

Trajectory-Planning and Normalized-Variable Control for Parallel Pick-and-Place Robots

Peyman Karimi Eskandary*

PhD Graduate

Department of Mechanical Engineering
& Centre for Intelligent Machines
McGill University
Montreal, Quebec, H3A 0G4
Email: pkarimie@cim.mcgill.ca

Bruno Belzile

Postdoctoral Fellow

Department of Mechanical Engineering
& Centre for Intelligent Machines
McGill University
Montreal, Quebec, H3A 0G4
Email: bruno@cim.mcgill.ca

Jorge Angeles

Professor

Department of Mechanical Engineering
& Centre for Intelligent Machines
McGill University
Montreal, Quebec, H3A 0G4
Email: angeles@cim.mcgill.ca

ABSTRACT

Trajectory-planning and an efficient control scheme play a crucial role in improving the performance of pick-and-place robots. This paper introduces a novel method of trajectory-planning with cycle time and path constraints. Assuming that a smooth trajectory is given, to be followed within a prescribed cycle time, the newly proposed method of trajectory-planning removes the torque peaks of the actuators by a suitable scheduling of the velocity of the moving plate. Since pick-and-place robots are usually expected to meet the end-poses in a certain time span, while disregarding the intermediate poses, the velocity can be tuned properly around the critical points of the trajectory by means of a time-scaling function. Moreover, the authors report the formulation of a linear quadratic regulator (LQR) controller with normalized variables, to be used in conjunction with our trajectory-tracking control scheme for an in-house-developed Schönflies-motion generator. This parallel robot offers a functionally symmetric, single-loop architecture, with an isostatic kinematic chain, and virtually unlimited rotatability of its gripper. A comparison between two actuation systems developed by the authors is conducted via simulation results.

1 Introduction

The unprecedented growth of automation in different industrial sectors calls upon a wide range of robots for special tasks. Manipulation tasks fall into two major configuration categories: trajectory-tracking, as needed for additive or subtractive manufacturing, and pick-and-place operations (PPO). A trajectory-tracking task requires the robot to follow a desired path space with a given schedule. On the other hand, PPO refers to carrying a payload from the starting pose to the final pose, regardless of the poses visited in-between, as long as the payload does not collide with the environment. Moreover, the PPO speed over the trajectory is usually not crucial, as long as it meets the desired cycle time. Pick-and-place robots have broad applications in industry, in operations such as moving parts between workstations. The focus of this paper is on pick-and-place robots with four degrees of freedom (dof) producing Schönflies motions, namely, Schönflies-motion generators (SMGs). These robots are capable of translation in three independent directions, and rotation about one axis of fixed orientation (usually vertical) [1], similar to the motion of a waiter's tray.

While serial architectures have a relatively large workspace, parallel architectures benefit from a higher load-carrying capacity, operation precision and speed. These features led to more emphasis in recent years on parallel manipulators,

*Address all correspondence related to ASME style format and figures to this author.

a.k.a. Parallel-Kinematics Machines (PKMs). Several parallel architectures have been proposed to generate Schönflies-motions [2–7]. Current parallel SMGs are mostly designed based on either the three-limb Delta architecture [8] enhanced with a telescopic Cardan shaft to transmit a rotation to the base plate (BP), or a four-limb architecture [9], also derived from the Delta architecture. The well-known SMGs in the market are ABB Robotics’ IRB FlexPicker [10] and Adept Technology Inc.’s Quattro s650H.

Recently, our team picked one of the four parallel SMG architectures proposed by Lee and Lee [11, 12] to develop a novel SMG. In contrast to the popular above-mentioned parallel architectures, the two-limb architectures proposed by the Lees benefit from overall simplicity, isostaticity, high rotatability of the “moving platform”¹ (MP), and a fully symmetric layout [13, 14]. The Isostatic Schönflies Motion Generator (ISMG) designed and fabricated at the Centre for Intelligent Machines (CIM), McGill University, is termed the Peppermill Carrier (PMC) because of the formal and functional similarity of its gripper-carrying rod² to the familiar peppermill, seen in Italian restaurants³. The putative MP in our case consists of two coaxial screw joints with identical pitches, but of opposite hands. Each limb of the PMC is run by one novel actuator, dubbed the C-drive (cylindrical drive). This drive operates based on a cylindrical differential mechanism [15, 16], capable of generating the two-dof cylindrical motion subgroup [1], i.e., independent translational and rotational motions, with the former along a direction parallel to the axis of the latter.

Beyond the architecture and the kinematics of the PMC, another element that must be considered is the control strategy. Indeed, high-speed operations make it necessary to have a carefully tuned controller. In order to achieve that goal, linear quadratic control theory is an interesting avenue for systems with multiple inputs and multiple outputs (MIMO). The LQR is one solution to minimize a cost function and obtain optimal feedback gains [17]. The formulation on which the LQR is based was first proposed by Kalman [18]. While the LQR is intended for linear systems, it can also be applied to their non-linear counterparts, such as the well-known inverted pendulum problem [19]. It can also be applied to trajectory tracking [20].

Several methods of trajectory-planning have been proposed to improve the performance of manipulators, each with different constraints and objectives. Bourbonnais et al. used the spline stochastic approach to minimize the cycle time of a pick-and-place robot [21], while others tried to keep the motion jerk continuous with kinematic constraints [22–25]. Moreover, having knowledge about the dynamic response of a robot plays an essential role in trajectory-planning; therefore, many methods have been established based on the robot model [26, 27]. In addition, model-based trajectory-planning becomes a crucial tool to find the optimum feasible path for cable robots [28–30].

The focus of this paper is the development of a new scheme for trajectory-planning with cycle time and path constraints. The goal is to keep constant the cycle time and the desired path of the Peppermill, as imposed by the user, while scheduling the velocity in such a way that the maximum torque of the motors is kept within given bounds. One of the main obstacles to speed up a PPO is the required torque peaks, which exceed the nominal characteristics of the actuators and result in deterioration of trajectory-tracking precision. Moreover, the removed torque peaks allow the user to increase the speed of operation. The velocity scheduling introduced in this paper modifies the time step of the time array, which belongs to the prescribed trajectory. This novel technique is applicable to any given trajectory upon identifying the critical points, in order to improve the performance of pick-and-place robots.

