REPORT NO. UIUCDCS-R-88-1469 UILU-ENG-88-1775

A QUALITATIVE APPROACH TO RIGID BODY MECHANICS

by

Paul E. Nielsen

November 1988

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

A QUALITATIVE APPROACH TO RIGID BODY MECHANICS

BY

PAUL ERIC NIELSEN

B.S., University of Illinois, 1981

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1988

Urbana, Illinois

©Copyright by
Paul Eric Nielsen
1988

Abstract

In order for a program to interact with the world as well as people do, we must provide it with a great deal of commonsense about the way things work. Reasoning about the geometric interactions and motions of objects is an important part of that commonsense. Some of the most complex problems we solve involve reasoning about mechanical devices, such as gears, cams, and clocks.

Qualitative mechanics is the symbolic analysis of the motions and the geometric interactions of physical objects. This thesis describes a theory for analysis of rigid body mechanisms, an important subset of qualitative mechanics problems. This theory has been implemented and tested on several mechanisms including a mechanical clock. Beginning with drawings of the parts involved we compute a discrete symbolic description showing changes in position and motion of the parts of the mechanism as well as its global behavior.

Acknowledgements

First and foremost I want to thank my advisor, Ken Forbus, for initiating and directing this research. I am grateful for his help and support throughout. His guidance, insightful comments, and encouragement is very much appreciated.

Thanks to Donald Michie for encouraging my initial interest in artificial intelligence and providing a place for that interest to grow.

I want to thank Mehdi Harandi for serving on my committee and for his support in the early days of my graduate work.

Thanks to the other members of my committee, especially Gerald DeJong and Steven Lu for their insightful and useful comments.

A special thanks to all the members of the Qualitative Reasoning Group: Dennis DeCoste, Brian Falkenhainer, Boi Faltings, John Hogge, Hyeonkyeong Kim, Gordan Skorstad, Janice Skorstad, and Barry Smith. They are friends, colleges, critics, and guides. The many discussions we had were invaluable.

Thanks to Betty Cheng for her thesis macros and for being a delightful officemate and friend.

Thanks to It's About Time Clock Repair and The Time Museum for providing an exciting background in horology.

Finally, thanks Lucja for the love and support you have given me.

This research was funded by Office of Naval Research, Contract No. N00014-85-K-0225 and by a University of Illinois Cognitive Science/Artificial Intelligence fellowship.

Table of Contents

List of Tables

List of Figures

Part I

Introduction

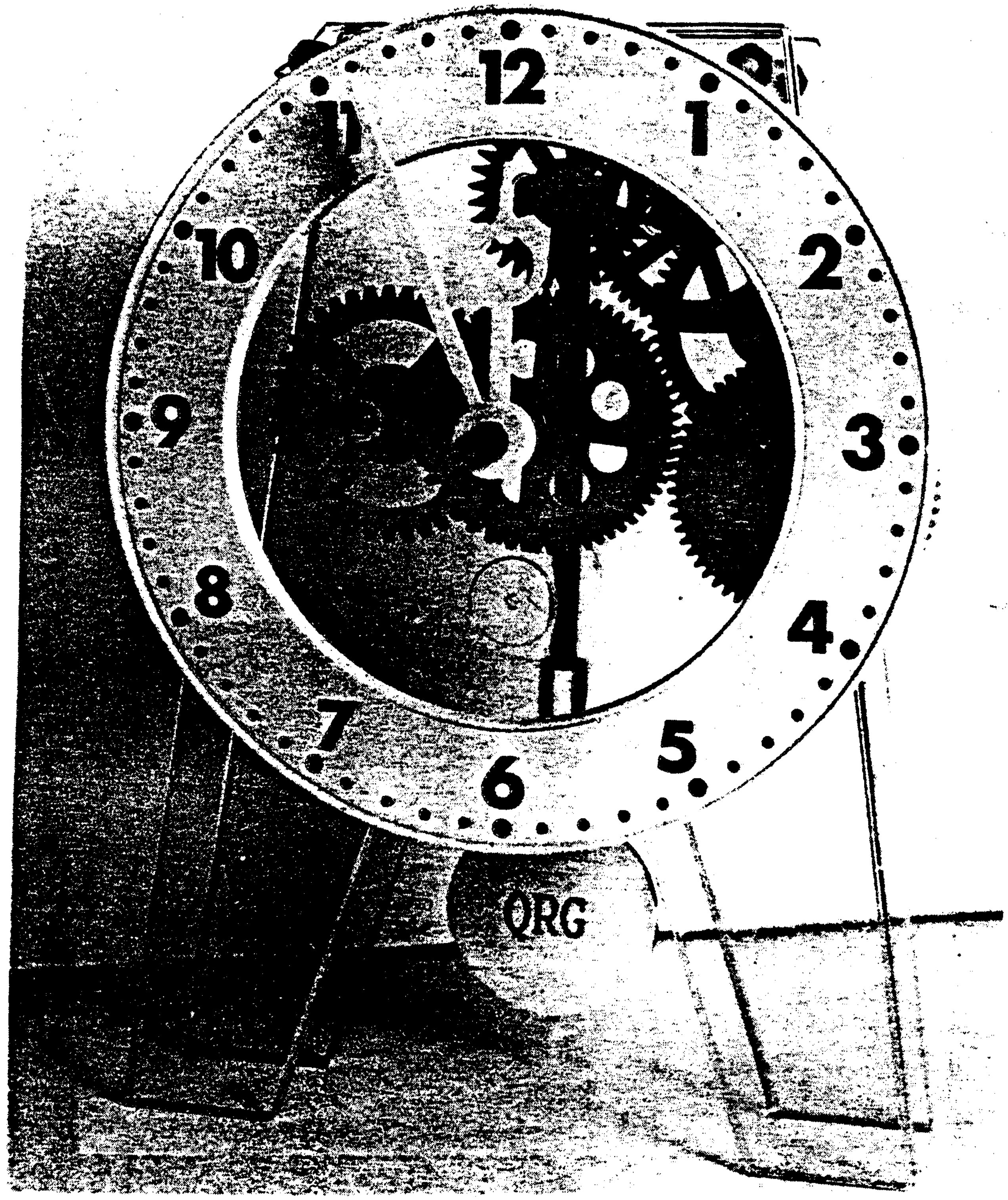


Figure 0.1: The QRG clock

Chapter 1

Introduction

Mechanical problems pervade our everyday experiences. Our interactions with solid objects include tool manipulation, opening doors and windows, and rearranging objects. These actions involve such operations as grasping, pushing, pulling, and turning. People vary greatly in their ability to understand mechanical behavior. However, most people understand that unsupported objects fall, round objects roll, pushed objects move, and solids cannot pass through solids. People are able to visualize how the parts of a mechanism will interact with each other and are fascinated when an unusual behavior is encountered. Thus, a theory of mechanical reasoning is fundamental to understanding common sense reasoning.

Classic approaches to mechanics (Newton, 1686; Reuleaux, 1876) rely on extensive mathematical formulations which are incomprehensible to most people. More recent work in symbolic approaches (Gelsey, 1987; Joskowicz, 1987b; Pu & Badler, 1988a; Stanfill, 1985) rely on shape recognition or *a priori* functional knowledge and are unable to reason about new, unusual, or variant interactions.[1].

Qualitative Mechanics (hereafter QM) is a generative model of mechanical analysis which determines behavior from surface geometries of the components and descriptions of external forces affecting the system. This thesis presents a combination of metric and symbolic approaches to mechanics which is general enough to describe the behavior of unknown rigid body devices in a discrete, symbolic manner. We do *not* rely on functional descriptions nor on shape recognition. The important features of this research are reasoning about multiple interacting objects, each of which may be only partially constrained and each of which may be in motion, and the ability to exhaustively generate potential behaviors of a device. Our theory completes work begun in kinematics by Faltings (Faltings, 1987a; Faltings, 1987b) and combines it with theories of qualitative dynamics, based on work by Forbus (Forbus, 1984).

This theory has been implemented as a system of computer programs which take as input diagrams of a mechanism's components and produce a behavioral description of the mechanism. Specifically, we take drawings of fixed motion, rigid objects; determine how they will interact by performing collision tests; partition these interactions into equivalent behaviors; combine these behaviors for all components in the mechanism; propagate external forces acting on the system; and thus produce an envisionment showing possible changes in motion and position throughout the mechanism (figure 1.1).

[1]We return to these alternative approaches in chapter 10

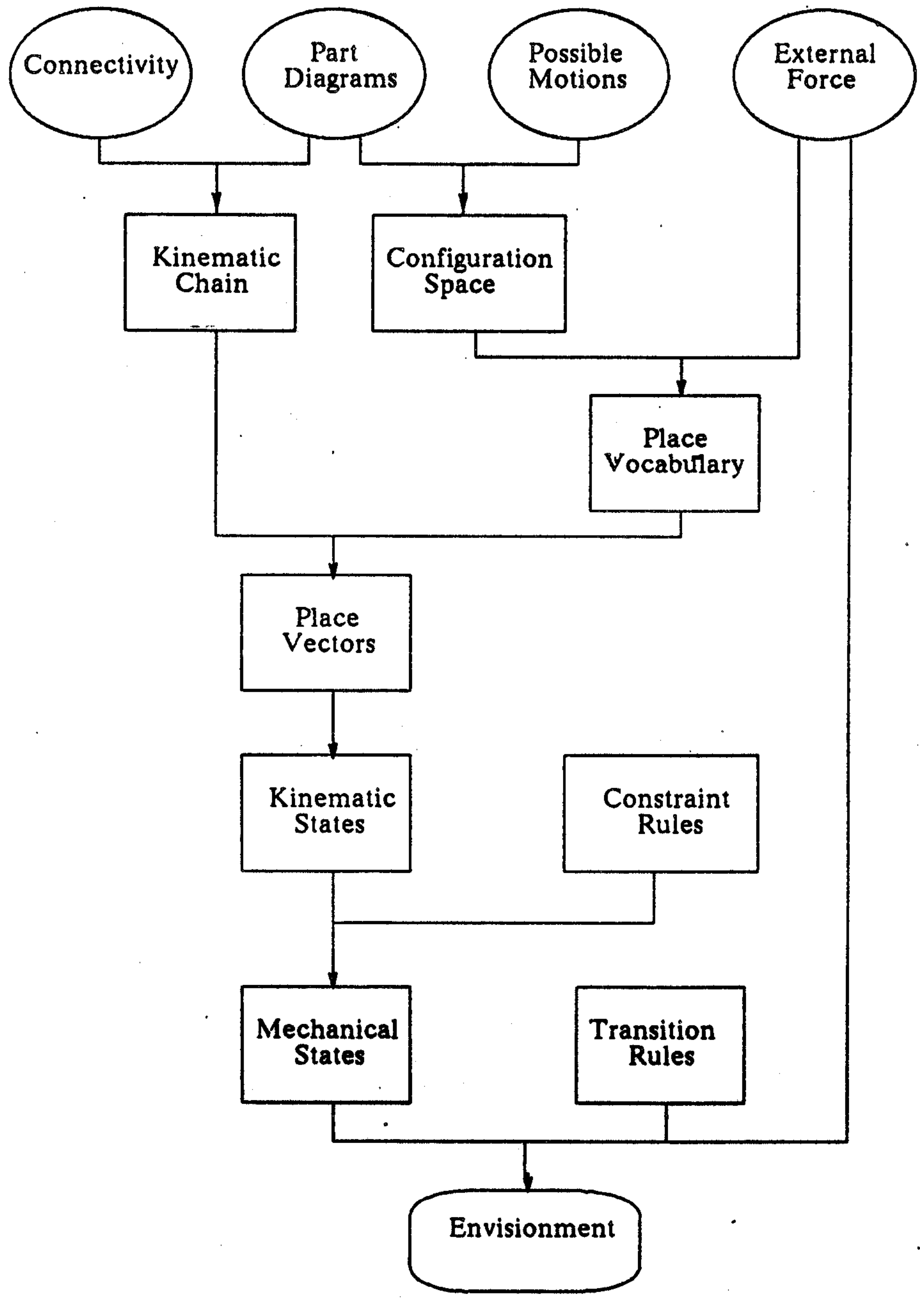


Figure 1.1: Information flow of Qualitative Mechanics

Understanding the operation of mechanical clocks has been an open problem in qualitative physics. Consequently, throughout this thesis we use the mechanical clock as a source of examples to illustrate the stages of analysis. In section 9.1 we will present a complete analysis of the mechanical clock shown in figure .1. However, these techniques are applicable to a wide variety of problems in the mechanisms world. The mechanisms world consists of machines, and a machine is "any device consisting of two or more resistant, relatively constrained parts which may serve to transmit and modify force and motion so as to do work"(Cowie, 1961).

1.1 Introductory Example

One goal of this research is to produce a complete qualitative analysis of several mechanical clocks. A common clock escapement is the *recoil* or *anchor* escapement. The following descriptions show how horologists[2] describe the behavior of this escapement.

Horologists call these drawings *action sequences*. In addition to gaining a basic understanding of this escapement for further examples, the reader should observe in these descriptions that no rate information is necessary – only the direction of motion. Second, from an infinite number of continuous, possible configurations only a few are necessary to characterize the behavior. Finally, the only parameters mentioned are motion, impulse or push, shape, and connectivity. In the following chapters we will develop a theory of mechanics which allows reasoning about mechanical devices from these types of description rather than using specific rate information, force magnitudes, or continuous configuration locations.

Landies (Landies, 1983) describes the action of this escapement as follows (Figure 1.2):

> (1) Tooth is about to leave exit pallet, *b*. (2) Tooth falls on face of entry pallet, *a*. (3) As entry pallet moves in, it pushes the scape wheel, *c*, back (recoil). (4) As entry pallet withdraws and releases tooth, the scape wheel, resuming forward motion, gives impulse through the tooth to the pendulum.

de Carle (de Carle, 1975) describes the action of this escapement as follows (Figure 1.3):

> The escape wheel is rotating in a clockwise direction. The pallets are fixed to the pallet arbor and the crutch is fixed to the pallet arbor. The crutch controls and gives impulse to the pendulum. As the pendulum swings to the left it allows the tooth *A* of the escape wheel to slide along the impulse face *B* of the pallet pad and at the same time imparts impulse to the pendulum. Eventually the tooth *A* drops off the pallet pad and the tooth *C* drops on to the pallet pad *D* and as the pendulum continues to swing to the left this locking becomes deeper and by reason of the curve of the pad the escape wheel is made to recoil. When the pendulum has reached the end of its journey and starts to return the escape tooth *C* will then give impulse to the pallet and so to the pendulum. This cycle is repeated on the pallet pad *B*. Fig. 1.3 (1).– Pendulum about to swing to the left. Fig. 1.3 (2).–Tooth about to escape from left hand pallet and drop on to right hand pallet. Fig. 1.3 (3).–Impulse imparted to right hand pallet.

[2]Those who study time or timepieces.

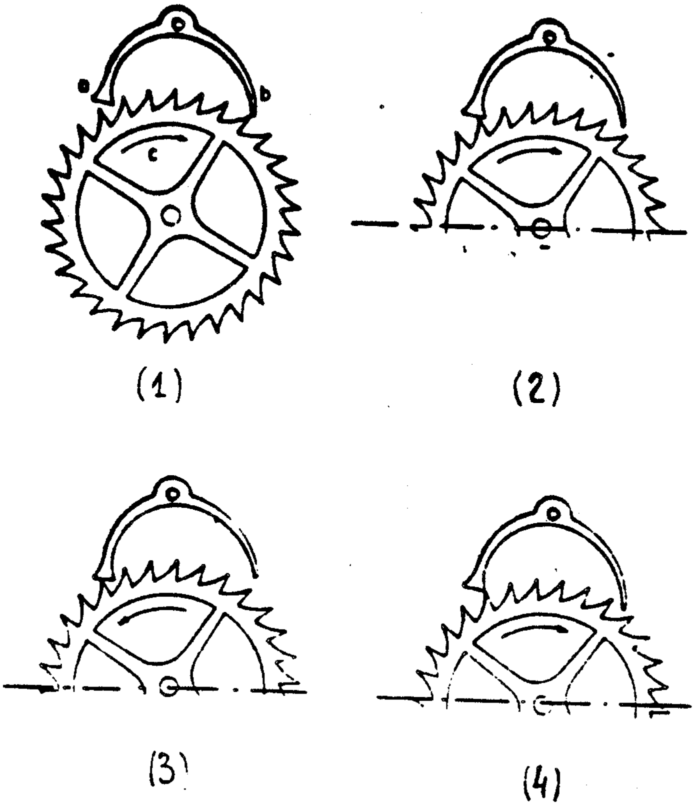


Figure 1.2: Anchor escapement 1

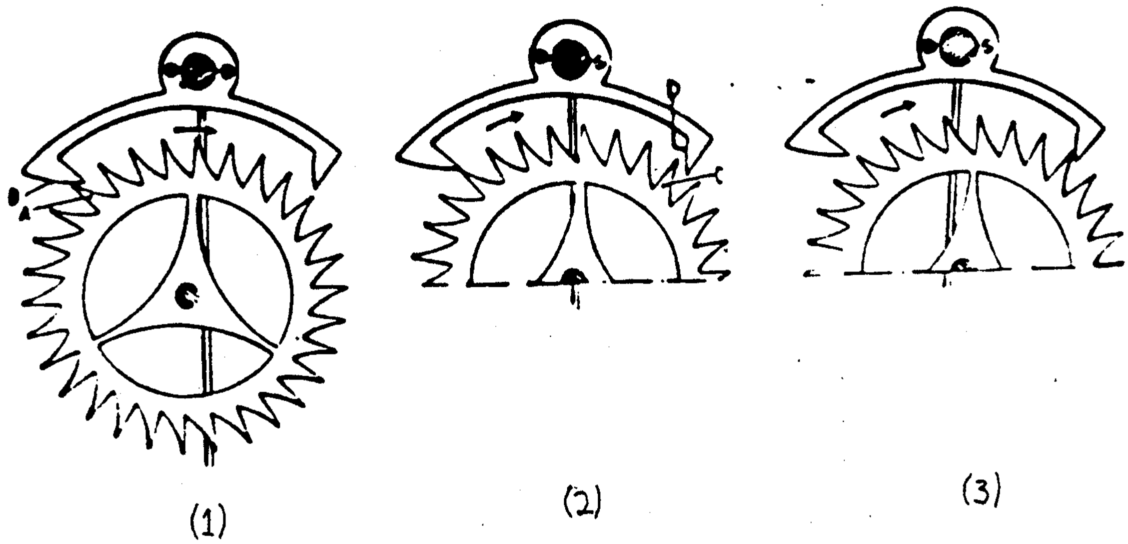


Figure 1.3: Anchor escapement 2

In following this example the reader may have mentally animated the parts of this escapement and perhaps discovered further potential behaviors, such as the pallets jammed between the teeth or the pallets clear of the teeth. We want the behaviors produced by our analysis not only to include the intended behavior but also other physically possible behaviors. These may be eliminated by further analysis or may indicate design flaws.

1.2 Overview

This thesis contains four major parts: introduction, theory, practice, and discussion. The introduction (which you are currently reading) provides familiarity with the problem domain, shows the results we expect, and examines the potential usefulness of these results. The theory portion describes the principles of qualitative mechanics. The practice portion describes two implementations of these ideas and the result of our analysis on several extended examples. In the discussion section we examine related work and possible extensions.

The theory section begins with a description of the spatial concepts necessary for describing constrained, mechanical motions. This requires a qualitative representation of direction and selecting a global reference frame for this representation. We assume this frame is based on Cartesian coordinates. Since we are concerned with the effects of motion on location, our representations of location must be allow the prediction of the next possible configurations of objects after movement. Once location has been represented we focus on the methods for determining changes in locations and how these may be influenced. We consider the effects of motion on position, position on motion, and the interactions between the two.

The overall achievement of this theory is a technique for envisioning mechanical systems, from geometric and dynamic influence descriptions, without requiring prior knowledge of the functionality of the system or its components. Complete envisionments produced by this technique are an exhaustive description of the device's possible behavior.

The practice portion describes two implemented programs which perform qualitative mechanical reasoning. The first of these, `Alex`, reasons about constrained motions and external forces on geometries. `Alex` is capable of analyzing many mechanical problems and performing stability analysis of general rigid bodies. Examples are taken from mechanisms and the blocks world.

The second program, `Clock`, adds the ability to reason about changes in location basing its analysis on discrete, symbolic representations of space. To implement reasoning about mechanisms at the system level, we employ abstractions that increase description tractability while minimizing information loss. This allows us to provide both focused reasoning at the component level and general behavioral descriptions at the system level. In section 9.1 we demonstrate this program on a complete mechanical clock.

The discussion portion begins with an overview of related work. That chapter covers previous AI work in kinematics, naive physics, and qualitative physics. Section 11.1 provides a summary and conclusions. Finally we discuss the potential applications of this work and future work.

1.3 Motivation

Though a variety of approaches to mechanical analysis exist, there are many reasons to prefer a qualitative model. In this section we discuss some of the advantages of qualitative models over mathematical models, and examine how other artificial intelligence systems will benefit from qualitative, model-based reasoning. We do *not* claim that qualitative analysis by itself is powerful enough to completely analyze every rigid body mechanical device. But even when a more precise model is needed later there are advantages to first performing a qualitative analysis.

1.3.1 Less Complex

The inputs required for qualitative analysis are less complex than numerical analysis. Qualitative representations require less precise information and fewer input equations. This allows reasoning to begin before all relevant parameter values are chosen. Numerical analysis requires rate and magnitude information for forces and motions. It also requires either exact matching when comparing locations or a fuzzy value, which may give false matches. QM needs no rate or magnitude information, and provides a way of comparing location without exact matching, based on relevant, behavioral expectations.

Qualitative models allow the user to work at a higher level of abstraction. Most important decisions in the design process should occur early, but designers may neglect these stages and rely more on the mathematical formulations because of the availability of tools to assist them(Lu, 1987). Qualitative representations enable computers to become more active in these early stages of the design process.

Qualitative reasoning can guide more precise analyses. Even when qualitative reasoning is insufficient, it provides an explicit representation of ambiguities that narrows the need for more precise information (de Kleer, 1975). Rather than mathematically formulate the entire mechanism we may be able to focus examination on a few parameters over a limited range.

1.3.2 Exhaustive

Qualitative solutions may be generated exhaustively because the problem space is finite. This means that all states can, in principle, be generated and all paths explored. Numerical solutions have infinite solution spaces. A numerical simulator only follows a single path through state space while total envisionments show all paths. Qualitative envisionments rule out impossible behaviors and transitions. In numerical simulation one can never be certain that a behavior *cannot* occur, only that it can occur.

Generating solutions exhaustively is especially important in verifying reliability. One wants to know every way a device could fail (with respect to the assumed geometry), and how to detect or prevent it. In design problems one could determine that any mechanism of a given form is impractical.

1.3.3 Generative

A generative approach is essential for mechanical analysis. While there are advantages to maintaining libraries of common mechanisms, they are not sufficient for innovative design.

Approaches that rely entirely on part recognition or functional descriptions are severely limited for reasons which will be discussed here. First, consider the individual objects. We can never have a complete set of part behaviors because there are an infinite number of possible variations in the shapes of objects.

Second, even if we restrict consideration to a subset of all possible objects, it is the interactions between objects which determines their behavior, not just the object itself. For example, gears only exhibit rotation transfer behavior when their teeth mesh properly with other gears; a gear on a flat surface would behave as a wheel; and that same gear might serve as a spacer to keep narrow objects a fixed distance apart. Considerations of the interactions between objects would make an enumeration of all possible behaviors grow exponentially with each object added.

Shape approximations are insufficient because detailed distinctions between shapes may be crucial. For example, a small hole in an otherwise flat surface will typically be insignificant, but will affect its interaction with a peg. Small notches in surfaces may be overlooked when rolling, but may cause the objects to catch when sliding.

Finally, we cannot even restrict consideration to a small number of interacting objects and provide functional descriptions of the interactions. There are arbitrary possible perturbations in the separation between object's surfaces that will affect their behavior. For example, some gears will jam if they are too close together even if the teeth are aligned and mesh properly.

A generative approach allows the analysis of new, unusual, and variant interactions. It also allows multiple perspectives to be used in problem analysis. For example, a coarse approximation may be computed to determine overall behavior and a more detailed analysis may be used in problem diagnosis.

1.3.4 Cognitively Natural

Qualitative models seem more cognitively natural. People typically use qualitative concepts and will often ignore the precise numerical information even when it is available (de Kleer, 1983). Humans are not very good at formal mathematical reasoning and will gain more insight from a qualitative explanation than when presented with an differential equation.

Not everyone who works with a design will have the mathematical ability of the engineer who designed it. Often people need this qualitative insight in order to understand the mathematical equation.

1.3.5 Causal

Qualitative models can provide causal reasoning. They can attribute behavior to certain parts of the system and to other behaviors. Thus qualitative simulation can give justifications for its results as well as explain why another decision was eliminated.

If the result of a numerical simulation disagreed with an engineer's experience, the engineer would tend to disregard the program's results, believing them to be erroneous. While qualitative models are no less prone to errors, they can justify their decisions, which serves as a validity check. This justification indicates which parameters were involved in the conclusion and how they may be adjusted.

Part II

Theory

Chapter 2

Basics

This chapter establishes the fundamental representations used by QM.

2.1 Direction

A concept of direction is essential in spatial reasoning to describe vector quantities, such as force and motion. In this section we discuss qualitative representations for vectors.

2.1.1 Translational Direction

When people give a direction in space, without resorting to pointing, diagrams, or mathematics, they typically use words such as "right, left"; "up, down"; and "front, back" relative to some frame of reference. To reason about a single dimension we assign "+" to the first of each of these pairs and "–" to the second with "0" meaning center. This description corresponds to the signs of the numerical values in a Cartesian coordinate system or to the signs of the cosine and the sine, respectively, in a polar coordinate system.

In order to reason about directional quantities in a discrete, symbolic way we extend this representation to multiple dimensions by combining values for each single dimension in an ordered list.[1] A direction which corresponds to an axis may be represented exactly, for example, $(+, 0, 0)$ indicates the vector is along the X axis. Directions which do not correspond to axes are represented by spatial regions. For example any direction to the lower left of the origin will be written $(-, -)$ or $(-, -, 0)$, where 0 indicates there is no magnitude along the Z axis. In higher dimensions, vectors such as $(+, +, +)$ would represent spatial regions. Because it is important to consider the lack of magnitude, we introduce a distinguished vector consisting of all zeros.

Figure 2.2 shows this notation graphically. In two dimensions there are nine possible qualitative directions:

(0, 0) a point indicating no value in any direction (alternatively, a value equal to that of the chosen reference frame

$(+, 0)$ increasing parallel to the X axis

[1]This capability is important not only for three dimensional analysis, but also for configuration space where dimensions may be higher than three.

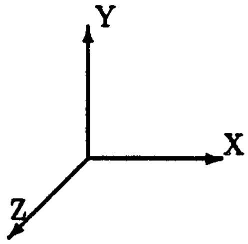


Figure 2.1: Right handed Cartesian coordinate system

Axis of rotation	Positive rotation
x	y to z
y	z to x
z	x to y

Table 2.1: Rotational directions

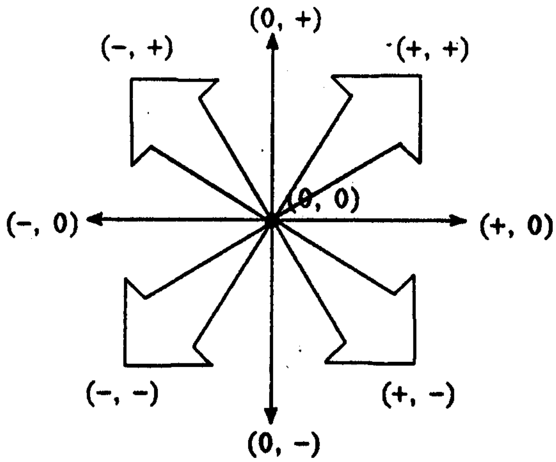


Figure 2.2: Qualitative translational directions

(−, 0) decreasing parallel to the X axis

(0, +) increasing parallel to the Y axis

(0, −) decreasing parallel to the Y axis

(+, +) the upper right quarter of the plane; any simultaneous increase in both the X and Y components without regard to the relative magnitude of these increases

(+, −) the lower right quarter of the plane; any simultaneous increase in the X component and decrease in the Y component

(−, +) the upper left quarter of the plane; any simultaneous decrease in the X component and increase in the Y component

(−, −) the lower left quarter of the plane; any simultaneous decrease in both the X and Y components

Definition 1 (All Possible Translational Directions) *The set of all possible translational directions, $\mathcal{T}$, is the Cartesian product of the sets of all possible translational directions for each dimension. The set of all possible translational directions for a single dimension is {+, 0, −}.*

Definition 2 (Translational Direction) `trans-dir(x)` *is a function on vector quantities. It is the ordered tuple of the signs of the direction component of vector* **x**.

Definition 3 (Zero Translational Direction) $\forall$ t ϵ $\mathcal{T}$, `Zero-Translation(t)` *is true iff for all dimensions of* t *the component value is 0.*

2.1.2 Rotational Direction

People typically describe direction of rotation either by the direction of a tangent on the described curve, or using words such as "counter-clockwise" and "clockwise". We represent a counter-clockwise rotation as "+" and a clockwise rotation as "−" when looking along a positive axis toward the origin. Thus the way we normally perceive the movement of the hands of a clock is (−) in two dimensions or (0, 0, −), in three dimensions (clockwise about the Z axis). This representation of rotational directions corresponds to the signs of numerical values in a right handed Cartesian coordinate system (Figure 2.1 and table 2.1).

Definition 4 (All Possible Rotational Directions) *The set of all possible rotational directions, $\mathcal{P}$, is the Cartesian product of the sets of all possible rotational directions about each axis. The set of all possible rotational directions about a single axis is {+, 0, −}.*

Definition 5 (Rotational Direction) `rot-dir(x)` *is a function on rotational quantities. It is the ordered set of the signs of* x*'s rotation about each axis.*

Definition 6 (Zero Rotational Direction) $\forall$ (r ϵ $\mathcal{P}$), `Zero-Rotation(r)` *is true iff for all axes of* r *the component value is 0.*

2.1.3 Vector Quantities

Two important vector quantities are force and motion. In classical mechanics these quantities are represented by direction and a magnitude. The previous sections demonstrated qualitative representations for direction. Rather than metrically represent magnitudes, we associate causal information with the external force quantities. We assume forces ultimately arise from external influences on the system. Without knowing how these come about we will associate a unique identifier with each external influence and refer to this as the *cause* of the force. The set of these causes is the *external-agents.* Knowing the cause of an effect is useful in resolving ambiguity because it allows a method of comparison. (Section 2.3 describes the way this causal information is used.)

Motions and forces change over time, but in any given instant an object can move in only one translational and one rotational direction. For now we will talk about time as instants and assume functions from objects to quantities map to instants of time. In section 5.1 we provide a discrete, symbolic representation of time.

Definition 7 (Motion) *$\forall$(t ϵ $\mathcal{T}$, o ϵ Objects),* `TransMotion(o, t)` *indicates the instantaneous linear motion of object* o *is in direction* t. *$\forall$(r ϵ $\mathcal{P}$, o ϵ Objects)* `RotMotion(o, r)` *indicates the instantaneous rotational motion of* o *is in direction* r.

Law 1 (Uniqueness of Motion) $\forall(o \epsilon Objects), \exists(t \epsilon \mathcal{T}) \mid TransMotion(o,t)$.
$\forall(o \epsilon Objects), \forall(t_1, t_2 \epsilon \mathcal{T}) TransMotion(o,t_1) \wedge TransMotion(o,t_2) \Rightarrow t_1 \equiv t_2$.
$\forall(o \epsilon Objects), \exists(r \epsilon \mathcal{P}) \mid TransMotion(o,r)$.
$\forall(o \epsilon Objects), \forall(r_1, r_2 \epsilon \mathcal{P}) \mid TransMotion(o,r_1) \wedge TransMotion(o,r_2) \Rightarrow r_1 \equiv r_2$.

The *uniqueness of motion* law says any object must have exactly one translational motion, and exactly one rotational motion at any given time. We consider non-movement to be in the direction consisting of all zeros.

There may be several forces (e.g., gravity, friction, push, etc.) acting on various surfaces of an object in different directions at the same time. Because the forces discussed so far only act on a surface, and we are concerned with their effects on the whole object, we introduce *distributed force*, *torque*, and *net force* which act at the object level. Distributed force is a set of forces in translational directions which influence an entire object. Torque is a set of forces in rotational directions which influence an entire object. The net force is a single force, with both translational and rotational components, which would have the same effect on an object as all of its distributed forces and torques combined. Later sections will show how to determine these quantities.

Definition 8 (Force) *$\forall$ (t ϵ $\mathcal{T}$, s ϵ Surfaces, c ϵ External-agents),* `Force(s, t, c)` *is true iff an agent,* c, *causes a force on surface* s *in the* t *direction.*

Definition 9 (Distributed Force) *$\forall$ (t ϵ $\mathcal{T}$, o ϵ Objects, c ϵ External-agents),* `Distributed-Force(o, t, c)` *is true iff* c *causes a force on object* o *in the* t *direction.*

Definition 10 (Torque) *$\forall$ (r ϵ $\mathcal{P}$, o ϵ Objects c ϵ External-agents), Torque(o, r, c) is true iff* c *causes a torque on object* o *in the* r *rotational direction.*

Definition 11 (Net Force) *$\forall$ (t ϵ $\mathcal{T}$, r ϵ $\mathcal{P}$, o ϵ Objects),* `Net-Force(o, t, r)` *indicates that the direction of the vector sum of all distributed forces acting on object* o *equals* t, *and the direction of the sum of all torques acting on object* o *equals* r.

2.2 Choosing a Frame of Reference

While there may be no preferred frame of reference, selection of a good frame of reference can greatly simplify analysis of a problem and reduce ambiguity (the "tilting the head" effect). Typically a reference frame will be selected or imposed by other considerations. However, to create a truly autonomous mechanical reasoner we present heuristics for automating selection of a reference frame which is useful in the analysis of mechanisms. This frame allows us to discuss force and motion using the above representations.

There are two stages in selecting a reference frame:

1. Select a fixed object to anchor the reference frame.

2. Select the axes for the orientation of the frame.

2.2.1 Finding a Fixed Location

The fixed object, by definition, will have no motion relative to this frame. When selecting a fixed object to define a reference frame there are three considerations :

1. The fixed object is chosen to minimize movement of the predetermined axes of rotation.

2. If there are no rotating links, the link with the greatest number of contacts is fixed

3. An outside agent may elect to select the fixed link in order to assist other analysis or study kinematic inversion.[2]

2.2.2 Orienting the Axes

Orientation of the reference frame is based on a preanalysis of the known vector quantities. These include external forces, surface normals, directions to the center of rotation, etc. (Surface normals of rotating objects are not considered because there will be an infinite number of them under all configurations.) For example if the only external force is gravity, its direction might become an axis. The following heuristics can be used to orient the frame of reference:

- Axes are selected from among the quantitative representations of free directions and the directions of the external forces.

- Further axes are selected to maximize parallel and perpendicular relations between surfaces. (Quantities which are normal to each other make natural axes.)

- If there are not enough vectors which are normal to each other, the vector cross product can be used to create new normal vectors.

- If there is more than one set of potential axes, the set which divides the initial directional quantities into the greatest number of separate quadrants and has the greatest number of directional quantities falling on an axis is selected. This selection minimizes ambiguity.

[2]Kinematic inversion is the creation of new mechanisms by fixing different links in a kinematic chain.

		sign(x) -	0	+
sign(y)	-	-	-	N1
	0	-	0	+
	+	N1	+	+

N1 :	if X > Y then sign(X)
	if X < Y then sign(Y)
	if X = Y then 0

Table 2.2: [X] + [Y]

		sign(x) -	0	+
sign(y)	-	+	0	-
	0	0	0	0
	+	-	0	+

Table 2.3: [X] * [Y]

For example, consider the clock escapement shown in Figures 1.2 and 1.3. Since both objects are rotating about pins we will choose one of these pins as the fixed frame. As both make the same number of contacts with other objects either pin may be arbitrarily chosen. The vector quantity used to initially orient the frame is gravity (the vertical axis).[3] It it further oriented about an axis which is normal to each of the surfaces (the Z axis coming out of the paper). The third axis is selected to be normal to these two (the horizontal axis).

The heuristics for choosing a qualitative frame of reference minimize information lost and make distinctions which are relevant to the problem. They are heuristic because an exact solution does not exist. Any qualitative frame of reference we choose will represent different vectors with the same symbol, and the distinction between these vectors may be critical to understanding the operation of the mechanism. Without advance knowledge of the operation of the mechanism we cannot know what distinctions are critical. In the future, once we have determined the function of the mechanism in one frame of reference we might use that information to suggest another frame of reference which could reveal more detail.

2.3 Qualitative Vector Arithmetic

The qualitative arithmetic of Forbus (Forbus, 1984), shown in tables 2.2 and 2.3, provides a way to manipulate sign information directly without knowing the actual magnitudes of the quantities involved. For example if two sources each cause a positive force in the same direction the net force will be positive. If there are forces in both the positive and negative directions, more information about relative magnitudes is needed to resolve the ambiguity. For example, two forces resulting from the same cause will have equal magnitudes and will cancel each other if they are in opposite directions.

[3] When gravity is considered, people usually prefer to make it act in a downward direction; this could be added as a minor extension to this algorithm.

$$\text{X axis rotation} = \begin{vmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{vmatrix}$$

$$\text{Y axis rotation} = \begin{vmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{vmatrix}$$

$$\text{Z axis rotation} = \begin{vmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

Table 2.4: Vector rotation

In general when two forces act in opposite directions the outcome is ambiguous, and we resolve this ambiguity by branching. Representing the relevant considerations of ambiguity to the designer may be useful at this point. The designer may elect to use a more precise analysis to resolve this ambiguity or simply increase the values of those parameters which will achieve the desired outcome. For example, the force due to friction opposes the force driving a clock. Knowing that the force driving the clock is caused by a spring or a weight, without determining the actual magnitudes, clock designers have been known to add weight or put in heavier springs until it works.

Though these formulations are simple, they enable us to represent a variety of spatial problems. Some of the more common problems we encounter in spatial reasoning involve determining ninety degree rotations and computing half planes (or half spaces).

We have extended qualitative arithmetic to compute these for vectors represented in the manner specified above. These extensions make use of the vector dot product $(a_1 \cdot a_2 + b_1 \cdot b_2 + c_1 \cdot c_2)$ and vector rotation formulas (table 2.4). In the vector rotation formulas θ is the angle of rotation. A θ of 90° causes the cosines to become zeros and the sines to become ones, so the only math we need consider is addition and multiplication of signs.

Definition 12 (Half Plane) $\forall$ *(*`x, y` $\in$ *$\mathcal{T}$),* `Half-Plane(x, y)` *is true iff the sign of the vector dot product of* `x` *and* `y` *is + or 0.*

Definition 13 (Open Half Plane) $\forall$ *(*`x, y` $\in$ *$\mathcal{T}$),* `Open-Half-Plane(x, y)` *is true iff the sign of the vector dot product of* `x` *and* `y` *is +.*

Definition 14 (Rotate-90) $\forall$ *(*`x, y` $\in$ *$\mathcal{T}$* `r` $\in$ *$\mathcal{P}$),* `Rotate-90(x, y, r)` *is true iff* `y` *is the vector which is perpendicular to* `x` *by the smallest rotation in the rotational direction* `r`*.*

2.4 Object Representations

2.4.1 Objects and Surfaces

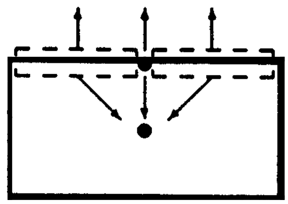


Figure 2.3: Surfaces of a block

Rigid objects are represented by the set of their surfaces. Surfaces, in turn, are distinguished by the qualitative direction of the surface normal *and* the direction from the surface to the center of rotation.[4] (We will introduce further distinctions in section 4.2.) For tractability the centers of rotation are assumed to be fixed.[5] For example, the top of a two dimensional block consists of three qualitatively distinct surfaces, all of which have a surface normal in the up direction but whose directions to the center of rotation are down-right, down, and down-left (Figure 2.3).

Places where the surface normal is not defined (i.e., corners) need special consideration. Since there is no behavioral difference between being in contact at a concave corner and being in contact with the two surfaces distinctly, we can represent concave corners as simultaneous contact with multiple surfaces. Convex corners, however, allow a greater range of motions than can be expressed by any combination of surface contacts. We represent convex corners explicitly by the set of adjacent surfaces. (From now on when we refer to a corner we will mean a convex corner.)

Definition 15 (Surface) $\forall$ *(*x ϵ *Objects),* `Surface(x, p)` *is true if* p *is a point (or set of qualitatively equivalent, connected points) on the exterior of object* x.

