A summary on 'A robust layered control system for a mobile robot' by R. A. Brooks(1985), AI Laboratory, MIT

Thalagala B.P. 180631J

April 30, 2020

When implementing an autonomous intelligent mobile robot there are several requirements to be fulfilled by the control system of the robot and there are different approaches to fulfill those requirements.

Requirements of a control system[1],

1. A Robot Probably has ***multiple goals*** which may have varying priorities depending on the context at the moment, therefore control system must be able to identify the situation of environment and give the priority to what is necessary while providing required services to the low level objectives.

2. A robot has ***multiple sensors*** which in nature has some error in their readings comparison to the real physical quantities that are measured. Therefore control system must be able to model the environment as much as accurate by using available sensor data wisely.

3. The robot must have some ***robustness***. For that the control system must be able to drive the robot with some performance by using available sensor data even though one or more sensors are not in proper working condition.

4. The processing power of the control system must be able to be increased in case of adding more components to the robot in order to increase its capabilities . This feature is known as ***additivity*** of the control system.

The underlying process of most of the approaches in fulfilling these requirements has three basic steps.

Problem is decomposed into several sub-problems
↓
Sub-problems are solved
↓
Solutions to sub-problems are composed together to make up the final solution

Depending on the way of decomposing the problem(at the first step), the architecture of control system of the mobile robot differs. The conventional method is known as *Sense-Plan-Act (SPA) Architecture*. Here the problem is decomposed into functional modules depending on the internal workings of the solution.This is regarded as a horizontal decomposition of the problem into vertical slices[1].

The overall function of this kind of control system is, taking information from the environment using sensors, planing the actions need to be taken and finally executing the planned tasks using actuators.The sub problems of SPA architecture can be shown as follows.

Sensors → | Perception | Modeling | Planning | Task Execution | Motor Control | → Actuators

Figure 1: Sense-Plan-Act (SPA) Architecture

All the sub-problems need to be solved to a certain extent and composed together essentially, before testing a mobile robot designed under this category. Implementing a solution for only one sub-problem is not sufficient to operate the robot with any amount of performance.

In addition to that changes in control system cannot be done considering only one sub-problem(a piece in above figure) alone. That is if some change in a piece is needed, either we have to keep the interfaces to adjacent pieces unaltered or do the necessary changes of the functions in adjacent pieces.

Note: *From this point onward most of the information will be directly related to the mobile robot made by Rodney A. Brooks in 1985, at Artificial Intelligence Laboratory, MIT. Therefore not general.*

Subsumption Architecture

In Subsumption Architecture[1] the problem is decomposed based on the desired external actions of the mobile robot.This is regarded as a vertical decomposition of the problem into horizontal slices[1].

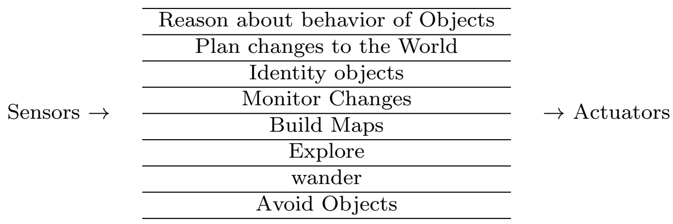

Figure 2: Subsumption Architecture

To understand this architecture there are some concepts to be identified at first,

Levels of competence

A level of competence describes a specific set of behaviors which the robot should have over all the environments that it will deal with.When the level of competence increases even more specific behaviors and additional constraints are added to get the desired final robot behavior. This can be viewed as follows and notice that a particular lower level competence is included in the immediate higher level.

Level	Behavior
0	Avoid contact with objects
1	Move leisurely in the environment without hitting objects
2	Explore the reachable environment
3	Build maps of the environment and plan routes from one place to another
4	Become aware of the changes in *'static'* environment
5	Understand objects in current environment and perform task related to specified objects
6	Create and execute plans
7	Understand behavior of objects in environment and modify plans accordingly

Layers of Control

In this subsumption architecture[1], sub-control systems can be built corresponding to each level of competence and they are able to operate individually. These sub-control systems are called as control layers (zeroth level, first level, ...).

Particular lower layer is not aware about its upper layer which sometimes inject necessary data to the lower layer to limit the normal flow of data in lower layer to achieve desired robot behavior. And the level of competence can be increased to the next level by adding a new control layer on top of the existing layers.

