### AggiE-Challenge Project Executive Summary

**Development of a Mobile Motion Capture System with Extensive Features**

<table>
<thead>
<tr>
<th><strong>Principal Investigator</strong></th>
<th>Reza Langari</th>
<th><em>Professor, Department of Mechanical Engineering</em></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Graduate Assistant</strong></td>
<td>Hyosang Moon</td>
<td><em>Department of Mechanical Engineering</em></td>
</tr>
<tr>
<td><strong>Student Members</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Group 1</td>
<td>Eric Gros</td>
<td>Department of Biomedical Engineering</td>
</tr>
<tr>
<td></td>
<td>Karla Gonzalez-Coronado</td>
<td>Department of Industrial and System Engineering</td>
</tr>
<tr>
<td></td>
<td>Romina Del Bosque</td>
<td>Department of Biomedical Engineering</td>
</tr>
<tr>
<td>Group 2</td>
<td>Hari Shrestha</td>
<td>Department of Mechanical Engineering</td>
</tr>
<tr>
<td></td>
<td>Logan Rector</td>
<td>Department of Mechanical Engineering</td>
</tr>
<tr>
<td></td>
<td>Paul C. Pereira</td>
<td>Department of Mechanical Engineering</td>
</tr>
<tr>
<td>Group 3</td>
<td>Chokote W. Kamdeu</td>
<td>Department of Mechanical Engineering</td>
</tr>
<tr>
<td></td>
<td>Minsung Choi</td>
<td>Department of Mechanical Engineering</td>
</tr>
<tr>
<td></td>
<td>Timothy Omoniyi</td>
<td>Department of Mechanical Engineering</td>
</tr>
<tr>
<td>Group 4</td>
<td>Jack Reid</td>
<td>Department of Mechanical Engineering</td>
</tr>
<tr>
<td></td>
<td>Tyler Romero</td>
<td>Department of Computer Science and Engineering</td>
</tr>
</tbody>
</table>
Development of a Mobile Motion Capture System with Extensive Features

Objectives
The proposed study addresses an efficient method to capture and analyze the human arm motions with relatively lower cost than the conventional laboratory installed motion capture systems (e.g. Vicon or Optotrack). By developing a mobile motion capture system, the kinematic and biophysical (i.e. EMG – electromyography) information of the arm motions can be acquired in the human subject’s actual daily activities.

In addition, to enrich the motion information and to get a better understanding on the human motor behavior, extensive features are also implemented beyond the captured motion kinematics. Based on an inverse dynamics model of human arm, the system will enable us to read the required joint torque for the captured motion. In addition, the actual muscular force for the selected muscles will be plotted based on the physiological model of the skeletal muscles (e.g. Hill-based model). We can monitor the activations of each skeletal muscle during the motion through the acquired EMG data. However, since the EMG from each channel is an electrical action potential from the muscle, we cannot directly compare them in a standardized way. By quantifying the activations of each muscle in a more generalized format (i.e. force value) compare to the original EMG signal, we can interpret the subject’s motion intent in a form of voluntary muscle contractile force values.

The main objective of this project is to develop a platform which can acquire measurable data during human motions in daily activities. We ultimately expect that the developed platform will enable us to gather enormous amount of human motion data and form a database. From this database, strategies or governing rules on human arm reaching motions can be extracted (or identified) for better understanding on human motor behavior. At last, based on the database and identified governing rules of human arm reaching motions, an optimized control algorithm for various human interactive robots (e.g. exoskeletons, prostheses and haptic devices) can be designed.

Basic Concept
For the system’s mobility, a wireless IMU (Inertial Measurement Unit) is attached on each arm link. Since the human arm can be modeled as rigid linkage system, the full kinematics (i.e. three positions and three orientation angles) of each arm link can reconstruct the entire arm motion. The wireless modules each with a single channel surface EMG unit are also attached on the selected major skeletal muscles which dominate joint DOF (Degree Of Freedom) motions. The captured kinematic and biophysical information is transferred to a host PC device and the reconstructed motion kinematics and extensive feature values are represented through a user interface application.

