TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Department of Electrical Power Engineering and Mechatronics

CONSENSUS-BASED DEPTH CONTROL BY USING MULTIPLE SENSORS

MITME SENSORIGA KONSENSUSPÕHINE SÜGAVUSKONTROL

MASTER THESIS

Student: Andrijana Cekovska

Student code: 177287MAHM

Supervisor: Prof. Dr. Mart Tamre, Head of Department

Tallinn 2019

(*On the reverse side of title page*)

AUTHOR'S DECLARATION

Hereby I declare, that I have written this thesis independently.

To the best of my knowledge, no academic degree has been applied for based on this material. All works, major viewpoints and data of the other authors used in this thesis have been referenced.

Author: ....

/signature /

Thesis is in accordance with terms and requirements

....

/place and date/

Supervisor: …........................

/signature/

Accepted for defence

....

/date and place/

Chairman of theses defence commission: ....

/name and signature/

Department of Electrical Power Engineering and Mechatronics

THESIS TASK

Student: Andrijana Cekovska, 177287MAHM

Study program: MAHM 02/13 - Mechatronics

main specialty: automation, robotics, control engineering

Supervisor: Head of Department, Prof. Dr. Mart Tamre, +372 620 3202

Consultant: Prof. Dr. Stjepan Bogdan, Head of Department
UNIZG – Faculty of Electrical Engineering and Computing, Laboratory for Robotics and Intelligent Control Systems (LARICS), Zagreb, Croatia,
+385 1 6129 644, stjepan.bogdan@fer.hr

Thesis topic:

(in English) *Consensus-based Depth Control by Using Multiple Sensors*

Thesis tasks and time schedule:

No	Task description	Dates of Completion
1.	Familiarization with subCULTron Project and consensus theory	23.02 – 28.02.2019
2.	Recreating simulation results from previous work and analysis	01.03 – 15.03.2019
3.	Implementation of trust algorithm on scheduled acoustic communication	16.03 – 06.04.2019
4.	Analysis of simulation results	07.04 – 13.04.2019
5.	Implmentation of consensus algorithm together with trust algorithm on scheduled acoustic communication / Improvement and additional analysis on the simulation results	14.04 – 30.04.2019
6.	Work on aMussel agents and implementation of trust algorithm	01.05 - 14.05.2019

Language: English **Deadline for submission of thesis:** 21.05.2019

Student: Andrijana Cekovska ....

Supervisor: Prof. Dr. Mart Tamre ....

Consultant: Prof. Dr. Stjepan Bogdan ....

CONTENTS

LIST OF FIGURES

LIST OF ACRONYMS AND ABBREVIATIONS

CoCoRo	Collective Cognitive Robots
subCULTron	Submarine Cultures Perform Long-Term Robotic Exploration of Unconventional Environmental Niches
aPad	artificial Pad surface vehicle
aMussel	artificial Mussel agent
aFish	artificial Fish agent
LAN	Local Area Network
GSM	Global System for Mobile Communications
GPS	Global Positioning System
LED	Light Emitting Diode
LiPo battery	lithium-ion polymer battery
USB	Universal Serial Bus
UART	Universal Asynchronous Receiver-Transmitter
SPI	Serial Peripheral Interface
I2C	Inter-Integrated Circuit
CAN	Controller Area Network
PSoC	Programmable System on Chip
IDE	Integrated Development Environment
BLE	Bluetooth Low Energy
GUI	Graphical User Interface

FOREWORD

The work in this master thesis was completed at the Faculty of Electrical Engineering and Computing, at the Laboratory for Robotics and Intelligent Control Systems (LARICS) in Zagreb, Croatia. The work was associated with subCULTron Project and was directed towards implementation of consensus-based trust algorithms on a scheduled acoustic protocol for depth control of underwater multi-agent system. The work on the Project in LARICS was supervised by Prof. Stjepan Bogdan and Asst. Barbara Arbanas.

Here, I would like to express my gratitude to Prof. Stjepan Bogdan for the opportunity to take part in this Project and to work closely with the research staff who have already done significant work in the filed.

I would like to thank Asst. Barbara Arbanas for her assistance and help throughout the process, which contributed to quality of the work. I would also like to thank Asst. Goran Vasiljevic for his assistance with the testing of the system's functionalities.

Finally, I would like to thank my supervisor at Tallinn University of Technology, Prof. Mart Tamre for approving the work on this Project, which resulted with nice collaboration between the two Faculties.

Keywords: subCULTron, multi-agent system, consensus algorithms, trust algorithm, depth control

Andrijana Cekovska

Tallinn, 2019

INTRODUCTION

The underwater environment has always been challenging for exploration and research as great part of the world's underwater habitat remains unexplored [1], but also the environmental changes in the marine ecosystems such as water pollution and loss of biodiversity have been a topic for discussions on how to preserve the underwater ecosystems [2]. A step towards understanding of the impacts that these changes might have on the ecosystems is monitoring the marine environment and gathering data which can be used for further analysis. And such, with the growing trend of underwater robotics, underwater systems for exploration and monitoring of the underwater ecosystems are being developed. Some of these systems are developed to work as an artificial swarm, which include different types of underwater agents that are designed to work in dynamically changing environments. One project of this kind is CoCoRo Project (Collective Cognitive Robots) for creating a swarm of cognitive, autonomous robots with ability of interaction [3]. Another Project which explores the idea of artificial swarm of autonomous robots is the SubCULTron Project, supported by European Union Horizon 2020 research and innovation program [1], [4]. This Project is still ongoing and provides methods for developing an autonomous artificial swarm of agents for monitoring the underwater ecosystem, which makes it highly attractive. As such, the work in this thesis is closely associated with one of the agents in the subCULTron swarm. The subCULTron swarm consists of three different types of underwater agents named as aPad, aMussel and aFish and are able to communicate with each other depending on their purpose in the swarm, to exchange information and to make decisions based on already existing information, implemented algorithms, but also to work in adaptive way, in order to fulfil the end goal of the system: long-term underwater monitoring of the ecosystem in the canals in Venice, Italy [4].

The agents can also be considered in groups, forming multi-agent systems, depending on a specific requirement. For this purpose, consensus and consensus-based algorithms have been explored and implemented for the multi-agent system. The main idea behind these algorithms is to have all the agents exchange information with each other through communication and finally to agree on a mutual value for the parameter of interest. This will enable the agents to cooperate between each other, and the system, therefore, to achieve autonomy in operation.

Notable work has been done in the area of consensus theory and algorithms, such as the work in [6] by Olfati-Saber et al., where different types of consensus algorithms are discussed, and some principles are mathematically proven. One of the main purposes of the consensus algorithms is to ensure convergence of the algorithm, i.e., to ensure that the agents' values will converge to one common value which will be used in the system. Another type of consensus-based algorithms is the trust algorithms, which consider the notion of trust values between the agents in the system. The main goal is to have all the agents in the system reach an agreement on every agent's trustworthiness in the group.

Notable work in the area of trust algorithms is done by Haus et al. [8], where the trust values calculations depend on direct communication between the agents and exchange of information, but also on information obtained through neighbours' observations. The notation of agent's confidence is used and adaptation law for the observation function is given. Moreover, the convergence of the trust algorithm is mathematically proven [8]. Considering the subCULTron swarm itself, newly developed algorithms have been implemented. One example is implementation of average consensus algorithm for scheduled acoustic communication by Arbanas et al. given in [9]. With this communication protocol defined, the problem with interference when all the agents in the group try to communicate at the same time, is eliminated. There are however certain problems which are still to be solved, such as how to detect agent's malfunction by the group, which greatly affects the outcome of the consensus algorithm implemented, and therefore affects the work of the system.

This kind of problem, specifically associated with the depth control of the aMussel agents of the subCULTron swarm, will be addressed and analysed in this work.

The chapters of the thesis, therefore, are organized as follows: introduction to subCULTron Project and the swarm of agents with description of the agents' functionalities; previous work on the system regarding the topic which will be discussed in the work; problem statement and proposed solution; overview and analysis of methods and algorithms which are of importance for the proposed solution; presentation of the simulation results; discussion for the implementation of the proposed solution on the aMussel agents and finally summary of the work done.

1 SUBCULTRON UNDERWATER PROJECT

The SubCULTron Project (Submarine cultures perform long-term robotic exploration of unconventional environmental niches) represents a swarm of autonomous robots, envisioned for a long-term monitoring and exploration of the underwater habitat in Venice, Italy [4]. As this region and its underwater world offer an environmental habitat which is strongly influenced by industry, inhabitants, tourism, the Venice Lagoon is selected as an experimental area [12]. The main goal of the project is to have an underwater system which will be capable of autonomous operation for few days or even weeks, without the need for human intervention [13]. As such, the system consists of three different autonomous agents:

- artificial lily pads on the water surface;
- artificial mussel on the sea-ground;
- artificial fish in between these two layers.

Each of these underwater agents has a special purpose in the system, which contributes to the end goal of the system. The artificial lily pad, referred to as aPad operates on the water surface and serves as a communication hub between the swarm and the scientists who are monitoring and operating the swarm, through short-range and long-range communication modules [13]. This surface vehicle has multiple purposes: it serves as a charging station which is capable of exchanging energy with the aMussel and aFish agents, and for this purpose, the aPads are equipped with solar panels. The aPad also possess docking system, which is used for docking the agents for charging or relocation [13]. For communication purposes, it is equipped with GPS which provides absolute location in the global coordinate system, as well as Wi-Fi modules and antennas which provide communication between the agents on the water surface and nanomodem unit for underwater acoustic communication and localization [14], [15]. The aPad agent is shown in Figure 1.1.

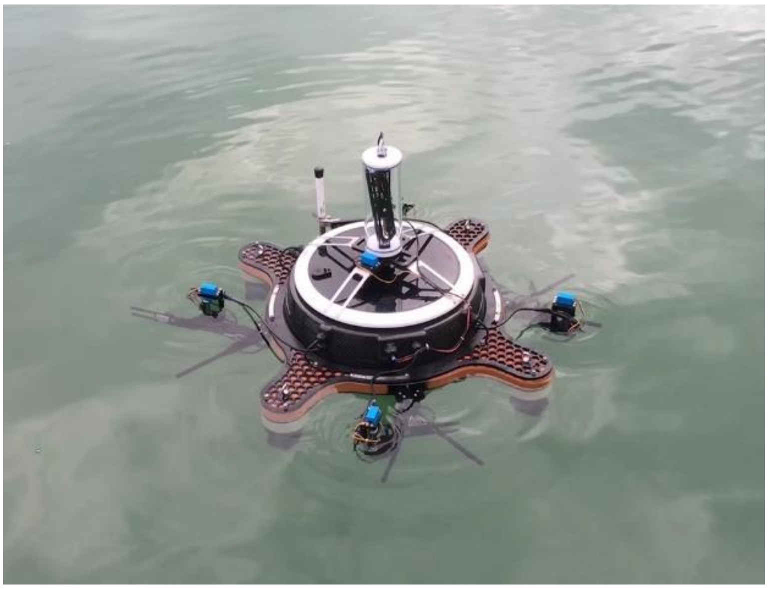

Figure 1.1: aPad agent [11]

The artificial Mussel robot on the other hand, referred to as aMussel, is an underwater agent, which is used to monitor the natural habitat, to gather information about biological agents such as algae, and fish, but also to gather information about certain parameters such as temperature, pressure, turbidity, ambient light [4], [13]. The aMussel agent has one degree of freedom, which allows the agent to move either vertically upwards when heading to the surface, or vertically downwards when diving to the bottom, for which operation it has buoyancy system implemented [13], [14]. The aMussel agent also has gripping legs at the bottom, for anchoring on the seabed, as it is originally intended. Since the aMussel agents' main operation is monitoring the ecosystem, they are equipped with various sensors such as pressure sensor, ambient light sensor, temperature sensor, turbidity sensors, and some of the aMussel agents are also equipped with sensors for measuring dissolved oxygen [15]. The agents are also equipped with different types of modules for communication between each other or with other agents in the swarm, in order to share data. These communication modules, both for underwater communication and communication on the surface, include wireless LAN, Bluetooth, acoustic communication, as well as electric sense and modulated Blue/Green light [9], [13]. As the aMussel agents are of interest in this work, other aMussel functions and capabilities will be provided in Section 1.1 - Section 1.3. The aMussel agents are shown in Figure 1.2.

Figure 1.2: aMussel agents [9]

The third agent in the system is the artificial fish, referred to as aFish, which serves as communication link between the aPad and group of underwater aMussels. The aFish is equipped with sensors, same as the aMussel agents, for measuring temperature, pressure, conductivity. The aFish is also equipped with the same communication modules as the aPads, Wireless LAN, Bluetooth and acoustic communication, as well as electric sense and Blue/Green light communication as the aMussel agents [13], [14], [15]. The aFish agent is shown in Figure 1.3.

