

ISA EXPO 2005 Technical Conference

A Model for Human-like Reasoning in Programmable Controllers



<http://www.compsim.com>

Helena Keeley

Tom Keeley



Knowledge
Enhanced
Electronic
Logic

Discrete Manufacturing Systems

- PLCs are commonly used to
 - Orchestrate the operation of equipment
 - Coordinate the movement of material and data on the factory floor
- By:
 - Stepping the process through a series of steps
 - By sensing status (reading inputs)
 - By controlling actuators (producing outputs)

Human Interaction with PLCs

- Humans provide services not provided by the PLC
 - Start and stop the system
 - Load variables (Counts and Values)
 - Monitor the system
 - Operate equipment
 - Service equipment

- HMI facilities
 - Pushbuttons / Keypads
 - Indicator Lights
 - Alphanumeric Displays
 - Graphics Terminals / HMI software

Common PLC Programming Languages

- Supported by PLC
 - Relay Ladder Logic (RLL)
 - Sequential Function Charts (SFC)
 - Function Blocks IEC 1131-3 / 61499
 - Structured Text
 - Instruction Lists

- Using Specialty Cards
 - PC functionality providing support for conventional programming languages (C, Java...)
 - Specialty functions (Custom Settings)

- Supported by HMI Terminals
 - Custom tools / languages

Device Focus of Programming

- PLC Programming Languages focused on program flow and processing of Input and Output data
 - IF, Then, Else logic
 - I/O
 - Limited data processing capabilities
- PC Programming Languages in Specialty Cards focused on data processing
- HMI Programming Languages focused on presentation and integration with PLC data structures

Humans in the Loop

- Provide Supervisory Functions
- Provide Maintenance (corrective and preventative)
- Requires
 - Ability to interpret massive amounts of information
 - Appropriate judgment when reacting to operational pressures
 - In depth understanding of the objectives and limitations of the system

Humans may be the Weak Link

- It has been estimated that 90% of all workplace accidents identify human error as the cause.
 - "Human factors in accident modeling", by A.M.Feyer 1998
- While the industry is / has addressed human safety related issues with redundancy and other techniques their primary focus has been on equipment failure.
- Relatively little work has been done to address Human Reasoning (or Errors in Judgment) in the supervision and maintenance of the systems.

Only Incremental Improvement

- Industrial Automation Suppliers have made incremental improvements to the manufacturing process in the last 30 years.
 - Smaller and faster PLCs (more, better, faster, cheaper)
 - Alternative networks (tiered or flat architectures)
 - More I/O cards (filled out product lines)
 - Specialty cards (niche fillers)
 - RF, Redundancy, Software Tools....
- Yielding (since the PLC was invented)
 - PLC's do the same job they have been doing
 - Humans do the same job they have been doing

Why?

- Only humans have had the ability to interpret complex situations and make judgmental decisions
 - These complex situations have been difficult or impossible to effectively “program”.
- Industry has accepted that they would have to live with humans in the loop and have tried to address the situation with:
 - Increased training (have they?)
 - Better documentation (is it ever sufficient or correct?)
 - Backup personnel (do they?)

Judgmental Decisions

- Involve the interpretation of information
 - To select from a series of alternatives
 - To perform a series of relative tasks (or make relative adjustments)
- Involves balancing alternatives
- Involves understanding how each action may impact other parts of the problem domain
- This is a data fusion problem

Human Judgment in Manufacturing

- Performance optimization
 - Inventory Management
 - Performance Modifications
 - Asset Management
 - Diagnostics / Prognostics
- The human-in-the-loop *attempts* to apply written company policies when responding to judgmental issues.
- Or, if no policies are documented, the human-in-the-loop makes his/her own decisions and the company accepts the potential errors as acceptable.

