Model-Based Diagnosis of Power-Station Control Systems

Dong-Yih Bau, University of Paris XI
Patrick J. Brézillon, University of Paris VI

M ANY APPROACHES HAVE been investigated for defining and solving the diagnostic task. One knowledge-based monitoring system\(^1\) detects faulty behavior in a dynamic system based on sensor measurements. The system triggers an alarm when a predetermined threshold is exceeded, and the process operator responds manually according to the diagnostic clue.

Knowledge-based systems have also been used to diagnose a wide range of automated control systems, which detect faults in the system being controlled and execute sequences of actions to eliminate the faults without operator intervention. Such faulty behavior can be detected directly by observing the controlled system’s functioning, or inferred indirectly by observing the control system’s functioning.

Since control systems are themselves subject to failures, they often contain redundant and backup controllers to complete their fault repairs. If a control system displays abnormalities during the correction process, a knowledge-based system can help process operators complete the diagnosis in their tasks of maintaining and repairing the control system.

The off-line diagnosis of automated control systems has three special features:

1. Reasoning concerns the overall system (both the controlled system and the control system).
2. Diagnosing the control system also involves diagnosing the controlled system.
3. Many redundant observations must be analyzed to identify all abnormal behavior in the control system.

We propose a two-step diagnosis that first establishes a synthetic conclusion, and then a model-based explanation. We use this approach in SEPT (Suivi d’Evenements dans les Postes du Transport), a functioning knowledge-based system for diagnosing control systems in extra-high-voltage power networks.\(^2\)\(^-\)\(^4\)

Diagnosing control systems

In nuclear power plants, chemical-processing plants, aerospace systems, and so on, the control system includes many controllers that are integrated in the system being controlled. These controllers must detect faulty behavior on line and act quickly to eliminate it. Faulty behavior can have catastrophic effects, such as system damage or indirect disturbance of clients. The control system’s design aims at shutting down the potentially damaged part of the system while keeping the rest functioning.

The control design is a simple, direct (or almost direct) mapping between the faulty behavior and the control system’s action.
sequences. The control system contains redundant and backup controllers to ensure proper operation during the automated correction process, even if some controllers are faulty. Using redundant controllers means that several controllers can cooperate to detect and repair the same fault in the system. Backup controllers participate in the corrective process when triggered by a controller malfunction. Thus, all started controllers compete and cooperate over time, and their overall actions ensure the elimination of any faulty system behavior.

To describe such a complex system more formally: Let \( P \) be a system performing some functions, and let \( C \) be a control system comprising a set of controllers \( \{C_1, C_2, \ldots, C_n\} \). If a fault appears in \( P \), \( C \) tries to diagnose and eliminate it. In this sense, \( C \) is both an automated diagnostic system and a corrective system. The detection by \( C \) of a fault in \( P \) leads \( C \) to repair \( P \): when \( P \) behaves normally, we say the controllers are “sleeping.” From here on, we’ll use \( C+P \) to refer to the overall system (the control system plus the controlled system).

**The control model.** We can obtain a description of \( C+P \) from the intended design. This type of “deep” description is given in terms of structure, function, behavior, and \( P \)’s fault features, which constrain \( C \)’s functioning. Thus, the description is both a normal model and a fault model: With respect to \( C \), the control model is a normal model that describes how \( C \) normally works; with respect to \( P \), it is a fault model because the description includes \( P \)’s fault features. A representation of the description is called a “control model.”

**Faults and abnormalities.** To distinguish \( P \)’s faulty behaviors from those of \( C \), we’ll use “fault” for \( P \) and “abnormality” for \( C \). To determine faults and abnormalities, we must observe \( P \) and \( C \). We can obtain fault information directly by observing \( P \), or we can infer it by observing \( C \). Faults and abnormalities are thus application-dependent.

