Summary of Subsumption Architecture Research Article

1. Introduction (Vinasirajan Viruthshan - Pg 2 - 5)

Because of the rapid changes in the boundary conditions sequence architecture is not suitable for real time reactions. There are some important factors that a robot control system should have.

Out of multiple goals, the robot should be able to identify higher priority goals according to the situations. Robot should be able to collect the data from multiple sensors and it should be able to make decisions even if the readings are not that accurate. Even if some sensors fail, it should be able to continue its process without losing the process. Adequate power should be given for the robot to work with multiple sensors.

There were some other approaches which were made to make the control systems better. By letting the user to code for parallelism and to code for exception paths whenever the robot finds any abnormal instances. Usage of multiple sensors in real time, will allow one to get a clear understanding about the environment even if a sensor falsely detects something. For an example sonar sensor and IR sensor can be used to measure the distance, but both might give different accuracy reading for different instances. To add more additional functions to the robot, more processing power will be needed. By utilising the existing power of processors, or processors can be upgraded to be faster systems, or else by adding more processors, robot will be able to perform fast even after adding new functions.

It doesn't mean that complex control system should have complex behaviour. System should be kept simple. If the components' complexity increase, proper ways should be followed to reduce the complexity. Robots should be able to work with its' independent decisions without human intervention. Robot must model the world in 3- dimension. Relational maps should be used instead of absolute coordinate systems.

The physical robot has measurements like 17 inches diameter and about 30 inches from the ground. Driver mechanism of Real-world Interface of Sudbury and Massachusetts Three parallel drive wheels were connected. Motors are controlled by single microprocessor. Body is connected with steering and always point the wheels direction. a ring of twelve Polaroid sonar time of flight range sensors oriented symmetrically around the body. two Sony CCD cameras connected at the head. There is a plan to add feelers. The main intel 8031 processor communicate with offboard processor. The radios are modified Motorola digital voice encryption units.

2. Levels and Layers(Namina Wijetunga - Pg 6 - 9)

2.1 Level of Competence

Most engineering problems are solved by breaking down the problem into several subproblems and solving each subproblem. Mobile robot builders have broken the problems into several parts namely sensing, mapping sensor data into a world representation, planning, task execution and motor control. But this is considered as a horizontal decomposition of a problem into vertical slices. The slices form a chain through which the

information flows from the robot's environment through the robot and back to the environment via action.

Instead of this approach, the researcher suggests decomposing the problem vertically as the primary way of slicing the problem. The problem is sliced based on desired external manifestations of the robot control system.

To implement this, a number of **levels of competence** (which are known as an informal specification of a desired class of behaviours for a robot over all environments it will encounter) will be defined for the robot.

Higher level of competence states that it has a more specific desired class of behaviour and it has control over the bottom levels. They have additional constraints on that class.

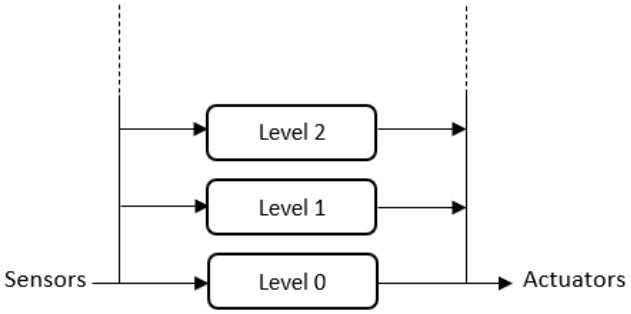


Figure 1

2.2 Layers of Control

In subsumption architecture, the key idea is constructing layers for the control system and adding a new layer on top of the existing layer to move to the next higher level of overall competence. A complete robot control system is started to build which achieves the level 0 competence and it is tested and debugged thoroughly. Without altering the system, the next layer is built on top of the 0^{th} layer and it can access the data of level 0. With the aid of level 0, this level achieves the level 1 competence. Level 0 is unaware that there is a level above it. A working robot is there once level 0 is finished and additional layers can be added layer to improve the robot.

Here, individual layers can be working on individual goals separately and parallelly. The key feature here is no need to take early decisions on which goal should be achieved first. Therefore, this architecture has **multiple goals.** The subsumption architecture has **multiple sensors**. Looking at the sensors, the sensor fusion problem can be ignored here as all the sensors are not necessary to feed into the central representation. But the robot may use the sensor values at the same time. Other layers may process them in their own ways to achieve their own goals.

This architecture is said to be **robust** because of 2 reasons. Since it has multiple sensors, if their results can be used, the system is robust. The lower levels are well debugged, and they continue to run when higher levels are added. In case of a failure of a higher level, still, the lower levels continue to run, and some functionality is still happening in the robot. As the fourth feature, this architecture has **additivity**. It is handled by making each new layer

running on its own processor. Also, each layer can be spread easily over many loosely coupled processors.

2.3 Structure of Layers

When decomposing each layer, the freedom is given to use different decompositions for different sensors – set, task – set pairs based on the intentions of the problem.

3. A Robot Control System Specification Language (Saeedha Nazar - Pg. 9-11)

Internal structure of modules and the way in which they communicate are two primary aspects of the components in this architecture.

