Position Control of Tendon-Driven Fingers

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

***Abstract*— Conventionally, tendon-driven manipulators implement some force control scheme based on tension feedback. This feedback allows the system to ensure that the tendons are maintained taut with proper levels of tensioning at all times. Occasionally, whether it is due to the lack of tension feedback or the inability to implement sufficiently high stiffnesses, a position control scheme is needed. This work compares three position controllers for tendon-driven manipulators. 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 feedback, the controller nominally maintains the internal tension on the tendons by implementing a two-tier architecture with a range-space constraint. 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. Occasionally, however, a position-based control law is desired. In cases where tension feedback is not available or the control bandwidth does not allow for sufficiently high joint stiffnesses, a pure position controller is needed. Many researchers have presented tension-based controllers for tendon-driven fingers [1]–[6], while none (to our knowledge) have presented a position-based controller.

The challenge is to develop a controller that can achieve the desired performance while maintaining suitable tensions on the tendons. According to the needs of our implementation, three criteria arise. 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. Finally, 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. Assuming no tension sensing is available, this objective can be nominally achieved by eliminating the null-space motion amongst the tendons.

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.

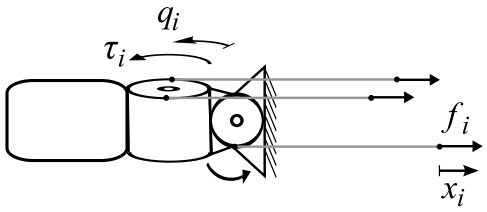

Fig. 1. Schematic of a simple finger with tendons.

This work compares several position controllers for tendon-driven manipulators. A new controller is presented that achieves superior transient performance than equivalent proportional-integral (PI) based controllers. This controller implements a two-tier architecture 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 humanoid hand.

The paper starts with an introduction to the kinematics of the finger. The control laws are then presented and their transfer functions are analyzed. Finally, the experimental results from Robonaut-2 are presented and discussed.

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 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; it contains the joint radii data mapping tendon tensions to joint torques. For the system to be tendon controllable, R must be full row rank and there must exist an all-positive column matrix, $\boldsymbol{w}$, such that $R\boldsymbol{w} = 0$ [7]. 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. Throughout this work, bold symbols represent column matrices.

This same R expresses the relationship between the tendon and joint velocities. Based on the principle of virtual work, the contribution of the joint motion to the tendon velocity equals $R^T\dot{\boldsymbol{q}}$. Assuming a constant R, the net displacement of the tendons is a sum of the joint contribution plus the change in length, $\boldsymbol{l}$, of the tendon. The symbol Δ denotes the difference in a respective variable from one configuration to another.

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

We will model the tendon as a linear spring and assume it remains taut. We will also assume that the tendons all have the same stiffness value, k_t, since the difference in tendon lengths is not sufficient to warrant a significant difference in stiffness. The following analysis relates $\Delta\boldsymbol{l}$ to the change in tendon tensions and then joint torques.

$$\begin{aligned} \Delta\boldsymbol{f} &= k_t\Delta\boldsymbol{l} \\ \Delta\boldsymbol{\tau} &= R\Delta\boldsymbol{f} \\ &= k_tR\Delta\boldsymbol{l} \end{aligned} \tag{4}$$

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

$$\begin{aligned} \Delta\boldsymbol{l} &= \tfrac{1}{k_t}R^+\Delta\boldsymbol{\tau} + \Delta\boldsymbol{l}_{int} \\ \Delta\boldsymbol{l}_{int} &\doteq (I - R^+R)\,\boldsymbol{\delta}, \end{aligned} \tag{5}$$

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 side of (5) represents the change in length due to external loads, while the second term represents the change in length due to the internal tensions. This results in the following final relation for the tendon displacement from (3).

$$\Delta\boldsymbol{x} = R^T\Delta\boldsymbol{q} + \frac{1}{k_t}R^+\Delta\boldsymbol{\tau} + \Delta\boldsymbol{l}_{int} \tag{6}$$

In the absence of tension feedback, the only way to keep the internal tension constant is to eliminate $\Delta\boldsymbol{l}_{int}$. This implies that the actuator motion, $\Delta\boldsymbol{x}$, 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.

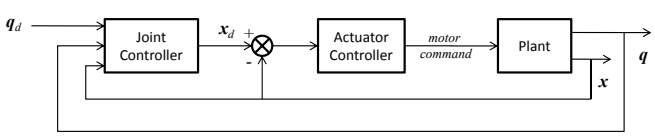

Fig. 2. The control architecture. The lower, actuator position loop allows the Finite-Difference controller to actively constrain the actuator motion to the range-space.

