Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

DMDL for choreography

Hongyi Li and R. W. Brockett

***Abstract*— A dynamical motion description language (DMDL) is devised. The study of the control of unstable mechanical systems is quite challenging, involving difficult problems caused by limited communication rates, quantization errors, and parasitic dynamics. Trajectories following problems addressed here in a novel way, are particularly difficult and had heretofore not been investigated in a language context. We contrast kinematics with dynamics, MDL with DMDL, and Aristotle $mv = F$ with Newton $m\dot{v} = F$. We define the language atoms of DMDL based on the partition of trajectory space (function space) generated by the state space. Useful descriptions of language elements are defined in the phase space instead of the time domain.**

I. INTRODUCTION

In this paper, we describe the mathematical foundation for designing a control system that processes choreographic scripts. Here a choreographic script is a sequence of symbols defining dynamic motion segments which will be discussed in detail in section III. For example, dancers and gymnasts use choreographic scripts to plan their body movements in their performances. The interaction of discrete and continuous objects is one of the fundamental difficulties encountered in application digital computing to understanding or control of physical systems, [1]. To illustrate, consider controlling a vehicle on Mars from a base station on Earth. It is not a good idea to send a detailed analog command specifying the motor torque or velocity at each time spot, due to the time delay and the limited channel capacity. It is better to send high-level discrete commands such as "move", "turn", "stop", or composite commands such as "avoid the rock". Along this line Brockett defined a language for Kinematic machines, called Motion Description Language (MDL), [2], [4]. Later, Krishnaprasad's group at University of Maryland extended MDL to incorporate interrupts from sensors, which they refered to as MDLe, [6]. Sastry's group at UC Berkeley took a graph theoretic approach to the motion control of robots, which they called control primitives. Its structure is similar to object oriented programming languages, such as C++, [7]. It is of equal importance to investigate the inverse problem. Namely, how does one use analog inputs to generate a discrete motion (an element of a cluster of trajectories sharing a number of common properties)? For example, when we speak the word "hello", we produce an acoustic trajectory with time index. If it is close in some metric to the 'normal' acoustic trajectory of "hello" we can recognize the word correctly. Or we can say, the map from trajectory space (continuous) to symbol space (discrete) is not one to one. There are more than one trajectories corresponding the same symbol. Brockett's pulse driven dynamical systems captures this property. A pulse space is defined as a subset of the phase space. The corresponding dynamical system changes discretely under pulse inputs. An analog counter is defined as a pulse driven dynamical system, [1].

Hongyi Li is with the 21st Century COE Project Office, Tokyo Denki Univeristy, Hatoyama, Hiki, Saitama 350-0394, Japan hongyi@ham.coe.dendai.ac.jp

R. W. Brockett is with the Division of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138, USA brockett@hrl.harvard.edu

II. PREMILIARY

Kinematic machines are models of the form

$$\dot{x} = G(x)u; \quad y = h(x)$$

where u, x, and y are functions of time. The input u is subject to some regularity condition, say u belongs to the set of m-dimensional bounded piecewise continuous functions U. x has its range in n-dimensional state coordinate space X. y has its range in p-dimensional output space Y. G is a $n \times m$ matrix that depends only on x. $h : X \longrightarrow Y$ is an actuator-to-output coordinate map. The symbolic inputs are called atoms. The atoms of the MDL are triples of the form (u, k, T). If at time t_0 the machine receives an input string (u_1, k_1, T_1), $\cdots$, (u_i, k_i, T_i) , the state x evolves according to

$$\begin{array}{lll} \dot{x} = G(x)(u_1 + k_1(y)); & y = h(x); & t_0 \le t \le t_0 + T_1 \\ \vdots & \vdots & \vdots \\ \dot{x} = G(x)(u_i + k_i(y)); & y = h(x); & \\ & t_0 + T_1 + \cdots T_{i-1} \le t \le t_0 + T_1 + \cdots + T_i & \end{array}$$

where $k : Y \longrightarrow X$ is a feedback function belonging to a function space K, [2].

The theorem of the completeness of MDL is called expressiveness of Affine Modal Segments in [3]. In order to be able to implement a system, which interprets a family of atoms, it is necessary to index the possible atoms in a finite way. A natural choice is to take an affine linear form. By an *affine atom* we understand that (u, k, T) is of the form

$$\begin{bmatrix} v_1 \\ v_2 \\ \cdot \\ \cdot \\ \cdot \\ v_m \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ \cdot \\ \cdot \\ \cdot \\ u_m \end{bmatrix} + \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1p} \\ k_{21} & k_{22} & \cdots & k_{2p} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ k_{m1} & k_{m2} & \cdots & k_{mp} \end{bmatrix} \begin{bmatrix} y_1 - d_1 \\ y_2 - d_2 \\ \cdot \\ \cdot \\ y_p - d_p \end{bmatrix}$$

with the u_i, k_{ij}, d_i, and T all being real numbers. Where v_i is the input to the machine, y_i is measurement and d_i is the target position.

