CONTROL OF SYSTEMS BY PARALLEL ACTUATORS

STEPHEN MONTGOMERY-SMITH

Abstract. We describe a particular control method for a system controlled by several actuators with the same control constants. We show under certain assumptions that the control constants for the whole system can be obtained immediately from the control constants for a single actuator. This greatly simplifies the work in finding the control constants. Also, no gain scheduling is required. The authors have been unable to find any prior work in this direction, and so believe this is a rather new approach.

1. Introduction

In this paper, we describe a system whose state, which we will call the *end effector position*, is given by an element of a differential manifold $\eta \in \mathcal{M}$. We suppose that the end effector position is determined by the *values* of n actuators:

$$\boldsymbol{\ell} = (\ell_1, \ell_2, \dots, \ell_n) \in \mathbb{R}^n. \tag{1}$$

We write $L : \mathcal{M} \to \mathbb{R}^n$ for the function which maps the end effector position to the actuator values. Note that we can allow the problem to be over-constrained, that is, n can be bigger or equal to the dimension of $\mathcal{M}$. Note that the elements of $\mathcal{M}$ are generalized positions in the sense of the Euler-Lagrange formalism [1].

Examples of these are cable-driven parallel robots, consisting of a fixed rigid frame, and a floating rigid body called the end effector. The end effector is manipulated via eight cables attached to actuators. Each actuator is clamped to the fixed frame. The value, ℓ_k, of the kth actuator is the length of cable issued by the actuator. The manifold $\mathcal{M}$ is the six dimensional space of poses, that is, positions with orientations. See [2, 3, 10] for an introduction to parallel robots. See [8, 9] for information about cable-driven parallel robots. The algorithm described in this paper was tested upon a cable-driven parallel robot built by the NASA Johnson Space Center, which we describe in another paper [7].

We assume that the only measurements we can take are the values of the actuators, and that the only method of control is to command a force at each actuator, which converts into a force on the end effector. We assume that once the end effector force is known, one can determine the trajectory of the end effector via a frictionless, quadratic Hamiltonian system. The force is a generalized force obeying the principle of virtual work, and we will call it the *end effector force.*

The method of control is to calculate the end effector position using the actuator values. Then, from the difference of the actual end effector position from the requested end effector position, is calculated the required acceleration of the end effector. From this is calculated the required end effector force. Finally, we find the actuator forces to effect this.

Our main contribution is to show that the control constants for computing the required acceleration of the end effector are the same as the control constants used to control a single actuator, making it easier to determine the control method.

The benefit of this approach was demonstrated when developing the cable-driven parallel robot that is to be described in [7]. The control constants were calculated having access to and performing trials upon only one actuator. When the controller was tried on the full parallel robot, it worked the first time. If we had been forced to find the control constants by trial and error for the whole system, this would have been very difficult.

This result requires two things. First that the actuator response is linear. This would be violated if, for example, if there is significant non-linear friction. Second, it requires that the each actuator by itself can be controlled by the same linear controller. For example, with the cable-driven parallel robot, if there is significant flexing or stretching of cables, this would require a different control method for each actuator because the orientations and/or lengths of the cables aren't necessarily identical.

Since the control constants can be found by considering a controller for a single actuator, then the same control constants can be used irrespective of the end effector position, that is, no gain scheduling is required. Thus the formula for resonant frequencies can also be calculated from the resonant frequencies of a single actuator.

Key words and phrases. Parallel actuator, linear control theory, differential manifold, tangent bundle, cotangent bundle, wrench set.

We believe the methods and results in Sections 5 and 6, where we show that the control constants come from those described in Section 4, are new. For this reason, we know of no prior work in this direction, and thus we have no references. But it is possible that these methods were developed in a completely different context, and that the author is simply unaware of them. (Early versions of this paper were submitted to other control theory journals, and they replied that this paper was out of their purview, suggesting to the author that no-one else had considered this approach.)

2. A brief primer on manifolds, tangent bundles, and cotangent bundles

While there is a well known abstract definition of a differential manifold (see, for example [6]), for our purposes we will consider a *manifold of dimension* m to be the inverse image of zero under an infinitely-differentiable function $F : \mathbb{R}^p \to \mathbb{R}^{m-p}$, whose Jacobian matrix is full rank in a neighborhood of $\mathcal{M} = F^{-1}(0)$.

