3-2002

Phase Regulation of Decentralized Cyclic Robotic Systems

E. Klavins
California Institute of Technology

Daniel E. Koditschek
University of Pennsylvania, kod@seas.upenn.edu

Recommended Citation

E. Klavins and Daniel E. Koditschek, "Phase Regulation of Decentralized Cyclic Robotic Systems", *International Journal of Robotics Research* 21(3), 257-275. March 2002. http://dx.doi.org/10.1177/027836402320556430

Postprint version. Published in *International Journal of Robotics Research*, Volume 21, Issue 3, March 2002, pages 257-275. Publisher URL: http://ijr.sagepub.com/content/21/3/257.abstract.

Phase Regulation of Decentralized Cyclic Robotic Systems

Abstract

We address the problem of coupling cyclic robotic tasks to produce a specified coordinated behavior. Such coordination tasks are common in robotics, appearing in applications like walking, hopping, running, juggling and factory automation. In this paper we introduce a general methodology for designing controllers for such settings. We introduce a class of dynamical systems defined over n-dimensional tori (the cross product of n oscillator phases) that serve as reference fields for the specified task. These dynamical systems represent the ideal flow and phase couplings of the various cyclic tasks to be coordinated. In particular, given a specification of the desired connections between oscillating subsystems, we synthesize an appropriate reference field and show how to determine whether the specification is realized by the field. In the simplest case that the oscillating components admit a continuous control authority, they are made to track the phases of the corresponding components of the reference field. We further demonstrate that reference fields can be applied to the control of intermittent contact systems, specifically to the task of juggling balls with a paddle and to the task of synchronizing hopping robots.

For more information: Kod*Lab

Disciplines

Electrical and Computer Engineering | Engineering | Systems Engineering

Comments

Postprint version. Published in *International Journal of Robotics Research*, Volume 21, Issue 3, March 2002, pages 257-275. Publisher URL: http://ijr.sagepub.com/content/21/3/257.abstract.

E. Klavins
Computer Science Department
California Institute of Technology
Pasadena, CA 91125, USA
klavins@caltech.edu

D. E. Koditschek
Department of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, MI 48109, USA
kod@umich.edu

Phase Regulation of Decentralized Cyclic Robotic Systems

Abstract

We address the problem of coupling cyclic robotic tasks to produce a specified coordinated behavior. Such coordination tasks are common in robotics, appearing in applications like walking, hopping, running, juggling and factory automation. In this paper we introduce a general methodology for designing controllers for such settings. We introduce a class of dynamical systems defined over n*-dimensional tori (the cross product of* n *oscillator phases) that serve as reference fields for the specified task. These dynamical systems represent the ideal flow and phase couplings of the various cyclic tasks to be coordinated. In particular, given a specification of the desired connections between oscillating subsystems, we synthesize an appropriate reference field and show how to determine whether the specification is realized by the field. In the simplest case that the oscillating components admit a continuous control authority, they are made to track the phases of the corresponding components of the reference field. We further demonstrate that reference fields can be applied to the control of intermittent contact systems, specifically to the task of juggling balls with a paddle and to the task of synchronizing hopping robots.*

KEY WORDS—coupled oscillators, hopping robots, juggling robots, behavior composition

1. Introduction

Many tasks in robotics require that cyclic actions be coordinated. A walking or running robot must coordinate its legs, which repeatedly convert motor torque into forward thrust, according to some desired gait—a schedule for the delivery of leg thrusts. Thus, the legs must deliver their thrusts according to some rhythm: whether simultaneously; in alternating groups of three; or some other pattern. If each leg is supposed to have some degree of independence—for example, to deal with terrain variations local to it—then, in the absence of some coordination mechanism, the phase relationships between the legs will surely destabilize. In this paper we address the problem of modifying the controller of an individual cyclic subsystem (e.g., a leg, or motor subunits corresponding to some modular constituent of an ensemble) so that it uses information obtained from other subsystems (e.g., the other legs) in such a manner that the entire system (e.g., the robot) stabilizes around a desired phase relationship (e.g., a gait).

In more formal terms, the paper addresses the problem of composing, “in parallel,” limit cycles. Namely, we assume we are presented with a number of independent controlled systems that each, separately, exhibit a limit cycle—an attracting periodic orbit. We seek a procedure for yoking together their controllers in such a fashion that the resulting coupled closed loop system exhibits a single limit cycle corresponding to a specified relationship among the original decoupled systems. The procedure should ideally be formal—relying simply on the presence of the individual attractors, independent of the details of the mechanisms that stabilize the individual subsystems. It should also be correct—accompanied by a proof that the coupled system does indeed exhibit the specified periodic attractor.

Our goal is to introduce tools that rely on feedback, introduce a minimum number of communications links among components and depend as little as possible on any central control element. Our hypothesis is that these tools are best thought of compositionally (Klavins and Koditschek 2000; Klavins 2000; Klavins, Koditschek, and Ghrist 2000)—that is, that components or groups of components that behave correctly in isolation may be composed by altering their control algorithms in a formulaic fashion to achieve coordination. The

The International Journal of Robotics Research
Vol. 21, No. 3, March 2002, pp. 257-275,

resulting modularity should give rise to scalability, enabling the design of large, complex nonlinear and robust controlled systems.

1.1. Motivation

Formal techniques for building decentralized control architectures with provable properties will become increasingly important as robots and, more generally, physically embedded computing systems, become more and more complex (Tennenhouse 2000). A simple, three degree of freedom juggling robot (Rizzi, Whitcomb, and Koditschek 1992), for example, may contain several processors which acquire sensory data, compute estimates and deliver control commands. A complex automated factory may require the coordination of hundreds of robots each performing processing, sensing and actuating locally (Rizzi, Gowdy, and Hollis 1999; Park, Tilbury, and Khargonekar 1999). If MEMS devices for distributed manipulation (Donald et al. 1999) can be used for assembly then possibly thousands or millions of microscopic actuators will require coordination. Ideally, to design controllers for such systems, we desire a provably correct method for controlling each component, using local information as well as information from neighboring components, so that some global task is performed by the entire collection of components. Similar goals have been realized, to some extent, in distributed computing (Lynch 1996). However, in physically embedded computing systems, any control methodology, distributed or otherwise, must account for real-world mechanical forces, inaccurate sensing and a partially actuated environment. Addressing these latter complications is the main challenge to decentralized, modular control.

We are convinced that compositional methods of this kind are possible. Previous work by the second author and collaborators has led to several functioning (laboratory) robots that possess relatively sophisticated dexterous behaviors formed by coupling previously working simpler constituents in an analogous manner. For example, robot "batters" have been constructed in compositions involving a one degree of freedom "vertical" limit cycle controller inspired by Raibert's hoppers (Koditschek and Bühler 1991). A planar version of the batter is composed of the Raibert vertical cycle with a one degree of freedom (Bühler, Koditschek, and Kindlmann 1990) "horizontal" fixed point controller that might be loosely construed as a nonlinear P-D (that is, a variant on the traditional proportional-derivative design (Koditschek 1989)). A subsequent spatial version is composed of the same Raibert vertical cycle with a two degree of freedom horizontal generalized P-D (Rizzi, Whitcomb, and Koditschek 1992).

This approach to the parallel composition of cycles with point attractors can claim some degree of generality. For example, the controller for a functioning two degree of freedom brachiating robot (Nakanishi, Fukuda, and Koditschek 2000) may be interpreted as the parallel composition of a repetitively swung simple pendulum with a nonlinear P-D. Work in progress suggests that similar parallel compositions of a repetitively swung pogo stick with a nonlinear P-D may yield effective controllers for multi-jointed (Saranli, Schwind, and Koditschek 1998) and even multi-legged (Altendorfer et al. 2000) running. These last examples, moreover, begin to hint at a hypothetical explanation of how running animals exhibit ground reaction forces characteristic of pogo sticks (Full and Koditschek 1999). We stress the qualification "some degree" of generalization, since no formal stability proofs have yet been carried through for these follow-on examples of cycle-with-point attractor compositions. For the original batting examples the necessary correctness proofs have been developed: there is indeed an attracting invariant "vertical" submanifold whose restriction dynamics exhibits the form of the Raibert vertical hopper (Rizzi and Koditschek 1994). But for none of these examples has a general and formal coupling procedure been articulated; a compositional "recipe" common to the batters, the brachiator and the runners is not yet evident.

Several desirable consequences quickly follow when a composition technique for embedded controllers is defined. The "sequential composition" of point attractors enjoys a formal and completely general control theoretic semantics via graphs that locate attractors relative to their neighbor's basins (as delimited, for example, by their Lyapunov functions) (Burridge, Rizzi, and Koditschek 1999). Although this notion is conceptually simple, it is nevertheless quite useful: a well established tradition of sequential composition in AI—preimage backchaining (Lozano-Perez and Mason 1984)—can be rendered formally and correctly in this manner. Formal composition methods can also ensure scalability. For example, the first author has developed methods for composing hybrid factory robot programs into concurrent, multi-robot systems (Klavins and Koditschek 2000). Based on this method, a provably correct compiler (Klavins, Koditschek, and Ghrist 2000) has been built for (a simplified, simulated version of) a modular factory (Rizzi, Gowdy, and Hollis 1999).

In contrast to the compositions just cited, the parallel composition of cycles, the focus of this paper, remains a less developed area. We have some experience with successful empirical efforts of this kind. The batters, above, have been joined to produce "jugglers"—again, both planar (Bühler, Koditschek, and Kindlmann 1994) and spatial (Rizzi, Whitcomb, and Koditschek 1992) versions—by properly "interleaving" the two constituent vertical cycles. However, no correctness proof has heretofore existed, much less a clear report of any formal coupling principles suitable for applications in other domains. It is toward such a formal coupling principle that this paper is addressed.

1.2. Specific Contributions

Parallel compositions of the sort described above, involving point attractors or points and one cyclic attractor, satisfy the following steady state behavior: the attractor of the coupled system is the cross product of the component attractors. In contrast, the obvious first complication arising from parallel constructions of several cyclic attractors is that their cross product is not directly related to any of the components, defining, instead, an n-torus, which does not in itself correspond to any natural robotic task that we have encountered. A composition of limit cycles resulting in a limit cycle, therefore, must also specify the limiting *phase relationships* desired of the components. The central contribution of the paper is to propose a simple but general formalism for specifying these phase relationships along with a general method for realizing the specification with a reference dynamical system. We then introduce a number of example problems to suggest how these reference dynamics might eventually be the basis of a formal composition technique for a large class of physically relevant cyclic systems.

We begin in Section 2 with the observation that any cyclic component system can be described in terms of *phase coordinates* consisting of phase and phase velocity, thus giving each system the same "interface" and introducing the model space of the composition—the n-torus. We then propose a simple means of specifying the desired phase relationships with a *connection graph*, whose nodes denote the component cyclic systems and whose edges, labeled "in phase" or "out of phase," denote their desired phase differences at steady state. Next, we present a general procedure for building a gradient-like vector field on the model space whose coupling functions are governed by the connection graph. Finally, we identify algebraically the entire forward limit set of the closed loop coupled system, affording, thereby, a check that the desired limit cycle is essentially globally asymptotically stable.

We use the reference dynamics to compose several one degree of freedom cyclic components in three progressively more complicated examples evocative of the physically functioning juggling and running robots introduced above. These examples are chosen to suggest the range of control affordance that a more general composition procedure must encompass. It is well established in control theory that differences in actuation structure present radically different opportunities for point stabilization, ranging from arbitrarily achievable dynamics (e.g., in the case of fully actuated first order systems), to systems that are smoothly unstabilizable (e.g., in the case of nonholonomically constrained mechanisms (Brockett 1983)). The corresponding stabilizability classification for limit cycles is far less established, hence we select examples that correspond to physical settings of interest. The examples illustrate how the same abstract connection graph can serve as the basis for compositions of cyclic systems of various types, each type corresponding to an intuitively different affordance over the subsystems' phases.

In our first example we consider highly abstracted components, each taking the form of a first order fully actuated subsystem—in phase coordinates, $\dot{\theta}_i = f_i(\theta_i, v_i), i = 1, \ldots, n$, where f_i is invertible with respect to the control, v_i, for each *phase*, $\theta_i \in S^1$. Since appropriate "inverse dynamics" style state feedback, $v_i = f_i^{-1}(\theta_i, u_i)$, can arbitrarily reshape the vector field, $\dot{\theta}_i = u_i$, it is clear that the phase θ of each component of the system is directly controllable. Such models are not uncommon in the coupled oscillator literature, and, for the purposes of this paper, offer a simple setting in which to illustrate the application of the proposed composition technique. Specifically, in Section 2.4 we construct coupling controllers arising from two different connection graphs (Figure 1) for a set of six first order cyclic components. We use the algebraic characterization of the forward limit set on the cross product space,

$$T^6 = \underbrace{S^1 \times S^1 \times \ldots \times S^1}_{6 \text{ times}}$$

to discover that the first connection graph is effective while the second is not.

