Beginners Guide to Swarm Robotics

Beginners Guide to Swarm Robotics

Michael Roa

Chapter 1

Introduction

Robots; from nature.

Swarm robotics is a branch of robotics that deals with communities of robots and their behavior. This can be immediately overwhelming at first but with time and due diligence it can be understood in its entirety. The manual you are about to read will help you build the fundamentals needed to create, program, and utilize a swarm robot in a realistic application. The first step to understanding swarm robotics is to understand the what and why of swarm robotics, then how and why this guide will help you.

What is Swarm Robotics?

The term 'swarm robotics' is generally the mimicking of swarm behavior found in nature with regards to robotics (Tan, Ying). This can be utilizing large amounts of simplistic robots, or small numbers of complicated robots. The behavior the robots exhibit, no matter their complexity of number, are based on similar swarm behaviors in nature. It takes inspiration from swarm behavior witnessed in nature, such as flocks of birds, schools of fish or armies of ants. A single robot designed to act as an animal is not implementing swarm robotics, but a group of robots acting as a herd may be defined as such. These creatures and their robotic counter parts seek to act as larger community of independent agents towards some goal. To conclude, swarm robotics is the mimicking of swarm behavior found in nature regarding robots cooperating towards some goal.

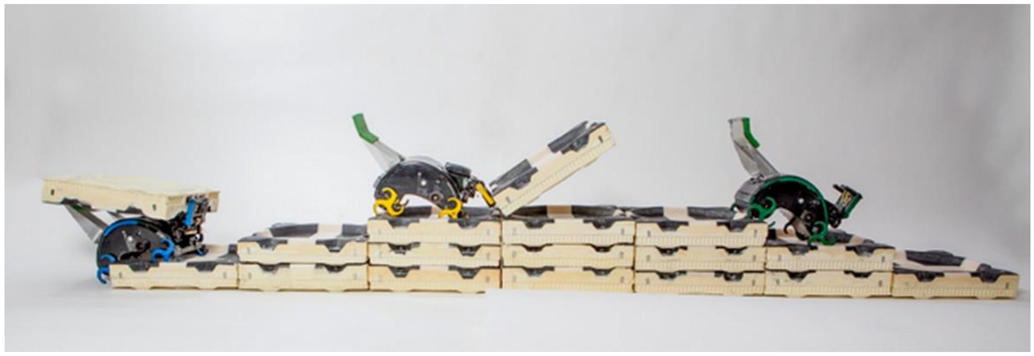


FIGURE 1.1 Robots working together to build a structure, an example of swarm robots in practice.

Why study swarm robotics?

The application of swarm robotics is far reaching and broad; extraterrestrial exploration, manufacturing automation, disaster relief, environmental research (Ying Tang, Zhong-yang Zheng, Reinhard Gerndt & Stefan Krupop). Anywhere a robot can be used, instead a group of robots with the capabilities of the single robot can be deployed to increase effectiveness. The field of swarm robotics has only recently reached a level of development that allow for commercial viable application (Ferrer, Eduardo). It is projected to receive heavy investment soon alongside most other robotics-based fields. Finally, this manual has been written with the intent to assist the approach of the NASA Swarmathon competition (Kuijt, Dr David), a yearly contest held to assist NASA's research on the deployment of swarm robots to Mars.

How can this manual help me?

Swarm robotics is a very hard field to define a 'start point'. It requires some degree of understanding in the fields of robotics, mathematics, and computer science. Current 'starter' swarm robot kits focus purely (Reinhard Gerndt & Stefan Krupop) on 'large-scale' deployments with 'low complexity'. These are not suitable to research regarding the NASA Swarmathon competition.

This manual is intended as a starters guide to build, program, and test your first 'swarm robot'. It will teach the most necessary concepts of robotics, mathematics, and computer science in a manageable format.

Chapter

2

ROS & C++ Primer

Bread and butter of robotics programming.

C++ is a key programming language in regards to swarm robotics. It is considered a key language across the entire field of computer science. Its nature as a heavily used 'low-level' (by modern definition) programming language makes it a perfect fit to program robots with. ROS on the other hand is a standardized package for robotics with a publisher-subscriber system. ROS is used to handle input data from the sensors of the robot whereupon it 'publishes' it to a 'subscriber', but we will go more into detail of this soon. This section will identify concepts you should brush up on before heading onwards.

Key Parts of C++

C++ is a very extensive language, and thus one should have some prior skill with it before continuing. The key parts of C++ that any robotics programming should understand can be broken down into the following; syntax, algorithms, and object-oriented programming. Syntax refers to the style and logic of the code. Syntax can vary between languages, but it is important to remain consistent with your syntax especially with group programming. This will help organize your efforts as a team and avoid confusing bugs due to someone else's inconsistencies. Algorithms play a major role in the treatment of data. Robotics has heavy use of algorithms to filter 'fuzzy'(inaccurate) data from sensors into more accurate data. Understanding algorithms will give you a leg up on programming patterns and optimizing code for the robots. Finally understand the core concepts of object-oriented programming. Understanding the intricates of object-oriented programming can greatly assist you when programming robots. As robots have limited memory use of object-oriented concepts like the ‘stack’ and ‘heap’ can prevent your code from causing memory overflow issues.

Publisher Subscriber Model of ROS

ROS (Robotics Operating System) is an open source C++ library that uses a publisher-subscriber model to communicate data of the robot. ROS is used as it provides an easy library to pull data from sensors on the robot and command data to various components of the robot. The main concept to understand from ROS is the publisher subscriber data-model. The ‘publisher’ pushes the data of a ‘topic’ to a ‘subscriber’. For a more practical example, imagine a car; the publisher of the car is your speedometer, the topic of the car is the speed of the car, and the subscriber of the car is you. In this case, when you push on the accelerator your speedometer increases accordingly. It is ‘publishing’ the speed of the car, one of many ‘topics’ or data points of the car. You as a subscriber then see the speedometer and react accordingly. If you are going to fast you slow down, or vice versa. This example shows a practical application of the core concept behind ROS.

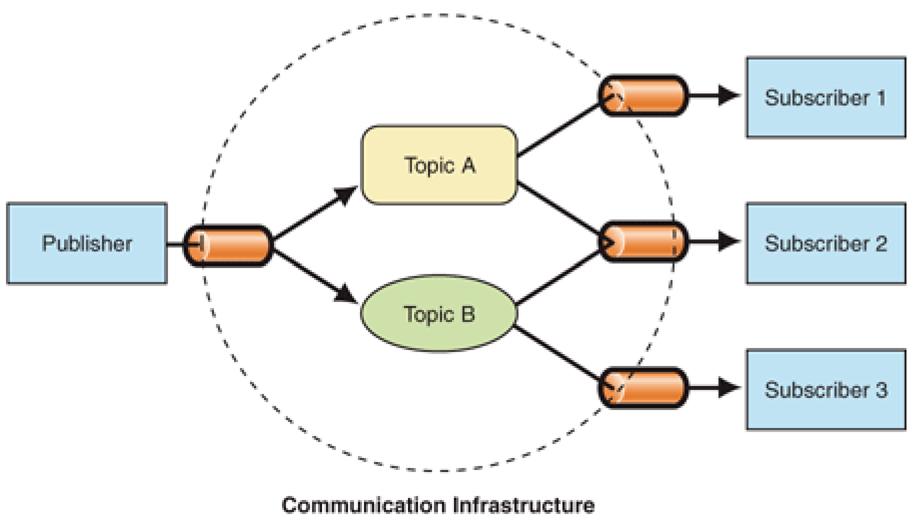


FIGURE 2.1 A publisher subscriber model.

