MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Impedance Control of a Delta Robot

Bortoff, Scott A.; Sanders, Haley; Girindhar, Deepika

TR2023-090 July 15, 2023

Abstract

A delta robot is an attractive platform for robotic applications involving contacts and collisions, such as object assembly, because of its low mass and inertia (for low impedance and high speed), low link and joint compliance (for precision), and mechanical simplicity (for low cost). For these types of applications, impedance control is desirable, enabling a task-level controller to modulate the manipulator impedance to minimize the transfer of energy, momentum and force between the manipulator and the environment or task. In this paper, a feedback linearizing control law in task space is derived and used to construct an impedance controller for a three degree of freedom delta robot. Because the robot is a complex closed chain, neither the forward kinematics nor the feedback linearizing control law can be expressed analytically, in closed form. However we show that both can be computed algorithmically. We also show how tactile sensors, integrated into the gripper, may be used in an outer loop feedback to modify, and specifically reduce, the robot impedance. This is useful for manipulating objects of relatively low mass, or where transfer of energy, momentum or force from the robot to an object to be grasped must be minimized. We demonstrate the controller in simulation for a soft grasping primitive, and also in a laboratory experiment, where it plays speed chess.

World Congress of the International Federation of Automatic Control (IFAC) 2023

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Impedance Control of a Delta Robot

Scott A. Bortoff, * **Haley Sanders,** ** **Deepika Giridhar** ***

* *Mitsubishi Electric Research Laboratories, Cambridge, MA 02139 USA (e-mail: bortoff@merl.com)*
** *Brigham Young University, Provo, UT 84604 USA (e-mail: hpsanders98@gmail.com)*
*** *Ohio State University, Columbus, OH, 43202, USA, (e-mail: giridhar.9@osu.edu)*

Abstract: A delta robot is an attractive platform for robotic applications involving contacts and collisions, such as object assembly, because of its low mass and inertia (for low impedance and high speed), low link and joint compliance (for precision), and mechanical simplicity (for low cost). For these types of applications, impedance control is desirable, enabling a task-level controller to modulate the manipulator impedance to minimize the transfer of energy, momentum and force between the manipulator and the environment or task. In this paper, a feedback linearizing control law in task space is derived and used to construct an impedance controller for a three degree of freedom delta robot. Because the robot is a complex closed chain, neither the forward kinematics nor the feedback linearizing control law can be expressed analytically, in closed form. However we show that both can be computed algorithmically. We also show how tactile sensors, integrated into the gripper, may be used in an outer loop feedback to modify, and specifically reduce, the robot impedance. This is useful for manipulating objects of relatively low mass, or where transfer of energy, momentum or force from the robot to an object to be grasped must be minimized. We demonstrate the controller in simulation for a soft grasping primitive, and also in a laboratory experiment, where it plays speed chess.

Keywords: Robotics, mechatronic systems, nonlinear control systems.

1. INTRODUCTION

Delta robots are used in industry primarily for high speed pick-and-place operations. For these applications, high-gain PID control is sufficient, with actuation typically provided by a set of base-mounted servomotors via high-ratio gear trains. Servomotor reference trajectories are computed from a task space reference trajectory using the inverse kinematics, which are expressible analytically, in closed form. However, the delta robot's low mass and high mechanical stiffness make it an attractive platform for other applications such as robotic assembly, where collisions and contact with objects in the environment are commonplace. In these applications, when an end effector comes in contact with an object to be manipulated, a collision occurs, and the robot will transfer mechanical energy and momentum to that object. In many situations, it is critical to minimize this transfer, in order to avoid damage, for example. However, the conventional high-gain PID with actuation via a high-ratio gear train, both intended to reject disturbances in order to provide precise position tracking for pick-and-place operations, can be too stiff when contacts and collisions occur.

There are several strategies to reduce the transfer of energy and momentum from a manipulator to an object to be grasped or manipulated. The approach velocity can be reduced, which is commonly done but obviously reduces productivity. Manipulator joints can be designed with a reduced gear ratio, making them back-drivable, or may employ torque / force feedback. This is a strategy used in many cobots. Alternatively, mechanical compliance can be introduced to soften the collision, and many end effectors have been developed that effectively exploit this strategy. However, for some applications, these approaches may be insufficient. Considering more fundamental modifications to the manipulator, the mass of the manipulator may be reduced and direct drive actuation employed, effectively reducing the robot inertia and joint friction. In addition, advanced control can be used to modulate the manipulator impedance depending on the task. For example, position and velocity feedback gains can be reduced as a manipulator approaches an object. This can reduce the transfer of energy, momentum and force during and after a collision.

In this paper we derive a feedback linearizing control algorithm for a three degree of freedom delta robot, which can be used as a task-space impedance controller. Feedback linearization of a serial link robot manipulator is well-studied and arguably no longer a subject of active research (Spong and Vidyasagar (2004); Lynch and Park (2017)). However, the delta robot (and similar mechanisms such as the Gough-Stewart platform) is a complex, closed-chain mechanism, which makes the control derivation less than obvious. This is because the forward kinematics are not expressible as an analytic, closed-form mathematical expression, so neither the manipulator Jacobian, nor the feedback linearizing control law is expressible analytically. However, we show how both may be computed using

