Reactive control of a dual-arm ping-pong ball juggling robot

Yuquan Wang

Abstract— **This report summarizes a pingpong-ball juggling control strategy for a dual-arm robot. We assume that each arm has 7 Degrees of freedom (DOF) and the robot end-effector is a racket.**

Assuming that the perception (vision) system is able to provide accurate estimate of the ball trajectories, we use constraint-based programming to reactively control the robot such that the robot is able to handle the in-herent randomness of such a pingpong-ball paddling task.

In the preliminery experimental validation, we model the striking process with simplifed flight dynamics and impact dynamics.

I. Introduction

Cascade juggling of more than one ball combines deterministic process with random fluctuations [1]. The control or stabilization of such an inherent stochastic system requires sensory feedback in real time.

Constraint based programming provides a versatile framework for combining several different constraints into a single robot control scheme [2]. We take advantage of the redundancy of a robot manipulator to improve the execution of a reactive tracking task. Particularly due to the dexerity demand from more than one pingpong ball, we explore the redundancy of the robot by formulating proper task-dependent measures and/or constraints and integrating them with an optimization framework [3].

The rest of the report is organized as follows: we introduce how we model the juggling with certain assumptions in Sec. II, we propose an reactive control strategy in Sec. III and we conclude this report by introducing the experimental setup in Sec. IV

II. Modelling

A. Flight and impact dynamics

We are not aiming for using a robot hand which is able to open or close fingers [4], instead we simplify the catch/throw as paddling. According to the early pioneers in [5], we should model the striking process with proper flight dynamics of the ball and the impact dynamics between the ball and the racket.

Baically we can model the ball as point mass under gravity force using the following state space model:

$$\begin{bmatrix}\dot{\boldsymbol{b}}\\ \ddot{\boldsymbol{b}}\end{bmatrix} = \begin{bmatrix}0 & I\\ 0 & 0\end{bmatrix}\begin{bmatrix}\boldsymbol{b}\\ \dot{\boldsymbol{b}}\end{bmatrix} + \begin{bmatrix}0\\ \boldsymbol{g}\end{bmatrix} \tag{1}$$

where $\boldsymbol{b} \in \mathbb{R}^3$ represents the position of the ball and $\boldsymbol{g} = [0, 0, -9.8]^\top$ denotes the gravity force. We can construct an oberver of the flighting ball dynamics by feeding the discrete version of (1) as a state space model into a Kalman filter.

We assume that components of the ball's velocity tangent to the plane of impact remains unchanged after the impact. Along the normal direction of the impact plane, we define a positive restitution coefficient α to model the impact dynamics:

$$(\dot{\boldsymbol{b}}'_n - \boldsymbol{v}'_n) = -\alpha(\dot{\boldsymbol{b}}_n - \boldsymbol{v}_n), \tag{2}$$

where $\boldsymbol{b}'_n$ and $\boldsymbol{v}'_n$ denote the normal components of the ball and paddle velocites right after impact, while $\dot{\boldsymbol{b}}_n$ and $\boldsymbol{v}_n$ are prior to impact [5].

Taking the robot inertia into account, we can assume that paddle is way heavier than the ball such that $\boldsymbol{v}'_n = \boldsymbol{v}_n$. Then we can simplify (2) as:

$$\dot{\boldsymbol{b}}'_n = \dot{\boldsymbol{b}}_n + (1+\alpha)(\boldsymbol{v}_n - \dot{\boldsymbol{b}}_n). \tag{3}$$

B. Dual-arm motion and the Juggling strategy

From the principal components in three-ball cascade juggling [1], we know that the candidate trajectory of the two end-effectors of a dual-arm robot resembles the trajectory shown in Fig. 3.

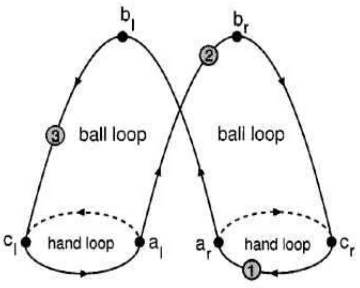

Fig. 1. Paths of the three balls and the two hands during cascade juggling. Between throw ($a_{l,r}$) and catch points ($c_{l,r}$) the balls describe a parabolic path which peaks at b_l, while the hands (subscripts l, r) follow a more or less elliptic path

Fig. 1: Fig.1 taken shamelessly from [1]

We can summarize from Fig. 3 that the coordination of the two arms primarily depends on the throwing/catching. As the common robot arm consists of several revolute joints, we can use cyclic motion to model the end-effector(hand) loop.

