Stigmergy for Multi-Robot Coverage
Design Documentation

Michael Chadwick

November 19, 2015

Contents

Chapter 1

Design Documentation

1.1 Summary of Proposal

1.1.1 Project Outline

"A robotic swarm is composed of a large number of simple physical robots. From the local interactions between the robots and the interactions of the robots with the environment, an efficient global intelligence emerges. Multi-robot coverage is the problem in which a swarm of robots needs to coordinate decentralised in order to effectively and efficiently cover an unknown environment. Examples include various monitoring, rescuing, and patrolling scenarios. The purpose of this project is to set up an experimental demonstrator, i.e. 'a dark room', in which multi-robot coverage experiments can be conducted using e-puck robots, implementing the stigmergy principle as observed in ant colonies. Ants use chemicals, called pheromones, to communicate with each other via the environment. However, despite of a few reports of using chemicals in robotic experiments, this is not a straightforward approach due to difficulties in implementation and limited extendability. Therefore, we take advantage of a glow-in- the-dark foil (i.e. a foil covered by phosphorescent material which absorbs UV light and re-emits the absorbed light at a lower intensity for up to several minutes after the original excitation). As robots need to emit light to the glow-in-the-dark foil, each e-puck robot is equipped with a UV-LED pointing toward the floor. The glowing trails will take up the role of natural pheromones."

1.1.2 Project Aims

The Aims of the Project are to construct a testing arena for the e-Puck robotic platform, notably a "dark room," which will allow the usage of the robot's lights to leave localised messages on flooring which can store and emit light. Successfully fulfilling the aim will mean that other users of the e-Puck system have a basis to create their own dark room. This may improve research in swarm robotics or, if used for demonstration purposes, can bolster interest in applicable fields.

The primary Objective of the project are to build said arena with dimensions small enough to fit on a circular table approximately 60 centimetres in radius. As a secondary Objective, completion of the project will produce a program for the e-Puck robotic system to demonstrate the effectiveness of the environment the robot will be placed in. The program will initially be an implementation of StiCo[6], but may have modifications dependant on time constraints.

1.1.3 Changes to Original Specification

The original requirements document stated that the program will be similar to StiCo and HybaCo[4]. The language used suggests that the final program will be a variant of the two algorithms. This is no longer the case – the program will be an implementation of the StiCo algorithm and a separate, modified program will only be available should there be enough time to change the original implementation. This is to allow the time to properly implement the StiCo algorithm as the compiled code is used as a proof of concept that the constructed dark room is a viable environment for the testing of light based communication using the e-Puck system.

1.1.4 Relevant Research and Analysis

There has been some research on Stigmergic algorithms. The main algorithm used within this project will be StiCo[5, 6, 7]. When implemented, the algorithm can help to reduce the total area of terrain covered by multiple robots which would improve efficiency and the total area being patrolled upon. This algorithm will be used to show that the dark room that will be built is functional.

BeePCo is another, different algorithmic solution in stigmergic robotics. Whilst it is not applicable for the project due to it's reliance on direct communication between agents and would be a very different implementation compared to StiCo, it can be useful with fewer robots to help monitor all of the area by maintaining distance through network connections.

HybaCo is a combination of both StiCo and BeePCo – initially running the latter algorithm whilst a direct connection is available and then switching to StiCo when connecting to other robots is no longer possible, perhaps due to range limitations[1]. If time permits, the final deliverables will include HybaCo, by creating a state machine to switch between the children algorithms StiCo and HybaCo

1.2 System Design

1.2.1 Project Components

Anticipated components for the project include documentation for each stage of development, source code and it's compiled version of the StiCo algorithm for the e-Puck robotic platform. The final component will be a constructed arena for testing the source code with the e-Puck system, along with blueprints that are provided in the Design phase of the project.

These together will complete the aim of providing a dark room for researching and testing of algorithms for the e-Puck system, along with a demonstration program to show whether the constructed arena is successful.

1.2.2 Proposed Data Structures

Currently, it the project will not contain advanced data structures such as Sets, Queues, Arrays or Stacks. The program will have basic data-type variables to store the radius of the circular path the robot will take, a variable to check whether the robot has scanned a light path successfully and, time permitting, an integer variable to be used as a state machine - running StiCo initially then BeePCo/HybaCo should the user wish the state to change. The changing of state can be called by pressing a button on the robot, which will make the agent to execute the selected algorithm.

