Position Control of Tendon-Driven Fingers with Position Controlled Actuators

Muhammad E. Abdallah, Robert Platt Jr., Brian Hargrave, Frank Permenter

Abstract— **Conventionally, tendon-driven manipulators implement some force-based controller using either tension feedback or dynamic models of the actuator. The force control allows the system to maintain proper tensions on the tendons. In some cases, whether it is due to the lack of tension feedback or actuator torque control, a purely position-based controller is needed. This work compares three position controllers for tendon-driven manipulators that implement a nested actuator position controller. A new controller is introduced that achieves the best overall performance with regards to speed, accuracy, and transient behavior. To compensate for the lack of tension control, the controller nominally maintains the internal tension on the tendons through a range-space constraint on the actuator positions. These control laws are validated experimentally on the Robonaut-2 humanoid hand.**

I. Introduction

Tendon transmission systems are often used in the actuation of fingers for high degree-of-freedom (DOF) hands. The remote actuation allows for significant reductions to the size and weight of the fingers, features that are important for dexterous manipulation. Since the tendons can only transmit forces in tension, the number of actuators must exceed the manipulator DOF's to achieve fully determined control of the finger. This redundancy entails a null-space that is needed to maintain some minimum level of tensioning on the tendons.

Accordingly, an ideal control law for such a system would be a force-based controller with tension feedback. Through the feedback, the tendons can always be kept taut and appropriate levels of tensioning can be maintained. Even without the tension feedback, the force controller may employ dynamic models of the actuator to estimate the tensions. Occasionally, however, such force controllers are not viable and a purely position-based control law is desired. In cases where tension feedback is not available, a lower force control loop is not developed, or the control bandwidth does not allow for sufficiently high joint stiffnesses, a position-based controller is needed.

Many researchers have presented force-based controllers for tendon-driven fingers. Most of the systems have applied such force control using tension feedback. This includes almost all the robotic finger applications [1]–[5], as well as a few larger manipulators [6], [7]. Others have applied the force control using the alternative modeling approach, using either dynamic models of the actuation [8]–[10] or static models of the tendon [11]. A purely position-based controller, however, is desired that requires neither force sensing or modeling.

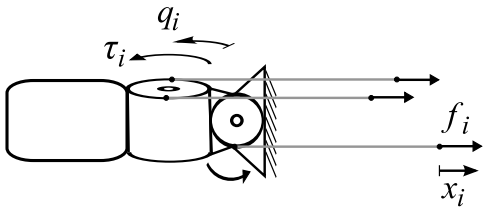

Fig. 1. Schematic of a simple finger with tendons.

The challenge is to develop a controller that can achieve the desired performance while maintaining suitable tensions on the tendons. This work will focus on three criteria. First, the controller needs to produce a fast response time with low steady-state error. Second, it needs to produce no transient overshoot. The overshoot can cause spikes in the tension as either the tendons fight each other or hard limits are struck. Eliminating the overshoot thus becomes important in the absence of tension feedback, and it applies to both the joint and actuator spaces. Third, the controller must be able to maintain the internal, or null-space, tensions on the tendons. Given some initial state of tensioning, the controller needs to maintain the internal tensions to keep the tendons from either going slack or applying excessive loads. Without tension sensing, this objective can be nominally achieved by eliminating the null-space motion amongst the tendons.

This work compares several position controllers for tendon-driven manipulators. A new controller is presented that achieves superior transient performance compared to equivalent proportional-integral (PI) based controllers. This controller implements a nested actuator position loop with a range-space constraint to eliminate the null-space motion. The controllers are validated experimentally on the three DOF fingers of the Robonaut-2 (R2) humanoid hand.

II. Finger Kinematics

Before introducing the control laws, an understanding of the finger kinematics is needed. For that purpose, consider the schematic of a representative tendon-driven manipulator shown in Fig. 1. $\boldsymbol{q}$ and $\boldsymbol{\tau}$ represent the column matrices of positions and actuated joint torques, respectively. $\boldsymbol{x}$ and $\boldsymbol{f}$ represent the column matrices of tendon positions and tensions, respectively. The relationship between the n joint

M. Abdallah is with the Manufacturing Systems Research Lab, General Motors R&D, Warren, MI 48090, USA muhammad.abdallah@gm.com

R. Platt was with the Johnson Space Center, NASA, Houston, TX 77058, USA robert.platt-1@nasa.gov

B. Hargrave and F. Permenter are with Oceaneering Space Systems, Houston, TX 77058, USA {bhargrave, fpermenter}@oceaneering.com

Patents are pending on this work.

