CONF-860638--4 DR-1806-8

860009720
UCRL- 94286
PREPRINT

MASTER

RECEIVED BY OSTI APR 30 1986

AN EXPERT SYSTEM TO CONTROL A FUSION ENERGY EXPERIMENT

UCRL--94286

DE86 009720

R.R. Johnson
T. Canales
D. Lager

This paper was prepared for submittal to:
1986 AMERICAN CONTROL CONFERENCE
Seattle, WA.

June 18-20, 1986

Lawrence Livermore National Laboratory

This is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint is made available with the understanding that it will not be cited or reproduced without the permission of the author.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

AN EXPERT SYSTEM TO CONTROL A FUSION ENERGY EXPERIMENT

Rowland R Johnson, Tom Canales and Darrel Lager

Lawrence Livermore National Laboratories
P.O. Box 808
Livermore, California 94550

Abstract

This paper describes a system that automates neutral beam source conditioning. The system achieves this with artificial intelligence techniques by encoding the behavior of several experts as a set of if-then rules in an expert system. One of the functions of the expert system is to control an adaptive controller that, in turn, controls the neutral beam source. The architecture of the system is presented followed by a description of its performance.

Introduction and Motivation

Magnetic Fusion Energy (MFE) experiments are done with systems that are composed of several complex subsystems. It is clear that in order for MFE to be feasible that most, if not all, of the subsystems must be operated automatically. The goal of this project is the automation of one of these subsystems.

A major component of all MFE experiments is plasma heating. This is accomplished with a device called a neutral beam source depicted in Figure 1.

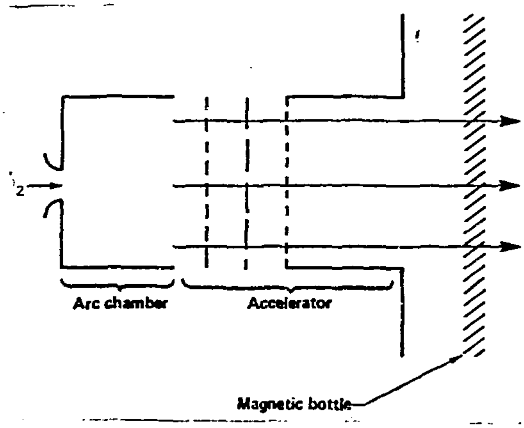

Figure 1
Neutral Beam Source Device

Plasma is generated by dumping deuterium gas into an arc chamber where it is ionized. The resulting plasma enters the accelerator which heats the plasma and injects it into the magnetic bottle. There are three major parameters used in source operations. Parc is the power delivered to the arc chamber. Vaccel and Iaccel are the potential applied to the accelerator grids and the resulting current seen by these grids. In order for fusion to be sustained Parc must be at least 75Kw and Vaccel must be at least 75 Kv.

A newly manufactured or freshly overhauled source cannot operate reliably at these power levels. One of the reasons for this is one of the high voltage components inside the source may have a dust particle or other imperfection on it. When the accelerator voltage is raised the particle causes arc-over. If the power level is too high the arc causes a pit to form and ruins the source. If it is the proper level the arc will burn away the particle and leave a smooth surface. With the particle gone the voltage may be increased to a new level to burn away the next particle. The process of starting at low power levels and moving to higher power levels is known as source conditioning.

Conditioning and the day to day operation of a source are done under the supervision of an operations staff. Source conditioning may be viewed as a conventional control system with the operator being the controller. However, replacing the operator with a conventional control system is unworkable since it relies, in part, on being able to model the conditioning process and source in a precise way. This is very difficult due to the highly non-linear nature of the arcs and plasmas involved.

