Anatomically Correct Testbed Hand Control: Muscle and Joint Control Strategies

Ashish D. Deshpande, Jonathan Ko, Dieter Fox and Yoky Matsuoka

{add, jonko, fox, yoky}@cs.washington.edu
University of Washington Seattle, WA, USA

Abstract— Human hands are capable of many dexterous grasping and manipulation tasks. To understand human levels of dexterity and to achieve it with robotic hands, we constructed an anatomically correct testbed (ACT) hand which allows for the investigation of the biomechanical features and neural control strategies of the human hand. This paper focuses on developing control strategies for the index finger motion of the ACT Hand. A direct muscle position control and a force-optimized joint control are implemented as building blocks and tools for comparisons with future biological control approaches. We show how Gaussian process regression techniques can be used to determine the relationships between the muscle and joint motions in both controllers. Our experiments demonstrate that the direct muscle position controller allows for accurate and fast position tracking, while the force-optimized joint controller allows for exploitation of actuation redundancy in the finger critical for this redundant system. Furthermore, a comparison between Gaussian processes and least squares regression method shows that Gaussian processes provide better parameter estimation and tracking performance. This first control investigation on the ACT hand opens doors to implement biological strategies observed in humans and achieve the ultimate human-level dexterity.

I. Introduction

Human hands are capable of many dextrous grasping and manipulation tasks. For example, with proper training and practice we can learn to play musical instruments with our hands, use chopsticks, and perform daily tasks such as cooking and writing. However, none of the existing robotic hands are capable of demonstrating human levels of dexterity. Dexterity of movements is achieved in the human hand in part due to the biomechanics of the hand and in part due to the neuromuscular control. To be able to understand and analyze human levels of dexterity, and to achieve it with robotic hands we need to mimic the biomechanics as well as the neuromuscular controls in the robotic hands.

We have recently completed the assembly of an anatomical robotic hand, called Anatomically Correct Testbed (ACT) Hand (Figure 1), with the following three research goals in mind: (1) an experimental testbed to investigate the biomechanics and neural control of human hand movements, (2) a physical prototype to test new surgical techniques for impaired hands and (3) a tele-manipulator for precision tele-operation and prosthetics. Unlike other anthropomorphic robotic hands [7], [4], [5], [15], [25], the ACT Hand is anatomically correct. The local nonlinear interactions between the muscle excursions and joint movements are mimicked in the ACT Hand by bone shapes that match human
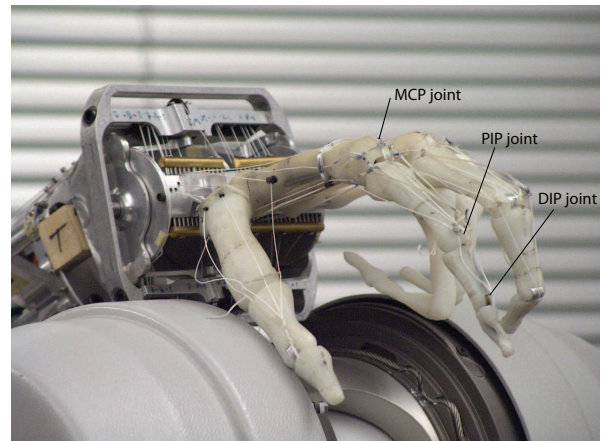

Fig. 1. Anatomically Correct Test-bed (ACT) Hand. The skeletal structure and tendon routing in each finger are anatomical. The hand is controlled by DC motors that are located in the forearm and connected the tendons.

bones and by the properties of the tendon hood that connects the actuators to the finger bones [35], [34], [11]. The anatomical properties allow us to implement neuromuscular control strategies without having to compensate, in control software, for the differences in robotic hand structure and human hand anatomy.

The next step is the development of control strategies for the ACT Hand. We are interested in taking two distinct approaches for the controller design for the ACT Hand. One is to treat the ACT Hand strictly as a redundant robotic system and control the motors to achieve the desired finger motions and forces. Another approach is to implement the neuromuscular controls observed in the human hands as closely as possible in the ACT Hand. Our ultimate goal is to investigate both of these approaches, and understand the relative advantages and disadvantages. In this paper we present the control strategies based on the robotics perspective and discuss the implications for developing biological controls.

With six to nine muscles devoted to move each finger having four to five degrees of freedom, the human hand forms a system with actuation redundancy. The web of tendons connecting the hand muscles to the bones is arranged such that most of the hand muscles are multi-articulate (i.e. each muscle controls more than one joint). The Central Nervous System (CNS) likely takes advantage of these biomechanical properties in developing neuromuscular control strategies to achieve manipulation dexterity [33], [22].

The biological data has shown that position as well as force control are used during grasping tasks [32]. The passive

biomechanics of the hand is critical in achieving the stable, dextrous movements and manipulation [17]. It is proposed that humans resolve the actuation redundancy by modulating the muscle forces in synergy either for simplicity of the solution or to save energy [32], [2]. The joint stiffness and end point impedance are controlled actively to avoid errors [21] When we design the biological controller, we intend to incorporate these facts found in the neuroscience literature.

