Leveraging Control of a Robotic Hand to play Für Elise

Evan Strittmatter, December 19th 2023

Abstract - Here we demonstrate an implementation of an optimal control algorithm to play the opening of Für Elise on a custom dimension piano with the Wonik Allegro hand in simulation space. The problem is first broken down into its separate component. We derive the dynamics of the individual fingers of the robot, and define a objective function which applies universally across each separate component. We demonstrate convergence for each task, and then provide time captures as well as a video of the full performance.

I. Introduction

As the demand for versatile and human-like robotic systems continues to grow across industries including healthcare, manufacturing, waste disposal and space exploration, the need for sophisticated control of adaptable robotic hands becomes increasingly apparent [1][2][3][4]. The ability to mimic human hand movements with precision and agility is pivotal for these robotic systems to seamlessly integrate into diverse environments and perform a wide array of tasks. Moreover, the pursuit of achieving human-like dexterity raises intriguing questions at the intersection of bio-mechanics, control algorithms, and sensory feedback. Producing music offers an unique lens through which to demonstrate control of exact motion, combined with pinpoint timing, which is integral to the tasks described above [5]. Here we demonstrate the successful implementation, in simulation space, of a robotic hand, shown in Fig 1. performing Für Elise.

In the following sections we will provide a brief background on control of the allegro hand and the state of Musical Robot applications. We then dive into the specific dynamics of the Allegro Hand, and how we have implemented control over the system. We present a piecewise approach to the problem, breaking it down into a series of achievable components. Finally we present the results of our findings, analyse the models abilities and shortcomings, and provide recommendations for future steps.

II. Background

This project focuses around control of the allegro hand within a music application as this unique task introduces additional complexity to standard optimal control problems. For this implementation, rather than a control solution be satisfied simply by reaching a target position; the time at which the robot reaches this position is critical for a successful completion of the task. If the robot hits a key too early or too late the melody of the song is disrupted.

Fig. 1. The Allegro hand is a 16 DOF robotic hand consisting of four fingers, each with 4 joints. For this performance we are only utilising the index, middle, and ring fingers on the robot.

We have previously conducted a simulation control project utilizing a combined system of the Franka-Emika Arm, and the Allegro hand, without the added constraints of time complexity. As part of this project we implemented a control algorithm applied to a real world system which takes human inputs from a remote controller and moves the wrist joint of the arm in the direction provided by the user.

With the emerging field of musical Robotics [6], projects involving time incorporated control have been presented. Details for achieving time scaling on control built upon those presented in reference [7].

III. Dynamics of the Allegro Hand

Here we define the dynamics of the Allegro Hand.

A. Configuration, State, and Control

Here we define the state of the system. Let q = [q_{index}, q_{middle}, q_{ring}, q_{thumb}]

$q_{index} \in R^4$, and is defined with the joint angles θ_1,θ_2,θ_3,θ_4 where θ_2 - θ_4 represent respectively the angle of the MCP, PIP, and DIP joints in the human finger, and θ_1 is the rotation in the base joint perpendicular to this motion. The same Joint angle breakdown applies for q_{middle}, and q_{ring}. q_{thumb} on the other hand mirrors the joints of the human thumb, providing all the same flexion and extension now with only two joints on the finger instead of 3, but with 2 joints at the base of the finger as opposed to 1.

The control inputs motor torques onto the individual servo motors at each of these joint positions. u = [u_{index}, u_{middle}, u_{ring}, u_{thumb}] where u_{index}, u_{middle}, u_{ring}, and u_{thumb} $\in$

R^4, where each value is a torque applied to the motor in the same schema as defined above for $\theta' s$

We define the state of the robot as x = [q , $\dot{q}$]

The dimensions, mass of each components, and other mechanical specifications of the allegro hand are available at [8], and the XML file containing this information was supplied by [9]].

B. Manipulator Equation, and Dynamics

From Euler-Lagrange equations we can derive the manipulator equations for the fingers under gravity as

$$M(q)\ddot{q} + b(q, \dot{q}) = u \tag{1}$$

Where $M(q)$ is the mass matrix of an individual finger, $b(q, \dot{q})$ is the bias term. Reorganizing the manipulator equation above and the definition of state from above we can get the first order Dynamics of a single finger

$$\dot{x} = f(x, u) = \begin{bmatrix} \dot{q} \\ M(q)^{-1}(u - b(q, \dot{q})) \end{bmatrix} \tag{2}$$

IV. Control Method - Operational Space control

For the problem of control the goal is to close to the desired target position and hit the key with enough force to play a note, but also to strike the note at in accordance with the melody of the song. There are a multitude of potential control methods which can be used to solve this task. The data presented here is gathered with a PID control implementation. We also provide a comparison for am implementation with operational space control.