Preprint submitted to 2012 IEEE International Conference on Robotics and Automation. Received September 2, 2011.

torques and the m tendon tensions follows, where $m > n$.

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

$R \in \mathbb{R}^{n\times m}$ is known as the tendon map, consisting of the joint radii data. For the system to be tendon controllable, 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 [12]. Inversely, the solution for $\boldsymbol{f}$ follows, where R^+ is the pseudoinverse of R, I is the identity matrix, and $\boldsymbol{\lambda}$ is arbitrary.

$$\begin{aligned} \boldsymbol{f} &= R^+\boldsymbol{\tau} + \boldsymbol{f}_{int} \\ \boldsymbol{f}_{int} &\doteq (I - R^+R)\,\boldsymbol{\lambda} \end{aligned} \tag{2}$$

$\boldsymbol{f}_{int}$ represents the *internal tensions*, lying in the null-space of R and producing zero net torques. The matrix $[I - R^+R]$ provides the projection operator into the null-space of R. Given quasi-static conditions, $\boldsymbol{f} = \boldsymbol{f}_{int}$ whenever zero external forces act on the finger. The bold symbols throughout represent column matrices.

This same R expresses the relationship between the actuator and joint velocities. Based on the principle of virtual work, the actuator velocity equals $R^T\dot{\boldsymbol{q}}$ when the tendons are inextensible [11]. 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.

$$\dot{\boldsymbol{x}} = R^T\dot{\boldsymbol{q}} + \dot{\boldsymbol{l}} \tag{3}$$

Assuming a constant R and integrating these velocities:

$$\boldsymbol{x} = R^T\boldsymbol{q} + \Delta\boldsymbol{l}, \tag{4}$$

where $\Delta\boldsymbol{l}$ is a change in length relative to the zero-positions length, i.e. the length of the tendons when $\boldsymbol{x} = 0$ and $\boldsymbol{q} = 0$.

We will model the tendons as linear springs of stiffness k_t and assume they remain taut. If the system is defined such that the zero-positions length equals the unstretched length of the tendons, the forces become proportional to $\Delta\boldsymbol{l}$.

$$\begin{aligned} \boldsymbol{f} &= k_t\Delta\boldsymbol{l} \\ \boldsymbol{\tau} &= k_tR\Delta\boldsymbol{l} \end{aligned} \tag{5}$$

Solving for $\Delta\boldsymbol{l}$ in this expression reveals both a range-space and null-space component:

$$\begin{aligned} \Delta\boldsymbol{l} &= \tfrac{1}{k_t}R^+\boldsymbol{\tau} + \Delta\boldsymbol{l}_{int} \\ \Delta\boldsymbol{l}_{int} &\doteq (I - R^+R)\,\boldsymbol{\delta}, \end{aligned} \tag{6}$$

where $\boldsymbol{\delta}$ is arbitrary. $\Delta\boldsymbol{l}_{int}$ represents the change of length in the null-space of R, i.e. the change in length that effects only the internal tensions, not the joint torques. Hence, the first term on the right hand side of (6) represents the change in length due to external loads, while the second term represents the change in length due to the internal tensions. Substituting into (4) provides the final relation for the tendon displacement.

$$\boldsymbol{x} = R^T\boldsymbol{q} + \frac{1}{k_t}R^+\boldsymbol{\tau} + \Delta\boldsymbol{l}_{int} \tag{7}$$

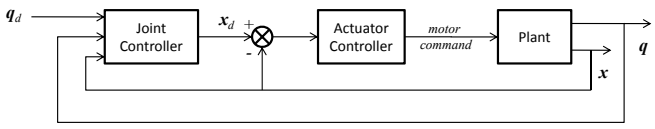

Fig. 2. The control architecture. The lower, actuator position loop allows the Finite-Difference controller to actively constrain the actuator motion to the range-space.