2 Trajectory-Planning

The cycle time of operations plays a crucial role in the performance of a robot. While the cycle time is important in the trajectory-tracking task, the precise motion schedule over the path is just as essential. For instance, a welding or soldering robot needs to exactly follow a prescribed path at a given speed; otherwise, the weld could be oversized or weak at different points along the path. This requirement leads the designer to construct a robust robot capable of performing any desired task programmed by the user. Pick-and-place robots, however, are usually designed to meet the given end-poses within a certain time, regardless of the motion schedule. This allows the user to modify the trajectory by, for example, minimizing the kinetic energy of the system and removing the actuator torque peaks by synthesizing the trajectory with smooth blendings [31], to prevent shaking. A second approach is the velocity-management of the trajectory to satisfy the required cycle time, as well as to prevent unnecessary actuator torque peaks. Removing the actuator torque peaks enables the robot to operate faster. In order to distribute a given cycle time properly, velocity scheduling needs to be implemented. The method proposed in this paper focuses on modifying and optimizing any given prescribed trajectory vs. time. These trajectories are analyzed by means of the kinematics and dynamics models of the SMG to find the torque required from each of the four motors. This task can also be conducted on the robot by measuring the motor torques from current readouts. Whether simulating the task or conducting an experiment, the highest torque-demanding points of the trajectory can be identified. After identifying the critical instants in the cycle time, the speed of motion should be lowered around those instants. Instead of changing the desired pose, the time step is manipulated to decrease the speed. By keeping the originally prescribed pose, only the time step is enlarged by means of a time-step scaling function C_t . Consequently, the velocity and acceleration are decreased around

¹The quotation marks indicate the use of the generic term, while recognizing that the peppermill is far from being a “platform” *stricto sensu*

²which plays the role of the moving platform of PKMs

³Hence the moniker

the critical time. In order to better describe the desired relationship, the exponential function is chosen due to its continuous and smooth profile:

$$C_t = 1 + \sum_{i=1}^n A_i e^{-3(t-t_i)^2/B_i^2} \quad (1)$$

where A_i and $2B_i$ are, respectively, the proportional magnitude and the length of the slowed (scheduled) region around the i th critical point, which happens at time t_i , as shown in Fig. 1(a). Since the width of the scheduled region does not have sharp ends, $2B_i$ is defined when C_t settles down to 5% of the A_i value. Moreover, the total number of scheduled regions is assumed to be n within the cycle time T .

As can be seen in Fig. 1(a), element-wise multiplication of the C_t array at sampled instants by the time step array results in a longer cycle time than the original cycle time, since the integral of the scaling function is greater than T . In other words, the extra time spent to slow down the critical parts of the trajectory, which is the area trapped between the C_t curve and the horizontal line $C_t = 1$, results in not meeting the desired cycle time. Therefore, the overall speed of the operation should be increased to compensate the added time due to the lowering of the velocity at certain points. The scaling function is thus modified to:

$$C_t = \left(1 + \sum_{i=1}^n A_i e^{-3(t-t_i)^2/B_i^2}\right) \frac{T}{T + \sum_{i=1}^n \int_0^T A_i e^{-3(t-t_i)^2/B_i^2} dt} \quad (2)$$

where the integrals in the denominator are readily calculated numerically, using Romberg's scheme [32]. As depicted in Fig. 1(b), the modification scales down C_t , making the dark and the light gray areas equal (by making the integral value of the modified C_t equal to T). Once the C_t is properly modified, one can element-wise multiply it by the time step array of the desired trajectory. After conducting the inverse kinematics analysis, the joint-space variables, namely, Ψ , $\dot{\Psi}$, and $\ddot{\Psi}$, are modified accordingly.

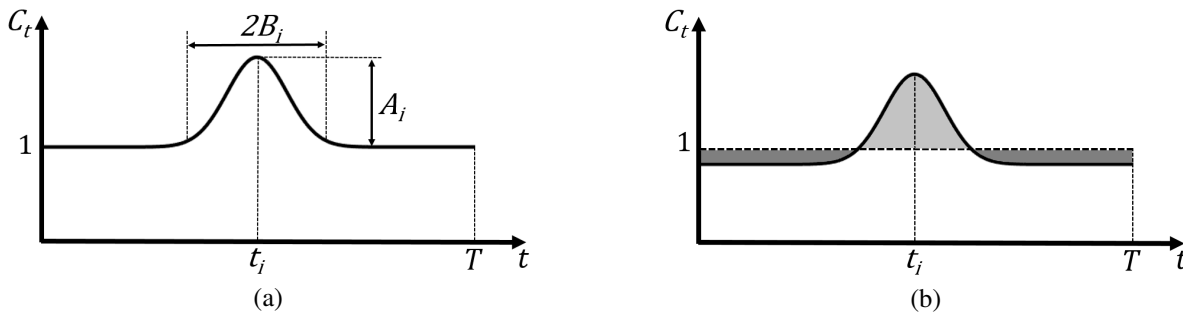


Fig. 1. Time-step scaling function for velocity scheduling: (a) original; (b) modified

One can fully automate the process of velocity-scheduling by determining A_i based on the ratio of the i th peak to the root mean square (RMS) value of the acceleration or torque plot over one cycle of operation. Also, B_i can be assigned based on the width of the peak evaluated at the RMS value.

3 The PMC Modeling

The PMC isostatic architecture shown in Fig. 2(a), a closed CRRHRRC chain⁴, consists of two C-drives, two limbs, and the peppermill [14]. The identical C-drives bear axes located in two horizontal planes, in such a way that the C-drive axes are normal to the Z-axis. The C-drive works based on the synchronized rotation of two nuts, run by two parallel screws, one right-, one left-hand, with the same pitch, as depicted in Fig. 2(b). This synchronization is achieved by means of a belt-pulley transmission, which transfers the rotation and translation of the nuts to a strain-wave-gear (SWG) transmission, a.k.a. Harmonic Drive, mounted on the third parallel axis. The SWG transmission offers high, compact, and self-locking gear reduction. These features render the SWG-driven C-drive (for brevity, the C-drive) robust to changes of the external load.

⁴R, H and C stand for revolute, helical (screw) and cylindrical joint, correspondingly, underlined characters denoting actuated joints

