

MIT Open Access Articles

Optimal Shape and Motion Planning for Dynamic Planar Manipulation

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

Citation: Taylor, Orion and Alberto Rodriguez. "Optimal Shape and Motion Planning for Dynamic Planar Manipulation." Robotics: Science and Systems XIII, July 2017, Cambridge, Massachusetts, Robotics: Science and Systems Foundation, July 2017.

As Published: <http://dx.doi.org/10.15607/rss.2017.xiii.055>

Publisher: Robotics: Science and Systems Foundation

Persistent URL: <https://hdl.handle.net/1721.1/130051>

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Terms of use: Creative Commons Attribution-Noncommercial-Share Alike



Optimal Shape and Motion Planning for Dynamic Planar Manipulation

Orion Taylor and Alberto Rodriguez

Department of Mechanical Engineering, Massachusetts Institute of Technology
<tayloro, albertor>@mit.edu

Abstract—This paper presents a framework for optimizing both the shape and the motion of a planar rigid end-effector to satisfy a desired manipulation task. We frame this design problem as a nonlinear optimization program, where shape and motion are decision variables represented as splines. The task is represented as a series of constraints, along with a fitness metric, which force the solution to be compatible with the dynamics of frictional hard contact while satisfying the task.

We illustrate the approach with the example problem of moving a disk along a desired path or trajectory, and we verify it by applying it to three classical design problems: the rolling brachistochrone, the design of teeth of involute gears, and the pitch curve of rolling cams. We conclude with a case study involving the optimization and real implementation of the shape and motion of a dynamic throwing arm.

I. INTRODUCTION

Jai alai players use a *cesta* to catch and throw a ball at high speeds and with high accuracy (Figure 1). The cesta is an evocative example of the interplay between shape and motion. Their coordination allows players to transfer a large amount of energy to a ball while controlling its trajectory.

One might see jai alai as an interesting but contrived example of dynamic manipulation. An example where both shape and motion play key roles in determining the interaction between end-effector and object. On the contrary, in many solutions for robotic manipulation and locomotion, contact is designed and planned for pointy feet and finger tips. In these solutions, shape is usually of little relevance, and motion planning alone is tasked with controlling interaction. One might choose to see these solutions as equally contrived.

This paper starts from the assumption that shape *and* motion are design freedoms, and studies the problem of simultaneously optimizing them for planar manipulation tasks. The main contribution is a general framework to design shapes and to plan motions that work together to accomplish kinematic or dynamic tasks such as reaching a goal state, follow a path or a trajectory, or optimize a fitness function. A significant part of the contribution is in the representations for motion, shape, and tasks, that enable the optimization. The long term goal of our work is a better understanding, and the development of tools to effectively use shape and actuation in manipulation.

The proposed approach to design shape and motion, and the structure of this paper, is as follows:

- The **problem** has the form of a nonlinear program where shape and motion are decision variables. The dynamics-kinematics of planar contact are represented

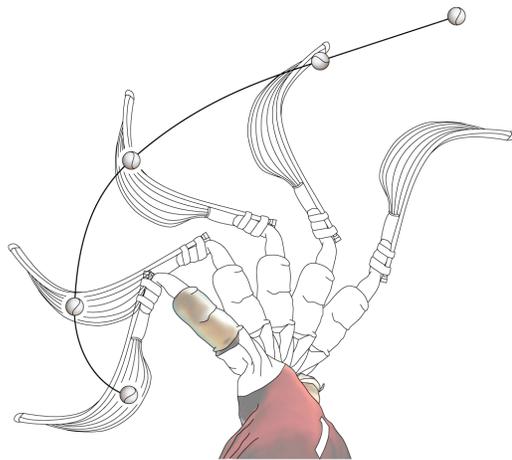


Fig. 1. Jai alai player throwing a ball. The cesta allows players to transfer a large amount of energy to the ball while controlling its rolling trajectory.

as constraints, similar to previous works on trajectory optimization through contact [13, 15].

- The **representation** of shape, motion, interaction, and task, is in terms of splines and dynamic-kinematic constraints at collocation points. Section III describes the system, Section IV the interaction and task constraints, and Section V their spline parametrization.
- We **illustrate** the approach in Section VI with the toy example task of moving a disk along a desired path or trajectory. We show the differences between optimizing either the manipulator's shape, or its trajectory, or both. A key observation is that often both shape and motion can be used to satisfy the same task, and that there is an inherent nullspace in their combined design space.
- We **verify** the approach in Section VII by formulating classical problems with known solutions: gear tooth profiles, rolling cams of variable transmission, and the rolling brachistochrone. The optimization approach yields correct solutions, and offers flexibility in studying variations.
- We **implement** the problem of planar dynamic throwing in Section VIII. In the optimized solutions, the shape and throwing trajectory cooperate with gravity to maximize reach and respect the frictional limits of the interaction between a throwing palm and a ball.

We finish with a discussion of the main challenges involved in simultaneously optimizing shape and motion and promising directions for future work.

II. RELATED WORK

Shape optimization for contact The field of shape optimization for contact interactions is a broad subset of mechanism design for automation. Examples include the design of part feeders, traps, fences, finger shapes, gear teeth, and cams. Caine [3] develops a framework and set of computational tools for designing the shapes of features in a vibratory bowl feeder. Similarly, Brokowski et al. [2] optimize the shape of a curved fence used to reorient parts traveling along a conveyor belt. Rodriguez and Mason [20, 21, 19] build a framework for computing end effector shapes for 1 DOF actuators and desired contact interactions based on sets of contact normals. Gear design is a relatively large field with extensive recent work [5, 11, 10, 24, 25] on methods to design shapes and pitch curves of circular and non-circular gears.

Trajectory optimization through contact The nonprehensile manipulation and locomotion communities have been especially interested in motion optimization involving frictional contact. Lynch and Mason. [12] introduced a control system for a one joint nonprehensile manipulator, enabling it to preform various dynamic tasks such as throwing and catching with a flat palm/arm. Ryu et al. [22] and Lippiello et al. [9] both create control frameworks for stabilizing and driving planar rolling systems. Becker and Bretl [1] design a set of control inputs for a sphere rolling on a table such that the cumulative rotation of the sphere is invariant with respect to its size- a unique example of motion planning that takes shape uncertainty into account. Posa et al. [15] propose one of the very few frameworks for trajectory optimization for systems that undergo intermittent frictional interaction based on a complementarity formulation for contact resolution.