Definition 16 (Surface Normal) $\forall$ *(*d ϵ $\mathcal{T}$, p ϵ *Surfaces),* `Surface-Normal(p, d)` *is true if* d *is the direction of the surface normal at the surface* p.

Definition 17 (Origin Direction) $\forall$ *(*d ϵ $\mathcal{T}$, p ϵ *Surfaces),* `Origin-Dir(p, d)` *is true iff* d *is the translational direction from a surface,* p*, on an object to the center of rotation of that object.*

Definition 18 (Corner) $\forall$ *(*x, s_1, s_2 ϵ *Surfaces),* `Corner(x, s1, s2)` *is true if* x *has zero extent, and the smallest angle which passes through free space between* s_1 *and* s_2 *is greater than 180°.*

[4]If an object cannot rotate, all centers of rotation are equally good. By convention we use the zero direction.

[5]Either we are given the direction from a surface to its center of rotation in advance or a set of possible centers is indicated, any one of which will be the center during the behavior of interest.

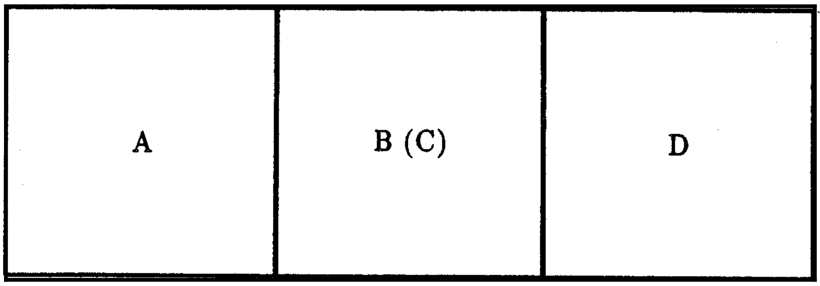


Figure 2.4: Indirect contact

2.4.2 Contact

In order for an object to affect another object there must be some kind of contact between them. (If we think of effects such as gravity and magnetism as a field, we can reason about the contact between this field and an object.) Objects that can interact pairwise are called *kinematic pairs.* Because we are reasoning about changes occurring over time, these objects need not continually be in contact but only potentially be in contact. Even when they are not in contact, information about their next possible contacts may be significant. (In chapter 4 we examine the effects of kinematic pairs which are not in contact.) The `Contact` relation describes which parts of objects are in contact.

Definition 19 (Contact) $\forall$ *(*`x, y` $\in$ *Surfaces),* `Contact(x, y)` *is true if* `x` *touches* `y`.

Definition 20 (Kinematic Pair) $\forall$ *(*`x, y` $\in$ *Objects),* `x` *and* `y` *form a* `kinematic pair` *if they have surface contact or their motion envelopes intersect.*

2.4.3 Kinematic Chains

When one object participates in multiple kinematic pairs, we need to indicate that the properties of this object will be identical in both representations. The `Linkage` relation does this. It says that if the force or motion is known for one of the objects, that property will hold over the other object as well. For example, assume blocks A and B are in contact, blocks C and D are in contact, and we later learn that B and C are the same object seem from different sides (see Figure 2.4). Any motion transmitted from A to B will instantaneously affect C and may affect D.

Definition 21 (Linkage) $\forall$ (`obj`$_1$, `obj`$_2$ $\in$ *Objects),* `Linkage(obj`$_1$`, obj`$_2$`)` $\Rightarrow$
$[\forall\ (t_1,\ t_2 \in \mathcal{T},\ r_1,\ r_2 \in \mathcal{P}),\ (\textit{Net-Force}(obj_1,\ t_1,\ r_1) \wedge \textit{Net-Force}(obj_2,\ t_2,\ r_2)) \rightarrow$
$(t_1 \equiv t_2 \wedge r_1 \equiv r_2)] \wedge$
$[\forall (t_3,\ t_4 \in \mathcal{T}),\ \mathtt{TransMotion}(obj_1,\ t_3) \wedge \mathtt{TransMotion}(obj_1,\ t_4) \rightarrow t_3 \equiv t_4] \wedge$
$[\forall (r_3,\ r_4 \in \mathcal{P}),\ \mathtt{RotMotion}(obj_1,\ r_3) \wedge \mathtt{TransMotion}(obj_1,\ r_4) \rightarrow r_3 \equiv r_4]$

Because behavior is only influenced through contact we distinguish objects that may directly or indirectly affect each other's behavior from other objects that just happen to be part of the description. A *kinematic chain* is composed of the closure of kinematic pairs. *Kinematic chains* help solve the *frame problem* (McCarthy & Hayes, 1969) in this context since only parts of the same kinematic chain will ever influence the behavior of other parts.

Definition 22 (Kinematic chain) *A* `kinematic chain` *is a set of kinematic pairs such that each pair shares an object with at least one other kinematic pair in the chain.*

Chapter 3

Mechanical Motion

This chapter shows how the representations in the previous chapter may be used to reason about mechanical motion. First we eliminate motion directions which are impossible. Then we consider how motion propagates between objects that are in contact. Finally, we consider the effects of external influences to the system.

3.1 Blocking

This section answers two questions. Given contact between an object and an obstacle

1. How will the motion constraints of the obstacle prevent the object from moving?
2. Which motions of the obstacle must be constrained to prevent the object from moving in some direction?

A *mechanical constraint* is a reaction force which absolutely prevents a body from moving in a certain direction. Constrained motion is essential to understanding mechanics because constraints guide motion so that the objects can "move only in definite paths relative to the other parts"(Cowie, 1961). For example, a piston in a cylinder is constrained from moving any direction except along the cylinder. The opposite of a constraint is a *freedom*. When an object is completely free to move in any direction, its path will be determined by the forces acting on it. For example, a ball thrown in the air has complete freedom.

The mechanisms we consider are partially constrained. Certain paths of motion are prevented by contact between rigid members, but there may be some "play" between the parts where the path of motion must be determined by analysis of the forces.

In this section we show how to determine the effects of constrained objects and partially constrained objects on the allowable motions of other objects. An object which is part of the fixed frame of reference is completely constrained. An object, by default, is free to move in each direction unless it is specifically constrained.

Definition 23 (Constraint) *$\forall$ (*`t` *$\in \mathcal{T}$,* `o` *$\in$ Objects),* `TransConstraint(o, t)` *is true when object* `o` *is absolutely prevented from moving in direction* `t`*.*
$\forall$ (`r` *$\in \mathcal{P}$,* `o` *$\in$ Objects),* `RotConstraint(o, r)` *is true when object* `o` *is absolutely prevented from rotating in direction* `r`*.*

$$
\begin{aligned}
&\forall\ (obj,\ obst,\ p,\ q,\ sn)\\
&\quad [\mathit{RigidBody}(obst) \wedge \mathit{RigidBody}(obj) \wedge \mathit{Surface}(obst,\ p) \wedge \mathit{Surface}(obj,\ q) \wedge\\
&\quad\ \mathit{Contact}(p,\ q) \wedge \mathit{Surface\text{-}Normal}(p,\ sn) \wedge \mathit{Origin\text{-}Dir}(p,\ o_1) \wedge \mathit{Origin\text{-}Dir}(q,\ o_2) \wedge\\
&\quad\ (\forall d_1)\ [\mathit{Open\text{-}Half\text{-}Plane}(\text{-}sn,\ d_1) \Rightarrow \mathit{TransConstraint}(obst,\ d_1)] \wedge\\
&\quad\ (\forall r_1 \exists x_1)\ [\mathit{Rotate\text{-}90}(\text{-}sn,\ x_1, r_1) \wedge \mathit{Open\text{-}Half\text{-}Plane}(x_1, o_1)\\
&\qquad\qquad \Rightarrow \mathit{RotConstraint}(obst,\ r_1)]\ ]\\
&\quad \Longrightarrow\\
&\quad [(\forall d_2)\ [\mathit{Open\text{-}Half\text{-}Plane}(\text{-}sn,\ d_2) \Rightarrow \mathit{TransConstraint}(obj,\ d_2)] \wedge\\
&\quad\ (\forall r_2 \exists x_2)\ [\mathit{Rotate\text{-}90}(\text{-}sn,\ x_2, r_2) \wedge \mathit{Open\text{-}Half\text{-}Plane}(x_2, o_2)\\
&\qquad\qquad \Rightarrow \mathit{RotConstraint}(obj,\ r_2)]\ ]
\end{aligned}
$$

Table 3.1: The law of contact constraint

Definition 24 (Freedom) $\forall$ ($t \in \mathcal{T}$, $o \in$ *Objects*), `TransFreedom(o, t)` *is true when object* `o` *is not prevented from moving in direction* `t`.
$\forall$ ($r \in \mathcal{P}$, $o \in$ *Objects*), `RotFreedom(o, r)` *is true when object* `o` *is not prevented from rotating in direction* `r`.

The motion constraints which may be imposed when two objects are in contact are given in Figure 3.1. This says that if an obstacle is "sufficiently" constrained it will prevent the following motions of an object in contact:

- Translational motion into the open half plane centered on the object's surface normal at the point of contact.
- Rotational motion clockwise about any point which lies in the open half plane centered ninety degrees clockwise from the object's surface normal at the point of contact.
- Rotational motion counter-clockwise about any point which lies in the open half plane centered ninety degrees counter-clockwise from the object's surface normal at the point of contact.

Previous approaches to qualitative kinematics (and classical kinematics) have assumed all objects are fixed except the object of interest. In those systems, "sufficiently" constrained meant *completely* constrained. That over-simplifies the problem and prevents reasoning at the system level.

An obstacle may be only partially constrained yet still prevent other objects from moving in some directions. The minimum set of directions in which an obstacle must be constrained in order to be ***sufficiently constrained*** is exactly the same as the set of directions of imposed constraints. That is, an obstacle is sufficiently constrained if it is unable to move in any of the following directions:

- Translational motion into the open half plane centered on the object's surface normal at the point of contact.

- Rotational motion clockwise about any point which lies in the open half plane centered ninety degrees clockwise from the object's surface normal at the point of contact.
- Rotational motion counter-clockwise about any point which lies in the open half plane centered ninety degrees counter-clockwise from the object's surface normal at the point of contact.

Figure 3.1 illustrates this law graphically for the two dimensional case. Consider an object (the block) in contact with an obstacle (the wall). The inverse surface normal of the wall at the point of contact is to the right (Figure 3.1.1). When the wall is prevented from moving translationally into the half plane centered on this surface normal (*up-right*, *right*, or *down-right*) (Figure 3.1.2); prevented from rotating *counter-clockwise* about a point which is in the half plane centered ninety degrees counter-clockwise of the surface normal (Figure 3.1.3); and prevented from rotating *clockwise* about an point which lies in the half plane centered ninety degrees clockwise of the surface normal (Figure 3.1.4) — then the block will be unable to move translationally into the half plane centered on this surface normal (*up-right*, *right*, or *down-right*) (Figure 3.1.2); unable to rotate *counter-clockwise* about a point which is in the half plane centered ninety degrees counter-clockwise of the surface normal (Figure 3.1.3); and unable to rotate *clockwise* about a point which lies in the half plane centered ninety degrees clockwise of the surface normal (Figure 3.1.4).

A stack of blocks is a simple example of the need for partially constrained obstacles. Though the floor is the only obstacle which is completely constrained, any block in the stack is prevented from moving in any downward direction because it has contact along a surface with the surface normal in the "down" direction and the block (or floor) it is in contact with is constrained in all downward directions.

In Figure 3.2, if the ramp is completely constrained, the block is still free to move in *any* direction. The wedge does not sufficiently constrain motion in the downward direction since it is free to move translationally in the *down-right* direction. The block may move in a downward direction by pushing the wedge down and to the right. If the wedge could not be pushed to the side (perhaps a catch on the ramp) the wedge could not move in any of the directions required by the shape of the surface between the block and the wedge, and consequently the block could not move downward.

3.1.1 Constraints Imposed at Corners

For any two surfaces that are in contact, the direction of the surface normal of one will be opposite to the direction of the surface normal of the other. Thus any time a surface is in contact with a corner we may use the reverse of the surface's surface normal as the surface normal of the corner. For example, in the above law, to determine the constraints imposed by the corner on the surface, we may use the reverse of the surface's surface normal as the surface normal of the corner.

There is a degenerate case, however, where the surface normal is not defined for either surface. When one surface slides off another there will be an instant when contact between two convex corners can occur. To determine the applicable motion constraints in this case we must first determine which of the adjacent surfaces can come into contact with the opposite corner. The area of blocked space enclosed by a corner is determined from the reverse surface normals of the adjacent surfaces. The intersection of the half planes defined

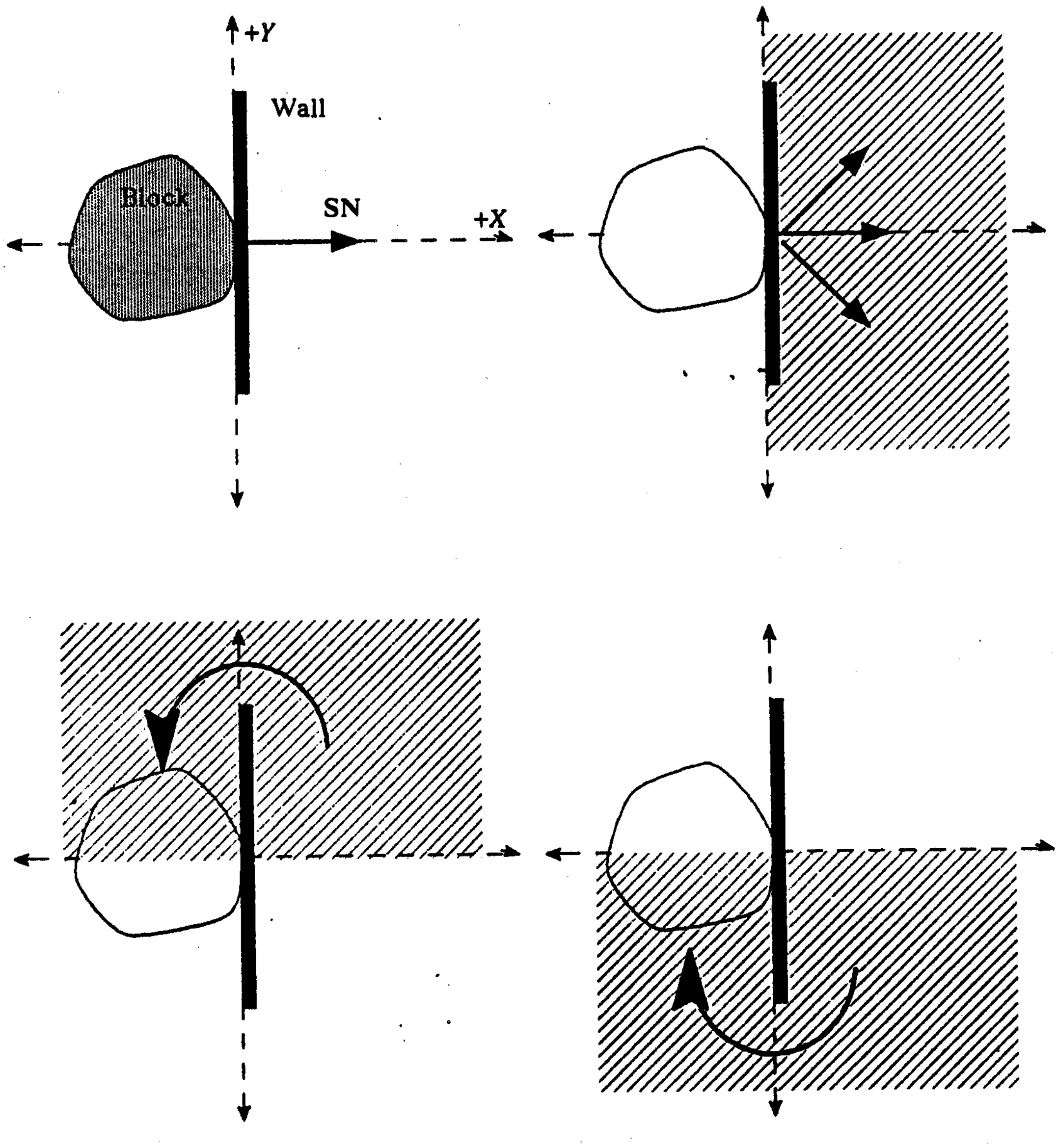


Figure 3.1: Constraints imposed by surface contact

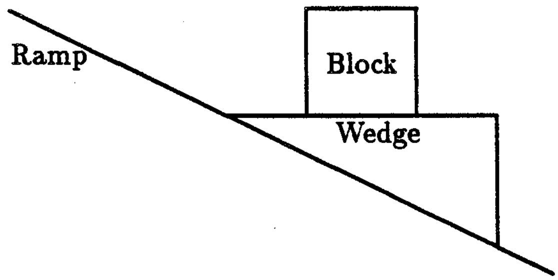


Figure 3.2: Block on wedge on ramp

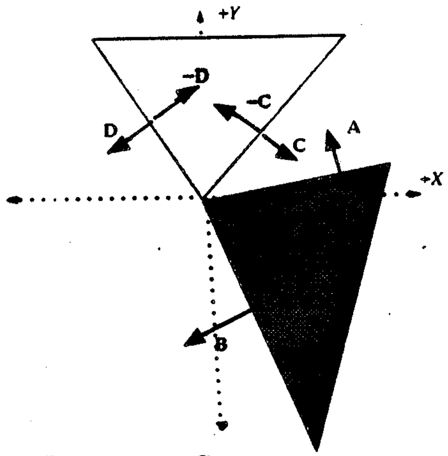


Figure 3.3: Corner contacts

by those surface normals is blocked space. If the surface normal of another surface lies in one of these directions, then that surface may contact that corner, otherwise it cannot. For example, in Figure 3.3 the directions given by the intersection of the half planes defined by the reverse surface normals of the upper triangle is in the directions (+, +), (0, +) and (–, +). Since the surface normal at A is in the (–, +) direction contact may occur; however, the surface normal at B is in the (–, –) direction and may not come into contact with the corner.

Definition 25 (Potential Contact)
$\forall$ *(*`obj, obst` ϵ *Objects,* `sur, cor,` p_1, p_2 ϵ *Surfaces,* `sn,` d_1, d_2 ϵ $\mathcal{T}$*),*
`Corner(cor,` p_1`,` p_2`)` $\wedge$ `Surface(obst, sur)` $\wedge$
`Surface(obj, cor)` $\wedge$ `Surface(obj,` p_1`)` $\wedge$
`Surface(obj,` p_2`)` $\wedge$ `Surface-Normal(`p_1`,` d_1`)` $\wedge$
`Surface-Normal(`p_2`,` d_2`)` $\wedge$ `Surface-Normal(cor, sn)` $\wedge$
`Half-Plane(-`d_1`, sn)` $\wedge$ `Half-Plane(-`d_2`, sn)`
$\Rightarrow$ *Can-Contact(cor, sur)*

Once we know which of the adjacent surfaces may come into contact with each of the corners, we can find the constraints imposed at the corner by taking the intersection of the set of constraints imposed by each of these potential contacts. For example in Figure 3.3 the upper block may constrain by the intersection of the constraints imposed by contact between the corner of the upper block and surface A, and the constraints imposed by contact between the corner of the lower block and surface C.

Definition 26 (Corner Constraints) *The constraints imposed by contact between two convex corners is the intersection of those imposed by contact between each corner and each adjacent surface that can come into contact.*

3.2 Pushing

We have seen how a fixed body will prevent motion. Now we explore how a moving body will transfer motion. Again there are two questions. Given contact between an object and a moving body:

1. How will the motion of the body affect the object?
2. What motions of the body will affect the object?

The law describing the motions an object must undergo when in contact with a moving body are given in Figure 3.2. This says that if a body is moving "into" an object, the object must move in at least one of the following ways, and if none of these motions are possible the body cannot move:

- Translational motion into the open half plane centered on the body's surface normal at the point of contact,
- Rotational motion clockwise about some point which lies in the open half plane centered ninety degrees clockwise from the body's surface normal at the point of contact, or

$$
\begin{array}{l}
\forall\ (obj,\ obst,\ p,\ q,\ sn) \\
\quad [RigidBody(body) \wedge RigidBody(obj) \wedge Surface(body,\ p) \wedge Surface(obj,\ q) \wedge \\
\quad\ Contact(p,\ q) \wedge Surface\text{-}Normal(q,\ sn) \wedge Origin\text{-}Dir(p,\ o_1) \wedge Origin\text{-}Dir(q,\ o_2) \wedge \\
\quad [(\exists d_1)\ [Open\text{-}Half\text{-}Plane(\text{-}sn,\ d_1) \wedge TransMotion(body,\ d_1)] \vee \\
\quad\ (\exists r_1 x_1)\ [Rotate\text{-}90(\text{-}sn,\ x_1, r_1) \wedge Open\text{-}Half\text{-}Plane(x_1, o_1) \\
\qquad\quad \wedge\ RotMotion(body,\ r_1)]\ ]] \\
\quad \Longrightarrow \\
\quad [(\exists d_2)\ [Open\text{-}Half\text{-}Plane(\text{-}sn,\ d_2) \wedge TransMotion(obj,\ d_2)] \vee \\
\quad\ (\exists r_2 x_2)\ [Rotate\text{-}90(\text{-}sn,\ x_2, r_2) \wedge Open\text{-}Half\text{-}Plane(x_2, o_2) \\
\qquad\quad \wedge\ RotMotion(obj,\ r_2)]\ ]
\end{array}
$$

Table 3.2: The law of motion transfer

- Rotational motion counter-clockwise about some point which lies in the open half plane centered ninety degrees counter-clockwise from the body's surface normal at the point of contact.

The directions in which a body can move and still have *motion into* the object is exactly the same as the directions the object must undergo. Namely, the body is moving into the object if it has any of the following motions:

- Translational motion into the open half plane centered on the body's surface normal at the point of contact
- Rotational motion clockwise about any point which lies in the open half plane centered ninety degrees clockwise from the body's surface normal at the point of contact
- Rotational motion counter-clockwise about any point which lies in the open half plane centered ninety degrees counter-clockwise from the body's surface normal at the point of contact.

Imagine the block is moving right in Figure 3.4. In this case the block in the "body" and the boulder is the "object". SN indicates the surface normal of the block which is the same as the negative surface normal of the boulder. Because the direction of motion of the block is into the boulder, the boulder must move in one of the following directions:

- translationally right,
- translationally down right,
- translationally up right,
- clockwise about a point down from the contact,
- clockwise about a point down left from the contact,

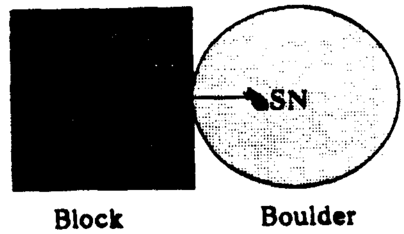


Figure 3.4: Block pushing a boulder

- clockwise about a point down right from the contact,
- counter-clockwise about a point up from the contact,
- counter-clockwise about a point up left from the contact, or
- counter-clockwise about a point up right from the contact.

3.3 External Forces

In addition to forces which arise by contact between objects, there are external forces (such as gravity, springs, and friction) that may affect the system. We can consider the effects of these influences without knowing how they arise. These forces may be constantly active or may be active only in certain configurations.

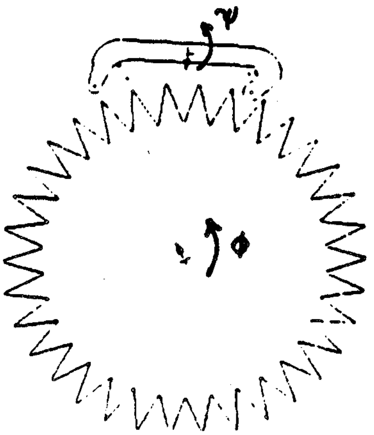
In QM we model a force as acting on a (possibly imaginary) surface of the object. For example, a dropped brick would be pushed downward by gravity at its center of mass. The surface normal is irrelevant, but the direction to the center of rotation is important because a torque will be produced if the direction to the center of rotation is not collinear with the direction of the force. Table 3.3 shows all the directions in which a single force may influence an object. Once we determine the set of distributed forces and torques acting on an object we can add them, using the qualitative arithmetic (section 2.3) to obtain the net force on an object.

To determine the instantaneous motion from a given configuration, we first eliminate the constrained motions and then apply motion transfer. If there is a directed net force and no current motion of the object in that direction, the object will instantly move in the given direction. If the object has non-zero current motion or zero force, its current motion will persist to the next instant. (We will consider changes of motion that require some time interval in section 5.3.1). Finally, if there is no consistent set of unconstrained motions, the object cannot move.

$$
\begin{array}{l}
(\forall\ obj,\ q) \\
\qquad [\mathit{RigidBody}(obj) \wedge \\
\qquad\ \mathit{Surface}(obj,\ q) \wedge \\
\qquad\ \mathit{Origin\text{-}Dir}(q,\ o_1) \wedge \\
\qquad\ \mathit{Force}(q,\ d_f,\ c)] \\
\qquad \Longrightarrow \\
\qquad [(\forall d_1)\ [\mathit{Open\text{-}Half\text{-}Plane}(d_f, d_1) \\
\qquad\qquad \wedge\ \mathit{Not}(\mathit{TransConstraint}(obj,\ d_1))] \\
\qquad\qquad \Rightarrow\ \mathit{Distributed\text{-}Force}(obj,\ d1) \\
\qquad\ (\exists r_1 x_1)\ [\mathit{Rotate\text{-}90}(d_f, x_1, r_1) \\
\qquad\qquad\quad \wedge\ \mathit{Open\text{-}Half\text{-}Plane}(x_1, o_1) \\
\qquad\qquad\quad \Rightarrow\ \mathit{Torque}(obj,\ r_1)]\ ]
\end{array}
$$

Table 3.3: The law of force distribution

Clock Escapement **Configuration Space**



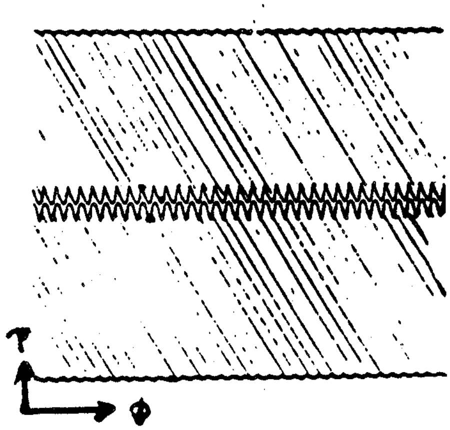


Figure 4.1: Clock escapement and corresponding configuration space

Within a configuration space we distinguish regions by whether or not the objects are overlapping. Overlapping regions are blocked and non-overlapping regions represent free space. The boundaries between free and blocked regions in configuration space represent configurations where the objects are in contact. The particular configuration space representations we use are based on work by Faltings (Faltings, 1987b).

Figures 4.1 and 4.2 illustrate configuration space representations for a recoil escapement and a gear pair respectively. Shaded regions represent blocked space and unshaded, free space with solid lines representing contact. The configuration space for the gear has been enlarged to show free space between the segments. Chapter 9 provides more examples of configuration spaces.

Definition 27 (Configuration Space) *A* `configuration` *is a specific choice of values for the motion parameters of an object. A* `configuration space` *is the space of all configurations.*

4.2 Place Vocabularies

If we represented locations as points, a complete spatial analysis of each possible location of an object would take infinitely long. Thus the space of consideration must be partitioned. If one is not careful, a partitioning maybe too grainy or lack relevant information. Therefore,

provides interesting suggestions for recognition of lower pairs once this configuration space has been computed, and Gelsey (Gelsey, 1987) has developed an algorithm for identification of lower pairs from part geometry.

Chapter 4

Location

In this chapter we develop a generative approach to qualitative kinematics based on *configuration space* representations of object interaction. This may be used to abstractly compare position and facilitate reasoning at varying levels of detail.

4.1 Metric Diagram

People are much better at spatial reasoning when provided with a diagram than with an English explanation. Shapes have more geometric features than can comfortably be encoded by qualitative spatial vectors. We advocate the metric diagram/place vocabulary (hereafter MD/PV) (Forbus et al., 1987) approach to spatial reasoning. A *metric diagram* is an oracle for answering simple spatial questions, such as intersection and other spatial relations. It may take the form of drawings, equations, or a camera. From the metric diagram we compute a discrete, symbolic representation of space (the place vocabulary) to make the features relevant to the questions we expect to answer explicit. Specifically, the geometric questions we expect to answer by observation are the following:

1. How can the objects touch?
2. Are there configurations of the objects where they do not touch?
3. What configurations are illegal?
4. What configuration will result when one or more objects are moved in a specified manner?

We have found configuration space (Lozano-Perez & Wesley, 1979) to be a useful representation for reasoning about the location of mechanical pairs. Each axis of the configuration space corresponds to one degree of freedom (allowable motion) of one of the objects. Unconstrained objects will have six degrees of freedom each (three translational motions and three rotational motions), but components of mechanisms, by definition, are relatively constrained. "Single-degree-of-freedom mechanisms are the forms used most frequently"(Erdman & Sandor, 1984), so each kinematic pair typically only needs a two dimensional representation after the lower pair restrictions[1] have been imposed.

[1] Lower pair analysis falls outside the scope of our research because there are only six such pairs, and they require a twelve dimensional configuration space for analysis. Recent work by Joskowicz (Joskowicz, 1987b)

Clock Escapement Configuration Space

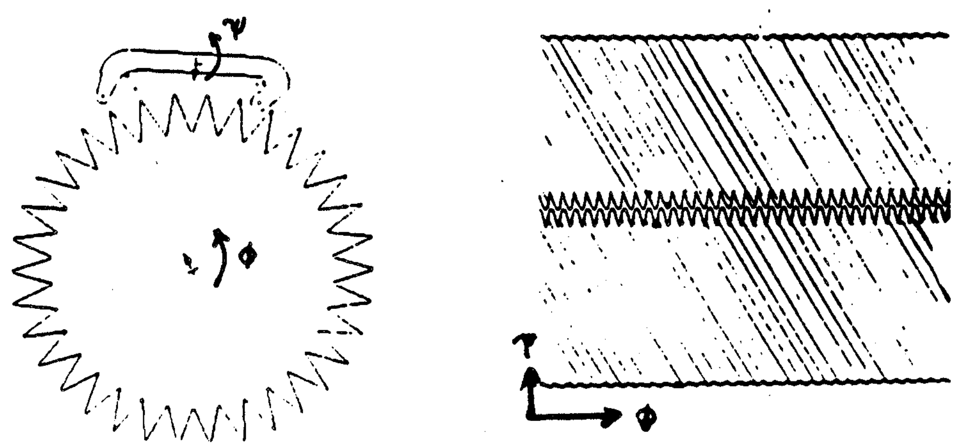


Figure 4.1: Clock escapement and corresponding configuration space

Within a configuration space we distinguish regions by whether or not the objects are overlapping. Overlapping regions are blocked and non-overlapping regions represent free space. The boundaries between free and blocked regions in configuration space represent configurations where the objects are in contact. The particular configuration space representations we use are based on work by Faltings (Faltings, 1987b).

Figures 4.1 and 4.2 illustrate configuration space representations for a recoil escapement and a gear pair respectively. Shaded regions represent blocked space and unshaded, free space with solid lines representing contact. The configuration space for the gear has been enlarged to show free space between the segments. Chapter 9 provides more examples of configuration spaces.

Definition 27 (Configuration Space) *A* **configuration** *is a specific choice of values for the motion parameters of an object. A* **configuration space** *is the space of all configurations.*

4.2 Place Vocabularies

If we represented locations as points, a complete spatial analysis of each possible location of an object would take infinitely long. Thus the space of consideration must be partitioned. If one is not careful, a partitioning maybe too grainy or lack relevant information. Therefore,

provides interesting suggestions for recognition of lower pairs once this configuration space has been computed, and Gelsey (Gelsey, 1987) has developed an algorithm for identification of lower pairs from part geometry.

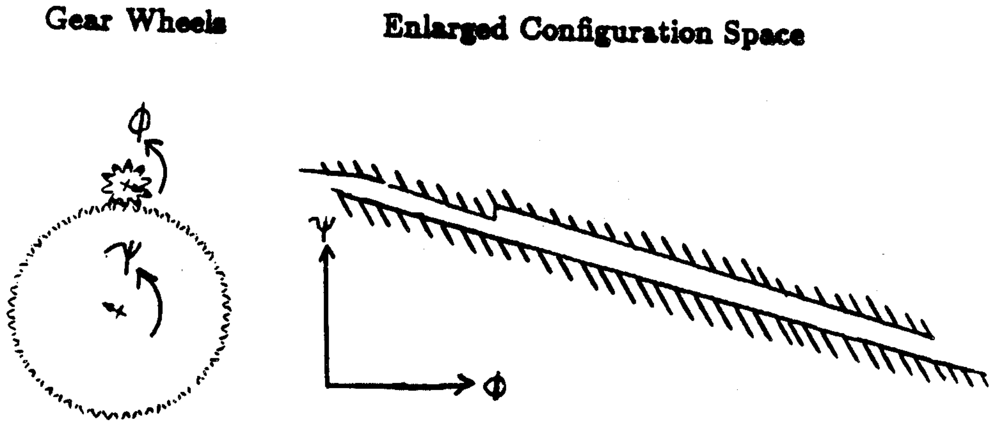


Figure 4.2: Gear wheels and corresponding configuration space

a central problem of symbolic spatial reasoning is finding an appropriate representation to describe space as a finite number of regions rather than as an infinite number of points.

We adopt the concept of a *place vocabulary* (introduced in Forbus, 1981) which partitions space according to the kind of reasoning to be performed. A connected region of space in which all points share common properties is a *place*. The set of all places covering the space of interest is a *place vocabulary*.

Different reasoning schemes necessitate different divisions. In mechanisms the shape of the surfaces of the objects in contact affect their possible behaviors, and the next possible contacts define the behavioral predictions of objects. Faltings (Faltings, 1987b) provides a place vocabulary for his configuration space representations based on constraint equations. Here we will develop place vocabularies based on the possible motions of the objects.

As we have seen, contact constitutes one relevant property of distinguished places. However, contact alone is not enough. When objects are not in contact we can use information about possible motions to determine the next possible contacts. Our place vocabulary consists of regions where contact is equivalent (according to the qualitative direction of the surface normal) and non-contact regions which are divided according to the next contacts attainable through motion.

Places in the place vocabulary consist of four types of symbolic descriptors distinguished by contact and allowable motion. These are *constraint segments*, *joins*, *free space divisions*, and *full faces*. The properties of each are discussed below.

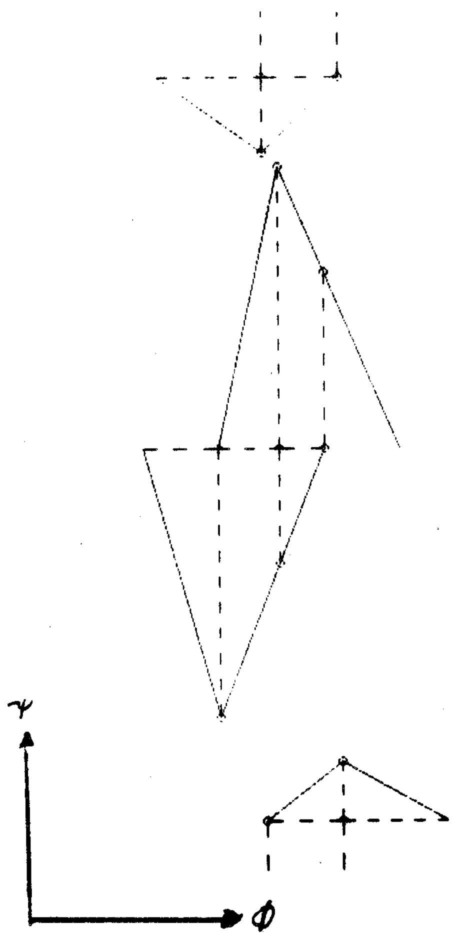


Figure 4.3: Place vocabulary for scape wheel and pallets

4.2.1 Constraint Segments

Constraint segments (hereafter CSEG's) form the boundaries between free and blocked regions in configuration space. They represent configurations where the objects contact. We further distinguish CSEG's by the direction of their surface normal for two reasons:

1. The orientation of the surface restricts the possible motions of the objects and determines the direction of the forces transmitted by contact.

2. Forces change when surface orientations change, which subsequently changes the mechanism's behavior.

Definition 28 (CSEG) *A CSEG is a locus of points characterized such that both legal and illegal configurations can be reached from them by an arbitrarily small motion.*

Law 2 (Motion Constraint) *A CSEG prevents motion into the open half plane centered on its normal into blocked space.*

The exterior lines in Figures 4.1 and 4.2 as well as the solid lines in Figure 4.3 represent constraint segments. Figure 4.3, which has been enlarged to show detail, represents the place vocabulary for the clock escapement shown in 4.1.

4.2.2 Joins

Most of the previous work on qualitative kinematics overlooks the points where surfaces meet.[2] Analysis at a point is only slightly more complex than along a smooth surface, but including them roughly doubles the size of the resultant place vocabulary. We feel that the behavioral continuity is worth this expense.

Definition 29 (Concave Join) *A join is* concave *if the CSEG on one side of the join lies in the same half plane as the surface normal of the CSEG on the opposite side of the join. The constraints imposed by an concave join are the union of the constraints imposed by the adjacent surfaces.*

Definition 30 (Convex Join) *A join is* **convex** *if the CSEG on one side of the join does not lie in the same half plane as the surface normal of the CSEG on the opposite side of the join. The constraints imposed by an convex join are the intersection of the constraints imposed by the adjacent surfaces.*

Note that either definition may be used for a straight join. In Figure 4.3 joins are represented by small circles.

4.2.3 Free Space Divisions

Some interesting kinematic pairs (such as clock escapements) do not stay in contact and may produce intermittent motions. Most common kinematic pairs need to have some play between the parts to reduce friction. Free space divisions (FSD's) partition regions of open space. They provide behavioral distinctions when the objects are not in contact.

[2]The only exception being (Shoham, 1985), which only treats points.

The number and form of the FSD's affect the complexity of the resultant place vocabulary. The minimum FSD's should at least originate where the shape of the contact changes, since this corresponds to a qualitative change in the behavior of the mechanism (other origins are possible subject to additional information about free space), then follow the free directions of motion of each individual object.[3] This partitions free space according to the next possible contact, given motion of a single object.

Definition 31 (Free Space Division) *A* `FSD` *is the locus of configurations originating from any "significant" configuration and following one of the degrees of freedom. An FSD will terminate at the nearest configuration in which the object is once again in contact.*

Our choice of FSD's corresponding to possible motions of the objects helps minimize the ambiguity of qualitative analysis, but there may still be more than one possible next place (e.g., when both objects move). We can eliminate all but a few choices, but without more metric information (rates and distances), there is inherent ambiguity. In Figure 4.3 free space divisions are shown as dashed lines.

4.2.4 Full Faces

Open areas of space between constraints, free space divisions, and joins are termed *full faces.* They impose no restrictions on the current movement of the objects, but serve to answer the question "Where can this go next?" In Figure 4.3 full faces are the open areas between the segments.

Definition 32 (Full Face) *A* `full face` *is an open area of the configuration space, possibly bounded by CSEG's, joins, or FSD's.*

If the parts have no interaction with each other, their entire configuration space will be one unbounded full face.

4.3 Place Vectors

While an N dimensional configuration space (where N is the summation of the number of links, L, times the degrees of freedom of that link, F_L) would characterize the configuration of the entire mechanism, this configuration space would be enormous. Further, the knowledge this provides does not justify its complexity. Instead, we represent the static state of the mechanism as a *place vector* consisting of one place from each kinematic pair in the kinematic chain. When the mechanism has only one kinematic chain, the place vector consists of $L - 1$ places characterizing the mechanism's overall configuration. This may be thought of as several lower dimensional projections of the N dimensional space.

Place vocabularies for kinematic pairs may be stored. This provides a modular approach to mechanism analysis. When parts are changed, we need only construct the place vocabulary of the new parts, not the entire mechanism. It also allows us to create a library of the place vocabularies of common mechanisms which may be smoothly integrated with new mechanisms.

[3] Falting's approach (Faltings, 1986) was more concerned with reasoning about kinematic constraint and constructed FSD's that follow the shape of the curve which produced the constraint.

Information may be propagated across place vocabularies by establishing a *correlation* between sets of places. This allows one to show how the same object participates in two place vocabularies.[4] For example, if one wanted to express that one object was rotating twice as slow as another, a correlation could be set up between places in the first object and places offset by 0° and 180° for the second object.

