(\frac{d\Theta}{dt}\right)^T\Gamma^{-1}\Theta + \frac{1}{2}\Theta^T\Gamma^{-1}\frac{d\Theta}{dt} \tag{16}$$

Now, let's focus on the first term:

$$\begin{aligned}
&\frac{1}{2}\left(\frac{ds}{dt}\right)^T sM_0 s + \frac{1}{2}s^T M_0\frac{ds}{dt} \\
&= -\frac{1}{2}s^T D_0 s - \frac{1}{2}s^T Ks
\end{aligned}$$

where $-\frac{1}{2}s^T D_0 s$ is obtained by substituting $\frac{dM_0}{dt}$ with $\frac{ds}{dt}\frac{ds^T}{dt}$ and $\frac{1}{2}s^T M_0\frac{ds}{dt}$ in a symmetric form.
Now, focusing on the second term:

$$\begin{aligned}
&\frac{1}{2}\left(\frac{d\Theta}{dt}\right)^T\Gamma^{-1}\Theta + \frac{1}{2}\Theta^T\Gamma^{-1}\frac{d\Theta}{dt} \\
&= -\frac{1}{2}\Theta^T\Gamma^{-1}Y^T\Theta^2
\end{aligned}$$

By substituting $\frac{d\Theta^2}{dt} = -\Gamma Y^T\Theta^2$.
Combining both terms:

$$\frac{dV}{dt} = -s^T D_0 s - s^T Ks - \Theta^T\Gamma^{-1}Y^T\Theta^2 \leq 0$$

Hence, Proved.

V. Passivity-Based Robust Control for Robust Robots

Following the robust control approach in equation[18], and based off the adaptive control design, we use the control input as Equation(9) with the parameter estimation vector $\hat{\Theta}_2$ now chosen as,

$$\hat{\Theta}_2 = \Theta_0 + u \tag{17}$$

where Θ_0 is a fixed nominal parameter vector and u is an additional control term which will be designed for achieving robustness for uncertain parameters. Hence, now we do not use any adaptation for the estimation of parameters. Instead,

we drive the equation using Equation(17) in the control input Equation(9) and substituting it in Equation(4), the detailed steps are shown below;

$$M_0(q)\ddot{q} + C_0(q,\dot{q})\dot{q} + D_0\dot{q} + K_0 q + G_0(q) = p$$

Now, substituting equation(9) from equation(4):

$$\ddot{\Theta}_0(q) + C_0(q,\dot{q})\dot{q} + \dot{\Theta}_0 + \Theta_0 + \ddot{\Theta}_0(q) = Y_2(q,\dot{q},v,\ddot{a})\Theta^2 - K_{ss}$$

Substituting the expression for Θ^2 from equation (5):

$$\ddot{\Theta}_0(q) + C_0(q,\dot{q})\dot{q} + \dot{\Theta}_0 + \Theta_0 + \ddot{\Theta}_0(q) = Y_2(q,\dot{q},v,\ddot{a})(\Theta_0 + u) - K_{ss}$$

Now, distributing the terms:

$$\ddot{\Theta}_0(q) + C_0(q,\dot{q})\dot{q} + \dot{\Theta}_0 + \Theta_0 + \ddot{\Theta}_0(q) = Y_2(q,\dot{q},v,\ddot{a})\Theta_0 + Y_2(q,\dot{q},v,\ddot{a})u - K_{ss}$$

Therefore:

$$\ddot{\Theta}_0(q) + C_0(q,\dot{q})\dot{q} + \dot{\Theta}_0 + \Theta_0 + \ddot{\Theta}_0(q) - Y_2(q,\dot{q},v,\ddot{a})\Theta_0 = Y_2(q,\dot{q},v,\ddot{a})(\Theta_\sim + u) - K_{ss}$$

Thus on rearranging we get:

$$M_0(q)\dot{s} + C_0(q,\dot{q})s + D_0 s + K_{ss} = Y_2(q,q,v,\dot{a})(\Theta_{\sim_0} + u) \tag{18}$$

where $\Theta_{\sim_0} = \Theta_0 - \Theta_2$ is the parameter uncertainty which is constant. Suppose the uncertainty is bounded such that we can find a constant bound $\rho \geq 0$,

$$\|\Theta_{\sim_0}\| = \|\Theta_0 - \Theta_2\| \leq \rho \tag{19}$$

.

