Simulation of Motion Capture Data using the Port-Hamiltonian approach

By Brian Wilcox, Undergraduate Mechanical Engineering MIT

Advised by Josep-Maria Font Llagunes, UPC Biomechanical Engineering Group (BIOMEC) and

Arnau Dòria-Cerezo, Dept. Electrical Engineering, and Inst. of Industrial and Control Engineering

I. INTRODUCTION

Models and simulations of mechanical systems offer great tools to understand the behavior of physical movement. In this paper, we focus on the Port-Hamiltonian approach. This pH approach is viable for biomechanical systems because it is simply to connect multiple body segments, external assistive devices, actuators, and more under the same methodology. Under this Port-Hamiltonian formulation, we can conduct inverse dynamics, forward dynamics, and control design in a way that remains consistent with the fundamental framework and that is easily implemented in computer simulations. Throughout this paper, we have described a pH model of a simple biomechanical system. We show how the pH model is suitable for the simulation of human generated motion data.

II. MODEL

Below is a figure of the 2 link arm which we will be modeling

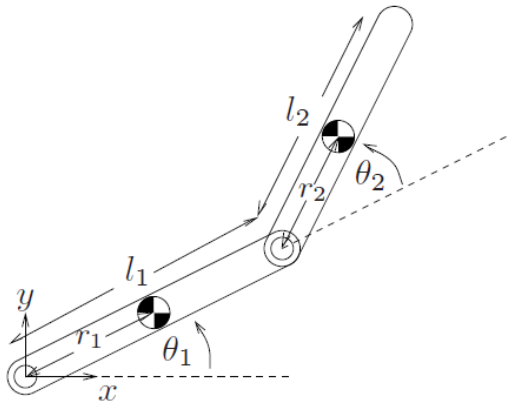

Figure 1: 2-link planar manipulator

Port-Hamiltonian for Arm link

We use the Port-Hamiltonian modeling approach because it allows us to model any component of a system based on its energy and connect multiple components together. Here we will use this approach to model each arm of our 2 link system.

The Hamiltonian involves the total kinetic and potential energies of the system:

$$H_i = T_i + V_i \tag{1}$$

$$T_i = \frac{p_{\theta i}^2}{2I_i} + \frac{p_{xi}^2}{2m_i} + \frac{p_{yi}^2}{2m_i} \tag{2}$$

$$V_i = m_1 g y_i \tag{3}$$

where H_i is the Hamiltonian, T_i is the kinetic energy, and V_i is the potential energy

$$p_{\theta i} = I_i \dot{\theta}_i \ , p_{xi} = m_i \dot{x}_i \ , p_{yi} = m_i \dot{y}_i$$

where I_i is moment of inertia of center of mass, m_i its mass

and x_i and y_i are coordinates of the center of mass in xy plane

From the Hamiltonian expression in (1), we can form the Port-Hamiltonian of the system:

$$\begin{bmatrix} \dot{\mathbf{q}}_\mathbf{i} \\ \dot{\mathbf{p}}_\mathbf{i} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial H_i}{\partial q_i} \\ \frac{\partial H_i}{\partial p_i} \end{bmatrix} + \begin{bmatrix} 0 \\ G \end{bmatrix} \mathbf{u} \tag{4}$$

where $\mathbf{q_i} = [\theta_i, x_i, y_i]$ and $\mathbf{p_i} = [p_{\theta i}, p_{xi}, p_{yi}]$

and G is a coordinate transformation vector for the input generalized forces or torques in u

where

$$\frac{\partial H_i}{\partial q_i} = \frac{\partial V_i}{\partial q_i} = m_i g \tag{5}$$

and

$$\frac{\partial H_i}{\partial p_i} = \dot{\mathbf{q}}_\mathbf{i} \tag{6}$$

2-link Arm Interconnection

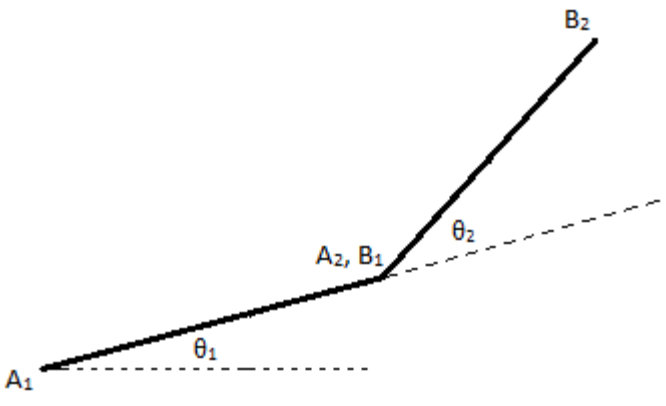

Figure 2: 2-link arm Interconnection

In order to connect our two links, we can define the following constraints:

Connected at A_2 and B_1

$$(x_{A2},\ \dot{x}_{A2},\ y_{A2},\ \dot{y}_{A2}) = (x_{B1},\ \dot{x}_{B1},\ y_{B1},\ \dot{y}_{B1}) \tag{7}$$

$$F_x^{B1} = -F_x^{A2}$$

$$F_y^{B1} = -F_y^{A2}$$

Fixed at A_1

$$(x_{A1},\ \dot{x}_{A1},\ y_{A1},\ \dot{y}_{A1}) = (0, 0, 0, 0) \tag{8}$$

Relate q_i:

$$x_1 = r_1 \cos\theta_1 \tag{9}$$

$$y_1 = r_1 \sin\theta_1$$

$$\dot{x}_1 = -r_1 \sin\theta_1\, \dot{\theta}_1$$

$$\dot{y}_1 = r_1 \cos\theta_1\, \dot{\theta}_1$$

$$x_2 = l_1 \cos\theta_1 + r_2 \cos(\theta_1 + \theta_2)$$

$$y_2 = l_1 \sin\theta_1 + r_2 \sin(\theta_1 + \theta_2)$$

$$\dot{x}_2 = -(l_1 \sin\theta_1 + r_2 \sin(\theta_1 + \theta_2))\dot{\theta}_1 - r_2 \sin(\theta_1 + \theta_2)\, \dot{\theta}_2$$

$$\dot{y}_2 = (l_1 \cos\theta_1 + r_2 \cos(\theta_1 + \theta_2))\dot{\theta}_1 + r_2 \cos(\theta_1 + \theta_2)\dot{\theta}_2$$

In order to simplify algebra later on:

$$a = -r_1 \sin\theta_1 \qquad (10)$$

$$b = r_1 \cos\theta_1$$

$$c = -(l_1 \sin\theta_1 + r_2 \sin(\theta_1 + \theta_2))$$

$$d = -r_2 \sin(\theta_1 + \theta_2)$$

$$e = (l_1 \cos\theta_1 + r_2 \cos(\theta_1 + \theta_2))$$

$$f = r_2 \cos(\theta_1 + \theta_2)$$

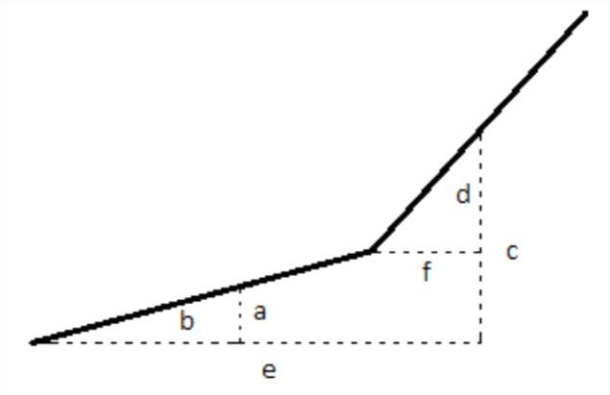

Figure 3: Components of 2-link arm

III. DYNAMICS

Previously, we eliminated the constraints in our port-Hamiltonian system in order to convert our pH system into standard form. Here, we will derive the dynamic equations in port-Hamiltonian form. We'll start with the general Hamiltonian for a mechanical system:

$$H(q,p) = \frac{1}{2}\mathbf{p^T M^{-1}(q)p} + \mathbf{V(q)} \qquad (11)$$

$where\ H(q,p)\ is\ the\ Hamiltonian\ , p\ the\ momentum, M(q)\ the\ Mass\ Matrix, V(q) the\ potential\ energy$

Since we are interconnecting two systems, we must add their Hamiltonians:

$$H_{12} = H_1 + H_2 = \frac{1}{2}\mathbf{p_{12}}^{\mathbf{T}}\mathbf{M_{12}^{-1}p_{12}} + \mathbf{V_{12}} \qquad (12)$$

$$\text{where } \mathbf{p_{12}} = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix}, \mathbf{V_{12}} = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}$$

And the diagonal mass matrix, $\mathbf{M_{12}} = \begin{bmatrix} \mathbf{M_1} & 0 \\ 0 & \mathbf{M_2} \end{bmatrix}$, where each link's mass matrix $\mathbf{M_1}$, and $\mathbf{M_2}$ is constructed such that $\mathbf{p} = \mathbf{M}\dot{\mathbf{q}}$ for all coordinate states. Therefore, $\left[p_{\theta i}, p_{xi}, p_{yi}\right] = \mathbf{M_i q_i}$,where $\mathbf{q_i} = [\theta_i, x_i, y_i]$

For our system, we've been using 3 coordinate states to describe each arm ($\mathbf{q_i} = [\theta_i, x_i, y_i]$). However, because x_i and y_i depend on θ_i, we can fully describe our system with θ_i. Therefore, our 2-link system can be reduced from 12 states to 4 states (θ_1 and $\theta_2$and related momentum p_1 and p_2).