In the absence of tension feedback, the only way to keep the internal tension constant is to eliminate the internal motion, $\Delta\boldsymbol{l}_{int}$. This implies that the actuator position must lie in the range-space of R^T. Assuming we have zero external forces and an accurate kinematic model, staying in the range-space will keep the static tensions on each tendon constant, preventing the tendons from either going slack or being overloaded. Of course, an external load may cause the actual tensions to drop to zero or to reach excessive highs, however, the tensions will return to their original state once the load is removed.

III. Control Laws

Based on these kinematics, a set of control laws can now be presented. The controller implements a two-tiered architecture with an upper loop regulating joint positions and a lower loop regulating actuator positions. Shown in Fig. 2, the upper loop passes actuator position commands down to the lower loop. Not only is it common for actuators to operate a well-tuned position controller, but this hierarchy also exists to accommodate the range-space constraint needed by the first of the three control laws presented here. We assume here that the lower loop has been tuned to maximize performance with a first-order response behavior (i.e. without overshoot).

A. Finite-Difference Law

The first control law implements a discrete version of a velocity controller, feeding back the current position of the actuators combined with incremental changes from the joint error. We thus refer to it as the *Finite-Difference* controller. Based on the velocity relation in (3) with a constant tendon length, the commanded position is:

$$\boldsymbol{x}_d = \boldsymbol{x} - k_pR^T\Delta\boldsymbol{q}, \tag{8}$$

where $\Delta\boldsymbol{q} = \boldsymbol{q} - \boldsymbol{q}_d$, $\boldsymbol{q}_d$ is the desired joint position, and k_p is a scalar constant gain.

This control law commands actuator increments only in the range-space, thus introducing no internal motion. It works well in producing a fast response that closes the steady-state error and maintains an over-damped behavior. The problem, however, is that it does not actively constrain the actuator positions to the range-space. Although the increment always lies in the range-space, the actual positions can deviate due to external disturbances or actuator saturation effects. These errors are then propogated due to the incremental nature of the controller. This effect is exacerbated by the inability of the tendons to resist compression. Consider thus

Preprint submitted to 2012 IEEE International Conference on Robotics and Automation. Received September 2, 2011.

any case in which the finger is externally constrained: the tendons opposing the joint error in tension remain restrained, while the tendons supposedly in compression run away. This internal motion will dissipate the internal tension, possibly even leaving the finger uncontrollable due to the slack in the tendons.

To resolve this problem, the output of the control law needs to be projected into the range-space of the finger. This will allow the lower actuator loop to actively servo to the range-space. That projection is achieved by the operator R^+R. Noting that R^+R is symmetric, the new commanded position follows.

$$\begin{aligned} \boldsymbol{x}_d &= R^+R\Big(\boldsymbol{x} - k_p R^T \Delta\boldsymbol{q}\Big) \\ &= R^+R\boldsymbol{x} - k_p R^T \Delta\boldsymbol{q} \end{aligned} \tag{9}$$

This results in our final *Finite-Difference* control law. This controller produces the same positive transient and steady-state performance as (8); however, it resists the internal motion even when disturbed. Note that the initial relation in (8) could have been implemented with a single-loop controller, setting the motor input proportional to $R^T\Delta\boldsymbol{q}$. The range-space constraint of (9), however, requires the nested actuator position loop of Fig. 2.

B. Feed-Forward Law

The second control law is the first of two laws based on PI compensators. This law implements a feed-forward term for the final position of the actuators with a PI term to eliminate steady-state error. From (4), the predicted position of the actuator equals $R^T\boldsymbol{q}_d$, given a constant tendon length. Since the kinematic model may not be perfect, the PI compensator is needed to eliminate the errors. Referred to as the *Feed-Forward controller*, the commanded position follows.

$$\boldsymbol{x}_d = R^T\boldsymbol{q}_d - R^T\left(k_p\Delta\boldsymbol{q} + \int k_i\Delta\boldsymbol{q}dt\right) \tag{10}$$

The feed-forward term results in a fast rise-time, while the PI term results in zero steady-state error. Unfortunately, we will see that any non-zero PI gain unavoidably causes overshoot in the transient response. Such overshoot is quite undesirable as previously described. Accordingly, only low gains can be used, resulting in a step response with a fast rise time but slow settling time.