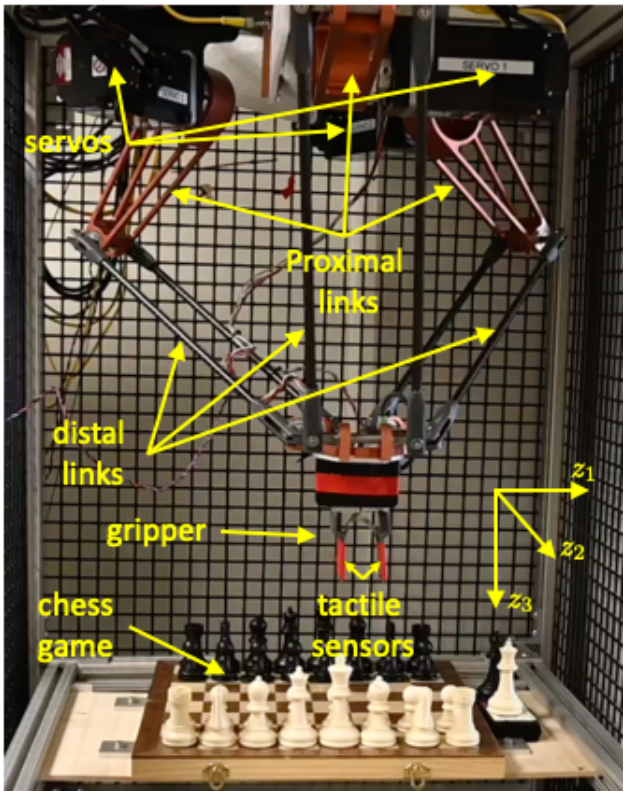

Fig. 1. MERL delta robot playing chess.

an efficient iterative algorithm. This is the basis of an impedance controller, where the manipulator impedance can be modulated independently in each of the three Cartesian directions.

As a second means to modulate manipulator impedance, we introduce tactile sensors mounted on the gripper, used in an outer-loop feedback control. We show how this can reduce manipulator impedance within the sensor bandwidth. This is useful in applications where a very soft touch is needed. We discuss real-time implementation issues with the control algorithm, and provide experimental validation with our delta robot under impedance control playing speed chess. For this example, the soft-touch impedance control with tactile feedback improves robustness of grasping and also releasing the chess pieces with respect to the small amount of position uncertainty of the pieces and also the contact with the chess board. Further, the design, with low mass links, low impedance control and direct drive, is inherently safe for human interaction.

This paper is organized as follows. In Section 2, a dynamic model is used to construct the feedback linearizing control algorithm. The tactile sensing feedback is explained in Section 3. Real-time control realization for the laboratory delta robot playing speed chess is described in Section 4, and concluding remarks provided in Section 5.

2. A SOFT GRASPING DELTA ROBOT

In this section we derive a feedback linearizing feedback control algorithm for the delta robot that serves as an inner-loop for the soft-grasping outer-loop derived in Section 3. Referring to Figs. 1-2, the delta robot, invented by Clavel (1990), consists of three (or more) identical,

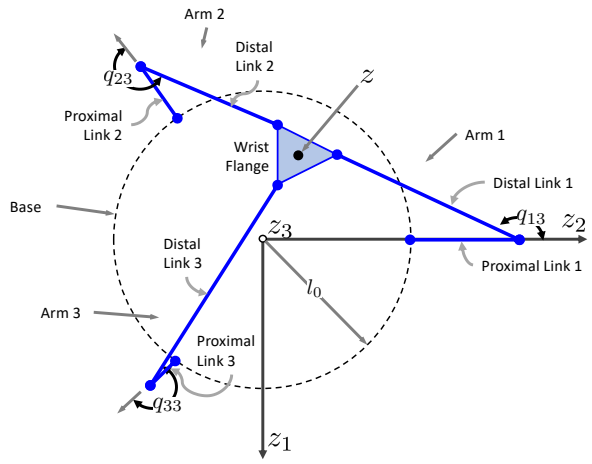

Fig. 2. MERL delta robot looking up.

under-actuated arms, arranged symmetrically about the z_3-axis (pointing down). Each arm has a proximal link, attached to the servomotor shaft at the base, and a pair of parallel distal links that are connected to the proximal link by universal joints. The six distal links are connected to the wrist flange by universal joints, so that the two distal links associated with each arm remain parallel. A gripper is mounted to the wrist flange. The servomotor angles are measured, but the universal joint angles are not. The configuration provides three translation degrees of freedom for the gripper; its orientation is fixed.

As a platform for soft grasping and more generally, robotic assembly, the delta robot offers several advantages over serial link robotic manipulators. It is relatively low mass, with carbon fiber distal links and direct-drive, base-mounted servomotors. These characteristics reduce robot impedance. Yet, its links are mechanically stiff, since the distal links incur no bending moments due to the universal joints. As such, the robot has high precision and repeatability. In addition, the translational and rotational dynamics are decoupled at the wrist, simplifying control, although the particular robot considered here lacks an articulated wrist.

The manipulator dynamics are derived by first expressing the dynamics for each unconstrained arm, lumping the two parallel distal links together because of their constrained motion. Following Bortoff (2018, 2019), the holonomic coupling constraint that represents the connections of the three arms to the wrist flange is incorporated using a constrained Lagrangian or Hamiltonian approach. The resulting index-3 differential algebraic equation (DAE) is stabilized using Baumgarte (1972, 1983), giving an index-1 DAE. Bortoff (2018) shows the DAE has 18 differential equations and variables, and six algebraic equations and variables, but the solution evolves on a six-dimensional, invariant, zero-dynamics manifold, and is mathematically equivalent to the solution of a conventional set of six dynamic equations written in six generalized coordinates. This formulation results in a single set of singularity-free equations (no switching among systems using dynamic state selection is required), despite the robot's kinematic singularities. It is computationally efficient and useful for derivation of model-based control algorithms.

