

263C Project Report: Typing with a Simple Robot

Zhi Li Preston Hill Xuan Yang

Abstract—This project aims to make a typing robot using feedback control. The report includes the motivations for building a typing robot, the mechanical and electrical design of the robot system and application of PD control with gravity compensation. The report also shows the simulation and physical experiment results of the robot system and analysis of the task performance.

I. OVERVIEW

The goal of this project is to build a robotic arm that can type on a keyboard automatically by using robotics control systems. The structure of the robot is a simple robotic arm with 3 DOFs (Degree of Freedoms) to move around over a generic keyboard(As is shown in Fig 1). There are four aims to design and build this robotic arm:

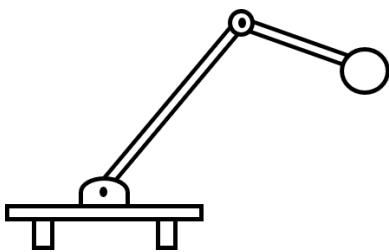


Fig. 1. Simple structure of the 3-DOF typing robot

(1) To master the content learned in 263C Control of Robotic Systems class and attempt many control methods, so that we may better understand them. Then we may use those different control methods on this simply built prototype to compare each and find out which method works best.

(2) To build this robot is very meaningful. In our daily life, there is always a lot of stuff that needs typing, which consumes much of our free time. This robot can help people to do this kind of simple typing work over and over again, so they can leave the robot to autonomously type while performing other tasks, which helps them save much time to do what they want.

(3) This robot can also work as an assisting robot to help people who have difficulty typing or using their hands, so that those disabled people will be also able to finish their work by using this robot.

(4) This robot could serve as a platform for further study. This robot could be a prototype for derivative designs that could accomplish other tasks. By changing the end effector of the robot, it could become a such as Whack-A-Mole game player, piano player, or artist for example. If wheels were added to the rail, it could also become a ball-picker to help sports players such as tennis, badminton and ping pong players pick up balls during training.

II. OBJECTIVES OF CONTROLLER DESIGN

In our project, feedback control is realized to make robot arm type on a keyboard. The basic goal for the typing task is to rapidly and precisely type the desired keys. The feedback control will focus on the response to the step input which is the position of desired key.

Controller performance is evaluated by the following heuristics:

(1) The time to strike a desired key.

The controller is designed so that rapid transient performances will be achieved, i.e. the response time of the robot arm when given a desired configuration should be minimized. Additionally, other transient performance indices are also examined, such as the overshoot, and settling time.

(2) The accuracy of pressing the key.

For the typing task in the project, based on the assumption that the controller could drive the robot arm to correctly press the desired key, we will try to minimize the steady state error (between the desired and actual configuration of the end effector) for each key stroke.

III. METHODS

A. Assumptions

Assumptions in Simulation

- Each link is considered a rectangular prism, and each motor is considered a cylinder for simulation
- Viscous friction is negligible
- Non-linearities such as backlash, and PWM deadzones are negligible
- Measurements are accurate

B. Solidworks Model

We used Solidworks to design the 3D model of our robot, shown in Fig 2. Our robot is primarily constructed from three parts: a rail, link one, and link two, which constitute the three degree of freedoms.

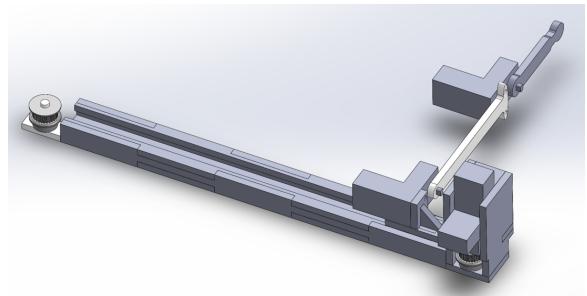


Fig. 2. 3D model of the typing robot

As Fig 2 shows, the motor on one side of the rail drives a pulley to move the sled back and forth using a timing belt.

Two additional motors are used to change the joint angles of robotic arm, connected to the sled. We laser cutting pieces from MDF and glued them together with epoxy to reduce cost and complexity, and the resultant robot is shown in Fig 3.