Three examples of manifolds that are typically used in robotics are the manifold of positions $\mathbb{R}^3$, the manifold of orientations SO_3, and the manifold of poses SE_3, which by identifying the set of three by three matrices with $\mathbb{R}^9$, and the set of symmetric three by three matrices with $\mathbb{R}^6$, are respectively created by

$$\begin{aligned} &F_1 : \mathbb{R}^3 \to \mathbb{R}^0, \\ &F_1(x) = 0, \\ &\mathbb{R}^3 = F_1^{-1}(0); \end{aligned} \tag{2}$$

$$\begin{aligned} &F_2 : \mathbb{R}^9 \to \mathbb{R}^7, \\ &F_2(R) = (R^T R - I, \det(R) - 1); \\ &SO_3 = F_2^{-1}(0); \end{aligned} \tag{3}$$

$$\begin{aligned} &F_3 : \mathbb{R}^{12} \to \mathbb{R}^7, \\ &F_3(R, x) = (F_2(R), F_1(x)), \\ &SE_3 = F_3^{-1}(0). \end{aligned} \tag{4}$$

To each point $\eta \in \mathcal{M}$ is associated its *tangent space* $T_\eta\mathcal{M}$, that is, the m-dimensional subspace of $\mathbb{R}^p$ which is tangent to the manifold at η. Thus $T_\eta\mathcal{M}$ can be thought of as the set of end effector velocities. We call the disjoint union of all tangent spaces the *tangent bundle* $T\mathcal{M} = \bigcup_{\eta\in\mathcal{M}} T_\eta\mathcal{M}$.

Also associated to each point $\eta \in \mathcal{M}$ is the dual space to the tangent space, which is called the *cotangent space*, and is denoted $T^*_\eta\mathcal{M}$. Thus $T^*_\eta\mathcal{M}$ can be thought of as the set of end effector forces that can be applied to the end effector, with the duality defined as mapping an end effector velocity and an end effector to their inner product, the rate of change of work done by the end effector. The disjoint union of the cotangent spaces is called the *cotangent bundle* $T\mathcal{M} = \bigcup_{\eta\in\mathcal{M}} T_\eta\mathcal{M}$.

In the first example $\mathbb{R}^3 = F_1^{-1}(0)$, the tangent space is the set of translational velocities, and the cotangent space is the set of forces. In the second example $SO_3 = F_2^{-1}(0)$, the tangent space can be identified with the set of angular velocities, and the cotangent space with the set of torques. In the third example $SE_3 = F_3^{-1}(0)$, the tangent space can be identified with the set of twists, and the cotangent space with the set of wrenches.

3. Mathematical description of the system

Given the position of the end effector that is a function of time, $\eta(t)$, we have its velocity and acceleration which are functions of time taking their values in $T\mathcal{M}$ given by

$$\varphi = \dot\eta, \quad \alpha = \dot\varphi. \tag{5}$$

We define the linear operator $\Lambda_\eta : T_\eta\mathcal{M} \to \mathbb{R}^n$ by the directional derivative, which in local coordinates is

$$\Lambda_\eta\theta = \theta \cdot \frac{dL(\eta)}{d\eta}. \tag{6}$$

Often we simply write Λ for Λ_η when there is no confusion.

Then, from the velocity φ, we can calculate the rate of change of the actuator values:

$$\dot{\boldsymbol{\ell}} = \Lambda_\eta\varphi. \tag{7}$$

We suppose that there is a linear operator $T = T_\eta : \mathbb{R}^n \to T^*\mathcal{M}$, which converts actuator forces $\boldsymbol{f} = (f_1, f_2, \dots, f_n)$ to the end effector force τ:

$$\tau = T_\eta\boldsymbol{f}. \tag{8}$$

By the principle of virtual work, we have

$$\varphi \cdot T_\eta\boldsymbol{f} = \dot{\boldsymbol{\ell}} \cdot \boldsymbol{f} = \Lambda_\eta\varphi \cdot \boldsymbol{f}. \tag{9}$$

Hence

$$T_\eta = \Lambda_\eta^T. \tag{10}$$

Next we describe the equations of motion. Suppose that the system is given by a Lagrangian

$$l(\eta, \varphi) = \tfrac{1}{2} M_\eta(\varphi, \varphi) - v(\eta), \tag{11}$$

where $M_\eta = M$ is a positive definite bilinear operator on $T_\eta\mathcal{M}$. This includes kinetic energy of the actuators

$$\tfrac{1}{2} m_0 |\dot{\boldsymbol{\ell}}|^2, \tag{12}$$

where m_0 is the effective mass of the each actuator (the notion of 'effective' is explained in Section 4 below). In local coordinates we can describe Λ and M as matrices, then the kinetic energy of the actuators is given by

$$\tfrac{1}{2} m_0 \varphi^T \Lambda^T \Lambda \varphi, \tag{13}$$

and so we must have that the matrix

$$M - m_0 \Lambda^T \Lambda \tag{14}$$

is positive semi-definite.