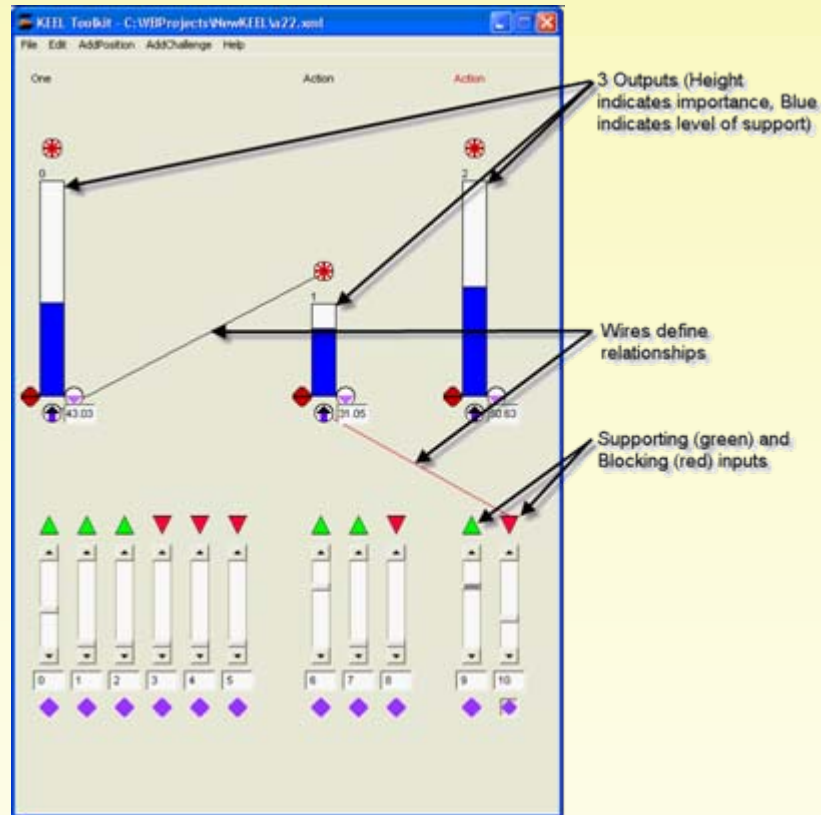
Programming Judgmental Reasoning

- Requires a way to program the “importance of information” in a dynamic environment
 - The importance of information will change based on the environment.
 - A single piece of information may carry different levels of importance in different parts of the problem domain.
- Requires a way to define how each piece of information interacts with other pieces of information
- Requires that decisions and actions are completely explainable and auditable

Hypothesis

- This requires a graphical language
 - Because judgmental decisions and subjective actions are image processing functions (right brain functions)
 - Solutions based on conventional text based languages would be too complex, hard to develop, hard to maintain.
- This requires a “dynamic” graphical language
 - So the reasoning behind judgmental decisions can be visualized.
 - So existing decisions and actions can be completely explained and audited.

Sample KEEL Language Segment



The dynamic nature of this language segment can be tested at:

<http://www.compsim.com/demos/d18/ShowKEELDemo0.html>

Sample Problem

- A machine operator is given a work order that defines expectations for the day's production.
- The operator controls the speed of the machine.
- By increasing the speed of the machine, more parts are produced, but quality degrades and wear and tear on the machine is increased.

Sample Problem (cont)

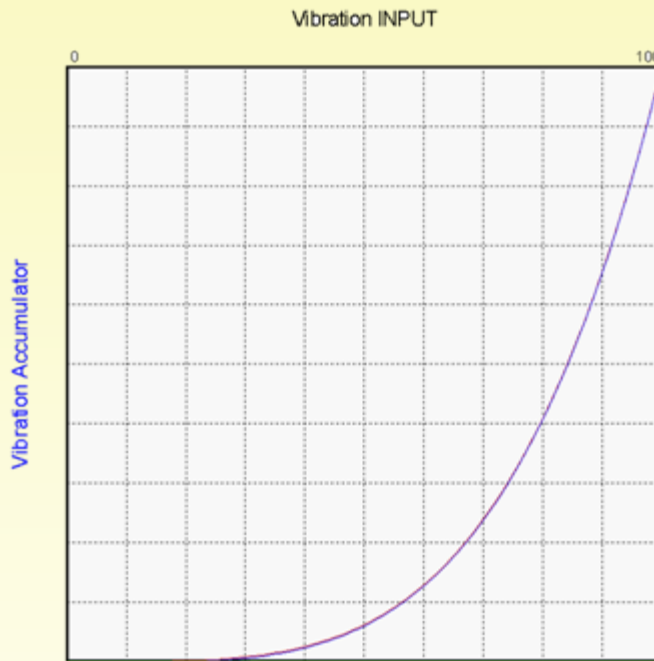
- During the day, the operator's supervisor comes by and informs the operator of the need to increase production to satisfy a new order.
- The operator increases the speed "a little bit" and production increases.
- The operator detects a new vibration in the equipment.
- After ignoring the vibration for a while, a maintenance technician is called.
- Production is stopped while the technician looks over the system.....