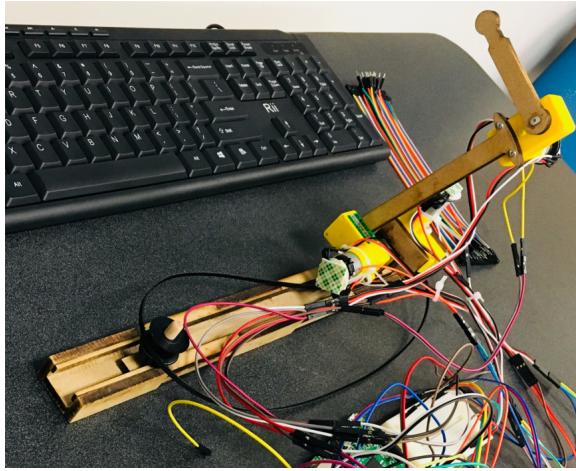


Fig. 3. Real shape of the typing robot

Based on a geometric relation with the keyboard shown in Fig 4, we determined the length of link one to be 115 mm, and link two to be 64 mm. according to the size of the keyboard and the angle between those two links. This geometric relation set the minimum and maximum desired joint angles for each joint. The desired joint limits for joint two should be 30° and 120° for joint two when the end effector reaches the two boundaries of the keyboard as shown in Fig 4. However, after construction the joint limits were found to be obstructed by the thickness of the links, so in order maintain the desired workspace over the keyboard, the initial distance between the rail and keyboard to was increased to 55 mm. This caused the joint limits to be larger, ensuring the robot can reach the desired workspace.

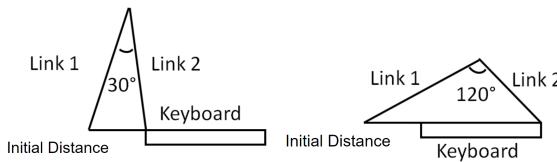


Fig. 4. Geometric relationship between robot and keyboard to determine the length of two links

C. Forward and Inverse Kinematics

Forward kinematics is used to determine the coordinate and position of the end-effector. We assigned the frames of the robot as shown in Fig 5. The base frame is on the rail with the same height as the keyboard. Let frame $\{1\}$ coincide with frame $\{0\}$ with the X axis pointing vertically up. Let frame $\{2\}$ be at the first joint with the X axis along the link one. Finally, frame $\{3\}$ is at the second joint with the X axis pointing along link two. Frame $\{4\}$ is at the end-effector. From this, the DH parameters were tabulated as shown in Table I,

and the homogeneous transformation matrix from base frame to end effector frame is

$${}^0T = \begin{bmatrix} c_{23} & -s_{23} & 0 & a_1 + L_2 c_{23} + L_1 c_2 \\ s_{23} & c_{23} & 0 & L_2 s_{23} + L_1 s_2 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Where $c_{23} = \cos(\theta_2 + \theta_3)$ and $s_{23} = \sin(\theta_2 + \theta_3)$.

Since the 3 DOF robot is being used in free space (with six degrees of freedom), only the position information is needed for inverse kinematics and control. Let $p_d = [p_x \ p_y \ p_z]^T$ be the desired position, then the inverse kinematics can be obtained as follows:

$${}^0T(4, 1 : 3) = \begin{bmatrix} a_1 + L_2 c_{23} + L_1 c_2 \\ L_2 s_{23} + L_1 s_2 \\ d_1 \end{bmatrix} = p_d = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \quad (2)$$

$$d_1 = p_z \quad (3)$$

$$c_3 = \frac{(p_x - a_1)^2 + p_y^2 - L_1^2 - L_2^2}{2L_1L_2} \quad (4)$$

$$s_3 = -\sqrt{1 - c_3^2} \quad (5)$$

$$\theta_3 = \text{Atan2}(s_3, c_3) \quad (6)$$

$$k = \frac{(p_x - a_1)^2 + L_1^2 + p_y^2 - L_2^2}{2L_1} \quad (7)$$

$$\theta_2 = \text{Atan2}(\sqrt{(p_x - a_1)^2 + p_y^2 - k^2}, k) + \text{Atan2}(p_y, p_x - a_1) \quad (8)$$

Where the small joint limits play a large role in the existence of solutions for the physical robot.

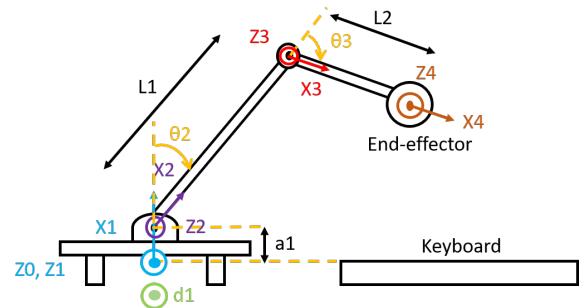


Fig. 5. Frame assignment on the typing robot

TABLE I
DH TABLE

| i | α_{i-1} | a_{i-1} | d_i | θ_i |
|-----|----------------|-----------|-------|------------|
| 1 | 0 | 0 | d_1 | 0 |
| 2 | 0 | a_1 | 0 | θ_2 |
| 3 | 0 | L_1 | 0 | θ_3 |
| 4 | 0 | L_2 | 0 | 0 |

