# 2. Intelligent Control

## 2.1 Techniques in Intelligent Control

Intelligent Control (IC) was originally proposed by Fu [1971], to increase the flexibility and extend the range of then current automatic control systems. The approach used techniques from fields of artificial intelligence, operational research, and dynamic control to serve, reason, plan and act in an "intelligent" or "smart" manner [Saridis and Valavanis, 1988]. This list can be augmented to include computer science, as advanced concepts are being employed to manage the overall system complexity and its command and control infrastructure.

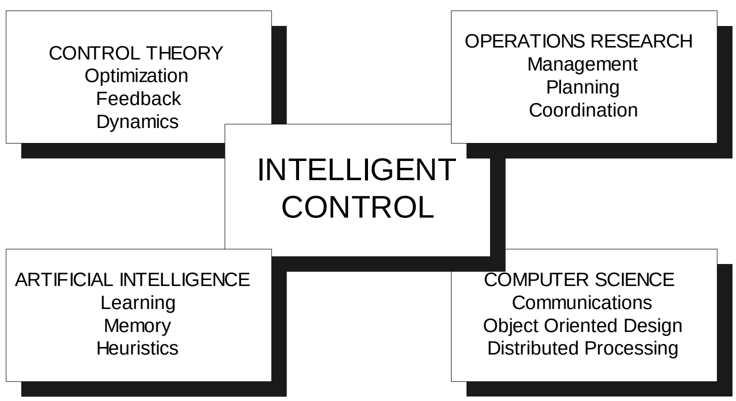


Fig. 2.1. Techniques in intelligent control.

IC systems are not defined in terms of specific algorithms; they employ techniques which can sense and reason about their environment and execute commands in a flexible and robust manner [Antsaklis and Passino, 1992]. The technological drive for autonomy in many complex systems has motivated research into various aspects of IC: system architectures, learning control, sensory processing and data fusion, world modeling, etc. [Albus, 1991]**.**

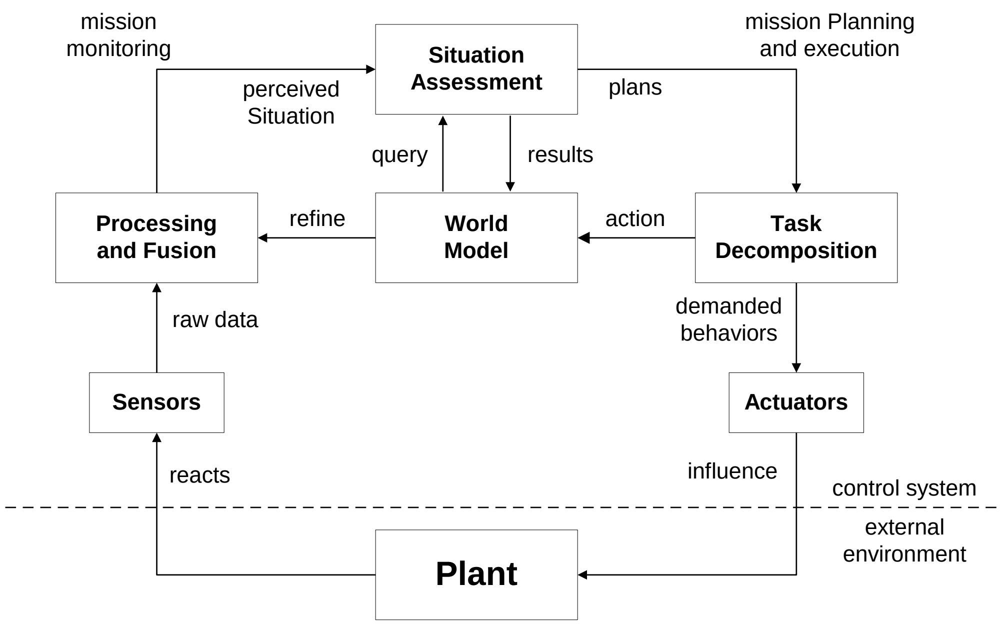


Fig. 2.2. Various components for intelligent behavior.

## 2.2 Control System Architectures

Central to an overall IC systems design is the architecture which determines how complexity is managed and which modules are necessary for implementing the desired behavior, as well as specifying the command and control infrastructure necessary to link and manage the distributed processes efficiently. The two important components in any system architecture are the hierarchical functional decomposition of the problem into simpler subtasks, and the command and control management infrastructure which sends messages both vertically and horizontally to neighboring submodules.

### 2.2.1 Hierarchical Control Architectures

Hierarchical control architectures can be used to implement systems which require separate sensing, planning and execution phases, and to resolve complexity at various levels of understanding [Saridis, 1989]. Those systems are designed according to the principle of *increasing intelligence with decreasing precision*. The highest levels in control hierarchy use intelligent reasoning strategies to understand the processed data. As the information moves down the control structure, the information and processing algorithms become less intelligent and more precise.

