Imperial College London

MEng Individual Project

Imperial College London

Department of Computing

TODO

Author:
Akshat Tripathi

Supervisor:
Prof. Andrew Davison

Second Marker:
Prof. Antoine Cully

January 26, 2023

Abstract

TODO

Acknowledgements

TODO

Contents

List of Figures

Chapter 1

Introduction

1.1 Objectives

The field of distributed robotics is a growing one, partly due to the growth of the number of applications involving it (autonomous vehicles and drone delivery systems) and partly because of increased interest in areas which would greatly benefit from it, such as the Lunar Gateway Project, or the Mars 2020 Perseverance Mission.

An open problem in this distributed robotics is that of effective inter-robot communication. Inter-robot communication provides several benefits to distributed robotics; it allows robots to 1. acquire a more accurate view of their environment, 2. gain access to a larger map of the world, 3. improve their path planning by incorporating others' plans and 4. carry fewer resources, such as sensors and instead rely on others in the swarm.

A subset of distributed robotics research is dedicated to finding ways to allow such communication in situations where some of the robots may act with hostility. The source of this hostility could be unnatural, where a bad actor solely seeks to disrupt the system, or it could be a natural consequence of competition in the environment, such as a self-driving car trying to prevent others from overtaking it.

This hostility must be protected against if the benefits of communication and collaboration are to be seized. This is the principal focus of this Master's Thesis.

1.2 Challenges

1.3 Contributions

Currently, the main contributions of this paper can be found in the background research section and some of the preliminary results on the effectiveness of attacks on robot networks. This is subject to change with time.

Chapter 2

State of the Art

The first half of this chapter will provide the theoretical background for this thesis. First, we will discuss the field of multi-robot systems, providing the reader with an understanding of how the field has evolved and how it may further evolve. Next, we examine RobotWeb [1], the research that this thesis seeks to build upon. Finally, we discuss various security issues present in the field to arrive at the research question for this thesis.

2.1 Multi-Robot Systems

The study of multi-robot systems concerns itself with studying how to allow multiple robots to operate in the same environment [2]. Multi-robot systems have several advantages over single-robot systems; they are more effective, efficient, flexible, and resilient [3]. These robots can behave competitively or collaboratively, coordinate statically or dynamically, communicate explicitly or implicitly, consist of homogeneous or heterogeneous robots, and make decisions centrally or decentrally [4].

2.1.1 Competitive vs Collaborative Behavior

Multiple robots which share a common goal are considered to be behaving collaboratively, whereas if each robot aimed to complete its own goal at the expense of all others, it would be said to be behaving competitively [4]. Examples of collaboration range from teams of robots constructing a lunar habitat [5] to exploring unknown environments [6].

2.1.2 Static vs Dynamic Coordination

If a multi-robot system operates using a set of predetermined rules, then it can be said to be coordinating itself statically. A possible set of rules would be that each robot must maintain a certain distance between it and all others. Dynamically coordinated multi-robot systems would instead make decisions whilst performing the task and may communicate to do so [4].

2.1.3 Explicit vs Implicit Communication

Most multi-robot systems communicate explicitly by sending messages to each other via a hardware communication interface, for example, a wifi module [4]. However, there is still a sizeable minority of approaches that send messages through their environment (implicit communication) and rely on others to sense these messages to receive them. An example of implicit communication is found in [7], where the authors use it to allow a team of robots to play a game of football for the RoboCup Simulation League [8].

2.1.4 Homogeneous vs Heterogeneous Robots

Multi-robot systems can either contain robots with identical hardware, which are known as homogeneous systems, or individual robots may have different hardware, making them heterogeneous

systems. Heterogeneous systems allow a greater degree of specialisation within a multi-robot system but also add additional decision-making complexity.

2.1.5 Centralised vs Decentralised Decision Making

A multi-robot system is said to have centralised decision-making if all robots communicate with a central agent, which may or may not be a robot itself, to receive instructions. Centralised schemes perform better with smaller groups of robots and in static environments, they also introduce a single point of failure in the central agent [4]. Decentralised schemes, however, avoid vesting authority into a central agent and instead treat each agent as an equal part of the system, which allows them to avoid the problems associated with centralised schemes. However, decentralised schemes lose the guarantee that they will converge to an optimal solution, as decisions are made with incomplete information. In addition to centralised and decentralised schemes, multi-robot systems may also be organised in a hierarchical manner, where some robots would be chosen as local leaders, but no global leader would exist.

2.2 Robot Web

This thesis seeks to build upon the work done in "A Robot Web for Distributed Many-Device Localisation" [1], which describes a method for *heterogeneous* robots in a *decentralised* multi-robot system to *collaborate* via *explicit communication* to localise *dynamically*.

