UNIVERSITÉ de Liège

FACULTE DES SCIENCES APPLIQUEES

The Wedge Billiard
Experiments on the Flatland setup

Master thesis submitted by

Fabien Defays

in partial fulfillment of the requirements
for the degree of

Diplôme d'Etudes Approfondies en Sciences Appliquées

September, 2003

Acknowledgments

First, I would like to express my gratitude to advisor, Rodolphe Sepulchre. I thank him for his advices, his dedication and all the opportunities he gave me during the two years I have spent in the Department.

I am indebted to Kevin Lynch for inviting me to Northwestern University, and all the members of the lab for their kind hospitality. I wish to express special thanks to Jim Lovsin for the help he gave me and the realization of mechanical parts.

I am grateful to the members of the Jury, B. Boigelot and J. Destiné for accepting to read and evaluate this thesis.

I would like to thank Olivier Brüls for his precious advices during writing.

I thank all my colleagues at the University of Liège. Especially Manuel Gerard with whom I have shared two incredible years. I thank him for the help he gave me during this thesis, for the Matlab script that generates all the simulation results, for his patience , and also for his cheerfulness making these two years unforgettable.

Last, but not least, I would like to thank the members of my family and Valérie who helped and inspired me over these years.

Contents

List of Figures

Chapter 1

Introduction

The design of robots reproducing animal behaviors is an active field of research. The main properties of such robots are their abilities to coordinate their motion and to manipulate objects. In this context, juggling covers a class of challenging problems for researchers. The wedge billiard is a particular juggling configuration studied in this thesis.

The wedge billiard is also an interesting benchmark for theoretical investigations of impact control problems encountered in legged robotics. The dynamics and control analysis of such mechanisms is rendered difficult by the underactuated and discrete nature of their control. The robotics community has made impressive advances over the last decades in building legged robots. Much of this research has focused on mechanisms that can sustain stable locomotion in the absence of actuation. Less research has been devoted to the "active control" of these mechanisms, that is the design of control laws that stabilize an otherwise unstable motion, with the notable exception of the work of Buehler, Koditschek and coworkers who pioneered the active stabilization of juggling machines [2]. The design of stabilizing feedback laws for Buehler planar juggler has also been considered by Lynch [5] and by Sepulchre and Gerard [8].

The stabilization problem considered in [8] is the simplest impact control problem beyond its one-dimensional version: the celebrated bouncing ball or line-juggler model, extensively studied in the literature [7]. Considering the two-dimensional wedge billiard or planar-juggler introduces few additional complexity in the model but raises stabilization questions that are more directly connected to the theoretical issues encountered in legged robotics.

The wedge billiard is an elementary dynamical system : a ball moves on an inclined plane and undergoes elastic collisions with two intersecting edges forming a fixed angle θ with the gravity field direction. Lehtihet and Miller have shown in [4] that depending on the angle θ, the uncontrolled system exhibits a variety of dynamical phenomena, from chaotic motions to periodic orbits.

In order to test the algorithms developed theoretically in the aforementioned works, an experimental setup of the wedge billiard is under development at the University of Liège. A multipurpose testbed, called Flatland, is already available in the *Laboratory for Intelligent Mechanical Systems*, at Northwestern University (Illinois-USA). It consists of an air hockey table which can float lightweight planar objects with nearly zero friction. The table is mounted on a pivoting frame which allows to vary the effects of gravity by varying the table tilt.
The present work studies the adaptation of the Flatland setup to reproduce the dynamical behavior of the ideal uncontrolled wedge billiard.
The control of the edges presented in this work is only used to compensate for the energy losses at the impacts. The problem of stabilizing periodic orbits is not tackled. Beside the non-elasticity of impacts, other implementation problems as the thickness of the elements or the trajectory tracking of the arms are addressed.

In previous works [4, 8], the wedge billiard has been studied under the assumption of elastic impacts, which means that when the system is uncontrolled the energy of the ball does not decrease during the experiment. This assumption is never met in practice and non-elastic impacts lead to energy losses which fundamentally alter the motion of the ball. Therefore, discrete-time control laws for the actuated edges are designed to compensate for these energy losses. Following [8], the mirror algorithm proposed by Buehler and Koditschek [2] is adapted to the present framework : the discrete-time control laws are converted into continuous-time reference trajectories that the actuated edges must track. The position of the arms is a function of the ball position, which is given by a vision system.

The thesis is organized as follows. The first part is dedicated to the theoretical approach. In chapter 2, we derive the dynamical model of the wedge billiard under the elasticity hypothesis. Effects of non-elastic impacts are presented in chapter 3. A control strategy, based on the mirror law, is proposed to tackle energy losses at impact times. The second part of the thesis deals with the mechanical implementation of the wedge billiard. The Flatland setup is depicted in chapter 4. Chapter 5 presents some difficulties encountered with the mechanical implementation of the wedge billiard. Finally, the experimental results are analyzed and compared with simulation results.

Part I

Dynamics and Control of the Wedge Billiard

Chapter 2

Dynamics of a billiard in a gravitational field

This chapter describes the motion of a particle in a constant gravitational field. The particle, considered as a point mass, undergoes collisions with two intersecting edges. Figure 2.1 illustrates the system under consideration.
In this chapter, the system is studied under the assumption of elastic collisions (energy conservation at impact). In the absence of control, the two edges form a fixed angle θ with the direction of gravity. Depending on the angle θ and on the initial conditions of the ball, the system exhibits a variety of dynamical phenomena. For $\theta < 45$ deg, chaotic and stable behaviors coexist with periodic orbits. When $\theta > 45$ deg, the motion of the particle is completely chaotic. $\theta = 45$ deg is a special case ; for this angle, the motion 2 DOF of the particle can be decoupled into two 1 DOF bouncing ball dynamics, each bouncing elastically on a rigid floor.
Effects of non-elastic collisions and control solutions are presented in chapter 3.

2.1 Description of the system

Let θ be the half-wedge angle as shown in Fig 2.1, $\underline{r}$ the position of the ball (unit point) and $\underline{v} = v_r\underline{e}_r + v_n\underline{e}_n$ its velocity. $(\underline{e}_r, \underline{e}_n)$ is an orthogonal frame attached to the fixed intersection point with $\underline{e}_r$ parallel to $\underline{r}$.

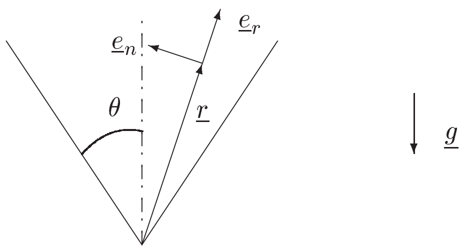


Figure 2.1: The wedge billiard

The mass point evolves along a simple parabolic motion until it collides with an edge. The collisions are assumed to be elastic. Since the motion is unaffected by the mass of the particle, we choose to work with a unit mass point.

The energy conservation of the system is expressed by:

$$E = \frac{1}{2}(v_r^2 + v_n^2) - < \underline{r}, \underline{g} > \tag{2.1}$$

2.2 Discrete billiard map

Stability of periodic orbits is studied via the Poincaré-map relating the state from one impact to the next one. The discrete-state vector, noted $x[k]$, consists of continuous-time variables $x(t)$ evaluated at impact time $t[k]$. Because the continuous-time variables can be discontinues at impact times, we use the notation $x^-(t[k])$ for pre-impact values and $x^+(t[k])$ for post-impact values. Following [8] the discrete-time state will denote post-impact values, thus $x[k] = x^+(t[k])$.

Change of variables. According to [4] and [8], we use the state variables

$$\begin{aligned} V_r &= \frac{v_r}{\cos\theta} \\ V_n &= \frac{v_n}{\sin\theta} \end{aligned}$$

the discrete state vector being

$$x[k] = \begin{pmatrix} V_r[k] \\ V_n[k] \end{pmatrix}$$

The discrete wedge-billiard map is the composition of a parabolic flight map and an impact map. The flight map integrates the continuous-time equation of motion between two successive impacts time, yielding

$$\mathcal{F} \quad \equiv \quad x^-(t[k]) = F_1(x[k-1]). \tag{2.2}$$

The impact rule expresses post-impact variables as a function of pre-impact variables in the form:

$$\mathcal{I} \quad \equiv \quad x[k] = F_2(x^-(t[k])). \tag{2.3}$$

Thus the billiard map, expressing the discrete-time state of the ball as a function of the previous discrete-time state, is given by

$$\mathcal{B} \quad \equiv \quad x[k] = F_2(F_1(x[k-1])). \tag{2.4}$$

In [8], Sepulchre and Gerard review the billiard map developed by Lehtihet and Miller [4]. They analyze the stabilization problem of periodic orbits. Rotational actuation of the arms around their fixed intersection point is used to stabilize one particular orbit of the system. Here, we focus on the unactuated system : the edges are at their equilibrium position. We consider the uncontrolled billiard map.

The impact rule $\mathcal{I}$ adopted assumes that collisions are perfectly elastic. Under this assumption, the tangential velocity is conserved and the normal velocity reverses. Moreover, in the absence of control the energy is conserved. The impact map takes the analytical form :

$$\mathcal{I} \quad \equiv \quad \begin{pmatrix} V_r[k] \\ V_n[k] \end{pmatrix} = \begin{pmatrix} V_r^-(t[k]) \\ -V_n^-(t[k]) \end{pmatrix} \tag{2.5}$$

The flight map is entirely determined by the wedge geometry, that is by the parameter $\alpha = \tan\theta$. Two type of motions are to be considered for the flight map. Type A describes the ball rebouncing on the same edge while type B corresponds to a collision with the opposite edge. Note that the sign of the component of the velocity in the $\underline{e}_n$ direction is conserved in A-motion and changes in

B-motion. Following [8] and [4], we develop the flight maps $\mathcal{F}_\mathcal{A}$ and $\mathcal{F}_\mathcal{B}$ corresponding respectively to A and B motion :

$$\mathcal{F}_\mathcal{A} \equiv \begin{pmatrix} V_r^-(t[k+1]) \\ V_n^-(t[k+1]) \end{pmatrix} = \begin{pmatrix} V_r[k] - 2|V_n[k]| \\ -V_n[k] \end{pmatrix} \tag{2.6}$$

$$\mathcal{F}_\mathcal{B} \equiv \begin{pmatrix} V_r^-(t[k+1]) \\ V_n^-(t[k+1]) \end{pmatrix} = \begin{pmatrix} |V_n[k]| - V_r[k] - |V_n^-(t[k+1])| \\ \sqrt{4E + 2\frac{1-\alpha^2}{(1+\alpha^2)^2}(|V_n[k]| - V_r[k])^2 - V_n^2[k]} sign(V_n[k]) \end{pmatrix} \tag{2.7}$$

The condition that indicates A-motion instead of B-motion is given by

$$V_n^2[k]\sin^2\theta + (V_r[k] - 2|V_n[k]|)^2\cos^2\theta \le 2E \tag{2.8}$$

Condition (2.8) means that A-motion is selected as long as the energy constraint (2.1) is satisfied after being mapped by $\mathcal{F}_\mathcal{A}$. The equality corresponds to the limit case when collision occurs at the wedge vertex.

Composing the flight maps $\mathcal{F}_\mathcal{A}$, $\mathcal{F}_\mathcal{B}$ and the impact map $\mathcal{I}$, one obtains the discrete billiard maps $\mathcal{B}_\mathcal{A}$ and $\mathcal{B}_\mathcal{B}$. $\mathcal{B}_\mathcal{A}$ is the billiard map when impacts k and $k+1$ occur on the same edge while $\mathcal{B}_\mathcal{B}$ is the map for impacts k and $k+1$ on different edges.

$$\mathcal{B}_\mathcal{A} \equiv \begin{pmatrix} V_r[k+1] \\ V_n[k+1] \end{pmatrix} = \begin{pmatrix} V_r[k] - 2|V_n[k]| \\ V_n[k] \end{pmatrix} \tag{2.9}$$

$$\mathcal{B}_\mathcal{B} \equiv \begin{pmatrix} V_r[k+1] \\ V_n[k+1] \end{pmatrix} = \begin{pmatrix} |V_n[k]| - V_r[k] - f_1[k] \\ -sign(V_n[k])f_1[k] \end{pmatrix} \tag{2.10}$$

with

$$f_1[k] = \sqrt{4E + 2\frac{1-\alpha^2}{(1+\alpha^2)^2}(|V_n[k]| - V_r[k])^2 - V_n^2[k]}$$

Properties of the mapping

The properties of the system are explored by constructing a map of the plane which relates the velocity components v_r and v_n at impact points. The wedge is symmetric with respect to the gravitational field, its boundaries are equivalent. Thus, we will only consider the absolute value of v_n. Before examining the numerical results obtained with the billiard map, it is useful to emphasize some general properties of the mapping.