Solving the Euler-Lagrange equations [1], we obtain the equations of motion

$$\tau = M\alpha + \mu(\eta), \tag{15}$$

where in local coordinates

$$\mu(\eta) = \varphi \cdot \frac{\partial M_\eta}{\partial \eta}(\varphi, \cdot) - \frac{\partial}{\partial \eta} v(\eta). \tag{16}$$

We define the *no-load forces* to be the actuator forces if the actuators are not attached to the system, that is, the Lagrangian is given simply by equation (12):

$$\boldsymbol{f}_0 = m_0 \ddot{\boldsymbol{\ell}}. \tag{17}$$

Thus differentiating equation (7), we obtain (in local coordinates)

$$\boldsymbol{f}_0 = m_0 \Lambda_\eta \alpha + m_0 \varphi \cdot \frac{d\Lambda_\eta}{d\eta} \varphi. \tag{18}$$

For the example of the cable-driven parallel robot, the cable tensions are given by

$$\text{cable tensions} = \boldsymbol{f}_0 - \boldsymbol{f}. \tag{19}$$

Next, we need inverse functions to L and T, which we call Y and F. We define the *set of admissible actuator values*, $\mathcal{L} \subset \mathbb{R}^n$, to be the range of the function L. We suppose that we have a *forward kinematics* function, $Y : \mathcal{L} \to \mathcal{M}$, which is a left inverse to L. Because of possible measurement errors, Y should produce decent answers even if the actuator values are merely close to $\mathcal{L}$. For example, this could be implemented using the Newton-Raphson Method.

For the inverse function of T, we need some more definitions. Given $\boldsymbol{f}_b, \boldsymbol{f}_0 \in \mathbb{R}^n$, we suppose that we have a predefined set $\mathcal{C}_{\boldsymbol{f}_b,\boldsymbol{f}_0} \subset \mathbb{R}^n$. Here $\boldsymbol{f}_b$ is the command force required to overcome actuator resistance such as back-EMF, $\boldsymbol{f}_0$ is the no-load actuator forces, and $\mathcal{C}_{\boldsymbol{f}_b,\boldsymbol{f}_0}$ is the set of those $\boldsymbol{f}$ such that it is permissible to command forces $\boldsymbol{f} + \boldsymbol{f}_b$ to the actuators.

Typically this is a convex set defined by a finite number of linear constraints. For the example of cable-driven parallel robots, we might say $\boldsymbol{f} \in \mathcal{C}_{\boldsymbol{f}_b,\boldsymbol{f}_0}$ if and only if the tensions in the cables, $\boldsymbol{f}_0 - \boldsymbol{f}$, is never below a given predefined value, and the command forces $\pm(\boldsymbol{f} + \boldsymbol{f}_b)$ don't exceed the actuator hardware limits.

Then we define the *wrench set* to be the set of achievable forces:

$$\mathcal{W} = \{(\tau, \boldsymbol{f}_b, \boldsymbol{f}_0) \in T^*\mathcal{M} \times \mathbb{R}^n \times \mathbb{R}^n : \exists \boldsymbol{f} \in \mathcal{C}_{\boldsymbol{f}_b,\boldsymbol{f}_0} \text{ such that } T\boldsymbol{f} = \tau\}. \tag{20}$$

We suppose that we have a function $F : \mathcal{W} \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ that provides a right inverse to the map defined by T in the following manner:

$$T(F(\tau, \boldsymbol{f}_b, \boldsymbol{f}_0)) = \tau, \tag{21}$$

such that

$$F(\tau, \boldsymbol{f}_b, \boldsymbol{f}_0) \in \mathcal{C}_{\boldsymbol{f}_b,\boldsymbol{f}_0}. \tag{22}$$

Because there are more actuators than the number of degrees of freedom (that is, the dimension of $\mathcal{M}$, computing the actuator forces in the last step is an over-constrained problem. For the example of cable-driven cable robots, there are many approaches in the literature [4, 5, 8].

Finally we need a way to approximate the difference between two end effector positions by an element of the tangent space. That is, there is a function $\triangle : \mathcal{M} \times \mathcal{M} \to T\mathcal{M}$ such that $\triangle(\eta_1, \eta_2)$ is in $T_{\eta_1}\mathcal{M}$, so that with respect to a 'reasonable' coordinate system about η_1 that

$$\eta_2 \approx \eta_1 + \triangle(\eta_1, \eta_2). \tag{23}$$

For example, if $\mathcal{M}$ is a Riemannian manifold, we could define it as the direction of a geodesic from η_1 to η_2. If $\mathcal{M}$ is a Lie group, we could define it as the direction of a one parameter subgroup from η_1 to η_2. With the example of SO_3, the latter can be identified with angle of rotation from one to the other, multiplied by the unit vector in the direction of the axis of this rotation.

