MPC Project: Trajectory of Toy Cup-and-Ball During Free Fall

Kingston Chua 3037390459, Ning Zhang 3037405708, Peter Wang 3036342321

(Video link: https://drive.google.com/file/d/1AYZ0SgvgbdP_9CcuVBxeXW6qCf4D2J43/view?usp=sharing)

***Abstract* - In this report, we explore the application of model predictive control (MPC) in the ball-and-cup game. When playing the ball-and-cup game, the ball's motion transitions from circular motion to free fall, which makes it a nonlinear problem. Due to the complexity of solving nonlinear models, we simplified our model and focused on the ball's trajectory during freefall given initial conditions. MPC was then used to find the optimal path the cup must move to catch the ball.**

I. INTRODUCTION

The cup-and-ball is a traditional children's toy. This toy typically consists of a ball and cup that are bounded together with string. The objective of the game is to swing the ball from rest and catch the ball using the cup. Figure 1 shows an example of a successful attempt at landing the ball into the cup.

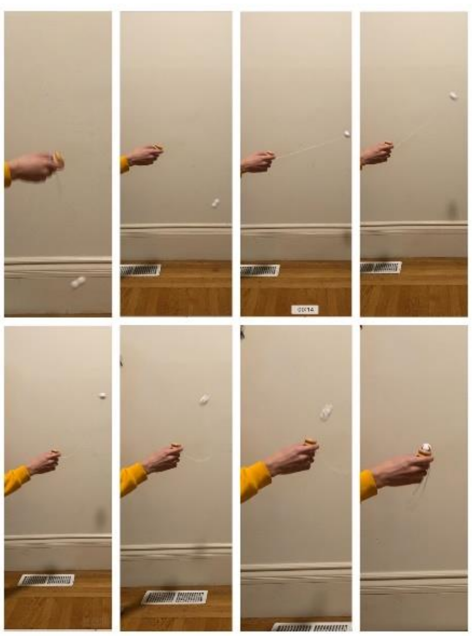


Figure 1: Playing Procedure

Although the game may seem simple at first glance, the ball-and-cup game has been the subject of intrigue in robotics research due to the toy's nonlinear system, presence of contact forces, and need for precise positioning as a terminal goal. Typically, when solving this problem through model-based approaches, the problem is broken down into two stages: (i) the swing-up stage, and (ii) the free-fall stage. During the swing-up stage, the cup and ball can be modeled as a cart with an inverted pendulum problem, where the cup is the cart, and the ball is a rigid pendulum [1]. By applying a horizontal acceleration to the cup, it is possible to change the ball's angular velocity. At a certain angle and angular velocity, the string between the ball and cup will no longer be taut, which marks the start of the free-fall stage. In the free fall stage, the ball can be represented as a projectile motion problem with the initial conditions taken from the final conditions of the swing-up stage.

In this paper, we simplify the cup-and-ball problem to only consider the free-fall stage and use a closed-loop control design to minimize the velocity of the cup while achieving the same position of the ball at the end of the simulation.

II. PROBLEM STATEMENT

When setting up the model, we made several assumptions. First, that there is no friction due to air. The effect due to the curvature and rotation of the earth is negligible. Also, that the string is massless and does not apply any forces to the ball or cup. Finally, the ball has a given instantaneous velocity the moment free-fall occurs.

The model can be represented as two point masses as seen in Figure 2.

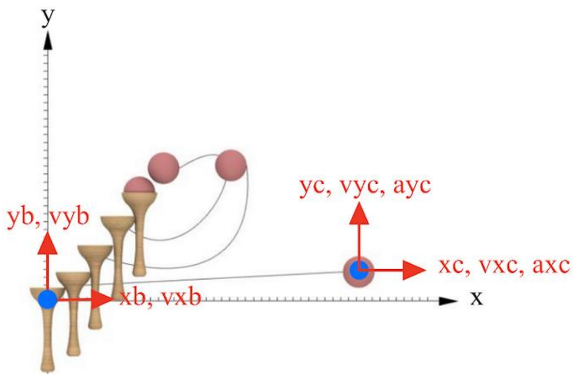


Figure 2: Model where the cup and ball are represented as point masses.

Each point mass can be described by the following linear system:

$$\begin{cases} \dot{x}_b = v_{xb}(t) \\ \dot{y}_b = v_{yb}(t) \\ \dot{v}_{xb} = 0 \\ \dot{v}_{yb} = -g \\ \dot{x}_c = v_{xc}(t) \\ \dot{y}_b = v_{yc}(t) \\ \dot{v}_{xc} = a_{xc}(t) \\ \dot{v}_{yc} = a_{yc}(t) \end{cases} \quad (1)$$

Where x_b, y_b, v_{xb}, and v_{yb} are the position and velocity of the ball in the corresponding direction on the Cartesian plane. The states x_c, y_c, v_{xc}, and v_{yc} represent the position and velocity of the cup in the corresponding direction. The inputs a_{xc} and a_{yc} are the acceleration of the cup. The acceleration due to gravity is assumed to be g = 9.8 m/s^2.