C. Pure PI Law

To avoid the overshoot problem of the previous controller, the third control law implements only a PI compensator. Referred to as the *Pure PI controller*, that relation follows.

$$\boldsymbol{x}_d = -R^T\left(k_p\Delta\boldsymbol{q} + \int k_i\Delta\boldsymbol{q}dt\right) \tag{11}$$

Compared to the previous law in (10), this law can be tuned to prevent overshoot and can thus achieve a faster settling time. As shown in the next section, a purely first-order response can be achieved by setting $k_i = ak_p$, where a^{-1} is the time-constant for the actuator position loop. In theory, this control law can thus be tuned to provide the same performance as the finite-difference controller. In practice, however, the two are not equal. Comparing (9) and (11), the two are identical except for the swapping of the position feedback for the integrator. Eliminating the inherent lag of the integrator allows the Finite-Difference controller to implement higher gains, and thus a faster response, before the onset of instability or overshoot.

IV. Transfer Function Analysis

To understand the performance of these controllers, consider the transfer function for each. The following analysis provides the theoretical validation for the claims of the previous section. Experimental validation will follow in the next section.

To start the analysis, consider the equation of motion for the finger.

$$M\ddot{\boldsymbol{q}} + \boldsymbol{\eta} = \boldsymbol{\tau} + \boldsymbol{\tau}_e \tag{12}$$

M is the joint-space inertia matrix. $\boldsymbol{\eta}$ represents the sum of the Coriolus, centripetal, gravitational, and frictional forces. And $\boldsymbol{\tau}_e$ represents the torques produced by external forces. For our purposes here, zero external forces are assumed.

These dynamics need to be expressed as a function of the actuator positions. Consider first the joint torques from (4) and (5):

$$\boldsymbol{\tau} = k_t R\left(x - R^T\boldsymbol{q}\right). \tag{13}$$

Substituting this back into the equation of motion,

$$\frac{1}{k_t}\left(M\ddot{\boldsymbol{q}} + \boldsymbol{\eta}\right) + RR^T\boldsymbol{q} = R\boldsymbol{x}. \tag{14}$$

The passive forces here are scaled by the inverse of the tendon stiffness. For dexterous fingers, this term becomes negligible due to both the low inertia of the fingers as well as the high stiffness of the tendons. For example, the R2 finger weighs only 0.125 lbs while the tendon stiffness is on the order of 400 lbs/in. We will thus neglect these passive forces. In addition, we will model the actuator with a first-order transfer function and a time-constant of a^{-1}. Accordingly, the relation can be expressed in the Laplace domain as follows, where $\boldsymbol{Q}(s)$ and $\boldsymbol{X}(s)$ represent the Laplace transforms of $\boldsymbol{q}(t)$ and $\boldsymbol{x}(t)$, respectively.

$$\begin{aligned} RR^T\boldsymbol{Q} &= R\boldsymbol{X} \\ &= \frac{a}{s+a}R\boldsymbol{X}_d. \end{aligned} \tag{15}$$

Consider now the transfer function for the **Finite-Difference controller** (9). Assuming the motion is limited to the range-space as expected, we can substitute $\boldsymbol{x} = R^T\boldsymbol{q}$.

$$\boldsymbol{x}_d = R^+R\left(R^T\boldsymbol{q}\right) - k_pR^T\Delta\boldsymbol{q} \tag{16}$$

Substituting the transform of this result into (15) produces the following transfer function for the control law, revealing a desirable first-order response.

$$\boldsymbol{Q} = \frac{ak_p}{s+ak_p}\boldsymbol{Q}_d \tag{17}$$

Preprint submitted to 2012 IEEE International Conference on Robotics and Automation. Received September 2, 2011.