System Architecture and Group Interrelationships
The overall system structure is shown in Fig. 1. As shown in the figure, the entire task consists of 7 modules: 1) Dead Reckoning algorithm, 2) Inverse kinematics module, 3) Inverse dynamics module, 4) Varying moment arm computation module, 5) Hill-based muscle model, 6) EMG processing module, and 7) User interface software. Among those, 2), 6) were achieved during the last semester and 7) is considered to be implemented in a Labview environment after all other parts are developed. Therefore, 4 groups have been involved in this project for this semester.

Group 1 (2 BMEN and 1 ISEN students) has contributed to develop the real-time Hill-based muscle model based on the references [1, 2]. The physiology based model outputs the generated muscle force from two inputs, the muscle kinematics and the muscle activation signals. The former is acquired from the output of the Varying moment arm computation module and the later gets the input from the output of the EMG processing module. In the end, a parameter optimization scheme will also be implemented to adjust the model parameter values for each individual by using a machine learning algorithm such as, genetic algorithm.

Group 2 (3 senior MEEN students) has been involved to develop the Dead Reckoning algorithm. The algorithm generates the position and orientation of each IMU unit with respect to the global frame. Since the IMU sensor outputs the linear accelerations and angular velocities, the strapdown integration method should be implemented to acquire the position and orientation angle information.

Group 3 (3 senior MEEN students) has developed an inverse dynamics module of the human arm. This module computes the required joint torque values for the captured motion kinematics. Here, the arm is assumed as a 4 DOFs (i.e. 3 DOFs shoulder and 1 DOF elbow) serial chain of 2 rigid links. In order to derive the equations of motion in 3D human arm motion, the Lagrangian method is utilized.

Group 4 (1 MEEN and 1 CSCE students) has developed the Varying moment arm computation module. This module allows us to obtain the muscle kinematics (i.e. excursion and its rate of change) from the captured joint kinematics. It is known that the moment arm value of human skeletal muscle is varying across the joint motion due to complex anatomical structures inside the body.
Major Accomplishments and Future Works

**Group 1: Hill-based Muscle Model**

In the end of the last semester, this group could come up with the model in a closed form of equation,

\[
0.1433 \cdot a \cdot (0.5\text{MPa}) \left[ \frac{m}{2p_y} \right] \sin(2\alpha) \cdot \exp \left[ -0.5 \left( \frac{\Delta L_{CE} [n] - \phi_m}{\phi_i} \right)^{2} \right]
\]

\[
0.1074 + \exp \left[ -1.3 \sinh \left( \frac{2.8}{0.5(a+1)(2L_{CE} + 8L_{CE} \beta)} \frac{\Delta L_{CE} [n] - \Delta L_{CE} [n-1]}{\Delta t} + 1.64 \right) \right]
\]

At first, they tried to solve (1) for the length change of the contractile element at the current time step, \( \Delta L_{CE} [n] \). During the procedure, they could obtain some advanced knowledge on the symbolic math functions in MATLAB. Despite their efforts, however, they finally concluded that it is impossible to solve the equation in a closed format due to the complexity and the interdependency of multiple variables within it.

From [1], the students could re-organize (1) into 5 pieces based on Fig. 2: 1) Length change of the series element (SE) \( \Delta L_{SE} \), 2) Length change of the contractile element (CE) \( \Delta L_{CE} \), 3) Force of the contractile element (CE) \( F_{CE} \), 4) Force of the parallel element (PE) \( F_{PE} \), and 5) Final force output calculation. The resulting equations are as following 5 equations:

\[
\Delta L_{SE} [n-1] = \frac{0.03L_{t}}{S_{SE}} \ln \left( \frac{\left( e^{\frac{3}{2}F_{CE}[n-1]} - 1 \right)F_{CE}[n-1]}{0.5F_{CE, max}} \right)
\]

where \( \Delta L_{SE}[n-1] \) and \( F_{CE}[n-1] \) are the length change of SE and the generated force of CE at n-1 time step, respectively. \( L_{t} \) represents the tendon slack length, \( S_{SE} \) is the shape factor of SE and \( F_{CE, max} \) indicates the maximum force value of the CE. Once the length change of SE is obtained from (2), the length change of CE becomes