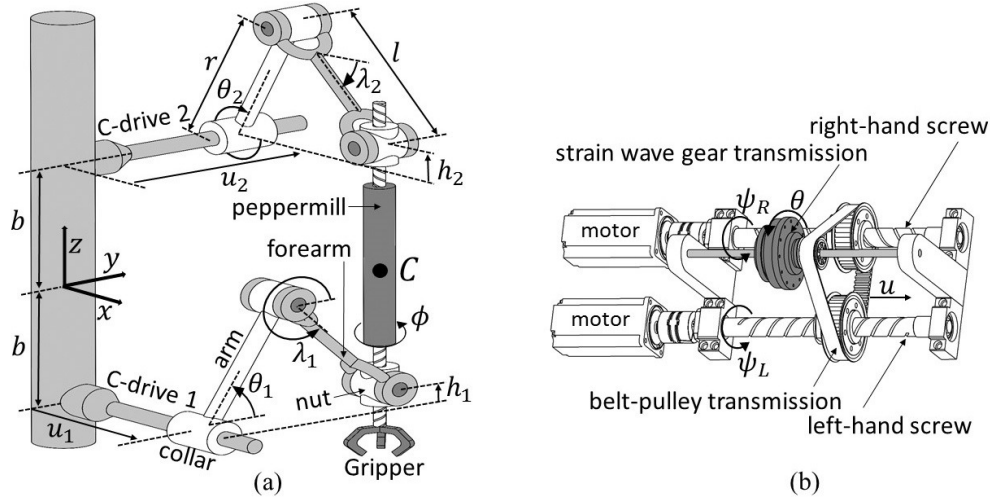


Fig. 2. PMC schematic view: (a) CRRHRRRC architecture; (b) SWG-driven C-drive

The kinematic model of the PMC is established based on the geometric relations between the desired four-dof trajectory, $\mathbf{x} \equiv [x_C \ y_C \ z_C \ \phi]^T$, and the angular displacement of the four motors implemented in the two C-drives; these displacements are grouped in the four-dimensional array $\boldsymbol{\psi}$. The formulation is developed in two steps: first, finding the pose of the C-drive output (arm) in terms of \mathbf{x} ; second, using the 4×4 Jacobian matrix \mathbf{J}_C of the two C-drives to map the collar pose into $\boldsymbol{\psi}$, namely,

$$\boldsymbol{\psi} = \mathbf{J}_C \mathbf{d} \quad (3a)$$

with

$$\boldsymbol{\psi} \equiv \begin{bmatrix} \psi_{1L} \\ \psi_{1R} \\ \psi_{2L} \\ \psi_{2R} \end{bmatrix}, \quad \mathbf{J}_C \equiv \frac{1}{p} \begin{bmatrix} 2\pi & pG & 0 & 0 \\ -2\pi & pG & 0 & 0 \\ 0 & 0 & 2\pi & pG \\ 0 & 0 & -2\pi & pG \end{bmatrix}, \quad \mathbf{d} \equiv \begin{bmatrix} u_1 \\ \theta_1 \\ u_2 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} x_C \\ \alpha_1 + \beta_1 \\ y_C \\ 2\pi - \alpha_2 - \beta_2 \end{bmatrix} \quad (3b)$$

where

$$\alpha_1 = \arctan\left(\frac{h_1}{y_C}\right), \quad \alpha_2 = \arctan\left(\frac{h_2}{x_C}\right), \quad \beta_i = \arccos\left(\frac{r^2 - l^2 + k_i^2}{2rk_i}\right) \quad (3c)$$

$$k_1 = \sqrt{y_C^2 + h_1^2}, \quad k_2 = \sqrt{x_C^2 + h_2^2}, \quad h_i = z_C + (-1)^i \frac{q\phi}{2\pi}, \quad i = 1, 2 \quad (3d)$$

in which, for $i = 1$ and 2 , ψ_{iR} and ψ_{iL} are the angular displacements of the right-hand and left-hand screws of the i th C-drive, respectively, while u_i and θ_i , arrayed in vector \mathbf{d} , denote the translation and the rotation of the i th C-drive output (arm). The pitch of the right- and left-hand screws implemented in the C-drive are p and $-p$, respectively, while the pitch of the right- and left-hand screws in the peppermill are q and $-q$. Furthermore, G is the total gear-reduction ratio due to the belt-pulley transmission and the SWG. Also, r and l are the length of the arms and forearms, respectively.

The mathematical model of the PMC is established based on the concept of the natural orthogonal complement (NOC) [33] in order to find the motor torques $\boldsymbol{\tau}$ based on the external load and the prescribed trajectory. By means of the NOC, the dynamics model becomes a reduced system of four equations in four unknowns. In total, the PMC has 11 rigid bodies: four screws (two for each C-drive), two arms, two forearms, two nuts, and the peppermill. The mathematical model of interest is derived as:

$$\boldsymbol{\tau} = \mathbf{I}\dot{\boldsymbol{\Psi}} + \mathbf{C}\boldsymbol{\Psi} - \boldsymbol{\gamma} - \boldsymbol{\eta} - \boldsymbol{\delta} \quad (4a)$$

with the definitions

$$\boldsymbol{\tau} \equiv [\tau_{1L} \ \tau_{1R} \ \tau_{2L} \ \tau_{2R}]^T, \mathbf{I} \equiv \sum_{i=1}^{11} \mathbf{T}_i^T \mathbf{M}_i \mathbf{T}_i, \mathbf{C} \equiv \sum_{i=1}^{11} (\mathbf{T}_i^T \mathbf{M}_i \dot{\mathbf{T}}_i + \mathbf{T}_i^T \mathbf{W}_i \mathbf{M}_i \mathbf{T}_i) \quad (4b)$$

$$\boldsymbol{\gamma} + \boldsymbol{\eta} + \boldsymbol{\delta} \equiv \sum_{i=1}^{11} \mathbf{T}_i^T (\mathbf{w}_i^G + \mathbf{w}_i^E + \mathbf{w}_i^D), \mathbf{M}_i \equiv \begin{bmatrix} \mathbf{I}_i & \mathbf{O} \\ \mathbf{O} & m_i \mathbf{1} \end{bmatrix}, \mathbf{W}_i \equiv \begin{bmatrix} \boldsymbol{\Omega}_i & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{bmatrix}, \mathbf{T}_i \equiv \frac{\partial \mathbf{t}_i}{\partial \boldsymbol{\Psi}} \quad (4c)$$

where $\mathbf{1}$ and \mathbf{O} denote the 3×3 identity and zero matrices, respectively. The *angular-velocity dyad* \mathbf{W}_i of the i th body is defined based on the angular-velocity matrix $\boldsymbol{\Omega}_i$ of the i th body, the cross-product matrix (CPM)⁵ of the angular-velocity vector $\boldsymbol{\omega}_i$. Furthermore, \mathbf{I}_i and \mathbf{M}_i are the inertia tensor and the *inertia dyad* at the c.o.m. of the i th body, respectively, while \mathbf{T}_i is the twist-shaping matrix of the i th body, which maps the vector array of the motors speeds, $\boldsymbol{\Psi}$, into the twist vector \mathbf{t}_i of the i th body. Moreover, \mathbf{w}_i^G , \mathbf{w}_i^E and \mathbf{w}_i^D are the gravitational, external and dissipative wrenches exerted on the i th body, respectively; correspondingly, $\boldsymbol{\gamma}$, $\boldsymbol{\eta}$ and $\boldsymbol{\delta}$ are the generalized-force vectors, stemming from the gravity, external loads, and dissipation. The twist \mathbf{t}_i of and the wrench \mathbf{w}_i exerted on the i th body are evaluated at the body c.o.m.; these are defined below:

$$\mathbf{t}_i \equiv \begin{bmatrix} \boldsymbol{\omega}_i \\ \mathbf{v}_i \end{bmatrix}, \quad \mathbf{w}_i \equiv \begin{bmatrix} \mathbf{n}_i \\ \mathbf{f}_i \end{bmatrix} \quad (5)$$

where \mathbf{v}_i and $\boldsymbol{\omega}_i$ represent the velocity of the i th body c.o.m, and its angular velocity, respectively, while \mathbf{n}_i and \mathbf{f}_i are the moment and the force acting on the i th body, the latter at the body c.o.m..

