A Bio-inspired Filtering Framework for the EMG-based Control of Robots

Panagiotis K. Artemiadis and Kostas J. Kyriakopoulos

Abstract— **There is a great effort during the last decade towards building control interfaces for robots that are based on signals measured directly from the human body. In particular electromyographic (EMG) signals from skeletal muscles have proved to be very informative regarding human motion. However, this kind of interface demands an accurate decoding technique for the translation of EMG signals to human motion. This paper presents a methodology for estimating human arm motion using EMG signals from muscles of the upper limb, using a decoding method and an additional filtering technique based on a probabilistic model for arm motion. The decoding method can estimate, in real-time, arm motion in 3-dimensional (3D) space using only EMG recordings from 11 muscles of the upper limb. Then, the probabilistic model realized through a Bayesian Network, filters the decoder's result in order to tackle the problem of the uncertainty in the motion estimates. The proposed methodology is assessed through real-time experiments in controlling a remote robot arm in random 3D movements using only EMG signals recorded from able-bodied subjects.**

I. Introduction

Control interfaces for robots that are based on signals measured directly from the human body have received increased attention during the last decade. In this paper a control interface is proposed, that uses electromyographic (EMG) signals from muscles of the upper limb in order to control a remote robot arm. In particular, a decoding model that translates the recorded muscle activity to arm motion, in conjunction with a bio-inspired filtering technique based on the modeling of the arm movement, provides the motion commands to the robot arm, in real-time.

EMG signals have often been used as control interfaces for robotic devices. However, in most cases, only discrete control has been realized, focusing only for example at the directional control of robotic wrists [1], or at the control of multi-fingered robot hands to a limited number of discrete postures [2]. A small number of researchers have tried to build continuous models to decode arm motion from EMG signals. The Hill-based muscle model [3], whose mathematical formulation can be found in [4], is more frequently used in the literature [5]. However, only a few degrees of freedom (DoFs) were analyzed (i.e. 1 or 2), since the nonlinearity of the model equations and the large numbers of unknown parameters for each muscle make this analysis rather difficult. The authors have used EMG signals in the past to control in a continuous way a single DoF of a robot arm in [5], as well as 2 DoFs during planar catching tasks in [6] and random 2 DoFs motion in [7].

Most of the previous works in the field, do not incorporate the fact that the resulted EMG-based estimates for motion should be human-like. In other words, kinematic and dynamic characteristics that govern human arm movements can be identified, modeled, and finally incorporated into an EMG-based control interface. This would certainly improve the system accuracy, while it would increase system robustness to unforeseen cases, since muscles activation patterns never seen during training could cause a decoding model to fail.

In this paper, a methodology for controlling an anthropomorphic robot arm using EMG signals from the muscles of the upper limb, is proposed. Surface EMG electrodes are used to record from muscles of the shoulder and the elbow. The user performs random movements in the 3D space, having visual contact with the robot arm. The system architecture is divided into two phases: the training and the real-time operation. During the training phase, the user is instructed to perform random arm movements in the 3D space. Motion data are recorded through a magnetic position tracking system. EMG recordings from 11 muscles are also collected. Using EMG and motion data, a decoding model is trained to map muscle activations to joint angles computed through the motion data. Moreover, a Bayesian Network realized through a probabilistic graphical model is trained using the joint angles from the training data. This model is used to model the human arm movement, by describing joint angle dependencies during random arm motions. Having trained the decoding and the graphical model, the real-time operation phase commences. During this phase, the user can teleoperate in real-time an anthropomorphic robot arm in the 3D space. The estimates of the human arm motion is done using only EMG recordings. Then, the estimated joint angles are inputed to the graphical model, which provides a filtered version of them, based on the dependencies modeled during training. This filtering plays a significant role, since it has been proved that improves the accuracy of the motion estimates, in cases where the decoding model fails to track the real human arm motion. Finally, a control law that utilizes the final motion estimates is applied to the robot arm actuators. The efficacy of the proposed method is assessed through a large number of experiments, during which the user controls the robot arm in performing random movements in the 3D space.

