Understanding Linkages

Howard E. Shrobe
Symbolics, Inc. and MIT AI Lab.
hes@zermatt.lcs.mit.edu

Abstract

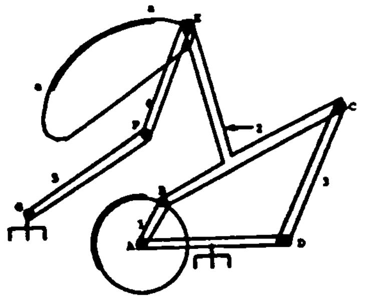
Mechanical linkages are used to transmit and transform motion. In this paper we investigate what it might mean to "understand" a linkage, i.e. how one would explain the functioning of the system. We present a system capable of understanding a variety of relatively simple linkage mechanisms found in standard references.

Our system extracts its understanding by analyzing the results of a numerical simulation of the mechanism. It proceeds through several stages: The simulator builds a trace of its reasoning which is parsed and analyzed, leading to a structuring of the mechanism into driving and driven components. The trajectories of the coupling points are then analyzed to find interesting qualitative features, as are the curves representing the histories of angular deflections of rocker arms. Next the system looks for symbolic relationships between the features and conjectures a causal relationship between them. Finally, this causal relationship is verified by geometric reasoning. This process produces explanations very much like those in standard texts.

1 Motivation

Mechanical linkages are used to transmit and transform motion. In this paper we investigate what it might mean to "understand" a linkage, i.e. how one would explain the functioning of the system. We constrain ourselves to simple mechanisms with a single degree of freedom. Figure 1[1] shows a "dwell mechanism"[2] with its explanation reproduced from [1]. (We have highlighted parts of this explanation). This paper presents a system which can "understand" this linkage.

For those not familiar with linkages, we note that the set of links 1,2, and 3 together with the fixed frame, is a "four-bar linkage" (with joints A,B,C and D) and that the pair of links 4 and 5 (with joints F and G) is a "dyad". Link 2 is the "coupler" of the four-bar linkage; since point E is on the coupler, the curve it traces is called a "coupler curve". Four bar linkages are extremely flexible driving mechanisms; they can create a large number of coupler curves exhibiting a broad variety of shapes. The shape of the curve is a function of the (relative) sizes of the links and the position on the coupler link used to trace the curve; [6] catalogues several thousand coupler curves for rocker crank mechanisms (a specific type of four bar linkage).



The lengths of the links comply with the conditions: BC = 2 AB, DC = 5.2AC, EC = 3.6AB, EF = 3.6AB, GF = 11.4AB, AD = 6AB, GD = 8.4AB and AG = 11AB. Link 4 is connected by turning pairs E and F to link 2 of *four-bar linkage ABCD* and to link 5 which oscillates about fixed axis G. When point B of crank 1 travels along the part of the circle indicated by a heavy continuous line, point E of connecting rod 2 *describes a path of which portion a-a approximates a circular arc* of radius FE with its centre at point F. During this period link 5 almost ceases to oscillate, i.e. *it practically has a dwell.*

Figure 1: A Dwell Mechanism and Its Explanation

Several observations about the explanation of figure 1 are worth emphasizing:

- The explanation is reductionist. The behavior of the whole is derived from a decomposition of the mechanism into modules and from the behaviors of the modules: the device consists of "four-bar linkage ABCD" driving the pair of links 4 and 5.
- A crucial component of the explanation is a characterization of the shape of the curve traced by point E of link 2. The characterization is in terms of qualitative features: "point E ... traces a path of which portion a-a approximates a circular arc of radius FE with its center at point F".
- The explanation does not emphasize the sort of local causal propagations made popular in [2; 3; 10; 11].
- To the extent that causation enters into the explanation it occurs at a relatively high level of abstraction:

[1] In this picture, the links are draw as bars, except that link 2 has a long finger projecting from it to point E making it look link an inverted T. Circles are used to indicate the joints between the links. The "ground" symbols are used to indicate that link AB is rigidly connected to the fixed frame and that joint G connects link 5 to the fixed frame.

