

Ball in double hoop: demonstration model for numerical optimal control [★]

Martin Gurtner Jiří Zemánek

*Faculty of Electrical Engineering
Czech Technical University in Prague
Technická 2, 166 27 Praha 6, Czech Republic
{martin.gurtner,jiri.zemaneck}@fel.cvut.cz*

Abstract: Ball and hoop system is a well-known model for the education of linear control systems. In this paper, we have a look at this system from another perspective and show that it is also suitable for demonstration of more advanced control techniques. In contrast to the standard use, we describe the dynamics of the system at full length; in addition to the mode where the ball rolls on the (outer) hoop we also consider the mode where the ball drops out of the hoop and enters a free-fall mode. Furthermore, we add another (inner) hoop in the center upon which the ball can land from the free-fall mode. This constitutes another mode of the hybrid description of the system. We present two challenging tasks for this model and show how they can be solved by trajectory generation and stabilization. We also describe how such a model can be built and experimentally verify the validity of our approach solving the proposed tasks. *

* All codes and drawings are available at <http://github.com/aa4cc/flying-ball-in-hoop>

Keywords: Trajectory planning; Optimal control; System modeling; Hybrid systems; Control education; Mechatronic systems

1. INTRODUCTION

Ball and hoop system is well-established among laboratory models used in teaching control. Its origin dates back to Wellstead (1980, 1983) who introduced it as a simple model having qualitatively the same dynamics as the liquid slop problem; in addition, Wellstead (1980, 1983) and Fabregas et al. (2011) also described how this model can be used in teaching of linear control theory. More specifically, the model was used in a regime where the ball is close to the stable position on the hoop and where a linear approximation of motion dynamics is valid.

In contrast, we consider full repertoire of motions the system offers. We describe the system as a hybrid model with a mode where the ball rolls on the outer hoop and with a mode where the ball has dropped out and entered a free-fall mode. This description enables us to generate dynamically richer trajectories (for instance, the ball can go through the top of the hoop or even fly in free fall); nevertheless, such trajectories also require more advanced control techniques. First, one needs to find a way to generate such trajectories. Second, the trajectories have to be stabilized; in other words, one cannot simply take the controls corresponding to a desired trajectory, apply it to the system and expect to observe the desired trajectory. This model is ideal for a demonstration of how these techniques, trajectory generation and stabilization, can be applied because it is relatively simple and allows

one to pose tasks which are rather difficult to solve by other means. We propose and solve two such tasks. Specifically, we formulate each task as an *Optimal Control Problem (OCP)* with nonlinear dynamics and limited control input. We solve the OCPs and show how to stabilize the obtained trajectories.

In addition, we added another hoop inside the outer one. This further extends the variety of tasks that can be solved on this system; one can, for example, swing up the ball on the outer hoop, let it fall and stabilize it on the inner hoop. It is worthy of note that the stabilization of the ball on the inner hoop can be viewed as stabilization of a disk on a disk; that itself is an interesting problem (for references, see Ryu et al. (2013)).

This paper is organized as follows. In Section 2, we derive a mathematical model of the system. In the following section, we present two examples of interesting and challenging tasks for this system and propose an approach solving them. In Section 4, we describe hardware setup of the system and verify the viability of the proposed approach solving the tasks.

2. MODELING

The system is cartooned in Fig. 1 and the modes of the hybrid description are displayed in Fig. 2. The hybrid description has three modes: the ball rolls on the outer hoop (S1), the ball is in free fall (S2), and the ball rolls on the inner hoop (S3).

Before we delve into the derivation of equations of motion for each mode, let us devote a few words to coordinate

[★] This work was supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS16/232/OHK3/3T/13, and partially by the Czech Science Foundation within the project P206/12/G014.

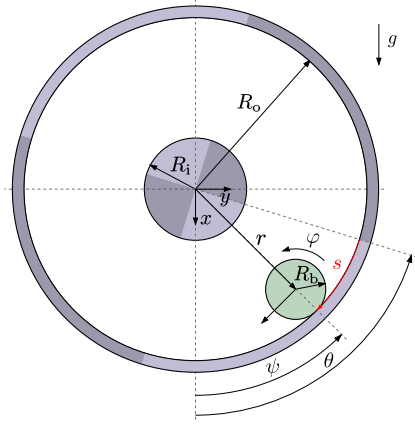


Fig. 1. A sketch of the ball and hoop system.

systems in which we will describe the motion of the ball. Due to the rotational symmetry, polar coordinates are a natural choice for modes S1 and S3. In contrast, Cartesian coordinates are more suitable for the free-fall mode (S2) because then the equations of motion are linear. In fact, it turns out that the description of motion in S2 in polar coordinates is strongly nonlinear. Thus, from the implementation point of view (i.e. to increase numerical stability), one should use polar coordinates for modes S1 and S3 and Cartesian coordinates for S2. Nevertheless, transformations of the coordinates would make equations in this paper unnecessarily complicated, and thus we stick to polar coordinates for all three modes.