The exact algorithmic decomposition is problem dependent, although it generally consists of global reasoning routines, local planning algorithms and low level coordination and control techniques. Each module in the control hierarchy has just enough resources (access to data and functions) to perform its task, and knowledge is stored in a distributed manner throughout the system.

### 2.2.2 Subsumption Control Architectures

Subsumption control system architectures are based on entirely *reactive* behavior rather than the traditional *sense*, *plan*, *action* cycle associated with hierarchical architectures [Brooks, 1986]. It is based on the principle that complex (and useful) behavior can arise from the collective actions of many simple subtasks. Each processing unit implements a subtask such as *wander*, *explore* or *avoid objects*, etc., and each unit receives sensor signals directly and sends commands to the actuators. The overall behavior is than a composite of the individual subtasks. The sensory processing algorithms are tightly coupled with the functions that send commands to the actuators and this is significantly different from hierarchical control systems where the sensory processing and control elements are separated [Brooks, 1990]. New behaviors can be easily introduced as the perceptron and navigation routines form part of the same task. However, this also means that the sensory processing may be redundant, with different modules performing the same data fusion techniques.

### 2.2.3 Model Reference Architectures

Both of these architectures have desirable features. The hierarchical structure allows each module to be assigned a unique task and object oriented programming techniques can be used to implement such a structure efficiently. Subsumption systems use the sensory data to influence the commands issued to the actuators directly, and complex behaviors are observed from the combination of simple tasks. The combination of these two philosophies forms an alternative system representation scheme; the approach of Albus with various model reference system architectures [Albus, 1991, 1992; Abus et al., 1990]. The system is decomposed vertically into various levels of abstraction and reasoning, and horizontally into sensory processing, world modeling and task decomposition modules.

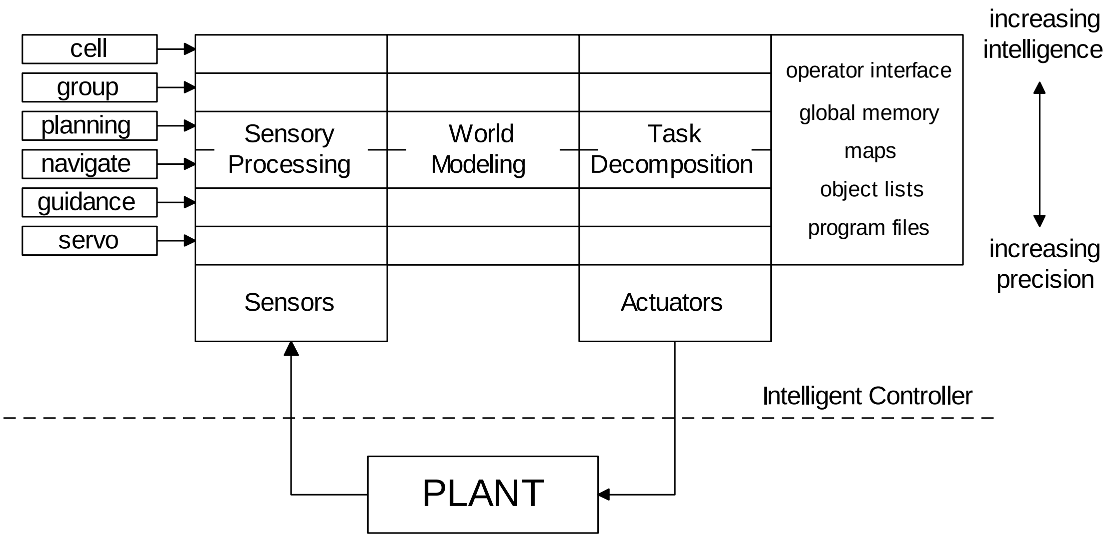


Fig. 2.3. Model reference system architecture.

In order to increase both the flexibility of current autonomous systems and their rate of development, the system is designed to satisfy the following requirements:

- <u>extensibility</u>, both functional and temporal;
- a <u>flexible</u> human/computer interface;
- <u>real-time</u> operation;
- <u>distributed</u> systems which support graceful degradation; and
- <u>application-independent</u> development.

This architecture provides a convenient framework for the conceptual development of Intelligent Autonomous Systems (IAS) [Corfield, *et al*, 1991]. IASs must operate in hazardous, ill-defined, time-varying environments and complete their assigned tasks safely. They use algorithms which can learn from their interaction with the environment, resolve ambiguous and uncertain situations and operate in a fail-safe fashion [IEEE Cont. Sys. Mag., 1993, IEEE Expert, 1991].