# Product Specification & Design

## (a) Overview of the Project

#### Project Aim

**The AVINSoR Project**[[1]](#footnote-1) aims to demonstrate the ability of an embedded system to perform autonomous *exploratory* navigation in a miniature AGV (Autonomous Ground Vehicle)[[2]](#footnote-2). The *perception* and *behaviour* involved in the autonomous navigation process should be *defined* (created) and *refined* (adjusted) through a supervisory *training process* – which would involve some form of input from a human ‘trainer’ (essentially this process should be like training a pet, until the pet learns how to behave appropriately on its own).

#### Defining Autonomous Navigation

If navigation is defined as the ability of a *navigator* to: (1) know ones position in the current moment, (2) have an accurate idea of the destination location, (3) to plot a route from the current location to the destination, and (4) to follow the route; then *autonomous navigation* would be the ability to do everything mentioned earlier *without* any external input; apart from*,* perhaps, the *desired destination –* to ultimately command the autonomous navigation system to plan and follow a route to a particular desired destination. However, even this type of input would be unnecessary in exploratory systems (such as AVINSoR) after the *training process* - which aims to define a set of *features of interest* and the appropriate behaviour(s) associated with them.

#### Machine Learning

Essentially, the system will utilize a number of machine learning methods (via digital logic circuitry, and executed algorithms).

## (b) Technical Specification Summary

The initial concept of the AVINSoR project is illustrated in its abstract systems block diagram form in **Figure 1**. The concept involves:

1. The ability of a “Visual Perception” system to *define* and *refine* *feature vectors* of objects from sensory inputs (from both, the *main* visual sensor, and the *optional* auxiliary sensors).
2. The ability of a “Behavioural Patterns Memory” to *learn* and remember digital output patterns as a function of time (and perhaps other factors).
3. A “Cognition” system remembers the link(s) between selected feature vectors and sequence of selected appropriate behaviour to execute.

All of the above systems require supervisory *training* commands to control the process of ‘learning’ via *defining* and *refining* feature patterns, feature vectors, output functions and the links between them.



|  |
| --- |
| ***Fig 1.*** *Diagram illustrating the primary concept of the AVINSoR project as hardware sub-systems.* |

#### Communication Link with the Human Supervisor’s PC

The most obvious deduction from the diagram is the wireless communication medium between the robot and the human supervisor/trainer’s PC, made possible through a transceiver on both sides. Though the exact wireless communication medium is yet to be defined (IEEE 802.11 WiFi, Radio, etc.), this would most likely be implemented by a Wi-Fi transceiver module on both sides.

#### Hardware: Robot-Side

**The *entire* system illustrated within the robot block would be prototyped using multiple development boards (MCU and/or FPGA based) throughout the course of the project.** Unfortunately, the project timescale does not allow for the development of an entirely new embedded framework from scratch, nor the design of a *final* single-board embedded-system prototype PCB (which would be the ideal deliverable *if* the project timeline was larger, and the budget greater).

##### Implementation of Supervised Machine-Learning Logic

**As the key suggests, the system(s) which emulate visual perception, cognition and behavioural patterns are, as of yet, abstract** - as neither the exact logical/mathematical model nor the exact hardware approach has been precisely defined (a greater depth of research is required, as well as iterative simulations and iterative practical experimentation). The approach could use either a processor to execute the necessary machine-learning algorithms, a set of custom-defined logic (implemented using an FPGA), or both.

##### Digital IO Ports

**The digital input from the camera** module is a specialized input channel dedicated to input from a specific imaging device. It would require a specific hardware port compatible with the chosen module (e.g. USB). There may also be a need to upgrade to a stereo camera module to implement the execution of more advanced algorithms in the future, so compatibility must be considered.

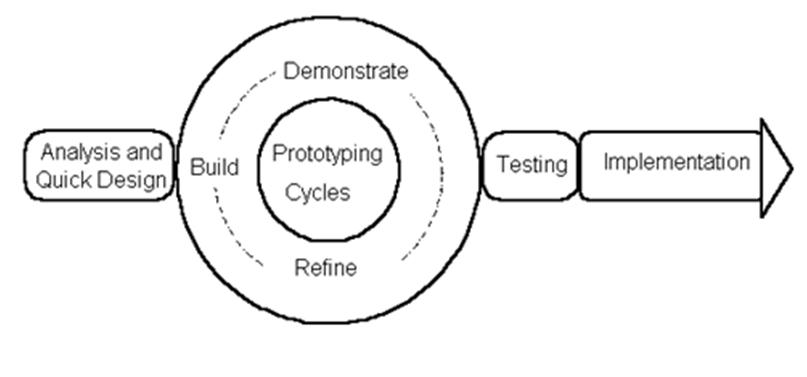
**The digital input of Supervisory ‘Training’ Commands** via the wireless communication link with the PC should not be considered a physical port as such, but rather, a *definition* of the input received.

**The digital outputs to actuators** are essentially an array of (typically 5v) digital signals which can be interfaced with controller modules for servos, dc motors, etc.

