Horizontal Throwing Arm: Activation Time Sweep and Nonlinear Optimization

Big Bearing Gang: Ronak Roy, Jenny Zhang, David von Wrangel, Ethan Hammons

Introduction

How do humans throw? How do we intuitively know when to start extending the arm during the swing to throw faster? When do we move the wrist and extend the forearm relative to shoulder movement to create the most effective whipping motion and maximize forward velocity of the projectile thrown?

This project explores arm action dynamics as affected by relative link lengths, looking at a highly simplified model of a 3-dof arm in a 2D plane. We hope to answer these questions:

- By switching the first and second joints from off to full voltage at some activation times, what enables the maximum projectile velocity?
- By applying a voltage profile, what trajectory leads to the highest endpoint velocity?
- How do the two control schemes compare? Is the improvement, if any, of the voltage profile sufficient to justify its added complexity?
- How does changing the lengths of the links affect the maximum achievable velocity?

System Model

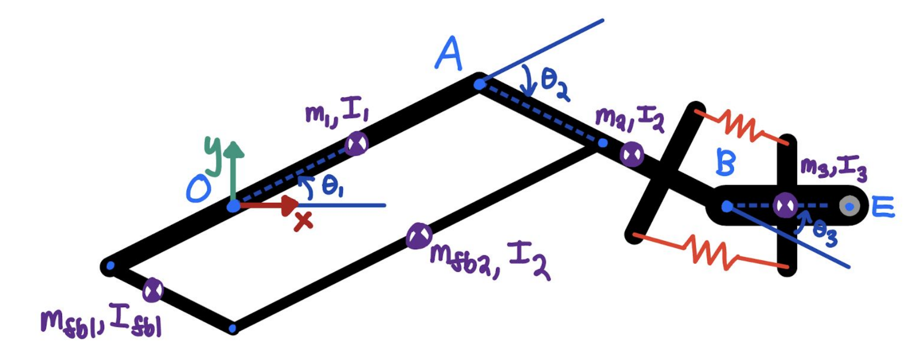


Figure 01: Drawing of the robot arm

Model Details:

- θ_1 driven directly
- θ_2 driven via parallelogram four-bar linkage
- θ_3 underactuated–rubber bands to vary stiffness of wrist joint

Activation Time Sweep

MATLAB Activation Time Sweep Setup

- Joint 'activated': apply maximum torque possible given state
- Joint 'deactivated', assume no relative angular velocity
- Sweep across [-1, 1] at 0.01s resolution for 2nd joint activation time relative to 1st
- If joint limits violated during rollout, take max velocity of trajectory before violation

Results

- Overall able to reach higher speeds when motors are allowed to run for longer times