Robots in the Robot Web move along predefined paths, estimating their location via internal odometry. When a robot senses another, it communicates its measurement to the other robot, and then both robots use the measurement to update their location estimates. Since we live in an imperfect world, each sensor measurement carries with it a small amount of noise, which is reflected in the Robot Web by a degree of uncertainty attached to each robot's location estimate and represented by a Gaussian distribution.

This section will introduce the reader to the core concepts used in the Robot Web and assemble them to provide the reader with an understanding of how the Robot Web functions and some of its limitations.

2.2.1 Factor Graphs

A factor graph is an undirected bipartite graph used to represent the factorisation of a probability distribution $p(X)$. A probability distribution can be said to be factorised if it is written in the form:

$$p(X) = \prod_i f_i(X_i) \tag{2.1}$$

The nodes of a factor graph can either represent variables (X_i) or factors (f_i). There are several different ways to draw factor graphs, but we will use the one defined in [9], where factors are drawn as squares and variables are drawn as circles.

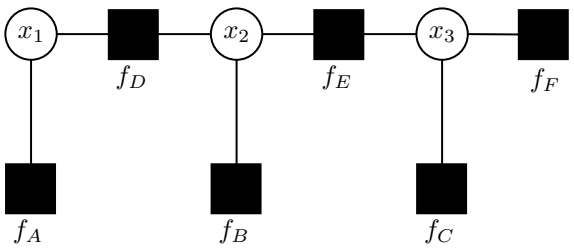

Figure 2.1: An example of a factor graph

The above factor graph represents the following factorisation:

$$p(X_1, X_2, X_3) = f_A(X_1)f_B(X_2)f_C(X_3)f_D(X_1, X_2)f_E(X_2, X_3)f_F(X_3) \tag{2.2}$$

Assuming that each variable takes discrete values, suppose we wanted to find the probability that $X_1 = z$ for some value of z using the above factor graph. Then we would need to find:

$$p(X_1 = z, X_2, X_3) = \sum_{i=X_2} \sum_{j=X_3} p(X_1 = z, X_2 = i, X_3 = j) \tag{2.3}$$

And by 2.2 we get:

$$p(X_1 = z, X_2, X_3) = \sum_{i=X_2} \sum_{j=X_3} f_A(z)f_B(i)f_C(j)f_D(z, i)f_E(z, j)f_F(j) \tag{2.4}$$

which can be rearranged to form:

$$p(X_1 = z, X_2, X_3) = f_A(z) \sum_{i=X_2} \left(f_D(z, i)f_B(i) \left(\sum_{j=X_3} f_E(z, j)f_C(j)f_F(j) \right) \right) \tag{2.5}$$

Similarly, if we wanted to find the probability that $X_2 = z$ for some z we would need to find:

$$p(X_1, X_2 = z, X_3) = f_b(z) \left(\sum_{i=X_1} f_D(i, z)f_A(i) \right) \left(\sum_{j=X_3} f_E(z, j)f_C(j)f_F(j) \right) \tag{2.6}$$

Noticing how the sum over X_3 in both 2.5 and 2.6 is the same, we may want to "cache" the result when dealing with large factor graphs, to improve performance. To do this we can associate calculations with nodes in the factor graph. We call these associations "messages".

The general form of a message from variable i to factor j is the product of the messages from all other neighbouring factors [10]. Put formally:

$$m_{x_i \to f_j} = \prod_{s \in N(i) \setminus j} m_{f_s \to x_i} \tag{2.7}$$

The general form of a message from factor j to variable i is the product of the messages from all other neighbouring variables and the factor applied to all other variables except i [10]. Put formally:

$$m_{f_j \to x_i} = \left(\sum_{X_j \setminus x_i} f_j(X_j) \right) \left(\prod_{k \in N(j) \setminus i} m_{x_k \to f_j} \right) \tag{2.8}$$

Finally, the marginal value of a variable is simply the product of all incoming messages to it [10].

$$p(x_i) = \prod_{s \in N(i)} m_{f_s \to x_i} \tag{2.9}$$

2.2.2 Belief Propagation

The above equations are used by the Belief Propagation algorithm, an iterative message-passing algorithm used to calculate the marginal value for each variable in a factor graph [10]. Each iteration of Belief Propagation has 3 phases:

1. Variables send messages to each of their neighbouring factors 2.7.
2. Factors send messages to each of their neighbouring variables 2.8.
3. Each variable updates its "belief" (its estimated marginal value) 2.9.

The original Belief Propagation algorithm was designed to be used in tree-like graphs, i.e. graphs without loops [10]. However, empirical evidence has shown that "Loopy-BP" can still converge to provide useful results in a variety of problem domains [10].

2.2.3 Gaussian Belief Propagation

A special case of the Belief Propagation algorithm is Gaussian Belief Propagation, which applies to problems where all variables follow a Gaussian distribution, and all factors are Gaussian functions of their inputs.

