EXTENDING THE DEVICE-ORIENTED QUALITATIVE SIMULATION
METHOD TO MECHANICAL DEVICES

Pearl Pu
U-155
Computer Science Department
University of Connecticut
Storrs, CT 06268
U.S.A
(Tel. 203-486-2122)
(Email pu@carcvax.uconn.edu)

TOPIC: Principles in Qualitative Simulation

ABSTRACT: Mechanical systems, especially those which exhibit intermittent
motions, provide a good basis for the investigation of device
behaviors. A key observation is that the spatial configuration of
such systems changes periodically. So far only simple links or
conduits as in deKleer [3] have been used to model the
connection between a pair of objects in the device-oriented approach.
We offer a solution which uses a separate representational entity,
called the connection frame, to model the spatial relationships
between a pair of objects and how those relationships achieve force
or velocity propagation. The connection representation is assumed
supplied as part of the design knowledge of the mechanism, though
it could be just as readily computed by other spatial connection
determination methods. We describe a framework constructed to
simulate the behaviors of both ordinary and intermittent mechanical
systems, with an emphasis on force and velocity propagation reasoning.
In general, it appears that continuous motion can usually be
modeled by velocity propagation while intermittent motion is best
approached by force propagation.

STATUS: Research

DOMAIN: Mechanical Devices

Language: CommonLisp, Flavors

EFFORT: Part of a PH.D Thesis
Extending the Device-Oriented Qualitative Simulation to Mechanical Devices

Pearl Pu
University of Connecticut

ABSTRACT

Mechanical systems, especially those which exhibit intermittent motions, provide a good basis for the investigation of device behaviors. A key observation is that the spatial configuration of such systems changes periodically. So far only simple links or conduits as in deKleer [3] have been used to model the connection between a pair of objects in the device-oriented approach. We offer a solution which uses a separate representational entity, called the connection frame, to model the spatial relationships between a pair of objects and how those relationships achieve force or velocity propagation. The connection representation is assumed supplied as part of the design knowledge of the mechanism, though it could be just as readily computed by other spatial connection determination methods. We describe a framework constructed to simulate the behaviors of both ordinary and intermittent mechanical systems, with an emphasis on force and velocity propagation reasoning. In general, it appears that continuous motion can usually be modeled by velocity propagation while intermittent motion is best approached by force propagation.

1. Introduction

Besides offering a good basis for the study of behavior reasoning issues, intermittent mechanical devices are also important in characterizing systems such as clutches, brakes, clocks, etc. According to [1], the behaviors of these systems have been rather difficult to characterize by mathematical approaches, such as differential equations. Traditionally, designers have been using the graph method. However, the graph method does not provide a causal account for the behavior reasoning of such systems. Recent successes in qualitative physics [3,6,8,2] have shown that in order to facilitate the common sense reasoning of physical systems, a more direct and causal notion of behavior reasoning is called for.

After examining the various approaches in qualitative physics, we found that the device-oriented ontology by deKleer [3] is most suitable for the behavior reasoning of mechanical systems (see Pu [10] for detailed discussion). However, we observe that the spatial configuration of intermittent systems changes periodically. Thus, instead of using a simple link or a conduit as in deKleer [3] to model the connection between a pair of object, a separate representation entity, called the connection frame, is used. Such a frame is robust enough to accommodate for the changeable spatial configuration of intermittent systems. Furthermore, our constraint modeling is also different from deKleer's. We provide a systematic way for the users to establish the constraints from physical models.

Our representation strategy is highly object-oriented. A system is represented by a graph of nodes. Nodes belong either to the component or to the connection class. Message passing is used to model the flow of force or velocity inside the system. Detailed discussion is describe in a later section.

The next section gives a more detailed overview of the problem of qualitative simulation of intermittent systems. Section 3 characterizes the domain of mechanical systems and their motions, suggesting much on the choice of our representation and reasoning strategies. Section 4 outlines the modeling primitives and discusses their rationals. Section 5 describes a simulation system that produces behaviors of intermittent systems. Examples are used to illustrate the strength of the representation technique.

Figure 1: The spring-driven cam mechanism

1 The terms, object-oriented and device-oriented, are used interchangeably.
2. The Problem

We are interested in a qualitative description of how intermittent devices behave. Illustrated in Figure 1 is an intermittent device called the spring-driven ratchet mechanism.

