VALIDATION OF AN INVERSE KINEMATICS ALGORITHM FOR THE ARM

Joris M. Lambrecht and 1, 2 Robert F. Kirsch
1Department of Biomedical Engineering, Case Western Reserve University, Cleveland, OH
2 Louis Stokes Cleveland VA FES Center of Excellence, Cleveland, OH
email: joris.lambrecht@case.edu, web: fescenter.org

SUMMARY
Implantable neuroprostheses are a promising technology for restoring arm function in individuals with high-level tetraplegia but controlling these multi-joint devices is a challenge. Commanding the hand in global space may be more intuitive, offloading individual joint control to the neuroprosthesis. However, the redundancy in the number of joints relative to the hand position and orientation makes this conversion nontrivial. Inverse kinematics algorithms have been developed to resolve this problem in the fields of ergonomics and computer animation, but are not often validated relative to typical human movements. For this study, kinematic data was collected from two able-bodied subjects while performing various reaching movements. Algorithm predicted and actual trajectories were compared; percentage of variance-accounted-for, calculated across all reaches and joints, was greater than 95% in both subjects. The algorithm was found to be sufficiently fast for real-time use and has been implemented in a virtual reality environment for testing arm command strategies.

INTRODUCTION
High-level cervical spinal cord injuries can result in complete paralysis of the arm. Implantable neuroprostheses have been developed and implemented to reanimate the arm, restoring reaching and grasping functions [1]. Controlling these neuroprostheses is a challenge because there are many degrees freedom (dofs) to control and few voluntary actions with which to command them. Head orientation, head and neck EMG, EOG, EEG, and tongue movements are potential command sources. In all of these cases, commanding the position and orientation of the hand may be more intuitive than controlling individual joints. The neuroprosthesis would then be responsible for determining joint angles.

If scapular kinematics and trunk movement are ignored, placing the hand in space is controlled by 7 dofs. However, the position and orientation of any object in space can be defined by 6 dofs. This results in a redundancy when determining joint angles from a goal position/orientation. Inverse Kinematics (IK) algorithms resolve this redundancy by using optimization [2] or behavior-based rules [3].

The purpose of this study was to determine if an IK algorithm could accurately predict the joint angles trajectories typically made by able-bodied subjects from the calculated position and orientation of the hand.

METHODS
Motion Capture: Two healthy 26 year-old male subjects participated in the study after giving informed consent. Kinematic data was captured at 30 Hz using a dual camera Optottrak System (Northern Digital, Inc, Waterloo, Ontario). Figure 1 shows a schematic of the experimental setup. Bony landmarks [4] were used to define anatomical coordinate systems with respect to infrared emitting diode (IREN) clusters placed on the upper arm, forearm, and hand. The glenohumeral joint center (GH) was estimated using a regression method [5] based on scapular landmarks. Each subject was strapped to a rigid chair to minimize translation of GH, which was defined as the origin of the global coordinate system (x-right, y-superior, z-posterior). This coordinate system was chosen to match the original ISG standards [4] and the Dynamic Arm Simulator [6]. A bar with 8 holes (4 on top and 4 on the front face) was placed in front of the subject. The subjects were asked to, starting from a rest position, place a peg in a hole and return to the rest position. The bar was placed in 4 positions (See Fig 1 A-D), and 3 grip types were used, resulting in 96 different reaches. Each reach was repeated at least 3 times.

Data Analysis: Joint angles were calculated from bony landmark data. Gaps in the joint angle data—caused by intermittent loss of visibility of IRED clusters—were interpolated using a piecewise cubic Hermite method. If any gaps were longer than 5 frames, that reach was ignored. Joint angles were then low-pass filtered at 4 Hz, and segmented into reaches. Next, endpoint position, \( p_G \), and orientation, \( R_G \), were calculated via forward kinematics of the simplified arm model (Fig 2).

\[
\]
This eliminates any discrepancy introduced by using the simplified arm model when comparing results to the IK prediction which also uses this model.

