**University of Kent**

**AVINSoR** (**A** **V**isually **I**ntelligent **N**avigation **S**ystem f**o**r application in **R**obotics)

INTERIM REPORT

**Gurtej Singh Birring**

gsb4@kent.ac.uk

**Dr. Peter Lee**

Date of Submission: 19/12/2014

# Introduction

## Background

The AVINSoR project can be considered an *experimental* project with the underlying philosophy emphasizing the design of a *biologically-inspired intelligent autonomous navigation system*. The idea stems from interest in the study of human and animal cognition, particularly with regards to:

1. **Learning, Problem-Solving, and Decision-Making.**
2. **Visual, Sensory and Spatial Perception.**
3. Logical/mathematical *models* and *simulations* of the above faculties/processes/phenomena.
4. *Algorithms*, *hardware and software implementations* that allow the above processes/phenomena to occur in digital electronics and computer systems.

It may be possible that a human-trainable autonomous navigation system can be developed that entirely eliminates the need to program the way the system perceives sensory data and responds to it (through decision-making and actuator control), and that the perceptions and behaviors of the navigation system can be taught, refined or completely erased through the commands of a human user (who will be referred to as the ‘*trainer-user’* throughout this report).

Figure 1. Hierarchical diagram illustrating the different subject areas of background research required for abstract system design.

### Autonomous Navigation

**Navigation** is the ability to [1]:

1. Know ones position in the current moment.
2. Have an accurate idea (*spatial*) of the desired destination.
3. Plot a route from the current location to the destination.
4. Follow that route.

Hence, ***autonomous* navigation**, would be the ability to perform all the above actions *without* any external input (apart from, perhaps, data that spatially indicates or defines the *desired destination*, except in *visual* *exploratory systems* where there is no particular need to define a particular destination but instead the *visual or sensory* *features of interest* to navigate *towards* – or *against)*.

## Aim

The ultimate aim of the AVINSoR project is to develop an *intelligent* system able to perform *autonomous exploratory* navigation on a miniature ALV (Autonomous Land Vehicle) platform; the word *intelligent* entailing the system’s ability to ***define*** and ***refine***visuospatial perceptions, behavior patterns (movement) and decision-making skills via *supervisory input*.

The need for *supervisory input* may be partially superseded by intelligent algorithm(s) that attempt to automate the *training* process, though this is in no way a *functional requirement* due to time constraint.

## Survey of Current Methods/Theories/Models

#### Current Methods of Autonomous Navigation

An autonomous navigation system was implemented on a small quadcopter using the Odroid-U2 development platform[[1]](#footnote-1). The method utilizes *visual* and inertial sensors for GPS-independent *3D pose estimation* which would be especially useful in a GPS-denied environment, and *3D reconstruction of overflown terrain* to find potential landing sites and generate landing trajectories for the purpose of *autonomous landing* [2].

DelFly Explorer, a dragonfly imitating ornithopter, with its initial origins as an undergraduate BSc project at Delft University of Technology, attracted media attention for being the world’s smallest and lightest self-navigating drone [3]. The ornithopter is currently able to perform collision-free flight in a given room through the implementation of stereo-vision based obstacle detection algorithms (fixated towards areas of low visual texture) and a control algorithm which generates a short-term flight path on a single miniature microcontroller development board weighing only 4g [4] [5].

# Functional Specification

## System Concept

The initial concept of the AVINSoR project is illustrated in its abstract systems block diagram form in **Figure 2**. 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.

  
**Figure 2.** A diagram illustrating the fundamental concept behind all three prototypes to be developed. Obviously, Prototype 1 would utilize a wired USB connection instead of wireless connection, and the digital camera input would have to be omitted due to hardware restrictions.