For simplicity, we will not state the time dependence explicitly. At first, we derive equations of motion for each mode and after that, we describe guards and reset maps for transitions between the modes.

2.1 Outer hoop (S1)

A model of a ball rolling in a hoop was derived by Wellstead (1983). For reader's convenience—and because Wellstead made a small mistake in the derivation of the model—we rederive the model.

We use *Euler-Lagrange equation* to derive the model. Let us choose angles ψ and θ (see Fig. 1) as the generalized coordinates. We assume that the ball rolls without slipping. In addition, we do not consider the dynamics of the hoop as we assume that we can directly command the angular acceleration of the hoop $\ddot{\theta}$. In fact, the acceleration will be our input to the model and it constitutes a rheonomic (time-varying) constraint. This constraint reduces the number of independent coordinates to one.

The kinetic co-energy of the ball is

$$T^* = \frac{1}{2}mv^2 + \frac{1}{2}I(\dot{\varphi} + \dot{\theta})^2, \quad (1)$$

where m is the mass of the ball, I is the moment of inertia of the ball and $\varphi = \frac{s}{R_b} = \frac{R_o}{R_b}(\theta - \psi)$. The translational velocity v and the angular velocity $\dot{\varphi}$ of the ball can be expressed as follows

$$v = -(R_o - R_b)\dot{\psi}, \quad (2a)$$

$$\dot{\varphi} = \frac{R_o}{R_b}(\dot{\theta} - \dot{\psi}). \quad (2b)$$

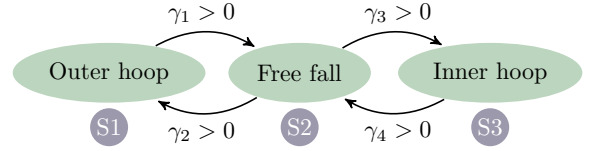


Fig. 2. Modes of the hybrid description of the ball and hoop system.

Thus, the kinetic energy expressed in the generalized coordinates is

$$T^* = \frac{1}{2}m(R_o - R_b)^2\dot{\psi}^2 + \frac{1}{2}I\left(\frac{R_o + R_b}{R_b}\dot{\theta} - \frac{R_o}{R_b}\dot{\psi}\right)^2. \quad (3)$$

Potential energy of the ball is given by

$$V = -mg(R_o - R_b)\cos\psi \quad (4)$$

and the system content modeling friction of the ball is

$$D = \frac{1}{2}b\dot{\varphi}^2 = \frac{1}{2}b\frac{R_o^2}{R_b^2}(\dot{\theta} - \dot{\psi})^2. \quad (5)$$

Defining the Lagrangian $\mathcal{L} = T^* - V$, an equation of motion of the ball can be computed by Euler-Lagrange equation as

$$\frac{d}{dt}\left(\frac{\partial\mathcal{L}}{\partial\dot{\psi}}\right) - \frac{\partial\mathcal{L}}{\partial\psi} + \frac{\partial D}{\partial\dot{\psi}} = 0 \quad (6)$$

which results in the following differential equation

$$\bar{a}_o\ddot{\psi} + \bar{b}_o\dot{\psi} + \bar{c}_o\sin\psi + \bar{d}_o\dot{\theta} = \bar{e}_o\ddot{\theta}, \quad (7)$$

with coefficients

$$\bar{a}_o = m(R_o - R_b)^2 + I\frac{R_o^2}{R_b^2}, \quad (8a)$$

$$\bar{b}_o = b\frac{R_o^2}{R_b^2}, \quad (8b)$$

$$\bar{c}_o = mg(R_o - R_b), \quad (8c)$$

$$\bar{d}_o = -\bar{b}_o, \quad (8d)$$

$$\bar{e}_o = I\frac{R_o}{R_b}\left(\frac{R_o}{R_b} + 1\right). \quad (8e)$$

2.2 Free fall (S2)

The motion of the center of the ball in free fall can be easily described in Cartesian coordinates because the only force acting on the ball is the gravitational force. Considering the Cartesian coordinate system shown in Fig. 1, we have