In general, physical systems will not offer such direct affordance over phase velocity. For example, in action-angle coordinates (Arnold 1991), (E_i, θ_i)—the classical representation of phase for the one degree of freedom mechanical systems of present interest—a lossless closed loop component exhibits the dynamics $\dot{\theta}_i = \omega_i(E_i)$ and $\dot{E}_i = 0$. Physically realizable control inputs act on the acceleration variable of such a system and typically enter both the action and the angle dynamics in a complicated manner,

$$\begin{aligned} \dot{\theta}_i &= \omega_i(E_i, u_i) \\ \dot{E}_i &= f_i(\theta_i, E_i, u_i). \end{aligned}$$

However, many cyclic component systems of interest in robotics—in particular, all of those mentioned in the motivation section above—are regulated by means of periodically applied changes in "stiffness" or some similar physical variable that adjust the action on a cycle-by-cycle basis because they can be actuated only intermittently. The term intermittent means that during certain periods, there is no affordance at all. Thus, $f_i \equiv 0$, and $\partial\omega_i/\partial u_i \equiv 0$ except when θ_i is contained in some *contact set* $\mathcal{A}$. When such a subsystem operates in isolation, it is sufficient to verify that the "stiffening schedule" stabilizes a specified periodic orbit with respect to any convenient coordinate system. However, when conceived as a component, the subsystem's stiffening schedule on the contact set must be altered with the dual purpose of regulating its own limit cycle as well as its phase relative to its specified neighbors.

In Section 3 we introduce two different examples of this situation, evocative, respectively, of the batting (Bühler, Koditschek, and Kindlmann 1994) and hopping robots (Koditschek and Bühler 1991) discussed above. In both cases we begin by introducing variants of the component subsystem controllers that were deployed on the physical machines: they yield asymptotically stable limit cycles for the component one degree of freedom systems with user-specified steady state total energy. We next address the parallel composition of two such cyclic components, defining the identical reference dynamics (on the two-dimensional torus) as a model for juggling in the first example, and for bipedal hopping in the second. For both examples, we show how to compute the phase coordinate transformations required to implement the composition. The reference dynamics then serves to guide modifications in the subsystem controllers that confer sensitivity to relative phase in addition to the prior affordance over individual energy (now interpreted as phase velocity, through the action variable, in the new coordinate systems). We finally show, in both cases, that the resulting coupled systems do indeed achieve the desired closed loop behaviors.

Similar though they are, these two settings introduce a further distinction in subsystem actuation structure: the recourse to deadbeat as opposed to asymptotically stabilizing control. For the juggling example, because the contact set is a lower dimensional surface visited instantaneously each cycle, it is easy to build a subsystem controller that works in a "deadbeat" manner, achieving the desired ball height for an isolated single ball after one hit. In contrast, for the hopping example, the component controllers for the two isolated subsystems achieve asymptotic stability with convergence in infinite time. Despite this difference in input structure, the composition procedures are quite similar. A comparison of eqs. (29) and (31), which define the return maps of these systems near equilibrium, shows that the second system incorporates a delay term, g, which marks the difference (and complicates the analysis). That our method applies equally well to each example is a hint of its possible generality.

A second key difference between these two examples underscores an additional feature of the composition method whose proper consideration lies beyond the scope of the present paper. The gradient-like reference dynamics on the two-dimensional torus entails two invariant cycles—the desired (essentially global) 180°-out-of phase attractor and an in-phase repellor. The latter structure can be considered a dynamical "obstacle" to be avoided by the coupled system. In synchronized hopping, there is a dedicated actuator for each cyclic system (each hopper can control itself) , while in juggling, a single actuator must switch its attention from each ball in the system necessitating a sort of interleaving control. Separating the phase of the balls thus becomes an important part of the composition and is naturally encoded by the presence of the repellor.[1]

In summary, the present paper addresses a small, next step toward the general goal introduced at the beginning of the paper. We introduce a general formalism for specifying the steady state cyclic behavior of a collection of cyclic components. Treating the component subsystems as oscillators suggests how to compose them using feedback terms based on the phases of neighboring oscillators to achieve the specified coupled behavior. If the components of the system are continuously actuated (i.e., they have direct control over their phase), the method is straightforward. If, on the other hand, the components have intermittent control over their phases (as with a paddle bouncing a ball or a leg delivering thrust to a robot), applying our method is less obvious. Nevertheless, we demonstrate that it is applicable to this important class of problems. The similarity of method in the different examples strongly suggests that a general formalism of the kind we desire should be possible to define and practice.

1.3. Related Work

1.3.1. Decentralized Control

By decentralized control of a system $\dot{\mathbf{x}} = f(\mathbf{x}, u)$ we mean first that $\mathbf{x}$ can an be broken into a number of subsystems $\mathbf{x} = \mathbf{x_1}; \ldots ; \mathbf{x_n}$ where the semicolon means vector concatenation. Second, we require that controllers can be found so that for each subsystem i we have that $\dot{\mathbf{x}}_\mathbf{i} = f_i(\mathbf{x}_{j_1}, \ldots , \mathbf{x}_{j_k}, u_i(\mathbf{x}_{j_1}, \ldots , \mathbf{x}_{j_k}))$ where the set $\{j_1, \ldots , j_k\}$ of neighbors of i is a proper subset of $\{1, \ldots , n\}$ (i.e., the control law for the ith system depends only on the neighbors of i). Various decentralized control schemes for legged locomotion (Beer et al. 1997; Calvitti and Beer 2000) are inspired by biological models of the stick insect (Cruse 1990) and other animals (Delcomyn 1980). In such schemes, each leg of a six-legged robot is considered to be animated by a separate processor and actuator, deciding what to do based on the state of certain neighboring legs. Of course, the state of the robot's mechanical body depends upon the positions and velocities of its center of mass frame as well as the states of all the legs and thus the system is decentralized only in the sense that the control of actuated components is decentralized. In Klavins et al. (2001) we present a simulation study suggesting that the methods in the present paper can exhibit behaviors similar to those observed in a simple model (Calvitti and Beer 2000) of this kind of coordination. We believe, but have not yet explicitly demonstrated, that the framework introduced here promotes a parsimonious view of the stick insect models, and one for which correctness of coordination proofs will be tractable. In the present paper we prove that similar coordination strategies are robust and stable, albeit in a vastly simpler setting. We believe such a parsimonious and sound foundation is a necessary precursor to the widespread adoption of any walking or coordination strategies.

Our general approach to building decentralized systems, inspired by the traditions of computer science (Abadi and Lamport 1993; Charpentier and Chandy 1999), is compositional (Burridge, Rizzi, and Koditschek 1999; Klavins and Koditschek 2000; Klavins 2000). That is, we seek methods for composing previously isolated subsystems into more complex systems via adjustments of their individual controllers as

1. We accomplish the change of attention of the paddle among the balls using an *attention function* (see eq. (25)), the details of which are beyond the scope of this paper but can be found in Klavins (2000).

we have already illustrated in our earlier remarks about "sequential" (Burridge, Rizzi, and Koditschek 1999), "parallel" (Bühler, Koditschek, and Kindlmann 1990), and "interleaved" (Rizzi, Whitcomb, and Koditschek 1992) control. There are many other informal uses of composition in robotics as well, such as the bottom up generation of flocking behaviors (Reif and Wang 1999), for which formal compositional treatments could be quite useful as well.

Our simple characterization of a "decentralized" system suffers from the same sort of ambiguity that afflicts some computer science characterizations of distributed systems. That is, in their simplest forms the available formalisms, differential equations and computational complexity, are blind to the issues of information flow and communication costs (of course, many of these problems in computer science have been addressed (Lynch 1996)). We do not address these problems here, although we believe they are of tremendous importance as systems become more complex. In Klavins et al. (2001) we begin to address the trade-off between centralized control structures, such as used in our hexapod robot (Saranli, Bueler, and Koditschek 2000), with decentralized control.

1.3.2. Coupled Oscillators

The study of the nervous system has inspired a significant volume of work on coupled oscillators. The goal has been to devise analytically tractable models of the neuron and of collections of neurons for the purpose of explaining observed neurophysiological phenomena. The typical approach is to treat neurons as oscillators of some kind and introduce various coupling terms (Guckenheimer and Holmes 1983). Researchers have examined various regular topologies of such couplings as well, as for example in Kopell (1995) and Ermentrout and Kopell (1994).

Oscillators appear in robotics whenever coordinated cyclic movements are required. As robotics researchers increasingly turn toward biology for inspiration, coupled oscillators are used synthetically, to engineer control algorithms, rather than as modeling tools. Generally one finds two apparently complementary approaches: feedforward shape generators and feedback driven networks of coupled oscillators. The first approach is inspired by the discovery of biological *central pattern generators*—oscillating groups of neurons that have been shown to produce rhythmic movements originating in the central nervous system (Pearson 1973; Cohen, Rossignol, and Grillner 1988; Cohen, Holmes, and Rand 1982) that seem associated with a feedforward style of control. In robotics, pattern generators are generally used to control repetitive movements. In the RHex hexapod robot (Saranli, Bueler, and Koditschek 2000), a simple, first order linear oscillator is used as a "clock" to generate an alternating tripod gait. Similar mechanisms are used to control the shape of snake like robots (Ostrowski, Desai, and Kumar 1998; Bloch et al. 1996) and in an underactuated two degree of freedom suspended leg (Lewis et al. 2000) to produce feedforward locomotion. In contrast, feedback methods, for example those inspired by the previously mentioned study of the slow-moving stick insect (Cruse 1990) which have begun to be analyzed (Calvitti and Beer 2000), couple internal oscillators with mechanical oscillators such as legs. The internal oscillators are then synchronized via some coupling function, yet are constantly disturbed by signals from the legs. In Hu (2000) a nonlinear oscillator model with feedback is used to coordinate leg motions in a biped.

In this work, synthesis of coordination control algorithms (rather than, say, modeling and analysis of complex biological systems) remains the primary goal, hence we take the view that oscillators should be abstracted to phase and phase velocity coordinates. Thus, instead of using a complex, nonlinear oscillator to generate motions we explicitly use simple gradient-like reference fields and concentrate on the means by which variously composed controllers may be introduced into variously configured control systems to produce coordinated movement in mechanical systems. A similarly synthetic view was used by, for example, Pratt and Nguyen (1995) to synchronize clocks in a mesh-connected network of processors.

2. Reference Fields: First Order Cyclic Systems

In this section we suppose that a task is represented as a limit cycle—an isolated invariant periodic attractor—on a torus of some dimension. For now, we examine model systems, whose subsystems are all first order actuated oscillators and suppose that a suitable computationally effective measure of *phase*—an abstraction to this model space from the state space of the robot system in question—can be found.[2] We also assume that all degrees of freedom are actuated. Later, in Section 3, we relax this assumption. We call the vector fields corresponding to the model dynamical systems we construct **Reference Fields**, roughly comparable to the target dynamics (Nakanishi, Fukuda, and Koditschek 2000) or templates (Full and Koditschek 1999) introduced in allied papers. We will use these fields to construct feedback controllers that attempt to make actual systems behave like the model systems we construct by *referring* to the value of the field at each point in configuration space.

Given a configuration space X with dynamics $\dot{\mathbf{x}} = F(\mathbf{x}, \mathbf{u})$, we define a **task** for this system to be a submanifold $M \subseteq X$ with a dynamical system of the form $\dot{\mathbf{y}} = G(\mathbf{y}) \in T_{\mathbf{y}}M$ defined over it. A control law $\mathbf{u} = \mathbf{g}(\mathbf{x})$ **performs** the task (M, G) if

1. M is an attractor (i.e., an asymptotically stable invariant set) and
2. the restriction dynamics on M is given by G.

In this paper, we wish to control n one-dimensional oscillators each of whose phase $\phi_i(t)$ can be thought of as a point

2. In Section 3.1, in which we consider intermittent contact systems, we actually show how to construct phase coordinates.

in the circle S^1. We take as a representation of the circle the interval $[0, 2\pi]$ with its end points identified. Thus, all values of ϕ_i are considered modulo 2π. We frequently consider smooth 2π-periodic functions of $\mathbb{R}$ and smooth functions on S^1 equivalently. Each oscillator also has a phase velocity $\dot{\phi}_i(t) \in \mathbb{R}$ which, as discussed in Section 1.2, we consider to be the control input for the oscillator. Thus, $\dot{\phi}_i = u_i$.

The configuration space of n such oscillators is the n-dimensional torus,

$$\mathbb{T}^n = \underbrace{S^1 \times \ldots \times S^1}_{n \text{ times}}.$$

The tasks over the torus that we consider in this paper are constant flows on subtori. They thus have the form $(\mathbb{T}^m, G)$ where $0 < m \le n$ and for $\theta \in \mathbb{T}^m$, $G(\theta) = \mathbf{a}$, $\mathbf{a} \in \mathbb{R}^m$. Since we are presently only interested in limit cycles, we take $m = 1$ so that $\mathbb{T}^m = S^1$. In Section 2.2, we show how to construct reference fields over $\mathbb{T}^2$ and in Section 2.3 we extend the idea to a particular task in n dimensions: the problem of coordinating the oscillators so that they exhibit particular phase relationships in a decentralized manner.