Abnormalities in \( C \) are detected at the controllers when they act to eliminate a fault in \( P \). We identify an abnormality by comparing the controller’s behavior both with its normal actions and with information on the fault in \( P \). If a controller’s behavior cannot be explained by its normal actions, it has experienced an “intrinsic malfunction.” Such an abnormality can appear for any fault in \( P \). If a controller has failed to detect or eliminate a fault when it should, it has behaved “incoherently.” An incoherence in one controller is not easy to detect because the redundant controllers can still eliminate the fault. Thus, a controller’s functioning can appear normal at the controller level but abnormal based on the fault.

**The diagnostic task.** The diagnostic task has three stages — intrinsic-abnormality diagnosis, fault diagnosis, and incoherence diagnosis — and it results in two kinds of output — a diagnosis and an explanation. In diagnosing a control system, the input is generally a structural and functional description of \( C+P \), and a set of observations of the controller actions in \( C \) and the fault in \( P \) (depending on their accessibility). The output then consists of a synthetic conclusion of the major abnormalities in \( C \) and the fault in \( P \), and a causal explanation of \( C+P \)’s actions starting with the fault, through \( C \)’s activation and (normal or abnormal) behavior, to the elimination of the fault in \( P \).

**Diagnosing control systems in French power networks.**

A French power network is a meshed network of power lines, whose nodes are extra-high-voltage substations containing several busbars or transformers, interconnected by cutoff systems (see Figure 1). Power lines are also connected to busbars via cutoff systems. Power lines, busbar sections, and transformers are parts of the controlled system \( P \).

The cutoff systems include equipment such as relays and circuit breakers. Relays detect faults and (depending on the fault features, especially the location) eliminate them by opening circuit breakers. In that sense, a cutoff system acts as a controller, and the set of cutoff systems corresponds to the control system \( C \).

The control system includes backup controllers to help eliminate faults when a circuit breaker refuses to open. For example, in Figure 1 consider a fault in Line 1, which is connected to Busbar A. If the circuit breaker of the Line 1 controller does not open, a lockout auxiliary relay (not shown) opens the circuit breakers in Coupler 1 and Coupler 2. (The opening of the circuit breakers is accomplished via the breaker-failure auxiliary relays in Couplers 1 and 2.)

When a fault is detected, the controllers usually

- detect short circuits,
- execute a sequence of relay operations,
- issue orders to trip appropriate circuit breakers,
- operate the circuit breakers, and

![Figure 1. A simplified extra-high-voltage substation. Shaded boxes are cutoff systems. PX1 and PX2 are redundant distance relays, PJB is a busbar relay, CB is a circuit breaker, and ADD is a breaker-failure auxiliary relay.](image-url)
The structural description changes as circuit breakers open or close. The functional specification indicates (in terms of the structure and fault features) how \( C \) detects a fault and how controllers start to function. It also describes how the started controllers normally operate in terms of structure, function, behavior, and fault features under various conditions.

Figure 2 is a simplified part of the SEPT functional description of relays and circuit breakers. The functioning is described as elementary sequences of events — called e-sequences — that define the links between events and the diverse conditions under which they appear.

**Faults and abnormalities.** Faults in power networks are called short circuits and can occur in busbars, lines, or transformers. A short circuit can be transient or permanent, resistant or nonresistant, monophased or polyphased. It is detected by the two controllers that connect the faulty part in \( P \) to other parts of \( P \). The fault is also detected by the many nearby controllers through electrical feeding paths. The fault features are not directly observed but are inferred from the recorded signal events in \( C \).

In power networks, fault detection depends on "protection zones," which correspond to distances and directions between the fault and the controllers. For instance, in Figure 3 the protection zones for Controller 1 are Zone 1 (the first 80 percent of the line from Controller 1), Zone 2 (the last 20 percent of Line 1, and Substation B at the other end of the line) and Zone 3 (any other part except Line 1). A relay time setting for each protection zone delays the trip order that the relay sends to the circuit breaker. Relay time settings are adjusted so that a minimum number of circuit breakers will be opened. In Figure 3 the relay time setting for Zone 1 is shorter than that for Zone 3, and since the fault is in Zone 1, the circuit breaker of Controller 1 must open first (before Controllers 2 or 3).