Under Gaussian Belief Propagation, each message can be interpreted as a Gaussian and so must contain sufficient information to produce one. A naive way of achieving this is to include a mean vector and a covariance matrix in each message. However, this approach is computationally expensive as it requires a full matrix multiplication whenever messages are multiplied which is an order $O(n^3)$ operation. An alternative approach is to use the *canonical form* of the multivariate Gaussian distribution.

The canonical form uses an *information vector* (η) and a *precision matrix* (Λ) defined as follows:

$$\eta = \Sigma^{-1}\mu \qquad\qquad \Lambda = \Sigma^{-1}$$

where Σ is the covariance matrix and μ is the mean vector. Now multiplying messages is made more efficient as it only requires the addition of both messages' η and Λ values, making it an order $O(n^2)$ operation in the worst case. A further performance improvement can be made by recognising that the precision matrix is a sparse matrix [10].

2.2.4 Lie Theory

Lie theory is a subset of group theory focussed on studying *Lie groups*. Lie theory is a vast and abstract field, from which we only need to borrow a few concepts. The first is that positions and rotations can be represented as Lie groups, for example, the group $SO2$ represents a rotation in 2D space and the $SE3$ group represents a rigid motion in 3D space. The second core concept is the *tangent space* which allows small deviations to be applied to the Lie group uniformly regardless of the value it operates on. This concludes our whirlwind tour of Lie theory, we invite the reader to read [11] for a more detailed tutorial.

2.2.5 Putting it all together

Now that we have covered all of the prerequisites to understanding how the Robot Web operates, we shall now demonstrate how they can be assembled into the Robot Web.

Every robot in the Robot Web needs to estimate its current location at all times, this is called localisation. One simple localisation method is to use odometry, which uses internal sensors to measure its displacement from its previous location. Since no sensor is perfect, this introduces a small amount of noise, which can be accurately modelled using a Gaussian distribution. The Robot Web simulates odometry using a factor graph, each known position of the robot maps to a pose variable, and the variables of each pair of successive positions are connected by an odometry factor.

On every timestep, the robot performs an iteration of Gaussian Belief Propagation to estimate its current position.

The Robot Web further improves the accuracy of robots' locations by allowing robots to measure each other using external sensors. When a robot senses another, it creates a factor in its factor graph between its and the other's latest pose variables. When each robot wants to send a message to another, it publishes the message to its **Robot Web Page**, which the other robot will eventually read and use to update its location estimate.

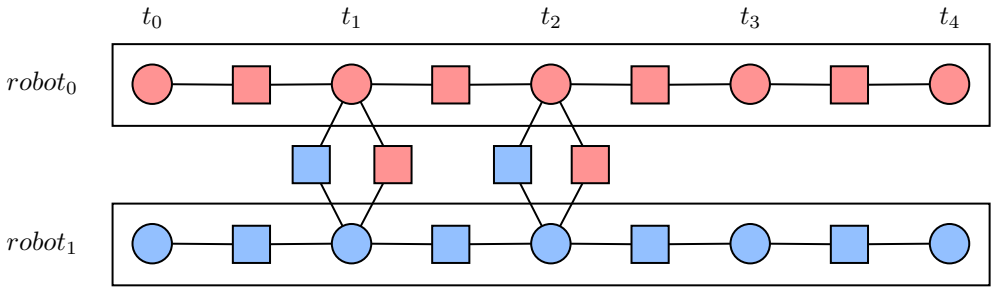

Figure 2.2: An example of a factor graph in the Robot Web. Each robot's variables are connected by odometry factors. At times t_1 and t_2, both robots sense each other, and so exchange measurements by creating inter-robot-measurement factors on the graph.

The Robot Web represents the locations and sensor measurements of all robots using general Lie groups, rather than any specific group. This has the consequence that any type of sensor or robot can be a part of the Robot Web. For example, a drone moving in 3D space can interact with a car moving on a plane.

2.2.6 Evaluation

The Robot Web has been shown to improve the accuracy of robot localisation, and most importantly the inter-robot measurements have been shown to provide further improvements over schemes where robots would only measure landmarks.

Furthermore, the Robot Web has proven to be robust to a large number of faulty inter-robot sensors reporting random measurements, with this robustness lasting until 70-80% of inter-robot sensors reported corrupted measurements.

