Human Interactive Pose Prediction Over Path Observed Through A Multibody Underactuated System (HIPPOPOTAMUS)

Matt McDermott

Abstract:

The goal of this project was to devise a novel technique to estimate the pose of a human user as they guide a compliant robotic arm through an arbitrary trajectory without the use of force sensors, inertial measurement units or a vision system. The forces of gravity, friction and inertia of a 3 DoF manipulator are cancelled out, decoupling movement in the x, y and z world frame axis and turning the end effector of the robot into a floating endpoint force detection machine. Using information about the bias of human strength as a function of position of the user's hand relative to the shoulder, trajectory data from the hand guided path is used to calculate cartesian components of forces and estimate the position of the human's shoulder relative to the base of the robot.

Introduction:

Most modern human pose estimation techniques utilize vision systems or in some cases inertial sensors fixed to various points on the human arm. These techniques have a number of advantages but share a common weakness in the fact that they rely on highly controlled conditions in order to function properly. For instance, in order to overcome some of the issues associated with people and objects being obscured in a vision system, multiple cameras must be set up surrounding the workspace. In the case of inertial sensors, the system will fail entirely if a user does not properly attach the sensors to their body. Inertial sensors are also extremely prone to drift over long periods of time. *Roetenberg et al.* quantify this drift to between 10 and 25 degrees error after 60 seconds of accelerometer measurement without outside correction [3].

The alternative strategy proposed in this paper involves using inertial parameters of the robot combined with known strength characteristics of the human arm to formulate an estimate of human pose without the use of external vision or inertial tracking equipment. A sufficient implementation of this technique has the potential to provide robotic systems with the knowledge of where their human user is located without the need for additional hardware. This means that this estimation strategy can be retroactively added to existing systems and provide insight into human pose with minimal implementation cost.

For instance, in the case of a collaborative human-robot team working in the same manufacturing cell, data taken from a manipulator's trajectory which can estimate where the human worker is standing could be used to update future path planning in order to make sure that future operations do not get in the way of the human's work space. Alternatively, if the system senses the human is guiding the robot through a path from outside their usual work zone, it could indicate to the robot that there is an obstruction somewhere in the workspace and to proceed with a higher level of caution.

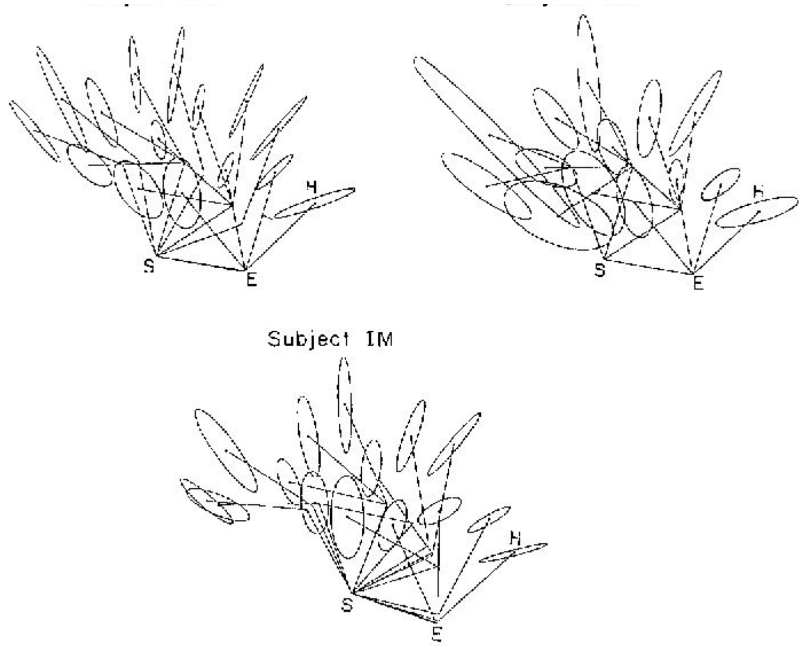
Another interesting application for this pose estimation strategy is in the case of rigid upper body exoskeletons. Because of the number of degrees of freedom in the human shoulder, it is difficult to design an exoskeletal frame that can move as freely as the human shoulder while still maintaining rotation about the same axis as the revolute joints of a human. For a system that enhances human joint movement it is of extreme importance to know the true state of each human joint so that movements can be computed correctly. Currently, joints of an exoskeleton must either rotate about the same axis as those of the human, which severely limits shoulder mobility, or the exoskeleton must include either a vision or accelerometer based human pose estimation system so that movements induced by the human can be interpreted correctly.