Ev3 Dev

On the note of ROS, we will use EV3 Dev libraries as the basis for all programming. This is an open source library geared towards the LEGO Mindstorms EV3 kit. This manual will cover some basic examples using EV3 but the library, and its implementation with ROS is well documented.

Chapter

3

Defining Robotics

Or; "Why is my toaster not a robot".

We defined swarm robotics earlier, but never addressed the question of what a robot is in general. If we continued without a definition it would like to try to drive a car with one wheel (this is not advisable). The next section and chapters will help define the general idea of robotics, separate it from swarm robotics and introduce a few pivotal terms that will be used throughout the manual.

What isn't a robot?

The definition for a 'robot' goes something along the lines of 'a machine capable of preforming a complex series of actions through external, or internal control.' (Oxford Definition) While thorough this definition is unwieldly especially for our purposes, so we will instead readjust it to make a more suitable one. To construct a better definition let's ask if a car is a robot? A car is a machine, capable of preforming the complex action of going forward at high speeds, through external and in more modern cases internal control. (Pagarkar, M. Habibulah) So why then is a car not a robot? Let's update the working definition to state that a robot is a machine that requires no immediate human intervention to execute a complex series of actions. Now a car falls solidly outside the definition of a robot. It is still able to go forward at high speeds, and in some cases operate autonomously but typically requires a human to be ready to take immediate control in many situations. So now let's examine the question "Is my toaster a robot?" Our current definition is 'A machine that requires no immediate human intervention to perform a complex series of actions through external or internal control.' This in mind, a toaster by some stretch can still be considered a robot. Another set of readjustment is in order then as sadly in the academic community, a toaster is not a robot. (Sadowski, Ed) This time we'll add that a robot must perform the complex actions using sensory information. With this, the robot may use gyro sensors, laser measurement, or any set of sensory information to execute its action. Now the toaster, like the car, is not a robot. With our full definition in mind we can begin examining what does fall under the idea of a robot and eventually fully define a swarm robot.

What is a robot?

Through our definition let's look at some robots to better build an idea of what is a robot. A robotic arm typically used for industrial manufacturing is a robot. These robots execute commands sent to them, using gyro sensors to measure their position. They are capable of very complex tasks such as assembling car components on an assembly line. (Acieta)

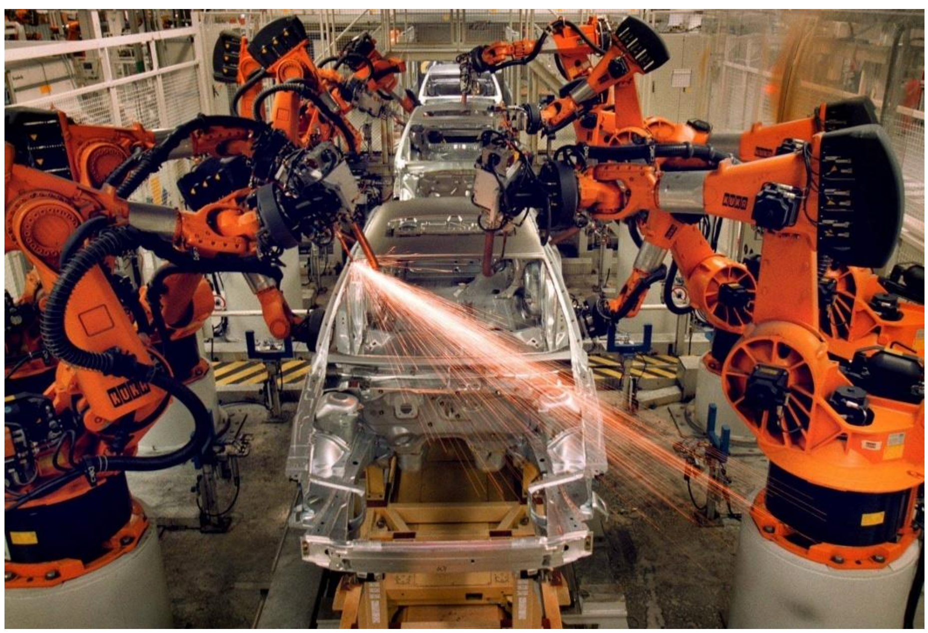


FIGURE 3.1 Robotic arms assembling a car.

Another type of robot is an 'autonomous drone.' They execute complex tasks such as flying, landing, and take-off without human intervention. As well they use GPS sensors to measure their position and fly wherever needed. (Drone Omega)

Flying drones have become very popular and are commercially available for use by the public. As well, they are significant players in the roles of national defense.

FIGURE 3.2 X-47 B Drone refueling

One last example of a robot is NASA's Mars explorer robots, *Spirit* and *Opportunity*. These robots were sent to Mars to execute scientific tasks (NASA, Mars Exploration) such as exploration, soil sampling and telemetry data collection. These are incredibly complex tasks and required a large set of sensors to execute each one. (NASA, Mars Exploration)

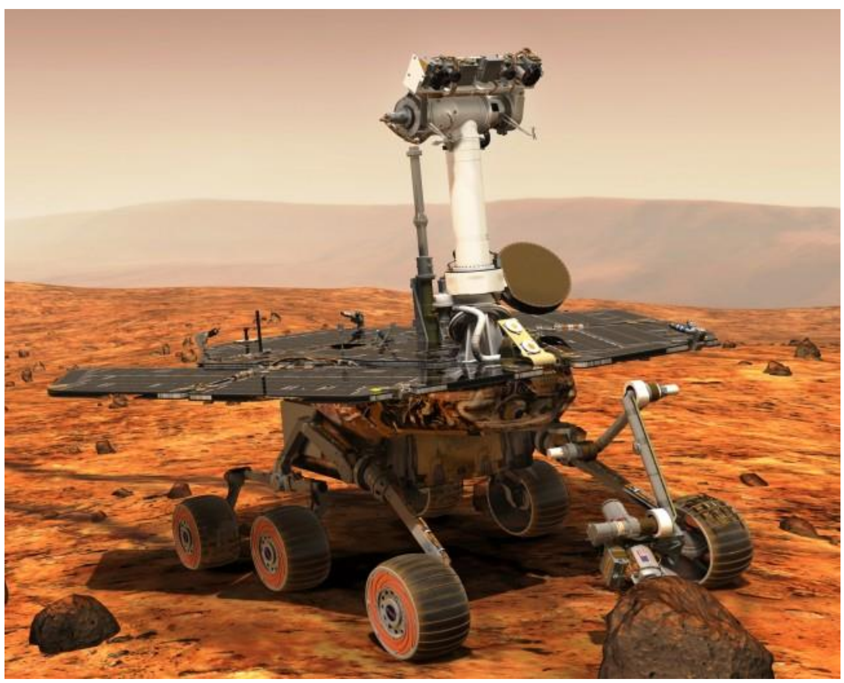


FIGURE 3.3 Artists rendition of the *Spirit* rover on Mars.