P. K. Artemiadis is with the Department of Mechanical Engineering, Massachusetts Institute of Technology (MIT), Cambridge, MA, 02139 USA. Email: partem@mit.edu

K. J. Kyriakopoulos is with the Control Systems Lab, School of Mechanical Eng., National Technical University of Athens, 9 Heroon Polytechniou Str, Athens, 157 80, Greece. Email: kkyria@mail.ntua.gr

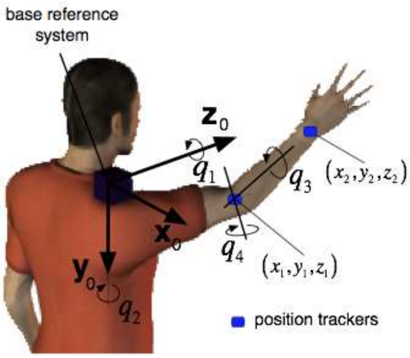

Fig. 1. The user moves his arm in the 3D space. Two position tracker measurements are used for computing the four joint angles. The tracker base reference system is placed on the shoulder.

II. Materials and methods

A. Background and Problem Definition

There is no doubt that the musculoskeletal system of humans is quite efficient, while very complex. Narrowing our interest down to the upper limb and not considering finger motion, approximately 30 muscles actuate 7 DoFs. In this study, we are focusing on the principal joints of the upper limb, i.e. the shoulder and the elbow. The wrist motion is not included in the analysis for simplicity. Since the method proposed here will be used for the control of a robot arm, equipped with 2 rotational DoFs at each of the shoulder and the elbow joints, we will model the human shoulder and elbow as having two DoFs too, without any loss of generality. The elbow is modeled with a similar pair of DoFs corresponding to the flexion-extension and pronation-supination of this joint. Hence, 4 DoFs will be analyzed from a kinematic point of view. For the training of the proposed system, the motion of the upper limb should be recorded and joint trajectories should be extracted. For this scope, a magnetic position tracking system was used, equipped with two position trackers and a reference system, with respect to which the 3D position of the trackers is provided. In order to compute the 4 joint angles, one position tracker is placed at the user's elbow joint and the other one at the wrist joint. The reference system is placed on the user's shoulder. The set-up as well as the 4 modeled DoFs are shown is Fig. 1. Let $\mathbf{T_1} = \begin{bmatrix} x_1 & y_1 & z_1 \end{bmatrix}^T$, $\mathbf{T_2} = \begin{bmatrix} x_2 & y_2 & z_2 \end{bmatrix}^T$ be the position of the trackers with respect to the tracker reference system. Let q_1, q_2, q_3, q_4 be the four joint angles modeled as shown in Fig. 1. Finally, by solving the inverse kinematic equations (see [8] for more details) the joint angles are given by:

$$\begin{aligned} q_1 &= \arctan 2\left(\pm y_1, x_1\right) \\ q_2 &= \arctan 2\left(\pm\sqrt{x_1^2 + y_1^2}, z_1\right) \\ q_3 &= \arctan 2\left(\pm B_3, B_1\right) \\ q_4 &= \arctan 2\left(\pm\sqrt{B_1^2 + B_3^2}, -B_2 - L_1\right) \end{aligned} \tag{1}$$

where

$$\begin{aligned} B_1 &= x_2\cos\left(q_1\right)\cos\left(q_2\right) + y_2\sin\left(q_1\right)\cos\left(q_2\right) - z_2\sin\left(q_2\right) \\ B_2 &= -x_2\cos\left(q_1\right)\sin\left(q_2\right) - y_2\sin\left(q_1\right)\sin\left(q_2\right) - z_2\cos\left(q_2\right) \\ B_3 &= -x_2\sin\left(q_1\right) + y_2\cos\left(q_1\right) \end{aligned} \tag{2}$$

where L_1 the length of the upper arm. The length of the upper arm can be computed from the distance of the first position tracker from the base reference system, i.e. $L_1 = \|\mathbf{T_1}\| = \sqrt{x_1^2 + y_1^2 + z_1^2}$. Likewise, the length of the forearm L_2 can be computed from the distance between the two position trackers, i.e.$L_2 = \sqrt{\left(x_2 - x_1\right)^2 + \left(y_2 - y_1\right)^2 + \left(z_2 - z_1\right)^2}$.

