The International Journal of
Robotics Research
32(2) 247–258
© The Author(s) 2013
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/0278364912468302
ijr.sagepub.com

Decoupled torque control of tendon-driven fingers with tension management

Muhammad E Abdallah[1], Robert Platt Jr[2] and Charles W Wampler[1]

Abstract

To facilitate human assembly tasks, Robonaut 2 is equipped with a dexterous, compact hand featuring fingers driven remotely by tendons. This work outlines the force-control strategy for the fingers, which are actuated by an “$n+1$” tendon arrangement. Existing tendon-driven fingers have applied force control through independent tension controllers on each tendon, in other words, in the tendon space. The coupled kinematics of the tendons, however, cause such controllers to exhibit a transient coupling in their response. This problem can be resolved by alternatively framing the controllers in the joint space of the finger. A joint-space torque control law is proposed here that demonstrates a decoupled response with a faster settling time than an equivalent tendon-space formulation. In addition, a tension distribution algorithm is presented here to translate joint torque commands into tendon tensions. It guarantees that each tendon tension respects both an upper and a lower bound, using an efficient, finitely-convergent algorithm. These two contributions provide for a compliant, well-controlled hand, aptly suited for unstructured interaction.

Keywords

Force control, tendon actuation, multifingered hands, manipulation, humanoid robots

1. Introduction

Robonaut 2 (R2) was designed by NASA and GM to support humans in unstructured assembly tasks. This motivated the need for a robotic hand that approximates the human hand in size and dexterity. Like many of the existing robotic hands, R2 implements a tendon-driven actuation within a force control framework. Remote actuation allows for compact design in a high degree-of-freedom (DOF) hand, while the force or torque control provides the compliance necessary for physical interaction. Since the tendons can only transmit forces in tension, the number of actuators must exceed the DOFs to achieve fully-determined control of the finger. It turns out that only one tendon more than the number of DOFs is needed (Murray et al., 1994). If arranged correctly, the $n+1$ tendons can independently control the n DOFs while always maintaining positive tensions. Such an “n+1” arrangement is very attractive due to its minimum number of actuators. Unlike an antagonistic “2n” arrangement, however, an “n+1” arrangement requires a coupled relation between the tendon and joint displacements, which introduces an extra layer of complexity to the control.

Many control strategies exist in the literature for the torque control of coupled, tendon-driven manipulators. These strategies consist of two general components: a method for tension distribution and a control law. The first component, tension distribution, refers to the process of determining a set of tendon tensions that give a desired set of joint torques while resolving the redundancy in the actuation. This redundancy leads to a null-space of tensions that can be used to ensure positive tensions in all tendons. Although the problem can be framed as a linear program (LP), existing systems have avoided conventional optimization techniques. Rather, simpler algorithms and heuristic approximations have been sought.

For the second component, the existing control laws can be categorized based on whether or not they employ tension feedback. Control laws that use tension feedback explicitly regulate the tendon tensions. Neglecting dynamics, they use a tension distribution algorithm (TDA) to translate the desired joint torques into desired tendon tensions and then apply an independent tension regulator for each tendon. We refer to this scheme as a *tendon-space controller*. Although this controller is widely applied, we show here that it

[1] General Motors R&D, Warren, MI, USA
[2] State University of New York, Buffalo, NY, USA

Corresponding author:
Muhammad E. Abdallah, General Motors R&D, Mail Code 480-106-224, 30500 Mound Rd, Warren, MI 48090, USA.
Email: muhammad.abdallah@gm.com

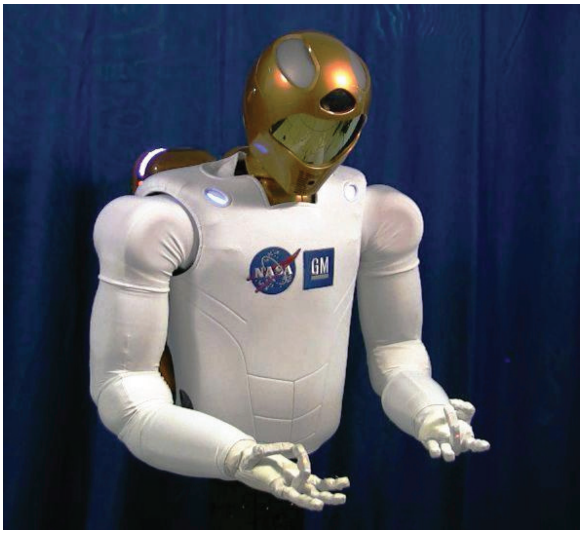

Fig. 1. The torque-controlled fingers of Robonaut 2 (R2).

introduces a transient coupling to the response of the finger. A command or perturbation to one joint can produce an undesired response in another joint. This first-order disturbance arises intrinsically from the kinematics of the tendons, rather than from bandwidth limitations in the controller.

Alternatively, existing control laws without tension feedback apply a dynamic control law, typically based on a computed torque formulation. They employ models of the actuator dynamics with acceleration estimates to predict the tendon tensions. Given the inherent error in such a scheme, tension values can only be roughly controlled and often high null-space tension is applied to compensate. Such higher tensions are undesirable, contributing to tendon fatigue and degrading performance with greater friction. On the plus side, this strategy offers full dynamic compensation and avoids the transient coupling of the tension-feedback methods.

Given the disadvantages of these two approaches, a new control strategy is desired that can both provide independent torque control at the joints and carefully protect the tendons. The tendons are often the weakest link in the system, failing due to a combination of shock and fatigue. A law based on tension feedback is thus desired. The law may neglect the dynamics of the finger, but the transient coupling of the tendon-space controllers is not acceptable.

We thus present a new control strategy for “$n+1$”-type fingers that features both a TDA and a new control law. The TDA improves on existing methods by providing both a user-defined upper bound and a user-defined lower bound for the tensions. It satisfies the range while minimizing the null-space tension and avoiding any nonlinear disturbance to the joint torques. These objectives are achieved with a compact algorithm that is simple, efficient, and fully converges in a (small) finite number of steps.

Our second contribution is a control law that employs tension feedback and decouples the joint torque response. Instead of regulating the tensions in the tendon space, it regulates the torques in the joint space. We thus refer to it as the *joint-space controller*. Not only does this formulation isolate the kinematic coupling, but it also demonstrates a faster response. Although the tendon-space induced coupling has been previously observed in Wimbock et al. (2008), we provide here the theoretical analysis for the behavior. The contrast with the joint-space formulation is validated by both theory and experiment.

The net product of these two components is a finger that offers a high level of tendon management underneath the surface while operating transparently of that actuation. The finger actively yields to excessive external forces and minimizes the tension levels throughout. All the while, it delivers force control without exhibiting any evidence of the underlying coupled actuation. The integrated control strategy is currently implemented on R2, where experiments validating its performance are conducted.

2. Previous work

Several TDAs have appeared in literature. Salisbury and Craig (1982) first presented a simple method for setting the lowest tension equal to zero, and our algorithm builds off their method. Based on the concept of rectifier functions, Jacobsen et al. (1984) present a quasi-analytical solution for antagonistic pairs, which is then extended to multiple DOF systems by Lee and Tsai (1993). This solution is efficient yet heavily system-dependent, requiring considerable analytical derivation for each system. Hirose and Ma (1991) present an algorithm that distorts the joint torques. Gabiccini et al. (2011) apply standard convex optimization techniques, where their system uniquely integrates internal contact forces on grasped objects with the redundant tendon tensions. Besides these methods, others have just elected to apply conservative levels of null-space tension that could hopefully cover all cases. These solutions all provide only for a lower bound, and only Salisbury and Craig (1982) minimize the null-space tension.