Therefore, we must derive a pH system with $\mathbf{q} = [\theta_1, \theta_2]^{\mathrm{T}}$ and $\mathbf{p} = \mathbf{M}(\mathbf{q})\dot{\mathbf{q}}$

In order to do this, we must reduce our states through the kinematic constraints based on the interconnection of the two links. Let's construct a matrix, A, that contains the constraints, such that

$$\dot{\mathbf{q}}_\mathbf{i} = \mathbf{A}\dot{\mathbf{z}} \qquad (13)$$

$$where\ \dot{\mathbf{z}} = \left[\dot{\theta}_1, \dot{\theta}_2\right]^{\mathrm{T}}\ and\ \mathbf{A}(\theta_1, \theta_2)$$

We may find the matrix **A** as the Jacobian of the states $\mathbf{q_i}$ such that

$$A = \mathbf{J_q}(\theta_1, \theta_2) = \begin{bmatrix} \frac{\partial q_1}{\partial \theta_1} & \frac{\partial q_1}{\partial \theta_2} \\ \frac{\partial q_2}{\partial \theta_1} & \frac{\partial q_2}{\partial \theta_2} \end{bmatrix} \qquad (14)$$

Where $\mathbf{J_q}$ is the Jacobian matrix of states $\mathbf{q_i}\ and\ \mathbf{q_1} = [\theta_1, x_1, y_1]\ \ and\ \mathbf{q_2} = [\theta_2, x_2, y_2]\ and\ \mathbf{q_i} = [\theta_i, x_i, y_i]$

We have now related our 6 coordinates to our desired 2 states.

The kinetic energy of the original states is:

$$T(q, p) = \frac{1}{2}\dot{\mathbf{q}}_\mathbf{i}^{\mathbf{T}}\mathbf{M}\dot{\mathbf{q}}_\mathbf{i} \qquad (15)$$

Where **M** is the diagonal matrix, in our study $\mathbf{M_{12}}$

Substituting (13) into (15):

$$T = \frac{1}{2}\dot{\mathbf{z}}^{\mathbf{T}}\mathbf{A}^{\mathbf{T}}\mathbf{M}\mathbf{A}\dot{\mathbf{z}}$$

$$M_* = \mathbf{A}^{\mathbf{T}}\mathbf{M}\mathbf{A}$$

$$T = \frac{1}{2}\dot{\mathbf{z}}^{\mathbf{T}}\mathbf{M}_*\dot{\mathbf{z}} \quad (16)$$

This reduces our states to the desired $\dot{\mathbf{q}} = \dot{\mathbf{z}} = \left[\dot{\theta}_1, \dot{\theta}_2\right]^{\mathrm{T}}$.

Now, to recover the Hamiltonian, we set

$$\mathbf{p} = \mathbf{M}_*\dot{\mathbf{z}} \quad (17)$$

And recover

$$H(q,p) = \frac{1}{2}\mathbf{p}^{\mathbf{T}}\mathbf{M}_*{}^{-\mathbf{1}}(\mathbf{q})\mathbf{p} + \mathbf{V}(\mathbf{q}) \quad (18)$$

Now we've obtained a new mass matrix $\mathbf{M}_*$, that interconnects the system

Explicitly, we computed $A\ and\ M$ below (using (10) from earlier to simplify)

$$A = \begin{bmatrix} 1 & 0 \\ a & 0 \\ b & 0 \\ 1 & 1 \\ c & d \\ e & f \end{bmatrix} and\ M = \begin{bmatrix} I_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & m_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & m_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & m_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & m_2 \end{bmatrix} \quad (19)$$

*Note that for the angular momentum and rotational kinetic energy of the links, more generally, $p_{\theta i} = I_i\omega_i\ and\ T_{rot} = I_i\omega_i^2\ where\ \omega_1 = \dot{\theta}_1\ and\ \omega_2 = \dot{\theta}_1 + \dot{\theta}_2$. This accounts for the angular velocity contribution from link 1 to link 2.

In order to calculate $\mathbf{M}_*$

$$\mathbf{M}_* = \begin{bmatrix} m_1(a^2+b^2) + m_2(c^2+e^2) + I_1 + I_2 & m_2(cd+ef) + I_2 \\ m_2(cd+ef) + I_2 & m_2(d^2+f^2) + I_2 \end{bmatrix} \quad (20)$$

When simplified, this mass matrix matches the mass matrix seen in models (same 2-link arms) found in other academic papers. One model (Dirksz and Sherpen, 2012) comes in the same form as the pH model explained here with mass matrix they've defined

$$a_1 = m_1 r_1^2 + m_2 l_1^2 + I_1 \qquad (21)$$
$$a_2 = m_2 r_2^2 + I_2$$
$$b = m_2 l_1 r_2$$