Controllers can malfunction or exhibit abnormal behavior due to equipment failure, relay mistiming, or trip-order errors. An incorrect protection zone for a fault can lead to subtler faulty behavior. To overcome these problems, the control system can eliminate a fault by different combinations of circuit-breaker openings.
depending on all controllers' behaviors (normal or abnormal). In Figure 3, if the circuit breaker of Controller 1 is not opened at the right time, then Controllers 2 and 3 must be opened at their relay time settings.

**Model-based diagnosis of the control system.** Figure 4 shows SEPT's architecture, emphasizing the model-based-explanation component. SEPT identifies abnormal behavior in C by interpreting error-prone signal events. As input, the diagnostic task receives the structural and functional description of C+P and a series of observations corresponding to the recorded signal events in C. The diagnostic task then performs associative reasoning that compiles diagnostic knowledge from the control model, yielding C's intrinsic abnormalities, fault features, and incoherence between C's functioning and the fault (see Figure 5).

Then, the explanation session begins by presenting the initial topology (structural specification) and the identified fault. As a spontaneous explanation, SEPT presents the normal started devices and the predicted "protection zone" of the controllers. The diagnostic task then begins a cycle of analyzing and predicting the signal events according to their temporal chronology (see Figure 6).

At each point in time, SEPT
- presents the signal events appearing at that time;
- analyzes the signal events up to that time to identify the successful e-sequences;
- presents the main current states of the overall system, identified abnormalities of C, and so on;
- predicts C's near-future functioning based on SEPT's understanding of overall system behavior so far;
- lets users ask questions for a deep understanding of the behavior of C+P; and
- chooses the next point in time and reiterates until there are no more signal events to treat.

Finally, SEPT explains how it eliminated (structurally isolated) the fault.

### Diagnosis output

The first step in SEPT's output consists of diagnoses for intrinsic abnormalities, faults, and incoherence.

**Intrinsic-abnormality diagnosis.** SEPT identifies all possible intrinsic abnormalities in C by comparing signal events with the diagnostic knowledge stored in the knowledge base. Such a database-like approach is different from the categorical classification often used in knowledge-based diagnostic systems.
A nonresistant, monophased short circuit appeared in the first part of Line 1 at time $t_0 + 10$ ms.

PX1 and PX2 in the Line 1 controller must detect the fault in Zone 1:
PX1 and PX2 in the Line 2 controller must detect the fault in Zone 3:
PJBs must detect the fault and must be oriented toward the Busbar Section A where the faulty line is connected.

** Continue (user can ask a question here)

$t = t_0 + 30$ ms (signal events)
1. Line 1 PX1.Zone1 $t_0 + 30$ ms
2. Line 2 PX1.Zone1 $t_0 + 30$ ms
3. Line 2 PX2.Zone1 $t_0 + 30$ ms
4. Coupler 1 PJB.DIR.A $t_0 + 30$ ms
Successful e-sequences:
(e-sequence.2. Line 1. PX1)
(e-sequence.1. Coupler 1. PJB)

** Continue
Inopportune e-sequences:
(e-sequence.2. Line 2. PX1)
(e-sequence.2. Line 2. PX2)

There was an untimely activation by "wrong protection zone" of PX1 and PX2 in the Line 2 controller that detected the fault in Zone 1 instead of Zone 3.

** Continue
Abnormal behaviors:
ab1: PJB in Coupler 2 controller did not emit direction signal PJB.DIR.A orienting towards the busbar section A to which the faulty Line 1 is connected.
ab2: The PX2 in the Line 1 controller did not emit a signal zone PX2.Zone1 like PX1.

** Continue
$t = t_0 + 51$ ms (signal events)
5. Line 1 DT.PX1.Mono $t_0 + 51$ ms
6. Line 2 DT.PX1.Mono $t_0 + 51$ ms
7. Line 2 DT.PX2.Mono $t_0 + 51$ ms
Successful e-sequences:
(e-sequence.3. Line 1. PX1)
(e-sequence.3. Line 2. PX1)
(e-sequence.3. Line 2. PX2)
and monophased authorization signals exist.

