

Toward Obstacle Avoidance in Intermittent Dynamical Environments

Robert R. Burridge, Alfred A. Rizzi, and Daniel E. Koditschek

Department of Electrical Engineering and Computer Science
The University of Michigan, Ann Arbor, Michigan, USA

Abstract

In this paper we discuss a robotic task requiring *dynamical safety* in the face of an intermittent environment. We define and offer examples of this notion. We then construct a dynamically safe composite controller from dynamically safe constituents, and present empirical evidence of its effectiveness.

1. Introduction

This paper and a companion paper [1] develop an approach to building up complexes of controllers from simpler constituents. These ideas, first set out in [2], arise from our previous experience with dynamically dexterous robot tasks. The composite controllers we develop result in more varied and capable closed loop robot behavior than can be accomplished by any one of the constituent controllers acting in isolation. The companion paper [1] presents a method for extending the domain of attraction of a goal set via an appropriate composition procedure. This paper introduces the notion of *dynamical safety* for systems with only intermittent contact between robot and environment. In other words, we seek to add obstacle avoidance capability to robot controllers that can already successfully manipulate objects that must be periodically released into a dynamical environment along the way toward their goal. Although all the discussion in the companion paper is limited to the dynamical setting, we believe that our framework for controller composition is useful over a much wider class of robot task domains. In contrast, the novelty of the obstacle avoidance problem considered here is peculiar to the intermittent setting.

To introduce the new notion of a *safe* control strategy we find it useful to add to the formal definition (i) a specific extended example along with (ii) some empirical results reflecting our efforts to implement the example on our three degree of freedom Bühgler robot [3].

1.1. Intermittent Dynamically Dexterous Tasks

We are interested in constructing machines that can interact with an environment possessing significant dynamics, and that can accomplish tasks affording only intermittent influence over that environment. As in the past, we focus on robot juggling (ball bouncing) tasks as good representatives of this larger class which we believe to include such capabilities as catching, throwing, and hopping. Our past work has resulted in a family of machines [3, 4] that exhibit excellent dexterity in a narrow domain, as well as an immature but growing body of theory to explain how [5].

This paper extends the work reported in [1]. In particular, we wish to develop methods by which we can guarantee that constituent feedback based dynamical manipulation strategies are *safe* with respect to certain generalized obstacles. We show that these ideas will afford the automatic construction of complex combinations which are themselves also guaranteed to be *safe* with respect to the same obstacles.

1.2. Dynamical Obstacles

Our robot, the three degree of freedom Bühgler [6], is equipped with a (roughly) one meter long flat paddle and a stereo vision system. The task we wish to explore – the “batted pick and place” – consists of a ball being thrown into the workspace without warning with the requirement that the robot capture it (prevent it from escaping the workspace), bat it as necessary, then bring it to rest at a pre-specified location on the paddle.

The edge of the workspace defined by the end of the paddle provides a natural obstacle since once the ball passes that boundary, it will never return. To avoid this obstacle, we need to make certain not only that the ball isn't currently *at* the boundary, but also that future interactions between the robot and the ball will not cause it to *reach* the boundary before we can hit it again. This observation motivates our definition of *safety* in Section 2.

1.3. Robot Control Architecture

All of the experimental work described in this paper has been implemented on the Bühgler robot [6, 3] as elaborated in [1]. This machine senses ball positions using 2 CCD cameras located above and outside the workspace, and senses impacts using a microphone attached to the paddle. The raw data is filtered and integrated by an observer, producing a “continuous” stream of estimates of the ball's position and velocity, as reported in [6]¹. The estimate of the ball's state is fed through a nonlinear transformation, or “mirror law”, $M(b)$, to arrive at a desired reference trajectory for the robot, which is in turn passed to a smoothing “follow-through” generator. The resulting reference trajectory is tracked by a high performance adaptive inverse dynamics controller [7]. The signal flow through this sequence of processing steps from camera image plane all the way down to the joint

¹Although space limitations prevent an exhaustive list of modifications to the original system, we will note that both the window manager and dynamical observers have been significantly modified to allow balls to be thrown into the workspace rather than carefully presented, as had been the case heretofore.

level torque commands amounts to a feedback control policy that we shall refer to as Φ . Different choices of observers or mirror laws give rise to a characteristic robot response and in this sense there are many such Φ that might be brought to bear on any particular problem.

