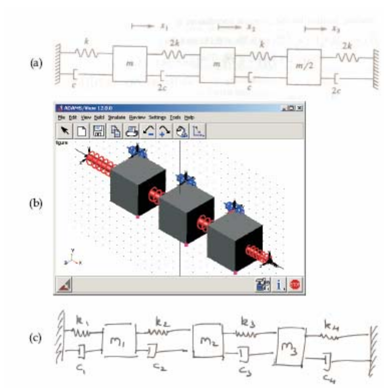
A review of: *A Sketch-Based Interface for the Design and Analysis of Simple Vibratory Mechanical Systems*

Chris Ellis

October 31, 2007

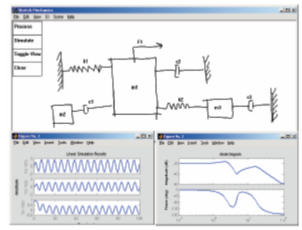
Sketch-based Design of Vibratory Mechanical Systems

- Rapidly construct diagrams of mechanical systems
- Simulate the motion of the system
- Graphs the results



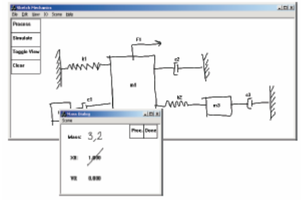
Sketch-based Interface

- Can draw masses, springs, dampers, forces, and grounds
- Can use any number of strokes in any order
- Can process and interpret strokes as simulated vibratory system



Sketch-based Interface (cont.)

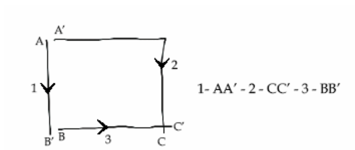
- Labels recognized symbols
- User can tap symbols and change parameters (also sketch-based)
- User can toggle between ink and "beautified" mode



Mark-group-recognize

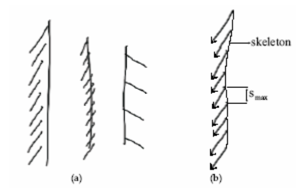
- Pull out easy-to-recognize symbols
- Cluster remaining strokes into symbols
- Perform more difficult recognition on remaining strokes

Marking Step



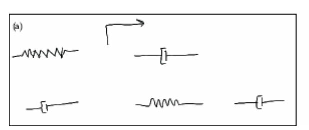
- Extract masses
  - Determine closed loop by finding minimum endpoint distance to close stroke chain
  - Threshold by 10% of total stroke length
  - Limit 5 strokes
- Allows for arbitrary stroke order, direction, and shape

Marking Step (cont.)



- Extract grounds
  - Test if separation between pair of most distant consecutive strokes < twice the average separation
  - Test if hashes all point to same quadrant (NW, NE, SW, SE)
  - Skeleton length w/in 5% of length of line from first hash to last
  - Find minimum of 4 hatch strokes meeting above criteria
  - Add hatches until a stroke breaks the pattern

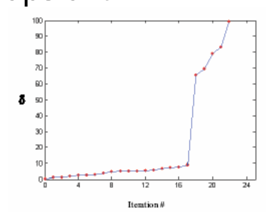
Grouping Step



- Still need to identify springs, dampers, and forces
- Concerns:
  - Can be arbitrarily close
  - Can be arbitrary size, shape, number of strokes
  - Do not know number of clusters

Grouping Step (cont.)

- Uses domain specific knowledge
  - Clusters are spatially distinct
  - Clusters do not overlap
- Use agglomerative hierarchical clustering
  - Each point initially belongs to its own group
  - Merge the two nearest-neighbor groups until dissimilarity score jumps drastically



Recognition Step

- For each cluster
  - Find segment points (speed/angle cusps)
  - Use least squares to fit arcs and lines to ink between the segment points
  - Extract nine features
    - (e.g. # of strokes, # line segments, # arc segments, # L intersections, # X intersections, # T intersections, # pairs of parallel lines, # pairs of perpendicular lines, avg distance between endpoints)
  - Classify

Recognition Step (cont.)

- Classification Assumptions
  - Assumes training features are distributed normally (Gaussian)
  - Assumes features are statistically independent
  - Neither assumption is true, but good enough
- Use Naïve Bayesian Classifier

$$P(x_j|D_i) = \frac{1}{\sigma_{i,j}\sqrt{2\pi}} exp[-\frac{(x_j - \mu_{i,j})^2}{2\sigma_{i,j}^2}]$$

Connectivity

- Similar to Grouping Step, connects nearest components together
  - Euclidian distance
  - bounding box center for mass
  - endpoints for others
  - Additional constraints
    - Springs or dampers connected to exactly one mass or ground
    - Mass or ground may have any number of springs or dampers

Finally…

- Results of the recognition used to construct motion equations
- Equations described as matrices and vectors
- Matrices and Vectors passed to Matlab
- Displacement vector returned

$D = \{d_0(t), d_1(t), \ldots d_{m-1}(t)\}$

Evaluation

- Testers generally had little to no experience with Tablet PCs
- Given ~30s of practice time
- No explanation of system functions

Evaluation (cont.)

- Worked correctly for majority of subjects
- Errors from unexpected input
  - Symbols too close
  - Symbols drawn strangely
  - False positives for masses

The End

Or is it?

(Yes it is)