Definition 33 (Place Vector) *A* **place vector** *is a set consisting of one place from each of the place vocabularies of each kinematic pair in a kinematic chain.*

Definition 34 (Correlation) *If X and Y are sets of places and {place vectors} is the set consisting of all place vectors in the mechanism.*
`PlaceCorresponds(X Y)`
$\Rightarrow$
$\forall$ *(p ϵ {place vectors}, x ϵ X, y ϵ Y) member(x, p) $\leftrightarrow$ member(y, p).*

[4]This is similar to the composition of kinematic pair relationships introduced by Gelsey (Gelsey, 1987).

Chapter 5

Envisioning

5.1 Time

The behavior of mechanisms changes in time. Elements move; contacts are made and broken; and forces can occur intermittently. Our model of time is based on Hayes' (Hayes, 1979) divisions of spacetime called *histories* (with extensions by Forbus, 1984 and Williams, 1986). Briefly, histories are bounded in time (events begin and end) as well as space. It is assumed that spatially isolated objects will not affect each other's behaviors. In section 2.4.3 we saw how to distinguish spatially isolated objects in mechanism. Even when kinematic pairs are not in contact we want to keep track of their next possible interactions, so the initial spatial extent of our representations does not change.

Histories are divided into contiguous, non-overlapping sequences of *episodes* and *events* which differ in their temporal extent; events occur in at instant while episodes persist for some interval of time. Within each episode/event the relevant behaviors do not change. For example, we can talk about a ball moving upward for some interval, it stops in an instant, and moved downward for some interval.

Instead of talking about functions from objects to quantities mapping on to instants of time, functions from objects to quantities will now map on to either episodes or events where the distinction is given by the equality change law (Forbus, 1984).

Law 3 (Equality change) ***A behavior lasts for an instant only when a change from some threshold occurs. In all other cases a behavior lasts for an interval of time.***

Temporal information resolves some of the ambiguity when computing transitions from a given state. Instantaneous changes will occur before those that require an interval. For example, because contact between surfaces changes in an instant it will occur before a dynamic influence which requires some interval.

Law 4 (Instant/interval change) ***All changes which occur in an instant will occur before those which require an interval. All changes which may occur in an instant will occur at the same time. Changes which require some interval will not occur at the same time unless there is a direct or indirect influence between them.***

Motion(Pallets) = 0	Motion(Scape) = -	Motion(G1) = -
Motion(G2) = +	Motion(G3) = +	Motion(G4) = -
Motion(G5) = -	Motion(G6) = +	Motion(G7) = -
Motion(G8) = -	Motion(G9) = +	Motion(G10) = +
Motion(G11) = -	Loc(Scape, Pallets) = SP-PL-22	Loc(G1, G2) = G1-2-PL-1
Loc(G3, G4) = G3-4-PL-1	Loc(G5, G6) = G5-6-PL-1	Loc(G6, G7) = G6-7-PL-1
Loc(G8, G9) = G8-9-PL-1	Loc(G10, G11) = G10-11-PL-1	

Figure 5.1: Sample mechanical state of the QRG clock

5.2 State

To describe what is happening in an episode or event we use the *qualitative state* representation (de Kleer, 1975), an abstraction of the notion of state in classical mechanics. The parameters which might appear in a mechanical state are represented abstractly, enabling an infinite number of possible behaviors to be divided into a finite number of qualitatively distinguished behaviors. Changes in state correspond to changes in behavior.

A QM state has two components, corresponding to changes in space and time: a *kinematic component* and a *dynamic component.* The kinematic component consists of a set of places from a place vector representing the location of the objects in a kinematic chain. The dynamic component gives the motion of each object as a qualitative vector.

To develop the full qualitative state space of the mechanism we combine every possible place with every possible motion of every object which exists in that place. Typically this is an enormous number of states, but we will see in section 8.1.6 how this number can be reduced.

Definition 35 (State) ***A state** is the combination of all unconstrained motions of each object with all places in a place vector.*

Figure 5.1 shows a representative state from the QRG clock. Gear names begin with "G" followed by a number. Place names are composed from the objects constituting that place vocabulary, a "PL" (for place), and a number. Since there is only one degree of freedom there is only one dimension of motion for each object.

5.3 Transitions

A *Transition* maps a qualitative state into the set of qualitative states that can occur next. Just as state is a combination of dynamic and kinematic information, state transitions are the combination of the changes in the dynamic component and the kinematic component of the current state. Kinematic changes are determined by the resultant of motion on place in the place vocabulary for each object. Dynamic changes are determined by the result

		Force Component			
		+	0	−	?
	+	+	+	0	$+ \vee 0$
Motion	0	$+_i$	0	$-_i$	?
Component	−	0	−	−	$- \vee 0$
	?	$+ \vee 0$	?	$- \vee 0$	?

Table 5.1: Effects of force on next possible motion

of force and constraint on motion for each motion in the current state as discussed above. When multiple influences are conflicting or an influence has ambiguous results there may be several possible next states. Our definition of transitions is based on that of (Forbus, 1986).

Definition 36 (Transition functions) *The functions* `Before`: `s` $\rightarrow$ {*states*} *and* `After`: `s` $\rightarrow$ {*states*} *represent the states which can lead to and be reached from a state, respectively. That is,* $\forall$ *(*`s`, s_1, s_2 ϵ *States)* s_1 ϵ `Before(s)` *exactly when there is a transition from* s_1 *to* `s`*, and* s_2 ϵ `After(s)` *exactly when there is a transition from* `s` *to* s_2.

Determining state transitions involves the rules of QM we have developed thus far, plus the additional complication of simultaneous dynamic and kinematic interactions necessary to model collisions. Because we previously only examined dynamic and kinematic descriptions in isolation, this interaction did not arise. In the rest of this section we will examine the information necessary to determine what mechanical transitions can occur from a given state.

5.3.1 Dynamic Transitions

We base our theory of force affecting motion on *limit analysis* (Forbus, 1984). Without knowing the magnitudes of the forces involved, we can (with possible ambiguity) determine their affects. If force is qualitatively proportional to acceleration, acceleration directly influences motion, and force is the only thing affecting either acceleration or motion[1] — we can assert that force directly influences motion.

Once the net force in the current mechanical state is known, we decompose it into its component directions for each dimension. Table 5.1 shows how to predict the possible next motions for each force and current motion. A value of *ambiguous* is represented by "?". An ambiguous change or a choice of possibilities indicates a branch in the predicted next motion. Those transitions which occur in an instant are distinguished by the subscript "i". All other transitions require some interval of time, and their current values may persist in the next state.

5.3.2 Feedback

[1]There may be multiple forces, but we assume mass does not change.

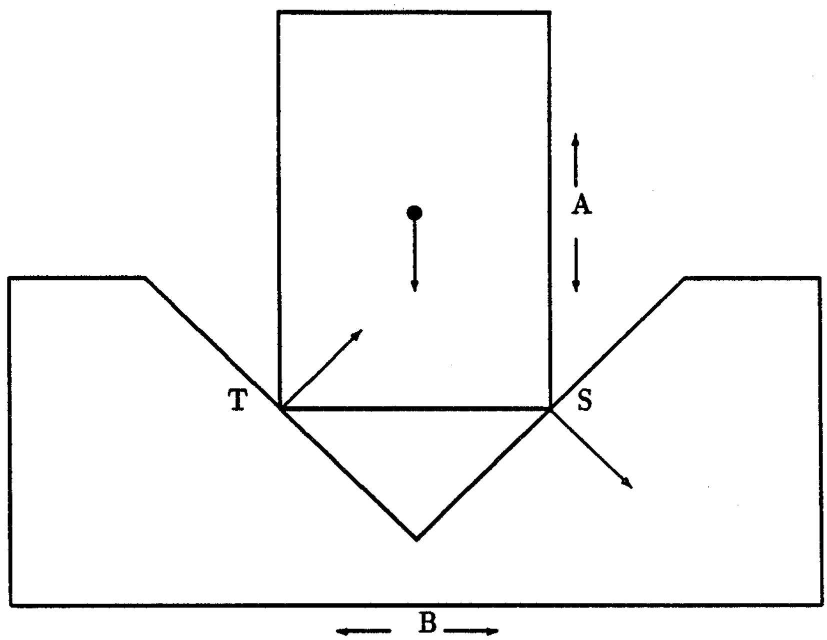


Figure 5.2: Force feedback example

It may be the case that the designer's arrangement of parts causes force to be transferred from an object back to itself through some chain of other objects. Typically this will cause the mechanism to jam. When the two forces act in opposite directions using the analysis we have developed thus far will yield an ambiguous net force. Consider Figure 5.2, where **A** can only move vertically and **B** can only move horizontally. If an external force is applied to **A** in the downward direction, there is a force transmitted to **B** at surface S which is to the left. This force in turn causes a force to be transmitted back to **A** on surface **T** which pushes **A** upward. Without knowing the relative magnitudes of the forces we might hypothesize that **A** may move up.

Every force represented to QM has associated causal information. This allows us to compare some forces with opposite signs. In addition to the qualitative arithmetic we know that opposite forces with the same cause will cancel. By including causal information in this example we know that the cause of **A** moving up is the same as the cause of **A** being pushed down so the net effect is no push.

Law 5 (Opposite Forces with Same Cause) *If the signs of the same component of two forces are opposite and both forces have the same cause, then the qualitative sum of these force components is zero.*

If the opposing forces were not equal in magnitude, there would be a net force transmitted back to the source which would tend to weaken the source. This weaking is a function of the source and would only go to zero when the magnitude of the source was zero. This law is only true of rigid objects because they transmit forces in both directions.

If a force with the same cause is transmitted by two different paths there is positive feedback and the mechanism may jam. When the force transferred back to an object acts in the same direction, the mechanism will still jam if the magnitudes of the forces are unequal; however, if the magnitudes of the forces are equal the mechanism will move freely. Positive feedback may be detected for further analysis by maintaining a list of the paths of objects which transmitted the force.

5.3.3 Collision

In this section we will explore the result of making new contact between objects. Modeling collision is a complex process in any theory of motion, so we resort to some simplifying assumptions. We will assume (1) the collision forces are great enough to overwhelm the external forces (This is a good assumption because the collision forces between rigid bodies are infinite.); (2) no two collisions will occur at exactly the same time; and (3) all collisions are inelastic.

The parameters to consider when bodies collide are mass, velocity, elasticity of collision, and rigidity of the objects. When two semi-rigid bodies come into contact they continue moving into each other for some interval. During this time, a restoring force arises that opposes the motion of these objects. When this force becomes great enough, the two objects stop moving relative to each other and depending on the type of collision, may begin moving away from each other.

When two rigid bodies transition from a state where they are not in contact to a state where they are in contact, a collision occurs. During a rigid body collision, a change in the motion of the two bodies may instantaneously occur in order to satisfy new constraints imposed by the new contact. The duration of the states both before and after the collision may persist for some interval.[2] Because we do not know the relative masses of the objects, the resultant motion is highly ambiguous. We can reduce this ambiguity by assuming the collisions are inelastic, and thus, the resultant mass is always greater than either of the original masses. The resultant motion of the coupled objects then is the qualitative vector sum of the directions of motion of the original masses.

5.4 Envisionment

A QM envisionment shows all possible behavioral sequences of a mechanism through changes in position and motion by determining the next possible transitions from every consistent mechanical state. We use the formal definition of envisionment from (Forbus, 1986).

Definition 37 (Envisionment) *An* `envisionment` *represents all possible qualitative states a particular system may take on and all legal transitions between them.*

Envisioning makes several types of reasoning possible such as "What happens next?"; "Can a behavioral sequence occur?"; and "Are there potentially dangerous states which can occur that I should be aware of?" Envisioning gives all possible activity of the system, not just a single path. Finally envisioning can serve as the basis for more complex reasoning systems, such as those discussed in section 11.2.

[2] This violates the instant/interval pair notion of time advocated by (Williams, 1986).

While an envisionment needs additional information to determine the histories that correspond to acceptable clock behavior, it recognizes that such behaviors exist and indicates several undesired behaviors. For example, in the clock it detects the pendulum beating repeatedly without the gear train moving and the pallets both clear of scape wheel, allowing the gear train to move freely.

There are also undesirable behaviors the envisionment recognizes that people might not consider, such as the recoil action of the escapement being powerful enough to drive the clock backwards some number of periods. Additional knowledge, not necessarily metric, can eliminate these behaviors. For example, we might include the knowledge that the forces transmitted by contact act slowly enough so that objects will not break contact due to the resultant motions. Chapter 9 shows envisionments for several mechanisms.

Part III

Practice

When investigating common sense reasoning, the intuitive nature of the task makes it easy to miss conceptual or inferential difficulties. The conclusions, after all, are obviously true. Implementation provides a crucial check of our theory, since the right consequences are not obvious to our machines. In the next chapters, we examine implementations of these theories in sufficient detail that they could be reconstructed by a competent AI programmer with reasonable effort.

Our implementation consists of two major programs, `Alex` and `Clock`. `Alex` reasons about what happens next from a given static snapshot. For mechanical analysis, given a configuration of objects, it determines the next possible instantaneous motions of these objects. It may also be applied to static analysis of relatively unconstrained objects. To determine the stability of a structure, for example, we confirm that no instantaneous motions are possible.

`Clock` is an envisioner for the rigid body mechanisms world. It uses the instantaneous motion analysis first implemented in `Alex` as a subsystem. `Clock` takes as input a metric diagram a which is the configuration space representations of rigid objects from Faltings, 1987b. It also requires a description of the external forces acting on the system. Its output is a complete collection of qualitative states and all kinematic and dynamic transitions between them.

The underlying rule engine used by these systems is *ATMoSphere*[3] (formerly ADB) which is based on an assumption-based truth maintenance system[4].

[3] Written by Forbus. (Unfortunately a description of *ATMoSphere* is not available at the time of this writing.)

[4] Written by de Kleer (de Kleer, 1986a; de Kleer, 1986b; de Kleer, 1986c).

Chapter 6

The Alex Program

`Alex`, the first implementation of the QM theory, determines the object freedoms and next possible motions from object geometries, current motions, and external forces. It solves a variety of static analysis problems, and is an initial step toward mechanism analysis.

Fahlman's BUILD program (Fahlman, 1977) used numerical methods to reason about the stability of block stacks during the construction of simple structures. "The heart of BUILD is the stability test, the module that looks at a state of the blocks world and decides whether anything is going to fall"(Fahlman, 1977). `Alex` can accomplish this stability analysis for many examples using purely qualitative methods.

6.1 Scene Representation

`Alex` requires knowledge of surfaces, contacts, fixed centers of rotation, fixed links (the reference frame), and external forces. Representing a problem for `Alex` requires a scene description and a set of situations. The scene description tells the number of dimensions, the individuals involved, the individuals' descriptions,[1] the individuals' surfaces, the external motion constraints, and the external forces. These descriptions are completely symbolic, using the vocabulary introduced previously. The situations provide kinematic information indicating which surfaces may be in contact at various times. Figure 6.1 gives the BNF form of `Alex`'s scene representation.

Beginning with the scene description, objects are named and represented by the set of their surfaces. Since force and motion is only transmitted by contact, an overall description of the shape of the object itself unimportant. Only the orientation of the surfaces which can come into contact and the direction from the surface to the object's center of rotation need be described. For example, we need not represent a design carved into the side of a gear, only the tooth surfaces. We say "`Surface(top (0, +) (0, --))`" which means a surface named "`top`" has its normal in the positive Y direction, and the direction from it to the center of rotation is in the negative Y direction.

Global constraints on motion, imposed by the reference frame, are associated with objects. These are distinguished according to whether they restrict translational or rotational motions. For example, `Translational-Constraint(gear (+ +))` means the gear object

[1] Individual descriptions are included for compatibility with other systems. All of our object descriptions specify that each object is a rigid body.

<problem description> ::= <scenario><state>
<scenario> ::= “(” **Defscenario** <number of dimensions> <name>
Individuals “(” <name>* “)”
Descriptions “(<object descriptions>* “)”
Surfaces “(” <list of object surfaces>* “)”
[<translational constraints>] [<rotational constraints>]
Forces “(” <force entry>* “))”
<object descriptions> :: = “(” **rigid-ob** <name> “)”
<list of object surfaces> ::= <name> “(” <surface entry>* “)”
<surface entry> ::= <name><translational direction><direction to center of rotation>
<translational constraints> ::= “(” <translational constraint entry>* “)”
<translational constraint entry> ::= “(” <name><translational direction>* “)”
<rotational constraints> ::= “(” <rotational constraint entry>* “)”
<rotational constraint entry> ::= “(” <name><direction to center of rotation>* “)”
<force entry> ::= “(” <name><translational direction>
<direction to center of rotation> “)”
<state> ::= “(” **Defstates** <configuration descriptor>* “)”
<configuration descriptor> ::= “(” <contact descriptor>* <force descriptor>*“)”
<contact descriptor> ::= “(” **Contact** <name><name> “)”
<force descriptor> ::= “(” **Force** <name><translational direction>
<direction to center of rotation> “)”
<number of dimensions> ::= *integer*
<name> ::= *string*
<translational direction> ::= *qualitative translational direction*
<direction to center of rotation> ::= *qualitative translational direction*

Figure 6.1: BNF scene description for **Alex**

is unable to move in any direction which would be into the +X and +Y quarter plane. `Rotational-Constraint(piston (+))` means the piston cannot rotate counter-clockwise.

Finally, the scene description may include external forces on objects, when forces are globally active. Forces are represented by their translational direction and the direction from the point of their activity to the object's center of rotation. For example, `Force(lever (- 0) (- -))` indicates the lever is being pushed down (– Y direction) and the direction to the center of rotation of the lever is in the –X, –Y direction. If its movement is not constrained we should predict that lever will move in the clockwise direction.

Following the scene description are a number of situations. These describe the kinematic and dynamic influences at an instant. The only dynamic influence we consider is force, and though we also discussed them in the scene description, often a force will not be constant or will change directions. For example, a hand may lift a block and set it down. The direction from the center of gravity to the center of rotation of a pendulum will change as it swings. Though we do not consider the transitions between these states, we are able to simultaneously represent and reason about the instantaneous effects of these changes.

The kinematic aspect of the situation consists of a number of surfaces which are in contact. These relations are stated as `Contact(a b)` which means that surface "a" is touching surface "b". `Alex` is not capable of automatically determining how contacts are made and broken; but rather, this information must be provided by some other agent.

6.2 Determining Activity

`Alex` assumes that every object is free to move in every direction and begins eliminating free directions according to the law of contact constraint (Figure 3.1).

Given objects with current motion, `Alex` tests to see which possible motions, if any, are consistent (Figure 3.2). If motion is possible `Alex` will determine all potential the directions of motions, which are the directions that yet need to be constrained to prevent motion.

Given forces acting on objects, it tests to see if the direction of the net force is consistent. If that direction is constrained, it tests to see if any direction which lies in the same half plane as the net force is consistent. This is similar to the law of force distribution (Figure 3.3), with an additional ordering on the resultant direction of motion. In typical mechanical links, all motions will be constrained except one and it is enough to distinguish the half plane of the resultant motion. For free objects this information needs to be augmented by a partial ordering of preferred motions.

The result of this analysis is a description of the possible motions of each object in each situation. This may be used for stability analysis: when the only possible motions are zero, the structure is stable. It may be used for mechanism analysis if an oracle were to tell the situation which resulted from each motion. In chapter 8 we will see how to build such an oracle and other considerations necessary for mechanism analysis.

Chapter 7

Extended Examples from Alex

This chapter presents several examples from **Alex** taken from blocks world and mechanism world problems. In the examples we will show the scene description, an intermediate view showing the free directions of motion, and the predicted resultant motions of the objects.

7.1 Blocks

7.1.1 Unsupported Block

Consider the block shown in Figure 7.1. The block has 12 surfaces (3 per side) and a single external force (gravity) acting at its center. A fixed table (not shown) only needs a single surface descriptor since it has no center of rotation. The description in Figure 7.2 shows two possible locations of this block, one where it is unsupported and the second where it has fallen onto the table. Entries which begin with a question mark are variables and may be replaced with any qualitative direction from the set of possible directions.[1]

In the first situation, when there are no contacts and the block is unsupported **Alex** determines the free directions of motions for all of the objects as shown in Figure 7.3.

Since **Alex** determined the block is free to move in all translational directions, when a force is applied to the block it will move in the direction of that force. **Alex** predicts that in the next instant the block will begin to fall in a downward direction (Figure 7.4).

In the second configuration, the bottom of the block has come into contact with the table, many of the previously free directions of motion of the block are eliminated.[2] The block is no longer free to move in any of the (0 −), (+ −), or (− −) translational directions and no longer free to rotate either (+) or (−) (Figure 7.5).

When the block is resting on the table **Alex** predicts no motion for any component. The structure is stable (Figure 7.6).

7.1.2 Arch

[1]Since every object must be free to move in at least one direction and preventing an object from standing still is not something one typically wants to do, by default **Alex** will not automatically constrain motion in the direction consisting of all zeros .

[2]Notice that **Alex** cannot predict this will happen as a consequence of the earlier situation. Our qualitative geometry contains no representation of position or location.

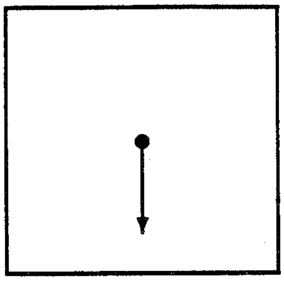


Figure 7.1: Block

The scene description of the arch show in Figure 7.7 is given in Figure 7.8. This is a 2 dimensional scene containing four objects: a cross bar, the left support block, the right support block, and a table. Each of these objects has several surfaces, but note that it is not necessary to represent every surface of every object, only those surfaces which may eventually come into contact with other surfaces.

Figure 7.7 shows an arch resting on a table where the table is completely constrained. The contact between the surface of the table and the bottom surfaces of the blocks prevent them from moving in the $(0\,-)$, $(+\,-)$, and $(-\,-)$ directions as well from rotating $(+)$ or $(-)$. Under the influence of gravity they will not move.

The cross bar is prevented from translational motion in the downward direction by the contact between its underside and either of the blocks. Since the center of rotation of the cross bar lies in a half plane which is 90 degrees counter-clockwise of the surface normal where the cross bar contacts block A, it is prevented from counter-clockwise rotation. Since the center of rotation of the cross bar lies in a half plane which is 90 degrees clockwise of the surface normal where the cross bar contacts block B, it is likewise prevented from clockwise rotation. Since these are the only motions the gravitational force can influence, the bar will not move and the structure is stable. Free directions of the arch are shown in Figure 7.9. The motions of the are shown in Figure 7.10.

7.2 Mechanisms

7.2.1 Gear Pair

Consider the gear pair shown in Figure 7.11. Each gear is constrained to rotational motion. This example illustrates way **Alex** propagates motion between solid objects. For any given motion of one of the gears we would like to be able to determine how the other gear will move.

Since we constrained everything except rotation the free directions of motion are as shown in Figure 7.13

Once the free directions of motion are determined **Alex** determines the consistent combinations of motions for each object in the system. For example if the driver gear turns

```
Block 2
      Individuals (block table)
      Descriptions ((rigid-ob block)
                    (rigid-ob table))
      Translational-Constraints ((table (?x1 ?y1)))
      Rotational-Constraints ((table (?r1)))
      Surfaces ((block (block-t-l (0 +) (+ --))
                       (block-t-m (0 +) (0 --))
                       (block-t-r (0 +) (-- --))
                       (block-l-t (-- 0) (+ --))
                       (block-l-m (-- 0) (+ 0))
                       (block-l-b (-- 0) (+ +))
                       (block-r-t (+ 0) (-- --))
                       (block-r-m (+ 0) (-- 0))
                       (block-r-b (+ 0) (-- +))
                       (block-b-l (0 --) (+ +))
                       (block-b-m (0 --) (0 +))
                       (block-b-r (0 --) (-- +)))
                (table (tb (0 +) (0 0))))
      Forces ((block (0 --) (0 0))
              (table (0 --) (0 0)))
Situations {}
           {Contact(tb block-b-l), Contact(tb block-b-m), Contact(tb block-b-r)}
```

Figure 7.2: Scene description of block and table

```
Translational-Freedom (Block (+ -))
Translational-Freedom (Block (+ 0))
Translational-Freedom (Block (+ +))
Translational-Freedom (Block (0 -))
Translational-Freedom (Block (0 0))
Translational-Freedom (Block (0 +))
Translational-Freedom (Block (- -))
Translational-Freedom (Block (- 0))
Translational-Freedom (Block (- +))
Translational-Freedom (Table (0 0))

Rotational-Freedom (Block (+)) = True
Rotational-Freedom (Block (-)) = True
Rotational-Freedom (Block (0)) = True
Rotational-Freedom (Table (0)) = True
```

Figure 7.3: Free directions of motion for block and table

clockwise (−) the motion is about an point located down from the contact indicated, which satisfies motion into the surface. From the set of motions the follower gear must undergo, the only one which satisfies rotation about an point located down left of the contact is motion counter-clockwise (+). In this example there are six consistent combinations of motion. There is no translational motion for either object in this example, so we will have omitted it from the descriptions (Figure 7.14).

7.2.2 Recoil Escapement

A recoil escapement is shown in Figure 7.15. The direction of the surface normal of the pallets at their point of contact with the scape wheel is **N**, the direction to the center of rotation of the wheel is **W**, and the direction to the center of rotation of the pallet is **P**. In this type of escapement, when the pallet arm contacts the scape wheel, the wheel is not constrained by the pallet arm. As a result the pallet's continued swing drives the wheel backward, causing the entire clock mechanism to move backward (recoil). This movement continues until the force of the clock drive is sufficient to overcome the inertia of the pallet. The gear shown in this example is in tooth contact with a gear connected to the scape wheel. `Alex` begins with the scene description shown in Figure 7.16. The free motions of the objects are shown in Figure 7.17.

In this configuration there are ten consistent combinations of motions for the objects. Rather than show all of these cases we will only investigate the recoil case. If we further tell the program the pallet is moving counter-clockwise there is only one consistent solution as shown in Figure 7.18.

7.3 Discussion

`Alex` only handles single state reasoning, and it will be very difficult to generalize this to multi-state reasoning. When position is changed through rotation the surface normals will change directions. We cannot determine whether this represents a qualitative change or not without re-representing the entire mechanism.

When ambiguity arises one can either take a liberal or conservative bias depending on the problem being solved. If we are trying to determine possible motions, we consider vectors which ambiguously fall outside the half plane are actually outside the half plane. For example, to maximize safety in determining stability, we assume that ambiguously constrained directions are underconstrained but ambiguous directions of motions may allow movement.

When rotation is not constrained we must either know the center of rotation or postulate a few locations for this point at the time of representation and test all of these locations. When problems are symmetric, or the majority of weight is shifted to one side of a support, determining centers of rotation may be obvious, but if it is not obvious, finding these might require more mathematical analysis than determining stability. Consider the arrangement shown in Figure 7.19. If the center of rotation lies above the support block `Alex` will say the structure is stable, but if the center of rotation lies elsewhere `Alex` predicts it may rotate.

```
Translational-Motion (Block (0 -))
Translational-Motion (Table (0 0))
Rotational-Motion (Block (0))
Rotational-Motion (Table (0))
```

Figure 7.4: Motions for block and table

```
Translational-Freedom (Block (+ 0))
Translational-Freedom (Block (+ +))
Translational-Freedom (Block (0 0))
Translational-Freedom (Block (0 +))
Translational-Freedom (Block (- 0))
Translational-Freedom (Block (- +))
Translational-Freedom (Table (0 0))
Rotational-Freedom (Block (0))
Rotational-Freedom (Table (0))
```

Figure 7.5: Free directions of motion for block and table

```
Translational-Motion (Block (0 0))
Translational-Motion (Table (0 0))
Rotational-Motion (Block (0))
Rotational-Motion (Table (0))
```

Figure 7.6: Motions for block and table

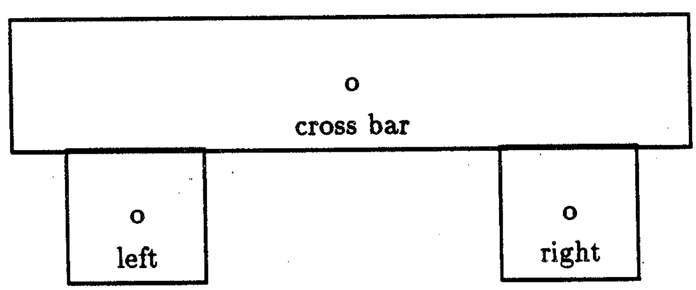


Figure 7.7: Arch

```
Arch 2
    Individuals (cross-bar left right table)
    Descriptions ((rigid-ob cross-bar)
                  (rigid-ob left)
                  (rigid-ob right)
                  (rigid-ob table))
    Surfaces ((cross-bar (cb-1 (0 --) (+ +))
                         (cb-2 (0 --) (-- +)))
              (left (lt-t-l (0 +) (+ --))
                    (lt-t-m (0 +) (0 --))
                    (lt-t-r (0 +) (-- --))
                    (lt-b-l (0 --) (+ +))
                    (lt-b-m (0 --) (0 +))
                    (lt-b-r (0 --) (-- +)))
              (right (rt-t-l (0 +) (+ --))
                     (rt-t-m (0 +) (0 --))
                     (rt-t-r (0 +) (-- --))
                     (rt-b-l (0 --) (+ +))
                     (rt-b-m (0 --) (0 +))
                     (rt-b-r (0 --) (-- +)))
              (table (tb (0 +) (0 0))))
    Translational-Constraints ((table (?x1 ?y1)))
    Rotational-Constraints ((table (?r1)))
    Forces ((cross-bar (0 --) (0 0))
            (left (0 --) (0 0))
            (right (0 --) (0 0))
            (table (0 --) (0 0))
            )
(defState ((Contact tb lt-b-l) (Contact tb lt-b-m) (Contact tb lt-b-r)
           (Contact tb rt-b-l) (Contact tb rt-b-m) (Contact tb rt-b-r)
           (Contact cb-1 lt-t-l) (Contact cb-1 lt-t-m) (Contact cb-1 lt-t-r)
           (Contact cb-2 rt-t-l) (Contact cb-2 rt-t-m) (Contact cb-2 rt-t-r)))
```

Figure 7.8: Scene description of arch

```
Translational-Freedom (Cross-Bar (+ 0))
Translational-Freedom (Cross-Bar (+ +))
Translational-Freedom (Cross-Bar (0 0))
Translational-Freedom (Cross-Bar (0 +))
Translational-Freedom (Cross-Bar (- 0))
Translational-Freedom (Cross-Bar (- +))
Translational-Freedom (Left (+ 0))
Translational-Freedom (Left (+ +))
Translational-Freedom (Left (0 0))
Translational-Freedom (Left (0 +))
Translational-Freedom (Left (- 0))
Translational-Freedom (Left (- +))
Translational-Freedom (Right (+ 0))
Translational-Freedom (Right (+ +))
Translational-Freedom (Right (0 0))
Translational-Freedom (Right (0 +))
Translational-Freedom (Right (- 0))
Translational-Freedom (Right (- +))
Translational-Freedom (Table (0 0))
Rotational-Freedom (Cross-Bar (0))
Rotational-Freedom (Left (0))
Rotational-Freedom (Right (0))
Rotational-Freedom (Table (0))
```

Figure 7.9: Free directions of arch

```
Translational-Motion (Cross-Bar (0 0))
Translational-Motion (Left (0 0))
Translational-Motion (Right (0 0))
Translational-Motion (Table (0 0))
Rotational-Motion (Cross-Bar (0))
Rotational-Motion (Left (0))
Rotational-Motion (Right (0))
Rotational-Motion (Table (0))
```

Figure 7.10: Motions of arch

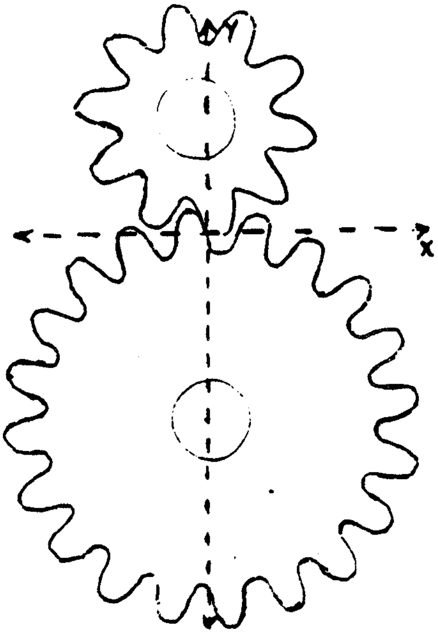


Figure 7.11: Gear pair

```
Gears2
     Individuals :  (driver, follower)
     Descriptions :  rigid-ob(driver), rigid-ob (follower)
     Translational-Constraints :  driver (?x1 ?y1), follower (?x3 ?y3)
     Surfaces :  Follower (f-1 (- -) (0 +)), (f-2 (0 -) (0 +)),
                          (f-3 (+ -) (0 +))
                 Driver (d-1 (+ +) (0 -)), (d-2 (0 +) (0 -)),
                          (d-3 (- +) (0 -))
     CONTACT (f-1 d-1)
```

Figure 7.12: Scene description of gear pair

```
Translational-Freedom(driver (0 0))
Translational-Freedom(follower (0 0))

Rotational-Freedom(driver (+))
Rotational-Freedom(driver (-))
Rotational-Freedom(driver (0))
Rotational-Freedom(follower (+))
Rotational-Freedom(follower (-))
Rotational-Freedom(follower (0))
```

Figure 7.13: Freedom of gear pair

```
Rotational-Motion(driver (+))
Rotational-Motion(follower (+))

Rotational-Motion(driver (+))
Rotational-Motion(follower (-))

Rotational-Motion(driver (+))
Rotational-Motion(follower (0))

Rotational-Motion(driver (-))
Rotational-Motion(follower (+))

Rotational-Motion(driver (0))
Rotational-Motion(follower (+))

Rotational-Motion(driver (0))
Rotational-Motion(follower (0))
```

Figure 7.14: Motions of gear pair

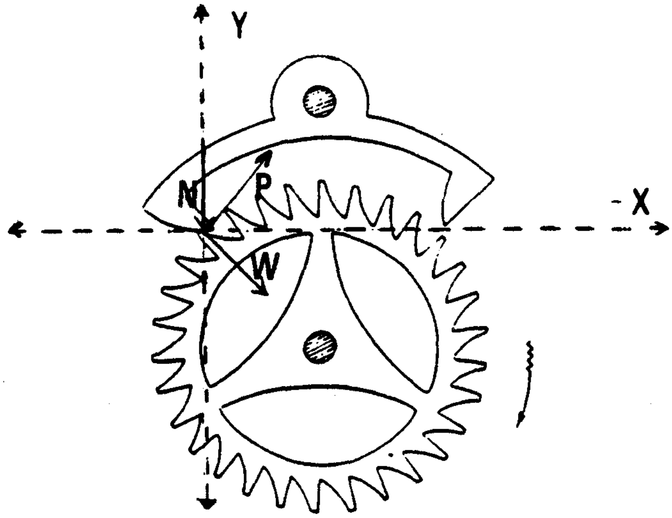


Figure 7.15: Recoil escapement

```
Recoil 2
       Individuals :  scape-wheel, pallet, gear
       Descriptions :  rigid-ob(scape-wheel), rigid-ob(pallet), rigid-ob(gear)
       Translational-Constraints :  scape-wheel(?x1 ?y1), pallet(?x2 ?y2),
                                    gear(?x3 ?y3)
       Surfaces :  Pallet (p-1 (0 -) (+ +)), (p-2 (- 0) (- +))
                   Scape-wheel (s-w-1 (0 +) (+ -)), (s-w-2 (+ 0) (- -)),
                               (s-w-b-1 (- -) (0 +)), (s-w-b-2 (0 -) (0 +)),
                               (s-w-b-3 (+ -) (0 +))
                   Gear (gear-top-1 (+ +) (0 -)), (gear-top-2 (0 +) (0 -)),
                        (gear-top-3 (- +) (0 -))
       CONTACT (s-w-b-3 gear-top-3), (s-w-1 p-1)
```

Figure 7.16: Scene description of recoil escapement

```
Translational-Freedom(scape-wheel (0 0))
Translational-Freedom(pallet (0 0))
Translational-Freedom(gear (0 0))

Rotational-Freedom(scape-wheel (+))
Rotational-Freedom(scape-wheel (-))
Rotational-Freedom(scape-wheel (0))
Rotational-Freedom(pallet (+))
Rotational-Freedom(pallet (-))
Rotational-Freedom(pallet (0))
Rotational-Freedom(gear (+))
Rotational-Freedom(gear (-))
Rotational-Freedom(gear (0))
```

Figure 7.17: Free directions of recoil escapement

```
Translational-Motion(scape-wheel (0 0))
Translational-Motion(pallet (0 0))
Translational-Motion(gear (0 0))

Rotational-Motion(scape-wheel (+))
Rotational-Motion(pallet (+))
Rotational-Motion(gear (-))
```

Figure 7.18: Motions of recoil escapement

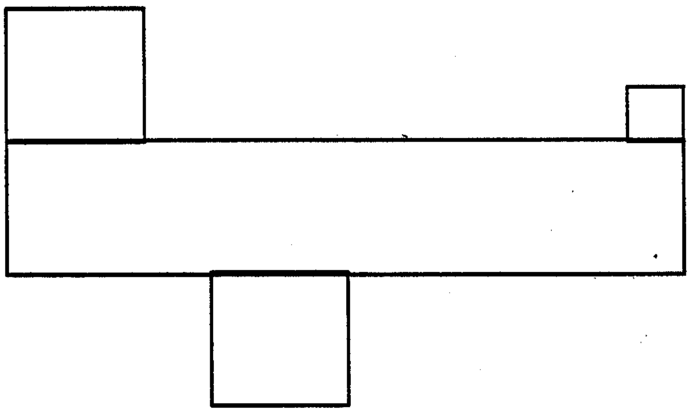


Figure 7.19: Ambiguously stable structure

Chapter 8

The Clock Program

Clock begins with the principles of blocking and pushing developed for Alex as a basis for determining motion changes. While Alex was only concerned with instantaneous motion changes, Clock considers changes that require some interval of time. We combine this with a richer geometric representation in the form of a configuration space, which allows us to determine how location changes over time.

Representing a scene for Clock requires entering drawings of the parts involved and indicating their relationship to each other. Scene descriptions are limited to two dimensional mechanisms each having one degree of freedom.[1] The user may indicate external forces that act on individual parts when they are in specific locations.

The behavioral descriptions produced by Clock are in the form of complete envisionments. These show not only next possible behavior, but all possible behaviors of the system from any legal configuration. Intermediate stages of analysis produce place vocabulary descriptions of the pairwise interactions of the component objects.

8.1 Kinematic Considerations

Our kinematic analysis is based on a place vector representation of the current configurations of all components of the mechanism. The algorithm shown in Figure 8.1 computes this place vector. Each stage will be discussed in some detail in the subsections which follow. Step 3.1 will be discussed at the end of the section so that the reader understands why abstraction is necessary.

8.1.1 Configuration Space

Clock uses the configuration space representations constructed by the program described in (Faltings, 1987b) as one of its inputs. Those representations serve as a metric diagram for constructing symbolic, spatial representations. We briefly describe that work here because an understanding of configuration space is essential to understanding the rest of the chapter.

Kinematic considerations arise from the geometry of the objects involved. Analysis begins from drawings of the parts of the mechanism. The boundary of a part consists of straight line segments and arcs. Explicit in these drawings is the concept of solid and open

[1] This limitation is imposed by the current implementation of the configuration space transformation.

1. Enter drawings of each component of the mechanism and
   indicate objects that form kinematic pairs.
2. For each kinematic pair, compute their configuration space representation,
3. For each configuration space
   3.1. Perform abstraction
   3.2. Generate place vocabulary
   3.3. Determine transitions between places
4. Establish kinematic chains
5. Combine places along kinematic chains to determine consistent place vectors

Figure 8.1: Algorithm for generating kinematic states

regions.[2] Parts are joined to form kinematic pairs by entering two objects in a drawing and their relative displacements. Finally the free directions of motion must be indicated, either the center of rotation or a vector parallel to the translational motion. These drawings serve as the basis for the configuration space representations constructed by the program described in Faltings, 1987b. The configuration space locations in which the objects are overlapping are distinguished from locations in which the objects are not in contact. The boundary between these regions represents regions in which the objects are in contact.

8.1.2 Building the Place Vocabulary

The configuration space representations of object pairs serve as the metric diagram for generating place vocabularies. `Clock` constructs place vocabularies from these configuration space representations in four broad stages corresponding to each of the divisions introduced in section 4.2.

- CSEG's represent the boundaries between free and blocked space. These may be determined directly from the configuration space. `Clock` groups adjacent CSEG's which have the same surface normal, and separates those which differ.
- FDS's represent distinctions of free space. They begin where the constraint segment's surface normal qualitatively changes directions, since this corresponds to a possible qualitative change in behavior. Since we are concerned with motion changes, they continue parallel to all possible directions of motion of each object until they contact another surface.
- Places where free space divisions intersect constraint segments and other free space divisions are marked as joins.

[2] In (Forbus, 1981) those were called parity.