A great number of control schemes exist for the position as well as the force control of robots with actuation redundancy [8], [23], [29]. Nakanishi et al. have presented a summary of the existing position and force control strategies for operational space control of a robotic arm [24]. The existing strategies exploit the redundancy either to optimize forces [14] and energy [9] or to satisfy contact constraints [12]. There exist many examples of human-like robot controls. Suzuki and Mayahara [31], and Tahara et al [30]. present examples of position control strategies for an anthropomorphic robotic shoulder joint and robotic arm, respectively, which are actuated by linear muscles.

Researchers have also developed control strategies for robotic hands that possess varying degree of anthropography. Zollo et al. [37] have presented controls for position control of finger which is under-actuated. Lee et al. [19] and Kim et al. [16] have presented stiffness control strategies for a single joint of a hand and for five fingers in simulation, respectively. Matsuoka and Afshar [20] estimate and compare biological and robotic solutions for the muscle forces for a given set of dynamic joint movements. Redundancy resolution for hand motion and force control for grasping has also been presented in [28], [36].

As a first set of control strategies for the ACT hand, we implemented two controllers: the *direct muscle controller* under which we control the individual muscle positions to achieve the desired finger position and the *force optimized joint controllers* under which we control muscle forces determined from joint torques necessary to achieve the desired finger positions. Both are reasonable approaches from the robotics perspective for the ACT hand control, and could likely act as building blocks for the biological control investigation.

Key components of both controllers are the mapping from joint angles to muscle lengths, and the derivative of this mapping, the moment arm matrix [11]. Considering the complex relationships between the ACT Hand joints and muscles we implemented Gaussian processes to determine the joint to muscle mappings. Gaussian processes are non-parametric regression techniques that are extremely flexible and robust, thereby providing the modeling capabilities necessary for the ACT Hand [27]. Specifically, we show how Gaussian process models can be used to learn the joint angle to muscle length mappings and the moment arm matrix based on example data. Our experimental evaluation demonstrates the benefits of using Gaussian processes for modeling the ACT Hand. We also evaluate and discuss the relative advantages and disadvantages of the two types of controllers, and the implications for designing a biologically inspired control

schemes. These investigations were carried out on the index finger of the ACT hand.

II. ACT Index Finger Description

In this section we give a brief overview of the biomechanics of the ACT index finger. More details can be found in our earlier publications [35], [34], [11]. The finger skeletal structure and the tendon routing mimic the human counterparts. All four bones have human bone contours which provide accurate tendon guidance. There are three joints with a total of four degrees of freedom, namely from proximal, MCP (metacarpal phalange) Ab-Ad, MCP flex, PIP (proximal inter-phalange), and DIP (distal inter-phalange). A human finger is controlled by 3 intrinsic muscles, namely, PI (palmar interossei) , RI (radial interossei) and LUM (lumbrical), and 4 extrinsic muscles, namely, EDC (extensor digitorum communis), EIP (extensor indicis profundus), FDS (flexor digitorum superficialis) and FDP (flexor digitorum profundus) [6]. In the ACT Hand, six of the seven muscles are realized by a brush-less DC motor. The muscle EDC is equivalent to EIP for single finger control [6]. Tendons are connected to the motor shafts after passing over pulleys. Each motor is connected to a miniature controller (Barrett Technology Inc. [3]) with an embedded photo-sensor and an encoder wheel (with 114 ticks/deg). The motor position can be provided at more than 500Hz per motor by a RTAI (Real Time Application Interface) Linux machine. The force generated in a tendon, $F_m(i)$, is directly proportional to the motor current $I_m(i)$:

$$F_m(i) = \frac{K_\tau}{r_m} * I_m(i), \tag{1}$$

where, K_τ is the motor torque constant and r_m is the radius of the motor shaft.

III. Method

A. Data Collection for Joint-Muscle Mappings

The two sources of data for the ACT index finger are the Vicon motion tracking system for joint angle determination, and the motors for muscle excursion and force data. The Vicon motion capture system uses cameras to track small reflective markers placed on the ACT index finger. Using information on camera placement and a model of the ACT index finger and marker locations, the system reconstructs the angles for each of the four joints at 120Hz.

Because PI muscle primarily abduct/adduct the MCP joint (and no other joints), the timing correlation between the PI muscle excursion data and MCP joint Vicon data were used to synchronize between the two systems. The ACT index finger was moved around passively through the full range of finger motions multiple times.

B. Moment Arm Matrix of the ACT Index Finger

In our previous work we have determined that, similar to the human hand, the moment arm matrix for the ACT Hand is non-constant and has nonlinear dependencies on the finger configuration [11]. In the case of the ACT index finger, the

moment arm is defined by a matrix R of dimension 6×4. The finger DOFs and the muscle excursions can be defined to be related by functions f_i as follows:

$$l_i = f_i(\underline{\theta}) \quad i = 1, ..., 6, \tag{2}$$

where $\underline{l}$ is the vector of muscle excursions ($\underline{l} = [l_1, l_2, l_3, l_4, l_5, l_6]^T$) and $\underline{\theta}$ is a vector of finger joint angles ($\underline{\theta} = [\theta_1, \theta_2, \theta_3, \theta_4]^T$). Then the moment arm is defined as:

$$\underline{\dot{l}} = R\underline{\dot{\theta}} \tag{3}$$

where,

$$R_{ij}(\theta) = \frac{\partial l_i}{\partial \theta_j} = \frac{\partial f_i}{\partial \theta_j} \quad i = 1, ..., 6 \text{ and } j = 1, ..., 4. \tag{4}$$

In essence, R is simply the gradient of the function $f_i(\theta)$.