Regarding tendon-space control laws, Salisbury and Craig (1982) again were the first to implement such a scheme using the Stanford/JPL hand. Working on the same hand, Starr (1990) implemented a similar scheme but with a model of the tendon-conduit hysteresis added. Other hands to implement such a scheme include the POSTECH and Utah/MIT hands, although the latter has an uncoupled tendon routing (Jacobsen et al., 1989; Lee et al., 1994). Such a scheme has also been applied to larger manipulators, most notably the CT-ARM (Ma et al., 1993) and Ogane’s arm (Ogane et al., 1996).

Lee (1991) was the first to introduce a computed torque-control law with no tension feedback. Kobayashi and Ozawa (2003) and Mulero-Martinez et al. (2006) both build on this framework with adaptive neural networks to learn the model parameters. Wimbock et al. (2008) then bridged the two control law camps with an impedance controller that both

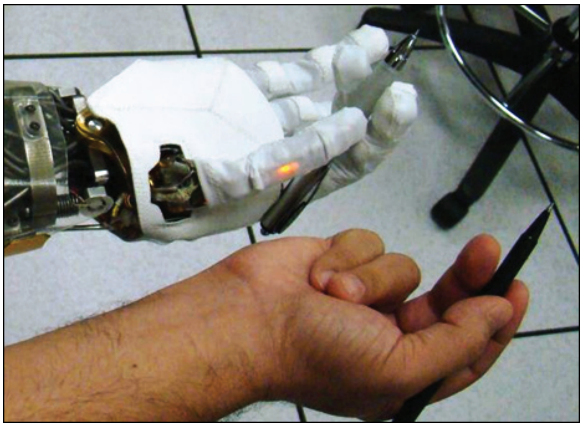

Fig. 2. The R2 hand and fingers approximate the size of a human hand.

utilizes tension feedback and models the actuator dynamics. Their impedance law, however, operates in the tendon space and thus exhibits the same transient coupling of the conventional tendon-space controllers as they themselves note.

The following presentation starts with a review of both the mechanical system as well as the kinematics of tendon-driven manipulators. After this review, the two components of the strategy are presented: the TDA followed by the control law. Finally, the experimental results from R2 are presented.

3. Mechanical system

R2 is an upper-body humanoid robot that consists of two arms, a torso, and a head containing vision equipment. Each arm has seven DOFs with a fully integrated humanoid hand attached. The arms are force-controlled using an implementation of series-elastic actuation. Further discussion of R2 is available in Diftler et al. (2011).

The R2 hand consists of five fingers including an opposable thumb. The fingers are actuated remotely via a cable or tendon transmission, where the actuators are integrated into the forearm of the robot and the tendons pass through the wrist structure. The resulting design achieves a high level of dexterity within an approximately human form-factor, as shown in Figure 2.

The fingers fall into two categories: the dexterous set and the grasping set. The dexterous set consists of the thumb, index and middle fingers. These fingers are fully actuated using an “$n+1$” tendon arrangement. Both the index and middle fingers have three DOFs while the thumb has four. The grasping set, on the other hand, consists of the ring and pinky fingers. These fingers are underactuated and less mobile; they serve the less-demanding role of supporting grasps on larger objects. See Abdallah and Wampler (2011) for a discussion of the torque control of tendon-driven underactuated fingers.

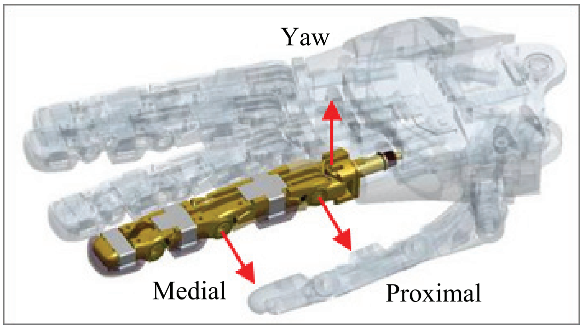

Fig. 3. A model of the R2 index finger. Motion of the distal joint is mechanically linked to the medial joint.

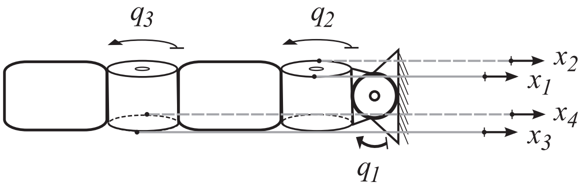

Fig. 4. Kinematic configuration.

This work will focus on the index finger of the hand, a model of which is shown in Figure 3. The finger has four tendons and three independent DOFs: a yaw, a proximal pitch, and a medial pitch. A fourth (distal) pitch joint exists at the end of the finger, but this joint is mechanically linked to the medial joint. The yaw joint is perpendicular to the pitch joints. The tendon configuration is seen in Figure 4. The finger weighs less than 60 g (2 oz) and is less than 10 cm (4 in) long.

Each tendon is actuated by a brushless DC motor with a planetary reduction gearhead. A ball-screw converts rotary to linear motion, which is then transmitted to the finger through a tendon-conduit arrangement. The tendons are made of vectran, a polymer fiber, and are sheathed in steel extension springs that bridge the wrist joints. The tendon tensions are sensed through a novel load-cell placed in the load path of the conduits (Abdallah et al., 2008). This design allows for compact packaging that does not require electrical connections to moving parts.

4. Kinematic and force models

Consider the finger schematic in Figure 4. Let $\boldsymbol{q}$ and $\boldsymbol{\tau}$ represent the column matrices of joint angles and actuated joint torques respectively, and let $\boldsymbol{x}$ and $\boldsymbol{f}$ represent the column matrices of tendon positions and tensions, respectively. (Throughout this work, bold symbols represent column matrices.) The relationship between the n joint torques and the m tendon tensions is given by the tendon map, $R \in \mathbb{R}^{n\times m}$, such that

$$\boldsymbol{\tau} = R\boldsymbol{f}, \tag{1}$$

where $m > n$. The entries of R are the signed pulley radii in the tendon routing path. For the system to be tendon controllable (where "tendon controllable" means that for any $\boldsymbol{\tau} \in \mathbb{R}^n$, there exists $\boldsymbol{f} \in \mathbb{R}^m_{\geq 0}$ that satisfies (1)), R must satisfy two conditions: it must be full row rank, and there must exist an all-positive column matrix lying in its null-space (Kobayashi et al., 1998).