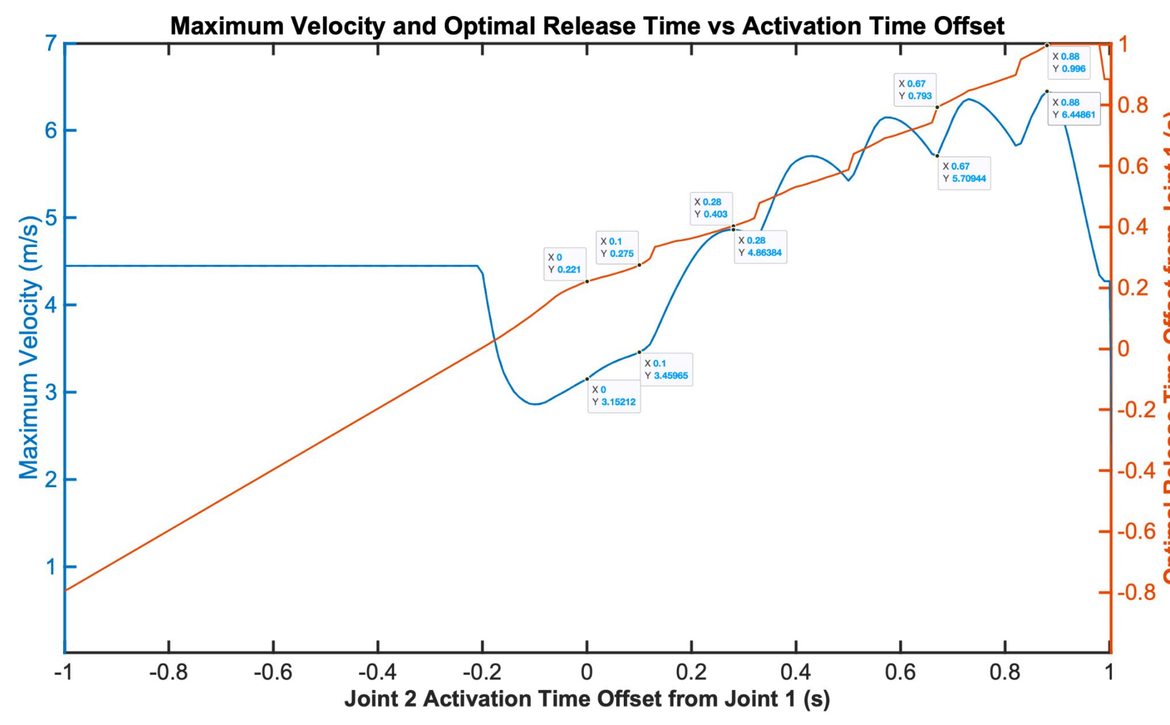


Figure 02: Activation Time Sweep Max Velocity

- Max velocity tapers off, marginal benefit of increasing runtime decreases
- Relative starting time of joints 1 and 2 important for reaching local maximums (depends on spring oscillation)
- Starting motor 2 after motor 1 is preferred to keep system moment of inertia low for initial acceleration

Note: Simulated for 9V operation for motor 1 and 6V operation for motor 2 to avoid brownout during hardware validation

Nonlinear Optimization

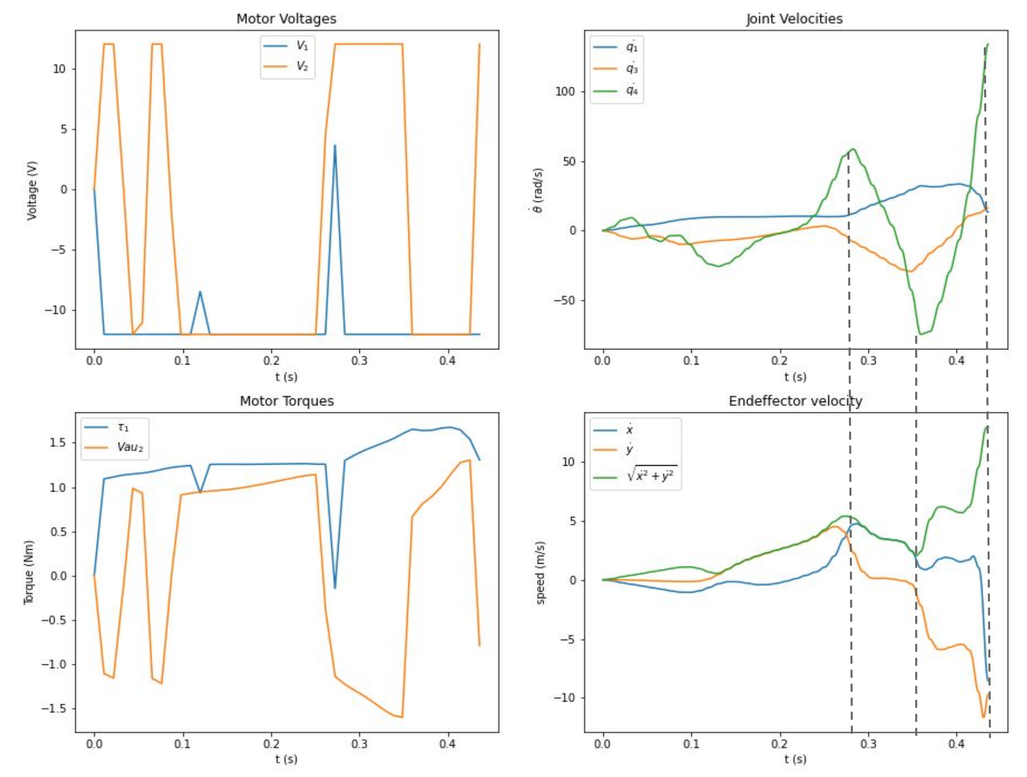


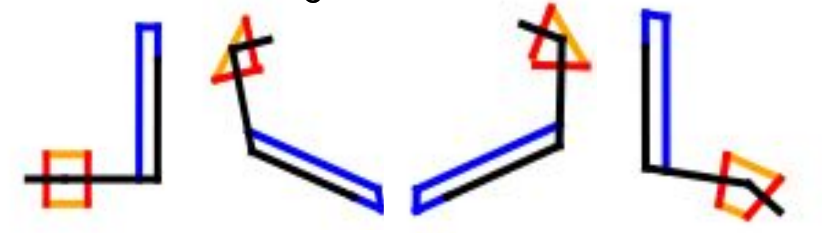
Figure 03: Nonlinear Optimization States