Theorem 2.1: (Brockett). If G is continuous functions of x whose components satisfy a Lipschitz continuity condition, then affine atoms can be used to generate an arbitrarily good approximation to any curve, which the mechanism is capable

0-7803-9568-9/05/$20.00 ©2005 IEEE

of generating.

Proof. Let $\bar{x}(t)$ be any solution of

$$\dot{x} = G(x)\bar{v}(t).$$

The standard Euler approximation to this solution, obtained by solving the difference equation

$$x(nh + h) = x(nh) + hG(x(nh))\bar{v}(nh),$$

converges to the true solution $\bar{x}(t)$, as step size h goes to zero. Let $u_i = \bar{v}(ih)$, $k_i = 0$, and $T_i = h$ we get a system which approximates the original system just like the Euler approximation. As h goes to zero, the solution of the system driven by the (u, k, T)'s approaches the original solution $\bar{x}(t)$. ∎

Pulse space and pulse driven systems are studied in [1]. It appears that in many cases neurobiological systems communicate by means of pulses rather than bilevel signals. One can attempt to define a pulse by imposing specifications directly on the functions of time. However, because of the character of pulse trains and the processes that generate them, it seems to be more efficient to characterize pulses in term of differential inclusions. This means, we put constraints on u and $\dot{u}$ and possible higher order derivatives. In this paper an example of a **double annulus** model of pulse space is constructed. The dynamical system

$$\dot{x} = -\sin(2\pi x) + u; \quad x(0) = 0$$

can count the number of pulses defined by the above pulse space.

III. Definition of DMDL

The MDL deals with kinematic systems. The differential equations for a kinematic system have no drift term (i.e. no inertia). However, most physical systems have inertia. They are described by differential equations with drift terms, called dynamical systems. A dynamical system can be approximated by a kinematic system only when its moment of inertia is small and its velocity is low. The language, which deals with dynamical systems, is called Dynamical Motion Description Language (DMDL).

Let's define a control system in the following way. Let X be a n-dimensional differentiable manifold. Let

$$\dot{x} = f(x, u), \quad x \in X \tag{1}$$

be a control system on X. By saying a control system can be choreographed, we mean that the system can accept a language as input, and outputs a trajectory choreographed as dictated by the language. In its most narrow sense, the language is a collection of strings of atoms. Each atom describes a cluster of trajectory segments.

A typical diagram of such control system is shown as Figure 1. In order to interpret a language, we need to define equivalence classes of trajectories using a certain equivalence relation. These equivalence classes will be our atoms. One can define such equivalence relation based on homotopy.

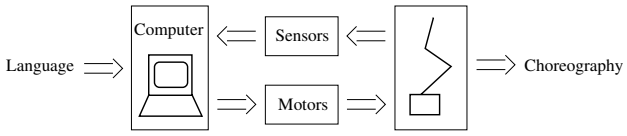

Fig. 1. A typical control system

Definition 3.1: . A feasible trajectory is a trajectory, which can be generated using a feasible control function. An equivalence class of trajectories is a collection of feasible trajectories, which can be deformed from one to another continuously. An atom is a symbol, which defines an equivalence class of trajectories.

Given a control system with k isolated equilibrium points $p_1, p_2, \cdots, p_k$ with l_i being the index of possible motion types. Possible atoms include:

- Stay at equilibrium p_i for time t_i,
- Moving around p_i of type l_i,
- Change from p_i to p_j of type l_i,
- circling around p_i, or a cluster of equilibrium points $p_{i_1} \to p_{i_2} \to \cdots p_{i_s} \to p_{i_1}$.

We can also consider discretization of a function space instead of the state space. The discretization should be fine enough to capture any interesting topology on the function space. First, let's define the trajectory space as

$$\mathcal{S} = \{x([0,T]) | \dot{x} = f(x,u) \quad \text{for some } x(0) = x_0 \\ \text{and some } u(t) \in \Omega \quad \text{for } t \in [0,T]\}.$$

Thus an element $x([0,T]) \in \mathcal{S}$ can be identified by pair $(x_0, u([0,T]))$ with $u(t) \in \Omega$ for $t \in [0,T]$. It maybe happen that two different $u([0,T])$'s result in same trajectory $x([0,T])$ Thus $\mathcal{S}$ is a subset of $X \times \Omega[0,T]$. In general, $\mathcal{S}$ is a function space of infinite dimension. The motion control problem is then to find an appropriate $u([0,T]) \in \Omega$ and/or initial state $x_0 \in X$ to generate the desired trajectory $x([0,t]) \in \mathcal{S}$. In choreography, we do not need to generate a trajectory $x([0,T])$ precisely. We want to reduce the problem to a problem in a finite dimensional space or finite state space. Thus we need to define alphabets on set $\mathcal{S}$.