In order to implement the controller on the ball and cup, we use a simplified and discretized model with sampling time Ts = 0.01 s:

$$\begin{cases} x_b(\mathrm{k}+1) = x_b(\mathrm{k}) + T_s v_{xb}(k) \\ y_b(\mathrm{k}+1) = y_b(\mathrm{k}) + T_s v_{yb}(k) \\ v_{xb}(k+1) = v_{xb}(k) \\ v_{yb}(k+1) = v_{yb}(k) - T_s g \\ x_c(\mathrm{k}+1) = x_c(\mathrm{k}) + T_s v_{xc}(k) \\ y_c(\mathrm{k}+1) = y_c(\mathrm{k}) + T_s v_{xc}(k) \\ v_{xc}(k+1) = v_{xc}(k) + T_s a_{xc}(t) \\ v_{yc}(k+1) = v_{yc}(k) + T_s a_{yc}(t) \end{cases} \quad (2)$$

With the state constraint:

$$L^2 > \left(x_b(\mathrm{k}) - x_c(k)\right)^2 + \left(y_b(k) - y_b(k)\right)^2 \quad (3)$$

Where L = 0.36 m is the length of the string when taut. This constraint is used to prevent the ball from undergoing constraint motion.

The initial conditions of the ball and cup are as follows:

$x_{b0} = 0.232\, m$
$y_{bo} = 0.274\, m$
$v_{xbo} = -1.149\, m/s$
$v_{ybo} = 0.971$ m/s
$x_{co} = 0\, m$
$y_{co} = 0\, m$
$v_{xco} = 0$ m/s
$v_{yc0} = 0\, m/s$

The initial conditions from the ball were derived from a paper called “Learning to Play Cup-and-Ball with Noisy Camera Observations.” In the paper, the nominal terminal state of the ball of the swing-up phase is the angle $\theta_0 = 2.44\, rad$ relative to the paper’s reference frame and angular velocity $\dot{\theta}_0 = 4.18\, rad/s$. These terminal states were converted to our reference frame and used as our initial conditions of free-fall.

The terminal state of the model was set to:

$$\begin{aligned} -0.0001 \le x_b(N) - x_c(N) \le 0.0001 \\ -0.0001 \le y_b(N) - y_c(N) \le 0.0001 \\ -0.0001 \le v_{xb}(N) - v_{xc}(N) \le 0.0001 \\ -0.0001 \le v_{yb}(N) - v_{yc}(N) \le 0.0001 \end{aligned}$$

The reason for setting the terminal constraints to 0.0001 instead of 0 is that it is easier for the pyomo solver to compute and reduces likelihood of infeasibility.

III. MPC Design

The state-space representation of our variables are as follows:

$$x_k = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} = \begin{bmatrix} x_b \\ y_b \\ v_{xb} \\ v_{yb} \\ x_c \\ y_c \\ v_{xc} \\ v_{yc} \end{bmatrix}$$

$$u_k = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} a_{xc} \\ a_{yc} \end{bmatrix}$$

The model was then written in linear matrix form as:

$$A = \begin{bmatrix} 0 & 0 & dt & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & dt & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & dt & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & dt \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ dt & 0 \\ 0 & dt \end{bmatrix}$$

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

We then created a cost function that would minimize the velocity of the cup during travel to reduce the collision between the cup and ball upon contact, which increases the chance to catch the ball. The resulting Constrained Finite Time Optimal Control (CFTOC) problem is as follows:

$$\min_{x_7, x_8} \sum_{k=0}^{N-1} x_k' Q x_k$$

$$x_{k+1} = Az_k + Bu_k, \qquad \forall k = \{0, \dots, N-1\}$$

$$(x_{1k} - x_{5k})^2 + (x_{2k} - x_{6k})^2 \leq L^2, \quad \forall k = \{0, \dots, N-1\}$$

$$x_0 = \bar{x}_0$$

$$x_N = \bar{x}_N$$

The IPOPOT quadratic solver was then used to compute the optimal states and inputs up to the finite horizon.

IV. Results

In this problem, we chose the CFOTC horizon N=3. The CFOTC was solved using the initial conditions defined in the problem statement. The optimal states and inputs of the time step were recorded and fed back into the CFOTC as initial conditions to compute the next optimal output of the next time step. This process is repeated M = 15 to create a closed-loop MPC simulation of the trajectories of the ball shown on Figure 3.

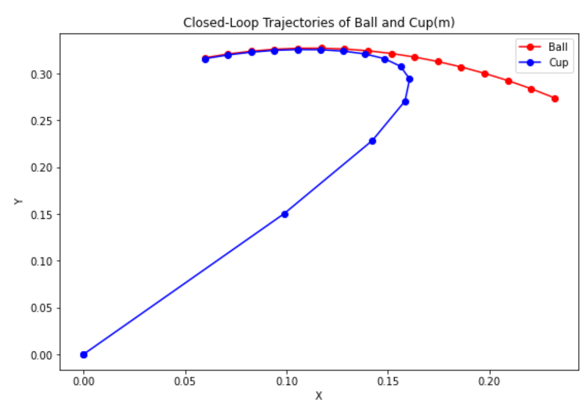


Figure.3 Trajectories of the ball and cup in MPC

We can see that after 7 steps of MPC simulation, the positions of ball and cup are very close, so that it will be quite possible to make sure that the cup is able to smoothly catch the ball in success. We assume this point was the best chance to catch the ball with the smallest velocity. And we also plotted the velocities of the ball and cup in Figure 4 to see how they change during this process. The blue line represents the cup, and the red line represents the ball, the first diagram is the velocity in x direction and the second diagram is in the y direction.

