34th Annual International Conference of the IEEE EMBS
San Diego, California USA, 28 August - 1 September, 2012

State-Space Control of Prosthetic Hand Shape

M Velliste[1,2], AJC McMorland[1], E Diril[1], ST Clanton[3], and AB Schwartz[1,2]

***Abstract*— In the field of neuroprosthetic control, there is an emerging need for simplified control of high-dimensional devices. Advances in robotic technology have led to the development of prosthetic arms that now approach the look and number of degrees of freedom (DoF) of a natural arm. These arms, and especially hands, now have more controllable DoFs than the number of control DoFs available in many applications. In natural movements, high correlations exist between multiple joints, such as finger flexions. Therefore, discrepancy between the number of control and effector DoFs can be overcome by a control scheme that maps low-DoF *control* space to high-DoF *joint* space. Imperfect effectors, sensor noise and interactions with external objects require the use of feedback controllers. The incorporation of feedback in a system where the command is in a different space, however, is challenging, requiring a potentially difficult inverse high-DoF to low-DoF transformation. Here we present a solution to this problem based on the Extended Kalman Filter.**

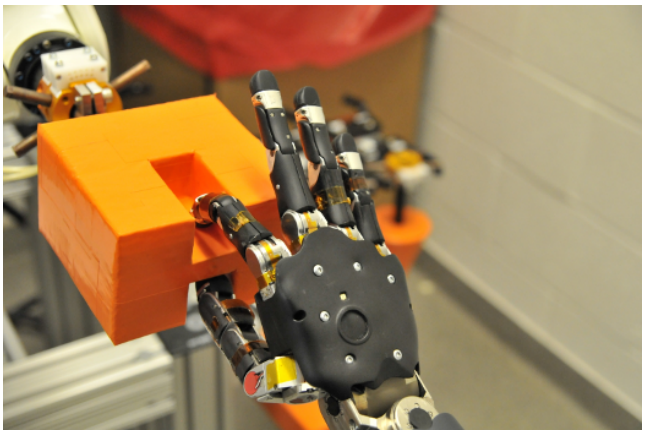

Fig. 1. MPL performing a pinch grasp of a slot object

I. Introduction

A. Background

This paper considers the problem of controlling a high-dimensional robotic system in a reduced-dimensional space. This problem occurs in neuroprosthetic hands and arms which provide a means for amputees, spinal cord injury (SCI) patients, and sufferers of locked-in syndrome to interact with the world [1], [2]. Recent technological developments promise a future where the dexterity of prosthetic hands will approach that of the human; impressive progress has been made in terms of human-like morphology and movable degrees of freedom (DoF). The human hand has 22 DoF, and the most complex robotic hand to-date, the Modular Prosthetic Limb (MPL, from Johns Hopkins University Applied Physics Laboratory, Fig. 1) has 17 DoFs, of which 10 are currently actively controllable [3].

Let us consider a simple discrete-time system where a user specifies a command, $\boldsymbol{\chi}_c[k] \in \mathbb{R}^N$ at time k, and the hand follows the command using internal servo control. N refers to the number of finger joints, and the command could be a velocity, position, torque etc. This would allow the user to form any desired hand shape, but would require the user to control many DoFs simultaneously. A high-dimensional interface may not be available to all users or may be difficult to control. Currently, in neuroprosthetic control [1], [2], fewer than 10 control DoFs are available. Hence, there can be a mismatch between the number of DoF required to drive a complex prosthetic, and the number of control DoFs available. One solution is to use a low-DoF control space that maps to the high-DoF hand space. An appropriately chosen control space might even be more intuitive than controlling individual finger joints for the user.

B. Low-DoF to high-DoF mapping

Low-DoF representations have been calculated from joint movement correlations during natural human grasping using principal component analysis (PCA) and the concept of eigengrasps [4]. Similar dimensionality reduction has been described for robotic grasping [5], [6]. As another example, the MPL hand allows arbitrary Reduced-Order Control tables to be specified to map low-DoF control space to finger joint angles [3], [7]. The correlated patterns of joint positions are most generally referred to as *hand synergies*.

