

**MINE 432**  
**Robotics and Industrial Automation**  
**Lecture 14**  
**Real-Time Supervisory Control**

**Definition**

A real-time control system is an integrated computer system that responds to interrupting external events to provide accurate, fast control or alarm action. Typical response times are between ~ 10 msec to 2 - 3 seconds depending on the purpose and application.

Intelligent Control is computer modeling techniques that employ elements that mimic the human-thought-process. Response times depend on the purpose and application:

Direct real-time: 10 msec to 2-3 seconds

Pseudo real-time: 2-3 seconds to 1-2 minutes

Supervisory Control is often run in "pseudo" real-time" since the module used is usually adjusting set-points for local control loops that are operating in "true" real-time.

In Intelligent Supervisory Control, as we have seen in the earlier chapters, the elements that are used might consist of one or more of the following items:

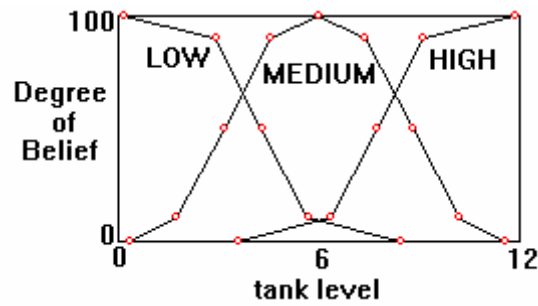
- rule-based modeling (expert systems)
- fuzzy logic inferencing
- artificial neural network modeling
- genetic algorithm optimization
- ability to explain and justify
- ability to adapt or learn from experience
- management of temporal-reasoning

A typical rule in a Supervisory System using a rule-based approach is as follows:

Rule Name: water\_valve\_high

```
IF tank level is definitely "high"
AND pump speed is "maximum"
THEN valve position change is "closed alot"
    DEFUZZIFY (valve position)
    FIND (pulp flowrate * )
    WAIT ("water_valve_high", 120 )
ELSE valve position change is not "closed alot"
```

Combining these rules with the power of Fuzzy Inferencing using fuzzy sets such as:



**Fig. 14.1.** Typical Fuzzy Sets covering the full universe of discourse of a variable.

provides a way to develop Fuzzy Associative Maps of two or more input variables into a third output variable so that rules are applied partially depending on the degree of belief in the concept in question.