Similar to the approach of Salisbury and Craig (1982), the general solution of (1) for $\boldsymbol{f}$ may be formed in terms of a matrix $W \in \mathbb{R}^{(m-n)\times m}$ whose rows orthogonally complete the row space of R. That is, the $m \times m$ matrix

$$P = \begin{bmatrix} R \\ W \end{bmatrix} \tag{2}$$

is invertible, and $RW^T = 0$. In other words, the columns of W^T span the right null-space of R. With this notation, the solution for $\boldsymbol{f}$ is

$$\boldsymbol{f} = R^+\boldsymbol{\tau} + W^T\boldsymbol{\lambda}, \tag{3}$$

where the superscript $(^+)$ indicates the pseudoinverse, and $\boldsymbol{\lambda}$ is arbitrary. In light of this, it is useful to augment (1) with a new vector $\boldsymbol{t} = W\boldsymbol{f}$, which will be defined as a measure of the internal tension. Accordingly, we may write

$$\bar{\boldsymbol{\tau}} = P\boldsymbol{f}, \tag{4}$$

where $\bar{\boldsymbol{\tau}} = \begin{pmatrix} \boldsymbol{\tau} \\ \boldsymbol{t} \end{pmatrix}$. We can always form W using mutually orthogonal rows of unit length, in other words $WW^T = I$, in which case $P^{-1} = \begin{bmatrix} R^+ & W^T \end{bmatrix}$ and (3) becomes

$$\boldsymbol{f} = R^+\boldsymbol{\tau} + W^T\boldsymbol{t}. \tag{5}$$

With a suitable design, such as the finger sketched in Figure 4, the conditions for tendon controllability can be satisfied with as few as $m = n + 1$ tendons. Henceforth, we will assume that the finger to be controlled is an "n+1" design, and accordingly, W is a unit row vector that is all-positive. In addition, $\boldsymbol{t}$ becomes a scalar and shall simply be denoted as t.

The matrix R also expresses the relationship between the actuator and joint velocities. Based on the principle of virtual work, $\dot{\boldsymbol{x}} = R^T\dot{\boldsymbol{q}}$ when the tendons are inextensible (Murray et al., 1994). Allowing for elastic tendons, the actuator velocity becomes the sum of this joint contribution plus the rate-of-change of the length, $\boldsymbol{l}$, of the tendons.

$$\dot{\boldsymbol{x}} = R^T\dot{\boldsymbol{q}} + \dot{\boldsymbol{l}} \tag{6}$$

Assuming the stiffness values for the tendons are all equal and the external torque is constant, it can be shown (see Appendix A) that this elastic change in the tendon length lies in the null-space of R, such that

$$\dot{\boldsymbol{l}} = W^T\dot{\theta}. \tag{7}$$

We define $\dot{\theta}$ as the *internal velocity*, which parameterizes the set of actuator velocities that produce no motion at the joints. In other words, it represents internal motion that is absorbed by the elasticity of the tendons. The velocities can thus be related by the augmented tendon map

$$\dot{\boldsymbol{x}} = P^T\dot{\bar{\boldsymbol{q}}}, \tag{8}$$

where $\bar{\boldsymbol{q}} = \binom{\boldsymbol{q}}{\theta}$.

We will model the tendons as linear springs and assume they remain taut. Indeed, one of the objectives of the TDA of the next section is to maintain positive tension at all times. We further assume that the tendons all have the same stiffness value, k_t, which holds if all tendons are made from the same material and have approximately the same length. Both conditions hold for R2, where differences in length due to tendon routing in the finger are negligible in proportion to the total tendon length running from the forearm to the finger. The tendon tension is thus proportional to the elastic stretch, $\Delta\boldsymbol{l}$, such that

$$\boldsymbol{f} = k_t\Delta\boldsymbol{l}. \tag{9}$$

Defining the zero actuator position to correspond to $\boldsymbol{q} = \boldsymbol{0}$ with zero stretch on the tendons, and assuming that R is constant, one may integrate (6) to obtain

$$\boldsymbol{x} = R^T\boldsymbol{q} + \Delta\boldsymbol{l}, \tag{10}$$

which, combined with (9), gives the final relationship between the tendon tensions and positions,

$$\boldsymbol{f} = k_t(\boldsymbol{x} - R^T\boldsymbol{q}). \tag{11}$$

Regarding the net forces on the finger, the internal joint-space dynamics are dominated by the tendon-actuated torques. The passive dynamics here include the inertial, Coriolus, centripetal, and gravitational forces. For dexterous, multi-fingered hands like those of R2, the fingers have relatively slight inertia and size, the very reasons they were actuated remotely in the first place. For even the slightest of deflections, these passive forces thus become negligible compared to the actuated forces. For example, the R2 finger weighs less than 0.6 N (0.1 lbs) while the tendon stiffness is on the order of 700 N/cm (400 lbs/in). Thus, for the upcoming analysis, we will neglect the passive forces, resulting in a force balance equation of

$$\boldsymbol{\tau} + \boldsymbol{\tau}_e = 0, \tag{12}$$

where $\boldsymbol{\tau}_e$ represents the external torques applied to the finger.

5. Tension distribution algorithm (TDA)

The first main component of the control strategy is the TDA, which determines the tendon tensions $\boldsymbol{f}$ that will be used to exert the desired joint torques $\boldsymbol{\tau}^*$. This component is necessitated by the redundant and unidirectional nature of the tendons. We present here a solution that ensures that the tensions fall within both an upper and a lower bound, while minimizing the internal tension. The lower bound on tension, $f_{min} \geq 0$, is intended to keep the tendon taut, whereas the upper bound, $f_{max} > f_{min}$, protects the tendon from damage.

In general, the single free parameter t is insufficient to satisfy both bounds. When no satisfactory solution is possible, we deem it more important to maintain the bounds than to match $\boldsymbol{\tau}^*$. Hence, we introduce a second parameter, α, to scale down the commanded torque such that $\boldsymbol{\tau} = \alpha\boldsymbol{\tau}^*$, where $0 \leq \alpha \leq 1$. The solution should minimize both the internal tension as well as the reduction in the desired torque. Formally, we state the problem as:

Given the constraints

$$f_{min} \leq \boldsymbol{f} \leq f_{max} \tag{13}$$

$$R\boldsymbol{f} = \alpha\boldsymbol{\tau}^* \tag{14}$$

find the $\boldsymbol{f}$ that minimizes $t = W\boldsymbol{f}$ from amongst the set of solutions that minimize $\|R\boldsymbol{f} - \boldsymbol{\tau}^*\|$.

The inequalities are understood to apply to each element.

This problem can be simplified by the following observation. In the first step of the problem, minimizing $\|R\boldsymbol{f} - \boldsymbol{\tau}^*\|$, two cases are possible: $\alpha = 1$ or $\alpha < 1$. Whenever $\alpha < 1$, the solution is unique and thus the second optimization step is not needed. In this case, the higher dimensional box represented by (13) and the line represented by (14) for $\alpha = 1$ do not intersect. Since the box is axis-aligned while the direction of the line is all-positive, the line is never parallel to any side of the box and thus the closest point on the box is always unique. This fact allows us to simplify the problem and express it as the two-stage LP in Problem 1.

Problem 1: Tension distribution problem

```
Data: $\boldsymbol{\tau}^*, R, W, f_{min}, f_{max}$
maximize $\alpha$;
subject to $R\boldsymbol{f} = \alpha\boldsymbol{\tau}^*, 0 \leq \alpha \leq 1, f_{min} \leq \boldsymbol{f} \leq f_{max}$ ;
if $\alpha = 1$ then
    minimize $t = W\boldsymbol{f}$;
    subject to $R\boldsymbol{f} = \boldsymbol{\tau}, f_{min} \leq \boldsymbol{f} \leq f_{max}$ ;
end
```

An additional condition is required to ensure that a feasible solution exists. Denoting the elements of W as w_i for $i = 1, \ldots, m$, one sees that if

$$\frac{f_{max}}{f_{min}} \geq \max_{i,j} \frac{w_i}{w_j}, \tag{15}$$

then there is a feasible point of the form $\alpha = 0, \boldsymbol{f} = \beta W^T$, for some $\beta \geq 0$. As the right-hand side of (15) is fixed by the design of the finger, we assume that the tension limits on the left-hand side are always chosen to satisfy this inequality.