Definition 3.2: . An **atom** l is a subset of trajectory space $\mathcal{S}$. An **Alphabet** $\mathcal{A} = \{l \,|\, l \subset \mathcal{S}\}$ is a collection of atoms which partitions $\mathcal{S}$. We can also say, alphabet $\mathcal{A}$ is a quotient space of trajectory space $\mathcal{S}/\sim$ where '$\sim$' is the equivalence relation defined by the atoms l's. A **language** $\mathcal{L}$ is subset of $\mathcal{A}^*$ which satisfies a given syntax and contains empty set Φ, where $\mathcal{A}^*$ is the free monoid over $\mathcal{A}$. We say alphabet $\mathcal{A}'$ is **coarser** than $\mathcal{A}$ (or $\mathcal{A}$ is **finer** than $\mathcal{A}'$) if there is an equivalence relation $\sim$ such that $\mathcal{A}' = \mathcal{A}/\sim$.

Thus we can have a hierarchy structure of alphabets based on coarseness ($\prec$),

$$\mathcal{A}_1 \prec \mathcal{A}_2 \prec \cdots \prec \mathcal{S}.$$

An example. Consider a first order system on a torus $T^2 = S^1 \times S^1$

$$\begin{cases} \dot{\theta}_1 = u_1 \\ \dot{\theta}_2 = u_2, \end{cases}$$

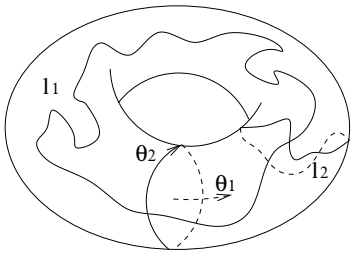

Fig. 2. Alphabets on a torus

with $(\theta_1, \theta_2) \in T^2$. We can define atoms

$$\begin{aligned} l_1 &= \text{ make a big circle in time } T \\ &= \{(\theta_1,\theta_2)([0,T])|\int_0^T d\theta_1 \\ &= 2\pi, \int_0^T d\theta_2 = 0\}, \\ l_2 &= \text{ make a small circle in time } T \\ &= \{(\theta_1,\theta_2)([0,T])|\int_0^T d\theta_1 \\ &= 0, \int_0^T d\theta_2 = 2\pi\}. \end{aligned}$$

They are illustrated in Figure 2. More generally, we can define an atom

$$l(\alpha,\beta,T) = \{(\theta_1,\theta_2)([0,T]) \mid \int_0^T d\theta_1 = \alpha, \int_0^T d\theta_2 = \beta\}.$$

Under the atom $l(\alpha, \beta, T)$, the initial and final states are related by

$$(\theta_1, \theta_2)(T) = (\theta_1, \theta_2)(0) + (\alpha, \beta).$$

With a sequence of $l(\alpha, \beta, T)$'s, we can describe almost all the trajectories. Thus we can define an alphabet

$$\mathcal{A} = \{l(\alpha,\beta,T)|\alpha,\beta \in R; \quad T \in R_+\}$$

and a language $\mathcal{S} = \mathcal{A}^*$. Such $\mathcal{S}$ is a finite dimensional language in the sense that $\mathcal{A}$ is homeomorphic to R^3. We can also define a coarse alphabet and languages on the coarse alphabet. For example, define atom

$$\begin{aligned} l(i,j,k) = \{(\theta_1,\theta_2)([0,k\Delta T]) \mid \int_0^{k\Delta T} d\theta_1 = i\Delta\alpha, \\ \int_0^{k\Delta T} d\theta_2 = j\Delta\beta\}. \end{aligned}$$

It captures the properties of state quantization and time discretization. One can consider simplified motions of diving in term of the above motion atoms. Let θ_1 be angular position of the diver's body with respect to a horizontal axis attached to his/her waist and θ_2 angular position with respect to a vertical axis along his/her body. Then $l(2\pi, 0, T_1)$ represents turning a full circle about waist in time T_1 while $l(0, 4\pi, T_2)$ means turning 2 full circles along body axis in time T_2. A complete diving trajectory can be segmented into a sequence of $l(\theta_{1i}, \theta_{2i}, T_i)$'s.