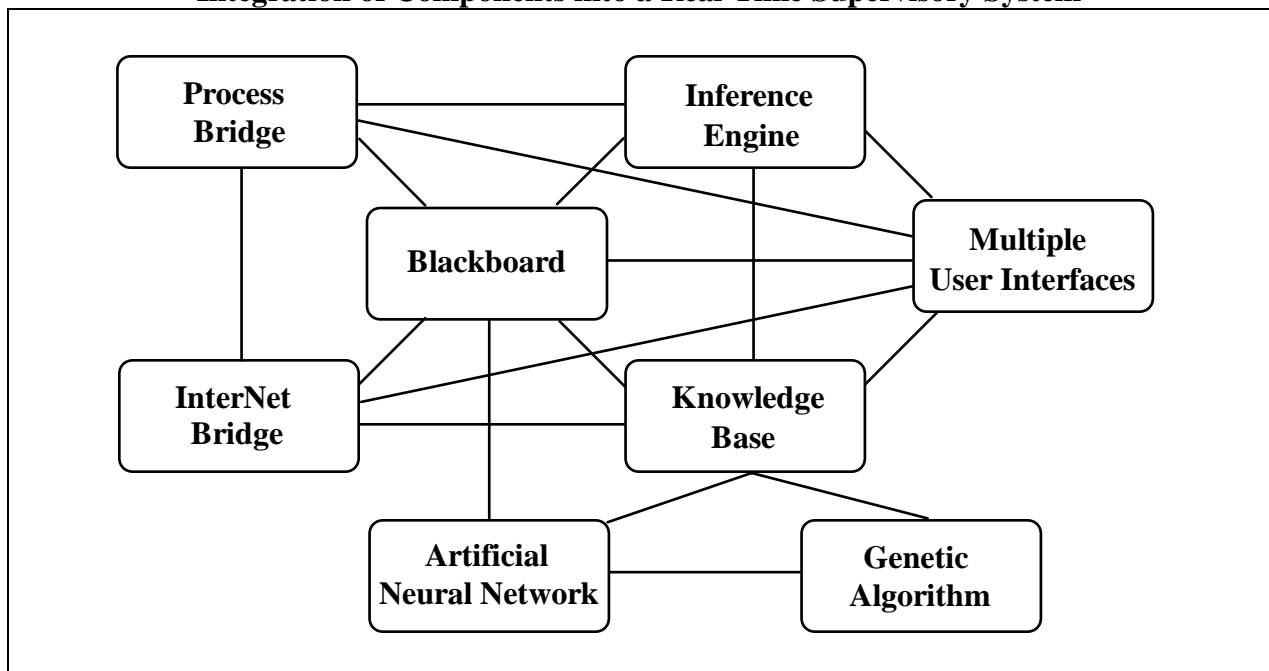
### Intelligent User Interfaces

In addition to the Inference Engine and Knowledge Base, a good real-time supervisory system will have the capability to locate "intelligence" directly within the User Interface:

Some of the items that might be found in these interfaces include:

- Process mimic diagrams with various I/O icons such as slider bars, dials, charts, etc.
- Trend diagrams of data
- Window for viewing and logging messages from the expert system
- Explanation and Justification capabilities for novice operators
- Messages can be filtered into classes depending on the needs of the workstation

### Integration of Components into a Real-Time Supervisory System



**Fig. 14.2.** Components that make up a Real Time Supervisory Control System.

There is now a trend toward the integration of intelligent supervisory modules with existing SCADA software packages such as Intellution's The FIX, Factory-Link, Real-Flex and others. The interface in these systems will look the same as a normal system without intelligence with the exception of a window for displaying messages from the Expert System. These messages might include alarm situations, process state analysis messages, explanations for control actions taken, scheduled events, etc.

Comdale Technologies' SmartWorX Suite is an example of this new approach to Intelligent Real-Time Supervisory Control.

Fig. 14.2 shows how the modules using the various components of CI can be integrated with other software: user interfaces, SCADA tools, network drivers, etc. Future systems will also allow connections to the Internet opening the plant to control from very remote locations.

## **Operating Systems**

Two operating system alternatives are useful to analyse.

UNIX-based

Windows-NT

First, UNIX-based systems can be selected but with the exception of QNX, this will mean using UNIX-based hardware systems such as SUN or IRIS (expensive hardware). QNX, on the other hand, is an operating system specifically designed for use in process control applications in a PC environment. UNIX-based as well, this O/S is designed for very rapid message-transfer and is the state-of-the-art for numerous real-time SCADA applications.

However, Windows-NT is poised to become the dominant OS in the near future and for easy data-transfer and compatibility between various software modules, WinNT is likely to be the preferred choice in the future especially since hardware continues to become faster and faster. WinNT will likely never be as fast as QNX but within 2 years, hardware will make it as fast as QNX today. Currently WinNT is about 4 times as slow as QNX.

## **Intelligent Manufacturing Systems**

Another important area to note for future expansion of CI and AI technologies is in the area of plant-wide manufacturing systems. One very important approach is the development of agent software which can be used to model a complex set of interacting processes by dividing-up the process into "holons". A "holon" is a word derived from Holography meaning the "parts that make up the whole". Holons can be combined together to make other larger holons. The interaction between holons is a complex optimization and scheduling problem that demands the use of "heuristic" methods such as are available in AI and CI techniques.

The key issues here are:

Agent-based modeling technologies

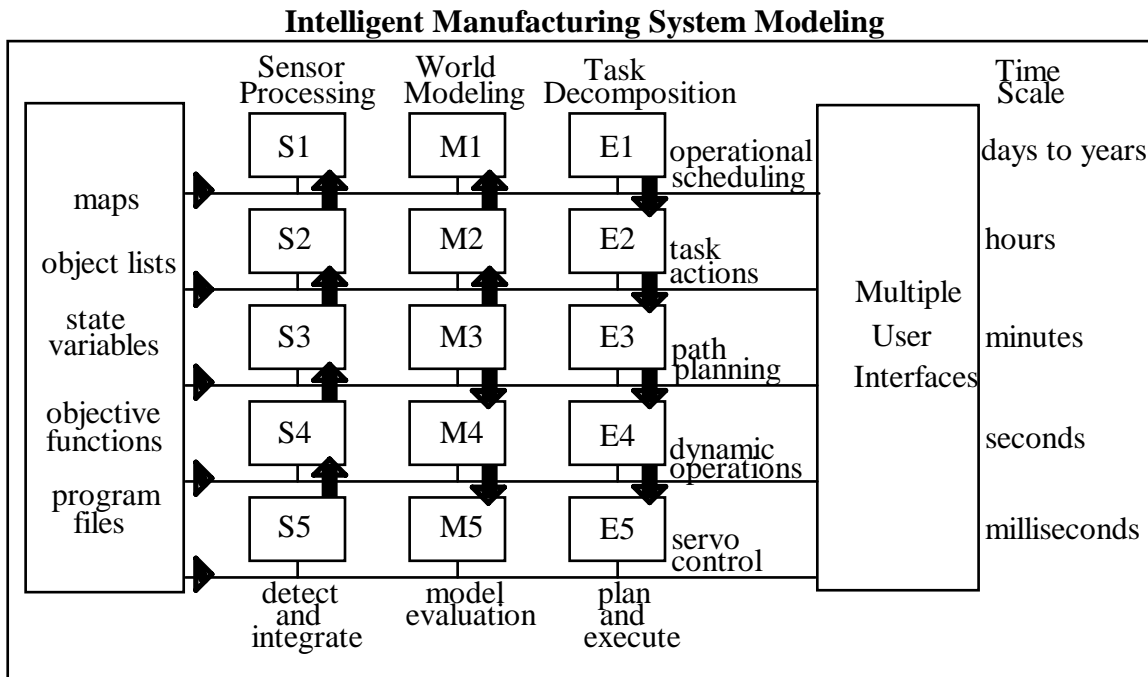
Elements of a complex process are considered to be holons