4. Control of a single actuator

Let ℓ denote actuator value, let f_c be the command force given to a single actuator, and f be the actual force supplied by this actuator.

Each actuator has an *effective* no-load mass m_0, which is the ratio $f/\ddot{\ell}$ when there is no load placed upon the actuator.

For the remainder of this section, we suppose that the actuator is carrying a passive load. We denote by m to be the effective mass of the actuator with this load. Thus no load corresponds to $m = m_0$, and we always have $m \geq m_0$. If the actuator is clamped so that it cannot move, this corresponds to $m = \infty$, which is a mathematical idealization representing when the passive load is very large.

For the purpose of making the analogue of these equations and the system controller equations clearer, we shall replace the actuator actual and command forces by actual and command accelerations

$$a = \frac{f}{m} = \ddot{\ell}, \tag{24}$$

$$a_c = \frac{f_c}{m}. \tag{25}$$

We look for a controller such that, given an actuator value ℓ_r, attempts to create a command acceleration a_c such that the actual actuator value, ℓ, is close to ℓ_r.

First we describe an *open-loop* controller:

$$f_c = m\ddot{\ell}_r + k_0\dot{\ell}_r, \tag{26}$$

or

$$a_c = \ddot{\ell}_r + \frac{k_0}{m}\dot{\ell}_r. \tag{27}$$

We call k_0 the back-EMF constant, since for electric motors this is a likely source of this term. This controller fails badly if there is any drift or noise in the system, because it makes no attempt to correct for error.

Next, suppose we also have a good *homogeneous closed-loop* controller. Denote the vector containing all time derivatives of order less than l of ℓ by

$$\boldsymbol{\ell} = [\ell, \dot{\ell}, \ddot{\ell}, \ldots, \ell^{(l-1)}]^T. \tag{28}$$

The controller is defined by appropriately sized constant matrices A, B, C, and D as:

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\ell \tag{29}$$

$$a_c = C\boldsymbol{x} + D\boldsymbol{\ell}. \tag{30}$$

By a good homogeneous closed-loop controller, we mean that if this is used to control the passively loaded actuator, then ℓ converges to 0 in a manner that is expeditious enough for our application.

An example is a PID controller

$$a_c = -(k_i{\textstyle\int}\ell + k_p\ell + k_d\dot{\ell}), \tag{31}$$

But it could be something more complex, such as a cascaded controller, or a linear quadratic Gaussian controller.

Note that if cable stretching or sagging plays a significant role in the cable-driven parallel robot, then this should be able to control a single actuator with a cable with similar stretching or sagging characteristics attached.

The results of this paper don't depend upon what definition of 'good' we use. Our assertion is that the parallel actuator driven robot controller behaves as well as the single actuator controller. Thus if the user knows the level of precision or stability required for the whole system, they merely have to check these same parameters for the single actuator.

We combine the open-loop and homogeneous closed-loop controller to obtain a good *closed-loop feed-forward* controller, that is, given a requested actuator value ℓ_r, and its vector of derivatives

$$\boldsymbol{\ell}_r = [\ell_r, \dot{\ell}_r, \ddot{\ell}_r, \ldots, \ell_r^{(l-1)}]^T, \tag{32}$$

we find the command acceleration a_c such that ℓ converges to ℓ_r in an expeditious manner. This can be created by applying the homogeneous closed-loop controller to

$$\ell_d = \ell - \ell_r \tag{33}$$

$$\boldsymbol{\ell}_d = \boldsymbol{\ell} - \boldsymbol{\ell}_r \tag{34}$$

to obtain

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\ell_d \tag{35}$$

$$a_c = \ddot{\ell}_r + \frac{k_0}{m}\dot{\ell}_r + C\boldsymbol{x} + D\boldsymbol{\ell}_d. \tag{36}$$

For example, with the PID controller it is

$$a_c = \ddot{\ell}_r + \frac{k_0}{m}\dot{\ell}_r - (k_i{\textstyle\int}\ell_d + k_p\ell_d + k_d\dot{\ell}_d). \tag{37}$$

5. The controller for the system by parallel actuators

For the controller, we introduce the state vector $\boldsymbol{\xi}$, which is a vector of elements from the tangent space, with the same number of components as the state vector $\boldsymbol{x}$ described in equation (29). We denote a matrix multiplied by a vector of elements from the tangent space as giving another vector of elements of the tangent space as follows:

$$(A\boldsymbol{\xi})_i := \sum_j A_{i,j}\xi_j, \tag{38}$$

where ξ_j means the jth component of $\boldsymbol{\xi}$.