Table 1. Kinematic Parameters.

Symbol	Description	Value
l_0	Base radius	0.165 m
l_1	Length of proximal link	0.2 m
l_2	Length of distal link	0.4 m
l_3	Width of wrist flange	0.0562 m
l_4	Height of gripper	0.075 m

Let $q_i \in \mathbb{R}^3$ denote the joint angles of arm i, ordered so that q_{i1}, is the ith servomotor angle, and the remaining two angles correspond to the universal joints, for $1 \leq i \leq 3$. The position of the geometric center of the gripper relative to the base frame, denoted $z \in \mathbb{R}^3$, is expressed in terms of arm 1 coordinates as

$$z = \psi(q_1) = \begin{bmatrix} l_2 \sin(q_{12}) \sin(q_{13}) \\ l_0 - l_3 + l_1 \cos(q_{11}) + l_2 \cos(q_{12}) \\ l_4 + l_1 \sin(q_{11}) + l_2 \sin(q_{12}) \cos(q_{13}) \end{bmatrix}, \quad (1)$$

with parameters listed in Table 1. Define $z = \psi_1(q_1) := \psi(q_1)$, $z = \psi_2(q_2) := R_2\psi(q_2)$ and $z = \psi_3(q_3) := R_3\psi(q_3)$ as the forward kinematics of arms 1, 2 and 3 respectively, where $R_2 = R_z(-2\pi/3)$, $R_3 = R_z(2\pi/3)$ and $R_z(\cdot)$ is the rotation matrix about the z_3-axis.

Following Bortoff (2018), define $q = [q_1^T \; q_2^T \; q_3^T]^T \in \mathbb{R}^9$ and $v = \dot{q}$. Then the robot dynamics are the index 1 DAE

$$\dot{q} = v \quad (2a)$$

$$M(q)\dot{v} + C(q,v) + D(v) + G(q) = H^T(q)\lambda + B(u + \tau_u) + \tau_v \quad (2b)$$

$$\ddot{h}(q,v,\dot{v}) + \alpha_1 \dot{h}(q,v) + \alpha_0 h(q) = 0 \quad (2c)$$

where the constraint $h(q) : \mathbb{R}^9 \to \mathbb{R}^6$ satisfies

$$h(q) = \begin{bmatrix} \psi_1(q_1) - \psi_2(q_2) \\ \psi_1(q_1) - \psi_3(q_3) \end{bmatrix} = 0, \quad (3)$$

$H(q)$ is the Jacobian of $h(q)$, $\lambda \in \mathbb{R}^6$ is the Lagrange multiplier, $\tau_u \in \mathbb{R}^3$ is the vector of *matched* disturbance torques, $\tau_v \in \mathbb{R}^9$ is the vector of *virtual* disturbance torques, $u \in \mathbb{R}^3$ is the control input torque vector, positive constants α_0 and α_1 are such that $s^2 + \alpha_1 s + \alpha_0$ is a Hurtwitz (stable) polynomial. Bortoff (2018) provides explicit formulas for M, C, D, G and B (the inertia matrix, Coriolis and centripetal torque, damping, gravity, and input vectors, respectively). Finally,

$$y = [q_{11} \; q_{21} \; q_{31}]^T \quad (4)$$

is the vector of measured servomotor angles. The other six joint angles are not measured. The distinction between τ_u and τ_v is explained in the next subsections.

Forward Kinematics

The delta robot is a complex [1] closed kinematic chain, as defined by Merlet and Gosselin (2008), which means that it is not possible to compute the forward kinematics (the function from measured servomotor angles to the location of the end effector) as a closed-form analytic expression. It is computed algorithmically from (3). Defining

$$x = [q_{12} \; q_{13} \; q_{22} \; q_{23} \; q_{32} \; q_{33}]^T, \quad (5)$$

whose elements are the unmeasured joint angles, rewrite (3) by reordering the arguments as

$$h(x,y) = 0. \quad (6)$$

[1] Meaning one link has a degree of connectivity ≥ 3.

For a given y, the solution x to these six nonlinear equations can be computed iteratively using, for example, Newton's method,

$$\frac{\partial h}{\partial x}(x_k, y) \cdot (x_{k+1} - x_k) = -h(x_k, y), \quad (7)$$

which converges locally and quadratically assuming the robot is not near a kinematic singularity (which implies $\frac{\partial h}{\partial x}$ is nonsingular). Then (1) is used to compute z.

Inverse Kinematics

The inverse kinematics, i.e., the function from z to y (and q) is expressed analytically by computing the angles of the planar triangle formed by each arm using the Law of Sines, since the length of each side is known, and then solving a quadratic equation. This gives two expressions for q_i for each arm, given a reachable z. One solution is for the arm pointed outward, which is the normal mode of operation, while the other is for the arm pointed inward, which is not used. Details are omitted.

Manipulator Jacobians

The manipulator Jacobian is the map between end effector velocity / force and joint angular velocity / torque, respectively, and is important for (a) modeling and simulating the effect of force disturbances on the end effector, and also for (b) control purposes, such as computing reference joint velocities from reference velocities of the end effector. For serial link robots, it is the mathematical Jacobian of the forward kinematics. For the delta robot, this is nontrivial because the forward kinematics are not expressible analytically. In fact, the two uses for the Jacobian described above require two different expressions for the Jacobian.