Data Structure Manipulation

With the StiCo algorithm, no user input would be required to manipulate the data structures in place. When a light trail is detected, the motor strength to each wheel will be swapped to allow the robot to turn in the other direction. If the HybaCo algorithm is implemented concurrently, buttons on the e-Puck system would be used to differentiate between running the StiCo implementation and the HybaCo algorithm.

1.2.3 Interface Design

As the project is using a robotic system, the interface between user and program is already defined. Extending the deliverables to include an implementation of the HybaCo algorithm will mean applying an event-driven stage in the program, to allow the user to press a button on the e-Puck system to define which algorithm should run. This will act as a basic state machine so the user may choose which algorithm each robot executes. Lights on the robot can be used to indicate which state has been selected.

Evaluating the Project will fall into two categories: the constructed dark room and the implementation of the algorithms, StiCo and potentially HybaCo too.

The dark room will be evaluated based on how little light gets through the covering and how the edges can keep the robots from leaving the sectioned area. Testing for the light levels can be done by looking at the constructed arena to see how strongly the flooring glows after normalising in the environment. A torch can then be shone into the arena to see whether the floor can successfully hold the light for an amount of time.

For evaluating the program, the robots need to interact with the light trails they leave behind in some form to help evaluate the effectiveness of the dark room. This is achieved by implementing the StiCo algorithm, where the robots will change direction when they come into contact with a light trail.