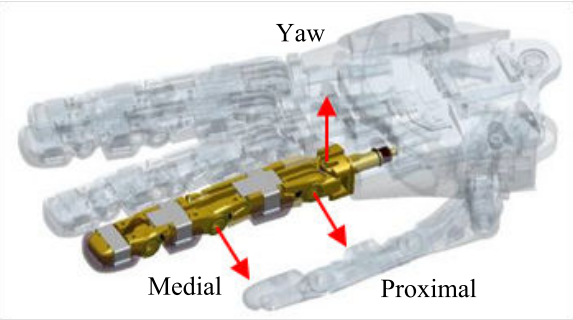

Fig. 3. A model of the Robonaut-2 index finger.

Next, consider the transfer function for the **Pure PI controller**. Substituting from (11), the following second-order function arises.

$$\boldsymbol{Q} = \frac{ak_p s + ak_i}{s^2 + a(1+k_p)s + ak_i}\boldsymbol{Q}_d \tag{18}$$

The system can be reduced to a first-order system by setting $k_i = ak_p$.

$$\boldsymbol{Q} = \frac{ak_p}{s + ak_p}\boldsymbol{Q}_d \tag{19}$$

This indicates that the Pure PI controller, in principle, can be tuned to produce the same exact result as the Finite-Difference controller. In practice, however, the implementation issues of communication delays and discrete processing rates favor the Finite-Difference controller.

Finally, consider the transfer function for the **Feed-Forward controller**.

$$\boldsymbol{Q} = \frac{a(1+k_p)s + ak_i}{s^2 + a(1+k_p)s + ak_i}\boldsymbol{Q}_d \tag{20}$$

As shown in the Appendix, this transfer function will necessarily overshoot given any non-zero gains. Of course, the overshoot can be slight and acceptable given relatively low k_i gains. With such low gains, however, the settling time will be considerably long. Not only will the system always overshoot in theory, but in practice, the overshoot is even greater due to the communication delays and actuator saturation effects of any implementation.

V. Experimental Results

A. Mechanical System

The control laws were tested on the index finger of the R2 humanoid hand [13]. A model of the finger is shown in Fig. 3. The finger has four tendons and three independent DOF's: a yaw, a proximal pitch, and a medial pitch. The yaw joint is perpendicular to both pitch joints, and the tendon mapping matrix follows.

$$R = \begin{bmatrix} 0.15 & 0.15 & -0.15 & -0.15 \\ 0.265 & -0.195 & 0.265 & -0.195 \\ 0 & 0 & 0.195 & -0.195 \end{bmatrix} \text{ in} \tag{21}$$

The system is actuated by brushless DC motors with planetary reduction gearheads. Ball-screws provide the linear conversion for the motor power, which is then transmitted to the finger through a tendon-conduit arrangement. This arrangement consists of a polymer cable threaded through a steel extension spring. Joint angles are sensed through Hall-effect sensors, and actuator positions are sensed by incremental encoders on the motors. An initialization routine tensions up the tendons and defines the encoder positions. All position commands for the joints are filtered by a trajectory generator to enforce a top speed.

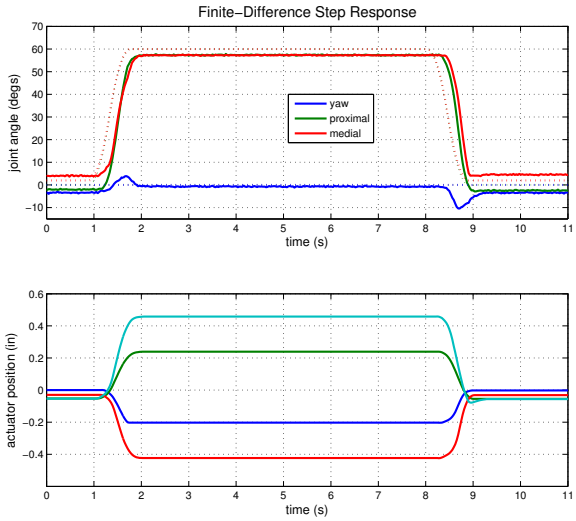

Fig. 4. A filtered step input (dotted line) is commanded to the joints. The controller produced a fast response with satisfactory steady-state error. No overshoot was exhibited in either the joint or actuator spaces, which was an important controller specification.