Phase space limits. Because of the energy conservation (2.1) and the geometry of the system

$$<\underline{r}, \underline{g}> = -h \le 0,$$

the phase space is bounded : $v_r^2 + v_n^2 \le 2E$. The equality corresponds to the case $h = 0$, that is the ball is at the intersection of the edges. This case yields to the equation of the bordering circle of the state space :

$$v_r^2 + v_n^2 = 2E \tag{2.11}$$

Balls encountering the wedge vertex. Equation (2.8) means that A-motion (successive bounces on the same edge) is indicated as long as a point of the phase space is still satisfying the energy constraint (2.1) after being mapped by $\mathcal{B}_\mathcal{A}$. Using equations (2.1) and (2.6), we express the condition to have m impacts on the same arm :

$$V_n^2[k]\sin^2\theta + (V_r[k] - 2m|V_n[k]|)^2\cos^2\theta \le 2E \tag{2.12}$$

The equality corresponds to the case where impact m occurs at the intersection of the edges. The points on the curves $\Gamma_m^\pm$ defined by

$$V_r[k] = 2\alpha m V_n[k] \pm \sqrt{2E - V_n[k]^2} \tag{2.13}$$

with m natural, are images of a collision at the wedge vertex. In particular (see Fig. 2.2), $\Gamma_0^{\pm}$ is the phase space limit and $\Gamma_1^{\pm}$ is obtained by using the equality in equation 2.1. It separates the phase space in two regions; A and B. A point in region A is mapped by $\mathcal{B}_{\mathcal{A}}$ and one in region B by $\mathcal{B}_{\mathcal{B}}$.

The transformations $\mathcal{B}_{\mathcal{A}}$ and $\mathcal{B}_{\mathcal{B}}$ are continuous. The boundary curve between the two regions generates the discontinuity in the sense that two nearby points, one of which is in region A, and the other in region B, are separated under the action map. The discontinuity is responsible for regions of global chaos.

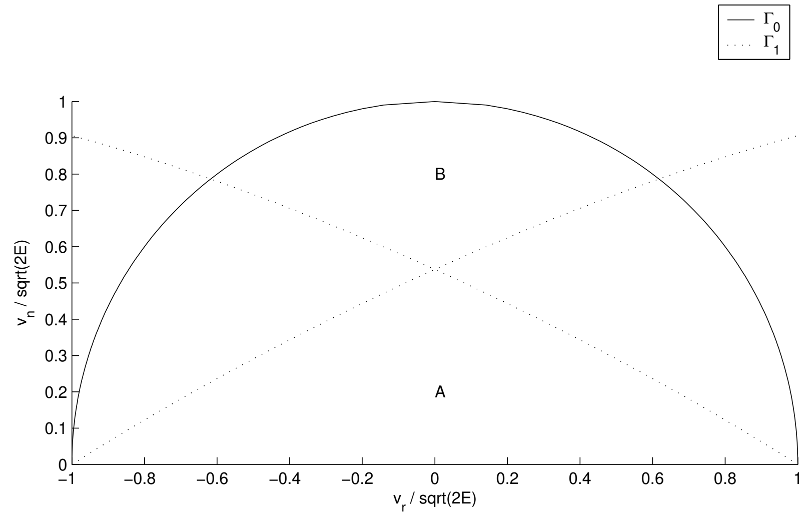


Figure 2.2: Partitioning of the phase space (v_r, v_n)

Period-one orbits. Period-one orbit in the continuous system correspond to a fixed point in the Poincaré map. From its definition, region A cannot contain such points : the ball cannot bounce indefinitely on the same edge, sooner or later the system reaches region B. Consequently, there is no fixed point for $\mathcal{B}_{\mathcal{A}}$. The fixed points are in region B.

Introducing the condition $(V_r[k], V_n[k]) = (V_r[k+1], V_n[k+1])$ in equation (2.10), we find the coordinates of the fixed point $(V_{n,e}, V_{r,e})$:

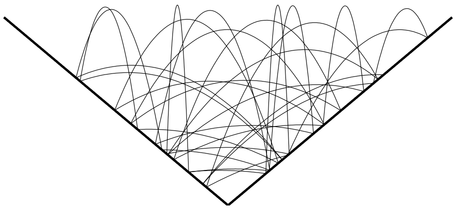
$$\begin{aligned} V_{n,e} &= 0 \\ V_{r,e} &= \tfrac{1+\alpha^2}{\alpha}\sqrt{\tfrac{1}{\alpha^2+3}} \end{aligned} \tag{2.14}$$

The stability of this point and the general structure of the map depend on the value of the wedge angle θ.

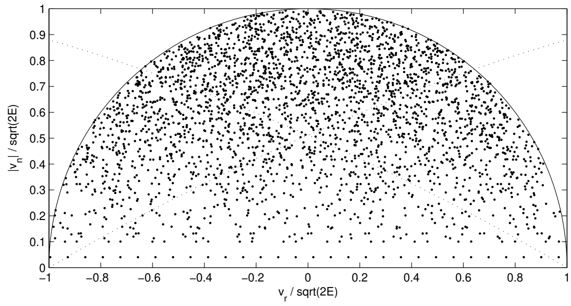
2.3 Dynamics associated with different wedge angle values

2.3.1 $\theta > 45$ deg : global chaos

For this range of θ, all the fixed periodic points turn out to be unstable. No island of stability can be found. The motion is completely chaotic. Figure 2.3 represent the phase space for $\theta = 50$ deg. In the phase space, chaos is characterized by an irregular cloud : the points seem to be disposed randomly.



(a)

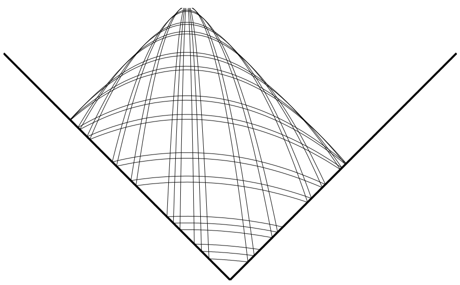


(b)

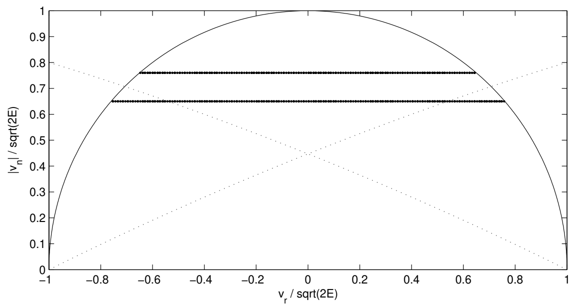
Figure 2.3: $\theta = 50$ deg : continuous trajectories *(a)* and phase space *(b)*.

2.3.2 $\theta = 45$ deg : square wedge configuration

The square wedge billiard configuration corresponds to a wedge angle of 45 deg, it is a very special case of the general wedge billiard. The motion is completely integrable and the phase space degenerates in two parallel lines (Figure 2.4).



(a)



(b)

Figure 2.4: $\theta = 45$ deg, continuous trajectories *(a)* and phase space *(b)*.

In the absence of control, the 2DOF independent motions decouples into two 1DOF independent motions : in the fixed frame $(\underline{e}_1, \underline{e}_2)$ attached to the wedge, the dynamics of the mass point $x = x_1\underline{e}_1 + x_2\underline{e}_2$ satisfies :

$$\ddot{x}_i = \frac{-\sqrt{2}}{2} g \quad ; \quad i = 1, 2.$$

In addition to period-one orbit, the system possesses a family of period two-orbits (Fig. 2.5). These periodic orbits are specific to the square wedge configuration and cannot be found for other wedge angles.

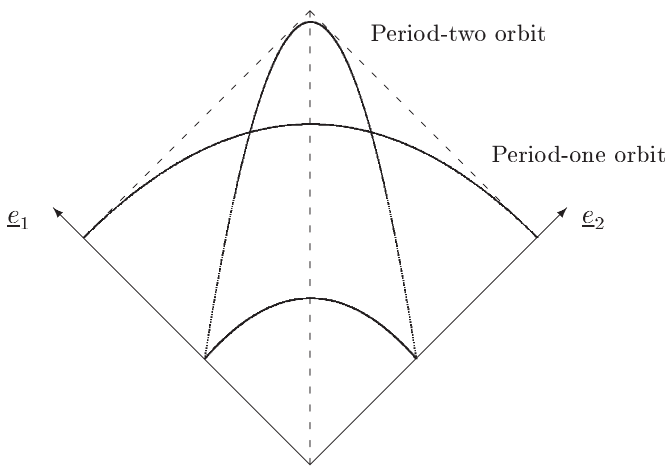


Figure 2.5: An illustration of the period-one and period-two orbits of the square wedge billiard for a fixed energy level.

2.3.3 $\theta < 45$ deg : from periodic orbits to chaos

This range of θ gives rise to a great variety of mapping structures. In [4], Lehtihet and Miller linearized the mapping around the period-one fixed point to describe the nature of the map. For $\theta < 45$ deg, the fixed point is surrounded by a near integrable subregion B* of B (Figure 2.6).

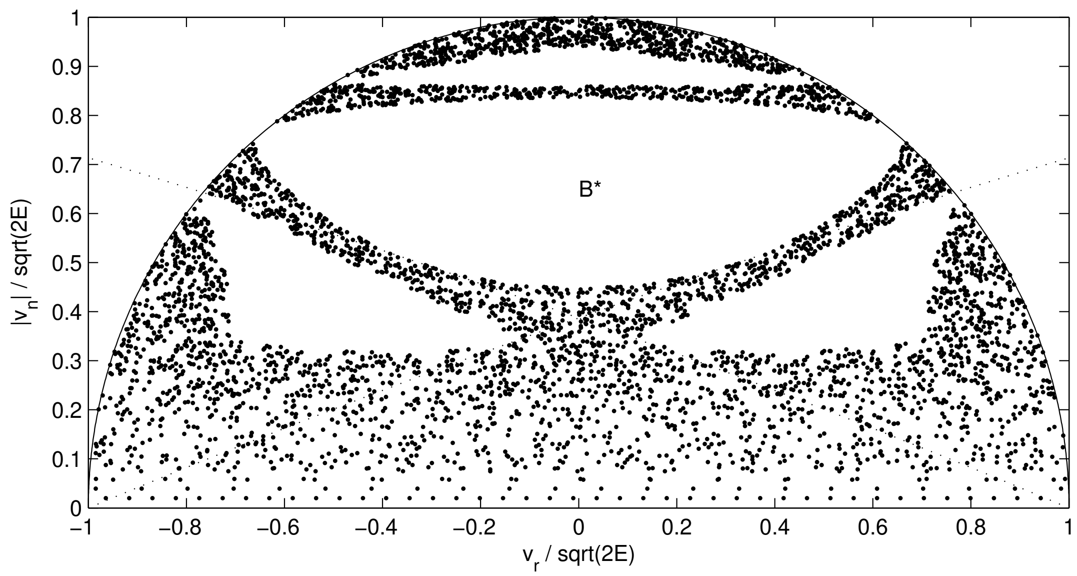
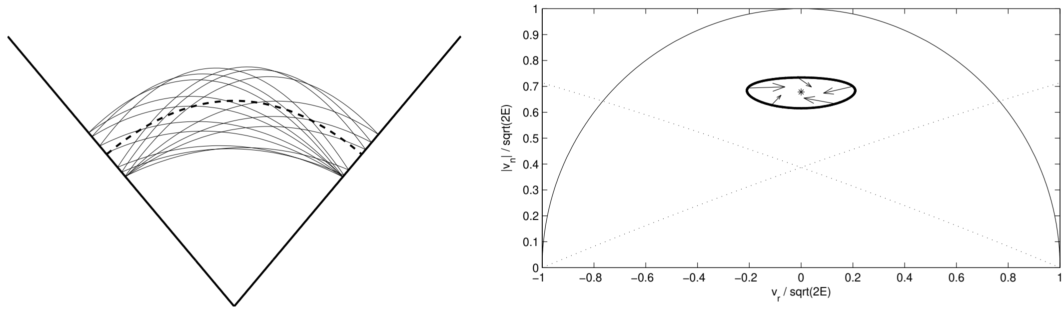


Figure 2.6: Phase space for $\theta = 40$ deg, white regions contain near integrable regions.