C. Mapping Between Joint Angles and Muscle Lengths

To get the mapping for the function $f_i(\theta)$, we use a machine learning technique called Gaussian process (GP) regression and standard least squares regression (LSR) as baseline.

1) Least Squares Regression: The least squares technique seeks to find the coefficients of a 3rd degree polynomial which minimizes the l2-norm error. The variables in this case are the joint angles, the output is muscle length. We learn a separate set of coefficients for each muscle. The parameter optimization is solved by QR decomposition. In general, a third degree polynomial would have 35 terms, however, after careful analysis of the data, some terms were removed to improve the performance of the regression. We also investigated polynomials of degrees higher than three, but found that these lead to increased errors due to overfitting. Overall, polynomials have the advantage of being relatively simple parametric functions while still producing a good fit. Our analysis of mapping results proved that polynomial functions produce a fit with low overall mean error (< 1mm) [10]. The gradient of the learned polynomial can be easily found by taking partial derivatives of the parametric function. This gradient represents the moment arm.

2) Gaussian Process Regression: Gaussian processes (GP) are non-parametric techniques for learning regression functions from sample data [27]. GPs have been used with great success in robotics applications such as reinforcement learning [13] and Bayesian filtering [18]. Assume that we have n d-dimensional input vectors: $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n]$. A GP defines a zero-mean, Gaussian prior distribution over the outputs $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n]$ at these values:

$$p(\mathbf{Y} \mid \mathbf{X}) = \mathcal{N}(\mathbf{Y}; 0, \mathbf{K} + \sigma_n^2\mathbf{I}), \tag{5}$$

The covariance of this Gaussian distribution is defined via a kernel matrix, $\mathbf{K}$, and a diagonal matrix with elements σ_n^2 that represent zero-mean, white output noise. The elements of the $n \times n$ kernel matrix $\mathbf{K}$ are specified by a kernel function over the input values: $\mathbf{K}[i, j] = k(\mathbf{x}_i, \mathbf{x}_j)$. By interpreting the kernel function as a distance measure, we see that if

points $\mathbf{x}_i$ and $\mathbf{x}_j$ are close in the input space, then their output values $\mathbf{y}_i$ and $\mathbf{y}_j$ are highly correlated.

The specific choice of the kernel function k depends on the application, the most widely used being the squared exponential, or Gaussian, kernel:

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \, e^{-\frac{1}{2}(\mathbf{x}-\mathbf{x}')W(\mathbf{x}-\mathbf{x}')^T} \tag{6}$$

The kernel function is parameterized by W and σ_f. The diagonal matrix W defines the length scales of the process, which reflect the relative smoothness of the process along the different input dimensions and σ_f^2 is the signal variance.

In our application, we wish to learn a mapping from joint angles to muscle lengths. This is done by conditioning Equation 5 on training data $D = \langle \mathbf{X}, \mathbf{Y} \rangle$, where $\mathbf{X}$ contains joint angles, $\underline{\theta}$, and $\mathbf{Y}$ the corresponding muscle lengths, $\underline{l}$, collected during the training phase. It can be shown that conditioning on training data and a test input $\mathbf{x}_*$ results in the following Gaussian predictive distribution over the output:

$$p(\mathbf{y}_* \mid \mathbf{x}_*, D) = \mathcal{N}(\mathbf{y}_*; \mathrm{GP}_\mu(\mathbf{x}_*, D), \mathrm{GP}_\Sigma(\mathbf{x}_*, D)) \tag{7}$$

with mean

$$\mathrm{GP}_\mu(\mathbf{x}_*, D) = \mathbf{k}_*^T[K + \sigma_n^2 I]^{-1}\mathbf{y} \tag{8}$$

and variance

$$\mathrm{GP}_\Sigma(\mathbf{x}_*, D) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T \left[K + \sigma_n^2 I\right]^{-1} \mathbf{k}_*. \tag{9}$$

Here, $\mathbf{k}_*$ is a vector of kernel values between $\mathbf{x}_*$ and the training inputs $\mathbf{X}$: $\mathbf{k}_*[i] = k(\mathbf{x}_*, \mathbf{x}_i)$. Note that the prediction uncertainty, captured by the variance GP_Σ, depends on both the process noise and the correlation between the test input and the training inputs. Here, we do not consider the prediction uncertainty but focus on the GP mean prediction (8). As can be seen, the complexity of the GP mean prediction is linear in the number of training points. In our experiments we found that GP prediction is efficient enough for finger control. If needed, however, more efficient predictions can be generated by sparsification of the GP [26].