[2] A dwell mechanism is one in which some part moves (in this case oscillates) most of the time, but for some period of time stands still (i.e. dwells).

"the shape of the coupler curve causes the dyad to have a dwell".

- The reductionist style of explanation stops at a higher level than is common in many other engineering domains where it is normal to reduce all behavior to the that of the modelling primitives. Here the modelling primitives are links and joint, yet the author does not attempt to provide a mechanistic explanation of how the shape of the coupler curve is related to the sizes of the links of four bar linkage ABCD.

The system described in this paper is capable of producing explanations of the sort found in figure 1 . Our approach is as follows:

- We numerically simulate a completely described linkage (i.e. one in which the lengths of all the links are known). The simulator is driven by geometric constraints.
- From this simulation we extract a "mechanism graph" a record of the local inferences of the geometric constraint engine which shows how motion propagates from link to link through the joints of the mechanism.
- The mechanism graph is "parsed" into a more structured form which decomposes the system into driving and driven modules. To the extent possible the parsed graph consists of standard building blocks (e.g. four-bar linkages, dyads). Each standard building block has a set of parameters known to be important to its behavior. The parsed graph also identifies as important the coupling points between the driving and driven modules.
- A complete simulation of the linkage is run, stepping the mechanism through its full range of positions (in our example this amounts to spinning link 1 through a full 360 degrees and for each step calculating the positions and orientations of all the remaining components). During this run, the trajectories of the coupling points and the history of values of other interesting parameters are recorded.
- The shapes of the captured curves are analyzed and qualitative features extracted.
- Finally, qualitative relationships between these features are derived and accounted for by geometric reasoning.

Section 2 describes the simulator and how it is extended to support the rest of this process. Section 3 then examines the process of mechanism extraction and section 4 describes curve characterization. Section 5 show how these facilities work together to constuct an explanation of the mechanism. Section 6 discusses to what degree the interpretation produced is an adequate "understanding" of the mechanism. Finally, in section 7 we compare our work with other work on understanding mechanisms; in particular, we will contrast this work with "qualitative kinematics".

2 The Simulator

Our simulator is based on Kramer's TLA [9] . However, since our work (at least for now) only involves planar mechanisms we have simplified TLA to a 2-D simulator (hence we call our simulator the Planar Linkage Assembler, or PLA).

2.1 Basic Object Types

The simulator is at its core a geometric constraint engine. This engine reasons about the following physical objects:

- **Links**: These are rigid bodies connected by joints. In PLA, all links are assumed to be aligned in parallel planes. Each link has its own local coordinate system. Each link also has a transformation matrix mapping its coordinate system into the global coordinate system. (We will often refer to the global coordinate system as the "fixed frame").
- **Joints**: A joint connects two links. PLA currently handles the following joint types (show in figure 2):
  1. **Revolute**: The two links are connected at a single point; they rotate relative to each other about this point. A hinge is a familiar example. These are sometimes called "turning pairs".
  2. **Pin in Slot**: A round "finger" from the first link slides in a guide path in the second link. The first link can translate along the direction of the slot; it can rotate relative to the second link as well. The guide track of a folding door is an example.
  3. **Prismatic**: The first link slides along the second link, but is not free to rotate relative to it. A piston in its cylinder is a familiar example.

The computational model has data structures representing links and joints. It reduces their behavior to the following computational constructs:

- **Markers**: Each marker is associated with a specific link. A marker has two components specified in the local coordinate system of its link: a point and an orientation. A marker can be thought of as a line extending from the point in the direction specified by the orientation. The marker also maintains its global position and orientation.
- **Constraints**: Constraints are the mechanism used to build a computational model of Joints. Each joint is modelled as a bundle of constraints. A constraint is imposed between two links by relating two markers, one from each link. PLA has three constraint types:
  1. **Coincident**: The two markers are forced to be at the same location in the global coordinate system. A revolute joint is modelled as a single coincident constraint.
  2. **Inline**: The location of the first marker is on the line described by the second marker. A Pin-In-Slot joint is modelled by a single Inline constraint.
  3. **Cooriented**: The two markers' orientations are forced to be the same in the global coordinate system. A prismatic joint is modelled as a combination of a Cooriented and an Inline constraint.
- **Anchors**: An anchor is a distinguished type of marker which is attached to the global coordinate system rather than to a link. In building the model, one often needs to position or orient a link relative to the global coordinate system. This is done by creating

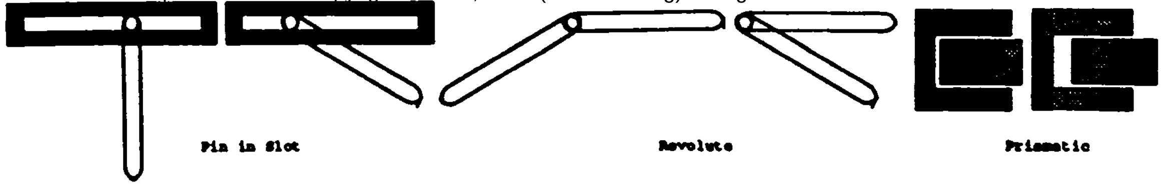


Figure 2: The Joints Modelled in PLA

an anchor and then imposing a constraint between the anchor and a marker on the link. A cooriented constraint can be used to fix the link's orientation; a combination of a cooriented and coincident constraint can be used to fix both the position and orientation of the link. The position or orientation of a driving link of the model (e.g. the orientation of link 1 in figure 1) is controlled by imposing a constraint between a marker on the link and an anchor. The simulator sets the value of the anchor's global orientation and then the constraint engine propagates this to the marker on the driving link.

2.2 The Constraint Engine

As in Kramer's TLA, the constraint engine solves the geometric constaints using local geometric techniques as much as possible. These tecniques take the form of "constraint" and "locus intersection" methods (described below). As the constraint engine runs, it monitors the degrees of freedom remaining to each link; it also records for each marker whether its global position and orientation is fixed. As the links' degrees of freedom are reduced and the markers' orientation and position become fixed, the methods are triggered. Each method moves or rotates a link to satisfy a constraint, further reducing the degrees of freedom available to the link and fixing the global position and orientation of some markers, enabling some other method to run. The process terminates when all degrees of freedom are removed.

The **constraint methods** are rule-like operators which are triggered by a pattern consisting of the type of the constraint, the presence of a fixed marker on one of the links and the degrees of translational and rotational freedom remaining to the other link. When a constraint method is triggered, it translates or rotates (or both) the link to satisfy the constraint. In doing so it reduces the degrees of freedom available to the link and it notes that a marker's orientation or position has become fixed. When a link is reduced to 0 degrees of rotational freedom, every marker on it becomes rotationally invariant in the global coordinate system; when a link is reduced to 0 degrees of both rotational and translational freedom, every marker on the link also becomes positionally invariant in the global coordinate system.

Figure 3 shows a constraint method for the coincident constraint used to model a revolute joint.

Locus intersection methods are used after the constraint methods. When a link's degrees of freedom have been sufficiently reduced, the markers on the link are constrained to move in simple curves. For example, if a link

```
If    There is a coincident constraint
              between Marker-1 and Marker-2
      Marker-1 has invariant global position
      Link-2 has 2 degrees of translational freedom
Then  Measure the vector from Marker-2 to Marker-1
      Translate Link-2 by this vector
      Reduce the translational degrees of freedom
              of Link-2 to 0
      Set Marker-2 to have invariant global position
```