To generate a qualitative simulation is to produce a description at a level of understanding close to human’s. A human is likely to describe the way this ratchet mechanism works in the following way according to Bickford [1]:

In the first stage (A), only the drive cam is in motion and the net torque on the wheel is zero. In the second stage (B), the cam has released the arm and the drive spring has moved the arm forward to contact the wheel, thus dumping all the stored energy into the wheel. This creates a drive torque that causes the wheel to rotate. In the third stage (C), a so-called non-overthrow tooth on the ratchet arm has engaged the inner row of teeth on the ratchet, exerting a stopping force on the wheel. The net torque is negative now and the wheel comes to a stop. In the fourth stage (D), the cam comes into contact with the arm, causing the arm to retract. The net torque on the wheel is zero.

Notice how the author has chosen to describe the behavior of the mechanism by going through the behavior of the constituent objects in an object-oriented manner. Notice the motion propagations in the description. Notice also how the description is broken up into four phases, with each phase corresponding to a particular spatial configuration of the mechanism.

Let us now pose the behavior reasoning problem of a device with respect to the spatial configuration of that device:

How to explicitly specify the different phases of the spatial configuration of an intermittent device, and how to construct a simulation algorithm to model the effect of the spatial configuration on motion propagation?

3. Domain Definition and Characterization

The basic physics of some mechanical systems is reviewed here. More importantly, we present two major observations that have guided us to the strategy of using highly object-oriented approach and message passing techniques to construct a general frame work for performing qualitative simulation.

3.1 The basics

According to Schwamb [11], a mechanism is a combination of rigid bodies so arranged that the motion of one compels the motion of the others. This suggests that a mechanism should be modeled as a graph of nodes, with each node representing a constituent part. In discussions that follow, we will use the terms “a constituent part”, “a component”, “a body”, “an individual object” interchangeably to mean a class of entities: the building blocks of a mechanism.

In [7], the modeling of a mechanical system is also deduced from the constituent parts. They are called subsystems, and are divided into the following categories:

- 1-port subsystems — they connect to the rest of the system through one port. There are the following kinds of 1-port subsystems:
  - Resistors (e.g., dampers or dashpots) — they dissipate energy.
  - Capacitors (e.g., a spring) — they store and release potential energy without loss in the ideal case.
  - Inertia subsystems (e.g., a mass with a velocity) — they store and release another kind of energy, kinetic energy.

- 2-port subsystems — they connect to the rest of the system through two ports. Force and velocity flow into the subsystem from one port and leave through the other. In the ideal case, power is conserved. That is, the product of incoming force and velocity is equal to the product of outgoing force and velocity. The lever system and the tooth-meshing two gear system are good examples.

- 3-port subsystems — they are further divided into 0-junction and 1-junction subsystems. In the 0-junction case, the forces from all of the three ports are equal, but the three velocities sum to zero. In the 1-junction case, the three velocities are equal, but the three forces sum to zero. In both cases, power is conserved. The 3-port subsystems do not necessarily concern three objects. In fact in the 0-junction case, the subsystem concerns two objects, while in the 1-junction case, it concerns one object with three forces.

While the 1-port subsystems are equivalent to the components of a mechanism, both 2-port and 3-port subsystems correspond to connections between pairs of objects. In our object-oriented paradigm, this characterization has made the hierarchical representation of physical entities very easy. In a scheme called the object hierarchy, we represent individual objects in the following tree structure: at the topmost level, all objects are three dimensional objects; at the second level, objects are either movable or stationary; under the movable objects, objects are either resistors, capacitors, or inertial capacitors; at the very bottom level, we have generic objects such as gears, ratchets, etc. The connection hierarchy has an abstract connection type at the top. At the second level, it has intermittent and regular (or continuous) connections. At the third level, further distinctions among the different types of connections are made.

3.2 Motion flow

We view a device’s behavior as some kind of propagation of flow throughout the device — as the flow travels, it brings motion to individual objects. But our notion of flow so far is still an abstract concept. What is flow?

To study how humans describe such motion flows during the course of explaining the behavior of a device, we did some survey. We found that in Bickford [1], for example, descriptions of
"force imparting from one body to another" and "force transferring from one body to another" are used frequently to indicate motion propagations in a device. However, forces are not always convenient to model motion propagations. Sometimes people switch to velocity propagations. In the case of the tooth-meshing gear train, the propagation of motion from the first gear to the second is best modeled by angular velocity flow. We have finally decided to use force and velocity flow to represent motion flow.