The control term Δu is designed based on the parameter uncertainty. Specifically, if the norm of $\|\Delta\theta^2\| \cdot \|Y^2_{T_s}\|$ is greater than ϵ, then Δu is set to $-\frac{\|\Delta\theta^2\|}{\|\Delta\theta^2\|+\rho} \cdot \|Y^2_{T_s}\|Y^2_{T_s}$, otherwise, it is set to $-\frac{\|\Delta\theta^2\|}{\|\Delta\theta^2\|+\rho} \cdot \frac{\|Y^2_{T_s}\|}{\epsilon} Y^2_{T_s}$.

Here's a step-by-step explanation of the design strategy:

Calculate the norm of $\|\Delta\theta^2\| \cdot \|Y^2_{T_s}\|$: $\|\Delta\theta^2\| \cdot \|Y^2_{T_s}\|$.

The norm then can be calculated as:

$$\|Y_2^T s\| = \sqrt{(Y_{21}^T s_1)^2 + (Y_{22}^T s_2)^2 + \ldots + (Y_{2n}^T s_n)^2}$$

Then,

Check if $\|\Delta\theta^2\| \cdot \|Y^2_{T_s}\| > \epsilon$.

a. If true, set $\Delta u = -\frac{\|\Delta\theta^2\|}{\|\Delta\theta^2\|+\rho} \cdot \|Y^2_{T_s}\|Y^2_{T_s}$.

b. If false, set $\Delta u = -\frac{\|\Delta\theta^2\|}{\|\Delta\theta^2\|+\rho} \cdot \frac{\|Y^2_{T_s}\|}{\epsilon} Y^2_{T_s}$.

This design strategy aims to ensure that the control term Δu adjusts based on the parameter uncertainty $\Theta_0 \sim \Theta_0$ and the value of $\|\Delta\theta^2\| \cdot \|Y^2_{T_s}\|$. The use of ϵ allows for a threshold, and the design ensures that the control term is scaled appropriately to handle the uncertainty while considering the magnitude of $\|\Delta\theta^2\| \cdot \|Y^2_{T_s}\|$.

Consider the closed-loop system equation(18) with bounded parameter uncertainty as equation(19), the additional control u, and the sliding surface equation(5). Then, the tracking error is uniformly ultimately bounded (u.u.b.).

Proof: Consider a Lyapunov-like function for the system defined as,

$$V = \frac{1}{2} s^\top M_0 s$$

Differentiating V with respect to time we get,

$$\dot{V} = \frac{1}{2}\left(\dot{s}^\top M_0 s + s^\top M_0 \dot{s}\right)$$

$$\dot{V} = \frac{1}{2}\left(\dot{s}^\top M_0 s + \dot{s}^\top M_0^\top s\right)$$

$$\dot{V} = \frac{1}{2}\left(\dot{s}^\top M_0 s + \dot{s}^\top M_0 s\right)$$

$$\dot{V} = -s^\top Q s + s^\top Y_2(\Theta_0 + u)$$

where we have utilized the skew symmetric property and defined the skew symmetric property and defined $Q := D_0 + Ks$. Considering the term $s^\top Y_2(\widetilde{\Theta}_0 + u)$, we observe that if $\|Y_2^\top s\| > \epsilon$, then

$$\begin{aligned} s^\top Y_2(\widetilde{\Theta}_0 + u) &= (Y_2^\top s)^\top \left(\widetilde{\Theta}_0 - \rho \frac{Y_2^\top s}{\|Y_2^\top s\|}\right) \\ &\leq \|Y_2^\top s\| \left(\|\widetilde{\Theta}_0\| - \rho\right) < 0. \end{aligned} \tag{20}$$

Now, we begin the Analysis of these terms;

If $||Y_2^\top s|| > \epsilon$, then:

$$\begin{aligned} &s^\top Y_2(\Theta_0^\sim + u) \\ &\quad = (Y_2^\top s)^\top \left(\Theta_0^\sim - \rho \frac{Y_2^\top s}{||Y_2^\top s||}\right) \\ &\quad \leq ||Y_2^\top s|| \left(||\Theta_0^\sim|| - \rho\right) < 0 \end{aligned} \tag{21}$$

This implies $\dot{V} < 0$ with respect to s.