Consider a simple 2nd order system

$$\ddot{\theta} = u. \tag{2}$$

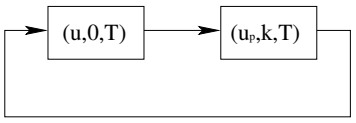

Fig. 3. Example of the syntax of DMD

We want to command (2) to turn a full circle. Assuming $\dot{\theta}(0) = 0$, we can integrate (2) to get the constraint on u to be $x(T) - x(0) = \int_0^T \int_0^t u(\sigma)d\sigma dt = 2\pi$. Then

$$\begin{aligned} x(T) - x(0) &= \int_0^T \int_0^t u(\sigma)d\sigma dt \\ &= \int_0^T \int_\sigma^T u(\sigma)dtd\sigma \\ &= \int_0^T u(\sigma)(T-\sigma)d\sigma \\ &= \int_0^T u(\sigma)g(\sigma)d\sigma \end{aligned} \tag{3}$$

where $g(\sigma) = (T - \sigma)$. By putting a metric $g(t)$ on the input space $\mathcal{U}$, making a full circle in θ space can be down by selecting u such that $\int_0^T u(t)g(t)dt = 2\pi$. $\dot{\theta}(T)$ is still arbitrary. If we want $\dot{\theta}(T) = 0$, then we have another integral constraint, $\int_0^T u(t)dt = 0$. So we need to find candidate solutions from these two constraints.

$$\begin{cases} \int_0^T u(t)g(t)dt = 2\pi \\ \int_0^T u(t)dt = 0 \end{cases} \tag{4}$$

Equation (4) has at least one solution when g is not a constant. One such u could be a function that has positive value around $t_{max} = \arg\max g(t)$, negative value around $t_{min} = \arg\min g(t)$, and zero elsewhere. For example,

$$u = \begin{cases} \frac{\pi}{(g(t_{max}) - g(t_{min}))\Delta t}, \\ \qquad \text{for } t \in [t_{max} - \Delta t, t_{max} + \Delta t], \\ -\frac{\pi}{(g(t_{max}) - g(t_{min}))\Delta t}, \\ \qquad \text{for } t \in [t_{min} - \Delta t, t_{min} + \Delta t], \\ 0, \qquad \text{otherwise.} \end{cases}$$

MDL applied to a kinematic machine is a context free language, i.e., any arbitrary string of atoms is in the language. But DMDL is a context sensitive language. We cannot cascade atoms arbitrarily to form a valid string. That is because we need to take the state of the dynamical system into account. One type of *syntax* to form a valid string is to cascade atoms, which share a common stationary point. For example, $(u, 0, T) \longrightarrow (u_p, k, T) \longrightarrow (u, 0, T) \longrightarrow \cdots$, where $(u, 0, T)$ changes the dynamical system from one equilibrium point/periodic orbit to another equilibrium point/periodic orbit and (u_p, k, T) stabilizes the system at one . It changes from one equilibrium point/periodic orbit to another one equilibrium point/periodic orbit ($u_p = 0$ for equilibrium points), shown in Figure 3.

IV. Useful descriptions of language element

Given a single pendulum, 2nd order system,

$$\ddot{x} + \dot{x} + \sin x = u, \tag{5}$$

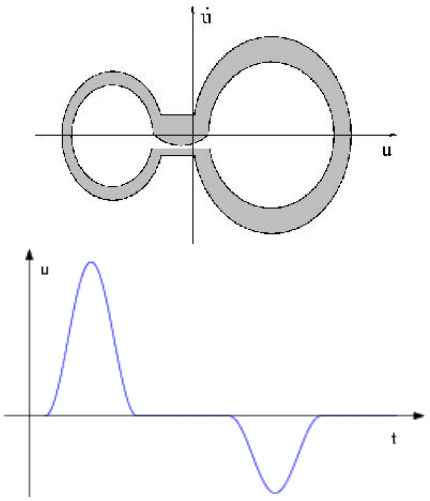

Fig. 4. Above: Dumbbell phase space. Below: A pulse from the dumbbell phase space.

we want to generate motion by selecting the torque u of the driving motor. The pendulum has two type of isolated equilibria, $(x, \dot{x}) = (2k\pi, 0)$ and $(x, \dot{x}) = ((2k+1)\pi, 0)$, $k \in Z$. The reason to put a damping term in (5) is that we want attractive equilibria at $(2k\pi, 0)$ for $k \in Z$. One type of motion we are interested in is to turn a full circle, i.e., change from $x = 2k\pi$ to $(2k+2)\pi$. We are interested in the description of $(u, \dot{u})$ to generate such motion. Suppose the initial state of the system (5) is in a neighborhood of $(x, \dot{x}) = (2k\pi, 0,)$, we can use pulse u to reset the initial velocity $\dot{x}$. With the new $\dot{x}$ in a certain range, system will turn a full circle and approach $((2k+2)\pi, \dot{)}$. In order to rest around the new equilibrium $(2k+2)\pi, 0)$, we need to send a negative pulse to bring $\dot{x}$ close to 0. The state x will then rest at the new equilibrium because of its damping term. Figure 4 shows the sketch of such a u in phase space and time domain respectively. It looks like a dumbbell. The following lemma will describe properties of pulse generated from the dumbbell phase plot. Let

$$S = (-\alpha_2 - \alpha_3\varepsilon, \alpha_1) \times (-\beta_1\varepsilon, \beta_2\varepsilon)$$

$$d_1(u, \dot{u}) = \begin{cases} 0, & \text{for } u < 0 \text{ or } (u, \dot{u}) \in S \\ 1, & \text{otherwise,} \end{cases}$$

$$d_2(u, \dot{u}) = \begin{cases} 0, & \text{for } u > -\alpha_3\varepsilon \text{ or } (u, \dot{u}) \in S \\ 1, & \text{otherwise.} \end{cases}$$