Regarding muscle recordings, based on the biomechanics literature [9], a group of 11 muscles, mainly responsible for the studied motion is recorded: deltoid (anterior), deltoid (posterior), deltoid (middle), pectoralis major, teres major, pectoralis major (clavicular head), trapezius, biceps brachii, brachialis, brachioradialis and triceps brachii. Surface bipolar EMG electrodes used for recording, are placed on the user's skin following the directions given in [9]. Raw EMG signals after amplification are digitized at the sampling frequency of 1 kHz and processed with a linear envelop.

B. EMG Decoding Model

Since the number of muscles recorded is quite large (i.e. 11), a low-dimensional (low-D) representation of muscle activations will be used instead of individual activations. The most widely used dimension reduction technique is principal component analysis (PCA). During the training period, the EMG recordings from each muscle are preprocessed, and then they are represented into a low-dimensional space, using the PCA algorithm. It was found that a 2-dimensional (2D) space could represent most the original high dimensional data variance (more than 96%). The authors have used the dimensionality reduction for muscle activations in the past for planar movements of the arm [7]. Therefore the details of the method application are omitted.

Having represented the muscle activations into a low-dimensional space, one can build a model that will use the EMG low-dimensional embeddings to estimate performed motion. Let $\mathbf{U}_t \in \mathbb{R}^2$ be the 2-dimensional vector of the low-dimensional representation of the 11 muscle recordings, at time $t = kT, \quad k = 1, \ldots,$ where T the sampling period. Let $\mathbf{y}_t \in \mathbb{R}^4$ be the vector of the arm joint angles at the same time instance. The model that will be used for decoding the EMG activity to performed motion is defined as

$$\begin{aligned} \mathbf{x}_{t+1} &= \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{U}_t + \mathbf{v}_t \\ \mathbf{y}_t &= \mathbf{C}\mathbf{x}_t + \upsilon_t \end{aligned} \tag{3}$$

where $\mathbf{x}_t \in \mathbb{R}^d$ is a hidden state vector, d the dimension of this vector and $\mathbf{v}_t$, υ_t zero-mean Gaussian noise vectors in process and observation equations respectively, i.e $\mathbf{v}_t \sim N\left(\mathbf{0}, \mathbf{W}\right)$, $\upsilon_t \sim N\left(\mathbf{0}, \mathbf{Q}\right)$, where $\mathbf{W} \in \mathbb{R}^d$, $\mathbf{Q} \in \mathbb{R}^4$ are the covariance matrices of $\mathbf{v}_t$, υ_t respectively. The matrix $\mathbf{A}$ determines the dynamic behavior of the hidden state vector $\mathbf{x}$, $\mathbf{B}$ is the matrix that relates muscle activations $\mathbf{U}$ to the state vector $\mathbf{x}$, while $\mathbf{C}$ is the matrix that represents the relationship between the joint kinematics $\mathbf{y}$ and the state vector $\mathbf{x}$.

Model training entails the estimation of the matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{W}$ and $\mathbf{Q}$. Given a training set of length m, including the low-dimensional embeddings of the muscle activations and the corresponding joint angles, the model parameters can be found using an iterative prediction-error minimization (i.e. maximum like-lihood) algorithm [10]. It must be noted that this kind of decoding model has been used by the authors in the past for similar EMG-based teleoperation scenarios [7]. However, this paper presents a significant improvement of the previously proposed methodology, by introducing the bio-inspired filtering approach analyzed below.

C. Modeling Human Arm Movement

The modeling of human arm movement has received increased attention during the last decades, especially in the field of robotics. This is because there is a great interest in modeling and understanding underlying laws and motion dependencies among the DoFs of the arm, in order to incorporate them into robot control schemes. In this paper, in order to model the dependencies among the DoFs of the arm during random 3D movements, we are going to use graphical models.

1) Graphical Models: Graphical models are a combination between probability theory and graph theory. They provide a tool for dealing with two characteristics; the uncertainty and the complexity of random variables. Given a set $\mathbf{F} = \left\{ \begin{array}{ccc} f_1 & \dots & f_N \end{array} \right\}$ of random variables with joint probability distribution $p(f_1, \dots, f_N)$, a graphical model attempts to capture the conditional dependency structure inherent in this distribution, essentially by expressing how the distribution factors as a product of *local functions*, (e.g. conditional probabilities) involving various subsets of $\mathbf{F}$. Directed graphical models, is a category of graphical models, also known as Bayesian Networks. A directed acyclic graph is a graphical model where there are no graph cycles when the edge directions are followed. Given a directed graph $G = (V, E)$, where V the set of vertices (or nodes) representing the variables $f_1, \dots, f_N$, and E the set of directed edges between those vertices, the joint probability distribution can be written as follows:

$$p(f_1, \dots, f_N) = \prod_{i=1}^{N} p(f_i \,|a(f_i)) \tag{4}$$

where, $a(f_i)$ the *parents* (or direct ancestors) of node f_i. If $a(f_i) = \emptyset$ (i.e. f_i has no parents), then $p(f_i \,|\emptyset) = p(f_i)$, and the node i is called the *root* node. Eq. (4) requires the parents of each variable, therefore the structure of the graphical model is required. This can be learned from the training data, using the algorithm presented below.

2) Building the Model: A version of a directed graphical model is a tree model. It's restriction is that each node has only one parent. The optimal tree for a set of variables is given by the Chow-Liu algorithm [11]. Briefly, the algorithm constructs the maximum spanning tree of the complete mutual information graph, in which the vertices correspond to variables of the model and the weight of each directed edge $f_i \to f_j$ is equal to the mutual information $I(f_i, f_j)$, given by

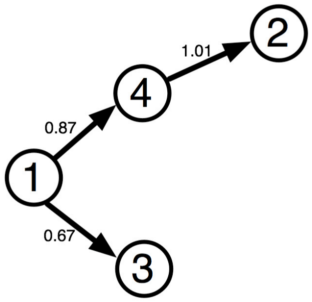

Fig. 2. The directed graphical model (tree) representing nodes (i.e. joint angles) dependencies. Node i corresponds to q_i. $i \to j$ means that node i is the parent of node j, where $i, j = 1, 2, 3, 4$. The mutual information $I(i, j)$ is shown at each directed edge connecting i to j, in bits.

$$I(f_i, f_j) = \sum_{f_i, f_j} p(f_i, f_j) \log \frac{p(f_i, f_j)}{p(f_i)\, p(f_j)} \tag{5}$$

where $p(f_i, f_j)$ the joint probability distribution function for f_i, f_j, and $p(f_i)$, $p(f_j)$ the marginal distribution probability functions for f_i, f_j respectively. Mutual information is a unit that measures the mutual dependence of two variables. The most common unit of measurement of mutual information is the bit, when logarithms to the base 2 are used. It must be noted that the variables $\left\{ \begin{array}{ccc} f_1 & \dots & f_N \end{array} \right\}$ are considered discrete in the definition of (5). Details about the algorithm of the maximum spanning tree construction can be found in [11].

In our case, we have a set of 4 discrete variables $\{q_1, q_2, q_3, q_4\}$. These variables correspond to the joint angles of the 4 modeled DoFs of the arm, in degrees, rounded to the nearest integer value. Discrete variables are used for simplifying subsequent algorithm for training and inference using the directed graphical model, without losing much information from the data, since the maximum error imposed by discretizing joint angles values to the nearest integer is 0.5 deg. Using joint angle data recorded during the training phase, we can build the tree model. The resulted tree structure is shown in Fig. 2. Therefore, using (4) and the tree structure (see Fig. 2), we can define the joint probability of the four variables representing joint angles by

$$p(q_1, q_2, q_3, q_4) = p(q_1)\, p(q_4 \,|q_1)\, p(q_3 \,|q_1)\, p(q_2 \,|q_4) \tag{6}$$

where $p(q_i \,|q_j)$, $i, j = 1, 2, 3, 4$, the conditional probability distribution of q_i, given its parent q_j. These conditional probabilities can be initially described by 2D histograms, constructed using the training data. I.e. for each value of the parent q_j, construct a histogram of the values taken by child q_i. In this way, we come up with conditional 2D histograms for each pair of parent-child. Since training data are finite, and there can be cases where a specific value for a joint angle was not observed, while values around this were observed, we are going to move one step further towards a continuous