Nonlinear trajectory optimization (SNOPT) via Direct Collocation

- Same Lagrangian Dynamics
- Objective: - Norm of final speed
- Constraints:
  - No initial configuration constraint
  - Initial torques, joint velocity, wrist joint position constrained to 0
  - Motor dynamic constraint as:

$$\tau_1 = J\frac{\dot{q}_1[i] - \dot{q}_1[i-1]}{h[i]} - k_t\frac{V_1[i] - k_t\dot{q}_1[i]}{R} + \nu\dot{q}_1[i]$$

  - Bound Voltages at 12V



Times: 0s 0.378s 0.514s 0.602s

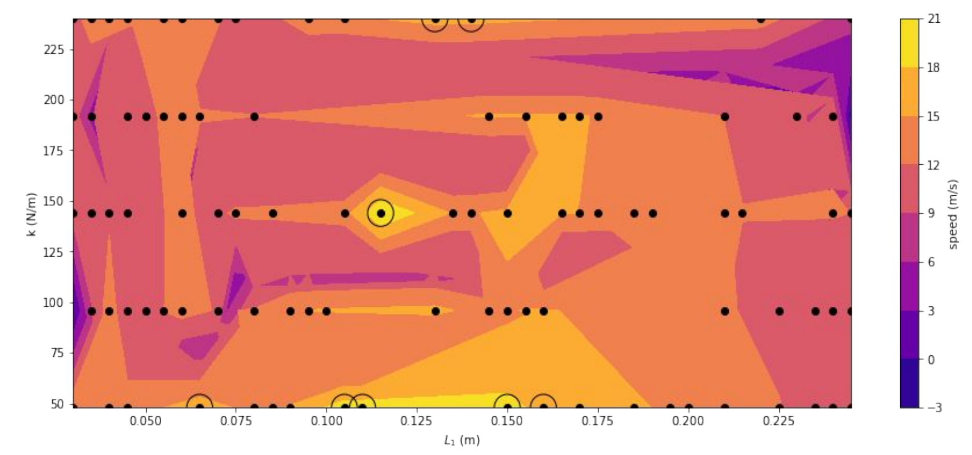


Figure 04: Link Length and Spring Stiffness Optimization

No significant structural trend as long as we can achieve maximum speed by:

- Joint "moving" into same direction
- Resonating the wrist (wrist velocity dominates)
  - No motor -> no speed constraint
  - First part of trajectory "pumps" the springs
  - Release time is driven by spring