Simultaneous shape and motion optimization Despite the abundance of work on either shape or motion optimization for contact interactions, there is relatively little work that approach both simultaneously. Reist and D’Andrea. [16, 17] optimize the motion and concavity of a paddle juggler capable of stably bouncing a ball without feedback. The approach is limited to the particular application, and the contact manipulation is limited to periodic instantaneous impacts. Chen [4] optimizes both the shape and control input for an underactuated throwing arm. Lynch [13] explores the design space (shape and motion) of a contact juggler for the task of butterfly juggling in a planar rolling system. This system is the closest work to this paper, and serves as primary inspiration for the proposed approach.

III. A PLANAR MANIPULATION SYSTEM

This paper focuses on a type of planar contact manipulation system consisting of two rigid bodies: a hand H and an object B , which share a single contact point, as illustrated in Figure 2. This section describes the notation and coordinates that we will use to describe their shapes, motions, and interactions. Throughout the paper we will use subscripts h and b referring to hand and object respectively.

In the paper we will make use of the following notation:

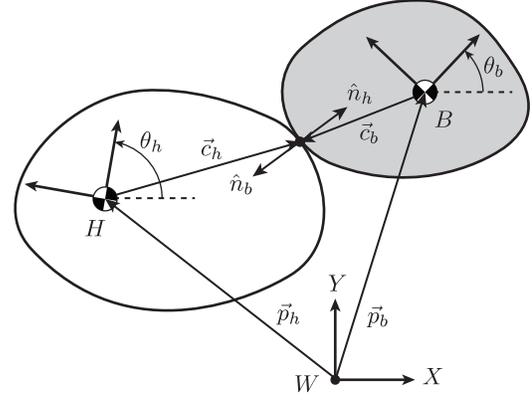


Fig. 2. Planar manipulation system consisting of a hand H manipulating an object B . We make use of an inertial reference frame W , and moving frames attached to the hand and ball. Their configurations are given by their position vectors and orientations (\vec{p}_h, θ_h) and (\vec{p}_b, θ_b) . Hand and object interact at contact point located at \vec{c}_h or \vec{c}_b and with normals \hat{n}_h and \hat{n}_b , all defined in the hand and object reference frames.

- $(\vec{p}_h, \theta_h) = (p_{h_x}, p_{h_y}, \theta_h)$ and $(\vec{p}_b, \theta_b) = (p_{b_x}, p_{b_y}, \theta_b)$ describe the planar poses of hand and object.
- $\vec{c}_h(s), \vec{c}_b(s) : [0, 1] \rightarrow \mathbb{R}^2$ parametrize the shape profiles of hand and object in their respective frames.
- s_h and s_b are the values of the parameter s at contact, thus $\vec{c}_h(s_h)$ and $\vec{c}_b(s_b)$ are the contact point in the hand and object reference frames.
- $\vec{v}(s) = \frac{d}{ds} \vec{c}(s)$ is the tangent vector to a shape at point $\vec{c}(s)$ in the hand and object reference frames.
- $\hat{n}(s) = R(\frac{\pi}{2}) \cdot \frac{\vec{v}(s)}{|\vec{v}(s)|}$ is the normalized outward facing surface normal to a shape at point $\vec{c}(s)$ in the hand and object reference frames.

Note that we will make frequent use of the rotation matrix about axis \hat{z} by θ radians, noted by $R(\theta)$.

The **motion** of the type of planar system we consider in this paper is then parametrized by the time-dependent functions:

$$\vec{p}_h(t), \theta_h(t), \vec{p}_b(t), \theta_b(t)$$

its **shape** is parametrized by the functions:

$$\vec{c}_h(s), \vec{c}_b(s)$$

and its **interaction** by the evolution of the contact point:

$$\vec{c}_h(s_h), \vec{c}_b(s_b)$$

which evolve in time with the parameters $s_h(t), s_b(t)$.

Note that the variables describing the system can be split into two categories: **design variables** $\vec{p}_h, \theta_h, \vec{c}_h, \vec{c}_b$ which describe parts of the system that can be directly controlled, i.e., the shape and motion of the hand and the shape of the ball; and **descriptor variables** $\vec{p}_b, \theta_b, s_h, s_b$ which describe underactuated degrees of freedom determined by the evolution of the design variables, i.e, the motion of the ball and the evolution of the contact point.

The system is also affected by the following constants, which we assume to be known: the mass of the object m , its moment of inertia I , gravity \vec{g} , and the coefficient of friction between hand and object μ .

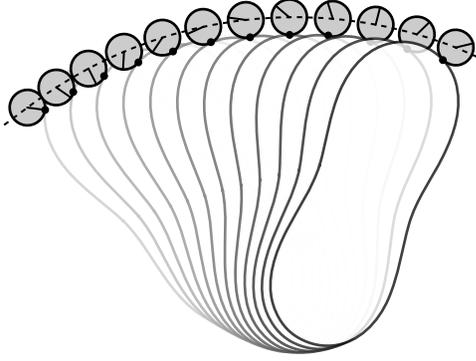


Fig. 3. Example of a simple planar manipulation task. Under gravity, the hand (white) moves an object (grey ball) along a trajectory. Note that the orientation of the object is linked to its displacement along the path.

IV. A PLANAR MANIPULATION TASK

We study two types of planar manipulation tasks framed as constrained satisfaction/optimization problems:

1. Produce a desired motion of the object, in the form of either a goal state, a path, or a trajectory to follow, e.g. move along a curve, as shown in Figure 3.
2. Optimize a behavior of the object defined by a fitness function, e.g., throw fast.

To simplify interactions, and for the sake of optimization complexity, we restrict the search for solutions to where hand and object interact with sticking or rolling contact, but do not slip with respect to each other.

The task then takes the form of a nonlinear optimization program with shape and trajectory as decision variables, subject to dynamic, kinematic, and task constraints. A big part of the work, which we describe in the two following subsections, is in finding a tractable way to formulate these constraints.

A. Kinematic and Dynamic Constraints

Before tailoring the system to any particular task, we need to make sure that the interactions it produces adhere to the laws of physics. To do so, we impose a series of kinematic and dynamic constraints that guarantee that contact is maintained with no penetration, that frictional forces are such that objects do not slide with respect to each other, and that the acceleration of the system is along the resultant of forces.