The map shown in Figure 2.6 is obtained for $\theta = 40$ deg. The map exhibits several near-integrable regions, each one containing a periodic point of certain sequences of transformations $\mathcal{B}_{\mathcal{A}}$ and $\mathcal{B}_{\mathcal{B}}$. These regions are a simple connected with a domain of global chaos corresponding to collisions at the wedge vertex.

Points in region B*, are transformed using exclusively the continuous transformation $\mathcal{B}_{\mathcal{B}}$. This case gives rise to a family of quasi-periodic orbits. The ball alternatively hits each edge and the trajectory of the ball stays in a neighborhood of the period-one orbit trajectory. In the state space, quasi-periodic orbits generates points on an invariant curve. Figure 2.7 represents a quasi-periodic orbit for $\theta = 40$ deg.



(a) (b)

Figure 2.7: $\theta = 40$ deg, illustration of quasi-periodic orbits near the fixed point corresponding to period-one trajectories, continuous trajectories *(a)* and phase space *(b)*. The vectors in the phase space point at the position of the next point $(v_r[k+1], v_n[k+1])$ -for some points $(v_r[k], v_n[k])$ in the phase space- corresponding to the state of the ball at the next impact.

For some wedge angle values, other fixed points exist in the phase space. They correspond to periodic orbits of period greater than one. As for period-one, these points are surrounded by a near-integrable region containing invariant curves. For example in Figure 2.6, the three second largest white areas correspond to near-integrable regions containing fixed points of period three. Plots IC_1 and IC_2 of Figures 2.8(phase space) and 2.9(continuous trajectories) represent such quasi-periodic trajectories.

For a fixed wedge angle $\theta < 45$ deg, the nature of the wedge billiard dynamics depends on the initial conditions. Figure 2.8 plot simulation results for different initial conditions $(v_r[0], |v_n[0]|)$. The continuous trajectories associated with the different simulations are plotted on Figure 2.9

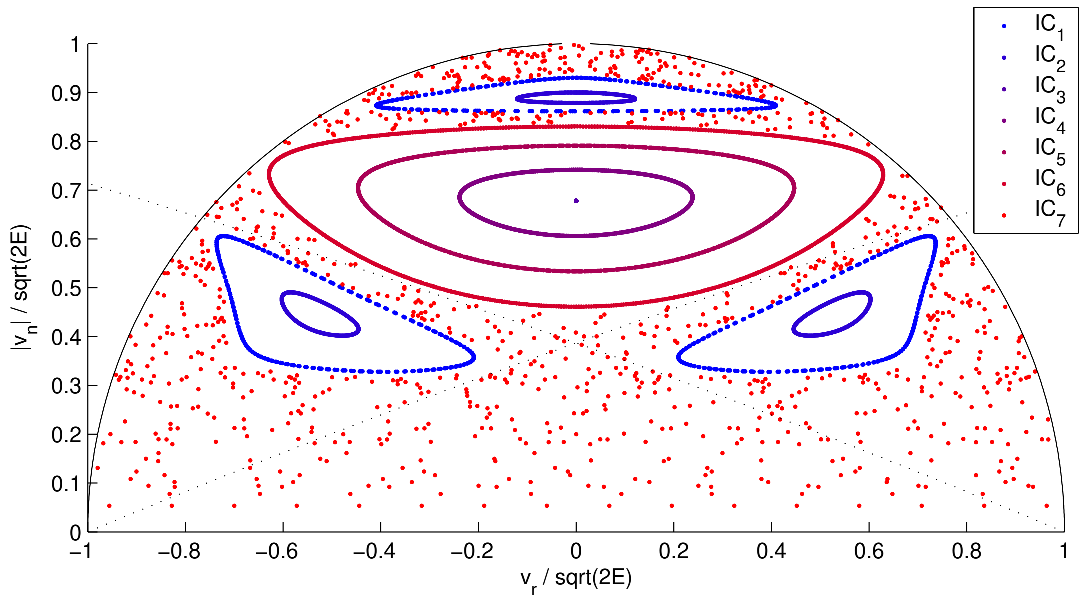
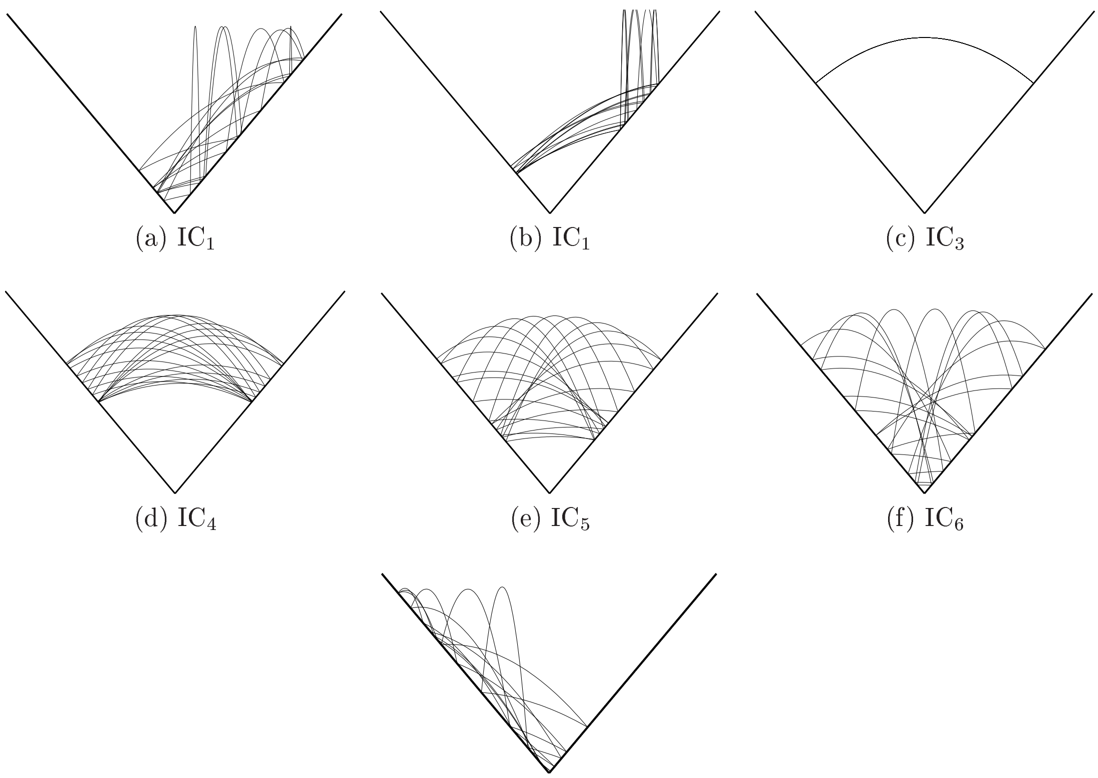


Figure 2.8: Phase space for $\theta = 40$ deg and different initial conditions. For each case, the initial state $(v_r[0], |v_n[0]|)$ is encircle.



(a) IC_1 (b) IC_1 (c) IC_3

(d) IC_4 (e) IC_5 (f) IC_6

(g) IC_7

Figure 2.9: Continuous trajectories for different initial conditions (IC). The seven plots correspond to the seven cases depicted in the phase space of Figure 2.8.

Chapter 3

Control of the wedge billiard

To implement the wedge billiard in a mechanical setup, we must take the effect of non-elastic collisions into account. A continuous-time tracking control method, based on the mirror law algorithm developed by Buehler and Koditschek ([2]) is proposed in this chapter.

3.1 Discrete Control laws and compensation of non-elastic collisions

We are interested in implementing an experimental setup of the wedge billiard. The main difficulty comes from the hypothesis of elasticity. Up to now, we have considered that the ball always hits the edge under elastic collisions, that is without loss of energy. This assumption does not hold in practice. The control of the two edges is used to correct the energy losses due to non elastic impacts.

3.1.1 Effects of energy losses at collisions

Let $\beta(t)$ and $\mu(t)$ denote respectively the angular deviation of the ball and of the edge at time t respect to the equilibrium angle of the edge. As defined in section 2.1, $\underline{r} = r\underline{e}_r$ denotes the position of the ball and $\underline{v} = v_r\underline{e}_r + v_n\underline{e}_n$ its velocity.[1].

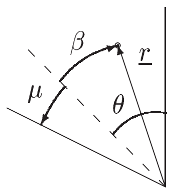


Figure 3.1: Ball bouncing on an edge.

Impact occurs at time $t[k]$ when $\mu(t[k]) = \beta(t[k]) = 0$. Assuming that the edge is rigid and not affected by the impact, the non-elastic impact rule yields :

$$r(t[k])\dot{\mu}(t[k]) - v_r^-(t[k]) = -e \cdot (r(t[k])\dot{\mu}(t[k]) - v_r^+(t[k])) \tag{3.1}$$

$$v_n^-(t[k]) = v_n^+(t[k])) \tag{3.2}$$

[1]The position of the ball must be taken from the edge rotation point. In the theoretical model presented in this section, this point is the same for the two edges and correspond to the intersection point. In the physical implementation, the edges are actuated on the top ; the angles $\beta(t)$ are different for the two edges.

where e is the restitution coefficient.

Using the polar velocity of the ball $\dot{\underline{r}} = \dot{r}\underline{e}_r + r\dot{\beta}\underline{e}_n$, equation (3.1) simplifies to :

$$\dot{\mu}(t[k]) - \dot{\beta}^-(t[k]) = -e \cdot (\dot{\mu}(t[k]) - \dot{\beta}^+(t[k])) \tag{3.3}$$

3.1.2 Discrete-time control laws

The control of the edge is used to compensate for the losses of energy at impacts. To reproduce the effect of elastic collisions, we want to impose

$$v_r^-(t[k])) = -v_r^+(t[k])) \tag{3.4}$$

Replacing this relation in equation 3.3 yields to the discrete control law of the edge :

$$\mu(t[k]) = 0 \tag{3.5}$$

$$\dot{\mu}(t[k]) = \frac{e-1}{e+1} \cdot \dot{\beta}^-(t[k]) \tag{3.6}$$

This system of equations is the discrete-time control laws of the edge required to simulate elastic collisions of the ball. The laws are evaluated at time $t[k]$ when impact occurs, $\beta^-(t[k])$ is the angular position of the ball before impact. The angular position is computed around the edge rotation point. The discrete laws are the same for the two edges, the only difference comes from the computation of $\beta^-(t[k])$, which is not the same for the two edges.

3.2 Continuous-time control law

3.2.1 Mirror law implementation

To implement the control laws in a mechanical setup, we must convert the discrete-time control laws (3.5), (3.6) in a continuous reference trajectory for the actuated edges. The mirror law proposed by Buehler and Koditschek [2] is easily adapted to the situation. It generates a reference trajectory that the actuated edge must track. The reference trajectory is chosen as :

$$\mu(t) = F_{\dot{\mu}}(x[k])\beta(t) + F_{\mu}(x[k]) \tag{3.7}$$

where $\beta(t)$ is the angular position of the ball at time t. This trajectory is valid for the time range $(t[k], t[k+2])$. The expression $F_{\mu}(x[k])$ is the discrete-time control law fixing the edge position at impact. The expression $F_{\dot{\mu}}(x[k])$ is the discrete-time control law fixing the arm velocity at impact.

The impact occurs when $\mu(t) = \beta(t) = 0$. Then, the deviation angle of the actuated edge is equal to

$$\mu(t[k+2]) = \frac{F_{\mu}(x[k])}{1 - F_{\dot{\mu}}(x[k])} \tag{3.8}$$

and its velocity to

$$\dot{\mu}(t[k+2]) = F_{\dot{\mu}}(x[k])\dot{\beta}^-(t[k+2]) \tag{3.9}$$

We see that the right member of equation (3.8) approximates the discrete expression $F_{\mu}(x[k])$ (the smaller $F_{\dot{\mu}}(x[k])$ is, the better the approximation).

3.2.2 Continuous-time control laws

The continuous laws are easily found from discrete control law (3.6) :

$$\mu(t) = \frac{e-1}{e+1} \cdot \dot{\beta}^-(t) \tag{3.10}$$

$$\dot{\mu}(t) = \frac{e-1}{e+1} \cdot \dot{\beta}^-(t) \tag{3.11}$$

Part II

Experiments of the wedge billiard

Chapter 4

The Flatland experimental setup

The wedge billiard has been implemented on a mechanical setup. It consists in the adaptation of the Flatland test-bed previously developed in the Laboratory for Intelligent Mechanical Systems of Northwestern University (Illinois-USA).