Chapter

4

Defining Swarm Robotics

Why would we ever send ants to Mars?

As with our definition of robotics, we made a very working definition in the intro chapter. Let's re-state it now before we begin the processing of integrating it with our new current robotics definition. We currently know swarm robotics as "the mimicking of swarm behavior found in nature with regards to robotics" With our memory refreshed let's dive right into making a more proper and suitable definition as we did with robotics in general.

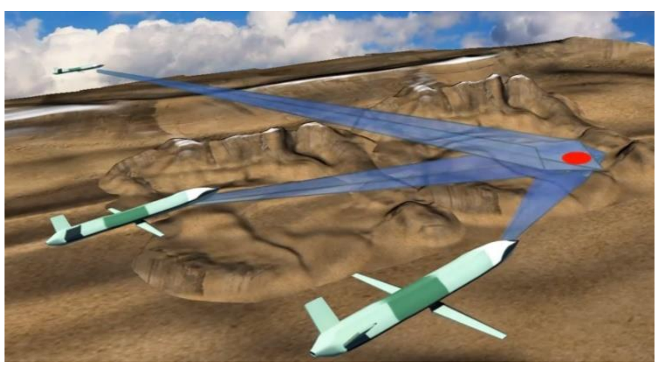
What is a swarm robot?

As before we will begin with first examining the flaws of the definition with real world examples. This time however, we'll be doing it inversely, instead finding what swarm robots fall outside of our current definition. A nice example is Intel's Shooting Star drones and their behaviors. The drones preform light shows at events (Intel, Shooting Star) creating dazzling patterns and shapes that if ever executed in nature would cause great concern to the emergent behavior of animals. But these drones fall entirely within the study o 'swarm robots yet our definition puts them outside. As before let's make a few adjustments to make it more suitable. Instead of forcing the robots to preform akin to nature, we'll broaden their capabilities with a definition that looks like "The execution of complex actions that either; mimic swarm behavior found in nature, or use extensive communication between two or more agents, with regards to robotics." Now, the beautiful performances of the Shooting Star more properly fit our definition.

FIGURE 4.1 Intel's Shooting Star creating a Christmas tree.

Examples of swarm robots

Now we can better find examples of swarm robots and get a gauge of how varied the field is considered. As we saw, Intel's Shooting Star is a 'swarm robot' project consistent of flying drone robots and a human controller prior to the complex action. Another one-to-one example of flying drones executing swarm robotic behaviors is current projects to organize military operations through drones on a much larger scale.



These robots will overwhelm sensors with coordinated flying, (NavalDrones) typically on battlefields. This behavior is considerably larger than the previous 'light show' displays but is still solidly within the realm of

FIGURE 4.1 DARPA project on swarm robotics for military application.

swarm robotics.

Now let's look at some examples that take place outside of the skies such as the ENEA robot fish swarm. (ENEA) These robots are expected to assist greatly measuring the effect of climate change and human activities effect on aquatic nature. They mimic fish swarms as they move based on sensor information that will help define the fish species preferred environment. They will help scientific better understand actual fish swarm behaviors and can be deployed to multiple environments. According to our definition these robots are clearly an application of swarm robotics.

FIGURE 4.2 ENEA robotic fish swarm.

And to wrap up our swarm robot examples we'll look at a classic swarm robot example. Harvard's micro-bots, (Rubenstein) while small, are making a big impression on the study of swarm robotics. These little robots, while simple, allow for gathering data on large scale deployments to test resilience, capability, and execution on groups that can range in the numbers of the hundreds. These robots cannot communicate between each other but are capable of preforming in large 'swarm behavior' like groups.

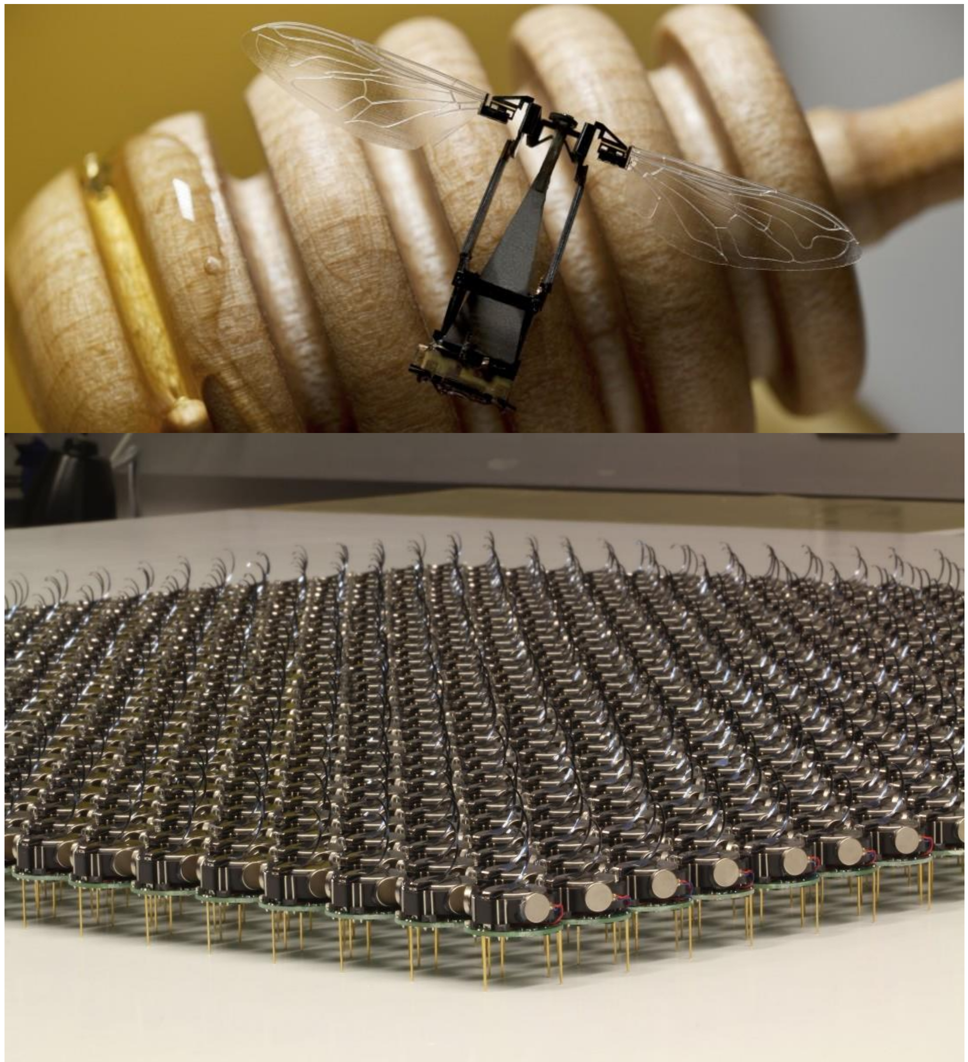


FIGURE 4.3 Examples of Harvard's microrobot swarms.