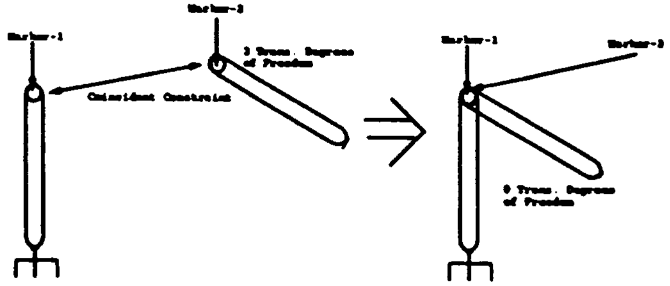


Figure 3: A Constraint Method for The Coincident Constraint

has 0 degrees of translational freedom and 1 degree of rotational freedom then every marker on the link (except the one about which the link rotates) is constrained to move in a circle. When two markers involved in a constraint are both restricted to move in simple curves, there are only a small number of locations that the markers can consistently occupy. For example, in simulating the linkage of figure 1, driving link 1 is rotated into its desired position, fixing the position of B; this means that link 2 is allowed only to rotate about B. Similarly, the position of D is fixed and link 3 may only rotate about D. C must, therefore, be at an intersection point of the circular paths allowed to the ends of links 2 and 3.[3]

2.3 Animating a Linkage

The motion of a linkage can be simulated by repeatedly incrementing the position or orientation of the driving link and allowing the constraint engine to "move (or rotate) the other links to their corresponding positions". A simple animation can be produced by showing successive snapshots.

At each step of the simulation, each link and marker is directed to record its position and orientation once the set of constraints has been solved.

[3]Notice that locus intersection methods methods lead to ambiguous results, since two circles may intersect at more than one point; the simulator must chose between the geometrically allowable results using physical principles such as continuity of motion.

The simulator can also attach "probes" to any marker in the linkage. By recording the position of the marker at each time step they can capture the complete trajectory of motion of the marker as in figure 1. This information is used later in analyzing the mechanism, see section 4 .

2.4 Building the Mechanism Graph

The simulator also has the ability to record its deductions, building up a mechanism graph similar to that in [2] . While building the mechanism graph, the simulator maintains in each link a special data structure we call a link-state-entry. This contains the number of degrees of rotational and translational freedom available to the link at that point in the solution process, as well as information normally associated with a justification-oriented truth maintenance system.

When a constraint method entry updates the state of a link, it creates a new state-entry for the link as well as a justification. The antecedents of the justification are the current link-state-entries of links entering into the constraint; the consequent of the justification is the new link-state-entry for the changed link. The justification also records the constraint which caused the update. The link-state-entries may be thought of as the nodes of the graph and the justifications as directed arcs from the old link-state-entries to the new one. Locus intersection methods similarly create new link-state-entries and build special justifications for them. The resulting graph records the process by which the simulator satisfies the constraints by incrementally moving the links and reducing their degrees of freedom.

3 Mechanism Extraction

The first step in understanding the linkage mechanism shown in figure 1 is *mechanism extraction*, in which the assembly is decomposed into sub-assemblies and the relationship between driving and driven components is established. The input to this process is the mechanism graph produced by the simulator. For a single degree of freedom linkage (all of our examples are single degree of freedom mechanisms) it is sufficient to simulate the linkage at a single time step and analyze the mechanism graph produced at that time step.

Mechanisms are identified as patterns of interaction within the mechanism graph; the patterns are identified by parsing rules like those shown in figure 4. The parsing rules in turn build up sub-structure. For the linkage of figure 1, the first rule characterizes link 1 as a crank; the second rule characterizes links 2 and 3 as a dyad. The third rule then notices that crank 1 drives the dyad formed by 2 and 3, characterizing 1,2, 3, together with the fixed frame as a four-bar linkage. Finally, links 4 and 5 are characterized as another dyad which is driven by marker E on the coupler of the four bar linkage.

The rules shown in figure 4 cover most uses of four bar linkages. Other mechanisms are identified by parsing rules similar to those shown above.