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 controlling joint positions and a lower loop controlling actuator positions. Shown in Fig. 2, the upper loop passes actuator position commands down to the lower loop. We assume here that the lower loop has been tuned to maximize performance with a first-order response behavior (i.e. without overshoot). 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.

A. Finite-Difference Law

The first control law implements a discrete version of a velocity controller, where the current positions of the actuators are continuously fed back and combined with a delta vector based on the joint errors. We thus refer to it as the *Finite-Difference* controller. Based on the kinematic relation in (6), the commanded position is:

$$\boldsymbol{x}_d = \boldsymbol{x} - k_pR^T\Delta\boldsymbol{q}, \tag{7}$$

where $\Delta\boldsymbol{q}$ here represents the joint position error $(\boldsymbol{q} - \boldsymbol{q}_d)$, and k_p is a scalar, constant gain. This control law zeros the null-space displacement term of (6), since none is desired, as well as the external torque term. The external torque term represents the stretch in the tendons due to an external torque. The feedback controller will makeup for this disturbance.

This controller works well in producing a fast response that closes the steady-state error and maintains an over-damped behavior. The problem is that it does not actively constrain the actuator positions to the range-space. Although the delta vector commanded to the actuators lies in the range-space, disturbances or actuator saturation effects can cause the actual positions to deviate from the range-space. This effect is exacerbated by the fact that the tendons cannot resist compression. Consider thus 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 ($\Delta\boldsymbol{l}_{int}$) 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, as shown in Appendix A. Noting that R^+R is symmetric, the new commanded position follows.

$$\begin{aligned} \boldsymbol{x}_d &= R^+R\Big(\boldsymbol{x} - k_pR^T\Delta\boldsymbol{q}\Big) \\ &= R^+R\boldsymbol{x} - k_pR^T\Delta\boldsymbol{q} \end{aligned} \tag{8}$$

This results in our final *Finite-Difference* control law. This controller produces the same transient and steady-state performance as (7); however, it resists the internal motion even when disturbed. Note that the initial relation in (7) could have been implemented with a single-loop controller, setting the motor command proportional to $R^T\Delta\boldsymbol{q}$. The range-space constraint of (8), however, requires the two-tiered hierarchy 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. If the system is initialized such that the initial positions, $\boldsymbol{x}$ and $\boldsymbol{q}$, are defined as zero, then the actuator positions that matches the desired joint positions without changing the length of the tendons are given by $R^T\boldsymbol{q}_d$. 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.

$$\boldsymbol{x}_d = R^T\boldsymbol{q}_d - R^T\left(k_p\Delta\boldsymbol{q} + \int k_i\Delta\boldsymbol{q}dt\right) \tag{9}$$

The feed-forward term results in a fast rise-time, while the PI term results in zero steady-state error. Unfortunately, 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.

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

Compared to the previous law in (9), 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 (8) and (10), the two are identical except that the position feedback of the first replaces the integral error term of the second. Effectively, a continuous time integral is thus used instead of the discrete and delayed time integral. This allows the Finite-Difference controller to implement higher gains, and thus a faster response, without instability or overshoot.

IV. Transfer Function Analysis

To understand the performance of these controllers, consider the transfer function for each. This analysis provides the theoretical validation for the claims of the previous section. The experimental validation follows in the next section.

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

$$M\ddot{\boldsymbol{q}} + \boldsymbol{\eta} = \boldsymbol{\tau} + \boldsymbol{\tau}_e \tag{11}$$

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.

If the system is initialized so that the initial positions and lengths are defined as zero, $\boldsymbol{x}_0 = \boldsymbol{q}_0 = \boldsymbol{l}_0 = 0$, then (3) gives us the following relationship for the actuator position.

$$\boldsymbol{x} = R^T\boldsymbol{q} + \boldsymbol{l} \tag{12}$$

Since $\boldsymbol{f} = k_t\boldsymbol{l}$, the joint torques thus become:

$$\begin{aligned} \boldsymbol{\tau} &= R\boldsymbol{f} \\ &= k_tR\left(x - R^T\boldsymbol{q}\right). \end{aligned} \tag{13}$$

Substituting back into the equation of motion,

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

Since the passive dynamics are scaled by the inverse of the tendon stiffness, which is a relatively large value, their contribution is not significant. This is further supported by the assumption that the manipulator inertia and first-order dynamics are insubstantial, as is commonly the case for dexterous fingers. We will thus neglect the effect of these dynamics. 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. Let $\boldsymbol{Q}(s)$ 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** (8). 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.