Throughout the work presented here, we presume that the state estimates emerging from the observer are correct (i.e. correctly converged to the true ball state), and that the robot will accurately track whatever reference trajectory we command. In fact we only require that the robot be “on track” whenever an impact occurs. Thus far our laboratory experiences have consistently supported these assumptions.

1.4. Sequential Composition of Controllers

As described in [1], our current program of research focuses on the use of multiple control strategies and switching schemes to carry out tasks that are impossible with a single continuous controller [8, 9].

For example, to carry out the “batted pick and place” task described in the companion paper we employ three control strategies – juggling (Φ_J), catching (Φ_C), and palming (Φ_P). Indeed, we foresee the need for several *deployments* of each of these strategies, each with different set points and gains. In order to organize these various strategies and deployments toward achieving the overall goal, we use a version of the backchaining algorithm originally described in [10] and extended in [1].

This algorithm uses knowledge of limit sets – sets of points, \mathcal{G}_Φ , in the phase space of the ball, to which a particular control policy, Φ , tends to bring the ball – and domains of attraction – sets of points, \mathcal{D}_Φ , from which Φ is guaranteed to deliver the ball to the associated limit set, \mathcal{G}_Φ – in order to partition the ball’s state space into regions where individual controllers should be active. The resulting partition of the state space provides the basis for a switching mechanism between the controllers which is guaranteed to drive the system to the goal. It turns out that if the component controllers are “safe” with respect to an obstacle, then the composite controller has the same property. While local estimates of the domains may be achieved by standard linear analysis, we wish to work with the larger global domain for which analytical estimates may not be available. This motivates the experimental determination of \mathcal{D}_Φ presented in the sequel.

2. Safe Behavior

2.1. Setting

Let $b \in \mathcal{B} \approx \mathbb{R}^6$ be the full state of the ball (position and velocity) in Cartesian coordinates. Let $r \in \mathcal{R} \approx \mathbb{R}^6$ be the state of the robot in joint space.

The ball in flight will be modeled by Newtonian dynamics with gravity pointing down along the z -axis. Due to the simplicity of the ball flight dynamics, we can derive a closed form expression for the ball position at time t in the future as a function of present state,

denoted

$$b(t) = F^t(b). \quad (1)$$

When the ball and paddle collide we use the standard restitution model of collisions (see Synge and Griffith [11] for a discussion of restitution models). In short, we assume that only the ball's velocity component normal to the paddle is affected, while neither the tangential component nor the velocity of the robot is altered by impact. Unless the ball and robot are in continuous contact, it is natural to divide the trajectory of the ball into epochs of time punctuated by collisions. The k^{th} epoch starts with the ball in state b_k , and ends immediately after the next impact, in state b_{k+1} .

Since the robot has no effect on the ball except at contact, we will ignore from now on the trajectory of the actuator system, and only consider the state of the robot at an impact event. Given that the robot strategies used here are feedback strategies (entirely based on the state of the ball), it follows that we can compute a return map of the form

$$b_{k+1} := f_{\Phi}(b_k) \quad (2)$$

for a particular robot strategy Φ .

We can now make formal the notions of limit sets and domains of attraction introduced in Section 1.4. Suppose there is an attracting set, \mathcal{G} , arising from iteration of f_{Φ} . We call \mathcal{G} the goal of Φ , and the domain of attraction of Φ to \mathcal{G} is given by

$$\mathcal{D}_{\Phi}(\mathcal{G}) = \{b \in \mathcal{B} \mid f_{\Phi}^{\infty}(b) \in \mathcal{G}\}. \quad (3)$$

2.2. Definition

To introduce the notion of safety we must first consider what an obstacle is. Let the set of all ball states which are unacceptable be denoted by \mathcal{O} . This set includes the positions where the ball is in a physical obstacle, as well as all states where the ball is beyond the cylinder of the workspace, or too fast to be viewed by, or outside the field of view of the current vision system.

The time until a ball, b , will hit this obstacle set is given by

$$\tau_{\mathcal{O}}(b) = \min t : F^t(b) \in \mathcal{O}, \quad (4)$$

while the time to next impact for a particular controller, Φ , is given by $\tau_{\Phi}(b)$.