The control loop is shown as a block diagram in Figure 1. It is labeled throughout with superscript numbers that correspond to the steps given below.

(1) Obtain the requested end effector position η_r.
(2) Compute the requested acceleration

$$\alpha_r = \ddot{\eta}_r. \tag{39}$$

(3) Measure actuator values $\boldsymbol{\ell}$.
(4) Calculate the actual end effector position:

$$\eta = Y(\boldsymbol{\ell}). \tag{40}$$

(5) Find the $\triangle$ difference between the actual end effector position and the requested end effector position:

$$\theta_d = \triangle(\eta, \eta_r) \tag{41}$$

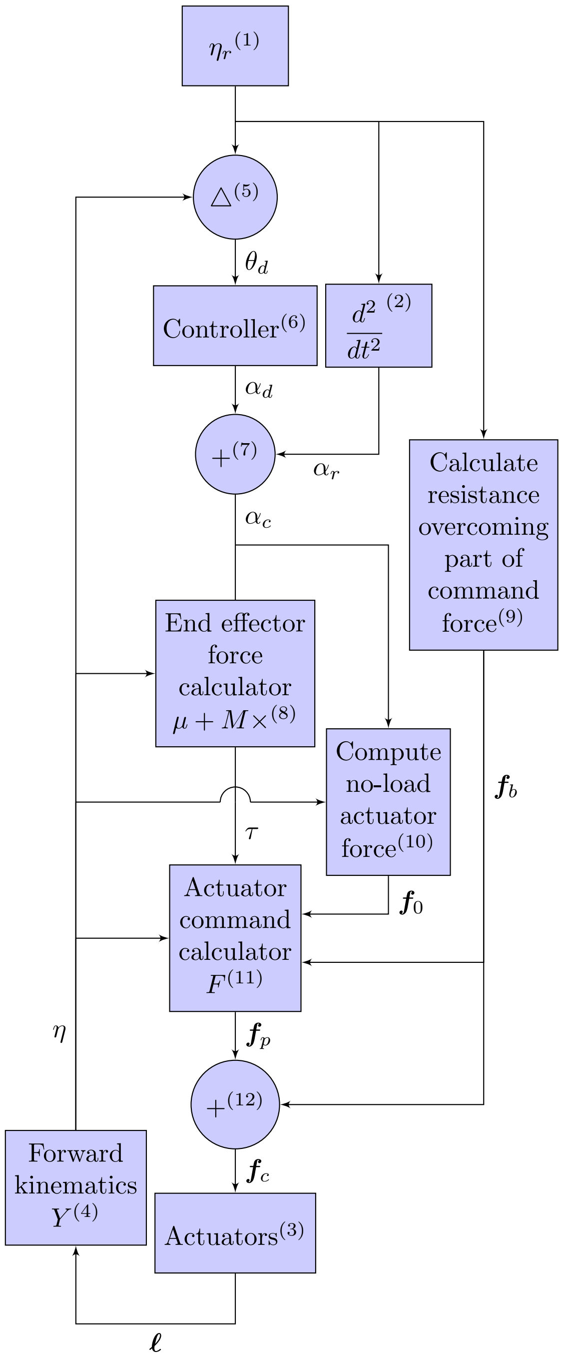

FIGURE 1. Block diagram of the controller for the robot driven by parallel actuators.

and compute the vector of derivatives

$$\boldsymbol{\theta}_d = [\theta_d, \dot{\theta}_d, \ddot{\theta}_d, \ldots, \theta_d^{(l-1)}]^T. \tag{42}$$

(6) Calculate the control part of the command end effector acceleration:

$$\dot{\boldsymbol{\xi}} = A\boldsymbol{\xi} + B\theta_d \tag{43}$$

$$\alpha_d = C\boldsymbol{\xi} + D\boldsymbol{\theta}_d. \tag{44}$$

For example, the PID controller would be:

$$\alpha_d = -(k_i \textstyle\int \theta_d + k_p\theta_d + k_d\dot{\theta}_d). \tag{45}$$

(7) Calculate the command end effector acceleration:

$$\alpha_c = \alpha_r + \alpha_d. \tag{46}$$

(8) Determine the command end effector force to be applied to the end effector:

$$\tau_c = \boldsymbol{\mu} + M\alpha_c. \tag{47}$$

(9) Calculate the requested rate of change of the lengths of the cables

$$\dot{\boldsymbol{\ell}}_r = \Lambda(\eta_r)\varphi_r, \tag{48}$$

and find the resistance overcoming part of the command force for the actuators

$$\boldsymbol{f}_b = k_0\dot{\boldsymbol{\ell}}_r. \tag{49}$$

(10) Use equation (18) to calculate the no-load actuator forces.