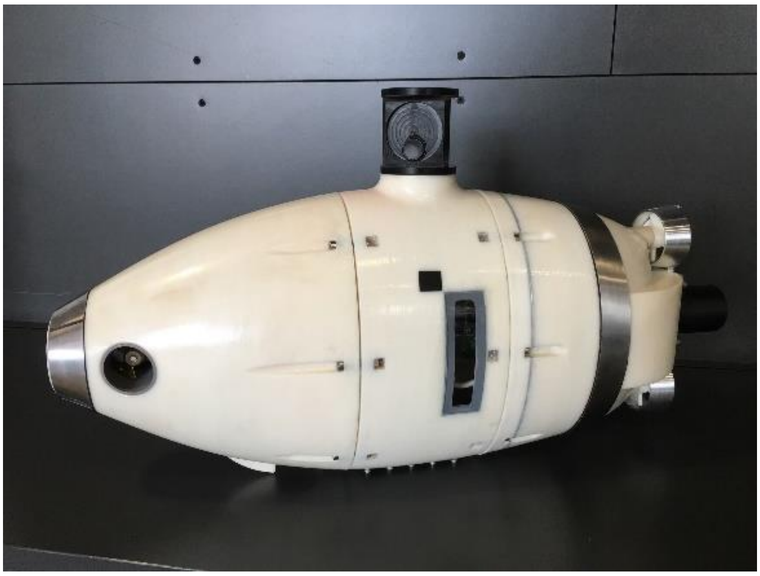

Figure 1.3: aFish agent [9]

1.1 aMussel hardware structure and electronics

For gathering information from the environment and transmitting the data for analysis, the aMussel agent is equipped with variety of sensors and communication modules. The main hardware structure is given in Figure 1.4.

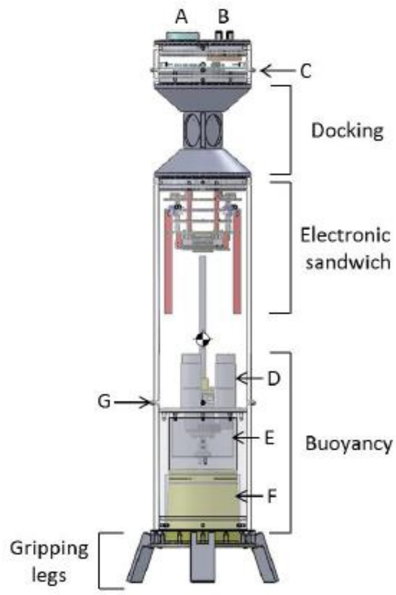

Figure 1.4: aMussel basic structure: A – acoustics, B – ambient light, turbidity sensor, C – electric sense receivers, D – motors, E – piston, F – rolling diaphragm, G – electric sense emitters [14]

Buoyancy system is implemented at the bottom of the aMussel agents. The buoyancy system is a diaphragm-based system, which consists of a piston with a rolling diaphragm and three motors placed under 120 degrees. One of the motors is equipped with incremental encoder, which is used for driving the piston. The rolling diaphragm which is attached to the bottom of the platform, is pushed and pulled by actuating the linear driving mechanism, which causes the volume of the platform to increase or to decrease, and in this way to change the depth of the aMussel agent underwater [14]. When the piston is completely out, the volume is the highest, therefore the force exerted causes the aMussel agent to head to the surface. When the piston is completely in, the volume is the lowest, and it causes the aMussel agent to sink to the bottom. This provides the aMussel agent with one degree of freedom, without a possibility for active movement [13], [14]. Moreover, the aMussel agent has gripping legs under the buoyancy system, which are used when it is needed the aMussel agent to be anchored to the seabed, as shown in Figure 1.4.

Above the buoyancy system, as can be seen in Figure 1.4, there is the “electronic sandwich”, which consists of modulated Blue/Green light and novel type bio-inspired sensing mechanism, called electric sense, as well as Raspberry PI unit, Raspberry PI adapter and Power board [13], [14], [15].

Above the “electronic sandwich”, the docking system can be found. As it was mentioned in Chapter 1, the docking system is used from the aPad platform, to dock the aMussel agent for charging or relocation. At the top of the aMussel, different types of sensors can be found, such as turbidity, ambient light, pressure, and temperature sensors. The aMussel agent is also equipped with inertial measurement unit. Some aMussel units are also equipped with sensors for measuring dissolved oxygen [15]. All of the sensors are used for measuring the corresponding parameters of the underwater ecosystem.

In order to be able to communicate between each other, as well as with the other agents, the aMussel agents are equipped with communication modules given as:

- Wireless LAN, Bluetooth, GSM modules for operation on the water surface;
- Geolocation (GPS) for determining the agents position while on the surface [15];
- Nanomodem unit for underwater acoustic communication between the aMussel agents themselves or for communication with the aPad surface vehicles;
- Short-range communication with electric sense and modulated Blue/Green light.

Additionally, the aMussel agents have LEDs for monitoring the internal state of the agents, as well as camera for taking pictures underwater, while positioned at the seabed [14].

For powering the aMussel, each aMussel agent is equipped with two LiPo batteries (5000 mAh, 3,6 V), with one as principal power source and one as backup source [14].

1.2 aMussel software modules

The aMussel agents are equipped with 32-bit ARM core microprocessor unit based on the Programmable System on Chip from Cypress Semiconductor, with ability for hibernation and with minimal energy consumption [14], [15]. The microprocessor can disable the power supply of sensors and modules, therefore reducing power consumption. As the computational power of the microprocessor is low, additional raspberry PI unit is installed, where the Wi-Fi communication, as well as storing large amount of data is performed via the raspberry PI unit [15]. Furthermore, Wi-Fi dongle is also provided to the raspberry PI board, which enables wireless network for the aMussel agents [15]. The system has different types of interfaces such as USB, UART, SPI, I2C, CAN interface for connecting sensors or external connectivity [14]. The aMussel agents also have Real Time Clock for recording data in real time. The information, therefore, are either send via some communication module or are stored in the internal flash, of 512 Mb [14].

The system architecture using FreeRTOS operating system is developed for the microcontrollers of the aMussel agents [16]. Communication and device drivers for GPS, GSM, Bluetooth, turbidity sensors have been implemented and tested [16]. The architecture of the operating system of subCULTron is given in Figure 1.5.

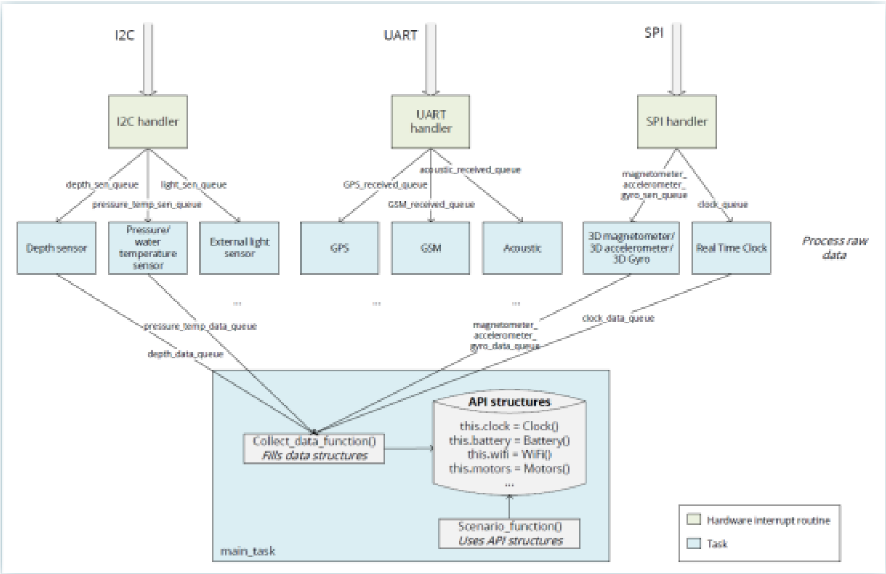

Figure 1.5: Architecture of the operating system of subCULTron [16]

For the software development part of the aMussel agents, as well as the aPad surface vehicles, PSoC Creator IDE is being used. The commands send via Bluetooth to the aMussel agents is performed by using the Docklight tool [18].

1.3 aMussel underwater communication

When discussing methods for underwater communication, there are not many options for reliable communication without loss of data on greater distances [19]. Radio waves, for example, do not propagate well underwater [19]. Optical signals in the blue/green region experience attenuation when propagating on distances beyond 100 m [19]. Therefore, for long-range underwater communication, acoustics communication is widely used. The sound wave propagates as pressure wave, and it is therefore possible to travel longer distances in kilometres. The bandwidth of operation is defined in order of few kHz, where the frequencies in the 100 kHz region can be used for shorter distances [19].

There are various underwater acoustic modems available, which vary in size, cost, range of operation. And such, the acoustic modems can be in diameter of 0,05 m to even 0,5 m [19]. Depending on the frequency of operation, the frequency of acoustic modems can vary from few to hundreds of kHz having into consideration the maximal distance which they need to cover [19]. Searching for more convenient solution for underwater communication, the Intelligent Sensing and Communication research group at the University of Newcastle has developed low-cost acoustic nanomodems [15], [20]. These nanomodems operate in the acoustic frequency range of $24-28\ \mathrm{kHz}$, with acoustic data rate of $40\ \mathrm{bits/s}$, where unicast or broadcast messages of up to 7 bytes can be send. The maximal covered range is around $2\ \mathrm{km}$ [20]. The power supply of the nanomodems is $3-6{,}5\ \mathrm{Vdc}$.

For underwater communication in the subCULTron Project, i.e., communication between the aMussel agents themselves, and the other agents as well, few methods are being used:

- Long-range acoustic communication - all the aMussel agents are equipped with low-cost nanomodems, as described in this Section 1.3;

- Short-range communication: modulated blue/green light and electric sense.

The electric sense communication as short-range communication can be used in muddy or turbid water [1], [14]. As it is not very expensive, and it can easily be implemented on an underwater agent, it can be used for obstacle avoidance, object recognition, or docking [1]. The docking refers to the possibility for the underwater agent to attach to the station for charging or data exchange, in this case to the aPad, which is used as a recharge station. The electric sense can be used for detecting the electric line reflected by polarized object, which will cause the agent to avoid the obstacles [1]. When the electric sense is fused with inertial measurement, for example, it can also be used for object recognition [1]. The electric sense is complemented by modulated blue/green light [1], [14], [15]. As an active sense, the modulated light works on the principle of emitting light, which is reflected by the obstacles and detected by photoreceptors. The modulated light sense is not much affected by turbid water, but it can be severely affected by external light source, when close to the surface [1], [14]. The range of sensing is around $1\ \mathrm{m}$. However, the modulated light sense has small cone of perception, therefore many devices are needed for avoiding blind spots [1], [14].

For this work, only the acoustic communication will be considered, as long-range communication is needed for communication between the aMussel agents, which are of interest.

As it was described in this Section 1.3, the aMussel agents are equipped with nanomodem unit, shown in Figure 1.6.

Figure 1.6: Nanomodem for acoustic communication [15]

The underwater acoustic communication though can have its drawbacks, as discussed in [15], such as delay of packet transmission and loss of data, due to the slow propagation of the sound in water, multipath, and attenuation [15]. Another problem which can occur when utilizing underwater acoustic communication is the interference when two or more agents try to communicate at the same time [15]. Due to this, collision can happen, and the will be lost.

Having these problems into consideration, scheduled acoustic communication protocol, used together with average consensus algorithm, has been developed for the purpose of this Project [9]. In the following chapters, the implemented scheduled acoustic protocol will be discussed in detail.

2 PREVIOUS WORK ON THE SYSTEM AND ACHIEVEMENTS

As the aMussel agents are equipped with various types of sensors, such as pressure and ambient light sensors, the primary purpose of the aMussel agents is envisioned to be monitoring of the underwater environment, while being positioned at the seabed [4], [13], [14]. However, these parameters which are being monitored by the aMussel agents can vary on different depth [13], [14]. Therefore, depth control algorithm is being developed, which allows the aMussel agents to be positioned at certain depth in the canals and monitor the environment for the parameters of interest. Each of these agents can run the depth control algorithm with a reference input from its own implemented pressure sensor. Since the work in the thesis does not assess the depth control algorithm itself, but rather considers the depth control of the agents from different perspective and addresses a specific issue regarding the reference value which is given as an input to the depth control algorithm, analysis regarding the stability of the system while performing the depth control algorithm with a given reference value, the time needed for reaching a steady state during execution of the depth control algorithm, are out of the scope in this work and will not be examined further.