First, in order to model and simulate the effect of force disturbances, we compute the Jacobian that maps an external disturbance force $f \in \mathbb{R}^3$ that is applied to the gripper to the vector of virtual torques τ_v that is applied to *all* of the joints. First, sum the forward kinematics of each arm,

$$z = (\psi_1(q_1) + \psi_2(q_2) + \psi_3(q_3))/3. \quad (8)$$

Differentiating gives the 3×9 *virtual* Jacobian

$$J_v(q) = \frac{1}{3}\left[\frac{\partial \psi_1}{\partial q}(q_1) \quad \frac{\partial \psi_2}{\partial q}(q_2) \quad \frac{\partial \psi_3}{\partial q}(q_3)\right], \quad (9)$$

which is used to compute τ_v in (2b),

$$\tau_v = J_v^T(q) \cdot f. \quad (10)$$

This is non-singular, even at manipulator kinematic singularities.

The second formulation we denote the *control* Jacobian J_c, which is the map between end effector forces / velocities and *servomotor* torques / velocities. Following Okasha and Bortoff (2020), by the Implicit Function Theorem, there exists $g : \mathbb{R}^3 \to \mathbb{R}^6$ such that

$$h(g(y), y) = 0, \quad (11)$$

so that we may write $x = g(y)$, if $\frac{\partial h}{\partial x}$ is nonsingular in a neighborhood of (x, y). Write ψ_1, ψ_2 and ψ_3 in the (x, y)-coordinates to compute the Jacobians

$$\Psi_x := \left[\frac{\partial\psi_1}{\partial x_1}\ \frac{\partial\psi_1}{\partial x_2}\ \frac{\partial\psi_2}{\partial x_3}\ \frac{\partial\psi_2}{\partial x_4}\ \frac{\partial\psi_3}{\partial x_5}\ \frac{\partial\psi_3}{\partial x_6}\right] \in \mathbb{R}^{3\times 6} \quad (12a)$$

$$\Psi_y := \left[\frac{\partial\psi_1}{\partial y_1}\ \frac{\partial\psi_2}{\partial y_2}\ \frac{\partial\psi_3}{\partial y_3}\right] \in \mathbb{R}^{3\times 3}. \quad (12b)$$

Then, using the chain rule, the 3×3 control Jacobian is

$$J_c = \frac{\partial z}{\partial y} = \Psi_x \cdot \frac{\partial g}{\partial y} + \Psi_y\,, \quad (13)$$

where

$$\frac{\partial g}{\partial y} = -\left(\frac{\partial h}{\partial x}\right)^{-1} \cdot \frac{\partial h}{\partial y}. \quad (14)$$

For time-domain simulation of the effects of disturbance f, $\tau_v = J_v^T \cdot f$, and not $\tau_u = J_c^T \cdot f$, should be used, because it is analytic and globally defined, whereas J_c is undefined at kinematic singularities. Moreover, using J_c for simulation would make the DAE non-smooth because of the termination condition from (7), causing problems with DAE solver error control as described by Cellier and Greifeneder (1991) and Cellier (2006).

Gravity Compensation

At each sample time, the digital controller measures the servomotor angles y, computes the forward kinematics (7), giving q, z, and x, and then computes J_c. Referring to (2a), τ_u is computed at each sample time to cancel the effect of $G(q)$, by solving the six-dimensional, globally nonsingular linear system

$$[B \quad H^T(q)] \begin{bmatrix} \tau_u \\ \lambda \end{bmatrix} = G(q) \quad (15)$$

for τ_u and λ.

Exact Feedback Linearization in Joint Space

Applying the value τ_u computed in (15) to (2a), reordering the equations and expressing them in the (y, x)-coordinates results in the gravity-compensated model

$$\begin{bmatrix} \bar{M}_{11}(q) & \bar{M}_{12}(q) \\ \bar{M}_{21}(q) & \bar{M}_{22}(q) \end{bmatrix} \begin{bmatrix} \ddot{y} \\ \ddot{x} \end{bmatrix} + \begin{bmatrix} \bar{C}_1(q,v) \\ \bar{C}_2(q,v) \end{bmatrix} + \begin{bmatrix} \bar{D}_1(v) \\ \bar{D}_2(v) \end{bmatrix} = \begin{bmatrix} u \\ 0 \end{bmatrix}, \quad (16)$$

where the overbar denotes the reordering. Next write $\ddot{x}$ in terms of $\ddot{y}$ by differentiating (6) twice, giving

$$\frac{\partial h}{\partial y}\ddot{y} + \frac{\partial h}{\partial x}\ddot{x} + \dot{y}^T \frac{\partial^2 h}{\partial y^2}\dot{y} + \dot{x}^T \frac{\partial^2 h}{\partial x^2}\dot{x} = 0. \quad (17)$$

Solving for $\ddot{x}$ and substituting into (16) gives

$$\bar{M}_y \cdot \ddot{y} + \bar{D}_1 + a = u, \quad (18)$$

where, using (14), the 3×3 inertia matrix

$$\bar{M}_y = \bar{M}_{11} - \bar{M}_{12} \cdot \frac{\partial h}{\partial x}^{-1} \cdot \frac{\partial h}{\partial y} = \bar{M}_{11} + \bar{M}_{12} \cdot \frac{\partial g}{\partial y}, \quad (19)$$

and a includes the higher-order terms (the second two terms in (17) and $\bar{C}_1$). The feedback linearizing control in the y-coordinates is then

$$u = \bar{M}_y v + \bar{D}_1 + a, \quad (20)$$

where

$$v = k_p(r - y) + k_v(\dot{r} - \dot{y}) + \ddot{r}, \quad (21)$$

which, applied to (16) gives

$$(\ddot{y} - \ddot{r}) + k_v(\dot{y} - \dot{r}) + k_p(y - r) = 0, \quad (22)$$

where r, $\dot{r}$ and $\ddot{r}$ are the reference trajectory for y and its first two derivatives, $k_p = \mathrm{diag}\{k_{p1}\ k_{p2}\ k_{p3}\}$ and $k_v = \mathrm{diag}\{k_{v1}\ k_{v2}\ k_{v3}\}$ are position and velocity gains, respectively.