$$\boldsymbol{f}_0 = m_0\Lambda_\eta\alpha_c + m_0\varphi \cdot \frac{d\Lambda_\eta}{d\eta}\varphi. \tag{50}$$

(11) Determine whether $(\tau_c, \boldsymbol{f}_b, \boldsymbol{f}_0) \in \mathcal{W}$. If it isn't, declare that the end effector is out of its workspace, command the actuators to brake, and quit. Otherwise, calculate part of the actuator forces using

$$\boldsymbol{f}_p = F(\tau_c, \boldsymbol{f}_b, \boldsymbol{f}_0). \tag{51}$$

(12) Command the actuators with

$$\boldsymbol{f}_c = \boldsymbol{f}_p + \boldsymbol{f}_b. \tag{52}$$

(13) Go back to Step 1.

Note that in Steps 8, 10, and 11, the quantities M, $\boldsymbol{\mu}$, Λ, $\mathcal{W}$, and F are calculated from η and φ, but they could just as well be calculated from η_r and φ_r, the only change required being to equation (41) to

$$\theta_d = -\triangle(\eta_r, \eta) \tag{53}$$

6. THEORETICAL JUSTIFICATION FOR THE CONTROLLER

Assumption 1. *The interaction between the command force f_c, the actual force f, and the actuator* *value ℓ of a single actuator, is given by the linear system*

$$f + c_2\dot{f} + \cdots + c_n f^{(n-1)} + k_0\dot{\ell} + k_1\ddot{\ell} + \cdots + k_n\ell^{(n+1)} = f_c + \tilde{c}_2\dot{f}_c + \cdots + \tilde{c}_p f_c^{(p-1)}. \quad (54)$$

Note this linearity can be difficult to achieve if friction is significant in the actuators. Friction is highly non-linear, especially when $\dot{\ell}$ and f switch between having the same sign and having different signs, as could happen with an active load.

If the actuator has a passive load m as described in Section 4, then equation (54) becomes

$$\ddot{\ell} + c_2\ell^{(3)} + \cdots + c_n\ell^{(n+1)} + \tfrac{k_0}{m}\dot{\ell} + \tfrac{k_1}{m}\ddot{\ell} + \cdots + \tfrac{k_n}{m}\ell^{(n+1)} = a_c + \tilde{c}_2\dot{a}_c + \cdots + \tilde{c}_p a_c^{(p-1)}. \quad (55)$$

The original open-loop controller was derived from the assumption that equation (54) is a perfect description of the passively loaded actuator:

$$f_c + \tilde{c}_2\dot{f}_c + \cdots + \tilde{c}_p f_c^{(p-1)} = m(\ddot{\ell}_r + c_2\ell_r^{(3)} + \cdots + c_n\ell_r^{(n+1)}) + k_0\dot{\ell}_r + k_1\ddot{\ell}_r + \cdots + k_n\ell_r^{(n+1)}, \quad (56)$$

which in Section 4 was approximated with equation (26).

Assumption 2. *For every $m \in [m_0, \infty]$, equations* (33), (34), (35) *and* (36) *provide a good closed-loop feed-forward controller for system* (55).

This assumption can either be tested theoretically using eigenvalue analysis, or experimentally by loading various passive loads onto a single actuator. The latter approach doesn't require any knowledge of the coefficients in equation (54). We simply need to believe that such an equation exists.

Assumption 3. *The time scale of the corrections θ_d is much smaller than the time change of η_r, and the magnitude of θ_d and its derivatives are much smaller than that of η and its corresponding derivatives. This means we can assume that the time derivatives of η are negligible compared to η, and hence we can assume the matrices M, F, and the covector μ, and their derivatives, are constant in the time scales in which the controller operates. We also assume that equation* (23) *holds for time derivatives of both sides, and that the constants are uniformly controlled in the ranges achieved.*

The main result of this paper, which we state below, is described as an 'assertion' rather than a 'theorem,' as the proofs are not very rigorous.