For instance, one motion in world space that would be completely safe for a human at one elbow position could be very dangerous for the user if their elbow is oriented differently. Without a robust system of pose estimation, an exoskeleton with misaligned joints could accidentally break a user's elbow if pose is estimated incorrectly. The implementation of a force based human pose estimation system in a rigid upper body exoskeleton would allow non-aligned joints without the need for external tracking hardware. This has the potential to improve the robustness and usability of exoskeleton systems for both industrial and rehabilitation applications.

Background and Related Work:

Flash and Mussa-Ivaidi describe in detail the inertial and force characteristics of the human arm as a function of hand position relative to the shoulder [1]. The planar case of the human arm stiffness ellipse described in this study has its major axis pointed directly in the direction of the shoulder regardless of the position of the hand. Intuitively, this makes sense as a human generally has much more strength when pushing outwards away from the body than moving perpendicular to the direction of the shoulder. This is the reason why most humans can lift much higher weight in a bench press than in a butterfly exercise.

Fig. 1- Human Arm Stiffness Ellipse data from Flash and Mussa-Ivaidi



There have been sufficient works studying various methods of human pose estimation. Most notably, *Human Joint Angle Estimation and Validation with a Robot Arm*, describes a use of an Unscented Kalman Filter as well as Zero Velocity Updates to avoid sensor drift [2]. Although taking a different approach than the force based implementation described here, some characteristics such as the workflow described to estimate arm inertia are applicable in both cases.

Methodology:

The decision to use a homemade robotic arm was largely due to the accessibility of the hardware. Ideally, a fully backdrivable off the shelf product such as the Barrett WAM arm would have been used as system parameters are already well known and existing models have been devised to cancel the dynamics of the robot. Unfortunately, such systems are prohibitively expensive and building a direct drive manipulator from scratch was the next best thing.

In order to determine the forces exerted by the human arm on the robot, the forces of gravity, friction and robot inertia must first be cancelled out. While it may have been possible to calculate these forces and subtract them after motions are performed on the arm without active force cancellation, it is likely that the changing force/ inertia characteristics of the robot would have had an effect on the force exerted by the human on the system. For instance, moving the end effector through a region of space with higher endpoint inertia may cause the human to push harder in that space than otherwise necessary, even accounting for differences in gravity, friction, and endpoint inertia, adding noise to the data. For that reason, the forces of gravity, friction, and inertia were actively cancelled out via force control of the robot's three joints.

Gravity was cancelled out via a simple kinematic model. The lever arm between the effective center of mass of links 1 and 2 and their respective axis of rotation were calculated from the CAD model to start and then tuned slightly to account for differences in mass from 3d print infill, motor material and cable routing. These results were verified experimentally by moving the arm to various configurations and making sure each motor was outputting just enough torque to hold the end effector in place.

It is worth noting that the friction of the arm is low enough to begin with that if suspended upside down, each joint of the arm will swing back and forth with minimal change in magnitude between successive oscillations. Because the robot does not have any force sensors, there is no way to cancel out static friction as it can not measure any force being exerted on it until it begins movement. Despite this, because of the high fidelity of the encoders used on each joint (540CPR for J0 and J1, 49,152CPR for J2) the regime of kinetic friction can be cancelled out very shortly after movement begins.

For each joint, the damping coefficient β_i was determined by fixing the robot to the table such that the axis of rotation of the joint being examined was parallel with the direction of gravity. A simple control script was written to send torque commands to the motor in proportion to -β_i*Angular Velocity. β_i was slowly increased until pushing the joint and letting go resulted in the the joint accelerating away from its initial position, meaning that β_i had become large enough to overcome kinetic friction. β_i was then decreased to 80% of its critical value for safety reasons.

Robot inertia is by far the most difficult force to cancel out and first requires solving the equations of motion of the robotic arm. This process is non-trivial for a 3DOF manipulator in 3D space and requires the use of a powerful symbolic solver to obtain a closed form solution.