We define a controller Φ to be *safe* if there can be found a set $\mathcal{D}_{\Phi S} \subseteq \mathcal{D}_{\Phi}$ such that

$$\mathcal{D}_{\Phi S} \subseteq \{b \in \mathcal{D}_{\Phi} \mid \tau_{\Phi}(b) < \tau_{\mathcal{O}}(b)\} \quad (5)$$

and

$$f_{\Phi}(\mathcal{D}_{\Phi S}) \subseteq \mathcal{D}_{\Phi S}. \quad (6)$$

Assuming that the robot is acting according to Φ , then the first criterion requires that the domain not contain any ball states from which the trajectory of the ball will lead to the obstacle before the next contact. The second criterion – invariance to impacts – implies that once in the safe domain the robot will never knock the ball out (again assuming that the robot continues in Φ). Together, these criteria insure that once a ball is in $\mathcal{D}_{\Phi,S}$, it will neither leave that set nor hit an obstacle as long as the robot remains in Φ .

2.3. Two Examples with Formal Demonstration

For the remainder of this paper we will only consider purely configuration space obstacles which are vertically aligned generalized cylinders. That is to say that the obstacle set \mathcal{O} can be completely characterized by its projection onto the x-y plane of the ball's phase space.

2.3.1. The Horizontally Regulated Juggle

Consider the juggling strategy, Φ_J , used in [6] and [1]. It is easily demonstrated that any ball which lies at the horizontal set-point with zero horizontal velocity is in fact “trapped” in a positive-invariant submanifold of the ball's state space [5]. If we consider this two dimensional set of ball states as $\mathcal{D}_{\Phi,S}$ it follows that this juggling strategy will be safe provided the horizontal set-point lies outside the obstacle set. First we note that since this set is invariant under the controller Φ_J it follows that (6) is trivially satisfied. Satisfaction of (5) follows from the fact that $\tau_{\mathcal{O}} = \infty$ for any ball in $\mathcal{D}_{\Phi,S}$ since, by presumption, no ball in this set has any horizontal velocity and thus can not reach \mathcal{O} (since \mathcal{O} is limited to being a vertically aligned cylinder).

2.4. Local Safety

We have empirically verified that the final equilibrium-point for the juggling strategy mentioned above is asymptotically stable. From this fact it follows that there exists a positive definite matrix, P , such that the set defined by

$$\mathcal{D}_{\Phi,L}(\epsilon) = \{b \in \mathcal{B} \mid \|f_{\Phi}(\bar{b})\|_P^2 < \|\bar{b}\|_P^2 < \epsilon\} \quad (7)$$

is nonempty for some $\epsilon > 0$ (here \bar{b} stands for $b - \mathcal{G}_{\Phi}$). Furthermore there must exist an ϵ^* such that $\mathcal{D}_{\Phi,L}(\epsilon^*) \subseteq \mathcal{D}_{\Phi}(\mathcal{G})$ and $\mathcal{D}_{\Phi,L}(\epsilon^*) \cap \mathcal{O} = \emptyset$. This set is by design a positive invariant set under f_{Φ} and thus (6) is immediately satisfied. Verification of (5) however is more difficult. First we note that no impact between the robot and machine can occur within the obstacle set if we start the ball in $\mathcal{D}_{\Phi,L}(\epsilon^*)$ – this is a direct result of the invariance of this set and the lack of an intersection between it and \mathcal{O} . However in order to conclude that the overall system is *safe* we must also ensure that during the ball's flight (between impacts) the boundary of the obstacle is not crossed. From our simple Newtonian model for flight we know that the projection of a ball's flight onto the x-y plane will be a straight line, and as a result of the convexity of the set $\mathcal{D}_{\Phi,L}(\epsilon^*)$ we can conclude that that the

entire flight of a ball remains “over” the projection of this set onto the x-y plane and thus can never intersect the boundary of the obstacle set.

3. Experimental Determination of Safe Domains

In order to determine the domain of attraction for our juggling controller, a “typical” set point, p_0 , was chosen and the juggle controller was activated. Two series of experiments were conducted, as outlined below.

3.1. Notation

For the purposes of this section it is convenient to consider apex points rather than the full ball state. Essentially, an apex point has all ball states except \dot{z} . Furthermore, we will express the horizontal component of these apex points in polar coordinates, rather than cartesian. Thus we have ϕ and r and their velocities instead of x and y and their velocities. In apex space, our nominal set point is $p_0 = (-0.3, 0.55, 4.5, 0, 0)$, where the first value is the polar angular position, the second is the radial distance from the origin, the third is the vertical energy, and the last two are the velocities $\dot{\phi}$ and \dot{r} (which will always be zero at juggle set points).