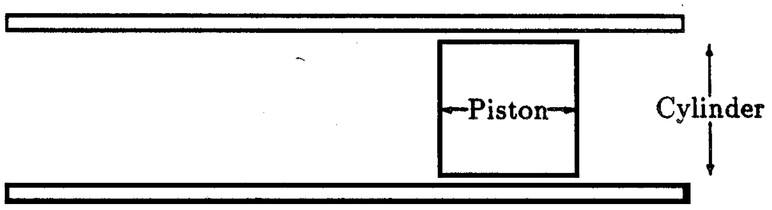


Figure 8.2: Piston in a cylinder (Arrows indicate free directions of motion)

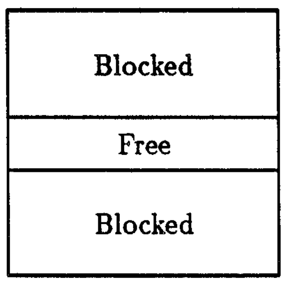


Figure 8.3: Configuration space for a piston in a cylinder

- Finally, full faces are detected by following paths of joins, CSEG's, and FSD's to form the smallest enclosing path of a region.

Because a region may not be fully enclosed, our program must look across free space from each join on a path to see if the opposite surface is part of the path. If not, the two paths are merged as borders of the same full face. For example, if a piston had only horizontal motion and its enclosing cylinder had only vertical motion (see figure 8.2), we might generate the configuration space shown in figure 8.3. Though both the upper and lower surfaces border the same full face region, there is no path of joins, CSEG's, and FSD's from the upper to the lower surface.

8.1.3 Finding Place Transitions

Movement of the objects in some direction may cause them to transition to the next adjacent place. Adjacency is the last piece of information we need from the metric diagram and having determined this, we are done with quantitative representations.

For each place and each motion we compute the set of next possible places from the partitioned form of the configuration space. Motions that would enter blocked space (i.e., the half plane centered on the reverse surface normal of a CSEG) are constrained, which means that their next place will be the same as the current place. If the width of the current place is zero in one of the directions of motion, the transition will occur instantaneously. Otherwise the transition will require some interval of time.

There are five considerations to determining the next adjacent place in the metric diagram. (1) For lines, such as CSEG's and FSD's, the normal defines a half plane where any direction into that half plane will cause a transition into the adjacent full face. (2) At each end of a line is a join which will be reached by motion along (or opposite to) the direction of the line segment. (3) The transitions from a join to a full face are indicated by the intersection of the half planes of the normals of the lines which meet at a join. (4) Places which have finite extent in some direction may remain in that place for motion in that direction. (Places with no extent in some direction must instantly leave that place for motion in that direction.) (5) Transitions from full faces as well as transitions from joins to lines are handled by reciprocity, that is, if a transition is installed from an object to a full face, a transition is installed in the opposite direction from the full face to the object.

The place vocabulary consists of the symbolic description of the joins, FSD's, CSEG's, and full faces. Each place is uniquely identified so that properties can be associated with it. For example, the adjacency of places in any given direction may be represented as `(Adjacent place1 place2` $<$ *direction* $>$`)`. A place representing a constraining surface has surface normals, and if there is more than one surface normal (i.e., a corner), we indicate whether they form an concave angle. Other information may be associated with a place, such as external forces.

Place vocabulary representations of kinematic pairs may be precomputed in the above manner and then stored. This allows reusability of kinematic pairs in various mechanisms and facilitates the development of a kinematic library.

8.1.4 Forging Kinematic Chains

The user indicates objects which are common between kinematic pairs. This includes objects which are rigidly coupled and share some direction of rotation or translation (i.e., two gears sharing a common shaft). This information causes `Clock` to establish a linkage between motions and forces of the solid objects.

`Clock` uses linkage and kinematic pairs to form kinematic chains. The kinematic chain is the transitive closure of kinematic pairs and common links.

8.1.5 The Place Vector

Recall that a place vector consists of one place from each place vocabulary along the kinematic chain. For each kinematic chain, we combine each place from a constituent pair's place vocabulary with each place from each of the other constituent place vocabularies. The resultant choice set is a set of place vectors, each of length equal to the number of kinematic pairs in the kinematic chain. The number of different place vectors is bounded by the product of the number of different places in the constituent place vocabularies. This set can be large, but in section 8.1.6 we will see how the size may be reduced to make analyzing complex mechanisms tractable.

At this point the user may enter further constraints on the place vectors based on outside analysis. These constraints have the effect of limiting the allowable choice combinations. For example, if a given gear (A) has a constant ratio with respect to some other gear (B), we might make statements of the form $Position(A) = x \Longleftrightarrow Position(B) = y \vee z$. This says that the first gear will only be in a given position if the second is in one of two positions. Because this is based on information not available to the system, the exact positions the user wants to specify may not be part of the current place vocabulary. To generate a new place vocabulary with this place one may need to go back to the metric descriptions and add further place distinctions.

8.1.6 Abstracting the Place Vocabulary

In this section we explore the effect of abstracting the configuration space on the size of the place vocabulary, showing how abstraction makes qualitative reasoning about complex mechanisms tractable. This is discussed out of the order in which it appears in the algorithm shown in Figure 8.1 because it is necessary that the reader have a clear idea of the problem in order to appreciate the necessity of abstraction.

A key problem in qualitative spatial reasoning is finding the right level of detail to support the needs of different reasoning methods. Analysis of failures may involve describing every surface imperfection, while gaining an initial understanding of overall device behavior is facilitated by abstracting away such details. The type of abstractions we need cannot be made by a purely qualitative analysis. When analyzing a single kinematic pair the place vocabulary may be copiously detailed, allowing intensive search for behavioral anomalies, but at the mechanism level this space must be condensed to provide tractability while still preserving overall behavior.

For example, the full place vocabulary for the clock escapement given in figure 4.1 consists of over 1300 places. This by itself is not an unwieldy number of distinct locations to consider if we wanted a detailed analysis of this pair. But when combining that information with 6000 to 60,000 places for each gear pair, the clock becomes intractable before we even start to consider dynamics. Information about periodicly recurring patterns[3] reduces the number of places to 96 for the escapement and 16 for a typical gear. But since our clock has 6 gear pairs, its entire place vocabulary after this optimization would consist of about 1,600,000,000 places.

To produce a detailed analysis we would like to preserve information about how a qualitatively unique surface on the original part interacts with each surface on its opposite pair. For example, at one level of detail we may be interested in knowing what behavior the fore face of a gear tooth has on the top of the opposite gear tooth. But when reasoning about long kinematic chains we cannot possibly keep track of where each surface comes into play. At the most detailed level CSEG's may be distinguished by a labeling of the object's surfaces which gave rise to them. Alternatively, the most abstract distinction of CSEG's is by the orientation of their surface normals because this orientation restricts the possible motions of the objects and determines the direction of forces transmitted by contact. Adjacent CSEG's with qualitatively equivalent surface normals may be represented by a single place in the place vocabulary.

This abstraction collapses all adjacent surfaces with qualitatively equivalent surface normals into one functionally equivalent surface which reduces the number of places on a

[3] Faltings handles recurring surface patterns by recording repetitive surfaces and performing the configuration space transformation only once for all of these surfaces. (Faltings, 1987b for details.)

gear to 12 and on the escapement to 80 and reduces the overall place vocabulary to about 240,000,000 places.

By reducing the resolution (enlarging the grain size), interactions between small irregularities on surfaces may be ignored. For example, the gear depicted in Figure 4.2 has certain interactions between the surfaces which prevent it from turning in the "up left" direction for a small interval. If this interval is below some ϵ it may be ignored at the risk of loss of accuracy. This reduces the number of places on a gear to 3 and the number of places on the escapement to 50. The total number of places is now 36,450.

When resolution is reduced, the gap between parts also becomes less apparent. If the gap size between two parts is less than some ϵ a new place is created which is bounded on two sides, conceptually this eliminates the play between the two parts and collapses three distinct places into one. It is important that only gaps be eliminated this way and not blocked space since that would alter the mechanisms behavior.

These abstractions allow us to reduce the number of places when considering gear behaviors to one. (The escapement remains 50, which is the total size of our place vocabulary.) The practical effect of this is that arbitrarily long gear chains contribute no more to the complexity of the mechanism than a single gear pair. This corresponds to the intuitive notion of a gear as producing a single constant behavior and validates other qualitative analysis' rules such as "A and B form a parallel gear pair if their only possible motion are two coordinated rotations."(Joskowicz, 1987a) However, our result was obtained from a first principles, geometric analysis. No "knowledge engineering" for specific parts or chains of parts is required.

The gear has been a basic element of machinery from its earliest beginnings (Dudley, 1969), and in some sense it is well understood. When inspecting a power train, simply observing gears gives one some general expectations of their behavior. However, explaining why gears bind requires a focus on the individual parts and more sophisticated observations of their interactions. A purely qualitative description would not benefit from additional observation, but by allowing new metric information to modify our symbolic representation we can construct a new qualitative description which depicts the conditions for gears to jam. Conversely, if we considered every possible way gears can jam we would never understand the overall behavior of the mechanism.

8.2 Adding Dynamic Considerations

The dynamic parameters we consider are force and motion. Force is the only influence on motion. Since an external force may only be active when an object is in limited positions we must consider the geometry and associate force with location. The algorithm shown in figure 8.4 shows how we compute the dynamic states of the system and combine it with the place vector to produce the mechanical states.

8.2.1 External Forces

We allow the user to associate external forces either with an object (for example, a spring may always push a certain gear clockwise) or with objects in various positions (for example, in certain positions a pendulum is pulled counter-clockwise by gravity, in others clockwise). This is done by associating forces with particular places in the place vocabulary. Associating

1. Input external forces associated with objects
2. For all objects, assume all possible motions
3. For each place vector
    3.1 For each object in the place vector
        3.1.1. For every legal combination of place and motion
            3.1.1. Compute the mechanical state of the system

Figure 8.4: Algorithm for generating mechanical states

force in various positions involves adding new metric information indicating where the force changes. This metric information is mapped back onto one of the configuration spaces in which the object participates, and the places indicated by this mapping will imply this force. Because these considerations fall outside any previous analysis, to cleanly associate a force with a place it may be necessary to add further divisions to the configuration space and recompute the place vocabulary. These new divisions are handled exactly the same as FSD's, except they extend throughout the space rather than stopping at a surface.

To transmit force between two rigid bodies they must be in contact (i.e., the current place must correspond to a CSEG). A force will be transmitted to a second object only if that force or motion is directed into the open half plane centered on the reverse surface normal of the contact. The direction of the resultant force is along the reverse surface normal despite the direction of the original force. (Section 3.2 formalized this.)

8.2.2 Individual Motion

Since we assumed each object has only one degree of freedom, the set of possible directions of motion for each is $\{+, 0, -\}$. Unless otherwise indicated we assume each object free to move in any of these directions. If we were to generate the dynamic state of the system it would have an upper bound of 3 to the power of the number of objects; however, we need not generate dynamic states explicitly.

Once the kinematic states are generated, these partial states are augmented by dynamic information to directly generate the set of complete possible states. Kinematic constraints are very powerful. Not all combinations of motion will be consistent. By finding consistent augmentations to consistent kinematic constraints much effort is saved. Further, all objects in the scene may not be part of the same kinematic chain and will have no control over the behaviors of others. These are best analyzed separately, but this distinction can only be realized through kinematic considerations.

8.2.3 Determining Mechanical States

Mechanical states are formed from consistent combinations of each place in each place vector with the allowable motions of each object described by that place vector. Since adjacent

1. For each mechanical state,
    1.1. Gather the applicable forces.
        1.1.1. Determine how forces propagate via connectivity
        1.1.2. Determine net force by qualitatively adding multiple forces
    1.2. Find kinematic transitions through motion and information associated with the place vocabularies
    1.3. Find dynamic transitions via limit analysis
    1.4. For each transition
        1.4.1. Calculate the resultant state from the hypothesized changes
            1.4.1.1. Enforce linkages
            1.4.1.2. Calculate the effects of collisions on motions
        1.4.2. If this state matches one previously generated, install a transition
        1.4.3. Otherwise, ignore the transition because it is inconsistent

Figure 8.5: Algorithm for computing state transitions

objects will constrain motion, we traverse adjacent kinematic pairs of the place vector *in order*, as defined by the linkage relation, combining the possible motions for each kinematic pair with the places described by the vector. Allowable motions are those which are not constrained by a previous choice or by a location. For example, if a linkage were set up between the motion of two objects and one of these objects had its motion predetermined then the motion of the second must be the same as that of the first. If there is more than one kinematic chain for the mechanisms there will be multiple sets of mechanical states independently describing different parts of the mechanism.

8.3 Transitions

Until now we have only considered static behaviors of the mechanism. We determined consistent combinations of place and motion without knowing how they can change. In this section we consider the transitions between place and motion which lead to behavioral descriptions of mechanical devices.

Just as state is a combination of dynamic and kinematic information, state transitions are the combination of the changes in the dynamic component and the kinematic component of the current state. The algorithm shown in figure 8.5 shows how we compute behavioral changes.

8.3.1 Force Propagation

Because we associated forces with places (section 8.2.1), `Clock` can determine the active external forces in a mechanical state by inspecting kinematic component. These external forces propagate through the kinematic chain either by linkage or by contact. Section 3.3 showed how forces propagate through contact. Those concepts are implemented in pattern directed rules. Forces are added by table loop-up (see Section 2.3).

8.3.2 Kinematic Transitions

`Clock` determines the set of next possible places for each place vocabulary in a place vector. When neither object described by a given place vocabulary has current motion, the next place will be the same as the current place. However, if either of the objects are in motion the set of next possible locations is determined by inspection of the place transitions under motion (as shown in section 8.1.3). If the set of possible transitions does *not* include the current place, this change will occur in an instant. If the current place is included in the set of possible transitions, this transition may require some interval to occur.

This timing information is important because changes which occur in an instant happen before those which occur in an interval (de Kleer & Bobrow, 1984; Williams, 1984). Upon detection of an instantaneous kinematic transition `Clock` hypothesizes a new state which consists of this change of place and all other places identical to their previous values. If all changes in place require some interval, the next possible locations are all combinations of the changes in place including those which remain in the current place.

8.3.3 Dynamic Transitions

If the kinematic components are identical but there are no forces active on any parts of the system, all current motions persist. If the kinematic component of the hypothesized next state is not identical to the kinematic component of the current state, all current motions persist. Because any qualitative change involves either achieving or leaving some threshold value, both of which will occur in an instant, we assume that changes in contact and changes in motion will not occur simultaneously without one causing the other.

If the kinematic components are identical and there are active forces on one or more parts of the system, we qualitatively add the forces on each component. During this addition forces with opposite direction but identical cause will cancel (see section 5.3.2).

If there is still a net force we use limit analysis (section 5.3.1) to determine how the force will affect current motions. Force will directly influence the direction of motion. This means that an object moving in some direction with force applied in the opposite direction may eventually stop moving; an object moving in some direction with force applied in the same direction may continue moving; an object not moving in some direction will instantaneously begin moving in that direction if a force is applied along that direction; and an object with no force applied will not change motion. When multiple influences are consistent or an influence has ambiguous results, there may be several possible next states. The next behavior may be any consistent combination of current and changed behaviors.

8.3.4 Enforcing Consistency

Though the transitions discussed thus far will result in a new mechanical state we have made no effort to insure that this state is consistent. For example linkages between places and motions need to be enforced. Any change to one object of the linkage will instantly cause the other to change.

Collision is somewhat more difficult. If our hypothesized mechanical state includes two objects which are in contact, but have inconsistent motions (the hypothesized motions of one or both objects is constrained) we have to resolve this conflict. The resultant direction of motion after collision is the qualitative sum of the motions of the two individual objects prior to the collision.

After determining the direction of the objects after the collision, the resultant motion must be compared with the motion of adjacent objects along the kinematic chain. If there is a violation between the newly determined motions of the objects involved in collision and other objects in the chain (violation of the law of transfer of motion), the motion of the object furthest from the collision is modified until this conflict is resolved. This check continues along the chain as long as any modification of motion has been made.

The modification of motion is based on a notion of nearness similar to a quantity space. We assign some ordering to the set of all directions and continue to select the nearest direction(s) as possible motions until the conflict is resolved or we exceed some threshold. For the problems we are solving that threshold is the zero vector.

Definition 38 (Nearness) *The directions nearest to a given vector,* **v** *are respectively as follows:*

- $\{v\}$
- $\{x \mid$ *Open-Half-Plane(v, x)*$\}$
- $\{x \mid$ *Zero-Vector(x)*$\}$
- $\{x \mid$ *Rotate-90(v, x, r)*$\}$, *where Not(Zero-Vector(r))*
- $\{x \mid$ *Open-Half-Plane(-v, x)*$\}$
- $\{-v\}$

8.3.5 Generating the Envisionment

`Clock` produces the complete envisionment of a mechanism consisting of all mechanical states and all transitions between these states. Once we have hypothesized a next state we search the previously generated set of all possible mechanical states to determine if this one matches any of them. If so we install a transition between these two states. If not the transition is ignored since the set of mechanical states generated previously is complete.

Chapter 9

Extended Examples from Clock

In this chapter we provide examples of mechanism envisionments which have been generated by the `Clock` program. The description of each example consists of the following:

- The arrangement of kinematic pairs
- The configuration space representation of these pairs
- The graphic representation of the place vocabulary
- A history taken from the envisionment

Complete envisionments for these devices are given in the appendices. The configuration space representations shown here were produced by the program described in (Faltings, 1987b). Recall that the direction of positive rotation is counter-clockwise. Thus motion in the positive X or Y directions in configuration space will represent a counter-clockwise motion of the corresponding part. When objects have rotational freedom these configuration spaces will wrap around (i.e., a rotation of 360° corresponds to a 0° rotation). In the caption, the object described by the horizontal axis is given first and the vertical axis second.

We will use the simplest place vocabulary representation of each mechanism after applying the abstractions discussed in section 8.1.6. For example, we will only investigate one of the periodically recurring patterns. This results in a much simpler display, but the reader must recognize this display repeats periodically. Referring back to the original configuration space representation will help one recognize the extent of the period.

9.1 Clock

One motivation for this thesis was to automatically produce qualitative behavioral descriptions of a complete mechanical clock. Though we have already seen partial analysis of this problem, in this section we present the overall mechanical clock analysis.

The QRG clock (shown in figure .1) consists of eleven gears (forming six gear pairs), a scape wheel, anchor pallets, a ratchet wheel, and a pawl. Figure 9.1 shows a flat layout of the parts of this clock, grouped into kinematic pairs. Because we are only concerned with rigid bodies, the spring is treated as an external force pushing gear 6 counter-clockwise and the ratchet wheel in the opposite direction. These pairs serve as the basis for generating configuration space representations.

9.1.1 Configuration Space

Figures 9.2 through 9.9 show the configuration space representations of the clock's kinematic pairs. Since each of the objects in the clock have a single rotational degree of freedom, each dimension of the configuration space corresponds to a rotation of an individual object. For example, in Figure 9.2 the pallets begin in an inverted position. A rotation of the pallets by 180° would correspond to a horizontal line in configuration space at half the height of the vertical axis. The region of free space in this area represents the normal operating region of this escapement, with the pallets facing downward.

Areas marked "blocked" represent configurations where the two objects would overlap, if that were possible. Areas marked "open" represent the play between the objects, where they are not in contact. Solid lines represent the divisions between free and blocked space corresponding to configurations where the objects are in contact. Labeling of open and blocked areas for the gears is omitted because the open area is too small. It corresponds to the smallest space between the solid lines; the larger spaces between the lines are blocked.

These configuration space representations are the inputs to the `Clock` program. While unintuitive, configuration space representations are preferable, because motion of solid objects may be thought of as a point moving among fixed obstacles. This simplification of representation is sufficient to indicate where the objects touch, the shape of the contact, and which locations are adjacent under motion.

9.1.2 Place Vocabulary

Figures 9.10 and 9.14 through 9.20 show the place vocabularies in configuration space. Figures 9.11 through 9.13 show representative arrangements of the escapement for each place in the central operating region. Notice how the abstractions have simplified the shape of the configuration spaces. Curves have been replaced by lines having the same qualitative slope, complex paths have been smoothed, and some of the play between parts has been eliminated. In particular a single place suffices to describe any configuration of a gear pair.

The place vocabulary consists of the spatial divisions of the configuration space and the next spatial division(s) we will reach for motion in each direction. The borders between free and blocked space in the configuration space (CSEG's) are grouped so that adjacent borders with the same slope form a single place. Points where the slope changes direction are marked as joins. From each of these joins, the free space is partitioned according to the possible motions of each individual part. These free space divisions (FSD's) are shown as dashed lines in the figures. For example, in the escapement if we are in place-86 and the pallet is moving clockwise (downward in configuration space) the next contact with the scape wheel will force it to turn counter-clockwise. We want to distinguish that set of locations from place-75, where a clockwise motion of the pallets will result in a contact causing clockwise motion of the scape wheel. Between these two places, along the FSD characterized by place-45, a clockwise motion of the pallets results in the pallets jamming in the scape wheel.

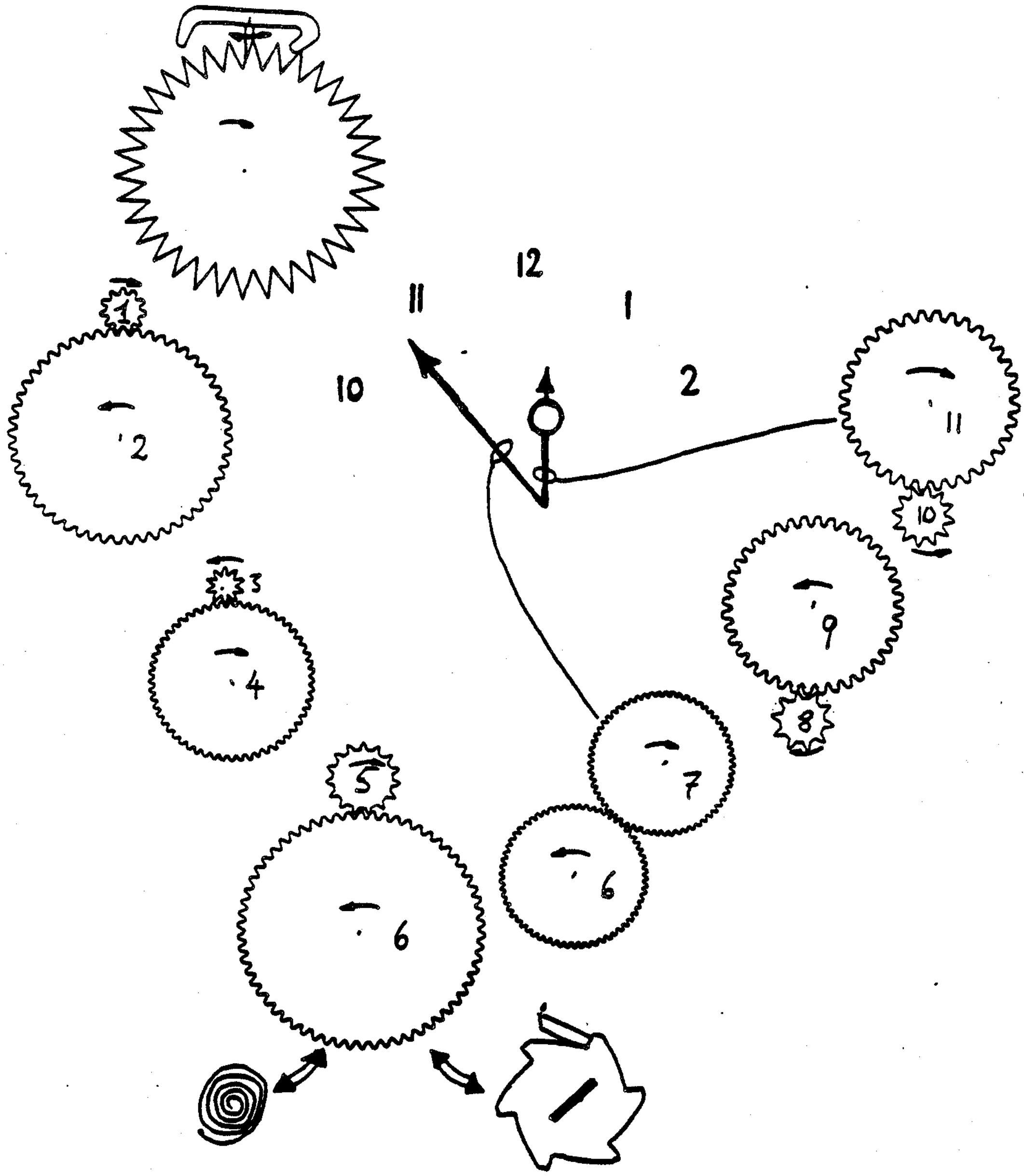


Figure 9.1: Two dimensional representation of the QRG clock

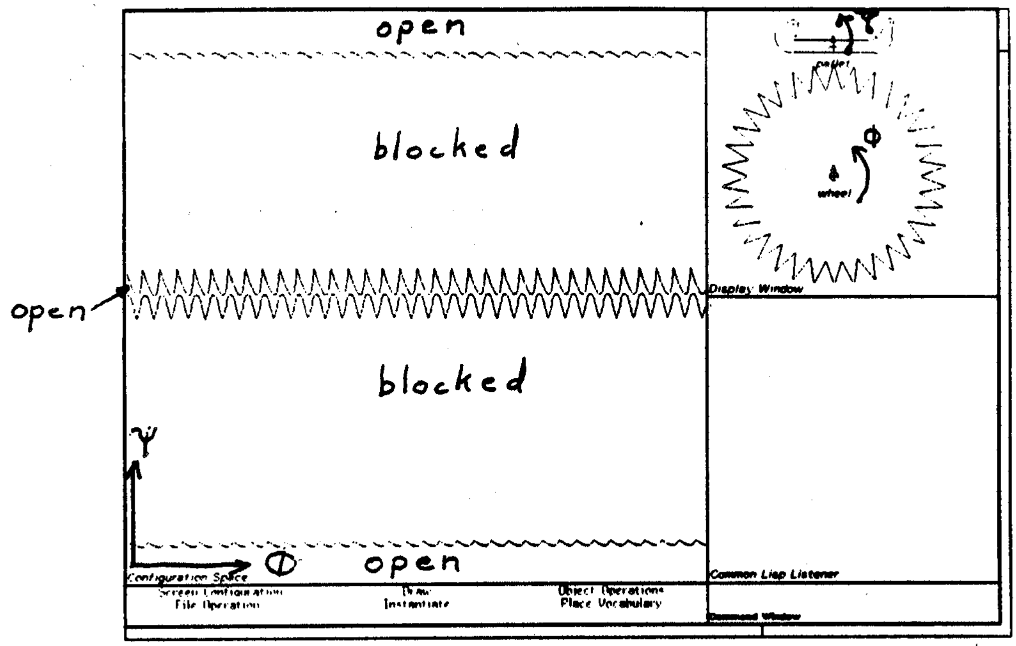


Figure 9.2: The configuration space for scape wheel and pallets

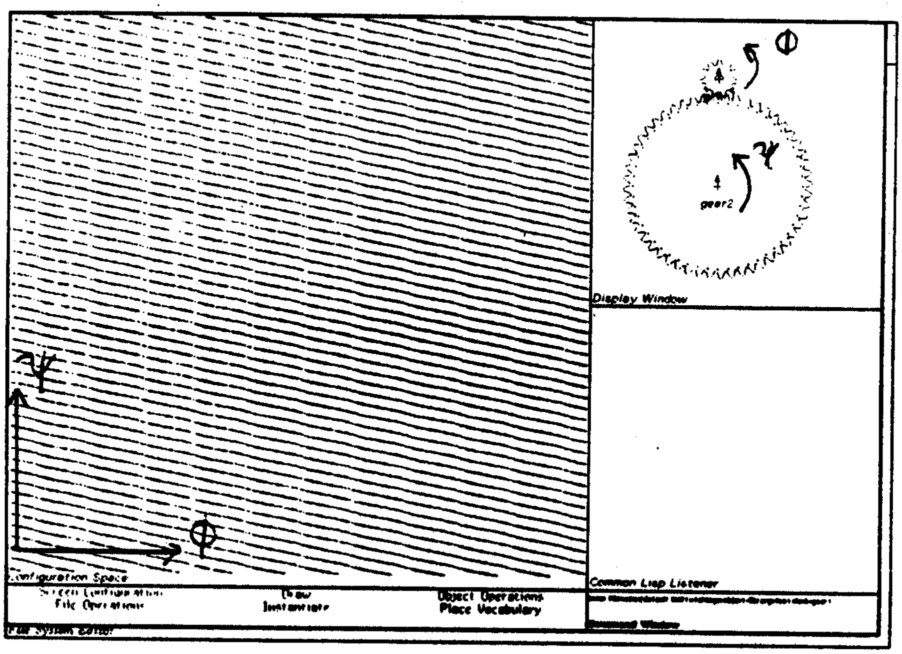


Figure 9.3: The configuration space for gears 1 and 2

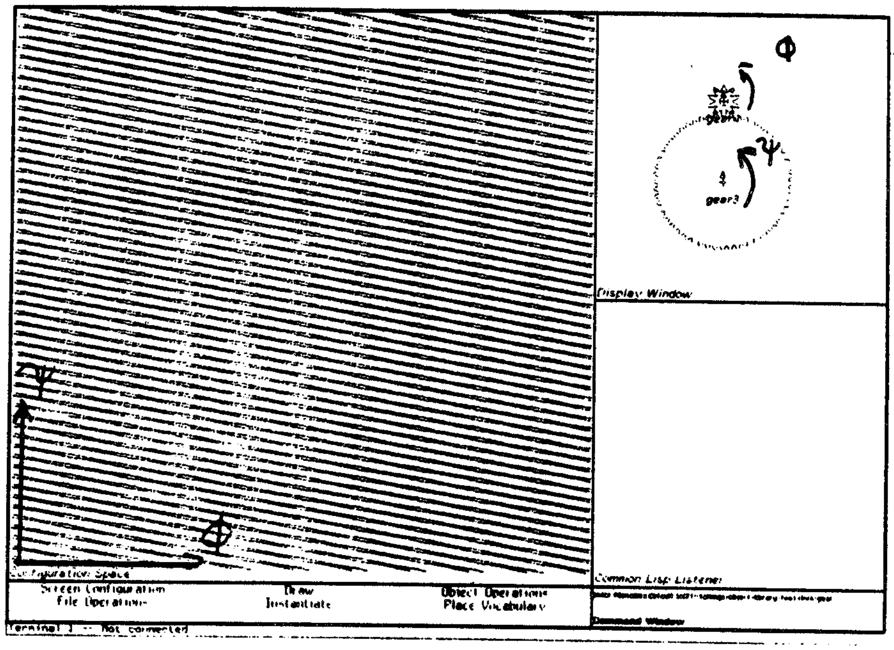


Figure 9.4: The configuration space for gears 3 and 4

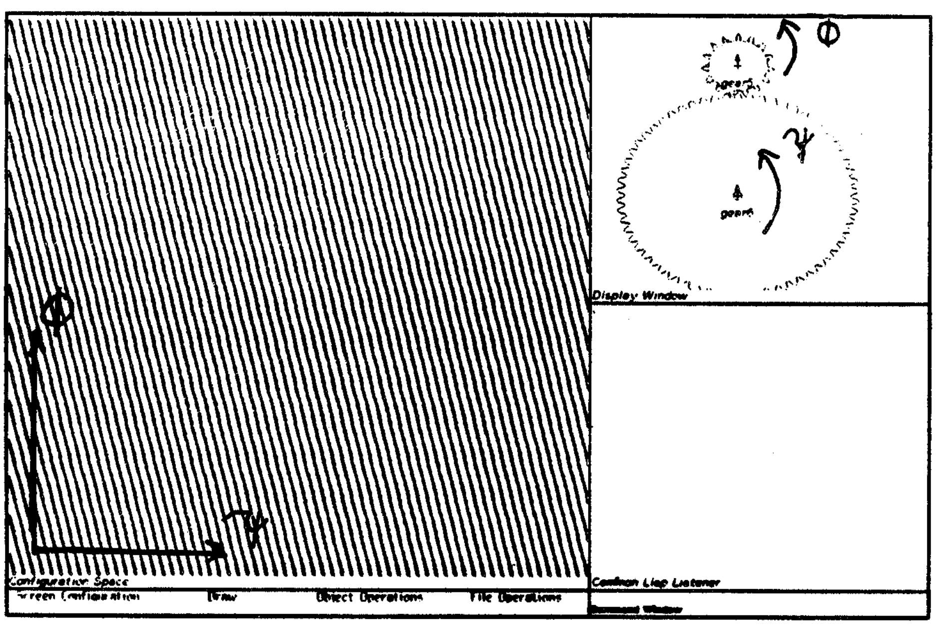


Figure 9.5: The configuration space for gears 6 and 5

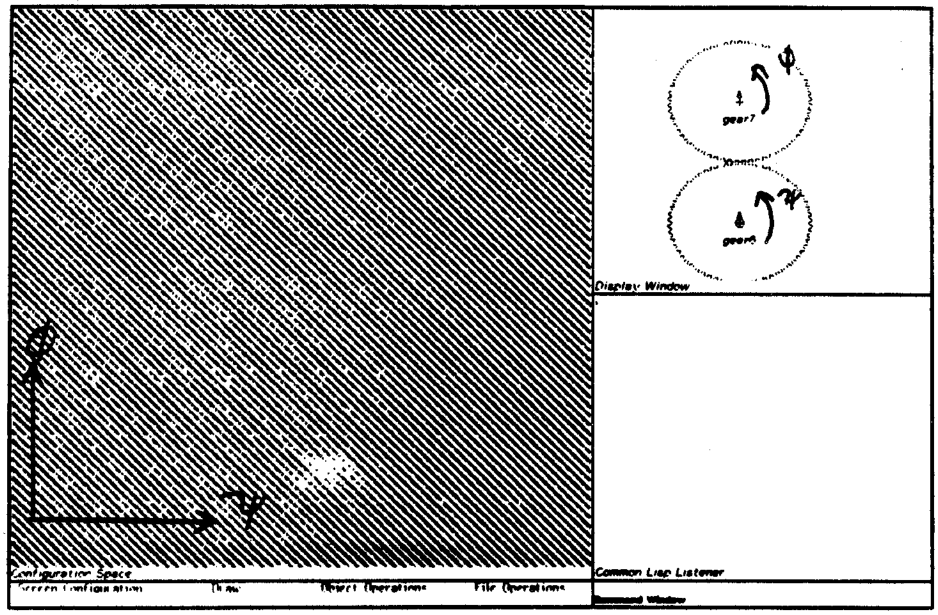


Figure 9.6: The configuration space for gears 7 and 6

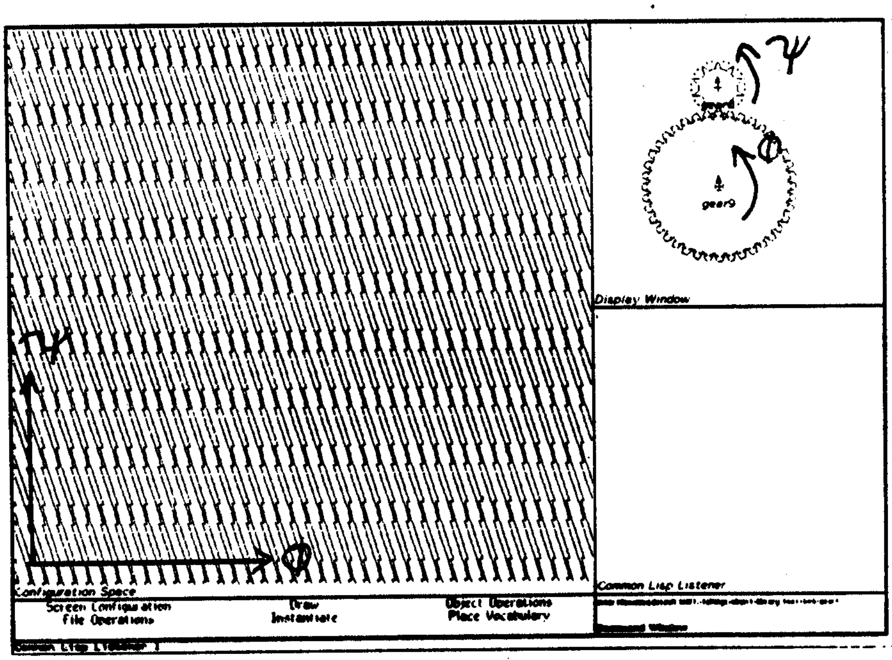


Figure 9.7: The configuration space for gears 9 and 8

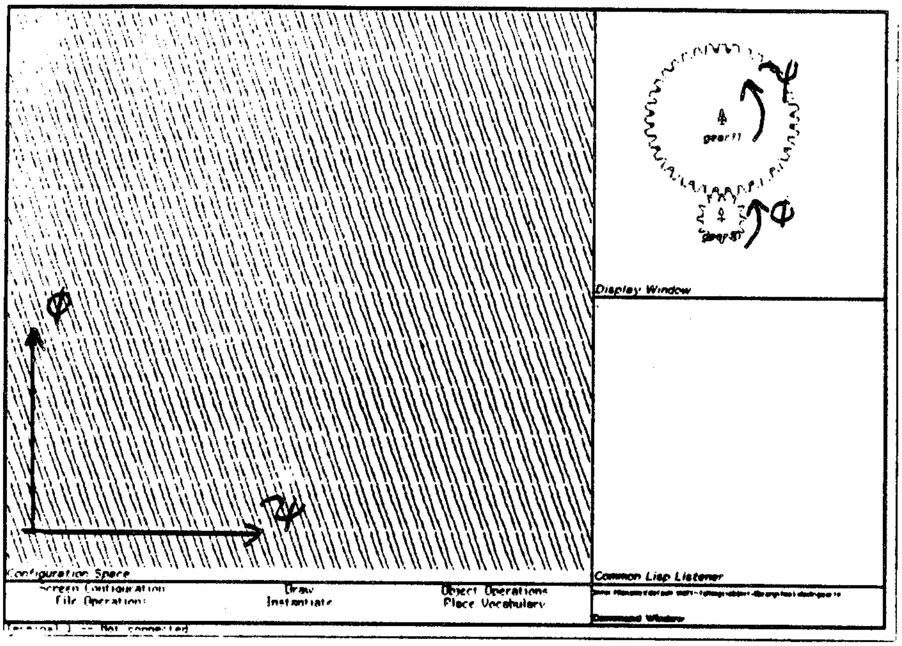


Figure 9.8: The configuration space for gears 11 and 10

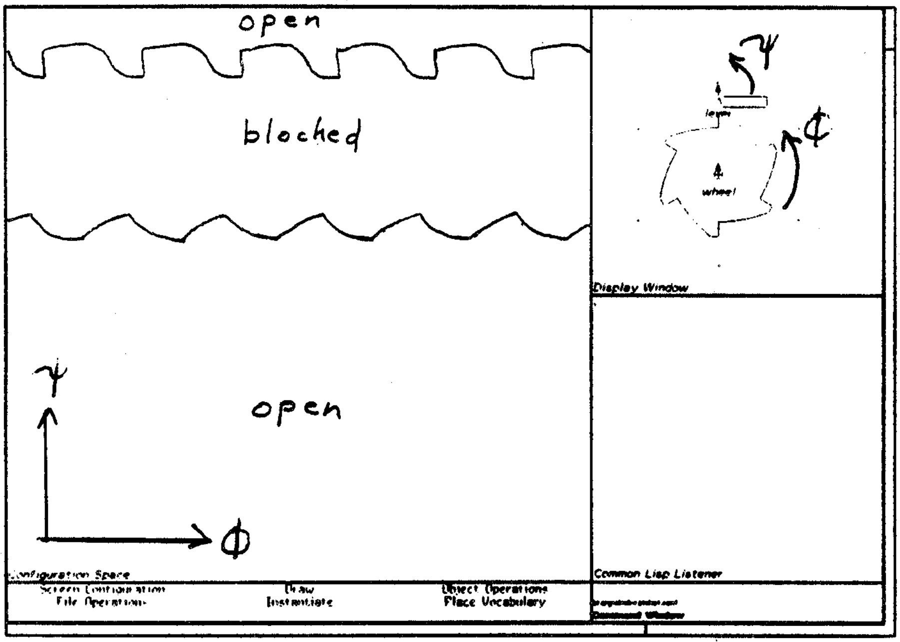


Figure 9.9: The configuration space for ratchet and pawl

Each place must be uniquely identified so that we can associate properties with it. For example, if a place arises from a CSEG we need to include information about surfaces and surface normals. A join must indicate whether the adjacent CSEG's form an concave angle. Every place must indicate the set of places adjacent to it in each direction. In this thesis, a distinct numbering serves as a symbolic name.