In our qualitative device behavior reasoning framework, we model those "imparting" and "transferring" forces and velocities in terms of a type of flowing objects called packets. Packets can travel from one object to the other in a device. Inside each packet, forces and velocities are qualitative variables and each has a magnitude, and a direction. 4

In section 5, we describe our qualitative simulation technique called the force-velocity propagation method. The simulator usually uses only one notion (either force or velocity propagation) at a time. There we will discuss when we use force propagation and when velocity propagation.

3.3 The cause, enablement, and effect of motion

For any object to move, it must receive some kind of sources. We call those sources the causes of the motion. For an isolated object, such sources must be applied externally. For an object, which is placed in a topology imposed by the geometry of a device, such sources can propagate throughout the topology and finally reach the object through the neighboring objects. Sometimes, however, an object can be prevented (purposely or accidentally) from moving by geometrical constraints, frictional forces, etc. We call these conditions the enablement/disablement of motion. Finally, if an object is allowed to move in a certain direction, and has the source to move in that direction, the object will exhibit some kind of behavior. We call the object's behavior the effect of motion. These three notions — cause, enablement/disablement, and effect — together define the causality of motion in our framework.

The definition of motion is therefore: if and only if an object has the cause and enablement for motion, can an object move. Following from this definition, the motion flow0 of a device is the propagation of motion from some external sources. Such a flow, however, can be prevented (temporarily, intermittently, etc) by such things as frictions and geometrical constraints.

For our particular domain, there are three forms of causes for an object:

- a force source;
- a velocity source;
- a source resulting from structural changes (e.g., the stretching and compressing of a spring).

Motion disablement arises from the following sources:

- frictions;
- geometrical constraints (e.g., a wall blocks an object from moving in a certain direction);
- forces of opposite directions (this is a more active form to prevent an object from moving in a certain direction).

As we will show later, we implement the notion of motion enablement/disablement by the inclusion of a degree-of-freedom field in a frame which describes an object. This field can be manipulated to reflect such fact as whether or not an object is blocked by another object.

3.4 Intermittentness

In the beginning, we observed that intermittent systems differ from regular systems in that the configuration of the former goes through phase changes. There are other characteristics of intermittent systems.

From the motion flow propagation point of view, intermittent devices have an interesting way to purposely achieve the propagation or prevention of motion flow. They do so by the manipulation of the spatial configurations of the devices. In the spring-driven ratchet mechanism, when the cam is in contact of the arm, this configuration prevents the spring from pushing the arm forward. When the cam leaves the arm, the spring starts to take effect, causing the arm to bump into the wheel. How can we characterize all of this?

Let us define the connection where two objects' spatial configuration goes through phase changes as an intermittent point. The cam and the arm form one intermittent point. The arm and the wheel form another. In the case of cam-and-arm, the physical contact between the two objects forms a geometrical constraint which prevents a force source from taking effect. Only when the objects start to move away from each other, such a force source can take effect. Thus, depending on the particular phase an intermittent point is in, it can do the following:

- Posing geometrical constraints (e.g., when the cam and the arm are in contact, the cam prevents the arm from being pushed by the spring);
- Letting go of geometrical constraints (e.g., when the cam

---

4The quantity space is used here for all of our qualitative variables. See [6] for the definition of quantity space and [ ].

*When the term motion flow is used, we mean either force or velocity flow.*
and the arm are not in contact, the arm is free to move counter-clockwise;  
• Providing a cause for motion (e.g., when the arm suddenly contacts the wheel, it provides a torque to the wheel).

In the past, only links and conduits have been used to model the connectivity of two objects. Those representational entities, however, are not robust enough to accommodate the spatial reasoning between pairs of objects of both ordinary and intermittent devices. We offer a representational primitive, called the connection frame, to describe both the geometrical and spatial relationships of pairs of objects, and how those relationships affect motion propagation. We discuss how we do that in section 4.3.

3.5 The dual functionality of an object

An object, viewed in isolation from a particular system, can be used to achieve several functions. Using the tooth-meshing gear train as an example, a gear can be a power receiver that receives external force or velocity from a motor through a shaft and supplies power to the rest of the system, or a gear can be a transporting object, merely transporting power. The distinction of the functionalities depends on the topology of the mechanism and where the gear is used. Following the "no-function-in-structure" principle in [3], we should not make such a distinction in the generic model of a gear. Thus in the generic model of an object, we implement both functionalities — an object can be a power source receiver that supplies power to the rest of the system, or as a power transporter. As we will show later, in the case of a force receiver, confluences are used to reason about the behavior of the object, and in the case of a transporter, velocity transferring rules are used.