Considering the aMussel agents as part of a swarm, i.e., multi-agent system, the agents, therefore, should be able to communicate with each other and exchange information [13]. Furthermore, as the multi-agent system will operate underwater, long-range underwater communication should also be established. Therefore, each aMussel agent is equipped with nanomodem unit, which enables long-range underwater acoustic communication. Because of the interference which can occur when all the agents try to communicate inside the swarm at the same time, in order to exchange information and consequently cause loss of data, scheduled communication protocol has been developed and established for the system, with round-robin scheduling mode [9], [15]. Moreover, as the aMussel agents are considered as part of multi-agent system with possibility to communicate between each other, the consensus theory, i.e., consensus algorithms are being utilized. The main purpose of consensus and consensus-based algorithms is to have all the agents, part of the multi-agent system, reach mutual agreement on the parameter of interest. For the analysed system, with sequenced communication protocol defined, average consensus algorithm has also been implemented and tested [9]. The authors in [9] demonstrate that the average consensus algorithm used with the sequenced communication protocol achieves convergence, with set of analysis conducted, having into consideration that the convergence of the average consensus algorithm has already been proven. Mathematical proof for the convergence of this solution however is still to be provided [9].

This solution which includes the average consensus algorithm, ensures convergence of the agents' values, even though the measurements received from the agents' sensors might be incorrect. Therefore, the agents will be able to reach an agreement on mutual value, however this value might differ greatly from the formally considered correct value [1], as the incorrect measurement will not be excluded in the calculations.

As the defined algorithm is crucial for addressing the problem and proposing the solution, overview of the implemented average consensus algorithm on the sequenced communication protocol will be given in Section 2.1. Moreover, to better illustrate the outcome of the implemented solution, simulation results[2] will also be presented in Section 2.1.

2.1 Average consensus algorithm for scheduled acoustic communication

Arbanas et al. in [9] propose new communication protocol for underwater communication which relies on scheduled transmission of the agents in multi-agent system. The defined protocol is designed in a way where all agents transmit their messages in their designated time slot. The period of transmission is defined and represented with T, and the duration of the actual transmission is given with T_d. To avoid collisions, T_d should always be smaller than T [9].

The transmission of the i-th message is given with, [9]:

$$t_i = t_1 + (i-1)T \tag{2.1}$$

where t_1 is the transmission of the first message, under assumption. Therefore, the start and duration of each time slot is given with (t_i, T). The time scheduling of the slots is decided to be round-robin.

[1] value which most of the agents in the multi-agent system have or hold similar values

[2] The presented simulation results are based on the original work by the authors presented in [9]. The tested scenarios and the results obtained are recreated for the purpose of this work, with consent from the authors, which are also supervising this work

The proposed algorithm is given as follows, [9]:

$$x(k+1) = P'x(k) \tag{2.2}$$

where P' is the Perron matrix, defined as:

$$P' = I - \varepsilon L' \tag{2.3}$$

where I represents the identity matrix,

L' is the graph Laplacian of the communication graph $G = \{V,\ E\}$, with V representing the vertices and E representing the edges in the graph of the physical topology.

The graph Laplacian is given with:

$$L' = D' - A' \tag{2.4}$$

where D' and A' are defined as follows, [9]:

$$D' = diag(Am) \tag{2.5}$$

$$A' = A\ diag(m) \tag{2.6}$$

and they represent the modified Laplacian matrix in order to switch the state of the system, originally given as:

$$D = diag[d_1, d_2, .. d_n] \tag{2.7}$$

where D is the degree matrix of the graph, with elements $d_i = \sum_{i \neq j} a_{ij}$ and zero off-diagonal elements;

$A = [a_{ij}]$ is the adjacency matrix of the graph.

The modified Laplacian matrix in the proposed algorithm uses the elements of the vector $m = (m_j)$ defined as, [9]:

$$m_j(t) = \begin{cases} 1, & t = t_1 + (i-1)T \\ 0, & otherwise \end{cases} \quad (2.8)$$

where $j = i \, mod \, n$,

i is the index of the time slot.

Moreover, an average consensus algorithm has also been implemented and adjusted to the communication protocol defined [9]. The consensus algorithm has been validated experimentally, having in consideration the fact that the convergence of the average consensus algorithm has already been proven [9]. Different scenarios were tested and presented in this work, where the influence of the step size ε is being analysed to the convergence speed and scalability of the method [9]. The discussed average consensus algorithm is given as:

$$x_i(k+1) = x_i(k) + \varepsilon \sum_{j \epsilon N_i} \left(x_j(k) - x_i(k) \right) \quad (2.9)$$

where $x_i(k)$ is previous state of agent i,

$x_j(k)$ is previous state of agent j,

$x_i(k+1)$ is current state of agent i,

ε is the step size,

N_i represents the neighbours of agents i ($j \epsilon N_i$ means agent j is neighbor of agent i).

The outcome of the tested solution under certain scenarios is convergence of the algorithm with the specified values/measurements given as inputs. However, in case of faulty sensor measurement, the consensus value will be calculated considering the incorrect measurement as well. For more information on the tested scenarios and the results, the reader is directed to [9].

To better illustrate the problem that is being addressed in this work, the convergence of the implemented algorithm will be shown for two cases: first when all the agents have similar values, and second when agent 4 have significantly higher value than the other agents in the group.

The convergence of the measurement values is demonstrated in Figure 2.1, for the case when all the agents in the group are assigned similar values. The measurement vector is chosen to be:

$$x = [23\ 30\ 32\ 35\ 25]$$

and the adjacency matrix for a selected topology of the system is also selected as:

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

These values will also be used for the Simulation phase of the proposed solution, to illustrate the improvement of this algorithm in same conditions.

The simulation is performed for step size taken as $\varepsilon = 0{,}2$.

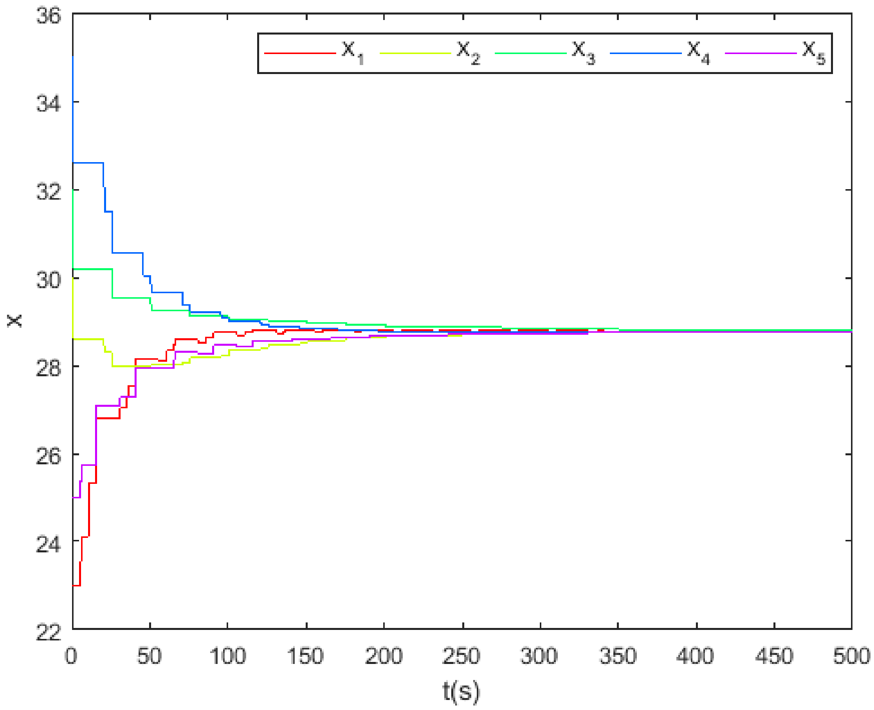

Figure 2.1: Convergence of the measurement values, with $\varepsilon = 0{,}2$ and all agents in the group having similar values (the result is obtained based on the work in [9])

It can be seen in Figure 2.1 that in case when all the agents in the swarm have similar values, the implemented solution leads to convergence towards a value which corresponds to the measurement values of the group.

In the following case, the outcome of this solution when agent 4 is assigned significantly higher value than the other agents in the group will be demonstrated. The adjacency matrix and the step size remain the same, whereas the measurement vector is now given as:

$$x = [23\ 30\ 32\ 120\ 25]$$

The result is shown in Figure 2.2.

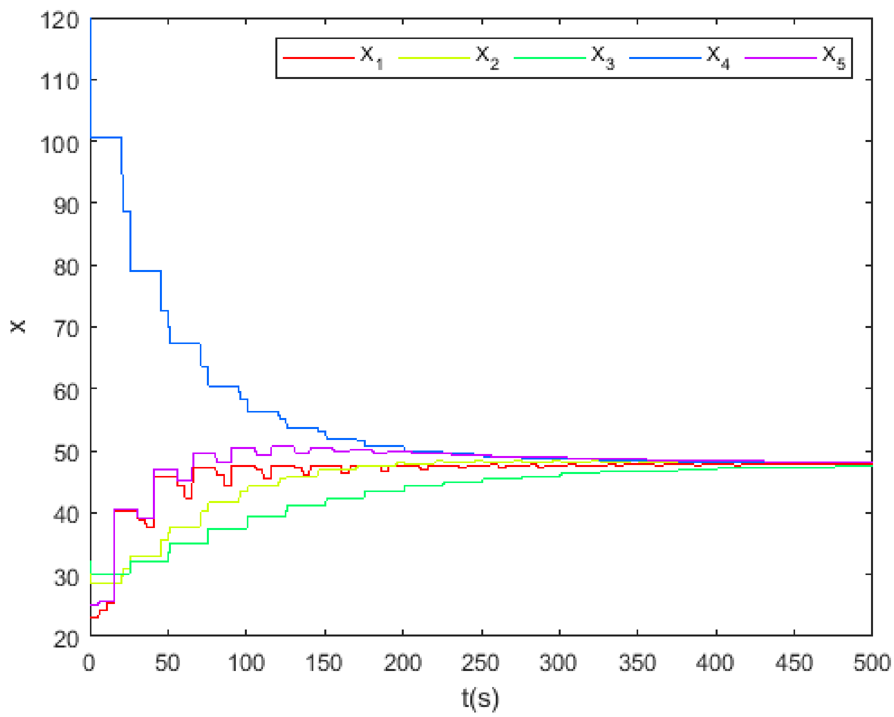

Figure 2.2: Convergence of the measurement values, with $\varepsilon = 0{,}2$ and agent 4 having significantly higher measurement value than the other agents in the group (the result is obtained based on the work in [9])

It can be seen from Figure 2.2 that the implemented algorithm achieves convergence on the measurement values, however the mutually agreed value is lot higher than the values which most of the agents in the group have. The agent 4 high measurement value was not detected by the group.

This leads to pose a question on how it is possible to detect sensor malfunction, when using the consensus algorithm. Now we are ready to state the problem and to propose a solution.

3 PROBLEM STATEMENT AND PROPOSED SOLUTION

To meet the main requirement of the system where each aMussel agent in the swarm hold prespecified depth, the main question which arises is how the aMussel agents can detect sensor malfunction, and moreover will it be possible for the aMussel agents to maintain their depth, despite of the sensor failure. The main reasoning behind the idea is not to have the depth control algorithm dependent on only one sensor measurement, i.e. the measurement from the agent's sensor which is implemented on the agent running the depth control algorithm. This reasoning leads to two possible scenarios:

1. As the agents are part of the swarm and the communication protocol between them has already been established, consensus-based algorithms can be implemented in a way that will allow exchange of information between the agents, therefore the reference signal to be used as an input for the depth control algorithm will be the mutually agreed value between the agents. This kind of approach will eliminate the individual influence of the agents' sensors measurements and it will take into consideration sensor measurement from multiple sensors;

2. As the agents are equipped with multiple different sensors, such as pressure sensor and ambient light sensor, the other possible solution is to have the agents switch the reference value for the depth control algorithm in case of sensor failure. Sensor failure/malfunction will be detected based on the given range of allowed sensor measurements in order to hold the required depth. In this case, the agent will either head to the surface or dive to the bottom, which will affect the sensor measurement and it will indicate sensor failure. When this occurs, the agent should then switch to the other reference signal provided to the depth control algorithm. Since the agent will no longer hold the required depth at that point, the ambient light sensor measurement might also have incorrect value, which value alone cannot be used as an input to the depth control algorithm. Therefore, a consensus-based algorithm should be implemented for the ambient light measurements of all the agents in the swarm, so the group is able to reach mutual agreement on this measurement value, which will not depend on only one sensor measurement. Later this mutually agreed value will be passed as a reference value to the depth control algorithm.