One important use of the structure produced by the parsing rules is to identify points of interest in the mechanism. This information is part of the representation of the assemblies identified by mechanism extraction. In particular, the system knows that the trajectory of coupler points of a four-bar linkage is usually interesting, particularly if the coupler point drives another identifiable mechanism. Also the system knows that the deflection angle of the rocker arm of a driven dyad is interesting.

```
If    There is a circle-circle locus intersection
      Link-2 has been constraind by a revolute joint
          to an anchor
      Link-1 has been constrained by a revolute joint
Then  Link-1 and Link-2 form a dyad
      Link-1 is the driving  link
      Link-2 is the rocker link

If    Marker M-1 is on Link-1
      The position of M-1 is set by a coincident
          constraint with an anchor
      The orientation of M-1 is set by a cooriented
          constraint with an anchor
      Marker M-2 is on Link-1
      There is a constraint between M-2 and M-3
      M-3 is on Link-2
Then  Link-1 is acting as a crank
      Link-2 is driven by Link-1

If    Dyad-1 is a revolute dyad with Driving
          Link C-1 and driven link R-1
      Crank-1 is acting as a crank driving C-1
      The constraint between Crank-1 and C-1
          is a coincident constraint
Then  Crank-1 and Dyad-1 form a four-bar linkage
      Crank-1 is the crank of the four-bar
      C-1 is the coupler of the four-bar
      R-1 is the rocker of the four-bar
```

Figure 4: Rules for Parsing a Mechanism Graph

4 Characterizing Curve Shape

At this point, mechanism extraction has parsed the linkage into 2 sub-assemblies (a four-bar linkage and a dyad) and established a driver-driven relationship between them (the dyad is driven by a coupler point on the four-bar). However, the overall behavior of the mechanism depends on a specific shape feature of the curve traced by point E (it contains a region which is a circular arc).

The next step of the analysis is to capture the relevant curves and to characterize their shapes. This is done by running a complete simulation of the linkage (i.e. by stepping the driving link through its complete range of motions); during this simulation, probes are attached to those points identified as interesting by the mechanism extraction: the coupler curve traced by point E and the angle of the rocker arm 3. There are two types of data collected: Trajectories (i.e. mappings from the driving parameter to location) and graphs (i.e. mappings from the driving parameter to a scalar value).

The following (basic image understanding) analyses are then performed:

- For each graph, the extrema of values are located (by finding the zero crossings of the first derivatives).
- For each graph, a segmentation into linear approximatons is performed using the "Split-Merge" technique.

- For each trajectory traced, the system calculates its Theta-S representation which maps position along the curve to orientation.
- For each trajectory traced, the segmentation of the theta-s curve is mapped back into a segmentation of the curve which approximates it with linear and circular segments.
- For each trajectory traced, the system calculates the radius and center curvature at each point.
- For each trajectory traced, the system calculates the points of self intersection.

Figure 5 shows the analysis of the coupler curve of the dwell mechanism of Figure 1 .

Similarly, the angle of the driven arm of the dyad is collected and analyzed. This is also shown in figure 5 .

5 Constructing an Understanding of a Mechanism

We have extracted from the simulation of the device a decomposition into driving and driven components. The decomposition has guided the choice of trajectories and displacement histories to collect. The analysis of these curves leads to a set of qualitative features characterizing the shapes of the curves. In the case of the dwell mechanism of figure 1, the system notes that:

- The angle of the rocker arm has a period of constant value.
- The coupler curve has a circular segment.

The final step in constructing an understanding of the mechanism is to notice relationships among these curve features as well as relationships between the curve features and metric properties of the linkage. It then must attempt to explain these relationships through geometric reasoning. In particular, the system notes that:

- The radius of curvature of the circular segment traced by the coupler point of the four bar linkage is nearly equal to the length of the driven arm of the dyad.
- The distance from the fixed end of the rocker of the dyad to the center of curvature of the circular arc trace by the coupler point is nearly equal to the length of the rocker of the dyad.
- There is a substantial overlap between the period during which the coupler of the four bar traces the circular arc and the period during which the rocker arm's angle holds steady.