1.2.4 Design Diagrams

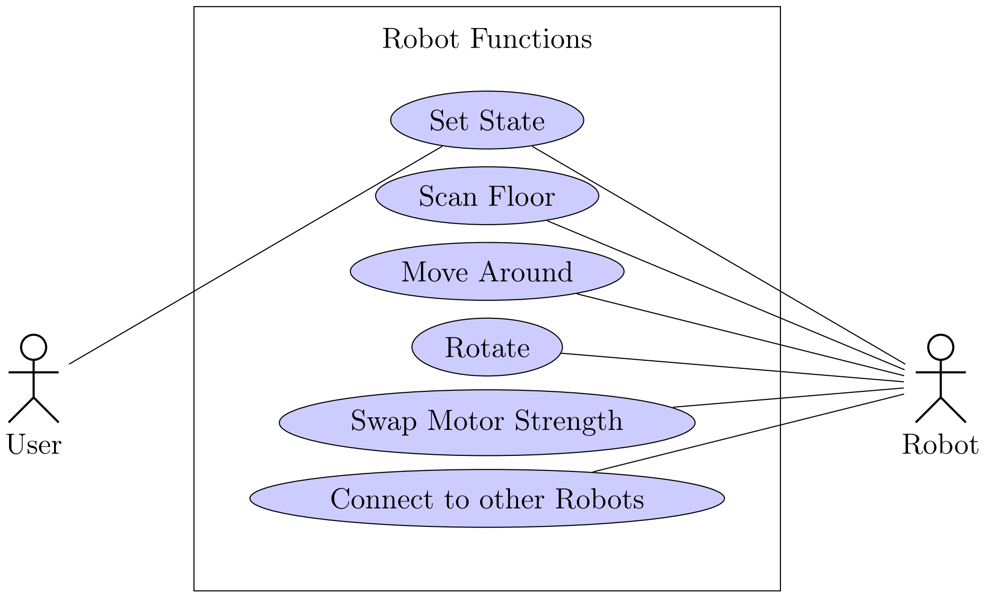


Figure 1.1: Use Case Diagram for proposed software solution

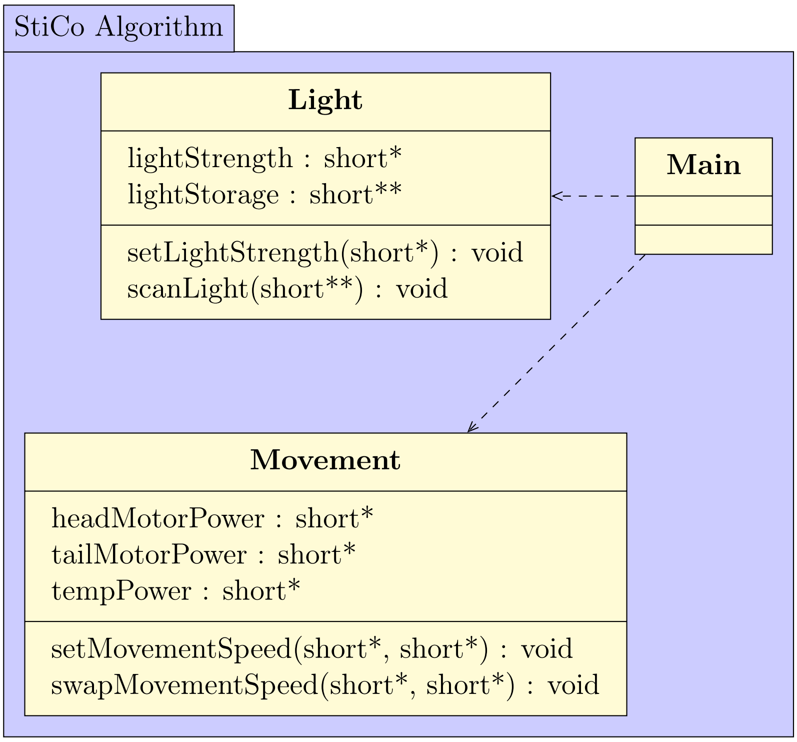


Figure 1.2: Class Diagram for StiCo Implementation

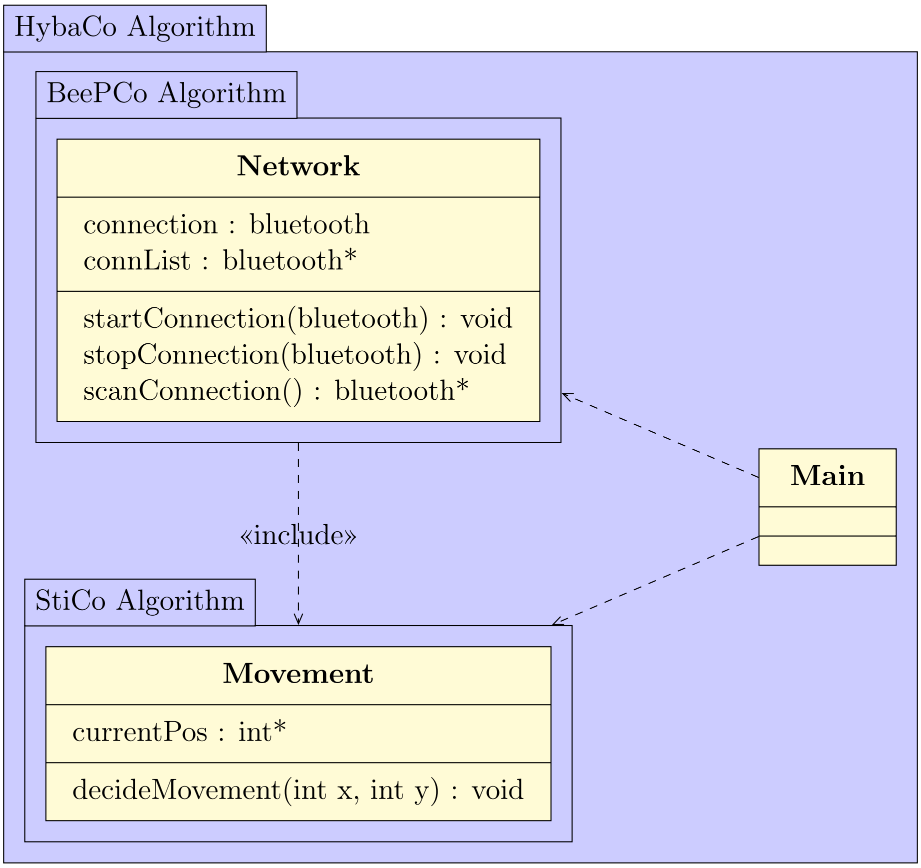


Figure 1.3: Class Diagram for HybaCo Implementation. Note that the StiCo package will include the Light and Movement classes defined in the previous diagram(Fig. 1.2) – Only additional information is shown here