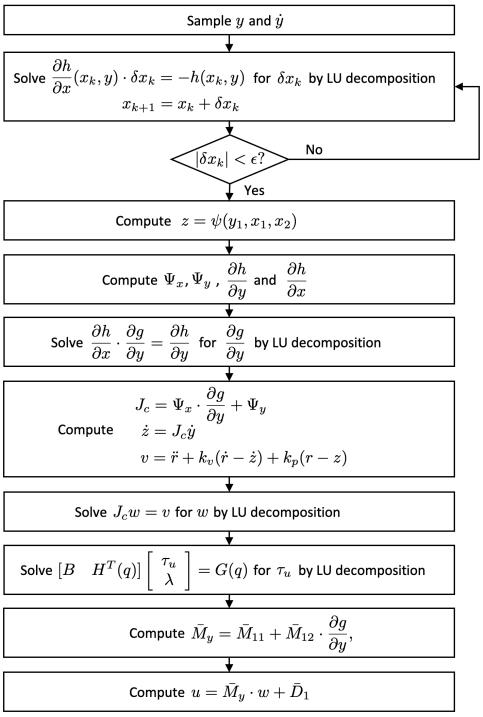

Fig. 3. Flow chart for approximate feedback linearization.

Approximate Feedback Linearization in Task Space

For the delta robot, the higher order terms a in (18) are small in magnitude. The Corriolis and centripetal torque $\bar{C}$ is small because the distal link and wrist flange masses are small, and the maximum joint velocities in practice rarely exceed 5 rad/s (for our robot), because the delta robot has a relatively small work volume. Therefore, we neglect a, and use the first two terms of (20) as an approximate feedback linearization in the y-coordinates.

Finally, to express the approximate feedback linearization in Cartesian task space (z-coordinates), substitute $\ddot{z} = J_c\ddot{y} + \dot{J}_c\dot{y} \approx J_c\ddot{y}$ into (18) giving

$$u = \bar{M}_y J_c^{-1} v + \bar{D}_1, \quad (23)$$

which, when applied to (18) gives

$$\ddot{e} + k_v\dot{e} + k_p e \approx 0,$$

where $e = z - r$, and r, $\dot{r}$ and $\ddot{r}$ in (21) are redefined to be the reference trajectory and its derivatives in the task coordinates z. Control (23) and (15) decouple the dynamics in each Cartesian direction, and render them (approximately) linear and second order. Outer loop gains k_p and k_v may be adjusted to modify the manipulator impedance independently in each direction. Fig. 3 is a flow chart for the complete approximate feedback linearizing control algorithm.

3. TACTILE SENSOR FEEDBACK

A parallel gripper is attached to the underside of the wrist flange, with its axis aligned with the z_1-axis. The two fingers do not operate independently; only the distance between them can be controlled. Denote the right finger position relative to the center of the end effector as z_4. Then the robot impedance is ratio of force applied to the gripper left or right finger, f_i, $i = 1, 2$, respectively, to the

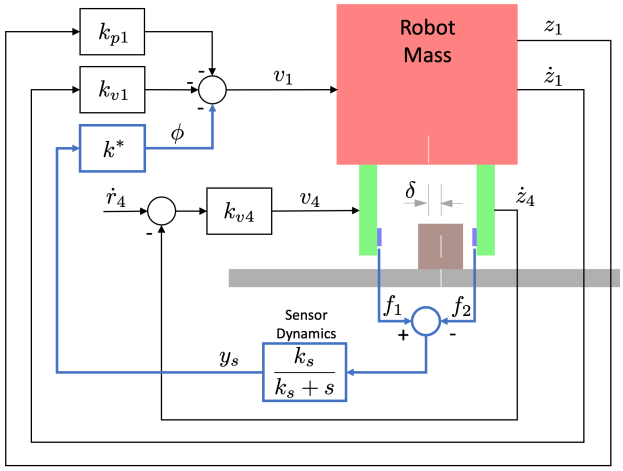

Fig. 4. Delta robot with tactile sensor (blue) feedback integrated into the fingers (green) of a parallel gripper, positioned to grasp a block (brown).