Experimental Methods

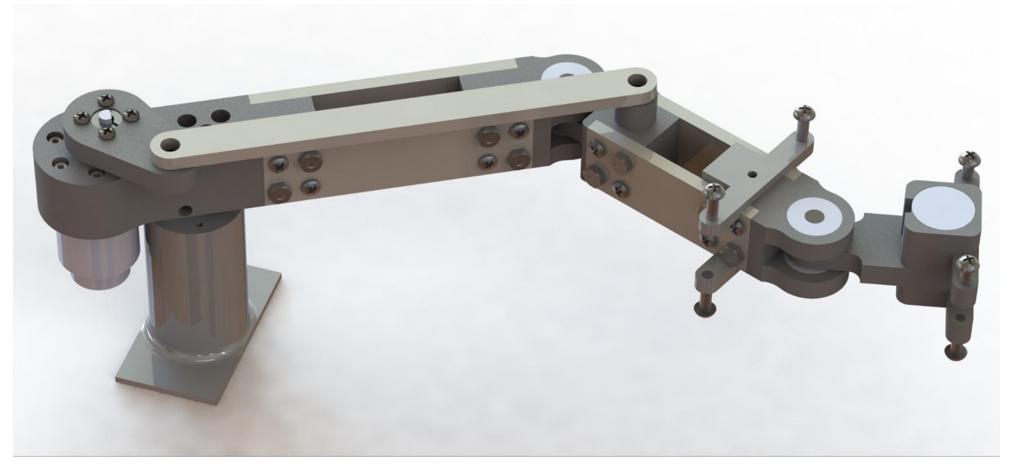


Figure 05: CAD of the robot arm

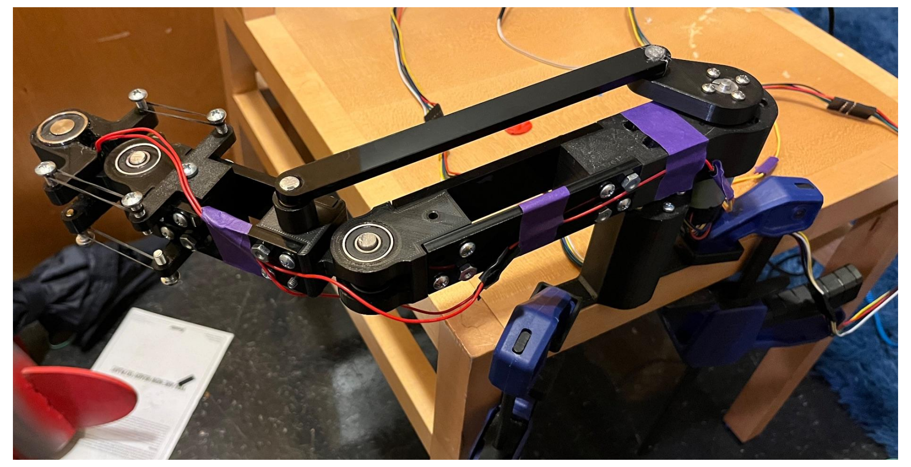


Figure 06: Fabricated hardware

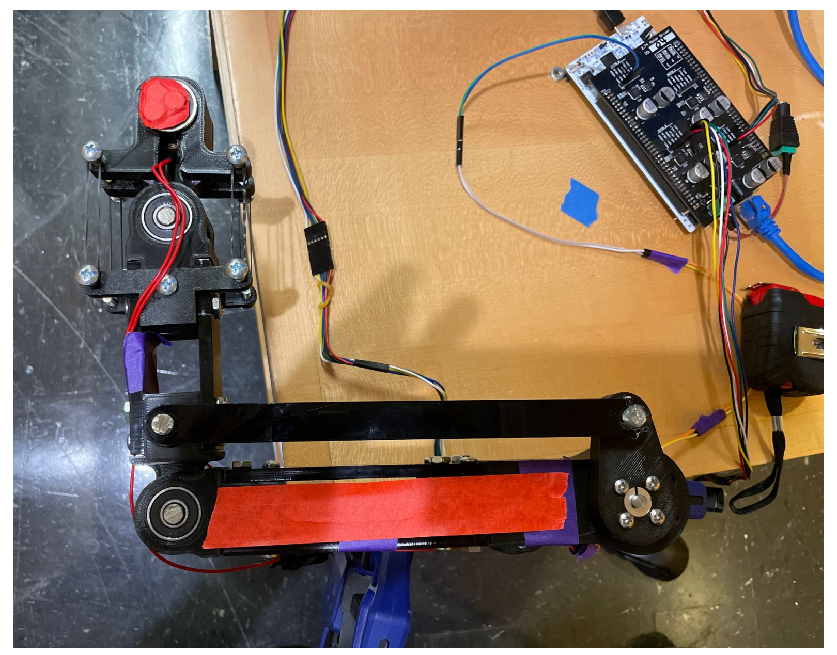


Figure 07: Hardware starting configuration

Configuration:

$\theta_1 = 0°$, $\theta_2 = -90°$, $\theta_3 = 0°$

rubber bands for 3rd joint

lightweight, easily visible projectile

MATLAB script passes voltage profile into Mbed controller

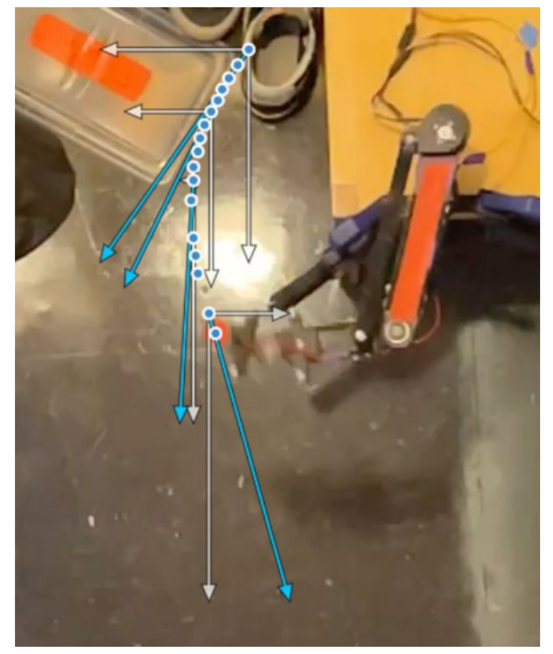
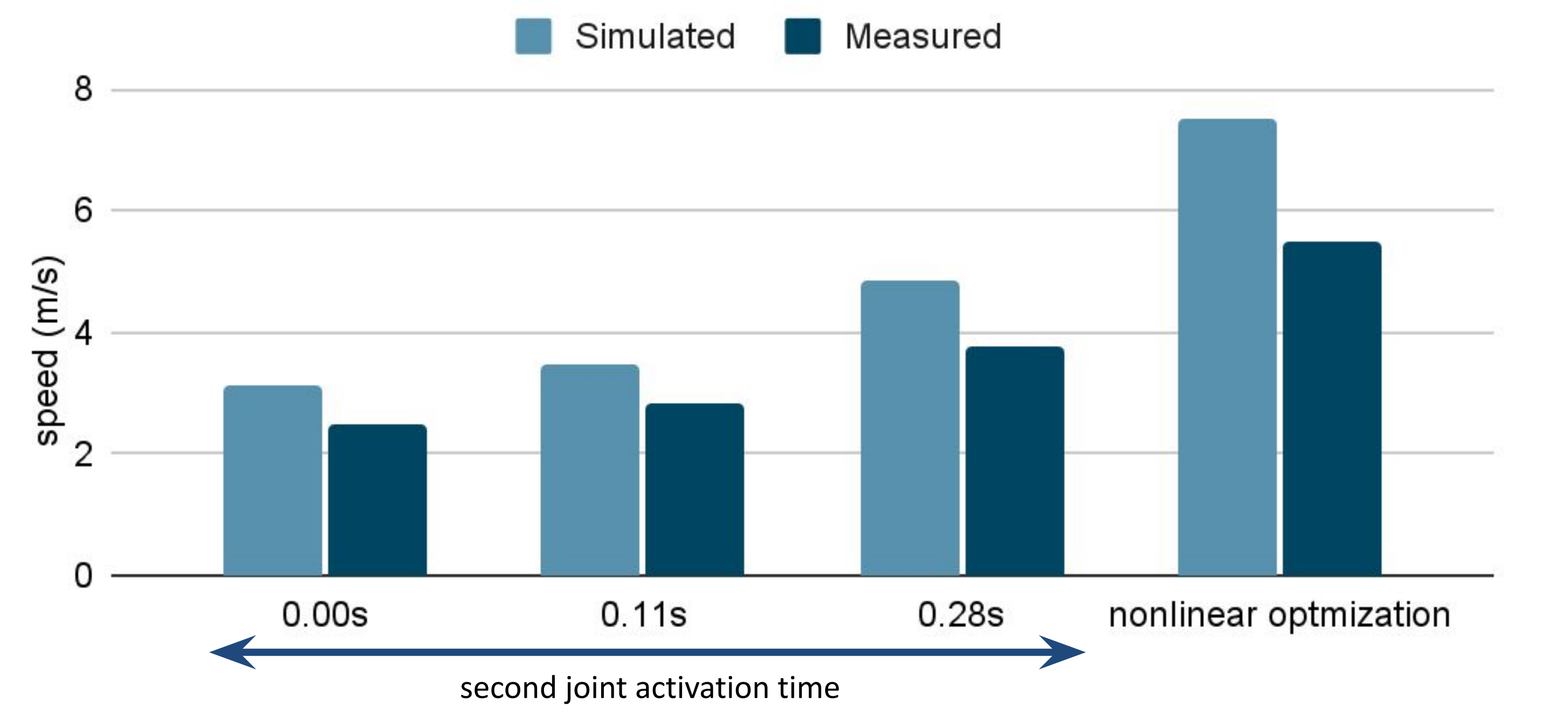


Figure 08: Video analysis

Experimental Results



Discussion

The experimental results compared to our simulated predictions only average around 20% error. This can be accounted for by the singularity of the four-bar linkage not being as smooth in hardware as in simulation, the motor controller struggling to apply enough power to match our desired torque curves, and inaccuracies in system parameters.

Conclusion

- The trajectory of an arm when throwing exploits the hard limitations of each joint–by accelerating each link to load the next link into being initially pushed as far back as possible, and then flicking the segment outward, the velocity of the endpoint can be maximized.
- When comparing our two optimization types, non-linear yields an almost 50% increase in throwing speed.
- When it comes to link lengths, the non-linear optimizer found no significant difference in changing them. This is due to it always exploiting the end-effector spring.

Acknowledgements

Thank you to Prof. Sangbae Kim, TA Elijah Stanger-Jones, and TA Se Hwan Jeon for all their advice and help.