1.2.5 Project Pseudo-code

Algorithm 1 StiCo Algorithm[6]

Require: Each robot can deposit/detect pheromone trails

```
Initialise: Choose circling direction (CW/CCW)
loop
    while (no pheromone is detected) do
        Circle around
        deposit Pheromone
    end while
    if (interior sensor detects pheromone) then
        Reverse the circling direction
    else
        while (pheromone is detected) do
            Rotate
        end while
    end if
end loop
```

Algorithm 2 BeePCo Algorithm[3]: Differentation Cycle

every T_{QR} do
if $(h_i < threshold_{QR})$ **then**
    $QR_i = true$
    broadcast $hd = \{0, h_{QR}\}$
else
    $QR_i = false$
end if

Algorithm 3 BeePCo Algorithm[3]: Pheromone Propagation Cycle

while hd is received **do**
  if $(\ hd[1] < threshold_{hopcount}\ )$ **then**
    $h_i = h_i + hd[2]$
    broadcast $hd = \{hd[1] + 1, hd[2].K_{HOPDECAY}\}$
  else
    drop hd
  end if
  go to BeePCo Move Cycle
end while

Algorithm 4 BeePCo Algorithm[3]: Move Cycle

if (pheromone received) **then**
  PS-guided moving decision
else
  Keep moving in the direction of the last move
  Broadcast communication link request
  Establish local communication links
end if

Algorithm 5 BeePCo Algorithm[3]: Moving Decision

if $(\ h_i > 0\ )$ **then**
  for All the received pheromones (p) of the robot **do**
    $diff_X = p_{Sender_X} - currentCoordinate_X$
    $diff_Y = p_{Sender_Y} - currentCoordinate_Y$
    $\theta = ArcTangentQuadrant(diff_Y, diff_X)$
    $component_X = p.hd * cos\theta$
    $component_Y = p.hd * sin\theta$
    $Sum_X + = component_X$
    $Sum_Y + = component_Y$
  end for
end if
$magnitude = \sqrt{(Sum_X)^2 + (Sum_Y)^2}$
$\theta_{destination} = ArcTangentQuadrant(Sum_Y, Sum_X)$
Apply 180 degrees shift to $\theta_{destination}$
Clear all received pheromones

Algorithm 6 BeePCo Algorithm[3]: Decay Cycle

```
for every T_DECAY do
    h_i = h_i.K_TIMEDECAY
end for
```

Algorithm 7 HybaCo Algorithm[2]

```
time = 0
loop
    if Links to Neighbours Exist then
        Apply BeePCo using BEE Pheromones
    else
        Apply StiCo using ANT Pheromones
    end if
end loop
```

Data Dictionary

Attribute name	Description	Found in entity	Occurrence
lightStrength	Stores current agent's light intensity	Light	Whenever the agent needs to change light intensity
lightStorage	Stores localised light intensity	Light	When the agent scans the floor for localised messages
headMotorPower	Stores left wheel movement strength	Movement	When movement speed is modified or change in rotation direction
tailMotorPower	Stores right wheel movement strength	Movement	When movement speed is modified or change in rotation direction
tempPower	Stores headMotorPower	Movement	Used each time the robot needs to change rotational direction
currentPos	Stores agent's position	Movement	Used during HybaCo execution. Used to calculate moving decision
connection	BeePCo var.; output for other agents	Network	Used during HybaCo execution. Denotes broadcast status
connList	BeePCo var.; list of linked agents	Network	Used during HybaCo execution. Stores information of connected agents to obtain location – part of Moving Decision Pseudo-code