4.1 Overview

The experimental test bed used for these experiments has been developed at Northwestern University by Craig Black for his Master's Thesis entitled *Planning and Control of Planar Batting Using Nonlinear Optimization*. Illustrations and part of the description comes from Craig Black Thesis [1].

The Flatland setup (Fig 4.1) consists of an air hockey table which can float lightweight planar objects with nearly zero friction. The table is mounted on a pivoting frame which allows to vary the effects of gravity in the plane by varying the table tilt. There is an extruded 80/20 aluminum superstructure which the lights, the camera and the arms are mounted on.

The robot arms are made of lightweight aluminum links with small harmonic drive motors. For position feedback of the parts on the table, there is a color CCD camera mounted above the table. This allows to track colored blobs at 60 Hz. Analysis of the vision data is done by a separate Cognachrome vision board, and centroid information is sent by a serial link to a 300 MHz Pentium II PC.

4.2 Table, frame

The air table is a standard, commercially available air hockey game table. The factory air blower had been replaced with a larger and more powerful model. This allows to float lightweight plastic objects of moderate size with very little friction (our test disk is 3 inch in diameter and 0.5 inch thick). The table is mounted on a custom built pivoting steel frame. The frame may be tilted to orient the table at discrete angles in either direction. There are 21 settings available at 0.0, 2.5, 5.0, 10.0, 20.0, 30.0, 40.0, 50.0, 60.0, 70.0, and 80.0 degrees. The amount of gravitational force affecting the table is then the sine of this angle times full gravity.

To this steel frame is attached modular framing of 80/20 aluminum extrusion. This outlines the perimeter of the table and allows to attach the arms at any point in the plane. This frame is extended upwards to accommodate an adjustable camera mount. The framing also supports raised lighting on either side.

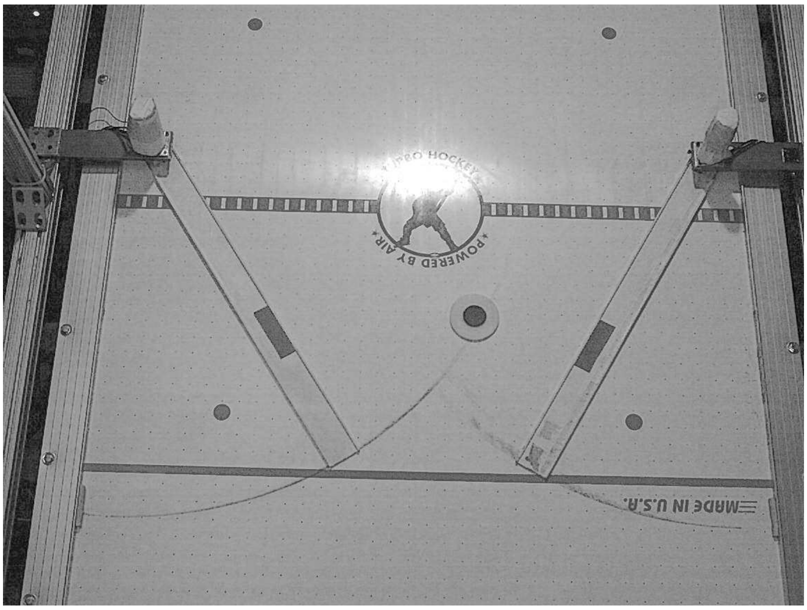


Figure 4.1: Picture of the flatland

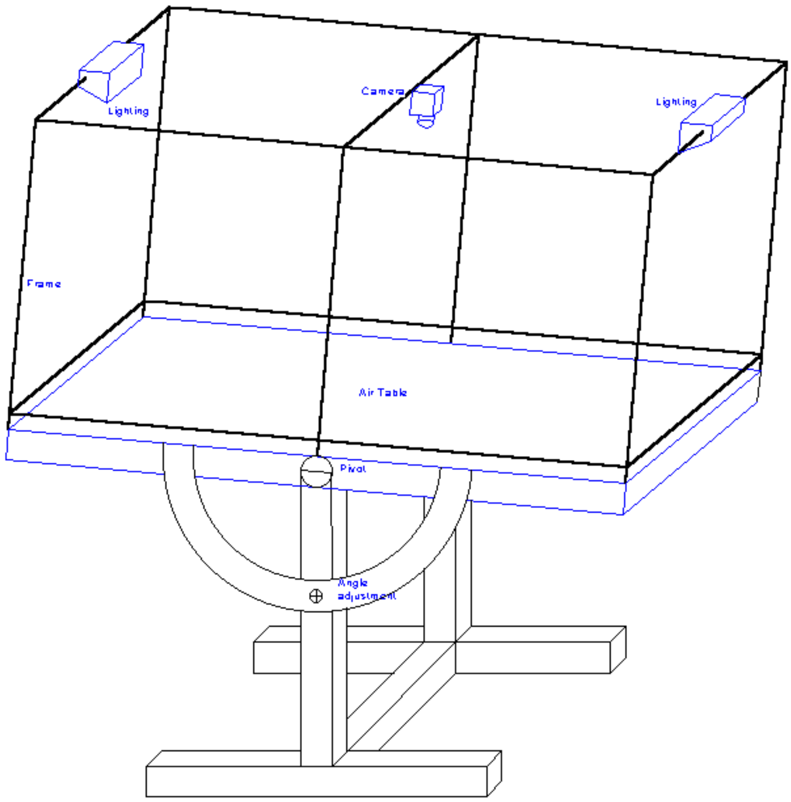


Figure 4.2: Isometric view of Flatland testbed. The puck floats on the air table, which is tiltable to vary the effects of gravity. The overhead camera tracks object position.

4.3 Vision system

4.3.1 Introduction

An overhead color camera provides data at a rate of 60Hz with a resolution of 200x250 pixels. The data are analyzed using the Cognachrome Vision System (Newton Research Labs). The Cognachrome can track up to three different colors, it outputs a standard NTSC signal of the tracked blob data for debugging, which is displayed on a television.

4.3.2 Communication protocol

The Cognachrome communicates with the computer via RS-232 serial line. Various information about the tracked object can be transmitted including position of the centroid, aspect ratio, angle, area and size (a complete list can be found in the Cognachrome Manual [9] on section 5.3.1.). The amount and type of data outputted directly affect how fast the vision board can compute and send information.

For our purposes, we wish to know the position of puck. We track the position of a color circular blob sticked in the center of the puck. To insure the tracked object is the puck, we will also compute the aspect ratio. Motor encounter only gives a relative position of the arm, we wish to use the vision to set the initial position of the arm(s): we put one long color blob with a high aspect ratio on each arm [1]. We use the two same colors as Craig Black [1] and David Choi [3] : dark blue -for the puck- and light blue -for the arm(s). We also track the frame number to insure the tracked object is the puck.

The protocol string sets theses parameters and describes how they are outputted. As suggested in [9], we chose hexadecimal transfer (good compromise between ease of debugging and efficiency). The protocol string begins with the header which we specify to be character '$' and the current frame number (mod 256). Then follows the data of the tracked object for each channel (only the first channel for our application). Capital letters are the variables the Cognachrome system outputs, the small 'x' (apart the first one) specify the format used (hexadecimal) and the number specify how many characters are displayed. Small letters are markers to make the reading easier.

Protocol String:

```
h: protocol header ="$%02xG"
a: protocol chan A ="x%04xXy%04xYs%02xS"
b: protocol chan B ="h%02xCt%02xA"
c: protocol chan C =""
```

Example of string and interpretation:

```
$a5x9d4by22afs0bhb9t12h20ta1
```

[1]We developed two methods to initialize the angular position of the arms. One is based on the vision system (accuracy of 1 degree), the other consist in pushing the arms to the table edges (accuracy of 0.2 degree).

$	:	start of a frame,
a5	:	the frame number (to insure frame rate),
x9d4b	:	x position (column number * 256) of the center of the largest blob (range 0-25600),
y22af	:	y position (row number *256) of the center of the largest blob (range 0-32640),
s0b	:	aspect ratio of the blob times 10 (10 = square or circle, 20 = oval or rectangle twice as long as it is wide),
hb9	:	x position (column number) of the first arm (range 0-200),
t12	:	orientation of the first arm (range 0-180),
h20	:	x position (column number) of the second arm (range 0-200),
ta1	:	orientation of the second arm (range 0-180),

4.3.3 Color training

1. Cut out a circular blob.
2. Train the Cognachrome system. This is done by pressing 's' in the `tpuvis` window while placing the desired color in the middle of the camera viewing area.
3. Grow or shrink the region. Tweaking is done by using '<' and '>' to shrink or grow the tracking color threshold.
4. Incrementally train the system by using 'j'.
5. Save by typing 'S'.

See Cognachrome manual [9], section 5.2.2. for more information.

4.3.4 Coordinates calibration

The vision system works out the puck position in the camera coordinates. To make the implementation and the results interpretation easier, we change the coordinates referential. The center of coordinates in the new referential (X_0, Y_0) is the intersection of the two arms.
The change of coordinates is expressed as follows :

$$\begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{pmatrix}}_{A} \begin{pmatrix} x_{cam} \\ y_{cam} \\ 1 \end{pmatrix}$$

with (X, Y) the position of the puck in the global coordinates (we work with inches) and (x_{cam}, y_{cam}) the camera coordinates. The third column of the matrix A refers to the translation of the coordinates center and the sub-matrix $\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$ refers to the rotation α and the scale factor K_S. To emphasize this, we can write the previous equation in another form :

$$\begin{aligned} \begin{pmatrix} X \\ Y \end{pmatrix} &= \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x_{cam} \\ y_{cam} \end{pmatrix} + \begin{pmatrix} a_{13} \\ a_{23} \end{pmatrix} \\ &= \begin{pmatrix} K_{Sx} & 0 \\ 0 & K_{Sy} \end{pmatrix} \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} x_{cam} \\ y_{cam} \end{pmatrix} + \begin{pmatrix} X_0 \\ Y_0 \end{pmatrix} \end{aligned}$$

To compute matrix A :

a) place the puck on the table,

b) measure the global position (X, Y) on the table (air holes are used as global coordinates as they are placed 1" from each-other)),

c) run code that outputs (x_{cam}, y_{cam}).

At least three measurements must be taken. Matrix A is computed using Matlab as follows:

```
XX = [ X1,Y1,1 ; X2,Y2,1 ; ... ; Xn,Yn,1];
xx = [ x1,y1,1 ; x2,y2,1 ; ... ; xn,yn,1];
A = XX / xx;
```

4.3.5 Performances and vision error

Stationary performances of the system had been evaluated previously by Craig Black ([1]). For any given stationary point, there is a centroid position variation of 0.12 inch in the x direction and 0.04 inch in the y direction. This is due to the camera signal and its relatively low resolution. Positions in the center of the frames have little discrepancy, while the corners tend to give larger errors.

Because of the slow vision rate (60Hz), we tilt the air hockey table of a small angle to increase dynamical efficiency. We choose to work with a slope of 10 degrees. Dynamical performances have not been computed precisely.

4.4 Actuators and arms

The two arms are made of lightweight aluminum links with small harmonic drive motors (1.1 amps max). The encoders are included on the motors and have a resolution of 1000 counts/revolution. The DC servo motors are integrated with harmonic drive gearing with a ratio of 100:1. This gives the encoders a shaft end resolution of 400 000 counts/revolution with quadrature.

The main hardware for the system is housed in a central enclosure that includes the power supplies, amplifiers, and the robot I/O. It serves as an external go-between for the central computing hardware and the robot manipulators. This includes switches for the power supplies for the motors and amps. Each channel (0-3) has a switched motor connection and encoder.

Both the amplifiers and the encoders feed into a Servo To Go, Inc. ISA bus servo I/O card. This controller card is capable of controlling up to eight axes at a time. This includes eight digital to analog converters (DACs) for eight channels of analog output. There are an accompanying eight channels of encoder input. It also has interval timers capable of generating interrupts.

Because of implementation facilities, the two arms are not actuated at the intersection point - as studied before - but on the top. The configuration is represented on Figure 4.1.

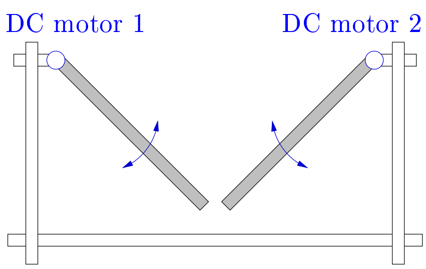


Figure 4.3: Schematic representation of the arms actuation.