The first attempt at obtaining inertia cancellation torques involved using MatLab SimScape MultiBody (SM). SM provides an intuitive workflow of establishing reference frames and kinematic chains and supports directly importing solid bodies from Autodesk Inventor. Unfortunately, SM does not give the user access to the actual equations of motion so two parameter sweep scripts were written in order to generate enormous lookup tables for state prediction and virtual endpoint inertia. Force exerted on the end effector of the robot by the human can be calculated by determining the difference between the predicted states as a function of states in the previous timestep and then using the virtual endpoint inertia of the previous state to determine the direction and magnitude of force applied to the robot to achieve the difference in current state.

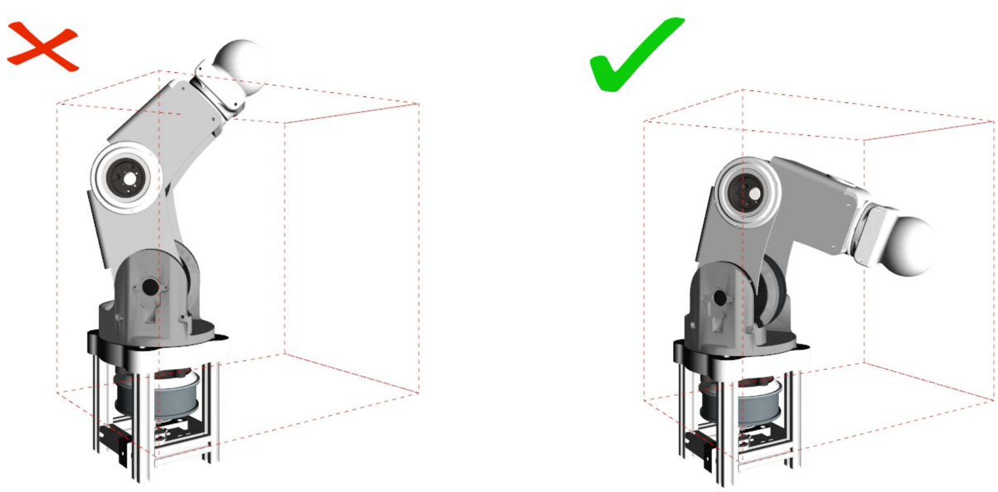
One limitation of calculating state prediction and endpoint inertia from SimScape Multibody is that the parameter sweep returns discrete values while the real world robot is acting in a continuous workspace. Because of this, these tables had to be linearized for each value which resulted in slight error for each estimate. This process is not only very computationally expensive, but is inflexible to changes in system parameters. For instance, after physical testing began it became apparent that the end effector needed to be swapped for a ball for easier grasping, however, this meant that both lookup tables needed to be regenerated. Even for a relatively “small” lookup table with 8 distinct values for each of the 6 states (J0,J1,J2 position and velocity), 262,144 simulations must be run which takes over 7 hours to complete.

Due to limitations in accuracy and flexibility, it became apparent that obtaining an analytical solution to the equations of motion was a superior strategy. Using PyDy, a python workflow and package array for use in complex multibody dynamics, equations of motion were obtained using a formulation of Lagrangian Dynamics referred to as *Kane’s Method.* Lagrangian Dynamics involves establishing generalized coordinates, speeds and kinematic differential equations for a system and solving the system in terms of energy so that constraint forces do not need to be considered explicitly [5].

Link mass obtained from the gravity cancellation script was used to correct component masses in the CAD model of the robot. The updated inertial dyadics were calculated using Autodesk Inventor iProperties and were used along with link lengths and masses to fully define each link of the robot. Because inertial cancellation was to be stacked on top of the gravity and friction cancellation torques, the only loads for this simulation on the system were the torques applied at the three joints.