4 Modeling Primitives

4.1 The general strategy

We take a highly object-oriented approach to model the structures of a mechanical device. The simulation strategy follows directly from the object-oriented ontology — force and velocity packets are used to propagate behavior. Thus a system is a topology of component and connection nodes. When there is a force packet acting on a component, the component goes through state changes according to its confluence models. When there is a velocity packet acting on a component, the component passes it to the next object using something called velocity transferring rules. The connection between a pair of objects deduces behavior propagation from the spatial relationship of the pair. In this section, we outline the format of both component and connection frames and discuss the relevance of the format with respect to behavior propagation.

4.2 The component frames

The frame to model the structure and local behaviors of an individual object has the following slots:

• Degree-of-freedom
• Physical variables
• Velocity transferring rules
• Confluence rules

The degree-of-freedom slot specifies explicitly whether or not an object can move in a certain direction. Since we are dealing with mechanisms, an object usually undergoes guided motion, and thus has few degrees of freedom. According to Paul's classification in [9], there are the following possible degrees of freedom in planar mechanisms:

• Revolute (e.g., a gear is mounted on a shaft and is allowed to rotate on the shaft) — a rotational freedom in both clock and counter-clockwise directions around an axis.
• Slide (e.g., a linkage mounted on a fixture and is allowed to slide along that fixture) — a translational freedom in both positive and negative directions along an axis.

In intermittent mechanisms, however, the motion of some objects is often purposely constrained. Such constraints are geometrical constraints imposed by the contact of the object itself and its neighboring objects. To be able to reason about the behavioral propagation of intermittent objects under those geometrical constraints, we must be able to know when such constraints arise and how the constraints affect behavior propagation. We discuss the reasoning of geometrical constraints in section 4.3. However, we must show here how such constraints affect behavioral propagation. The degree-of-freedom slot in the component node is used to reflect the presence or absence of geometrical constraints imposed on an object. Thus for an object which is continuously connected to its neighbors, the degree-of-freedom slot is a constant. For an object with intermittently connected neighbors, this slot changes. Such changes depend on a certain type of message coming from the connection node where geometrical constraints are determined. The arm of the spring-driven ratchet device, for example, periodically receives a message from the cam-arm connection node and changes its degree-of-freedom slot to enable or disable the arm's motion tendency in a certain direction, depending on the message.

The behavior rules take the degree-of-freedom as their pre-conditions. That is, for an object to move in a certain direction, the degree-of-freedom in that direction has to be satisfied. Following our dual functionality view of an object, we use two kinds of rules to reason about the behavior propagation within an object — velocity propagation rules if the object is a transporter, and confluence rules if the object is an external force receiver.

The confluence equations of an object are due to deKleer [3]. Confluences are qualitative differential equations which govern the dynamic behaviors of an object.

---

*Nodes and frames are used interchangeably.
*By local behavior, we mean the reason of the behaviors of an object as viewed in isolation.
*Motion in both directions are allowed, since objects often undergo reciprocating motions, for instance, swinging.
4.3 The connection frames

In a mechanism, there are two different types of connections that could exist between a pair of objects with respect to spatial contact. The continuous connection between a pair of objects reflects the constant nature of the spatial relationship of that pair. In intermittent connections, however, spatial relationships are transient and go through phase changes.

In a continuous connection, we explicitly specify the geometrical constraints of the connection and their effect on motion propagation. Thus a continuous connection frame has the following slots:

- Geometrical constraints (e.g., gear-meshing, belt-coupling, cross-belts-coupling, rigid-bar-connecting, etc.) — they are preconditions, and for our simulator to infer the consequent behavior propagation, these conditions have to be satisfied.

- Behavior propagation rules — the rules specify how motion of an object will be transferred to another object. For the tooth-meshing gear train, the velocity transferring rule propagates the ratios of the angular speed as specified by this equation \( \omega_2 = (t_1/t_2) \cdot \omega_1 \) where \( \omega_1 \) and \( \omega_2 \) are the angular velocities of the first and second gears, and \( t_1 \) and \( t_2 \) are the teeth counts of the first and second gears respectively.

In an intermittent connection, we are interested in knowing when the connected objects are spatially connected, and when they are not. Right now the contact detection is assumed supplied as part of the design knowledge of the mechanism. Ideally the contact detection information should be acquired by a geometric program which computes the contact of two objects using the configuration space technique. [4,5] provides a good example.

The connection frame looks like the following:

- Spatial configurations — they are the different spatial configurations of an intermittent connection.