3.1 Finite State Machines

Each module is a finite state machine augmented with some instance variables, which can hold Lisp data structures.

Each module consists of a number of input and output lines. The most recently arrived message is always available for inspection while messages can be lost if they newly arrive on an input line before the last was inspected. There is a distinguished input to each module called reset.

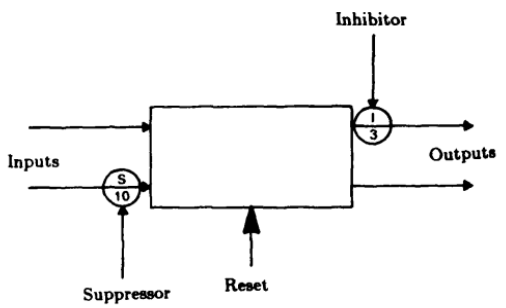
Initially, all modules start in the distinguished state named NIL. When a signal is received on the reset line the module switches to state NIL. A state can be specified into four types:

1. **Output** - *Computed as a result of the function of the module's input buffers and instance variables. A new specified state is then entered.*
2. **Side effect** - *One of the module's instance variables is set to a new value computed as a function of its input buffers and variables. A new specified state is then entered.*
3. **Conditional dispatch** - *A predicate on the module's instance variables and input buffers is computed and depending on the outcome one of two subsequent states is entered.*
4. **Event dispatch** - *A sequence of pairs of conditions and states to branch to are monitored until one of the events is true. The events are in combinations of arrivals of messages on input lines and the expiration of time delays.*

3.2 Communication

A visual representation of using finite state modules for the purposes of communications is shown below.

- Input/output line connect modules. These lines can be thought as



wires, each with sources and a destination.
- Outputs may be inhibited and inputs may be suppressed.
- For both suppression and inhibition we write the time constants inside the circle. In our specification language we write wires as a source (i.e. an output line) followed by a number of destinations (i.e. input lines).

4 . A Robot Control System Interface (Rajkumar Kishokkumar - pg. 11-15)

4.1 Zeroth Level

Zeroth level prevent the robot from contact with other objects. If something approaches the robot it will move away. It will halt when it is going to collide with object. For that this level has some modules. Motor module will maintain the communication with physical robot and accept the motion commands. Sonar module will create the map of obstacles. The collide module sends a signal according to the sonar module. The feel force module generates a signal resultant force. The runway module monitors the force and sends command to motor module if it is important.

4.2 First Level

In this layer robot directed to go anywhere aimlessly without hitting obstacles. The wander module generates a new heading for the robot every 10 seconds. The avoid module gets the result of force from the zeroth level and combine it with heading and produce a modified heading. Its output suppresses the out from runaway module and enter the motor module.

4.3 Second Level

level two gives exploratory mode behaviour to the robot according to the visual observations. In this layer, the grabber module makes sure that level 2 has control of motors and temporarily inhibiting a number of paths in lower level. During the inhibiting time, sensors will give plan to pathplan module. The monitor module monitors the motor module compute the position relative to initial position. Th integrate module gather reports from monitor module and sends recent result to integral line. The pathplan module handle the goal specifications and attempt to reach the goal. If the position of the robot is close to the desired position its output sends to the straighten module. the straighten module responsible to the final orientation of the robot. It sends the commands directly to the motor module. Monitors and integral make sure it.

The zeroth level and first level operations are happening during normal operations of second level.

5. Performance (Selani Indrapala - pg. 15-19)

When writing this paper, the physical robot and communication links were not complete enough to support tests. Therefore, only layers 0,1 and part of layer 2 were tested.

5.1 A simulated Robot

The simulation accounts for all the errors and uncertainties in the real world. They took into consideration slight deviations in motion and thermal and humidity effects. The simulation runs off a clock which runs at the same rate as the actual robot. The simulation runs in realtime and also drives graphic displays of robot state and module performance monitors.

5.2 Zeroth and First Level

Only the first level control system was connected. The robot wandered aimlessly without colliding with any obstacles.

5.3 Second Level

The second level control system was connected with 2 goals fed to the goal line of the grabber module. In trying to achieve the goals the lower level wandering was suppressed. The goals weren't exactly achieved and the turn and forward motions had a 5% error. As soon as the goal was achieved satisfactorily, the robot started wandering again.

6. Implementation Issues(Selani Indrapala)

The objective for developing the layered control system was additivity and processing power. Since it is decomposed into asynchronous processors with low bandwidth communication and no shared memory assists in achieving this objective. However, the modules have access to Lisp programs which require additional computational power.

6.1 A spatial processor

To carry out the hard Lisp computations of any given module, approximately 60 trivial processors, with a one bit alu each would be sufficient. This network could easily be fitted in a single chip.

6.2 Sizing The Processors

16 states (8 according to the processors presented in this paper) would be enough for all modules. However, in this implementation, a higher number of simpler states are used. These can be easily put in the corner of an array processor chip or many could be packed on a single chip.

6.3 Short Term Approaches

So far a single Lisp machine has been enough for simulations. However, when bringing on the vision work, that may ot be enough. The long term goal will be to use custom chips. However, in the short term, connection machine Hillis 85, NEWS network and cross omega network can be used for large scale simulation, communication within modules and between module communication respectively.