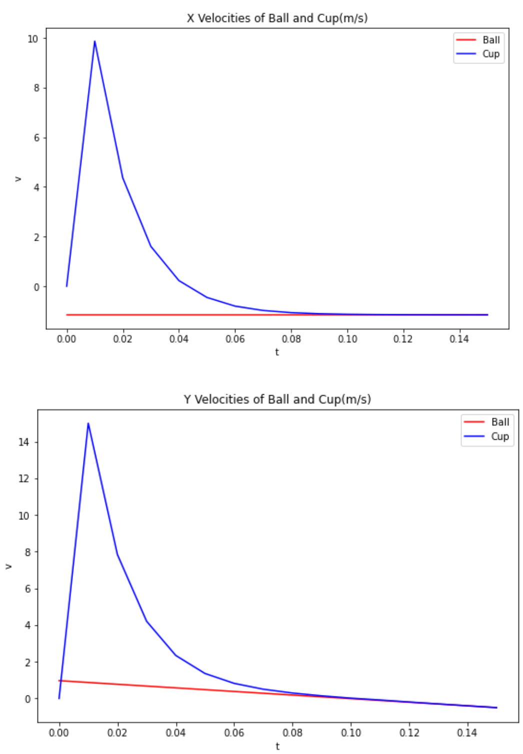
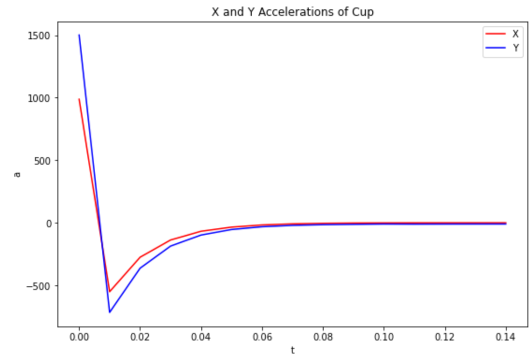


Figure.4 Velocity diagrams of the cup and the ball

From the velocity diagram, we can see that after about 0.07 s which was 7 steps of the simulation, the velocity of the ball and cup in both x and y direction matches each other, the trajectory diagram also shows that the position of them overlaps, which means the cup successfully catch the ball after 0.07 to 0.08 seconds in our MPC model.

The input by the MPC controller was also plotted in Figure 5.



From the acceleration diagram, we also notice that at around 0.07 s, the acceleration inputted on the cup becomes 0, which indicates that the cup has successfully reached the terminal constraint as specified in our MPC.

V. CONCLUSION

In this paper, we showed the use of a MPC controller to find the trajectory of a free-falling ball and control a cup to successfully catch the ball. Regarding future works, we originally tried to set a terminal constrain that make the cup catch the ball in more accurate position, but then we realized that the error or the noise in this case was not necessary to be considered since we are not going to do the catch test in real life. If we have chance to do the real test using a controller, design a closed-loop feedback control to deal with the discrepancy would be a crucial part in our project. We also thought about whether using velocity or acceleration as inputs in our project, after some tests, we decided to use acceleration as our input, we will try to use the velocity as inputs and compare the result if we have more time to do that.

REFERENCES

[1] Bujarbaruah, Monimoy, et al. "Learning to Play Cup-and-Ball with Noisy Camera Observations." 2020 IEEE 16th International

Conference on Automation Science and Engineering (CASE). IEEE, 2020.
[2] B. Nemec and A. Ude, “Reinforcement learning of ball-in-a-cup playing robot,” in 2011 IEEE International Conference on Robotics and Biomimetics, Dec 2011, pp. 2682–2987.
[3] T. Sakaguchi and F. Miyazaki, “Dynamic manipulation of ball-in-cup game,” in Proceedings of the 1994 IEEE International Conference on Robotics and Automation, May 1994, pp. 2941–2948 vol.4.
[4] A. Namiki and N. Itoi, “Ball catching in kendama game by estimating grasp conditions based on a high-speed vision system and tactile sensors,” in 2014 IEEE-RAS International Conference on Humanoid Robots, Nov 2014, pp. 634–639.
[5] D. Schwab, T. Springenberg, M. F. Martins, T. Lampe, M. Neunert, A. Abdolmaleki, T. Herkweck, R. Hafner, F. Nori, and M. Riedmiller, “Simultaneously learning vision and feature-based control policies for real-world ball-in-a-cup,” arXiv preprint arXiv:1902.04706, 2019.