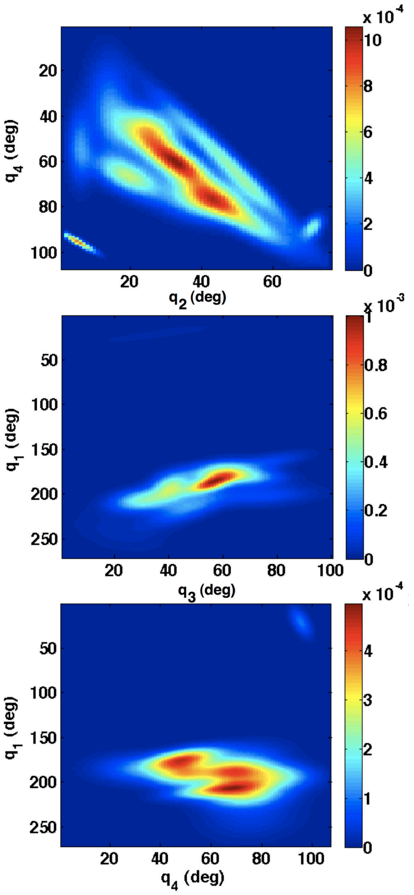

Fig. 3. Conditional probabilities represented in 2D histograms for each pair of the tree. Observed values for parents and children are shown in columns and rows respectively.

model by fitting a Gaussian Mixture Model (GMM) to the conditional distribution of a variable and its parent. For the root variable of the tree, i.e. q_1, a GMM will also be fitted. Therefore, the marginal probability distribution function of the root is given by

$$p\left(q_1\right) = \sum_{c=1}^{K_1} p_c \mathcal{N}\left(q_1; \mu_{1c}, \sigma_{1c}\right) \tag{7}$$

where K_1 the number of the Gaussian mixture components, p_c the mixing coefficients of the components and $\mathcal{N}\left(q_1; \mu_{1c}, \sigma_{1c}\right)$ a Gaussian with mean μ_{1c} and variance σ_{1c}. The conditional probability distribution of a child variable q_i given its parent variable q_j is given by

$$p\left(q_i \left| q_j \right.\right) = \sum_{c=1}^{K_j} p_c N\left(q_i; \mu_{ic} + \omega_{ic} q_j, \sigma_{ic}\right) \tag{8}$$

where ω_{ic} weights for the parent variable q_j calculated by the fitting procedure. Details about the GMMs and their fitting procedure (Expectation Maximization (EM)) can be found in [12]. Having obtained the continuous estimates of the 2D histograms, we can then revert to the tabular (discrete) representation of the conditional distribution, by using (8) for each parent-child pair value, and (7) for obtaining the 1-dimensional histogram for the root node. The resulted 2D conditional histograms are shown in Fig. 3. Observing the 2D histograms one can see that the conditional distributions are not Gaussian-like. Therefore, techniques like Principal Component Analysis (PCA) or Kalman Filter would definitely fail in describing arm movement and being a successful filter for our problem. However, a graphical model can incorporate these dependencies, therefore it can be used for modeling the complexity of arm movements.

3) Inference using the Graphical Model: Inference in probabilistic models in general, and in Bayesian Networks in our case, is the estimation of values of *hidden* nodes in a graph, given the values of the *observed* ones, where *hidden* nodes are the nodes that are not known either due to lack of measurement method or due to some missing measurements, and *observed* nodes are the nodes that are measured.

Sometimes a node is not observed, but we have some distribution over its possible values; this is often called "soft" or "virtual" evidence. Using the *junction tree algorithm* one can infer a hidden node, given a probability distribution over the possible values of the other, "observed" nodes. In this case, the *junction tree algorithm* finds the posterior marginal distributions for all the nodes in the tree. Since all nodes interact to each other according to the tree structure, there can be cases, where the prior distribution for a node deriving from its "virtual" evidence, is different from the posterior distribution after inference. Therefore, there can be cases where we have a prior probability distribution over the possible values of all the nodes of the tree, and by exploiting the tree connectivity (i.e. nodes dependency), find a *better* posterior probability distribution for the values of the nodes. This is the characteristic that will be used in our case for filtering the EMG-based estimates, using the tree structure built from the training data.