**IK Algorithm:** A hybrid analytical-numerical algorithm [2,7] was modified to match our coordinate system. The analytical portion of the solution converts the three-parameter shoulder Euler angle into a rotation, \( \phi \), about a vector extending from the shoulder to the wrist (swivel or pivot angle) and a rotation \( R_c \). \( R_a \) is the rotation required by the shoulder to put the wrist at the goal position if \( \phi = 0 \).

\[
R_{ctk} = (R_c R_R R_R R_R R_R R_R R_R) R_c^{t-1} = (R_c R_a) (R_R R_R R_R) R_c^{t-1}
\]

Since \( R_c(\theta) \) and \( R_c \) can be directly calculated from the goal position, only \( \phi, \theta_5 \), \( \theta_6 \), and \( \theta_7 \) are unknown. A constrained non-linear optimization algorithm was used to minimize the cost function:

\[
f(\theta, \theta_5, \theta_6, \theta_7) = \alpha^2 + k(\phi - \phi^*)
\]

where \( \alpha \) is the scalar angle value determined from the axis-angle form of \( R_{ctk} \), \( k \) is a weighting factor, and \( \phi^* \) is previous value of \( \phi \). The assumption was that shoulder pivoting is minimized during movements. The initial guess for the optimization was the previous frame. For the first frame, the actual joint angles were used.

**RESULTS AND DISCUSSION**

The metrics used to assess performance were ‘Variance Accounted For’ (VAF) and ‘Root Mean Square Error’ (RMSE)—of all joints, \( \theta \), and the shoulder pivot, \( \phi \) alone—and Orientation Error (\( \alpha \)). Figure 3 shows a representative reach made by subject 1 and the corresponding IK prediction.

**Figure 3:** A representative reach from subject 1. The solid gray line is the actual trajectory. The dashed black line is the IK predicted trajectory. (\( k=0 \))

Figure 4 shows the summary of performance across all reaches while varying the weighting factor, \( k \). For both subjects, the best performance was achieved with a very low value (\( k \leq 0.001 \)). For both subjects, across all tasks: VAF\( \phi > 95\% \), VAF\( \theta > 80\% \), RMSE\( \phi < 5^\circ \), RMSE\( \theta < 9^\circ \). For \( k = 10 \), the requirement to minimize the change in \( \phi \) overwhelmed the cost function so the orientation error, \( \alpha \), was much larger. For \( k = 0.01 \) and \( k = 0.1 \), \( \alpha \) remained small, but the performance relative to the actual trajectory suffered. In other words, the shoulder pivot angle changed more than expected and attempting to minimize change resulted in reduced similarity to the actual movements.

An additional kinematic data set was recorded while the subjects caught and tossed a cylindrical object. The object was tossed to the subject in various positions and orientations. A representative sample from subject 1 is shown in Figure 5. Across all the toss/catch data, for both subjects, VAF\( \phi > 85\% \) and RMSE\( \theta < 10^\circ \), indicating that the algorithm may be extensible to other arm movements.

**Figure 4:** Performance across all reaches for various weighting factors, \( k \). VAF was measured across all tasks (not an average). Error Bars are one standard deviation.

**Figure 5:** Representative toss/catch data from subject 1. The solid gray line is the actual movement. The dashed black line is the IK solution. (\( k=0 \)).

**CONCLUSIONS**

The IK algorithm presented by Tolani, et al. [2,7] was validated with kinematic data from two subjects. The optimization portion of the algorithm reached an accurate solution within 10 iterations, which easily runs faster than real-time assuming a 30 Hz update rate. The IK algorithm has therefore been found suitable for animation purposes (e.g., for testing arm command strategies) as well as eventual implementation in the control systems for an arm neuroprosthesis.

**ACKNOWLEDGEMENTS**

This work was supported by NIH/NICHD N01-HD-53403.

**REFERENCES**