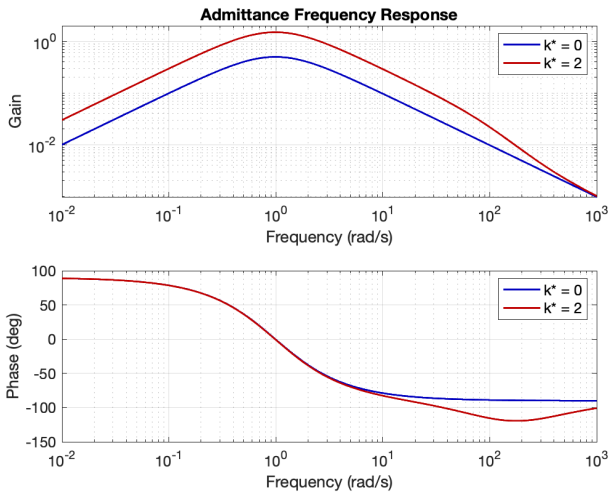

Fig. 5. Bode plot of robot admittance with tactile sensor feedback, with $k_p = 1$, $k_v = 2$, $k_s = 100$, and $k^* = 0$ (blue) and $k^* = 2$ (red).

velocity of the finger, $\dot{z}_1 + \dot{z}_4$, and is generally represented in the frequency domain.

The gains k_p and k_v in (23) shape the robot impedance. For contact applications, it is desirable to reduce the impedance, and it is common to set $k_p = 0$ and make k_v small. However, this can have an adverse affect on robustness with respect to disturbances and model uncertainty, and result in unacceptably large reference tracking errors. An alternative means to shape the impedance is to use tactile sensor feedback.

Consider the feedback system diagrammed in Fig. 4. Force sensors mounted to the insides of the fingers directly measure contact forces f_i. Their difference is amplified by k^*, fed back, and subtracted from v_1 in (21). Conceptually, when one finger contacts an object, the sensed force is amplified and fed back to provide an outer loop force in the same direction, effectively reducing the impedance of the robot.

A Bode diagram of the admittance (inverse of the impedance)

$$A(s) = \frac{s(Z_1(s) + Z_4(s))}{F(s)}$$

for the combined robot and gripper is shown in Fig. 5 for $k^* = 0$ (blue) and $k^* = 2$ (red). (Note: $A(s)$ depends on kinematic, dynamic and control parameters of the gripper and is omitted for space reasons.) The force sensor dynamics are modeled as a first-order low pass filter with bandwidth $k_s = 100$ rad/s. This particular gripper provides only velocity control, as shown in Fig. 4, and has a large gear ratio, making its effective inertia and impedance relatively large compared to the robot. Because of this, force feedback to the gripper velocity loop is not as effective. The effect of the tactile sensor feedback is to increase the robot admittance (decrease the robot impedance) in the sensor bandwidth, in this case by 10dB. This is a useful means to shape the robot impedance beyond modulation of the gains k_p and k_v, and more complex forms of compensation than just k^* are possible.

3.1 Soft Grasping Example

Consider the situation diagrammed in Fig. 4, where a parallel gripper is positioned to grasp a block (brown), which rests on a surface (gray). The block location is offset from the gripper centerline by an unknown distance δ. The objective is to close the fingers and grasp the block while minimizing the resulting horizontal sliding motion. Note that if the robot horizontal position z_1 was to be held fixed, then the block would slide a distance δ.

When the first finger ($i = 1$ or 2, depending on the sign of δ) contacts the block, this feedback actuates the robot to maintain the force f_i to be small relative to the nominal case ($k^* = 0$), reducing the transfer of momentum to the block after the first collision event. This feedback remains active before and after both of the collisions occur, and, together with the velocity control for the fingers, it naturally results in both fingers closing, contacting the block in sequence, with a final contact force of $f_i = \dot{r}_4 k_{v4}$. The grasping operation is accomplished with no hybrid switching or mode changing, and results in minimal perturbation of block.

The scenario was simulated for the delta robot dynamics (1) - (4), (9) and (10), with feedback control (23), using the Modelica realization described by Bortoff (2018) and Okasha and Bortoff (2020). The parameters of the robot are identified from our laboratory robot, and the block parameters represent a wooden Jenga piece. Static friction is modeled assuming wood-on-wood contact.

Figs. 6 and 7 show the results of a grasp simulation, comparing active tactile feedback (red) with the non-tactile feedback case (blue). For the both cases, the z_1 direction gains were $k_{v1} = 0.1$ and $k_{p1} = 0$, so that it had a very low impedance, while the other directions have higher gain. Note especially the block displacement which was nearly zero using the tactile feedback, whereas for the nominal case, the block slid 4 mm, nearly half the possible $\delta = 10$ mm.

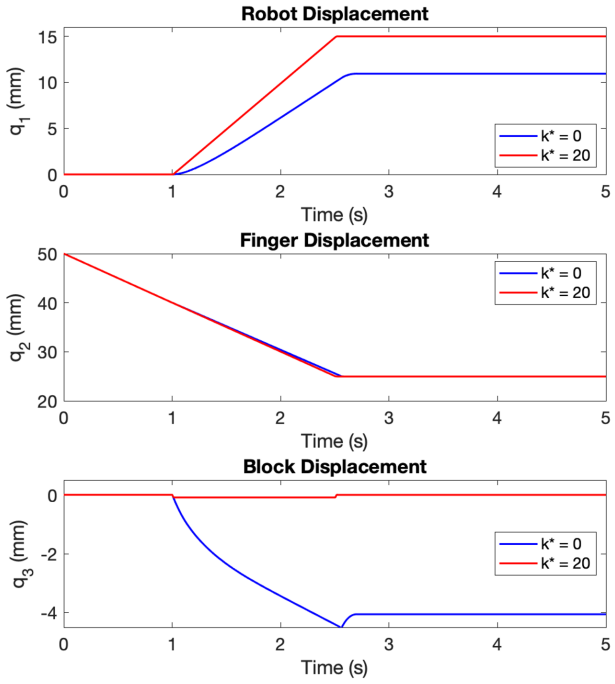

Fig. 6. Grasp simulation results showing resulting robot, finger and block displacement for $k^* = 0$ (blue) and $k^* = 2$ (red).