A. Robot Kinematics

The process of calculating the end effector position given the current joint angles is called Forward Kinematics (FK). The inverse of this, i.e. calculating the joint angles that result in an end effector position is called Inverse Kinematics (IK). Together FK and IK make up robot kinematics. Logically there is a multitude of possible joint angles, depending on the degrees of freedom of the robot, when using IK for a given end effector. For our case we are focusing on the four DOF fingers, from which the FK are derived as follows

$$D = L_1 * sin(\theta_1) + L_2 * sin(\theta_1 + \theta_2) + L_3 * sin(\theta_1 + \theta_2 + \theta_3) \tag{3}$$

Where L_1 - L_3 are the lengths of the arm segments, and D is the distance from the base of the finger, the end end effector position relative to the fingers base is calculated as follows

$$X = D * cos(\theta_0) \tag{4}$$

$$Y = L_1 * cos(\theta_1) + L_2 * cos(\theta_1 + \theta_2) + L_3 * cos(\theta_1 + \theta_2 + \theta_3) \tag{5}$$

$$Z = D * sin(\theta_0) \tag{6}$$

In order to implement a PID controller in joint space, we must use these equations to find θ values which will satisfy to the conditions of our problem. More specifically this means finding the joint angles for each finger, for each note to be played given the desired end effector pose. Operational Space control provides an alternative method for correlating the Cartesian and joint space coordinates.

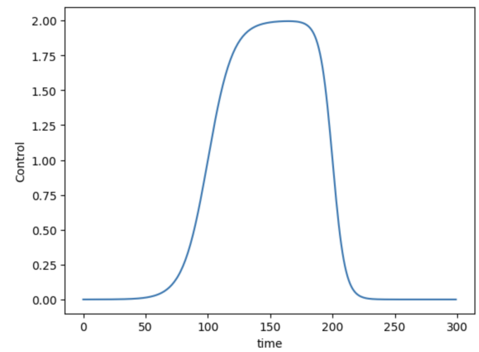

Fig. 2. Example of piecewise function to represent motor control inputs at the specific time within the melody for a single note to be played

B. PID Control

As mentioned above for the task of playing piano we simply need to hit the notes (be at the correct position with the correct velocity) at the right time (doesn't reach the desired position too soon or too late). We will worry about timing later on but from a note playing stand point the error is calculated as follows:

$$P_{error} = K_p * E(t) \tag{7}$$

$$I_{error} = \int E(t)\, dt \tag{8}$$

$$D_{error} = K_d * dE(t)/dt \tag{9}$$

$$TotalError = P_{error} + I_{error} + D_{error} \tag{10}$$

where

$$E(t) = qpos_{desired} - qpos \tag{11}$$

In order to incorporate time into calculating total error we want the control to be in the shape of the piecewise function in Fig 2.. Control inputs are therefor defined where E_t is the difference between the current and desired time.

$$Error_{time} = tanh(K_{t1} * (E_t + CompletionTime)) \\ -tanh(K_{t2} * E_t) \tag{12}$$

$$\ddot{q}_d = Error_{PID} * Error_{time} \tag{13}$$

where K_p, K_p, and K_p are selected by finding the minimum possible total error at the time the note is supposed to be played.

C. Operational Space Control

Operational Space Control utilises the same premise as a PID controller, however now we are working with Cartesian coordinates instead of in joint space. This makes it much easier to define our tasks in terms of end effector sites in Cartesian coordinates rather than using Inverse Kinematics to find the respective joint space coordinates. As such the $\ddot{x}_d$ is calculated by the same method as in $eq13$, however now $qpos$ is [x,y,z] rather than [θ_1,θ_2,θ_3,θ_4]

The Jacobian correlates the relationship between Cartesian and joint space and is classified as follows:

$$J(q) = \nabla Error_{PID} \tag{14}$$

which can be used to calculate the Operational space mass matrix, and control as follows

$$M(x) = (J(q) \cdot M(q)^{-1} \cdot J(q).T)^{-1} \tag{15}$$

$$u = J.T \cdot M(x) \cdot \ddot{x}_d + bias(q, \dot{q}) \tag{16}$$

This gives the robot the additional ability to incorporate striking the key with a specific velocity. This should be set to impart the required momentum onto the key such that the note will ring forcefully and correctly. This is much more difficult to formulate for an implementation which is exclusively in joint space.