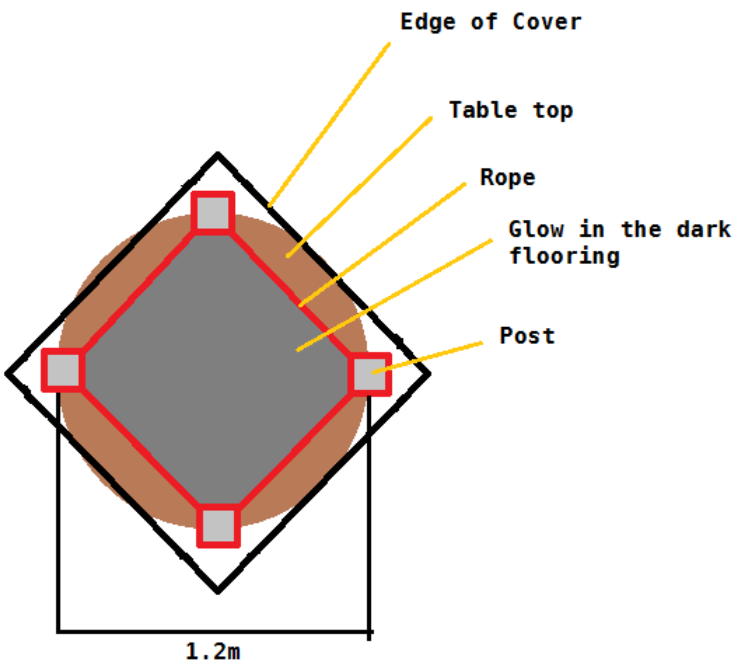


Figure 1.4: Top down view of the proposed dark room

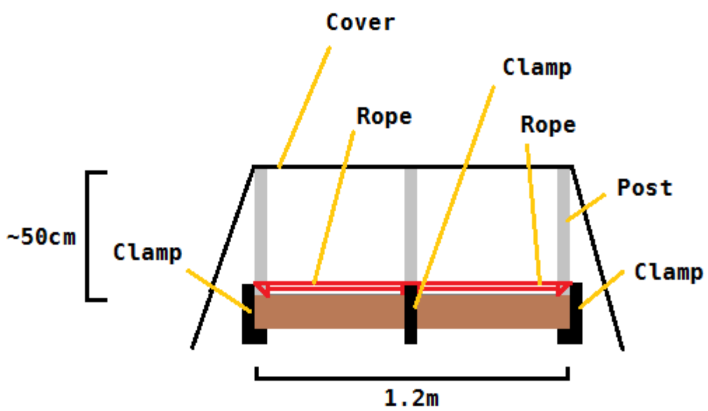


Figure 1.5: Side view of the proposed dark room

1.2.6 Arena Design

The design above is based on two main concepts – flexibility and ease of use. Once the glow in the dark flooring is placed, four posts are then clamped to the edges of the table. This ensures the flooring does not move and provides stability for the rope. Using a sheepshank knot, applied to the inner edges of the posts as suggested in the top down view(1.4), it is estimated that to completely fence the arena with one piece of rope it should approximately be 11 metres. Using rope means that the arena can be of whatever size is deemed appropriate, so long as there is enough to sufficiently cover the generated edges.

Assuming a circular table, four posts, a sheepshank knot and wrapping the rope around each post twice, the following algorithm calculates the minimum required length of rope (it is suggested that you round up to the nearest half a metre to make sure):

Algorithm 8 Rope Length Calculator (Measurements in centimetres)

1: $Edge = 3(\sqrt{2(r^2)}) + 2$ ▷ Multiplying by 3 is due to tripling the rope over
2: $WrappedPost = (postPerimeter \times 2) + 2$ ▷ Trailing 2's are for securing the rope
3: $TotalRequiredRope = 4(Edge + WrappedPost)$

1.3 Evaluation Design

1.3.1 Evaluation Criteria

As mentioned earlier in the document[1.2.3], the elements that fall under evaluation are the dark room to be constructed and the compiled program that will be placed on the e-Puck robotic system to demonstrate the effectiveness of the construct as a dark room.

The Dark Room

- The arena can contain multiple robots

- Arena limits stop the agents from escaping
- Light level is low enough to not affect the glow in the dark flooring
- Being able to view the robots without leaking light into the arena
- Portable and flexible; easy to set up

The Compiled Program

- Helps to evaluate the dark room
- Interacts with the light messages left on the flooring
- Implements the StiCo algorithm

1.3.2 User Evaluation

It is difficult to evaluate how user friendly the final products will be. This is because the user will have minimal interaction with the robots. The robots will flash to indicate a change in the algorithm that will be executed. To construct the dark room, the user will require basic knowledge of creating knots. There are no users to acquire feedback from.