Assertion 1. *Given Assumptions 1, 2, and 3, the algorithm described in Section 5 is a good controller for the parallel actuator driven robot.*

Pick a time t_0 which is in the range of times in which the controller performs the required corrections. Let $\eta_0 = \eta(t_0)$. Define

$$\theta = \triangle(\eta, \eta_0) \quad (57)$$

$$\theta_r = \triangle(\eta_r, \eta_0). \quad (58)$$

Thus

$$\eta \approx \eta_0 + \theta \quad (59)$$

$$\eta_r \approx \eta_0 + \theta_r, \quad (60)$$

and if we set

$$\varphi_r = \dot{\eta} \quad (61)$$

then

$$\varphi \approx \dot{\theta} \quad (62)$$

$$\varphi_r \approx \dot{\theta}_r, \quad (63)$$

and we have

$$\theta_d \approx \theta - \theta_r. \quad (64)$$

Let $n_{\mathcal{M}}$ be the dimension of $\mathcal{M}$, and define the $(n_{\mathcal{M}} \times n_{\mathcal{M}})$ matrix

$$N = M^{-1}\Lambda^T\Lambda. \quad (65)$$

Assertion 2. *With the same assumptions as Assertion 1, if the end effector is controlled by the algorithm given in Section 5, then we have*

$$\ddot{\theta} + c_2\theta^{(3)} + \cdots + c_n\theta^{(n+1)} + N(k_0\dot{\theta} + k_1\ddot{\theta} + \cdots + k_n\theta^{(n+1)}) \approx \alpha_c + \tilde{c}_2\dot{\alpha}_c + \cdots + \tilde{c}_p\alpha_c^{(p-1)} \quad (66)$$

where

$$\alpha_c \approx \hat{\alpha}_c + k_0 N\dot{\theta}_r \quad (67)$$

and $\hat{\alpha}_c$ is calculated thus:

$$\boldsymbol{\theta}_d = [\theta_d, \dot{\theta}_d, \ddot{\theta}_d, \ldots, \theta_d^{(l-1)}]^T \quad (68)$$

$$\dot{\boldsymbol{\xi}} = A\boldsymbol{\xi} + B\theta_d \quad (69)$$

$$\hat{\alpha}_c = \alpha_r + C\boldsymbol{\xi} + D\boldsymbol{\theta}_d. \quad (70)$$

Proof: Rewrite equation (54) for the jth actuator:

$$f_j + c_2\dot{f}_j + \cdots + c_n f_j^{(n-1)} + k_0\dot{\ell}_j + k_1\ddot{\ell}_j + \cdots + k_n\ell_j^{(n+1)} = f_{c,j} + \tilde{c}_2\dot{f}_{c,j} + \cdots + \tilde{c}_p f_{c,j}^{(p-1)}. \quad (71)$$

From equations (7), (49), (51), (52), and (63), we obtain

$$f_{c,j} = F_j(\tau_c, \bar{\boldsymbol{f}}_c) + k_0\Lambda\dot{\theta}, \quad (72)$$

and applying $T = \Lambda^T$ we obtain

$$T\boldsymbol{f}_c = \tau_c + k_0 \Lambda^T \Lambda \dot{\theta}_r. \tag{73}$$

Similarly, from equations (7) and (62), we have

$$T\boldsymbol{f} = \tau, \tag{74}$$

where τ is the actual end effector force, and

$$T\dot{\boldsymbol{\ell}} = \Lambda^T \Lambda \dot{\theta}. \tag{75}$$

Hence from Assumption 3, we obtain

$$\begin{aligned} &\tau + c_2\dot{\tau} + \cdots + c_n \tau^{(n-1)} \\ &\quad + \Lambda^T\Lambda(k_0\dot{\theta} + k_1\ddot{\theta} + \cdots + k_n\theta^{(n+1)}) \\ &\approx \tau_c + k_0\Lambda^T\Lambda\dot{\theta}_r + \tilde{c}_2(\dot{\tau}_c + k_0\Lambda^T\Lambda\ddot{\theta}_r) + \\ &\quad \cdots + \tilde{c}_p(\tau_c^{(p-1)} + k_0\Lambda^T\Lambda\theta_r^{(p)}). \end{aligned} \tag{76}$$

Next, subtracting μ, and then left multiplying by M^{-1}, and using Assumption 3 again, we obtain equation (66). The rest of the assertion follows by equation (64). Q.E.D.

Lemma 1. *The matrix N has a basis of eigenvectors, with eigenvalues in* $[0, m_0^{-1}]$.

Proof: Let

$$\tilde{N} = M^{-1/2}\Lambda^T\Lambda M^{-1/2}. \tag{77}$$

Clearly $\tilde{N}$ is symmetric and positive semi-definite, and since $M - m_0\Lambda^T\Lambda$ is positive definite, we have

$$m_0^{-1}I - \tilde{N} = m_0^{-1}M^{-1/2}(M - m_0\Lambda^T\Lambda)M^{-1/2} \tag{78}$$

is positive definite. (Here I denotes the $(n_{\mathcal{M}} \times n_{\mathcal{M}})$ identity matrix.) Hence $\tilde{N}$ has a basis of eigenvectors, with eigenvalues in $[0, m_0^{-1}]$. Also,

$$N = M^{-1/2}\tilde{N}M^{1/2}, \tag{79}$$

and hence N and $\tilde{N}$ are similar matrices. Q.E.D.