**The digital input from auxiliary sensors** are essentially an array of (typically 5v) digital inputs for external digital sensor modules or (if necessary) encapsulating circuitry.

## (c) Project Design Metrics

If the AVINSoR project extended beyond the 28-week lifeline, the expectation of the outcomes of the project would be rather different to what they are now. Ideally, the final design of a system with such aims (mentioned above) would be the development of an entirely new adaptable remote and versatile hardware framework which improves efficiency of the machine learning process through the implementation of custom logic (and auxiliary circuitry). In a 24-week project, the design of a hardware framework is *not* *a realistic possibility*, let alone a working prototype. The specified timeframe only allows for the design, simulation/execution and testing of the machine learning algorithms and of course, the auxiliary software (video frame capture, image enhancement, GUI) necessary for demonstration on a miniature AGV platform. It is also worth noting that the project plan takes a *spiral (iterative) development* approach (specifically *the rapid prototyping* model, with a greater emphasis on design/refine stage due to level of research and experimentation required) rather than the traditional sequential waterfall modelled approach.

 ***Fig 2.*** *Diagram illustrating The Rapid Prototyping Model of product development,*

The design metrics will be proposed baring this mind; and hence, the proposed metrics would be appropriated towards software design:

* **Usability**
  + AVINSoR design process should involve some form of Usability Metric in order to quantify the ease-of-use of AVINSoR’s User Interface(s). This would allow for an overall improvement in User Interface(s) – i.e. *enhanced* user experience, every time the current version is revised. This would mean that the level of training and/or documentation required to teach the user how to use the product/solution would be reduced, and the users – who are ultimately trying to implement the system in a robotic solution to allow the robot to be controlled via supervision and ultimately behave in a certain manner when certain objects or features in the environment are sensed – will be able to do so quicker (considering that other factors such as *effectiveness of the software algorithms* are not compromised and continue to improve - another metric?).
* **Reliability** 
  + Reliability covers a wide range of specifics with regards to software design. The metric of reliability – as a generalized measurement, would allow the management to maintain and improve on the performance quality of the product and ensure that efforts are made to improve robustness and dependability with every design iteration.
    - **Reliability of the algorithm:**
      * The consistency of the machine-learning algorithms to reproduce the same object/feature perception result *once* trained *for a particular object/feature.*
      * The consistency of the system to produce the same actuator output once *trained* to do so.
      * The consistency of the *cognition system* to initiate appropriate actuator behaviour for the specified feature vector after *training*.
    - **Reliability of the software operations**
      * All aspects of the robot-side software should work effectively when the robot-side CPU/processor(s) are being used intensively (it is not clear at this stage whether processing needs to be decentralized for certain aspects of the design or not). The quantification model used to ensure this may also take into the account the *resource-intensiveness[[3]](#footnote-3)* of each piece of software, so both metrics can be improved (or a separate quantification model could be used).
      * **If** the solution is designed to be used on a range of different development boards / embedded platforms (i.e. BeagleBoard, Raspberry Pi) for purposes of compatibility; then it may be a good idea to quantify the *similarity of performance* on the variance platforms. So the *dependability* of the system is consistent amongst the various platforms that it may target.
* **Speed (Execution and Response)**
  + As the system is
* **Maintainability**
  + The system should be maintainable to allow for versatility (adoption of the code for future problems),
  + The metrics associated with maintainability will primarily concern the quality of Object Orientation; hence, this is likely to involve the quantification of: class coupling and cohesion, class structure and design, quality of UML diagrams.
  + This would ensure that as the system gets more and more complicated during the *cyclic* process of *product development.*
* **Power Consumption**
  + As the system is designed to operate on a remote vehicle with a battery-pack, power consumption should be quantified and improved upon with every revision. As the number of design iterations increase, it is likely that the system (software solution) would grow in complexity; defining a *metric* would allow for the design to be conscious of power consumption with every revision (where as defining power consumption as a *constraint* would not allow for such

## (d) Primary Belbin Role

I believe that my primary Belbin team role would be that of a Complete Finisher; I am adamant on perfection of tasks I choose to undertake – especially with regards to elegance, structure, hierarchical arrangement and overall aesthetics of the hardware or software solution. Sometimes this means that important deadlines can be overlooked and that .

## (e) Importance of Team Dynamics in a Design Environment

# Manufacturing Technologies

## Objectives of the PCB Design Process

**The**

1. AVINSoR is an acronym for *A* *V*isually *I*ntelligent *N*avigation *S*ystem f*o*r Application in *R*obotics. [↑](#footnote-ref-1)
2. AGV is an acronym for an *A*utonomous *G*round *V*ehicle – i.e. a vehicle designed to operate on ground, which implements autonomous navigation/control. A miniature AGV is an AGV which is smaller and lighter than AGVs modelled around life-size human-operated vehicles (i.e. car, tank, etc.), may also be denoted [↑](#footnote-ref-2)
3. The term “*resource-intensiveness”* is used specifically with regards [↑](#footnote-ref-3)