An iteration in the conditioning process is described as follows. The operator decides on parameters and sends them to the source. The source runs for a short time in which measurements are taken and displayed before the operator. The combination of input parameters and output measurements are known as a shot. After several good shots at one power level the source becomes "conditioned" enough to run at that level. Under certain conditions an operator will then try to raise the power level for the next shot. If good shots are obtained then that power level is maintained. If not, the power level must be reduced or "deconditioning" will happen. Conditioning then is just a sequence of shots with the ultimate goal of running at high power levels.

Using expert system techniques for automatic conditioning is a viable alternative to a conventional controller since operators do quite well at the task of conditioning without exact models of the process. The purpose of an expert system is to encode the expertise that has been acquired by an operator through experience. No attempt is made to analyze this experience and arrive at a theory that explains it. Rather, the goal is to codify the manner in which an operator represents knowledge and reasons about the process.

The overall configuration used for conditioning is shown in Figure 2.

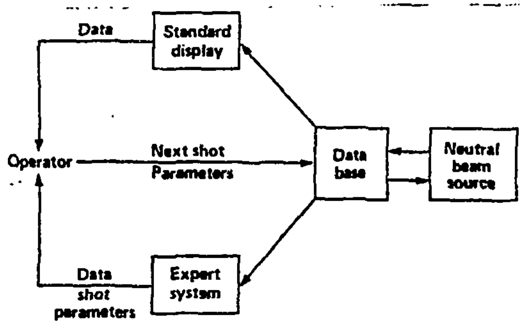

Figure 2
Source Conditioning Operations

Normally, the operator receives data from the standard display, determines shot parameters for the next shot and sends them to the database. When the expert system is being used it gets the same data provided by the standard display and derives shot parameters for the next shot. The operator reviews the suggestions made by the expert system and, if appropriate, uses them.

Ultimately, the operator will be removed from the loop and be called upon only in rare situations. The present configuration provides an effective means to extract the necessary information from the operator and encode it in the expert system. In addition, it insures that an error made by the expert system will not go unchecked and cause serious damage to the source.

Expert System Techniques

This system uses a forward chaining rule based inferencing technique. Essentially, the knowledge of an operator is encoded in a set of *rules*. The concept of a rule is much like that of an IF-THEN statement in a programming language. Rules have the form

(RULE <rule-name> <LHS> <RHS>)

where <LHS> stands for the left hand side and represents a condition. ... stands for the right hand side and represents an action. When a rule's <LHS> is true its corresponding <RHS> is evaluated. Currently, there are well over 300 such rules in the system. Basically, the system iterates on the following two steps. First, it finds a rule with a true <LHS>. Second, this rule's <RHS> is executed which may cause more rules to have a true <LHS>. This continues unti all the rules with a true <LHS> have had their <RHS>'s executed. The clauses that make up the <LHS> are LISP functions that make references to history database. The <RHS> is LISP code to modify the database.

Knowledge engineering is the *process of* developing a set of rules that encode an operator's knowledge. Although the concept of a rule is straight forward, knowledge engineering a difficult task. The difficulties arise when large numbers of interacting rules are used. Di cussions with operators result in new rules bein added. Often, old rules that used to work do no work now as a result of the new additions. As t number of rules grow it gets exponentially more difficult to add new rules.

The system provides a programming environme that facilitates rule development. The primiti that make up <LHS>s and <RHS>s are meant to reflect the operator's representation of the pro lem. Old rules are easily changed and new rules are just as easily added. In short, the task o knowledge engineering is not hindered by the tas of knowledge encoding.

One of the system requirements is that the results of several past shots must be examined b the rules. For example there are a number of cas where the operators use a rule of the form

IF the *last five* shots have meet some particula condition. (e.g. maximum-duration has been greater then 80)
THEN do something (e.g. raise vaccel by 2.0)

A mechanism that allows this kind of historical referencing of information from previous shots h been implemented by placing each piece of information about a shot in a named register. For each shot a set of these named registers is saved. These register sets can then be used to access t register values for any of the prior shots. Internally, a list of hash tables is used for the representation of the registers thus providing quick access. Currently the system uses shot register sets

SHOT-4, SHOT-3, SHOT-2, ,SHOT-1, SHOT-0 and SHOT

where, SHOT-4....SHOT-1 are those shots occurrin before the most recent shot,
SHOT-0 is the most *recently completed* sho
and, SHOT+1 is the shot to be fired next.