4.5 Software

A software architecture, written in C and implemented in the QNX real time operating system, has been developed previously by David Choi. We use this structure and only change the C code. The program uses drivers in QNX for the Servo-to-go card (hardware manual) that were ported by Tom Worsnopp.

The flow of Figure 4.4 presents the main loop of the Flatland controller program.

First the vision data is received from the serial port with the `getc(ser1)` command. If the header character '$' is seen, all the data for that frame are dumped into an array and converted into decimal numbers. The coordinates of the puck are extracted. From these data -or from the old ones- we compute physical parameters : discrete-time parameters (as the flight times) at impact times and continuous-time (energy, apex coordinates of the parabolic trajectory, velocity components) otherwise.
Then we compute a desired position of the arm using the control laws designed and we send the orders to the motors. If the system has just started up, the command is not the continuous-time control law designed but an initialization command : the arms are push to stops to calibrate the motor encoders.

The arm control is computed in the same thread as the vision. Thus, the sampling time is the same than the vision acquisition: 60 Hz.

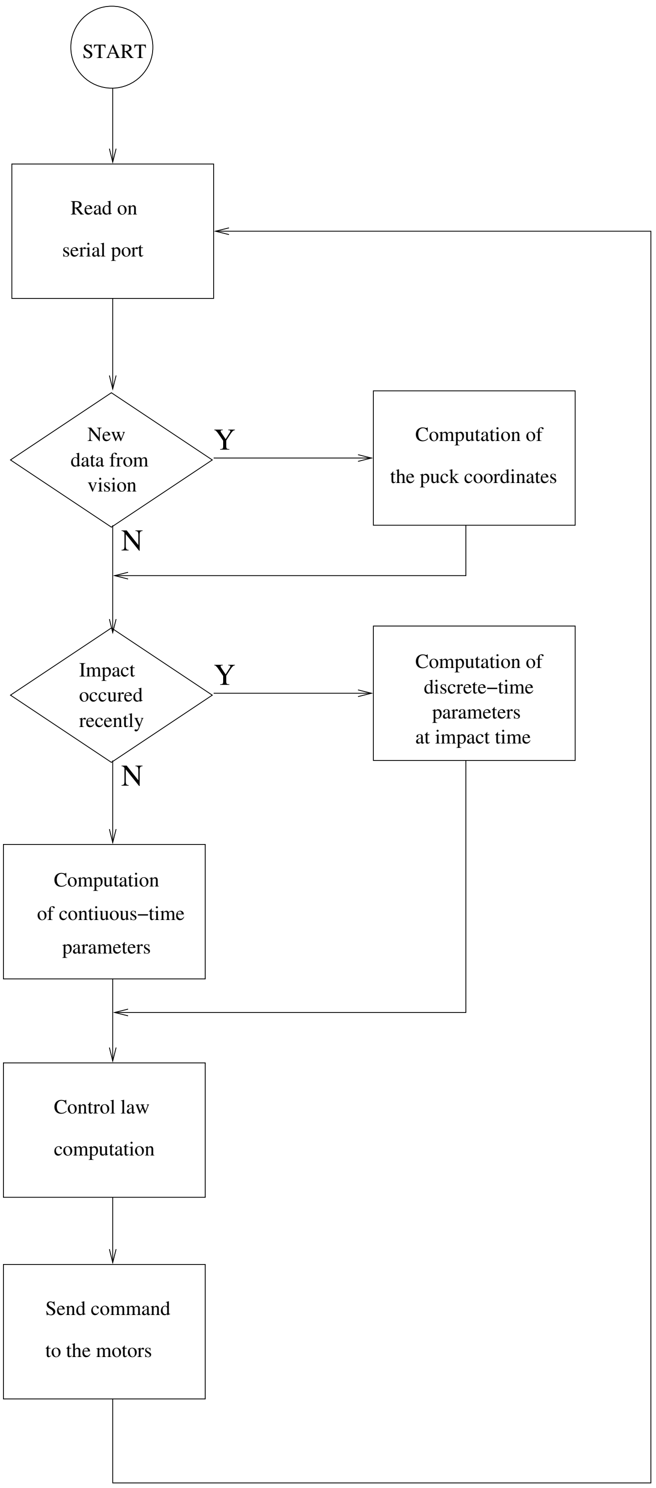


Figure 4.4: Control process flow

Chapter 5

Experiments implementation

This chapter describes some difficulties encountered with the physical implementation of the wedge billiard. First, the trajectory tracking problem of the arm is tackled. The tracking deals with the implementation of the continuous-time control laws. Since experiments have shown the sensibility to this parameter, we should pay much attention to this problem.
Then, we present the effects of the puck and the arm thickness.; two solutions are proposed to compensate for the effects. At last, we discuss the energy conservation and the restitution coefficient.
These difficulties give rise to dynamical limitations, all the properties of the theoretical model are not observed on the mechanical setup.

5.1 Motor controller - Trajectory tracking

5.1.1 Introduction

The continuous-time control laws developed in section 3.2 drive the angular deviation of the arm in order to compensate for the losses of energy at impact. The DC motor moves the arm tracking the reference trajectory; the encoder gives the orientation of the motor's shaft and is used as a feed-back information on the arm angular orientation.

The first experiments we have run on the setup show the importance of the tracking problem. To reproduce the dynamical motions observed in the model, the arm must implement the discrete-time control laws designed. We used the mirror algorithm to implement these laws in a continuous-time form. The similarities with the model depend on the accuracy of tracking at impact times.
One must be aware that not only the angular deviation of the arm matters at impact times but also the angular velocity.

For implementation facilities, we choose to control the motors at the same sampling frequency as the vision system acquisition : 60 Hz.
To obtain experimental results faithful to the simulations, the trajectory tracking should be improved in future experiments.

5.1.2 Control architecture

The control strategy developed is based on the structure illustrated on Figure 5.1. μ_{des} and $\dot{\mu}_{des}$ are the desired angular position and velocity computed from the continuous control laws, μ_{mes} is the angular position of the arm measured out from the encoder, u is the value of the voltage applied to the motor.

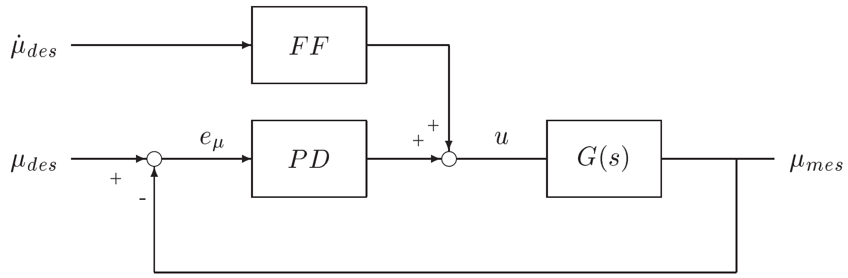


Figure 5.1: Motor controller

The control consists of a PD controller and a feed-forward action.

PD action. The PD controller outputs the sum of a proportional term and a derivative term. The input is the angular error (e_μ) between the value read from the encoder (μ_{mes}) and the command angle (μ_{des}). The process is sampled at a frequency of 60Hz and the derivative term is computed using three points. The output $u_{PD}[k]$ given at time $t(k)$ is given by

$$u_{PD}[k] = k_P e_\mu[k] + k_D \frac{4e_\mu[k-1] - 3e_\mu[k-2] + e_\mu[k]}{2\Delta t} \tag{5.1}$$

with Δt the sampling time (1/60), k_P and k_D the proportional and derivative gains.

Feed-forward action. The feed-forward action computes a voltage u_{FF} given the command velocity $\dot{\mu}_{des}$. This calculation is based on a electrical model of the actuator (motor). For a continuous current motor, the relation:

$$\dot{\mu} = k_M \cdot u \tag{5.2}$$

relates the angular velocity $\dot{\mu}$ of the motor with the power alimentation voltage u.
On the flatland setup, the arm is linked to the motor shaft. We model the arm influence by adding two differences in the model 5.2. *a)* the first effect comes from the inertia and from the friction of the arm on the table. It leads to a diminution of the gain k_M. *b)* the second difference comes from the effect of gravity. The weight of the arm is integrated by adding a term function of the angle μ. As the arm moved around its equilibrium position, we consider this term constant[1]. Thus, we have the following :

$$\theta = k_{F1} \cdot u + k_{F2} \tag{5.3}$$

with k_{F1}, k_{F2} value obtained from experiments.

5.1.3 Experimental results

To tune the gains of the controller, we worked without using the puck and the vision system. The trajectory tracked is a theoretical parabolic motion of a fictitious puck sliding freely on the inclined plane and bouncing on an horizontal arm. The trajectory is described by the following equations :

$$\begin{aligned} x(t) &= x(0) + \dot{x}(0)t \\ y(t) &= y(0) + \dot{y}(0)t - \frac{g^*}{2}t^2 \end{aligned}$$

with $g^* = g\sin(incl)$, *incl* is the inclination angle of the air hockey table.

[1] A more accurate model can be developed using a physical modeling approach. The main difficulty with this approach comes from the approximation of the friction coefficient. This term is very sensitive to the assembly, the performances change each time the arm is took apart and mounted back on. We prefer using an empirical approach.

Figure 5.2 plots trajectory tracking results for the left arm. The gains has been tuned using a 'fictitious' puck. For the experiments plotted, the command angle comes from the real system : the puck position is located with the vision system.

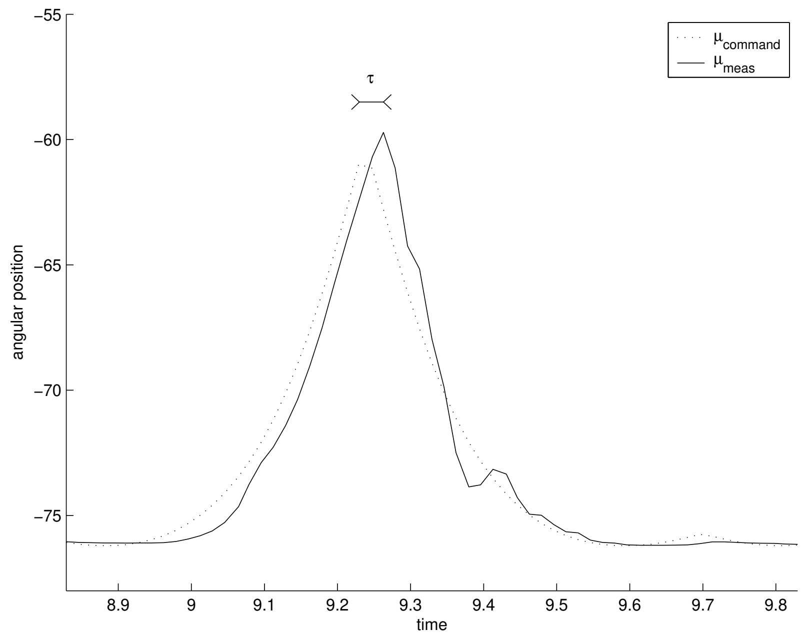


Figure 5.2: Trajectory tracking performances. $\mu_{command}$ is the command and μ_{mes}, the angular position measured with the vision system. Impacts occur when $\mu_{command}$ is maximum ($t = 9.23$).

Trajectory tracking performances.

- there is a delay τ between the command and the position measured. We have $\tau \equiv 0.02$ sec. τ is very small compare to the sampling period ($T_s = 1/60 = 0.016$),
- at impact, the angular error between the command and the real angle is roughly 1.5 deg.
- an overshoot occur after impact,
- trajectory tracking is poor just after impact,

We do not care about the two last items since we only care of the arm angular position and velocity at impact times. At impact velocity tracking is good as the slope of the two curves are similar.

Wedge-angle deviation. When comparing the experimental results with simulation data, we take into account the angular error at impact time : the simulation wedge-angle value is taken 2 deg bigger than the experimental one.

5.1.4 Improvements

Experiments have shown that the ball dynamics is very sensitive to the accuracy of the trajectory tracking. Improvements in the actual setup will increase the similarities of the experiments with the model. The first improvement should be to decrease the sampling time; we could control the motor at a frequency of 1000Hz in stead of 60Hz. Performances may also be improved by tuning more accurately the gains of the controller.

5.2 Thickness of the elements

From the theoretical approach adopted in chapter 2, the ball is assumed to be a point mass and the edge is line. The thickness of the elements are not taken into account. In a mechanical implementation, if the thickness of the elements is not considered, there is an angular deviation of the edge at impact. This deviation leads to modifications in the billiard dynamics.