D. Electrical Component and Design

The electrical components used to build the system include: an L293D circuit, Raspberry Pi, Hall Effect Sensors, Power Supply and Geared DC Motor. The L293D is an H bridge motor control circuit which has a high current output; the Raspberry Pi is the central processor where the control loop runs. The hall effect sensors detect changes in magnetic fields so a pair of them can function as an encoder using a phase offset between the pair to infer direction of rotation. The power supply provides six volts at a current appropriate for the motors and the motors are the output of the system and have a high gear ratio which allowed adequate torque generation.

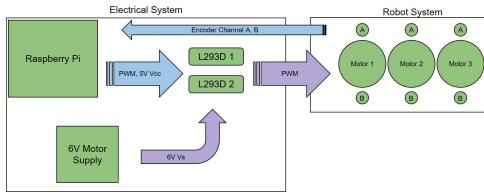


Fig. 6. Electrical Components and Design

Fig 6 shows the connections between the components. Blue arrows represent logic signals with low current from the Raspberry Pi, and purple arrows represent the high current signals required to drive motors. The Raspberry Pi outputs a logic PWM signal generated by software with small current and 5V to the two L293D circuits. Since the PWM signals are generated by software the maximum frequency of the signal is smaller than a PWM signal generated using electrical hardware, which the Raspberry Pi can also do. However, with three motors six independent signals were needed and the Raspberry Pi only provides two hardware based PWM signals. Each L293D amplified the logical PWM signals from the Raspberry Pi based on the motor supply voltage and current, and can power two motors. The PWM output of the L293D is suitable for direct application to the motors. Each L293D outputs two PWM signals with opposite magnitude and each signal is applied to each of the motor leads, causing the PWM signals to be bidirectional. Thus when the duty cycle is less than 50 %, the motor spins clockwise, and when the duty cycle is greater than 50 % the motor spins counterclockwise. Finally, the hall effect sensors output logic signals back to the Raspberry Pi allowing the computation of position and velocity. The control loop runs on the Raspberry Pi, where the joint position and velocity data from the hall effect sensors is used to adjust the output to the L293D circuit. As shown in Fig 7, the hall effect sensors are mounted in pairs to each motor. They are placed less than 3 mm away from a magnet with 4 north and 4 south poles. This magnet was attached to the rotor of the motor with a press fit. However due to the large motor housing causing an unstable connection to the magnet, the motor housing was filed down to allow the magnet to slide further onto the motor rotor.



Fig. 7. Motor Magnet and Hall Effect Sensor

E. Control Methods

Since the arm of the robot moves against gravity after each key stroke, the effect of gravity is non negligible. Thus PD Control with Gravity Compensation was used to control the robot. This control law may be expressed as

$$u = g(q) + K_P \tilde{q} - K_D \dot{q} \quad (9)$$

where $g(q)$ is gravity term of the robot system, $\tilde{q} = q_d - q$ is the difference between desired joint angle(q_d) and real time joint angle(q), and K_P is the proportional gain and K_D is the derivative gain, with desired velocity equal to zero [2].

Fig 8 shows the block diagram of PD control with gravity compensation.

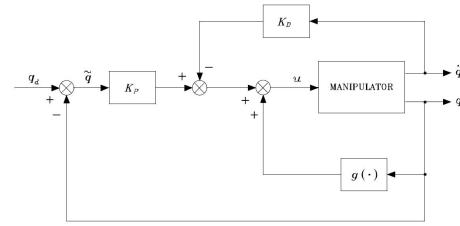


Fig. 8. Block Diagram [2]

The basic model of two links part of our robot is shown in Fig 5. Based on this model, the gravity term is calculated and can be expressed as:

$$g(q) = \left[\frac{(m_{l1}l_1 + m_mL_1 + m_{l2}L_1)c_2 + m_{l2}l_2c_{23}}{m_{l2}l_2c_{23}} \right] \quad (10)$$

Where c_2 is $\cos \theta_2$, c_{23} is $\cos(\theta_2 + \theta_3)$, m_{l1} and m_{l2} are the mass of each link, l_1 and l_2 are the distances of centers of mass from joint axes and L_1 and L_2 are the link lengths. From this control law, the control torque needed to drive joints two and three to the desired position is derived.