As it was explained in Section 2.1, having average consensus algorithm implemented in the multi-agent system, will only ensure convergence to a mutually agreed value of the agents, based on the actual measurements which are being exchanged via acoustic communication, even in the case when the sensor measurement might be incorrect.

The proposed solution for this problem is to use consensus-based trust algorithm together with the consensus algorithm, where the trust algorithm will work towards reaching a consensus on the trust values that each agent holds towards all the other agents in the group and these trust values will be used for determining the final consensus measurement value of the agents' measurements by using the consensus algorithm. The use of the trust algorithm will result with the agents being able to adapt the trust values which they hold towards their neighbouring agents and at certain point all the trust values towards one agent in the group will converge to a mutually agreed value between the agents. In case of lower trust values towards a particular agent, the measurement from the said agent will be handled properly by the group. This will ensure that the agents reach an agreement on the measurement value, but also it will ensure that the agreed value is correct, in terms that the agreed measurement value by the group will be closer to the values that most of the agents in the group have or hold similar values. Afterwards, all the agents will have the mutually agreed value by the swarm, including the agent which expressed sensor malfunction before. This will cause the said agent to regain and maintain the required depth.

As it was discussed in Chapter 2, due to the possibility of loss of data as interference can occur when the agents communicate at the same time using the underwater acoustic communication, the sequenced communication protocol is implemented. The consensus-based trust algorithm which will be implemented, will also work on the already established communication protocol in the multi-agent system.

Another major point in the proposed solution, which comes directly from the trust algorithm, is that the whole system is decentralized. Therefore, there is no central master agent which holds trust values for all agents in the swarm. Instead, all the agents should be able to hold trust values towards all the other agents in the group and moreover to be able to adapt the corresponding trust values when the agent with the turn to communicate, shares its values with the group. In meantime, all the agents should perform the consensus-based trust algorithm, as well as the consensus algorithm locally, and reach a consensus on the measurement value, in accordance with the obtained trust values towards the agents in the swarm.

In the following chapters detailed explanations of all the methods used in the proposed solution will be discussed and the results will be presented.

4 CONSENSUS AND CONSENSUS-BASED TRUST ALGORITHMS

Consensus and consensus-based algorithms have been widely explored when considering multi-agent/multi-vehicle systems for various applications, such as distributed network systems, formation and cooperative control [5], [6]. Great part of the work in this field focuses on the importance of reaching an agreement to a certain value between the agents in the system, following certain criteria, that the group should achieve. In [6], Olfati-Saber et al. provide analysis on the theoretical aspects of consensus algorithms in networked multi-agent systems. The authors also argue the importance of connectivity of the nodes, i.e. topology (fixed or switching), as well as the information flow which can be directed or undirected [6]. In the area of formation control, in [31], Babić et al. discuss the advantages that a group of vehicles would have over a single vehicle containing more expensive equipment, such as distributed operation and coverage of larger areas, higher computational power and redundancy.

As the decision making in multi-agent systems follow certain criteria which is defined in advance but may vary in time, based on changes in the environment or change of the end goal, lot of these systems are designed to operate in decentralized fashion [7]. As it is discussed in [7], [8], for two agents to complete a common mission, the notion of trust is being introduced. This trust value among the agents represent the trustworthiness of the agents in the group, under two assumptions:

- the agents are devoted in completing the common mission, and
- the agents are in fact capable of completing the end goal [7].

The trust-based approach in dynamic multi-agent systems has been assessed in some previous work, with the following adaptation law for the trust vector proposed, given in [7], [8]:

$$\dot{\xi}_i = \sum_{j \epsilon N_i} \xi_{ij}\left(\xi_j - \xi_i\right) \tag{4.1}$$

where ξ_i is the trust value of the i-th agent,

ξ_{ij} is the trust value of agent i towards agent j,

ξ_j is the trust value of agent j,

N_i represents the neighbours of agent i.

As the nature of the trust algorithms is consensus-based, all the agents in the group should be able to reach a consensus on the trust values towards other agents in the group [7], [8]. This will require convergence of the trust algorithm.

In [7], [8] the authors continue to argue about the use of observations from the agents when adapting the trust values, as well as its convergence, and propose an algorithm with the use of trust values as well as observation functions. Moreover, they provide mathematical proof of its convergence [8].

Since the trust algorithm will be used as part of the solution in this work, the trust algorithm will now be discussed in more details.

4.1 Graph theory considerations for consensus-based trust algorithm

In order to give an overview of how the multi-agent system operates, and later to discuss the trust algorithm, basic notation from the graph theory will be given. The agents in the multi-agent system can be represented with a control graph, whereas the agents represent the nodes of the graph and the edges' weights between the nodes in the graph are given with the trust values of the corresponding agents in the swarm.

Therefore, the graph is described as $G = \{V, E\}$, where G represents the graph with

$V = [1,\ 2, ..n]$ representing the nodes of the graph, i.e. the agents' states, and

$E = \{e_{ij}\}$, $i, j \in V$ representing the edges between the nodes of the graph [8].

From graph theory it is known that every graph can be either directed or undirected. If there is two-way communication between any two nodes of the graph, then the graph is undirected. On the other hand, if there is connection, for example, from agent i towards agent j, but the opposite is not necessarily the case, then the graph is directed [28], [29].

As the system is decentralized, and so is the control graph, Haus et al. in their work in [8] choose a consensus-based algorithm which determines the state of the agents, where the states of the agents are represented with the parameter of interest.

The algorithm is given as, [8]:

$$\dot{x}_i = \sum_{j \epsilon N_i} \xi_{ij}(x_j - x_i) \qquad (4.2)$$

where ξ_{ij} is the trust of agent i towards agent j and $0 \leq \xi_{ij} \leq 1$,

x_i is the measurement state of agent i,

x_j is the measurement state of agent j,

N_i represents the neighbours of agent i.

Given these preliminaries on how the multi-agent system is represented, in Section 4.2, the consensus-based trust algorithm will be elaborated.

4.2 Consensus-based trust algorithm in discrete time domain

For reaching an agreement on the agents' states, the authors in [8] use the algorithm given with (4.2).

Referring to previous work in the field, in [8] it is stated that the convergence of this algorithm is already proven. For the purpose of this work, it is enough to know that the convergence of the algorithm is ensured and mathematically proven. Therefore, the algorithm will be used with the guarantee of convergence and so the formal proof of the convergence will not be discussed in this work. For further details on this matter, the reader is directed to the work presented in [6], [8].

The main question which is of interest here, is how to calculate and update the trust values, which represent the weight values of the edges between the nodes in the graph topology as discussed in [8].

Since the system should work in decentralized manner and all the agents should hold values towards every other agent in the system, for the adaptation law both trust values as well as observation values are being considered.

The adaptation law has been analysed for both continuous and discrete time domain, however for the purpose of this work, only the algorithm defined for discrete time domain will be given. The trust protocol therefore is defined as follows, [8]:

$$\xi_{ij}(k) = \begin{cases} \xi_{ij}(k-1) + \dfrac{1}{d_{ii}+2} \displaystyle\sum_{k \epsilon N_i} \xi_{ik}(\xi_{kj} - \xi_{ij}) + (\tau_{ij}(\sigma_{ij}, \delta_{ij}) - \xi_{ij}) & , j \epsilon N_i \\ \xi_{ij}(k-1) + \dfrac{1}{d_{ii}+1} \displaystyle\sum_{k \epsilon N_i} \xi_{ik}(\xi_{kj} - \xi_{ij}) & , j \notin N_i \end{cases} \quad (4.3)$$

where ξ_{ij} is the trust value of agent i towards agent j, and agent j can be neighbour of agent i, but not necessarily. Both cases are considered;

ξ_{ik} is the trust value of agent i towards agent k, and agent k is neighbour of agent i;

ξ_{kj} is the trust value of agent k towards agent j, where agent k is neighbour of agent i and agent j is/is not neighbour of agent i;

N_i represents all the neighbours of agent i;

τ_{ij} is the observation function of agent i towards the neighbour's j trustworthiness and it depends on σ_{ij} and δ_{ij} parameters;

σ_{ij} parameter is defined as agent i confidence regarding its neighbour j;

δ_{ij} parameter is defined as neighbouring agent j performance as seen by agent i and it can have different interpretation in different applications [8]. For the analysis presented in this paper, δ_{ij} is held constant;

$\xi_{ij}(k-1)$ is the previous trust value of agent i towards agent j;

$\xi_{ij}(k)$ is the current trust value of agent i towards agent j.

The parameter d_{ii} is defined as the weighted-out degree of agent i and is given with the formula (4.4):

$$d_{ii} = \sum_{k \epsilon N_i} \xi_{ik} \tag{4.4}$$

where ξ_{ik} represents the trust value of agent i towards agent k, and agent k is neighbour of agent i.

Regarding the observation function τ_{ij}, two assumptions are made [8]:

1. The confidence value of agent's trustworthiness cannot have negative values;

2. The observation function has the following properties:

   - $0 \leq \tau_{ij} \leq 1$

   - $\frac{\partial \tau}{\partial \sigma} > 0$ for $\sigma > 0$

The following adaptation law for the trust values is being proposed, [8]:

$$\sigma_{ij}(k) = \sigma_{ij}(k-1) - \frac{K}{1 + K\frac{\partial \tau}{\partial \sigma}} \cdot \left(\tau_{ij}\left(\sigma_{ij}, \delta_{ij}\right) - \xi_{ij}\right), \quad K > 0 \tag{4.5}$$

where K is the adaptation gain and the other parameters are same as defined in (4.3).

The confidence value is adapted with respect to agents' observations. The observation function which is defined as observation function regarding the neighbour's trustworthiness is defined in [7], [8] as:

$$\tau_{ij} = e^{-\frac{\delta_{ij}{}^2}{\sigma_{ij}{}^2}} \tag{4.6}$$

where the parameters are same as defined in (4.3).

Referring to previous work in the field, the authors also give correlation of the confidence value where this value is named as reputation. The δparameter is also described to have different names assigned in different works, such as delay or throughput of a channel, or agent's gossip level [8].

Furthermore, different scenarios are being considered in the paper and discussed, where the results of the convergence of the proposed algorithm towards the trust values is shown. It is also demonstrated that these values depend on the adaptation gain, as well as the initial conditions of the parameters considered. Moreover, mathematical proof for convergence of the algorithm, using both trust and observation values is provided.

5 IMPLEMENTATION OF CONSENSUS-BASED TRUST ALGORITHM FOR SCHEDULED ACOUSTIC COMMUNICATION PROTOCOL

The proposed solution is continuation on the work which has been done on the system so far. The proposed solution refers to the problem of detecting sensor failure or sensor malfunctioning which measures the parameter which is needed as a reference signal to the depth-control algorithm, in the multi-agent system. The proposed solution consists of implementing the consensus-based trust algorithm, on the already established scheduled underwater acoustic communication protocol, considering already elaborated and discussed algorithms and methods.

There are four major points to be considered for the proposed solution:

i. The multi-agent system works in decentralized manner, therefore, all the agents in the system should always hold the trust values of the other agents at all times. This follows from the algorithm presented in Section 4.2;

ii. Since the communication is scheduled, the agents should be able to update the corresponding trust values towards the other agents, based on the information received from the agent which is currently transmitting following the protocol;

iii. After each update of the trust values, the agents should calculate the trust values towards the other agents using the trust algorithm, based on the newly received measurements;

iv. With the calculation of the trust values of the agents, the agents should be able to reach mutual agreement on the parameter, i.e. the sensor measurement of interest as final goal, by using the calculated trust values between the agents. The agreed value can be further used as an input for the depth control algorithm. For this purpose, consensus algorithm for the measurement value is performed in parallel with the trust algorithm for calculation of the trust values between the agents, which are then used to calculate the measurement value between the agents in the group.

There are multiple options to be explored when implementing the trust algorithm on the scheduled acoustic communication protocol. As it is already discussed in [8], different parameters influence the calculation of the trust values.

The topology of the system, the initial trust and confidence values, the initial performance of one agent as seen by another agent, the adaptation gain as well, will result with the final trust values agreed by applying consensus-based algorithms to differ. Additionally, the use of scheduled communication in the multi-agent system might affect the calculation of the trust values as well. According to the algorithm given with [8], each agent receives trust values from its neighbouring agents regarding the observed agent, through direct communication. Therefore, for one particular agent to be able to calculate the trust values towards the other agents, the neighbouring agent should have sent its trust values which it holds for the group. If this approach is taken, none of the agents in the multi-agent system will be able to calculate the trust values, until all of its neighbours take turn in communicating and share their information with the group. An alternative to this issue would be to have initial values set to all the agents, and to adapt these initial values as the agents take turn in communicating, following the scheduled communication protocol.