1.3.3 Assessing the Criteria

The criteria will be mainly assessed when the dark room is completed. Robots can be placed within the arena to assess the correct size. A hand-held torch will be used to test whether the flooring can hold a charge, and to see the difference between ambient light levels and directly exciting the flooring. Evaluating the robot's capabilities can be done using only one agent, by using a torch to simulate another agent's light trail and seeing a change in the behaviour.

1.3.4 People involved in the Evaluation

Evaluation will be primarily completed by one person, as only one is required to set up the robots and their states.

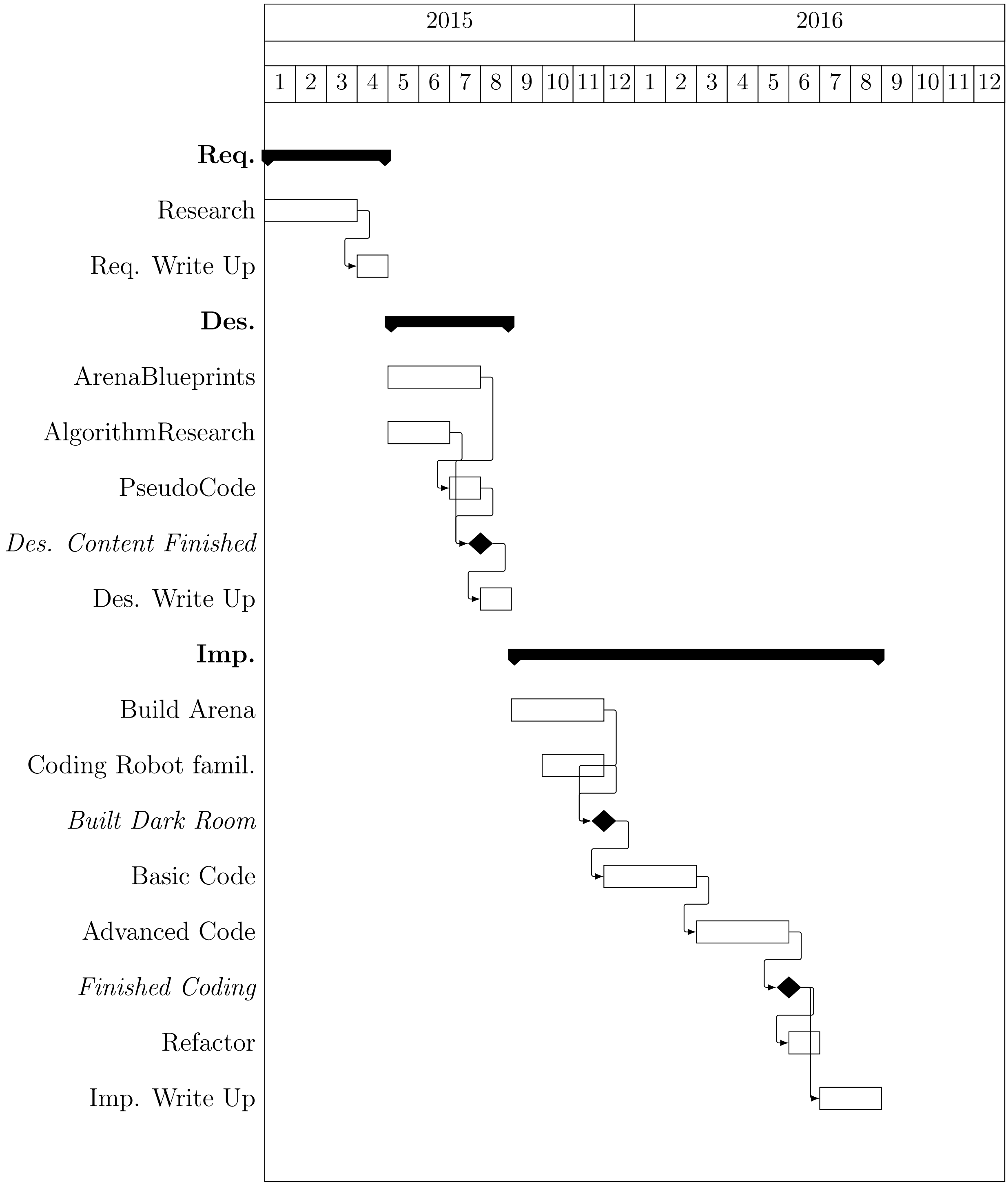
Human Data and Participants

There are no using of Human Data or Participants within this project – all parts are covered by one person and therefore does not require ethical limitations to be enforced upon the project.