There is one consideration which requires entering an external place distinction. When the pallets are facing to the right, gravity will turn them clockwise, when they are facing left, counter-clockwise. Without knowing about gravity, Clock cannot make this distinction on its own. To distinguish these sets of locations, two new place divisions, pallets facing down and pallets facing up, are introduced. It turns out that a place division already exists for the pallets facing down, the free space division passing horizontally through the operating region of the clock.

9.1.3 Kinematic States

Once the place vocabularies have been computed and stored for each kinematic pair, the user indicates by means of *linkages* whether the objects in one kinematic pair are involved in others. For example, the motions of gears 9 and 10 correspond, and the motion of gear 1 corresponds to that of the scape wheel. In the clock, these objects are directly coupled.

This information allows us to determine kinematic chains. The kinematic chain is the transitive closure of any object and the kinematic pairs or linkages involving this object. At this point Clock notices that there are two kinematic chains in this clock, one consists of the escapement and gears while the other involves the ratchet. Because there is neither a rigid connection between the ratchet and the rest of the mechanism nor a linkage between any part of the ratchet and any other part of the clock, the ratchet may be analyzed separately from the rest of the clock. For now we will leave off discussion of the ratchet and come back to it in the next section.

The place vector is formed by taking all combinations of a single place from each place vocabulary in a kinematic chain. Each of the gears in this kinematic chain only has a single place, and there are 58 places covering the escapement. Thus there will be 58 place vectors.

Place vectors provide the kinematic state of the mechanism. The full set of kinematic states is too long to list and largely repetitious, but a sample state is as follows:

```
Location(Gear11,Gear10) = <Gear10-Place-16>
Location(Gear9,Gear8) = <Gear8-Place-31>
Location(Gear7,Gear6) = <Gear6-Place-13>
Location(Gear6,Gear5) = <Gear5-Place-12>
Location(Gear3,Gear4) = <Gear3-Place-19>
Location(Gear1,Gear2) = <Gear1-Place-13>
Location(Scape-Wheel,Pallets) = <Esc-Place-24>
```

In this state the escapement is in one of the locations described by place-24, and each of the gear pairs are in their unique places. Place numbers are generated automatically by the program. Notice that even though there is only one place for each of the gear pairs, the abstraction of the place vocabularies generated several places before combining them.

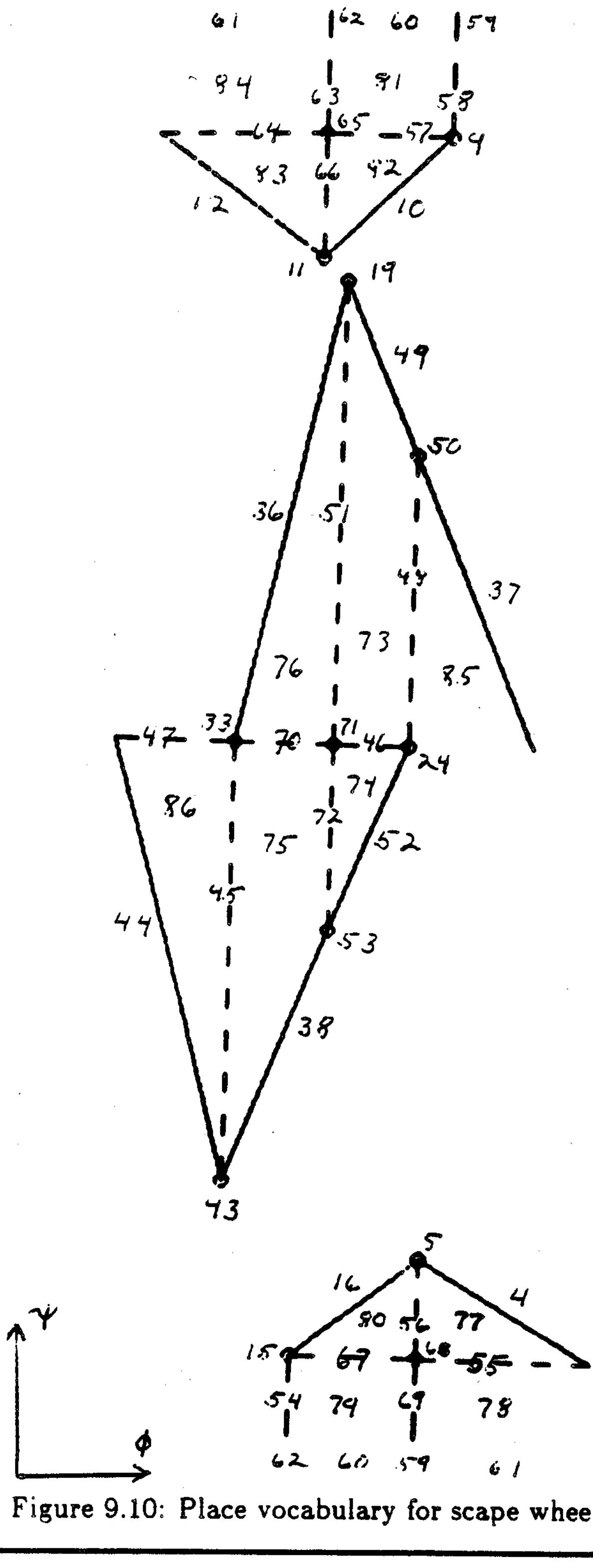


Figure 9.10: Place vocabulary for scape wheel and pallets

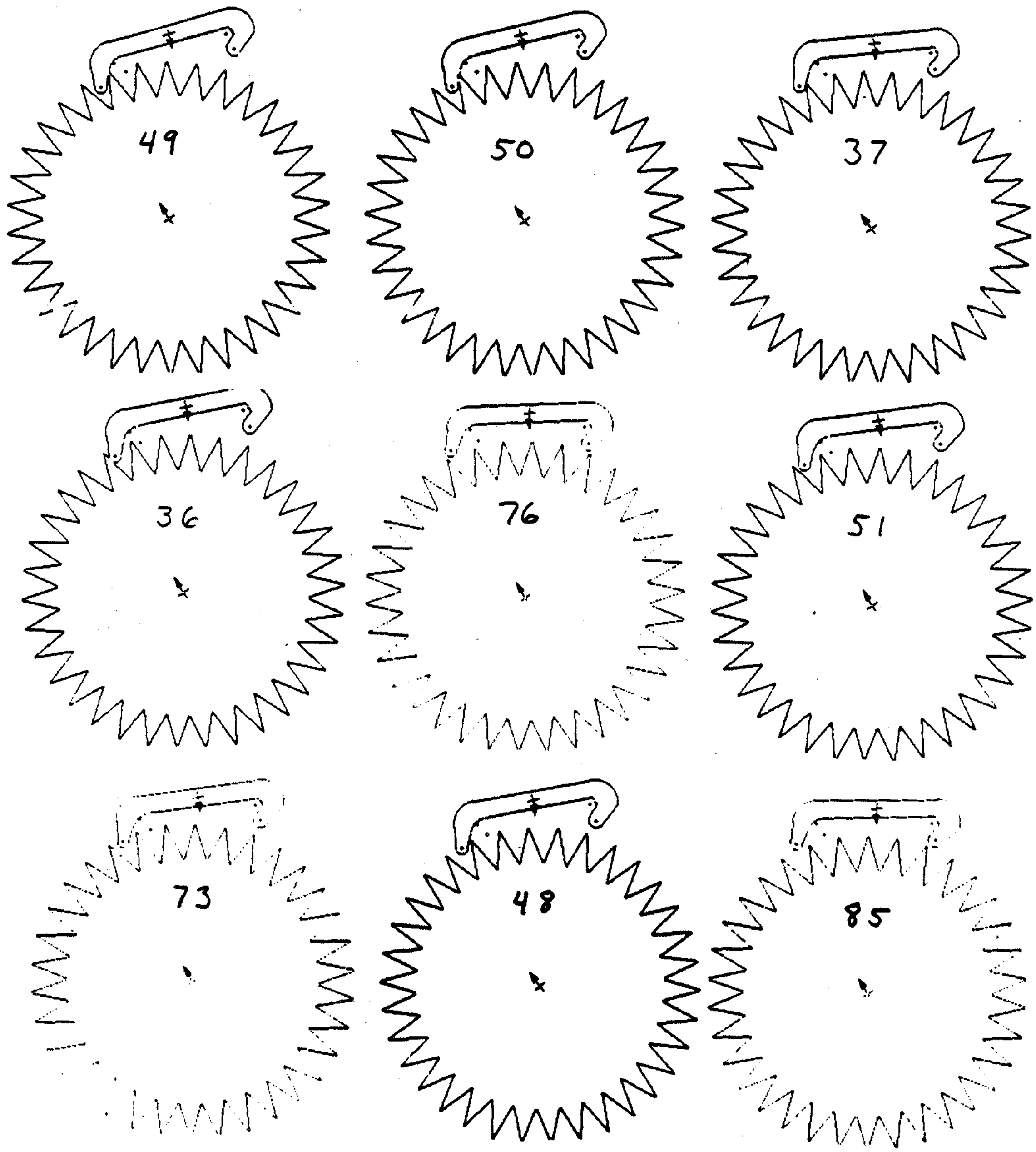


Figure 9.11: Sample configurations for the escapement's places

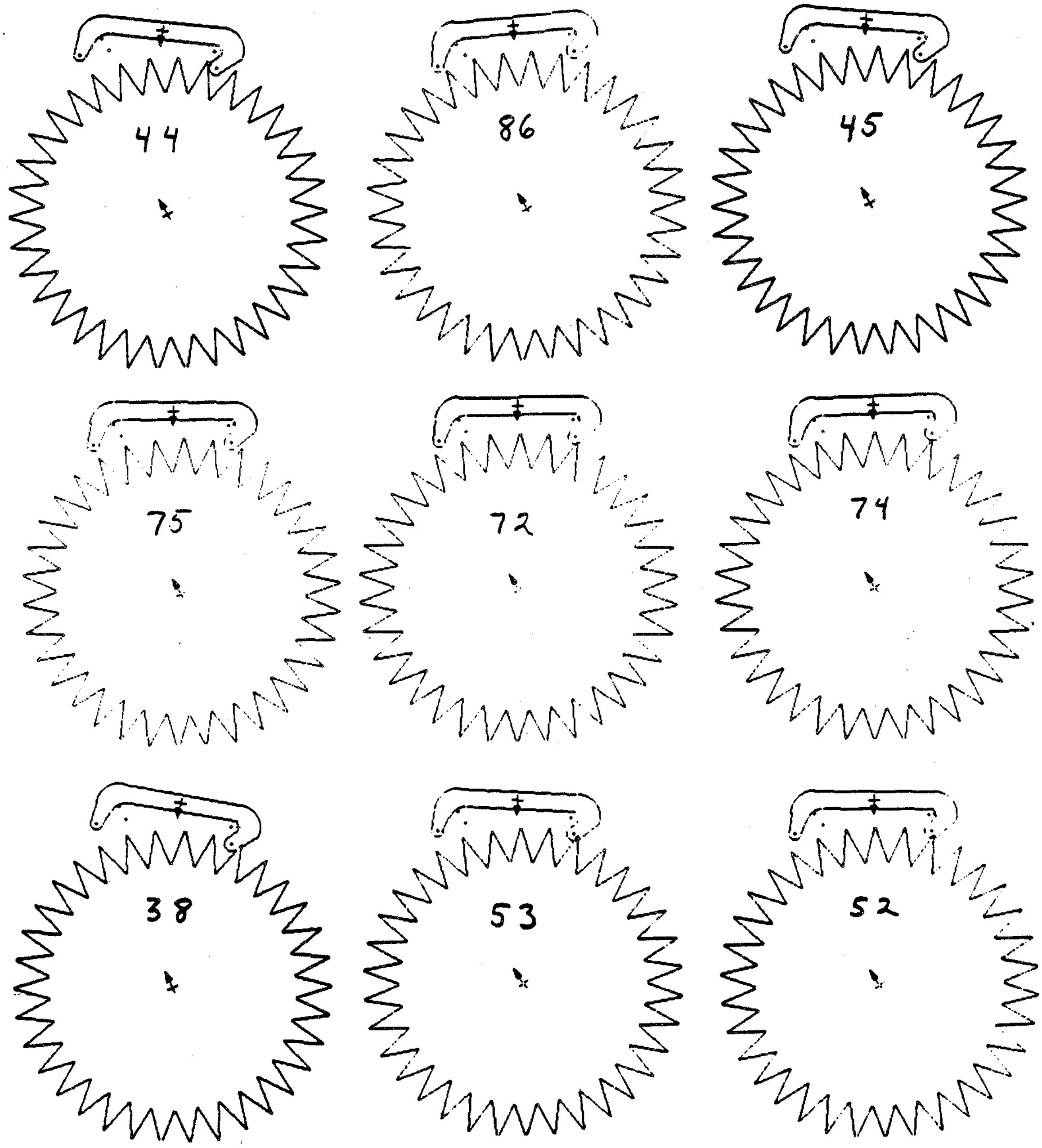


Figure 9.12: Sample configurations for the escapement's places

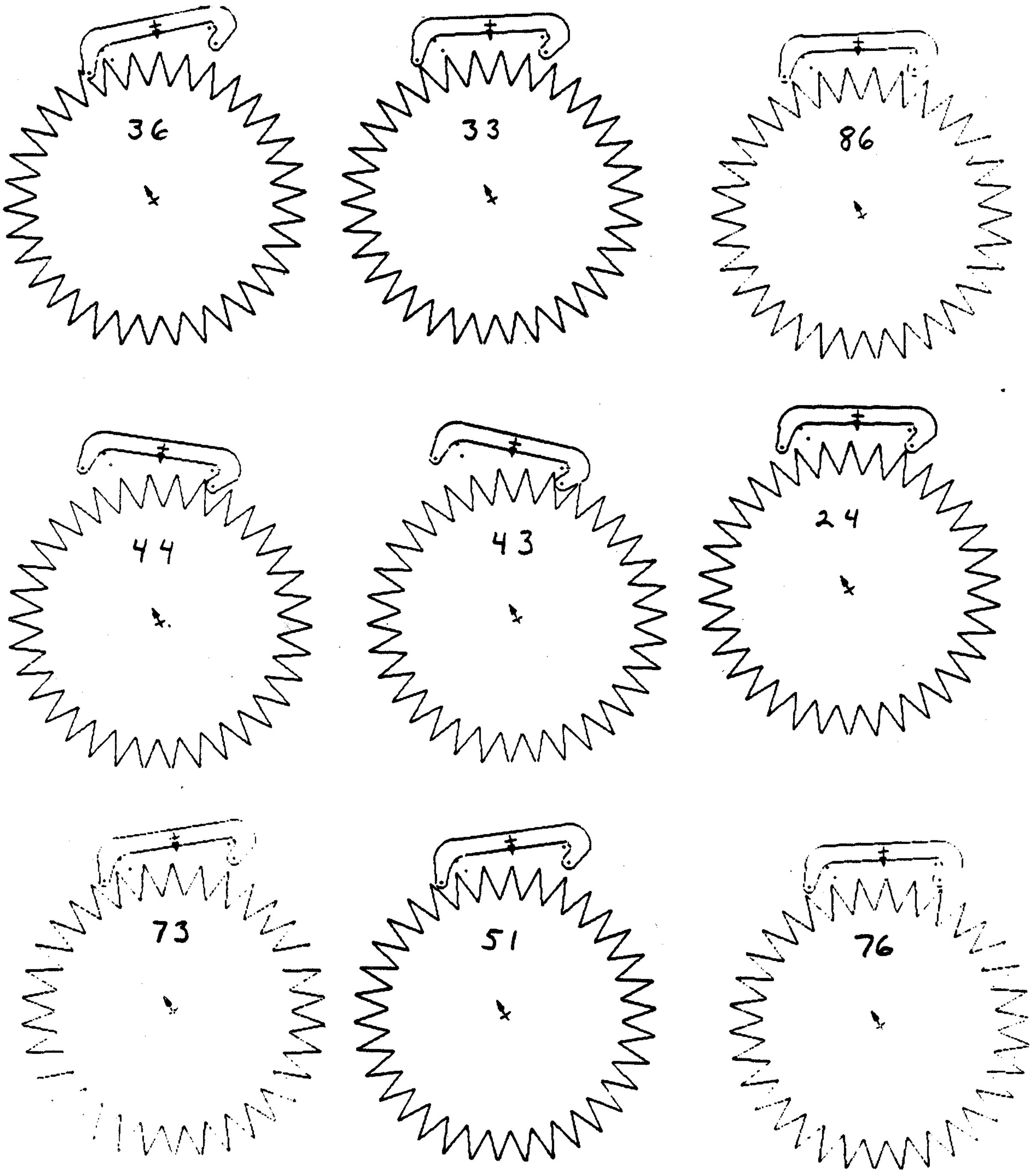


Figure 9.13: Sample configurations for the escapement's places

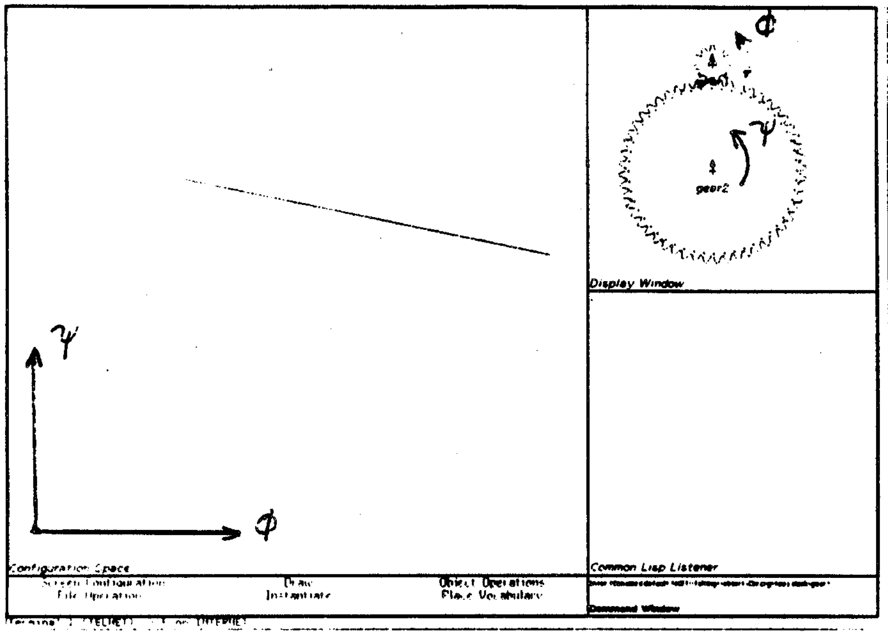


Figure 9.14: Place vocabulary for gears 1 and 2

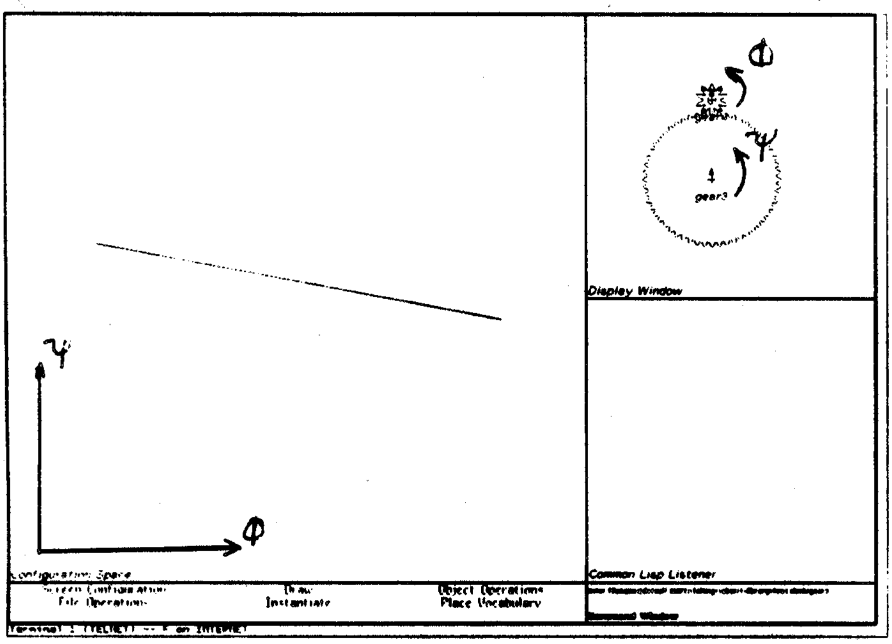


Figure 9.15: Place vocabulary for gears 3 and 4

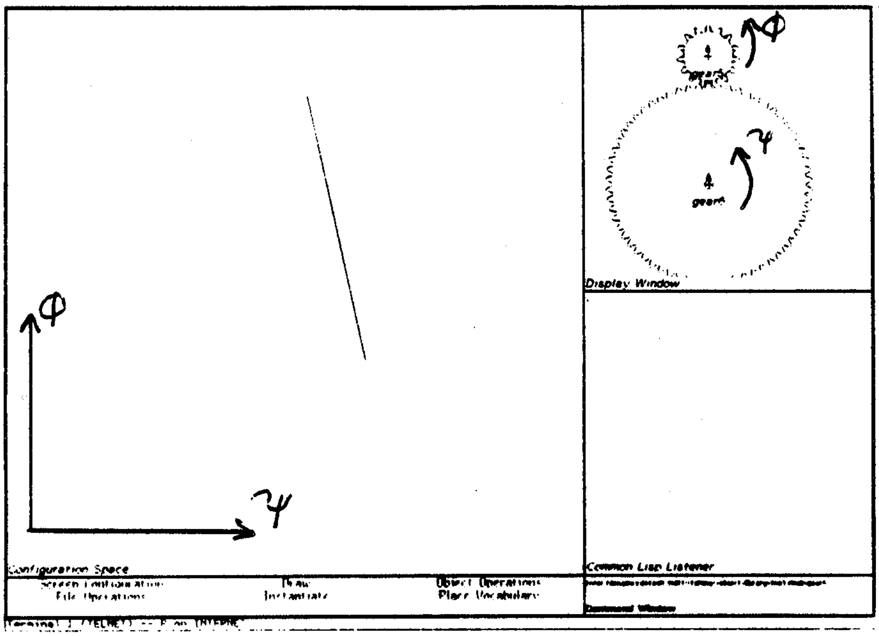


Figure 9.16: Place vocabulary for gears 6 and 5

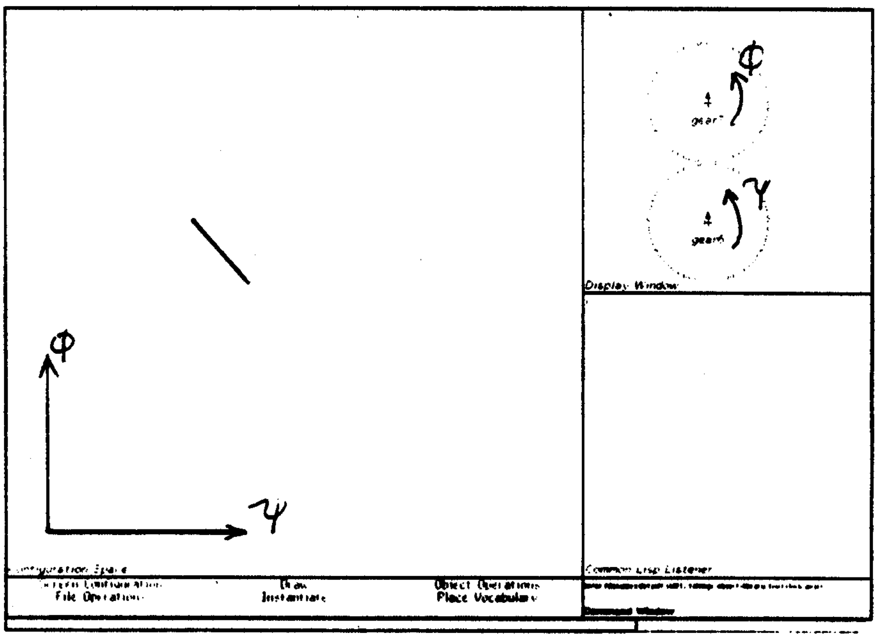


Figure 9.17: Place vocabulary for gears 7 and 6

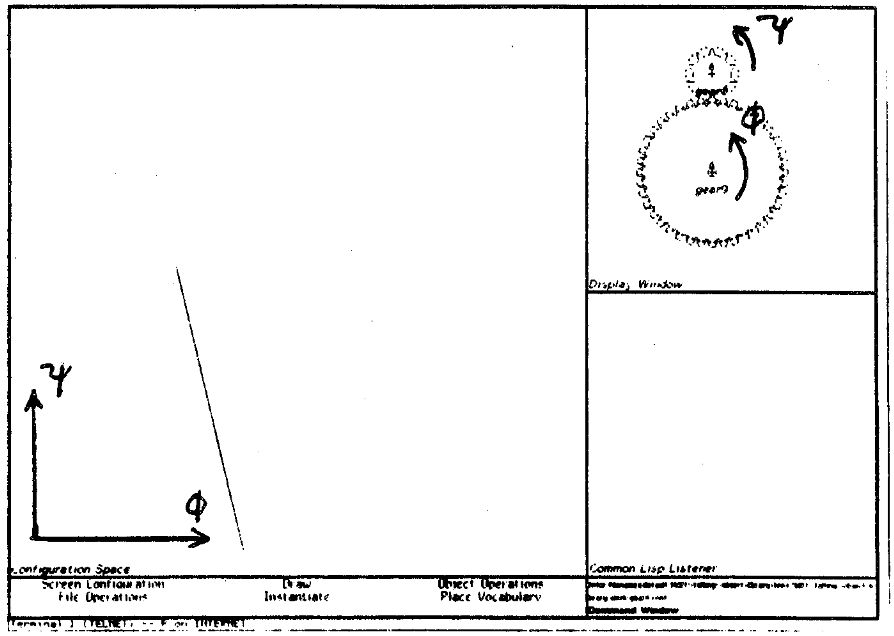


Figure 9.18: Place vocabulary for gears 9 and 8

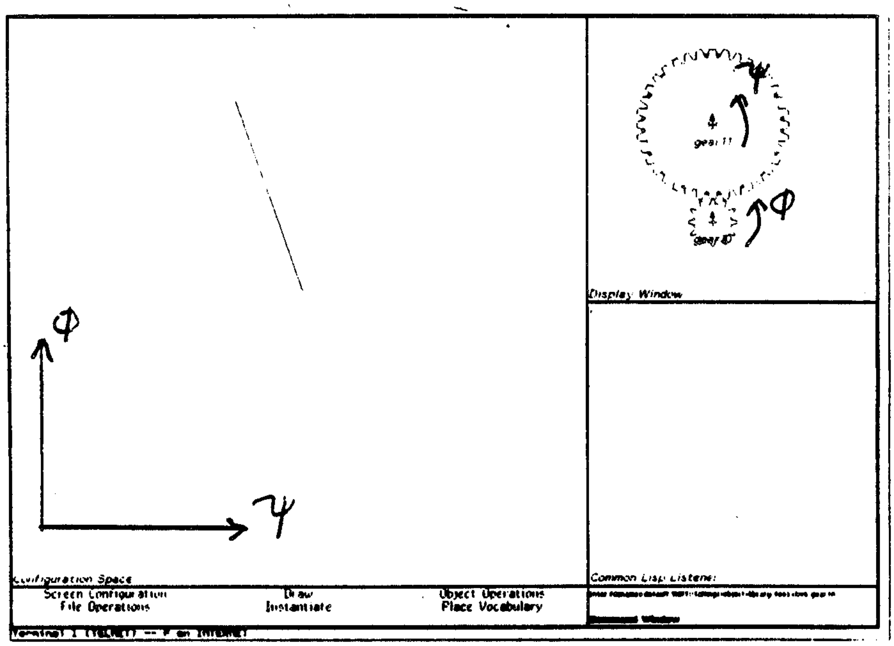


Figure 9.19: Place vocabulary for gears 11 and 10

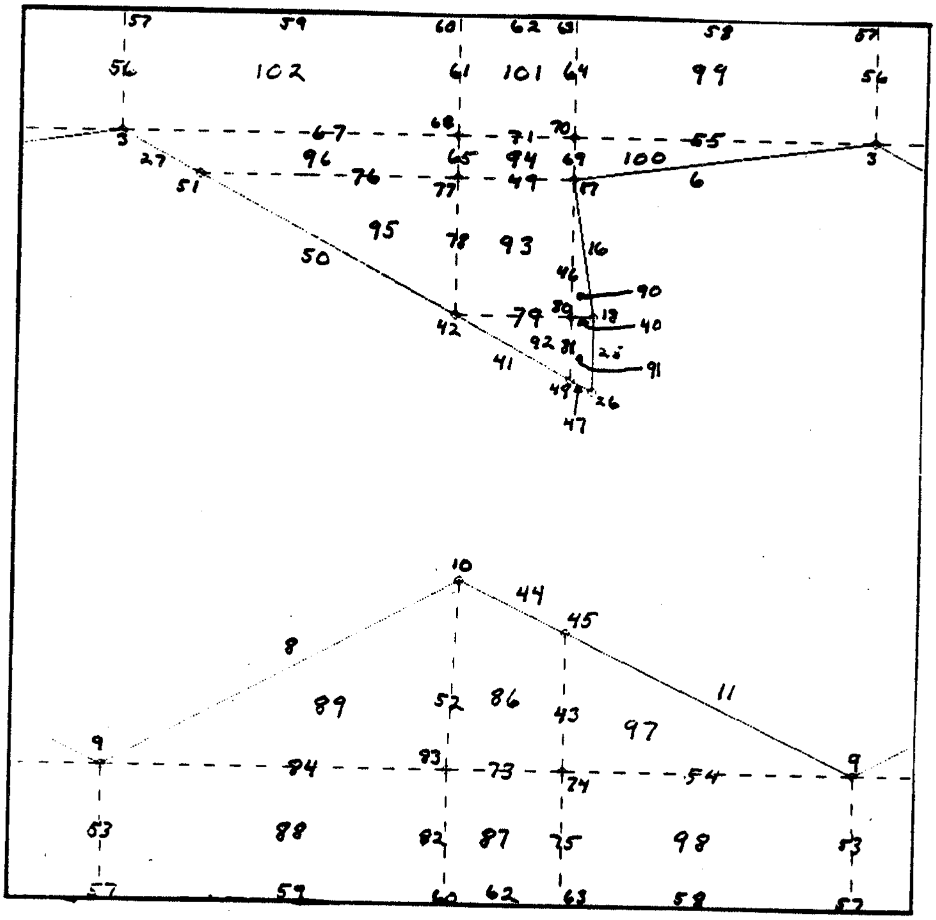


Figure 9.20: Place vocabulary for ratchet and pawl

9.1.4 Mechanical States

Each possible motion of each object is added to the kinematic state to form the full mechanical state. This new information if generated separately would be the dynamic state; however, when objects are in contact many possible motions will be ruled out, and linkages enforce the choices of motion for a second object once the first is created. Thus it is important to do kinematic analysis first to minimize the number of intermediate interpretations created. There are 462 consistent mechanical states for this clock, of which one is reproduced below:

```
Location(Gear11,Gear10) = <Gear10-Place-16>
Motion(Gear10) = +
Motion(Gear11) = -
Location(Gear9,Gear8) = <Gear8-Place-31>
Motion(Gear8) = -
Motion(Gear9) = +
Location(Gear7,Gear6) = <Gear6-Place-13>
Motion(Gear7) = -
Location(Gear6,Gear5) = <Gear5-Place-12>
Motion(Gear5) = -
Motion(Gear6) = +
Location(Gear3,Gear4) = <Gear3-Place-19>
Motion(Gear4) = -
Motion(Gear3) = +
Location(Gear1,Gear2) = <Gear1-Place-13>
Motion(Gear2) = +
Motion(Gear1) = -
Location(Scape-Wheel,Pallets) = <Esc-Place-24>
Motion(Pallets) = +
Motion(Scape-Wheel) = -
```

9.1.5 Mechanical Transitions

Having computed all possible qualitative states, we now consider transitions between states. These transitions may be caused either by forces causing change of motion (dynamic transition) or motions causing change of location (kinematic transition). If a collision occurs as a result of changing location, the kinematic transition may cause motions as well as location to change.

Appendix A gives the complete transition table for the mechanical clock. Since the gear pairs do not change places, locations are numbered according to the current place of the escapement. Further, since the motion of the scape wheel uniquely determines the motion of each gear, only the motion of the scape wheel and the pallets is given. This representation was selected to conserve space in this thesis; `Clock` maintains all the place and motion information as shown in the representative mechanical state given above.

The set of all possible mechanical states and transitions between them is shown graphically in figures 9.21 through 9.24. These are grouped according to behavioral cycles. Figure 9.21 shows the envisionment of the normal operating region of the clock. Figure 9.22 shows the region where the pallets are inverted but still interacting with the scape wheel. Figure 9.23 shows an interesting behavior, in that the pallets are in their downward positions, but the gears have just enough clearance to turn free of them. Finally 9.24 shows the behavior when the pallets are inverted and the pendulum is balancing in the upright position. Again the gears are turning freely.

Notice in the transition table (Appendix A) that some motions do not produce a transition from certain states. For example, place-4 with both the scape wheel and the pallets moving counter-clockwise (positive). These are transitions which are ruled out by constraint considerations. Because place-4 corresponds to a CSEG with surface normal in the (–,–) direction, it is impossible for the objects to be moving in any of the (+,0), (+,+), or (0,+) directions. while in that place. They must follow the surface of the CSEG. That is, positive motion of both the scape wheel and the pallets in place-4 would result in a configuration where the two solids would be inside each other.

Notice also that the next state is not necessarily uniquely determined. The most ambiguous transition is from place-73 with both the scape wheel and the pallets moving counter-clockwise. In this state, if the current motion persists, the next location will be any of place-48, place-49, or place-50. (See Figure 9.10). In place-48 the current motion persists. Place-49 and place-50 represent contact between the two objects, and these transitions represent collisions. Before collision the motion is at an angle. The surface normal of the contact is at an angle opposite to the direction of motion. Depending on more exact information about the slopes, the objects will either move up, down, or stop on the surface. That kinematic ambiguity accounts for seven of the next possible states.

The second source of ambiguity in this state is due to dynamic considerations. Gravity is pushing the pallets clockwise, and the spring is pushing the scape wheel clockwise. Since the objects may be in place-73 for some period of time, depending on the strength of the force, either of these objects (or both) could stop moving before they make any of the contacts mentioned above.

One of the ways this envisionment could be used in to determine whether a behavior can occur. In this case we want to know if a path through the envisionment exists which exhibits the clock behavior. Using de Carle's (de Carle, 1975) description of clock behavior (Section 1.1 and Figure 1.3) as the basis for clock behavior, search begins in place-36 with both the scape wheel and the pallets moving clockwise (36, –, –): *As the pendulum swings to the left it allows the tooth A of the escape wheel to slide along the impulse face B of the pallet pad.* From this state we go to state (33, –, –) then (86,–, –): *Eventually the tooth A drops off the pallet pad.* From (86,– –) the next state is highly ambiguous, but the explanation helps guide the search: *The tooth C drops on to the pallet pad D and as the pendulum continues to swing to the left this locking becomes deeper and by reason of the curve of the pad the escape wheel is made to recoil.* The next state in which the scape wheel is recoiling is (44, +, –). After this state de Carle describes two behaviors: *When the pendulum has reached the end of its journey and starts to return the escape tooth C will then give impulse to the pallet and so to the pendulum.* There are transitions to states (44, 0, 0) or (43, 0, 0) and then (43, –, +) which leads to (44, –, +). Then the pendulum reached the end of its journey it stops and begins moving in the opposite (plus) direction. Finally: *This cycle is repeated on the pallet pad B.* Briefly, the transitions are from (44, –, +) to (24, –, +) to (73, –, +) to (51, –, +) to (76, –, +). From this state there is another collision to (36, +, +). There are transitions to states (36, 0, 0) or (19, 0, 0) and then (19, –, –) which leads to (36, –, –), the starting state.