A holonic is simply an individual discrete element of a whole

Modeling methodology can be directly applied to control and operate a plant

Heuristic methods can be very helpful in providing "optimum" solutions

The diagram below shows the NASREM modeling environment being proposed to set up interacting models of very complex tasks. The different levels in this environment are analogous to the control level hierarchy shown later with the bottom level used for direct servo-control and the top level dedicated to long-term analysis across the corporation.



**Fig. 14.3.** NASA/NIST Standard Reference Modeling Environment System.

### Other Issues

The following issues must be considered in developing an Intelligent Supervisory System:

- time to develop a system
- system maintenance
- networking capabilities
- system security
- intelligent alarming
  - eliminate alarm 'showers'
  - persistence, inhibition & nuisance
  - institute an immediate action on entering or exiting an alarm state

A good tool should be capable of providing incremental development with rapid prototyping. SmartWorX for example, has a module called Expert Optimizer which assists a developer in translating the operating "facts" and "rules" into the code required by the system.

System maintenance is an important point since these systems are never finished -- they continue to grow and expand as new information comes to the attention of the developers or as the plant and ore situations change. Maintenance should be easily done by other personnel than the original developers without requiring extensive training.

A good tool will also have excellent networking capability to allow the incorporation of redundancy in the network. Mirrored modules can be made available on the network so that loss

of one work-station can be handled by the software. The back-up modules can be used instead to continue control of the process.

Security is an important feature of an intelligent supervision. Security must deal with on- and off-line development issues as well as access to output screens and input set-points. Data can be seen by multiple users but only certain variables can be changes. Passwords can be used to establish priority levels for multiple users to protect the integrity of the overall system.

Finally intelligent alarm systems are important for all plants. The context by which a plant is currently running needs to be known to provide correct advice and warnings. A plant may be starting-up, shutting-down, coming-out of an alarm situation, etc. Each of these contexts must be taken into account in determining how alarms are passed to the User Interface.

"Smart" Alarm systems can deal with alarm "showers" wherein a large influx of alarm signals is really not very useful -- only one underlying alarm is really important. These nuisance alarms need to be filtered out of the User Interface. Persistent alarms are also important issues to deal with. The time interval between consecutive messages can help but more can be done such as troubleshooting through a well-developed knowledge base to source-out the cause of the persistence. Actions can be programmed to take place upon either entering or exiting an alarm state. Intelligence can be incorporated into an overall alarm strategy to improve the environment for the plant operators as they work with the system.

### **Temporal Reasoning**

A good development tool must be able to reason about time. Functions need to be available to determine certain trends in data -- these include rates of change and time-averaging.

Scheduling of events are also important -- these include setting up actions to occur at some particular time, some particular time interval, after or before a particular occurrence, etc.

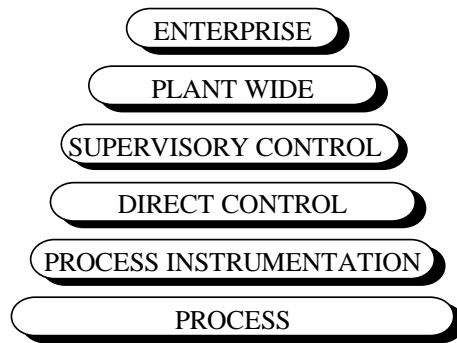
The ability to recognize certain data patterns or shapes can be very important in diagnosing certain problems. These might include concave-up or down, sudden drops or rises in data, spike-events (either up or down), multiple sequential changes, etc.