Our definition of swarm robotics can be considered robust enough that it captures many different types of swarm robots, big and small, future and current. With the definitions of swarm robotics and robotics in hand we can now find the most important terms and aspects for us.

Chapter 5

Robotics Primer

The need-to-know terms.

With our adjusted definition in hand, we can now determine what is and isn't a robot. We can then determine if a robot falls under swarm robotics or not. But this does not make us capable of understanding the aspects of either, to do this we have to first learn the most important concepts of robots. This chapter and the next will explain some of the more important concepts to begin understanding both fields, but I strongly encourage you to investigate each term thoroughly and seek other any other terms that will help you to understand how to operate with swarm robots.

Key terms of robotics

One of the most important terms which we have already been using but have no defined is a "sensor." In our case a sensor is any device that detects input from the physical environment and can send information back the main computer of the robot. Some examples of sensors include gyroscopes, touch, and light sensors.

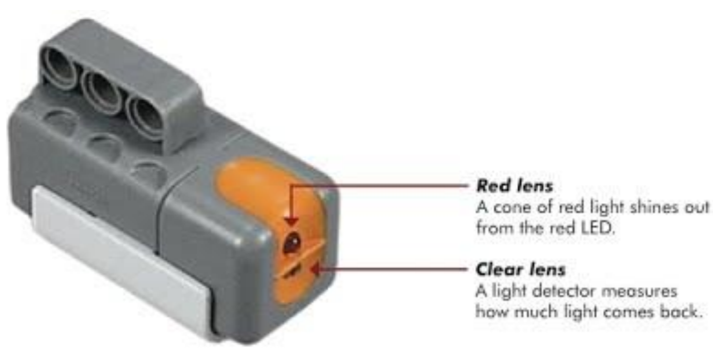


FIGURE 5.1 LEGO Mindstorms light sensor and operation

Another important term is localization. Localization is a catch-all phrase that deals with the abstract idea of position. To help define localization, ask yourself how you would navigate in a city. You may use street sights, local landmarks or just generally know the city by memory. All of these are a form of localization. The localization of a robot is attempting to give the robot an idea of where it is through local information. This local information can be its acceleration, camera feed, or GPS or any form of information that can assist with determining location.

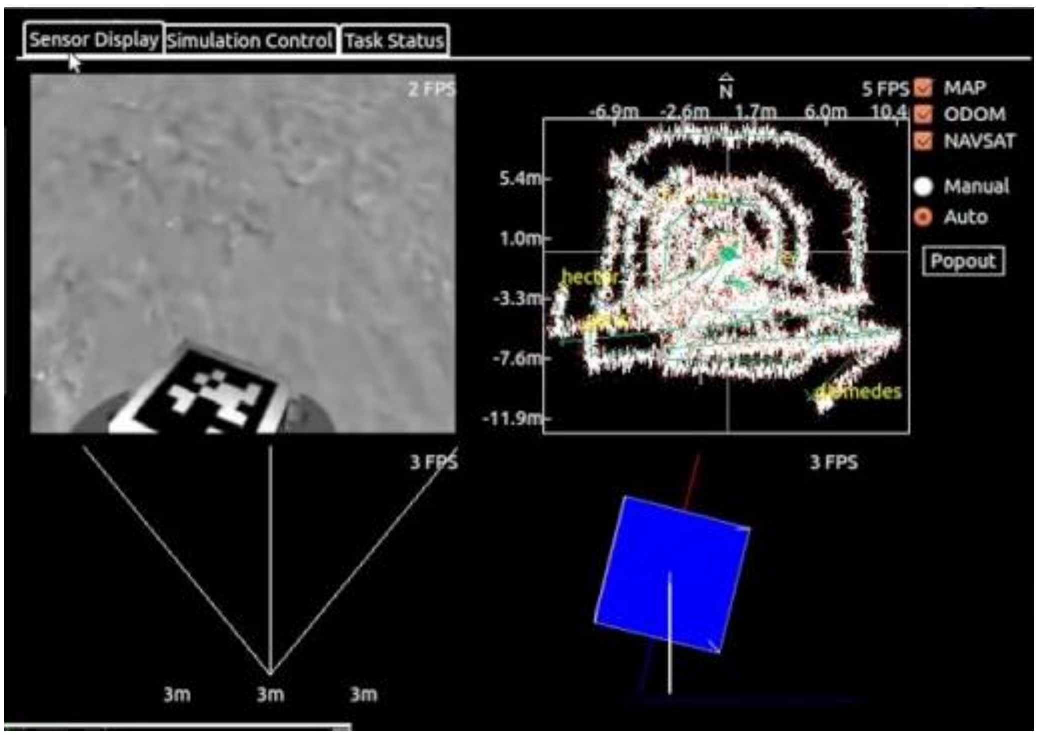


FIGURE 5.2 'Swarmie' GUI displaying localization information.

For our vignettes it is important that we understand the term "degrees of turn." Fairly self-descriptive this addresses the angle cover during a turning operation to come to the desired angle of heading. For example, if we want a robot to turn 90° degrees to the left, according to a unit circle, we want the robot to have a positive 90° degree-of-turn. Alternatively, if we want the robot to turn around then another 30° degrees we need a 180°+30° degree positively or negatively.