Definition 4.1: We say that $u : [0, \infty) \to R$ is a dumbbell-pulse train if

i. $\omega^2\alpha_1^2(1-\varepsilon)^2 < \dot{u}^2 + \omega^2(u-\alpha_1)^2$, and $\omega^2\alpha_2^2(1-\varepsilon)^2 < \dot{u}^2 + \omega^2(u-\alpha_2-\alpha_3\varepsilon)^2$, and $(u, \dot{u}) \notin [-\alpha_2-\alpha_3\varepsilon, \alpha_1]\times[-\beta_1(\varepsilon-\varepsilon^2), \beta_2(\varepsilon-\varepsilon^2)]$

ii. $\dot{u}^2 + \omega^2(u-\alpha_1)^2 < \omega^2\alpha_1^2$, or $\dot{u}^2 + \omega^2(u-\alpha_2-\alpha_3\varepsilon)^2 < \omega^2\alpha_2^2$, or $(u, \dot{u}) \in (-\alpha_2-\alpha_3\varepsilon, \alpha_1) \times (-\beta_1\varepsilon, \beta_2\varepsilon)$

iii. $|\ddot{u} + \omega^2 u - \alpha_1\omega^2|d_1(u, \dot{u}) + |\ddot{u} + \omega^2 u + (\alpha_2 + \alpha_3\varepsilon)\omega^2|d_2(u, \dot{u}) < \varepsilon$.

Lemma 4.1: (Dumbbell) If $u(\cdot)$ is a dumbbell-pulse train, then:

i. The period between successive pulses, T_d, approaches $\frac{4\pi}{\omega} + \frac{\alpha_3}{\beta_1} + \frac{\alpha_3}{\beta_2}$ as ε goes to zero.

ii. If $d_k(u(t_1)) = d_k(u(t_2)) = 0$ then the integral

$$I_k(t_1, t_2) = \int_{t_1}^{t_2} u(t) d_k(u(t)) dt$$

approaches $(-1)^{k-1}\frac{2\pi\alpha_k}{\omega}$, $k = 1, 2$, times the number of positive and negative pulses in $[t_1, t_2]$ respectively as ε goes to zero.

iii. If $u(t)$ begins and ends around 0, the time integral

$$\frac{1}{2\pi}\int \dot{\theta}(t)dt = \frac{1}{2\pi}\int \frac{\ddot{u}(t)u(t) - \dot{u}^2(t)}{u^2(t) + \dot{u}^2(t)}dt$$

is the number of positive (or negative) pulses.

We omit the proof, because it is almost identical to the proof of Lemma 3 in [5]. In the region of $d_1(u, \dot{u}) = 1$, u approaches one period of positive pulse $\alpha_1(1 - \cos(\omega t))$ as ε goes to zero. In the region of $d_2(u, \dot{u}) = 1$, u approaches negative pulse $\alpha_2(\cos(\omega t) - 1)$. The two strips in between give the separation time between positive and negative pulses. The 2nd order constraint for the pulse train prevents u from making tiny circles when $|\dot{u}|$ is small. We will find use of this pulses generator in the following theorem and determine those parameters.

Theorem 4.1: Given a 2nd order system

$$\ddot{x} + \dot{x} + \sin x = u, \quad x(0) = 0, \quad \dot{x}(0) = 0, \tag{6}$$

there exist parameters $(\omega, \alpha, \beta, \varepsilon)$ such that any dumbbell-pulse train u will increase x by 2π at the end of each period T_d. More precisely, if $u(0) = u(t_1) = 0$ and $\dot{u}(0) = \dot{u}(t_1) > 0$, then

$$(x, \dot{x})(t_1) = (\frac{1}{2\pi}\int_0^{t_1} \frac{\ddot{u}(t)u(t) - \dot{u}^2(t)}{u^2(t) + \dot{u}^2(t)}dt, 0) + O(\varepsilon).$$

Proof. We break this problem into four steps. First, we need to reset the initial velocity $\dot{x}$ to a large enough value using a big positive pulse in the region of $d_1(u, \dot{u}) = 1$. Second, We let the system flow by itself, i.e., u is zero or very small which corresponds to the thin strip below u-axis. Third, when $x(t)$ is close to 2π we need to reset $\dot{x}$ to around zero which is done by the small negative pulse in the region of $d_2(u, \dot{u}) = 1$. Last, we let the system flow by itself again corresponding the thick strip above u-axis. See Fig5 for the vector field of (6) with $u = 0$. The damping term will bring the state to the new equilibrium. The next cycle then starts.