While one can solve the tension distribution problem with any of several LP algorithms, the following algorithm is simpler and more efficient, taking advantage of the special structure of this particular problem. For comparison, Appendix B discusses solving the problem with the simplex method.

To simplify the algorithm, we introduce vectors $\boldsymbol{v} = R^+\boldsymbol{\tau}^*$ and $\boldsymbol{w} = W^T$ so that we may rewrite (5) as

$$\boldsymbol{f} = \alpha\boldsymbol{v} + t\boldsymbol{w}. \tag{16}$$

The solution algorithm is given in pseudocode as Algorithm 2 below.

Algorithm 2: Tension distribution algorithm (TDA)

```
Input  : $\boldsymbol{\tau}^*, f_{min}, f_{max}, \boldsymbol{v}, \boldsymbol{w}$
Output: $\boldsymbol{f}$ = solution of LP problem
// Assume $\boldsymbol{\tau}^*$ is feasible ($\alpha = 1$).
1 $\alpha = 1$; $t = \max_i \frac{f_{min} - v_i}{w_i}$;
2 $\boldsymbol{f} = \alpha\boldsymbol{v} + t\boldsymbol{w}$;
3 while $\boldsymbol{f} > f_{max}$ or $\boldsymbol{f} < f_{min}$ do
      // $\boldsymbol{\tau}^*$ is not feasible ($\alpha < 1$).
4     $h = \arg\max_i f_i$, $\ell = \arg\min_i f_i$;
5     Solve for $(\alpha, t)$:
          $\begin{bmatrix} v_h & w_h \\ v_\ell & w_\ell \end{bmatrix} \begin{pmatrix} \alpha \\ t \end{pmatrix} = \begin{pmatrix} f_{max} \\ f_{min} \end{pmatrix}$;
6     $\boldsymbol{f} = \alpha\boldsymbol{v} + t\boldsymbol{w}$;
7 end
```

5.1. Analysis

First, we wish to establish that the TDA converges in a finite number of steps. The key is to consider successive solutions in the (α, t)-plane, as shown in Figures 5 and 6. When condition (15) is satisfied, we are guaranteed to have a feasible solution with $\alpha = 0$, marked as point A in the diagram. In the figures, the lines labeled $f_\ell = f_{min}$ and $f_h = f_{max}$ are the most constraining tension limits at the current value of α; that is, h and ℓ are as determined by Step 4 of Algorithm 2. The feasible side of each constraint is indicated by an arrow.

As illustrated in Figure 5, the initial iterate computed at Step 1 of Algorithm 2 starts with $\alpha = 1$ and finds the smallest t that satisfies all the minimum tension constraints (point B). If this also satisfies all the maximum tension constraints, as illustrated by the solid line for $f_h = f_{max}$, we are done. If not, such as when $f_h = f_{max}$ is the dashed line, the algorithm iterates to find a feasible solution that maximizes α.

The iterative phase is illustrated in Figure 6. The current iterate is always at the intersection of the line $f_\ell = f_{min}$ and either $\alpha = 1$ or $f_h = f_{max}$. If the current iterate satisfies all the constraints, as at point C, the algorithm terminates and the solution is optimal, because no solution to the right

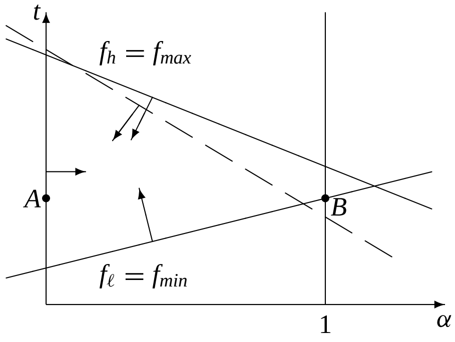

Fig. 5. Convergence proof: first iteration.

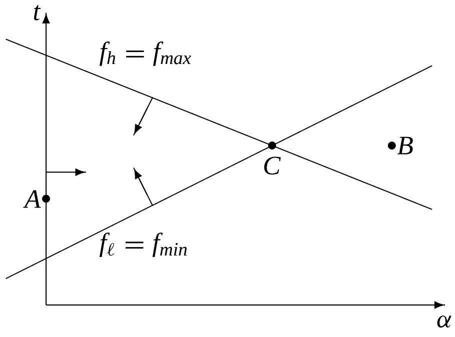

Fig. 6. Convergence proof: subsequent iterations.

of C satisfies the constraints. Otherwise, the current iterate violates either $f_{min} \leq \boldsymbol{f}$ or $\boldsymbol{f} \leq f_{max}$ or both, which means it lies in the v-shaped region to the right of C, such as at point B. The algorithm determines $f_h > f_{max}$ and $f_\ell < f_{min}$ as two of the constraints that are violated. Since all $w_i > 0$, we know that the line $f_h = f_{max}$ passes below or through B but above A, while line $f_\ell = f_{min}$ passes above or through B but below A. Therefore, the next iterate, C, at the intersection of the two lines, must lie between A and B, and in particular, α decreases at every iteration. There are a finite number of pairs (h, ℓ) to consider, and since α strictly decreases, we can never revisit any pair. Consequently, the algorithm must terminate in a finite number of steps at the optimal value for α.

In practice, the algorithm converges quickly, and in fact, there are simple conditions under which the algorithm must terminate by the first pass through the **while** loop. Let ${}^1\boldsymbol{f}$ be the initial value for $\boldsymbol{f}$ computed at line 2, let ${}^i\boldsymbol{f}$, $i = 2, 3, \ldots$, be the sequence of iterates computed at line 6 of Algorithm 2, and let α_i be the corresponding values of α. When $|\boldsymbol{\tau}^*|$ is large enough, ${}^1\boldsymbol{f}$ will violate the upper bound as shown by the dashed line case in Figure 5. The next iteration, ${}^2\boldsymbol{f}$, then solves for the intersection of the dashed line with the lower constraint, setting the current highest and lowest tensions at the boundaries. This solution is thus guaranteed to satisfy the constraints whenever ${}^2\boldsymbol{f}$ maintains the same relative order of the elements. Noting that the solutions share the same lower constraint while swapping $f_h = f_{max}$ for $\alpha = 1$ for the upper constraint, the relation between them follows.

$${}^2\boldsymbol{f} = \alpha_2\,{}^1\boldsymbol{f} + \frac{(1-\alpha_2)f_{min}}{w_\ell}\boldsymbol{w} \qquad (17)$$

This relation reveals two cases under which the algorithm terminates with ${}^2\boldsymbol{f}$. The first case is when $f_{min} = 0$. Since the first term on the right-hand side is just proportional to ${}^1\boldsymbol{f}$, it cannot change the order of the elements. The second case is when the elements of $\boldsymbol{w}$ are all equal, in other words, the null-space is isometric. The second term on the right will thus add an equal amount to each tension and will not change their order either. This occurs given a symmetric design of the joints, such as when the tendons are arranged on equal-diameter pulleys. More generally, the condition occurs when each row of R sums to zero.

Even if neither of these conditions holds exactly, it may happen that f_{min} is relatively small and the asymmetry of the finger design is low. Then, one can expect that in most cases ${}^2\boldsymbol{f}$ still is optimal, although further iterations may occasionally be needed. We explore this further in a numerical study of the R2 finger, next.