$$\ddot{x} = g \quad \text{and} \quad \ddot{y} = 0. \quad (9)$$

In order to get equations of motion in polar coordinates (r, ψ) (see Fig. 1), we only need to transform the coordinates by relations $x = r\cos\psi$ and $y = r\sin\psi$. This way, we get

$$\ddot{r} = r\dot{\psi}^2 + g\cos\psi, \quad (10a)$$

$$\ddot{\psi} = -\frac{1}{r}(g\sin\psi + 2\dot{\psi}\dot{r}). \quad (10b)$$

To fully describe the state of the system during free fall of the ball we also need to describe evolution of φ ; we assume that the angular velocity $\dot{\varphi}$ is constant during the free fall.

2.3 Inner hoop (S3)

A sketch of the ball rolling on the inner hoop is shown in Fig. 3. The equations of motion for the ball rolling on

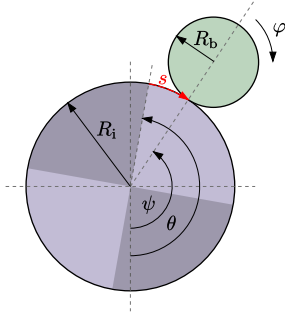


Fig. 3. A sketch of the ball rolling on the inner hoop.

the inner hoop can be derived along similar lines as for the ball on the outer hoop. The only difference is that the effective rolling radius is $R_i + R_b$ instead of $R_o - R_b$ and angle φ has the opposite orientation. Therefore, we skip the derivation and state directly the resulting equations of motion:

$$\bar{a}_i \ddot{\psi} + \bar{b}_i \dot{\psi} + \bar{c}_i \sin \psi + \bar{d}_i \dot{\theta} = \bar{e}_i \ddot{\theta}, \quad (11)$$

where

$$\bar{a}_i = m(R_i + R_b)^2 + I \frac{R_i^2}{R_b^2}, \quad (12a)$$

$$\bar{b}_i = b \frac{R_i^2}{R_b^2}, \quad (12b)$$

$$\bar{c}_i = mg(R_i + R_b), \quad (12c)$$

$$\bar{d}_i = -\bar{b}_i, \quad (12d)$$

$$\bar{e}_i = I \frac{R_i}{R_b} \left(\frac{R_i}{R_b} - 1 \right). \quad (12e)$$

2.4 Transition between modes

Transition from S1 to S2 Mode S1 (outer hoop) is valid as long as the centripetal force F_c acting on the ball is larger than the normal component of the gravitational force F_g with respect to the hoop. Mathematically, it is valid if

$$F_g \cos \psi + F_c = mg \cos \psi + m(R_o - R_b) \dot{\psi}^2 > 0. \quad (13)$$

If this condition does not hold, the ball drops of the hoop. It readily follows from (13) that the guard for transition from S1 to S2 is

$$\gamma_1 = -g \cos \psi - (R_o - R_b) \dot{\psi}^2 > 0. \quad (14)$$

The reset map for states r , \dot{r} , φ and $\dot{\varphi}$ is

$$r^+ = R_o - R_b, \quad (15a)$$

$$\dot{r}^+ = 0, \quad (15b)$$

$$\varphi^+ = (\theta^- - \psi^-) \frac{R_o}{R_b}, \quad (15c)$$

$$\dot{\varphi}^+ = \frac{R_o + R_b}{R_b} \dot{\theta} - \frac{R_o}{R_b} \dot{\psi}. \quad (15d)$$

The first three relations are apparent. The last one is derived from the inertial angular velocity ($\dot{\varphi}^- + \dot{\theta}^-$) of the ball and (2b). The remaining states transit to the mode S2 without any change, that is $\theta^+ = \theta^-$, $\dot{\theta}^+ = \dot{\theta}^-$, $\psi^+ = \psi^-$ and $\dot{\psi}^+ = \dot{\psi}^-$.