\[
\Delta L_{CE} [n] = \Delta L_{CE} [n] - \Delta L_{SE} [n]
\]
where $\Delta L_{PE}[n]$ is the length change of PE at $n$ time step. Note that this value is identical to the length change of the muscle itself. With the neural activation input (i.e. processed EMG signal) $a[n]$ and the result of (3), the force output of CE can be derived as

\[
F_{CE}[n] = \frac{0.1433a[n] \cdot F_{CE_{\text{max}}} \cdot \exp \left\{-0.5 \cdot \frac{\Delta L_{CE}[n] - \phi_m}{\phi_v} \right\}}{0.1074 + \exp \left\{1.3 \sinh \left(\frac{2.8 \Delta L_{CE}[n]}{0.5 \cdot [a[n]+1] \cdot [2L_{CE} + 8aL_{CE} \Delta t] + 1.64}\right)\right\}}
\]

where $\phi_m$ and $\phi_v$ are the shape parameters of empirical curves. $\alpha$ refers the percentage of fast fibers in the selected muscle and $\Delta t$ is the sampling time of computation. $L_{CE0}$ is the rest length of CE. The force from PE is derived by

\[
F_{PE}[n] = \left\{\frac{0.05F_{CE_{\text{max}}}}{e^{S_{PE}} - 1}\right\} \left\{e^{-\frac{S_{PE}[n]}{L_{\text{max}}(4L_{CE}+2L_{CE})}} - 1\right\}
\]

where $S_{PE}$ is the shape parameter of PE and $L_{\text{max}}$ is the maximum length of the selected muscle. Note that the length change of PE, $\Delta L_{PE}[n]$, is the output of Varying Moment Arm module. Finally, the total muscle force can be expressed as the summation of (4) and (5) as

\[
F_{tot}[n] = F_{CE}[n] + F_{PE}[n].
\]

The developed algorithm was tested with a simple experiment. A sEMG unit was attached on the subject’s biceps.
brachii muscle and the subject tugged his hand under a desk and applying an elbow flexion torque against the desk while he maintains his elbow angle as 90 deg. The measured EMG signal was implemented in the developed algorithm. Due to the isometric condition, the muscle kinematics input was set as zero. The experimental result is plotted in Fig. 3. As shown in the figure, the computed muscular force value follows the trend of measured EMG magnitude. The same algorithm is expanded to other muscles (i.e. triceps longhead, anterior and posterior deltoids) and implemented in the demonstration video. The resulting force plot of each muscle seems somewhat unrealistic. This may due to lack of parameter adjustment scheme which has not been implemented yet. As a future work for this group, the genetic algorithm will be implemented to optimize the physiological parameters involved in the developed model.

**Group 2: Dead Reckoning Algorithm**

Each of the 9 DOF Shimmer IMU sensors consists of 3-axis accelerometer, 3-axis gyroscope and 3-axis magnetometer. The sensor readings represent the linear accelerations, angular velocities and direction vector of the magnetic field of the planet with respect to the local frame within the sensor unit. The goal of Group 2 is to translate those local values into the global frame and to integrate them to get the position and orientation information in the fixed global frame.

As an initial approach, the filtered sensor data was rotated via quaternion then integrated to get the desired values. A low-pass and a high-pass Butterworth filters were applied to the accelerometer outputs and the gyroscope outputs, respectively. The algorithm was implemented as a batch-mode processing (i.e. offline) algorithm in the MATLAB. To verify the computation, Vicon Motion Capture system was utilized to record the motion of 3 reflective markers attached on the IMU sensor (see Fig. 4). Prior to the experiment, the IMU sensor’s local frame was aligned with the Vicon global frame and the sensor local frame was defined in the Vicon measure via cross products among the positions of attached markers. As shown in Fig. 5, the IMU and Vicon simultaneously capture the motion during a movement. The rotation of the sensor local frame (see Fig. 6) with respect to the fixed global frame is defined as Z-Y-Z frame rotations as shown in Fig. 7. These frames can be constructed by their appropriate rotation matrices, which are given as