2.1. Specifying Phase Relationships

As mentioned in Section 1.2, composing some number of cyclic tasks requires a specification of the desired phase relationships between them. We also would like to specify the communication structure to be used by the system, by stipulating which other oscillators are neighbors of any particular oscillator. To proceed, we define a *connection graph* as follows. Define C to be an $n \times n$ symmetric matrix over the set $\{0, 1, -1\}$ where $C_{i,i} = 0$ for all i. We interpret C as follows. If $C_{i,j} = 1$, then it is desired that oscillators i and j be in phase: $\phi_i - \phi_j = 0 (\text{mod } 2\pi)$. If $C_{i,j} = -1$, then it is desired that i and j be out of phase: $\phi_i - \phi_j = \pi (\text{mod } 2\pi)$. If $C_{i,j} = 0$ then no phase difference is specified—although one may be implied transitively via other connections.[3]

We describe two examples. The first is a specification suitable as the basis for the control of a six legged robot. It consists of six oscillators, one for each leg, connected so that there are two disjoint, fully connected in-phase tripods and one out-of-phase connection between a representative from each tripod. The second is a specification which produces an unintended behavior—that is, one where the connection matrix does not give a system that performs the task associated with it over the entire domain of the system. In Section 2.4 we demonstrate how to check these systems using the results in Section 2.3.

Let $\phi_1, \ldots, \phi_6$ represent the phases of each of six oscillators and suppose that oscillators one through three correspond to the first tripod and four through six, the second. And suppose one and six are connected out of phase. The connection matrix that realizes this scheme, illustrated in Figure 1(a), is

$$C_{alt} = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & -1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ -1 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}. \tag{1}$$

Another seemingly natural way to specify an alternating tripod is to suppose that the six legs are arranged in a ring, with each out of phase with its neighbors (Figure 1(b)). It turns out that this specification cannot be realized with our method, as is shown in Section 2.4.

$$C_{bad} = \begin{pmatrix} 0 & -1 & 0 & 0 & 0 & -1 \\ -1 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 & -1 \\ -1 & 0 & 0 & 0 & -1 & 0 \end{pmatrix}. \tag{2}$$

2.2. Two Dimensions

To illustrate our idea, we consider the task of regulating two oscillators with phases ϕ_1 and ϕ_2 so that (1) the rate of change of each phase is some desired value and (2) the phases are maximally separated. This is specified by the connection matrix

$$C_2 = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}. \tag{3}$$

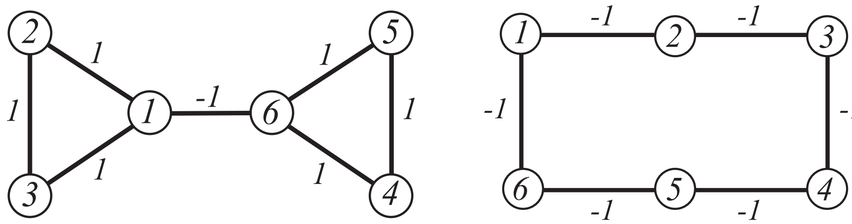
To define a reference system for this task, we proceed in two steps. First, we define a potential energy function V on the phase difference $\phi_1 - \phi_2$ which has a unique minimum at $\phi_2 = \phi_1 + \pi (\text{mod } 2\pi)$. Then we take the negative gradient ∇V and add a drift term $(1, 1)^T$ which suspends the gradient system in the two-dimensional torus. We define the potential energy function V over $\mathbb{T}^2$ by $V(\phi_1, \phi_2) = \cos(\phi_2 - \phi_1)$.

This function, shown in Figure 2(a), has the set $\{(\phi_1, \phi_2) \mid \phi_1 - \phi_2 = \pi (\text{mod } 2\pi)\}$ as its minimum. We next take the negative gradient and add a "drift" term, defining the reference field, to be

$$\mathcal{R}(\phi_1, \phi_2) = \kappa_1 \begin{pmatrix} 1 \\ 1 \end{pmatrix} - \kappa_2 \nabla V(\phi_1, \phi_2)\ . \tag{4}$$

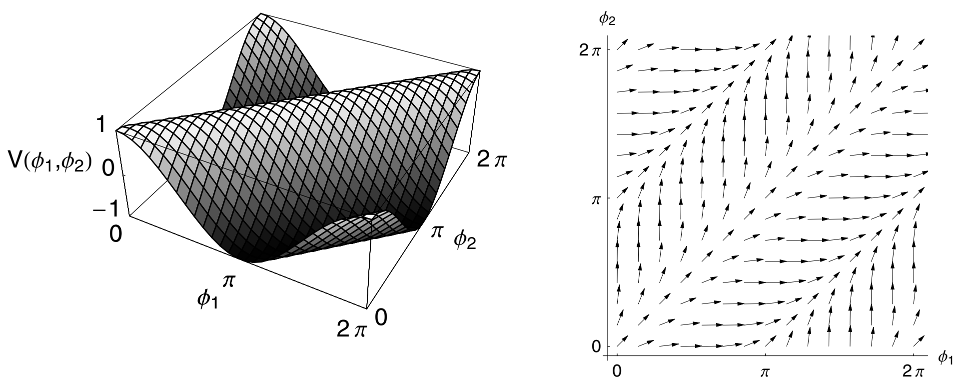
Here κ_2 is an adjustable gain which controls the rate of convergence to the limit cycle. The circles $\phi_2 = \phi_1$ and $\phi_2 = \phi_1 + \pi$ are equilibrium orbits (see Figure 2(b)). The first is unstable, the second is stable. Thus, this field performs the task specified by C_2. Speaking compositionally, we say that the two separate tasks $\dot{\phi}_1 = \kappa_1$ and $\dot{\phi}_2 = \kappa_1$ have been composed by adding to the system the coupling term $\kappa_2 \nabla V(\phi_1, \phi_2)$.

3. The restriction to 0 or π phase differences is not limiting: if we desire that $\phi_1 - \phi_2 = \alpha \mod 2\pi$, then we set $\phi_2' = \phi_2 + \alpha$ and require $\phi_1 - \phi_2' = 0 \mod 2\pi$ instead. For clarity of presentation, we avoid such coordinate changes.



(a) (b)

Fig. 1. Two different connection graphs. The system (10) obtained from (a) results in an "alternating tripod" behavior whereas the system (11) obtained from (b) does not.



(a) (b)

Fig. 2. The dimensional reference fields. (a) The artificial energy function V on the phase difference $\phi_1 - \phi_2$. (b) The reference field (4) obtained by suspending the negative gradient of V in the torus.

2.3. Multiple Oscillators and Arbitrary Connections

In this section we consider multiple oscillators and the further constraint that the reference fields we construct be in a certain sense decentralized. In particular, we build systems $\dot{\mathbf{x}} = \mathcal{R}(\mathbf{x})$ such that $\mathcal{R}_i$ depends only on a subset of the phases in the system—a set of "neighbors" that has been designated by the designer. We examine only systems whose task dynamics have $\dot{\phi}_i = a \in \mathbb{R}$ for all i and concentrate on the decentralized aspect of the problem. In particular, we propose the following method.

1. Identify the *connections* desired in the reference field. That is, specify with which neighbors the controller of a given oscillator must communicate.
2. Label the connections as either "in phase" or "out of phase."
3. Construct an energy function that respects these constraints.
4. Suspend (a generalized version of) the gradient of this function to complete the reference field and check that it meets the specifications given by the labeling.

The last step arises because the third step may not be able to realize the phase relationships specified in the second step while respecting the connections enforced in the first step. However, we supply an easy criterion to check this property.

DEFINITION 1. The **task specified** by a given connection matrix C is given by the set

$$M_C = \{(\phi_1, \dots, \phi_n) \mid \forall i, j \; C_{i,j} \neq 0 \Rightarrow \phi_i - \phi_j = \frac{\pi}{2}(1 - C_{i,j})\}$$

with the dynamics

$$(\dot{\phi}_1, \dots, \dot{\phi}_n) = \kappa_1(1, \dots, 1)\ .$$

We will give a reference field $\mathcal{R}$ based on C that performs this task in some instances, depending on the structure of C. As we have stated, however, we supply an easy criterion to check that a given reference actually performs the specified task.

From C we define a reference field $\mathcal{R}_C(\phi_1, \ldots, \phi_n)$ by setting

$$\dot{\phi}_i = \kappa_1 - \kappa_2 \sum_{j=1}^{n} C_{i,j} \sin(\phi_i - \phi_j), \tag{5}$$

where κ_1 and κ_2 are constant gains. The term $\sin(\phi_i - \phi_j)$ is similar to a simpler error term $\phi_i - \phi_j$, except it is continuous modulo 2π, simplifying the analysis. We will show that $\mathcal{R}_C$ arises from the suspension of a gradient-like system on the subtorus defined on the differences between the oscillators. In particular, let $V_{i,j} = -C_{i,j} \cos(\phi_i - \phi_j)$ and set V to be the energy function

$$V = \sum_{i<j} V_{i,j} \ . \tag{6}$$

To simplify our analysis, we will transform eq. (5) to a more convenient form. To this end we introduce the following notation and definitions.

DEFINITION 2. For each $n > 1$ we define the $\frac{n(n-1)}{2} \times n$ dimensional **difference matrix** L_n recursively as follows. First, set $L_2 = (1, -1)$. Then, for each $n > 2$ set

$$L_n = \left(\begin{array}{c|c} \mathbf{1} & -I_{n-1} \\ \hline \mathbf{0} & L_{n-1} \end{array} \right),$$

where I_n is the $n \times n$ identity matrix, the upper left hand part of the matrix is an $(n-1) \times 1$ vector of ones and the lower left hand part of the matrix is an $\frac{(n-1)(n-2)}{2} \times 1$ vector of zeros.

Let $\mathbf{x} = (\phi_1, \ldots, \phi_n)^T$. We will first define the system $\mathbf{y} = L_n \mathbf{x}$ obtained from the phases of the oscillators to be all possible differences between phases, corresponding to the connections in the graph. We express the reference field (eq. (5)) in this notation.

DEFINITION 3. Suppose the $n \times n$ dimensional connection matrix C is given. Let $\tilde{C}$ be the $\frac{n(n-1)}{2} \times \frac{n(n-1)}{2}$ dimensional diagonal matrix whose diagonal is $(C_{1,2}, C_{1,3}, \ldots, C_{n-1,n})$. Also, let $\mathbf{1} = (1, \ldots, 1)^T$ be a vector of n ones. Then the **reference field** associated with C is

$$\mathcal{R}_C(\mathbf{x}) = \kappa_1 \mathbf{1} - \kappa_2 L_n^T \tilde{C} s(L_n \mathbf{x}), \tag{7}$$

where $\mathbf{x} = (\phi_1, \ldots, \phi_n)^T$ and $s(\mathbf{y}) = (\sin(y_1), \ldots, \sin(y_{n(n-1)/2}))^T$.

The reference field R_C induces a vector field on the connections $\mathbf{y} = L_n \mathbf{x}$ given by

$$\begin{aligned} \dot{\mathbf{y}} &= G(\mathbf{y}) \triangleq L_n \dot{\mathbf{x}} = \kappa_1 L_n \mathbf{1} - \kappa_2 L_n L_n^T \tilde{C} s(\mathbf{y}) \\ &= -\kappa_2 L_n L_n^T \tilde{C} s(\mathbf{y}), \end{aligned} \tag{8}$$

since $L_n \mathbf{1} = 0$.

To understand the dynamics of eq. (8), and therefore of $\mathcal{R}_C$, we show that they are gradient-like. That is, its equilibrium states are the minima of some energy like-function. A function $U : X \to \mathbb{R}$ is a *LaSalle Function* (Lasalle 1961) (a generalized Lyapunov function) for the vector field $\dot{x} = F(x)$ if its image is compact and if $\dot{U}(x) \triangleq DU \cdot F(x) \leq 0$. We have

LEMMA 1. The energy function $V : \mathbb{R}^{\frac{n(n-1)}{2}} \to \mathbb{R}$ defined by eq. (6) is a LaSalle function for G.

Proof. Since $V(\mathbf{y}) = -\sum_{i<j} C_{i,j} \cos(y_{i,j})$, taking derivatives gives $DV = s(\mathbf{y})^T \tilde{C}$. Then

$$DV \cdot G(\mathbf{y}) = -\kappa_2 s(\mathbf{y})^T \tilde{C} L_n L_n^T \tilde{C} s(\mathbf{y}) = -\kappa_2 ||L_n^T \tilde{C} s(\mathbf{y})|| \leq 0,$$

since $\tilde{C}$ is symmetric. Finally, the image of V is contained in $[-\frac{n(n-1)}{2}, \frac{n(n-1)}{2}]$, proving the lemma. □

Let $\omega(Y)$ be the forward limit set of the set $Y \triangleq \mathbb{R}^{\frac{n(n-1)}{2}}$ under the dynamical system $\dot{\mathbf{y}} = G(\mathbf{y})$. Our first result is then:

THEOREM 1. The forward limit set $\omega(Y)$ of G is equal to $\{\mathbf{y} \mid G(\mathbf{y}) = 0\}$, the zeros of G.

This is the criterion for checking that the specifications given by C are met by R_C—we will discuss this after we prove the theorem.