5.2. R2 application

For R2, we desire to set a non-zero f_{min}. In normal operation, a small f_{min} keeps the tendons preloaded in tension to avoid sudden snapping from a slack to a taut state. Pretension also helps in control by raising the tendon tension measurements above the level of noise in the sensors.

Furthermore, the R2 fingers have a mildly asymmetric design. Consider the index finger of R2, which has the following tendon map and null-space.

$$R = \begin{bmatrix} 0.38 & 0.38 & -0.38 & -0.38 \\ 0.67 & -0.50 & 0.67 & -0.50 \\ 0 & 0 & 0.50 & -0.50 \end{bmatrix} \text{cm} \qquad (18)$$

$$W = [\, 0.347 \;\; 0.632 \;\; 0.490 \;\; 0.490 \,] \qquad (19)$$

One can see that rows 1 and 3 of R sum to zero, but row 2 does not.

Since $f_{min} \neq 0$ and the finger design is asymmetric, the tendon distribution algorithm may need more than two iterations. To assess how many iterations could be expected, we tested the algorithm in simulation.

The actual actuators for the robot were designed for a maximum of 220 N (50 lb) tension, producing a maximum joint torque of 2.9 Nm (26 in-lb) on the finger. To test the algorithm, the limits were set to $f_{min} = 8.9$ N (2 lb) and $f_{max} = 178$ N (40 lb). The value of f_{min} was selected conservatively; lower, more typical values reduce the chance of needing iterations. According to condition (15), a solution always exists given $f_{max} > 16$ N (3.6 lb).

The simulation tested all torque combinations spanning a range of $\tau_i \in [-5.5, 11]$ Nm ($[-50, 100]$ in-lb), with a 4.5 Nm (1 in-lb) resolution. An element of ${}^{2}\boldsymbol{f}$ exceeded the limits on less than 2% of the torque values. Even in those cases, the high and low values remained close to the limits. Out of the three million test points tried, the absolute maximum tension value was 183 N (41.1 lb), and the absolute minimum was 5.3 N (1.2 lb). Allowing for a third iteration, ${}^{3}\boldsymbol{f}$ exceeded the limits on effectively 0% of the values. The absolute extreme values returned were 179 N (40.2 lb) and 8.9 N (2.0 lb), respectively. Hence, we can confidently cap the iterations in the algorithm for R2 at three.

The advantage of this algorithm lies in two key points. First, it distributes the tensions with a simple, efficient algorithm. Second, it maintains a minimum preload tension and caps the maximum tension with a linear scaling of the desired joint torques. This feature protects the tendons from being overloaded either by the controller or by the environment. It also allows the system to avoid mechanically saturating the tensions without causing a nonlinear disturbance to the joint torques. The algorithm is general and works for any "n+1" arrangement that has all $w_i > 0$ and that satisfies the existence condition of (15).

6. Tendon-space control

The second main component of the control strategy is the control law. First, we will present the control law in the conventional tendon-space formulation. The tendon-space controller has been the standard approach for finger torque control. It simplifies the controller design by allowing the controller for each tendon to be analyzed and optimized independently; however, as we show here, it also introduces a coupled disturbance to the finger dynamics. This section will present both the control law and a closed-loop analysis of its behavior.

6.1. Control law

The tendon-space controllers consist of two steps. First, the desired tensions are found using the inverse of P,

$$\boldsymbol{f}_d = P^{-1}\bar{\boldsymbol{\tau}}_d, \tag{20}$$

where $\bar{\boldsymbol{\tau}}_d = \begin{pmatrix} \boldsymbol{\tau}_d \\ t_d \end{pmatrix}$ and t_d is selected by the TDA. Second, independent tension controllers servo to the desired tension at each tendon. The choice of the control law used for this tension servo does not affect the underlying behavior of the tendon-space formulation. In this case, we apply a PD compensator for the motor voltage command, $\boldsymbol{u}$, such that

$$\boldsymbol{u} = -k_p\left(\boldsymbol{f} - \boldsymbol{f}_d\right) - k_d\dot{\boldsymbol{x}}, \tag{21}$$

where k_p and k_d are the constant, scalar gains. For the dissipative term here, we used the position derivatives instead of the force derivatives due to their superior signal quality.

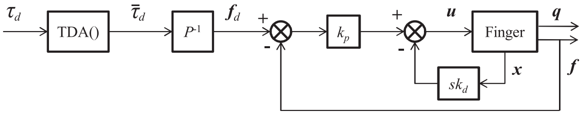

Fig. 7. The tendon-space control law. Reference torques must first be filtered by the tension distribution algorithm (TDA), which sets the desired internal tension.

A block diagram of this control law is shown in Figure 7. The desired torque must first be filtered by the TDA. Furthermore, a stiffness controller can be wrapped around this torque controller by setting the desired joint torques proportional to the joint error:

$$\boldsymbol{\tau}_d = -K(\boldsymbol{q} - \boldsymbol{q}_d), \tag{22}$$

where K is the diagonal stiffness matrix, and $\boldsymbol{q}_d$ is the reference joint position.

6.2. Analysis

To identify the net effect of this formulation, consider the closed-loop analysis of the finger. First, we will model the actuator with a typical second-order transfer function, $G(s)$, and assume that the actuators are all identical. Hence,

$$s\boldsymbol{X} = G(s)\,\boldsymbol{U}. \tag{23}$$

Let $\boldsymbol{X}(s)$ and $\boldsymbol{U}(s)$ represent the Laplace transforms of $\boldsymbol{x}(t)$ and $\boldsymbol{u}(t)$ respectively, where s is the Laplace variable. Similarly, $\boldsymbol{F}(s)$ and $\boldsymbol{Q}(s)$ will represent the Laplace transforms of $\boldsymbol{f}(t)$ and $\boldsymbol{q}(t)$, respectively.

To derive the transfer function, consider the Laplace transform of the control law (21) with a substitution from the actuator model.

$$s\left(G^{-1} + k_d\right)\boldsymbol{X} = -k_p\left(\boldsymbol{F} - \boldsymbol{F}_d\right) \tag{24}$$

Substituting from the tendon tension model (11) and assuming ideal tension sensing, we can solve for the closed-loop tension.

$$\left(k_p + \frac{G^{-1}}{k_t}s + \frac{k_d}{k_t}s\right)\boldsymbol{F} = k_p\boldsymbol{F}_d - s\left(G^{-1} + k_d\right)R^T\boldsymbol{Q} \tag{25}$$

Multiplying both sides by R, we will assume that there are no external forces and apply (12) to derive the following transfer function for the torque controller.

$$s\boldsymbol{Q} = \frac{k_p}{G^{-1} + k_d}\left(RR^T\right)^{-1}\boldsymbol{\tau}_d \tag{26}$$

For the final tendon-space transfer function, we will apply the stiffness law (22) on top of the torque controller. Also, we will assume that the motor dynamics are much faster than the control dynamics, such that $G(s) \approx G(0)$. Hence, the transfer function becomes

$$\boldsymbol{Q} = \left[I + s\tilde{K}^{-1}RR^T\right]^{-1}\boldsymbol{Q}_d, \tag{27}$$

where I is the identity matrix and $\tilde{K} = \frac{k_p}{G^{-1}(0)+k_d}K$. This final transfer function reveals a transient, first-order coupling induced by the kinematics of the finger (RR^T). Hence,

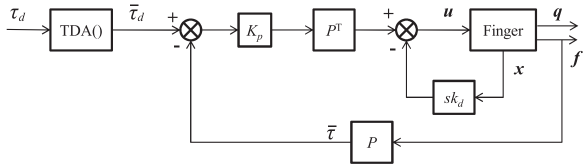

Fig. 8. The joint-space control law.

changes in the reference joint torque and position in one joint, as well as disturbances, will have unintended effects on the torque and position at other joints.