The most interesting thing about this architecture is that it naturally takes care of the requirements which were discussed at the beginning, that an autonomous mobile robot control system should fulfill.

1. ***Multiple goals***: As layers are able to work independently on multiple goals at the same time, there is no need of specifying what goal should be pursued by the robot since every goal is being pursued to a certain extent at a given moment and information given by all the layers can be used to take one final decision to be executed.

2. ***Multiple Sensors***: Since data form all sensors are available for each control layer and it is allowed to process the data in the way it wants to achieve its goal, there is no need of sending all the sensor data to a central processor to process.

3. ***Robustness***: Since a higher control layer is added only after its immediate lower layer is well debugged and tested in the environment, although there is any failure of operation in one layer, all the layers below that layer are continue to work, with the highest level of competence which corresponds to the working layer. Therefore robustness is naturally there in this architecture.

4. ***Aditivity***: Through allocating a processor for each control layer, the processing power of the robot can be increased whenever we want. All we have to do is when we adding a new higher level control layer use a dedicated processor for that layer.

Structure of Layers

In this architecture we do not need to consider all the inputs, all the objectives and all the behaviors within a single decomposition. Because we can decompose each layer in a different way to take out the maximum performance. In addition to that each layer is made up of several small asynchronous processors(*refer as modules*) which sends and receives messages from other modules, without any central control or any global memory. Moreover Inputs and outputs of the modules can be limited by other modules by sending messages. (*This is the main concept behind including lower layers inside the higher layers.*)

Robot Control System Instance for Subsumption Architecture

Zeroth Layer

Zeroth or the lowest layer of the mobile robot control system is designed to achieve *level zero competence*. That is this control layer makes sure that the robot does not make any contact with objects in the environment either moving or stationary.

I. If it is a stationary object and the robot is going to hit it, modules command the robot to stop where it is.

II. If some object is moving towards the robot, modules command the robot to move away from that object.

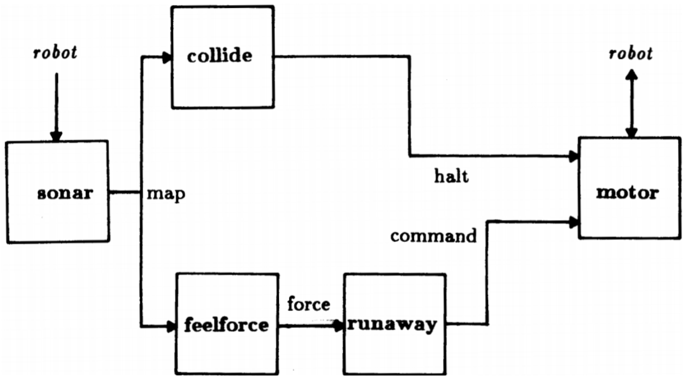

Figure 3: Modules and their interconnection in the zeroth layer[1]

Purpose of each module briefly, can be given as follows.

- **Sonar Module** has three sub functions as, taking sonar readings, check their validity and filter out invalid readings and build a robot centered map in polar coordinate system.
- **Collide Module** inspects the map produced by the sonar module and if a stationary object is detected which is on the way of robot, it sends a message through the halt line to the motor module.
- **Motor Module** accepts commands related to the motion of robot(for angular motion - angle and angular velocity(speed of turn)/for linear motion - distance to be traveled and velocity) and converts them into electrical signals which can then be executed by actuators. If a halt signal receives, it will be prioritized over any other received signals.
- **Feelforce Module** also inspects the map produced by the sonar module and associates(maps) every detected object to a corresponding repulsive (drive back) force and produce a resultant force considering every such force.

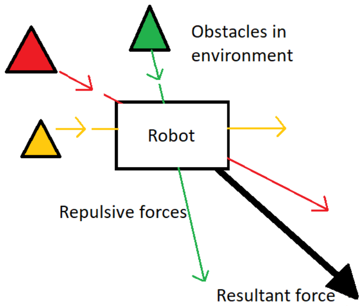

Figure 4: Determination of feelforce

- **Runaway Module** observes the resultant repulsive force produced by the feelforce module and sends necessary commands to the motor module.