Now we will compute parameters $(\omega, \alpha, \beta, \varepsilon)$ so that the system (6) will evolve as expected. Since we don't have the analytic solution to (6), we will use an approximation. When $\dot{x}$ is large, we can approximate (6) with $u = 0$ by

$$\ddot{x} + \dot{x} = 0 \tag{7}$$

We can integrate it to get $\dot{x} + x = const$. In order to reach point $(x, \dot{x}) = (2\pi, 0)$, we have $\dot{x} + x = 2\pi$. Thus we need $\dot{x}(0) = 2\pi$. We need the large positive pulse u to reset $\dot{x}(0)$.

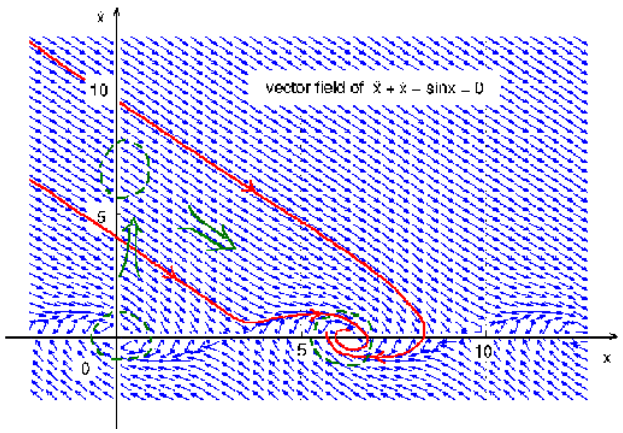

Fig. 5. Vector field

We can pick a large ω so that the duration of the pulse is very short. Then we can approximate (6) by

$$\ddot{x} = u. \tag{8}$$

Let $t_1 = \frac{2\pi}{\omega}$. From lemma 4.1.ii we then have

$$2\pi = \dot{x}(t_1) - \dot{x}(0) = \int_0^{t_1} u(t)dt = \frac{2\pi\alpha_1}{\omega} + h.o.t.$$

Thus $\alpha_1 = \omega$. Now we need to estimate the time t_2 to reach $x(t_2) = 2\pi$ to get β_1. We can use (7) for $x \in [0, \frac{3\pi}{2}]$ and the linearization of (6) for $x \in [\frac{3\pi}{2}, \frac{5\pi}{2}]$, i.e.,

$$\ddot{x} + \dot{x} + x - 2\pi = 0 \tag{9}$$

With initial $(x(t_1), \dot{x}(t_1)) = (0, 2\pi)$, the solution to (7) takes the form of $x(t) = 2\pi(1 - e^{-(t-t_1)})$. At $x(t_{21}) = \frac{3\pi}{2}$, we get $t_{21} = t_1 + \ln(4)$ and $\dot{x}(t_{21}) = 2\pi - x(t_{21}) = \frac{\pi}{2}$. With initial $(x(t_{21}), \dot{x}(t_{21}))$, the solution to (9) takes the form of

$$x(t) - 2\pi = e^{-\frac{t-t_{21}}{2}}[-\frac{\pi}{2}\cos(\frac{\sqrt{3}}{2}(t-t_{21})) + \frac{\pi}{2\sqrt{3}}\sin(\frac{\sqrt{3}}{2}(t-t_{21}))].$$

From $x(t_2) = 2\pi$, we get $\tan(\frac{\sqrt{3}}{2}(t_2 - t_{21})) = \sqrt{3}$. Thus

$$t_2 = t_{21} + \frac{2\pi}{3\sqrt{3}} = \frac{2\pi}{\omega} + \ln(4) + \frac{2\pi}{3\sqrt{3}}.$$

Then we need $\beta_1 = \frac{\alpha_3}{t_2 - t_1} = \alpha_3/(\ln(4) + \frac{2\pi}{3\sqrt{3}})$. At t_2, we have $\dot{x}(t_2) = \frac{\pi}{2}e^{-\frac{\pi}{3\sqrt{3}}}$. We need to use the small negative pulse u to bring $\dot{x}(t_2)$ to zero. Combine equation (8) and lemma 4.1.ii, we have $0 - \dot{x}(t_2) = -\frac{2\pi\alpha_2}{\omega}$. Thus $\alpha_2 = \frac{\omega}{4}e^{-\frac{\pi}{3\sqrt{3}}}$. Then we can select a small enough β_2 so that x will rest at the new equilibrium. Thus, one set of appropriate parameters for the dumbbell-pulse train is