The expression of the kinematics of contact in the form of constraints was already described by Montana [14]. The algebraic representation we use in this paper is similar to the one proposed by Lynch et al. [13] to describe contact juggling. For their expression, we will make use of the notation introduced in Section III to describe the shape and motion of the system. For simplicity of notation, we suppress the dependencies of \vec{c}_h , \vec{c}_b and their derivatives on s_h and s_b respectively.

Contact Constraint The contact points in the hand and object must be the coincident in the world reference frame:

$$\vec{p}_h + R(\theta_h) \cdot \vec{c}_h = \vec{p}_b + R(\theta_b) \cdot \vec{c}_b \quad (1)$$

Tangency Constraint At the point of contact, the vectors tangent to the hand and object must be opposing each other (note that \angle measures the orientation angle of a vector):

$$\theta_h + \angle \vec{v}_h = \theta_b + \angle \vec{v}_b - \pi \pmod{2\pi} \quad (2)$$

Rolling Constraint The speed of the contact point in the hand and object must be opposite:

$$|\vec{v}_h| \dot{s}_h = -|\vec{v}_b| \dot{s}_b \quad (3)$$

Inertia Constraint The angular acceleration of the object must be consistent with the sum of torques ($\sum r \times f = I\alpha$):

$$(R(\theta_b) \cdot \vec{c}_b) \times m(\ddot{\vec{p}}_b - \vec{g}) = I\ddot{\theta}_b \quad (4)$$

Friction Cone Constraint The contact force exerted by the hand on the object must be inside the friction cone. If $\vec{f}_h = R(-\theta_h)m(\ddot{\vec{p}}_b - \vec{g})$ is the contact force applied from the hand to the object, in the hand reference frame, then:

$$\pm \frac{\vec{v}_h}{|\vec{v}_h|} \cdot \vec{f}_h \leq \mu \hat{n}_h \cdot \vec{f}_h \quad (5)$$

We refer to the first three constraints as *kinematic constraints* which we impose to all problems in this paper, and the last two as *dynamic constraints* which we impose only when appropriate.

These constraints are only a local approximation to the physics of interaction. They do not explicitly prevent, for example, the hand and object from intersecting at some point other than the studied contact point due to their global shapes, or due to their local curvatures. The global problem, while important and interesting, is significantly more difficult to formalize. The local constraints have proven useful and sufficient for the problems analyzed in the paper.

B. Manipulation Task Constraints

The previous constraints narrow the set of possible manipulation systems to those that are physically sound. Now we explore additional constraints and the use of fitness functions to represent manipulation tasks.

Decision variable constraints It is common to reduce the dimension of the problem by directly restricting the range of acceptable values of decision variables α .

$$\alpha = k \quad (6)$$

$$k_1 \leq \alpha \leq k_2 \quad (7)$$

Common examples are to constrain the hand to rotate about a pivot $\vec{p}_h = (0, 0)$, or to fix the shape of the object, for example to be a circumference of radius r , $\vec{c}_b(s) = (r \cos(s), r \sin(s))$.

Initial and endpoint constraints We often want to constrain the hand or object to start from or reach a configuration:

$$\vec{p}_h(t_0|t_f) = \vec{k}_1 \text{ and/or } \theta_h(t_0|t_f) = k_1 \quad (8)$$

$$\vec{p}_b(t_0|t_f) = \vec{k}_2 \text{ and/or } \theta_b(t_0|t_f) = k_2 \quad (9)$$

to start from rest:

$$\dot{\vec{p}}_b(t_0) = \vec{0} \text{ and } \dot{\theta}_b(t_0) = \dot{s}_b(t_0) = 0 \quad (10)$$

$$\dot{\vec{p}}_h(t_0) = \vec{0} \text{ and } \dot{\theta}_h(t_0) = \dot{s}_h(t_0) = 0 \quad (11)$$

or to (additionally) start from a static equilibrium:

$$\ddot{\vec{p}}_b(t_0) = \vec{0} \text{ and } \ddot{\theta}_b(t_0) = \ddot{s}_b(t_0) = 0 \quad (12)$$

$$\ddot{\vec{p}}_h(t_0) = \vec{0} \text{ and } \ddot{\theta}_h(t_0) = \ddot{s}_h(t_0) = 0 \quad (13)$$

Implicit motion constraints In some cases, constraints only implicitly affect the decision variables. The most frequent use is to constrain the object or hand to move along a path, rather than a trajectory. These are formulated as general implicit nonlinear constraints:

$$F(\vec{p}_b, \vec{p}_h) = 0 \quad (14)$$

Regularization constraints Occasionally, we incorporate extra constraints to guide the solver to find or avoid a particular type of solution. The two most frequently used regularization constraints are fixing the x component of the hand shape to a given function:

$$c_{h_x}(s) = k(s) \quad (15)$$

and constraining each point of the hand shape to a line that varies with s :

$$k_1(s)c_{h_x}(s) + k_2(s)c_{h_y}(s) = k_3(s) \quad (16)$$

We often use (15) to enforce $c_{h_y} = f(c_{h_x})$ i.e. the hand shape passes the vertical line test. Similarly, (16) is used to enforce $|\vec{c}_h| = f(\angle \vec{c}_h)$. Both are effective at preventing self-intersecting hand shapes, which are undesirable.

Fitness metric Often, we want to optimize a behavior with respect to a performance metric, rather than satisfy a particular constraint. These become part of the cost function of the optimization problem. Two performance metrics we will study in this paper are throwing distance and travel time.

V. DISCRETIZATION

The shape and motion of the system are functions of space and time. The formulation in Section IV is continuous, but for optimization purposes, we use a discrete representation. We want a representation that supports smoothness C^2 and that is sparse, i.e., each decision variable has a limited domain of influence both in space and time.

To do so, we describe shape and motion as linear combinations of shifted basis functions, as illustrated in Figure 4. Let $\Phi(x)$ be any given decision variable, dependent on x , which could be either time t or space s . Then we construct:

$$\Phi(x) = \sum_{i=1}^N \alpha_i E(L(x) - i) \quad (17)$$

where:

- E is a cubic B-spline basis function with uniformly spaced knot points, which is twice differentiable (smoothness) and only non-zero in the interval $(-2, 2)$ (sparsity).