Transition from S2 to S1 This transition occurs when the ball hits the outer hoop. Thus the guard is

$$\gamma_2 = r - R_o + R_b > 0 \quad (16)$$

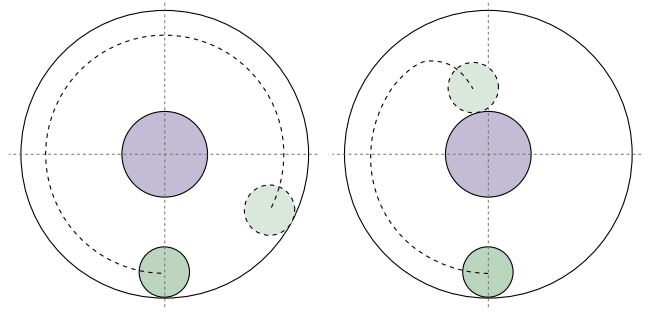


Fig. 4. A cartoon of the challenging tasks to be solved with the ball and hoop system.

and the corresponding reset map for ψ is

$$\psi^+ = \psi^- + \dot{\psi}_{\text{rot}}^- \quad (17)$$

where $\dot{\psi}_{\text{rot}}^-$ is derived from the angular velocity of the ball $\dot{\varphi}^-$ and can be computed from (2b):

$$\dot{\psi}_{\text{rot}}^- = \dot{\theta}^- - \frac{R_b}{R_o} \dot{\varphi}^-. \quad (18)$$

The remaining states transit unchanged, that is $\theta^+ = \theta^-$, $\dot{\theta}^+ = \dot{\theta}^-$ and $\psi^+ = \psi^-$.

Transition from S2 to S3 Analogously to the previous case, the transit occurs when the ball hits the inner hoop and the guard is

$$\gamma_3 = R_i + R_b - r > 0. \quad (19)$$

The reset map is also almost the same; the only difference is that the angular velocity $\dot{\psi}_{\text{rot}}^-$ here is computed as follows

$$\dot{\psi}_{\text{rot}}^- = \dot{\theta}^- + \frac{R_b}{R_i} \dot{\varphi}^-. \quad (20)$$

Transition from S3 to S2 Similarly to the transition from the outer hoop to free fall, the transition from the inner hoop to free fall occurs when the centripetal force becomes larger than the normal component of the gravitational force with respect to the hoop, that is

$$\gamma_4 = g \cos \psi + (R_i + R_b) \dot{\psi}^2 > 0.$$

and analogously to (15), the reset map is

$$r^+ = R_i + R_b, \quad (21a)$$

$$\dot{r}^+ = 0, \quad (21b)$$

$$\varphi^+ = -(\theta^- - \psi^-) \frac{R_i}{R_b}, \quad (21c)$$

$$\dot{\varphi}^+ = -\left(\frac{R_i - R_b}{R_b} \dot{\theta} - \frac{R_i}{R_b} \dot{\psi} \right), \quad (21d)$$

with the remaining states θ , $\dot{\theta}$, ψ and $\dot{\psi}$ transiting unchanged.

3. EXAMPLES OF INTERESTING TASKS

The hybrid description of the system allows us to pose visually appealing and from control perspective challenging tasks. In this section, we show two such tasks and describe control algorithms solving them.

Task 1: Roll the ball around the outer hoop

This task is sketched in Fig. 4 on the left side. In the beginning, the ball is in the steady position on the outer

hoop and the goal is to roll it around the hoop and get it back to the steady position. One might roughly imagine this task as the famous *Loop-the-Loop* stunt where a car rides around a vertical circle.

Naturally, this task is an instance of an *Optimal Control Problem (OCP)*. The ball is supposed to stay on the outer hoop and thus the hybrid model remains in the mode S1. To simplify the notation, let us define a state vector as

$$\mathbf{x} = [\theta, \dot{\theta}, \psi, \dot{\psi}]^\top. \quad (22)$$

and the state-space description corresponding to the mode S1 as

$$\dot{\mathbf{x}} = f_{S1}(\mathbf{x}, u), \quad (23)$$

where $u = \ddot{\theta}$ and

$$f_{S1}(\mathbf{x}) = \begin{bmatrix} x_2 \\ u \\ x_4 \\ \frac{1}{\bar{a}_o} (-\bar{b}_o x_4 - \bar{c}_o \sin x_3 - \bar{d}_o x_2 + \bar{e}_o u) \end{bmatrix}. \quad (24)$$