Sample Problem (cont)

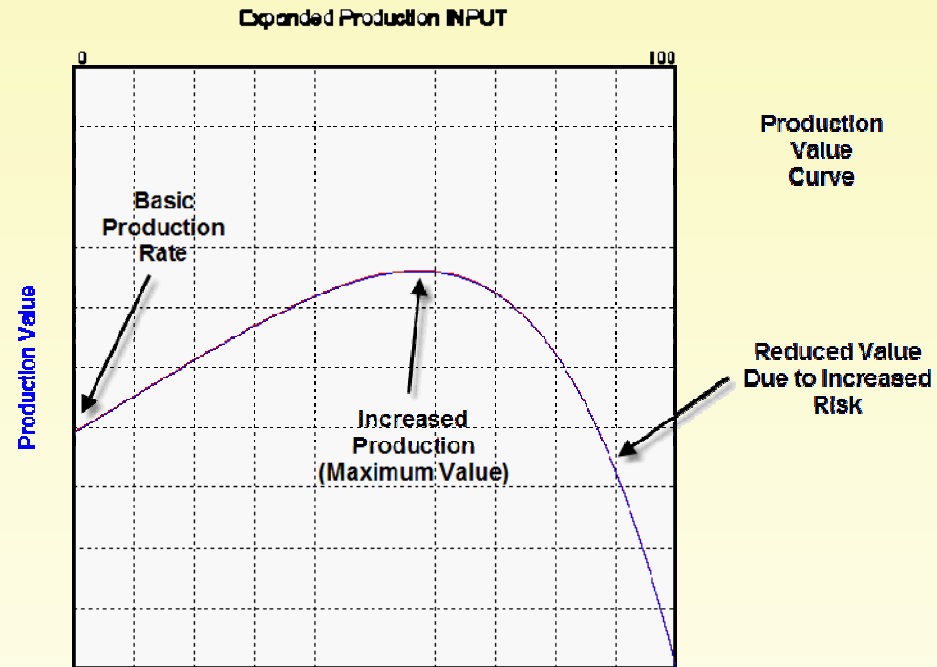
- Judgmental decisions:
 - The supervisor made a judgmental decision to try and escalate production by advising the operator of the new order.
 - The operator made a judgmental decision on how much to increase the speed of the machine.
 - The operator made another judgmental decision relative to the amount of vibration that justified calling the service technician.
 - The service technician made a judgmental decision about whether to stop production and analyze the system.

None of these judgmental decisions is auditable and if incorrect, they would be difficult to correct.