B. Finite-Difference Step Response

Two experiments were conducted with the Finite-Difference controller. The first experiment demonstrated the step response for a change in position. Starting at an initial joint position of $[0, 0, 2]$ degs, a step command of $[0, 60, 60]$ degs was commanded. The response is shown in Fig. 4. The joint moved quickly to the commanded position with the over-damped response desired, closing the steady-state error to about 3 degrees error. In the actuator space, the controller demonstrated the desired over-damped response as well.

C. Finite-Difference Disturbance Response

The second experiment tested the response of the Finite-Difference controller to external forces or disturbances. The initial version of the controller without the range-space projection, (8), failed under such conditions. The force created a joint error which the sliders attempted to compensate for as dictated by the kinematics. While the antagonist sliders *pulling* against the disturbance were restrained, the protagonist sliders *pushing* against it slid forward uninhibited. Since the joint errors were unaffected by this motion, the protagonist sliders continued to slide until they reached a hard stop or the force abated. This motion released the internal tension on the tendons, either reducing the passive

Preprint submitted to 2012 IEEE International Conference
on Robotics and Automation. Received September 2, 2011.

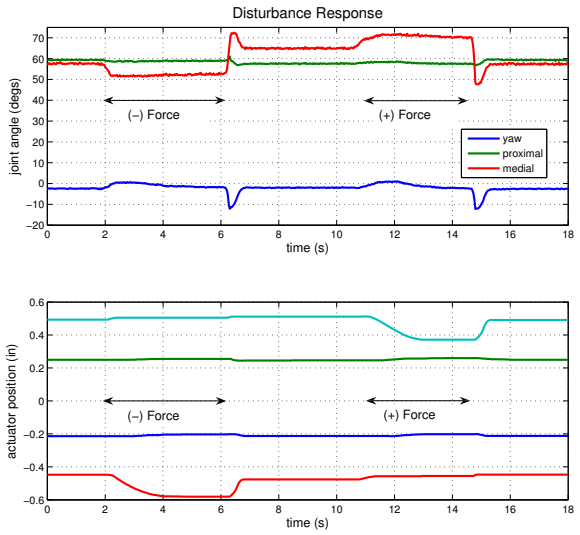

Fig. 5. A steady external force pushed the finger from time 2-6 s in one direction, and then from time 11-15 s in the opposite direction. Responding to the joint error, the protagonist tendon in each case slid forward until it was limited by the range-space constraint. Releasing the force, the tendons snapped back to position.

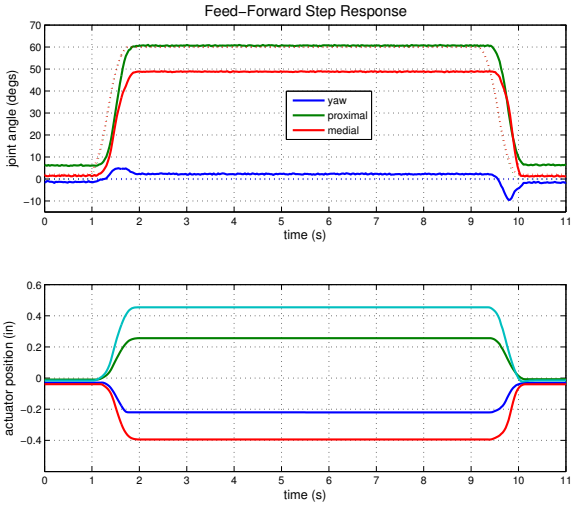

Fig. 6. The Feed-Forward controller with zero PI gains was applied here. The observed steady-state error of over 10 degs is due to errors in the kinematic model. The dotted line is the command to both joints.

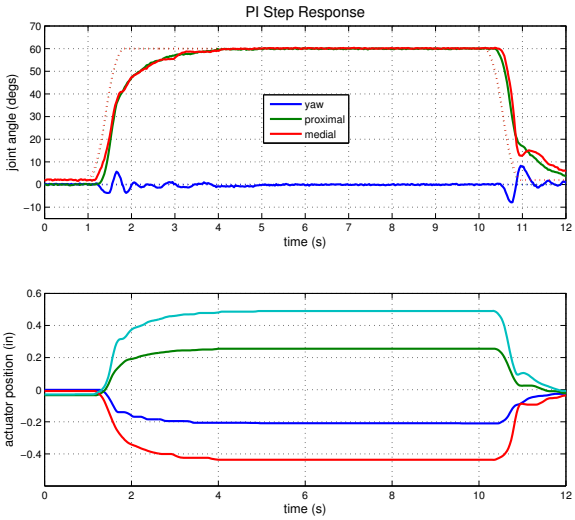

Fig. 7. The Pure PI controller eliminates the steady-state error. Its transient response, however, is significantly slower than the Finite-Difference controller.

stiffness of the joints or even introducing backlash due to the slack in the tendons.