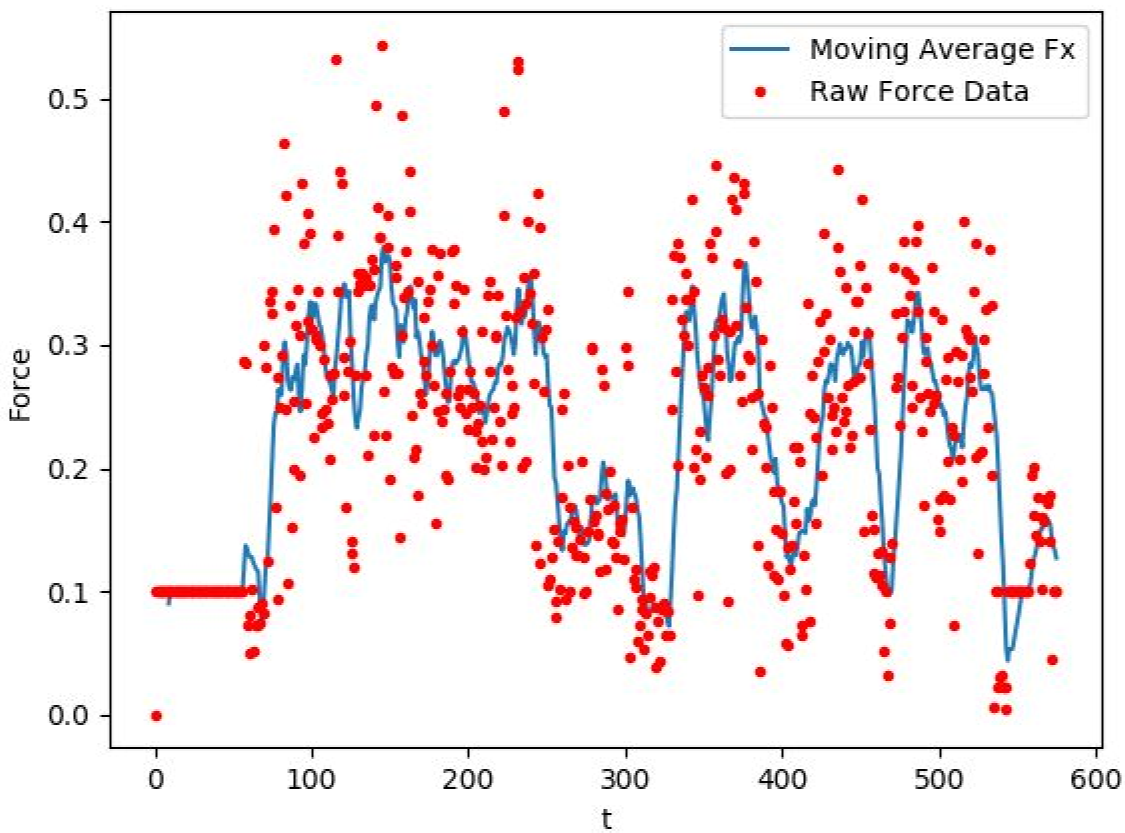
A function for state prediction simply involves plugging in current joint states into the equations of motion as initial parameters. The function used to measure endpoint inertia involves applying a differential force to the end effector in the x, y and z world frame axis for 1 second and recording the distance traveled by the end effector in the respective direction. Using $x = vi * t + 0.5 * a * t^2$ to calculate a and assuming no rotation for small displacement, I could be calculated for each axis via $F = Ia$. Once obtained, the statePredictor() and inertiaEstimator() functions were serialized using CloudPickle so that they would not have to be resolved upon each run of the program.

One limitation of the inertia estimator function is the fact that it quickly falls apart when attempted near the robot's vertical singularity. When the arm is fully extended pointing straight up in the air, a differential force pointing in the positive y direction (up) or in the direction perpendicular to the axis of rotation of joints 1 and 2, will result in no movement of the end effector due to the joints not bending due to kinematic constraints rather than inertia. This will be interpreted by program as infinite inertia in those directions, meaning that as the arm moves slightly away from singularity, the inertia cancellation loop thinks that it must send a command of near infinite torque to continue movement in that direction, which instantly saturates the actuators and creates a major safety hazard for the human holding on to the end effector. For that reason, a bounding box was created for a safe region of the workspace. Gravity and friction are cancelled in all regions but inertia is only negated within the bounded box. The robots pose relative to the bounding box was displayed in real time while test data was recorded to ensure that the end effector remained within bounds for the duration of each test.