4 The PMC Trajectory-Planning

One of the well-known benchmark tasks for SMGs accepted by the robotics community is the Adept test cycle [34]. This cycle is described as: 1) a 25-mm upward vertical translation; 2) a 300-mm horizontal translation with a simultaneous 180° clockwise rotation; 3) a 25-mm downward vertical translation; 4) and following the same path in reverse. The time for completing the industrial test cycle is used to rank pick-and-place robots in terms of speed. Currently, the record is three cycles per second, held by Adept's Quattro s650H. Similarly, the PMC is intended to perform the industrial test cycle. Since the torque peaks along the trajectory beyond the maximum torque affordable by the motors deteriorate the performance of the PMC, trajectory-planning by means of velocity scheduling is necessary. In order to schedule the MP velocity, the first step is the identification of the critical points. Extensive examination of the kinematics and dynamics models reveals that the torque peaks are located exactly at the same place where the maximum joint-space acceleration occurs. Therefore, the four critical points, listed in Table 1, are determined based on the peaks of the joint acceleration, shown in Fig. 3. Since the PMC has a symmetric architecture, the two motors of each C-drive have similar behavior to the other C-drive motors; therefore, only the acceleration of two motors is plotted in Fig. 3. The time-step scaling function is constructed based on the four acceleration peaks, plotted in Fig. 4. One can see that regions with C_t greater than unity are slowed down compared to the original trajectory, and vice versa.

An efficient method of velocity scheduling reduces the acceleration peaks by 35%, as illustrated in Fig. 5(a) by the dashed lines, compared to the original acceleration. Consequently, the motor torques are affected by velocity-scheduling, and the peaks are smoothed by 30%. One should remember that the cycle time is still the same and the PPO takes place at the same frequency.

⁵The CPM of $\boldsymbol{\omega}_i$ is defined as the partial derivative of $\boldsymbol{\omega}_i \times \mathbf{y}$ with respect to \mathbf{y} , $\forall \mathbf{y} \in \mathbb{R}^3$.

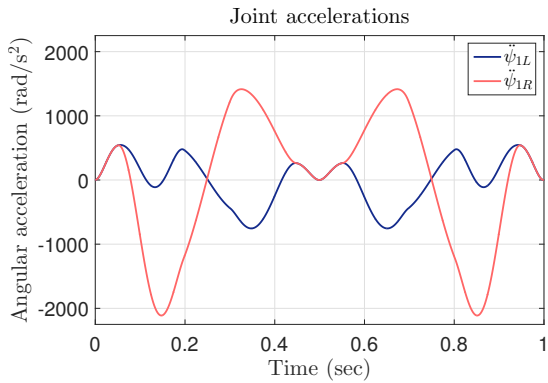


Fig. 3. Angular acceleration of the C-drive motors at 1 Hz

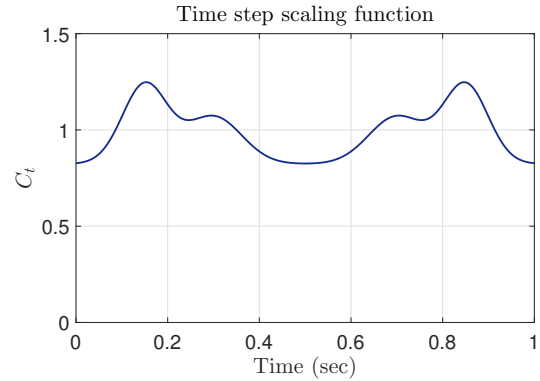
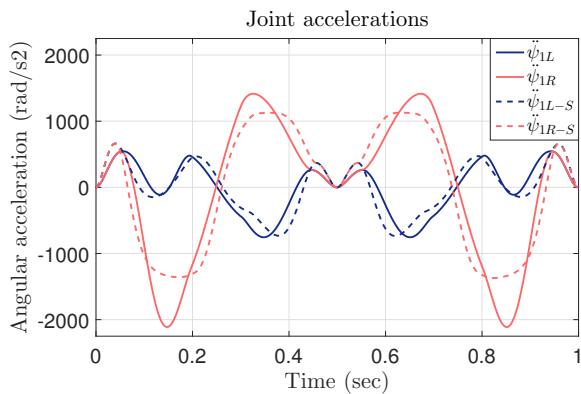


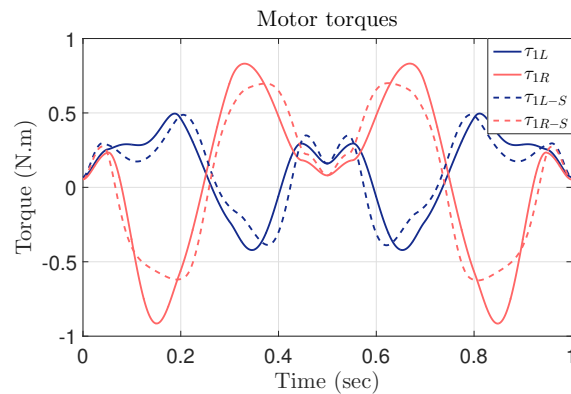
Fig. 4. Time-step scaling function for the PMC velocity scheduling at 1 Hz

Table 1. The PMC critical points on the test cycle at 1 Hz

t_i/T	A_i	B_i
0.15	0.5	0.15
0.31	0.3	0.12
0.69	0.3	0.12
0.85	0.5	0.15



(a)



(b)

Fig. 5. Velocity scheduling effect on the PMC motors: (a) acceleration; (b) torque

5 LQR Controller

As explained by Friedlaender [35], one way to control the PMC is to track the translational and angular positions of the C-drives directly. Indeed, the sensors required by feedback control are located at the actuators. Given that \mathbf{J}_C is dimensionally inhomogeneous— u_1 and u_2 being translations, while θ_1 and θ_2 are rotations—one has to redefine it to obtain a homogeneous Jacobian matrix. Furthermore, because of the pitch p of the C-drives, \mathbf{J}_C is also ill-conditioned. To obtain a new well-conditioned homogeneous matrix, the states are normalized with respect to the cycle frequency f , the pitch of the C-drive p , and the gear-reduction ratio G . Therefore, instead of $\mathbf{d} = \mathbf{J}_C^{-1}\boldsymbol{\psi}$, one has the new variables:

$$\boldsymbol{\sigma}_p = \begin{bmatrix} 2\pi f u_1 \\ G p f \theta_1 \\ 2\pi f u_2 \\ G p f \theta_2 \end{bmatrix} \quad (6)$$