Another use of this analysis is detection of faulty, erroneous, or unexpected behaviors. One way to do this would be to search the envisionment and determine if there are histories

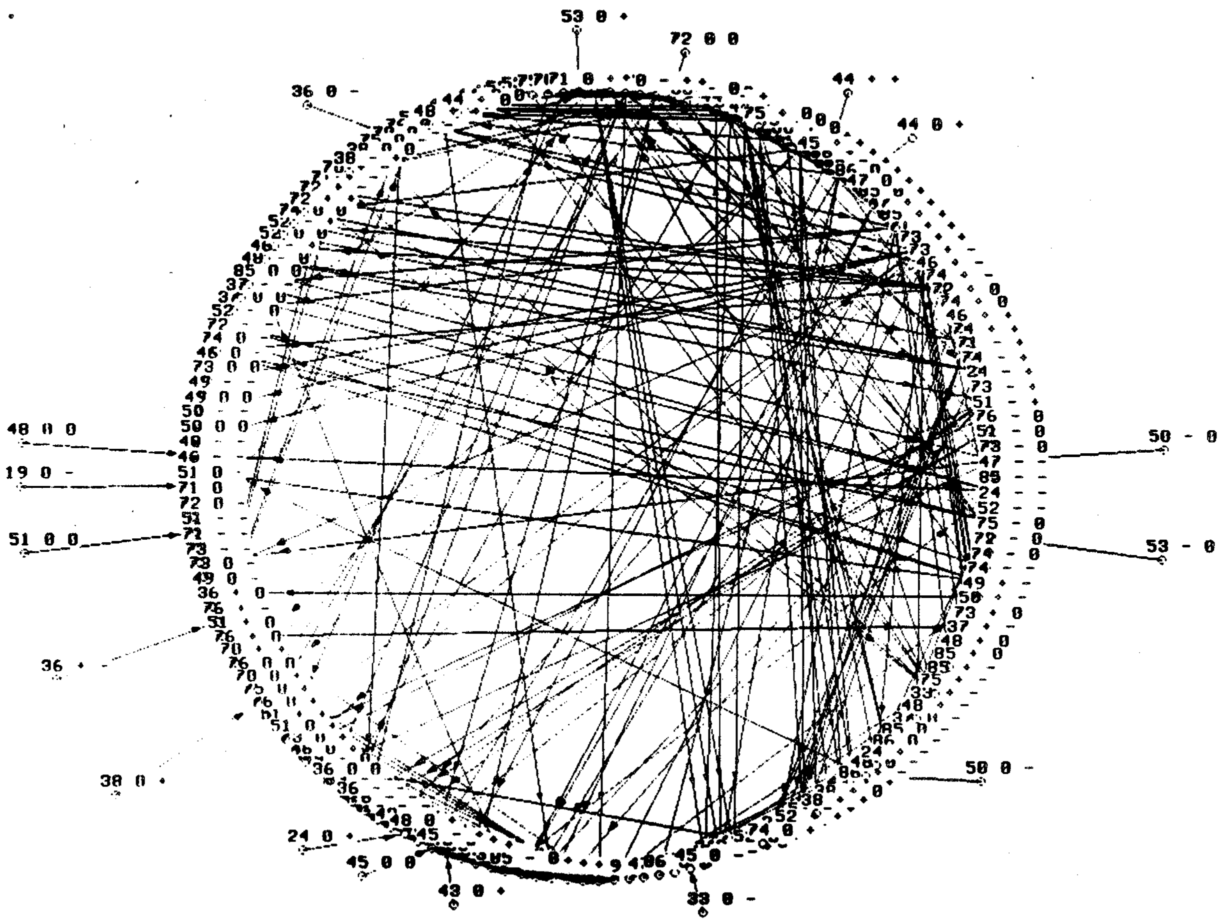


Figure 9.21: Envisionment of the QRG clock: Operating region

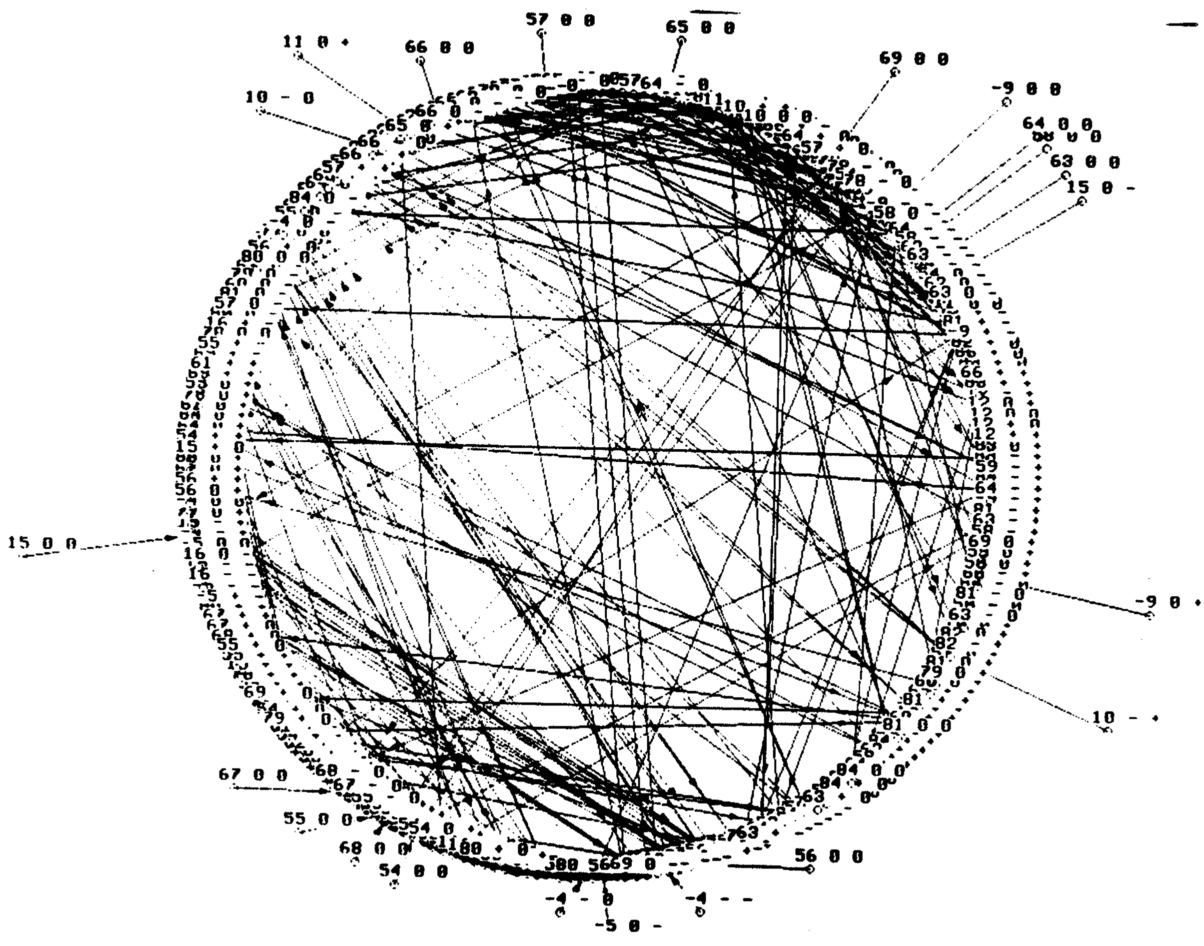


Figure 9.22: Envisionment of the QRG clock: Pallets inverted

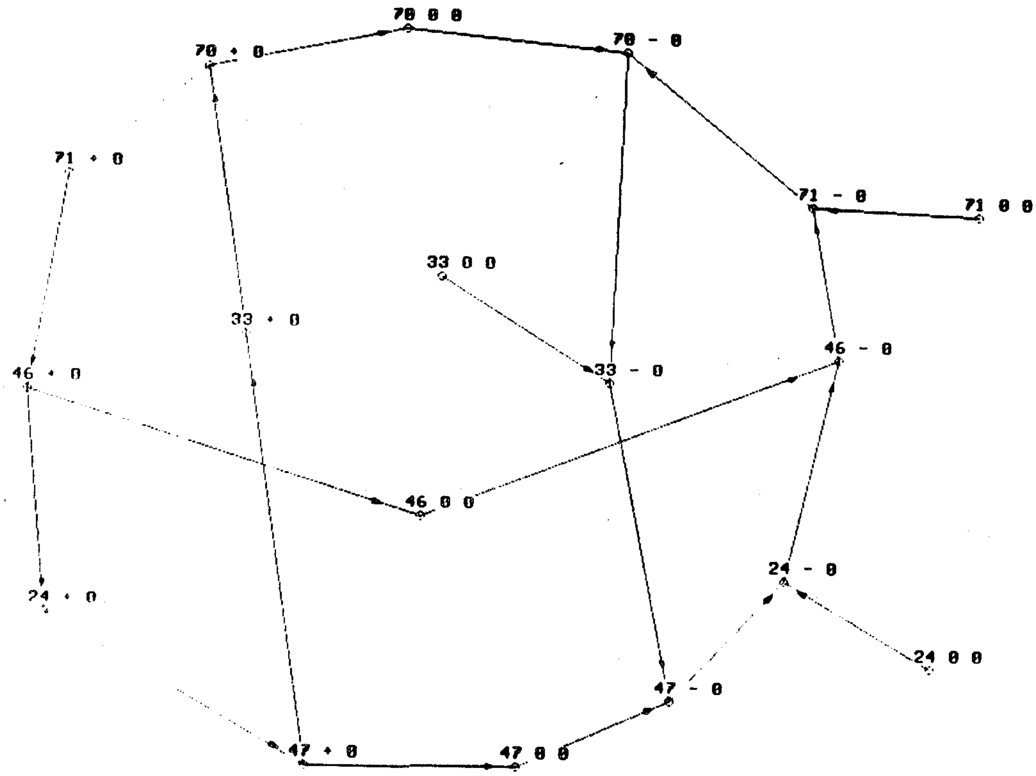


Figure 9.23: Envisionment of the QRG clock: Erroneous cycle

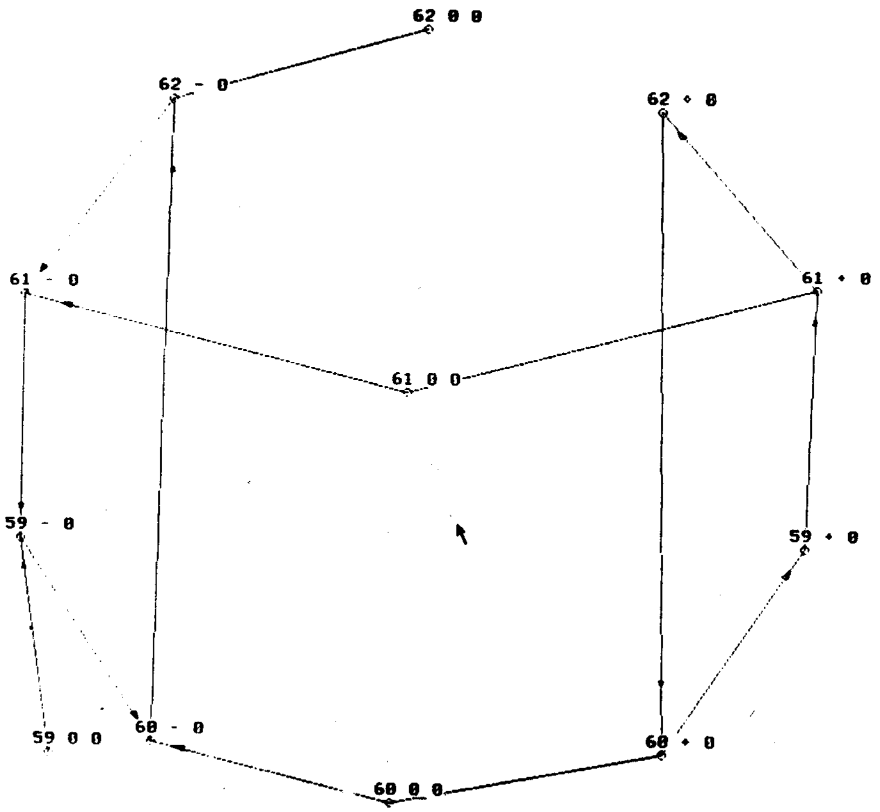


Figure 9.24: Envisionment of the QRG clock: Pallets inverted, gears spinning

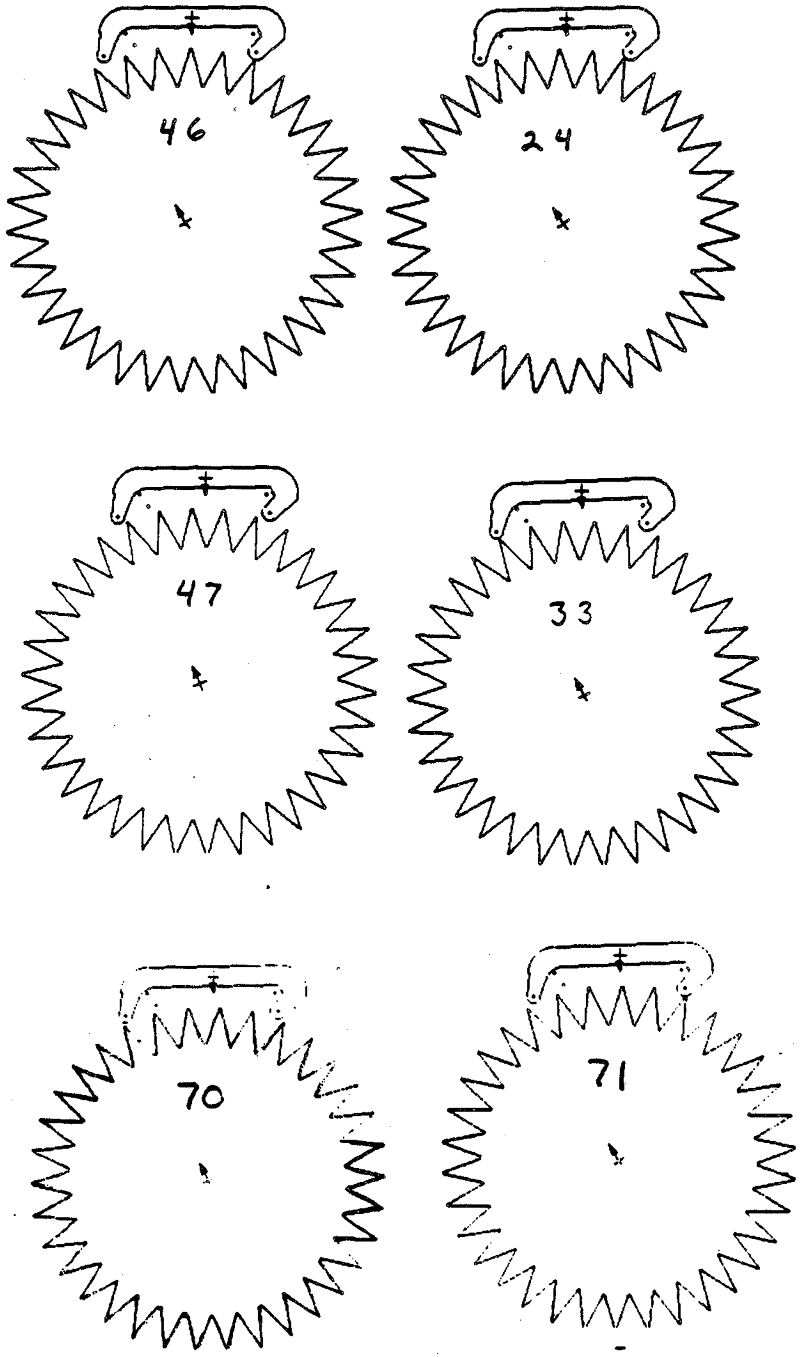


Figure 9.25: Clock history

which do not contain any of the states mentioned by de Carle's description of clock behavior. It happens that in the representation of this escapement there is a sequence of places where the gears will spin without actually contacting the scape wheel, clearly not a desirable clock behavior. This is shown in Figure 9.23. In these states, if the scape wheel starts with some initial momentum it may cycle through (46, +, 0), (24, +, 0),(47, +, 0), (33, +, 0), (70, +, 0), and (71, +, 0). After some period of time it will stop moving and continue cycling clockwise through (71, –, 0), (70, –, 0), (33, –, 0), (47, –, 0), (24, –, 0), (46, –, 0) until the spring winds down. Note that `Clock` cannot detect the fact that the spring will wind down. It only generates the behavioral sequences.

9.2 Ratchet

Though the ratchet is part of the mechanical clock, there is no rigid connection between it and the rest of the components of the clock. When `Clock` determines kinematic chains, a separate chain will be produced for the ratchet. This allows separate analysis of the ratchet and the rest of the clock.

We have already seen the configuration space and the place vocabulary for the ratchet in Figures 9.9 and 9.20. Figure 9.26 shows configurations of the ratchet in various places.

Not all places introduced to satisfy external requirements are legal. In the case of the ratchet, places where the pawl is pointing to the left cause a counter-clockwise force and places where the pawl is pointing right cause a clockwise force. When the pawl is straight up or straight down there will be no external force on it. In making external place distinctions the user will typically mark the place where the pawl is straight down as significant; however, the pawl can never be pointed straight down because that configuration corresponds to blocked space. The place vocabulary generator must be able to detect this case and ignore that division. To do this, except in degenerate cases, any division which does not cross an FSD or a CSEG is in blocked space and may be safely ignored. Degenerate cases occur when the entire configuration space is free space or when CSEG's which span the length of the configuration space are parallel to each other as well as an axis. These two cases must be detected in advance and handled specially.

9.2.1 Kinematic States

Analysis of the ratchet continues with the generation of the place vectors. Since there is only one place vocabulary in this kinematic chain, the place vector is the same as the place vocabulary. There are 72 places in the ratchet's place vocabulary which makes it more complex than the clock escapement with 58 places. Again the number of kinematic states makes they too expansive to list, but a sample state is as follows:

```
Location(Ratchet,Pawl) = <Rat-Place-26>
```

9.2.2 Mechanical States

Combining all possible motions of each object in each place yields the mechanical states of the ratchet. There are 588 consistent mechanical states. For example, the following are the only consistent mechanical states involving place-26:

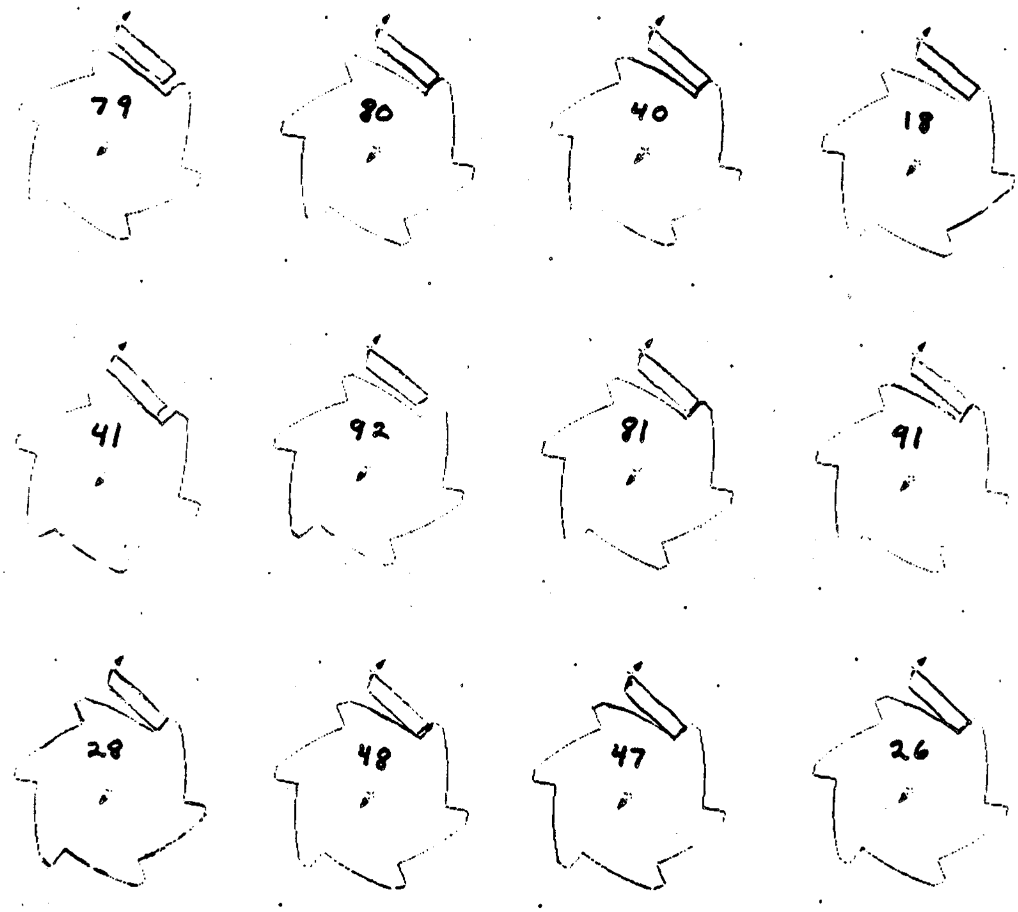


Figure 9.26: Sample configurations for the ratchet's places

```
Location(Ratchet,Pawl) = <Rat-Place-26>
Motion(Ratchet) = 0
Motion(Pawl) = 0

Location(Ratchet,Pawl) = <Rat-Place-26>
Motion(Ratchet) = 0
Motion(Pawl) = +

Location(Ratchet,Pawl) = <Rat-Place-26>
Motion(Ratchet) = -
Motion(Pawl) = +
```

9.2.3 Mechanical Transitions

We have focused on place-26 because it is the desired location for this device. The pawl is in contact with two surfaces on the ratchet wheel, one of which prevents the pawl from moving downward and the other prevents the ratchet from turning counter-clockwise. When the clock is operating the force of the spring will try to move the ratchet wheel counter-clockwise and gravity will pull the pawl down, leading to this jammed state. However, when the clock is being wound the ratchet wheel is still free to move clockwise.

Appendix B gives the complete transition table for the ratchet when the clock is operating and the net force on the ratchet wheel is due to the spring. Starting in place-102, with no motion on either object, one possible path through the envisionment is (102,0,0) → (102,+,–) → (67,+,–) → (96,+,–) → (76,+,–) → (95,+,–) → (78,+,–) → (93,+,–) → (79,+,–) → (92,+,–) → (81,+,–) → (91,+,–) → (26,0,0) from which there are no transitions: the mechanism stops. Figure 9.27 shows this behavioral sequence.

A second path of interest is the behavior which might be described as the pawl skipping across the top of the ratchet wheel. It frequently occurs when the wheel is moving very rapidly compared to the speed of the pawl. From the same starting state, one path characterizing this behavior is (102,0,0) → (102,+,–) → (61,+,–) → (101,+,–) → (71,+,–) → (94,+,–) → (69,+,–) → (100,+,–) → (6,+,+) → (3,+,+) → (102,+,+) → (102,+,0) → (102,+,–) from which the ratchet may continue to cycle. Figure 9.28 shows this behavioral sequence. To correct this the pawl should be started from some place "below" place-17. From there all transitions lead to eventual stoppage.

9.3 Scotch Yoke

A Scotch Yoke is used for converting purely rotational motion into purely translational motion. It consists of a cam which rotates about **O** and a yoke which is only free to move translationally along the **X** axis (Figure 9.29). The configuration space representation of this is shown in Figure 9.30. We partition the configuration space representation to create a place vocabulary (Figure 9.31) and compute the possible place transitions (Appendix C).

In this case since there is only one kinematic pair, it forms the kinematic chain. The number of kinematic states are the number of places in this place vocabulary. The possible motions of each component are combined with each kinematic state to yield the mechanical state, by assuming all unconstrained directions are possible motions.

Now we consider the influence of external forces on this system. If the cam is turned counter-clockwise one behavior of this mechanism is given in Figure 9.32. The cam continues

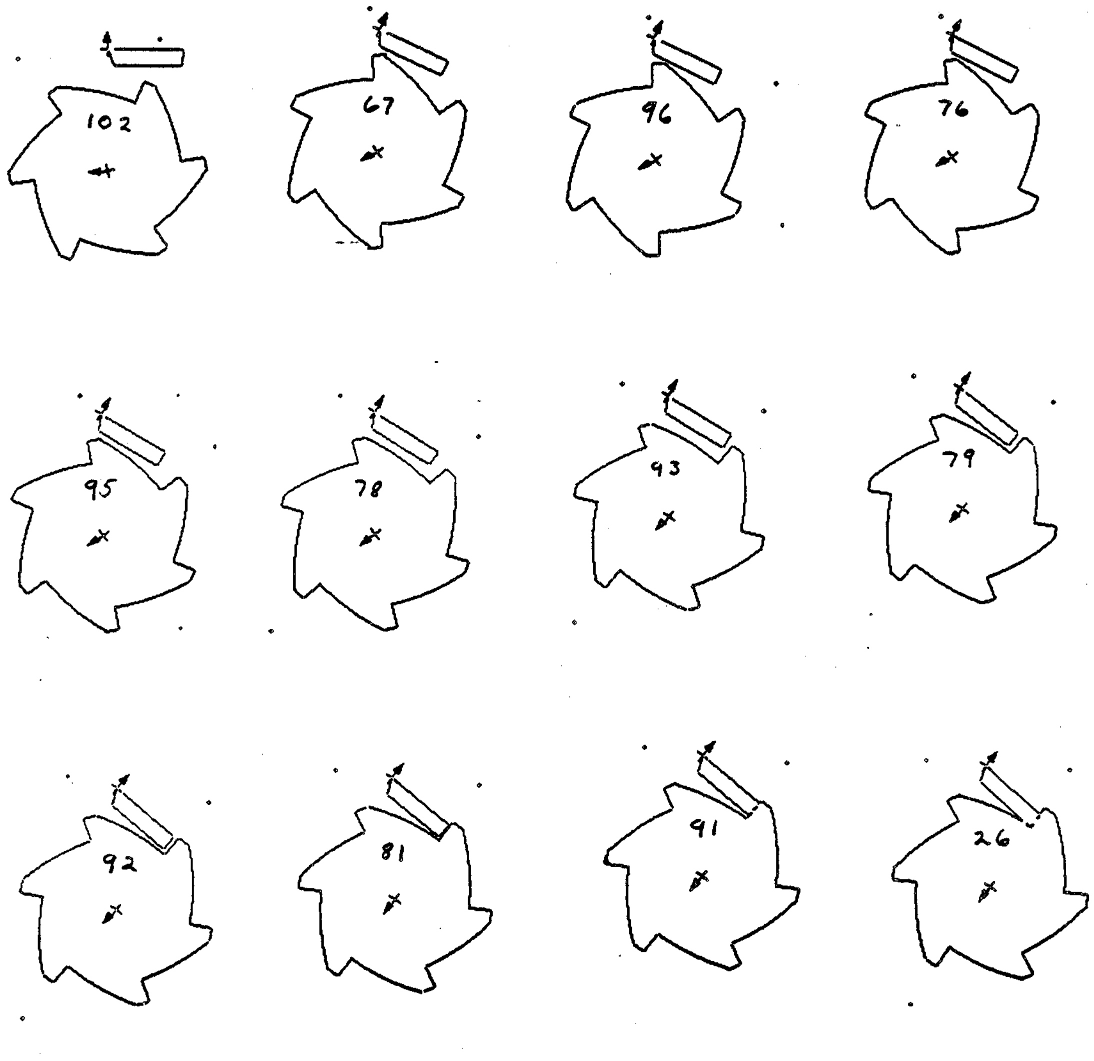


Figure 9.27: Ratchet history for jamming

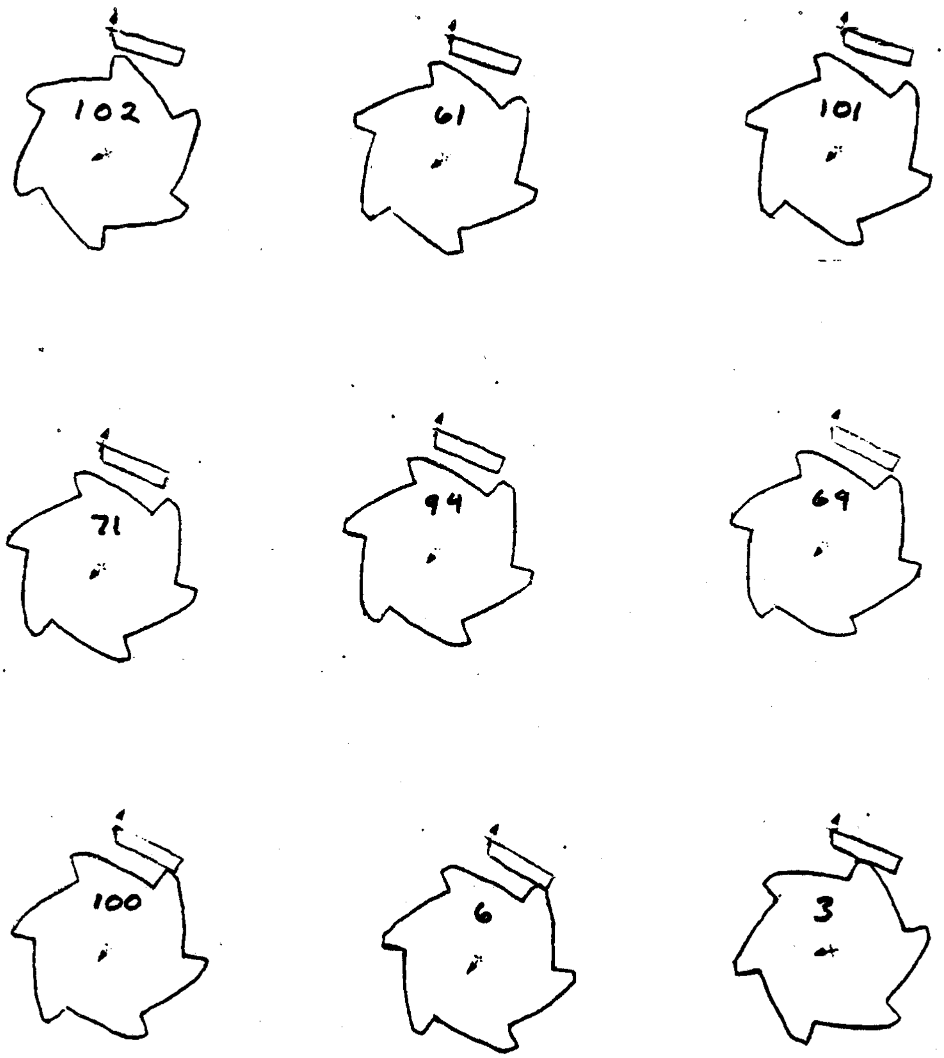


Figure 9.28: Ratchet history for bouncing

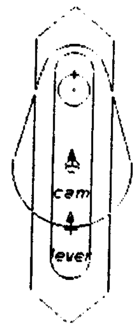


Figure 9.29: Scotch Yoke

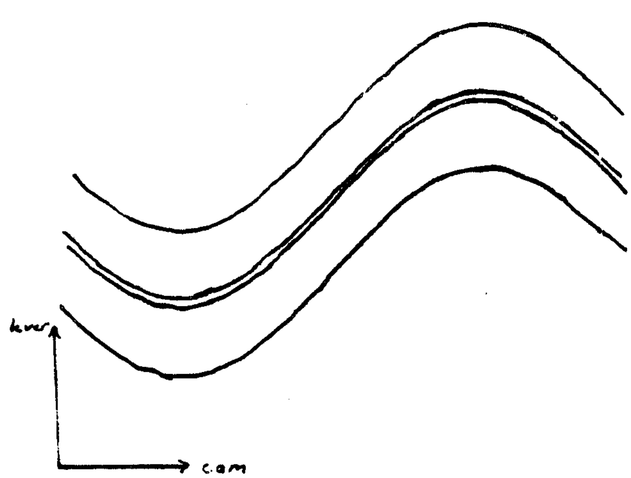


Figure 9.30: Scotch Yoke in Configuration Space

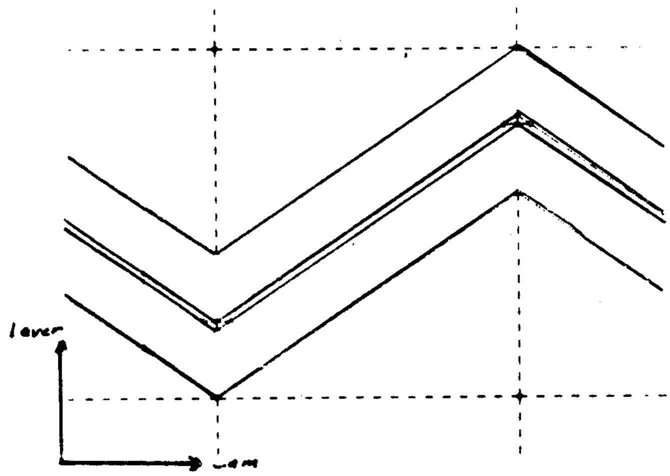


Figure 9.31: Place Vocabulary for Scotch Yoke

to rotate, but the yoke intermittently stops and changes directions. Appendix C gives the complete transition table for this mechanism.

9.4 Cam

The name "cam" is applied to various forms of revolving, oscillating, or sliding machine members which have edges shaped so as to impart a motion to a second part (a follower) which is usually variable and quite complex in many cases. The exact movement derived from any cam depends on the shape of the edge. Figure 9.33 shows one such cam. Its configuration space is given in Figure 9.34 and place vocabulary in Figure 9.35.

Unlike the Skotch Yoke, which only needs a force acting on one of the elements, the cam must have force on both objects. One drives the wheel and another directs the follower toward the cam wheel. If the force on the follower were not present, it would be pressed out of the way after one cycle, and the cam would turn freely. This force, however, increases ambiguity. Normally the cam pushes the lever through a sequence of complex motions. If the force on the follower is sufficient, it may push the cam wheel backward for some period of time. Some sample configurations of the cam are given in Figure 9.36.

9.5 Discussion

In this section we investigate the correctness of the envisionments produced by this method. There are four major questions we will address:

1. Are all of the individual states correct?

2. Are all of the individual state transitions correct?

3. Does the envisionment show the actual behavior of the mechanism?

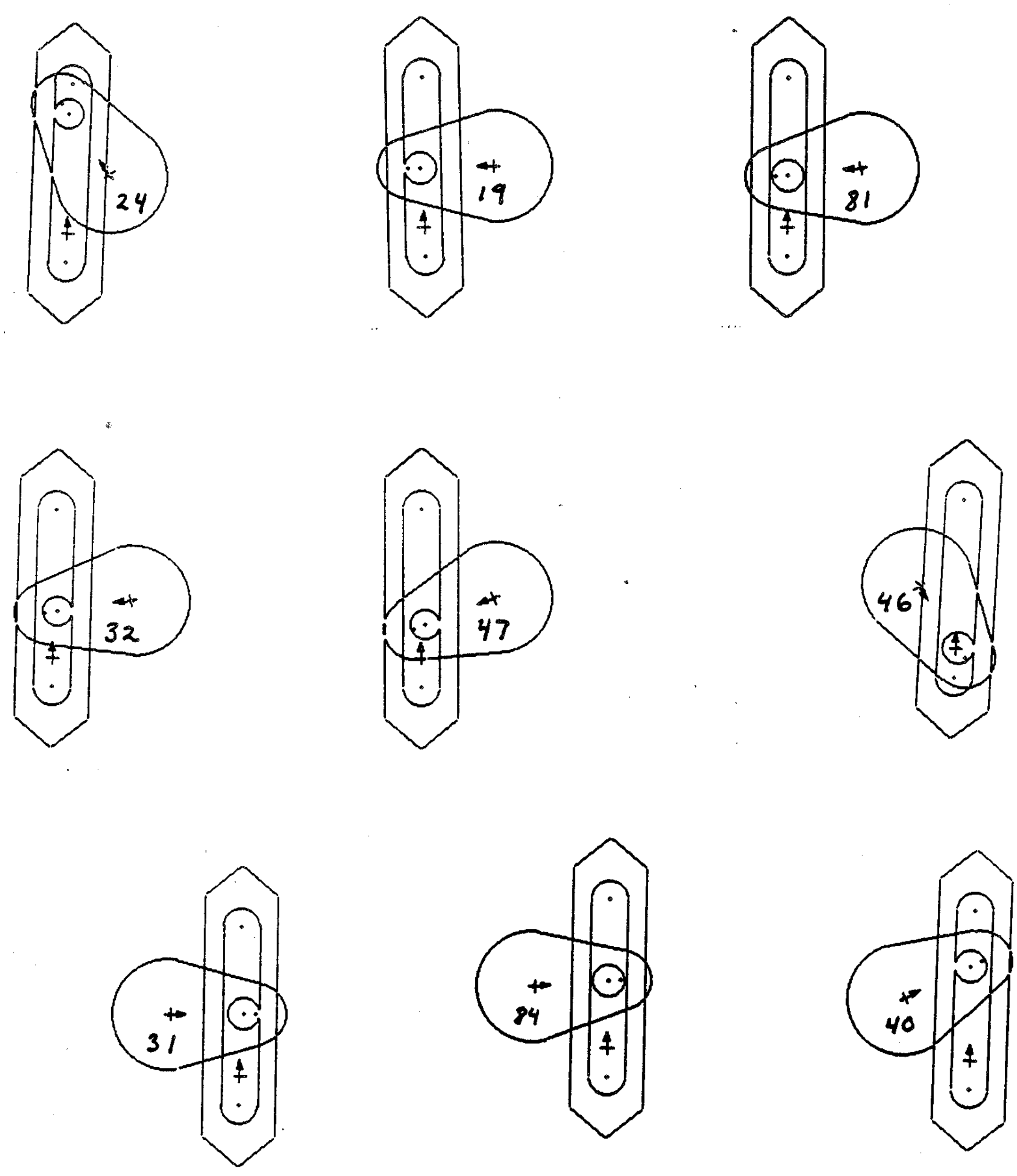


Figure 9.32: Scotch Yoke history

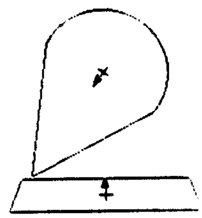


Figure 9.33: Cam

4. Are all the other behaviors in the envisionment possible?

By inspection we have determined that all of the states and single state transitions produced by `Clock` are possible, and further that these states and transitions correspond to possible behaviors. Since our algorithm determined all possible states and all possible transitions between single states, that result was expected. Not all of the possible states were generated because of abstraction (section 8.1.6), but all of the possible states and transitions for these abstract mechanisms were produced.

Within the envisionments produced here we can find histories corresponding to the expected behavior of the mechanism. These histories have been indicated throughout the examples shown.

Originally de Kleer claimed that every path through an envisionment must represent a physically realizable history (de Kleer, 1975) but Kuipers (Kuipers, 1986) showed that there could be paths which corresponded to counter-intuitive results.

In the case of the mechanical clock, consider the instant of contact between the pallets and the scape wheel. On each cycle one of three outcomes is possible: the pallets could forcefully jam into the scape wheel, they could exactly counteract each other, and finally the force of the scape wheel would suffice to immediately push the pallets backward. Due to ambiguity, any one of these outcomes is initially possible. However, once we commit to one of the outcomes it should hold unless the initial conditions change. Currently in our program there is no way to state that the initial conditions have not changed. Consequently successive cycle fluctuation between the various outcomes is possible.

Since we only know the direction of the forces, we have no reason to believe that their effects remain constant over time. That is, just because one branch was chosen in one iteration of a loop through the envisionment, the next iteration may choose another branch. To eliminate these behaviors we need more information about global considerations, such as energy.

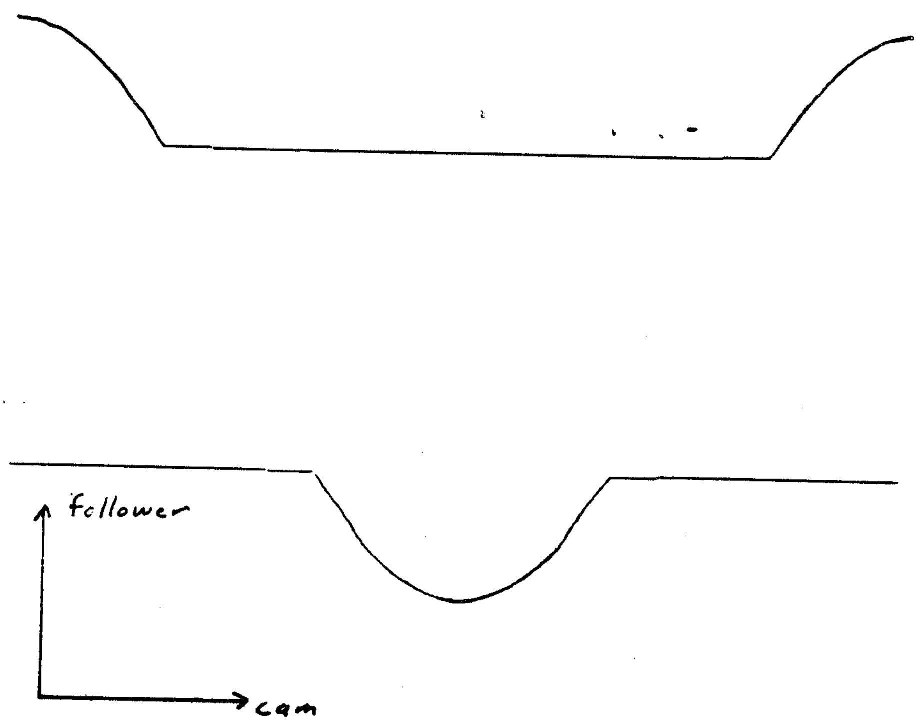


Figure 9.34: Configuration Space for Cam

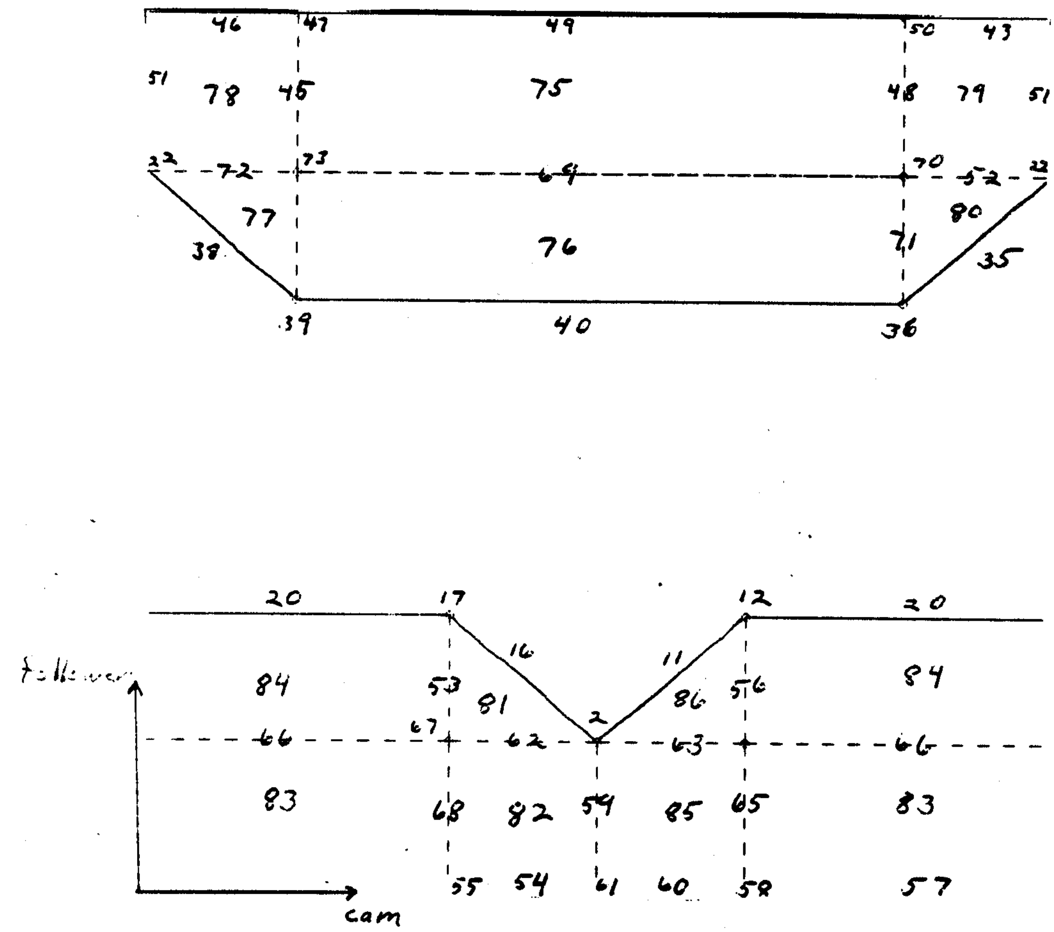


Figure 9.35: Place Vocabulary for Cam

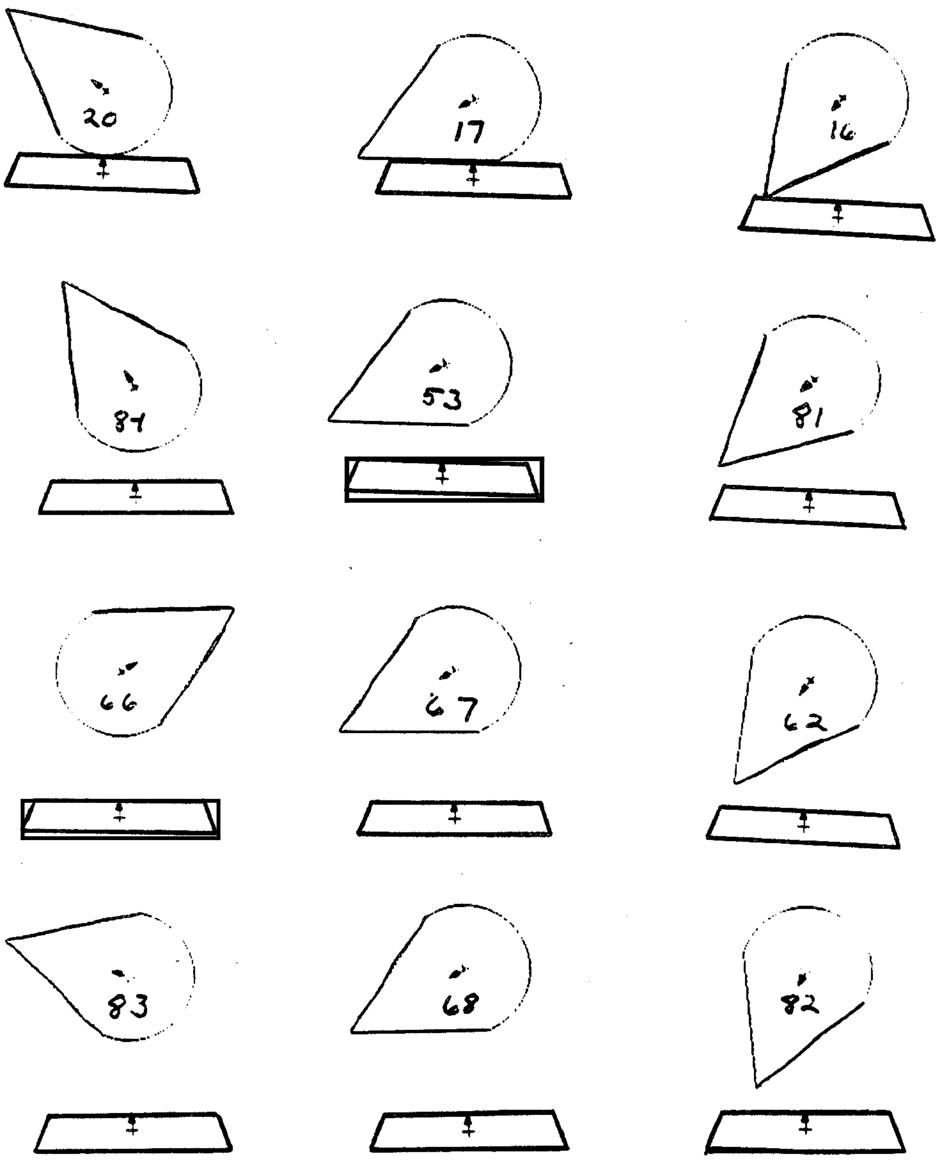


Figure 9.36: Sample Configurations of Cam

Energy should be considered during the history generation stage. This prevents the envisionments from becoming unnecessarily complex. Once a path through the envisionment is selected we must keep track of the branch chosen. When a path corresponding to an oscillation returns to the original node, we must select either the same path or one at a lower energy level as the previous iteration.