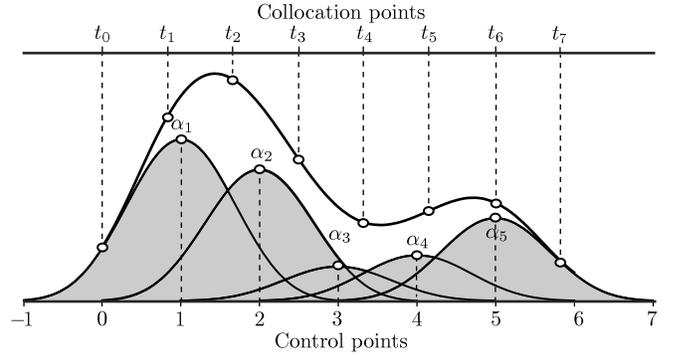


Fig. 4. Spline construction of the continuous decision variable $\Phi(x)$ with the discretized parameters $\alpha_1 \dots \alpha_N$, with $N = 5$, and with $M + 1 = 8$ collocation points where we will impose all constraints. Note that the domain of each basis is $(-2, 2)$ which gives sparsity to the spline representation.

- $\alpha_1 \dots \alpha_N$ are coefficients that weight the basis functions. Note that these become the discrete decision variables that discretize the continuous decision variable Φ .
- $L(\cdot)$ is a factor that non-dimensionalizes x as appropriate. For shape variables, s is already dimensionless, so $L(s) = s$. For motion variables, we transform time as $L(t) = \gamma \frac{t}{\tau}$, where τ is a global time constant that determines the duration of motion ($\frac{1}{\tau}$ is a decision variable in the optimization program, allowing trajectories of varying length), and γ is a normalizing constant that ensures each component of the trajectory has the same nominal duration regardless of the number of basis functions used in its representation (which allows varying resolution).

The use of basis functions allows us to compute in closed form the derivatives of the decision variables, which are necessary for many of the constraints described in Section IV:

$$\dot{\Phi}(x) = \sum_{i=1}^N \alpha_i \dot{E}(L(x) - i) \dot{L}(x) \quad (18)$$

$$\ddot{\Phi}(x) = \sum_{i=1}^N \alpha_i \ddot{E}(L(x) - i) \dot{L}(x)^2 \quad (19)$$

where we used that $\ddot{L}(x) = 0$.

Ideally, the constraints of motion would hold true at all times. However, for resolution purposes, we impose the motion constraints at $M + 1$ evenly distributed points along the trajectory, playing the role of *collocation* points in trajectory optimization [8]. In particular, each of the continuous motion constraints in a problem $G(t, \alpha_1, \dots, \alpha_N, \frac{1}{\tau})$ is imposed as $M + 1$ constraints in the optimization program $G_j(t_j = \frac{j}{M}\tau, \alpha_1, \dots, \alpha_N, \frac{1}{\tau})$ for $j = 0 \dots M$.

We extend this discretization to a periodic domain to represent closed shapes and periodic motions. The number of collocation points and basis functions we use varies with the problem and number of constraints. Our specific use of collocation drops any guarantees of integration exactness in favor of ease of implementation. However, we plan to explore more exact collocation methods in the future.

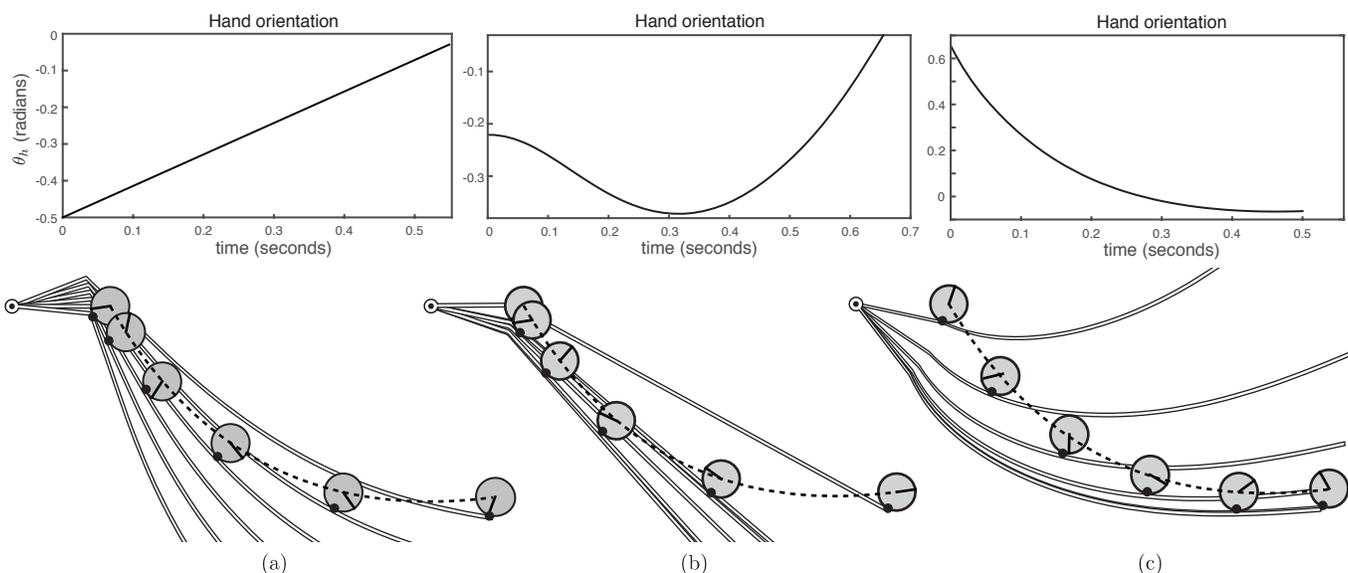


Fig. 5. Toy problem of moving a ball under gravity along a given path, drawn with a dotted line. There are infinite solutions in the shape-motion nullspace. Here we show two interesting cases: (a) Solution when the hand is forced to moved along a fixed trajectory, e.g., constant angular velocity, and only the hand shape is a design freedom. (b) Solution when the hand shape is fixed, e.g., a straight line, and only the hand trajectory is a design freedom. Note that the ball follows exactly the same path in both cases, although at different velocities. (c) Solution when both hand shape and trajectories are design freedoms. Note that in this case we can impose the trajectory of the ball, not just its path, illustrated by the fact that the ball moves at a constant speed along the curved path.

VI. ILLUSTRATIVE TOY PROBLEM

In this section we describe the method to optimize shape and motion with the simple problem of moving a ball under gravity along a desired path \vec{p}_b , with a rotational paddle.