Thus, upon combining Eqns.(3b) and (6), one obtains

$$\boldsymbol{\sigma}_p = \mathbf{J}_p \boldsymbol{\Psi} = \frac{pf}{2} \begin{bmatrix} 1 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \Psi_{1L} \\ \Psi_{1R} \\ \Psi_{2L} \\ \Psi_{2R} \end{bmatrix} \quad (7)$$

in which \mathbf{J}_p is dimensionally homogeneous, with unit of m/s, its condition number, based on the weighted matrix Frobenius norm, being unity. Thereafter, in order to improve the performance of the controller, state variables proportional to the time derivative of $\boldsymbol{\sigma}_p$ need to be added. To keep the same units, these four additional state variables are divided by $2\pi f$. The state vector obtained is

$$\boldsymbol{\sigma} = [\boldsymbol{\sigma}_p^T \boldsymbol{\sigma}_v^T]^T \quad (8a)$$

with

$$\boldsymbol{\sigma}_v = \mathbf{J}_v \dot{\boldsymbol{\Psi}} = \frac{p}{4\pi} \begin{bmatrix} 1 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \dot{\Psi}_{1L} \\ \dot{\Psi}_{1R} \\ \dot{\Psi}_{2L} \\ \dot{\Psi}_{2R} \end{bmatrix} \quad (8b)$$

By combining Eqns.(7–8b), one now has:

$$\boldsymbol{\sigma} = \begin{bmatrix} \mathbf{J}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_v \end{bmatrix} \begin{bmatrix} \boldsymbol{\Psi} \\ \dot{\boldsymbol{\Psi}} \end{bmatrix} \quad (9)$$

To simplify the control scheme, the C-drives are controlled, rather than the whole PMC. Therefore, two identical independent LQR controllers are used. As detailed by Friedlaender [35], the mathematical model of the C-drives is, if Coulomb friction is neglected,

$$\mathbf{M}\ddot{\boldsymbol{\Psi}} + \mathbf{D}\dot{\boldsymbol{\Psi}} = \boldsymbol{\tau} \quad (10)$$

where \mathbf{M} and \mathbf{D} are, respectively, the generalized-inertia and the damping matrices. To obtain a state-space representation, the above equation is rewritten as

$$\begin{bmatrix} \dot{\boldsymbol{\Psi}} \\ \ddot{\boldsymbol{\Psi}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{1} \\ \mathbf{0} & -\mathbf{M}^{-1}\mathbf{D} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Psi} \\ \dot{\boldsymbol{\Psi}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1} \end{bmatrix} \boldsymbol{\tau} \quad (11)$$

Using Eqns.(7–11), one obtains

$$\dot{\boldsymbol{\sigma}} = \mathbf{A}\boldsymbol{\sigma} + \mathbf{B}\boldsymbol{\tau} \quad (12a)$$

with

$$\mathbf{A} = \begin{bmatrix} \mathbf{J}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_v \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{1} \\ \mathbf{0} & -\mathbf{M}^{-1}\mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{J}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_v \end{bmatrix}^{-1}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{J}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_v \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1} \end{bmatrix} \quad (12b)$$

and $\boldsymbol{\tau}$ denoting the control vector; one can now compute the trajectory-tracking error, i.e., $\mathbf{e} = \boldsymbol{\sigma} - \boldsymbol{\sigma}_d$ where $\boldsymbol{\sigma}_d$ is the desired trajectory. Indeed, the time-derivative of the error vector is

$$\dot{\mathbf{e}} = \mathbf{A}\mathbf{e} + \mathbf{B}\mathbf{v} \quad (13)$$

where $\mathbf{v} = \boldsymbol{\tau} - \boldsymbol{\tau}_d$ and $\boldsymbol{\tau}_d$ is the desired torque vector obtained from inverse dynamics. With a system in the form of Eqn.(13), with state variables equal to zero at $T = 0$, the gains of a LQR controller can be found by minimizing the quadratic cost function

$$J = \int (\mathbf{e}^T \mathbf{Q} \mathbf{e} + \mathbf{v}^T \mathbf{R} \mathbf{v}) dt \quad (14)$$

where \mathbf{Q} and \mathbf{R} are, respectively, the state and the control weighting matrices. The former is a 8×8 matrix, the latter a 4×4 matrix, both being positive definite. Given a state feedback control law defined as $\mathbf{v} = -\mathbf{K} \mathbf{e}$ that minimizes the performance index J of Eqn.(14), matrix \mathbf{K} is given by [17]

$$\mathbf{K} = \mathbf{R}^{-1} (\mathbf{B}^T \mathbf{P} + \mathbf{N}^T) \quad (15)$$

In the above equation, \mathbf{P} is found by solving the algebraic Riccati equation (ARE)

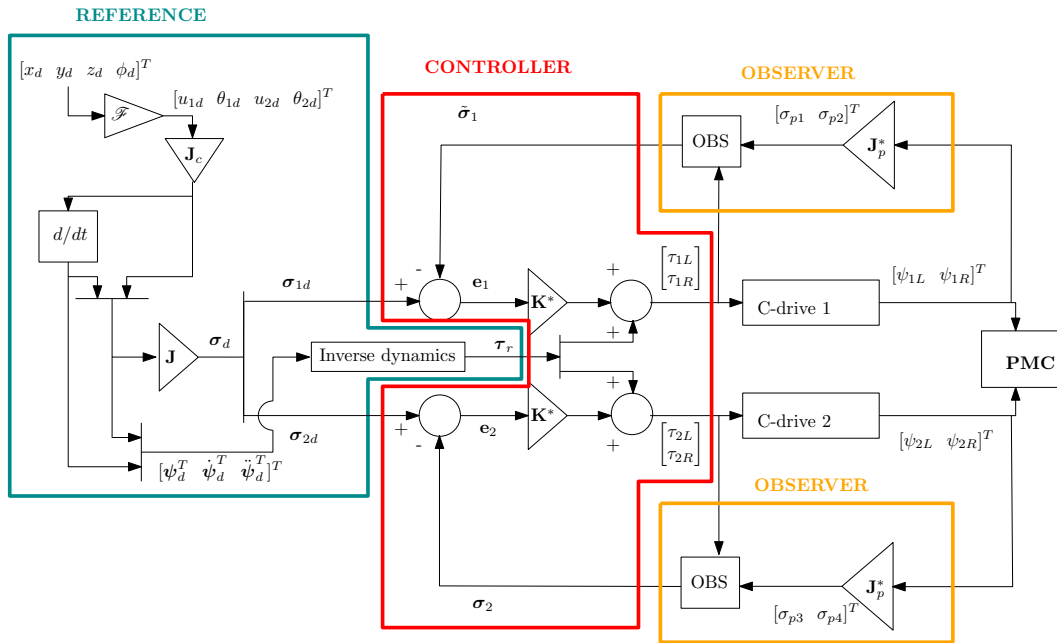


Fig. 6. Control Scheme with LQR; A minimum-order observer is used to estimate $\boldsymbol{\sigma}_v$.