During ball flight, the apex point should remain constant, thus each epoch can be represented by a single point in apex space.

3.2. Experiments

In the first set of experiments, a small region around the fixed point, \mathcal{N}_{p_0} , was chosen based on the steady-state performance of the controller. Every time the apex point entered \mathcal{N}_{p_0} , the robot “kicked” the ball in a random direction and then attempted to recover by juggling. If it succeeded in regulating the ball again, then all of the points after the “kick” were considered “good”. If the juggler lost the ball after batting it at least once, then all the apex points between the perturbation and the loss were considered “bad”.

In the second set of experiments, the robot regulated the ball to a different set point then switched on the original juggle controller. If it successfully negotiated the switch, “good” points were generated and the second set point was moved further away and the process repeated. If not, “bad” points were generated.

All “good” apex points from both experiments were combined into a single set, and all “bad” apex points into another.

3.3. Presentation

Due to the nature of the experiments, as well as very fast vertical regulation, the data set is sparse for apex points with z outside a roughly 10cm band. Furthermore, examination of all the data reveals that the height of the apex had no significant effect, thus we are able to ignore z and project all the data onto the four-dimensional horizontal state space, $(\phi, r, \dot{\phi}, \dot{r})$.

The velocity space, $(\dot{\phi}, \dot{r})$, was divided into twenty-five bins, or sectors, and the sets of good and bad points were divided among them. For each velocity sector, the corresponding positions, (ϕ, r) , were plotted. In figure 3 we show one “spine” of these plots, with \dot{r} near zero, and $\dot{\phi}$ varying from negative to positive as we progress down the page. The center plot shows the positions for horizontal velocities nearest zero.

In figure 2 we show the plots of r against \dot{r} and ϕ against $\dot{\phi}$ for all the recorded apex points, with superimposed ellipses approximating the “safe” regions of each.

3.4. Discussion

In figure 2, as $\dot{\phi}$ goes negative, we see fewer bad points on the right, and those on the left begin to creep in. There are also more failures toward the end of the paddle. As $\dot{\phi}$ goes positive, we see the opposite effect, as the bad points on the right creep in toward the set point.

The bad points appearing within each of the ellipses in figure 2 represent points outside of the other ellipse. Thus if all the bad points outside the r - \dot{r} ellipse were removed from the ϕ - $\dot{\phi}$ plot, none would remain inside the ϕ - $\dot{\phi}$ ellipse.

4. Conclusion

4.1. Combining Safe Controllers

Having chosen a “safe” ellipsoid approximating the safe domain for the juggle controller, we next chose several such controllers with different set points to cover a large portion of the zero-velocity configuration space. Using the partition method described in [1] we combined them together to create a larger, more robust controller whose domain was the union of all the individual domains. Figure 1 shows a representative run of such a controller when the ball is introduced outside the domain of the goal controller.

4.2. Future Work

The methods employed to generate data to determine the boundaries of the domain for juggling do not generate a rich enough set. We are currently looking at other methods, including simulation and analysis, to help determine the juggling domain.

Nonetheless, the conservative approximation we were able to derive enabled us to successfully build a safe complex controller that successfully negotiated ball states that would have been lost by a single juggle algorithm.

References

- [1] Robert R. Burridge, Alfred A. Rizzi, and Daniel E. Koditschek. Toward a dynamical pick and place. In *IROS*, pages 2:292–297, August 1995.

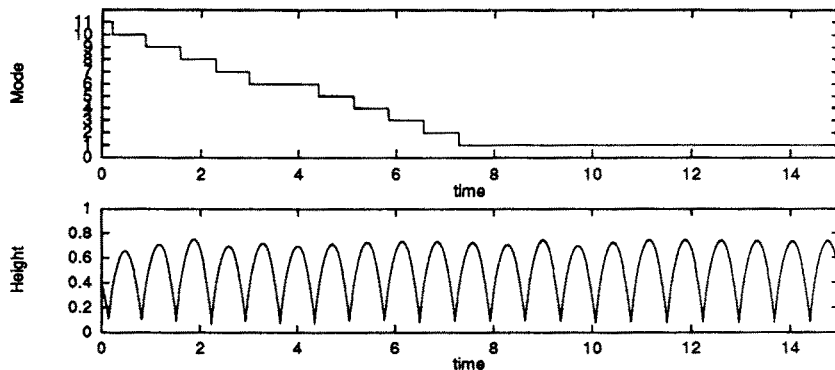


Figure 1. The current control mode indexed against time with the height of the ball plotted as a reference.