However, the control torque cannot be directly applied to the motors; we need to transfer the desired control torque to the duty cycle used to command motor rotation. Based on the motor circuit model (Fig 9), the relationship between control torque and desired voltage applied to the motors was derived.

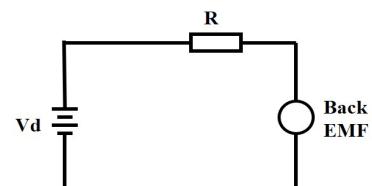


Fig. 9. Motor Circuit Model

$$V_d = \frac{Ru}{k} + k\omega \quad (11)$$

Where R is the resistance of motor, u is the control torque generated from control law, k is the torque constant of the motor and ω is angular velocity of the motor rotor. Then based on PWM theory, the duty cycle needed for the motors is:

$$D = \frac{V_d}{V} 100 \quad (12)$$

Where D is duty cycle of motor, and V is maximum working voltage of motor. For control of the first joint, we treat this motor control separately. We use position feedback to command the motor at the maximum duty cycle due to the high load.

F. Simulation

Considering the two link planar arm and the control law, a dynamic model was developed, with the inertia matrix given by

$$B(q) = \begin{bmatrix} b_{11}(\theta_3) & b_{12}(\theta_3) \\ b_{21}(\theta_3) & b_{22} \end{bmatrix} \quad (13)$$

Where

$$\begin{aligned} b_{11} = & I_{l1} + m_{l1}l_1^2 + k_{r1}^2 I_{m1} + I_{l2} \\ & + m_{l2}(L_1^2 + l_2^2 + 2L_1l_2 \cos \theta_3) \\ & + I_{m2} + m_{m2}L_1^2 \end{aligned} \quad (14)$$

$$b_{12} = b_{21} = I_{l2} + m_{l2}(l_2^2 + L_1l_2 \cos \theta_3) + k_{r2}I_{m2} \quad (15)$$

$$b_{22} = I_{l2} + m_{l2}l_2^2 + k_{r2}^2 I_{m2} \quad (16)$$

and I_{l1}, I_{l2} are the inertia of each link, $I_{m1} = I_{m2}$ are inertia of each motor and $k_{r1} = k_{r2} = 48$ is the gear ratio of motor. The coriolis and centrifugal Matrix is

$$C(q, \dot{q}) = \begin{bmatrix} h\dot{\theta}_3 & h(\dot{\theta}_2 + \dot{\theta}_3) \\ -h\dot{\theta}_3 & 0 \end{bmatrix} \quad (17)$$

Where

$$h = -m_{l2}L_1l_2 \sin \theta_3$$

Since the equations of motion obtained with Lagrange formulation give the analytic relationship between the joint torques and the joint positions, velocities and accelerations, these can be computed as

$$\ddot{q} = B^{-1}(q)(\tau - \tau') \quad (18)$$

Where

$$\tau'(q, \dot{q}) = C(q, \dot{q})\dot{q} + g(q) \quad (19)$$

The block diagram of the dynamics is shown in Fig 10

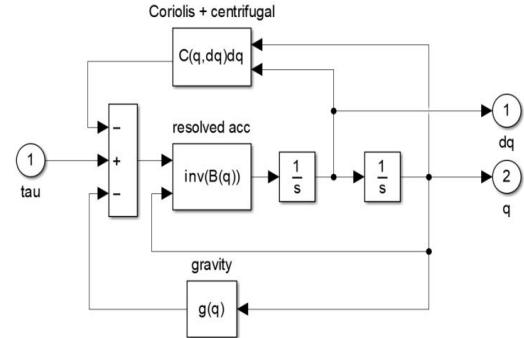


Fig. 10. Dynamics of robot [1]

The physical parameters of our robot are listed in Table II

TABLE II
PARAMETERS OF ROBOT

| Variables | Values | Explanations |
|-------------|-------------|--|
| m_{link1} | 0.005kg | mass of link1 |
| m_{link2} | 0.003kg | mass of link2 |
| L_1 | 115mm | link1 length |
| L_2 | 64mm | link2 length |
| l_1 | 70mm | distances of centers of mass from joint axes |
| l_2 | 40mm | the same as l_1 |
| m_{motor} | 0.06kg | mass of motor |
| k | 0.048 | motor inner parameter |
| R | 3.6Ω | motor resistance |
| V_{max} | 5V | motor maximum working voltage |
| K_p | 0.25 | stiffness |
| K_d | 0.125 | damping |
| T_{max} | 0.08N.m | Maximum torque of motor |

The simulation results will be shown in the results section.