Proof (of Theorem 1)**.** We will first show that $L_n L_n^T \mathbf{v} = 0$ if and only if $L_n^T \mathbf{v} = 0$. The result then follows from set equalities. So suppose that $L_n L_n^T \mathbf{v} = 0$. Let $\mathbf{w} = L_n^T \mathbf{v}$. Then $L_n \mathbf{w} = 0$ implies that $w_1 = w_2, w_1 = w_3, \ldots, w_{n-1} = w_n$. Thus, $\mathbf{w} = \alpha \mathbf{1}$ for some $\alpha \in \mathbb{R}$. Also, taking the transpose of $L_n \mathbf{1} = 0$ gives $\mathbf{1}^T L_n^T = 0$ from which we can conclude that $\mathbf{1}^T L_n^T \mathbf{v} = 0$. This means that $\mathbf{1}^T \alpha \mathbf{1} = 0$ which implies that $\alpha = 0$. Thus, $L_n^T \mathbf{v} = 0$. We conclude that $L_n L_n^T \mathbf{v} = 0 \Leftrightarrow L_n^T \mathbf{v} = 0$.

Now we have

$$\begin{aligned} \omega(Y) &\subseteq \{\mathbf{y} \mid \dot{V}(\mathbf{y}) = 0\} \\ &= \{\mathbf{y} \mid s(\mathbf{y})^T \tilde{C} L_n L_n^T \tilde{C} s(\mathbf{y}) = 0\} \\ &= \{\mathbf{y} \mid L_n^T \tilde{C} s(\mathbf{y}) = 0\} \\ &= \{\mathbf{y} \mid L_n L_n^T \tilde{C} s(\mathbf{y}) = 0\} \\ &= \{\mathbf{y} \mid G(\mathbf{y}) = 0\} \ . \end{aligned}$$

The first inclusion is by Lemma 1 and LaSalle's Invariance Principle. The second to last equality is because $L_n L_n^T \mathbf{y} = 0 \Leftrightarrow L_n^T \mathbf{y} = 0$. Finally, by definition, $\{\mathbf{y} \mid G(\mathbf{y}) = 0\} \subseteq \omega(Y)$ and thus, $\{\mathbf{y} \mid G(\mathbf{y}) = 0\} = \omega(Y)$. □

Since the $\dot{\mathbf{y}}$ system is not full rank, we work with the smaller system obtained by taking the first $n-1$ elements of $\mathbf{y}$, to analyze particular systems. From these differences, all other differences may be defined as long as the graph associated with the matrix C is connected. To this end, we define a projection

$$P_n = \left(\begin{array}{c|c} I_{n-1} & 0 \end{array} \right),$$

where the right part is an $(n-1) \times \frac{(n-1)(n-2)}{2}$ dimensional matrix. Then set $\mathbf{z} \triangleq P_n \mathbf{y} = (\phi_1 - \phi_2, \phi_1 - \phi_3, \ldots, \phi_1 - \phi_n)^T$.

We will also need a pseudoinverse of P_n, namely

$$P_n^\dagger = \left(\frac{I_{n-1}}{L_{n-1}} \right).$$

Then the $\mathbf{z}$ system is

$$\dot{\mathbf{z}} = H(\mathbf{z}) \triangleq -\kappa_2 P_n \dot{\mathbf{y}} = -\kappa_2 P_n L_n L_n^T \tilde{C} s(P_n^\dagger \mathbf{z}). \qquad (9)$$

Note that $\tilde{V} \triangleq V \circ P_n^\dagger$ is a LaSalle function for this system.

From the dynamical system defined on the connections between the oscillators, we can deduce the behavior of the total system $\dot{\mathbf{x}} = \mathcal{R}_C(\mathbf{x})$. We have

COROLLARY 1. The limit set $\omega(\mathbb{T}^n)$ of the system $\dot{\mathbf{x}} = \mathcal{R}_C(\mathbf{x})$ is equal to the set $\{\mathbf{x} \mid H(P_n L_n \mathbf{x}) = 0\}$. The dynamical system restricted to this set is simply $\dot{x}_i = \kappa_1$ for each i. Furthermore, if $\mathbf{z}^*$ is a stable fixed point of H, then the set $\{\mathbf{x} \mid P_n L_n \mathbf{x} = \mathbf{z}^*\}$ is a stable orbit of $\mathcal{R}_C$.

Thus, to check that the reference field $\mathcal{R}_C$ performs the task specified by C, we check that the stable orbits of $\dot{\mathbf{x}} = \mathcal{R}_C(\mathbf{x})$, which are given by the stable fixed points of $\dot{\mathbf{z}} = H(\mathbf{z})$, correspond to the task.

2.4. Examples

In this section we apply the above criterion to the examples in Section 2.1.

2.4.1. Example 1: An Alternating Tripod

Start with the connect graph C_{alt} defined in eq. (1) and let $\mathbf{x} = (\phi_1, \ldots, \phi_6)^T$. The system is then

$$\dot{\mathbf{x}} = \kappa_1 \mathbf{1} - \kappa_2 L_6^T \tilde{C}_{alt} s(L_6 \mathbf{x}),$$

where $\tilde{C}_{alt}$ is defined as in Definition 3. To understand this system, we use Theorem 1 and examine the system in eq. (8), given in this case by

$$\dot{\mathbf{y}} = -L_6 L_6^T \tilde{C}_{alt} s(\mathbf{y}),$$

where $\mathbf{y} = L_6 \mathbf{x}$ (without loss of generality we let $\kappa_2 = 1$). Now take the projection $\mathbf{z} \triangleq P_6 \mathbf{y}$ to the first five elements of $\mathbf{y}$ to get the system

$$\dot{\mathbf{z}} = H(\mathbf{z}) \triangleq \qquad (10)$$

$$\begin{pmatrix} 2\sin(z_1) + \sin(z_1 - z_2) + \sin(z_2) - \sin(z_5) \\ \sin(z_1) - \sin(z_1 - z_2) + 2\sin(z_2) - \sin(z_5) \\ \sin(z_1) + \sin(z_2) + \sin(z_3 - z_4) + \sin(z_3 - z_5) - \sin(z_5) \\ \sin(z_1) + \sin(z_2) - \sin(z_3 - z_4) + \sin(z_4 - z_5) - \sin(z_5) \\ \sin(z_1) + \sin(z_2) - \sin(z_3 - z_4) - \sin(z_4 - z_5) - 2\sin(z_5) \end{pmatrix}.$$

Setting $\dot{\mathbf{z}} = 0$ and solving for $\mathbf{z}$ gives 72 fixed points. Straightforward calculation of the eigenvalues of the Jacobian of H at each of these points shows that only one fixed point, namely $(0, 0, \pi, \pi, \pi)$ is stable and the rest are unstable. Using Corollary 1, we conclude that the task performed by this system is given by

$$M_{alt} = \{(\phi_1, \ldots, \phi_6) \mid \phi_1 - \phi_2 = \phi_1 - \phi_3 = 0 \text{ and } \phi_1 - \phi_4 = \phi_1 - \phi_5 = \phi_1 - \phi_6 = \pi\}$$

with the task dynamics $\dot{\phi}_i = \kappa_1$, which is equivalent to the task specified by the connection matrix C_{alt}.

2.4.2. Example 2: An Unintended Behavior

Now consider the specification given by C_{bad} and defined by eq. (2). In this case, $\dot{\mathbf{z}} = -P_6 L_6 L_6^T \tilde{C}_{bad} s(P_6^\dagger \mathbf{z})$ is given by

$$\dot{\mathbf{z}} = H(\mathbf{z}) \triangleq \qquad (11)$$

$$\begin{pmatrix} \sin(z_1 - z_3) + \sin(z_4) + \sin(z_1 - z_5) + sin(z_5) \\ \sin(z_2 - z_3) + \sin(z_2 - z_4) + \sin(z_4) + \sin(z_5) \\ -\sin(z_1 - z_3) - \sin(z_2 - z_3) + \sin(z_4) + \sin(z_5) \\ -\sin(z_2 - z_4) + 2\sin(z_4) + \sin(z_5) \\ \sin(z_4) - \sin(z_1 - z_5) + 2\sin(z_5) \end{pmatrix}.$$

The desired fixed point of this equation is $(\pi, 0, \pi, 0, \pi)$ which is indeed stable. However, the fixed points $(\pm\frac{2\pi}{3}, \mp\frac{2\pi}{3}, 0, \pm\frac{2\pi}{3}, \mp\frac{4\pi}{3})$ are also stable fixed points. Thus, the system (eq. (11)) does not perform the task specified by C_{bad} because it has five distinct, stable limiting behaviors. The fixed points of $\dot{\mathbf{z}} = H(\mathbf{z})$ need not be hyperbolic. In such cases, simple linearization around the fixed point is not sufficient to determine its stability and other methods must be used.

3. Application to Intermittent Contact Systems

To demonstrate the relevance of reference dynamical systems to robot control, we show in this section a means of constructing controllers for a kind of intermittent control problem, that is, where there is at least one degree of freedom which may only be actuated under certain circumstances. For example, we consider in this section the task of bouncing a ball on a paddle (a kind of juggling), wherein the robot may only actuate the ball during collisions. The rest of the time the ball is under the influence of gravity alone. The other example we consider is the control of hopping robots where there is a flight phase. While on the ground, the robot has some affordance over its trajectory while in the air it does not. The problem we address is whether the assumption of continuous actuation, made in Section 2, can be relaxed, allowing us to use the cyclic reference fields we have defined as the basis of control algorithms for these systems.

In Section 3.1 we describe how to change from the body coordinates of these systems to phase coordinates. In Section 3.2 we describe the tasks of bouncing a single ball and hopping a single leg. Then we show how to juggle two balls with one paddle and how to synchronize two hopping robots, using as a basis the two-dimensional reference field (eq. (4)), and analyze the stability of each control algorithm. The analyses in this section assume a particular connection strategy (eq. (3)) and do not address the question of coupling n arbitrary intermittent contact oscillators. Furthermore the two systems considered require substantially different treatments (although we attempt to show their similarities). Presently, a formal treatment of arbitrary networks of arbitrary intermittent contact oscillators does not exist. However, simulations such as shown in Figure 8, suggest that many such stable couplings can be designed.

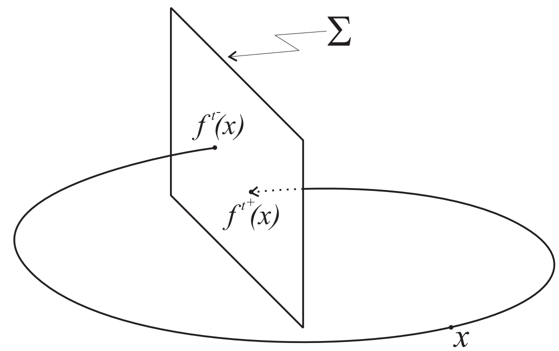


Fig. 3. The relationship between $t^-(x)$, $t^+(x)$ and x.

3.1. Phase Coordinates for Intermittent Systems

We first show how to obtain phase coordinates for intermittent cyclic dynamical systems, by which we mean systems for which a global cross section can be found. Let $f^t : \mathbb{R} \times X \to X$ be a flow on X. Formally, a global cross section Σ is a connected submanifold of X which transversely intersects every flowline. For any point $x \in X$, define the **time to return** of x to be

$$t^+(x) = \min\{t > 0 \mid f^t(x) \in \Sigma\} \tag{12}$$

and define the **time since return** of x to be

$$t^-(x) = \min\{t \geq 0 \mid f^{-t}(x) \in \Sigma\} . \tag{13}$$

The first return map, $\sigma : \Sigma \to \Sigma$, is the discrete, real valued map given by $\sigma(x) = f^{t^+}(x)$. Let $s(x) = t^+(x) + t^-(x)$. s is the time it takes the system starting at the point $f^{t^-}(x) \in \Sigma$ to reach Σ again. Now, define the **phase** of a point x by

$$\phi(x) = 2\pi \frac{t^-(x)}{s(x)} . \tag{14}$$

Notice that the rate of change of phase, $\dot{\phi}$, is equal to $2\pi/s$. The relationship of these functions to Σ is shown in Figure 3. Therefore, $\dot{\phi}$ is constant or piecewise constant, changing only when the state passes through Σ.

In the examples we give in Section 3.2, the function $h : X \to Y$, defined by $h(x, \dot{x}) = (\phi, \dot{\phi})$, is actually a change of coordinates where X is the space of position and velocity pairs and $Y = S^1 \times \mathbb{R}^+$. In juggling, we use the section $\Sigma \subseteq X$ defined by $x = 0$ which corresponds to the set of states where the robot may contact (and thereby actuate) the system. In hopping we use the section in X defined by $\dot{x} = 0$ and $x < 0$ which corresponds to the lowest point in a hop (see Sections 3.2.1 and 3.2.2 for the definitions of these systems). By construction, $h(\Sigma)$ will be given by the set $\mathcal{C} = \{(0, \dot{\phi}) \mid \dot{\phi} \in \mathbb{R}^+\}$ in both examples. In these intermittent control situations, it is only in $\mathcal{C}$ that $\dot{\phi}$ may be altered by the control input u. That is, we change $\dot{\phi}$ according to a control policy u to get the return map in phase coordinates $\sigma' : \mathcal{C} \to \mathcal{C}$ given by $\sigma'(0, \dot{\phi}) = (0, u(\dot{\phi}))$. We design the controller so that there is a unique and stable fixed point at some desired phase velocity $\dot{\phi}^* = \omega$.