The hyperparameters $\boldsymbol{\theta}_y$ of the GP are given by the parameters of the kernel function and the output noise: $\boldsymbol{\theta}_y = \langle \sigma_n, W, \sigma_f \rangle$. They can be determined by maximizing the log likelihood of the training data:

$$\boldsymbol{\theta}_y^* = \underset{\boldsymbol{\theta}_y}{\mathrm{argmax}} \, \log \, p(\mathbf{Y} \mid \mathbf{X}, \boldsymbol{\theta}_y). \tag{10}$$

This optimization can be performed efficiently using techniques such as conjugate gradient descent [27]. Finally, the gradient of the GP mean prediction function (8) gives the moment arm. For more details on GP the reader is referred to [18].

D. Controller Design

In general the control problem for the ACT Hand is to design controllers for the muscles to generate the desired hand posture, and forces and stiffness that affect the task. The task could be specified in a variety of ways, eg. in terms of end point positions joint stiffness or joint dynamics. For this paper, as a first try, we assume that we have desired joint angles.
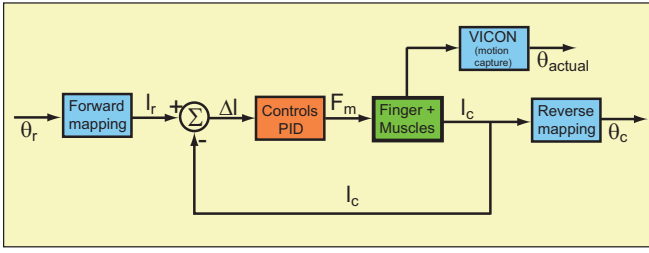
Fig. 2. Schematic for the direct muscle control. In each time step, the desired joint angles, $\underline{\theta}_r$, are mapped to desired muscle lengths, $\underline{l}_r$. The difference from the current muscle lengths, $\underline{l}_c$, is fed into the PID controller to determine the forces that directly control the muscles. A VICON tracking system provides ground truth joint angles, θ_{actual}, for evaluation purposes.

1) Direct Muscle Control: Figure 2 shows the schematics for the direct muscle control. Under this scheme, for a desired angle, $\underline{\theta}_r$, we determine the desired muscle lengths $\underline{l}_r$ using the mapping:

$$\underline{l}_r = f(\underline{\theta}_r). \quad (11)$$

The forward mapping function f is determined either by the least squares method or by using GPs. Using the encoder reading on each muscle the errors between the desired and the current positions, $\underline{l}_c$ is determined as:

$$\Delta \underline{l} = (\underline{l}_r - \underline{l}_c). \quad (12)$$

Based on the errors we set up a PID controller for each muscle to track the desired. Thus the muscle forces to applied based on the muscle positions errors are:

$$F_m = K_{Pl} * \Delta \underline{l} + K_{Dl} * \Delta \underline{\dot{l}} + K_{Il} * \int_t \Delta \underline{l}. \quad (13)$$

where K_{Pl}, K_{Dl} and K_{Il} are the PID gains. The motor currents are determined by using Equation 1.

This strategy is simple to implement as muscles are tracked independently. However, errors in the direction of muscle extension are not correctable with direct muscle control as muscles can only pull. Also, generating specific finger tip forces, necessary in grasping and manipulation, is difficult with this scheme.

2) Force-optimized Joint Control: Figure 3 shows the schematics for the force optimized joint torque control. This scheme requires the determination of the current joint angles. Since we do not have joint angle sensors (consistent with hand biology), we generate the reverse mapping to find the current joint angles using both the least squares method and the Gaussian processes given the current muscle positions

$$\underline{\theta}_c = g(\underline{l}_c). \quad (14)$$

The error in joint angles between the desired joint trajectory $\underline{\theta}_r$ and current joint angles $\underline{\theta}_c$ is then determined by

$$\Delta \underline{\theta} = \underline{\theta}_r - \underline{\theta}_c \quad (15)$$

We then set up a PID controller based on the angle errors
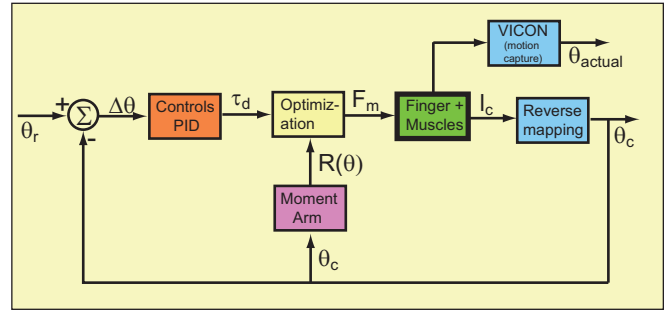

Fig. 3. Schematic for the force-optimized joint control. In each time step, the difference, $\Delta \underline{\theta}$, between the desired and estimated joint angles is fed into a PID controller to determine the necessary joint torques, τ_d. The muscle forces, F_m, needed to achieve these torques are computed via optimization using the current moment arm matrix $R(\theta_c)$. The moment arm depends on the current joint angle estimates, $\underline{\theta}_c$. These estimates are based on a reverse mapping from muscle lengths to joint angles.

to calculate for the desired joint torque, τ_{desrd}, as follows:

$$\tau_d = K_{P\theta} * \Delta \underline{\theta} + K_{D\theta} * \Delta \underline{\dot{\theta}} + K_{I\theta} * \int_t \Delta \underline{\theta} \quad (16)$$

where $K_{P\theta}$, $K_{D\theta}$ and $K_{I\theta}$ are the PID gains.