Now, we formulate the task as an OCP. In the initial time $T_0 := 0$, everything is steady and thus the initial condition is $\mathbf{x}(0) = [0, 0, 0, 0]^\top$. In a final time T_f , we require that the ball has rolled around the hoop, is back in the steady state and the hoop does not move; that imposes conditions $x_3(T_f) = -2\pi$, $x_4(T_f) = 0$ and $x_2(T_f) = 0$. Furthermore, to ensure that the ball does not drop out of the hoop, we require that the guard γ_1 is inactive during the whole trajectory. Instead of specifying the final time, we leave it as an optimization variable; we only require that the final time is smaller than a certain value T_{\max} . We also put bounds on the control u . Finally, the objective is to minimize the control u . Altogether, the OCP is

$$\begin{aligned} & \min_{u(t), \mathbf{x}(t), T_f} \int_0^{T_f} u^2(t) dt, & (25) \\ & \text{subject to: } x(0) = [0, 0, 0, 0]^\top, \\ & \quad x_2(T_f) = 0, \\ & \quad x_3(T_f) = -2\pi, \\ & \quad x_4(T_f) = 0, \\ & \quad \gamma_1(x(t)) < 0, \\ & \quad 0 < T_f < T_{\max}, \\ & \quad |u(t)| \leq u_{\max}, \\ & \quad \dot{\mathbf{x}}(t) = f_{S1}(\mathbf{x}(t), u(t)). \end{aligned}$$

To solve this OCP we use *direct collocation* (Hargraves and Paris (1987)) to discretize it to N equidistantly distributed knots in time and formulate it as a *nonlinear programming problem (NLP)*. This simplification, however, comes at a cost. NLP solvers usually does not guarantee optimality and provide only suboptimal solutions. Let us denote a (possibly suboptimal) solution provided by a NLP solver by u^* , \mathbf{x}^* and T_f^* .

By solving the NLP, we obtain a discrete trajectory of the system evaluated at certain time instants separated by sampling time $T_s = \frac{T_f}{N}$. Ideally, if the model captures the real system well enough and the sampling time T_s is short enough (say, a few milliseconds), an application of the control input $u^*(t)$ results in the desired state trajectory \mathbf{x}^* . Nevertheless, short sampling periods imply large number of knots N and thus make the NLP larger and possibly intractable. Furthermore, the model usually

mismatches the real system to some extent. Therefore, we need to stabilize the trajectory; we need to keep the system as close as possible to the desired trajectory.

Trajectory stabilization To stabilize the trajectory, we invoke the powerful framework of *Linear Quadratic Regulators (LQR)* and *neighboring extremals* (Bryson and Ho (1975)). We linearize the model around the trajectory and thus obtain a linear time-varying deviation model

$$\delta \dot{\mathbf{x}}(t) = \mathbf{A}(t) \delta \mathbf{x}(t) + \mathbf{B}(t) \delta u(t), \quad (26)$$

where $\delta \mathbf{x} = \mathbf{x} - \mathbf{x}^*$ and $\delta u = u - u^*$. The time-varying matrices are

$$\mathbf{A}(t) = \left. \frac{\partial f_{S1}(\mathbf{x}(t), u(t))}{\partial \mathbf{x}} \right|_{\mathbf{x}(t)=\mathbf{x}^*(t), u(t)=u^*(t)}, \quad (27)$$

$$\mathbf{B}(t) = \left. \frac{\partial f_{S1}(\mathbf{x}(t), u(t))}{\partial u} \right|_{\mathbf{x}(t)=\mathbf{x}^*(t), u(t)=u^*(t)}. \quad (28)$$

Now we design an LQR stabilizing the deviation model hence keeping $\delta \mathbf{x}(t)$ small and the system close to \mathbf{x}^* . We stick to a common notation and denote the state and input weight matrices by \mathbf{Q} and \mathbf{R} , respectively. The control law of the LQR is

$$\delta u(t) = -\mathbf{R}^{-1} \mathbf{B}^\top(t) \mathbf{S}(t) \delta \mathbf{x}(t), \quad (29)$$

where matrix $\mathbf{S}(t)$ is given by the solution of the differential Riccati equation

$$-\dot{\mathbf{S}}(t) = \mathbf{S}(t) \mathbf{A}(t) + \mathbf{A}^\top(t) \mathbf{S}(t) - \mathbf{S}(t) \mathbf{B}(t) \mathbf{R}^{-1} \mathbf{B}^\top(t) \mathbf{S}(t) + \mathbf{Q} \quad (30)$$

with final time condition $\mathbf{S}(T_f) = \mathbf{Q}$.

Finally, the applied control to the system is given by a sum of the nominal input $u^*(t)$ and the trajectory stabilizing term $\delta u(t)$:

$$u(t) = u^*(t) + \delta u(t). \quad (31)$$

Task 2: Get the ball on the inner hoop

The second task we would like to solve is to get the ball from the steady position on the outer hoop to the top of inner hoop and stabilize it there. This task is sketched in Fig. 4 on the right side. Once again, we formulate this problem as an OCP, find a trajectory solving the problem and design a regulator stabilizing this trajectory.