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

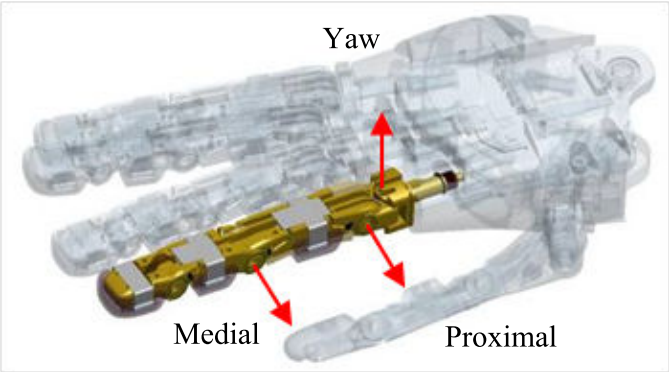

Fig. 3. A model of the Robonaut-2 robotic hand.

Next, consider the transfer function for the **Pure PI controller**. Substituting from (10), 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 Appendix B, 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 theoretically always overshoot, but in practice, the overshoot is heightened by the communication delays and actuator saturation effects of any implementation.

V. Experimental Results

A. Mechanical System

The control laws were tested on the primary fingers of the Robonaut-2 humanoid hand. A model of the hand 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.

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

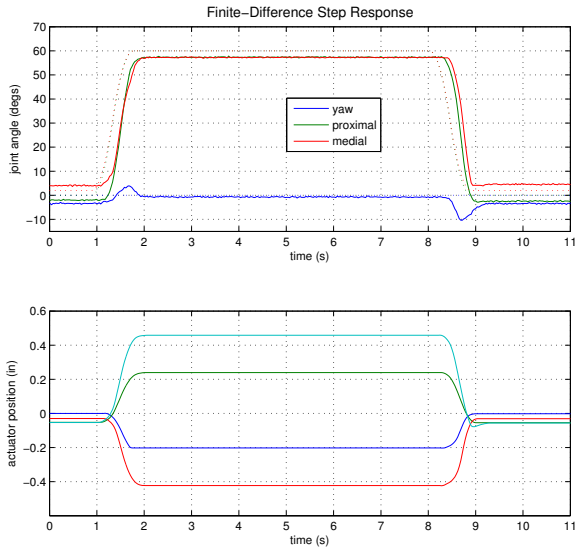

Fig. 4. A filtered step input (dotted line) is commanded to the joints. The controller produced a fast response with satisfactory steady-state error. No overshoot was exhibited in either the joint or actuator spaces, which was an important controller specification.

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. The processor operates at a rate of 350 Hz.

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 through a trajectory generator. The trajectory generator was tuned to provide the fastest stable response. 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. The yaw joint exhibited some coupled disturbance in the transience. This joint contains smaller radii making it more poorly conditioned than the others. 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 version of the controller without the range-space projection (7) 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*

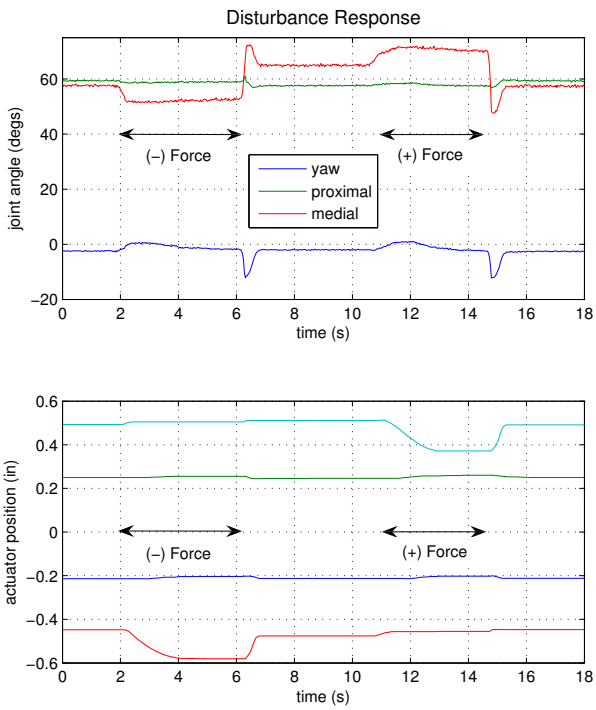

Fig. 5. A steady external force pushed the finger from time 2-6 s in one direction, and then from time 11-15 s in the opposite direction. Responding to the joint error, the protagonist tendon in each case slid forward until it is was limited by the range-space constraint. Releasing the force, the tendons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

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 was removed. This motion released the internal tension on the tendons, either reducing the passive 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 set time before being released. 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