The muscles need to apply appropriate pulling forces F_m to generate the desired torques. As there are 6 muscles and 4 finger joints, determination of the muscle forces F_m that generate the desired joint torques τ_d is a redundant problem. We know that the moment arm matrix, R, relates the muscle forces to the joint torques as follows:

$$\tau_{4\times1} = -R_{4\times6}^T * (F_m)_{6\times1}. \quad (17)$$

The moment arm matrix, R, is configuration dependent [11] and is determined by taking partial derivatives of forward mapping $\underline{l}_r = f_f(\underline{\theta}_r)$ as described above (see Equation 4).

There are multiple combinations of the muscle forces that can generate a specific joint torque. However, the muscles cannot push which sets up a constraint on muscle forces. To determine the muscle forces that generate the desired joint torques we set up an optimization problem as follows:

$$\begin{aligned} \text{determine} \quad & F_m & (18) \\ \text{with} \quad & \min(\sum |\tau_d + R^T * F_m|) \\ \text{with constraints} \quad & 0 \le F_m(i) \le F_{mMax}(i) \end{aligned}$$

where F_{mMax} is the vector of the maximum force generated by the muscles. This formulation allows for the exploitation of the muscle redundancy to set up biomechanical constraints. However, for this paper we have set up the optimization with only the min-max constraints. The optimized forces are determined by using a linear programming solver at each time step of control.

For the force-optimized joint controls, we derived the moment arm matrix, R, and current angles from the current muscle lengths using the reverse mapping for each time step in the controls. Since GP prediction requires more computational resources than a simple polynomial evaluation, it is clear that using GPs will lead to a slower overall controller loop. We will quantify this more fully in the results section.

TABLE I

MEAN ABSOLUTE MAPPING ERRORS WITH GP AND LSR (MM)

	EI	PI	FDP	LUM	FDS	RI
GP	0.1838	0.4193	0.1350	0.3192	0.1408	0.4567
LSR	0.7531	0.6615	1.2267	0.8082	0.4388	1.0320

IV. RESULTS

A. Validation of Mappings

Our first experiment tests the quality of the angle to muscle length mapping using both GP and least squares fitting. Figure 4 shows the difference between the least square regression and GP when it was projected to EI muscle length estimation. Table I shows the mean absolute error for all muscle length excursions when tested on a data set of over 200,000 angles and muscle lengths combinations. The data covers the physiological ranges of motion for all the finger joints. The actual excursion length was recorded directly from the encoder values, and the other estimations were from the Vicon joint angle information translated to muscle lengths. The GP approach shows better performance, but with a small penalty on speed.
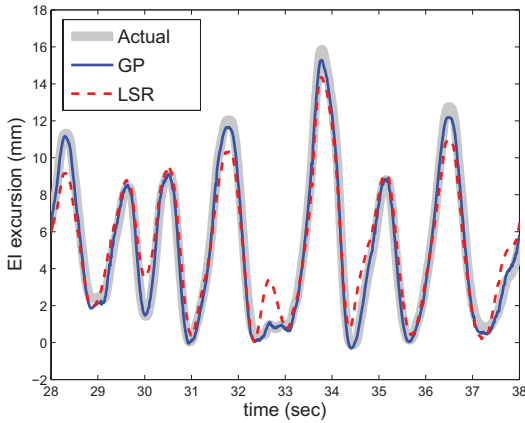

Fig. 4. GP vs least squares mapping for EI during a typical finger motion.

B. Direct Muscle Control

Next we conducted a tracking experiment by controlling the finger tip to move in a desired circular path by varying the MCP Ab-Ad and MCP Flex angles. Fig. 5 shows the desired trajectory for the angles against the muscle position control results with least square and GP techniques. This task was picked to show the combinational effect of both flexion and ab/adduction errors. The root mean squared error for this trajectory was 0.3233 radians using the least square mapping. The error for GP mapping was 0.0965 radians. The result using GP mapping is closer to the desired trajectory.

Figure 6 shows the desired and muscle lengths using GP and least square mapping. The two plots seem similar at first glance, however, the PI muscle has small but significant difference between GP and least squares mapping. This error (∼ 3mm) is enough to cause the differences in behavior shown in Figure 6. Since this trajectory was generated
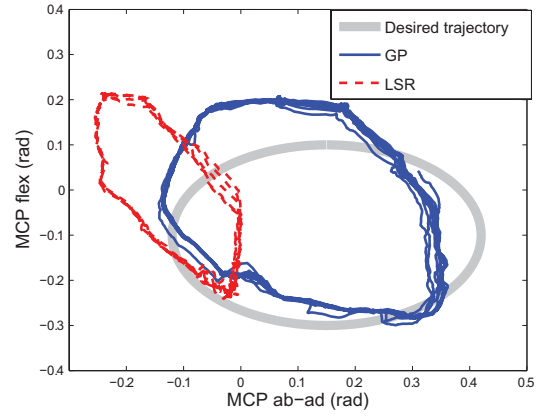