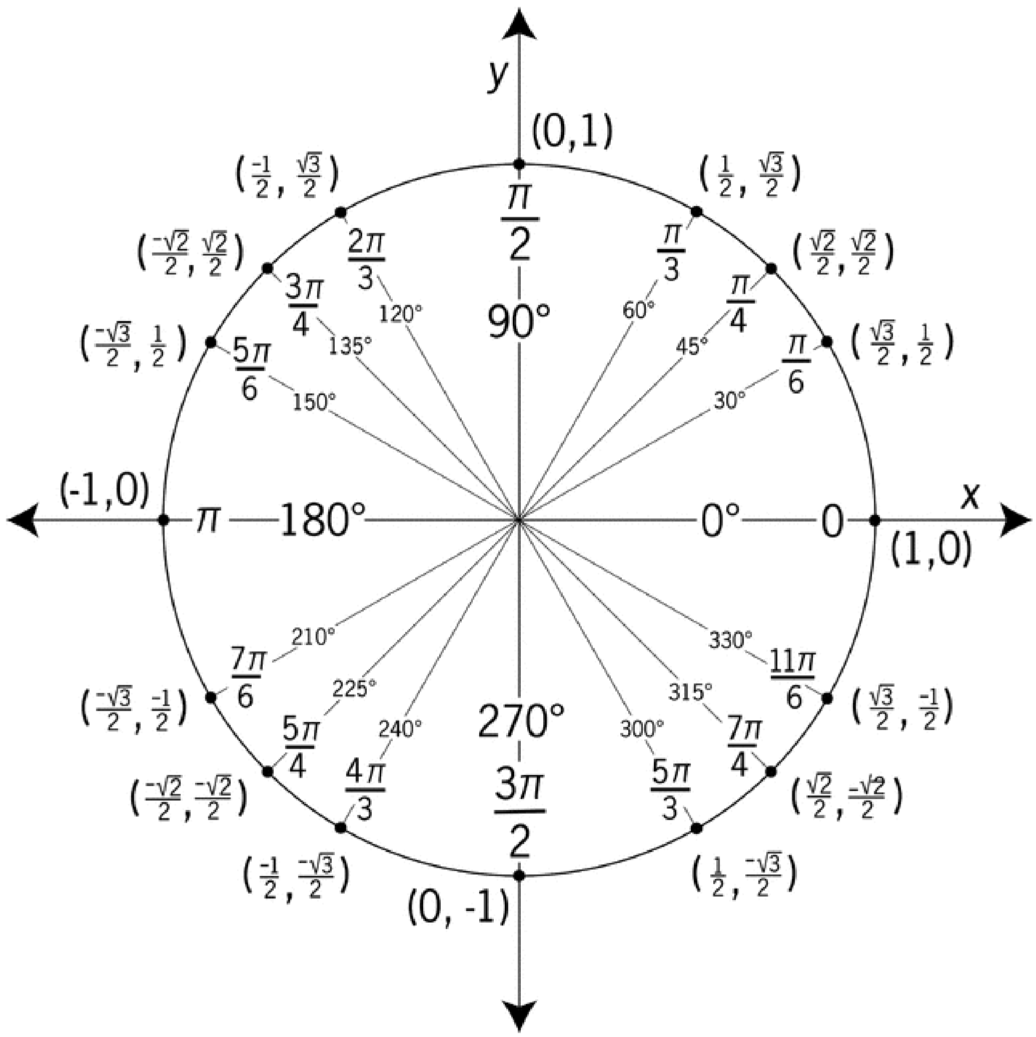


FIGURE 5.3 UNIT CIRCLE used to do degree-of-turn calculations

Another term of great important is actuators. An actuator is roughly any motor used on the robot. These come in many forms; Brushless DC motors, Stepper motors, AC Servo motors. These move apertures of the robot or even the entire robot. For our purposes we will work with two different actuators, two large AC servo motors with onboard sensors, and a medium AC servo motor with onboard sensors.

FIGURE 5.4 Large and Medium AC servo motors, respectively.

Another term used frequently is controller hardware. The controller of the robot is analogous to the human brain. It intakes signals and translates them for use and sends output signals to engage actuators or activate sensors. As with actuators, these come in many forms, analog to digital converters, robotics computer controllers, operational amplifiers. We are only concerned with the computer controller included with the LEGO EV3 Mindstorms set, which suits all our required tasks and goals.

The concept of a Kalman filter will be vital to continuing forwards as it is a programming algorithm that simplifies data input in a more workable format. As we stated earlier sensors collect data, but that does not mean all of it is useful. A Kalman filter is a computer algorithm that takes sensor data and filters 'noise' or unnecessary data into data that matters based on reinforced learning.

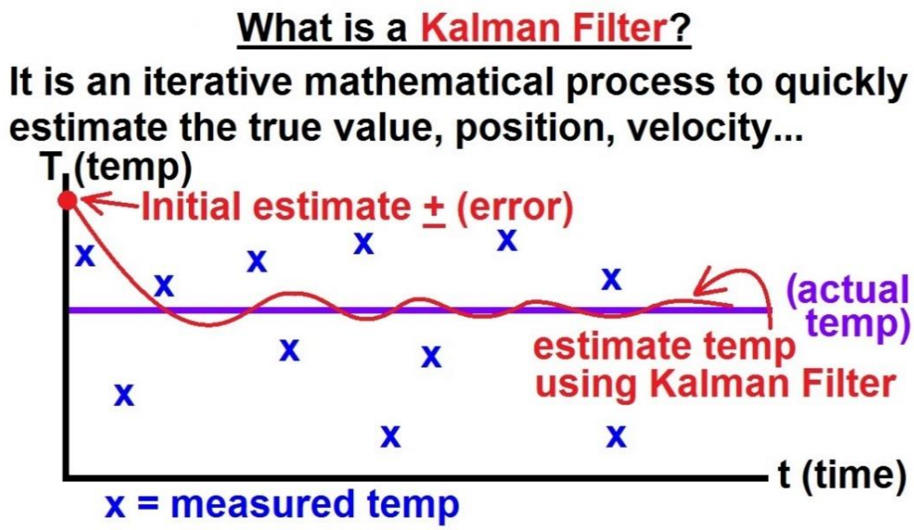


FIGURE 5.5 Kalman filter example with explanation.

The final term we will cover will cause a bit of *déjà vu*, as it is the term 'algorithms'. The definition of algorithms regarding robotics is instead the use of input to determine the robots next course of action. As with programming algorithms, which intake data to determine the programs output data, the concept of intake of data to output of data is similar. But for a robot, the algorithms are running constantly to gather data to determine the next course of action

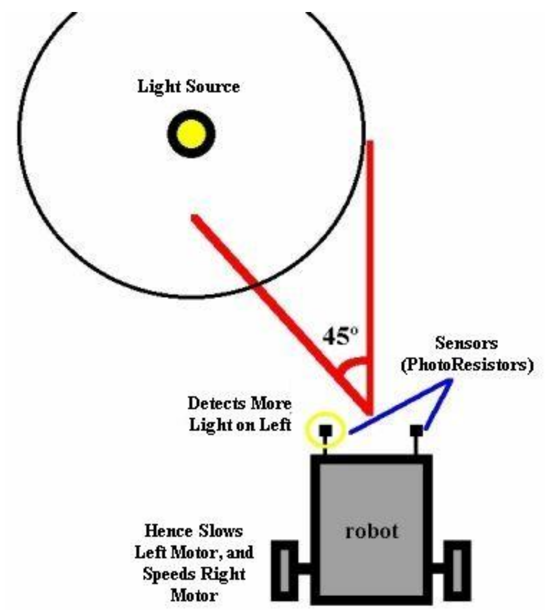


FIGURE 5.5 Robotic algorithm in action to turn towards light

Chapter

6

Swarm Robotics Primer

And now for more need-to-know terms.

Evidently, robotics has many terms. There are hundreds more to learn but now you know the required ones for the projects defined in this manual and to effectively program a basic swarm robot. To that note we have yet to learn the terms needed for swarm robotics specifically, but don't worry with a solid understanding of the terms needed for robotics, the swarm robotics terms come much easier.

Key terms for swarm robotics

Swarm robotics deals mostly in communication, so we shall start exactly there to begin our term databank. For swarm robots 'network communication' (Li, Ming) is vital towards communicating between robots. 'Network communication deals with all form of external signals sent from the robot for the purposes of conveying information to any other machine. Take note that the receiving machining does not have to be a robot to still be considered part of the network.