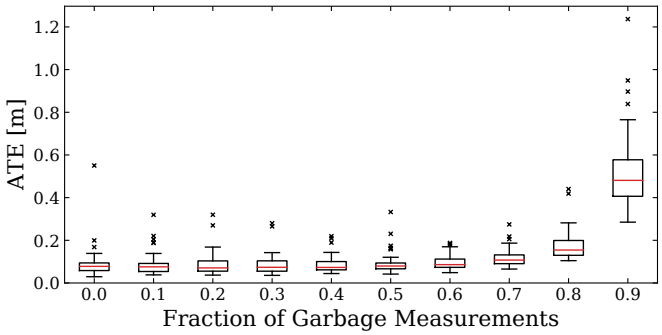

Figure 2.3: A graph showing that the RobotWeb is robust to up to 70-80% of "garbage" measurements, where faulty sensors report random measurements. ATE refers to the average Absolute Trajectory Error measured over 50 runs in an environment with 50 robots and 10 beacons running for 100 timesteps. Taken from [1, Figure 5]

Although the Robot Web is robust to many inter-robot sensors reporting random measurements, it is not robust to a bad actor which may instead report incorrect measurements designed to worsen the localisation of other members of the Robot Web. Possible attacks include but are not limited to:

1. Sending messages with extremely low standard deviations, to lull others into a false sense of security.
2. Sending these messages whilst assuming the identity of another robot.
3. Sending these messages from many nonexistent robots, also known as a Sybil attack.

2.3 Security Issues

In this section, we will discuss several general security issues that can arise in robot networks. We will focus on issues that affect the accuracy of a robot's internal model of the world. This excludes attacks which may result in an attacker gaining control over a robot, yet includes attacks performed by hijacked robots.

2.3.1 Denial of Service

A denial of service attack seeks to deny service. In a robot network such as the Robot Web, this would prevent one or more robots from being able to access messages sent by their peers, and essentially cut them off from the network.

The simplest way for an attacker to perform a DoS attack is to use a signal jammer, which can be constructed using off-the-shelf equipment [12]. This would continuously transmit signals within the range of frequencies allowed by the communication medium, both interfering with and irrecoverably corrupting any messages sent. More sophisticated attackers may craft harder-to-detect jammer attacks by mimicking legitimate messages or only transmitting when it senses communication [12]. In addition to these, there exist a whole host of jamming attacks (and defences) targetting specific communication protocols.

Another form of a DoS attack would be to disconnect specific robots from the network, using the network protocol's existing defences. For example, by convincing others that the target is a bad actor, triggering their defences to remove the target from the network.

2.3.2 Identity-Based Attacks

Identity-based attacks can wreak havoc on robot networks. Using the model defined by Douceur [13], we can describe a robot network as one consisting of E entities (robots) each claiming at least one identity i from the set of all identities I. When the network is under an identity-based attack, at least one of the following two properties will hold:

1. Two entities, $e1, e2$ will present the same identity i
2. The number of identities in I will exceed the number of entities in E.

If only the former holds, then the network is under a spoofing attack, whereas if only the latter holds, then the network is under a Sybil attack. Note: there is no guarantee that only one of these will hold at a time, and so we must be prepared to defend against both simultaneously.

Spoofing Attacks

Devices exchange information by sending packets or frames of data. For our purposes, we can ignore the differences between packets and frames, and use the terms interchangeably. Packets are used to encapsulate the data sent with relevant metadata, such as the source and destination IDs of the packet. This metadata is the main target of spoofing attacks.

In a spoofing attack, the attacker first finds the identity of a legitimate device, for example by first intercepting packets and then extracting the source ID from them. After this point, the attacker sends misleading packets impersonating the target.

This can have several benefits for the attacker. Firstly, they could covertly inject misinformation into the network by impersonating an already trusted robot. Secondly, they could trigger defences in the network to flag target for misinformation and remove it.

Sybil Attacks

In a Sybil attack, the attacker will create many fake identities to gain undue influence on the network. In a robot network, this would allow them to indirectly influence the actions of victim robots. For example, one could lead two self-driving cars to conclude that their best course of action is to crash, by claiming many false identities would be hit if they were not to.

Douceur [13] proves Sybil attacks are always possible in distributed systems where there is no central arbiter of truth. They present and prove four lemmas about identities in large-scale distributed systems. The first two lemmas are concerned with entities which directly validate the identities presented to them, whilst the second two lemmas involve entities which rely upon other, potentially untrustworthy, identities for validation. The lemmas are as follows:

1. If an attacker has ρ times as many resources as the weakest entity, then they can successfully present up to $\lfloor \rho \rfloor$ distinct identities.
2. If an entity doesn't validate all identities simultaneously, then an attacker can present an unbounded number of distinct identities.
3. If an entity trusts q other identities to validate an identity, then at least f attackers are required to perform the attack, where $f >= q$ or the resources commanded by the attackers exceed $q + f$ times the weakest entity's resources.
4. If all c non-attackers don't coordinate when they validate identities, and an entity again trusts q other identities for validation, then even a weak attacker can present $\lfloor \frac{c}{q} \rfloor$ identities.

We plan to solve this problem by treating the physical world as a central arbiter of truth, albeit with some degree of error, due to imperfections in sensors.

2.3.3 Physical Attacks

Finally, since this thesis focuses on the security of robot networks, we will discuss the possibility of physical attacks on the system, and our limited ability to protect against them. We define a physical attack as any attempt to compromise the ability of a robot to perform its task. This includes colliding with the robot but also includes more subtle tactics, such as obscuring the robot's sensors. Physical attacks could also be used similarly to Sybil attacks, where several attackers would surround a robot and feed it misinformation.