Using the aforementioned flight dynamics we can construct an observer of the ball trajectories with a Kalman filter, based on which we can online regulate the coordination of the two arms.

As we earlier assumed in Sec. II-A that the catch/throw is simplifed as paddling, we can use a similar strategy as shown in Fig. 2

e-mail: {yuquan}@kth.se

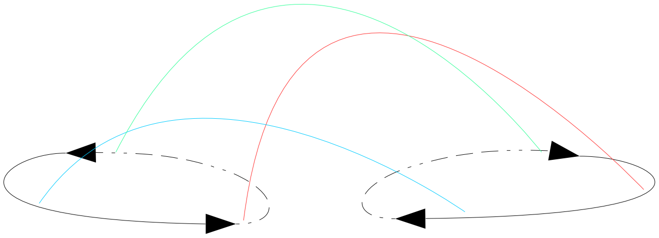

Fig. 2: By simplifing the cathc/throw with elastic collision between ball and racket, we can assign three collisions on the trajectory of each of the end-effectors. The tree distinct trajectories are marked with red-green-blue colors.

III. Reactive control solution

We decompose the control problem for each arm in different aspects in Sec. III-A and based on which we explain how to formate an optimization problem in Sec. III-B.

A. Constraints specification

As we do not restrict the moving space of the ball, we can not apply the mirror algorithm [6] to paddle the ball. As shown in Fig. 2, within one cycle of the ith paddle the jth ball has a nominal position $\boldsymbol{b}_j^i$ to impact with the ith paddle. Each nominal position is associated with a nominal direction $\boldsymbol{n}_j^i$.

Using the Kalman filter based on (1), we can predict the state of the ball and therefore we can calculate the position error $\boldsymbol{e_b}$ and the orientation error $\boldsymbol{e_n}$. The position error $\boldsymbol{e_b}$ could be calculated by directly taking the difference $\hat{\boldsymbol{b}}_j^i - \boldsymbol{b}_j^i$. We can use the direct design $\dot{\boldsymbol{e}} = -k\boldsymbol{e}$ to move the center of the racket to the landing position of the ball.

The orientation error $\boldsymbol{e_n}$ could be defined by unit quaternions, see [7]. Basically the idea is to use $\boldsymbol{e_n}$ to adjust the racket orientation such that the parabola of the ball points to the prespecified position.

Using the prespecified impact model and unit quaternion, we intuitively realize the correction with the following:

$$\boldsymbol{n}_j^{i\prime} = \frac{-\hat{\boldsymbol{n}}_j^i + \boldsymbol{n}_r}{2}, \tag{4}$$

where $\boldsymbol{n}_j^{i\prime}$ denotes the diretion of the ball after impact, $\hat{\boldsymbol{n}}_j^i$ denotes the direction of the ball prior to the impact, $\boldsymbol{n}_r$ denotes the normal direction of the racket.

As pointed out by [1] and constaint-based programming papers, e.g. [8], the smooth transition between constraints is important yet not easy to obtain. We can use change the translation and orientation by following timed sinus wave to achive a smoother transition.

B. Constraints integration

Suppose there are task-dependnet constraints and objectives η_0, $\boldsymbol{\eta}_1$ and η_2, we can calculate the motion of the robot end-effectors by formulating and solving optimization problems as: [3]

$$\begin{aligned}\min_{\dot{\boldsymbol{\theta}},\nu^2_{i=0,1,2}} \; & \frac{\partial \eta_3}{\partial \boldsymbol{\theta}}^\top \dot{\boldsymbol{\theta}} + \dot{\boldsymbol{\theta}}^\top Q \dot{\boldsymbol{\theta}} + w_1 \boldsymbol{\nu}_1^\top \boldsymbol{\nu}_1 + w_2 \nu_2^2, \\ \text{s.t.} & \frac{\partial \boldsymbol{\eta}_1}{\partial \boldsymbol{\theta}}^\top \dot{\boldsymbol{\theta}} + \boldsymbol{\nu}_1 = -\boldsymbol{k}_1(\boldsymbol{\eta}_1 - 0), \\ & \frac{\partial \eta_2}{\partial \boldsymbol{\theta}}^\top \dot{\boldsymbol{\theta}} + \nu_2 \geq -k_2(\eta_2 - b_2),\end{aligned}$$

where for $i = 1, 2$, $\boldsymbol{k}_i$ sets the convergence rates, b_i denotes the bounds and the weight w_i provides us a way to weight$\boldsymbol{\eta}_1$ and η_2 with respect to each other and the other objectives.