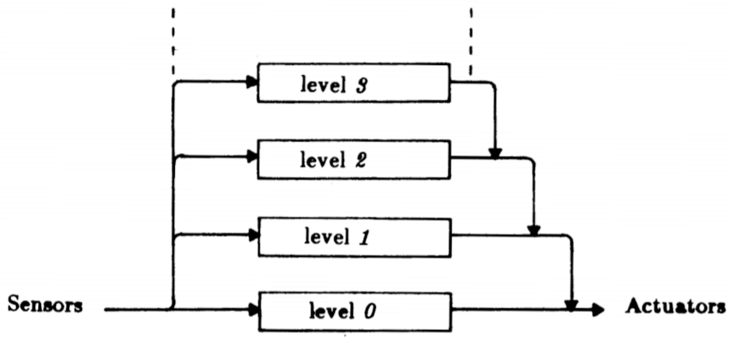

Figure 5: Layered Control System[1]

After building the zeroth level of the control system, which was made to achieve level zero competence, the robot is tested in the environment and then it is fully debugged. Then that layer in the control system will never be altered. All the other higher level layers are built on top of this layer. Higher layers are allowed to take data from lower layers and also permitted to inject necessary information to the lower layer modules. This can be represented as follows.

First layer

First layer with the aid of the zeroth layer achieves the *level one competence*.That is these layers together make sure that the robot moves leisurely without making any contact with either stationary or moving objects in the environment.

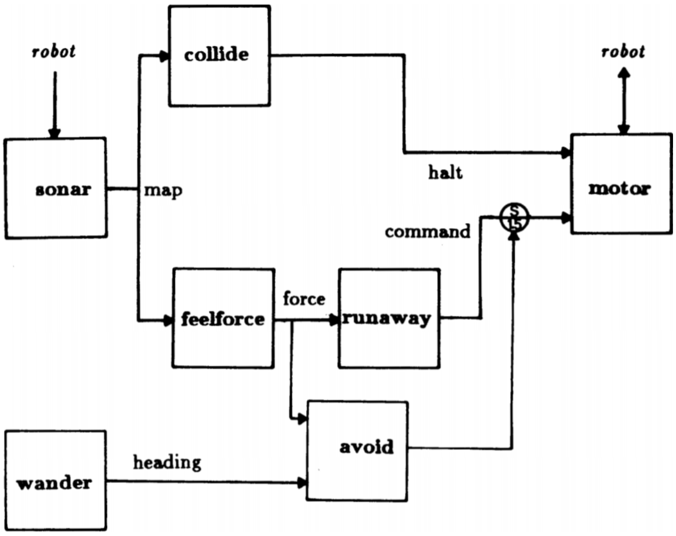

Figure 6: Interconnection between first layer modules with existing zeroth layer[1]

- **Wander Module** randomly chooses a direction repeatedly at every given amount of time to move.
- **Avoid Module** takes (1)the *suggested direction* from the wander module and (2)direction of the *resultant repulsive force* from the feelforce module in the zeroth layer, into account and then decides a direction to move. This new direction is then injected to the output line of the runaway by suppressing the normal data flow of that module.

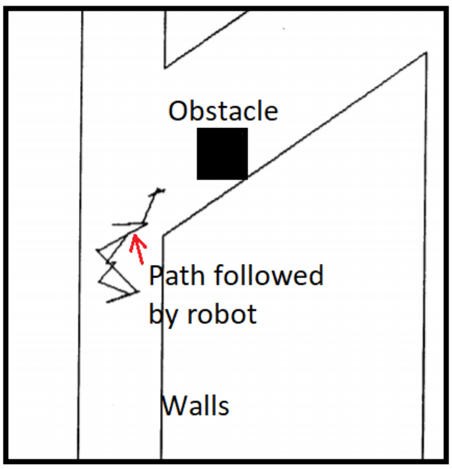

Figure 7: The path followed by the robot after implementing layer one and zero of the control system with level one competence(*a simulation result*)[1]

Conclusions

- Subsumption architecture introduces a way of decomposing the problem depending on the desired robot's behavior whereas in SPA architecture it is done by considering the internal workings(sensing/Modeling/planning/...) of the control system of the mobile robot.
- This architecture allows us to build complex robot control systems step by step, so there is no need of considering the whole problem at once to initiate the building of control system.
- Control of the system is not centralized and it is divided among several asynchronous processors which try to do their best individually so that all the goals are pursued to a certain extent.

Bibliography

[1] R. A. Brooks. A robust layered control system for a mobile robot. *Artificial Intelligent Memo-864,Artificial Intelligence Lab,Massachusetts Institute of Technology*, September 1985.