- Transferring Rules — they are rules to propagate motion between the two objects. Because we divide an intermittent connection into phases, we are able to specify different motion propagation rules based on the different phases. There is one rule per every phase.

We divide the spatial relationship between an intermittent pair of objects into the following phases:

- **Come into contact** — the phase where two objects are moving toward each other and eventually form a contact. Impact forces of the two objects need to be analysed in order to infer the behavior of the two objects.

- **In contact** — the phase where two objects stay in contact for a fixed period of time. Under this circumstance, the connected two bodies can either undergo motion together, or form an interlock. In the case of joined motion, the connection node sends out a force packet. In the case of an interlock, the node sends out a message that will disable the effected object from moving in a certain direction.

- **Leaving a contact** — the phase where the two bodies are moving away from each other. The output message enables the next object to move in a certain direction if the intercontact was a interlocking one.

- **Out of contact** — the phase where the two bodies are not in contact.

To achieve the reasoning of how the spatial relationship between intermittent objects effects motion propagation, the transferring rules can send out messages which can alter the enablement/disablement of motion of an object. For example, when a particular phase poses a geometrical constraint to an object, the transferring rule will send a message which will disable the degree-of-freedom of that object in a certain direction. That object will thus unable to move even when there is a motion source.

5 Simulating Behaviors

5.1 The simulation algorithm

The simulation of mechanical mechanisms proceeds in two dimensions, although not simultaneously. Illustrated in Figure 3 is a graphical representation of the simulation of a pair of tooth-meshing gears. Starting from all of the components that receive external forces or velocities, the propagation of behavior among the objects in a mechanism defines the object-to-object dimension for the simulation. The rules defined in each of the component and connection frames are responsible for guiding the simulator to traverse the topology of a mechanism. 10

Objects, upon receiving external force/velocity, will go through state transitions. The state transitions within an object define the state-to-state dimension for the simulation. The reasoning of state transitions is done by the use of confluence equations and velocity propagation rules. Notice, however, in simulating the tooth-meshing gears in Figure 3, one needs to invoke the confluence equations for gear1 only. Our force/velocity propagation can determine the angular velocity for gear2 directly based upon the transformer connection.

![](image)

Figure 3: The simulation of a pair of tooth-meshing gears

---

*Though the angular velocities are not real values, the teeth counts of each gear are real numbers. The symbolic ratio reasoning has enabled us to detect inconsistent mechanisms [10].*
The simulation relies on the force-velocity propagation method in deciding when to use force and when to use velocity to propagate. Details of f-v method can be found in [10]. In general, it appears that continuous motion can usually be modeled by velocity propagation while intermittent motion is best approached by force propagation.

The other major part of the simulation is message passing. Besides force and velocity packets, we have the following messages:

- Degree-of-freedom packets — they are intended to modify motion enablement or disablement of some components.
- Forward messages — they are used to propose state changes to the neighboring components. This way, we can reason about physically inconceivable mechanisms such as the three tooth-meshing gear train that forms an interlock.
- Backward messages — they are used to confirm state changes proposed by the forward messages. In the three tooth-meshing gear train example, the third gear will send a message that denies the other two gears from changing states or undergoing motion.

Now let's show the simulation algorithm:

1. Starting with an initial state of the device, and those objects that receive external forces, the simulator gets an object NEXT from a selector. 11
2. If this object has multiple neighbors and the incoming force or velocity messages are not consistent, the object will abort the simulation. Otherwise, simulate NEXT by invoking one of its rules. The invoked rule sends out either a force or velocity message.
3. The selector selects a connection node NEXT-CONNECT.
4. Simulate NEXT-CONNECT by invoking one of its rules. The invoked rule sends out a message.
5. If there are still unmarked nodes, the selector selects an object NEXT. Go to 2.
6. If all objects and connections are marked, then confirm all messages. At this point, all objects can change state if necessary.
7. Go back to 1, but with a different state of the system.

6 Conclusion

We have shown a device-oriented framework for the behavior reasoning of both ordinary and intermittent mechanical systems. Our framework is based on deKleer's device-oriented ontology. In addition, we have extended his theory to the domain of mechanical devices which require more modeling primitives. An important modeling primitive, the connection frame, provides robust expressive power to accommodate for the changeable spatial relationship of intermittent systems. The behavior of a device as a whole is deduced from the structure and local behaviors of the device's constituent parts. Within a component frame, rules are readily used to guide the simulator to infer either state changes if there is an input force, or velocity transfer. Within a connection frame, rules are used for the reasoning of behavior propagation from one object to the other. The force-velocity propagation method provides a direct causal account for behavior simulation.