In this thesis, we will not attempt to protect against attacks where several robots would physically collide, since this would require heavy hardware modifications to existing robots. Instead, we will focus on occlusion attacks and physical Sybil attacks, at the very least allowing a robot to detect them.

Several different techniques have been explored in preventing the types of identity-based attacks discussed in this chapter. In the 2$^{\text{nd}}$ half of this chapter, we will examine and evaluate these. We broadly group these approaches into 2 main groups; the first uses the physical characteristics of signal propagation to bind an identity to an entity, whilst the second exploits the fact that no entity can perform an unlimited amount of computation.

2.4 Wifi Fingerprinting

There are many ways for devices to communicate wirelessly, many of which use radiowaves. Wifi is the name of a family of networking protocols that allow for this, it derives from the IEEE 802.11 standard. In order to use Wifi, a device must have at least one wireless antenna which can transmit and receive within the bands specified in the IEEE 802.11 standard; usually 2.4GHz and 5GHz.

When a device sends a packet using Wifi, it transmits a radio signal for a given number of nanoseconds from its antenna. This signal will attenuate as it travels further and further through space. After a certain distance, also known as the communicating range of the antenna, the signal will fade into background noise. The signal leaves the antenna in all directions simultaneously, as a

radio wave. Eventually, a small part of this wave will reach the receiver's antenna and the packet will be decoded out of it.

Other parts of the wave will either diffract around corners, reflect off some large objects, or scatter off many small objects [14]. Consequently, this means that a receiver may receive a packet from several different directions simultaneously, that is that it could encounter different parts of the same wave from different directions at the same time. This phenomenon is known as multipath scattering.

Multipath scattering has two interesting properties; it is practically impossible to predict, ahead of time, the distribution of signals around a receiver and it is unlikely that two receivers will observe the same signal propagation with sufficient multipath scattering. These properties allow for devices to "fingerprint" every packet they receive, such that two different transmitters cannot have the same fingerprint unless they are simultaneously located at the same place.

However, implementing multipath scattering-based algorithms have some technical constraints; namely that the receiving antenna must be able to measure the signal in each direction, and that a sufficient amount of multipath scattering must occur. Usually, these algorithms struggle in outdoor environments, where the environment may not provide objects for multipath scattering to occur.

The following papers discuss methods to circumvent these constraints, focussing on securing robot networks from spoofing and Sybil attacks.

2.4.1 Guaranteeing spoof-resilient multi-robot networks

Gil et al, [15] provides an interesting approach to some of the aforementioned problems, most notably the problem of requiring expensive hardware to allow the receiving antenna to measure the signal in each direction. They do this by inventing an algorithm that allows them to build a "virtual spoofer sensor" only using commercially available wifi hardware, which creates a "spatial fingerprint" from each transmission in the network. They use the output from this sensor to calculate a confidence metric α indicating their algorithm's confidence that a robot's identity is its entity. Finally, they characterise the theoretical performance of the "virtual spoofer sensor" and provide empirical evidence to support their claims by undertaking several experiments.

The authors build the "virtual spoofer sensor" by building upon *Synthetic Aperture Radar* (SAR) techniques, which allow a single antenna to be used to simulate an antenna array. SAR involves moving the antenna to different locations and taking snapshots of the signals received. These snapshots are then combined using signal-processing techniques to emulate a multi-antenna array [16]. The "spatial fingerprint" calculated, is then compared to the fingerprints of other clients, and clients with identical fingerprints are assumed to be Sybil attackers.

The authors evaluate their algorithm in the context of the following problem statement. Given an environment with several "clients", each expecting service from mobile "servers", dynamically find the optimal layout for the servers such that each "client" is served. A subset of clients are assumed to be malicious and are carrying out Sybil attacks in order to influence the "servers" to move closer to them.

The authors perform four experiments to validate their hypotheses:

1. They compare the performance of the "virtual spoofer sensor" in both an indoor and simulated outdoor environment, as expected, finding that multipath scattering is more effective in indoor settings, but also that adding a single reflector to the environment vastly improves performance.
2. They compare the effect of a stationary, moving, and power-scaling Sybil attacker on the ability of the "virtual spoofer sensor" to correctly classify agents, resulting in no false negatives, but many false positives.
3. They evaluate their system on the multi-agent coverage problem [17], finding that it can provide near-optimal results even when there are $3\times$ more spoofed agents.
4. They apply their system to a drone delivery problem, where the "server" needs to visit each real "client" to deliver a package and again find that their system provides near-optimal

results when there are 3× more spoofed agents.