The mass-inertia matric becomes

$$\mathbf{M}_* = \begin{bmatrix} a_1 + a_2 + 2b\cos\theta_2 & a_2 + b\cos\theta_2 \\ a_2 + b\cos\theta_2 & a_2 \end{bmatrix} \qquad (3)$$

Which is equivalent to the Mass matrix in (20)

We next will solve for the dynamics in Port-Hamiltonian form, starting with the standard mechanical system form

$$\begin{bmatrix} \dot{\mathbf{q}}_\mathbf{i} \\ \dot{\mathbf{p}}_\mathbf{i} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -I & -R(q,p) \end{bmatrix} \begin{bmatrix} \frac{\partial H}{\partial q_i} \\ \frac{\partial H}{\partial p_i} \end{bmatrix} + \begin{bmatrix} 0 \\ G \end{bmatrix} u \qquad (22)$$

Becomes

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{p}_1 \\ \dot{p}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial H}{\partial q_1} \\ \frac{\partial H}{\partial q_2} \\ \frac{\partial H}{\partial p_1} \\ \frac{\partial H}{\partial p_2} \end{bmatrix} + \begin{bmatrix} 0 \\ G \end{bmatrix} u \qquad (23)$$

$$where\ R(q,p) = 0\ \ (neglecting\ friction), \frac{\partial H}{\partial p_1} = \dot{q}_1\ and\ \frac{\partial H}{\partial p_2} = \dot{q}_2$$

And

$$\frac{\partial H}{\partial q_1} = \frac{1}{2} p^{\mathrm{T}} \frac{\partial {M_*}^{-1}(q)}{\partial q_1} p + \frac{\partial V}{\partial q_1} \quad (24)$$

$$\frac{\partial H}{\partial q_2} = \frac{1}{2} p^{\mathrm{T}} \frac{\partial {M_*}^{-1}(q)}{\partial q_2} p + \frac{\partial V}{\partial q_2} \quad (25)$$

Also, note that $\dot{\mathbf{q}} = {\mathbf{M}_*}^{-1}\mathbf{p}$

These equations are now sufficient to solve for the dynamics of the system.

IV. MOTION CAPTURE

For the purpose of providing realistic trajectories for our arm model to simulate, we captured the motion of a human subject in the UPC Biomechanics Lab at ETSEIB.

The passive optical system in the lab uses markers made of retro-reflective material to reflect light that is generated near the cameras lens. These markers are illuminated using Infra-red (IR) lights mounted on the cameras. The markers are attached directly to the skin or surface of the subject.

The system in the UPC Biomechanics Lab uses 18 OptiTrack V100 IR cameras placed around the lab, surrounding the space where experiments are taken. For capture of the arm, three passive markers were attached to the skin surface of the subject studied, one on the shoulder joint, elbow joint, and the wrist joint. After capturing the motion, the software ARENA processes the motion using data of all the cameras and adjusts the data to obtain 3 dimensional trajectories of the markers. This trajectory is then saved and converted for analysis within the software MATLAB.

V. INVERSE KINEMATICS

After obtaining the trajectory data from the motion capture, we need to determine the joint angles. For our 2D model, we take the position data in the plane of motion (x-y plane). We then calculate the absolute angles, θ_i, for each step in the motion such that

$$\theta_i = \tan^{-1}\left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i}\right) \quad (26)$$

Since we are looking at the relative angle of the joint, we assign our generalized coordinate in our system such that

$$q_i = \theta_i - \theta_{i-1} \quad (27)$$

For $i = 1, \dots, k$

Note that $\theta_0 = 0$ in our system with k links.

Once we have the positions for our generalized coordinates, we may take its derivative to find the velocity and once again for acceleration

$$\frac{d}{dt} q = \dot{q} \quad (28)$$
$$\frac{d}{dt} \dot{q} = \ddot{q}$$

In Matlab, we employ *atan2*, the four quadrant inverse tangent, in order to solve for the joint angles.

For velocity and acceleration, we use the numerical differentiation functions *diff* with a step size *h* equal to the time step used in the motion capture.

$$q_1 = \theta_1 \quad (29)$$
$$q_2 = \theta_2 - q_1$$

VI. BIOMECHANICAL PARAMETERS ESTIMATION

To describe a realistic model of a human arm, we approximated values for the dimensions and mass of the subject's arm using accepted biomechanical ratio approximations for humans. The following ratios are provided from the biomechanics textbook Biomechanics and Motor Control of Human Movement by David Winter (1990)

Ratios	Variable	Value
Upper arm weight/ body weight	K1	0.028
Forearm-hand weight/ body weight	K2	0.022
Upper arm center of mass/ segment length	C1	0.436
forearm-hand center of mass/ segment length	C2	0.682
upper arm radius of gyration/ segment length	D1	0.322
forearm-hand radius of gyration/ segment length	D2	0.468