V. Problem Formulation - Für Elise

In this section we define the objective function and the constraints chosen to complete the introduction for one of the most recognizable songs on piano.

A. Individual Finger Objectives

In order to solve this problem we broke the task of playing a song down into its individual components which is a series of notes in order, played at specific time intervals. This means we need an objective function that not only states position and velocity for each of the robotic fingers, but additionally incorporates time in to calculating its error[7].

We simplified the problem, mapping the hand-note pattern of a human counterpart to be reflected in the trajectory of the robot. This means we preemptively matched which fingers needed to play which specific notes on the piano, and defined the objective function for a universal finger as follows:

$$Error_{state} = K_p*(qpos_{desired}-qpos)+K_d*(qvel_{desired}-qvel) \tag{17}$$

$$Error_{time} = 1 + tanh(K_t * (time_current - time_desired)) \tag{18}$$

$$u = Error_{state} * Error_{time} \tag{19}$$

The values for $K_p, K_d, and K_t$ in the PD control are tuned to minimize the worst error of any particular note.

B. Combined Motion

After the specific time at which the note is supposed to have been played is reached, the desired value inputs to the objective function are modified to the value of the next note in the series. The series of notes played was tailor picked so that the same finger would never play two notes on different keys back to back. This allows us to synchronously combine motion of two different fingers, so while one finger is moving downwards to play a note, the previous finger is resetting in preparation to play the next note. This is an important fundamental skill of playing music in accordance with the rhythm of the song.

C. Constraints

Equality constraints: In order to prevent unrealistic motion, the model is constrained such that the different components cannot overlap one another, or the objects in the surrounding world (ie the piano.)

Inequality constraints: On top of this motion of each individual joint is constrained between -15 and 100 degrees, and the joint torques are constrained below 0.7 Nm.

VI. Results

In this section we will validate the provided control trajectory, confirming the trajectory has converged to an optimal solution, and that the robot successfully performs the song Für Elise.

A. Solver Convergence

We first look to see the solver has converged to a locally optimal solution to each of its separate trajectories for each individual note. To do this we collect the total error in the current end effector position at each iteration step. Fig 3. shows the plot of this error vs iteration count and we can see that the solver approximately reaches its desired end effector goal by the desired time. Note that the D# which is played by the middle finger doesn't quite reach its desired end effector pose by the 600th iteration. This is because the parameters are constants across control for all notes and are tuned to minimize total error, summed across every moment a note should be played. The problem could be broken down further to provide each finger and note its own individual objective function and controller, and have each of these pieced together for a more optimal solution.

B. Trajectory Demonstration

Although some notes do not quite reach there exact target position by the time the note should be played, for the purposes of playing this particular song they appear close enough to constitute successful completion of the piece. We have attached a video with this submission, and in addition the moments the fingers make contact with the piano are presented in Fig 4.

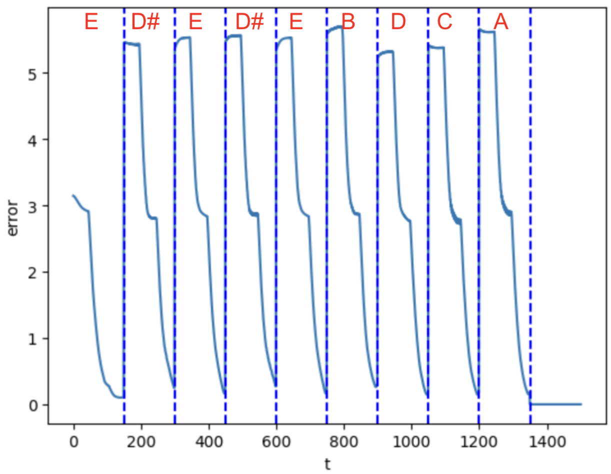

Fig. 3. Objective function loss against iteration count. The vertical dashed blue lines represent the desired point in time for the note to be played in line with a professional recording[10] of the song. The note played at each position is also labeled to the left of its respective dashed line.