### **High Speed Intelligent Monitoring**

Intelligent real-time systems must make high-level decisions and diagnose unexpected events. They acquire data automatically, apply heuristic methods to interpret sensor readings and feed advice out to the process or up to the user via a friendly man-machine interface. Although, data are acquired directly, decision-making can be slow for effective action. Data must be filtered before passing to the knowledge base to ensure efficient processing.

In analyzing control of an industrial process, a hierarchy can be delineated (see below):

- Level 0 - Process Instrumentation,
- Level 1 - Direct control,
- Level 2 - Supervisory control,
- Level 3 - Plant-Wide control
- Level 4 - Enterprise



**Fig. 14.4.** Control System Hierarchy.

At the lowest level, instruments sense, monitor and manipulate process variables. Devices are connected to units such as single-loop controllers, PLCs or DCSs, which apply a combination of sequential or continuous-time logic. Collecting, presenting, and managing sensor data requires numerical methods. For a system to respond to external events fast, carrying out several activities simultaneously, the system must operate within a multitasking environment that can efficiently handle priority interrupts and multiple module actions.

Neither of the two upper control levels need to operate in real- or "pseudo" real-time but there is much interest today, in applying Artificial Intelligence (AI) in process control. There are many examples of non-real-time AI applications at level 3 and 4 for diagnosis and advice [1,2,3], but most interest in real-time AI is at level 2. The question addressed here is:

*"Can AI techniques be introduced into level 1 to provide support for symbolic supervisory control?"*

### **Computational Intelligence**

AI is the branch of computer science dealing with symbolic, non-algorithmic problem-solving. "von-Neumann" - type computers are designed for fast, accurate number-crunching. So how can AI move into an intensive numerical computing environment?

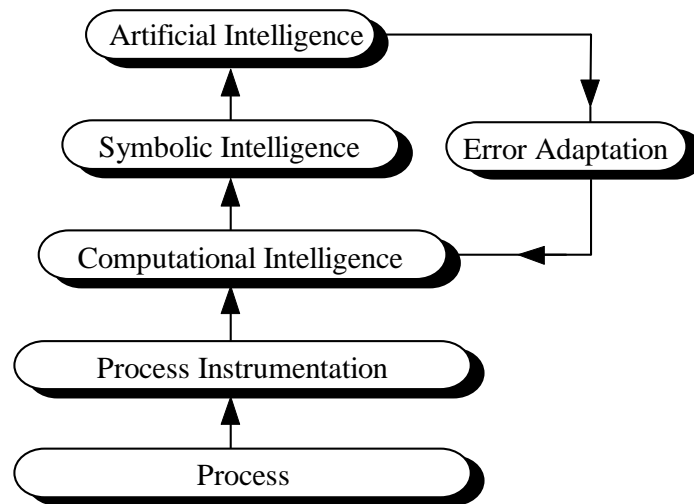
The answer is found in a newly evolving paradigm -

#### ***Computational Intelligence (CI).***

Coined by Bezdek in 1993, CI consists of *"low level" knowledge in the style of the mind* [4]. CI consists of "primitive" concepts, in the AI sense, supporting the beginning of symbolic knowledge. These "elements" are inputs to an AI structure that processes the symbols heuristically. "Primitives" are basic numerical operations: addition, subtraction, multiplication, division, and comparison that make up any complex structure to output "elements". Current hardware handles such structures efficiently and so, CI can provide support for AI methods. CI comprises Fuzzy Logic, Artificial Neural Networks and Genetic Algorithms.

This definition seems limiting, as CI should not rely only on pure arithmetic. Rules of thumb can aid the search for symbolic input for the high-intelligence level. This can increase performance speed, but usually creates error in the output. To bring AI into the lowest levels of the control hierarchy, CI modules must create "elements" very fast. By introducing AI approaches into a CI module, we cause certain error, but gain on speed. The key trade-off in real-time is always: **accuracy versus processing speed.**

Considering that error derives from using heuristics in CI, it is proposed that feedback from AI to CI can detect such error and provide interpretation. AI can test symbolic output from cooperating sensors and recognize, tune out or reduce the error (see below).