This provides a sliding window of six register sets accessible at any one time and is depicted Figure 3.

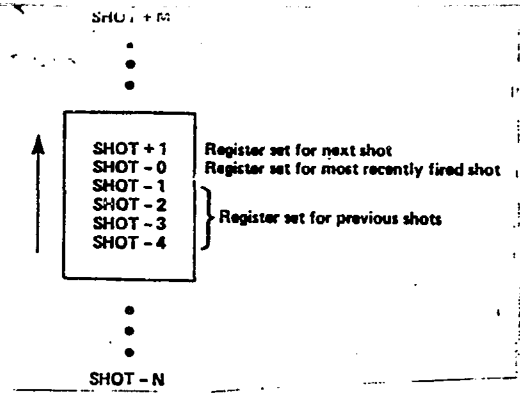

Figure 3.

When a shot is fired each shot register set SHOT-i becomes SHOT-(i+1), (e.g. SHOT+1 becomes SHOT-0, SHOT-0 becomes SHOT-1 ,etc.) SHOT-N is discarded and a new shot register set is created that becomes SHOT+1.

The value of a register is obtained with the lisp form

(<shot register set> <register name>)

For example,

(SHOT-3 Avg-Vaccel)

yields the average value of Vaccel for SHOT-3. The lisp form

(SETR <register name> <value>)

sets <register name> to be <value> in SHOT-0. This is used primarily to assign values for the most recent shot. Similarly, the lisp form

(SETR-NEXT <register name> <value>)

sets a register in SHOT+1. The primary use in this case is to assign suggested shot parameters for the next shot.

Controlling the Source

Source conditioning requires that two processes be controlled. The primary goal of the system is to control the conditioning process through the use of expert system techniques. In order to accomplish this it is necessary to control the accelerator process. Furthermore, the accelerator process can be controlled with conventional control system techniques. This observation resulted in a system that is a hybrid of expert and control system techniques.

Specifically, the operators (and the expert system) control the conditioning process by varying the accelerator voltage, Vaccel, and the density of the beam relative to the optimum density for that Vaccel. At optimum density the beam is most uniformly focused. They express the relative beam density in terms of Delta-Iaccel as amperes above (over-dense) or below (under-dense) the accelerator current at optimum density, Iaccel0. The notation Delta-Iaccel = [-1.5, -.5] means that Iaccel should be .5 to 1.5 amperes under-dense. The hardware for the source, however, is controlled by varying Vaccel and Parc, the arc power. The expert system uses an adaptive controller to make the transformation.

The relationship between the expert system and the controller for the neutral beam source is depicted in Figure 4.

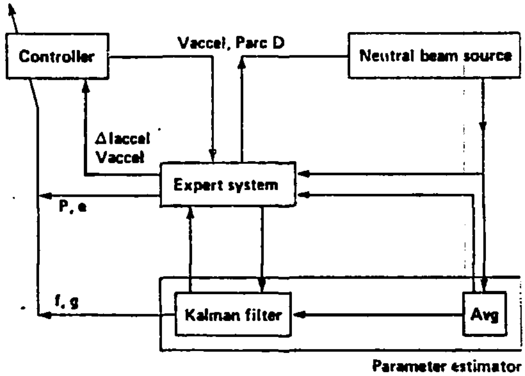

Figure 4
Controller Configuration

After a shot the digitized signals from the source are sent to the expert system and are processed by a set of averaging routines that are robust against interrupts and other forms of drop-outs in the signals. The outputs from the averagers are sent to the expert system and to a Kalman filter used as a parameter estimator. The parameters from the Kalman filter and the desired Vaccel and Delta-Iaccel from the expert system are given to the controller equations to compute the desired Vaccel and Parc for the next shot. The expert system exercises control over the Kalman filter by making it ignore measurements when the averages have large error bounds — when there were too many interrupts during a shot, for example. It also monitors the innovations sequence from the filter for diagnosing the health of the source. When the innovations become large it means that the model in the filter is unable to adequately describe the measurements from the source — usually because of a hardware failure. Rules in the expert system tell it to ignore large innovations after changes in Vaccel since it takes a few shots for the parameter estimator to adjust the model.