Using the ratios above and direct measurements of the subject's mass and arm lengths, we obtain the needed arm parameter dimensions below:

Parameter	Variable	Value
Mass of body (whole)	W	75 kg
Mass of arm upper arm	M1	K1*W =2.1 kg
Mass of forearm-hand	M2	K2*W = 1.65 kg
Length of upper arm	L1	0.29 m
Length of forearm-hand	L2	0.29 m
Center of mass upper arm	R1	C1*L1 = 0.1264 m
Center of mass forearm-hand	R2	C2*L2 = 0.1978 m
Inertia of Upper arm	I1	M1*(L1*D1)^2 = 0.0183 kg*m^2
Inertia of forearm-hand	I2	M2*(L2*D2)^2 = 0.0304 kg*m^2

VII. COORDINATE CORRECTION

The model used for our system assumes rigid body links. While the motion capture data provides the trajectory of the system, noise and the position of markers, which are affected by skin movement and other factors, reveal data that is not consistent with our rigid body model. The rigid body constraints of our system relate the natural coordinates (positions) with the properties and geometry of the rigid body. In addition, the joint constraints relate the relative motion of the links that are interconnected. A method to correct the raw data in order to fit our rigid body model is described by Seemann, Stelzner, and Simondis (2005) in a paper on the correction of motion capture data. In this method, the constraints, C, should satisfy the condition,

$$\mathbf{C}(\mathbf{q}) = \mathbf{0} \tag{30}$$

In our system, we have the following constraints,

$$\mathbf{C} = \begin{bmatrix} l_1 \cos q_1 - x_1 \\ l_1 \sin q_1 - y_1 \\ l_1 \cos q_1 + l_2 \cos(q_1 + q_2) - x_2 \\ l_1 \sin q_1 + l_2 \sin(q_1 + q_2) - y_2 \end{bmatrix} \tag{31}$$

However, as mentioned, the measured coordinates, q_m , solved through inverse kinematics don't satisfy this condition,

$$\mathbf{C}(\mathbf{q_m}) \neq \mathbf{0}$$

Therefore, in order to fit the coordinates within the constraints of our rigid body model, we utilized methods, as outlined by Seemann, Stelzner, and Simonidis in their work on motion capture data, in order to generate "corrected" coordinates. Their resulting equation is

$$\mathbf{q_{i,r+1}} = \mathbf{q_{i,r}} - \left[\mathbf{C_q}(\mathbf{q_{i,r}})\right]^{-1}\mathbf{C}(\mathbf{q_{i,r}}) \quad (32)$$

Where $q_{i,r=0} = q_{i,m}$, and $C_q(q_{i,r})$ is the Jacobian matrx of $C(q_{i,r})$.

This method calculates the *ith* term of the new position set, *r+1*, by calculating the difference between the *ith* term of the initial data set, *r*, and the rigid body kinematic error. Then, each iteration of this solution reduces the rigid body kinematic error such that the constraints, $C(q)$, converge to zero. Note that the solution converges after two or 3 iterations.

Once we have the corrected position coordinates, which we will now refer to as $\tilde{q}_i$, we must correct the velocity and acceleration such that the following constraints are also satisfied

$$\mathbf{C_q}(\tilde{q}_i, t)\dot{\tilde{\boldsymbol{q}}}_{\boldsymbol{i}} = \mathbf{C_t}(\tilde{q}_i, t) \quad (33)$$

$$\mathbf{C_q}(\tilde{q}_i, t)\ddot{\tilde{\mathbf{q}}}_{\mathbf{i}} = \mathbf{Q_d}(\tilde{q}_i, \dot{\tilde{q}}_i, t) \quad (34)$$

Here we are ensuring kinematic consistency by saying that the rigid body constraints of the velocity terms must be maintained after differentiating the new corrected position coordinates.

Note that the terms C_t and Q_d follow from differentiation of the constraints:

$$\mathbf{C_t}(q, t) = \mathbf{C_q}(q, t)\dot{\mathbf{q}} \quad (35)$$

$$\mathbf{Q_d}(q, \dot{q}, t) = \mathbf{C_q}(q, t)\ddot{\mathbf{q}} \quad (36)$$

After applying these methods in Matlab, we now have obtained corrected coordinates and their derivatives that are consistent with our rigid body model.

VIII. INVERSE DYNAMICS

A goal of this paper was to determine the input forces/torques produced by the motion that we captured in the lab. In order to do this, we must solve for the inverse dynamics of the motion. The port-Hamiltonian form is useful to solve this problem.