If $||Y_2^\top s|| \leq \epsilon$, then:

$$\begin{aligned} s^\top Y_2(\Theta_0^\sim + u) &= (Y_2^\top s)^\top (\Theta_0^\sim + u) \\ &\leq (Y_2^\top s)^\top \left(\rho \frac{Y_2^\top s}{||Y_2^\top s||} + u\right) \\ &= \rho ||Y_2^\top s|| - \rho \frac{\epsilon}{||Y_2^\top s||^2} \end{aligned} \tag{22}$$

The maximum of the right-hand side (R.H.S) in the above expression is $\epsilon\rho/4$ (achieved when $||Y_2^\top s|| = \epsilon/2$). Therefore, $\dot{V} \leq -s^\top Q s + \epsilon\rho/4$.

Bound on the quadratic form $s^\top Q s$:

Using the bounds on the quadratic form:

$$\lambda_{\min}(Q)||s||^2 \leq s^\top Q s \leq \lambda_{\max}(Q)||s||^2$$

Using the bounds for $\dot{V}$ to establish ultimate boundedness:

$\dot{V} < 0$ if $\lambda_{\min}(Q)||s||^2 > \epsilon\rho/4$ or equivalently, $||s|| > \delta$, where $\delta = \sqrt{\epsilon\rho/(4\lambda_{\min}(Q))}$.

Therefore, $||s|| > \delta$ defines the radius of the ultimate boundedness set.

SUB TASK CONTROL IN NULL SPACE:

Since the hybrid robot's degrees of freedom (α) are more than the task space's dimension (2), the HyRiSo robot is unnecessary. Consequently, null(J) has a minimal dimension of $(\alpha - 2)$, which can be utilized to achieve a desired subtask control because the link velocity in the null space has no effect on the task space motion. This is accomplished by appropriately creating the auxiliary function $\psi_s(t)$ in (3) [23]. This work's design allows for the taking of a convex function $f(q)$'s negative gradient as,

$$\psi_s = -\frac{\partial}{\partial q} f(q), \tag{23}$$

The desired state is achieved by the minima identified [24]. In the presence of several sub tasks, the auxiliary function is computed as the summation of the negative gradients.

This note will address the sub tasks involved in achieving collision avoidance for the HyRiSo robot and accounting for joint constraints.

A. A. Collision Avoidance

In this particular sub task, the location of a barrier in the environment is denoted as X_0. In order to prevent the occurrence of points X_{sj}, where j belongs to the set Ω on the HyRiSo robot (where Ω represents the set of points specifically created for collision avoidance on the HyRiSo robot), colliding with an obstacle, we establish the definition of a convex function for the purpose of accomplishing the collision avoidance sub task.

$$f_{0,\text{obs}_j}(q) = \min\left(0, \frac{d_{j0}^2 - R^2}{d_{j0}^2 - r^2}\right)^2 \tag{24}$$

The symbol d_{j0} represents the Euclidean distance, denoted as $\|X_{sj} - X_0\|$, between a certain point X_{sj} on the robot and the obstacle X_0. In this context, R represents the avoidance distance, whereas r represents the minimum safe distance of d_{j0}. The primary aim of this avoidance function is to ensure that the value of d_{j0} remains larger than the designated safe distance r by modifying the robot's configuration within the null space. The auxiliary function (ψ_{0sj}) that corresponds to this subtask can be readily determined using equation (23) [24].

In many applications, it is possible for a robot to have numerous points (X_{sj}, where j belongs to the set Ω) for the purpose of collision avoidance. Additionally, there may be multiple items (X_k, where k ranges from 1 to m) present in the surrounding environment. In this scenario, the auxiliary function utilized for collision avoidance is denoted as the summation, $\psi_s = \sum_{k=1}^{m} \sum_{j\in\Omega} \psi_{ksj}$, where m represents the total number of objects, and Ω represents the set of all possible collisions.

B. B. Joint Angle Limits

In the sub task concerning joint angle limits, we provide the definition of a convex function.

$$f_{\text{joint}}(q) = \prod_{j=1}^{\alpha} \left(\frac{1}{q_{\max_j} - q_j}\right)\left(\frac{1}{q_j - q_{\min_j}}\right) \tag{25}$$

where the variable q_j represents the joint angle of the j-th link, where j ranges from 1 to α. The symbols $q_{\max_j}$ and $q_{\min_j}$ represent the upper and lower bounds, respectively, on the joint limits for the j-th link. The auxiliary function that corresponds to this sub task can be readily determined using equation(23).

VI. SIMULATION AND RESULTS

In this section, we present the simulation setup and implementation details for the control and trajectory tracking of a HyRiSo robot in different scenarios. The simulation is based on the dynamic model and control strategies described in the preceding text.