We start by exploring these two problems: (Prob. 1) For a given fixed hand motion, is there a *hand shape* that forces the ball to travel along the desired path? and (Prob. 2) For a given fixed hand shape, is there a *hand motion* that forces the ball to travel along the desired path?

We can formulate both problems as a shape-motion pair that satisfies the following constraints:

- **Kinematic:** contact, tangency and rolling.
- **Dynamic:** inertia and friction.
- **Fixed decision variables:**
 - The object is a ball $\vec{c}_b(s) = (l \cos(s), l \sin(s))$.
 - The hand pivots about the origin: $\vec{p}_h(t) = \vec{0}$.
 - (Prob. 1) Fixed hand motion $\theta_h(t) = k \cdot t$, for example to constant velocity.
 - (Prob. 2) Fixed hand shape $\vec{c}_h(s)$, for example, to a straight line.
- **Task:** The ball moves along a desired path given as a level set $G(\vec{p}_b(t)) = 0$. In this example we use a parabola.

Figure 5a and Figure 5b show the outcome of the optimization. Both solutions satisfy all constraints and succeed in transporting the ball along the desired path, while rolling under the effect of gravity on the moving hand. This is an illustrative example of the nullspace that exists in the shape-motion design space. It is ultimately the combination of both that produces the desired object manipulation, but in many cases we can reproduce the effect of a motion in a shape, as well as the effect of a shape in a motion.

Note in the previous examples that the ball traverses the path at different speeds. Freeing both shape and motion in the optimization problem, and exploiting their nullspace, give us enough design freedom to control the trajectory $\vec{p}_b(t)$ along which the ball will move, not just its path. The problem has a very similar formulation to the previous one, but we instead replace the task constraint with a stricter constraint on the ball motion $\vec{p}_b(t) = \vec{p}_b^*(t)$, and remove the constraints on the shape or motion of the hand. The solution, illustrated in Figure 5c, succeeds in moving the ball along the desired path, but now, for example, with constant speed.

VII. CLASSICAL PROBLEMS

In this section, we formulate three classical problems with known solutions to validate the proposed optimization approach.

A. Rolling Brachistochrone

A brachistochrone curve is the path that allows an object to travel from A to B in the shortest amount of time, when starting from rest at A and accelerated by gravity \vec{g} . A classical result in mechanics is that when the object is a frictionless bead, the brachistochrone is a section of a cycloid [23]. Rodgers [18] showed that in the case of a rolling disk the brachistochrone is also a cycloid.

We represent the problem of the rolling brachistochrone with the proposed framework, where the path is a non-moving hand and the object is a disk, whose rolling trajectory down the hand $\vec{p}_b(t)$ becomes the brachistochrone when it achieves minimum travel time. We impose the following constraints:

- **Kinematic:** contact, tangency and rolling.
- **Dynamic:** inertia. Since the disk rolls without slipping (infinite friction), we omit the friction cone constraint.

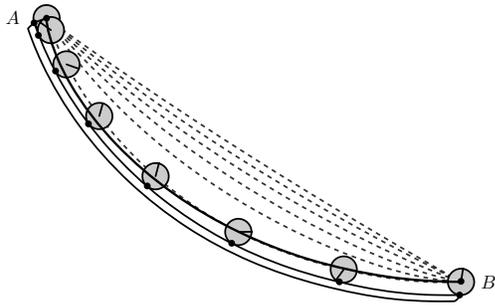


Fig. 6. Brachistochrone for a rolling cylinder. The analytical solution of the trajectory of the center of the cylinder converges (dotted line) to a cycloid (continuous line), when reducing the time to traverse from A to B .

- **Fixed decision variables:**

- The object is a disk $\vec{c}_b(s) = (l \cos(s), l \sin(s))$.
- The hand is static $\vec{p}_h(t) = \vec{0}$ and $\theta_h(t) = 0$.

- **Initial and endpoint constraints:**

- The disk starts at rest:
 $\vec{p}_b(t_0) = \vec{0}$, $\dot{\theta}_b(t_0) = \dot{s}_b(t_0) = 0$.
- The disk starts at $\vec{p}_b(t_0) = A$.
- The disk ends at $\vec{p}_b(t_f) = B$.

- **Task:** The duration of the trajectory is $\tau = T$.

We can formulate the problem by adding a cost on T to the objective function. From our experience, this is subject to local minima and very sensitive when approaching the real minimum time. In practice, we make the convergence more robust by stepping on T outside of the optimizer. The process starts with T equal to the time it takes for the roller to traverse a straight line from A to B , and gradually ask the program to find paths with smaller and smaller values of T until a solution cannot be found. Figure 6 shows the sequence of solutions converging to a cycloid.

B. Involute Gears

Gear design is a classic shape design problem. A pair of gears should mesh while maintaining a time-invariant gear ratio. The fundamental law of gearing [7] states that these two properties are equivalent to constraining the *line of action* to pass through the *pitch point* at all times. Figure 7 illustrates these concepts.

Involute gears, i.e., gears with teeth shaped as involute curves, are a popular solution that satisfy the above properties. One unique (and useful) property of the meshing between involute gears is that the line of action is constant throughout contact. As a consequence, the *pressure angle* α , which determines the amount of power that can be transmitted through the gear train, is constant throughout the meshing.

We now formulate the problem of finding gear shapes that satisfy the above properties with the proposed optimization approach. More concretely: for a given center distance between gears l , gear ratio r , and pressure angle α , find gear teeth that mesh adequately. We will indeed recover involute gears. In this case both hand and object are a pair of meshing gear teeth, and we impose the following constraints:

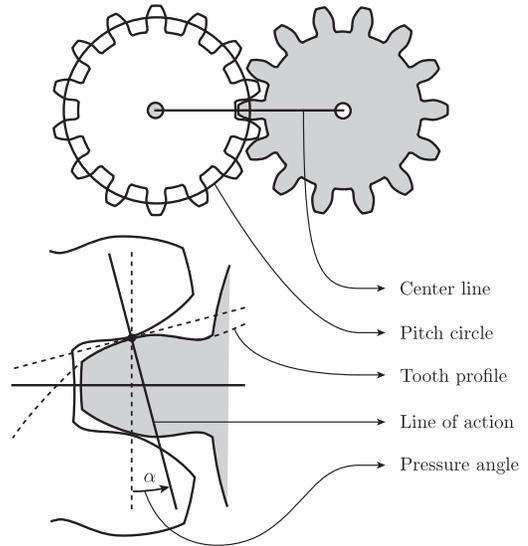


Fig. 7. Anatomy of a gear. The interaction between gears is largely determined by their tooth profile. The *center line* is the line connecting the two rotation centers, the *line of action* is orthogonal to the contact tangent between the two gears, hence is the direction along which force is transferred from one gear to another. The *pressure angle*, complementary to the angle between the line of action and the center line, is key in the design of gears.