Operation Policy as Curves

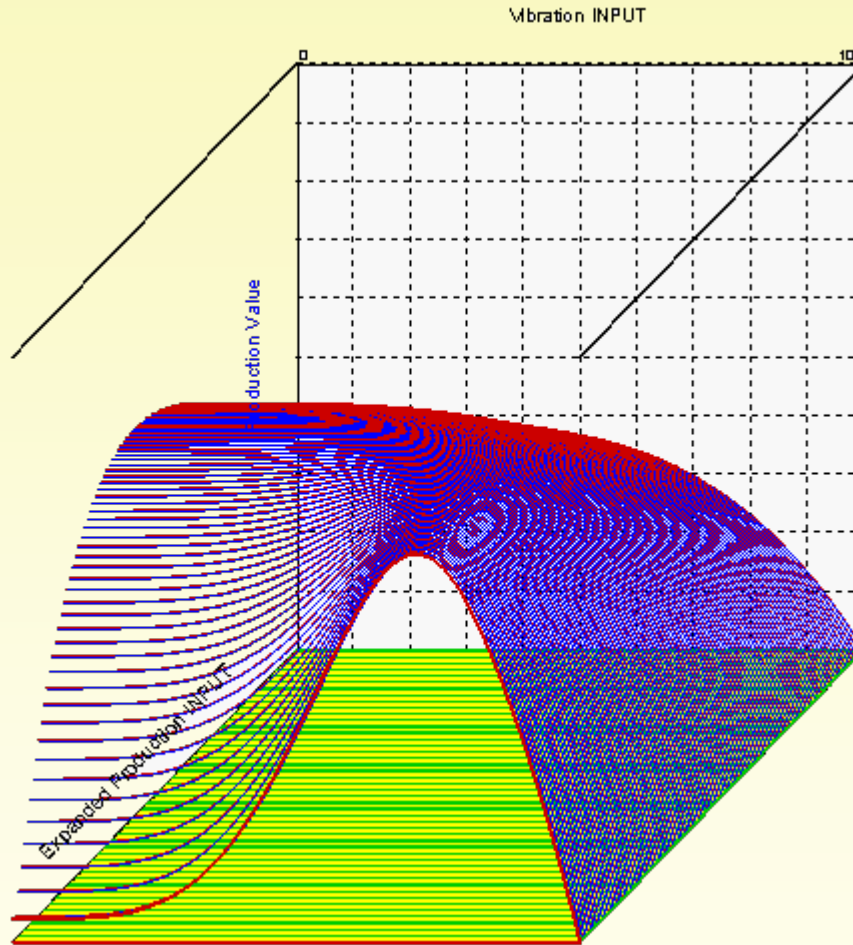


VIBRATION IMPACT ON MACHINE SPEED AND PRODUCTION RISK



PRODUCTION VALUE CURVE

Policy Described as a 3D Curve

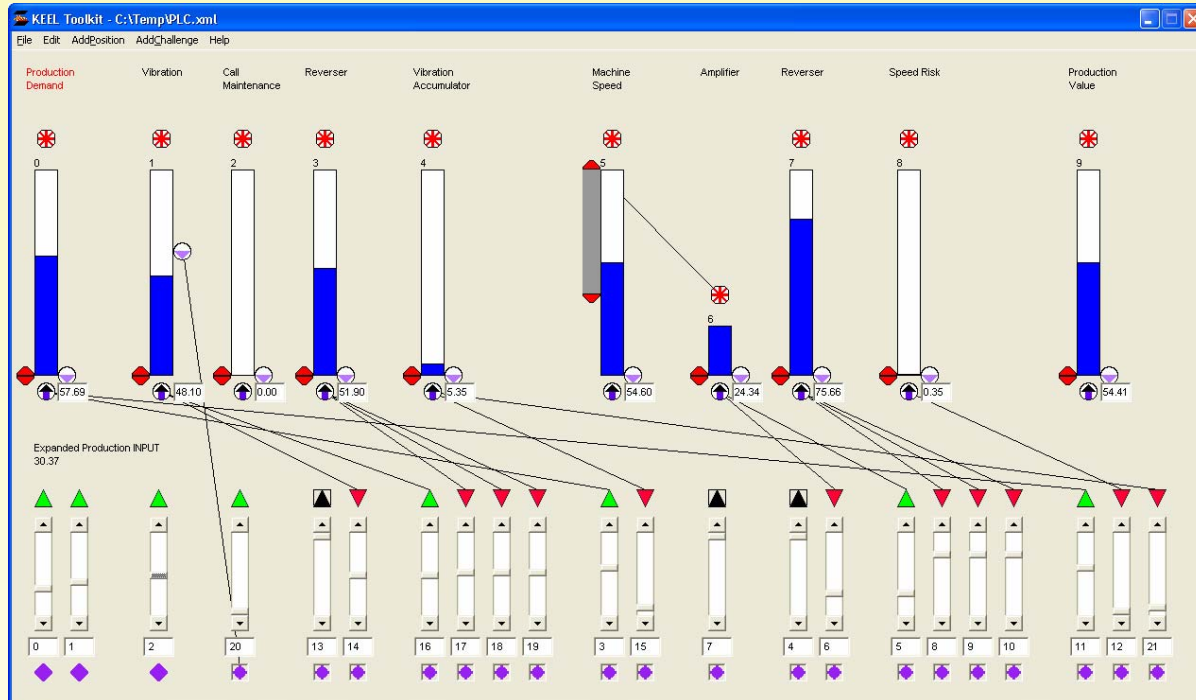


Every point on the surface of the curve defines the policy for operating speed based on demand and vibration.

By defining this “curve” as the company’s policy, the actions are completely explainable and auditable.

Any attempt to override the policy would also be auditable and subject to appropriate penalties.

Policy Defined Using the KEEL Dynamic Graphical Language



This model can be saved as Structured Text for operation on a PLC (or C, C++, Java, VB, VB.NET, Flash).

The dynamic nature of this language segment can be tested at:

<http://www.compsim.com/demos/d21/intelligentplc.htm#intelligentplc>

What PLCs Do (Today)

- Orchestrate the operation of the factory floor during normal operations according to fixed rules.
- Move data throughout the factory floor and exchange it with enterprise systems.
- Interact with human operators through HMI devices.

Non-Judgmental Actions

Summary: What PLCs Can Do

- PLCs can take on *some* of the judgmental decisions commonly left to human operators and supervisors and thus address *some* of the workplace accidents caused by human error.
- Using a dynamic graphical language these complex dynamic, non-linear, multi-dimensional, inter-related problem sets can be addressed with relative ease.

Incorporate Judgmental Decisions and Actions