- [2] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek. Dynamical pick and place. Technical Report CSE-TR-235-95, University of Michigan, Ann Arbor, MI, 48105, April 1995.
- [3] Alfred A. Rizzi, Louis L. Whitcomb, and D. E. Koditschek. Distributed real-time control of a spatial robot juggler. *IEEE Computer*, 25(5), May 1992.
- [4] Martin Bühler. *Robotic Tasks with Intermittent Dynamics*. PhD thesis, Yale University, New Haven, CT, May 1990.
- [5] A. A. Rizzi and D. E. Koditschek. Further progress in robot juggling: Solvable mirror laws. In *Int. Conf. Rob. and Aut.*, pages 2935–2940, 1994.
- [6] Alfred A. Rizzi. *Dezterous Robot Manipulation*. PhD thesis, Yale University, 1994.
- [7] Louis L. Whitcomb, Alfred Rizzi, and Daniel E. Koditschek. Comparative experiments with a new adaptive controller for robot arms. *IEEE Transactions on Robotics and Automation*, 9(1):59–70, 1993.
- [8] Roger W. Brockett. Asymptotic stability and feedback stabilization. In Roger W. Brockett, Richard S. Millman, and Hector J. Sussman, editors, *Differential Geometric Control Theory*, chapter 3, pages 181–191. Birkhäuser, 1983.
- [9] Daniel E. Koditschek. Assembly: Another source of nonholonomy in robotics. In *Proc. American Control Conference*, pages 1627–1632, Boston, MA, June 1991. American Society of Control Engineers.
- [10] Tomás Lozano-Perez, Matthew T. Mason, and Russell H. Taylor. Automatic synthesis of fine-motion strategies for robots. *The International Journal of Robotics Research*, 3(1):3–23, 1984.
- [11] J. L. Synge and B. A. Griffith. *Principles of Mechanics*. McGraw Hill, London, 1959.

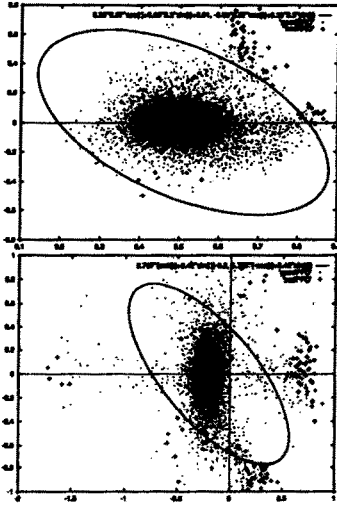


Figure 2. r plotted against \dot{r} , and ϕ plotted against $\dot{\phi}$ for all the data points, with superimposed ellipses approximating the safe domains.

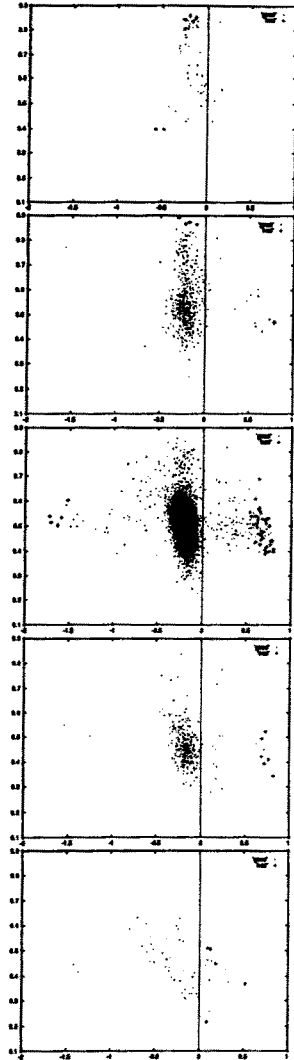


Figure 3. In all plots, $-0.1 < \dot{r} < 0.1$. As the plots proceed down the page, $\dot{\phi}$ moves between the intervals bounded by $-1.0, -0.5, -0.2, 0.2, 0.5,$ and 1.0 .