The present controller solved that problem by using the range-space projection, as the following experiment demonstrated. A steady external force was applied to the finger tip causing a displacement in the medial joint. Shown in Fig. 5, a negative force was applied for a set time and then released, followed by a positive force that was applied for a similar time. Given the subsequent joint displacement, the protagonist tendon slid forward a limited distance, as dictated by the range-space constraint. Upon release of the force, the actuators snapped back to kinematically consistent positions. The controller is thus able to nominally preserve the internal tensions initially placed on the tendons.

D. Feed-Forward & PI Step Responses

The same step response experiment was conducted with the other two controllers. First, the Feed-Forward controller was applied without any feedback ($k_p = k_i = 0$). Using this controller, the system would respond at the maximum speed of the joints; however, significant steady-state errors ensued. A sample response is shown in Fig. 6, where an error of over 10 degs resulted. Throughout our experiments, the PI gains could be increased only slightly without producing significant overshoot. Applying such low gains would result in a system with the same fast rise time but a very slow settling time. A satisfactory balance between overshoot and settling time for our implementation could not be found.

Consider now the Pure PI controller for the same step experiment. The system was tuned to its fastest response resulting in gains of $k_i = 3$ and $k_p = 1$. Since the actuator time-constant was observed to be 0.2 seconds, k_p should theoretically equal 0.6 for the first-order response of (19). In practice, we were able to slightly increase k_p for a faster response, although some higher-order oscillations start to appear. The results of the experiment are shown in Fig. 7. This controller did the best job of eliminating the steady-state error without overshoot; however, its response is much slower than the Finite-Difference controller. Those higher-order oscillations can be eliminated by reducing k_p.

VI. Discussion

Selecting a position controller for a tendon-driven finger involves balancing tradeoffs between several factors. First, the performance should achieve both satisfactory speed and

accuracy. Second, it should eliminate overshoot in both the joint and actuator spaces. Finally, it should constrain the actuator motion to the range-space of R^T. In the absence of tension feedback, this is the only way to maintain the initial internal tension applied to the tendons.

These criteria arise from the accummulated experience of the authors with the humanoid hand of R2. In the context of this application, the *Finite-Difference* controller provides the best overall performance. Although it does not fully eliminate the steady-state error as the other controllers do, it is significantly faster with accuracy that is sufficient for many purposes. Its accuracy can be further increased in one of two ways. First, increasing k_p will reduce the error. If this produces overshoot, the trajectory generator can then be slowed down. Alternatively, a small integral term with a limited range can be added to close off the final error.

Applications that are concerned more with the steady-state rather than the transient behavior may better suit one of the other two controllers. The *Pure PI* controller will provide zero steady-state error without overshoot, but it will require the longest rise time. With a faster rise-time, the *Feed-Forward* controller can also eliminate the steady-state error; however, it will provide overshoot.

References

[1] J. Salisbury and J. Craig, "Articulated hands: Force control and kinematic issues," *International Journal of Robotics Research*, vol. 1, no. 1, pp. 4–17, 1982.

[2] S. Jacobsen, J. Wood, D. Knutti, and K. Biggers, "The Utah/MIT hand: Work in progress," *Intl. Journal of Robotic Research*, vol. 3, no. 4, pp. 21–50, 1984.

[3] Y. Lee, H. Choi, W. Chung, and Y. Youm, "Stiffness control of a coupled tendon-driven hand," *IEEE Control Systems Magazine*, vol. 14, no. 5, pp. 10–19, 1994.

[4] T. Wimbock, C. Ott, A. Albu-Schaffer, A. Kugi, and G. Hirzinger, "Impedance control for variable stiffness mechanisms with nonlinear joint coupling," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Nice, France, Sept. 2008.