$$\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} - (\mathbf{P} \mathbf{B} + \mathbf{N}) \mathbf{R}^{-1} (\mathbf{B}^T \mathbf{P} + \mathbf{N}^T) + \mathbf{Q} = \mathbf{0} \quad (16)$$

which is solved using Matlab's continuous-time algebraic Riccati equation solver (CARE). Knowing \mathbf{P} , \mathbf{K} is known from Eqn.(15), the complete control law now being readily available:

$$\boldsymbol{\tau} = \mathbf{K} (\boldsymbol{\sigma}_d - \boldsymbol{\sigma}) + \boldsymbol{\tau}_d \quad (17)$$

The first term of the above equation is the feedback part containing the linear-quadratic regulator, the second the feed-forward reference torque computed with the inverse dynamics model. The complete control scheme is illustrated in Fig. 6. In this figure, the superscript * refers to the part of the associated variable relative to only one C-drive, i.e.,

$$\mathbf{J}^* = \frac{p}{2} \begin{bmatrix} f-f & 0 & 0 \\ f & f & 0 & 0 \\ 0 & 0 & 1/(2\pi) & -1/(2\pi) \\ 0 & 0 & 1/(2\pi) & 1/(2\pi) \end{bmatrix} \quad (18)$$

In order to obtain the variable Ψ , an observer is necessary to avoid numerical differentiation, which would introduce noise. Many solutions exist; here the minimum-order observer is chosen, as it generally converges more quickly than its full-order equivalent [36]. The latter would also estimate Ψ with a lower accuracy, as it can be directly measured at the actuators. The observer used is Luenberger's [37]. First, the state space matrices \mathbf{A} and \mathbf{B} of the system are recalled:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{A}_{12} \\ \mathbf{0} & \mathbf{A}_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_v \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{1} \\ \mathbf{0} & -\mathbf{M}^{-1}\mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{J}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_v \end{bmatrix}^{-1}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{0} \\ \mathbf{B}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{J}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_v \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1} \end{bmatrix} \quad (19)$$

The minimum-order observer is defined by

$$\mathbf{\kappa} \equiv \boldsymbol{\sigma}_v - \mathbf{L}\boldsymbol{\sigma}_p \Leftrightarrow \tilde{\mathbf{\kappa}} \equiv \tilde{\boldsymbol{\sigma}}_v - \mathbf{L}\boldsymbol{\sigma}_p \quad (20a)$$

$$\dot{\tilde{\mathbf{\kappa}}} = \mathbf{U}\tilde{\mathbf{\kappa}} + \mathbf{V}\boldsymbol{\sigma}_p + \mathbf{Z}\boldsymbol{\tau} \quad (20b)$$

where $\tilde{\mathbf{\kappa}}$ and $\tilde{\boldsymbol{\sigma}}_v$ are, respectively, estimations of $\mathbf{\kappa}$ and $\boldsymbol{\sigma}_v$. The matrix \mathbf{L} is made of the observer feedback gains and the matrices \mathbf{U} , \mathbf{V} , and \mathbf{Z} are defined as

$$\mathbf{U} = \mathbf{A}_{22} - \mathbf{L}\mathbf{A}_{12}, \quad \mathbf{V} = \mathbf{U}\mathbf{L}, \quad \mathbf{Z} = \mathbf{B}_2 \quad (21)$$

The estimation of the state vector is then

$$\tilde{\boldsymbol{\sigma}} = \boldsymbol{\Phi}\tilde{\mathbf{\kappa}} + \boldsymbol{\Lambda}\boldsymbol{\sigma}_p \quad (22)$$

with

$$\tilde{\boldsymbol{\sigma}} = [\boldsymbol{\sigma}_p^T \tilde{\boldsymbol{\sigma}}_v^T]^T, \quad \boldsymbol{\Phi} = [\mathbf{0} \ \mathbf{1}]^T, \quad \boldsymbol{\Lambda} = [\mathbf{1} \ \mathbf{L}^T]^T \quad (23)$$

The gains of the observer can be obtained with a pole-placement algorithm, such as the one included in Matlab's Control Systems Toolbox. The observer's diagram is shown in Fig. 7.

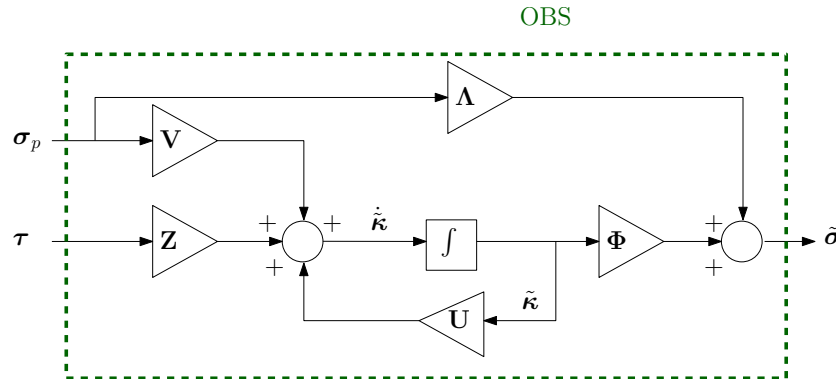


Fig. 7. Minimum-order observer's diagram.

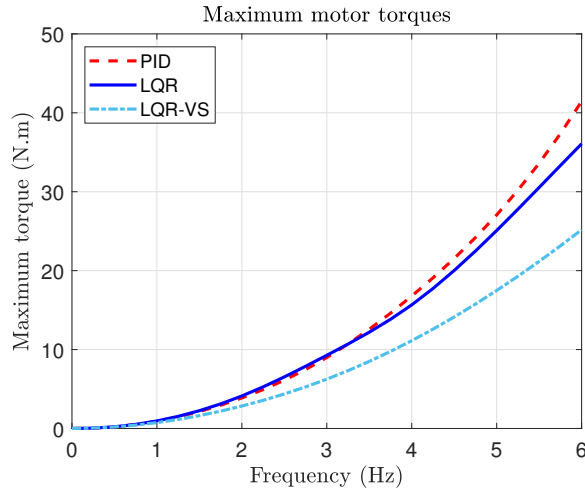


Fig. 8. Maximum torque of the actuators along the trajectory (VS stands for velocity scheduling)

5.1 Simulation Results

With the forward dynamics simulation model of the PMC, the LQR can be compared to the standard PID controller used in the previous sections. As can be seen in Fig. 8, when the frequency is increased, the LQR control scheme proves to be more adequate with respect to the maximum torque that must be provided by the actuators. The combined effect of the LQR scheme and the velocity scheduling significantly reduces the maximum torque required.