IV. RESULTS

A. Simulation Results

The initial condition of the system is $q(0) = [0, -\frac{\pi}{2}]^T$, and the desired position is $q_d = [-\frac{\pi}{2}, -\frac{\pi}{3}]^T$. The response of the system to this step input is shown in Figure 11

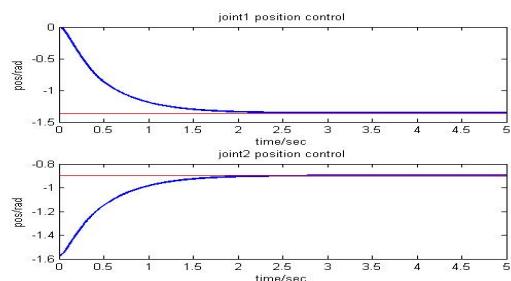


Fig. 11. Step response of planar arm

For this input, the rise time is less than one second, there is no overshoot and the settling time is approximately two seconds. These transient performance metrics are acceptable for our robotic system. The steady state error for position is shown in Fig 12

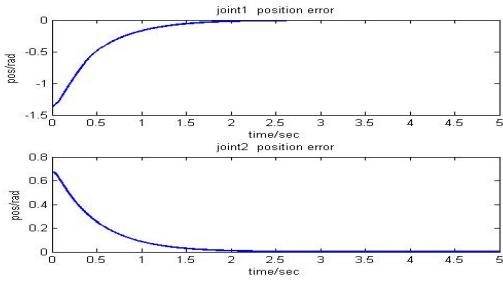


Fig. 12. Error of 2 joint angles

The steady state error for each link joint decays to zero after two seconds. Therefore, from the simulation, the controller is acceptable for a step input.

Additionally, the gravity term $g(q)$ is also examined in simulation to verify that the torque of the motor ($T_{max} = 0.08N.m$) is large enough to lift the arm. The result is shown in Fig 13.

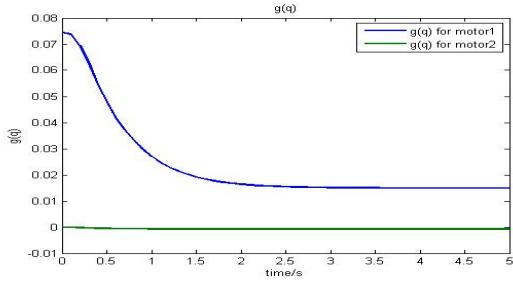


Fig. 13. Gravity term of simulation

From this figure, the gravity term is always less than $0.08N.m$ during the entire simulation.

The control torque generated from the control law must also be examined, since the physical robot will experience saturation at 100% or 0% duty cycle. The saturated control torque and the corresponding response of system is in Fig 14, where it can be seen that the control torque does not exceed the torque limits of the motor.

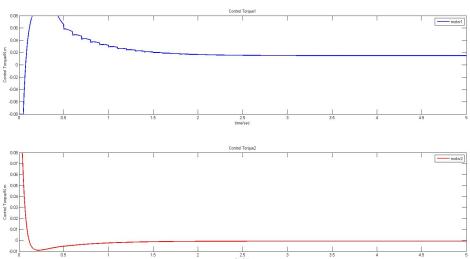


Fig. 14. Control Torques

From the simulation, we can conclude that the transient and steady state performance meet our design objectives.

B. Physical Experimental Results

Snapshots of the physical system during a key pressing task are shown in Fig 15.

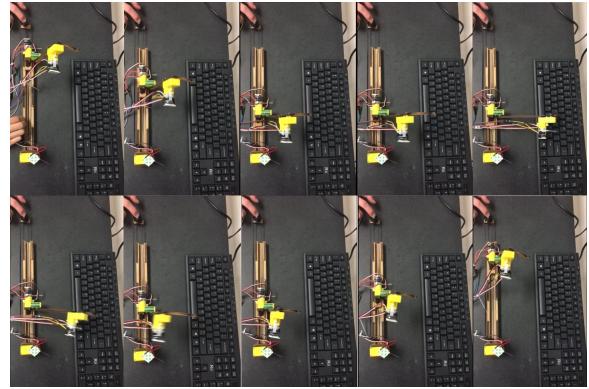


Fig. 15. Experimental Result

The robot was used to press the “I” key on a keyboard, during which the joint angles, and joint errors were recorded onto a text file for analysis. Shown below are the joint angles of each joint with respect to control loop count, during the PD control with gravity compensation. The joint angles and errors were recorded during the control loop for each joint, and once the control loop was stopped measurements were no longer recorded.