Experimental Results:

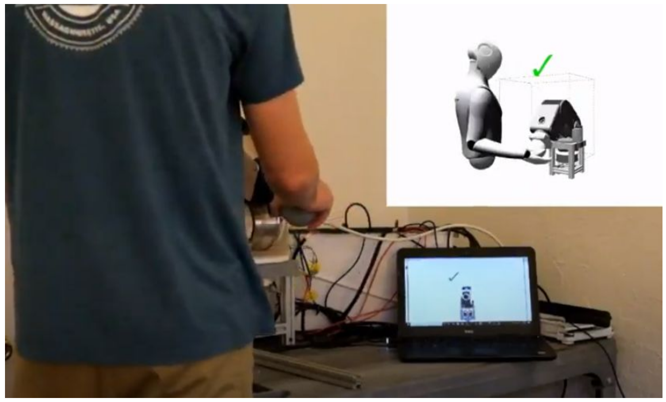
Position data taken directly from manipulator trajectory is relatively noisy and must first be filtered before it can be used in a meaningful way. Below is a plot of the raw data from a trajectory used to calculate the magnitude of the x component of force exerted on the end effector of the robot. A moving average filter is applied to smooth out some of the noise. This process is repeated for all three cartesian components of force.

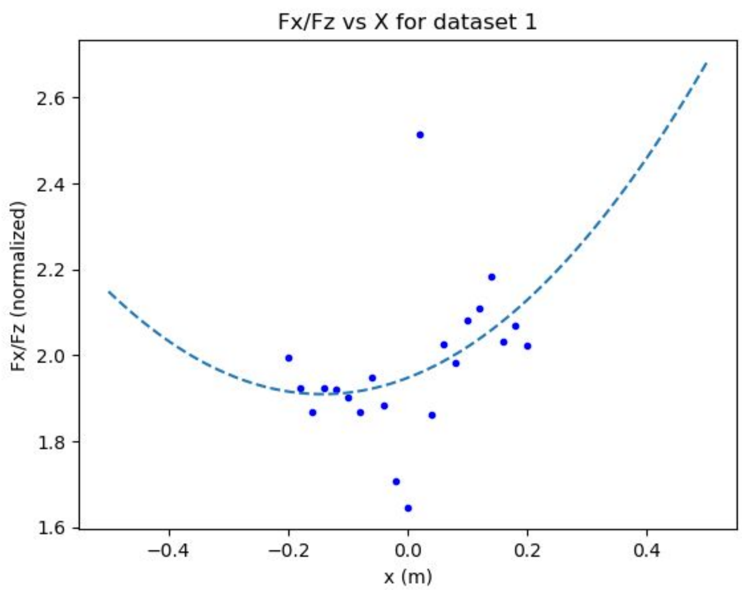


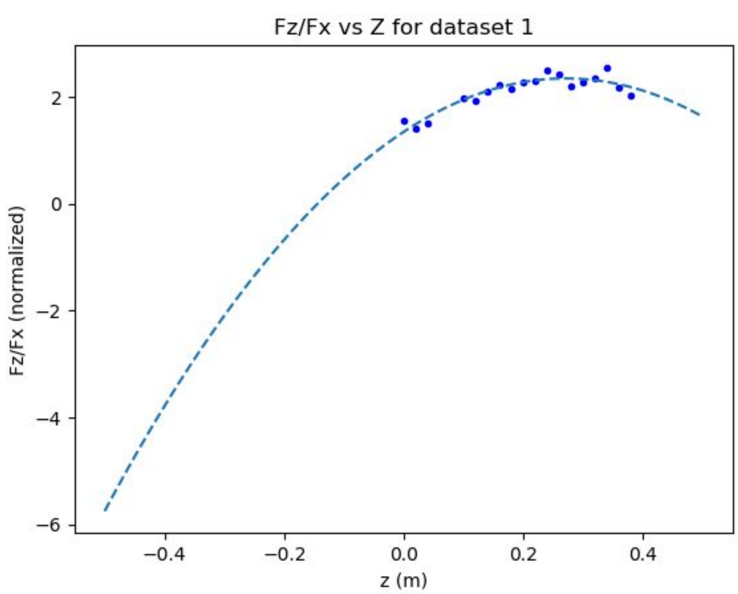
After the moving average filter is applied, for each axis the respective location of where each force is applied is digitized into 51 distinct bins and the average force of the top 50% of values in each bin is calculated. The decision to include only the top half of force measurements at each location is due the fact that a human can exert a low force on the end effector at nearly every location in the workspace but can only exert high forces in certain locations. Skewing the data towards higher force measurements makes this effect more apparent.