1.3.5 Project Conclusion

The conclusion of the project is expected to be a small arena to be used to simulate darkness or night time situations. The robot will be interacting with light trails on the flooring left by external influences in the manner that StiCo defines.

1.4 Project Review: Design Stage



1.4.1 Current Progress

The project is currently going slightly ahead of schedule. I am reading on how to implement a coding solution for the e-Puck robotic system and looking for materials to make the posts used to hold up the Cover for the dark room.

1.4.2 Changes within the project

The algorithm which will be implemented is now defined as the StiCo algorithm. If there are more unforeseeable changes, such as having to change the dimensions of the arena, the proposed solution should be adequately adaptable.

Bibliography

[1] Bastian Broecker et al. "Hybrid Insect-Inspired Multi-Robot Coverage in Complex Environments". English. In: *Towards Autonomous Robotic Systems*. Ed. by Clare Dixon and Karl Tuyls. Vol. 9287. Lecture Notes in Computer Science. Springer International Publishing, 2015, pp. 56–68. ISBN: 978-3-319-22415-2. DOI: 10.1007/978-3-319-22416-9_8.

[2] Bastian Broecker et al. "Social Insect-Inspired Multi-Robot Coverage". In: *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. AAMAS '15. Istanbul, Turkey: International Foundation for Autonomous Agents and Multiagent Systems, 2015, pp. 1775–1776. ISBN: 978-1-4503-3413-6. URL: http://dl.acm.org/citation.cfm?id=2772879.2773431.

[3] Ipek Caliskanelli, Bastian Broecker, and Karl Tuyls. "Multi-Robot Coverage: A Bee Pheromone Signalling Approach". English. In: *Artificial Life and Intelligent Agents*. Ed. by Christopher J. Headleand, William J. Teahan, and Llyr Ap Cenydd. Vol. 519. Communications in Computer and Information Science. Springer International Publishing, 2015, pp. 124–140. ISBN: 978-3-319-18083-0. DOI: 10.1007/978-3-319-18084-7_10.

[4] Michael Chadwick. *Stigmergy for Multi-Robot Coverage: Project Requirements Document*. 2015. URL: https://github.com/Chuddington/FYP-Stigmergy/blob/master/reqOnly.pdf (visited on 11/12/2015).

[5] Bijan Ranjbar-Sahraei, Gerhard Weiss, and Ali Nakisaee. "A Multi-robot Coverage Approach Based on Stigmergic Communication". English. In: *Multiagent System Technologies*. Ed. by IngoJ. Timm and Christian Guttmann. Vol. 7598.

Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, pp. 126–138. ISBN: 978-3-642-33689-8. DOI: 10.1007/978-3-642-33690-4_13.

[6] Bijan Ranjbar-Sahraei, Gerhard Weiss, and Ali Nakisaee. "Stigmergic Coverage Algorithm for Multi-robot Systems (Demonstration)". In: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 3*. AAMAS '12. Valencia, Spain: International Foundation for Autonomous Agents and Multiagent Systems, 2012, pp. 1497–1498. ISBN: 0-9817381-3-3, 978-0-9817381-3-0. URL: http://dl.acm.org/citation.cfm?id=2343896.2344079.

[7] Bijan Ranjbar-Sahraei, Gerhard Weiss, and Karl Tuyls. "A Macroscopic Model for Multi-robot Stigmergic Coverage". In: *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems*. AAMAS '13. St. Paul, MN, USA: International Foundation for Autonomous Agents and Multiagent Systems, 2013, pp. 1233–1234. ISBN: 978-1-4503-1993-5. URL: http://dl.acm.org/citation.cfm?id=2484920.2485159.