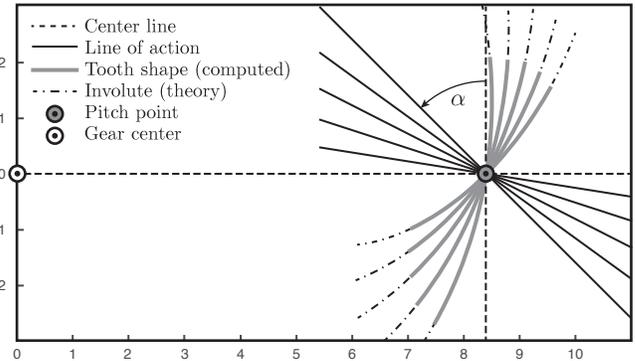


Fig. 8. Gear tooth profiles obtained for five different pressure angles α . The figure shows the corresponding line of action for each desired pressure angle (orthogonal to the gear tooth profile at the pinch point) and the recovered gear tooth profiles. These correspond very accurately to involute curves, known to provide constant pressure angle.

- **Kinematic:** contact, tangency and rolling.

- **Fixed decision variables:**

- The driving gear rotates about the origin $\vec{p}_h(t) = \vec{0}$.
- The driven gear rotates about the point $\vec{p}_b(t) = (l, 0)$.

- **Task:**

- Constant gear ratio r . We achieve this by fixing the trajectory of the gears $\theta_h(t) = \omega t$, $\theta_b(t) = -r\omega t$.
- Constant pressure angle α . That is:
 $\theta_h + \angle \vec{v}_h(s_h) = \alpha + \pi \pmod{2\pi}$

Note that in this case we do not impose dynamic constraints, since the meshing between gears can be seen as a purely kinematic/geometric problem. Another distinction from other problems, is that in this case we are looking for both the shape of hand and object (both gears). It should also be noted that we

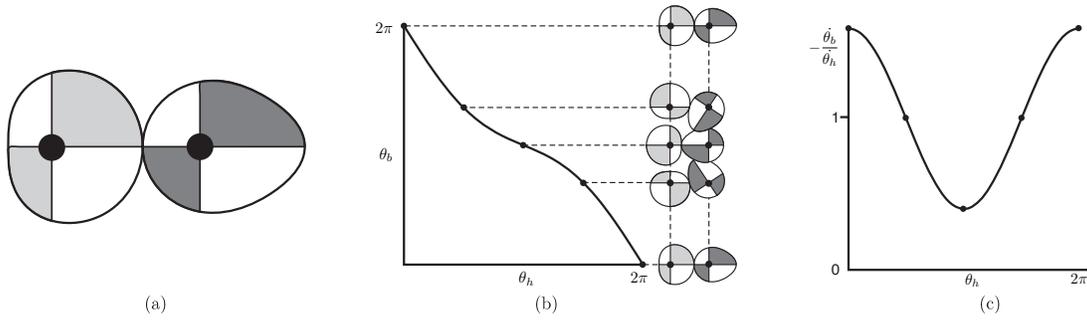


Fig. 9. (a) Pair of pitch curves with a desired transmission profile (b), and transmission ratio $h(\theta_1) = \frac{\dot{\theta}_2}{\dot{\theta}_1}$ (c).

make use of the property that meshing gears are approximately in rolling contact near the pitch point.

Figure 8 shows the obtained shapes corresponding to the section of a single gear tooth for pressure angles $\alpha = (.5, .6, .7, .8, .9) \cdot \frac{\pi}{2}$ and gear ratio $r = 1.5$. These curves can then be assembled into entire gear profiles. The resulting profiles match exactly with the expected analytical result corresponding to involute gears, also depicted in Figure 8.

C. Pitch Curve of non-Circular Gears

Every planar gear is characterized by a *pitch curve*, an imaginary smooth curve that defines its perimeter. The pitch curves of two meshing gears are in rolling contact as the gears rotate. For a circular gear, the pitch curve is a circle.

A classic problem in noncircular gear design is that of finding a pair of pitch curves $R_1(\theta_1), R_2(\theta_2)$ for meshing gears with a given transfer function $h(\theta_1) = \frac{\dot{\theta}_2}{\dot{\theta}_1}$, and center distance l . The typical approach [24] is to limit the problem to pitch lines that contact along the center line, in which case:

$$R_1(\theta_1) + R_2(\theta_2(\theta_1)) = l, \quad h(\theta_1) = \frac{R_1'(\theta_1)}{R_2'(\theta_2(\theta_1))}$$

due to contact and rolling constraints. The solution to these equations is then:

$$R_1(\theta_1) = \frac{Lh(\theta_1)}{1+h(\theta_1)}, \quad R_2(\theta_2(\theta_1)) = \frac{L}{1+h(\theta_1)}$$

Alternatively, we formulate the problem with the proposed framework, and, if desired, remove the above limitation. In this case, the hand and object shapes are the pitch curves. If we want to design a pair of pitch curves with transfer function $h(\cdot)$, the solution should satisfy:

- **Kinematic:** contact, tangency and rolling.
- **Fixed decision variables:**
 - The driving gear rotates about the origin $\vec{p}_h(t) = \vec{0}$.
 - The driven gear rotates about the point $\vec{p}_b(t) = (l, 0)$.
 - The orientations of both gears are $\theta_h(t) = \omega t$ and $\theta_b(t) = -H(\omega t)$, where $H'(\theta) = h(\theta)$.
- **Periodic boundary constraints:** We impose that the contact point resets after a full rotation of the gear $s_h(0) \equiv s_h(t_f) \bmod N_h$ and $s_b(0) \equiv s_b(t_f) \bmod N_b$.

Figure 9 shows an example with transfer function $h(\theta) = 1 + \frac{1}{1.707} \cos(\theta)$ (the same as an example in [10]). The resulting pitch curves align closely with the analytical result.