With network communication understood we can introduce an important concept in the form of a 'hivemind' for a swarm network. A hivemind is a central controller that the swarm robots bounce information from and receive commands from. The 'hivemind' can control the entire group of robots and coordinate the robot, decentralizing their algorithm from the local (only on the robot) to a more global (the entire network) orientation.

Another key term for swarm robotics is 'deployment and complexity'. This defines the number of robots and the general nature and number of sensors on each robot respectively. A 'small deployment' may between two and five robots to the network while a 'large deployment' can be in the realm of twenty to one hundred. This goes hand in hand with the complexity of the group. The Harvard micro bots would be low complexity as they have few sensors per robot and these are very simplistic sensors. Juxtaposed, the Mars rovers *Spirit* and *Opportunity* are high complexity robots, featuring multiple high-level sensors with intense complexity per sensor. By our definition lets attach the term to some of the robotic examples we saw earlier. The Harvard micro bots are a high deployment low complexity network, while the NASA Mars rovers are a low deployment high complexity network (though not really as they aren't a swarm robot group, but let's consider them as such.)

We will define the term 'emergent behavior' for swarm robot groups. This conveys behaviors preformed without any prior definition that the robots display 'emergently.' They are essentially unplanned behaviors executed by the groups that can be beneficial or detrimental to the goal at hand. Think of a robot whose goal is to grab a cube and return it home area. With a basic algorithm the robot will grab the cube, turn to home and drive until it reaches its home. Now suppose there is a cube between the robot with the cube and the home. The robot may accidently push the second cube to the home with its current algorithm. This pattern falls under the definition of 'emergent behavior' as it is entirely unplanned.

Finally, a little more *déjà vu* as we re-introduce an old friend, the term "algorithms." In this situation we are concerned with group algorithms that the robots may want to execute. Instead of dealing with any single robot's sensors this algorithm may deal with the collective groups sensors and execute decisions based on this. A previously the basic definition is the same, the algorithm intakes data from the robots, and executes commands based on this. However, a swarm algorithm may execute certain patterns for the entire group of robots as opposed to a single set of code, or single robot.

Chapter

7

Required Materials for Proceeding

What is needed for a robot.

To proceed with the next section, it will require a monetary investment. We will buy a robotics kit for learning that includes a set of sensors and materials that will ease our introduction. While the kit is defined as a "child's toy" it is an ironically power tool to teach the complex nature of robotics in a manageable fashion.

What is a LEGO Mindstorms kit?

Danish toy manufactures LEGO have released an educational kit geared towards teaching robotics. This kit the "LEGO Mindstorms" kit includes many useful components to build a very basic robot. These kits have been used at both a hobbyist and academic level in recent times due to their cheap cost and extent of teaching capability. The kits may be purchased for around $400 and come with instructions to assemble a series of robots, the LEGO materials to do so, and a few sensors. The price may be high, but the resulting education can be massive.

Why a LEGO Mindstorms kit?

As stated, the LEGO Mindstorms kit is a very low-cost set. But this isn't the only caveat of the kit. The main benefit of the kit is that many open source libraries can interact with the kit and intake data from the sensors. ROS offers a library to interact with the robot and learning how to use such a library is pivotal to our goals. Compared to building a custom robot with Arduino components, a Mindstorms robot offers a more uniform set of components. This is very important as the manual requires standardization to help cover all issues with no otherwise unforeseen lack of information. As well the Mindstorms kit can be assembled within minutes without any tools. This can help quickly set you on the path towards programming the robot avoiding any fiddling time lost with breadboards, wire testing and other setbacks. And if I have no reinforced the idea that the Mindstorms kit is cheap then allow me to restate this one more time. The cheapness is vital to the idea of swarm robotics and allowing for a low-cost medium scale deployment.

Where to buy a Mindstorms kit?

I suggest first investigating your University department head to ask for funding towards buying a set of maybe 3 or 4 LEGO Mindstorms kits. Otherwise they may be purchased online at the LEGO website, Amazon, or physically at hobby stores. Alternatively, many schools have the kits on standby for research purposes and can be rented out from the library or department resources office. Investigate these options to find which is most suitable to your situation.

Chapter

8

Assembling a robot

Let's play with LEGOs again.

The EV3 LEGO Mindstorms kit makes a fantastic starting ground to begin our first robot. It is simple to assemble, comes with a large set of sensors to work with, and is easily expandable for more intense projects. On top of this, there are open source libraries to program the robot. But before all of this we must first make a robot to work with.

Designing a robot

There is no absolute proper way to build the robot we will use. Rather it will build a more robust understanding to teach you how to design a robot suited to the goals. This will have the effect of teaching you exactly what components you are working with, where, and how they operate in a fashion more unique to your own design and understanding. So instead of a step-by-step assembly instruction this will be a general guide toward assembling your robot with your expected materials. This section may be a bit confusing but try your best to assemble the robot according to the general ideas defined here. Even if you can't, this can lead to emergent behavior so if the robot can accomplish the tasks defined in the chapter 8 it can be used to test swarm robotics applications. We will use the EV3 manual parts list to reference parts for use.

Key design components

There are a few absolutely required robotic assembly configurations. The first of which is the inclusion of a chassis to hold the EV3 computer controller. Luckily your kit includes a part 4540797, which is suited to the task. The frame will serve as the central design component by which everything must be attached. I suggest expanding the length with a 4611705 beam towards either end and deeming this the "front" of the robot. Towards the front of the robot attach the large motors to the left and right on the inside of the frame. To the motors attached wheels 4634091 on the outside. Your robot should look comparatively like this figure.

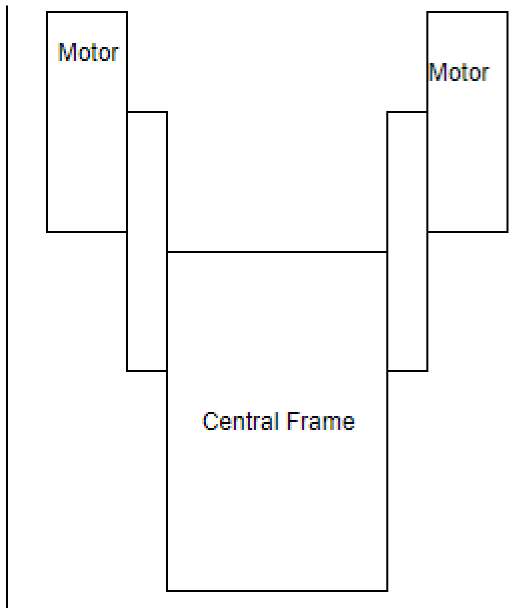
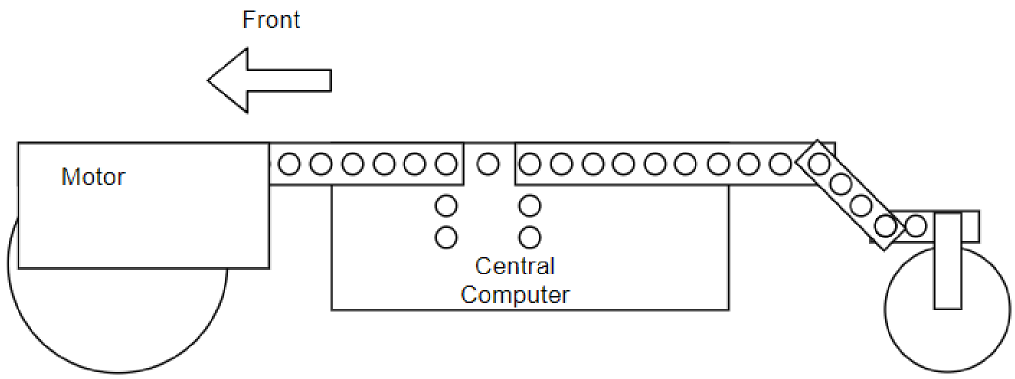