PCA-based synergies would have a linear mapping to the high-DoF space, but non-linear mappings are also possible. The method we describe here is concerned with how to implement feedback control when an arbitrary mapping function, $\mathcal{G}()$, is used such that

$$\boldsymbol{p}'_c = \mathcal{G}(\boldsymbol{p}_c) \tag{1}$$

where $\boldsymbol{p}_c \in \mathbb{R}^M$ stands for the command position in the low-DoF control space, and $\boldsymbol{p}'_c \in \mathbb{R}^N$ is the command in high-DoF joint space. The control space would form an M-dimensional synergy manifold within the N-dimensional joint space. Under this scheme, the user would be able to form any desired hand shape within the constraints of the synergy manifold, but would be unable to correct any perturbations away from the manifold. External perturbations

[1]Systems Neuroscience Institute, [2]Department of Neurobiology and [3]School of Medicine, University of Pittsburgh, 4074 BST3, 3501 Fifth Avenue, Pittsburgh, PA 15261

Support contributed by JHU-APL 972352 and DARPA N66001-10-C-4056.

The authors would like to acknowledge Shinsuke Koyama for useful discussions.

are inevitable in any practical robotic implementation, especially in prosthetic applications where the environment can be highly unpredictable. Therefore, a feedback controller in the high-DoF space is needed.

C. Feedback control

Here we consider how to implement feedback control when the command is specified as a reduced-dimensional velocity $\boldsymbol{v}_c \in \mathbb{R}^M$. The first consideration is to obtain a real-time position update $\boldsymbol{p}_c$ from the low-DoF velocity $\boldsymbol{v}_c$. This in turn can be converted to the high-DoF finger command $\boldsymbol{p}'_c$ through the synergy function (Eq. 1).

The simplest way to obtain position from velocity is to integrate:

$$\boldsymbol{p}_c[k] = \boldsymbol{p}_c[k-1] + \boldsymbol{v}_c[k]\Delta t \tag{2}$$

where Δt is the length of a discrete time step. But this can lead to a mismatch between the commanded and actual movements whenever external perturbations are present. Therefore, a velocity controller based purely on integration cannot work. A better solution needs to employ feedback, so that when the fingers run into an obstacle, the velocity-to-position converter would be informed and the position appropriately not incremented:

$$\boldsymbol{p}_c[k] = \boldsymbol{p}_f[k] + \boldsymbol{v}_c[k]\Delta t \tag{3}$$

This relies on the feedback position $\boldsymbol{p}_f$ being in the low-DoF space. Since the sensors return feedback in the high-DoF space, we would have to use the inverse $\mathcal{G}()$ function to convert the high-DoF joint position feedback $\boldsymbol{p}'_f$ to low-DoF feedback position $\boldsymbol{p}_f$. This would effectively implement a second level control loop that controls for velocity, on top of the joint-level position servo control in the hand (Fig. 2). The position feedback may need to be low-pass filtered to avoid accumulation of sensor noise in the feedback loop, although this will introduce delay into the feedback.

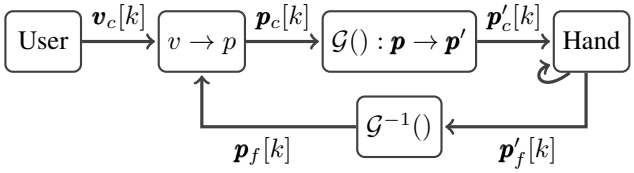

Fig. 2. Simple feedback control scheme incorporating low- to high-DoF mapping. Curved arrow represents low-level joint position servo-control.