We want to control the system so that the return map σ has a stable fixed point at some x^*. Whether or not $h^{-1}(0, \omega) = (0, \dot{x}^*)$ depends on the dimension of Σ. If $dim\ \Sigma = 1$, as it will be in the examples we supply, then the preimage of ω is indeed a point.

The main point of this section concerns the composition or interleaving of two such cyclic systems. That is, we suppose we have the system $(x_1, \dot{x}_1, x_2, \dot{x}_2) \in X^2$ with corresponding phase coordinates $(\phi_1, \dot{\phi}_1, \phi_2, \dot{\phi}_2) \in Y^2$. As before, system i may only be actuated when $\phi_i = 0$. In the examples we consider, we suppose that it is undesirable for the systems to be actuated simultaneously. Thus, the set of states where $\phi_1 = \phi_2 = 0$ should be repelling. One task that realizes these goals is (M, G) where

$$\begin{aligned} M &\triangleq \{(\phi_1, \dot{\phi}_1, \phi_2, \dot{\phi}_2) \mid \phi_1 = \phi_2 + \pi (\text{mod } 2\pi)\} \\ G(\phi_1, \phi_2) &\triangleq (\dot{\phi}_1, \dot{\phi}_2) = (\omega, \omega) . \end{aligned} \tag{15}$$

The constraint $\phi_1 = \phi_2 + \pi (\text{mod } 2\pi)$ encodes our desire to have the pair of phases as far from the situation $\phi_1 = \phi_2 = 0$ as possible.

To analyze and control such a system, we restrict our attention to the sections $\Sigma_1 \subseteq Y^2$ and $\Sigma_2 \subseteq Y^2$ defined by $\phi_1 = 0$ and $\phi_2 = 0$, respectively. For now, suppose that the flow alternates between the two sections. Let $g^t = H \circ F^t \circ H^{-1}$ be the flow in Y^2 conjugate to the flow in X^2 where $F = (f, f)$ and $H = (h, h)$ and

$$\tau_i(w) = \min\{\tau > 0 \mid g^\tau(w) \in \Sigma_{3-i}\} .$$

Start with a point $w \in \Sigma_1$. Let $w' = g^{\tau_1}(w)$ and $w'' = g^{\tau_2}(w')$. We have $w' \in \Sigma_2$ and $w'' \in \Sigma_1$, so we have defined the return map on Σ_1. Now since g is parameterized by the control inputs u_1 and u_2 we get

$$\begin{aligned} w = (0, \dot{\phi}_1, \phi_2, \dot{\phi}_2) &\mapsto w' = (\Phi_1, u_1, 0, \dot{\phi}_2) \\ &\mapsto w'' = (0, u_1, \Phi_2, u_2) . \end{aligned}$$

Thus, the phase velocity updates $u_1(w)$ and $u_2(w')$ must be found so that eq. (15) is achieved.

In Section 3.4, we will show that simply choosing $u_1 = \mathcal{R}_1(\Phi_1, 0)$, the first component of a 1 : 1 reference field as in eq. (4), and $u_2 = \mathcal{R}_2(0, \Phi_2)$ for well chosen values of κ_1 and κ_2 results in a controlled system that realizes the task (Definition 1).

3.2. Two Examples of Intermittent Contact Tasks

In Section 3.2.1 we consider as an example of a cyclic, intermittent contact problem, the task of bouncing a ball vertically on a piston to a desired height. In this fairly simple task, the paddle can hit the ball at just the right velocity to achieve the desired height in one collision (assuming such actuation is within the torque limits of the paddle), as described in Bühler, Koditschek, and Kindlmann (1994). In Section 3.2.2 we describe a model of a one degree of freedom hopping robot inspired by Raibert's hopper (Raibert, Brown, and Chepponis 1984), similar to that studied in Koditschek and Bühler (1991). In this model, the robot hops on a spring loaded leg and has control over the stiffness of the spring. We assume that it may instantaneously change the stiffness of the spring just before the decompression phase, at the point of maximal compression, thereby roughly simulating the effect of a pneumatic piston. With the control algorithm we give for this task, the robot may only *approach* the desired hopping height asymptotically, so that the discussion at the end of Section 3.1 only approximately applies. Nevertheless, when regulating two such hoppers, the same control idea—to "sample" the reference field at the cross section—applies in this situation as well. In Section 3.3 we describe how to juggle two balls at once and how to synchronize two hoppers. Then in Section 3.4 we give analytical and numerical evidence that this method is correct.

3.2.1. Juggling

Consider a system wherein a paddle with position p controls a single ball with position x to bounce, repeatedly, to a prespecified apex. Suppose the paddle always strikes the ball at $p = x = 0$ and instantaneously changes its velocity according to the rule

$$\dot{x}_{new} = -\alpha\dot{x} + (1+\alpha)\dot{p}. \tag{16}$$

The constant α is the coefficient of restitution. We suppose that the velocity of the paddle is unchanged by collisions. Evidently, a paddle velocity of $\dot{p} = (\alpha-1)/(\alpha+1)\dot{x}$ will set $\dot{x}_{new} = -\dot{x}$. Now define $\eta = \frac{1}{2}\dot{x}^2 + \gamma x$ to be the total energy of the system, where $\gamma \approx 9.81$ describes the force due to gravity (assume the mass of the ball is 1). By conservation of energy, $\dot{\eta} = 0$ between collisions. Set η^* to be the desired energy (corresponding to a desired apex). Define a reference trajectory for the paddle to follow by $\mu = cx$ where

$$c = \frac{\alpha-1}{\alpha+1} + k(\eta - \eta^*)$$

is constant between collisions. μ is called a *mirror law* because it defines a distorted "mirror" of the ball's trajectory. As the ball goes up the paddle goes down and *vice versa*. The gain, k, adjusts how aggressively the controller minimizes the energy error. If we assume that the paddle follows the reference trajectory very closely, the dynamics of the paddle are then a function of the ball position so that the system is effectively two-dimensional (the position and velocity of the ball). The mirror law controller can be shown (Bühler, Koditschek, and Kindlmann 1994) to drive the ball to the height corresponding to the energy η^*.

Using eq. (14) we define the phase ϕ of the ball so that $\phi = 0$ when it leaves the paddle, $\phi = \pi$ at the highest point of its flight, and $\phi = 2\pi$ as it hits the paddle again. Suppose the ball rebounds from a collision with the paddle with velocity $\dot{x}_0$. By integrating the dynamics $\ddot{x} = -\gamma$ and noting that collisions occur when $x = 0$, we obtain the time since the last impact and the time between impacts, a computationally effective instance of eqs. (12) and (13), as

$$t^- = \frac{\dot{x}_0 - \dot{x}}{\gamma} \quad \text{and} \quad s = t^- + t^+ = \frac{2\dot{x}_0}{\gamma}, \tag{17}$$

respectively. The change of coordinates $h : (\mathbb{R}^+ \times \mathbb{R}) - (0,0) \mapsto S^1 \times \mathbb{R}^+$ from ball coordinates to phase coordinates is given by $h(x, \dot{x}) = (\phi, \dot{\phi})$ where, following the recipe (eq. (14)), we take

$$\phi = \frac{\pi(\dot{x}_0 - \dot{x})}{\dot{x}_0} \quad \text{and} \quad \dot{\phi} = \frac{\pi\gamma}{\dot{x}_0}\,. \tag{18}$$

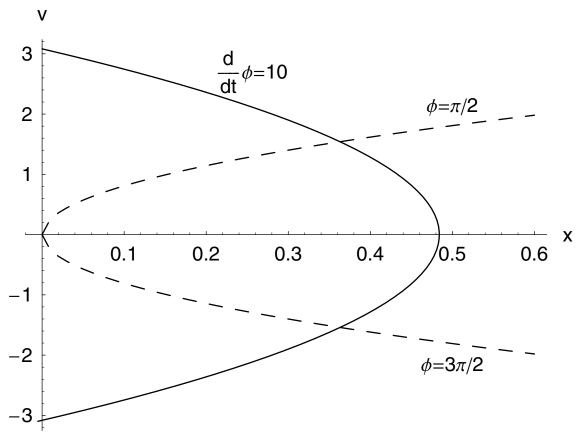
In Figure 4(a) we illustrate the relationship between phase, phase velocity and the orbits of the juggling model.

3.2.2. Hopping

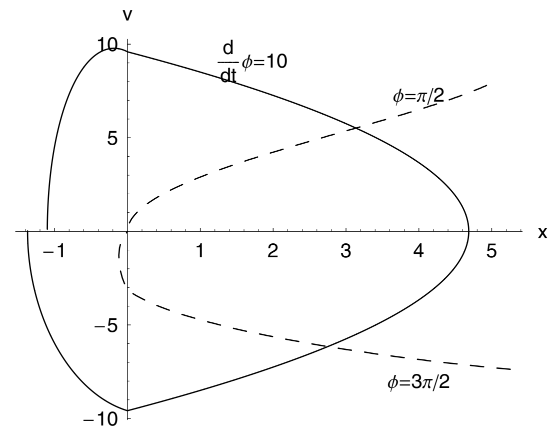
We model a single, vertical hopping leg, a mass $m = 1$ attached to a massless spring leg, by a dynamical system with three discrete modes: flight, compression and decompression. The latter two modes each have the dynamics of a linear, damped spring. Flight mode is entered again once the leg has reached its full extension. The equations of motion are

$$\ddot{x} = \begin{cases} -g & \text{if } x > 0 & \text{flight} \\ -\omega^2(1+\beta^2)x - 2\omega\beta\dot{x} & \text{if } x < 0 \wedge \dot{x} < 0 & \text{compression} \\ -\omega_2^2(1+\beta_2^2)x - 2\omega_2\beta_2\dot{x} & \text{if } x < 0 \wedge \dot{x} > 0 & \text{decompression,} \end{cases} \tag{19}$$

where ω and β are parameters which determine the spring stiffness $\omega^2(1+\beta^2)$ and damping $2\omega\beta$ during compression.



(a)



(b)

Fig. 4. The relationship between phase, phase velocity and the orbits of the (a) juggling and (b) hopping systems. Shown with black lines are typical one-cycle orbits, corresponding to a certain phase velocity, and with dashed lines, the curves $\phi = \pi/2$ and $\phi = 3\pi/2$.

This model is similar to that studied in Koditschek and Bühler (1991) where a period of thrust at the beginning of decompression was used to stabilize the hopper. We abstract the dynamics of thrust and suppose that, during decompression, thrust simply results in a change in spring stiffness and damping. Thus, ω_2 and β_2 are control inputs in our model.

Let x_b be the lowest point that the robot reaches in a given hop, just before decompression. In the appendix we show that choosing $\beta_2 = \beta$ and $\omega_2 = \omega\nu(x_b)$ where

$$\nu = (1 - k_b)e^{\beta\pi}/(1 - x_b)$$

results in a feedback controller that stabilizes the system to have its maximal compression point at k_b. In fact, it can be shown that the discrete, real-valued *return map* $f : \mathbb{R}^- \to \mathbb{R}^-$ that takes the maximal compression point of cycle k to the maximal compression point of cycle $k + 1$ is

$$x_{b,next} = f(x_b) = \frac{(1 - k_b)x_b}{1 - x_b}. \quad (20)$$

To determine the phase of this system, it suffices to derive the period, $s(x_b)$, of a cycle starting at $(x_b, 0)$. The value of s is obtained by summing the decompression time t_d, the flight time t_f, and the compression time t_c. It is shown in the appendix that

$$\begin{aligned} s(x_b) &= t_d + t_f + t_c \\ &= \frac{(\pi - \theta_l)e^{\beta\pi}(1 - x_b)}{1 - k_b} \\ &\quad - \frac{2}{\gamma}\omega(1 - k_b)\sqrt{1 + \beta^2}e^{\beta\theta_l}\left(\frac{x_b}{1 - x_b}\right) + \frac{\theta_l}{\omega}, \end{aligned} \quad (21)$$

where $\theta_l \triangleq \tan^{-1}(\frac{1}{\beta})$. Given the period corresponding to a particular x_b, we define the phase of a point $(x, \dot{x})$ to be $\phi(x, \dot{x}) = 2\pi t^-(x, \dot{x})/s(x_b)$. In Figure 4(b) we illustrate the relationship between phase, phase velocity and the orbits of the hopping model.

It can be shown that s is a diffeomorphism on $(-\infty, 0)$. We may, therefore, work equally well with the conjugate map,

$$g(T) \triangleq s \circ f \circ s^{-1}(T), \quad (22)$$

representing each orbit of the system (eq. (19)) uniquely by its period.

3.3. Implementing the Reference Dynamics

Knowing how to juggle one ball, or control one leg, should somehow lead us to a way of juggling two balls or synchronizing the controllers of two legs. We now show how to use the reference field (eq. (4)) in the case that $A : B = 1 : 1$ to accomplish both of these tasks with only slight modifications to the control algorithms presented above.

3.3.1. Juggling Two Balls

For a two ball system with ball positions x_1 and x_2, we obtain two phases ϕ_1 and ϕ_2. The velocity $\dot{\phi}_i$ is reset instantaneously upon collisions, corresponding to the update rule (eq. (16)).