The parameter estimator/Kalman filter is Potter's algorithm for recursive least squares as described by Bierman [4]. The algorithm assumes the system being modeled is time invariant and so after a large number of measurements (shots) it makes very small changes to the parameters. As a result it is unable to track changes in the system when Vaccel is increased as the source becomes conditioned. This difficulty was overcome by the introduction of a "forgetting factor" from Ljung [3]. Its effect was to produce a non-zero steady state gain thus keeping the filter "open" and able to track changes in the system.

The model used in the parameter estimator was taken from Theil [1] who observed a linear relationship between the arc power and Iaccel current for a given source at a given state of conditioning. This is expressed as

$$\text{Parc} = f * \text{Iaccel} + g \qquad (1)$$

where Iaccel and Parc are averaged measured values from the last shot, and f and g are are the parameters estimated by the Kalman filter.

The controller is given a desired Vaccel and Delta-Iaccel by the expert system. It uses the Child-Langmuir law

$$\text{Iaccel0} = P * \text{Vaccel}^{e} \qquad (2)$$

to compute Iaccel0, the Iaccel at the optimum perveance P. The values for P and e are determined experimentally for each source by maintaining Vaccel constant and varying Parc until a calorimeter indicates the beam is uniformly focused. The Iaccel at the optimum is measured. This process is repeated for a new Vaccel 10Kv higher. The optimum perveance, P, and e are then calculated from two equations with two unknowns using (2) and the known Vaccels and Iaccels.

A range of acceptable accelerator currents [IaccelL, IaccelU] is computed by adding Iaccel0 to the upper and lower values for Delta-Iaccel. Equation (1) is used to solve for IaccelN, the accelerator current that would be obtained on the next shot given the Parc setting used on the last shot and the f and g parameters computed by the Kalman filter. If IaccelN is within the interval [IaccelL, IaccelU] then the value for Parc is left unchanged from the last shot and Vaccel is sent to the source as the setting for the next shot. Otherwise equation (1) is again used to compute the desired arc power, ParcD, for the next shot using f, g, and IaccelL the lower acceptable value of accelerator current. Vaccel and ParcD are then sent to the source as settings for the next shot.

In the final expert system there were two other Kalman filters which estimated the parameters for models of the filament voltage/current, and the arc voltage/current. In some systems the parameters from these models are necessary to control the filament and arc power supplies, but it was not necessary for the sources used in this experiment. Although these models were not used to control the source, they were used by the expert system to monitor its health.

Stability

The foremost stability concern is to insure combination of Vaccel and Parc that will damage the source is never recommended. This can occur if Parc is too far different from the optimum perveance value causing the beam to diverge and strike the walls of the source. Damage can also occur if Vaccel is too high for the present conditioning level of the source causing pitting of the source components due to arcing. Both the operators and the expert system solve these problems by making only incremental changes in Vaccel and Parc and monitoring the effects of a change on the %-on time. If divergence or arc-over occurs the hardware shuts down the power supply for 10 milliseconds reducing the %-on time when several such interrupts happen during a shot. If the %-on time is too low the expert system keeps decreasing Vaccel until the shots are successful again.