Having found these overlaps, the system conjectures that the dyad has a dwell period which is *caused* by the coupler arm moving through a circular arc whose curvature is the same as the length of the driven arm of the dyad and whose center of curvature is at the location occupied by the dyad's joint when the circular arc is entered.

Notice that this conjecture does not itself refer to any specific metric information from the simulation. If we can support the conjecture with reasoning which also does not depend on metric information specific to this linkage, then we will have deduced something which can apply to a broader class of devices.

The final step is to use geometric reasoning to support the conjecture. The geometric knowledge needed to support the conjecture is very basic:

- Two circles intersect in at most two points.
- The center of a circle (the center of curvature of a circular arc) is the unique point equidistant (by the radius) from more than two points on the circle.

The reasoning supporting the conjecture is quite simple (and we omit it for brevity) It completes the interpretation of the mechanism using no metric information from the simulation but only qualitative shape features of the curves and symbolic relationships between joint positions. Any other mechanism satisfying these symbolic relationships will have the same behavior. General information has been extracted from the simuation of this specific device.

6 Adequacy of the Interpretation

An understanding of a mechanism should:

- Decompose the mechanism into understandable submechanisms.
- Explain how the behavior of the whole arises from that of the parts.
- Assign a purpose to each of the components.
- Enable redesign by highlighting what interactions lead to the desired behavior.

Our explanation of the dwell mechanism meets all these criteria. It decomposes the linkage into two well known sub-linkages and explains how the shape of the coupler curve causes the dyad to dwell. The two sub-linkages have well understood purposes.

We also claim that this explanation of the mechanism enables redesign. Although our system is not a redesign system, we claim that a redesign system could use the kind of information we generate.

We have run our system on several linkage mechanisms from [1] and several kinematics textbooks. The mechanisms handled include frequency multipliers, quick returns and dwells built from four bar linkages, dyads, scotch yokes, planetary gears (an extension to the domain) pantographs, etc.

7 Comparison to Other Approaches

There have been other projects on understanding kinematic mechanisms, (e.g. [4; 7; 5]). These have been concerned mainly with determining when state transitions occur, typically as a results of establishing and breaking contact between bodies. Although this is an important and difficult issue in general, it does not occur in the linkage domain: links are always in contact and the contact only occurs at the joints. This makes linkage simulation much easier than, for example, determining the behavior of a clock or a transmission.

Even more fundamentally, the kind of explanation that seems appropriate for understanding certain linkage mechanisms is quite different from those generated by these sys-