FIGURE 7.1 Robot assembly phase one

We must now support the rear end of the robot with a free turning wheel. As with the extensions towards the front, mirror them insert a 373726 axle with a 4509912 angled beam in between angled down. Then attach a 4142865 to the end followed by a 421175 crossbeam. Now attach a 4107085 angular block. To this block attach wheel 4587275 to complete the assembly. The robot should now resemble the diagram.



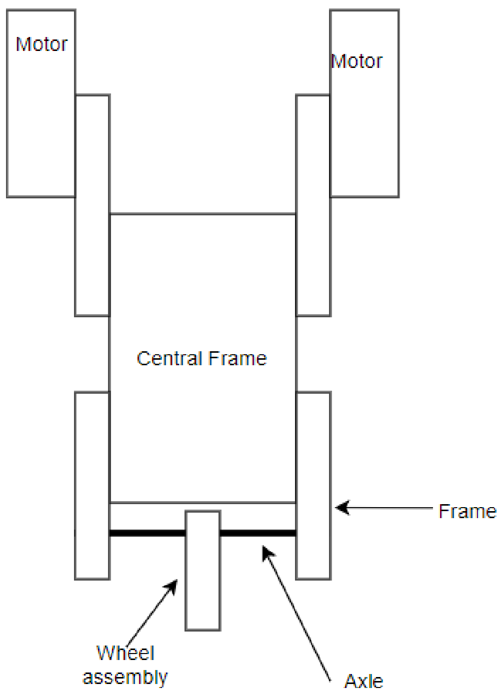


FIGURE 7.2,7.3 Robot assembly phase two

For the last addition attach a touch sensor to one side the motors that reaches well past the wheel diameter. Take note that attaching it to one side will offset the engine speed of that side and cause drifting, this is a challenge solvable by your own ingenuity. Perhaps centralizing the design will mitigate this.

Chapter

9

Basic robot programming vignettes

Recursively running into objects.

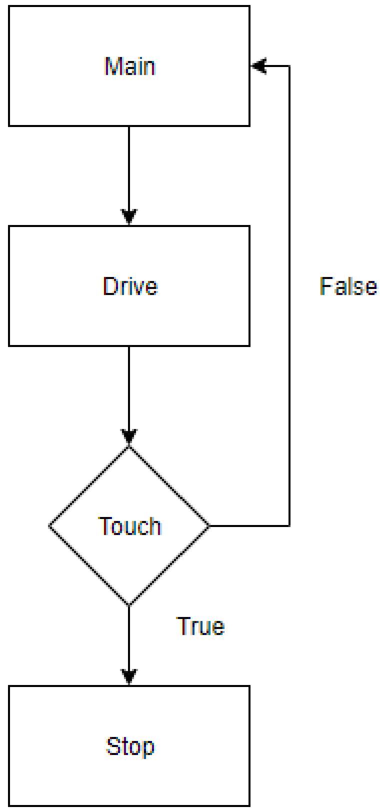
Now that we have a robot to work with we can begin programming it. This section will teach a very basic series of programs that can be expanded upon. It will offer very basic robot functionality based on general algorithms to use. Take note that this section will only address one robot and commanding that robot with onboard controls. We will avoid communication networks for now to first understand how a single robot operates.

Your first robot program

Learning a new skill means you start with the basics, after all most crawl before running. In this case, we'll be running into walls before we do anything else. Our robot should have an attached touch sensor, this sends a signal as to if is pressed or not. So, for our first program we'll send a drive command to the wheels, until it hits a wall.

Program Design

Our program loop will look something like this flowchart:



To break this down, lets start from the top. Our main loop will act as the heartbeat of the robot and run all subsequent code, it calls the drive function. This function will move the robot forward by commanding the motors to move, after which it will call the touch loop. The touch loop will check the state of the touch sensor, if it is not activated or false, it will recursively call the main function starting the loop over. If it instead returns true, the robot will cease all activities.

FIGURE 9.1 Program flowchart for one robot

As with the assembly phase, this chapter is not designed to code the robot for you, but rather supply a framework to do so. The flowchart, your background knowledge of code, and any supplemental ROS based readings should greatly help if you lack the immediate necessary skills.

Designing a swarm robot program

This program can be expanded to have swarm robot capabilities. Not necessarily in the traditional sense, but as a test to try network communication between systems. This flowchart would mirror the design of a program meant to do so.