We know that the dynamics of a mechanical system are described by

$$\dot{\mathbf{p}} = -\frac{\partial H}{\partial q} - R(q,p)\frac{\partial H}{\partial p} + \mathbf{u} \quad (37)$$

Our goal is to solve for the inputs, therefore

$$\mathbf{u} = \dot{\mathbf{p}} + \frac{\partial H}{\partial q} + R(q,p)\frac{\partial H}{\partial p} \quad (38)$$

In our example, we use

$$\mathbf{u} = \mathbf{M}_*(\mathbf{q})\ddot{\mathbf{q}} + \frac{\mathbf{1}}{\mathbf{2}}\mathbf{p}^{\mathbf{T}}\frac{\partial \mathbf{M}_*^{-\mathbf{1}}(\mathbf{q})}{\partial \mathbf{q}}\mathbf{p} + \frac{\partial \mathbf{V}}{\partial \mathbf{q}} \quad (39)$$

So long as the Mass matrix, M_*, and the inverse kinematics (including velocity and acceleration) of the generalized coordinates are known, we can simply solve for the input joint torques using Matlab.

IX. FORWARD DYNAMICS SIMULATION

Now that we have obtained the input torques for the motion captured in lab, we want to forward simulate the motion of the arm using our port-Hamiltonian model.

We want to use the ode23s solver in Matlab to simulate the model. Since we are using the pH model, we must define the states in Matlab that we intend to integrate as $\dot{\mathbf{q}}$ and $\dot{\mathbf{p}}$. This means that the ode23s solver will numerically integrate to find the position, $\mathbf{q}$,and momentum, $\mathbf{p}$,of each arm link.

Remember from (22)

$$\begin{bmatrix}\dot{\mathbf{q}}_\mathbf{i}\\ \dot{\mathbf{p}}_\mathbf{i}\end{bmatrix} = \begin{bmatrix}0 & I\\ -I & -R(q,p)\end{bmatrix}\begin{bmatrix}\frac{\partial H}{\partial q_i}\\ \frac{\partial H}{\partial p_i}\end{bmatrix} + \begin{bmatrix}0\\ \mathbf{G}\end{bmatrix}\mathbf{u}$$

And from (18)

$$H(q,p) = \frac{1}{2}\mathbf{p}^{\mathbf{T}}\mathbf{M}_*^{-\mathbf{1}}(\mathbf{q})\mathbf{p} + \mathbf{V}(\mathbf{q})$$

Therefore, in Matlab, we must create the states to be integrated such that

$$\dot{\mathbf{q}} = \frac{\partial H}{\partial p_i} = \mathbf{M}_*^{-\mathbf{1}}\mathbf{p} \quad (40)$$

And

$$\dot{\mathbf{p}}_\mathbf{i} = -\frac{\partial H}{\partial q_i} + \mathbf{u} \qquad (41)$$

Where we must construct the Mass Matrix, $\mathbf{M}_*$ and $\frac{\partial H}{\partial q_i}$ in Matlab using the integrated states $\mathbf{q}$ and $\mathbf{p}$, respectively. In order to apply forward simulation to our model, we must also read in the inputs, u , that we obtained during inverse dynamics for every step.

Note that the initial points of the simulation should correspond to the initial values of the position and momentum of the corrected motion data.

After simulation, we plotted the position response, $\mathbf{q}$, and compared it to the corrected positions. The simulation follows the motion for a couple seconds before the error sharply grows. This is likely due to errors involved with numerical integration.

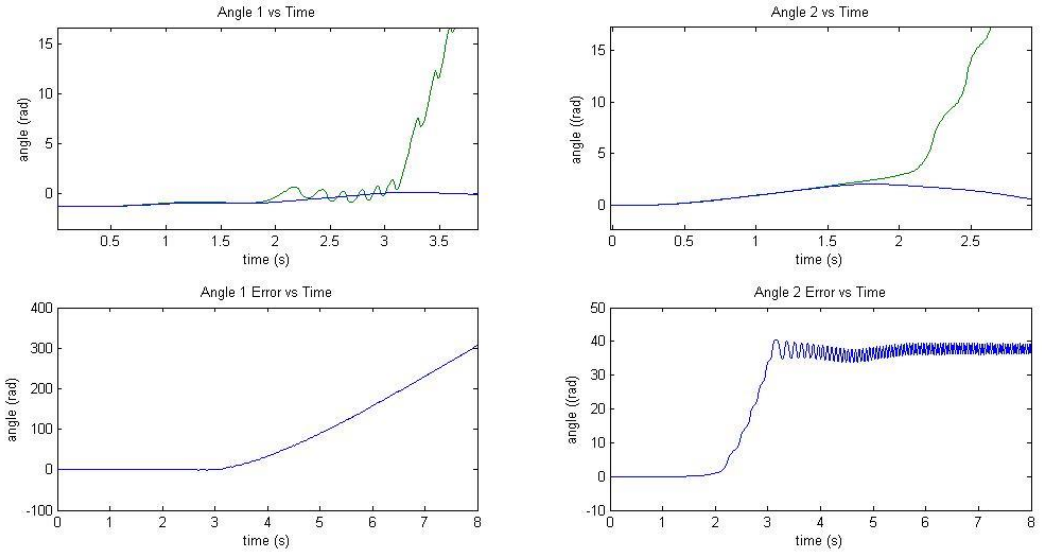

Figure 4: Plot of forward simulation results (green) vs the corrected reference trajectory (blue) with error plots

The above figure shows the simulation results from one of the motion capture trials conducted in the lab. The forward simulation results only hold accurate for about the first 2-3 seconds before accumulating significant errors. The errors grow once the slope of the reference motion increases greatly. This indicates that as the motion becomes sharper, the MATLAB ode solver increases in tracking

error. Our goal is to next improve the simulation by creating a control design that will solve for this error.