2.4.2 Lightweight Sybil-Resilient Multi-Robot Networks by Multipath Manipulation

Huang et al. [18] take a different approach to Gil et al. [15]; instead of relying upon the environment to provide multipath scattering, they actively cause it by using backscatter tags. This offers two main advantages over Gil et al.: 1. the environment has a lesser effect on the performance of the Sybil attack detector and 2. the robots' antennae no longer need to move when capturing a fingerprint. Using the captured signal information, the robots again compute a fingerprint per transmission, normalise it to mitigate the effects of any power scaling attacks, and finally compare the normalised fingerprint to those of all others, treating any identities with sufficiently similar fingerprints as Sybil attackers.

Backscatter tags scatter signals that they encounter by rapidly absorbing and reflecting them. Backscatter tags also simplify the fingerprinting process, since robots are no longer required to perform small movements for SAR, nor must the software engage in expensive linear algebra to construct a multi-antenna array.

A key property of backscatter tags is that they operate passively and don't require a power supply. This reduces the cost of implementing this scheme as tags can be simply and inexpensively attached to robots. Another useful property is that the backscattering of the final signal is highly correlated with the positions of the tags, transmitting and receiving antennae, meaning that if two identities have very similar backscattering patterns, they are likely to originate from the same entity.

When a robot receives a signal, it receives a raw signal, which is first smoothed out with a moving average filter, to create the backscattered signal. Then it decodes a message out of the backscattered signal, and uses it, with signal processing techniques to deduce how much each tag contributed to the backscattering, and when each tag was "activated". Then the robot does more filtering to remove any backscattering from the environment, the resulting signal will be used to construct a signature for the transmission.

The authors then evaluate their implementation in both an indoor office environment and an outdoor rooftop environment. They find that their method is virtually indifferent to the surrounding environment, as they measure an average true positive rate of 97.6% and an average false positive rate of 5.1%.

2.4.3 Conclusions

Although both sets of authors extensively test their system, they make some problematic assumptions, which may be exploited by attackers.

1. They do not account for Sybil attacks using multiple antennae, which could transmit the same message at the same time, but with variable power levels. Each antenna would create a different multipath scatter, and if the relative powers between them were varied, then they could theoretically construct many false fingerprints.

2. They also do not account for collaboration between different attackers, which would function similarly to the previous vulnerability, where geographically distributed attackers could simultaneously send the same message, with different power scales, creating another set of false fingerprints. This method could produce a larger range of false identities but may encounter synchronisation problems between attackers.

3. Attackers could leverage methods similar to Huang et al. and physically augment their antennae to manipulate their multipath scattering, for example, one could create moveable barriers to prevent some scattering from occurring.

4. Both sets of authors assume that any noise from the environment will not be malicious; Gil et al. assume that it will follow a Gaussian distribution, whilst Huang et al. assume that standard signal processing techniques would be sufficient to filter it out. Both of these assumptions fail to account for the possibility that coordinated attackers emit noise above background levels, but not so high that it would seriously interfere with transmissions. This

would disrupt every transmission's signature and would prevent any pair of transmissions from looking alike.

2.5 Proof of Work

Proof of Work is underpinned by the insight that no entity in a network can perform unlimited computation. This leads to defences reliant on the idea that generating identities should be computationally expensive, to prevent any entity from being able to create an unbounded number of them. A PoW identity generation scheme would express an identity as the solution to some puzzle, and thus the identity can be validated by checking if it is a valid solution to the puzzle. This imposes two constraints: 1. the puzzle must be hard to solve and 2. the solution to a puzzle must be easy to verify, which means that, formally, the puzzle belongs to the *NP-Complete* class of problems.

A problem with this strategy is that it requires wasting computational resources, which may be in short supply for the embedded systems used in robotics. Furthermore, it requires that normal robots constantly pay a steep price for their security even without the presence of attackers.

Gupta et al. [19] design an iterative algorithm (**GMCom**) that solves the 2 problem, where if attackers spend T resources, whilst J new non-attacking identities are presented, then each non-attacker only needs to spend $O(\sqrt[2]{TJ} + J)$ resources. They refer to this property as the *assymetry* of the algorithm, as attackers must spend many more resources than non-attackers.

GMCom organises identities into a group, where a subset of them form a committee. When a new identity tries to join the group, it must first solve a puzzle set by the committee. Occasionally, the committee will seek to purge all attackers from the group, by issuing a *purge puzzle*, which must be solved by all identities before a new committee is formed, otherwise the identity will be purged from the group. This limits the wasted computation that good entities must perform since they only need to solve a single puzzle when entering a group, and occasionally thereafter, whilst an attacker would need to solve puzzles for each identity it claims, and would need to solve them repeatedly to avoid being purged.

However, this approach is not without its limitations. The authors assume that attackers will always only be able to command a fraction, α, of the total resources of the system, however, the heterogeneous nature of robots means that this may not always be guaranteed. For example, in a drone delivery system, each drone would likely only possess a small amount of computational power, but an attacker may attack the system using their desktop computer.