[5] M. E. Abdallah, R. Platt, C. W. Wampler, and B. Hargrave, "Applied joint-space torque and stiffness control of tendon-driven fingers," in *IEEE Intl. Conf. on Humanoid Robots*, Nashville, TN, December 2010.

[6] S. Ma, S. Hirose, and H. Yoshinada, "Design and experiments for a coupled tendon-driven manipulator," *IEEE Control Systems Magazine*, vol. 13, no. 1, pp. 30–36, 1993.

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

[8] J. J. Lee, "Tendon-driven manipulators: Analysis, synthesis, and control," Ph.D. dissertation, Univ. of Maryland, College Park, MD, 1991.

[9] H. Kobayashi and R. Ozawa, "Adaptive neural network control of tendon-driven mechanisms with elastic tendons," *Automatica*, vol. 39, pp. 1509–1519, 2003.

[10] J. Mulero-Martinez, F. Garcia-Cordova, and J. Lopez-Coronado, "Position control based on static neural networks of anthropomorphic robotic fingers," *Advances in Neural Networks*, vol. 3972, pp. 1188–1197, 2006.

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

[12] H. Kobayashi, K. Hyodo, and D. Ogane, "On tendon-driven robotic mechanisms with redundant tendons," *International Journal of Robotics Research*, vol. 17, no. 5, pp. 561–571, May 1998.

[13] M. Diftler, J. Mehling, M. Abdallah, N. Radford, L. Bridgwater, A. Sanders, R. Askew, D. Linn, J. Yamokoski, F. Permenter, B. Hargrave, R. Platt, R. Savely, and R. Ambrose, "Robonaut 2 — the first humanoid robot in space," in *IEEE Intl. Conference on Robotics and Automation (ICRA)*, May 2011.

Appendix

Proof of Overshoot Claim

In section IV, the transfer function for the Feed-Forward controller (20) is presented with the claim that it must always overshoot. To validate that claim, consider the following analysis. That transfer function can be expressed in the following general form, where c_1 and c_2 are positive constants.

$$\boldsymbol{Q} = \frac{c_1 s + c_2}{s^2 + c_1 s + c_2}\boldsymbol{Q}_d \tag{22}$$

This system will necessarily overshoot, regardless of whether the denominator is under-, critically-, or over-damped. If the system overshoots when the denominator is overdamped, then it will necessarily overshoot in the other damping cases. Hence, it will suffice us to show that the system will always overshoot given overdamped poles.

Given the assumption of an overdamped plant, the system has two distinct real poles denoted as a and b.

$$\begin{aligned} a &= -\frac{1}{2}c_1 + \frac{1}{2}\sqrt{c_1^2 - 4c_2} \\ b &= -\frac{1}{2}c_1 - \frac{1}{2}\sqrt{c_1^2 - 4c_2} \end{aligned} \tag{23}$$

Hence, $c_1^2 > 4c_2$ and $b < a < 0$. Since motion of the multiple joints are decoupled, we can consider a single joint independently. Given a step input of 1,

$$\begin{aligned} Q &= \frac{c_1 s + c_2}{s(s-a)(s-b)} \\ &= \frac{ab - (a+b)s}{s(s-a)(s-b)}, \end{aligned} \tag{24}$$

where $c_1 = -a - b$, and $c_2 = ab$. After expanding this expression using the partial fractions technique, the inverse Laplace reveals the following response in the time domain.

$$q(t) = 1 + \left(\frac{a}{b-a}\right)e^{at} - \left(\frac{b}{b-a}\right)e^{bt} \tag{25}$$

This step response overshoots if its maximum is greater than 1. Solving for the critical point, the peak value can be expressed as a function of t_{max}, the time at which it occurs.

$$q(t_{max}) = 1 + \frac{a}{b}e^{at_{max}} \tag{26}$$

This value is always greater than one, indicating that this overdamped system must always overshoot the input. Since the system will always overshoot even when the poles are overdamped, it will exhibit overshoot much more so under the other possible scenarios.

Preprint submitted to 2012 IEEE International Conference on Robotics and Automation. Received September 2, 2011.