A problem with this control scheme is that the inverse may be difficult to derive for an arbitrary $\mathcal{G}()$ function. Here, we propose a method that uses a state-space filter as a controller (Fig. 3). It is based on an Extended Kalman Filter, which obviates the need for explicitly calculating the inverse transform and optimally filters noisy feedback and command. The State-Space Hand Controller (SSHC) subsumes the roles of three of the components in Fig. 2: the integrator, the forward transform and the inverse transform. It represents the upper layer of a two-layer hierarchical control system, and relies on a lower layer position controller at each finger joint.

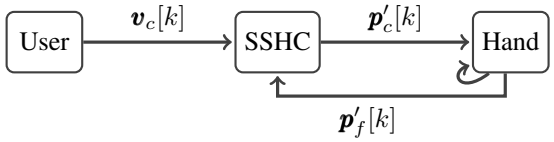

Fig. 3. Control scheme incorporating the State-Space Hand Controller (SSHC) as the upper layer of a hierarchical control system. The lower layer is joint position servo control (curved arrow).

II. Methods

A. The State-Space Hand Controller (SSHC)

The action of the SSHC can be thought of as preventing deviations of the hand state from the low-DoF synergy manifold while allowing velocity control of hand state along the manifold. If external forces cause the hand state to *deviate* from the manifold, then the feedback controller brings it back to the nearest point on the manifold as soon as the external influence is removed. In contrast, if external forces move the hand state *along* the manifold when command velocity is zero, then the hand state will remain at the new position on the manifold. In other words, the SSHC allows for compliance within the control space, but prevents deviations from it.

The state of the system is modeled in terms of position $\boldsymbol{p} \in \mathbb{R}^M$ and velocity $\boldsymbol{v} \in \mathbb{R}^M$ in low-DoF control space:

$$\boldsymbol{p}[k] = \boldsymbol{p}[k-1] + \boldsymbol{v}[k-1]\Delta t + \boldsymbol{\omega}_p[k] \quad \boldsymbol{\omega}_p \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_p) \tag{4}$$

$$\boldsymbol{v}[k] = \boldsymbol{v}[k-1] + \boldsymbol{\omega}_v[k] \quad \boldsymbol{\omega}_v \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_v) \tag{5}$$

This represents a random walk model encompassing the assumptions that, in the absence of measurements to the contrary, velocity would not change much from one time step to the next, and that position is likely to evolve as the integral of velocity. $\boldsymbol{\omega}_p$ and $\boldsymbol{\omega}_v$ are multivariate Gaussian noise terms with zero mean and covariance matrices $\boldsymbol{Q}_p$ and $\boldsymbol{Q}_v$ respectively. The complete state vector $\boldsymbol{x}_k$ and its associated noise covariance $\boldsymbol{Q}_k$ are defined as

$$\boldsymbol{x}_k \triangleq \begin{bmatrix} \boldsymbol{p}[k-d_f] \\ \boldsymbol{v}[k-d_f] \\ \vdots \\ \boldsymbol{v}[k+d_c] \end{bmatrix} \quad \boldsymbol{Q}_k \triangleq \begin{bmatrix} \boldsymbol{Q}_p & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}_v & \dots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{Q}_v \end{bmatrix} \tag{6}$$

The form of the current state $\boldsymbol{x}_k$ at discrete time k results from the need to keep track of sensor feedback, which arises from the true state at the earlier time $k - d_f$, and that the command velocity will not be reflected in the true state until the future time $k+d_c$. Accordingly, high-DoF finger position feedback $\boldsymbol{p}'_f \in \mathbb{R}^N$ and low-DoF user command velocity $\boldsymbol{v}_c \in \mathbb{R}^M$ are modeled as measurements in the state-space

filter:

$$\boldsymbol{p}'_f[k] = \mathcal{G}(\boldsymbol{p}[k-d_f]) + \boldsymbol{\epsilon}_f[k-d_f] \quad \boldsymbol{\epsilon}_f \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_f) \quad (7)$$

$$\boldsymbol{v}_c[k] = \boldsymbol{v}[k+d_c] + \boldsymbol{\epsilon}_c[k+d_c] \quad \boldsymbol{\epsilon}_c \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_c) \quad (8)$$