![Fig. 4. Vicon markers placed on IMU.](image)

![Fig. 5. Simultaneous Vicon and IMU motion test.](image)

![Fig. 6. Local frame defined on the sensor unit](image)

![Fig. 7. Three frame rotation to represent the local frame rotation with respect to the global frame.](image)
where \( XYZ \) and \( xyz \) represent the fixed global frame and the sensor local frame, respectively. Since the sensor unit’s orientation is changing with respect to time, these rotation matrices will be time-varying as well. To determine these values, the gyroscope measurements should be taken into account. The local angular velocities of the sensor can be defined as

\[
\omega_{xyz} = \begin{bmatrix}
\omega_x \\
\omega_y \\
\omega_z 
\end{bmatrix}
\]

where each component of the vector represents the angular velocity reading with respect to each axis of the gyroscope module. The angular velocity can also be represented as a function of three frame rotation angular rates (i.e. \( \dot{\psi}, \dot{\theta}, \) and \( \dot{\phi} \)) in a mixed coordinate system:

\[
\omega_{\text{mixed}} = \dot{\psi} \hat{K} + \dot{\theta} \hat{J} + \dot{\phi} \hat{k}.
\]

By applying appropriate rotation matrices, the mixed coordinate representation can be converted into the local frame, which is identical to the sensor measurement (8). The resulting equation becomes

\[
\omega_{xyz} = \dot{\psi} (R_{xyz}^{XY})^T \begin{bmatrix}
0 \\
0 \\
1
\end{bmatrix} + \dot{\theta} (R_{xyz}^{XY})^T \begin{bmatrix}
0 \\
1 \\
0
\end{bmatrix} + \dot{\phi} (R_{xyz}^{XY})^T \begin{bmatrix}
0 \\
0 \\
1
\end{bmatrix}.
\]

In order to avoid singularities, all rotation matrices above were converted into quaternion format in the MATLAB symbolic representation. Since (8) is equal to (10), we can solve two equations for the frame rotation angular rates \( \dot{\psi}, \dot{\theta}, \) and \( \dot{\phi} \) as using the MATLAB symbolic math:

\[
\dot{\psi} = -\frac{4\omega_y \sin \frac{\psi}{2} - 4\omega_x \sin \frac{3\psi}{2} - 5\omega_z \sin \frac{\psi}{2} + 2\omega_z \sin \frac{\psi}{2} + \omega_x}{\sin \theta \left(4 \sin^4 \frac{\psi}{2} - 3 \sin^2 \frac{\psi}{2} + 1\right)}
\]

\[
\dot{\theta} = \frac{4\omega_z \sin \frac{\psi}{2} + 4\omega_x \sin \frac{3\psi}{2} - 5\omega_y \sin \frac{\psi}{2} - 2\omega_x \sin \frac{\psi}{2} + \omega_y}{\left(4 \sin^4 \frac{\psi}{2} - 3 \sin^2 \frac{\psi}{2} + 1\right)^2}
\]

\[
\dot{\phi} = -\frac{2\omega_x \cos \theta - 2\omega_z \sin \theta}{2 \sin \theta} - \frac{\cos \psi \left(2\omega_z \cos \theta + 4\omega_x \sin \frac{\psi}{2} \cos \theta\right) - 2\omega_z \cos \theta}{\sin \theta \left(2 \cos^2 \psi - \cos \psi + 1\right)}.
\]

Here, since the frame rotation angles \( \psi, \theta, \) and \( \phi \) are still unknown, those values are approximated by the previous time step values as:
where $\Delta t$ is the sampling time. Then the frame rotation angular velocities can be represented in the global frame as

$$\omega_{XYZ} = \psi \hat{K} + \theta R_z(\psi) R_y(\theta) \hat{J} + \phi R_z(\psi) R_y(\theta) R_z(\phi) \hat{K}$$ \hspace{1cm} (15)$$

and the global sensor orientation can be computed in the same form as

$$\Theta_{XYZ} = \psi \hat{K} + \theta R_z(\psi) R_y(\theta) \hat{J} + \phi R_z(\psi) R_y(\theta) R_z(\phi) \hat{K}.$$ \hspace{1cm} (16)$$