Apparently, the trajectory we are looking for starts and ends in different modes of the hybrid model. This poses especially difficult problem to the trajectory generation because in the discretization of the corresponding OCP, one does not know how many knots should be assigned to each mode. Fortunately, the requirement that the ball ends up on the inner hoop can be reformulated to a requirement that the ball ends up on the outer hoop at specific angle ψ_{des} with specific angular velocity $\dot{\psi}_{\text{des}}$. This is due to the fact that the free-fall mode is uncontrolled and thus the state of the system leaving the free-fall mode is fully determined by the state of the system entering the free-fall mode. Therefore, we can find a trajectory solving Task 2 by the same approach as we used for Task 1.

In addition to ψ_{des} and $\dot{\psi}_{\text{des}}$, we also require that the hoop has specific angular velocity $\dot{\theta}_{\text{des}}$ at the moment when the ball drops out of it. That is because by reset map (15d), $\dot{\theta}_{\text{des}}$ determines $\dot{\varphi}$ during the free fall and thus by reset map (20) also $\dot{\psi}$ just after the ball lands on the inner hoop.

Table 1. Parameters of the demonstration model

Description	Parameter	Value	Unit
Radius of the outer hoop	R_o	95.8	mm
Radius of the inner hoop	R_i	43.8	mm
Radius of the ball	R_b	7.7	mm
Inertia of the ball	I	$1.28 \cdot 10^{-6}$	kg m^2
Mass of the ball	m	0.032	kg
Friction of the ball	b	$1.4 \cdot 10^{-6}$	N m s
Gravitational constant	g	9.81	m s^{-2}

Therefore, we compute $\dot{\theta}_{\text{des}}$ so that $\dot{\psi}$ after the impact is minimized.

The OCP for Task 2 has exactly the same form as (25); the only difference here is that the conditions for the final states are

$$x_2(T_f) = \dot{\theta}_{\text{des}}, \quad (32a)$$

$$x_3(T_f) = \psi_{\text{des}}, \quad (32b)$$

$$x_4(T_f) = \dot{\psi}_{\text{des}}. \quad (32c)$$

The remaining steps to solve this task are almost the same as with the previous one. The only additional step is to design a regulator stabilizing the ball on the inner hoop. After the ball lands, we hand over the control to a stationary LQR designed for this purpose.

4. EXPERIMENTAL VERIFICATION

To verify the validity of our approach, we implemented the proposed control algorithm and tested it on a real hardware setup.

4.1 Construction of the model

The hardware setup used for the verification is displayed in Fig 5. Parameters of the setup are summarized in Table 1. The hoop is 3D-printed¹ and it has imbedded rubber o-rings serving as high-friction rails for a metal ball. The hoop is attached to a BLDC motor driven by a *PearControl ESC3* regulator enabling us to control the motor in a speed and current mode. To simplify the control algorithm, we let the current control to the motor regulator and used the speed mode to set directly a desired angular velocity $\dot{\theta}_{\text{des}}$. Nevertheless, in Section 2 we considered $\dot{\theta}$ to be the control input to the model, thus also the output of the regulator. Therefore, to get θ_{des} , we numerically integrate the output of the regulator $\dot{\theta}$. Judging from the experiments, the control system performs surprisingly well in spite of this simplification.

4.2 Implementation

The control law was discretized by sampling of (31) and it was implemented in *Simulink*. We used *Support Package for Raspberry Pi Hardware* to generate code, compile it, and run it on *Raspberry Pi 3*. *Raspberry Pi 3* has sufficient computational power that it also allows us to measure the position of the ball by processing images from *Raspberry Pi camera module*. The image-processing algorithm was implemented in *Python* and *OpenCV*. Both, the image processing and control algorithm run at 50 Hz.

¹ The drawings and codes are available at <http://github.com/aa4cc/flying-ball-in-hoop>