## Functional Requirements

|  |  |
| --- | --- |
| Requirement ID (Implementation Priority) | Requirement |
| Controller Board to PC Communication Channel | |
| FR1 | **The communication link should allow for *transmission* *and* *reception* of data between the Controller Board and the PC (ideally, simultaneously); *and* at suitable bandwidth. [[2]](#footnote-2)**  This will be necessary to allow for observation of the controller board’s ***current state***. Hence the trainer-user will be able to make necessary adjustments to visual perceptions, behavior-patterns memory and cognitive ***synapses***. [[3]](#footnote-3)  The term ***current state*** denotes all data *received by* or *held in* memory of the controller board – all of which may be necessary to observe for the purpose of training(in case of trainer-user) and *debugging* (in case of system developer).  The **controller board’s** ***current state*** includes:  The **sensory data** received.  **“Object” perceptions / feature-patterns** held in memory.  **Behavior-patterns and actuator models** held in memory.  **Cognitive “synapses”** held in memory.   * **System settings** held in memory. |
| FR2 | The communication link should be used to update controller board firmware. |
| Controller Board Software - Cognitive System(s) | |
| FR3 | A **visual perception** **system** should be able to *form* and *reform* (improve) feature patterns that exist to represent objects from sensory data.[[4]](#footnote-4) Both processes (formation and enhancement) should be executed through trainer supervision. |
| FR4 | An **actuator control system/behavioral pattern memory** should be able to store **patterns and models** that function to calculate appropriate signals to send to actuators in order to allow appropriate movement. |
| FR5 | A **cognition system** should allow for *synaptic associations* to be made between visual perceptions and behavioral patterns – with the purpose to initiate appropriate behavior for every *active* visual/sensory perception.  Initially, the learning of these associations will be *“forced” through trainer supervision*, as the timescale of the project does not allow for intelligent automation of synaptic connection generation (a vast and complex field).  intelligent decision-making model)  This system will function to allow appropriate behavioral patterns to be calculated and executed when corresponding object(s) are detected *or* sensory patterns are active. |
| Graphic User Interface (GUI) | |
| FR6 | * The **Graphic User Interface (GUI)** will allow for the trainer-user to view controller board’s ***current state*** (sensory data, “memories”, and “synaptic connections”) in an easy-to-interpret graphical manner. |
| FR7 | * The **Graphic User Interface (GUI)** should allow the trainer-user to *train* the system through easy-to-use graphical features. |

## Deliverables

### Software Deliverables

* **Training Software / GUI**The software that the human trainer-user will use on a personal computer in order to establish and maintain a communication link with the robotic device in order to *observe* data of the “*current state*” received from the controller board executing the *AVINSoR firmware*; and subsequently use the software to *train* the system to allow for autonomous navigation.
* **Controller Board Software**Essentially, the software that is executed on MATLAB® (prototype 1) or the O/S executed on the Controller Board processor that will allow for visuospatial cognition and decision-making to occur.

Application source code will be modular and object orientated (OOA and OOD diagrams will be provided).

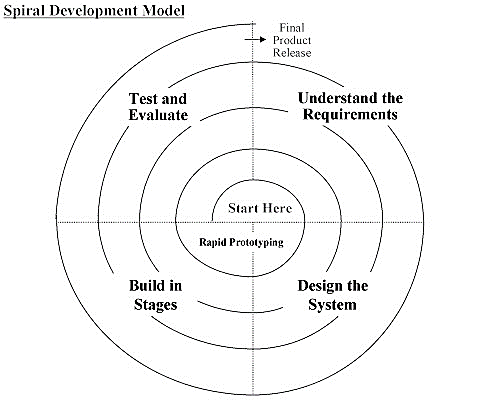
### Hardware Deliverables

* **Demonstration Robot (Prototype 3)**The final prototype will be developed with the sole purpose of *final demonstration* in mind, The robot will be based around a suitable processor-based controller board (essentially a single-board computer), connected to various sensors and actuators (servo motors) through external circuitry if needed.

# Project Plan

## The Development Methodology

A spiral (iterative) development approach will be used in the actuation of this project, instead of the traditional sequential waterfall modelled approach. This would ensure that there is a consistent balance between research, design, implementation (actual hardware and software development), and testing throughout the entire timeline of the project.



**Figure 3.** A diagram illustrating the concept of iterative development. [6]

Such approach would also ensure that there is always a demonstrable prototype at hand at any given moment (during the “active” phase) – this would ensure that all specific areas of research and development are being utilized and integrated into the demonstration model/prototype.

   
**Figure 4.** A diagram illustrating how the 36-week project timescale is split between the *research phase* and the *active phase*; and the individual prototype timespans.

## Gantt Chart and Tracking Progress

**Figure 5.** Original project Gantt chart.

According to the above Gantt chart, entire Initial Design should be complete by the end of Week 12. This may not be the case due to little time allocated towards the project during Week 12, hence Christmas period would be used to catch up with the work.

**PLEASE NOTE: A new Gantt chart needs to be designed due to the introduction of spiral development methodology (Prototypes 1, 2 and 3); also, Christmas Break and Easter Breaks have not been included though they will be used for project work.**

## Development: Prototypes

#### Prototype 1

As evident through the Gantt charts, during the initial stages of the project, time will be allocated towards the development of the fundamental part of the project: the **cognitive system** (which would essentially allow for *trainable* autonomous navigation) and the **software implementation of it**. Hence, the LEGO® Mindstorms® NXT robotic kit will be used, for a number of reasons:

1. **Availability.** Many LEGO® Mindstorms® NXT robotic kits are available in the School of Engineering and Digital Arts via the project supervisor.
2. **Its ability to be easily interfaced with MathWorks® MATLAB®**.MathWorks® MATLAB® is a popular (industry-standard) numerical computing environment (IDE) and fourth-generation (high-level) programming language with many complex readily-available libraries (MATLAB® *Toolboxes*) that allow for easy design and implementation of:
   1. **Machine-learning algorithms for data-mining and pattern-recognition** (via the Neural Network Toolbox™).
   2. **Image processing algorithms** and **Visual Object Detection and Recognition algorithms** (via the Image Processing Toolbox™, Computer Vision System Toolbox™, and the Image Acquisition Toolbox™)
   3. **Geographical/spatial mapping algorithms** of the environment (via the Mapping Toolbox™).