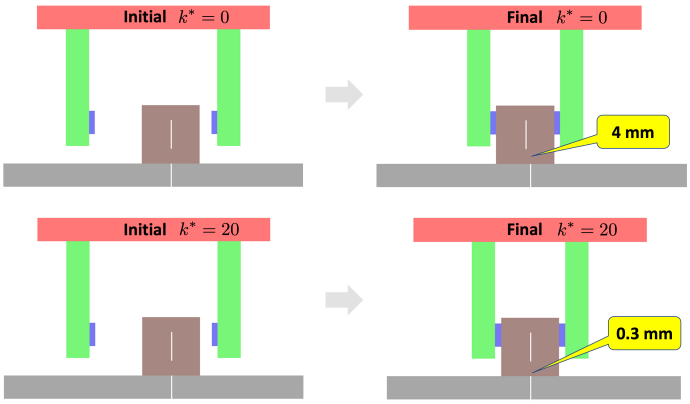

Fig. 7. Initial and final displacement of grasp simulation, showing approximately 4mm motion of the block with no tactile sensor feedback (top right), but less than 1mm motion with $k^* = 2$ (bottom right). The white lines on the block (brown) and surface (gray) are aligned in the initial positions (left).

4. CHESS PLAYING DEMONSTRATION

The MERL delta robot shown in Figs. 1 and 8, is a custom-built research robot intended for assembly control experiments. Each machined aluminum proximal link is directly actuated by a Mitsubishi Electric HG-KR-73B rotary AC servomotor with 0.75 kW rated power, 2.4 Nm rated torque, and a 22-bit rotary encoder resolution, giving a task-space resolution of less than 5μm. The distal links are are hollow carbon fiber tubes to reduce weight. A parallel gripper by New Scale Robotics, with a modified interface and control, is mounted to the underside of the aluminum wrist flange. Each servomotor is driven by an MR-J4-B servo amplifier in torque mode, which is in turn controlled by a MR-MC210 motion control board that is

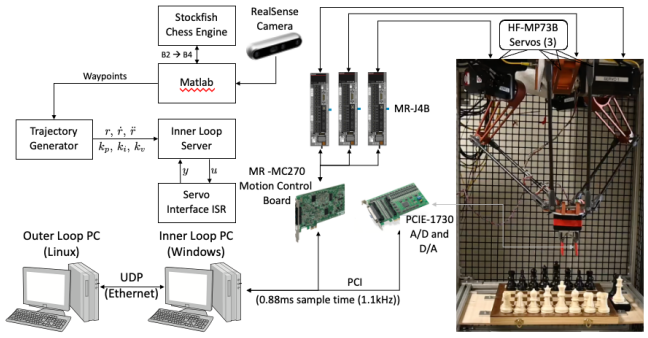

Fig. 8. System diagram.

installed in a PCI slot of a PC. The gripper is actuated by a commanded velocity, provided by an analog signal from a D/A board installed in the same PC. Tactilus piezoelectric tactile sensors mounted to the inside of each gripper finger, are also interfaced through the same A/D board.

As a demonstration of impedance control (21), (23) and the soft-touch force feedback, a speed chess demonstration was programmed. Of course, robots playing chess is not novel; the classic Mechanical Turk played chess in the 18th century! But in July 2022, a chess playing robot broke a 7-year-old player's finger at a chess tournament in Moscow, as reported by CNN [2], highlighting the risks of human interaction with a manipulator not intended for such purposes. The contribution here is to demonstrate a low-mass, direct drive robot controlled by the task-space variable-impedance feedback (23), which is inherently safer for human-robot interaction, yet quite fast.

The system can play chess against a human opponent or against itself. Referring to Fig. 8, the open-source StockFish chess engine was interfaced to Matlab and computes moves. When playing a human opponent, a RealSense camera monitors the board, determines the human's chess move, and communicates that to StockFish, which computes the robot's next move. From this, a set of waypoints is computed, and a minimum-time trajectory generator computes a reference trajectory and feedback gains for the next move. During play, chess pieces are grasped by the robot, and either moved to a new location on the board, or removed from play (if it has been captured). Feedback gains are reduced as a function of distance from the board, which minimizes the impact when the piece is put on the board. Each piece is subject to a small amount of position uncertainty, and each piece has a different shape. (The vision system is not used to determine precise locations of each piece; rather, they are all assumed to be in the center of their square.) Between each move, the robot moves to a home position so that it does not obstruct the view of the camera, and is not in the way of a human player.

The software is organized in a client-server architecture, with communication among the components by UDP. Processes run on two computers. An interrupt service routine (ISR) runs on the first PC, and is triggered by the MR-MC210 motion control board every 0.88ms to measure the servomotor angles and velocities, and also to write the

[2] https://www.cnn.com/2022/07/25/europe/chess-robot-russia-boy-finger-intl-scli/index.html

previously computed torques to the servos. The inner-loop control (23), computed as in Fig. 3, is realized as a server running on the same PC. It is triggered every time the ISR runs. The server takes approximately 90μs to compute (23), which is about 10% of the available time between samples. Because of the high sample rate (1.1kHz), the Newton solver (7) runs only 1-2 iterations to reduce error to less than 8 decimal places. The rest of the control consists of evaluation of linear and nonlinear functions, and solving linear systems of dimension 3, 6 or 9 by LU decomposition. All software is written in C.