D. Filtering Motion Estimates using the Graphical Model

Using model equation (3), at every time instance[1], motion estimates are provided through the vector $\mathbf{y} = \left[\begin{array}{cccc} \hat{q}_1 & \hat{q}_2 & \hat{q}_3 & \hat{q}_4 \end{array}\right]^T$[2]. However, from the definition of the model (3), these *prior* motion estimates belong to a Gaussian distribution given by

$$\hat{q}_i \sim \mathcal{N}\left(\mathbf{C}_i \mathbf{x}, \sigma_i^2\right), \quad i = 1, 2, 3, 4 \tag{9}$$

where $\mathbf{C}_{i_{1\times d}}$ is the i^{th} row of the matrix $\mathbf{C}$ of the model, i.e. $\mathbf{C} = \left[\begin{array}{cccc} \mathbf{C_1} & \mathbf{C_2} & \mathbf{C_3} & \mathbf{C_4} \end{array}\right]^T$, and σ_i^2 the variance of each prior Gaussian distribution[3]. Therefore, at every time instance, the EMG-based decoding model outputs a prior distribution for every joint angle of the human arm. These distributions are considered as "soft" evidences for the nodes of the tree shown in Fig. 2. Then, using the junction tree algorithm, we get the posterior distributions for each tree node, which correspond to our final estimates for the human arm joint angles. The total architecture is depicted in Fig. 4. The way this filtering technique improves the overall system accuracy is assessed through various experiments, analyzed in the Section III.

[1] The EMG-based decoding model outputs motion estimates $\mathbf{y}$ at the frequency of the EMG acquisition, i.e. 1 kHz.

[2] Subscripts t denoting time instances are omitted for simplicity.

[3] Covariance matrix $\mathbf{Q}$ is defined as diagonal matrix during model fitting.

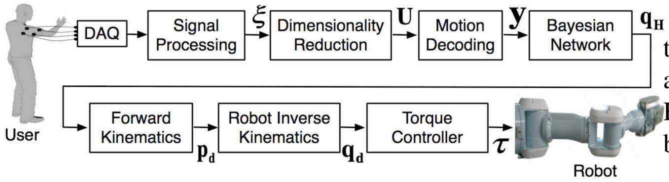

Fig. 4. The total system architecture.

E. Robot Control

Having computed the final estimates for the human joints angles $\mathbf{q_H} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}^T$, we can then command the robot arm. However, since robot and user's links have different length, the direct control in joint space would lead the robot end-effector in different position in the 3D space than that desired by the user. Consequently, the user's hand position should be computed by using the estimated joint angles, and then we can command the robot to drive its end-effector at this point in space. This is realized by using the forward kinematics of the human arm to compute the user's hand position and then solving the inverse kinematics for the robot arm to drive its end-effector to the same position in the 3D space. Hence, the final command to the robot arm is in joint space. Therefore, the subsequent robot controller analysis, assumes that a final vector $\mathbf{q_d} = \begin{bmatrix} q_{1d} & q_{2d} & q_{3d} & q_{4d} \end{bmatrix}^T$ containing the 4 desired robot joint angles is provided, where these joint angles are computed through the robot inverse kinematics as described above.

A 7 DoF anthropomorphic robot arm (PA-10, Mitsubishi Heavy Industries) is used. Only four DoFs of the robot are actuated (joints of the shoulder and elbow) while the others are kept fixed at zero position via electromechanical brakes. The arm is horizontally mounted to mimic the human arm. The robot motors are controlled in torque. In order to control the robot arm using the desired joint angle vector $\mathbf{q}_d$, an inverse dynamic controller is used, defined by:

$$\tau = \mathbf{I}\left(\mathbf{q_r}\right)\left(\ddot{\mathbf{q}}_\mathbf{d} + \mathbf{K_v}\dot{\mathbf{e}} + \mathbf{K_P}\mathbf{e}\right) + \mathbf{G}\left(\mathbf{q_r}\right) + \mathcal{C}\left(\mathbf{q_r}, \dot{\mathbf{q}}_\mathbf{r}\right)\dot{\mathbf{q}}_\mathbf{r} + \mathbf{F_{fr}}\left(\dot{\mathbf{q}}_\mathbf{r}\right) \tag{10}$$

where $\tau = \begin{bmatrix} \tau_1 & \tau_2 & \tau_3 & \tau_4 \end{bmatrix}^T$ is the vector of robot joint torques, $\mathbf{q_r} = \begin{bmatrix} q_{1r} & q_{2r} & q_{3r} & q_{4r} \end{bmatrix}^T$ the robot joint angles, $\mathbf{K_v}$ and $\mathbf{K_p}$ gain matrices and $\mathbf{e}$ the error vector between the desired and the robot joint angles, i.e.