We next take advantage of the fact that the flow $G^t = H \circ F^t \circ H^{-1}$, described in Section 3.1 and instantiated here, has the very simple form $(y_1, \dot{y}_1, y_2, \dot{y}_2) \mapsto (y_1 + \dot{y}_1 t, \dot{y}_1, y_2 + \dot{y}_2 t, \dot{y}_2)$ between collisions. For each ball i we define a mirror law, μ_i which the paddle should follow when it is about to hit ball i. First, define Φ to be the phase of ball two when the next ball one collision occurs. Then

$$\Phi_1 = \frac{\dot{\phi}_2}{\dot{\phi}_1}(2\pi - \phi_1) + \phi_2.$$

Now, for the first ball we require that, after its next collision,

$$\dot{\phi}_{1,new} = \frac{-\pi\gamma}{\dot{x}_{new}} = \frac{-\pi\gamma}{-\alpha\dot{x}_1 + (1+\alpha)c_1\dot{x}_1} = \mathcal{R}_1(\Phi_1), \quad (23)$$

where c_1 is the coefficient in the mirror law trajectory $\mu_1 = c_1 x_1$ and $\mathcal{R}_1(\phi)$ is the first component of the reference field (eq. (4)). Let η_1 be the energy of the first ball. Solving for c_1 and using the fact that when $x_1 = 0$, the potential energy is 0 so that $\dot{x}_1 = \sqrt{2\eta_1}$, gives

$$c_1 = \frac{1}{(1+\alpha)\sqrt{2\eta_1}}\left[\alpha\sqrt{2\eta_1} - \frac{\pi\gamma}{\mathcal{R}_1(\Phi_1)}\right]. \quad (24)$$

A similar expression for c_2 can be obtained in terms of $\mathcal{R}_2(\Phi_2)$. This gives us a mirror law for each ball. Combining these trajectories into a single trajectory of the form

$$\mu = q(Q_1, Q_2)\mu_1 + (1 - q(Q_1, Q_2))\mu_2 \quad (25)$$

that allows the paddle to manage both systems requires an attention function $q : \mathbb{T}^2 \to [0, 1]$ which is 1 before ball one hits and 0 before ball two hits. This is beyond the scope of this paper, but is discussed in Klavins (2000). We will assume in Section 3.4 that, away from the situation where both balls strike the paddle simultaneously, the paddle can service both mirror laws in an interleaved fashion.

3.3.2. Synchronized Hopping

Now suppose we have two physically unconnected hoppers, operating simultaneously, with states $(x_1, \dot{x}_1)$ and $(x_2, \dot{x}_2)$. We will show how to control both hoppers so that they are kept out of phase (one is at its highest point while the other is at its lowest point) and so that they stabilize at a desired hopping height x_b^* (or period T^*). We do this essentially by changing the set-points, now denoted $k_{b,i}$, for each hopper according to the phase of the other hopper. This corresponds to changing the period and thus allows us to regulate the relative phase of the hoppers.

To apply our phase regulation algorithm we reset the gains $k_{b,i}$, each time a leg reaches its lowest point, according to the reference field (eq. (4))

$$k_{b,i} \leftarrow \mathcal{R}(\phi_j) \triangleq k_b - k_s \sin(\phi_j), \quad (26)$$

where $j = 3-i$ and k_s is a gain about which we will have more to say later. The parameter k_b sets the desired lowest point in a cycle (which defines the hopping height and, equivalently, the period). Recall that k_b corresponds to period. It appears in the first term of the phase regulation expression instead of phase velocity because it is convenient to later analysis. Using the fact that changing $x_{b,i}$ is equivalent to changing the period T_i, this amounts to a period adjustment scheme for each leg that pushes them out of phase with each other. However, a leg does not respond immediately to the reset because control is asymptotic and not deadbeat. It must, therefore, be shown that this simple method indeed achieves the desired result.

We have defined a system that may be described by the state vector $x = (\phi_1, \phi_2, T_1, T_2) \in \mathbb{T} \times \mathbb{R}^+ \times \mathbb{R}^+$ which evolves as follows. We have $\dot{\phi}_i = 2\pi/T_i$ until some ϕ_i becomes $2\pi \equiv 0$. At this point, its desired hopping height is changed according to eq. (26) and the period is reset according to the assignment $T_i \leftarrow g_{k_{b,i}}(T_i)$. The system then continues similarly.

3.4. Analysis

We now analyze the local stability of the controlled systems described in Section 3.3. To analyze such systems, we derive the *return map* of each, as described in Section 3.1. That is, we let $\Sigma_1 = \{(\phi_1, \phi_2, \dot{\phi}_1, \dot{\phi}_2) \mid \phi_1 = 0\}$ and define a map $F : \Sigma_1 \to \Sigma_1$ that gives the phase of oscillator two, and the phase velocities of both oscillators, just before the first oscillator's phase becomes zero. We assume that zero phase crossings alternate. That is, start with a point $\mathbf{w} \in \Sigma_1$, integrate the system forward to obtain a point in Σ_2, then integrate again to get a point in $F(\mathbf{w}) \in \Sigma_1$. Then, compute the Jacobian $J_{\mathbf{w}^*}$ at the fixed point $\mathbf{w}^*$ of F corresponding to the out of phase situation given by eq. (15). If the eigenvalues of the $J_{\mathbf{w}^*}$ lie within the unit circle in the complex plane, then $\mathbf{w}^*$ is a stable fixed point of F. In both models we show this to be the case for certain parameters of the system.

Notice that when $A : B = 1 : 1$, then $\mathcal{R}(0, \phi_2) = \mathcal{R}(\phi_1, 0)$. To simplify notation in this section, we redefine $\mathcal{R} : S^1 \to \mathbb{R}$ to be the reference field restricted to $\phi_1 = 0$. Therefore, with $A{:}B = 1{:}1$,

$$\mathcal{R}(\phi) = \kappa_1 - \kappa_2 \sin(\phi). \quad (27)$$

3.4.1. Local Analysis of Juggling

Supposing that the paddle in the juggling system exactly tracks the reference trajectory (eq. (25)), we may consider the juggling system as equivalent to the system $(\phi_1, \phi_2, \dot{\phi}_1, \dot{\phi}_2) \in \mathbb{T}^2 \times \mathbb{R}^2$ where $\dot{\phi}_i$ is constant except for discrete jumps made when $\phi_i = 0$. These jumps are governed by the reference field (eq. (4)). That is, when $\phi_i = 0$, $\dot{\phi}_i$ instantaneously becomes $\mathcal{R}(\phi_1, \phi_2)$.

A point in Σ_1 has the form $w = (0, \phi_2, \dot{\phi}_1, \dot{\phi}_2)$. This maps to the point $w' = (\Phi_1, 0, \mathcal{R}(\phi_2), \dot{\phi}_2) \in \Sigma_2$ where Φ_1 is the phase of the first system when the trajectory of the total system first intersects Σ_2. w' in turn maps to the point $f(w) = (0, \Phi_2, \mathcal{R}(\phi_1), \mathcal{R}(\Phi_1))$ where Φ_2 is the phase of the second system when the trajectory next intersects Σ_1. The phases Φ_1 and Φ_2, which can be obtained via the point-slope formula for a line (in the ϕ_1, ϕ_2 plane), are given by

$$\Phi_1 = \frac{\mathcal{R}(\phi_2)}{\dot{\phi}_2}(2\pi - \phi_2) \quad \text{and} \quad \Phi_2 = \frac{\mathcal{R}(\Phi_1)}{\mathcal{R}(\phi_2)}(2\pi - \Phi_1)\ . \quad (28)$$

Let $(x, y, z) = (\phi_2, \dot{\phi}_1, \dot{\phi}_2)$. Then, expanding $f(w)$, we obtain a discrete, real valued map on Σ_2 given by

$$\begin{aligned} x_{k+1} &= \frac{\mathcal{R}\left[\dfrac{\mathcal{R}(x_k)}{z_k}(2\pi - x_k)\right]}{\mathcal{R}(x_k)}\left[2\pi - \frac{\mathcal{R}(x_k)}{z_k}(2\pi - x_k)\right] \\ y_{k+1} &= \mathcal{R}(x_k) \\ z_{k+1} &= \mathcal{R}\left[\frac{\mathcal{R}(x_k)}{z_k}(2\pi - x_k)\right] . \end{aligned} \tag{29}$$

Since none of the functions depend on y, we can treat y as an output of this system. Thus, analytically, it will suffice to treat eq. (29) as an iterated map of the variables $(x, z) \in S^1 \times \mathbb{R}^+$ given by $F(x_k, z_k) = (x_{k+1}, z_{k+1})$. We have the following fixed point conditions:

PROPOSITION 1. $F(x, z) = (x, z)$ if and only if $\mathcal{R}(x) = \mathcal{R}(2\pi - x) = z$.

We omit the proof, which is straightforward algebra (note that the values of x are always taken modulo 2π since $x = \phi_2 \in S^1$). For the reference field we are using, we have:

COROLLARY 2. If $\mathcal{R}(\phi) = \kappa_1 - \kappa_2 \sin(\phi)$, then the only fixed points of F are (π, κ_1) and $(0, \kappa_1)$.

We wish to show that the first fixed point, (π, κ_1), is stable, since it corresponds to the situation where the two subsystems are out of phase and at the desired velocity. To do this, we examine the Jacobian:

$$J_{(\pi,\kappa_1)}F = \begin{pmatrix} \dfrac{\kappa_1^2 - 3\kappa_1\kappa_2\pi + \kappa_2^2\pi^2}{\kappa_1^2} & \kappa_2\left(\dfrac{\kappa_2\pi}{\kappa_1} - 1\right) \\ \dfrac{\pi(\kappa_1 - \kappa_2\pi)}{\kappa_1^2} & -\dfrac{\kappa_2\pi}{\kappa_1} \end{pmatrix} . \tag{30}$$

Values of κ_1 and κ_2 which guarantee that the eigenvalues of eq. (30) lie within the unit circle are not difficult to find. For example, if the desired phase velocity κ_1 is given, then we can choose κ_2, which adjusts how aggressively the balls are pushed out of phase, to be $\kappa_2 = \frac{1}{2}\kappa_1/\pi$:

PROPOSITION 2. If $\kappa_2 = \frac{1}{2}\kappa_1/\pi$ then the eigenvalues of $J_{(\pi,\kappa_1)}F$ are both 1/2. The point (π, κ_1) is a stable fixed point of F under these conditions.

The proof is just a calculation: simplify eq. (30) using the constraint and compute the eigenvalues. In practice, it is not difficult to find other parameters which make F stable. For a given κ_1, we first choose κ_2 to be quite small and increase it slowly while the controller remains aggressive, yet still stable. It is also simple to show

PROPOSITION 3. The eigenvalues of $J_{(0,\kappa_1)}F$ are 0 and $1 + \frac{4\pi\kappa_2}{\kappa_1}$. The point $(0, \kappa_1)$ is an unstable fixed point of F.

This follows from the fact that κ_1 and κ_2 are both positive. Proposition 3 shows that the situation in which the two balls collide with the paddle simultaneously is repelling: the system is driven away, locally, from this "obstacle."

3.4.2. Local Analysis of Synchronized Hopping

In deriving the return map of the juggling system (eq. (29)) we used the fact that the paddle can strike the ball at just the right velocity to realize the reference field directed updates to phase velocity. In the hopping system, however, adjusting the spring stiffness in decompression does not allow for this. That is, in the juggling system, after a ball one hit, the phase velocity is adjusted according to $\dot{\phi}_1 \leftarrow \mathcal{R}(\phi_2)$. However, in the hopping system, when leg one reaches the bottom-most part of a cycle, the spring stiffness is adjusted so that the period of the hopper is reset according to $T_1 \leftarrow g_{k_b}(T_1)$ where g is eq. (22) and $k_b = \mathcal{R}(\phi_2)$ is obtained from the reference trajectory.

3.4.3. Derivation of the Return Map

Define to be $\Sigma = \{(\phi_1, \phi_2, T_1, T_2) \mid \phi_1 = 0\}$. Assuming that resets of the legs alternate, we construct the return map $F : \Sigma \to \Sigma$. We begin with a point $(0, \phi_2, T_1, T_2)$ just before resetting the period of hopper one. This evolves until a reset of hopper two. If we suppose that Φ_1 is the phase of hopper one just before hopper two is reset, then, just after the reset we have the point $(\Phi_1, 0, g(\mathcal{R}(\phi_2), T_1), T_2)$, where $g(x_b, T) = g_{k_b}(T)$ is as in eq. (22). This point evolves back to Σ so that the state just before hopper one is reset for a second time is $(0, \Phi_2, g(\mathcal{R}(\phi_2), T_1), g(\mathcal{R}(\Phi_1), T_2))$ where Φ_2 is the phase of the second hopper just before the second reset of the first hopper. Calculating Φ_1 and Φ_2 we have

$$\begin{aligned} \Phi_1 &= \frac{T_2}{g(\mathcal{R}(\phi_2), T_1)}(2\pi - \phi_2) \\ \Phi_2 &= \frac{g(\mathcal{R}(\phi_2), T_1)}{g(\mathcal{R}(\Phi_1), T_2)}(2\pi - \Phi_1). \end{aligned}$$

Letting $x = T_1$, $y = \phi_2$ and $z = T_2$ we obtain that the three dimensional, discrete, real-valued return map $F(x, y, z) = (x', y', z')$ corresponding to the oscillating system (eq. (19)) is defined by

$$\begin{aligned} x' &= g(\mathcal{R}(y), x) \\ y' &= \frac{g(\mathcal{R}(y), x)}{g[\mathcal{R}(\frac{z}{g(\mathcal{R}(y),x)}(2\pi - y)), z]} \\ &\quad \left[2\pi - \frac{z}{g(\mathcal{R}(y), x)}(2\pi - y)\right] \\ z' &= g\left[\mathcal{R}(\frac{z}{g(\mathcal{R}(y), x)}(2\pi - y)), z\right]. \end{aligned} \tag{31}$$

It is instructive to compare these equations with the return map (eq. (29)) for juggling—the difference being the appearance of

g which accounts for the lag between the assertion of control and its effect.