Chapter 3

Project Plan

3.1 Current State

Since this thesis involves extending the Robot Web, I'll briefly explain how the current codebase works. The codebase simulates a set of agents moving along fixed paths, in the Robot Web. It takes a configuration file as an input, which defines how many agents should exist, what the paths should be, where beacons should be placed and how much noise exists in the system. The simulator can also run in headless mode, where it logs each robot's estimated pose against time, which can be used to calculate summary statistics.

The codebase was originally designed to work with identical "agents" (robots) moving along fixed paths. Since attackers weren't present, the code was able to use a single connected factor graph, rather than a distributed set of fragments, which was a problem as it didn't make it easy for agents to control their message passing. The consequence of this was that it made it harder to implement attacks and defences.

First, I changed the code to allow different types of agents to exist within a configuration. Then I refactored the code to distribute fragments of the factor graph to each agent, which would allow them to attack others/defend themselves by taking control of which messages they send and receive. I've also added a host of different configurable tracks for the robots to move along, which can be seen below. I think will help with experimentation later in the project.

I've also implemented some basic attacks on the system. These include:

1. The "overconfidence attack" where an attacker sends a bad reading with a tiny standard deviation.

2. The "spoofing attack" where an attacker sends these measurements but uses the ID of a different agent.

3. The Sybil attack, where an attacker sends bad readings from many phantom agents. An interesting side note here is that the readings for the Sybil attack don't need to modify the standard deviation in the messages, which means that victims won't be able to easily distinguish them from correct messages.

Right now these attacks are in a basic state since they don't try to defeat any countermeasures that can be used.

Finally, I've written an experiment runner script, which would generate several configurations, run the simulation using them, and graph the results. This is mainly to automate experimentation.

3.2 Milestones

Mid-Feb 2023	Fully finish refactoring taking into account feedback from an early January meeting
End-Feb 2023	Finalise some ideas on how to defend against all attacks on robot localisation
Mid-Mar 2023	Implement the defenses
End-Mar 2023	Exam revision
Mid-Apr 2023	Extend RobotWeb to allow non-robot localisation
End-Apr 2023	Implement/Extend defences for this
Mid-May 2023	Run experiments on physical robots
End-June 2023	Write report.

3.3 Stretch Goals

The main stretch goal I have in mind is to further extend the security to allow robots to share their path information. This will be harder to do than the other defences, as it is only possible to find out if a robot has lied in the future when it either takes or doesn't take the path.

Chapter 4

Evaluation Plan

I plan to evaluate this thesis by simulating many different scenarios, varying the paths the robots will take, the number of attackers in the system, the types of attackers in the system and if I find multiple defences, then varying the different defences too. From these runs, I will compute an error metric measuring the deviation of each robot from its actual path. I then plan to compare the results to scenarios where there are no attackers, no defences and no Robot Web, to see how each effect the localisation of the robots.

I hope to observe that the defences I will implement are effective at preventing attackers from being able to influence the network, this would mean that the case where there are no attackers would have a similar error to the case where there are attackers and defences.

I also hope that the defences are not invasive, and so wouldn't require major changes to the hardware of robots. This includes computational resources, which are likely to be limited.

I would also like to evaluate the defences in a real-world environment.

Chapter 5

Conclusion

5.1 Ethical Considerations

"A new device merely opens a door; it does not compel one to enter"
(Lynn White [20])

From the discovery of coffee leading to the "Age of Enlightenment" to the invention of Boolean algebra creating our modern digital age, history has repeatedly shown us that it is impossible to fully understand the implications of new discoveries and nascent technology. With this in mind, we provide a short discussion of potential ethical issues which, we believe, may arise as a consequence of the research conducted in this thesis.

As with all research enabling autonomous robotics, we must consider potential military applications. Many militaries today already use unmanned aerial combat vehicles in their operations, if they were to incorporate this research, they may be able to improve their performance by allowing them to share information securely. However, we do not believe that this is likely to occur as militaries tend to have highly centralised structures, where each robot would have prior knowledge about other trusted robots in the network. Whereas our research focuses on providing security to untrusted robots in decentralised networks.

Another potential misuse of this research would be enhancing the capabilities and security of surveillance robots. In this scenario, an authoritarian regime would use robots to continuously monitor their citizens. The robots would communicate with others in their immediate surroundings to coordinate their search and could be vulnerable to cyber attacks where several are hijacked. The hijacked robots would then send incorrect messages to prevent certain areas from being searched. However, it is unlikely that this research would be an ideal candidate for implementing such a dystopia, as a single party would own the robots and would find it much simpler to implement centralised security measures.

Alongside the unethical misuses of this research, there exist several scenarios where it would confound unethical groups. One intended use case is to implement a common robotic infrastructure for autonomous robots owned by many different parties. Here the decentralised nature of the infrastructure would provide asymmetric robustness against hackers or governments seeking to unilaterally disrupt and destroy the infrastructure as they would not be a single point of failure.