As a multi-agent system is being considered here, the agents should be provided with initial information about the swarm. This will mean that the agents will have information about the topology of the group, but also information about the initial trust values of the agents in the group. Once the initial set up of the system is established, the agents will be able to start communicating within the group and exchange data. Therefore, for performing the analysis, second approach will be adopted.

Another point to be considered is how these initially set values will be adapted through communication. The trust algorithm provides an adaptation law for the confidence values of the agents, therefore the observation function is calculated and included in the trust values calculation. The observation function however, depends on the performance of the agents seen by other agents, parameter δ, despite the dependence on the adaptation of confidence values. The performance of the agents, here presented as δ, can be defined differently based on the application. This value in the cited work [8] is being held constant throughout the algorithm, once initially set. As the trust values are being calculated for the agents, the trust values are used as edge weight between the agents and are used for calculating the measurement value, with the use of (4.2). This formula is used to calculate the consensus value of the agents' measurements in the group, as discussed in [8]. The δ parameter can be adapted based on the consensus values calculated by the agents, and this parameter will change in every step, which will cause the adaptation law provided in [8] by Haus et. al to depend on two parameters. Since in the cited work, the simulation results are obtained by adapting the confidence values of the agents only, and δ values are held constant, in this work, the δ values will also be used as constant.

The initial δ values therefore, can be either assigned for all agents before the start of operation, or the algorithm can be designed in a way that all of the agents in the group will take turn in communication at least once and share their measurement values, and based on these information the initial values of δ parameter will be determined.

For implementation of the algorithm on the scheduled acoustic communication protocol, the formulas given with (4.2) – (4.6) are being used.

As the formal proof of the trust algorithm convergence is given in [8] and the convergence of the consensus algorithm is also proven in some previous work, cited in [8], for the purpose of this work, the trust algorithm, as well as the consensus algorithm will be used without having to provide the proof of convergence. Having provided the discussion on the topic, the results of implementation of the trust algorithm, using the scheduled acoustic communication protocol will be presented.

6 SIMULATION RESULTS

For analysis of the system's behaviour, the initial parameters values will be defined for the agents in the group and multiple possible scenarios will be tested.

The simulation results which will be presented in this chapter will also be performed on pre-defined topology of the system. The topology of the system can vary depending on the application, however for our simulation analysis the topology given in Figure 6.1 was chosen.

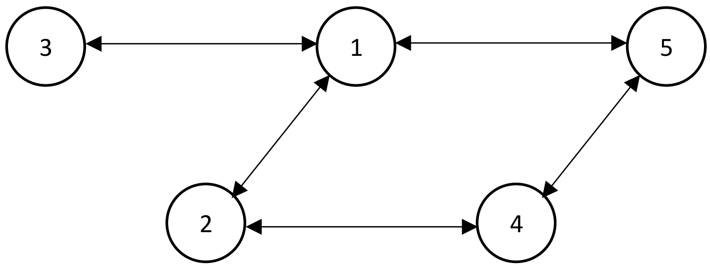

Figure 6.1: Topology of the multi-agent system

The workflow of the proposed solution is defined as follows:

i. Sequenced communication protocol, which determines the sequence of communication for the agents on the channel. The transmission on the channel is broadcast, therefore all of the agents in the group are able to receive the message of the agent who is currently transmitting;

ii. Trust algorithm execution, which will calculate the agents' trust values towards the other agents in the group, for all agents separately and will update the trust values towards the agent who is currently sending on the channel and afterward will continue executing the trust algorithm again;

iii. Execution of consensus algorithm in parallel with the trust algorithm, while using the trust values calculations towards each agent in the group obtained from the trust algorithm, to calculate the final measurement value for the group.

In the proposed solution, there are two consensus-based calculations which are performed in parallel, therefore, the simulation analysis will be presented in two parts:

1. First part will be dedicated to analyses of how the trust algorithm depends on different parameters, as well as whether the trust algorithm reaches convergence in time and how fast this convergence is achieved;

2. Second part will be dedicated to analyses of the outcome of the consensus algorithm and whether it achieves convergence on the measurement value, by using the calculated trust values, when only one agent in the swarm transmits at the dedicated time slot.

For the simulations performed, initial conditions were set for the system and they are defined as follows:

- Adjacency matrix A which is formed based on the pre-defined topology of the system:

$$\mathrm{A} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

- Initial trust values of the agents, where the agents at the beginning consider themselves as the most trustworthy:

$$\xi = \begin{bmatrix} 1 & 0.1 & 0.1 & 0.1 & 0 \\ 0.1 & 1 & 0 & 0 & 0.1 \\ 0.1 & 0 & 1 & 0 & 0 \\ 0.1 & 0 & 0 & 1 & 0.1 \\ 0 & 0.1 & 0 & 0.1 & 1 \end{bmatrix}$$

- Initial confidence values:

$$\sigma = \begin{bmatrix} 0 & 0.8 & 0.5 & 0.6 & 0 \\ 0.7 & 0 & 0 & 0 & 0.4 \\ 0.6 & 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 & 0.7 \\ 0 & 0.7 & 0 & 0.8 & 0 \end{bmatrix}$$

- The initial performance of agent j as seen by agent i given with δ and determined as the error in the measurements between two neighbouring agents, in the range of [0-100], where the initial measurement vector is assigned as:

$$x = [23\ 30\ 32\ 35\ 25]$$

Other parameters which will be considered for the simulations are:

- Adaptation gain initially set to $K = 0{,}02$;
- Communication period for the sequenced communication $T = 5$ s.

For the tested scenarios, it is considered that every agent in the group has transmitted on the channel once and the δ values are calculated and assigned. It is also considered that there is two way communication between the neighbouring agents, therefore, $A_{ij} = A_{ji}$.

6.1 Simulation results of trust convergence on scheduled communication protocol

1. In the first scenario, the convergence of the trust values towards each agent will be analysed using the initial values which are already set. At moment 500 s, the measurement given by agent 4 will significantly change its value compared to the measurement values of the other agent, and the newly obtained measurement vector is

$$x = [23\ 30\ 32\ 120\ 25]$$

The δ value towards agent 4 will also significantly increase. The results are shown in Figure 6.2 and Figure 6.3.

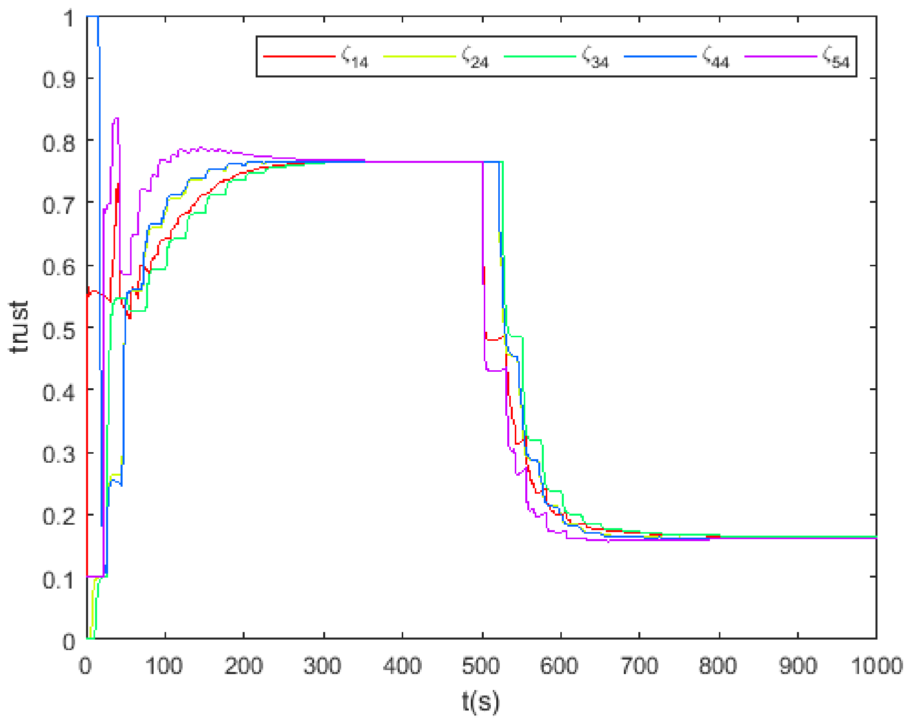

Figure 6.2: Trust values towards agent 4

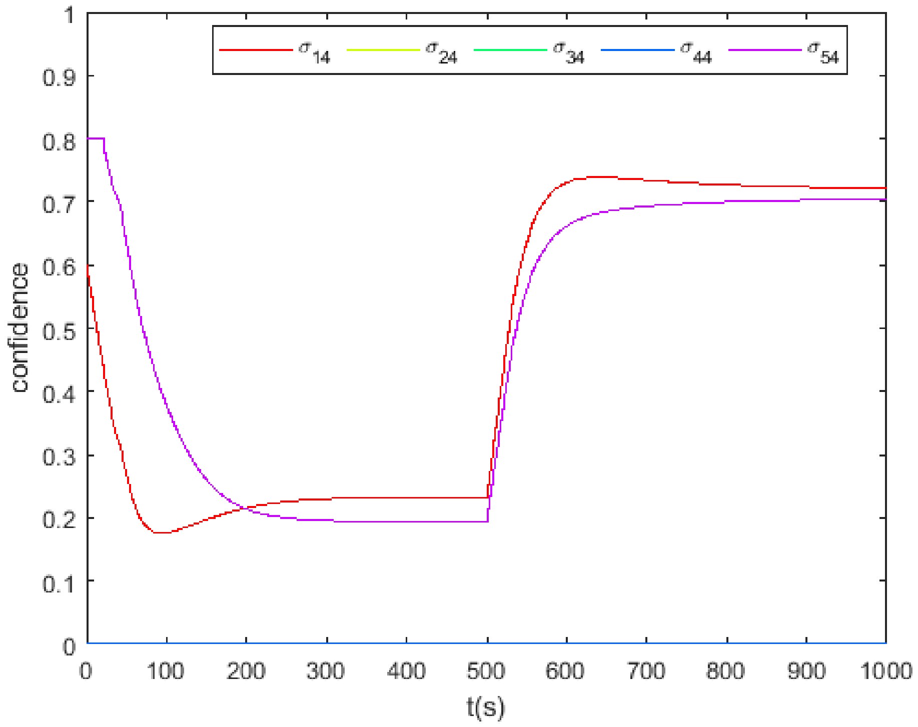

Figure 6.3: Confidence values of the agents regarding agent 4

As can be seen from Figure 6.2, all trust values from the agents in the group towards agent 4 are converging. In the time interval $[0 - 500]$ s the trust values towards agent 4 converge to high value, which depend on the initially set parameters and the measurement values assigned to the agents. At moment 500 s, the value of agent 4 is significantly increased, and this results with the trust values of the other agents in the group towards agent 4 to significantly decrease. Again, they converge to a mutual trust value. This indicates that agent 4 is no longer trustworthy in the group. The confidence values of agent 1 and agent 5 regarding agent 4 in the time interval of $[0 - 500]$ s is lower, as agent 4 is trustworthy and this is recognized by its neighbours, which now adapt their confidence values with respect to agent 4. The opposite happens starting from time $t = 500$ s. This behaviour can be seen in Figure 6.3.

2. In the second scenario, the influence of the adaptation gain on the outcome of the algorithm is analysed. Here it is explored how fast the convergence of the trust values are reached when setting the adaptation gain parameter to 10 times higher value than the original value of $K = 0{,}02$. The initial set up for the group is same as in Section 6.1 scenario 1. The results are given in Figure 6.4 and Figure 6.5.