This section proposes several solutions to reduce differences coming from the elements thickness.

5.2.1 Description of the system

Let $\mu(t)$ and $\beta(t)$ denote respectively the angular deviation of the arm and of the puck respect to the equilibrium edge position; $\underline{r} = r\underline{e}_r$ denotes the puck's center position given by the vision system.

Figure 5.3 represents a sketch of the situation when thickness is taken into account. The distance $d_{a,p}$ between the arm and the puck is given by

$$d_{a,p} = r \cdot \sin(\beta + \mu) - \underbrace{R_{puck} + T_{arm}/2}_{\alpha}. \tag{5.4}$$

Collision occurs when $d_{a,p} = 0$, though when:

$$r \cdot \sin(\beta + \mu) = \alpha. \tag{5.5}$$

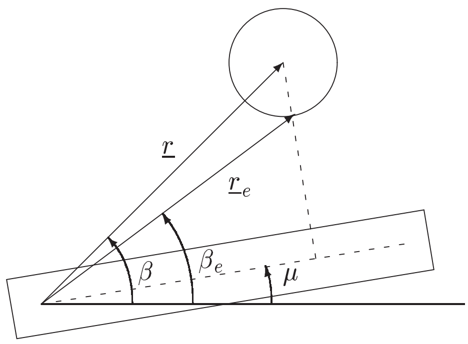


Figure 5.3: Schematic representation of one arm and the puck.

5.2.2 Influence of the thickness on the mapping

As impact occur before $\mu(t) = \beta(t)$, the continuous-time control laws (equations 3.10, 3.11) taken at impact times, do not lead to the same discrete-time laws as in the model (equations 3.5, 3.6). The continuous-time law can be written in the form

$$\mu_{th}(t) = k_1\beta(t) \tag{5.6}$$

with $\mu_{th}(t)$ the theoretical command of the arm (angular deviation respect to the equilibrium edge position) and $\beta(t)$ the angular deviation of the puck's center. k_1 is a constant parameter function of the restitution coefficient value (cf. equation 3.10). Using the relation 5.5 in the continuous law 5.6 leads to the following discrete laws at impact :

$$\mu[k] = \frac{1 + 1/k_1}{\sin^{-1}\frac{\alpha}{r[k]}} \tag{5.7}$$

Without modifications, elements thickness modify the impact map and the flight map of the wedge billiard.

Flight map. The flight-time between two successive impacts is reduced because impact occur sooner in experiments than in the theoretical model. This change influences the flight map.

One can choose to modify the arm-control in order to obtain the same experimental flight map in experiments than in the model (equations 2.6, 2.7). To obtain the same flight-time, impacts must occur when the puck deviation is null $\beta(t) = \mu_{eq}$. This condition change the discrete-time control laws, at impact we have :

$$\mu[k] = -\alpha \tag{5.8}$$

We can integrate this modification in the mirror law algorithm and compute new continuous-time control laws, but this modification will only solve the problem for the flight map, not for the impact map.

Impact map. If the arm orientation is modified at impact, the impact rule (equations 2.6) is no more valid. Indeed the problem comes from the velocity components. One can impose the orientation of the arm at impact to obtain the same impact in the model and in the experiments. Once more, this solution do not solve the problem as this solution change the flight map.

5.2.3 Solutions

Mechanical compensation. A specific design of the arms can compensate for the elements thickness. The solution proposed is represented on Figure 5.4. The arm have a form of a 'L', with the length L_{comp} (the small side of the 'L', cf sketch) equals to α ($\alpha = R_{puck} + T_{arm}/2$).

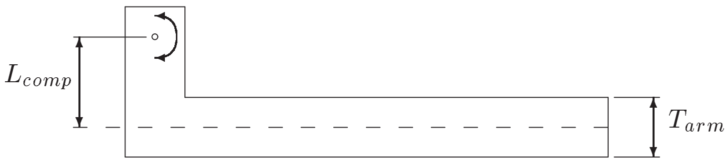


Figure 5.4: Schematic representation of an arm designed to compensate for the effect of the puck and the arm thickness.

This mechanical adaptation is the easiest solution to compensate for the thickness of the puck and of the arm. However we did not implement it on the Flatland setup. The main reason is that the setup is also use for other experiments and we did not want to apply too many mechanical modifications. We preferred software solutions.

Software compensation. In section 5.2.2, adaptation of the discrete-time control laws had been proposed. A difficulty comes from the fact that both modifications change the same discrete law. Thus it is not be possible to find a control law able to adapt the flight map and the impact map.

The solution implemented in the Flatland setup consist in taking the position of the puck not from its center point but from the nearest point to the arm[2] (Figure 5.3). The relations between the two are given by :

$$\sin(\beta_e - \beta) = \frac{R_{puck}}{r}$$

$$r_e = \sqrt{r^2 - R_{puck}^2}$$

with β, β_e the deviation of the puck computed respectively to the center point of the puck and to the point of the puck close to the arm. This solution overcomes the problem of the puck thickness but does not tackle the arm thickness.

[2]This point is different for both arms

5.3 Compensation of non-elastic collisions

The experimental setup should reproduce the motion of a puck freely sliding and bouncing under elastic collisions. The continuous-time control law designed in chapter 3 compensate for the effect of non-elastic collisions. However, as perturbations arise with the experimental implementation, the energy is not perfectly conserved.

Figure 5.5 represents the evolution of E during an experiment. Impacts occur when there is a discontinuity in the slope of y_{puck}, that is a discontinuity in $\dot{y}_{puck}$.
During this experiment, the energy of the puck varies between 1850 and 2550, around an average value of 2100. This corresponds to an accuracy of 20 %.

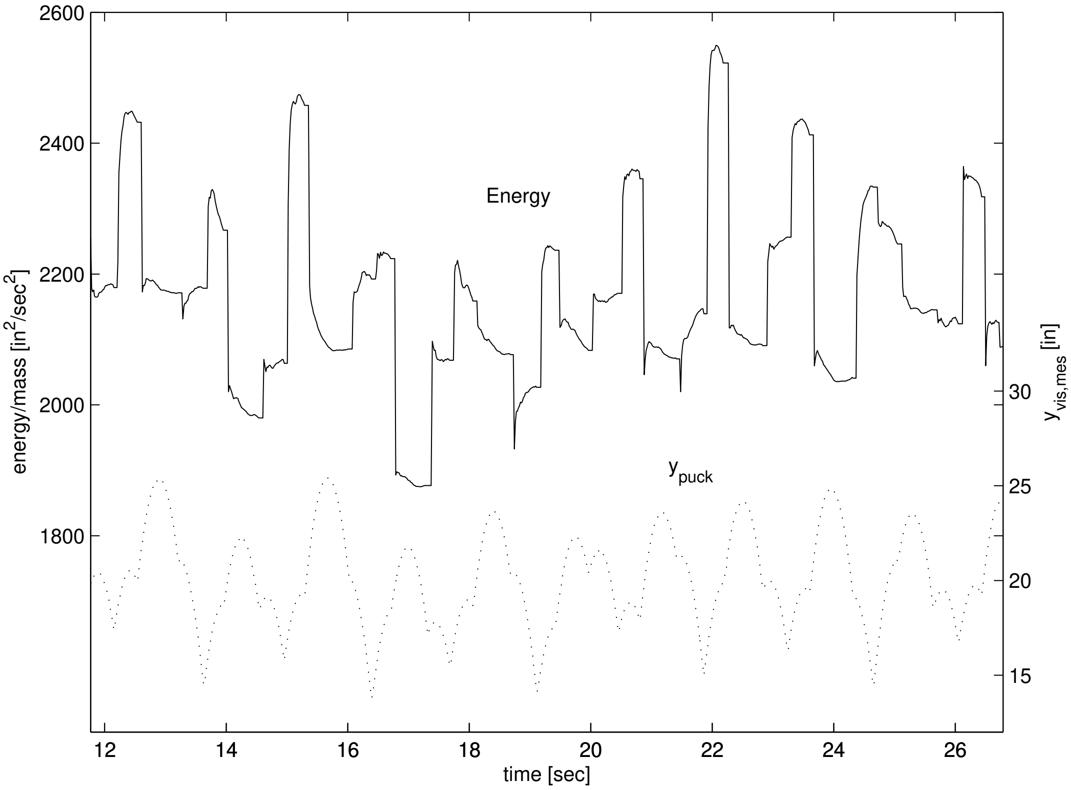


Figure 5.5: Evolution of the energy during experiments. The dash line plots the evolution of the puck height.

The accuracy on the energy conservation can be improved with a feed-back on the energy. The control strategy consists in computing the real-time energy of the puck and adapting the control law. The continuous-time control law is modified as follows :

$$\mu_{th}(t) = k_1\beta(t) + k_2\Delta E \tag{5.9}$$

with ΔE the error between the current energy and the target energy (which remains constant along the experiment).
A new source of perturbation appears with the computation of the energy. E is evaluated in real-time with the vision system, the imprecision arises when computing the kinetic energy from three points.

In practice this control strategy did not give good results. Once the trajectory tracking would be optimized, the energy feed-back may be useful to reproduce the motion of a puck bouncing under elastic collisions.

Discussion on e and implementation

The restitution coefficient e depends on the mechanics of the system, especially on the arm (edge) material and the puck (ball) material[3]. In order to maximize e the materials of the arm and of the puck should be chosen as hard as possible. But to respect the edge rigidity hypothesis, the weight of the puck must be negligible compare to the weight of the arm. The last constraint comes from the actuator limitations. The motor used in the mechanical setup is not enough powerful to allow using a heavy arm.

In the Flatland setup the arms are made of aluminum. The puck used is a classical air hockey game puck, made of plastic. The restitution coefficient is about 0.45[4](Figure 5.6).

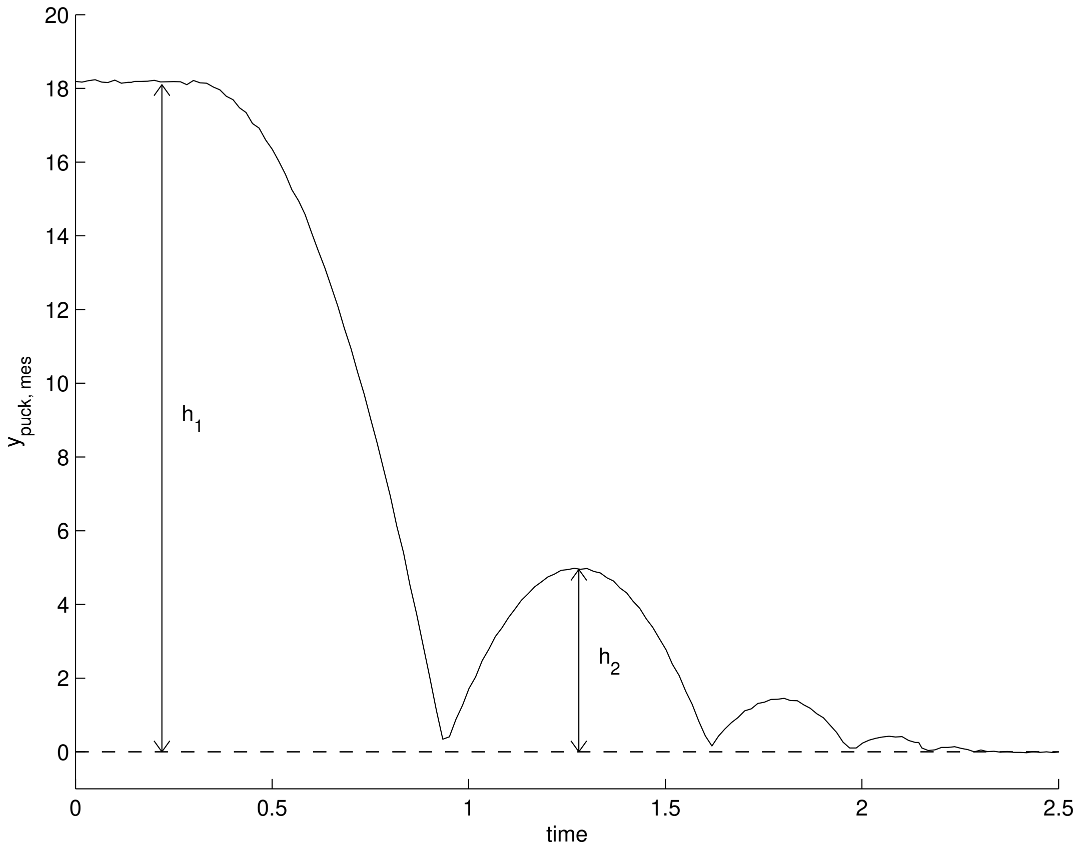


Figure 5.6: Evaluation of the restitution coefficient : $e = \sqrt{\frac{h_2}{h_1}}$

[3]When talking about the mechanical setup, we use the terms *arm* and *puck* in stead of *edge* and *ball* used in the theoretical approach.