$$(\omega, \alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \varepsilon)$$
$$= (\omega, \omega, \tfrac{\omega}{4}e^{-\frac{\pi}{3\sqrt{3}}}, \alpha_3, \alpha_3/(\ln(4) + \tfrac{2\pi}{3\sqrt{3}}), \beta_2, \varepsilon)$$

where ω is large and $\alpha_3, \beta_2, \varepsilon$ are small. ■

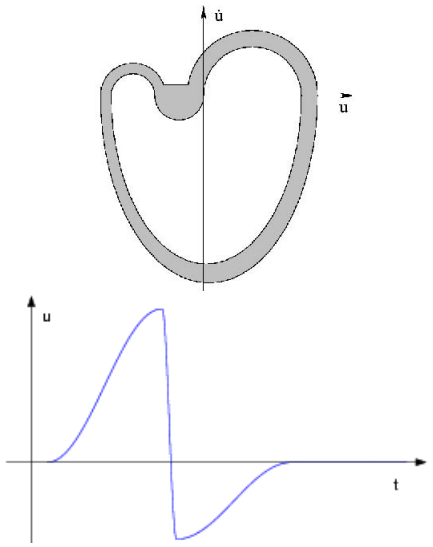

Fig. 6. Above: Bean phase space. Below: A pulse from the bean phase space.

The dumbbell pulse space has a thin belt below, as shown in Figure 4above. This means dumbbell-pulses are not robust, i.e., they are sensitive to perturbations. To improve on this we construct a bean pulse space that overcomes this shortcoming. Let

$$b_1(u,\dot{u}) = \begin{cases} 0, & \text{for } u < 0 \text{ or } \dot{u} < 0 \text{ or } (u,\dot{u}) \in S \\ 1, & \text{otherwise,} \end{cases}$$

$$b_2(u,\dot{u}) = \begin{cases} 0, & \text{for } u > 0 \text{ or } \dot{u} < 0 \text{ or } (u,\dot{u}) \in S \\ 1, & \text{otherwise.} \end{cases}$$

$$b_3(u,\dot{u}) = \begin{cases} 0, & \text{for } u > 0 \text{ or } \dot{u} > 0 \text{ or } (u,\dot{u}) \in S \\ 1, & \text{otherwise.} \end{cases}$$

Definition 4.2: We say that $u : [0,\infty) \to R$ is a bean-pulse train if

i. $\omega_1^2\alpha_1^2 < \dot{u}^2 + \omega_1^2(u-\alpha_1)^2$ and $\omega_1^2\alpha_2^2 < \dot{u}^2 + \omega_1^2(u + \alpha_2 + 2\alpha_3\varepsilon)^2$ for $\dot{u} \geq 0$,
$\omega_2^2(\alpha_1+\alpha_2+\alpha_3\varepsilon)^2 < \dot{u}^2 + \omega_2^2(u-\alpha_1+\alpha_2+\alpha_3\varepsilon)^2$
or $\dot{u}^2 + \omega_1^2(u+\alpha_3\varepsilon)^2 < \omega_1^2\alpha_3^2\varepsilon^2$ for $\dot{u} < 0$

ii. $\dot{u}^2 + \omega_1^2(u-\alpha_1)^2 < \omega_1^2\alpha_1^2(1+\varepsilon)^2$ or $\dot{u}^2 + \omega_1^2(u + \alpha_2 + 2\alpha_3\varepsilon)^2 < \omega_1^2\alpha_2^2(1+\varepsilon)^2$
or $\dot{u}^2 + \omega_2^2(u - \alpha_1 + \alpha_2 + \alpha_3\varepsilon)^2 < \omega_2^2(\alpha_1 + \alpha_2 + (\frac{1}{2}\alpha_1 + \frac{1}{2}\alpha_2 + \alpha_3)\varepsilon)^2$,
or $\dot{u} < \beta\varepsilon$ for $u \in (-\alpha_2 - \alpha_3\varepsilon, \alpha_1)$

iii. $|\ddot{u} + \omega_1^2 u - \omega_1^2\alpha_1|b_1(u,\dot{u}) + |\ddot{u} + \omega_1^2 u + \omega_1^2(\alpha_2 + 2\alpha_3\varepsilon)|b_2(u,\dot{u}) + |\ddot{u} + \omega_2^2 u - \omega_2^2(\alpha_1 - \alpha_2 - \alpha_3\varepsilon)|b_3(u,\dot{u}) < \varepsilon.$

Where $\alpha_1, \alpha_2, \alpha_3, \beta, \omega_1, \omega_2, \varepsilon > 0$, $2\alpha_3 - \alpha_1 - \alpha_2 > 0$, $\omega_2 \gg \omega_1$.