X. CONTROL DESIGN

As we have seen, even with correct input torques, forward simulation with numerical integration in Matlab suffers errors the longer the simulation time. This is an issue for any simulation that hopes to produce a full motion. Here we explore a couple control design methods that will help the simulation track the trajectory of the reference motion.

PD Control

We first look at a PD control design, as used in a thesis by Olga Pätkau (2014), that tracks the difference between the reference position and the generalized coordinates. The control design stabilizes the simulation by acting like springs and dampers on the joints.

The PD controller adds torque in order to produce the desired reference motion.

The new input, $\mathbf{u}$, becomes

$$\mathbf{u} = Q_{in} = Q_{IDA} + Q_{PD} \quad (42)$$

Where Q_{IDA} is the inputs we solved for in our inverse dynamic analysis.

Q_{PD}, the inputs from the controller, are determined using

$$Q_{PD} = K_p e_q + K_d \dot{e}_q \quad (43)$$

Where K_p is the stiffness matrix and K_d the damping matrix.

And where the tracking error is

$$e_q = \left(q_{ref} - q\right)^{\mathrm{T}} \quad (44)$$

$$\dot{e}_q = \left(\dot{q}_{ref} - \dot{q}\right)^{\mathrm{T}} \quad (45)$$

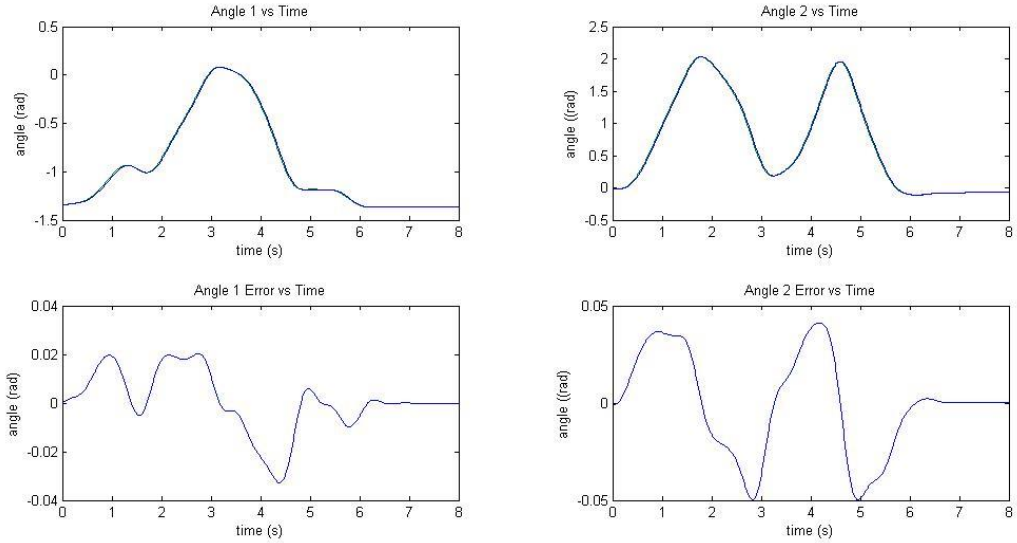

Figure 5: Plot of PD Control simulation (green) vs corrected reference trajectory (blue) with error plots

The plots in Figure 5 show significant improvement from the forward simulation. The simulation tracks the reference motion with nice accuracy. We should be careful to note that, given a wrong initial vector, the forward dynamics simulation would fail. However, with proper tuning of the gains, K_p and K_d, the controller should be able to stabilize the simulation to follow the reference motion.

Alternative Control Design Method

Another control design method described by Dirksz and Scherpen (2013) demonstrates tracking control on rigid-joint robots with only position measurements. This method has an advantage over the PD controller because it only requires the position of the motion. Their method is derived by transforming the PH system into another PH system in terms of the tracking error.

The transformation for the general pH system results changes the coordinates such that

$$\bar{q} = q - q_d(t) \tag{46}$$

$$\bar{p} = p - M(q)\dot{q}_d(t) \tag{47}$$

Where q_d refers to the reference motion. $\bar{q}$ and $\bar{p}$ are tracking errors for the system.