Furthermore, if one considers the trajectory-tracking error, the choice seems even more obvious, as it remains relatively small with the LQR control scheme, but it increases rapidly as the cycle time is reduced, as shown in Fig. 9, at 4 Hz. Thus, the PID controller is an appropriate choice at frequencies around 1 Hz and lower, but, to achieve higher speeds, other control strategies, such as LQR, should be investigated. The tracking error is further reduced when the scheduled trajectory is used.

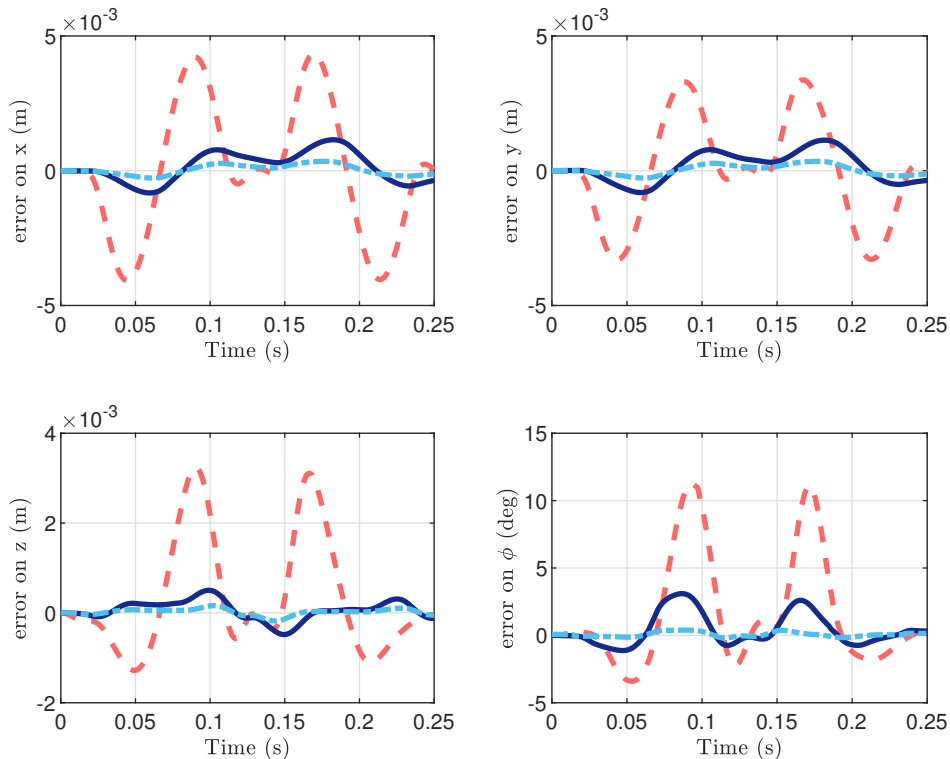


Fig. 9. Trajectory-tracking error at 4 Hz (PID: dashed curve, LQR: solid curve, LQR with velocity scheduling: dash-dot curve)

6 Conclusions

The authors introduced a novel method of trajectory-planning to properly schedule the velocity of the MP of a parallel SMG along the industry-adopted test cycle, while keeping the path and cycle time constant. The effectiveness of the proposed method was shown on the PMC, a parallel two-limb SMG, performing the above-mentioned test cycle. The reported kinematics and dynamics models of the PMC were used to find the critical points and demonstrate the smoothed torque peaks due to trajectory-planning.

Moreover, it was shown with simulation that a control strategy based on the linear quadratic regulator can help achieve better trajectory tracking while also reducing the maximum torque needed at high velocity. Furthermore, by means of the normalized homogeneous state variables, the isotropic Jacobian matrix of the C-drive was obtained. Together, velocity scheduling and trajectory-tracking LQR control made it possible to improve the speed and accuracy of the PMC.

Acknowledgements

We gratefully acknowledge the support of NSERC (Canada's Natural Sciences and Engineering Research Council) through grant No. 4532-2010, the FRQNT (*Fonds de recherche du Québec - Nature et technologies*) Postdoctoral Research Scholarship, the James McGill Professorship of Mechanical Engineering, and the McGill Engineering Doctoral Award.