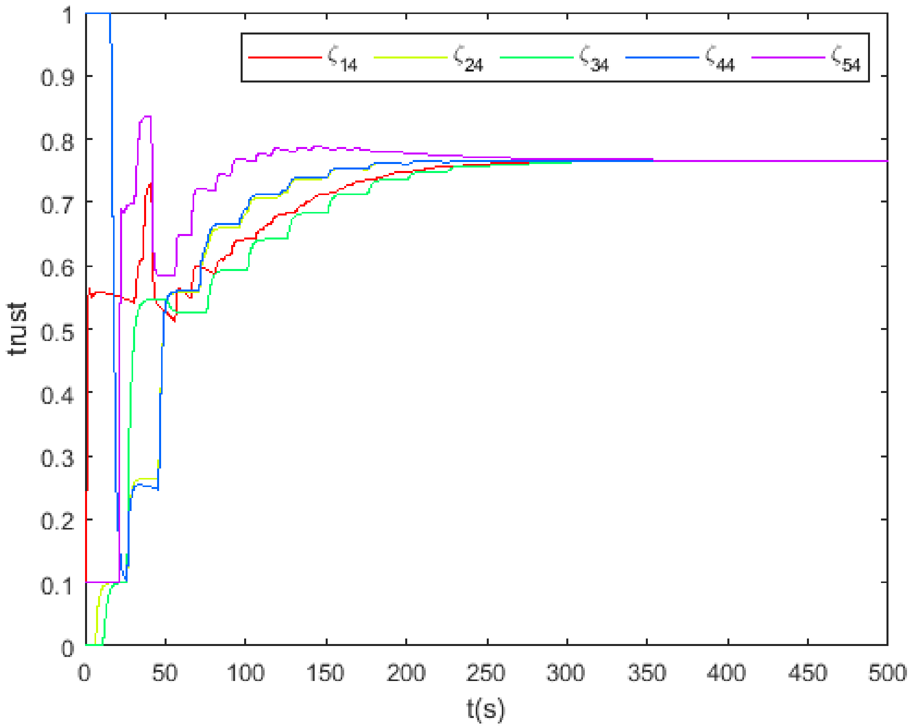

Figure 6.4: Trust values towards agent 4, with adaptation gain $K = 0{,}02$

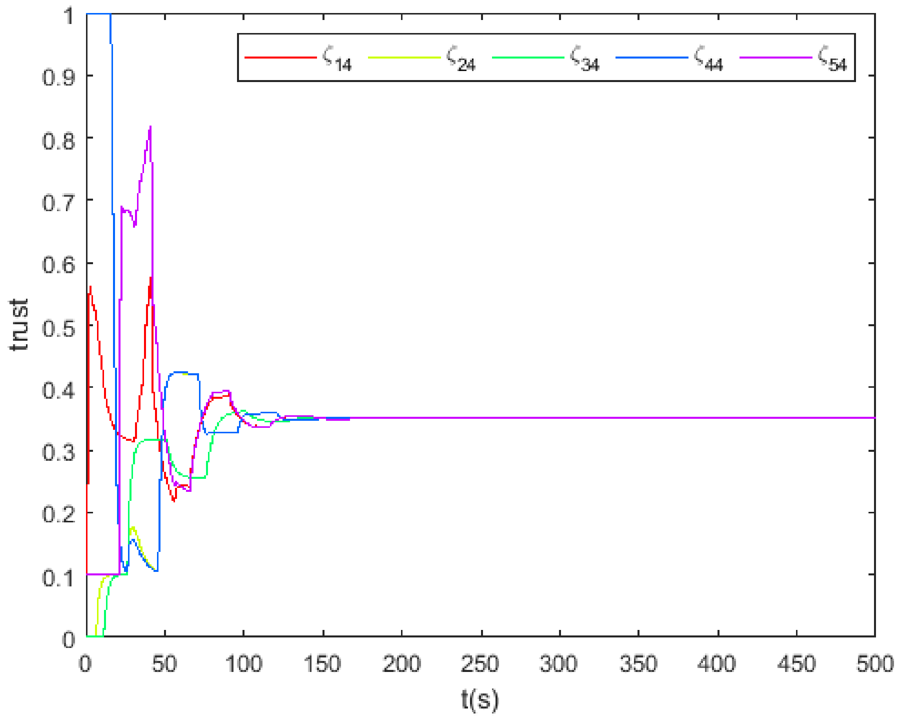

Figure 6.5: Trust values towards agent 4, with adaptation gain $K = 0{,}2$

As can be seen in Figure 6.4 the agents reach consensus on the trust values towards agent 4. In this case with lower adaptation gain ($K = 0{,}02$) the consensus is reached slower than in the second case where the adaptation gain is 10 times higher ($K = 0{,}2$), as can be seen in Figure 6.5. The adaptation gain though affects the trust calculations as well. Since the trust values are calculated much faster, i.e. the change in the confidence adaptation is higher, this results with convergence towards different trust value towards agent 4 compared to the first case, when the adaptation gain is 10 times smaller. However, since all the agents adapt their trust values to the new adaptation gain value, the final agreed trust value towards the other agents in the group also change.

In Figure 6.6 it is shown whether the agents can reach mutual agreement on the measurement values, when the adaptation gain is $K = 0{,}2$ and the trust values towards the agents in the group differ from the case when $K = 0{,}02$.

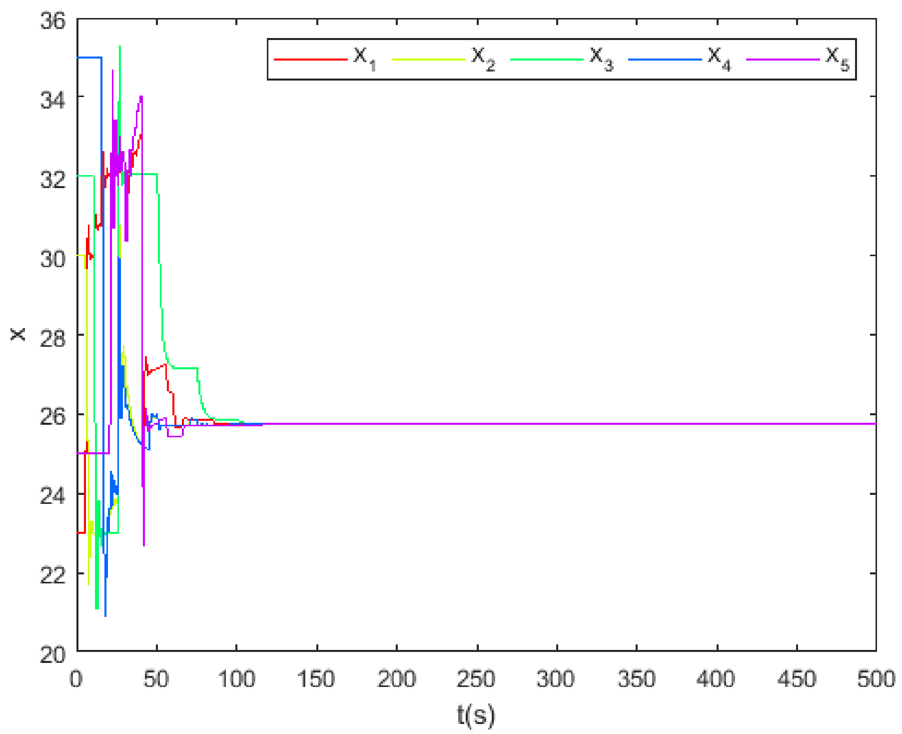

Figure 6.6: Convergence of measurement values with adaptation gain $K = 0{,}2$

From Figure 6.6, it can be seen that the agents reach mutual agreement which corresponds to the initial measurement values of the agents, despite the change of the trust values in the group. The consensus agreement between the agents on the measurement values will be discussed in Chapter 6.2.

3. In this scenario, the influence of the communication period of the channel on the trust values is analysed. Since the agents are calculating the trust values locally until they receive information from the agent which is currently sending on the channel, and at that moment all of the agents update the corresponding trust values which are being received from the sending agent, as the transmission is broadcast and continue calculating the trust values with the newly trust values obtained. Therefore, it is understandable to analyse how this trust values will change if the agents send their values with higher delays. The tested value of the communication period is set to be 4 times higher than the original communication period of $T = 5$ s.

   The initial set up of the group is same as Section 6.1, scenario 2, with $K = 0{,}02$ and the communication period of $T = 5$ s. Therefore, the outcome of this scenario is same as shown in Figure 6.4. However, the result will also be shown for simulation period of $[0 - 1000]$ s. Next, the communication period is increased 4 times than the originally set communication period to $T = 20$ s. Here, the convergence time of the agents' trust values towards agent 4 will be examined. Since the agents update their trust values much slower, the convergence time is also expected to be slower as well. The results are given in Figure 6.7 and Figure 6.8.

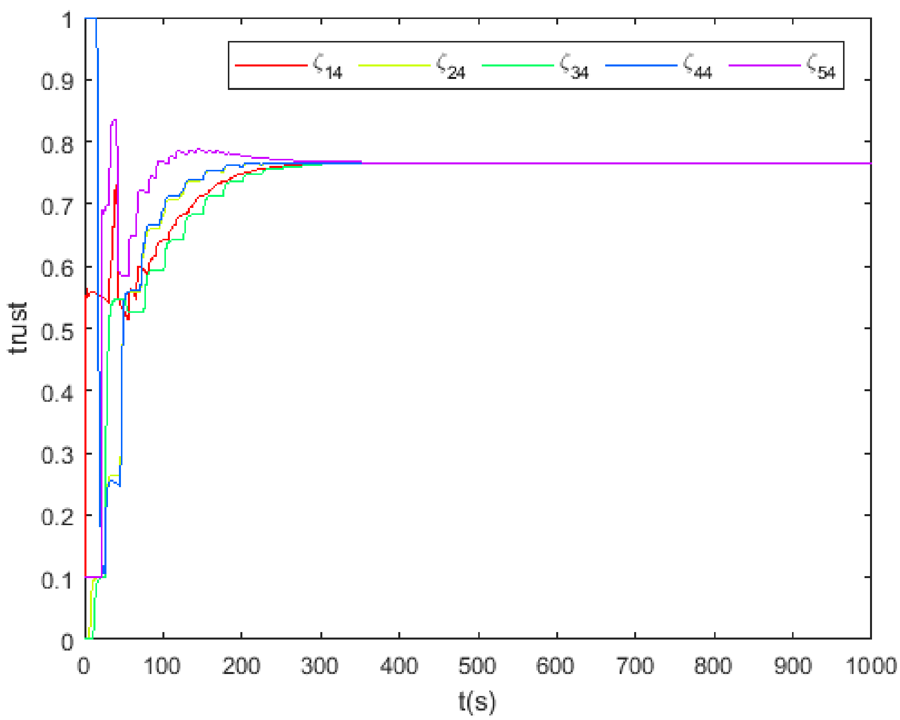

Figure 6.7: Trust values towards agent 4 with communication period $T = 5$ s

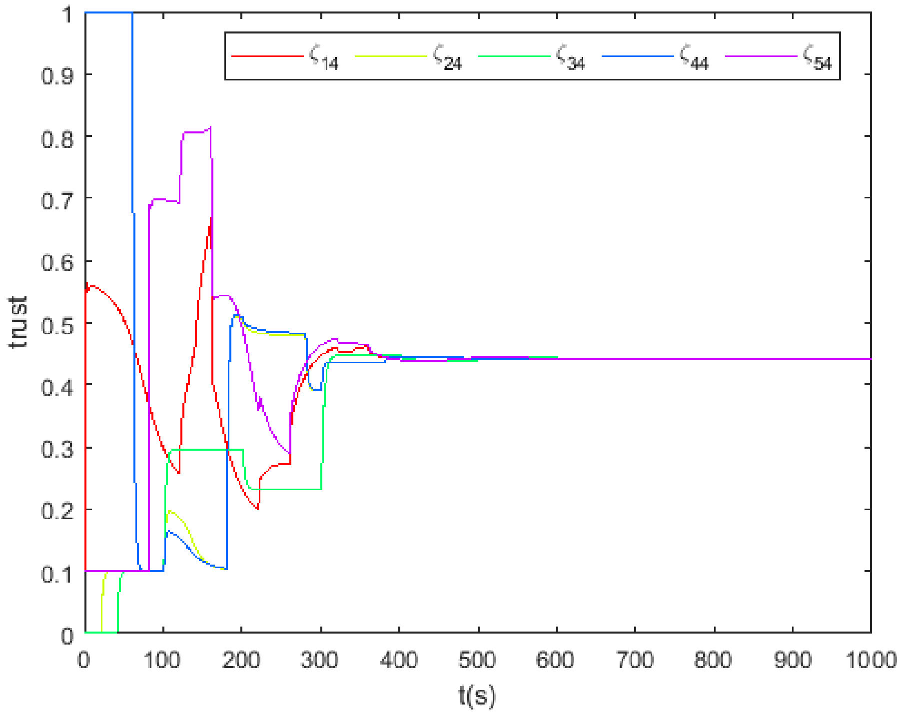

Figure 6.8: Trust values towards agent 4, with communication period $T = 20$ s

As can be seen from Figure 6.9, the agents reach convergence on the trust values towards agent 4 slower when the communication period is higher. It can also be seen that the trust values converge towards smaller value as well. Since the trust values towards the other agents change as well, this outcome might still be acceptable. To test whether the agents will reach an agreement on the measurement value, when the communication period is $T = 20$ s and the trust values towards the agents change, according to Figure 6.8, in Figure 6.9 it is shown the convergence towards mutual value between the agents.