The expert system and adaptive controller must deal with the situation where the estimates of the f and g parameters somehow become incorrect causing a bad value of Parc to be recommended. If this makes the source have too low a %-on time for subsequent shots there will not enough data for the Kalman filter to update and correct the f and g parameters. So the controller may continue to recommend improper values for Parc. A rule checks if there have been 5 consecutive shots with too low a %-on time and Vaccel was decreased twice in that interval then the expert system stops and displays a message for the operator. This condition most often occurs due to a hardware failure in the source, rather than bad parameter estimates.

Rule Organization

For each shot the system evaluates, in sequence, four separate groups of rules. Each such evaluation is called a phase and accomplishes a specific task. The four phases are signal processing, analysis, set-parameters, and display. Depending on the shot, the signal processing phase executes the appropriate signal processing routines. The display phase determines what waveforms the operator should see. A detailed discussion of the analysis and set-parameter phases follows.

One objective of the analysis phase is to derive a high level description of the shot. To this end a descriptive set of registers are used. The registers shots-since-last-increase, shots-since-last-decrease, and shots-since-last-change describe the current state of Vaccel. The mode register contains 'ADVANCING if the current strategy is to try to increase Vaccel and 'RETREATING otherwise. The territory register describes where Vaccel in relationship to where the source is conditioned. 'OLD means that Vaccel is below where the source is known to be conditioned. 'NEW represents that Vaccel is equal or greater than the conditioned level. Finally, 'BRAND-NEW means that Vaccel is at a point where significant accelerator on-time has not been obtained.

For example, consider the high-level description, short-term-trend, from the rule

```
(rule short-term-trend
      (and (shot-0 %-on)
           (shot-1 %-on))
      (setr short-term-trend (/ (shot-0 %-on)
                                (shot-1 %-on))))
```

The <LHS> for this rule is true if the percentage on-time for this and the last shot have been computed. If so, the <RHS> is executed and sets a register called short-term-trend that is the ratio of these two percentages. This register is used in the set-parameters phase to determine if conditioning is beginning to happen.

The analysis phase also runs the Kalman filters if there was sufficient on-time to assure accurate results. The differences between the expected and actual measurements are analyzed. Since the models appear to be incomplete the large differences that are expected in certain situations are ignored. A large unexplained error is indicative of possible source problems and the operator is informed.

The source has an elaborate mechanism for detecting, handling, and reporting faults. The reported faults are checked for consistency with each other and with other data. Inconsistencies imply source problems and are reported.

Finally, a search is made of all waveforms for any unexplained high frequency energy. Such an occurrence is often symptomatic of problems that, if allowed to persist, may lead to source damage.

The set-parameters phase uses the information produced by the signal processing and analysis phases to determine the suggested parameters for the next shot. For example, consider the rule in Figure 5.

```
(rule set-vaccel-suggested-110
   (and (equal (shot-0 territory) 'old)
        (equal (shot-0 mode) 'advancing)
        (>= (shot-0 shots-since-last-change) 3)
        (< (shot-0 short-term-trend) 1.2)
        (< (max (shot-0 max-duration)
                (shot-1 max-duration)
                (shot-2 max-duration))
           100.0))
    (prog()
     (setr-next v-accel-setpt-suggested
                (1- (shot-0 v-accel-setpt)))
     (setr-next min-delta-iaccel-suggested -2.0)
     (setr-next max-delta-iaccel-suggested -1.0)))
```

Figure 5.

This rule looks for a situation where the source is being operated in old territory, i.e. it has been conditioned at this level. The current strategy is to advance Vaccel when possible. The last change was to increase power but the last 3 shots have been the same. During these last 3 shots the maximum contiguous on-time has been less than 100 milliseconds Finally, there has been no improvement in the short term. The action of this rule is to suggest that Vaccel be reduced by 1. In additioned the desired value of Iaccel is 1 to 2 amps under the value required for operating on perveance. This reflects the operator's desire to run slightly underdense when changing levels.