[4]In a mechanical setup, restitution coefficient greater than 0.6 are difficult to obtain.

Sensitivity

On a first approach, one could think that bigger is e^5, better are the experiments. This not true as the sensitivity of the control law increases with the value e :

$$S = \frac{du}{de} = \frac{2}{(e+1)^2}\beta$$

5.4 Conclusion

The implementation problems encountered in this chapter limit the experimental possibilities. We are not able to reproduce on the Flatland setup, all the dynamical phenomena analyzes in chapter 2.

Because of the arm thickness and of the angular error between the command and the real position at impact, the wedge angle value is not constant during experiments. The value varies in a range around the equilibrium angle.
The square wedge billiard configuration is too singular to be implemented in the mechanical setup. We keep the two other situations, with a wedge-angle value greater than 45 deg and a value lower than 45 deg. We focus on the two cases $\theta = 30$ deg and $\theta = 50$ deg for the experiments.

Because of the little accuracy of the experiments, we are not able to reproduce complex dynamical motions. Periodic orbits correspond to an unstable fixed point in the Poincaré map; this situation is too singular to be implemented. On the other hand chaos is not very sensitive and is easily implemented. For quasi-periodic orbits, we focus on the B* region.

[5] $e \in [0,1]$: $e = 1$ corresponds to elastic collisions, if $e = 0$ all the energy is dissipated at impact, the puck does not bounce on the arm.

Chapter 6

Experimental results analysis

The experimental objective consists in reproducing on the Flatland setup the dynamical motions of the wedge billiard studied in chapter 2. The main difficulty encountered with experiments comes from noise perturbations which are inherent to experimental setups.

6.1 Graphical tools

Experimental results are marred by noise perturbations. This noise is inherent to the mechanical implementation. The sources are numerous : model uncertainties, delay from the trajectory tracking, limitation of the mechanics, ... Experimental noise yields to a poor similarity between simulation phase spaces and experimental phase spaces. The Fourier transform is then a useful tool as we will see in the sequel.

Phase space. Perturbations inherent to experimental data have an important effect on the phase space. The main properties as the space limit and the wedge vertex line remain unchanged, but dynamical phenomena associated with some wedge angle values are not observed with experimental data. Especially the problems appear with quasi-periodic motion.

For quasi-periodic motion, the state of the puck does not evolve continuously along the invariant curve. The Figure 2.7(b) illustrates the situation, the vectors plotted from some points (state of the puck at time $t[k]$) to the next point in the map (state of the puck at time $t[k+1]$). If the curve was generated continuously, the arrows would be tangent to the curve ; this is not the case here. Because of perturbations, the experimental points are not generated on an invariant curve but in a region around the average curve, along a ring. The thickness of the ring depends on the noise level. With the Flatland setup, the noise level is so important that the ring degenerates in an oval shape centered onthe fixed point.

Fourier transform. The frequency spectrum is helpful to quantify the noise level and to detect the periodicity of the trajectories. We analyze the normalized Fourier transform of vr, the tangential velocity, on the left arm. Since the function is real, the signal is symmetric with respect to the Nyquist frequency ; only the left half of the signal is plotted.

Periodic and quasi-periodic systems correspond to a(several) peak(s) in the Fourier transform; the position of the peak(s) is related to the frequency of the periodic orbits. For example, a peak at 0.2 on the X-Axis corresponds to an orbit of period 5. The noise adds a constant term at all frequencies (white noise)

Figure 6.1 represents the Fourier transform for $\theta = 40$ deg and different initial conditions. The cases depicted correspond to the cases analyzed in section 2.3.3, the continuous trajectories and the phase space representations are plotted respectively in Figures (2.9) and (2.8).

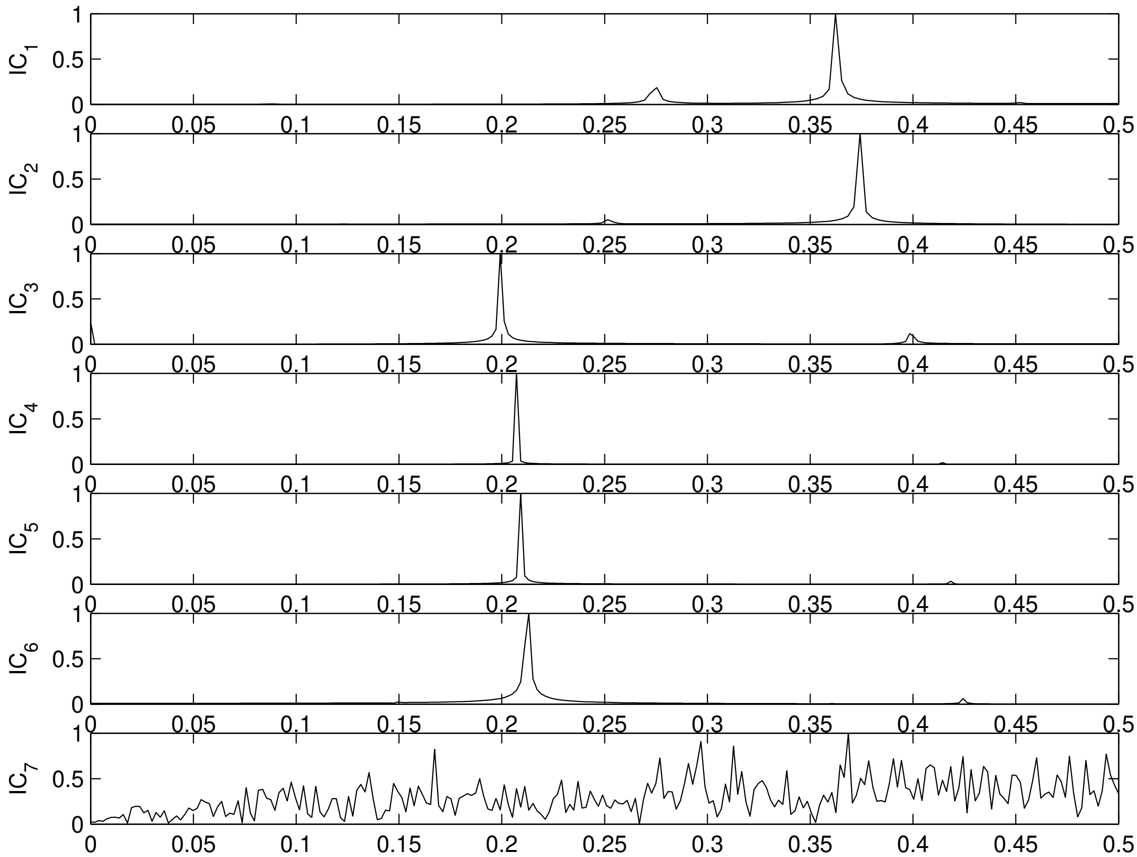


Figure 6.1: Fourier transforms for $\theta = 40$ deg and different initial conditions.

6.2 $\theta = 50$ deg : experimental results

When the wedge angle is greater than 45 degrees, the motion is completely chaotic. On the mechanical setup, the wedge angle is limited by the length of the arms. Experiments are run with $\theta = 50$ deg.

Continuous trajectories. Figure 6.2 depicts experimental data. The continuous trajectories have been measured with the vision system. The arms are represented at their equilibrium angle position [1]. The dash lines are at a distance $R_{puck}/2$ of the arms edges. Since the puck position given by the vision system is taken from its center, impacts should occur on the dash lines if the arms are at the equilibrium angle at impact times.

The continuous-time trajectories are useful to detect the average value of the arm angle at impact. However, they are not sufficient to characterize the dynamical motions.

[1] During experiments the arms are moving following the continuous-time control law based on the mirror algorithm : $\mu(t) = -K\beta(t)$. When the puck moves away from the equilibrium angle, the arm moves on the other direction, with a factor K.

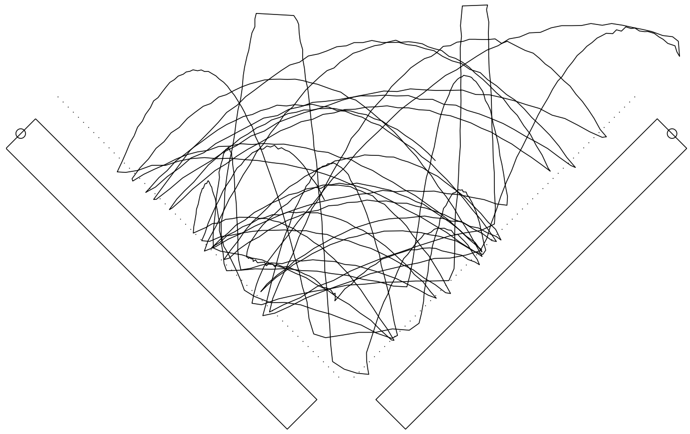


Figure 6.2: Experimental trajectories of the wedge billiard for $\theta = 50$ deg. The arms are drawn at equilibrium position. The dash lines are the locus where impacts occur if the arm is at the equilibrium angle at impact time.

Fourier transform. Chaotic motions are not easy to sustain during experiments. The problems arise when the puck does not hit the arm in the central region. Near the motor the angular error due to the arm thickness is important ; near the arm end, the rigidity hypothesis does not apply.

Thus, the data collected during experiments for $\theta = 50$ deg are limited. It can become a limitation in the frequency analysis since the most numerous the data are, the most accurate is the Fourier transform. The Fourier transform of v_r from experimental results and simulation results are plotted on Figure 6.3. The two plots have the same aspects with a white noise[2] added for the experimental data added. The noise presence was announced in the previous chapter; it comes from the experimental perturbations inherent to the the mechanical implementation.

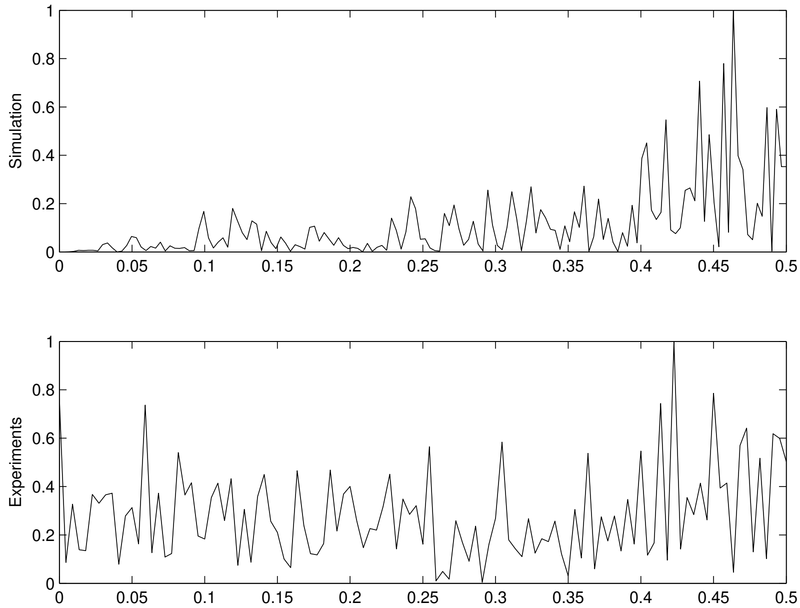


Figure 6.3: Fourier transform signal of the wedge billiard for $\theta = 50$ deg. Above are simulations results, under are experimental data

[2]The Fourier transform of a white noise correspond to a constant term over the all frequential range

6.3 $\theta = 30$ deg : experimental results

Quasi-periodic and periodic trajectories appear when the wedge angle is smaller than 45 degrees. In the mechanical setup, we focus on $\theta = 30$ deg and quasi-periodic orbits around the fixed point of period one (region B* in the phase space, cf Figure 2.6). The continuous-trajectories of experiments are plotted on Figure 6.2.