Part IV

Conclusion

Chapter 10

Related Work

There have been a number of investigations of common sense theories of kinematics, most notably (Davis, 1986; de Kleer & Brown, 1984; Forbus, 1981; Faltings, 1987a; Faltings, 1987b; Gelsey, 1987; Joskowicz, 1987a; Kuipers, 1986; Laughton, 1985; Pu & Badler, 1988a; Pu & Badler, 1988b; Shoham, 1985). This chapter reviews relevant AI research in naive physics and qualitative physics.

10.1 Naive Physics

Naive Physics was introduced by Hayes (Hayes, 1979; Hayes, 1985) as an attempt to represent the common sense knowledge that people have about the physical world. Rather than formalizing highly specialized information in a narrow domain, as had been done in previous AI research, he proposed to capture general information in a much larger domain. Hayes believes that "we are never going to get an adequate formalization of common-sense by making short forays into small areas, no matter how many of them we make."

One of the most interesting parts of Hayes' naive physics concerns how this knowledge is to be organized. Hayes views a naive physics as consisting of a collection of *clusters*, theories which internally are "tightly connected" but only "loosely connected" to other theories (in the sense of number of shared predicate symbols and constants). Some examples Hayes provides of clusters are measuring scales; shape, orientation, and dimension; inside and outside; histories; energy; assemblies; support; substances and physical states; forces and movement; and liquids. Our work addresses the support and the forces and movement clusters, but requires concepts from other clusters.

10.1.1 Shoham's Naive Kinematics

Naive Kinematics (Shoham, 1985) was an attempt to apply the methodology of Naive Physics to the field of kinematics. That paper presents theorems of translational and rotational freedom when an object's surface is in contact with one, two, or three points on fixed obstacles. Shoham makes the simplifying assumptions, based on early work in kinematics (Reuleaux, 1876), that all objects are everywhere smooth, that only one object can move, space is fixed, and discussion is restricted to two dimensions and three degrees of freedom, including rotational. Because the analysis is done on a point by point basis it seems unlikely that this approach will be applicable to understanding real mechanical objects. His theory has not, to our knowledge, been tested by implementation. Furthermore, many of his theorems are non-constructive and hence of unclear utility.

Even if the object were such that some number touchpoints would do, one needs some method of determining where these points should be located. A program asked, "Is this object free to translate?" which counted touchpoints up to two and then returned *ambiguous*, would be useless. Even the smallest surface consists of infinite points. When we are concerned with the practical aspect of determining behavior from geometry, we would prefer a program which would show where to place obstacles about an arbitrary object in order to constrain it in some manner, and when faced with an object surrounded by obstacles determine the constraints already imposed. Our theory does that, Shoham's does not.

10.2 Qualitative physics

Qualitative physics is similar to the naive physics of Hayes, but there are important differences. Both want to identify the core knowledge that underlies physical intuition. Naive physics ignores the actual mechanisms used to derive inferences while Qualitative physics also studies *how* inferences are drawn. Naive physics does not propose to make a computer program that can use the resulting formalism in some sense. Hayes believes that this detracts from the main goal and believes the results of a working program may be deceptive. Adherents of qualitative physics believe the building of programs helps to expose flaws in the theory. When working in a highly intuitive domain there is a strong tendency to overlook critical considerations which may be caught by a program. Finally, qualitative physics is concerned with modeling expert reasoning as well as that exhibited by the person on the street, while naive physics pursues common sense theories—even though they may be approximate, incorrect, or inaccurate.

10.2.1 NEWTON

NEWTON was the first qualitative physics program (de Kleer, 1975). It solved problems involving a point mass sliding on single, two-dimensional surfaces (the "roller coaster" world). That thesis was the first to postulate that even when you wanted to have an exact solution, you might first want to do a qualitative simulation to guide the formulation of the problem. The notion of envisioning was introduced in NEWTON.

The problem of objects interacting with surfaces is interesting as a subproblem of almost all kinematics problems, but the representation of objects moving in a "roller coaster" world is essentially one dimensional. NEWTON'S inability to focus on more than one moving object was a fundamental weakness for a general mechanics problem solver.

10.2.2 FROB

FROB was built to explore the qualitative and geometric knowledge necessary to understand motion through space under the influence of gravity (Forbus, 1981). The moving objects are balls (actually point masses) in a two dimensional world which may collide with each other as well as surfaces, or they may leave the surface and act only under the influence of gravity. FROB introduced much of the framework for qualitative kinematics used here, including the concept of place vocabularies.

FROB was only concerned with two dimensions and two degrees of freedom. While it was capable of investigating interaction between two moving bodies, the only question it attempts to answer is whether contact is possible. Interaction between moving objects was constrained by non-moving surfaces, and the way the behavior of the system might change as a result of contact between moving bodies was not investigated.

10.2.3 Qualitative Process Theory

Qualitative Process (QP) theory (Forbus, 1984) characterizes the dynamic aspects of qualitative physics. The fundamental notion of QP theory is that a physical process acts through time to bring about change, and that all changes in physical systems are caused (directly or indirectly) by physical processes. QP Theory can decide which processes are active, decide what changes can happen, and deduce possible behaviors of a system.

In QP theory quantities are expressed in terms of inequalities in a partial ordering called the *quantity space*. Processes typically begin and end when inequalities change. For example if two objects are in heat contact a heat flow process will occur until their temperatures are equal. Influences tell what can cause a quantity to change. Qualitative proportionalities tell how quantities are affected by other quantities. Together qualitative proportionalities and influences form a qualitative language for differential equations.

10.2.4 History generation

Kuipers (Kuipers, 1984; Kuipers, 1986) explores the possibility of generating histories directly rather than generating complete envisionments. The QSim system generates all possible next states from the current state and then prunes them according to heuristics, real world constraints, and user selection. Because only one path is explored the results may not be complete and may overlook potential disaster states.

10.3 Joskowicz' Kinematic Analysis

Joskowicz (Joskowicz, 1987a; Joskowicz, 1987b) presents a qualitative analysis of mechanisms by recognition of lower pairs from configuration space representations. Having done this he uses propagation of motion parameters to determine motions of all the components. He considers changes in contact between objects, but not as a result of the normal operation of the device. Instead, an outside agent makes and breaks pairs, as might happen in an automobile transmission.

10.4 Machine Geometry

Gelsey (Gelsey, 1987) begins with a CAD/CAM model of a mechanical device and uses constructive solid geometry to produce a kinematic analysis of lower pairs, cams, and gears. This approach allows three dimensional modeling of complex devices such as differential mechanisms and piston and crankshaft mechanisms. Though this approach lacks the generality to analize arbitrary pairs, a preanalysis of lower pairs and known components would help limit the complexity of a QM analysis.

Chapter 11

Discussion

11.1 Summary

We have presented a theory of qualitative mechanics sufficient to produce a qualitative, symbolic descriptions of mechanical behavior in the form of an envisionment. These ideas, combined with Falting's work, have for the first time allowed the qualitative simulation of a mechanical clock. While we have used the mechanical clock as a motivation these ideas are widely applicable to analysis of mechanisms containing pairs that are force coupled or that change the shape of the surfaces that come into contact.

We have presented three aspects of the kinematic analysis of a rigid body device; the constraint of motion, the transfer of motion, and the propagation of external forces. In order to do this we developed a logical theory of rigid body interactions. This theory provides a symbolic framework for geometric descriptions and laws describing mechanical constraint and motion transfer.

The principles of blocking and pushing introduced here allow us to determine the instantaneous behavior of a mechanism. Typically a mechanism will have all of its motions constrained but one and the only significant forces transmitted through contact. To determine the instantaneous behavior of the mechanism from any given configuration we first compute the constrained motions. After the impossible motions have been eliminated, the intersection of the free directions and the motions transferred by pushing should yield a single set of consistent motions for each part of the mechanism. If there is no consistent set the mechanism cannot move.

A *place* serves as a symbolic description of space. From our understanding of reference frames we have constructed place vocabularies which are partitioned according to the relevant consideration of possible motions. This reduces the complexity and ambiguity of qualitative mechanical analysis. We further constructed place vectors representing configuration at the system level rather than the component level. We also demonstrated a method of abstracting geometric information to reduce the complexity of symbolic information, so that entire mechanisms can be analyzed with less of a combinatoric explosion.

We believe that QM is not only useful when representing common sense reasoning, but also, expert level analysis. We believe the differences between the two types of reasoning lie both in faulty mental models and the complexity of the resultant descriptions. All of the models presented here should be comprehensible to a child yet the combination of these ideas is sufficient to explain such devices as the mechanical clock.

A reader contemplating the conclusions rendered by `Clock` might perhaps dismiss them as "obvious". I feel this criticism validates the hypothesis that this is indeed common sense reasoning. The truth is that the principles discussed here are not obvious to computers. Much work remains to be done before they have even a childish understanding of our physical world, but I believe that this work represents a significant step toward that understanding.

11.2 Applications

Because of the ubiquitous nature of mechanisms in our society, we expect our results to be useful in a wide variety of applications. Some of these potential applications are discussed here.

11.2.1 Determining Activity

Determining activity involves answering the question, "What is going on?" Given a geometric description of a device we would like to determine what it does. We might only be concerned with the current situation or with the overall behavior. An envisionment is the qualitative answer to this question. QM envisionments describe activity from the geometry and from high level representations of force and motion. Envisionment is preferable to numerical simulation for answering this question because envisionment describes all possible activities, while a simulation only follows a few paths and risks oversight.

Other information the envisionment provides directly includes prediction and postdiction: "What happens next?" and "What just happened?" We can answer these questions by, respectively, tracing forward or backward through the envisionment. Mechanisms, typically, are designed to stablize behavioral anomalies, so in our domain the question of prediction has less solutions than the problem of postdiction. However, if an unstable behavior is detected we could look back and determine how it could get there.

11.2.2 Suitability

Given some description of the desired behavior and a complete envisionment of some device we would like to determine whether the device is capable of displaying the described behavior. This involves comparing the state sequence of the behavioral description with the envisionment and searching to see if such a sequence can occur. There might be further information about undesirable states, behaviors that we would not want to happen during the operation of the device in question.

11.2.3 Behavioral Indexing

Rather than classify parts by shape, a qualitative analysis could compare the envisionments between mechanism components to determine if they exhibit analogous behavior (Falkenhainer, 1986). This could be used to provide indexing between components in a part library or between successive design revisions. For example, even though the geometry of a pulley, a rod, and set of gears is radically different they all exhibit similar behavior, transfer of angular motion.

11.2.4 Maintenance, Fault Detection, and Diagnosis

Qualitative models offer an advantage over mathematical models in that qualitative models can be causal. They can attribute behaviors to certain parts of the system and explain the considerations involved in achieving a conclusion. When a device is not behaving as desired we would like to be able to localize faults and suggest possible causes. One way to do this would be to compare a qualitative description based on the design with the observed behavior of the device. If the the predicted and observed behaviors do not match we could reconstruct the causal chain which determined why the observed transition was not possible.

11.2.5 Control

There are three major focuses of investigation in mechanical systems:

1. The input forces
2. The arrangement of parts
3. The output behavior

Our system determined the last when given the first two. Now we would like to investigate the problem of determining either the first of the second when given the other two. The problem of determining the input forces necessary to produce a given behavior from a specific arrangement of parts is *control*, and the problem of determining the arrangement of parts when given the forces and desired behavior is *design*. Currently there are no good methods to automatically help in these stages.

A qualitative control system would accept the desired behavior of a mechanism in the form of a state sequence. Determining the external forces would be a matter of noticing transitions within the state sequence. Change in motion must be caused either by contact or through some external force, but contact may be ruled out from a description of the interaction of the parts. Using the techniques of section 3.3 we could tell which direction(s) of force would affect behavior in the desired manner.

11.2.6 Reverse Engineering

The first step in automating design is reverse engineering. Given a complete behavioral description a device and a partial description of its components we would like to determine what other objects must be present in order for the device to display this behavior. We can approach this problem through causal information about motion constraint. This allows us to postulate where an object's surface might be, what its orientation is, and what its motion constraints must be.

11.2.7 Design and Manufacturing

A designer can propose a configuration space representation of a behavior he wants to implement, and in the worst case we might automatically cut a cam with the profile of the configuration space cut into it. The closer the cam follower is to a point, the better this would work.[1] However, configuration spaces are not intuitive, and most designers would find it easier to design the cam in the first place than try to understand configuration space.

What we propose as an alternative is a tool to help the designer through design revision at the early stages. Most designers can produce a rough implementation of the behavior they wish to achieve or begin with a cam discussed above. As design revision proceeds the qualitative reasoner could serve as a designer's apprentice. Given a behavioral description of a mechanism and the geometry of the parts we could determine if the proposed machine can exhibit the desired behavior. This would provide feedback prior to extensive mathematical analysis and indicate why a device may not work.

Reuleaux considers the process of machine development to be the replacement of force-closure by pair- and chain-closure and goes even further to say "we must consider it as an essential characteristic of future machine-development.(Reuleaux, 1876)" There are many tools to assist the designer once the design has been formalized, but a qualitative analysis will assist in the earlier stages, and its analysis may be successively enriched by more exact mathematical representations as the degree of formalism increases.

11.2.8 Expert Systems

In current expert systems the knowledge representation is stratified. They can neither work above *nor below* their level of expertise. They suffer from incompleteness and do not degrade gracefully when a problem falls beneath their area of competence. Qualitative models can provide a low level analysis which completely describes device behavior and can answer simple questions easily.

Another problem expert systems suffer from is GIGO (garbage in / garbage out) behavior. The system will try to provide an answer for whatever input the user provides. Qualitative models provide a simple way of input verification. Symptoms which do not correspond to any possible behavioral states or state transitions may be isolated for further verification.

11.2.9 Robotics

For a robot to interact with the physical world and manipulate its environment it must be able to accomplish tasks such as turning knobs, opening doors, lifting boxes, and stacking objects. Except in a highly artificial environment these tasks all require deep knowledge of the basic underlying principles of mechanics discussed in this paper. We can neither equip a robot with advance knowledge of all the objects it might encounter, nor can it be expected to solve complex equations to predict the effects of an approaching truck. Qualitative mechanics provides representations and process control for everyday interactions with the physical world.

11.3 Further Work

11.3.1 Multiple perspectives

A more detailed analysis of a mechanism might look at the way a component participates in the overall behavior of the mechanism, then look at a more detailed analysis of that

[1] Current work by Joskowicz (Joskowicz & Addanki, 1988) explores design from configuration space.

component to provide further insight into the behavior of the mechanism. For example, some gears will only turn freely in one direction. If we construct an approximate description of ideal gears which turn freely, we should then go back and see if the behavior of the gear in the overall mechanism violates its unidirectional constraint.

11.3.2 Combining open and closed pairs

Our analysis of qualitative mechanics best represents open (force closed) pairs such as would be found in intermittent motion devices and at the early stages of mechanism design. The problem of representing closed (kinematically coupled) pairs has received much more attention and there exists an exact mathematical analysis of their behaviors (Denavit & Hartenberg, 1955). No such solution exists for open pairs, and because of their infinite variety, such a solution is unlikely.

Work by Gelsey (Gelsey, 1987) demonstrates that rule based methods exist for the identification and analysis of closed pairs. We assume this analysis occurred prior to the configuration space representations received as inputs to our program. Our next step is to combine our methodology with that of Gelsey to provide a more complete mechanism analysis.

11.3.3 Combining approaches

People use a wide spectrum of methods for the analysis of mechanisms. The first principles approach presented here provides a method for analysis of new mechanisms as well as a way of storing information for analysis of identical parts which reoccur in other mechanisms. Other approaches to mechanics need to be explored and the results integrated with the techniques presented here.

We would like to allow for functional descriptions when such information is known. This additional information may help reduce the ambiguity of the qualitative analysis. It is necessary to use both functional and first principles descriptions when comparing the desired behavior of a device with its predicted behavior, as would be done in mechanism design.

We would like to explore case based and analogical reasoning to allow analysis of similarity in mechanisms. To some extent similar mechanisms behave in a similar fashion; however, those approaches fail when the differences become significant. For example, scape wheels, ratchet wheels, and gears are similar yet exhibit different behavior. An integrated approach would provide concise analysis of comparable mechanisms, deep analysis of novel mechanisms, and the ability to determine the limits of similarity.

11.3.4 Increasing scale

As the complexity of mechanism analysis increases we will be unable to produce complete solution spaces explicitly in the form of total envisionments from a first principles approach. We must either reduce the size of the envisionments by varying the depth of the analysis or find alternative control methods to limit the size of the space which must be generated.

One limitation sometimes imposed for tractability is exploring only a single history of a behavior. That approach corresponds to an envisionment in which we have sufficient information to resolve all ambiguities. It may suffice for certain tasks, such as determining whether a given behavior might occur, and, if it could reason backward as well as forward, it could tell how an undesired behavior might be achieved. Some problems with this approach include :

1. a certain amount of ambiguity is inherent in everyday problems;

2. in design problems we would want to begin with all possible behaviors and successively constrain undesirable behaviors;

3. that approach does not suffice for exploring device safety and determining all failure modes.

To a large extent the type of questions we expect to answer will force our selection of control. We want to look at the form of the question we are trying to answer and from that use minimal information to produce the simplest explanation. For example, to determine what happens next or what might have just happened, we only need to generate the adjacent states from the current state. In (Falkenhainer & Forbus, 1988) they discuss methods of explicitly representing modeling, simplifying, and operating assumptions based on analysis of the question posed.

11.3.5 Leaving configuration space

Producing a configuration space representation of a mechanism is time intensive, and the results are not intuitive. Faltings (Faltings, 1988) proposes a method of computing place vocabularies directly from the component parts without the need for a configuration space. While his theory has not yet been implemented and tested empirically, he claims[2] that it may increase the speed of computing place vocabularies by as much as two orders of magnitude! We would like to determine if the results are compatible with the place vocabularies and abstractions of place vocabularies described in this thesis. We would also like to see this analysis extended to more complex shapes.

11.3.6 Geometric interface

To better assist with mechanical analysis we need to get geometric input from different sources. Our current implementation accepts inputs similar to that produced by current CAD/CAM systems. We would like to further the interaction between this system and the real world by exploring the way visual analysis would provide the geometric inputs required by this system.

To analyze existing mechanisms and diagnose faults we need to represent the actual mechanism, not an idealized design. We need to distinguish the surfaces of objects and determine which side is solid and which is open. Work in vision has shown that the humans compute a symbolic representation of an image at a very low level (Marr, 1982). QM provides demands for the type of information that must be there, and expectations of other information that will focus attention.

11.3.7 Extension to higher dimensions

[2] Personal communication.

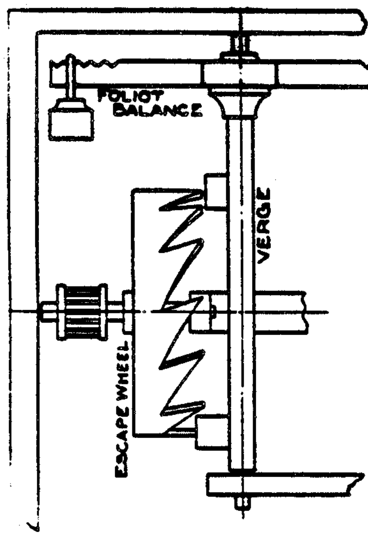


Figure 11.1: Verge escapement

While most mechanical problems may be analyzed by finding a suitable projection into two dimensions there are some which cannot. Consider the verge escapement (shown in figure 11.1). While each of its components have only a single degree of freedom, the fact that each component is rotating about an shaft perpendicular to the other makes this an inherently three dimensional problem.

The representations discussed in this thesis are easily extensible to higher dimensions, and the laws of qualitative dynamics will work with any number of dimensions, though their complexity will increase exponentially. Partitioning place vocabularies is somewhat more complex in higher dimensions, each added dimension introduces a new place descriptor and, again, the number of possible place transitions increases as the exponent. The major problem in reasoning in higher dimensions is finding a suitable input representation. More work needs to be done in representation of depth perception and transforming these representations to higher dimensional metric diagrams such as N dimensional configuration spaces.

11.3.8 Adding detail

To get finer detail while still maintaining qualitative representations we could resort to comparisons such as (steeper-than $slope_1$ $slope_2$). This creates a quantity space representation for angles, but will only be applicable when comparing angles in the same quadrant. Work in progress by Hyeonkyeong Kim explores better representations of angular quantities.

Another way to get finer detail is to increase the number of directional divisions. For example people might describe direction as some combination of north, south, east, and west at one level of detail, but use terms such as north-northeast for more precise directional information. This would allow us to express concepts such as gear ratio by comparing slopes in configuration space.

11.3.9 Deformable bodies

Mechanical engineers use the rigid body assumption as an abstraction because it reduces the number of considerations they must make in the early design phases. However, not all bodies can be represented as rigid, and as design revision continues they will want representations closer to actual behaviors.

There are many non-rigid bodies which may be modeled with little change to the theories discussed here. In section 5.3.3 we discussed the qualitative representation of a collision between semi-rigid bodies, and in fact we constructed a model of this prior to our model of rigid body collisions. This representation fits the quasi-static assumption used by mechanical engineers.

We would like to explore representations of very elastic materials and the effect this will have on the possible kinematic transitions. For example, if the pallet arms of our clock escapement were made of rubber, it would be possible to transition from the pallets pointed downward to the pallets in the inverted position.

The interactions of completely deformable objects, such as liquids or fibers, and rigid objects may be represented by a configuration space, but they require a new interpretation of blocked space. Rather than representing illegal configurations, it will represent different behavioral regions. We would like to explore the effects of these changes on our qualitative models.

We would like to add a richer representation of material composition to the representations discussed in this thesis. This will enable us to use other qualitative domain theories to reason about pneumatic and hydraulic systems.

Bibliography

Cowie, A. *Kinematics and Design of Mechanisms.* International Textbook Co., Scranton, PA., 1961.

Davis, E. *A Logical Framework for Solid Object Physics.* Technical Report TR No. 245, New York University, Computer Science Department, 1986.

de Carle, D. *Watch and Clock Encyclopedia.* Bonanza Books, New York, 1975.

de Kleer, J and Bobrow, D. Qualitative reasoning with higher-order derivatives. In *Proceedings National Conference on Artificial Intelligence*, American Association of Artificial Intelligence, 1984.

de Kleer, J and Brown, J. A qualitative physics based on confluences. *Artificial Intelligence*, **24**:(7 – 83), 1984.

de Kleer, J. *Qualitative and Quantitative Knowledge in Classical Mechanics.* Technical Report TR-352, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA., 1975.

de Kleer, J. Assumptions and ambiguities in mechanistic mental models. In Dedre Genter and A. S. Stevens (Eds.), *Mental Models*, Erlbaum, 1983.

de Kleer, J. An assumption-based tms. *Artificial Intelligence*, **28**(2), 1986.

de Kleer, J. Extending the atms. *Artificial Intelligence*, **28**(1), 1986.

de Kleer, J. Problem solving with the atms. *Artificial Intelligence*, **28**(2), 1986.

Denavit, J and Hartenberg, R. A kinematic notation for lower-pair mechanisms based on matrices. *Journal of Applied Mechanics*, **22**:(215 – 221), 1955.

Dudley, D. *The Evolution of the Gear Art.* American Gear Manufacturers Association, Washington, D.C., 1969.

Erdman, A and Sandor, G. *Mechanism Design: Analysis and Synthesis.* Volume 1, Prentice-Hall, Inc., Englewood Cliffs, NJ., 1984.

Fahlman, S. E. *A Planning System for Robot Construction Tasks.* Technical Report AI-TR-283, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1977.

Falkenhainer, B and Forbus, K. D. Setting up large-scale qualitative models. In *Proceedings National Conference on Artificial Intelligence*, American Association of Artificial Intelligence, 1988.

Falkenhainer, B. *An Examination of the Third Stage in the Analogy Process: Verification-Based Analogical Learning.* Technical Report UIUCDCS-R-86-1302, Department of Computer Science, University of Illinois, 1986. A summary appears in *Proceedings of the Tenth International Joint Conference on Artificial Intelligence,* 1987.

Faltings, B. *A Theory of Qualitative Kinematics in Mechanisms.* Technical Report UIUCDCS-R-86-1274, University of Illinois, 1986.

Faltings, B. Qualitative kinematics in mechanisms. In *Proceedings of IJCAI 87*, International Joint conference on Artificial Intelligence, Milan, Italy, 1987.

Faltings, B. *Qualitative Place Vocabularies for Mechanisms in Configuration Space.* Technical Report UIUCDCS-R-87-1360, University of Illinois, 1987.

Faltings, B. The use of the metric diagram in qualitative kinematics. In *Qualitative Physics 2nd Workshop*, IBM Paris Scientific Center, Paris, France, 1988.

Forbus, K. *A Study of Qualitative and Geometric Knowledge in Reasoning about Motion.* Technical Report TR-615, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA., 1981.

Forbus, K. *Qualitative Process Theory.* Technical Report TR-789, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA., 1984.

Forbus, K. *The Logic of Occurrence.* Technical Report UIUCDCS-R-86-1300, Department of Computer Science, University of Illinois, 1986.

Forbus, K, Nielsen, P, and Faltings, B. The inferential structure of qualitative kinematics. In *Proceedings of IJCAI 87*, International Joint conference on Artificial Intelligence, Milan, Italy, 1987.

Gelsey, A. Automated reasoning about machine geometry and kinematics. In *Third IEEE Conference on Artificial Intelligence Applications*, 1987.

Hayes, P. J. The naive physics manifesto. In Donald Michie (Ed.), *Expert Systems in the Micro-Electronic Age*, Edinburgh University Press, 1979.

Hayes, P. J. The second naive physics manifesto. In J. Hobbs and R. Moore (Eds.), *Formal Theories of the Commonsense World*, Ablex, 1985.

Joskowicz, L and Addanki, S. From kinematics to shape: an approach to innovative design. In *Proceedings National Conference on Artificial Intelligence*, American Association of Artificial Intelligence, 1988.

Joskowicz, L. *A Framework for the Kinematic Analysis of Mechanical Devices.* Technical Report TR No. 313, New York University, 1987.

Joskowicz, L. Shape and function in mechanical devices. In *Proceedings National Conference on Artificial Intelligence*, American Association of Artificial Intelligence, 1987.

Kuipers, B. Commonsense reasoning about causality: Deriving behavior from structure. In D. Bobrow (Ed.), *Qualitative Reasoning about Physical Systems*, The MIT Press, 1984.

Kuipers, B. Qualitative simulation. *Artificial Intelligence*, **29**:(289 – 338), 1986.

Landies, D. S. *Revolution in Time.* Belknap Press of Harvard University Press, Cambridge, MA., 1983.

Laughton, S. *Explanation of Mechanical Systems Through Qualitative Simulation.* Technical Report AITR85-19, University of Texas at Austin, 1985.

Lozano-Perez, T and Wesley, M. An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, **22**:(560 – 570), 1979.

Lu, S. 1987. Personal Communication.

Marr, D. *Vision.* W.H. Freeman and Co., San Francisco, 1982.

McCarthy, J and Hayes, P. Some philosophical problems from the standpoint of artificial intelligence. In *Machine Intelligence*, Edinburgh University Press, 1969.

Newton, I. *Philosophiae Naturalis Principia Mathematica.* London, 1686.

Pu, P and Badler, N. *Behavior Propagation Simulation of Intermittent Mechanical Systems.* Technical Report, University of Pensylvania, 1988.

Pu, P and Badler, N. *Design Knowledge Capturing for Device Behavior Reasoning.* Technical Report, University of Pensylvania, 1988.

Reuleaux, F. *The Kinematics of Machinery, Outlines of a Theory of Machines.* Macmillan and Co., London, 1876. Translated and Edited by Alex B. W. Kennedy, C.E.

Shoham, Y. Naive kinematics: One aspect of shape. In *Proceedings of the IJCAI 85*, pages (436 – 442), International Joint Conference on Artificial Intelligence, 1985.

Stanfill, C. MACK, a program which deduces the behavior of machines from their forms. *SIGART Newsletter*, **93**:(12 – 16), 1985.

Williams, B. The use of continuity in a qualitative physics. In *Proceedings National Conference on Artificial Intelligence*, American Association of Artificial Intelligence, 1984.

Williams, B. Doing time: Putting qualitative reasoning on firmer ground. In *Proceedings National Conference on Artificial Intelligence*, American Association of Artificial Intelligence, 1986.

Appendix A: Clock Transitions

Current Location	Current Direction of Motion (scape wheel / pallets)								
	- -	- 0	- +	0 -	0 0	0 +	+ -	+ 0	+ +
4	77 - -	77 - 0	5 0 0 5 - - 77 - +	77 0 -	4 - +		77 + - 15 + - 4 0 - 4 0 0		
5	80 - - 16 - -			56 0 -	5 - -		77 + - 4 + -		
9	82 - - 10 - -	57 - 0	81 - +		9 - -	58 0 +	83 + - 12 + -	64 + 0	84 + +
10	82 - - 11 0 0 11 - +	82 - 0	82 - +		10 - -	82 0 +			9 + + 82 + + 10 0 + 10 0 0
11			83 - + 12 - +		11 - +	66 0 +			82 + + 10 + +
12			83 - + 9 - + 12 0 + 12 0 0		12 - + 12 + 0 12 + + 12 + - 12 0 +	83 0 +	11 + + 11 0 0 83 + - 12 0 0 12 + 0	83 + 0	83 + +
15	78 - -	55 - 0	77 - + 4 - +	54 0 -	15 - +		79 + -	67 + 0	80 + + 16 + +
16	15 - - 80 - - 16 0 - 16 0 0			80 0 -	16 - - 16 + 0 16 + + 16 + - 16 0 -		80 + -	80 + 0	80 + + 5 + - 5 0 0 16 0 0 16 + 0
19	76 - - 36 - -			51 0 -	19 - -		73 + - 49 + -		
24	74 - - 52 - -	46 - 0	73 - +		24 - 0	48 0 +	86 + - 44 + -	47 + 0	85 + +
33	86 - -	47 - 0	85 - + 37 - +	45 0 -	33 - 0		75 + -	70 + 0	76 + + 36 + +
36	33 - - 76 - -			76 0 -	36 - -		76 + -	76 + 0	76 + + 19 + - 19 0 0 36 0 0 36 + 0
37	85 - -	85 - 0	50 - + 85 - + 37 - 0	85 0 -	37 - -		85 + - 33 + - 37 0 -		

Current Location	Current Direction of Motion (scape wheel / pallets)								
	- -	- 0	- +	0 -	0 0	0 +	+ -	+ 0	+ +
38		75 - 0 38 - +	75 - +		38 - +	75 0 +		38 + +	53 + + 75 + +
43			86 - + 44 - +		43 - +	45 0 +			75 + + 38 + +
44			86 - + 24 - +		44 - +	86 0 +	43 + + 43 0 0 86 + - 44 0 0 44 + 0	86 + 0	86 + +
45	86 - -	86 - 0	86 - +	43 0 0 45 - -	45 - +	33 + + 33 - + 45 - +	75 + -	75 + 0	75 + +
46	74 - -	71 - 0	73 - +	74 0 -	46 - 0	73 0 +	74 + -	24 + 0 46 0 0	73 + +
47	86 - -	24 - 0	85 - +	86 0 -	47 - 0	85 0 +	86 + -	33 + 0 47 0 0	85 + +
48	73 - -	73 - 0	73 - +	24 + - 24 - - 48 - -	48 - -	50 - + 48 - +	85 + -	85 + 0	85 + +
49	73 - -	73 - 0	19 0 0 19 - - 73 - + 49 - 0	73 0 -	49 - -		73 + - 50 + - 49 0 -		
50	73 - -	73 - 0	73 - + 49 - +	48 0 -	50 - -		85 + - 37 + -		
51	76 - -	76 - 0	76 - +	71 0 - 51 - -	51 - -	19 0 0 51 - +	73 + -	73 + 0	73 + +
52	74 - - 53 - - 52 - 0	74 - 0	74 - +		52 - +	74 0 +			24 + + 74 + + 52 0 +
53	75 - - 38 - 0	75 - 0	75 - +		53 - +	72 0 +			74 + + 52 + +
54	78 - -	78 - 0	78 - +	62 0 - 54 - -	54 - +	15 + + 15 - + 54 - +	79 + -	79 + 0	79 + +
55	78 - -	68 - 0 55 - +	77 - +	78 0 -	55 - +	77 0 +	78 + -	15 + 0 55 + +	77 + +
56	80 - -	80 - 0	80 - +	68 0 - 56 - -	56 - +	5 0 0 56 - +	77 + -	77 + 0	77 + +

Current Location	Current Direction of Motion (scape wheel / pallets)								
	- -	- 0	- +	0 -	0 0	0 +	+ -	+ 0	+ +
57	82 - -	65 - 0 57 - -	81 - +	82 0 -	57 - -	81 0 +	82 + -	9 + 0 57 + -	81 + +
58	81 - -	81 - 0	81 - +	9 + - 9 - - 58 - -	58 - -	59 0 + 58 - +	84 + -	84 + 0	84 + +
59	81 - -	60 - 0	79 - +	58 0 -	59 - 0	69 0 +	84 + -	61 + 0	78 + +
60	81 - -	62 - 0	79 - +	81 0 -	60 - 0	79 0 +	81 + -	59 + 0 60 0 0	79 + +
61	84 - -	59 - 0	78 - +	84 0 -	61 - 0	78 0 +	84 + -	62 + 0 61 0 0	78 + +
62	84 - -	61 - 0	78 - +	63 0 -	62 - 0	54 0 +	81 + -	60 + 0	79 + +
63	84 - -	84 - 0	84 - +	65 0 - 63 - -	63 - -	62 0 + 63 - +	81 + -	81 + 0	81 + +
64	83 - -	9 - 0 64 - -	84 - +	83 0 -	64 - -	84 0 +	83 + -	65 + 0 64 + -	84 + +
65	83 - -	64 - 0	84 - +	66 0 -	65 - -	63 0 +	82 + -	57 + 0	81 + +
66	83 - -	83 - 0	83 - +	11 0 0 66 - -	66 - -	65 0 + 66 - +	82 + -	82 + 0	82 + +
67	79 - -	15 - 0 67 - +	80 - +	79 0 -	67 - +	80 0 +	79 + -	68 + 0 67 + +	80 + +
68	79 - -	67 - 0	80 - +	69 0 -	68 - +	56 0 +	78 + -	55 + 0	77 + +
69	79 - -	79 - 0	79 - +	59 0 - 69 - -	69 - +	68 0 + 69 - +	78 + -	78 + 0	78 + +
70	75 - -	33 - 0	76 - +	75 0 -	70 - 0	76 0 +	75 + -	71 + 0 70 0 0	76 + +
71	75 - -	70 - 0	76 - +	72 0 -	71 - 0	51 0 +	74 + -	46 + 0	73 + +
72	75 - -	75 - 0	75 - +	53 - - 72 - -	72 - +	71 0 + 72 - +	74 + -	74 + 0	74 + +
73	71 - - 51 - - 46 - -	51 - 0 73 - -	19 0 0 19 - - 51 - + 49 - + 73 - 0	46 0 - 73 - -	73 - -	49 - + 73 - +	50 + - 49 + - 24 + - 48 + - 46 + - 73 0 -	50 + - 49 + - 48 + 0 73 + -	50 + - 50 0 0 50 - + 49 + - 49 0 0 49 - + 48 + + 73 0 + 73 0 0 73 + 0

Current Location	Current Direction of Motion (scape wheel / pallets) - -	 - 0	 - +	 0 -	 0 0	 0 +	 + -	 + 0	 + +
74	53 - - 52 - - 72 - - 74 - 0	72 - 0 74 - +	71 - + 46 - + 72 - +	52 - - 74 - -	74 - +	46 0 + 74 - +	52 + + 52 0 0 52 - - 74 0 - 74 0 0 74 + 0	52 + + 74 + +	24 + + 46 + + 52 + + 74 0 +
75	43 0 0 43 - + 38 - 0 45 - - 75 - 0	45 - 0 75 - +	33 - + 45 - + 70 - +	38 0 0 75 - -	75 - +	70 0 + 75 - +	53 + + 53 0 0 53 - - 72 + - 38 + 0 75 0 - 75 0 0 75 + 0	53 + + 72 + 0 38 + 0 75 + +	71 + + 53 + + 72 + + 38 + + 70 + + 75 0 +
76	33 - - 70 - - 36 - -	36 - - 76 - -	36 + + 36 0 0 36 - - 76 - 0	70 0 - 76 - -	76 - -	36 + + 76 - +	71 + - 70 + - 51 + - 76 0 -	51 + 0 76 + -	19 + - 19 0 0 36 + + 51 + + 76 0 + 76 0 0 76 + 0
77	68 - - 56 - - 55 - - 77 - 0	56 - 0 77 - +	5 0 0 5 - - 56 - + 4 - +	55 0 - 77 - -	77 - +	4 - + 77 - +	15 + - 4 + - 55 + - 77 0 - 77 0 0 77 + 0	4 + - 77 + +	4 + - 4 0 0 4 - + 77 0 +
78	59 - - 61 - - 69 - - 78 - 0	69 - 0 78 - +	68 - + 55 - + 69 - +	61 0 - 78 - -	78 - +	55 0 + 78 - +	62 + - 54 + - 61 + - 78 0 - 78 0 0 78 + 0	54 + 0 78 + +	15 + + 55 + + 54 + + 78 0 +
79	62 - - 60 - - 54 - - 79 - 0	54 - 0 79 - +	15 - + 54 - + 67 - +	60 0 - 79 - -	79 - +	67 0 + 79 - +	59 + - 69 + - 60 + - 79 0 - 79 0 0 79 + 0	69 + 0 79 + +	68 + + 69 + + 67 + + 79 0 +