The computation procedure within the algorithm is as follows:

1) Compute the frame rotation angular velocities (i.e. $\psi$, $\theta$ and $\phi$) by (11) to (13),
2) Update the frame rotation angles by (14),
3) Update the rotation quaternion and normalize it,
4) Compute the global accelerations by using the updated rotation quaternion and compensate the gravity acceleration,
5) Compute the global angular velocity by (15),
6) Compute the global velocities by using the computed global accelerations and global angular velocities,
7) Compute the global position value by numerical integration of global velocities, and
8) Compute the global sensor orientation value as (16).

The developed algorithm was applied to the captured experimental data of a random motion. The resulting 3D trajectory of IMU Dead Reckoning Algorithm is compared to the Vicon measures in Fig. 8(a). As shown in the figure, the developed algorithm does not track the reference trajectory (i.e. Vicon data) and blows up the resulting trajectory in a short moment. Fig. 8(b) shows the accelerations. The upper figure represents the low-pass filtered sensor measure (i.e. local accelerations) while the lower one plots the rotated and compensated accelerations (i.e. global accelerations without gravity effect). Fig. 8(c) compares the angular velocities. Upper figure represents the high-pass filtered sensor measure (i.e. local angular velocities) while the lower one shows the rotated angular velocities (i.e. global angular velocities).

Group 2 could not come up with a reasonably well-performed result from their algorithm. We could not figure out the exact problem in our code. In the future, more advanced methods such as, complementary Kalman Filter with quaternion model will be adopted for this project. In the demonstration video, the arm kinematics was captured via Vicon system instead of the proposed IMU-Dead Reckoning Algorithm.

**Group 3: Inverse Dynamics Module**

Since the human arm can be modeled as a serial chain of several rigid links, the required joint torque for a certain motion can be computed from the rigid body equations of motion (EOM). The inverse dynamics module is the system's
The EOM of multi-links in 3D space is quite complex. Therefore, we utilized the energy method (i.e. Lagrangian mechanics) rather than the conventional Newtonian method. Prior to develop the inverse dynamics module, the joint coordinate system is defined as Fig. 9.

The kinetic energy of each link is derived by

\[ KE_i = \frac{1}{2} m_i v_i^T v_i + \frac{1}{2} \omega_i^T I_i^0 \omega_i \]  \hspace{1cm} (17)

where \( m_i \) and \( I_i^0 \) refer the mass and the moment of inertia matrix of \( i \)-th link, respectively. \( v_i \) represents the velocity vector of COM of the link with respect to the global frame while \( \omega_i \) indicates the angular velocity vector of the link with respect to the global frame. Here, since \( I_i^0 \) is with respect to the global frame, it is time-varying according to the motion. For convenience, \( I_i^0 \) is represented in the local frame attached on the link (i.e. \( I_i^l \)) to get a constant matrix value and it is converted into the global frame by the rotation matrix as

\[ I_i^0 = (R_0^l)^T I_i^l (R_0^l). \]  \hspace{1cm} (18)

The potential energy of \( i \)-th link is computed as

\[ PE_i = m_i g x_{i,z} \]  \hspace{1cm} (19)

where \( g \) is the gravitational acceleration (=9.81 m/s\(^2\)) and \( x_{i,z} \) represents the vertical position component of the link COM. Then the EOM can be derived by the Lagrangian method

\[ \tau_i = \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} \]  \hspace{1cm} (20)

where \( q_i \) is the generalized coordinate (in our case, each joint DOF angle) and \( \tau_i \) is the corresponding joint torque. The Lagrangian \( L \) is defined as

\[ L = \sum KE_i - \sum PE_i. \]  \hspace{1cm} (21)

Due to the complexity of the computation, the EOM was derived by MATLAB symbolic math function.