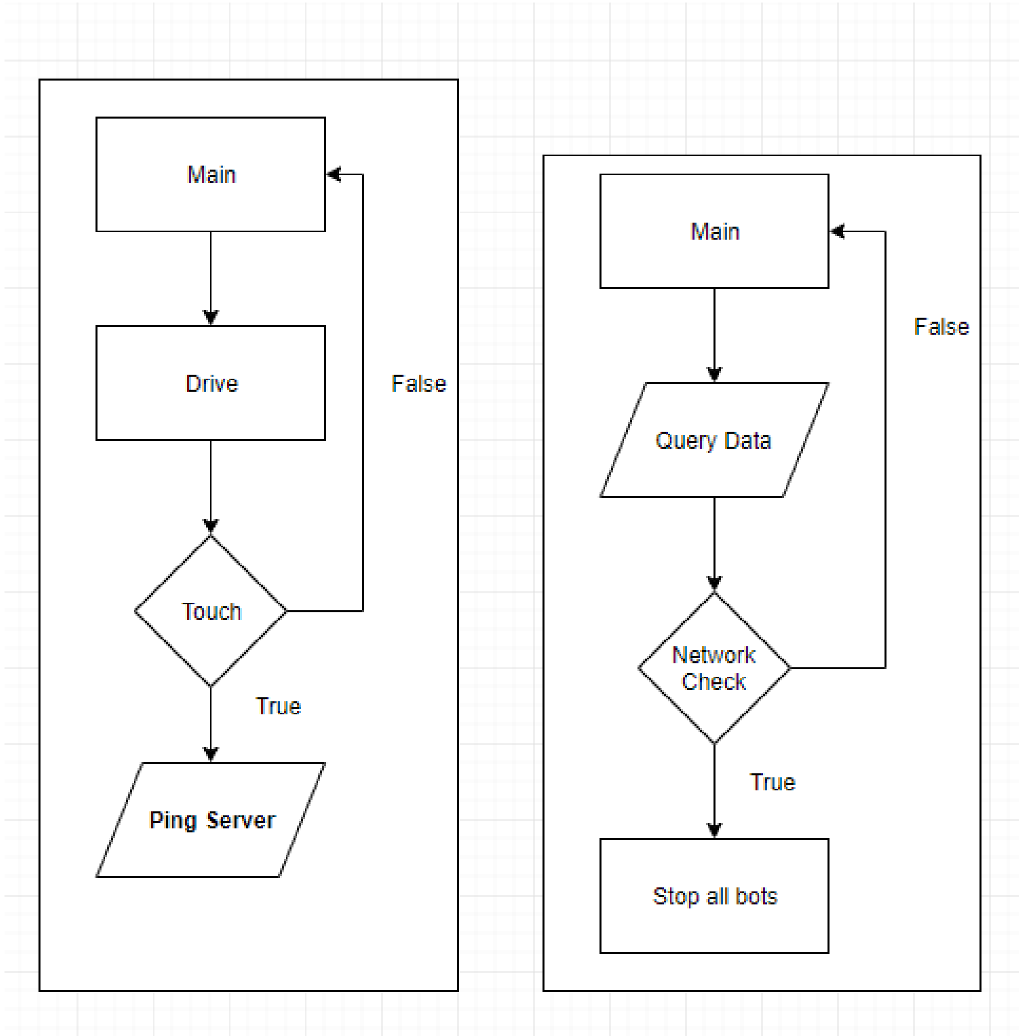


FIGURE 9.2 Program Flowchart for a communicated environment

This example can then be expanded to instead command another robot through the network check loops. If you have the opportunity further expand this program to create a more dynamic swarm robot behavior, maybe instead having the computer command the robot to turn until a certain number of walls have been hit.

Chapter

11

Conclusion

The road to a swarm robotics expert.

COngratulations! If you have made it to this chapter, it means you have some a rudimentary understanding of regarding swarm robotics. You can now be considered "introduced" to the concepts. But the road of swarm robotics is a long arduous one, and it does not end here.

What next?

Now that you have reached the end of the manual I suggest you begin your own experimenting. We only employed two sensors for the robots to use at maximum, and thus there more sensors available for you to use. Outside of this, third party sensors can increase your robot complexity ten-fold, maybe an RGB camera sensor will drastically change how you approach the vignettes. Past this experiment with your robots and their capabilities. Maybe specialize the robots so that one robots preform only one task, another approach of swarm robotics. The multiple robots offer you a plethora of opportunities to do interesting applications and find new emergent behaviors. Outside of the kit, seek our swarm robotics resources for learning and to teach you new skills. But the most important part is to move beyond this manual, seek more information and keep making bigger, and better swarm robots. After all, nature never stops evolving, and swarm robots are nature inspired!

Bibliography

Acieta. n.d. *Assembly Robotics.* Accessed 5 10, 2018. https://www.acieta.com/automation-application/assembly-robotics/.

Dongdong Xu, Xingnan Zhang, Zhangqing Zhu, Chunlin Chen, and Pei Yang. 2014. "Behavior-Based Formation Control of Swarm Robots." *Mathematical Problems in Engineering.*

ENEA. 2015. *Innovation Technology: Robotic fish to protect Venice from rising waters.* 10 22. Accessed 5 10, 2018. http://www.enea.it/en/news-enea/news/innovation-technology-robotic-fish-to-protect-venice-from-rising-waters.

Exploration, NASA Mars. n.d. *Spirit and Opportunity.* Accessed 5 10, 2018. https://mars.nasa.gov/programmissions/missions/present/2003/.

Ferdochuk, Denis, interview by Michael Roa. 2018. *2018 Swarmathon Challenges* (04 05).

Ferrer, Eduardo Castello. n.d. *TThe blockchain: a new framework for robotic swarm.* Cambridge: MIT Media Lab.

Frike, Matthew, interview by Michael Roa. 2018. *Challenges of 2018 Swarmathon* (04 18).

Iñaki Navarro, Fernando Matía. 2013. "An Introduction to Swarm Robotics." *ISRN Robotics*, 10.

Intel. n.d. *Intel Shooting Star.* Accessed 5 10, 2018. https://www.intel.com/content/www/us/en/technology-innovation/aerial-technology-light-show.html.

Kuijt, Dr. David, interview by Michael Roa. 2018. *MC Swarmathon 2018 Challenges* (March 10).

M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo. 2012. *http://iridia.ulb.ac.be/IridiaTrSeries/link/IridiaTr2012-014.pdf.* Technical, Bruxelles: Universite Libre de Bruxelles.

M.Habibullah Pagarkar, Kaushal Parekh , Jogen Shah , Jignasa Desai , Prarthna Advani , Siddhesh Sarvankar , Nikhil Ghate. n.d. "AUTOMATED VEHICLE CONTROL SYSTEM." Chembur, Mumbai.

Markets, Research and. 2018. *PR New Wire.* Jan 08. Accessed 03 26, 2018. https://www.prnewswire.com/news-releases/global-adaptive-robotics-market-2017-2021-market-to-grow-at-a-cagr-of-272---key-vendors-areirobot-rethink-robotics-softbank-group-universal-robots--yaskawa-motoman-300579051.html.

Michael Rubenstein, Alejandro Cornejo, Radhika Nagpal. 2014. "Programmable self-assembly in a thousand-robot swarm." *Science*, 345.

Ming Li, Kejie Lu, Hua Zhu, Min Chen, Shiwen Mao, B. Prabhakaran. 2008. "Robot swarm communication networks: Architectures, protocols, and applications." *Communications and Networking in China.*

Moses, Mealine, interview by Michael Roa. 2018. *Challenges of 2018 Swarmathon* (04 18).

NavalDrones. n.d. *LOw-Cost Unmanned aerial vehicle Swarming Technology (LOCUST) .* Accessed 5 10, 2018. http://www.navaldrones.com/LOCUST.html.

Nield, David. 2018. *Science Alert.* 04 04. Accessed 04 04, 2018. https://www.sciencealert.com/nasa-wants-swarms-buzzing-robot-bees-marsbees-to-explore-mars.

Omega, Drone. n.d. *How GPS Drone Navigation Works.* Accessed 5 10, 2018. http://www.droneomega.com/gps-drone-navigation-works/.

Reinhard Gerndt, Stefan Krupop. 2010. *Research Gate.* Jan. Accessed March 26, 2018. https://www.researchgate.net/publication/228458125_EcoBe_Mixed_Reality_Robot_Kit-An_Entry-Level_System_for_Teaching_Cooperative_Robotics.

Sadowski, Ed. 2012. *Is a toaster a robot?* April 18.

Sean, interview by Michael Roa. 2018. *Challenges of 2018 Swarmathon* (04 05).

Tan, Ying. 2013. "Swarm Robotics: Collective Behavior Inspired by Nature." *Journal of Computer Science & Systems Biology.*

Ying Tang, Zhong-yang Zheng. n.d. "Research Advance in Swarm Robotics." *Defence Technology*, 18-39.