To determine the effect of the human force ellipsoid bias, for each axis binned and averaged forces in that direction were divided by forces in the orthogonal axis applied at the same location (ex: x vs Fx/Fz) . A quadratic least squares fit was applied to each plot to determine the critical point and thus the most likely coordinate of the human shoulder. Shown below are the results of two datasets.

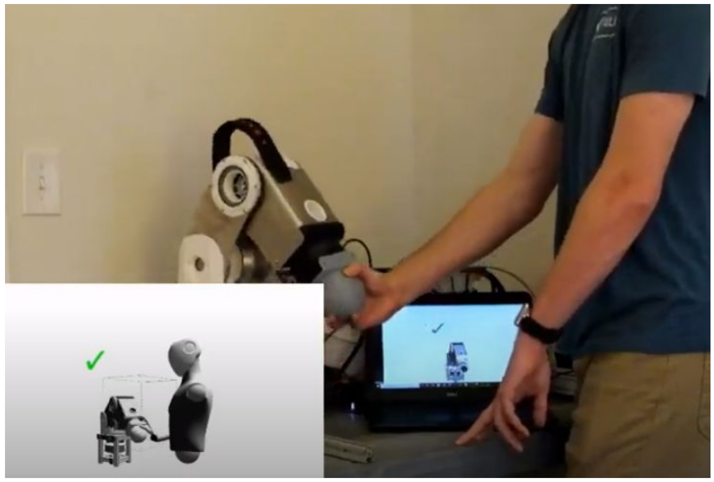
Dataset 1: Standing with shoulder to the left of robot

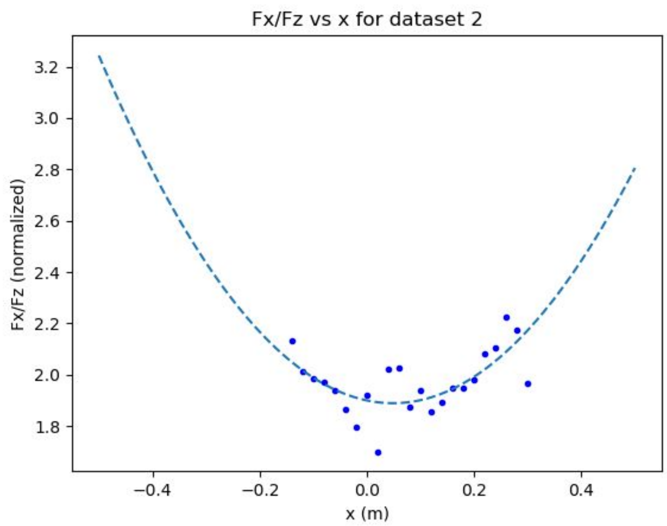


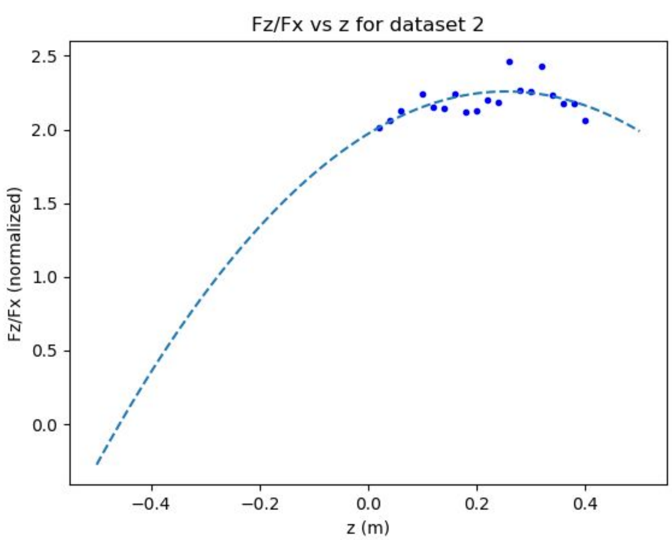




Dataset 2: Standing with shoulder to the right of robot







There is a clear shift in the x coordinate of the critical point of the best fit curve from one test to the next. This is due to the fact that the human force ellipse is becoming aligned with the xz coordinate axis (and therefore minimizing the ratio of Fx/Fz) at different x locations due to movement of the human shoulder. Alternatively, the critical point of the best fit curve plotted against the z axis shows little movement between the two tests. This makes sense due to the fact that the human shoulder is in about the same z location for both data sets. Ultimately, due to the fact that the human user is standing at a fixed height, the y position was held constant at the true value rather than being estimated through the same technique as x and z.