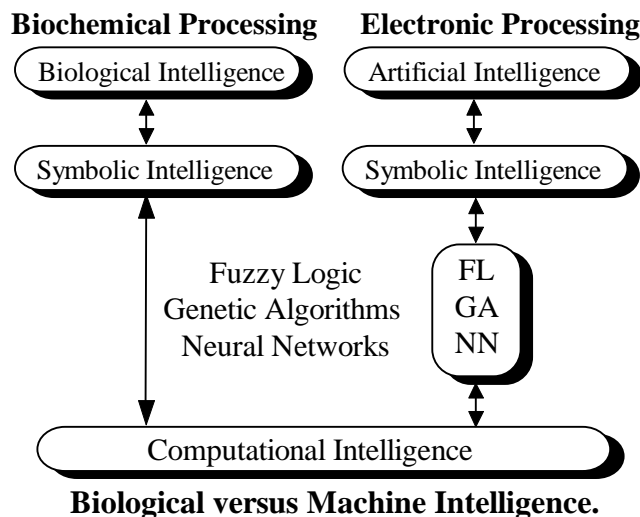


**Fig. 14.5,** Error Detection in AI module to correct the CI Module.

To assist AI in making rapid decisions intelligently, CI needs the following items:

- IF/THEN rules (inferences and relationships)
- prior knowledge (to direct the CI process)
- symbolic "elements" (output from CI module)

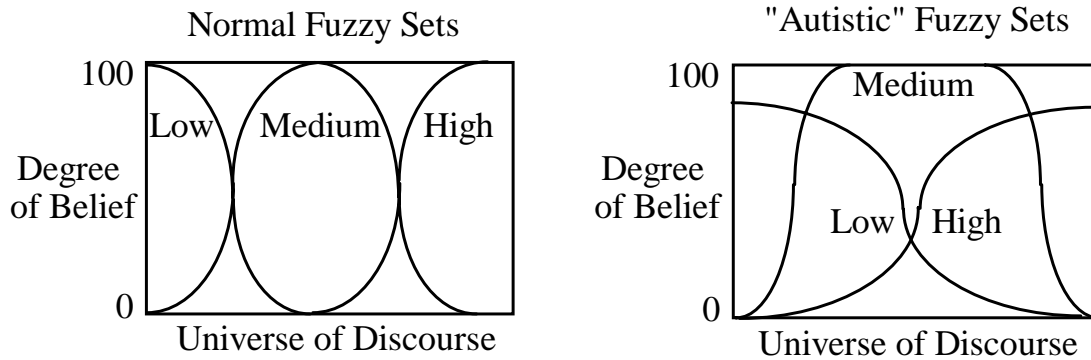
The approach resembles the hierarchy of human intelligence depicted below:



**Fig. 14.6.** Comparison of Biological Process of Information with Electronic Processing.

Biological Intelligence consists of manipulating symbols supported by low-level numerical processing to generate belief in a particular symbol. This organization is mirrored in the arrangement of AI with CI to form the basis for rapid problem-analysis. Like humans, computers work better when tasks are divided between symbolic and numerical analysis.

Within Biological Intelligence, the ability of autistic savants to carry out rapid and accurate data calculations, musical recall, etc., are examples of how the human brain can perform unusually accurate real-time computation. Whether output is intelligent or not is determined by those who interact with such exceptional people. Perhaps they use fuzzy sets with very broad support characteristics (see below). While savants may be able to tell the day of the week for a particular date, they rarely understand the significance of the date in question.



**Fig. 14.7.** Comparison of "Normal" fuzzy sets with "Autistic" fuzzy sets.

### Application of CI in Real-Time

CI can be applied for many real-time tasks:

- high-speed data acquisition and filtering
- time series analysis and pattern recognition
- direct control and smart monitoring
- real-time supervisory control

### A Real-Time Supervisory Application

An intelligent SCADA system for continuous casting of steel billets has been developed which relies on the use of a CI module for real-time data processing.

Continuous casting, involves pouring molten steel into a water-cooled copper mould. The semi-solid strand is pulled from the mould by rotating pinch-rolls. Casting speed is linked to mould metal level through a standard PID loop. Level changes are reflected by changes in casting speed. Metal level varies appreciably when turbulent conditions exist from "ropy" stream conditions. The machine oscillates to strip the solidifying shell from the mould wall. Displacement is usually sinusoidal but when binding occurs, these signals are distorted.