Proof of Assertion 1: We use Lemma 1 to obtain σ_1, $\sigma_2, \ldots, \sigma_{n_{\mathcal{M}}}$ a basis of eigenvectors of N, with corresponding eigenvalues m_1^{-1}, $m_2^{-1}, \ldots, m_{n_{\mathcal{M}}}^{-1}$, where m_1, $m_2, \ldots, m_{n_{\mathcal{M}}} \in [m_0, \infty]$. We also form a dual basis π_1, $\pi_2, \ldots, \pi_{n_{\mathcal{M}}}$ that satisfies

$$\sigma_i \cdot \pi_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j, \end{cases} \tag{80}$$

so that for any $\psi \in \mathbb{R}^{n_{\mathcal{M}}}$ we have

$$\psi = \sum_{i=1}^{n_{\mathcal{M}}} (\psi \cdot \pi_i)\sigma_i \tag{81}$$

$$\pi_i \cdot N\beta = m_i^{-1}(\pi_i \cdot \beta). \tag{82}$$

By Assumption 3, we can assume that all these vectors and eigenvalues are constant. For each $1 \leq i \leq n_{\mathcal{M}}$, dot product the equations in Assertion 2 by π_i. Then it may be seen that the resulting equations satisfy the hypotheses of Assumption 2, with y replaced by $\theta \cdot \pi_i$, with y_r replaced by $\theta_r \cdot \pi_i$, and with m replaced by m_i. Thus $\theta \cdot \pi_i$ is well controlled by $\theta_r \cdot \pi_i$.

Then it follows by equation (81) that θ is well controlled by θ_r. Therefore by equations (59) and (60), we have that η is well controlled by η_r. Q.E.D.

7. Conclusions

We have shown, under certain assumptions, that the control constants for a system of several parallel actuators can be derived simply from knowledge of the control constants of a single actuator. The benefit of this is that finding the control constants for a single actuator is much easier than finding the control constants for the whole system. Also, we can show that no gain scheduling is required, that is, the control constants do not require adjusting depending upon where the end effector lies.

The assumptions are (1) that each actuator has a linear response to commands, (2) that the time and distance scales of the control changes are much smaller than the respective scales of the requested position, and (3) that the control constants work well no matter what passive load is placed upon each single actuator. Also, (4) there is also the implicit assumption that the same control constants work for each of the actuators.

The first assumption seems to be very important, in particular, our limited experience is that this approach does not work well if the effect of friction is large. The fourth assumption means that this approach can be flawed if there are other non-proportional links in the mechanism between the actuator and end effector, for example, if there is a stretchable cable between them, and the lengths of the cables are different for different actuators.

Even with these limitations, we do believe that this approach is definitely of theoretical interest. And perhaps future work can overcome some of these difficulties.

References

[1] V.I. Arnold, Mathematical Methods of Classical Mechanics, 2nd Ed, Springer-Verlag, 1989.

[2] Jaime Gallardo-Alvarado, Kinematic Analysis of Parallel Manipulators by Algebraic Screw Theory, Springer, Switzerland, 2016.

[3] Grigore Gogu, Structural Synthesis of Parallel Robots, Springer, The Netherlands, 2008.

[4] Nicholas I. M. Gould, Philippe L. Toint, (2000). A Quadratic Programming Bibliography (PDF). RAL Numerical Analysis Group Internal Report 2000-1.

[5] Marc Gouttefarde, Johann Lamaury, Christopher Reichert, and Tobias Bruckmann, A Versatile Tension Distribution Algorithm for n-DOF Parallel Robots Driven by $n+2$ Cables, IEEE Transactions on Robotics, Vol. 31, no. 6, 2015.

[6] John Lee, Introduction to Smooth Manifolds. Graduate Texts in Mathematics. 218 (Second ed.). New York London: Springer-Verlag. ISBN 978-1-4419-9981-8. OCLC 808682771, 2012.

[7] Stephen Montgomery-Smith, Cecil Shy, Joshua Sooknanan, Derek Bankieris, Tension control of Cable-Driven Parallel Robots, in preparation.

[8] Andreas Pott, Cable-Driven Parallel Robots: Theory and Application, Springer, 2018.

[9] Sen Qian, Bin Zi, Wei-Wei Shang and Qing-Song Xu, A Review on Cable-driven Parallel Robots, Chinese Journal of Mechanical Engineering volume 31, Article number: 66 (2018).

[10] Hamid D. Taghirad, Parallel Robots: Mechanics and Control, CRC Press, 2013.

Department of Mathematics, University of Missouri, Columbia, MO 65211, U.S.A.