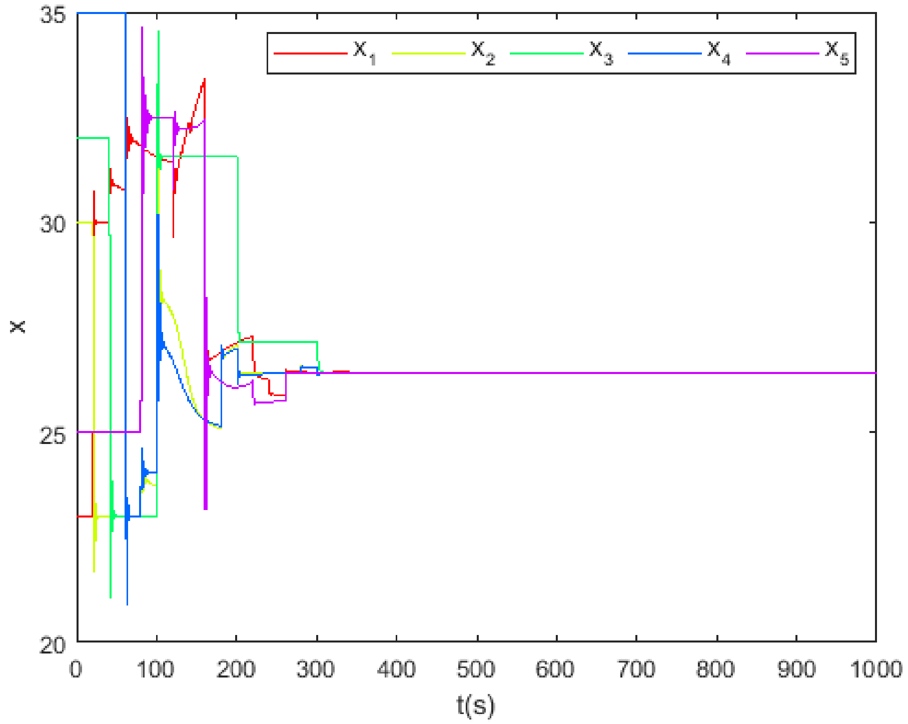

Figure 6.9: Mutual agreement of the agents on the measurement value, with $T = 20$ s

From Figure 6.9 it is possible to see that the agents still reach a consensus over the measurement value in an acceptable range. Therefore, the communication period might affect the trust calculations of the agents, however sicne this applies to all trust values, the agents are still capable of reaching an agreement on the measurement value.

6.2 Simulation results of consensus algorithm on scheduled acoustic communication protocol, by using trust algorithm

This part is dedicated to analyzing the final measurement value which will be agreed by the agents in the multi-agent system, based on the trust values which each of the agents hold towards its neighbors. Since the end goal of implementation of the trust algorithm is in a way directed towards eliminating or reducing the influence of the incorrect measurement value which is obtained by some agent in the group, this part will examine how close is the mutually agreed value to the value that most of the agents in the group measure.

Therefore, multiple scenarios will be assessed, and the results will be presented. The same initial parameters are used as given at the beginning of Chapter 6.

1. The first scenario examines the behaviour of the system when all the agents have the same measurement values. δ values in this case are 0. All the other initial parameters correspond with the parameters given at the beginning of Chapter 6.

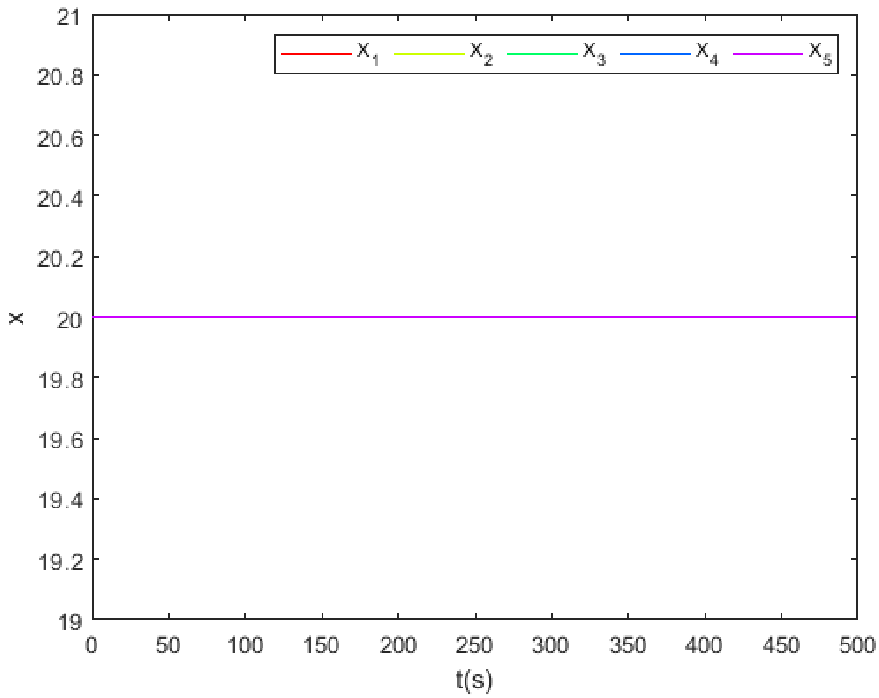

Figure 6.10: Convergence of the measurement value with $\delta = 0$

As can be seen from Figure 6.10 there is no difference in the mutually agreed value by the agents in the swarm from the actual (assigned for the purpose of simulations) measurement values of the agents.

2. The second scenario in this part examines the agreement on the measurement value by the group in case all the agents in the group have different measurements, although none of the agents have considerably higher/lower value than the rest of the agents in the group. The measurement vector is given as

$$x = [23\ 30\ 32\ 35\ 25]$$

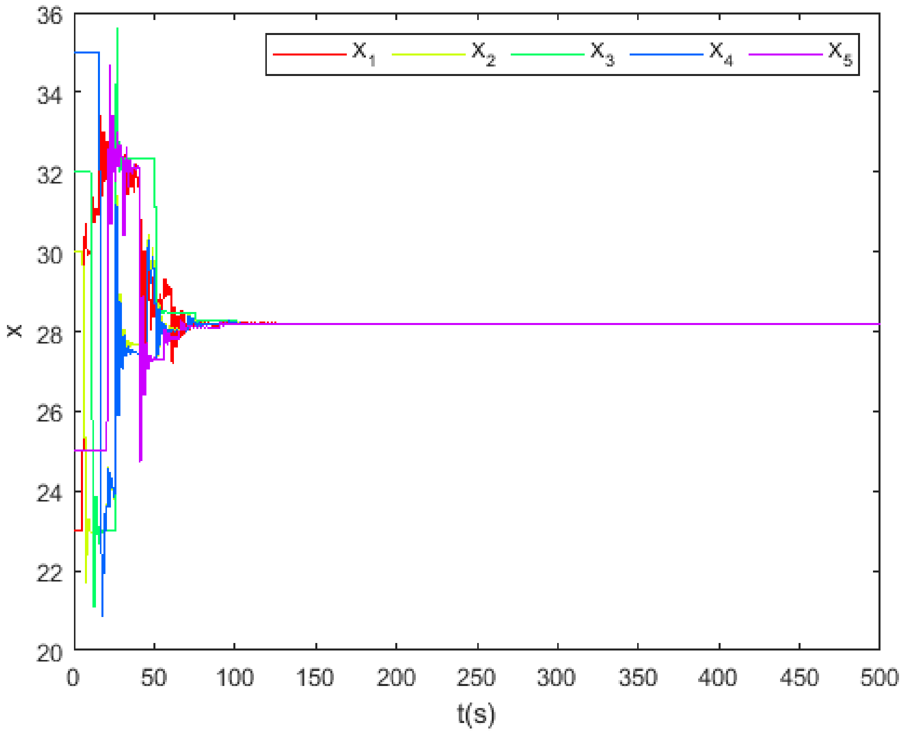

Figure 6.11: Convergence of the measurement values when the agents have similar values

From Figure 6.11 it can be seen that all the agents reach consensus on the measurement value, using the trust values they hold towards their neighbours and the agreed value corresponds to the agents' measurements.

3. Next scenario in this part analyses the outcome of the implemented algorithms in case when one of the agents in the group experience sensor malfunctioning. For the simulation purposes, this is represented with the said agent having considerably higher value compared to the rest of the agents in the group. In this case, we expect the final value agreed by the group to correspond to the measurements that most agents in the group have.

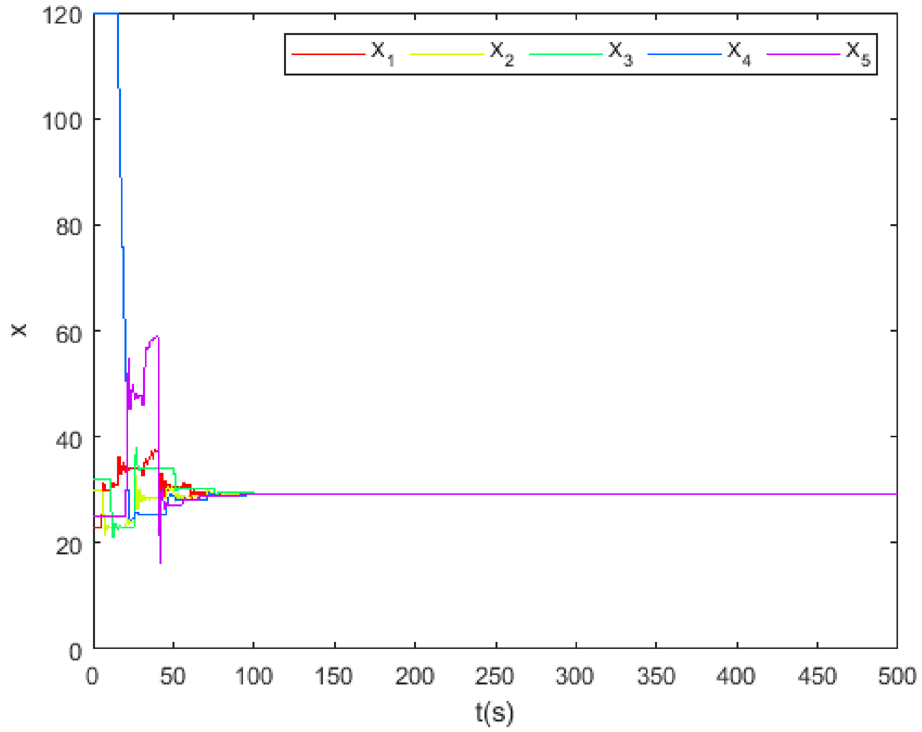

Figure 6.12: Convergence of the measurement values when agent 4 gives incorrect measurement

As it was assumed at the beginning, the group reaches an agreement and the agreed value corresponds to the measurements of most agents in the group. The group detects the high measurement given from agent 4, and adapts the trust values towards this agent, which result to diminish the influence which this measurement has upon the group. This is shown in Figure 6.12.

4. Last scenario shows how the agreed final measurement value changes, in accordance with Section 6.1, scenario 1. Initially all the agents in the group have similar measurements, and at moment $t = 500$ s, agent 4 gives significantly higher value than the other agents in the group. Therefore, the corresponding δ values also change (increase). The change in the trust values towards agent 4, is discussed in Section 6.1, scenario 1, and here the change of the agreed final value in this scenario will be presented. The result can be seen in Figure 6.13.