Another major advantage of MathWorks® MATLAB® is its ability to generate equivalent code in lower level languages such as C and C++. This would allow us to easily port the **cognitive system** developed in MATLAB® to lower-level compilers if there is ever a need to do so - though it is advised that the controller-board used for Prototypes 2 and 3 be selected specifically with ease of MATLAB® interfacing in mind.

1. LEGO® Mindstorms® NXT is a modular robotic kit. **Sensor and actuator circuitry and corresponding software code has already been developed**, saving time in the early stages - as a robotic platform (a functioning micro autonomous ground vehicle) would be readily available at the early stages of the project so high-priority tasks such as the *design and development of the cognitive system can be prioritized*.

#### Prototype 2 and Prototype 3

**Prototype 2** would involve the employment of the **cognitive system**(developed for Prototype 1) in an entirely new controller board with greater computing power. The new controller board could be an MCU-based development board, in which case the **cognitive system** and all other **software systems** (developed later to meet functional requirements) would have to be generated in the form of **low-level** **firmware source-code**. However, **a processor-based board with the ability to execute a lightweight/embedded operating system such as Android™ or Linux® would be more preferable**. The latter option would allow for easy interfacing with a camera module (with little or no programming required, due to the availability of *drivers* and *software libraries*), hence saving time and allowing the **entire *trainable* *biologically-inspired autonomous navigation* *system* to be developed as a single *executable* software with modular components** – this would be much easier to develop, debug and test**. Prototype 2 will be an improvement to Prototype 1 because:**

1. The ability to connect a camera module (or stereo-camera module) in order to receive visual data – which is not possible with the LEGO® Mindstorms® NXT.
2. Greater computing power; necessary to allow for on-board computation and fast execution of complex algorithms.
3. Possibility of a *wireless* communication link between the Controller Board and the PC; to allow for a more *application-specific/realistic* demonstration.

**Prototype 3 is the name given to the final demonstrable prototype** that would be prepared before the Viva Voce Examination on Week 28. Essentially, this would be an upgrade to Prototype 2; no changes to the Controller Board will be made, though *external* ‘sensor interface circuitry’ and ‘actuator control circuitry’ may be modified or replaced if absolutely necessary. The difference between Prototype 2 and Prototype 3 would largely be in the software the Controller Board executes.

# Progress to Date

## Initial Design

**The major focal-point in the development of Prototype 1 is the design and development of the Cognition System,** this involves defining the ‘faculties’ that constitute the Cognition System, how they interrelate with respect to function and data flow, and the specific operational flow of the processes and sub-processes. The architecture is largely determined by psychological theories and models specifically chosen for implementation in order to carry out specific functions similar to our own (as humans).

**Figure 6.** The current proposed architecture of the Cognition System to be written using MathWorks® MATLAB® and executed using LEGO® Mindstorms® NXT.

|  |  |
| --- | --- |
| Faculty Name | Description/Function |
| Visuospatial Cognition Faculties | |
| Visual and Sensory Perception | Functions to collate all sensory data (including past data from memory, and future data through *visuospatial imagination*) in order to create feature patterns that are representative of outside phenomena (objects). |
| Spatial Perception | Functions to collate all sensory data, feature patterns (representative of objects), even robotic movement (direction and speed) in order to form a spatial map; the map will allow relative perception of the *current location* and (if necessary) *destination location)*. |
| Visuospatial Imagination (OPTIONAL) | Visual Imagination allows humans to represent perceived objects and phenomena in the form art and speech (literature), but more importantly, allows humans to anticipate behavior of objects, or the result of human interaction with *the objects in question –* thus allowing analytical decisions to be made. Hence, synthesis of an *imagination faculty* is becoming increasingly popular in the field of cognitive robotics, with many potential advantages in applications such as: dynamic terrain events (reasoning where to go in order to avoid damage), explicit collaboration (reason about when help is needed), etc. [7] |
| Other Cognition Faculties | |
| Decision Making Faculty | The human mind’s decision making process is a logically structured process; it is worth noting that Off-Line Simulations of decision-making models and processes are becoming ever more popular in the field of cognitive psychology, and much of the research may be appropriated towards implementation of these models and processes in computer systems. [8] |
| Motor Cognition Faculty | Produces appropriate and highly refined signals to control actuators, to allow for accurate and appropriate movement with response to stimuli, but in accordance to cognitive decisions. |