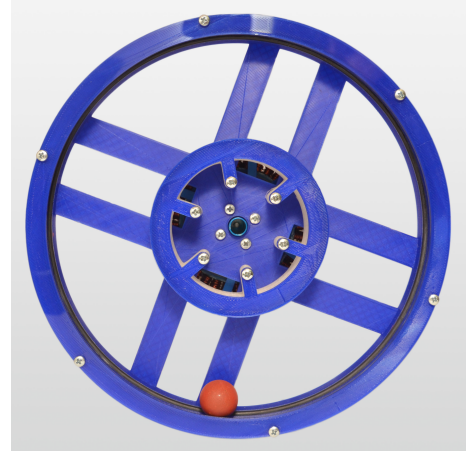
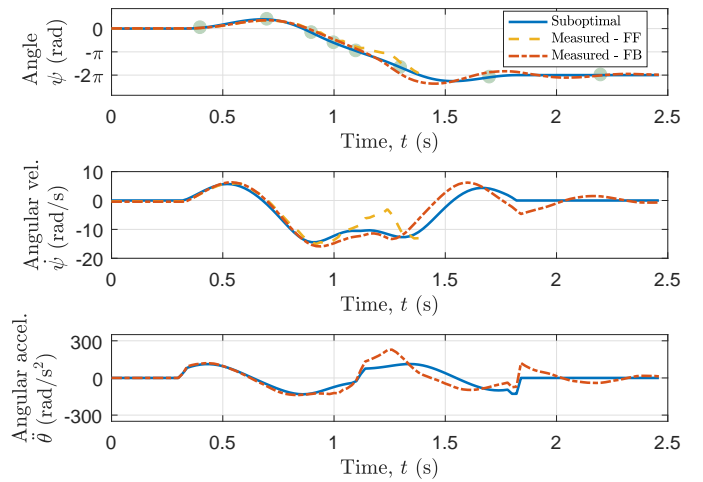
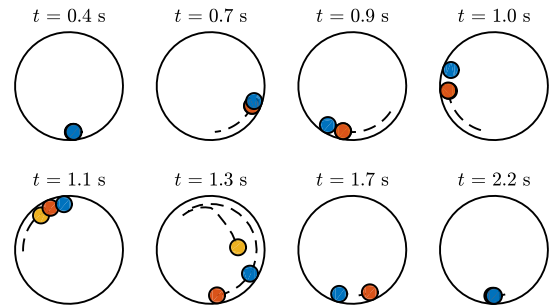


Fig. 5. A photo of the real hardware setup.

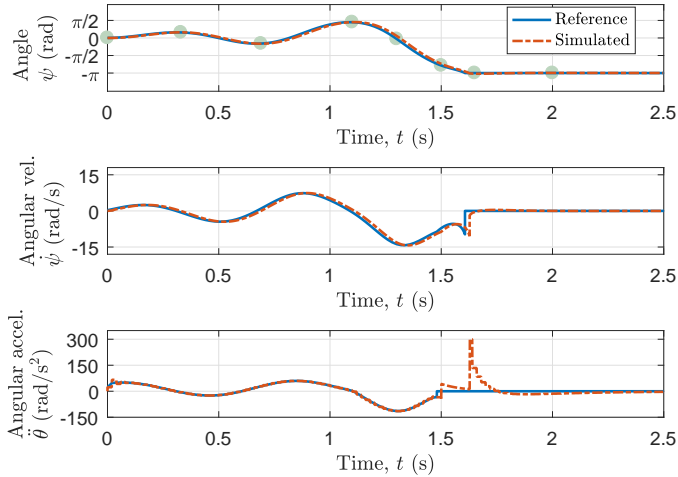


(a) Graphs

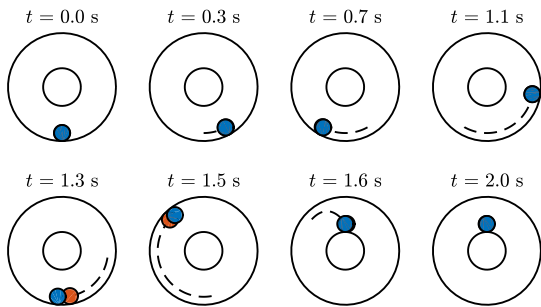


(b) Time instants

Fig. 6. A comparison of a nominal trajectory for Task 1 with a trajectory measured on the real system with feedback (FB) and without feedback (FF). The comparison has the form of (a) graphs and (b) visualizations of the position of the ball at several time instants. The time instants are also displayed in (a) by green dots. The colors of the balls in (b) correspond to the colors of trajectories in (a). The trajectory without feedback ends up at approximately $t = 1.2$ s because at that time the ball drops out of the hoop. The dimensions in (b) are not in scale.



(a) Graphs



(b) Time instants

Fig. 7. A comparison of a nominal trajectory for Task 2 with a trajectory obtained by simulation of the model with the trajectory stabilizing regulator. The comparison has the form of (a) graphs and (b) visualizations of the position of the ball at several time instants. The time instants are also displayed in (a) by green dots and the colors of the balls in (b) correspond to the colors of trajectories in (a). The dimensions in (b) are not in scale.