References

- [1] Hervé, J. M., 1999. "The Lie group of rigid body displacements, a fundamental tool for mechanism design". *Mech. Mach. Theory*, **34**(5), pp. 719–730.
- [2] Brogårdh, T., 2001. Device for relative movement of two elements. U.S. Patent No. 4,741,207.
- [3] Angeles, J., and Morozov, A., 2006. Four-degree-of-freedom parallel manipulator for producing Schönflies motions. U.S. Patent No. 7,127,962.
- [4] Gallardo-Alvarado, J., Abedinnasab, M. H., and Lichtblau, D., 2016. "Simplified Kinematics for a Parallel Manipulator Generator of the Schönflies Motion". *J. Mech. Robot.*, **8**(6), p. 061020.
- [5] Corbel, D., Gouttefarde, M., Company, O., and Pierrot, F., 2010. "Actuation Redundancy as a Way to Improve the Acceleration Capabilities of 3T and 3T1R Pick-and-Place Parallel Manipulators". *J. Mech. Robot.*, **2**(4), p. 041002.
- [6] Altuzarra, O., Zubizarreta, A., Cabanes, I., and Pinto, C., 2009. "Dynamics of a four degrees-of-freedom parallel manipulator with parallelogram joints". *Mechatronics*, **19**(8), pp. 1269–1279.
- [7] Huang, T., Bai, P., Mei, J., and Chetwynd, D. G., 2016. "Tolerance Design and Kinematic Calibration of a Four-Degrees-of-Freedom Pick-and-Place Parallel Robot". *J. Mech. Robot.*, **8**(6), oct, p. 061018.
- [8] Clavel, R., 1990. Device for the movement and positioning of an element in space. U.S. Patent No. 4,976,582.
- [9] Pierrot, F., Shibukawa, T., and Morita, K., 2003. Four-degree-of-freedom parallel robot. U.S. Patent No. 6,516,681.
- [10] Brantmark, H., and Hemmingson, E., 2001. "FlexPicker with PickMaster revolutionizes picking operations". *Ind. Robot An Int. J.*, **28**(5), pp. 414–420.
- [11] Lee, C., and Lee, P., 1990. "Isoconstrained mechanisms for fast pick-and-place manipulation". In *Geom. Methods Robot. Mech. Res. Theory Appl.*, Y. Lou and Z. Li, eds. Saarbrücken: LAP Lambert Academic Publishing, pp. 95–112.
- [12] Lee, P.-C., and Lee, J.-J., 2012. "Singularity and workspace analysis of three isoconstrained parallel manipulators with schoenflies motion". *Front. Mech. Eng.*, **7**(2), pp. 163–187.
- [13] Harada, T., and Angeles, J., 2014. "Kinematics and Singularity Analysis of a CRRHRC Parallel Schönflies Motion Generator". *Trans. Can. Soc. Mech. Eng.*, **38**(2), pp. 173–183.
- [14] Karimi Eskandary, P., and Angeles, J., 2018. "The dynamics of a parallel Schönflies-motion generator". *Mech. Mach. Theory*, **119**, pp. 119–129.
- [15] Harada, T., Friedlaender, T., and Angeles, J., 2014. "The development of an innovative two-DOF cylindrical drive: Design, analysis and preliminary tests". In 2014 IEEE Int. Conf. Robot. Autom., IEEE, pp. 6338–6344.
- [16] Karimi Eskandary, P., and Angeles, J., 2018. "The translating Π -joint: Design and applications". *Mech. Mach. Theory*, **122**, pp. 361–370.
- [17] Trentelman, H. L., Stoorvogel, A., and Hautus, M., 2001. *Control theory for linear systems*. Springer.
- [18] Kalman, R. E., 1960. "Contributions to the Theory of Optimal Control". *Bol. la Soc. Mat. Mex.*, **5**(2), pp. 102–119.
- [19] Wang, H., Dong, H., He, L., Shi, Y., and Zhang, Y., 2010. "Design and Simulation of LQR Controller with the Linear Inverted Pendulum". In 2010 Int. Conf. Electr. Control Eng., IEEE, pp. 699–702.
- [20] Divelbiss, A. W., and Wen, J. T., 1997. "Trajectory tracking control of a car-trailer system". *IEEE Trans. Control Syst. Technol.*, **5**(3), pp. 269–278.
- [21] Bourbonnais, F., Bigras, P., and Bonev, I. A., 2015. "Minimum-Time Trajectory Planning and Control of a Pick-and-Place Five-Bar Parallel Robot". *IEEE/ASME Trans. Mechatronics*, **20**(2), pp. 740–749.
- [22] Liu, H., Lai, X., and Wu, W., 2013. "Time-optimal and jerk-continuous trajectory planning for robot manipulators with kinematic constraints". *Robot. Comput. Integr. Manuf.*, **29**(2), apr, pp. 309–317.

-
- [23] Perumaal, S. S., and Jawahar, N., 2013. "Automated Trajectory Planner of Industrial Robot for Pick-and-Place Task". *Int. J. Adv. Robot. Syst.*, **10**(2), p. 100.
- [24] Gasparetto, A., Lanzutti, A., Vidoni, R., and Zanotto, V., 2011. "Validation of Minimum Time-Jerk Algorithms for Trajectory Planning of Industrial Robots". *J. Mech. Robot.*, **3**(3), p. 031003.
- [25] Menon, M. S., Ravi, V. C., and Ghosal, A., 2017. "Trajectory Planning and Obstacle Avoidance for Hyper-Redundant Serial Robots". *J. Mech. Robot.*, **9**(4), p. 041010.
- [26] Boscariol, P., and Gasparetto, A., 2013. "Model-based trajectory planning for flexible-link mechanisms with bounded jerk". *Robot. Comput. Integr. Manuf.*, **29**(4), pp. 90–99.
- [27] Azizi, M. R., and Naderi, D., 2013. "Dynamic modeling and trajectory planning for a mobile spherical robot with a 3Dof inner mechanism". *Mech. Mach. Theory*, **64**, pp. 251–261.
- [28] Zhang, N., and Shang, W., 2016. "Dynamic trajectory planning of a 3-DOF under-constrained cable-driven parallel robot". *Mech. Mach. Theory*, **98**, pp. 21–35.
- [29] Gosselin, C., and Foucault, S., 2014. "Dynamic Point-to-Point Trajectory Planning of a Two-DOF Cable-Suspended Parallel Robot". *IEEE Trans. Robot.*, **30**(3), pp. 728–736.
- [30] Jiang, X., and Gosselin, C., 2016. "Dynamic Point-to-Point Trajectory Planning of a Three-DOF Cable-Suspended Parallel Robot". *IEEE Trans. Robot.*, **32**(6), dec, pp. 1550–1557.
- [31] Gauthier, J. F., Angeles, J., and Nogleby, S., 2008. "Optimization of a Test Trajectory for SCARA Systems". In *Adv. Robot Kinemat. Anal. Des.* Springer Netherlands, Dordrecht, pp. 225–234.
- [32] Kutt, H. R., 1975. "The numerical evaluation of principal value integrals by finite-part integration". *Numer. Math.*, **24**(3), pp. 205–210.
- [33] Saha, S. K., and Angeles, J., 1991. "Dynamics of Nonholonomic Mechanical Systems Using a Natural Orthogonal Complement". *J. Appl. Mech.*, **58**(1), p. 238.
- [34] Nabat, V., de la O Rodriguez, M., Company, O., Krut, S., and Pierrot, F., 2005. "Par4: very high speed parallel robot for pick-and-place". In 2005 IEEE/RSJ Int. Conf. Intell. Robot. Syst., IEEE, pp. 553–558.
- [35] Friedlaender, T., 2015. "Design, Control and Testing of a Pick-and-Place Robot and its Novel Actuators". PhD thesis, McGill University.
- [36] Kailath, T., 1980. *Linear systems*. Prentice-Hall.
- [37] Bryson, A., and Luenberger, D., 1970. "The synthesis of regulator logic using state-variable concepts". *Proc. IEEE*, **58**(11), pp. 1803–1811.

List of Tables

1 The PMC critical points on the test cycle at 1 Hz 6

List of Figures

1	Time-step scaling function for velocity scheduling: (a) original; (b) modified	3
2	PMC schematic view: (a) CRRHRRRC architecture; (b) SWG-driven C-drive	4
3	Angular acceleration of the C-drive motors at 1 Hz	6
4	Time-step scaling function for the PMC velocity scheduling at 1 Hz	6
5	Velocity scheduling effect on the PMC motors: (a) acceleration; (b) torque	6
6	Control Scheme with LQR; A minimum-order observer is used to estimate σ_v	8
7	Minimum-order observer's diagram.	9
8	Maximum torque of the actuators along the trajectory (VS stands for velocity scheduling)	10
9	Trajectory-tracking error at 4 Hz (PID: dashed curve, LQR: solid curve, LQR with velocity scheduling: dash-dot curve)	10