In conclusion, there are many scenarios where this research may be misused to the detriment of humanity, yet we are not convinced that this research would be the most appropriate in those examples. Furthermore, given how this research seeks to defend against bad actors, we believe it is more likely to be a benefit to humanity.

Appendix A

First Appendix

Bibliography

[1] Murai R, Ortiz J, Saeedi S, Kelly PHJ, Davison AJ. A Robot Web for Distributed Many-Device Localisation. CoRR. 2022;abs/2202.03314. Available from: https://arxiv.org/abs/2202.03314.

[2] Farinelli A, Iocchi L, Nardi D. Multirobot Systems: A Classification Focused on Coordination. IEEE transactions on systems, man, and cybernetics Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society. 2004 11;34:2015-28.

[3] Roldán-Gómez JJ, Barrientos A. Special Issue on Multi-Robot Systems: Challenges, Trends, and Applications. Applied Sciences. 2021;11(24). Available from: https://www.mdpi.com/2076-3417/11/24/11861.

[4] Yan Z, Jouandeau N, Ali A. A Survey and Analysis of Multi-Robot Coordination. International Journal of Advanced Robotic Systems. 2013 12;10:1.

[5] Stroupe A, Huntsberger T, Okon A, Aghazarian H, Robinson M. Behavior-based multi-robot collaboration for autonomous construction tasks. In: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems; 2005. p. 1495-500.

[6] Burgard W, Moors M, Fox D, Simmons R, Thrun S. Collaborative multi-robot exploration. In: Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065). vol. 1; 2000. p. 476-81 vol.1.

[7] Pagello E, D'Angelo A, Montesello F, Garelli F, Ferrari C. Cooperative behaviors in multi-robot systems through implicit communication. Robotics and Autonomous Systems. 1999;29(1):65-77. Available from: https://www.sciencedirect.com/science/article/pii/S0921889099000391.

[8] Kitano H, Asada M, Kuniyoshi Y, Noda I, Osawa E. RoboCup: The Robot World Cup Initiative. In: Proceedings of the First International Conference on Autonomous Agents. AGENTS '97. New York, NY, USA: Association for Computing Machinery; 1997. p. 340–347. Available from: https://doi.org/10.1145/267658.267738.

[9] Kschischang FR, Frey BJ, Loeliger HA. Factor graphs and the sum-product algorithm. IEEE Transactions on Information Theory. 2001;47(2):498-519.

[10] Ortiz J, Evans T, Davison AJ. A visual introduction to Gaussian Belief Propagation. CoRR. 2021;abs/2107.02308. Available from: https://arxiv.org/abs/2107.02308.

[11] Solà J, Deray J, Atchuthan D. A micro Lie theory for state estimation in robotics. CoRR. 2018;abs/1812.01537. Available from: http://arxiv.org/abs/1812.01537.

[12] Pelechrinis K, Iliofotou M, Krishnamurthy S. Denial of Service Attacks in Wireless Networks: The Case of Jammers. Communications Surveys & Tutorials, IEEE. 2011 06;13:245-57.

[13] Douceur JR. The sybil attack. In: International workshop on peer-to-peer systems. Springer; 2002. p. 251-60.

[14] Hidayab M, Ali AH, Abas Azmi KB. Wifi signal propagation at 2.4 GHz. In: 2009 Asia Pacific Microwave Conference; 2009. p. 528-31.

[15] Gil S, Kumar S, Mazumder M, Katabi D, Rus D. Guaranteeing spoof-resilient multi-robot networks. Autonomous Robots. 2017;41(6):1383-400.

[16] Kumar S, Gil S, Katabi D, Rus D. Accurate Indoor Localization with Zero Start-up Cost. In: Proceedings of the 20th Annual International Conference on Mobile Computing and Networking. MobiCom '14. New York, NY, USA: Association for Computing Machinery; 2014. p. 483–494. Available from: https://doi.org/10.1145/2639108.2639142.

[17] Cortes J, Martinez S, Karatas T, Bullo F. Coverage control for mobile sensing networks. IEEE Transactions on Robotics and Automation. 2004;20(2):243-55.

[18] Huang Y, Wang W, Wang Y, Jiang T, Zhang Q. Lightweight Sybil-Resilient Multi-Robot Networks by Multipath Manipulation. CoRR. 2019;abs/1912.04613. Available from: http://arxiv.org/abs/1912.04613.

[19] Gupta D, Saia J, Young M. Peace Through Superior Puzzling: An Asymmetric Sybil Defense. In: 2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS); 2019. p. 1083-94.

[20] Thorndike L. Medieval Technology and Social Change. By Lynn White, jr..(New York: Oxford University Press. 1962. Pp. ix, 194, 10 plates. $6.00.). Oxford University Press; 1962.