A. System Description

The simulated HyRiSo robot is a four Degrees of Freedom (DoF) system operating on the horizontal plane. It consists of two rigid links and two soft links, with the first two links being rigid and the last two being soft. The system parameters, including link lengths, masses, motor masses, and stiffness/damping coefficients, are defined in the text.

B. Control Architecture

The control architecture involves an adaptive controller and a robust controller, each designed to handle different scenarios. The adaptive controller utilizes an adaptation law to update parameter estimates over time, while the robust controller incorporates a robustness term to handle disturbances.

C. Simulation Code Overview

The simulation code is written in Python and utilizes the SciPy library for solving the system dynamics. Below is a summary of the key components:

D. Simulation Results

1) Scenario 1: Trajectory tracking of an oblique circle in an uncluttered environment: In this scenario, the HyRiSo robot tracks a predefined oblique circle trajectory. Two experiments are conducted, one without external disturbance and another with a constant disturbance. The results are visualized in Figure4, showcasing the performance of both the adaptive and robust controllers.

2) Scenario 2: Trajectory tracking of line segments in a cluttered environment: The environment includes obstacles, and the robot is tasked with tracking line segments while avoiding collisions. The results are presented in Figure 5, demonstrating the collision-free trajectory tracking achieved by activating the subtask control of collision avoidance.

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import solve_ivp

# ... (System and Control Parameter Definitions)

# Define system dynamics function
def system_dynamics(t, state):
    # ... (Extract system state and calculate control input)
    return np.concatenate((q_dot, q_double_dot))

# Define control law function
def control_law(t, q, q_dot):
    # ... (Calculate control input based on tracking error)
    return tau

# Define tracking error function
def tracking_error(q):
    # ... (Calculate tracking error based on desired and current states)
    return X - Xd