7. Joint-space control

This coupling induced by the previous controller can be eliminated by a simple transformation to the joint space. This section presents the new joint-space formulation.

7.1. Control law

To derive the joint-space control law, we first need to map the actuator model to the augmented joint space. From the actuator model (23) and the velocity transformation (8), the joint velocity relates to the motor command as follows.

$$\begin{aligned} s\bar{\boldsymbol{Q}} &= P^{-T}G\boldsymbol{U} \\ &= G\tilde{\boldsymbol{U}} \end{aligned} \tag{28}$$

From this expression, $\tilde{\boldsymbol{u}} = P^{-T}\boldsymbol{u}$ is defined as the joint-space equivalent of the motor command. Note that this result depends on the assumption that the actuators are all identical, so that G commutes with P^{-T}, as is the case with R2.

Given this transformation, we would now like to apply a linear, decoupled servo in the augmented joint space, analogous to the tendon-space servo (21).

$$\tilde{\boldsymbol{u}} = -K_p\left(\bar{\boldsymbol{\tau}} - \bar{\boldsymbol{\tau}}_d\right) - K_d\dot{\bar{\boldsymbol{q}}} \tag{29}$$

K_p here is a diagonal gain matrix composed diagonally of k_pI and k_{pi}, which correspond to the joint and internal spaces respectively. On the other hand, let $K_d = k_dI$ for reasons that will become apparent.

Mapping this relation back to the actual motor command provides us with our joint-space control law.

$$\begin{aligned} \boldsymbol{u} &= -P^T\left[K_p\left(\bar{\boldsymbol{\tau}} - \bar{\boldsymbol{\tau}}_d\right) + k_d\dot{\bar{\boldsymbol{q}}}\right] \\ &= -P^TK_p\left(\bar{\boldsymbol{\tau}} - \bar{\boldsymbol{\tau}}_d\right) - k_d\dot{\boldsymbol{x}} \end{aligned} \tag{30}$$

The block diagram for this scheme is shown in Figure 8. Having a scalar k_d here allowed us to translate the damping term to the tendon space. The collocation achieved by using the actuator sensing, $\dot{\boldsymbol{x}}$, instead of the joint sensing, $\dot{\boldsymbol{q}}$, increases the stability of the system. Otherwise, the position feedback and damping terms would lag the actuation by the unmodeled dynamics of the tendon-conduit transmission.

7.2. Analysis

To analyze the closed-loop behavior, consider the Laplace transform of this control law combined with the actuator model.

$$s\left(G^{-1} + k_d\right)\boldsymbol{X} = -P^TK_p\left(\bar{\boldsymbol{\tau}} - \bar{\boldsymbol{\tau}}_d\right) \tag{31}$$

Substituting from the tendon tension model (11),

$$s\left(G^{-1} + k_d\right)\left(\boldsymbol{F} + k_tR^T\boldsymbol{Q}\right) = -k_tP^TK_p\left(\bar{\boldsymbol{\tau}} - \bar{\boldsymbol{\tau}}_d\right). \tag{32}$$

We can then solve for the closed-loop joint torques by mapping both sides to the range space with R and by noting that RR^T is nonsingular.

$$\begin{aligned} \left[k_pI + s\left(G^{-1} + k_d\right)\left(k_tRR^T\right)^{-1}\right]\boldsymbol{\tau} = \\ k_p\boldsymbol{\tau}_d - s\left(G^{-1} + k_d\right)\boldsymbol{Q} \end{aligned} \tag{33}$$

Assuming once again the condition of zero external forces from (12), the transfer function for the joint-space torque controller follows.

$$s\boldsymbol{Q} = \frac{k_p}{G^{-1} + k_d}\boldsymbol{\tau}_d \tag{34}$$

Again, assume that the motor dynamics are much faster than the control dynamics. The corresponding transfer function for the stiffness controller reveals a decoupled, first-order response given that K, and therefore $\tilde{K}$, is diagonal.

$$\boldsymbol{Q} = \left[I + s\tilde{K}^{-1}\right]^{-1}\boldsymbol{Q}_d \tag{35}$$

8. Experimental results

The behavior of the full control strategy can now be validated experimentally. First, the theoretical closed-loop behavior of the two control laws is demonstrated in simulation. Then, the same tests are conducted on the R2 index finger. Finally, the performance of the torque regulation and the underlying tension management are displayed. The tests capture the strong correspondence between the theoretical and experimental results, as well as the overall superiority of the joint-space control law.

8.1. Simulation

The first test features a simulation modeling the two transfer functions: the tendon-space (27) and joint-space (35) relations. The model was developed in Simulink™, and the tendon map (R) modeled the R2 index finger.

The experiment tested the response to a step input in both the proximal and medial pitch joints. The joints were given a step input of $\boldsymbol{q} = (0, 45, 90)^T$ deg, first in the positive direction and then in the negative direction. Shown in Figure 9, the results demonstrate the coupling effects of the tendon-space formulation driving the medial joint in the wrong direction before converging to the reference values.

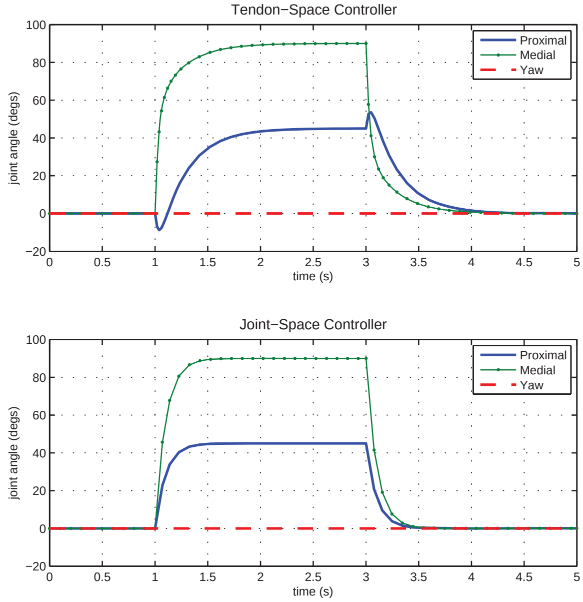

Fig. 9. Simulation results for the closed-loop transfer functions. A positive then negative step input of $\boldsymbol{q} = (0, 45, 90)$ deg was commanded.

The joint-space formulation, on the other hand, eliminated this coupling with an independent first-order response.

Despite the simplicity of these transfer functions, they effectively capture the essential behavior of the two formulations. This behavior was consistently demonstrated throughout different tests and validated on the R2 platform, as we will see next, as well as on independent testbeds.

8.2. *R2 step response*

The second test applied the same step input to the R2 finger. The results are shown in Figure 10. The tendon-space controller exhibited the aforementioned transient coupling, coupling that disturbed the tension tracking, delayed the response time, and produced unsightly motion in the finger. This behavior was consistently demonstrated throughout our tests; it also surfaced in the response to external disturbances.

The joint-space controller, on the other hand, eliminated the coupling and significantly increased the speed of the response. For both moving joints, the joint-space controller reduced the settling time by almost 25%. The medial joint dropped from a settling time of 1.3 s to 0.97 s. The proximal joint dropped from a settling time of 2.2 s to 1.7 s. This gain in speed is a result of the decoupling effect, rather than control gains. This difference is highlighted in Figure 11, where the response of the proximal joint for both laws is displayed.