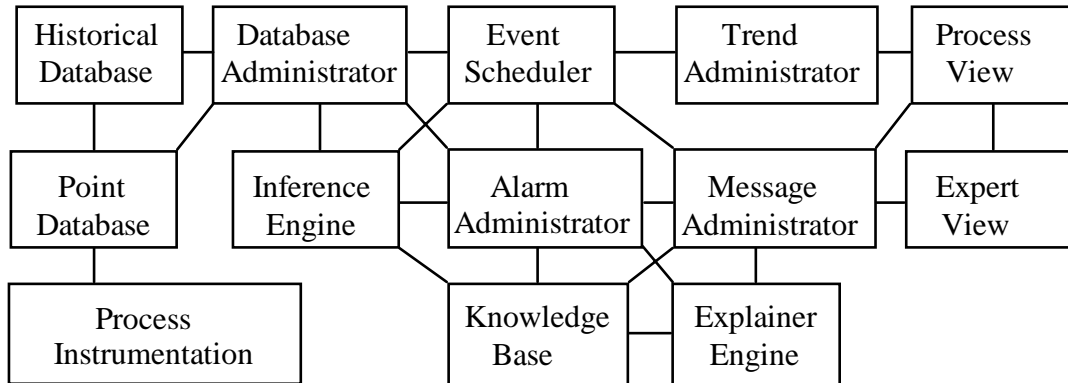
The following variables are monitored:

- Metal level position and variation
- Casting speed
- Cooling water flow and temperature
- Mould temperature, displacement and loads

The sensors and devices comprise the following:

- casting speed / metal level sensors
- mould thermocouples (TC)
- mechanical sensors - LVDT
- single loop PID controllers

The objective was to create the "**Intelligent**" **Mould** to monitor and control the process of continuous casting. ProcessVision, a SCADA development tool from Comdale Technologies, was used to build the application. The multi-tasking nature of a real-time ProcessVision system is depicted in below. Each module interacts with others in a true multi-tasking fashion with appropriate interrupts and priority scheduling as required.



**Fig. 14.8.** Typical Configuration of ProcessVision.

High-level supervisory decision-making occurs in the inference engine, the knowledge base and the explainer module. ProcessVision (PV) operates under QNX, a UNIX-based truly-distributed real-time multitasking operating system running on a PC. To accomplish rapid data acquisition within the SCADA, a Keithley-Metrabyte DAS-20 plug-in board was used.

### **Building the CI module**

Our research group at UBC has been working to interpret patterns from sensor responses from many field trials conducted over the past 20 years. Specific curve shapes from thermocouple time responses (temperature peaks, drops, etc.) are related to specific billet defects. These correlations make up the knowledge base which detects surface defects - bleeds/laps and depressions.

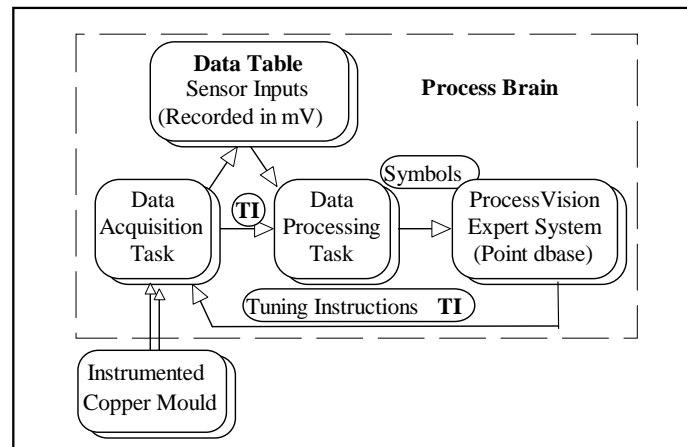
When a temperature drop and rise propagates down the mould, a defect is indicated. The drop to base temperature ratio together with upset duration define the significance and extent of a defect. These inputs are used to predict defects. A data acquisition rate of 20 Hz captures all important features of these traces. The "pseudo" real-time Expert System (ES) is not fast enough to process data intelligently at this rate, so application of a CI module is a necessity.

A multi-threading 'C' program was written for the CI module. The objective was to acquire data without delay while processing data. The two main threads are distinguished in Fig. 14.8.

The first program, *main* task, runs in an infinitive loop to execute the following functions:

- collect sensor data and store in RAM-resident table
- receive tuning instructions from ES
- create data processing task

The *processing* task receives data from the *main* task and sequentially filters inputs from each channel. Up to 5 functions per channel are applied:



**Fig. 14.9.** Structure of the "Intelligent" Mould.

- average** - calculate ave. over specified number of points; pass key-word-triplet to PV.
- minmax** - find minimum and maximum values; pass two key-word-triplets to PV.
- storedata** - place collected data (in volts) into a file.
- compare** - combine data from 2 channels to calculate negative strip time using inputs from an LVDT and the metal level sensor, at 200 Hz.
- valley** - this is an example of shape recognition and feature extraction that finds a "valley" shape in the data. The algorithm uses prior knowledge to direct the search, and a "window" technique to locate the minimum, left and right maximums. The search is set up by the **TI** signal from the AI module. The function can recognize 5 valleys in the data table and pass 20 key-word-triplets to PV (see Table below).
- calibration** - convert input volts to actual values.