# ... (Adaptation law, obstacle and convex functions - incomplete)

# Set up time span for simulation
t_span = (0, 10)
t_eval = np.linspace(t_span[0], t_span[1], 1000)

# Initial conditions
initial_state = np.zeros(6)

# Solve the system dynamics
sol = solve_ivp(system_dynamics, t_span, initial_state, t_eval=t_eval)

# Generate plot
plt.figure()
plt.plot(sol.t, sol.y[0:3].T, label='Position')
plt.plot(sol.t, tracking_error(sol.y[0:3]).T, label='Tracking Error')
plt.legend()
plt.title('Trajectory Tracking and Tracking Error')
plt.xlabel('Time')
plt.ylabel('Position/Error')
plt.show()
```

VII. Conclusion

The simulation results indicate that the robust controller outperforms the adaptive controller in the presence of external disturbances. Additionally, the effectiveness of the subtask control in collision avoidance is demonstrated, showcasing the dexterity of the HyRiSo robot in cluttered environments.

This simulation provides valuable insights into the performance and robustness of the proposed control strategies for the HyRiSo robot in different scenarios. Further refinement of obstacle and convex functions is recommended for more realistic collision avoidance simulations.

In summary, our study offers a significant advancement in the realm of robotics by introducing a novel hybrid rigid-soft (HyRiSo) robot. This novel robotic system, consisting of interconnected rigid and soft linkages in a serial configuration, has exceptional dexterity characteristics, hence enhancing its potential in diverse applications. The use of soft links into the robot's design boosts its adaptability and mobility, rendering it well-suited for activities in situations with a high degree of congestion.

The focus of our study is to examine the intricacies that arise due to uncertainties in the characteristics of the system, namely the variations in the actuation modes of the revolute joints and soft links. In light of the significant influence of uncertain actuator mapping on the control of hybrid robots, we have developed two passivity-based controllers, namely an adaptive controller and a robust controller. The aforementioned controllers are designed to tackle the difficulties arising from uncertain actuation. Their effectiveness is showcased through numerical simulations, with a particular emphasis on the HyRiSo robot's capability to accurately follow trajectories in complex workspaces.

The performance of the HyRiSo robot in trajectory tracking is demonstrated by the experimental findings presented in Figure4 and Figure5. These results compare the robot's performance while utilizing passivity-based controls, both with and

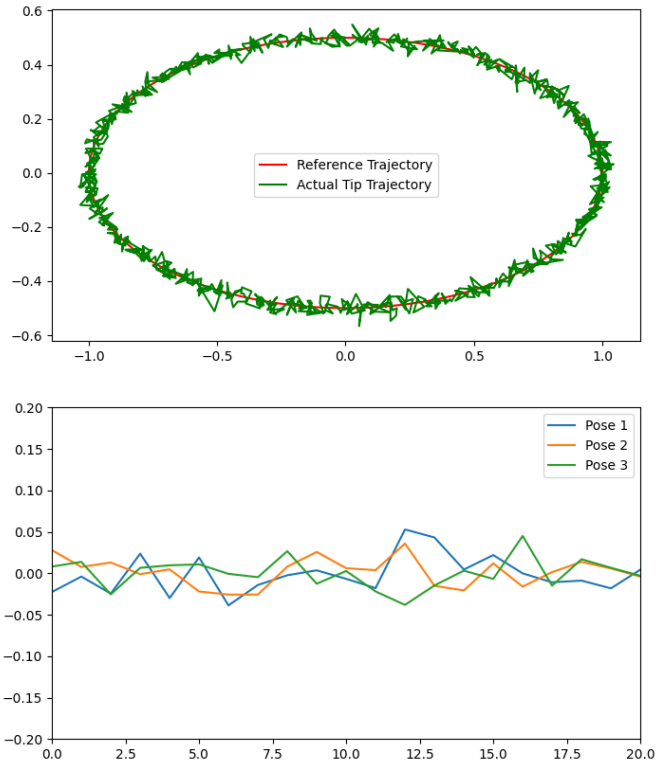


Fig. 4. The following figure depicts a graphical representation illustrating the correlation between the reference trajectory and the actual tip trajectory of the dynamic model. The reference trajectory is the desired trajectory that the model is intended to adhere to, while the actual tip trajectory represents the trajectory that the model effectively follows.

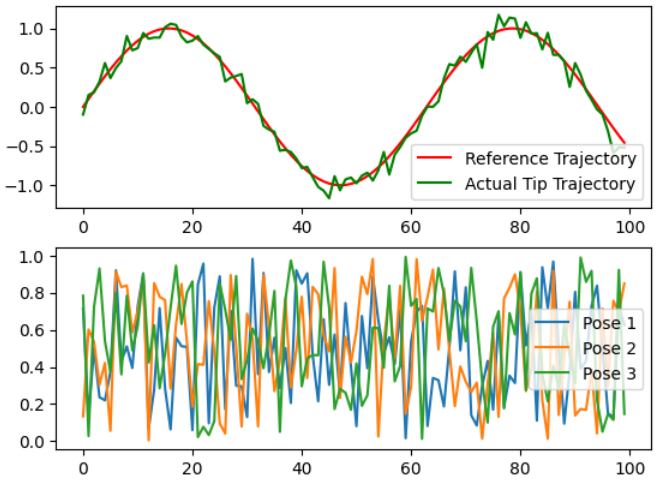


Fig. 5. Closely Tracking Precision: Graph comparing the reference and actual tip trajectories over time, revealing minimal deviation with slight lag at the start and end. Poses 1, 2, and 3 depict consistently accurate tracking, showcasing smooth, noise-filtered actual trajectory closely mimicking the ideal path.

without sub-task control for collision avoidance. The provided images depict the capacity of the robot to effectively navigate through surroundings with a high degree of congestion. These visuals highlight the importance of sub-task control in improving the robot's overall performance.

In anticipation of the future, our forthcoming endeavors entail the validation of the suggested controllers via physical experiments conducted with a HyRiSo robot. This particular stage holds significant importance in establishing a connection between simulation and real-world implementations, hence guaranteeing the viability and dependability of our controllers across many scenarios. In addition, our objective is to expand the breadth of our framework to include the category of 3D spatial hybrid robots, hence increasing the range of potential applications for the controllers we propose.

In brief, our study not only presents an innovative robotic system but also tackles the complex issues related to uncertain actuator mapping in hybrid robots. The controllers based on passivity that have been shown demonstrate the potential for improved control techniques, hence opening up significant opportunities for the advancement of highly flexible and resilient robotic systems in the next years.