Development Aids

In the course of developing this system it became apparent that a number of debugging tools would be necessary to increase the knowledge engineer's ability to quickly identify problems during rule development. The same kinds of tools used by traditional programming efforts (e.g. symbolic debuggers) would be very useful if applied to rules. For instance, it is known that a rule will fire only when its <LHS> evaluates to true. It should then be possible in cases where a rule fails to fire, to apply a stepper function to the evaluation of each clause in the <LHS> to determine why the rule failed. This conclusion led to the development of a rule-tracer based on the trace facility resident on the Symbolics LISP machine used for developing the expert system. The rule-tracer provides the knowledge engineer with the ability to break, print information, or step through the evaluation of a rule. Additionally, these features can be invoked either prior to or at run time.

The design of the system also allows dynamic run time evaluation of rules. In addition to being able to add new rules at run time, this makes possible a simple backup and retry capability that prevents having to re-run the entire system when simple errors are encountered. As an illustration of these tools, assume that the rule in Figure 5 should have fired, but for some reason has not. The stepper function can be invoked on the left hand side of this rule. Through an interactive process of questioning the outcome of each step, and noting where adjustments may be necessary, the rule can be re-evaluated, backed-up and retried. After discussions with the expert it is concluded that the LHS condition is too restrictive and that the threshold of 100 milliseconds used in the third clause should be raised to 150 milliseconds. This detailed stepwise execution of the system enhances the direct involvement of the expert at the lowest levels of the knowledge engineering phase.

Performance

Measuring the performance of the system is difficult because conditioning is a non-repeatable process. Therefore, it is not possible to condition the same source first with an operator and then with the expert system. Since it is not possible to determine performance on a comparative basis two other metrics were used to determine performance.

The first is a measure of "does it successfully condition a source". Unfortunately, testing time was limited and it was not possible to use the system for the entire range on any one source. However, the system was used several times in the 55-75 Kv range and successful conditioning was achieved.

The second is a measure of "does it make the same decisions that an operator makes". It is important to note that individual operators vary with each other and, at times, with themselves. Nonetheless, an attempt was made to determine the operator's judgement of the quality of each suggestion made by the system. These judgements fall into four categories. Most were in the first category where the suggestions were judged to be the same as an operator. Some fell into the second category in which the system's suggestion was judged to be better than the operators. This occurred mostly because the system's methods of computing averages, on time, etc. are more precisc than that of the operators. In the third category are suggestions that operators felt were incorrect. However, the number of such suggestions was not large enough to prevent conditioning. Finally, in the fourth category are the catastrophic suggestions in the sense that, if followed, damage would be done to the source. There were none in this category.

Conclusion

It is interesting to note that when this project began it was felt that very robust sophisticated signal processing and control algorithms would be needed to insure the expert system never received error-prone data. Instead, quite the opposite has proven true in that it is relatively easy to write rules for the expert to detect and circumvent bad data. The result is that relatively simple signal and control algorithms could be used, substantially reducing the implementation time for the system.

Acknowledgements

This work was performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Labs under contract number W-7405-ENG-48.

The authors are indebted to Bob Pancotti, Gary Preckshot, and John Woodruff for the help received in implementing the system. Bob Wyman conceived the concept of building the system and supported it. Bill Labiak, Ken Autio, Ken Gillespie, and Jim Sullivan provided the operator expertise through many, many hours of conversation.

References

[1] E. Thiel, "Simulation of Automatic Control of MFTF-B Neutral Beams," Lawrence Berkeley Labs Report LBL-19252

[2] G. Pollock, "Automation of Multiple Neutral Beam Injector Controls at Lawrence Livermore Laboratory," 7th Symposium on Engineering Problems of Fusion Technology, 1977

[3] Lennert Ljung and Torsten Soderstrom, "Theory and Practice of Recursive Identification", MIT Press, Cambridge, Massachussetts, 1983, pp 273-274.

[4] Gerald J. Bierman, "Factorization Methods for Discrete Sequential Estimation", Academic Press, New New York, 1977, p30.