Messages of several kinds carry information and travel inside a device, much like the way a human would describe the behavior propagation of the device.

The work described is being implemented in Flavors and Common Lisp on the Symbolics Lisp machines. There are a dozen of objects and several intermittent mechanisms and their behaviors that our system can represent and simulate. They include several gear trains, the spring-driven cam mechanism, and a mechanical clock.

Appendix: Examples

We show some example objects and a connection written in our framework. The syntax of our system is discussed in detail in [10].

Node id: gear
Degree-of-freedom:
free(gear, clockwise) supposes the gear is facing out; so is the x-axis free(gear, counter_clockwise)
State-variables: f, v, teeth
1. [x] external force w.r.t. angular velocity
2. [x] teeth count of the gear
Velocity transferring rules:
Incoming(v) = outgoing(v)
Incoming(sgn(v)) = outgoing(sgn(v))
Confluence Rules:
if f=0 and v=0 then v stays as +
if f>0 and v<0 then v stays as 0
if f<0 and v>0 then v becomes 0
if f<0 and v<0 then v becomes 0
if f>0 and v>0 then v becomes +
if f>0 and v<0 then v becomes -
Node id: gear-meshing
Geometric constraint precondition:
mode(connected, continuous)
gear-meshing(gear1, gear2)
state( teeth(gear1), teeth(gear2))
Transferring rules:
if sign(velocity(gear1)) = +, then sign(velocity(gear2)) = +
if sign(velocity(gear1)) = -, then sign(velocity(gear2)) = -
if sign(velocity(gear1)) = 0, then sign(velocity(gear2)) = 0
% above rules take care of sign velocity(gear2) = (teeth(gear1)/teeth(gear2)) * velocity(gear1)
% this rule takes care of ratio propagation of the angular velocity
Node id: spring
Degree-of-freedom:
free(spring, x_plus)
free(spring, y_minus)
State-variables v, x, f
1. linear velocity, unstretched or compressed
2. output force resulting from stretching
3. or compressing
Confluence Rules:
if v<0, and v=0 then y stays 0, x stays 0
if v>0, and v=0 then x stays 0, y stays 0
if v<0, and v>0 then x stays 0, y stays +
if v<0, and v=0 then y becomes 0, x becomes 0
if v>0, and v<0 then x becomes 0, y becomes 0
if v>0, and v=0 then x becomes 0, y becomes 0

10Due to the space limit, the simulator's control algorithm is not discussed here. We do take care of the synchronisation of parallel motions, as it was discussed in [10].
11The selector is basically a control unit that takes care of sequencing and synchronisation problems. It is supported by Petri-Net specifications [10].
Node id: (cm, arm)
Special configurations:
mode=connection, intermittent
in-contact
leave-contact
out-of-contact

Transferring rules:
I Notice the transitions from a phase to another is not necessarily
the same for all connections.

if phase=come-into-contact(cm, arm)
then next-phase=leave-contact,
force(cm)=clockwise I can impart some force to the
arm initially

if phase=in-contact(cm, arm)
then next-phase=leave-contact,
not-free(cm, counter_clockwise)

if phase=leave-contact(cm, arm)
then next-phase=out-of-contact,
free(cm, counter_clockwise)

if phase=out-of-contact(cm, arm)
then next-phase=come-into-contact
free(cm, counter_clockwise)

References

[2] Randall Davis. Diagnostic reasoning based on structure

[3] Johan deKleer. A qualitative physics based on conflu-

In *Proceedings Tenth International Joint Conference on*
*Artificial Intelligence,* August 1987.

[5] Boi Faltings. *Qualitative Place Vocabularies for*
*Mechanisms in Configuration Space.* Technical Re-
port UIUCDCS-R-87-1360, University of Illinois at
Urbana-Champaign, 1987.

*Intelligence,* 24, 1984.

[7] Dean Karnopp and Ronald Rosenberg. *System Dynam-
ics: A Unified Approach.* John Wiley & Sons, New
York, 1975.

[8] Benjamin Kuipers. Commonsense reasoning about
causality: deriving behavior from structure. *Artificial*
*Intelligence,* 24, 1984.

[9] Burton Paul. *Kinematics and Dynamics of Planar Ma-

[10] Pearl Pu. *Qualitative Simulation of Ordinary and In-
termittent Mechanical Mechanisms.* Technical Report,
Ph.D thesis, Computer Science Department, University of
Pennsylvania, Forthcoming.