The objective function has three aspects: (1) We always minimize the joint velocities measure $\dot{\boldsymbol{\theta}}^\top Q \dot{\boldsymbol{\theta}}$ as the robot is redundant and we want the minimum norm solution. We use the positive diagonal matrix Q to weight the joint velocities against each other and the other objectives. (2) The slack variables $\boldsymbol{\nu}_i$ fix the potential infeasibility induced by the corresponding constraints. (3) On top of the above two aspects, we optimize the manipulability measure $\eta_3 = -\sqrt{\det JJ^\top}$.

IV. Validation

We can use dynamics in a practical setup to improve the real-time performance [9]. Whereas in the current form of the simulation, we restrict ourselves to the kinematics. Namely, we generate robot joint velocities to perform the paddling task.

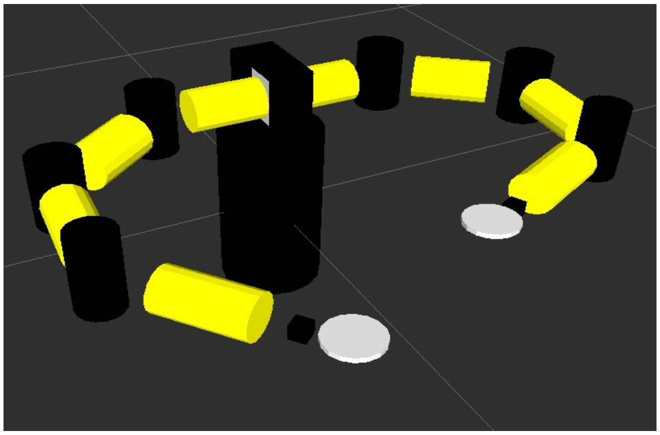

Fig. 3: The tailor made dual-arm robot model for the paddling simulation.

In order to improve the coordination between the two arms, we use parallel threads to perform the *rostopic* communication. Basically each arm uses an exclusive thread to communicate with ROS. Please find more details in the code.

References

[1] A. Post, A. Daffertshofer, and P. J. Beek, “Principal components in three-ball cascade juggling,” *Biological Cybernetics*, vol. 82, no. 2, pp. 143–152, 2000.

[2] J. De Schutter, T. De Laet, J. Rutgeerts, W. Decré, R. Smits, E. Aertbeliën, K. Claes, and H. Bruyninckx, “Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty,” *The International Journal of Robotics Research*, vol. 26, no. 5, pp. 433–455, 2007.

[3] Y. Wang, F. Vina, Y. Karayiannidis, C. Smith, and P. Ögren, "Dual arm manipulation using constraint based programming," in *19th IFAC World Congress*, Cape Town, South Africa, August 2014.

[4] J. Kober, M. Glisson, and M. Mistry, "Playing catch and juggling with a humanoid robot," in *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on*. IEEE, 2012, pp. 875–881.

[5] A. A. Rizzi, L. L. Whitcomb, and D. E. Koditschek, "Distributed real-time control of a spatial robot juggler," *Computer*, vol. 25, no. 5, pp. 12–24, 1992.

[6] M. Buehler, D. E. Koditschek, and P. J. Kindlmann, "Planning and control of robotic juggling and catching tasks," *The International Journal of Robotics Research*, vol. 13, no. 2, pp. 101–118, 1994.

[7] Y. Wang, C. Smith, Y. Karayiannidis, and P. gren, "Whole body control of a dual-arm mobile robot using a virtual kinematic chain," *International Journal of Humanoid Robotics*, vol. 13, no. 01, p. 1550047, 2016. [Online]. Available: http://www.worldscientific.com/doi/abs/10.1142/S0219843615500474

[8] N. Mansard, O. Khatib, and A. Kheddar, "A unified approach to integrate unilateral constraints in the stack of tasks," *Robotics, IEEE Transactions on*, vol. 25, no. 3, pp. 670–685, 2009.

[9] A. Nakashima, Y. Sugiyama, and Y. Hayakawa, "Paddle juggling of one ball by robot manipulator with visual servo," in *Control, Automation, Robotics and Vision, 2006. ICARCV'06. 9th International Conference on*. IEEE, 2006, pp. 1–6.