3.4.4. Local Stability of the Return Map

It can be shown that the point (T^*, π, T^*) is a fixed point of this system, where $T^* = s_{k_b}(k_b)$ is the period corresponding to the set-point k_b. We now wish to show

PROPOSITION 4. The point (T^*, π, T^*) is a stable fixed point of the system defined by eq. (31) when the synchronization gain k_s is chosen to be

$$\frac{1}{b\pi k_b}(a + c - bk_b)\left[2k_b - 2 + \sqrt{1 - 4k_b + 3k_b^2}\right]. \quad (32)$$

The proof is given in the appendix.

3.4.5. Numerical Studies and Simulations

We have simulated various combinations of hoppers and jugglers with various couplings and observed that our method stabilizes each system as expected. Figure 5 shows a simulation of the 1 : 1 juggling system described in Section 3.3.1. Besides the relationship on the synchronization gain κ_2 required to show Proposition 2, other settings of the gains are satisfactory as well. Increasing κ_2 increases the response time of the system. However, a higher setting results in eq. (29) becoming period two stable, then period four and so on. Eventually, too high a setting apparently gives chaotic behavior.

Although Proposition 2 requires a 1 : 1 coupling, we have in fact observed in simulation that any ratio $A : B$ that we tried could be stabilized provided that κ_2 is small enough. Figure 6, for example, shows the "hit-points" (the phase of ball two for a ball one collision and *vice versa*) in a simulation of a system with $A : B = 3 : 4$ which stabilizes after only a few ball/paddle collisions.

In Proposition 4, constraining the value of k_s to a function of k_b achieves analytical simplicity but is hardly necessary. Numerical simulations of the synchronized hopper system suggest a wide interval of k_s settings around the guaranteed values in eq. (32) yield stability. In Figure 7, we show a simulation starting from arbitrarily chosen initial conditions which eventually stabilizes at the desired hopping height and phase relationship. In our simulations, with k_s suitably small, we could not find initial conditions that did not eventually stabilize—leading us to believe that the system is in fact globally asymptotically stable.

Of course, the general theory laid out in Section 2, and particularly in Section 2.3, may be applied to intermittent contact systems with more than two oscillators. Figure 8 shows a six-hopper system as an example of this. The connection matrix for the system is eq. (1) and thus specifies an *alternating tripod* wherein two groups of three are synchronized out of phase.

4. Conclusion

In this paper we have explored a means of coupling cyclic systems. Our approach has been to develop a class of reference systems with first order dynamics which serve as model systems for controlling more complex types of systems, such as the intermittent contact systems presented in Section 3. The model systems are constructed using a specification of the phase relationships desired among the component systems as well as the communication, or neighbor structure to be used. It is our belief that developing such specification/implementation methods for robotics is crucial to developing formal methods and dynamical-systems-based programming languages for robotics. These in turn would address scalability and modularity problems that prevent robotics from benefitting from the "recursive explosion" that, for example, computer science has enjoyed.

There are several issues that were beyond the scope of the present paper or which remain to be explored. The classification of which connection graphs give rise to reference systems that perform the tasks specified by the graphs will eliminate the need for checking the Jacobian of each equilibrium orbit. We hope to develop tools for recursive compositions of cyclic systems: a coupled system, at equilibrium or not, has a phase which can be coupled to the phase of yet another system to produce yet another cyclic system. Finally, we have implemented these control ideas in our hexapod robot (Saranli, Bueler, and Koditschek 2000). Tuning certain gains allows us to examine a design space described by the degree of decentralization as well as the amount of feedback from the environment the control algorithm uses. The results suggest that performance may improved by occupying a suitable portion of this space. We have begun to report on our results (Klavins et al. 2001), and a more thorough treatment is forthcoming.

Appendix

Single Hopper Return Map

To integrate the system in eq. (19), we change coordinates in the compression and decompression phase (Bühler, Koditschek, and Kindlmann 1994) so that the system

$$A = \begin{bmatrix} 0 & 1 \\ -\omega^2(1+\beta^2) & -2\omega\beta \end{bmatrix}$$

is in real canonical form (Hirsch and Smale 1974). The change of basis is given by

$$W = \begin{bmatrix} \omega\sqrt{1+\beta^2} & \frac{\beta}{\sqrt{1+\beta^2}} \\ 0 & \frac{1}{\sqrt{1+\beta^2}} \end{bmatrix}.$$

In the new coordinates we have the system

$$\begin{bmatrix} \dot{y} \\ \ddot{y} \end{bmatrix} = B \begin{bmatrix} y \\ \dot{y} \end{bmatrix}, \quad B = WAW^{-1} = \omega \begin{bmatrix} -\beta & 1 \\ -1 & -\beta \end{bmatrix}.$$

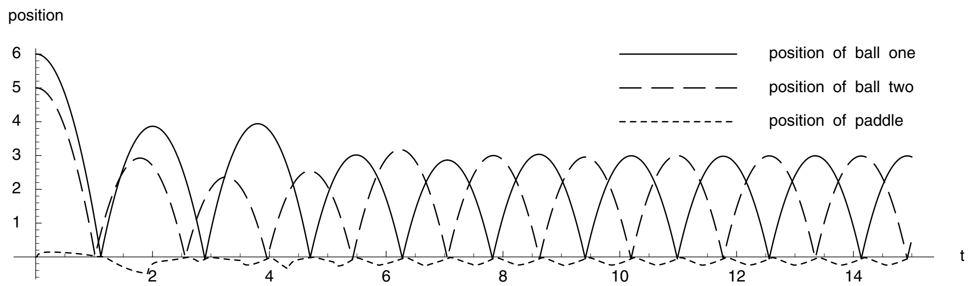


Fig. 5. A simulation of the 1 : 1 juggling system described in Section 3.3.1. The positions of the balls and paddle are shown as functions of time.

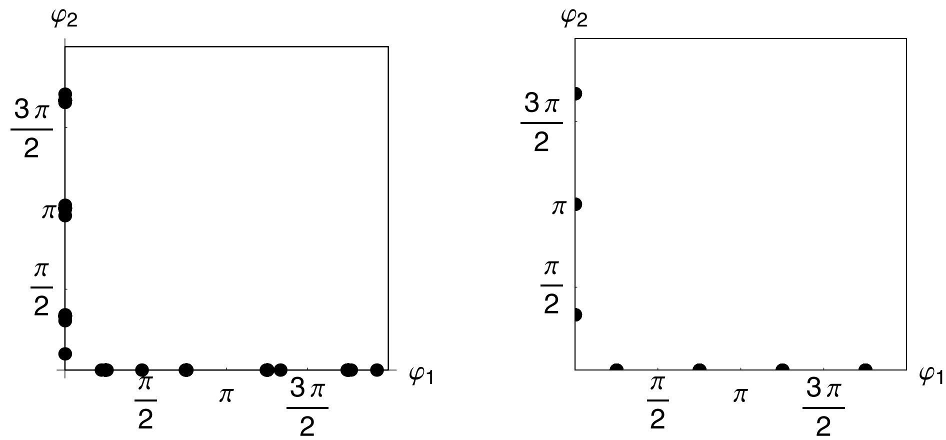


Fig. 6. A simulation of the juggling system for $A : B = 3 : 4$. Each dot corresponds to a ball-paddle collision in a simulation with 40 collisions. (a) The first 20 collisions are scattered around the equilibrium orbit. (b) The second 20 are at the equilibrium orbit, showing 3 ball one hits for every 4 ball two hits.

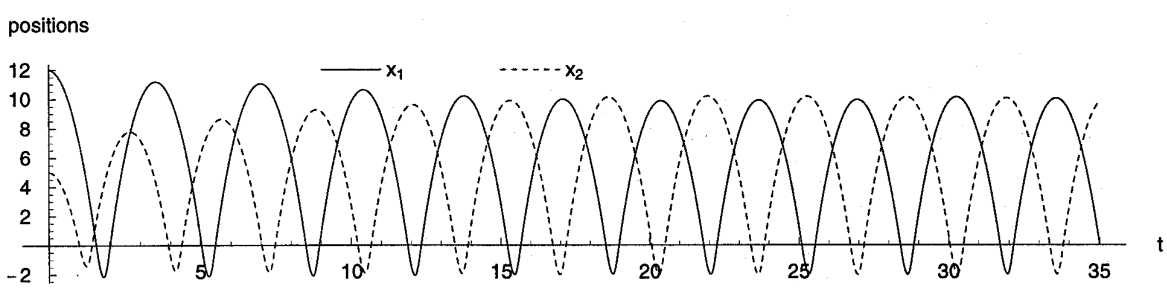


Fig. 7. A simulation of the 1 : 1 synchronized hopping system described in Section 3.3.2. The positions of the leg masses are shown as functions of time.

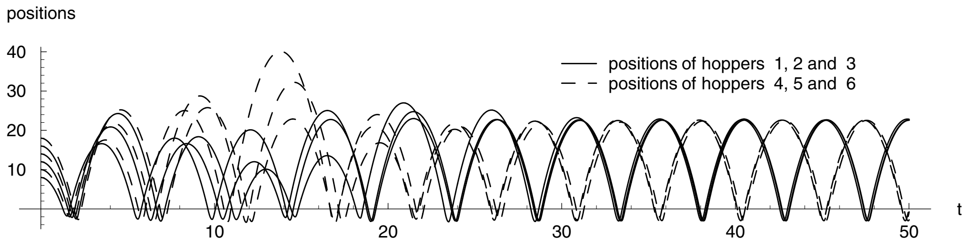


Fig. 8. A simulation of six hopping robots synchronized into two out of phase groups. The two groups consisting of robots one, two and three and robots four, five and six respectively, are forced to be in phase with other members of their groups.

We define energy and angle to be

$$E_c \triangleq [x, \dot{x}] W^T W \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$$

$$\theta_c \triangleq \tan^{-1}\left(\frac{\dot{y}}{y}\right) = \tan^{-1}\left(\frac{(1+\beta^2)^{-1/2}\dot{x}}{\omega(1+\beta^2)^{1/2}x + \beta(1+\beta^2)^{-1/2}\dot{x}}\right).$$

The subscript c denotes “compression.” In these coordinates, the compression phase becomes

$$\begin{aligned} \dot{E}_c &= -2\omega\beta E_c \\ \dot{\theta}_c &= -\omega. \end{aligned}$$

A similar expression is obtained for θ_d and E_d where the d stands for “decompression”:

$$\begin{aligned} \dot{E}_d &= -2\omega_2\beta_2 E_d \\ \dot{\theta}_d &= -\omega_2. \end{aligned}$$

Thus, both θ_c and θ_d have a constant rate of change (and decrease in time).

Now, starting at a point $(x_b, 0)$ corresponds to starting at a point (E_0, π). The lift-off point is (E_l, θ_l) and the touchdown point is (E_{td}, θ_{td}). Since energy is conserved in flight, we have that $(E_{td}, \theta_{td}) = (E_l, \theta_l - \pi)$. Now, integrating eq. (A1) gives

$$E_l = E_0 e^{2\beta_2(\theta_l - \pi)}$$

and we know $E_{td} = E_l$. Finally, let E_b be the energy at the next bottom point. Then

$$\begin{aligned} E_b &= E_{td} e^{2\beta(\theta_b - \theta_{td})} = E_{td} e^{2\beta(-\pi - (\theta_l - \pi))} = E_l e^{-2\beta\theta_l} \\ &= E_0 e^{2\beta_2(\theta_l - \pi)} e^{-2\beta\theta_l} = E_0 e^{2[\theta_l(\beta_2 - \beta) - \beta_2\pi]}. \end{aligned}$$

Now, $E_0 = \omega_2^2(1+\beta_2^2)x_b^2$ and $E_b = \omega^2(1+\beta^2)x_{b,next}^2$ and $\theta_l = \tan^{-1}(1/\beta_2)$. Therefore,

$$x_{b,next} = \frac{\omega_2\sqrt{1+\beta_2^2}}{\omega\sqrt{1+\beta^2}} x_b e^{\tan^{-1}(1/\beta_2)(\beta_2 - \beta) - \beta\pi}. \qquad \text{(A1)}$$

Substituting $\beta_2 = \beta$ and $\omega_2 = \omega\nu$ with $\nu = (1-k_b)e^{\beta\pi}/(1 - x_b)$ results in eq. (20).

Derivation of the Period of a Hop

Let t_0, t_l, t_{td} and t_b be the initial time at bottom, the lift-off time, the touchdown time and the next bottom time. Then $t_l = t_d$ is the time of decompression,$t_{td} = t_d + t_f$ is the sum of the decompression time and the flight time, and $t_b = t_d + t_f + t_c$.