Command velocity is treated as measurement because it can be noisy. $\boldsymbol{\epsilon}_f$ and $\boldsymbol{\epsilon}_c$ are multivariate Gaussian noise terms with zero mean and covariance matrices $\boldsymbol{R}_f$ and $\boldsymbol{R}_c$ respectively. The complete measurement vector $\boldsymbol{z}_k$ along with its associated noise covariance $\boldsymbol{R}_k$ are defined as:

$$\boldsymbol{z}_k \triangleq \begin{bmatrix} \boldsymbol{p}'_f[k] \\ \boldsymbol{v}_c[k-d_f-d_c] \\ \vdots \\ \boldsymbol{v}_c[k] \end{bmatrix} \quad \boldsymbol{R}_k \triangleq \begin{bmatrix} \boldsymbol{R}_f & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_c & \dots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{R}_c \end{bmatrix} \quad (9)$$

The inclusion of sensor feedback in the measurement makes the SSHC a feedback controller. Filtering is performed using standard Extended Kalman Filter (EKF) equations:

$$\hat{\boldsymbol{x}}_{k|k-1} = \boldsymbol{F}_k \hat{\boldsymbol{x}}_{k-1|k-1} \quad (10)$$

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{F}_k \boldsymbol{P}_{k-1|k-1} \boldsymbol{F}^\top + \boldsymbol{Q}_k \quad (11)$$

$$\tilde{\boldsymbol{y}}_k = \boldsymbol{z}_k - h\big(\hat{\boldsymbol{x}}_{k|k-1}\big) \quad (12)$$

$$\boldsymbol{S}_k = \boldsymbol{H}_k \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^\top + \boldsymbol{R}_k \quad (13)$$

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^\top \boldsymbol{S}_k^{-1} \quad (14)$$

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k \tilde{\boldsymbol{y}}_k \quad (15)$$

$$\boldsymbol{P}_{k|k} = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k)\, \boldsymbol{P}_{k|k-1} \quad (16)$$

where

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{I\Delta t} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \dots & \boldsymbol{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{I} \end{bmatrix} \quad h(\hat{\boldsymbol{x}}_k) = \begin{bmatrix} \mathcal{G}(\hat{\boldsymbol{p}}[k-d_f]) \\ \hat{\boldsymbol{v}}[k-d_f] \\ \vdots \\ \hat{\boldsymbol{v}}[k+d_c] \end{bmatrix} \quad (17)$$

where $\boldsymbol{I}$ is an $M \times M$ identity matrix and $\boldsymbol{0}$ is a zero matrix, and $\boldsymbol{H}_k = \frac{\delta h}{\delta x}$ is the Jacobian of $h()$, calculated numerically.

The use of the EKF allows the measured position to be in a different space than the state estimate because the measurement prediction function $h()$ allows use of the arbitrary (differentiable) mapping function $\mathcal{G}()$ to predict the measurement from current state. This is the step that obviates the need for the inverse $\mathcal{G}()$.

The final output command $\boldsymbol{p}'_c \in \mathbb{R}^N$ to the lower-layer finger joint controllers is calculated as

$$\boldsymbol{p}'_c = \mathcal{G}(\hat{\boldsymbol{p}}[k+d_c+1]) \quad (18)$$

$$\hat{\boldsymbol{p}}[k+d_c+1] = \hat{\boldsymbol{p}}[k-d_f] + \sum_{d=-d_f}^{d_c} \hat{\boldsymbol{v}}[k+d]\Delta t \quad (19)$$

where $\hat{\boldsymbol{p}}$ and $\hat{\boldsymbol{v}}$ are taken from $\hat{\boldsymbol{x}}_{k|k}$. This command effectively represents estimated position at time $k+d_c+1$ in the future. It is designed this way because d_c, by definition, is how long it takes for the fingers to get to their commanded position. Fig. 4 summarizes the information flow and timing relationships between the state and measurement variables.