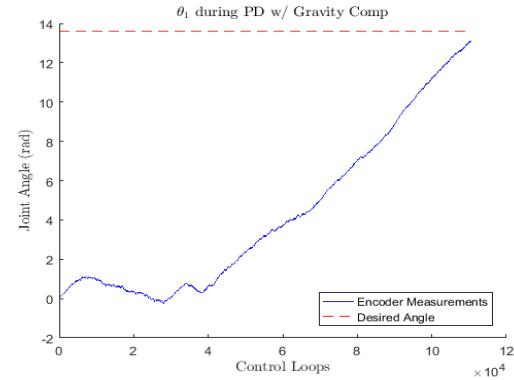


Fig. 16. θ_1 Joint Angle during Control

In Figure 16, the joint angle of the first motor is shown during control. During the start of the motion, the motor struggles to pull the sled along the track, and also struggles to overcome the internal friction of the motor. These forces cause wild vibrations of the encoder magnet (mounted to an unconstrained motor rotor), causing false measurements and unpredictable behavior. After the motor accelerates and spins at a more consistent speed the vibrations are smaller in magnitude, and the encoder can more accurately measure position. Despite the smooth appearance however, the measurements are still extremely noisy as shown in the other joint angles which reached their desired joint angles more quickly, and thus the finer details of the measurements are more apparent. Additionally, the measurement did not reach the desired value before the control loop was stopped as seen in the figure. This is because the PD controller causes the desired torque, and consequently voltage and PWM to reach zero before the desired value is reached. This effect is primarily caused by the derivative gain. In the physical system however, the motors

cannot be backdriven so the inertia of the arms plays no role in the position. Thus when the control input is zero the motor stops instantly. Additionally, the motors cannot operate for small voltages, less than 3V. This causes a deadzone in the PWM spectrum where despite receiving a PWM signal the motors will not turn. This exacerbates the problem described previously, since the motor will shut off not just when the control input is zero, but when it is close to zero as well. From testing the motors we found that the PWM deadzone was large, from about 30 % duty cycle to 70 % duty cycle. Outside this range the motor operates as expected, with duty cycle increasing or decreasing motor velocity. Finally, from this Figure, the measurements look smooth, but the resolution of the encoders was an additional limitation of the system. Due to the gear ratio and number of poles on the magnets used by the hall effect sensors to count rotations, the encoder resolution may be given by Equation 20

$$\theta_{min} = \frac{2\pi}{k_r \cdot p} = 0.0164\text{rad} \quad (20)$$

Where $k_r = 48$ is the gear ratio of the motors, and $p = 8$ is the number of magnetic poles, for 4 north and 4 south poles. This minimum resolution is the resolution of angular position on the output shaft of the motor gearbox.

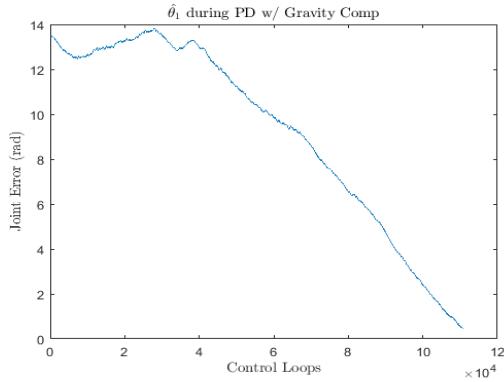


Fig. 17. θ_1 Joint Error during Control

In Figure 17, the joint error of the first motor is shown during control.

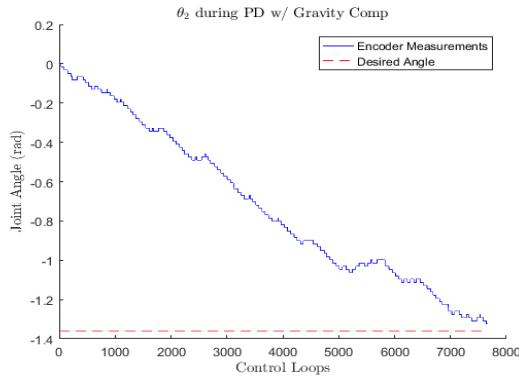


Fig. 18. θ_2 Joint Angle during Control

In Figure 18, the joint angle of the second motor is shown during control. Due to the decreased distance of the desired joint angle, the control loop approaches the desired value more quickly than for motor one. Thus it is possible to see the discretization of the joint angles due to encoder limitations. Additionally, as discussed before the measurements are quite noisy due to continuous vibrations of the encoder magnet. These vibrations are caused by the modifications made to the motors to allow the magnets to be easily mounted to the motor rotor. The motor housing was filed down to give more space for the encoder magnet, but simultaneously, the motor rotor was less secure and free to move nearly one millimeter in any direction. Unlike for the first motor however, the second motor has no trouble starting its motion from the rest position, initialized as zero.