Fig. 5. MCP angles while moving the finger in a circular trajectory with direct muscle control
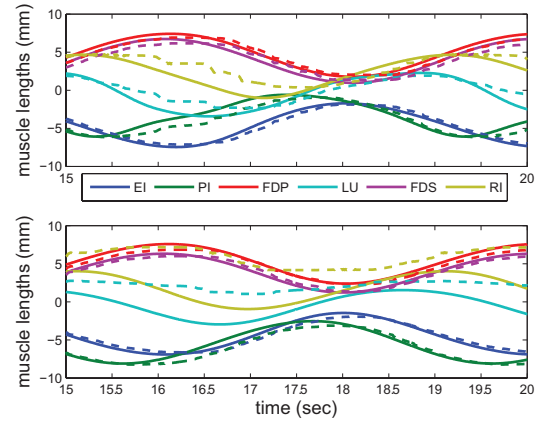

Fig. 6. Desired and actual muscle lengths while moving the finger in a circular trajectory. Desired lengths are shown by the solid lines. Dashed lines show muscles lengths actually obtained. Top plot shows the run made using GP mapping, and the run with least squares mapping is shown below.

analytically, ground truth muscle excursions were unavailable for comparison. The cycle rate for the main controller loop using GP was 208 Hz while least squares performs at 250 Hz. The difference in speed between the GP and least squares controller loop does not appear to affect the resultant finger motions.

C. Force-optimized Joint Control

Next we conducted same tracking experiment by controlling the fingertip to move in a desired square path by varying the MCP Ab-Ad and MCP Flex angles. Fig. 7 shows the desired trajectory for the angles against results from implementing the force optimized joint control results with least square and GP techniques. The root mean squared error for this trajectory was 0.07 radians using the least square mapping. The error for GP mapping was 0.1 radians.

Fig. 8 shows the computed muscle forces given the desired torque at MCP joint. The desired torque was computed using Equation 16 and the muscle forces were computed with optimization set up in Equation 18. There are several discrete muscle force changes that are commanded when the desired joint torque changes its sign (e.g. see time ∼ (30.5)
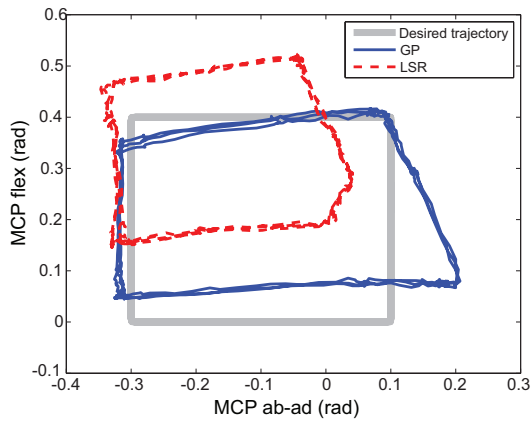
Fig. 7. MCP angles while moving the finger tip in a circular trajectory with force optimized joint control.
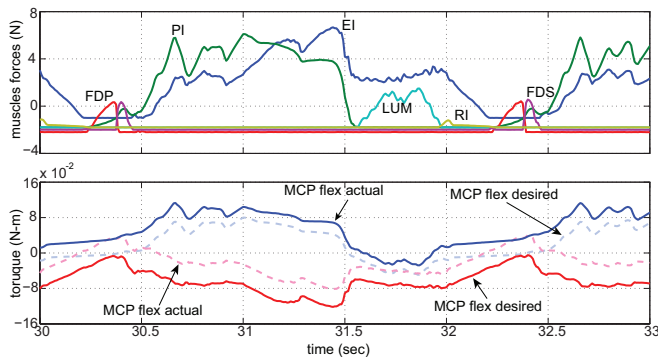

Fig. 8. Torques and forces during finger tip rotation. Red, blue and green lines show EI, FDP, and PI forces respectively.

seconds) due to the nature of the minimization equation. The bottom figure shows the desired and the actual torques at the MCP joint. The desired torque as described in Equation 16 and the actual torque is computed using the moment arm mapping and the muscle forces as given in Equation 17. The plots show that the optimization computed the correct muscle force combinations to achieve the desired torque although the desire torques are not matched precisely. Rapid changes in the muscle forces are undesirable as these cause jerky motions. These must be investigated further. The cycle rate for the main controller loop using GP was 143 Hz while least squares performs at 250 Hz

V. DISCUSSION

A. Gaussian Processes versus Least Square Regression

In our analysis, the GPs provided significantly better position tracking than the least square regression technique. As can be seen from the data [11], the relationship between angles and muscle lengths are smooth, but highly non-linear and complex. GPs are especially suitable for modeling these mappings. In addition, moment arms can be computed from the GP-based mappings as the gradients of the GP mean function (8). One drawback of the GPs is the time it takes to compute the output in the control loop. For the direct muscle control, the GPs were used only once in mapping

the desired angles θ_r to desired muscle length l_r, but for the joint control, the GPs were used twice, once for mapping current muscle lengths l_c to current angles θ_c, and again to obtain the moment arm used for force optimization. As we incorporate more biologically correct terms, we will need to make additional estimations that GPs would be useful for. We will thus investigate the use of sparse GPs for more efficient predictions [26].