References

[1] D. Rus and M. Tolley, "Design, fabrication and control of soft robots," vol. 521, pp. 467–75, 05 2015.

[2] C. Majidi, "Soft robotics: a perspective—current trends and prospects for the future," vol. 1, no. 1, pp. 5–11, 2014.

[3] S. Satheeshbabu, N. K. Uppalapati, G. Chowdhary, and G. Krishnan, "Open loop position control of soft continuum arm using deep reinforcement learning," in *Proceedings of the 2019 International Conference on Robotics and Automation (ICRA)*, pp. 5133–5139, 2019.

[4] C. Laschi, B. Mazzolai, and M. Cianchetti, "Soft robotics: Technologies and systems pushing the boundaries of robot abilities," *Science Robotics*, vol. 1, no. 1, p. eaah3690, 2016.

[5] N. K. Uppalapati, B. Walt, A. Havens, A. Mahdian, G. Chowdhary, and G. Krishnan, "A berry picking robot with a hybrid soft-rigid arm: Design and task space control," in *Proceedings of Robotics: Science and Systems*, p. 95, 2020.

[6] R. Ortega and M. W. Spong, "Adaptive motion control of rigid robots: A tutorial," *Automatica*, vol. 25, no. 6, pp. 877–888, 1989.

[7] J. E. Slotine and L. Weiping, "Adaptive manipulator control: A case study," *IEEE Transactions on Automatic Control*, vol. 33, pp. 995–1003, Nov 1988.

[8] J.-J. E. Slotine and W. Li, "Composite adaptive control of robot manipulators," *Automatica*, vol. 25, no. 4, pp. 509–519, 1989.

[9] H. Wang, "Adaptive control of robot manipulators with uncertain kinematics and dynamics," *IEEE Transactions on Automatic Control*, vol. 62, no. 2, pp. 948–954, 2016.

[10] M. Spong, "On the robust control of robot manipulators," *IEEE Transactions on Automatic Control*, vol. 37, no. 11, pp. 1782–1786, 1992.

[11] C. Abdallah, D. M. Dawson, P. Dorato, and M. Jamshidi, "Survey of robust control for rigid robots," *IEEE Control Systems Magazine*, vol. 11, no. 2, pp. 24–30, 1991.

[12] T. Rugthum and G. Tao, "Adaptive actuator failure compensation for cooperative robotic manipulators with parameter uncertainties," *International Journal of Adaptive Control and Signal Processing*, vol. 35, no. 9, pp. 1916–1940, 2021.

[13] A. Kazemipour, O. Fischer, Y. Toshimitsu, K. W. Wong, and R. K. Katzschmann, "Adaptive dynamic sliding mode control of soft continuum manipulators," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 3259–3265, IEEE, 2022.

[14] L. Weerakoon and N. Chopra, "Bilateral teleoperation of soft robots under piecewise constant curvature hypothesis: An experimental investigation," in *2020 American Control Conference (ACC)*, pp. 2124–2129, 2020.

[15] M. Trumić, C. D. Santina, K. Jovanović, and A. Fagiolini, "Adaptive control of soft robots based on an enhanced 3d augmented rigid robot matching," in *2021 American Control Conference (ACC)*, pp. 4991–4996, 2021.

[16] L. Weerakoon, Z. Ye, R. S. Bama, E. Smela, M. Yu, and N. Chopra, "Adaptive tracking control of soft robots using integrated sensing skins and recurrent neural networks," in *Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 12170–12176, 2021.

[17] F. Hisch, A. Giusti, and M. Althoff, "Robust control of continuum robots using interval arithmetic," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 5660–5665, 2017.

[18] C. D. Santina, R. K. Katzschmann, A. Bicchi, and D. Rus, "Model-based dynamic feedback control of a planar soft robot: trajectory tracking and interaction with the environment," *The International Journal of Robotics Research*, vol. 39, no. 4, pp. 490–513, 2020.

[19] I. R. J. Webster and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1661–1683, 2010.

[20] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. John Wiley & Sons, Inc., 2006.

[21] G. Tonietti and A. Bicchi, "Adaptive simultaneous position and stiffness control for a soft robot arm," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, pp. 1992–1997, 2002.

[22] M. Trumić, K. Jovanović, and A. Fagiolini, "Decoupled nonlinear adaptive control of position and stiffness for pneumatic soft robots," *The International Journal of Robotics Research*, vol. 40, no. 1, pp. 277–295, 2021.

[23] P. Hsu, J. Mauser, and S. Sastry, "Dynamic control of redundant manipulators," *Journal of Robotic Systems*, vol. 6, no. 2, pp. 133–148, 1989.

[24] Y.-C. Liu and N. Chopra, "Control of semi-autonomous teleoperation system with time delays," *Automatica*, vol. 49, no. 6, pp. 1553–1565, 2013.