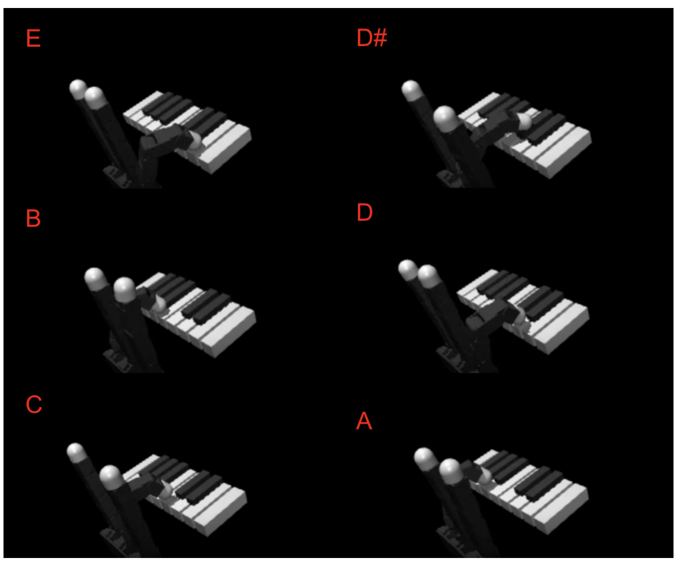

Fig. 4. Still frame captures of the moments the fingers contact the specific notes they are supposed to play. These moments correspond with their respective dashed blue line in Fig 3.

VII. Conclusion

We have demonstrated successful implementation of a robotic hand performing the opening and most recognisable part of Für Elise. Although we can see the current algorithm is not perfectly optimal for every subdivision of the the total task, it is enough to play with precision a complex song. This demonstrates the successful implementation of a robotic control algorithm that incorporates both accurate motion and precises timing.

Given the current state of the project, we provide two paths forward. From a simulation standpoint this solution can be improved by providing free motion to the wrist joint instead of having it mounted in place. This will allow for more notes across a wider range of keys to be played, which is critical for a complete performance of a song. In addition this will more accurately represent a human playing, as the hand can now be angled parallel with the keys rather than perpendicularly as shown here. To model this accurately the wrist motion must be constrained by the physical limitations of the system it is modeling. In particular its velocity and position should be constrained. Another potential direction this project could be taken is implementation on a physical system. This would allow for a live demonstration, and if combined with the Franka Emika arm, as has already been accomplished, and the simulation recommendations above, could lead to an exciting live performance of any song on piano. We recommend pursuing both tasks, as the proposed combined systems provides an exciting example of the potential and ability of musical robotic systems.

VIII. References

[1] Camarillo, David B., Thomas M. Krummel, and J. Kenneth Salisbury Jr. ”Robotic technology in surgery: past, present, and future.” The American Journal of Surgery 188.4 (2004): 2-15.

[2] Evjemo, Linn D., et al. ”Trends in smart manufacturing: Role of humans and industrial robots in smart factories.” Current Robotics Reports 1 (2020): 35-41.

[3] Satav, Anushka G., et al. ”A state-of-the-art review on robotics in waste sorting: scope and challenges.” International Journal on Interactive Design and Manufacturing (IJIDeM) (2023): 1-18.

[4] Gao, Yang, and Steve Chien. ”Review on space robotics: Toward top-level science through space exploration.” Science Robotics 2.7 (2017): eaan5074.

[5] Weinberg, Gil, et al. ”Robotic musicianship.” RO-MAN. 2008.

[6]Long, Jason, et al. ”A comparative evaluation of percussion mechanisms for musical robotics applications.” 2015 6th International Conference on Automation, Robotics and Applications (ICARA). IEEE, 2015.

[7]Lynch, Kevin M., and Frank C. Park. Modern Robotics: Mechanics, Planning, and Control. University Press, 2021.

[8]WONIK ROBOTICS. Allegro Hand User Manuel. 26 Dec. 2022

[9] Zakka, Kevin, and Yuval Tassa. “MuJoCo Menagerie: A Collection of High-Quality Simulation Models for MuJoCo.” 7 Sept. 2022.

[10] ”Beethoven - Für Elise” YouTube, uploaded by Rousseau, 21 May 2021,www.youtube.com/watch?v=wfF0zHeU3Zs.