Finally, a key advantage of GPs is their ability to generate consistent uncertainty estimates for their predictions (see Equation 9). Here, we do not take advantage of this ability. In the future, incorporation of GP uncertainty will allow us to build more robust controllers and to perform active learning by choosing training points that minimize uncertainty.

B. Muscle Control vs. Joint Control

Controlling the muscle positions directly requires less calculations in the control feedback loop and it results in good position tracking. However, this scheme has a number of limitations. Firstly, since tasks are defined in the joint angle space the desired trajectories for the muscle length control depend on the mappings between the joint angles and the muscle lengths, and any error in the mappings are not correctable. One solution to this problem is to use an online finger joint tracking system to actively correct the angle to muscle mappings similar to human visual feedback. Secondly, since the muscles cannot push, any errors in the extension of the muscles are not correctable without coordinated counteraction by other muscles. Generating the finger tip forces necessary in grasping and manipulation is also difficult with the position control. And most importantly, the neuromuscular controls for human hand requires control of specific joint stiffness [1], [2] which is not possible with the direct muscle position control.

The joint angle controller achieved more flexibility and incorporation of control optimization that is related to the biological control, but it was not possible to achieve a smooth circular trajectory. There are many possibilities for correcting this such as speeding up the GPs, getting force sensors, etc. Another factor is that we ignored the finger and motor dynamics, considering low finger mass ($\sim$ 20g) and low gearing at the motor. Incorporating the full dynamics of the finger, including the reflected motor inertias will improve our controllers.

C. Redundant Control Optimization

The joint control allows for the utilization of actuation redundancy in the finger. Determination of muscle forces to generate the desired joint torques is a redundant problem whose solution can be determined to match the neuromuscular control strategies [1], [2]. This approach allows for the possibility of incorporating the hand biomechanics by setting up additional constraints in the optimization problem. For example, we can set up the control problem to optimize on movement or muscle activation smoothness, specific use of selected muscles, etc. We can also utilize the actuation

redundancy to modulate the finger stiffness while satisfying the joint torque requirements.

D. Other Controllers and Comparison with the Biological Control Strategies

In this paper, we only tried a direct muscle position controller and a force-optimized joint controller for joint angle tracking. To design a more comprehensive controller other tasks need to considered. For example, it is important to evaluate fingertip or contact-point force controller for object interaction tasks and Cartesian fingertip controller for tracing tasks. With the ACT Hand mechanical structure, we believe that even these standard robotic controllers would allow more dexterity than what has been possible in the past.

In parallel, we are also interested in implementing the biological neural control strategies which are investigated in our laboratory and in the literature. For example, we are investigating the use of stiffness modulation for humans [1], [2]. The stiffness control has been shown to utilize the intrinsic muscles in a special way [21]. We will mimic such control strategies to understand the advantage of the biological control strategies.

Ultimately, we plan to compare the biological and engineering control strategies side by side to compare the relative advantages and disadvantages. While we currently believe that the biological control is currently superior to a purely robotic control, there may be engineering solutions that could allow easier augmentation for disabled hand control repair or prosthetic control.