The trajectory stabilizing regulator, as a state regulator, needs to know values of all states. States $x_1 = \theta$ and $x_2 = \dot{\theta}$ do not need to be measured because they are given by integration of the input $u = \ddot{\theta}$. Angle of the ball ψ is measured by a camera and image processing. Nevertheless, the angular velocity $\dot{\psi}$ is rather difficult to measure directly and thus has to be estimated. For this purpose, we used a discrete *Extended Kalman Filter (EKF)* in the standard setting.

Beside the model mismatch and state estimation, in the reality, one also usually has to deal with a latency in the measurement. In our hardware setup, we measured the latency of the measurement of ψ_{meas} to be approximately two control periods (40 ms). Thus, at control period k , we at first estimate the state $\hat{\mathbf{x}}[k-2]$ by EKF based on the previous estimate $\hat{\mathbf{x}}[k-3]$, $\psi_{\text{meas}}[k]$ and $u[k-3]$ and then compensate for the delay by two-step prediction:

$$\hat{\mathbf{x}}[k-1] = \hat{\mathbf{x}}[k-2] + T f_{S1}(\hat{\mathbf{x}}[k-2], u[k-2]), \quad (33)$$

$$\hat{\mathbf{x}}[k] = \hat{\mathbf{x}}[k-1] + T f_{S1}(\hat{\mathbf{x}}[k-1], u[k-1]), \quad (34)$$

where T is the control period and it equals to 20 ms.

4.3 Verification

Regarding Task 1, a reference trajectory \mathbf{x}^* obtained by the solution of (25) together with a trajectory measured on the real hardware setup are shown in Fig. 6. Apparently, the validity of the control algorithm is verified; it is able to follow the desired trajectory to such an extent that the ball ends up in the desired position and does not drop out of the hoop. We have not yet implemented the solution of Task 2 on the real hardware setup. Nevertheless, the proposed solution works very nicely in simulations as Fig. 7 shows. Solutions of both tasks are also presented in a video clip available at <https://youtu.be/484GN4KBQnc>.

5. CONCLUSION

We showed that the well-known ball and hoop system can be used not only to demonstrate some aspects of linear control theory but also to demonstrate strengths of numerical optimal control. We extended the ball and hoop system by another hoop and described this extended system by a hybrid model. We presented two tasks for the extended system that are ideal for a demonstration of trajectory generation and stabilization and solve them. As the model of the extended system is simple, one can also use it for a demonstration of control techniques that are based on *Sum-Of-Squares* programming. For instance, our proposed solution of the presented tasks can be extended by *LQR trees* (Tedrake et al. (2010)) to work also for varying initial conditions. Furthermore, the inner hoop can be replaced by something else. For example, one can replace it by a beam and modify the Task 2 to “get the ball on a beam and stabilize it there”. Or, the inner hoop can be replaced by a figure-eight shaped rails and the “butterfly” task can be solved on the system (Cefalo et al. (2006)).

REFERENCES

- Bryson, J.A.E. and Ho, Y.C. (1975). *Applied Optimal Control: Optimization, Estimation and Control*. CRC Press.
- Cefalo, M., Lanari, L., and Oriolo, G. (2006). Energy-based control of the butterfly robot. *IFAC Proceedings Volumes*, 39(15), 1–6.
- Fabregas, E., Farias, G., Dormido-Canto, S., Dormido, S., and Esquembre, F. (2011). Developing a remote laboratory for engineering education. *Computers & Education*, 57(2), 1686–1697.
- Hargraves, C.R. and Paris, S.W. (1987). Direct trajectory optimization using nonlinear programming and collocation. *Journal of Guidance, Control, and Dynamics*, 10(4), 338–342.
- Ryu, J.C., Ruggiero, F., and Lynch, K.M. (2013). Control of nonprehensile rolling manipulation: Balancing a disk on a disk. *IEEE Transactions on Robotics*, 29(5), 1152–1161.
- Tedrake, R., Manchester, I.R., Tobenkin, M., and Roberts, J.W. (2010). Lqr-trees: Feedback motion planning via sums-of-squares verification. *The International Journal of Robotics Research*, 29(8), 1038–1052.
- Wellstead, P.E. (1980). Scale models in control systems engineering. *Transactions of the Institute of Measurement and Control*, 2(3), 137–155.
- Wellstead, P.E. (1983). The ball and hoop system. *Automatica*, 19(4), 401–406.