** Continue
Abnormal behaviors:
ab1: PJB in Coupler 2 controller did not emit direction signal PJB.DIR.A orienting towards the busbar section A to which the faulty Line 1 is connected.
ab2: The PX2 in the Line 1 controller did not emit a signal zone PX2.Zone1 like PX1.

** Continue
$t = t_0 + 70$ ms (signal events)
8. Line 2 DJ.OFF $t_0 + 70$ ms
(Topology graphics are updated, after the opening of the CB in the Line 2 controller)
Successful e-sequences:
(e-sequence.5. Line 2. CB)
The circuit breaker CB in the Line 2 controller opens at time $t_0 + 70$ ms.

Figure 6. A SEPT explanation (the second output step).

In this diagnosis, each piece of diagnostic knowledge is given as a conjunction of signal events (described in an If-Then rule) and a list of possible abnormalities. For example:

If (1) the circuit breaker in line controller X emits an open-in-monophase signal at time $t$;
(2) the redundant distance relay in line controller X emits a monophased trip order at time $t'$; and
(3) $t'$ appears within $[t' + 10$ ms, $t' + 30$ ms],

Then the sequence of signal events is correct.
Possible abnormalities:
(1) is true and (2) is false.
(1) and (2) are true, but (3) is false.

 Such rules are compiled from the knowledge in the control model. The more frequent violations of the conjunction are defined by the expert and specify the possible abnormalities. This piece of diagnostic knowledge does not refer to the fault in the controlled system.

Fault diagnosis. SEPT identifies a fault's location — the most important feature — by weighted voting among controllers that do not have intrinsic abnormalities. Lines, busbars, and transformers are candidates for the fault location. A positive vote means the fault might have appeared at that location, while a negative vote means the opposite. The location with the most accumulated votes defines the fault location. This method uses a heuristic-statistic type of approach and is empirical, but it is still affected by the way controllers are started by the fault (especially given the notion of a protection zone).

SEPT also identifies other fault features. For example, the following rule describes how to identify a monophased fault:

If a fault is identified in the first part of one line and the distance relay PXi for that line has emitted monophased trip order DT.PXI.Mono,
Then the fault must be monophased.

Incoherence diagnosis. This diagnosis checks for incoherence between C's functioning and the identified fault. Further controller malfunctions can be identified, such as untimely activation or improper activation by the wrong protection zone. These problems can occur if the sensors are unreliable. The following rule is a piece of diagnostic knowledge of this type:

If (1) a nonresistant fault is identified in the first part of Line X;
(2) there is an electrical feeding path between Line X and Line Y; and
(3) the distance relay PXi in the Line Y controller detects the fault in Zone 1 instead of Zone 3,
Then the distance relay PXi in the Line Y controller was started by the wrong protection zone.

Explanation output

The model-based-explanation system shown in Figure 4 has three parts: a control model, an explainer, and dynamic knowledge. The explainer is a general explanation engine that reasons on a readily usable form of the control model to build and use dynamic knowledge, which contains items like predicted e-sequences, successful e-sequences, dynamic system states, and question traces. Here we discuss only the control model and the explainer.
Control model. The control model comprises the substation’s initial structure and the functional specification of relays and circuit breakers. (We detail only the latter here.) The functional specification is a set of e-sequences, each with corresponding canned text that can be instantiated with the appropriate components for explanation.

E-sequences. Using a frame-like formalism, an e-sequence contains five major slots: component, inputs, outputs, preconditions, and postconditions (see Figure 7). A component is an element of P or C. Inputs are incoming signal events that will cause the appearance of outputs (outgoing signal events at specified relay time settings), as long as the preconditions and postconditions are verified at the proper time. Preconditions specify the conditions that must be true before the inputs appear. Postconditions specify the conditions that must be true when the outputs appear.