REFERENCES

[1] P. Afshar, "A neuromuscular framework for motor control," Ph.D. dissertation, Carnegie Melon University, 2007.
[2] R. Balasubramanian and Y. Matsuoka, "Biological stiffness control strategies for the anatomically correct testbed (ACT) hand," in *IEEE International Conference on Robotics and Automation*, 2008.
[3] Barrett, "Barrett technology inc." http://www.barrett.com, 2008.
[4] A. Bicchi, "Hands for dexterous manipulation and robust grasping: A difficult road toward simplicity," *IEEE Transactions on Robotics and Automation*, vol. 16, pp. 652–662, 2000.
[5] K. B. Biggers, S. C. Jacobsen, and G. E. Gerpheide, "Low level control of the UTAH/MIT dexterous hand," in *IEEE International Conference on Robotics and Automation*, 1986.
[6] P. W. Brand and A. M. Hollister, *Clinical Mechanics of the Hand*, 3rd ed. Mosby, 1999.
[7] M. C. Carrozza, P. Dario, F. Vecchi, S. Roccella, M. Zecca, and F. Sebastiani, "The cyberhand: on the design of cybernetic prosthetic hand intended to be interfaced to the peripheral nervous system," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003.
[8] J. J. Craig., *Introduction to Robotics*. Addison Wesley, 1989.
[9] H. Cruse and et al., "On the cost functions for the control of the human arm movement," *Biological Cybernetics*, vol. 62, pp. 519–528, 1990.
[10] A. D. Deshpande, R. Balasubramanian, J. Ko, , and Y. Matsuoka, "Investigation of variable moment arms for the index finger through the ACT hand," *Submitted to Jounral of Biomechanics*, 2008.
[11] A. D. Deshpande, R. Balasubramanian, R. Lin, B. Dellon, and Y. Matsuoka, "Understanding variable moment arms for the index finger mcp joints through the ACT hand," in *IEEE BIOROB*, 2008.
[12] A. D. Deshpande and J. E. Luntz, "A method to characterize and exploit actuation redundancy in mobility and manipulation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.
[13] Y. Engel, P. Szabo, and D. Volkinshtein, "Learning to control an octopus arm with gaussian process temporal difference methods," in *Advances in Neural Information Processing Systems 18*. MIT Press, 2006.
[14] J. M. Hollerbach and K. C. Suh, "Redundancy resolution of manipulators through torque optimization," *IEEE Journal of Robotics and Automation*, vol. RA-3, pp. 308–316, 1987.
[15] S. C. Jacobsen, D. F. Iversen, R. T. Knutti, and K. B. Biggers, "Design of the UTAH/MIT. dexterous hand," in *IEEE International Conference on Robotics and Automation*, 1986.
[16] B.-H. Kim, B.-J. Yi, I. H. Suh, S.-R. Oh, and Y.-S. Hong, "A biomimetic compliance control of robot hand by considering structures of human finger," in *IEEE International Conference on Robotics and Automation*, 2000.
[17] J. Knutson, K. Kilgore, J. Mansour, and P. Crago, "Intrinsic and extrinsic contributions to the passive moment at the metacarpophalangeal joint," *Journal of Biomechanics*, vol. 33, pp. 1675–1681, 2000.
[18] J. Ko and D. Fox, "Bayesian filtering using gaussian process prediction and observation models," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008.
[19] Y.-T. Lee, J.-H. Kim, W.-K. Chungi, and Y. Youm, "Stiffness control of the coupled tendon-driven robot hand," in *International Conference on Systems, Man and Cybernetics*, 1993.
[20] Y. Matsuoka and P. Afshar, "Neuromuscular strategies for dynamic finger movements: A robotic approach," in *Proceedings of the 26th Annual International Conference of the IEEE EMBS*, 2004.
[21] T. E. Milner and D. W. Franklin, "Impedance control and internal model use during the initial stage of adaptation to novel dynamics in hum," *Journal of Physiology*, vol. 567, pp. 651–664, 2005.
[22] T. E. Milnera, D. W. Franklinb, H. Imamizub, and M. Kawatob, "Central control of grasp: Manipulation of objects with complex and simple dynamics," *NeuroImage*, vol. 36, pp. 388–395, 2007.
[23] Y. Nakamura, *Advanced Robotics Redundancy and Optimization*. Addison Wesley, 1991.
[24] J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal, "Operational space control: A theoretical and empirical comparison," *The International Journal of Robotic Research*, vol. 27, pp. 737–757, 2008.
[25] S. Narasimhan, D. M. Siegel, and J. M. Hollerbach, "Condor: An architecture for controlling the UTAH-MIT dexterous hand," *IEEE Transactions on Robotics and Automation*, vol. 5, pp. 616–627, 1989.
[26] J. Quinonero Candela and C. Rasmussen, "A unifying view of sparse approximate Gaussian process regression," *Journal of Machine Learning Research*, vol. 6, 2005.
[27] C. Rasmussen and C. Williams, *Gaussian processes for machine learning*. The MIT Press, 2005.
[28] J. K. Salisbury and B. Roth, "Kinematic and force analysis of articulated mechanical hands," *ASME Journal of Mechanisms, Transmissions, and Automation in Design*, vol. 105, pp. 35–41, 1983.
[29] L. Sciavicco and B. Siciliano, *Modelling and Control of Robot Manipulators*. Springer, 2000.
[30] M. Suzuki and T. Mayahara, "A method of real-time coordination of redundant muscles for multi-dof robot joints," in *SICE Annual Conference 2007*, 2007.
[31] K. Tahara, Z.-W. Luo, , and S. Arimoto, "On control mechanism of human-like reaching movements with musculo-skeletal redundancy," in *International Conference on Intelligent Robots and Systems*, 2006.
[32] F. J. Valero-Cuevas, "Predictive modulation of muscle coordination patterns scales fingertip force magnitude over the voluntary range." *Journal of Neurophysiology*, vol. 83, pp. 1469–1479, 2000.
[33] ——, "An integrative approach to the biomechanical function and neuromuscular control of the fingers," *Journal of Biomechanics*, vol. 38, p. 673684, 2005.
[34] M. V. Weghe, M. Rogers, M. Weissert, and Y. Matsuoka, "The ACT hand: Design of the skeletal structure," in *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, 2004.
[35] D. D. Wilkinson, M. V. Weghe, and Y. Matsuoka, "An extensor mechanism for an anatomical robotic hand," in *International Conference on Robotics & Automation*, 2003.
[36] Y. Yin, Z. Luo, M. Svinin, and S. Hosoe, "Hybrid control of multifingered robot hand for dexterous manipulation," in *IEEE International Conference on Systems, Man and Cybernetics.*, 2003.
[37] L. Zollo, S. Roccella, E. Guglielmelli, M. C. Carrozza, and P. Dario, "Biomechatronic design and control of an anthropomorphic artificial hand for prosthetic and robotic applications," *IEEE/ASME TRANSACTIONS ON MECHATRONICS*, vol. 12, pp. 418–429, 2007.