## Development Tools

### Software Development Tools and Libraries

* **MathWorks® MATLAB®** and **MATLAB® *toolboxes* (Prototype 1, 2, 3)**Used for system modelling, coding, and implementation.*Available on all School of Engineering and Digital Arts machine.*
* **RWTH – Mindstorms NXT Toolbox for MATLAB (Prototype 1)**Used to interface MathWorks® MATLAB® with LEGO® Mindstorms® NXT.*Released free (open-source) under the GNU General Public License by RWTH Aachen University.*
* **Digia® Qt® (Prototype 2, 3)**A cross-platform C++ Integrated Development Enviroment. Used to develop the training GUI and PC-side software. May also be used to develop Android or Linux executable software application for the Controller Board.  
  *Released free (open-source) under the GNU General Public License by RWTH Aachen University.*
* **Android Studio and Android SDK Tools (Prototype 2, 3)**May be used to develop Android executable software application for the Controller Board.  
  *Released free (open-source) under Creative Commons Attribution 2.5 License.*

### Hardware

* **LEGO® Mindstorms® NXT Robotic Kit (Prototype 1)***Available at the School of Engineering and Digital Arts through project supervisor.*
* **A Processor-Based “Development Board” or “Single-Board Computer” with the ability to run Android™ or Linux® Operating System (Prototype 2, 3)**  
  *Examples include the Odroid series, Beagle Board, Raspberry Pi, etc.  
  Will be sourced few weeks before the beginning of the Prototype 2 development phase.*
* **2 to 4 Servo Motors (Prototype 2, 3)**To be used as actuators in the demonstration model.
* **Digital ultrasonic sensor(s), IR emitters and detectors, etc. (Prototype 2, 3)**To act as auxiliary digital sensors to enhance quality of visual/visuospatial perceptions formed.
* **Radio Transceiver Module or Bluetooth Module (Prototype 2, 3)**To allow for wireless communication link between PC and Controller Board.
* **Digital Camera Module (USB) (Prototype 2, 3)**To allow for 15-30fps image capture, in order to form visual perceptions of the environment around the vehicles, and the travel path.

# References

|  |  |
| --- | --- |
| [1] | Oxford University Press, "Oxford Dictionaries," 2014. [Online]. Available: http://www.oxforddictionaries.com/definition/english/navigation. [Accessed 18 December 2014]. |
| [2] | R. Brockers, M. Humenberger and S. Weiss, "Towards Autonomous Navigation of Miniature UAV," in *Computer Vision and Pattern Recognition Workshops (CVPRW), CVPR2014*, Columbus, OH, USA, IEEE, 2014, pp. 631-637. |
| [3] | J. Vincent, "'Robot dragonfly' is the smallest and lightest self-navigating drone," 2013. [Online]. Available: http://www.independent.co.uk/life-style/gadgets-and-tech/robot-dragonfly-is-the-smallest-and-lightest-selfnavigating-drone-9008723.html. [Accessed 15 October 2014]. |
| [4] | Delft University of Technology, "DelFly Explorer," 2014. [Online]. Available: http://www.delfly.nl/explorer.html. [Accessed 15 October 2014]. |
| [5] | C. D. Wagter, S. Tijmons, B. Remes and G. d. Croon, "Autonomous Flight of a 20-gram Flapping Wing MAV with a 4-gram Onboard Stereo Vision System," Delft University of Technology, Delft Netherlands, 2013. |
| [6] | J. R. Chapman, "Project Management Training : Software Development Methodology," 1 January 2007. [Online]. Available: http://www.hyperthot.com/pm\_sdm.htm. [Accessed 19 December 2014]. |
| [7] | D. M. Lyons, S. Chaudhry and P. Benjamin, "A Visual Imagination Approach to Cognitive Robotics," in *Symposium on Understanding the Mind and Brain*, Tucson Arizona USA, 2010. |
| [8] | S. Nichols, S. Stich, A. Leslie and D. Klein, "Varieties of Off-Line Simulation," in *Theories of Theories of Mind*, Cambridge UK, Cambridge University Press, 1996, pp. 39-74. |
| [9] | P. E. R. L. Gregory, Eye and Brain, The Psychology of Seeing, Fifth Edition, Oxford UK: Oxford University Press, 1998. |

1. A 1.7GHz ARM Cortex-A9 Quad-Core processor based micro-sized development platform designed to execute Android and Ubuntu-Linux operating systems. [↑](#footnote-ref-1)
2. Suitable bandwidth is a [↑](#footnote-ref-2)
3. The term “*synapse(s)”* is used to denotelinks established between visual perceptions and behavior patterns in the cognition system. [↑](#footnote-ref-3)
4. The term *sensory data* is used to denote data from all sensors (including cameras) connected to the controller board. [↑](#footnote-ref-4)