An e-sequence is generally expressed as a temporal relation between signal events, accounting for some conditions: “The appearance of the outputs at time \( t + \text{delay} \) is caused by the appearance of the inputs at time \( t \) and the true value of diverse conditions either before time \( t \) or at time \( t + \text{delay} \).” For example, e-sequence.3 in Figure 2 would be represented as:

Component: Line
Input: PXi.Zonel at time \( t \);
Output: DT.PXiMono at time \( t + 20 \text{ ms} \);
Preconditions: “The fault is monophased,”
“Monophased authorization signal exists”;
Postcondition: “Fault exists relative to PXi at \( t + 20 \text{ ms} \).”

More formally, we can describe an e-sequence’s temporal causality as:

If (1) inputs are true at \( t \),
(2) preconditions are true before \( t \); and
(3) postconditions are true at \( t + \text{delay} \);
Then outputs are true at \( t + \text{delay} \).

From this, the explainer can identify a successful e-sequence when all corresponding outputs exist in the central printer.

Explainer. The explainer has six major modules: a predictor, a postanalyzer, topology graphics, a How module, a Why module, and a What module (see Figure 4). The predictor and the postanalyzer are the two major modules. The topology graphics module manages the graphic structure. The user interface corresponds to three predefined question types — how, why, and what — which are handled by the corresponding modules. The explainer performs a postanalysis-and-prediction cycle through the recorded signal events and can give spontaneous explanations. A user can ask questions at any step of the cycle. All the examples here concern e-sequences in Figure 2, the explanation in Figure 6, and the implemented e-sequence 3 in Figure 8.

Figure 7. The five slots of an e-sequence.

Figure 8. Part of the implementation of e-sequence.3.
The task is to find the e-sequences at the next step. The predictor looks for two successive e-sequences to circumvent the problem of signal loss at one step. The predicted e-sequences at these two steps are then checked. These are only potential e-sequences; they are not systematically presented to the user. The depth of the prediction is a trade-off between minimizing execution time and explaining signal loss (in signals emitted by the relays or circuit breakers).

For all potential e-sequences specified in the downstream slot of the identified e-sequence, the predictor checks all the preconditions except the initial operating-status signal (which might be emitted but not recorded). If the preconditions are not true, the predictor lacks reason to predict the e-sequence, even if the inputs exist.

For example, e-sequence.2 in Figure 6 is successful at time $t_0 + 30$ ms (Line 1 and PX1 are the objects involved). The predictor searches the e-sequences in the output of e-sequence.2 and finds e-sequence.3. It checks the preconditions “fault mono-phased” (from Figure 2) by calling the corresponding predicate “fault phase-?”.

Since the fault is mono-phased, the precondition is true before $t_0 + 30$ ms and the predictor predicts the e-sequence.3 at time $t_0 + 50$ ms. (Indeed, e-sequence.3 is successful at $t_0 + 51$ ms, as shown in Figure 6.)

**Postanalyzer.** The postanalyzer has several tasks. It must identify successful and untimely e-sequences, explain abnormal behavior from the diagnosis, explain failures of predicted e-sequences, and explain how the initial fault was eliminated (structurally isolated).

An e-sequence is successful when all its outputs can be associated with observations. This subtask involves simple pattern matching of the recorded signal events with the outputs. Once an e-sequence is successful, the postanalyzer checks the postconditions and, if necessary, the initial operating status signal. In Figure 6, for example, after $t_0 + 30$ ms the postanalyzer finds that e-sequence.3 is successful by its output DT.PXI.Mono at time $t_0 + 51$ ms (the predicted time is $t_0 + 50$ ms). It then calls the predicate “presence-signal-p” (see Figure 8) to establish the preconditions “monophased authorization signal exists.” The specified time delay is not considered here because there is a signal noise problem. Note that checking by calling the corresponding predicate could cause many important new states to be established.