Current Location	Current Direction of Motion (scape wheel / pallets) - -	- 0	- +	0 -	0 0	0 +	+ -	+ 0	+ +
80	15 - - 67 - - 16 - - 80 - 0	16 - - 80 - +	16 + + 16 0 0 16 - -	67 0 - 80 - -	80 - +	16 + + 80 - +	68 + - 67 + - 56 + - 80 0 - 80 0 0 80 + 0	56 + 0 80 + +	5 + - 5 0 0 16 + + 56 + + 80 0 +
81	65 - - 63 - - 57 - -	63 - 0 81 - -	62 - + 63 - + 60 - + 81 - 0	57 0 - 81 - -	81 - -	60 0 + 81 - +	9 + - 58 + - 57 + - 81 0 -	58 + 0 81 + -	59 + + 60 + + 58 + + 81 0 + 81 0 0 81 + 0
82	11 0 0 11 - + 10 - - 66 - -	66 - 0 82 - -	65 - + 57 - + 66 - + 82 - 0	10 - - 82 - -	82 - -	57 0 + 82 - +	10 + + 10 0 0 10 - - 82 0 -	10 + + 82 + -	9 + + 57 + + 10 + + 82 0 + 82 0 0 82 + 0
83	12 + - 12 0 0 12 - +	12 - + 83 - -	9 - + 12 - + 64 - + 83 - 0	12 + - 83 - -	83 - -	64 0 + 83 - +	11 + + 11 0 0 66 + - 12 + - 83 0 -	66 + 0 83 + -	65 + + 66 + + 64 + + 83 0 + 83 0 0 83 + 0
84	9 - - 64 - - 58 - -	58 - 0 84 - -	59 - + 58 - + 61 - + 84 - 0	64 0 - 84 - -	84 - -	61 0 + 84 - +	65 + - 64 + - 63 + - 84 0 -	63 + 0 84 + -	62 + + 61 + + 63 + + 84 0 + 84 0 0 84 + 0
85	24 - - 47 - - 48 - -	48 - 0 85 - -	50 - + 37 - + 48 - + 85 - 0	47 0 - 85 - -	85 - -	37 - + 85 - +	33 + - 37 + - 47 + - 85 0 -	37 + - 85 + -	37 + - 37 0 0 37 - + 85 0 + 85 0 0 85 + 0
86	44 + - 44 0 0 44 - + 86 - 0	44 - + 86 - +	24 - + 44 - + 47 - +	44 + - 86 - -	86 - +	47 0 + 86 - +	43 + + 43 0 0 44 + - 45 + - 86 0 - 86 0 0 86 + 0	45 + 0 86 + +	33 + + 47 + + 45 + + 86 0 +

Appendix B: Ratchet Transitions

Current Location	Current Direction of Motion (ratchet wheel / pawl)								
	- -	- 0	- +	0 -	0 0	0 +	+ -	+ 0	+ +
3	100 - - 6 - -	55 - 0	99 - +		3 + -	56 0 +	96 + - 27 + 0	67 + 0	102 + +
6	100 - - 17 - - 6 0 0 6 - 0	100 - 0	100 - +		6 - 0 6 - + 6 - - 6 + + 6 0 +	100 0 +			3 + + 100 + + 6 0 + 6 0 0
8	9 - - 89 - - 8 0 - 8 0 0			89 0 -	8 + +		89 + -	89 + 0	89 + + 10 + - 10 0 0
9	98 - -	54 - 0	97 - + 11 - +	53 0 -	9 + +		88 + -	84 + 0	89 + + 8 + +
10	89 - - 8 - -			52 0 -	10 + -		86 + - 44 + -		
11	97 - -	97 - 0	45 - + 97 - + 11 0 0 11 - 0	97 0 -	11 - 0 11 - + 11 - - 11 + - 11 0 -		97 + - 9 + - 11 0 - 11 0 0		
16	90 - -	90 - 0	17 - + 90 - + 16 0 0 16 - 0	90 0 -	16 + -		90 + - 18 0 -		
17	93 - -	49 - 0	94 - +	46 0 -	17 + -	69 0 +	90 + - 16 + -		100 + + 6 + +
18	91 - -	40 - 0	90 - + 16 - +	28 0 -					
26			91 - + 47 - +			28 0 +			
27			96 - + 3 - +			96 0 +		96 + 0	96 + +
28	91 - -	91 - 0	91 - +	26 0 0		18 - +			
40	91 - -	80 - 0 40 - -	90 - +	91 0 -	40 + -	90 0 +	91 + -	18 0 0 40 + -	90 + +
41			92 - + 42 - + 41 0 + 41 0 0		41 + -	92 0 +	48 + - 92 + -	92 + 0	92 + +

Current Location	Current Direction of Motion (ratchet wheel / pawl)								
	- -	- 0	- +	0 -	0 0	0 +	+ -	+ 0	+ +
42			95 - + 50 - +		42 + -	78 0 +	92 + - 41 + -	79 + 0	93 + +
43	86 - -	86 - 0	86 - +	74 0 - 43 + -	43 + +	45 - + 43 + +	97 + -	97 + 0	97 + +
44	86 - -	86 - 0	10 0 0 10 - - 86 - + 44 0 0 44 - 0	86 0 -	44 - 0 44 - + 44 - - 44 + - 44 0 -		86 + - 45 + - 44 0 - 44 0 0		
45	86 - -	86 - 0	86 - + 44 - +	43 0 -	45 - 0 45 - + 45 - - 45 + - 45 0 -		97 + - 11 + -		
46	93 - -	93 - 0	93 - +	80 0 - 46 + -	46 + -	17 0 + 46 + +	90 + -	90 + 0	90 + +
47			91 - + 48 - + 47 0 + 47 0 0		47 + -	91 0 +	26 0 0 91 + -	91 + 0	91 + +
48			92 - + 41 - +		48 + -	81 0 +	91 + - 47 + -	91 + 0	91 + +
49	93 - -	77 - 0 49 - -	94 - +	93 0 -	49 + -	94 0 +	93 + -	17 + - 17 + + 49 + -	94 + +
50			95 - + 51 - + 50 0 + 50 0 0		50 + -	95 0 +	42 + - 95 + -	95 + 0	95 + +
51			96 - + 27 - +		51 + -	96 0 +	95 + - 50 + -	76 + 0	96 + +
52	89 - -	89 - 0	89 - +	83 0 - 52 + -	52 + +	10 0 0 52 + +	86 + -	86 + 0	86 + +
53	98 - -	98 - 0	98 - +	57 0 - 53 + -	53 + +	9 + + 9 - + 53 + +	88 + -	88 + 0	88 + +
54	98 - -	74 - 0 54 - +	97 - +	98 0 -	54 + +	97 0 +	98 + -	9 + 0 54 + +	97 + +
55	100 - -	70 - 0 55 - -	99 - +	100 0 -	55 + -	99 0 +	100 + -	3 + 0 55 + -	99 + +

Current Location	Current Direction of Motion (ratchet wheel / pawl) - -	- 0	- +	0 -	0 0	0 +	+ -	+ 0	+ +
56	99 - -	99 - 0	99 - +	3 + - 3 - - 56 + -	56 + -	57 0 + 56 + +	102 + -	102 + 0	102 + +
57	99 - -	58 - 0	98 - +	56 0 -	57 + 0	53 0 +	102 + -	59 + 0	88 + +
58	99 - -	63 - 0 58 0 0	98 - +	99 0 -	58 + 0	98 0 +	99 + -	57 + 0	98 + +
59	102 - -	57 - 0 59 0 0	88 - +	102 0 -	59 + 0	88 0 +	102 + -	60 + 0	88 + +
60	102 - -	59 - 0	88 - +	61 0 -	60 + 0	82 0 +	101 + -	62 + 0	87 + +
61	102 - -	102 - 0	102 - +	68 0 - 61 + -	61 + -	60 0 + 61 + +	101 + -	101 + 0	101 + +
62	101 - -	60 - 0 62 0 0	87 - +	101 0 -	62 + 0	87 0 +	101 + -	63 + 0	87 + +
63	101 - -	62 - 0	87 - +	64 0 -	63 + 0	75 0 +	99 + -	58 + 0	98 + +
64	101 - -	101 - 0	101 - +	70 0 - 64 + -	64 + -	63 0 + 64 + +	99 + -	99 + 0	99 + +
65	96 - -	96 - 0	96 - +	77 0 - 65 + -	65 + -	68 0 + 65 + +	94 + -	94 + 0	94 + +
67	96 - -	3 - 0 67 - -	102 - +	96 0 -	67 + -	102 0 +	96 + -	68 + 0 67 + -	102 + +
68	96 - -	67 - 0	102 - +	65 0 -	68 + -	61 0 +	94 + -	71 + 0	101 + +
69	94 - -	94 - 0	94 - +	17 0 - 69 + -	69 + -	70 0 + 69 + +	100 + -	100 + 0	100 + +
70	94 - -	71 - 0	101 - +	69 0 -	70 + -	64 0 +	100 + -	55 + 0	99 + +
71	94 - -	68 - 0 71 - -	101 - +	94 0 -	71 + -	101 0 +	94 + -	70 + 0 71 + -	101 + +
73	87 - -	83 - 0 73 - +	86 - +	87 0 -	73 + +	86 0 +	87 + -	74 + 0 73 + +	86 + +
74	87 - -	73 - 0	86 - +	75 0 -	74 + +	43 0 +	98 + -	54 + 0	97 + +
75	87 - -	87 - 0	87 - +	63 0 - 75 + -	75 + +	74 0 + 75 + +	98 + -	98 + 0	98 + +
76	95 - -	51 - + 76 - -	96 - +	95 0 -	76 + -	96 0 +	95 + -	77 + 0 76 + -	96 + +
77	95 - -	76 - 0	96 - +	78 0 -	77 + -	65 0 +	93 + -	49 + 0	94 + +
78	95 - -	95 - 0	95 - +	42 + - 78 + -	78 + -	77 0 + 78 + +	93 + -	93 + 0	93 + +
79	92 - -	42 - + 79 - -	93 - +	92 0 -	79 + -	93 0 +	92 + -	80 + 0 79 + -	93 + +
80	92 - -	79 - 0	93 - +	81 0 -	80 + -	46 0 +	91 + -	40 + 0	90 + +
81	92 - -	92 - 0	92 - +	48 + - 81 + -	81 + -	80 0 + 81 + +	91 + -	91 + 0	91 + +

Current Location	Current Direction of Motion (ratchet wheel / pawl)								
	- -	- 0	- +	0 -	0 0	0 +	+ -	+ 0	+ +
82	88 - -	88 - 0	88 - +	60 0 - 82 + -	82 + +	83 0 + 82 + +	87 + -	87 + 0	87 + +
83	88 - -	84 - 0	89 - +	82 0 -	83 + +	52 0 +	87 + -	73 + 0	86 + +
84	88 - -	9 - 0 84 - +	89 - +	88 0 -	84 + +	89 0 +	88 + -	83 + 0 84 + +	89 + +
86	83 - - 52 - - 73 - - 86 0 - 86 0 0 86 - 0	52 - 0 86 - +	10 0 0 10 - - 52 - + 44 - + 86 0 +	73 0 - 86 + -	86 + +	44 - + 86 + +	45 + - 44 + - 74 + - 43 + - 73 + - 86 + 0	45 + - 44 + - 43 + 0 86 + +	45 + - 45 0 0 45 - + 44 + - 44 0 0 44 - + 43 + +
87	60 - - 62 - - 82 - - 87 0 - 87 0 0 87 - 0	82 - 0 87 - +	83 - + 73 - + 82 - + 87 0 +	62 0 - 87 + -	87 + +	73 0 + 87 + +	63 + - 75 + - 62 + - 87 + 0	75 + 0 87 + +	74 + + 73 + + 75 + +
88	57 - - 59 - - 53 - - 88 0 - 88 0 0 88 - 0	53 - 0 88 - +	9 - + 53 - + 84 - + 88 0 +	59 0 - 88 + -	88 + +	84 0 + 88 + +	60 + - 82 + - 59 + - 88 + 0	82 + 0 88 + +	83 + + 82 + + 84 + +
89	9 - - 84 - - 8 - - 89 0 - 89 0 0 89 - 0	8 - - 89 - +	8 + + 8 0 0 8 - - 89 0 +	84 0 - 89 + -	89 + +	8 + + 89 + +	83 + - 84 + - 52 + - 89 + 0	52 + 0 89 + +	10 + - 10 0 0 8 + + 52 + +
90	80 - - 46 - - 40 - - 90 0 -	46 - 0 90 - -	17 - + 46 - + 16 - + 90 0 + 90 0 0 90 - 0	40 0 - 90 + -	90 + -	16 - + 90 + +	18 0 - 16 + - 40 + -	16 + - 90 + -	16 + - 16 0 0 16 - + 90 + 0
91	48 + - 48 0 0 48 - + 47 + - 47 0 0 47 - + 81 - - 91 0 -	48 - + 47 - + 81 - 0 91 - -	80 - + 40 - + 48 - + 47 - + 81 - + 91 0 + 91 0 0 91 - 0	47 + - 91 + -	91 + -	40 0 + 91 + +	26 0 0 28 0 - 47 + -	28 0 0 91 + -	18 0 0 18 - + 40 + + 28 0 + 91 + 0

Current Location	Current Direction of Motion (ratchet wheel / pawl)								
	- -	- 0	- +	0 -	0 0	0 +	+ -	+ 0	+ +
92	41 + - 41 0 0 41 - + 92 0 -	41 - + 92 - -	42 - + 41 - + 79 - + 92 0 + 92 0 0 92 - 0	41 + - 92 + -	92 + -	79 0 + 92 + +	48 + - 81 + - 41 + -	81 + 0 92 + -	80 + + 81 + + 79 + + 92 + 0
93	42 + - 42 0 0 42 - + 79 - - 78 - - 93 0 -	78 - 0 93 - -	77 - + 78 - + 49 - + 93 0 + 93 0 0 93 - 0	79 0 - 93 + -	93 + -	49 0 + 93 + +	80 + - 79 + - 46 + -	46 + 0 93 + -	17 + + 49 + + 46 + + 93 + 0
94	77 - - 65 - - 49 - - 94 0 -	65 - 0 94 - -	68 - + 65 - + 71 - + 94 0 + 94 0 0 94 - 0	49 0 - 94 + -	94 + -	71 0 + 94 + +	17 + - 69 + - 49 + -	69 + 0 94 + -	70 + + 71 + + 69 + + 94 + 0
95	50 + - 50 0 0 50 - + 95 0 -	50 - + 95 - -	51 - + 50 - + 76 - + 95 0 + 95 0 0 95 - 0	50 + - 95 + -	95 + -	76 0 + 95 + +	42 + - 78 + - 50 + -	78 + 0 95 + -	77 + + 78 + + 76 + + 95 + 0
96	51 + - 51 0 0 51 - + 76 - - 27 - 0 96 0 -	27 - 0 96 - -	3 - + 27 - + 67 - + 96 0 + 96 0 0 96 - 0	51 + - 76 0 - 27 0 0 96 + -	96 + -	67 0 + 96 + +	77 + - 51 + - 76 + - 27 + 0 65 + -	65 + 0 96 + -	68 + + 67 + + 65 + + 96 + 0
97	74 - - 43 - - 54 - - 97 0 - 97 0 0 97 - 0	43 - 0 97 - +	45 - + 43 - + 11 - + 97 0 +	54 0 - 97 + -	97 + +	11 - + 97 + +	9 + - 11 + - 54 + - 97 + 0	11 + - 97 + +	11 + - 11 0 0 11 - +

Current Location	Current Direction of Motion (ratchet wheel / pawl)								
	- -	- 0	- +	0 -	0 0	0 +	+ -	+ 0	+ +
98	63 - - 58 - - 75 - - 98 0 - 98 0 0 98 - 0	75 - 0 98 - +	74 - + 54 - + 75 - + 98 0 +	58 0 - 98 + -	98 + +	54 0 + 98 + +	57 + - 53 + - 58 + - 98 + 0	53 + 0 98 + +	9 + + 54 + + 53 + +
99	70 - - 64 - - 55 - - 99 0 -	64 - 0 99 - -	63 - + 64 - + 58 - + 99 0 + 99 0 0 99 - 0	55 0 - 99 + -	99 + -	58 0 + 99 + +	3 + - 56 + - 55 + -	56 + 0 99 + -	57 + + 58 + + 56 + + 99 + 0
100	17 - - 6 - - 69 - - 100 0 -	69 - 0 100 - -	70 - + 55 - + 69 - + 100 0 + 100 0 0 100 - 0	6 - - 100 + -	100 + -	55 0 + 100 + +	6 + + 6 0 0 6 - -	6 + + 100 + -	3 + + 55 + + 6 + + 100 + 0
101	68 - - 71 - - 61 - - 101 0 -	61 - 0 101 - -	60 - + 61 - + 62 - + 101 0 + 101 0 0 101 - 0	71 0 - 101 + -	101 + -	62 0 + 101 + +	70 + - 71 + - 64 + -	64 + 0 101 + -	63 + + 62 + + 64 + + 101 + 0
102	3 - - 67 - - 56 - - 102 0 -	56 - 0 102 - -	57 - + 56 - + 59 - + 102 0 + 102 0 0 102 - 0	67 0 - 102 + -	102 + -	59 0 + 102 + +	68 + - 67 + - 61 + -	61 + 0 102 + -	60 + + 59 + + 61 + + 102 + 0

Appendix C: Skotch Yoke Transitions

Current Location	Current Direction of Motion (cam / yoke)								
	- -	- 0	- +	0 -	0 0	0 +	+ -	+ 0	+ +
3	76 - -	64 - 0	75 - + 8 - +	61 0 -	3 + 0		77 + -	65 + 0	78 + + 6 + +
6	3 - - 78 - - 6 0 -			78 0 -	6 + 0		78 + -	78 + 0	78 + + 7 + - 7 0 0
7	78 - - 6 - -			58 0 -	7 + -		75 + - 8 + -		
8	75 - -	75 - 0	7 0 0 7 - - 75 - + 8 0 0 8 - 0	75 0 -	8 + -		75 + - 3 + -		
11			73 - + 14 - +		11 + +	70 0 +			72 + + 16 + +
14			73 - + 15 - + 14 0 +		14 + 0	73 0 +	11 + + 11 0 0 73 + -	73 + 0	73 + +
15	72 - - 16 - -	54 - 0	71 - +		15 + 0	51 0 +	73 + - 14 + -	68 + 0	71 + +
16	72 - - 11 0 0 11 - + 16 0 0 16 - 0	72 - 0	72 - +		16 + +	72 0 +			15 + + 72 + +
19	80 - -	48 - 0	79 - + 24 - +	44 0 -	19 + 0		81 + -	45 + 0	82 + + 42 + +
22	43 - - 83 - - 22 0 -			83 0 -	22 + 0		83 + -	83 + 0	83 + + 23 + - 23 0 0
23	83 - - 22 - -			37 0 -	23 + -		84 + - 39 + -		
24	79 - -	79 - 0	40 - + 79 - + 24 0 0 24 - 0	79 0 -	24 + -		79 + - 19 + -		

Current Location	Current Direction of Motion (cam / yoke)								
	- -	- 0	- +	0 -	0 0	0 +	+ -	+ 0	+ +
27			80 - + 49 - +		27 + +	44 0 +			81 + + 32 + +
30			79 - + 31 - + 30 0 +		30 + 0	79 0 +	50 + - 79 + -	79 + 0	79 + +
31	82 - - 46 - -	41 - 0	83 - +		31 + 0	37 0 +	79 + - 30 + -	38 + 0	84 + +
32	81 - - 27 0 0 27 - + 32 0 0 32 - 0	81 - 0	81 - +		32 + +	81 0 +			47 + + 81 + +
33		33 0 0	76 - +		33 + 0	76 0 +		63 + 0	76 + +
35	71 - -	53 - 0 35 0 0		71 0 -	35 + 0		71 + -		
37	83 - -	83 - 0	83 - +	31 + - 31 - - 37 + -	37 + 0	23 0 0 37 + +	84 + - -	84 + 0	84 + +
38	79 - -	31 - 0 38 0 0	84 - +	79 0 -	38 + 0	84 0 +	79 + -	40 + -	84 + +
39	84 - -	84 - 0	23 0 0 23 - - 84 - + 39 0 0 39 - 0	84 0 -	39 + -		84 + - 40 + -		
40	79 - -	38 - 0	84 - + 39 - +	79 0 -	40 + -		79 + - 24 + -		
41	82 - -	43 - - 41 0 0	83 - +	82 0 -	41 + 0	83 0 +	82 + -	31 + 0	83 + +
42	19 - - 82 - - 42 0 -			82 0 -	42 + 0		82 + -	82 + 0	82 + + 43 + +
43	82 - - 42 - -			82 0 -	43 + 0		82 + -	41 + 0	83 + + 22 + +
44	80 - -	80 - 0	80 - +	27 0 0 44 + -	44 + 0	19 + + 19 - + 44 + +	81 + -	81 + 0	81 + +
45	81 - -	19 - 0 45 0 0	82 - +	81 0 -	45 + 0	82 0 +	81 + -	47 + +	82 + +

Current Location	Current Direction of Motion (cam / yoke)								
	- -	- 0	- +	0 -	0 0	0 +	+ -	+ 0	+ +
46	82 - - 47 - - 46 0 0 46 - 0	82 - 0	82 - +		46 + +	82 0 +			31 + + 82 + +
47	81 - - 32 - -	45 - 0	82 - +		47 + +	82 0 +			82 + + 46 + +
48	80 - -	50 - + 48 0 0	79 - +	80 0 -	48 + 0	79 0 +	80 + -	19 + 0	79 + +
49			80 - + 50 - + 49 0 +		49 + 0	80 0 +	27 + + 27 0 0 80 + -	80 + 0	80 + +
50			79 - + 30 - +		50 + 0	79 0 +	80 + - 49 + -	48 + 0	79 + +
51	71 - -	71 - 0	71 - +	15 + - 15 - - 51 + -	51 + 0	53 0 0 51 + +	71 + -	71 + 0	71 + +
52	71 - -	57 - 0 52 0 0		71 0 -	52 + 0		71 + -	53 + 0	
53	71 - -	52 - 0		51 0 -	53 + 0		71 + -	35 + 0	
54	72 - -	69 - 0 54 0 0	71 - +	72 0 -	54 + 0	71 0 +	72 + -	15 + 0	71 + +
55	71 - -	71 - 0	71 - +	69 0 - 55 + -	55 + 0	57 0 0 55 + +	71 + -	71 + 0	71 + +
56	71 - -	56 0 0		71 0 -	56 + 0		71 + -	57 + 0	
57	71 - -	56 - 0		55 0 -	57 + 0		71 + -	52 + 0	
58	78 - -	78 - 0	78 - +	66 0 - 58 + -	58 + 0	7 0 0 58 + +	75 + -	75 + 0	75 + +
59		60 - 0 59 0 0	76 - +		59 + 0	76 0 +			76 + +
60		62 - 0	77 - +		60 + 0	67 0 +		59 + 0	76 + +
61	76 - -	76 - 0	76 - +	63 0 0 61 + -	61 + 0	3 + + 3 - + 61 + +	77 + -	77 + 0	77 + +
62		63 - 0 62 0 0	77 - +		62 + 0	77 0 +		60 + 0	77 + +
63		33 - 0	76 - +		63 + 0	61 0 +		62 + 0	77 + +
64	76 - -	66 - 0 64 0 0	75 - +	76 0 -	64 + 0	75 0 +	76 + -	3 + 0	75 + +

Current Location	Current Direction of Motion (cam / yoke)								
	- -	- 0	- +	0 -	0 0	0 +	+ -	+ 0	+ +
65	77 - -	3 - 0 65 0 0	78 - +	77 0 -	65 + 0	78 0 +	77 + -	66 + 0	78 + +
66	77 - -	65 - 0	78 - +	67 0 -	66 + 0	58 0 +	76 + -	64 + 0	75 + +
67	77 - -	77 - 0	77 - +	60 0 0 67 + -	67 + 0	66 0 + 67 + +	76 + -	76 + 0	76 + +
68	73 - -	15 - 0 68 0 0	71 - +	73 0 -	68 + 0	71 0 +	73 + -	69 + 0	71 + +
69	73 - -	68 - 0	71 - +	70 0 -	69 + 0	55 0 +	72 + -	54 + 0	71 + +
70	73 - -	73 - 0	73 - +	11 0 0 70 + -	70 + 0	69 0 + 70 + +	72 + -	72 + 0	72 + +
71	51 - - 68 - - 15 - - 69 - - 55 - - 54 - - 71 0 -	51 - 0 55 - 0 71 0 0	56 - 0 35 - 0 51 - + 53 - 0 57 - 0 55 - + 52 - 0 71 0 +	68 0 - 54 0 - 71 + -	71 + 0	56 0 0 35 0 0 52 0 0 71 + +	55 + - 68 + - 69 + - 15 + - 51 + - 54 + -	55 + 0 51 + 0	55 + + 56 + 0 57 + 0 35 + 0 53 + 0 52 + 0 51 + +
72	11 0 0 11 - + 16 - - 70 - - 72 0 -	70 - 0 72 0 0	69 - + 54 - + 70 - + 72 0 +	16 - - 72 + -	72 + 0	54 0 + 72 + +	16 + + 16 0 0 16 - -	16 + +	15 + + 54 + + 16 + +
73	14 + - 14 0 0 14 - + 73 0 -	14 - + 73 0 0	15 - + 14 - + 68 - + 73 0 +	14 + - 73 + -	73 + 0	68 0 + 73 + +	11 + + 11 0 0 70 + - 14 + -	70 + 0	69 + + 70 + + 68 + +
75	66 - - 58 - - 64 - - 75 0 -	58 - 0 75 0 0	7 0 0 7 - - 58 - + 8 - + 75 0 +	64 0 - 75 + -	75 + 0	8 - + 75 + +	3 + - 8 + - 64 + -	8 + -	8 + - 8 0 0 8 - +
76	33 - 0 60 - 0 59 - 0 67 - - 76 0 -	67 - 0 76 0 0	66 - + 64 - + 67 - + 76 0 +	33 0 0 59 0 0 76 + -	76 + 0	64 0 + 76 + +	63 + 0 61 + - 33 + 0 59 + 0	61 + 0	3 + + 64 + + 61 + +

Current Location	Current Direction of Motion (cam / yoke)								
	- -	- 0	- +	0 -	0 0	0 +	+ -	+ 0	+ +
77	63 - 0 62 - 0 61 - - 77 0 -	61 - 0 77 0 0	3 - + 61 - + 65 - + 77 0 +	62 0 0 77 + -	77 + 0	65 0 + 77 + +	60 + 0 67 + - 62 + 0	67 + 0	66 + + 67 + + 65 + +
78	3 - - 65 - - 6 - - 78 0 -	6 - - 78 0 0	6 + + 6 0 0 6 - - 78 0 +	65 0 - 78 + -	78 + 0	6 + + 78 + +	66 + - 65 + - 58 + -	58 + 0	7 + - 7 0 0 6 + + 58 + +
79	50 + - 50 0 0 50 - + 30 + - 30 0 0 30 - + 48 - - 79 0 -	30 - + 79 0 0	31 - + 30 - + 40 - + 38 - + 24 - + 79 0 +	50 + - 30 + - 48 0 - 79 + -	79 + 0	40 - + 38 0 + 24 - + 79 + +	50 + - 30 + - 19 + - 24 + - 48 + -	24 + -	40 + - 40 0 0 40 - + 38 + + 24 + - 24 0 0 24 - +
80	49 + - 49 0 0 49 - + 80 0 -	49 - + 80 0 0	50 - + 48 - + 49 - + 80 0 +	49 + - 80 + -	80 + 0	48 0 + 80 + +	27 + + 27 0 0 44 + - 49 + -	44 + 0	19 + + 48 + + 44 + +
81	27 0 0 27 - + 32 - - 44 - - 81 0 -	44 - 0 81 0 0	19 - + 44 - + 45 - + 81 0 +	32 - - 81 + -	81 + 0	45 0 + 81 + +	32 + + 32 0 0 32 - -	32 + +	47 + + 32 + + 45 + +
82	47 - - 19 - - 45 - - 42 - - 46 - - 82 0 -	42 - - 82 0 0	43 + + 43 0 0 43 - - 42 + + 42 0 0 42 - - 41 - + 82 0 +	47 - - 45 0 - 46 - - 82 + -	82 + 0	43 + + 42 + + 41 0 + 82 + +	47 + + 47 0 0 47 - - 45 + - 46 + + 46 0 0 46 - -	46 + +	43 + + 42 + + 31 + + 41 + + 46 + +
83	43 - - 22 - - 41 - - 83 0 -	22 - - 83 0 0	22 + + 22 0 0 22 - - 83 0 +	41 0 - 83 + -	83 + 0	22 + + 83 + +	31 + - 37 + - 41 + -	37 + 0	23 + - 23 0 0 22 + + 37 + +
84	31 - - 38 - - 37 - - 84 0 -	37 - 0 84 0 0	23 0 0 23 - - 37 - + 39 - + 84 0 +	38 0 - 84 + -	84 + 0	39 - + 84 + +	40 + - 38 + - 39 + -	39 + -	39 + - 39 0 0 39 - +

Appendix D: Cam Transitions

Current Location	Current Direction of Motion (cam / follower) - -	- 0	- +	0 -	0 0	0 +	+ -	+ 0	+ +
2	82 - -	62 - 0	81 - + 16 - +	59 0 -	2 + +		85 + -	63 + 0	86 + + 11 + +
11	2 - - 86 - - 11 0 - 11 0 0			86 0 -	11 + +		86 + -	86 + 0	86 + + 12 + 0
12	86 - - 11 - -			56 0 -			84 + -	20 + 0	
16	81 - -	81 - 0	17 - 0 81 - + 16 0 0 16 - 0	81 0 -	16 - 0 16 - + 16 - - 16 + - 16 0 -		81 + - 2 + - 16 0 - 16 0 0		
17	84 - -	20 - 0		53 0 -	17 - 0 17 - - 17 + - 17 0 -		81 + - 16 + -		
20	84 - -	12 - -		84 0 -			84 + -	17 + -	
22	80 - - 35 - -	52 - 0	79 - +		22 + -	51 0 +	77 + - 38 + -	72 + 0	78 + +
35	80 - - 36 - 0 35 0 0 35 - 0	80 - 0	80 - +		35 - 0 35 - + 35 - - 35 + + 35 0 +	80 0 +			22 + + 80 + + 35 0 + 35 0 0
36		40 - 0	76 - +		36 - 0 36 - + 36 + + 36 0 +	71 0 +			80 + + 35 + +
38			77 - + 22 - + 38 0 + 38 0 0		38 + -	77 0 +	39 + 0 77 + -	77 + 0	77 + +
39			77 - + 38 - +			74 0 +		40 + 0	76 + +
40		39 - +	76 - +			76 0 +		36 + +	76 + +
41		41 0 0	83 - +		41 + 0	83 0 +		55 + 0	83 + +
43	79 - -	50 - 0 43 0 0		79 0 -	43 + 0		79 + -		

Current Location	Current Direction of Motion (cam / follower)								
	- -	- 0	- +	0 -	0 0	0 +	+ -	+ 0	+ +
45	78 - -	78 - 0	78 - +	73 0 - 45 + -	45 + -	47 0 0 45 + +	75 + -	75 + 0	75 + +
46	78 - -	46 0 0		78 0 -	46 + 0		78 + -	47 + 0	
47	78 - -	46 - 0		45 0 -	47 + 0		75 + -	49 + 0	
48	75 - -	75 - 0	75 - +	70 0 - 48 + -	48 + -	50 0 0 48 + +	79 + -	79 + 0	79 + +
49	75 - -	47 - 0 49 0 0		75 0 -	49 + 0		75 + -	50 + 0	
50	75 - -	49 - 0		48 0 -	50 + 0		79 + -	43 + 0	
51	79 - -	79 - 0	79 - +	22 + - 22 - - 51 + -	51 + -	51 + +	78 + -	78 + 0	78 + +
52	80 - -	70 - 0 52 - -	79 - +	80 0 -	52 + -	79 0 +	80 + -	22 + 0 52 + -	79 + +
53	84 - -	84 - 0	84 - +	67 0 - 53 + -	53 + +	17 0 0 53 + +	81 + -	81 + 0	81 + +
54		55 - 0 54 0 0	82 - +		54 + 0	82 0 +		61 + 0	82 + +
55		41 - 0	83 - +		55 + 0	68 0 +		54 + 0	82 + +
56	86 - -	86 - 0	86 - +	64 0 - 56 + -	56 + +	12 0 0 56 + +	84 + -	84 + 0	84 + +
57		58 - 0 57 0 0	83 - +		57 + 0	83 0 +			83 + +
58		60 - 0	85 - +		58 + 0	65 0 +		57 + 0	83 + +
59	82 - -	82 - 0	82 - +	61 0 0 59 + -	59 + +	2 + + 2 - + 59 + +	85 + -	85 + 0	85 + +
60		61 - 0 60 0 0	85 - +		60 + 0	85 0 +		58 + 0	85 + +
61		54 - 0	82 - +		61 + 0	59 0 +		60 + 0	85 + +
62	82 - -	67 - 0 62 - +	81 - +	82 0 -	62 + +	81 0 +	82 + -	2 + 0 62 + +	81 + +
63	85 - -	2 - 0 63 - +	86 - +	85 0 -	63 + +	86 0 +	85 + -	64 + 0 63 + +	86 + +
64	85 - -	63 - 0	86 - +	65 0 -	64 + +	56 0 +	83 + -	66 + 0	84 + +
65	85 - -	85 - 0	85 - +	58 0 0 65 + -	65 + +	64 0 + 65 + +	83 + -	83 + 0	83 + +
66	83 - -	64 - 0 66 - +	84 - +	83 0 -	66 + +	84 0 +	83 + -	67 + 0 66 + +	84 + +
67	83 - -	66 - 0	84 - +	68 0 -	67 + +	53 0 +	82 + -	62 + 0	81 + +

Current Location	Current Direction of Motion (cam / follower)								
	- -	- 0	- +	0 -	0 0	0 +	+ -	+ 0	+ +
68	83 - -	83 - 0	83 - +	55 0 0 68 + -	68 + +	67 0 + 68 + +	82 + -	82 + 0	82 + +
69	76 - -	73 - 0 69 - -	75 - +	76 0 -	69 + -	75 0 +	76 + -	70 + 0 69 + -	75 + +
70	76 - -	69 - 0	75 - +	71 0 -	70 + -	48 0 +	80 + -	52 + 0	79 + +
71	76 - -	76 - 0	76 - +	36 0 0 71 + -	71 + -	70 0 + 71 + +	80 + -	80 + 0	80 + +
72	77 - -	22 - 0 72 - -	78 - +	77 0 -	72 + -	78 0 +	77 + -	73 + 0 72 + -	78 + +
73	77 - -	72 - 0	78 - +	74 0 -	73 + -	45 0 +	76 + -	69 + 0	75 + +
74	77 - -	77 - 0	77 - +	39 0 0 74 + -	74 + -	73 0 + 74 + +	76 + -	76 + 0	76 + +
75	73 - - 45 - - 69 - - 75 0 -	45 - 0 75 - -	47 - 0 45 - + 49 - 0 75 0 + 75 0 0 75 - 0	69 0 - 75 + -	75 + -	49 0 0 75 + +	70 + - 48 + - 69 + -	48 + 0 75 + -	50 + 0 49 + 0 48 + + 75 + 0
76	39 0 0 39 - + 40 - 0 74 - - 76 0 -	74 - 0 76 - -	73 - + 69 - + 74 - + 76 0 + 76 0 0 76 - 0	40 0 0 76 + -	76 + -	69 0 + 76 + +	36 + + 36 0 0 71 + - 40 + 0	71 + 0 76 + -	70 + + 69 + + 71 + + 76 + 0
77	38 + - 38 0 0 38 - + 77 0 -	38 - + 77 - -	22 - + 38 - + 72 - + 77 0 + 77 0 0 77 - 0	38 + - 77 + -	77 + -	72 0 + 77 + +	39 + 0 74 + - 38 + -	74 + 0 77 + -	73 + + 74 + + 72 + + 77 + 0
78	22 - - 72 - - 51 - - 78 0 -	51 - 0 78 - -	51 - + 46 - 0 78 0 + 78 0 0 78 - 0	72 0 - 78 + -	78 + -	46 0 0 78 + +	73 + - 72 + - 45 + -	45 + 0 78 + -	47 + 0 46 + 0 45 + + 78 + 0
79	70 - - 48 - - 52 - - 79 0 -	48 - 0 79 - -	50 - 0 48 - + 43 - 0 79 0 + 79 0 0 79 - 0	52 0 - 79 + -	79 + -	43 0 0 79 + +	22 + - 51 + - 52 + -	51 + 0 79 + -	43 + 0 51 + + 79 + 0

Current Location	Current Direction of Motion (cam / follower)								
	− −	− 0	− +	0 −	0 0	0 +	+ −	+ 0	+ +
80	36 − 0 35 − − 71 − − 80 0 −	71 − 0 80 − −	70 − + 52 − + 71 − + 80 0 + 80 0 0 80 − 0	35 − − 80 + −	80 + −	52 0 + 80 + +	35 + + 35 0 0 35 − −	35 + + 80 + −	22 + + 52 + + 35 + + 80 + 0
81	67 − − 53 − − 62 − − 81 0 − 81 0 0 81 − 0	53 − 0 81 − +	17 − 0 53 − + 16 − + 81 0 +	62 0 − 81 + −	81 + +	16 − + 81 + +	2 + − 16 + − 62 + − 81 + 0	16 + − 81 + +	16 + − 16 0 0 16 − +
82	55 − 0 54 − 0 68 − − 82 0 − 82 0 0 82 − 0	68 − 0 82 − +	67 − + 62 − + 68 − + 82 0 +	54 0 0 82 + −	82 + +	62 0 + 82 + +	61 + 0 59 + − 54 + 0 82 + 0	59 + 0 82 + +	2 + + 62 + + 59 + +
83	41 − 0 58 − 0 57 − 0 65 − − 83 0 − 83 0 0 83 − 0	65 − 0 83 − +	64 − + 65 − + 66 − + 83 0 +	41 0 0 57 0 0 83 + −	83 + +	66 0 + 83 + +	55 + 0 68 + − 41 + 0 57 + 0 83 + 0	68 + 0 83 + +	67 + + 68 + + 66 + +
84	64 − − 66 − − 56 − − 84 0 − 84 0 0 84 − 0	56 − 0 84 − +	12 0 0 12 − − 56 − + 20 − 0 84 0 +	66 0 − 84 + −	84 + +	20 0 0 84 + +	67 + − 66 + − 53 + − 84 + 0	53 + 0 84 + +	17 + − 17 0 0 20 + 0 53 + +
85	61 − 0 60 − 0 59 − − 85 0 − 85 0 0 85 − 0	59 − 0 85 − +	2 − + 59 − + 63 − + 85 0 +	60 0 0 85 + −	85 + +	63 0 + 85 + +	58 + 0 65 + − 60 + 0 85 + 0	65 + 0 85 + +	64 + + 65 + + 63 + +
86	2 − − 63 − − 11 − − 86 0 − 86 0 0 86 − 0	11 − − 86 − +	11 + + 11 0 0 11 − − 86 0 +	63 0 − 86 + −	86 + +	11 + + 86 + +	64 + − 63 + − 56 + − 86 + 0	56 + 0 86 + +	12 + 0 11 + + 56 + +

Index

BIBLIOGRAPHIC DATA SHEET	1. Report No. UIUCDCS-R-88-1469	2.	3. Recipient's Accession No.

4. Title and Subtitle A QUALITATIVE APPROACH TO RIGID BODY MECHANICS	5. Report Date November 1988
	6.
7. Author(s) Paul E. Nielsen	8. Performing Organization Rept. No. R-88-1469
9. Performing Organization Name and Address Dept. of Computer Science 1304 W. Springfield Ave. Urbana, IL 61801	10. Project/Task/Work Unit No.
	11. Contract/Grant No. N00014-85-K-0225
12. Sponsoring Organization Name and Address Office of Naval Research	13. Type of Report & Period Covered Technical
	14.

15. Supplementary Notes

16. Abstracts

In order for a program to interact with the world as well as people do, we must provide it with a great deal of commonsense about the way things work. Reasoning about the geometric interactions and motions of objects is an important part of that commonsense. Some of the most complex problems we solve involve reasoning about mechanical devices, such as gears, cams, and clocks.

Qualitative mechanics is the symbolic analysis of the motions and the geometric interactions of physical objects. This thesis describes a theory for analysis of rigid body mechanisms, an important subset of qualitative mechanics problems. This theory has been implemented and tested on several mechanisms including a mechanical clock. Beginning with drawings of the parts involved we compute a discrete symbolic description showing changes in position and motion of the parts of the mechanism as well as its global behavior.

17. Key Words and Document Analysis. 17a. Descriptors

Qualitative Mechanics
Qualitative Physics
spatial reasoning
geometric modeling
artificial intelligence

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement unlimited	19. Security Class (This Report) UNCLASSIFIED	21. No. of Pages 155
	20. Security Class (This Page UNCLASSIFIED	22. Price

FORM NTIS-35 (10-70) USCOMM-DC 40329-P71