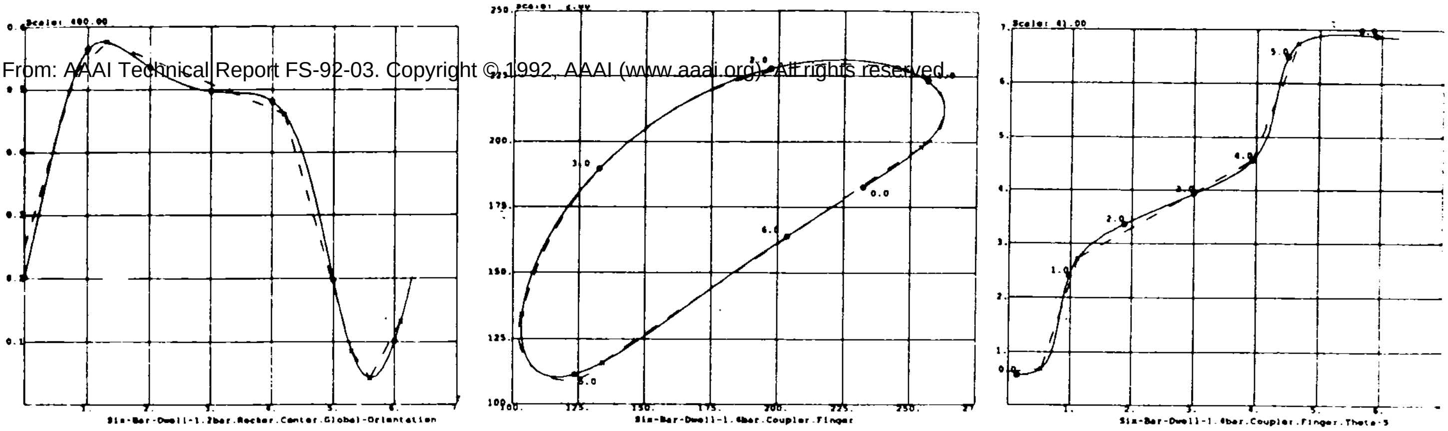


Figure 5: Analyses of The Coupler Curve and of The Rocker Arm Angle

tems, most of which, with the exception of [5; 7], fall within the qualitative reasoning paradigm.

One system [8] attempts to apply the qualitative reasoning paradigm to linkages. This system is capable of a certain form of envisioning of the behavior of four-bar linkage. However, the system as described in the paper does not predict the shape of coupler curves, nor does it deal with more complex systems which use four linkages as driving mechanisms.

However, as far as we can tell, there is no qualitative reasoning paradigm capable of deriving the shape properties of coupler curves; and this is one of our central concerns. This seems to require access to a more complete numerical description of the relative sizes of the links and the placement of the coupler point. In contrast to Kim's system which requires only inequality between certain link sizes, our system requires quantitative ratios between the links' sizes.

8 Summary

We have shown a system that can "understand" linkages, producing an explanation very similar to that given in textbooks on mechanical design.

Our system begins with numerical simulation, extracting from this a mechanism graph. The graph is then parsed into familiar modules bearing a driver-driven relationship to one another. This identifies interesting points in the mechanism whose trajectories are extracted and qualitatively characterized. Symbolic relationships between curve features are then noticed and used to generate conjectures about the functioning of the mechanism. Finally, geometric reasoning is used to support the conjecture, establishing the qualitative conditions which must obtain for the observed behavior to result.

This process has been to shown to extract general design principles from specific mechanisms, perhaps leading to the ability to support a redesign system.

References

[1] I.I. Artobolevsky. *Mechanisms in Modern Engineering Design*. MIR Publishers, Moscow, 1975.

[2] Johan deKleer. Causal and teleological resoning in circuit recognition. Technical Report TR-529, Massachussetts Institute of Technology, AI Lab., Cambridge, Mass., September 1979.

[3] Kenneth D. Forbus. Qualitative process theory. Technical Report TR-789, Massachussetts Institute of Technology, AI Lab., Cambridge, Mass., July 1981.

[4] Kenneth D. Forbus, Paul Nielsen, and Boi Faltings. Qualitative kinematics: A framework. Technical Report UILU-ENG-87-1739, University of Illinois at Urbana-Champaign, Urbana, Illinois, June 1987.

[5] Andrew Gelsey. The use of intelligently controlled simulation to predict a machine's long-term behavior. In *Proceedings of the National Conference on Artificial Intelligence*, pages 880–887. AAAI, 1991.

[6] J.A. Hrones and G.L. Nelson. *Analysis of The Four-bar Linkage*. MIT Press and John Wiley & Sons, New York, 1951.

[7] L. Joskowicz and E.P. Sacks. Computational kinematics. Technical Report CS-TR-300-90, Princeton University, Princeton, N.J., April 1990.

[8] Hyun-Kyung Kim. Qualitative kinematics of linkages. Technical Report UILU-ENG-90-1742, University of Illinois at Urbana-Champaign, Urbana, Illinois, May 1990.

[9] Glenn A. Kramer. Solving geometric constraint systems. In *Proceedings of the National Conference on Artificial Intelligence*, pages 708–714. AAAI, 1990.

[10] Benjamin J. Kuipers. Qualitative simulation. *Artificial Intelligence*, 29:289–338, 1986.

[11] Brian Williams. Qualitative analysis of mos circuits. *Artificial Intelligence*, 24:281–346, 1984.