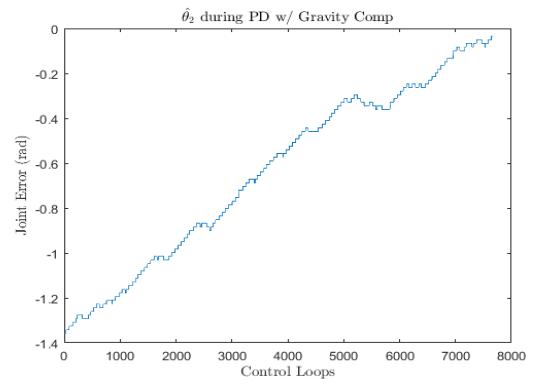


Fig. 19. θ_2 Error Angle during Control

In Figure 19, the joint error of the second motor is shown during control.

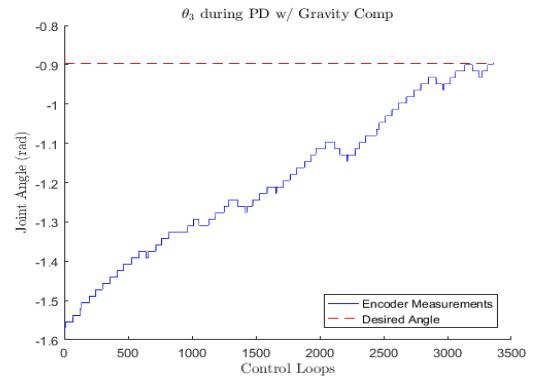


Fig. 20. θ_3 Joint Angle during Control

Joint three measurements are shown in Figure 20, with the same discontinuities and nonlinearities as the second joint. Since the desired motion for this joint was shorter still than for joint two, the encoder discretization can be clearly seen.

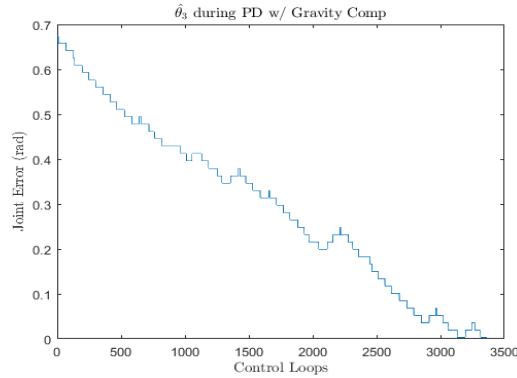


Fig. 21. θ_3 Error Angle during Control

In Figure 21, the joint error of the third motor is shown during control.

V. DISCUSSION

- The magnet used for encoder measurements does not stop at the same time as the motor rotor due to the inertia of the magnet and the high angular velocity of the motor. This causes measured angles to be larger in magnitude than actual values. This problem was addressed partially by subtracting the error for each measurement. The error was determined by measuring the position immediately after the control loop finished execution, then waiting for one second and measuring the position a second time. The difference between these measurements is the error caused by the inertia of the magnet, since the arm does not move after the motor inputs are zero.
- The vibration of the motor rotor is huge due to changing the modification made to the motor housing. This generates highly inaccurate encoder readings, propagating into the velocity measurements. We used a moving average technique, similar in effect to a low pass filter to help filtering out the “noise” from random fluctuations. Average velocity was calculated over 100 velocity measurements to compensate for the vibration errors.
- In the physical experiment, the robot did not hit the key precisely. The probable reasons for this inaccuracy are: backlash of the gear, the vibration of the motor’s rotation axis, the dead zone of the motor and imperfect measurements.
- The first motor needed an initial force to move the platform because of static friction between the sled and the rail. This friction is due to warping after the gluing of the laser cut parts causing a tight tolerance on the interface. Initial force is also needed for joint two when link one moves back to the home position after striking a key. Although the simulation indicates the motor has enough torque to lift the mass of the two links and motor, the physical experiment failed to do so. This may be because the simulation did not take many factors into consideration as discussed in the assumptions. Additionally, modifying the motor housing caused the motor rotor to gain some ability to move, reducing the torque output of the motor due to misalignment.

- We specified the order of motor movements in order to prevent collisions with the keyboard. For each key, the actuation order should be joint one, then joint two, then joint three. For moving back to the home position between key strokes, actuation the order should be joint two, then joint three, then joint one. Performing the actuation in this order allows the robot to move predictably. This order is necessary to prevent the end effector from dragging along the keyboard. Consider for example, if joint one was able to move more quickly than joints two and three after striking a key due to the gravitational forces on those two joints. In this case the end effector may not reach clearance in the vertical direction before joint one actuation begins, causing the end effector to press unwanted keys or suffer damage to the robot.