Now, integrate $\dot{\theta}_d = -\omega_2$ to get $\theta_l = \theta_0 - \omega_2 t_d$ so that

$$t_d = \frac{1}{\omega_2}(\theta_0 - \theta_l) = \frac{1}{\omega\tau}\left[\pi - \tan^{-1}(1/\beta)\right].$$

We next need the velocity $\dot{x}_l$ at lift-off. This can be found using the equation

$$E_l = (0, \dot{x}_l) W^T W (0, \dot{x}_l)^T = \dot{x}_l^2.$$

Thus,

$$\dot{x}_l = \sqrt{E_l} = -\omega\tau\sqrt{1+\beta^2} x_b e^{\beta(\theta_l - \pi)}.$$

Now integrating the flight phase gives that

$$t_f = \frac{2\dot{x}_l}{\gamma} = -\frac{2}{\gamma}\omega\tau\sqrt{1+\beta^2} x_b e^{\beta(\theta_l - \pi)}.$$

Lastly, integrating $\dot{\theta}_c = -\omega$ from θ_{td} to $-\pi$ and solving for t_c gives

$$t_c = \frac{1}{\omega}\tan^{-1}(1/\beta).$$

Summing the three times and substituting the value for τ given in the main text yields the desired result (eq. (21)).

Proof of Proposition 4

We describe the salient points of the proof of this theorem. Essentially, we linearize F and show that the linearized system is stable at (T^*, π, T^*). To compute the Jacobian of the map F, first, define

$$\begin{aligned} a &\triangleq (\pi - \theta_l)e^{\beta\pi} \\ b &\triangleq \frac{1}{\gamma} 2\omega e^{\beta\theta_l}\sqrt{1+\beta^2} \\ c &\triangleq \theta/\omega \\ \delta &\triangleq \frac{k_b k_s \pi b}{(1-k_b)(a - bk_b + c)}. \end{aligned}$$

Straightforward computation of partial derivatives yields that the Jacobian evaluated at (T, π, T) is equal to

$$\begin{pmatrix} \frac{1}{1-k_b} & \frac{T\delta}{\pi} & 0 \\ \frac{\pi(2+\delta)}{T(1-k_b)} & 1+3\delta+\delta^2 & \frac{\pi(2-k_b+\delta(1-k_b))}{T(k_b-1)} \\ \frac{\delta}{k_b-1} & -\frac{T\delta(1+\delta)}{\pi} & \frac{1}{1-k_b}+\delta \end{pmatrix}. \quad \text{(A2)}$$

Finding the characteristic polynomial of eq. A2 and substituting eq. (32) for k_s gives

$$-\lambda^3 + \xi_1\lambda + \xi_0, \quad \text{(A3)}$$

where

$$\xi_0 = \frac{1}{(k_b-1)^2} \text{ and}$$

$$\xi_1 = \frac{k_b}{(k_b-1)^2}(6 - 7k_b - 4\sqrt{1-4k_b+3k_b^2}).$$

We may now show the following:

LEMMA A1. The roots of eq. (A3) all have magnitude less than one whenever k_b is negative.

Proof. Suppose ρ_1, ρ_2 and ρ_3 are the roots of eq. (A3). Then

$$(\lambda-\rho_1)(\lambda-\rho_2)(\lambda-\rho_3) = \lambda^3 - \xi_1\lambda - \xi_0.$$

Thus,

$$\begin{aligned} \rho_1+\rho_2+\rho_3 &= 0 \\ \rho_1\rho_2+\rho_1\rho_3+\rho_2\rho_3 &= -\xi_1 \\ \rho_1\rho_2\rho_3 &= \xi_0. \end{aligned} \quad \text{(A4)}$$

Now, it can be shown that when $k_b < 0$ the coefficients of eq. (A3) satisfy the conditions $0 < \xi_0 < 1$ and $-1 < \xi_1 < 0$ giving the following conditions on the roots

$$\begin{array}{c} \rho_1+\rho_2+\rho_3 = 0 \\ 0 < \rho_1\rho_2+\rho_1\rho_3+\rho_2\rho_3 < 1 \\ 0 < \rho_1\rho_2\rho_3 < 1 \end{array}.$$

Using these conditions it is straightforward to show that two of the roots are complex conjugates, the other is real and negative and all have magnitude less than one.

Now, since the eigenvalues of eq. (A2) all have magnitudes less than one, we can conclude that (T^*, π, T^*) is a stable fixed point of the system (eq. (31)).

Acknowledgments

The authors thank Robert Ghrist for his helpful comments and his work on a predecessor of this paper (Klavins, Koditschek, and Ghrist 2000). This work is supported in part by DARPA and ONR under grant N00014-98-1-0747, "Computational Neuromechanics: Programming Work in Machines and Animals."

References

Abadi, M., and Lamport, L. 1993. Composing specifications. *ACM Transactions on Programming Languages and Systems* 15(1):75–132.

Altendorfer, R., Saranli, U., Komsuoglu, H., Koditschek, D. E., Brown, B., McMordie, D., Moore, N., Buehler, M., and Full, R. 2000. Evidence for spring loaded inverted pendulum running in a hexapod robot. In *International Symposium for Experimental Robotics*, Honolulu, HI.

Arnold, V. I. 1991. *Mathematical Methods of Classical Mechanics*. Translated by K. Vogtmann and A. Weinstein. Springer, 2nd edition.

Beer, R., Quinn, R., Chiel, H., and Ritzmann, R. 1997. Biologically inspired approaches to robotics. *Communications of the ACM* 40(3):31–38.

Bloch, A. M., Krishnaprasad, P. S., Marsden, J. E., and Murray, R. 1996. Nonholonomic mechanical systems with symmetry. *Archive for Rational Mechanics and Analysis* 136:21–99.

Brockett, R. W. 1983. Asymptotic stability and feedback stabilization. In *Differential Geometric Control*, eds. R.W. Brockett, R.S. Millman, and H.J. Sussman, chap. 3, 181–191. Birkhäuser.

Bühler, M., Koditschek, D. E., and Kindlmann, P. J. 1990. A family of control strategies for intermittent dynamical environments. *IEEE Control Systems Magazine* 10(2):16–22.

Bühler, M., Koditschek, D. E., and Kindlmann, P. J. 1994. Planning and control of robotic juggling and catching tasks. *International Journal of Robotics Research* 13(2).

Burridge, R., Rizzi, A., and Koditschek, D. E. 1999. Sequential composition of dynamically dexterous robot behaviors. *International Journal of Robotics Research* 18(6):534–555.

Calvitti, A., and Beer, R. D. 2000. Analysis of a distributed model of leg coordination. I. Individual Coordination Mechanisms. *Biological Cybernetics* 82(3):197–206.

Charpentier, M., and Chandy, K. M. 1999. Towards a compositional approach to the design and verification of distributed systems. In *World Congress on Formal Methods in the Development of Computing Systems (FM'99)*, Volume 1708 of *Lecture Notes in Computer Science*, eds. J. Wing, J. Woodcock, and J. Davies, 570–589. Springer-Verlag.

Cohen, A. H., Holmes, P. J., and Rand, R. H. 1982. The nature of the coupling between segmental oscillators of the lamprey spinal generator for locomotion: A mathematical model. *Journal of Mathematical Biology* 13:345–369.

Cohen, Avis H., Rossignol, Serge, and Grillner, Sten (eds.). *Neural Control of Rhythmic Movements in Vertebrates*. NY: Wiley Inter-Science.

Cruse, H. 1990. What mechanisms coordinate leg movement in walking arthropods? *Trends in Neurosciences* 13:15–21.

Delcomyn, F. 1980. Neural basis of rhythmic behavior in animals. *Science*.

Donald, B. R., Suh, J., Darling, R. B., Böhringer, K.-F., Baltes, H., and Kovacs, G. 1999. Cmos integrated organic ciliary actuator arrays for general-purpose micromanipulation tasks. *Journal of Microelectromechanical Systems* 8(4):483–496.

Ermentrout, G. B., and Kopell, N. 1994. Inhibition-produced patterning in chains of coupled nonlinear oscillators. *SIAM Journal of Applied Mathematics* 54:478–507.

Full, R. J., and Koditschek, D. E. 1999. Templates and anchors: Neuromechanical hypotheses of legged locomotion. *Journal of Experimental Biology* 202(23):3325–3332.

Guckenheimer, J., and Holmes, P. 1983. *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*. New York: Springer-Verlag.

Hirsch, M. W., and Smale, S. 1974. *Differential Equations, Dynamical Systems, and Linear Algebra*. Academic Press.

Hu, J. J. 2000. *Stable Locomotion Control of Bipedal Walking Robots: Synchronization with Neural Oscillators and Switching Control*. PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.

Klavins, E. 2000. Automatic compilation of concurrent hybrid factories form product assembly specifications. In *Hybrid Systems: Computation and Control Workshop, Third International Workshop*, Pittsburgh, PA.

Klavins, E. 2000. The construction of attention functions for phase regulation. Technical Report CSE-TR-422-00, University of Michigan.

Klavins, E., Koditschek, D. E., and Ghrist, R. 2000. Toward the regulation and composition of cyclic behaviors. In *Proc. Fourth International Workshop on the Algorithmic Foundations of Robotics 2000*, Dartmouth, NH.

Klavins, E., Koditschek, D. E. 2000. A formalism for the composition of concurrent robot behaviors. In *Proc. IEEE Conference on Robotics and Automation*.

Klavins, E., Komsuoglu, H., Koditschek, D. E., and Full, R. J. 2001. The role of reflexes versus central pattern generators in dynamical legged locomotion. In *Neurotechnology for Biomimetic Robots*. MIT Press.

Koditschek, D. E. 1989. The application of total energy as a lyapunov function for mechanical control systems. In *Control Theory and Multibody Systems*, Vol. 97 of *AMS Series in Contemporary Mathematics*, eds. J. Marsden, Krishnaprasad, and J. Simo, 131–158.

Koditschek, D. E., and Bühler, M. 1991. Analysis of a simplified hopping robot. *International Journal of Robotics Research* 10(6):587–605.

Kopell, N. 1995. Chains of coupled oscillators. In *Handbook of Brain Theory and Neural Networks*, ed. M. Arbib, 178–183. Cambridge: MIT Press.

LaSalle, J. P. 1961. *Stability by Liapunov's direct method*. Mathematics in science and engineering. vol. 4. New York: Academic Press.

Lewis, M. A., Etienne-Cummings, R., Cohen, A. H., and Hartmann, M. 2000. Toward biomorphic control using custom AVLSI chips. In *Proceedings of the 2000 International Conference on Robotics and Automation*, San Francisco.

Lozano-Perez, T., and Mason, M. T. 1984. Automatic synthesis of fine-motion strategies for robots. *International Journal of Robotics Research* 3(1):3–24.

Lynch, N. 1996. *Distrbuted Algorithms*. Morgan Kaufman.

Nakanishi, J., Fukuda, T., and Koditschek, D. E. 2000. A brachiating robot controller. *IEEE Transactions on Robotics and Automation* 16(2):109–123.

Ostrowski, J., Desai, J. P., and Kumar, V. 1998. Optimal gait selection for nonholonomic locomotion systems. *International Journal of Robotics Research* 19(3):225–237.

Park, E., Tilbury, D. M., and Khargonekar, P. P. Forthcoming. 1999. Modular logic controller for maching systems: Formal representation and performance analysis using petri nets. *IEEE Transactions on Robotics and Automation* 15(6):1046–1061.

Pearson, K. 1973. The control of walking. *Scientific American* 235(6):72–86.

Pratt, G. A., and Nguyen, J. 1995. Distributed synchronous clocking. *IEEE Transactions on Parallel and Distributed Systems* 6(3):314–328.

Raibert, M. H., Brown, H. B., and Chepponis, M. 1984. Experiments in balance with a 3D one-legged hopping machine. *International Journal of Robotics Research* 3:75–92.

Reif, J. H., and Wang, H. 1999. Social potential fields: A distributed behavioral control for autonomous robots. *Robotics and Autonomous Systems* 27(3):171–194.

Rizzi, A. A., Gowdy, J., and Hollis, R. L. 1999. Distributed programming and coordination for agent-based modular automation. In *The Ninth International Symposium of Robotics Research*, Snowbird, Utah.

Rizzi, A. A., and Koditschek, D. E. 1993. Further progress in robot juggling: Solvable mirror laws. In *Proc. IEEE International Conference on Robotics and Automation*, Atlanta, GA.

Rizzi, A. A., Whitcomb, L. L., and Koditschek, D. E. 1992. Distributed real-time control of a spatial robot juggler. *IEEE Computer* 25(5):12–26.

Saranli, U., Buehler, M., and Koditschek, D. E. 2000. Design, modeling and preliminary control of a compliant hexapod robot. In *Proc. IEEE Conference on Robotics and Automation*.

Saranli, U., Schwind, W. J., and Koditschek, D. E. 1998. Toward the control of a multi-jointed monoped runner. In *Proc. IEEE Conference on Robotics and Automation*, 2676–2682.

Tennenhouse, D. 2000. Proactive computing. *Communications of the ACM* 43(5):43–50.