During testing, every 100 timesteps (around 7 seconds or so) this algorithm was run with the cumulative path data. A particle filter was constructed with the inverse of each particle's distance from the estimate from the force model as the first fitness metric. A basic kinematic fitness metric was also applied which multiplied every particle by a punishment factor for each point in the robot's end effector trajectory that is more than one arm length away. This punishment factor becomes less and less severe the longer the path data set becomes. The inclusion of kinematic data in the particle filter helps out early on when the SNR of the data may be low and it is difficult for the force based algorithm to make sense of human position. The bounding box surrounding the usable workspace helps keep the kinematic portion of the particle filter from becoming too powerful. Without this constraint on position, if the end effector were to travel too far to one side or the other, it would become very apparent where the human is not standing and would ruin some of the novelty of the demonstration.

Conclusion and Future Work:

Due to the nature of this project being relatively unique in its formulation there is plenty of room for improvement in future work. Because the robot involved in this project was not designed with the intention of being used as a haptic measurement instrument, certain aspects of the design are inconsistent with the hardware goals of this experiment. When the arm was originally designed in its 6DOF configuration, the two base joints needed to be as strong as possible to lift the mass of joints 3, 4 and 5. When used in a 3DOF configuration however, the same servo design used on the elbow (joint 2) could easily be reproduced for the lower two joints as well. This would mean lower rotating mass and less inertia of a heavy motor bell that would need to be counteracted in the dynamic force model, creating an overall lighter and more agile package. Due to the fact that no force sensors were used in the project, friction and dynamic forces could only be cancelled once the arm was in motion. The high mass of the rotor bell on joints 0 and 1 meant that a large amount of inertia must be overcome in order to begin motion. The use of an existing professional manipulator could produce more responsive torque control and trajectory estimation.

The two example data sets each last between 60 and 120 seconds. With better code optimization and more powerful hardware sampling at a higher rate, it would be interesting to see how quickly this system could estimate human pose. Additionally, while the scope of this project focused solely on the cartesian forces exerted on the end effector. It would be interesting to see if additional information could be gained about the human, such as what angle their elbow is raised up at by examining the torques exerted on the end effector.This experiment involved only determining the location of a static human.

Future work could expand upon this strategy to estimate the pose of a moving human. Furthermore, if a better method of endpoint inertia estimation can be developed for the robot that does not fall apart near the vertical singularity, it would be possible to extend this strategy of human pose estimation to the full workspace of the robot.

Lastly, the model devised in this paper works under the assumption that the human arm has zero inertia or more specifically, the human user will subconsciously exert extra force when necessary to overcome the inertia associated with their own swinging arm. While this is a reasonable assumption at low angular velocities, human arm inertia must be taken into account in order to apply this technique to more quickly moving trajectories.

References:

[1] Flash, T., Mussa-Ivaldi, F. Human arm stiffness characteristics during the maintenance of posture. *Exp Brain Res* 82, 315–326 (1990). https://doi.org/10.1007/BF00231251

[2] El-Gohary, Mahmoud, and James McNames. "Human Joint Angle Estimation with Inertial Sensors and Validation with A Robot Arm - IEEE Journals & Magazine." *IEEE*, IEEE, 12 Feb. 2015, ieeexplore.ieee.org/abstract/document/7041198

[3] D. Roetenberg et al., "Ambulatory position and orientation tracking fusing magnetic and inertial sensing," IEEE Trans. Biomed. Eng., vol. 54, no. 5, pp. 883–890, May 2007.

[4] Lin, Feng, and Robert D Brandt. "An Optimal Control Approach to Robust Control of Robot Manipulators." *Https://Ieeexplore-Ieee-Org.ezproxy.library.tufts.edu/Stamp/Stamp.jsp?Tp=&Arnumber=660845*, IEEE, 1 Feb. 1998.

[5] Gamalath, Kail. *An Introduction to Analytical Mechanics.* Alpha Science International, 2011.