$$\mathbf{e} = \begin{bmatrix} q_{1d} - q_{1r} & q_{2d} - q_{2r} & q_{3d} - q_{3r} & q_{4d} - q_{4r} \end{bmatrix}^T \tag{11}$$

$\mathbf{I}$, $\mathbf{G}$, $\mathcal{C}$ and $\mathbf{F_{fr}}$ are the inertia tensor, the gravity vector, the Coriolis-centrifugal matrix and the joint friction vector of the four actuated robot links and joints respectively, identified in [13]. The vector $\ddot{\mathbf{q}}_\mathbf{d}$ corresponds to desired angular acceleration vector that is computed through simple differentiation of the desired joint angle vector $\mathbf{q_d}$ using a necessary low-pass filter to cut off high frequencies.

III. Results

A. Hardware and Experiment Design

The proposed architecture is assessed through remote teleoperation of the robot arm using only EMG signals from the 11 muscles as analyzed above. The robot arm used is a 7 DoF anthropomorphic manipulator (PA-10, Mitsubishi Heavy Industries). The details of the experimental setup can be found in [14].

The user is initially instructed to move his/her arm randomly in 3D space as shown in Fig. 1. During this phase EMG signals and position trackers measurements are collected for not more than 3 minutes. These data are enough to train the EMG-based decoding model and the graphical model analyzed earlier. The models computation time is less than 1 minute. As soon as the models are estimated, the real-time operation phase takes place. The user is instructed to move the arm in 3D space, having visual contact with the robot arm. The position trackers measurements are not used during this phase. Using the proposed method, estimations about the human motion are computed using only the recorded EMG signals from the 11 muscles mentioned earlier. However, the position trackers are kept in place (i.e. on the human's arm) for offline validation reasons. The estimated hand trajectories versus the real ones, during real-time operation phase are shown in Fig. 5. Moreover the corresponding trajectories with and without the proposed bio-inspired filtering technique are illustrated, in order to prove the method efficiency. The proposed system was tested by 3 subjects in total with similar results.

B. Efficiency Assessment

Two criteria will be used for assessing the accuracy of the reconstruction of human motion using the proposed methodology. These are the root-mean-squared error (RMSE) and the correlation coefficient (CC). The latter describes essentially the similarity between the reconstructed and the true motion profiles and constitutes the most common means of reconstruction assessment for decoding purposes. The mathematical definitions of the criteria can be found in [7]. Real and estimated motion data were recorded for 40 seconds during the real-time operation phase. Using the hand forward kinematics, the criteria values are computed in Cartesian space and listed in Table I.

In general the proposed methodology was used very efficiently for controlling the robot arm. As it can be seen from the results, the bio-inspired filtering method improves significantly the decoding performance, while increases the system robustness in cases where the decoding method couldn't track the human arm motion.

IV. Conclusions and discussion

In this paper, a methodology for controlling an anthropomorphic robot arm using EMG signals from the muscles of the upper limb, is proposed. An EMG-based decoding model provides estimates of arm motion in 3D, in real-time. Then, a bio-inspired filter, realized through a Bayesian Network approach, improves the accuracy of the estimates, by using human motion characteristics learned during model training, mathematically formed using joint angle dependencies for 3D arm movements. The final motion estimates are used to control a robot arm in 3D space, in real-time.

TABLE I

COMPARISON BETWEEN THE PROPOSED METHODOLOGY USING THE BIO-INSPIRED FILTERING APPROACH AND A SINGLE DECODING MODEL WITHOUT FILTERING, IN CARTESIAN SPACE.

Method	CC_x	CC_y	CC_z	$RMSE_x\ (cm)$	$RMSE_y\ (cm)$	$RMSE_z\ (cm)$
Decoding with Bio-inspired Filter	0.97	0.96	0.96	1.56	1.79	1.89
Single decoding	0.85	0.79	0.76	5.89	9.14	5.78