VIII. DYNAMIC THROWING

Inspired by the jai alai cesta in Figure 1, we set to design and implement a planar 1DOF thrower. Optimal throwing, in the context of a rotating paddle, requires an agreement between shape and trajectory. To do so, we look for a combination of shape and throw trajectory that maximizes the distance travelled by the ball before hitting the ground. To avoid degenerate solutions, we impose the following ad-hoc constraints: 1) The angular acceleration of the hand is bounded by $\bar{\alpha}$, 2) the ball starts at rest at a given location A , 3) the ball is released within radius l of the origin, and 4) we constrain to rolling interactions.

We define a problem that optimizes the distance the ball travels before hitting the ground with the following constraints:

- **Kinematic:** contact, tangency and rolling.
- **Dynamic:** inertia and friction.
- **Fixed decision variables:**
 - The object is a ball $\vec{c}_b(s) = (l \cos(s), l \sin(s))$.
 - The hand pivots about the origin: $\vec{p}_h(t) = \vec{0}$.
 - Bounded angular acceleration: $|\ddot{\theta}_h(t)| \leq \bar{\alpha}$.
- **Initial and endpoint constraints:**
 - The system starts at static equilibrium as in (10)-(13).
 - The ball starts at $\vec{p}_b(t_0) = A$.
 - The ball leaves the hand within a radius $|\vec{p}_b(t_f)| = l$.

Overhand throw: The result, as seen in Figure 10 is a hand shape that is concave up near the center, and concave down near the tip.

Underhand throw: The resulting shape, seen in Figure 11, is concave down near the center, and concave up near the tip.

Experiments The experimental setup consists of a motor attached to a rigid base. The computed hand shapes for underhand and overhand throwing are fused into a two-ended lasercut hand profile. The motor controller (Galil DMC 4020) has position/velocity tracking functionality, allowing the system to execute the computed trajectories. The resulting motion was captured with a high speed camera. We also use a Vicon motion tracking system to measure $\vec{p}_b(t)$. It should be noted that there is no feedback and the initial positions were set manually. The optimization and experimental results are similar as seen in Figure 10 and Figure 11.

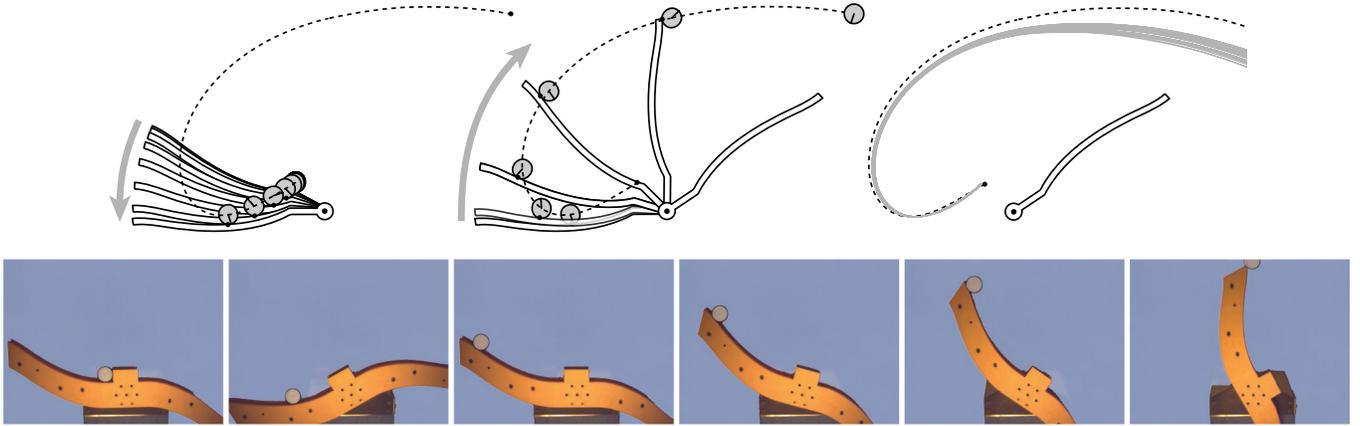


Fig. 10. The optimal overhand throw obtained is composed of two phases: a first gentle inclination where gravity accelerates the ball, followed by a fast upward stroke. The stream of pictures shows a real throw and the right most figure shows the ball trajectory spread over 30 throws.

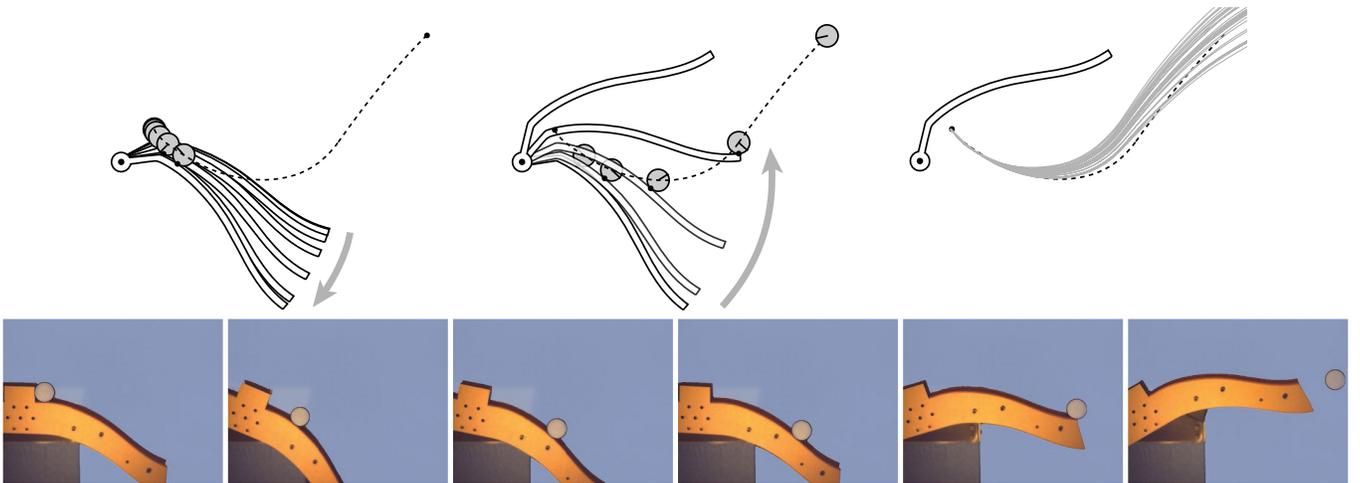


Fig. 11. The optimal underhand throw we obtain is composed of two phases: a first gentle inclination where gravity accelerates the ball, followed by a fast upward stroke. The stream of pictures shows a real throw and the right most figure shows the ball trajectory spread over 30 throws.