The *main* task collects data while the *processing* task filters recorded inputs. When the processing task finishes its routines, it "kills" itself. The *main* program recreates the processing task when it completes another acquisition cycle. The multi-threading design of the driver provides continuous data acquisition and processing of signals at up to 400 Hz.

Connection to PV uses the Comdale 3rd-Party Interface Library, communicating with the point database using keyword triplets. The Table below shows typical TC signal output.

<u>CI output for one sensor for one scan cycle</u>		
<u>Keyword Triplet</u>	<u>Value</u>	<u>Degree of Belief</u>
TC1.drop.@f	39.9	100
TC1.base.@f	132.7	100
TC1.span.@f	9.9	100
TC1.time.@f	8.6	100

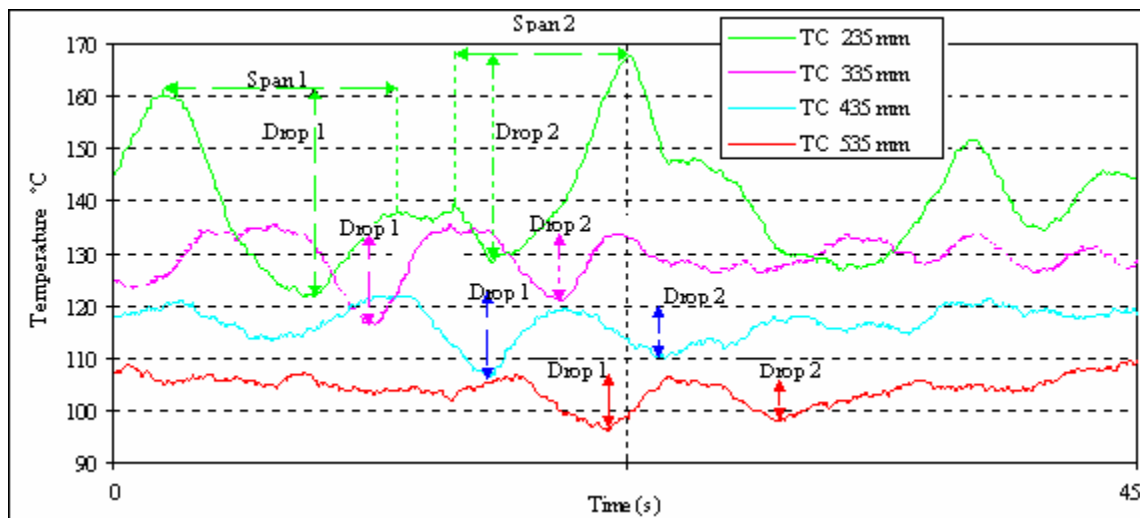
The system tracks 4 thermocouples (TC). The AI module uses Fuzzy Logic to interpret the temperature drop and span interval. Together with the number of valleys from each TC, a degree of belief (DoB) is established that a bleed/lap or depression has occurred.

## Results

The "**Intelligent**" Mould was implemented and tested at two Canadian mini mills in November 1994 and March 1995. The system was used for real-time monitoring. For the first time, negative-

strip-time could be followed on the screen in real-time. Correlations between metal-level fluctuations, casting speed and billet defects were clearly evident.

The system has established a clear association of ropy streams and a turbulent meniscus with on-line TC responses. These conditions lead to the formation of surface defects such as depressions, laps or bleeds. Depression defects were predicted and a high degree of mapping between predicted and actual defects was obtained. TC responses obtained during the March 1995 plant trial are shown below. Two temperature drops were detected for the period shown. The degrees of belief that a defect occurred were 98 and 97 respectively.