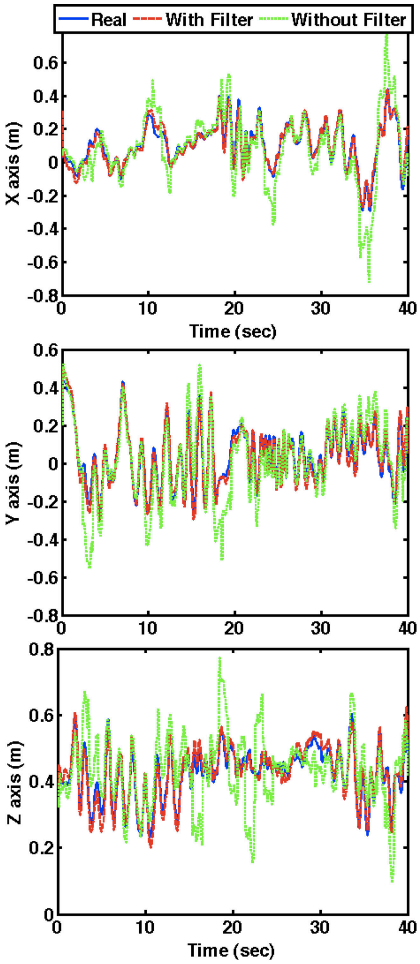

Fig. 5. Real and estimated human hand trajectory during the real-time operation phase. The estimated trajectory without the Bayesian Network filtering approach is also depicted.

The novelty of the method proposed here can be centered around two main issues. First, human arm movements are modeled using a Bayesian Network, revealing significant joint angle dependencies, and finally constructing a probabilistic model for arm movements that can be used in many research fields (i.e. robotics, graphics, biomechanics). Secondly, by using this Bayesian Network in order to filter the motion estimates of the EMG-based decoder, we achieved to improve the overall system performance, incorporating the human-like characteristics of the arm motion. In this way, the system is more robust in cases where the EMG-based decoding model could not track the human arm movements. This result enables the dexterous control of robotic devices, as presented in this study.

REFERENCES

[1] O. Fukuda, T. Tsuji, M. Kaneko, and A. Otsuka, “A human-assisting manipulator teleoperated by emg signals and arm motions,” *IEEE Trans. on Robotics and Automation*, vol. 19, no. 2, pp. 210–222, 2003.

[2] J. Zhao, Z. Xie, L. Jiang, H. Cai, H. Liu, and G. Hirzinger, “Levenberg-marquardt based neural network control for a five-fingered prosthetic hand,” *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 4482–4487, 2005.

[3] A. V. Hill, “The heat of shortening and the dynamic constants of muscle,” *Proc. R. Soc. Lond. Biol.*, pp. 136–195, 1938.

[4] F. E. Zajac, “Muscle and tendon: Properties, models, scaling, and application to biomechanics and motor control,” *Bourne, J. R. ed. CRC Critical Rev. in Biomed. Eng.*, vol. 17, pp. 359–411, 1986.

[5] P. K. Artemiadis and K. J. Kyriakopoulos, “Teleoperation of a robot manipulator using emg signals and a position tracker,” *Proc. of IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp. 1003–1008, 2005.

[6] ——, “Emg-based teleoperation of a robot arm in planar catching movements using armax model and trajectory monitoring techniques,” *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 3244–3249, 2006.

[7] ——, “Emg-based teleoperation of a robot arm using low-dimensional representation,” *Proc. of IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp. 489 – 495, 2007.

[8] ——, “Assessment of muscle fatigue using a probabilistic framework for an emg-based robot control scenario,” *Proc. of IEEE Int. Conf. Bioinformatics and Bioengineering*, 2008.

[9] J. R. Cram and G. S. Kasman, *Introduction to Surface Electromyography*. Inc. Gaithersburg, Maryland: Aspen Publishers, 1998.

[10] L. Ljung, *System identification: Theory for the user*. Upper Saddle River, NJ: Prentice-Hall, 1999.

[11] C. K. Chow and C. N. Liu, “Approximating discrete probability distributions with dependence trees,” *IEEE Transactions on Information Theory*, vol. 14(3), pp. 462–467, 1968.

[12] G. McLachlan and D. Peel, *Finite mixture models*. John Wiley & Sons, Inc, 2000.

[13] N. A. Mpompos, P. K. Artemiadis, A. S. Oikonomopoulos, and K. J. Kyriakopoulos, “Modeling, full identification and control of the mitsubishi pa-10 robot arm,” *Proc. of IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Switzerland*, 2007.

[14] P. K. Artemiadis and K. J. Kyriakopoulos, “Emg-based position and force control of a robot arm: Application to teleoperation and orthosis,” *Proc. of IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Switzerland*, 2007.