The controller parameters are shown in Table 1. In each case, the gains were tuned to maximize performance. As seen in the table, a higher stiffness value was applied to the yaw joint in both cases. The yaw joint is relatively ill-conditioned due to significantly smaller radii; hence, the higher stiffness better controlled its motion. The steady-state errors seen in the results can be addressed with a limited-range integrator.

Table 1. Controller parameters used in the experimental run.

	Tendon space	Joint space
k_d	0.01	0.01
k_p, K_p	0.01	$\begin{bmatrix} 0.05 & 0 & 0 & 0 \\ 0 & 0.05 & 0 & 0 \\ 0 & 0 & 0.05 & 0 \\ 0 & 0 & 0 & 0.003 \end{bmatrix}$
K	$\begin{bmatrix} 0.3 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.2 \end{bmatrix}$	$\begin{bmatrix} 0.3 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.2 \end{bmatrix}$

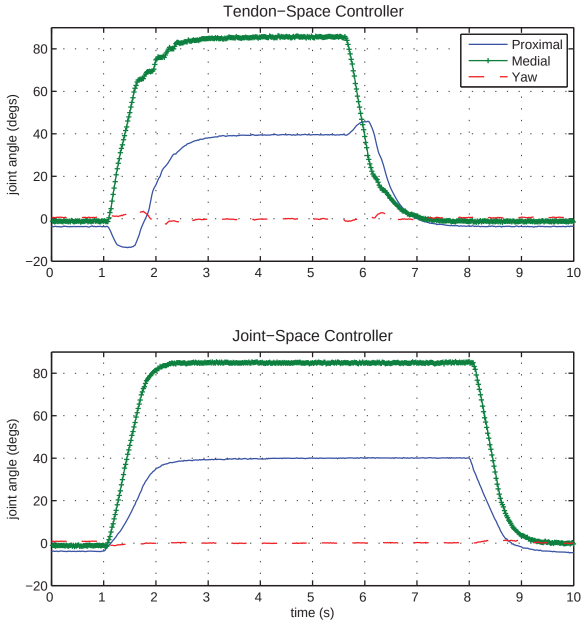

Fig. 10. Experimental results from R2 for the two control laws. The same step input of $\boldsymbol{q} = (0, 45, 90)$ deg was commanded. Note both the decoupled and faster response of the joint-space controller.

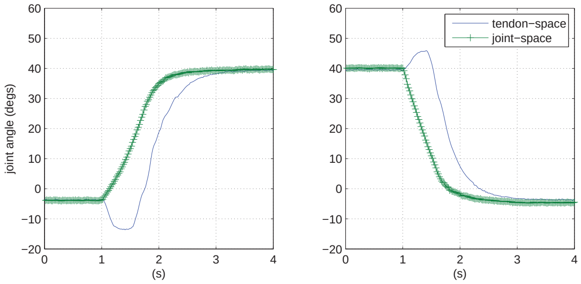

Fig. 11. Direct comparison of the proximal-joint response for both the tendon- and joint-space controllers.

8.3. *R2 tension and torque regulation*

The final test demonstrated the performance of the torque regulation in conjunction with the TDA. Instead of the stiffness controller, this experiment applied the torque controller for direct torque commands. With the finger sitting against a hard surface, the reference torque for the proximal joint ramped up from 0 to 1.7 N-cm (3 in-lb). The minimum and maximum tension were set at 4.4 N (1 lb) and 36 N (8 lb), respectively. At about 1.3 N-cm, the tensions saturated and the scaled solution kicked in for the distribution algorithm.

Results of the experiment are shown in Figure 12. The first figure shows the tension distribution satisfying the upper and lower limits. Ideally, the lowest tension should always equal 4.4 N, while the maximum tension should equal 36 N during saturation. Deviations from this ideal are due to errors in the torque regulation and sensor calibration. The errors increase with rising tensions due to the similarly increasing friction in the system. The second figure shows the torque regulation, where the displayed torques are computed from the tension feedback using (4). While the medial and yaw torques should remain at zero, the proximal should follow the reference torque (solid line) until saturation. At that point, it should follow the scaled solution (dashed line). The experiment demonstrates how saturation is achieved without nonlinear joint coupling.

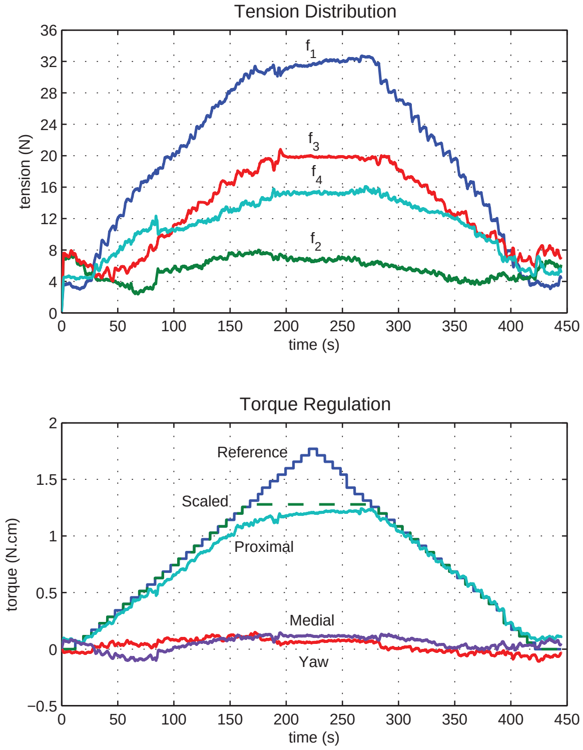

Fig. 12. With the finger in torque-control mode, the reference proximal torque is given a triangle input from 0 to 1.7 N-cm. The controller kept the tensions bounded by the range [4.4 N, 36 N], allowing the tensions to saturate without coupling the joint torques.

9. Conclusion

For tendon-actuated manipulators, joint-space control offers clear advantages over tendon-space control. Not only does it eliminate the coupled transience exhibited by tendon-space formulations, it also significantly increases the speed of the response. Furthermore, the transformation from tendon space to joint space is surprisingly simple. Compare the joint-space law (30) to the following re-expression of the tendon-space law (21):

$$\boldsymbol{u} = -P^{-1}k_p\,(\bar{\boldsymbol{\tau}} - \bar{\boldsymbol{\tau}}_d) - k_d\dot{\boldsymbol{x}} \qquad (36)$$

The difference is in the use of the transpose versus the inverse of P, where the transpose provides the decoupling of the joint space. This effect is reminiscent of the role of the Jacobian in the operational-space control of serial manipulators, where the inverse provides the decoupling.

The problem of tension distribution, faced by all torque controllers, is essentially a linear programming problem. Using knowledge of the system, we are able to reduce the problem to one more conducive to real-time implementation. The upper bound in the algorithm is needed to protect the tendons and to avoid nonlinear saturation effects. The adjustable lower bound allows the controller to either accommodate calibration errors in the tension sensors (with a higher bound) or to reduce the internal tension on the finger (with a lower bound).

Robotic hands designed for assembly applications need to interact with unstructured environments while approaching human size and dexterity. The TDA and joint-space control law together provide for a cohesive strategy that closely protects the tendons while operating transparently of the coupled actuation.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Acknowledgment

This work was conducted while R Platt was with the NASA Johnson Space Center. The authors would like to thank Brian Hargrave, Matt Reiland, Lyndon Bridgwater and the rest of the R2 team for their role in the control and design of the hand.