Matlab, running on the same PC, interfaces to the StockFish engine. A script runs after every move to determine a human player's move via the RealSense camera. Matlab maintains the state of the chess board, and computes a set of way points for the robot to execute its next move. These are sent to the Trajectory Generator, which computes the reference r and its first two derivatives to interpolate the way points. The reference is a C^2 (so $\ddot{r}(t)$ is continuously differentiable), and is nearly the minimum time trajectory, within upper bounds on position, velocity, acceleration and jerk. The reference follows straight lines between way points, with clothoid splines inserted at the intermediate way points to ensure the jerk remains bounded, which is important to minimize vibration.

Fig. 9 shows trajectories in the z-coordinates of one game, in which the robot plays itself. The tracking error is good during motion, but as can be seen in the lower plot, there is larger tracking error for the z_3 coordinate when it grasps or places pieces, because the gains are detuned when the robot is closer to the board.

5. CONCLUSION

A feedback linearizing control algorithm for a delta robot in task coordinates was presented, with detail provided about the steps for efficient computation. This is useful for impedance control of a delta robot, where its impedance is decoupled in each Cartesian direction, and can be modulated by a higher level control. Tactile sensors were shown to be useful to further modulate manipulator impedance. A chess playing demonstration was also described that uses both. These results may be useful in developing more transparent robots, with reduced mass, inertia and friction, that are especially useful for robotic applications involving contact and collision such as robotic assembly.

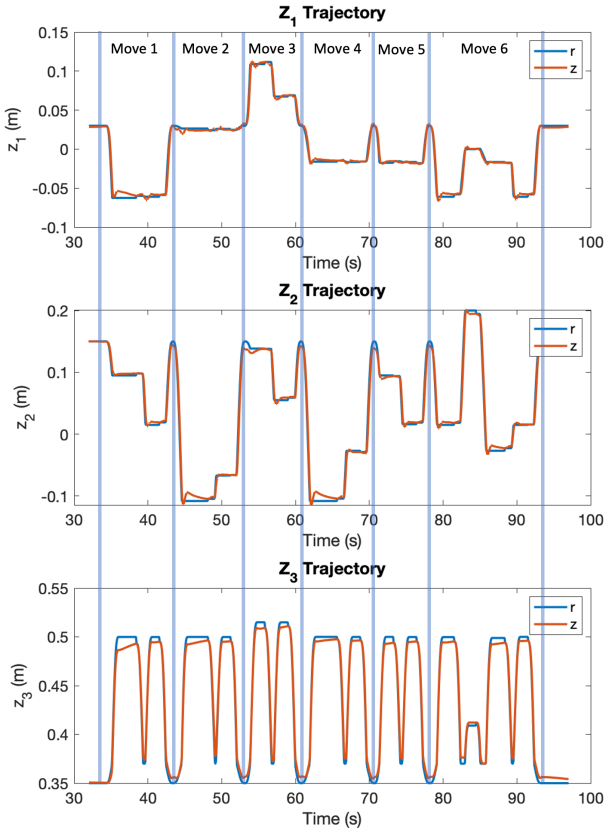

Fig. 9. First six moves of a chess game playing against itself. Home position is denoted by the vertical blue lines. In move 6, black captures a white piece, dropping it into a box next to the board.

REFERENCES

Baumgarte, J.W. (1972). Stabilization of constraints and integrals of motion in dynamic systems. *Computer Methods in Applied Mechanics and Engineering*, 1, 1–16.

Baumgarte, J.W. (1983). A new method of stabilization for holonomic constraints. *ASME Journal of Applied Mechanics*, 50, 869–870.

Bortoff, S.A. (2018). Object-oriented modeling and control of delta robots. In *IEEE Conference on Control Technology and Applications*, 251–258.

Bortoff, S.A. (2019). Using Baumgarte's method for index reduction in Modelica. In *Proceedings of the 13th International Modelica Conference*, 333–342.

Brinker, J. and Corves, B. (2015). A survey of parallel robots with delta-like architecture. In *Proceedings of the 14th IFToMM World Congress*.

Brinker, J., Corves, B., and Wahle, M. (2015). A comparative study of inverse dynamics based on Clavel's delta robot. In *Proceedings of the 14th IFToMM World Congress*.

Brinker, J., Funk, N., Ingenlath, P., Takeda, Y., and Corves, B. (2017). Comparative study of serial-parallel delta robots with full orientation capabilities. *IEEE Robotics and Automation Letters*, 2(1), 920–926.

Cellier, F.E. (2006). *Continuous System Simulation*. Springer.

Cellier, F.E. and Greifeneder, J. (1991). *Continuous System Modeling*. Springer.

Clavel, R. (1990). Device for the movement and positioning of an element in space. U.S. Patent 4,976,582.

Isidori, A. (1989). *Nonlinear Control Systems*. Springer-Verlag.

Lynch, K.M. and Park, F.C. (2017). *Modern Robotics: Mechanics, Planning and Control*.

Merlet, J.P. and Gosselin, C. (2008). *Springer Handbook of Robotics*, chapter Parallel Mechanisms and Robots. Springer.

Okasha, A. and Bortoff, S.A. (2020). Modelica-based control of a delta robot. In *Proceedings of the ASME 2020 Dynamic Systems and Control Conference*.

Spong, M.M. and Vidyasagar, M. (2004). *Robot Dynamics and Control*. Wiley.