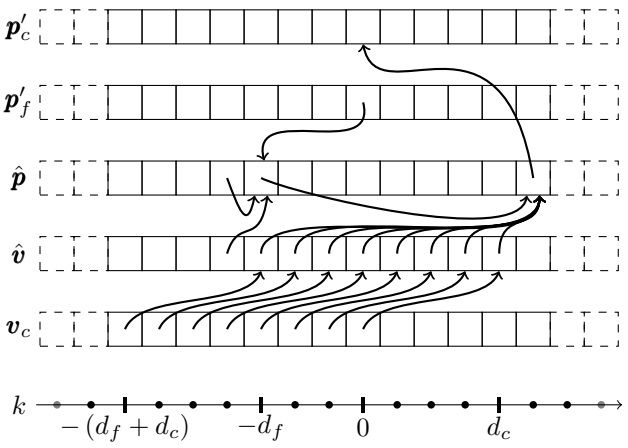

Fig. 4. Temporal relationships between state/measurement variable components.

B. Simulations

As a proof of concept of the SSHC, a simple simulation of a three-fingered hand (index and middle fingers plus thumb) was created, grasping a simple object as shown in Fig. 1.

1) Hand and object: Each finger was simulated as a single DoF that could move up and down in the range $-1 \dots 1$. With the hand fully open, the middle and index fingers would be up at 1 and the thumb down at -1. To grasp an object, each finger would move toward the center at 0. Idealistic servo control at each finger was simulated by having the actual position $\boldsymbol{p}'_a = (p_{middle}, p_{index}, p_{thumb})$ essentially follow command $\boldsymbol{p}'_c$ perfectly, except the command was passed through a 7-sample box-car filter to simulate inertia with a 3-sample control delay ($d_c = 3$). Sample interval was 20 ms. A slot object similar to Fig. 1 was simulated by simply imposing a limit on how far each finger could travel, meaning that when the finger hit the object limit, $\boldsymbol{p}'_a$ would stop following $\boldsymbol{p}'_c$. The limits were defined as $(0.6, 0.1, -0.1)$ for the middle finger, index finger and thumb respectively. Noisy feedback $\boldsymbol{p}'_f$ was simulated by adding random noise $\mathcal{N}(0, 0.3)$ to $\boldsymbol{p}'_a$, and passed through a filter to simulate a 1-sample delay ($d_f = 1$).

2) Control space: A $\mathcal{G}()$ function was defined so as to map two low-DoF control dimensions to the three fingers. Dimension 1 was mapped to flexion of all three fingers such that a positive command velocity v_{c1} in this one DoF alone would be enough to close the grip on an object. Dimension 2 was mapped to flexion of the middle finger alone so that a negative command velocity v_{c2} would allow separating it from the index finger to form a pinch grasp similar to Fig. 1. The control space can be thought of as a 2-dimensional manifold (a surface) within the 3-dimensional finger space, and the hand state at any one time would be a point in this space.

III. Results

Fig. 5 shows the SSHC performing two types of grasps of the simulated slot object which permits the index finger