Lemma 4.2: (Bean) If $u(\cdot)$ is a bean-pulse train, then:

i. The minimal time T_b for a complete pulse approaches $\frac{3\pi}{\omega_1} + \frac{\pi}{\omega_2} + \frac{\alpha_3}{\beta}$ as ε goes to zero.

ii. If $b_k(u(t_1), \dot{u}(t_1)) = b_k(u(t_2), \dot{u}(t_2)) = 0$ then the

integral

$$I_k(t_1, t_2) = \int_{t_1}^{t_2} u(t) s_k(u(t), \dot{u}(t)) dt$$

approaches $(-1)^{k-1} \frac{\pi \alpha_k}{\omega_1}$, $k = 1, 2$, times the number of positive and negative pulses in $[t_1, t_2]$ respectively as ε goes to zero.

iii. If $u(t)$ begins and ends around 0, the time integral

$$\frac{1}{2\pi} \int \dot{\theta}(t) dt = \frac{1}{2\pi} \int \frac{\ddot{u}(t)(u(t) - \alpha_1) - \dot{u}^2(t)}{(u - \alpha_1)^2(t) + \dot{u}^2(t)} dt$$

is the number of positive (or negative) pulses.

We omit the proof for the same reason as in Lemma 4.1.

Theorem 4.2: Given a 2nd order system

$$\ddot{x} + \dot{x} + \sin x = u, \quad x(0) = 0, \quad \dot{x}(0) = 0, \tag{10}$$

there exist parameters $(\omega, \alpha, \beta, \varepsilon)$ such that any bean-pulse train u will increase x by 2π at the end of each period T_b. More precisely, if $u(0) = u(t_1) = 0$ and $\dot{u}(0) = \dot{u}(t_1) > 0$, then

$$(x,\, \dot{x})(t_1) = (\frac{1}{2\pi} \int_0^{t_1} \frac{\ddot{u}(t)u(t) - \dot{u}^2(t)}{u^2(t) + \dot{u}^2(t)} dt,\, 0) + O(\varepsilon).$$

The proof is similar to that of Theorem 4.1 and is left to reader.

Take roller on Mars as an example. If we treat the roller as a dynamical system with states $(x, y, \theta, \dot{x}, \dot{y}, \dot{\theta})$, where (x, y) is the position and θ orientation of the roller. Then we can define two types of motion atoms: moving along its orientation l_a and moving perpendicular to its orientation l_p. Suppose the roller is differentially driven. We can define a set of pulse-train control inputs to move individual wheel forward and backward following given velocity profiles. Then atom l_a can be achieved by applying identical input pulse-train to both set of wheels. While atom l_p is not a possible motion. It can only be approximated. l_p can be approximated by applying same pulse-train to both wheel but phase shifted. When control the roller on Mars from a base station on Earth, we only need to transmit those symbolic motion string of atoms with desired parameters instead of exact analog inputs. It will greatly reduce the amount of data to be transmitted. One can also do optimal path planning in term of the above given atoms instead of the physical state space path planning.

V. CONCLUSIONS

A useful class of language elements (atoms) was constructed using phase space characterizations of the input space appropriate for a second order single pendulum system. Two types of phase space characterization were given: the dumbbell phase space and the bean phase space. We can identify these with pulse trains and then show that under either type of pulse train, the pendulum will turn one full circle after each pulse.

REFERENCES

[1] R. W. Brockett, Pulse Driven Dynamical Systems, *Systems, Models and Feedback: Theory and Applications* (A. Isidori and T. J. Tarn Ed.), pp 73-79, Birkhauser, Boston, MA; 1992.

[2] R. W. Brockett, *Lecture Notes on (k,k,T)*, Cambridge, MA; 1988.

[3] R. W. Brockett, On the Computer Control of Movement, *Proc. IEEE Conf. on Robotics and Automation*, Philadelphia, PA, 1988, pp 534-540.

[4] R. W. Brockett, Formal Languages for Motion Description and Map Making, *Proceedings of Symposia in Applied mathematics*, Providence, RI, 1990, pp 181-193.

[5] R. W. Brockett, Dynamical Systems and Their Associated Automata, *Systems and Networks: Mathematical Theory and Applications – Proceedings of the 1993 MTNS*(U. Helmke and R. Mennicken Ed.), vol. 77, pp 49-69, Akademie-Verlag, Berlin, German; 1994.

[6] V. Manikonda and P. S. Krishnaprasad and J. Hendler, Languages, Behaviors, Hybrid Architectures, and Motion Control, *Mathematical Control Theory* (J. Baillieul and Y. Willems Ed.), Springer-Verlag, 1998, pp 199-226.

[7] R. Murray and D. Deno and S. Sastr, Control Primitive for Robot Systems, *IEEE Tans. on Systems, Man and Cybernetics*, vol. 22, 1992, pp 183-193.