IX. DISCUSSION

Solver Our implementation uses SNOPT [6] for solving the presented nonlinear programs.

Multiple contact modes This work assumes rolling/sticking contact, and does not currently permit other contact modes (sliding, impact etc.). This is a limitation we plan to address in the future by exploring complementarity formulations.

Fitness objective vs. constraint satisfaction Posing some design objectives (i.e. travel time) as an optimization cost often results in local minima. We fix this by instead imposing the design objective as a constraint that iteratively increases or decreases. In the present implementation, this requires human supervision to determine the sequence of constraint values.

Problem precision and solution sensitivity We use intuition and trial and error to determine the number of control points and collocation points, which is not satisfying. Some regions of the solution require higher resolutions than others, and high resolution discretizations require more accurate initial guesses for solution convergence. The solution is also sensitive to the

number of collocation and control points, the relative scaling of costs vs. constraints, and the initial guess. A potential solution is to use multiscale optimization techniques that automatically increase resolution where necessary.

Shape regularization We need shape regularization for two reasons. First, the shape function has many extra degrees of freedom. For instance, $\vec{c}_h(s)$ and $\vec{c}_h(f(s))$ describe the same contour for all monotonically increasing functions f . This could be resolved if s corresponded to arc-length, however this is difficult to implement in practice. Without regularization, the control points representing shape tend to spread out unevenly, resulting in poor solutions. Second, any sufficiently general representation of shape is capable of self-intersections. Our formulation, which relies on local constraints, cannot prevent self-intersections, which are a global feature.

Though the regularization constraints shown in (15) and (16) are effective at addressing these issues, they require human supervision. We plan to explore alternative shape representations that reduce dimensionality, and to develop heuristics for finding more natural regularization constraints of a given task.

REFERENCES

- [1] Aaron Becker and Timothy Bretl. Approximate steering of a plate-ball system under bounded model perturbation using ensemble control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5353–5359, 2012.
- [2] M. Brokowski, M. Peshkin, and Kenneth Goldberg. Optimal curved fences for part alignment on a belt. *Transactions-America Society of Mechanical Engineers Journal of Mechanical Design*, 117:27–35, 1995.
- [3] Michael Caine. The design of shape interactions using motion constraints. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 366–371, 1994.
- [4] Po-Ting Chen. *Simulation and optimization of a two-wheeled, ball-flinging robot*. Masters thesis, University of California, San Diego, 2010.
- [5] Ana Cristescu, Bogdan Cristescu, and Andrei Laurenia. Generalization of Multispeed Gear Pitch Curves Design. *Applied Mechanics and Materials*, 659:559–564, 2014.
- [6] Philip E. Gill, Walter Murray, and Michael A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM review*, 47(1):99–131, 2005.
- [7] V. G. A. Goss. Application of analytical geometry to the form of gear teeth. *Resonance*, 18(9):817–831, 2013.
- [8] Charles R. Hargraves and Stephen W. Paris. Direct trajectory optimization using nonlinear programming and collocation. *Journal of Guidance, Control, and Dynamics*, 10(4):338–342, 1987.
- [9] Vincenzo Lippiello, Fabio Ruggiero, and Bruno Siciliano. The effect of shapes in input-state linearization for stabilization of nonprehensile planar rolling dynamic manipulation. *IEEE Robotics and Automation Letters*, 1(1):492–499, 2016.
- [10] Faydor L. Litvin, Alfonso Fuentes-Aznar, Ignacio Gonzalez-Perez, and Kenichi Hayasaka. *Noncircular Gears: Design and Generation*. Cambridge University Press, New York, New York, 2009.
- [11] Jen-Yu Liu and Yen-Chuan Chen. A design for the pitch curve of noncircular gears with function generation. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, volume 2, 2008.
- [12] Kevin M. Lynch and Matthew T. Mason. Dynamic nonprehensile manipulation: Controllability, planning, and experiments. *The International Journal of Robotics Research*, 18(1):64–92, 1999.
- [13] Kevin M. Lynch, Naoji Shiroma, Hirohiko Arai, and Kazuo Tanie. The roles of shape and motion in dynamic manipulation: The butterfly example. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1958–1963, 1998.
- [14] David J. Montana. The kinematics of contact and grasp. *The International Journal of Robotics Research*, 7(3):17–32, 1988.
- [15] Michael Posa, Cecilia Cantu, and Russ Tedrake. A direct method for trajectory optimization of rigid bodies through contact. *The International Journal of Robotics Research*, 33(1):69–81, 2014.
- [16] Philipp Reist and Raffaello D’Andrea. Bouncing an unconstrained ball in three dimensions with a blind juggling robot. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1774–1781, 2009.
- [17] Philipp Reist and Raffaello D’Andrea. Design of the pendulum juggler. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5154–5159, 2011.
- [18] Eric Rodgers. Brachistochrone and tautochrone curves for rolling bodies. *American Journal of Physics*, 14(4):249–252, 1946.
- [19] Alberto Rodriguez. *Shape for Contact*. Phd thesis, CMU-RI-TR-13-21, Carnegie Mellon University, 2013.
- [20] Alberto Rodriguez and Matthew T. Mason. Grasp Invariance. *The International Journal of Robotics Research*, 31(2):237–249, 2012.
- [21] Alberto Rodriguez and Matthew T. Mason. Effector Form Design for 1DOF Planar Actuation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 349–356, 2013.
- [22] Ji-Chul Ryu, Fabio Ruggiero, and Kevin M. Lynch. Control of nonprehensile rolling manipulation: Balancing a disk on a disk. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3232–3237, 2012.
- [23] Hector J. Sussmann and Jan C. Willems. 300 years of optimal control: from the brachistochrone to the maximum principle. *IEEE Control Systems Magazine*, 17(3):32–44, 1997.
- [24] Daniel CH Yang and Shih-Hsi Tong. Generation of identical noncircular pitch curves. *Journal of mechanical design*, 120:337–341, 1998.
- [25] I. Zarbski and T. Saaciski. Designing of non-circular gears. *Archive of Mechanical Engineering*, 55(3):275–292, 2008.