VI. FUTURE DIRECTIONS

Future plans include increasing the accuracy of the key striking motion, increasing the number of planar arms and using alternative end effectors to diversify the use of this robot for other tasks. To increase the accuracy of the key striking motion, new motors which have not been modified will be used to reduce the vibrations of the system. Additionally, new sensors which do not depend on a press fit magnet will be used to measure the position, like a potentiometer or optical encoder. These improvements will greatly reduce vibrations and improve the consistency of the system. Next, by increasing the number of planar arms in the system, the robot will be able to type punctuation using the shift key, or accomplish keyboard shortcuts such as copy and paste. Finally, some alternative end effectors could be used to change the function of the robot, such as by attaching a pen to create a drawing robot, or attaching a gripper to create a pick and place robot as discussed earlier.

REFERENCES

- [1] Craig, J. J. *Introduction to Robotics: Mechanics and Control*. 4th ed. Upper Saddle River, NJ: Prentice Hall, 2018.
- [2] Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. *Robotics: Modelling, Planning and Control*. Advanced Textbooks in Control and Signal Processing. London: Springer London, 2009.
- [3] Lynch, K. M. and Park, F. C. *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, 2017.
- [4] Siciliano, B. and Khatib, O. eds. *Springer Handbook of Robotics*. Berlin Heidelberg: Springer-Verlag, 2008.
- [5] Spong, M. W., Hutchinson, S., and Vidyasagar, M., *Robot Modeling and Control*, Wiley, 2005.

APPENDIX

A. Bill of Materials

The following Table III shows the parts purchased, their price and who purchased each part, along with a description.

TABLE III
BILL OF MATERIALS

| Part | Description | Cost | Purchaser |
|-------------------------|---|----------|-----------|
| Raspberry Pi | To control the robot and perform computations | \$36.49 | Zhi |
| Power Supply | To power the motors with high current | \$21.91 | Zhi |
| Power Supply Switch | To turn the power supply on and off | \$6.99 | Zhi |
| L293D | Control the motors | \$15.48 | Xuan |
| Keyboard | Test the robot on a sacrificial keyboard | \$9.00 | Xuan |
| Fuses | Fuses for power supply switch | \$9.98 | Preston |
| Zip Ties | For securing the encoders to the motors | \$3.85 | Preston |
| Gorilla Glue | For glueing the laser cut pieces together | \$7.77 | Preston |
| Breadboard Cables | To connect the electronics via breadboard | \$10.85 | Preston |
| DC Motors | To move the robot | \$7.90 | Zhi |
| Hall Effect Sensor Pair | To build encoders | \$38.85 | Zhi |
| Pulleys | To restrain the timing belt | \$14.70 | Zhi |
| Belt (too small) | A belt to move the sled along the track | \$3.86 | Zhi |
| Sanding Sponge | To sand the laser cut pieces, and remove glue | \$3.57 | Zhi |
| Screws | To mount the motors on laser cut parts | \$8.71 | Zhi |
| Wood Dowel Pin | To mount the pulley | \$4.20 | Zhi |
| Plastic Washer | To mount the motors on laser cut parts | \$3.23 | Zhi |
| Belt | Replaced the belt which was too small | \$4.26 | Zhi |
| Total | | \$211.60 | |

B. Work Division

The following Table IV summarizes how work was split between the members of our team.

TABLE IV
WORK DIVISION BY TASK

| General Task | Specific Task | Owner(s) |
|----------------|------------------------------|----------|
| CAD Model | CAD Rail | Preston |
| | CAD Link1 | Xuan |
| | CAD Link2 | Zhi |
| | CAD Motors | Zhi |
| Simulation | Develop Kinematics | Xuan |
| | Develop Inverse Kinematics | Xuan |
| | Measure Physical Parameters | Xuan |
| | Build dynamic model | Zhi |
| | Gain Optimization | Zhi |
| Software | Control Motors | Preston |
| | Read Encoders | Preston |
| | Calculate Gravity Force | Zhi |
| | Implement Control | Zhi |
| | Implement Inverse Kinematics | Xuan |
| | Implement Kinematics | Xuan |
| Physical Setup | Laser Cut Parts | Preston |
| | Build Parts and Glue | Preston |
| | Assemble Encoders | Preston |
| | Modify Motors | Zhi |
| | Setup Circuit | Zhi |
| | Connect to Pi and Run Code | Preston |