An untimely e-sequence is one not predicted, often because a controller was activated in an untimely manner or due to a wrong protection zone. Whatever the reason, such a controller will produce an e-sequence according to the protection zone it has found. The identification of an e-sequence is thus quite costly. This task can be simplified by checking only the initial events. Once the untimely e-sequence has been identified, the predictor is solicited to predict the next possible e-sequences. For example, the postanalyzer finds that e-sequence.2 in Figure 6 is untimely at time $t_0 + 30$ ms for PX1 and PX2 in the Line 2 controller. The postanalyzer then calls the predictor to predict the next e-sequence: e-sequence.3 for PX1 and PX2 in the Line 2 controller at time $t_0 + 50$ ms.

To account for potential signal-noise problems, the postanalyzer doesn’t explain the abnormal behavior identified in the diagnosis until after the predicted time of the related e-sequence. For example, the predictor predicts a potential e-sequence.1 for PJB in the Coupler 2 controller (not shown in Figure 6) at time $t_0 + 30$ ms. This e-sequence is not identified at time $t_0 + 30$ ms, and the corresponding abnormal behavior (ab1 in Figure 6) is explained the next time ($t_0 + 51$ ms) other signal events are identified.

Some abnormalities, usually involving the parallel functioning of relays in a controller, can be identified during diagnosis but not during model-based explanation. Their consistency is checked during diagnosis but their functioning is judged separately during explanation. This is the case in Figure 6 for abnormal behavior ab2 for the relays PX1 and PX2 in the Line 1 controller; the behavior is presented under the same conditions as ab1.

A potential e-sequence can be predicted by either the predictor or the How module, but the e-sequence might not be successful if its predicted time has passed. In explaining this, the postanalyzer checks all postconditions and the initial operating status signal. If one of the conditions is not respected, the e-sequence is unsuccessful. (Since this is only a potential e-sequence, the postanalyzer does not tell the user the reason.) If all the conditions are respected, the postanalyzer will tell the user (once the predicted time has passed) that it was incorrect to label the potential e-sequence as unsuccessful. Such a situation can occur because an e-sequence that is not predicted by the model is considered an abnormality. When the e-sequence has been predicted by the How module, the postanalyzer will furnish a delayed Why-Not explanation (explained later).

Once all signal events have been considered, the postanalyzer explains the fault elimination. It checks the isolation of the electrical feeding path between the fault location and other parts of the power network. Whatever the eventual abnormalities in the controllers, a fault is always eliminated by structural isolation. Signals emitted by circuit breakers signifying structural changes are used to calculate the structural isolation.

**What module.** To avoid a tiresome presentation and a long explanation, the postanalyzer does not automatically give users the detailed text of the successful e-sequences. Instead, the What module reveals the successful e-sequence, whether predicted or untimely, at the user’s request. For example, the following is a user’s question at the second “continue” line in Figure 6, and the response:

**What module.** The predicted e-sequences are not shown to the user, but the How module lets the user query them at any step. If a user asks a How question at successive
steps, the How module furnishes different explanations according to the progress of the diagnosis. The e-sequences predicted by the How module might need a delayed explanation. The e-sequences predicted by the How module might need a delayed explanation according to the progress of the diagnosis. The user can repeat the question at the last "continue" line of the control model defines a sequence of events from five primitives (component, inputs, outputs, preconditions, and postconditions) to establish the deep causal account of how the control device functions.

Indeed, we do not claim to store all possible abnormalities in the knowledge base. The weighted voting method can identify more than one fault location, and model-based explanation can identify other possible types of abnormal behavior and distinguish possible fault locations using the following heuristic strategies:

1. Explain the fault location using structural isolation.
2. Minimize the incoherence between the controller's functioning and the fault features.
3. Minimize the set of all identified abnormal behaviors.

Power networks often have different types of controllers, and therefore different control models in different substations. Because it uses a general explanation engine and a uniform knowledge representation for the control model, SEPT can be more easily adapted to large network problems with many substations. During system adaptation, users need only add knowledge to the system, without modifying the explanation engine.