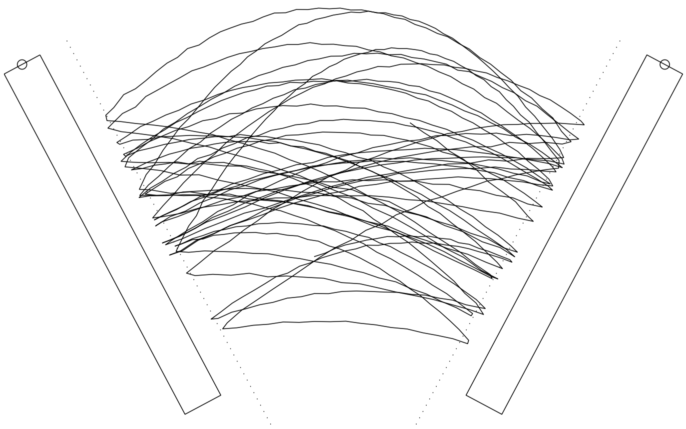


Figure 6.4: Experimental trajectories of the wedge billiard for $\theta = 30$ deg. The arms are drawn at their equilibrium position. The dash lines are the locus where impacts occur if the arm is at the equilibrium angle at impact time.

Comparison with simulations. The figure 6.5 compares the simulations results and the experimental data. The experiments are run with a wedge angle of 30 degrees; since the delay in the trajectory tracking introduces an error of 2 degrees at impact times, simulations are done with $\theta = 32$ degrees.

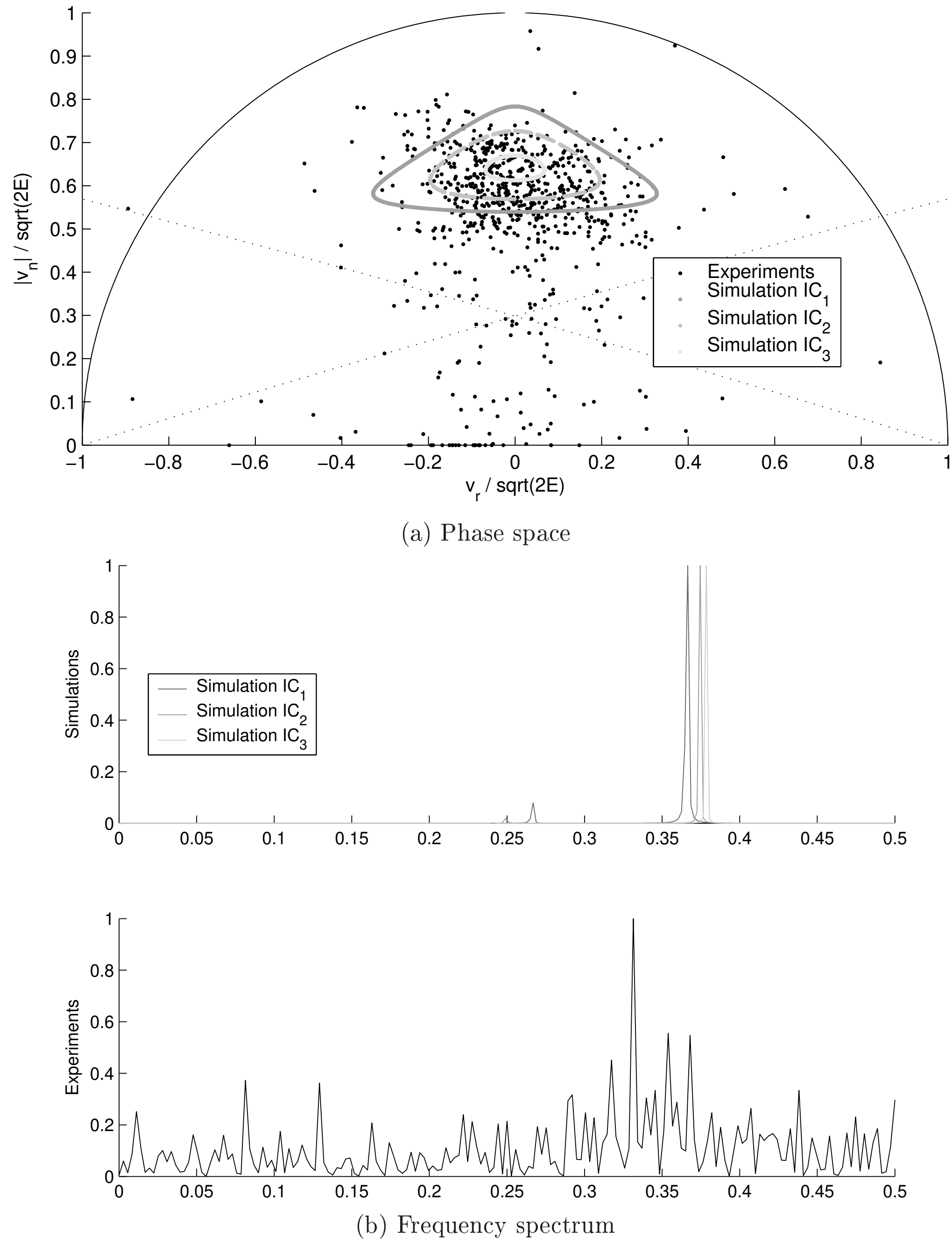
Perturbations in the model correspond to a change of initial conditions. Simulations are plotted for three different initial conditions.
In the phase space, almost all the experimental points are confined to the region B*. The quasi-periodic motion is observed on the continuous-time trajectories : the ball alternatively hits each arm. As explained in section 6.1, quasi-periodic motions with perturbations change the invariant curves. The curves degenerate in a points cloud around the fixed point of the plane.

Notwithstanding the noise inherent to experiments we observe a peak in the frequency spectrum of experimental data. However, the frequency where the peak appears does not correspond to the simulations results. This is due to the fact that the wedge angle at impact is not constant. Simulations results revealed that the peak frequency (around 0.37 for $\theta = 32$ deg) is very sensitive to the wedge angle.
In the simulations, the peak is located at a frequency 0.33 : it corresponds to a period-3 motion.

6.4 Comments

Some dynamical phenomena studied in the theoretical approach can be reproduce on the mechanical setup. Especially for 30 deg, quasi-periodic motions have been observed.

The poor similarity between the experimental results and the simulation results is due to the sources of perturbations encountered in the mechanical implementation. Before considering more complex problems as the stabilization of periodic orbits [8], improvements must be apply to the setup in order to reproduce more accurately the dynamical motion of the uncontrolled billiard.



(a) Phase space

(b) Frequency spectrum

Figure 6.5: Comparison between experimental and simulation results. $\theta_{exp} = 30\,\text{deg}$, $\theta_{sim} = 32\,\text{deg}$. Simulations results have been depicted for different initial conditions.

Chapter 7

Conclusion

This thesis is concerned with the experimental implementation of the wedge billiard configuration. The rich dynamical properties of the wedge billiard make it an interesting benchmark for stabilization of impact control systems.

When the wedge billiard is uncontrolled, various dynamical phenomena are observed as a function of the wedge angle θ [4]. When the wedge angle θ is below 45 deg, the system exhibits periodic, quasi-periodic and chaotic trajectories. For $\theta > 45$ deg, the motion appears completely chaotic. The value $\theta = 45$ deg is a special case and leads to a completely integrable system.

All the theoretical research efforts about this system [4, 8] rely on the assumption of elasticity, which means that, if friction is neglected, the energy of the ball is conserved while the edges are not actuated. This assumption is never met in practice and non-elastic impacts lead to energy losses which fundamentally alter the motion of the ball. In this thesis, discrete-time control laws for the actuated edges have been designed to compensate for these energy losses. Following [8], an adaptation of the mirror algorithm developed by Buehler et al [2] was used to convert the discrete-time control laws designed into continuous-time reference trajectories that the actuated edges must track.

The experimental setup is designed from an existing multipurpose testbed, called Flatland, developed in the *Laboratory for Intelligent Mechanical Systems*, in Northwestern University (Illinois-USA). It consists of an air hockey table which can float lightweight planar objects with nearly zero friction. This thesis presented the adaptation to reproduce the dynamical behavior of the ideal uncontrolled wedge billiard, without any stabilization objective. The control of the edges is only used in this work to compensate for the energy losses at the impacts.

Experimental results are affected by perturbations which are inherent to the mechanical implementation. In this work, we examined some of the sources of perturbations and proposed improvements. First, the trajectory tracking problem of the edges together with the implementation of the continuous-time control laws, reveals to be a crucial point and the source of the main differences between the experiments and the model.
Then, the influence of the thickness of both puck and arms are depicted. Effects were analyzed and solutions proposed to compensate for the effects and improve the similarities with the theoretical model.

The implementation difficulties encountered give rise to dynamical limitations. All the properties of the theoretical model can not be observed on experiments.
Comparisons between experimental results and simulation results are presented for two specific wedge angles. For $\theta = 50$ deg only chaotic motions were observed, as predicted by the model. For $\theta = 30$ deg, beside the chaotic motion associated with some specific initial conditions, we were able to reproduce quasi-periodic orbits predicted in the model.

The implementation of the dynamical behavior of the uncontrolled billiard studied in this thesis was a a first step towards the implementation of more complex robotics systems. Since the physical implementation reveals the high sensitivity of the dynamical motion to perturbations, it is important to further improve the performances of this single problem before dealing with stabilization of periodic orbits, which will be the next step in the wedge billiard implementation problem.

Appendix

Appendix A

Directions for use

The following steps explain how to run an experiment on the Flatland setup.

1. Plug in power to lights, air hockey table, amplifiers, television (for debugging), Cognachrome vision system fan, and CCD camera. Leave Cognachrome power unplugged.
2. Turn on the Motor Power, Amp Power and DAC 0 and 2 on the front face of the amp box.
3. Boot the computer in Windows 2000 (boot option1).
4. Start the program C:/Program Files/arc/win32bin/arc.exe.
5. Plug in power to the Cognachrome (will boot up vision software).
6. In the `tpuvis` window, type “`q`” to stop recording.
7. Check prototype strings by typing “`p`”. This is the format in which the data are sent to the computer. It only need to be checked once. The protocol strings should match those of section 4.3.2. To change the protocol, type the letter of the channel that you wish to modify and type in the new prototype string. If you do not need to change anything, just press Enter. After any changes, remember savin by typing “`s`”.
8. Type “`r`” to begin recording. A telnet window should start streaming vision data.
9. Reboot the computer in QNX (boot option 2, default).
10. In the terminal, set the correct baud rate by typing “`stty baud=38400 </dev/ser1`”
11. Remove the puck of the camera field and position the arms near their equilibrium position.
12. Compile and run code. Include curses and math librairies in the compile line (these are standard and no additional files are needed). Here is an example of compiling and running code `twoarms.c` :
    - compile : “`qcc twoarms.c servotogo.c -lncurses -lm -o twoarms`”
    - run : “`./twoarms`”

    The screen display that the Servotogo has been initialized and that the different threads have been started.

13. First the system initializes the encounters : the arms move to the table edges until stop. During this phase, put the puck between the arms in the camera field and hold it in such a way that it is seen by the camera.

14. When the first phase is finished, the control-law based on the mirror algorithm is applied, the arms will move near their equilibrium positions. When they are at stop, throw the puck to one arm in order to launch a sustained trajectory.
    If the puck falls above the arms, restart this step.

15. press "`x`" to stop the experiments

Bibliography

[1] C. K. Black, *Planning and control of planar batting using nonlinear optimization*, Master's Thesis, Northwestern University (2000)

[2] M. Buehler, D.E. Koditschek and P.J. Kindlmann, *Planning and control of robotic juggling tasks*, InternationalJournal of Robotics Research 13 (1994), no.2, 101-118.

[3] D. Choi, *Non prehensible underactuated self-assembly by designin stable limit sets*, `http://www.mech.northwestern.edu/lims/people/murphey/david/399.html`

[4] Lehtithet, H.E. Miller, and B.N., *Numericalstudy of a billiard in a gravitationnel field*, Physica 21D (1986), 93-104.

[5] K. M. Lynch and C. K. Black, *Recurrence, controllability and stabilization of juggling*, IEEE Trans. on Robotics and Automation **17**(2001), no. 2, 113-124.

[6] E. Ott *Chaos in synamical systems*, Cambridge University Press (1993).

[7] R. Ronsse *Analyse et Controle du Bouncing Ball.*, Engineering Thesis, University of Liege, (2002).

[8] R. Sepulchre and M. Gerard, *Stabilization of a planar juggler model*, submitted CDC 2003.

[9] A. Wright, R. Sargent, C. Witty, J. Brown, *Cognachrome vision system user's guide*, Newton Research Labs, edition 2.0.