The new Hamiltonian becomes

$$\bar{H} = \frac{1}{2}p^{\mathrm{T}}M^{-1}(q)p + \frac{1}{2}\bar{q}^{\mathrm{T}}K_p\bar{q} \tag{48}$$

Their method defines controller dynamics by

$$\dot{q}_c = K_d^{-1} K_c(\bar{q} - q_c) \tag{49}$$

And the state v, where

$$v = K_c(\bar{q} - q_c) \tag{50}$$

With the final control input realized by the equation

$$u = M(q)\ddot{q}_d + \frac{\partial(M(q)\dot{q}_d)}{\partial q}\dot{q}_d - \frac{1}{2}\frac{\partial \dot{q}_d{}^{\mathrm{T}} M(q)\dot{q}_d}{\partial q} + \rho(q) - K_p\bar{q} - v \tag{51}$$

The control gains used were taken from the paper by Dirksz and Scherpen (2013) on Tracking Control:

$$K_p = \begin{bmatrix} 80 & 0 \\ 0 & 80 \end{bmatrix}, \quad K_i = \begin{bmatrix} 50 & 0 \\ 0 & 50 \end{bmatrix}$$
$$K_d = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}, \quad K_c = \begin{bmatrix} 190 & 0 \\ 0 & 190 \end{bmatrix} \tag{52}$$

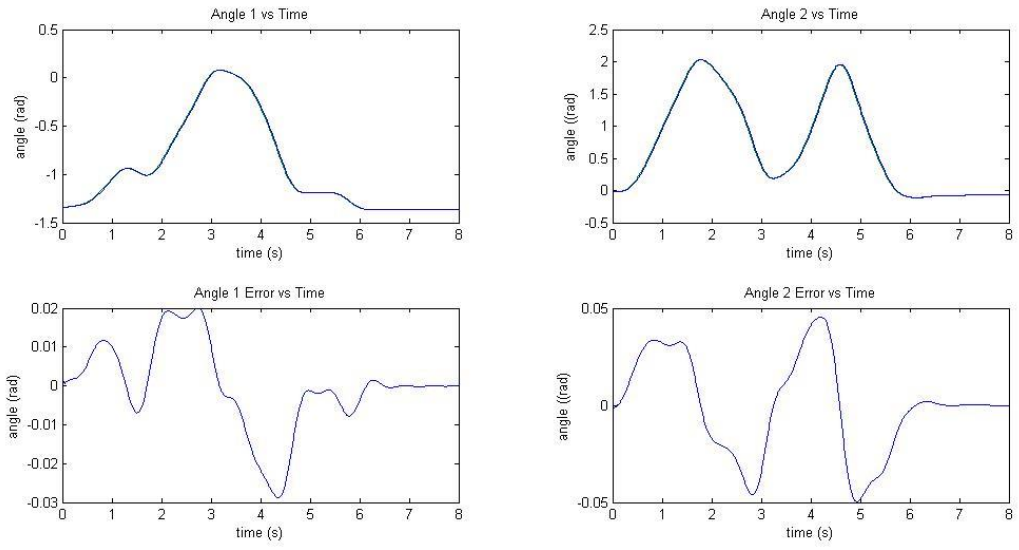

Figure 6: Plot of the Dirksz and Scherpen control simulation (green) vs the corrected reference trajectory (blue) with error plots

The controlled simulation using this alternative method performs well with good accuracy for the duration of the motion.

XI. RESULTS AND DISCUSSION

The simulation of motion capture data under the pH approach yields successful results with tracking control. For biomechanical systems, we must be careful about forming the appropriate mass matrix as well as correcting our data for rigid body motion. However, we still observe that, even though correct inverse dynamics is easily obtained under the pH model, forward simulation in Matlab suffers errors as a result of numerical integration. While PD control is a common, and successful, method for reducing the tracking error, we find that the control design described by Dirksz and Scherpen to offer the advantage of transforming the Port-Hamiltonian formulation of the system into a new form that is easily implemented without having to perform inverse dynamics directly. This paper has yet to explore the tuning of the control gains to improve the performance, but we should expect to see an improvement. Although not explored in this paper, the pH model can be adapted to include friction effects (if accurately modeled) or include biomechanical muscle activation forces as an input. This could be a future application of this pH process for simulating the motion capture of human motion.

REFERENCES

1. D. Dirksz, J. Sherpen
   On Tracking Control of Rigid-Joint Robots With Only Position Measurements
   IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, VOL. 21, NO. 4, JULY 2013

2. D. Winter. Biomechanics and Motor Control of Human Movement. Wiley. 1990

3. O. Pätkau
   Application of different Control Strategies to the FD Simulation of Human Gait

4. W. Seemann, G. Stelzner, C. Simonidis
   Correction of Motion Capture Data with Respect to Kinematic Data Consistency for Inverse Dynamic Analysis
   ASME 2005 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, September 24-28, 2005