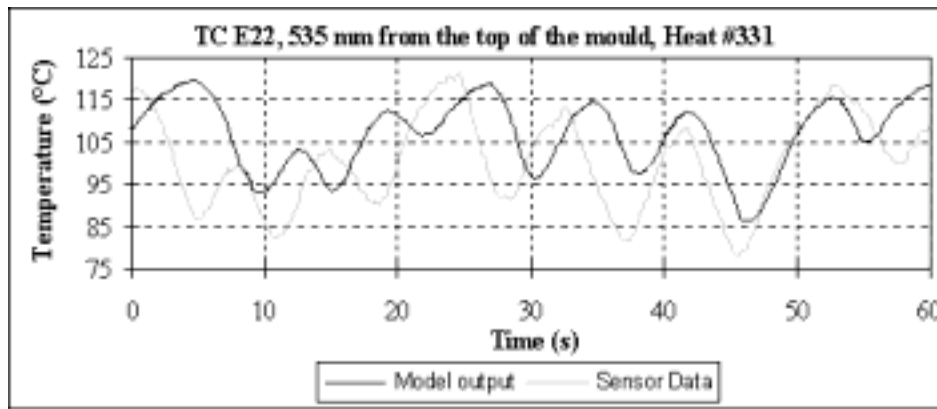


**Fig. 14.10.** Thermocouple traces from a continuous casting trial at Alta Steel - temperature drop indicate a surface defect formation (lap/bleed or depression).

The system correctly ignored other "apparent" drops. The CI module can be setup to include or exclude temperature depressions based on the desired degree of surface defect detection. Correlation with surface measurements was virtually perfect. Examination of the billet cast during this period proved the predictions correct: depressions were obvious on the billet surface. A mathematical model to predict temperature drop and duration based on depression measurements shows high correlation with the on-line detection system. The future plan for the "Smart Mould" is to use these defect predictions to provide an on-line quality rating for each cast billet.

To create a solid foundation for a real-time Quality Control System, we examined a number of billets for which detailed temperature trends were recorded. Surface defects were measured according to their distribution, position, and surface depth. These served as input to a 2-dimensional mould heat-transfer model. The depth and distribution of billet defects were translated into upsets in mould heat flux, assuming that the deepest depression represented about a 70% upset in heat extraction. Output from the model was obtained for several thermocouples located around and down the mould. Model output was plotted concurrently with real sensor data obtained during casting, as depicted below.

A clear correlation between these trends can be seen - one for the model temperature profile based on depression measurements and the second for the direct measurements taken from the TC. The modeled output has the same shape as the real thermocouple-time response.



**Fig. 14.11.** Comparison of temperature trend modeled from surface profile measurements to sensor-based temperature trend.

The differences in the position of peak and drop values are a result of assuming a 70% heat-flux upset in this analysis. Subsequent follow-up showed that 80% gave a better fit. The model assumes a constant casting speed, although speed varies appreciably during casting. Fluctuating casting speed also causes shifting of the thermocouple data. The analysis has established clear correlation with existing defects on the billets, sensor data, and our mould heat-flux model providing a basis for direct measurement of defects at the time of formation.

### Conclusion

A new paradigm known as Computational Intelligence is evolving in which intelligent numerical manipulation forms the underpinning of successful real-time AI. A parallel with Biological Intelligence is evident from man's ability to process numbers rapidly when necessary or whenever specialized tasks are required. The CI module acquires high-frequency data rapidly using AI techniques that allow monitoring and control of a continuous casting process. Field trials have confirmed the method can be applied to predict bleeds/laps and depressions. Our efforts are continuing to develop fast-responding modules to give sufficient accuracy for high-level AI processing. Effective real-time analysis of billet quality will lead to a useful performance criterion to give on-line analysis of cast product quality. The ultimate goal of this system is to provide on-line billet quality measurement. Both surface and internal defects are included in this system and an adaptable billet quality index is derived. Finally, with respect to real-time intelligent control, the following points are relevant:

- knowledge-based modeling is finding increased use within industrial plants
  - on-line diagnosis and training
  - real-time monitoring and control
- on-line learning will become more important in future generations of these technologies
- voice recognition and voice-activated messages will become more widely used
- operator replacement must be addressed as knowledge becomes a valuable commodity

### References

1. S. Kumar, J. Meech, I. Samarasekera, K. Brimacombe, "Knowledge Engineering an Expert System for Quality Problems in Continuous Billet Casting of Steel Billets", *Iron & Steelmaker*, 1993, 20(9), 29-36.
2. K. Otsuka et al., "Expert System for Blast Furnace Operation", *Sumitomo Search*, 1992, 50, 43-50.
3. R. Edwards & A. Mular, "An Expert System Supervisor of a Flotation Circuit", *CIM Bulletin.*, 1992, 69-76.
4. J.C. Bezdek, "What is Computational Intelligence?", *Computational Intelligence - Imitating Life*, 1994, IEEE World Congress on CI (WCCI), 1-12.