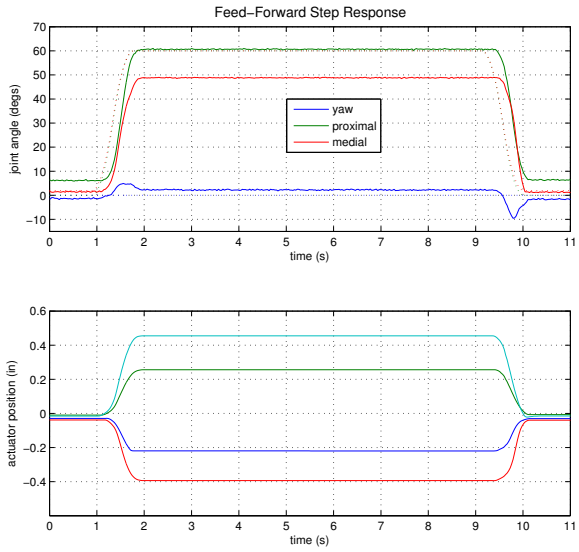

Fig. 6. The Feed-Forward controller with zero PI gains was applied here. The observed steady-state error of over 10 degs is due to errors in the kinematic model.

speed of the actuator; 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 could not be achieved.

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, however, we were able to increase k_p and achieve a faster response. 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. Note, the higher-order oscillations that are already starting to appear can be eliminated by reducing k_p.

VI. Discussion

Selecting a position controller for a tendon-driven manipulator involves balancing tradeoffs between several factors. First, the performance needs to achieve both satisfactory speed and accuracy. Second, it needs to eliminate overshoot in both the joint and actuator spaces. Finally, it needs to 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.

According to these requirements, the *Finite-Difference* controller provided the most suitable solution and is currently implemented on Robonaut-2. Although it does not fully

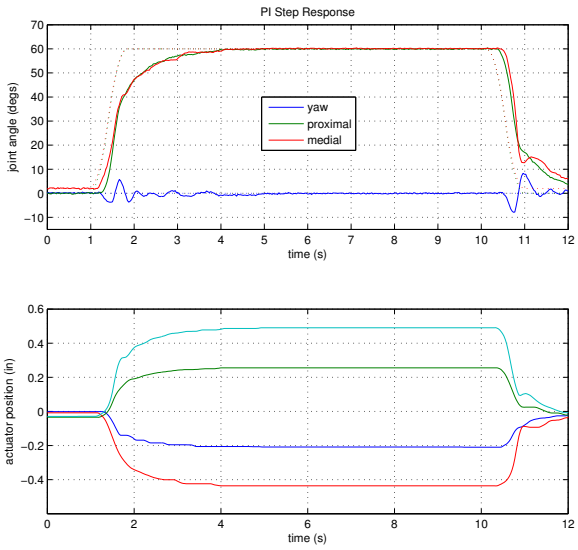

Fig. 7. The Pure PI controller eliminates the steady-state error. Its transient response, however, is significantly slower than the Finite-Difference controller.

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.

In addition, the Feed-Forward controller can be modified to implement a lag compensator instead of the PI term. This lag compensator represents a compromise with respect to the PI: it can reduce the overshoot, however it does so at the expense of the steady-state error. Furthermore, it will still not achieve the speeds of the Finite-Difference controller. Analysis of this control law is available in Appendix C.

APPENDIX

PROOF OF OVERSHOOT CLAIM

In section IV, the transfer function for the Feed-Forward controller (20) is presented. The section claims that this transfer function will always overshoot. That claim is validated here.

The 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 has been decoupled, we can consider a single joint independently. Given a step input of 1, the system can now be expressed as follows.

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

where $-c_1 = a + b$, and $c_2 = ab$. This expression can now be expanded using the partial fraction technique.

$$Q = \frac{1}{s} + \left(\frac{a}{b-a}\right)\frac{1}{s-a} + \left(\frac{b}{a-b}\right)\frac{1}{s-b} \tag{25}$$

The step response in the time domain can be found from the inverse Laplace transform of this expression.

$$q(t) = 1 + \left(\frac{a}{b-a}\right)e^{at} - \left(\frac{b}{b-a}\right)e^{bt} \tag{26}$$

This step response overshoots if its maximum is greater than 1. To find the critical point,

$$\begin{aligned} 0 &= \frac{dq}{dt} \\ &= a^2 e^{at_{max}} - b^2 e^{bt_{max}}. \end{aligned} \tag{27}$$

The peak value for the step response can now be found, where t_{max} is the time at which it occurs. Solving from (27), the peak value can be expressed as follows.

$$\begin{aligned} q(t_{max}) &= 1 + \left(\frac{a}{b-a}\right)e^{at_{max}} - \left(\frac{b}{b-a}\right)e^{bt_{max}} \\ &= 1 + \frac{a}{b}e^{at_{max}} \end{aligned} \tag{28}$$

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.

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] G. P. Starr, "Experiments in assembly using a dexterous hand," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 3, pp. 342–347, June 1990.
[4] 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.
[5] 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.
[6] 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.
[7] 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.