HE TWO OUTPUT STEPS - DIAGNOSIS and explanation — are complementary. The first gives a rapid synthesis based on the recorded signal events, while the second supplies a deeper causal account of the events and their effects. Keuneke also proposes a two-step diagnosis, where the functional representation has been extended to give a causal explanation for a malfunction hypothesis. In our scheme, on the other hand, the representation of the control model defines a sequence of events from five primitives (component, inputs, outputs, preconditions, and postconditions) to establish the deep causal account of how the control device functions.

Indeed, we do not claim to store all possible abnormalities in the knowledge base. The weighted voting method can identify more than one fault location, and model-based explanation can identify other possible types of abnormal behavior and distinguish possible fault locations using the following heuristic strategies:

1. Explain the fault location using structural isolation.
2. Minimize the incoherence between the controller's functioning and the fault features.
3. Minimize the set of all identified abnormal behaviors.

Power networks often have different types of controllers, and therefore different control models in different substations. Because it uses a general explanation engine and a uniform knowledge representation for the control model, SEPT can be more easily adapted to large network problems with many substations. During system adaptation, users need only add knowledge to the system, without modifying the explanation engine.

Acknowledgment
This research has been supported by a contract with Electricité de France.

References

Building intelligent robotic systems that can reason while functioning in unstructured environments is a challenging task, and intelligent, self-sufficient inference systems are essential if robots are to operate continuously in unpredictable environments. This book provides a basic foundation in autonomous mobile robotics—a crucial topic in artificial intelligence. It explores the successful deployment of industrial tele-operated and reprogrammable robots as well as more exacting scientific applications in remote, unstructured, and hazardous environments.

The tutorial provides a perspective on mobile robotics as a whole. This volume discusses topics in research areas such as sensor-based knowledge acquisition, problem solving under uncertainty, multi-sensor interpretation and integration, path planning and navigation, advanced parallel computing, and machine learning.

552 PAGES. DECEMBER 1991. HARDBOUND & TN O-8186-9018-6
CATALOG # 2018 - $80.00 MEMBERS $55.00

AUTONOMOUS MOBILE ROBOTS: Control, Planning, and Architecture — Volume 2
edited by S. S. Iyengar and Alberto Elfes

This companion text describes the navigational capabilities required to enable autonomous machines to operate within unstructured environments and presents many technically challenging problems. Volume 2 discusses potential applications that have further motivated advances in robotic component technologies, particularly in the areas of sensing, planning and control.

Autonomous Mobile Robots: Control, Planning, and Architecture covers, robot design and control, task-level planning, reactive planning, dynamic planning, prior planning for precompiled plans, real-world modeling, sensor control and global monitoring, and control of robotic systems. This tutorial also investigates wheeled mobile robots modeling and control, kinematic and dynamic modeling, control issues, system-level decision making and control, and AI approaches.

CATALOG # 2116 — $80.00 MEMBERS $55.00

ORDER TODAY!

1-800-CS-BOOKS
or FAX (714) 821-4010
in California call (714) 821 8380

IEEE COMPUTER SOCIETY PRESS TITLES

Duong-Yih Bau is a computer science engineer at Taipower in Taiwan. His research interests include the diagnosis of control systems, and explanation. Bau earned an BSc in nuclear engineering from National Tsing Hua University in Taiwan in 1977, an MS in computer science from the Asian Institute of Technology in 1986, and a PhD in computer science from the University of Paris XI in 1991.

Readers can reach Bau at No. 29, Lane 259, Ti-Cheng Road, Peitou, Changhua Hsien, Taiwan.

Patrick J. Brézillon is a researcher at the National Center of Scientific Research (CNRS) in France. His research interests include new architectures for knowledge-based systems to deal with such tasks as diagnosis, explanation, knowledge acquisition, and automatic learning. Brézillon received a BS in mathematics, an MS in applied mathematics, and a PhD in physical sciences, all from the University of Paris VI.

Readers can reach Brézillon at Laboria, Box 169, University of Paris VI, 4 Place Jussieu, F-