References

Abdallah M, Bridgwater L, Diftler M, et al. (2008) *Sensing the tendon tension through the conduit reaction forces.* Patent application 12/269552, USA.

Abdallah ME and Wampler CW (2011) Torque control of underactuated tendon-driven fingers. *Mechanical Sciences* 2(1): 83–90. (Available online at www.mech-sci.net/2/83/2011.)

Diftler M, Mehling J, Abdallah M, et al. (2011) Robonaut 2 – the first humanoid robot in space. In: *IEEE international conference on robotics and automation (ICRA)*, Shanghai, China, May 2011.

Gabiccini M, Branchetti M and Bicchi A (2011) Dynamic optimization of tendon tensions in biomorphically designed hands with rolling constraints. In: *IEEE international conference on robotics and automation (ICRA)*, Shanghai, China, May 2011.

Hirose S and Ma S (1991) Coupled tendon-driven multijoint manipulator. In: *IEEE international conference on robotics and automation (ICRA)*, Sacramento, CA, April 1991, pp. 1268–1275.

Jacobsen S, Ko H, Iversen E, et al. (1989) Antagonistic control of a tendon driven manipulator. In: *IEEE international conference on robotics and automation (ICRA)*, Scottsdale, AZ, May 1989.

Jacobsen S, Wood J, Knutti D, et al. (1984) The Utah/MIT hand: Work in progress. *International Journal of Robotic Research* 3(4): 21–50.

Kobayashi H, Hyodo K and Ogane D (1998) On tendon-driven robotic mechanisms with redundant tendons. *International Journal of Robotics Research* 17(5): 561–571.

Kobayashi H and Ozawa R (2003) Adaptive neural network control of tendon-driven mechanisms with elastic tendons. *Automatica* 39: 1509–1519.

Lee JJ (1991) *Tendon-driven manipulators: Analysis, synthesis, and control*. PhD Thesis, University of Maryland, College Park, MD.

Lee Y, Choi H, Chung W, et al. (1994) Stiffness control of a coupled tendon-driven hand. *IEEE Control Systems Magazine* 14(5): 10–19.

Lee JJ and Tsai LW (1993) Torque resolver design for tendon-driven manipulators. *ASME Journal of Mechanical Design* 115: 877–883.

Ma S, Hirose S and H Yoshinada (1993) Design and experiments for a coupled tendon-driven manipulator. *IEEE Control Systems Magazine* 13(1): 30–36.

Mulero-Martinez J, Garcia-Cordova F and Lopez-Coronado J (2006) Position control based on static neural networks of anthropomorphic robotic fingers. *Advances in Neural Networks* 3972: 1188–1197.

Murray R, Li Z and Sastry S (1994) *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL: CRC Press.

Ogane D, Hyodo K and Kobayashi H (1996) Mechanism and control of a 7 DOF tendon-driven robotic arm with NST. *Journal of the Robotics Society of Japan* 14(8): 1152–1159. (In Japanese.)

Salisbury JK and Craig JJ (1982) Articulated hands: Force control and kinematic issues. *International Journal of Robotics Research* 1(1): 4–17.

Starr GP (1990) Experiments in assembly using a dexterous hand. *IEEE Transactions on Robotics and Automation* 6(3): 342–347.

Wimbock T, Ott C, Albu-Schaffer A, et al. (2008) Impedance control for variable stiffness mechanisms with nonlinear joint coupling. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, Nice, France, September 2008.

Appendix A: Augmented velocity map

The velocities in (6) can be expressed with the augmented tendon map in (8) when $\dot{\boldsymbol{l}} \in \mathcal{N}(R)$. For the derivation, start with the force balance equation of (12). If K_t is the diagonal matrix of tendon stiffnesses, the time derivative of the torque becomes

$$\begin{aligned} \dot{\boldsymbol{\tau}}_e &= -R\dot{\boldsymbol{f}} \\ &= -RK_t\dot{\boldsymbol{l}}. \end{aligned} \qquad (37)$$

Solving for $\dot{\boldsymbol{l}}$ reveals both a range-space and null-space component:

$$\dot{\boldsymbol{l}} = -(RK_t)^+\,\dot{\boldsymbol{\tau}}_e + \mathcal{N}(RK_t)\,\boldsymbol{\lambda}, \qquad (38)$$

where $\boldsymbol{\lambda}$ is arbitrary. If the tendon stiffnesses all equal k_t, then $\mathcal{N}(RK_t) = \mathcal{N}(R)$. Using W as a basis for the null-space of R, (38) can be expressed as follows:

$$\dot{\boldsymbol{l}} = -\frac{1}{k_t}(R)^+\,\dot{\boldsymbol{\tau}}_e + W^T\dot{\boldsymbol{\theta}} \qquad (39)$$

When the external torque is instantaneously constant, the range-space term is eliminated. Hence, $\dot{\boldsymbol{l}}$ will all lie in the null-space and the augmented map in (8) is correct. Moreover, $\dot{\boldsymbol{\theta}}$ is defined as the *internal velocity*. Under these conditions, it parameterizes the space of tendon velocities that produce no change in the joint torques, and hence no motion in the joints. In the case of unequal stiffnesses, the result remains true if W is orthogonal to RK_t instead of just to R. Deviations from these assumptions represent disturbances for the feedback controller to correct.

Appendix B: Tension distribution by the simplex method

The standard form of the simplex method applies to a set of linear inequalities on variables restricted to the positive orthant. Renumber the tendons so that the elements of the internal tension vector $\boldsymbol{w}$ are in ascending order: $w_1 \le w_2 \le \ldots \le w_m$. Assuming that (15) holds, we have a feasible point at $\alpha = 0$, $t = f_{min}/w_1$. This feasible point is on the minimum tension constraint for tendon 1: $f_1 \ge f_{min}$. Let y be a new variable expressing the excess of f_1 over f_{min}, which must be nonnegative:

$$y := v_1\alpha + w_1 t - f_{min} \qquad (40)$$

Then the LP problem can be restated as

$$\max_{\alpha,y} \alpha$$

subject to

$$\begin{aligned} &0 \le \alpha \le 1 \\ &0 \le y \\ &v_i\alpha + w_i t(y) \ge f_{min}, \qquad i = 2, \ldots, m \\ &v_i\alpha + w_i t(y) \le f_{max}, \qquad i = 1, \ldots, m \end{aligned}$$

where $t(y)$ is the linear expression obtained by solving (40) for t. In this reformulation, the initial feasible point is $(\alpha, y) = (0, 0)$. Since the objective function is α, the simplex method slides first along $y = 0$, the minimum tension line for f_1, and then walks along a succession of minimum tension constraints until it either hits $\alpha = 1$ or a maximum tension constraint. At that point, the solution is feasible and against both a minimum and maximum constraint, so similar to the argument in Section 5.1, it must be optimal. This implies that the simplex method stops after at most m steps, since that is the total number of minimum tension constraints.

Either the solution method of Algorithm 2 or the above simplex method can be used. Experiments using the conditions of the R2 application show that the number of iterations is comparable, but Algorithm 2 runs about four times as fast due to simpler iteration logic and the fact that there is no need to convert the problem to standard simplex form. Optimization of the simplex method for this problem could reduce the difference in computational speed, but the resulting procedure will not match the utter simplicity of Algorithm 2.