Once the symbolic expression of EOM is derived, the actual joint torque values can be computed by putting the measured joint kinematics (i.e. joint angles, angular velocities and angular accelerations). Prior to that stage, the physical parameter values such as, mass, link length and moment of inertia of each link needs to be determined within the derived EOM. The link length and link radius can be measured on the human subject. However, there are some immeasurable parameter values; link mass, position of COM within the link and radius of gyration. All these parameter values can be approximated via anthropometric rules represented in Table I.

A random joint motion data was generated via manipulating a cosine function for each joint DOF and the developed inverse dynamics code was tested for the random motion data. The resulting plot is shown in Fig. 10. Note that the initial
torque values at time $t=0$ are due to non-zero angular acceleration values.

**Group 4: Varying Moment Arm Computation Module**

The human arm joint DOF motion is driven by muscle contraction. In other words, the contractile muscle force can be converted into the joint torque value with a certain moment arm. Generally, this moment arm value is varying along the motion and it can be derived as a function of joint DOF angle from the anatomical geometry of musculoskeletal model. Ideally, the moment arm values can be computed from the combination of anatomical data (i.e. insertion and origin points of the selected muscles, see [4-6]) and the geometry of each muscle wrapped around the obstacles modeled as spheres, cylinders, or combinations (see [7-9]). However, due to its complexities, Group 4 searched the previous researches on the empirical models (i.e. polynomial fit of the moment arm value with respect to each joint DOF angle) as an alternative method. Group 4 could find [10] for the shoulder muscles (i.e. anterior and posterior deltoids) and [8] for the elbow muscles (i.e. biceps brachii and triceps longhead). For the shoulder muscles, Group 4 contacted the authors of [11] and received the whole data set in a spreadsheet format.

As a result, the excursion length of the anterior deltoid (AD) is modeled as

$$E_{AD} = -0.0843 \text{cm} + \frac{0.00813 \text{cm}}{\text{Degree}} \zeta + \frac{0.0123 \text{cm}}{\text{Degree}} \theta - \frac{0.000429 \text{cm}}{\text{Degree}^2} \theta^2$$  \hspace{1cm} (22)

and the moment arm of AD with respect to the shoulder elevation and the humeral rotation angles become

$$r_{AD1} = 0.04955 \theta - 0.7367$$  \hspace{1cm} (23)

$$r_{AD2} = 0.47$$  \hspace{1cm} (24)

For the posterior deltoid (PD), the excursion length is
\[
E_{PD} = -0.038\text{cm} - \frac{0.00621\text{cm}}{\text{Degree}} \zeta + \frac{0.0954\text{cm}}{\text{Degree}} \theta - \frac{0.000558\text{cm}}{\text{Degree}^2} \theta^2
\]  

(25)

and the moment arm with respect to the shoulder elevation and the humeral rotation angles are

\[
E_{PD1} = 0.06405\theta - 5.503
\]

(26)

\[
E_{PD2} = 0.36.
\]

(27)

For the elbow flexion angle, the biceps brachii (BB) and the triceps longhead (TL) are involved. The moment arm of biceps brachii can be represented as

\[
r_{BB} = \begin{cases} 
0.2261\phi + 14.59 & \text{for } \phi \leq 30 \\
-\frac{1.268 \times 10^{-3} \text{mm}}{\text{Degree}^3} \phi^3 - \frac{3.146 \times 10^{-3} \text{mm}}{\text{Degree}^2} \phi^2 + \frac{0.9134 \text{mm}}{\text{Degree}} \phi - 3.118 \text{mm} & \text{for } \phi > 30
\end{cases}
\]

(28)

The moment arm of triceps longhead is

\[
r_{TL} = 1.375 \times 10^{-2} \phi^4 - 5.463 \times 10^{-5} \phi^3 + 7.676 \times 10^{-3} \phi^2 - 0.3383\phi - 21.82.
\]

(29)

The excursion length of BB and TL can be determined via simple equation:

\[
\Delta L = \Delta\theta \times r.
\]

(30)

The developed polynomial fitting equations are implemented in the demonstration video to compute the muscular force values. For more precise moment arm calculation, the original method (i.e. from the geometrical model of muscles based on the anatomical database) should be studied as a future work.

References