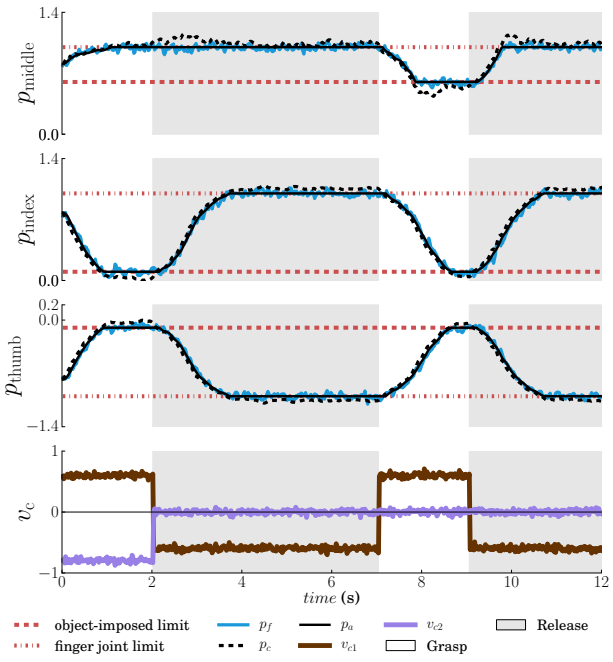

Fig. 5. Finger positions and low-DoF command velocities during simulated grasping of a slot object

and thumb to close further than the middle finger. During the first trial (from 0 s to 7 s), a pinch grasp, the second low-DoF command signal is used to differentially keep the middle finger away from the object during the grasp with the thumb and index finger. This demonstrates the ability of the SSHC to appropriately control the three finger joints based on 2-DoF velocity command to achieve the desired hand shape of a 3-DoF hand. During the second trial (from 7 s to 12 s), a full hand grasp, the control signal is chosen to create an interaction between the middle finger and the object: the object-imposed limit for the middle finger is higher than for the index finger, and only v_{c1}, which maps to both fingers, is modulated. This demonstrates the suitability of the SSHC for compliant control, where the hand is able to form to the object by interacting with it.

IV. Discussion

The SSHC is a novel control scheme that simultaneously controls velocity in a low-DoF command space, and position in a high-DoF actuator space. As a hierarchical control system, it implements an outer control loop that uses feedback from individual actuators (high-DoF output space) to maintain the position of the actuators in a coordinated pattern (low-DoF control manifold) while allowing velocity control along the manifold. It relies on an inner position control loop for each actuator. The controller can use an arbitrarily complex function to perform the mapping from low-DoF commands to high-DoF finger space, with the only restriction that the function be numerically differentiable.

In the particular example of prosthetic hand control, the SSHC enables feedback control to maintain the joint configuration within a hand synergy subspace while giving the user velocity control. To be useful for everyday living, prosthetics must cope gracefully with unforeseeable interactions with external objects, and so prosthetic actuators are typically compliant: they will yield to external forces. This means that there will frequently be a mismatch between commanded and actual positions. The SSHC provides a way to cope with this mismatch in a context where command and feedback are in different spaces. We expect this method to not only advance the development of highly dexterous prosthetic systems, but also to be of utility for the general problem of relating low-dimensional control to a high-dimensional effector.

References

[1] A. B. Schwartz, "Cortical neural prosthetics." *Annual Review of Neuroscience*, vol. 27, pp. 487–507, July 2004.

[2] M. Velliste, S. Perel, M. C. Spalding, A. S. Whitford, and A. B. Schwartz, "Cortical control of a prosthetic arm for self-feeding." *Nature*, vol. 453, no. 7198, pp. 1098–101, Jun. 2008.

[3] M. M. Bridges, M. P. Para, and M. J. Mashner, "Control System Architecture for the Modular Prosthetic Limb," *Johns Hopkins APL Technical Digest*, vol. 30, no. 3, 2011.

[4] M. Santello, M. Flanders, and J. F. Soechting, "Postural hand synergies for tool use." *The Journal of Neuroscience*, vol. 18, no. 23, pp. 10 105–15, Dec. 1998.

[5] M. Ciocarlie, C. Goldfeder, and P. Allen, "Dimensionality reduction for hand-independent dexterous robotic grasping," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, Nov 2007, pp. 3270 –3275.

[6] M. Ciocarlie, S. Clanton, M. Spalding, and P. Allen, "Biomimetic grasp planning for cortical control of a robotic hand," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, Sept 2008, pp. 2271 –2276.

[7] A. Harris, K. Katyal, M. Para, and J. Thomas, "Revolutionizing Prosthetics Software Technology," in *Proceedings of IEEE SMC*, 2011.