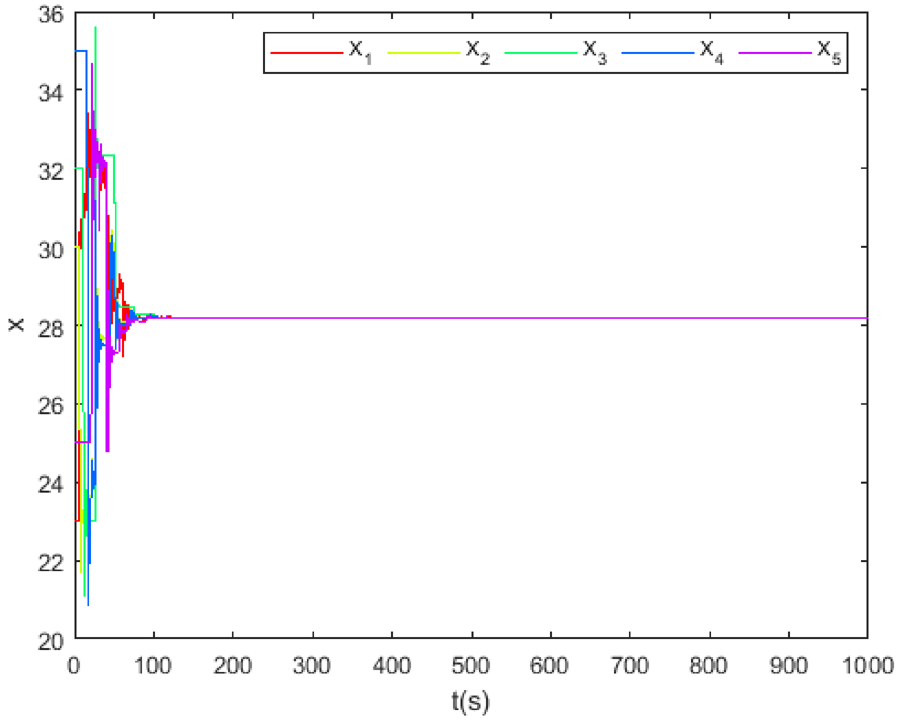

Figure 6.13: Convergence of the measurement value, with agent 4 giving incorrect measurment at $t = 500$ s

As can be seen from Figure 6.13, the mutually agreed value reached by the time $t = 500$ s did not change when the measurement value of the agent 4 significantly increased at $t = 500$ s. The trust and confidence values of the agents towards agent 4 adapted to the newly established situation, as shown before and the already agreed measurement value did not change.

7 WORK ON AMUSSEL AGENTS

The final stage of the work was dedicated to testing the proposed solution on the aMussel agents. Due to the limited time and the time-consuming hardware testing, this part was not fully completed. However, in this Chapter 7 the mode of operation of the aMussels will be elaborated, and the acoustic communication of the agents will also be discussed.

As it was discussed in Section 1.2, the aMussel agents are equipped with microcontroller unit from Cypress Semiconductors and PSoC Creator IDE is being used for programming the aMussels. As the computational power of the microcontroller unit is rather low [15], additional raspberry PI unit is provided. The programming of the aMussel agents is performed on the surface, whereas the acoustic communication can be tested underwater, or on the surface where it can operate on small distance of approximately 20 cm[1].

For connecting to the aMussel agents, BLE-USB bridge featuring Bluetooth 4.1 is being used [23]. In the identification phase, list of all found aMussel is given, as shown in Figure 7.1 and for each aMussel agent, the index and its number are shown.

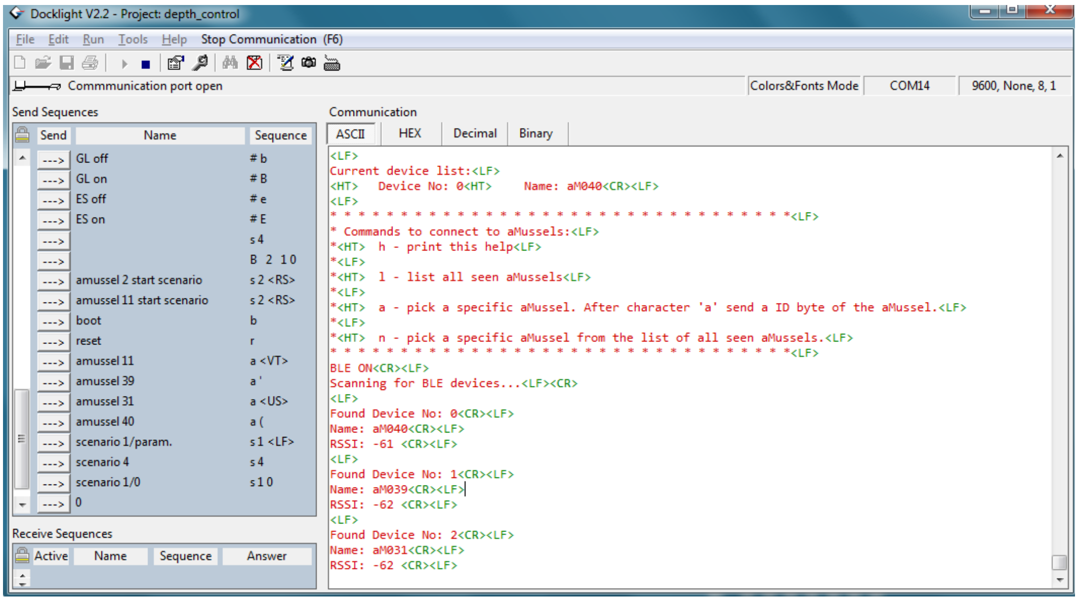

Figure 7.1: aMussel devices identification process

Since there are 120 aMussel units as part of the swarm, the numbers of the agents can vary from 1 to 120, depending on the agents which are being used for operation. For work with the aMussel agents, connection to one aMussel agent at a time is needed.

[3] this has been discussed with the supervisors

This is accomplished by entering the corresponding sequence of characters in the Docklight tool, which corresponds to the agent's assigned number, in range of 1 to 120. Once the connection is established, it is possible to test different functionalities of the aMussel agent, such as reading the pressure, temperature or ambient light, reading the battery voltage, testing the motors which results with linear movement of the piston in the buoyancy system, therefore causing the aMussel agent to either sink to the bottom or dive to the surface (in case the testing is done in testing area underwater). Other possible functionalities can also be tested. The sequence of characters which needs to be entered in order to observe some functionality, is programmed in the algorithm. In Figure 7.2 some examples of the defined sequence of characters can be seen, as well as the established connection with aMussel agent number 40.

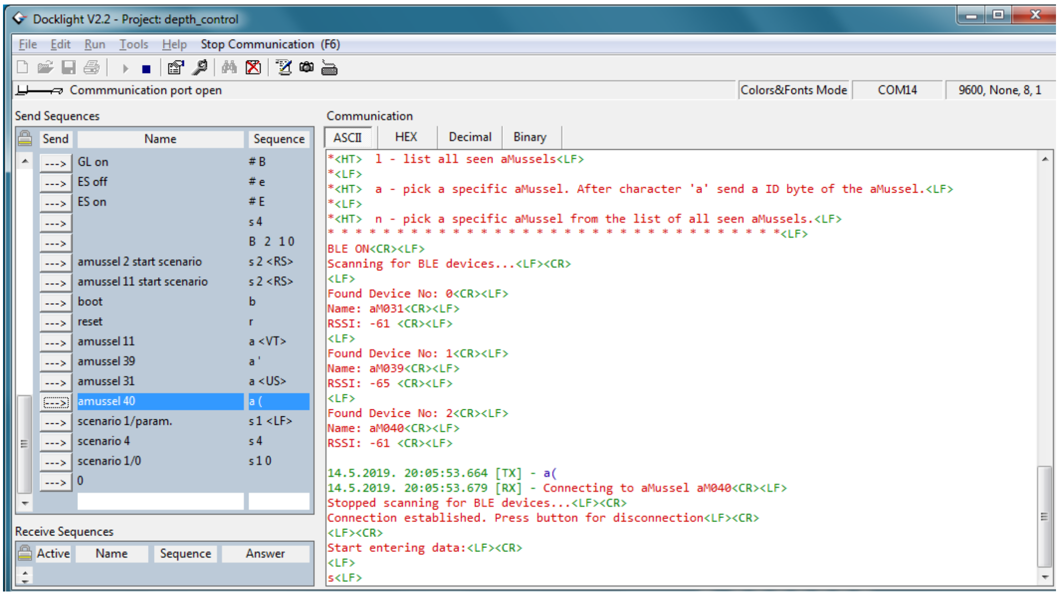

Figure 7.2: Established connection with aMussel agent no. 40

Once the work on one aMussel agents is finished, it is needed to disconnect from the agent and another identification of the active aMussel agents in the group to be performed. By using the corresponding character for the assigned number of the aMussel agent, the connection to the particular aMussel agent is established.

As the programming of the aMussel agents is performed wirelessly over network, and as discussed in Section 1.2, the Wi-Fi communication is provided with the raspberry PI unit, therefore, connection with the raspberry PI must be established. This is done by entering the bootload mode of each aMussel, and the programming is therefore performed using GUI interface, as shown in Figure 7.3.

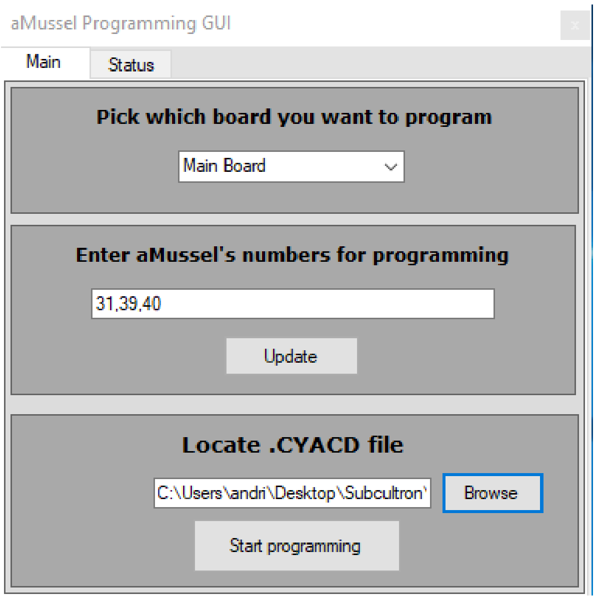

Figure 7.3: GUI interface for aMussel programming

Once the programming of the aMussel agents is finished, the whole process for establishing communication with each aMussel individually, should be performed again.

As for this work, establishing of acoustic communication between the agents is needed, testing of the acoustic communication, as well as the scheduled communication protocol defined was also tested.

Since the aMussel agents are equipped with nanomodem units, the work mode of the nanomodem is described in [20].

For establishing the acoustic communication on the agents which are active at that moment, configuration messages about the swarm need to be send via the UART-nanomodem unit. This includes information for all the agents in the swarm, including aPads, aFish and aMussels, with setting the corresponding bits of the maximal 7 bytes, which can be send in one message [20]. The nanomodem units can operate in unicast and broadcast mode, as given in [20], and therefore the corresponding set up for the acoustic communication mode needs to be performed.

For our application, only the broadcast mode was tested. Finally, the mode of communication needs to be defined, and for our case it is round-robin scheduling [9]. This is shown in Figure 7.4, where the agent number 40 receives the configuration parameters.

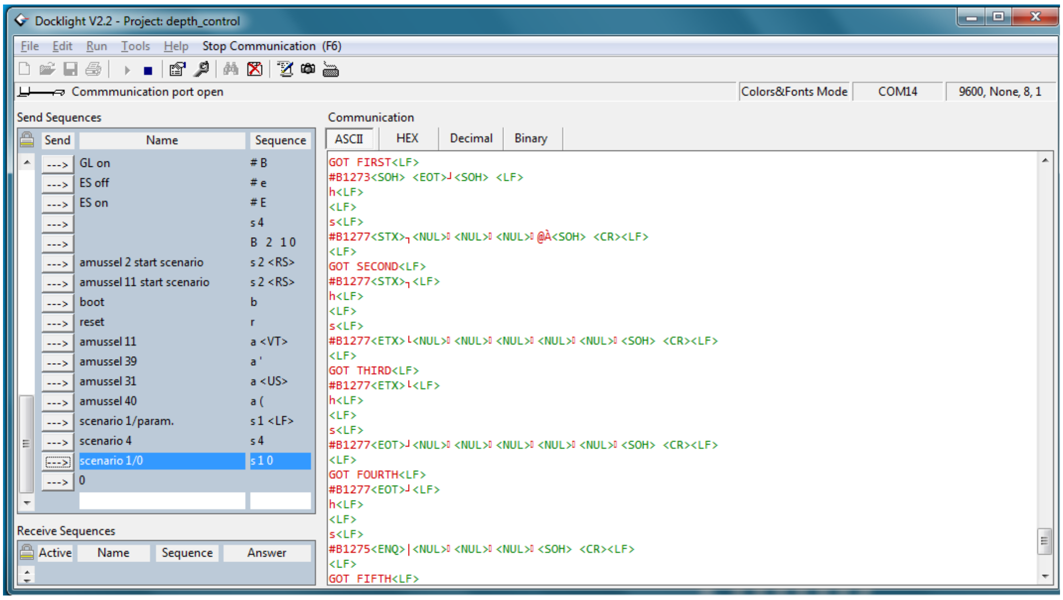

Figure 7.4: Configuration messages received by aMussel agent no. 40

Once the configuration of the system is finished, the corresponding scenario should be started on each aMussel agent. For the testing purposes, simple scenario was created, where each agent measures the ambient light and sends it over acoustics when its term for transmission on the channel comes. This is shown in Figure 7.5.

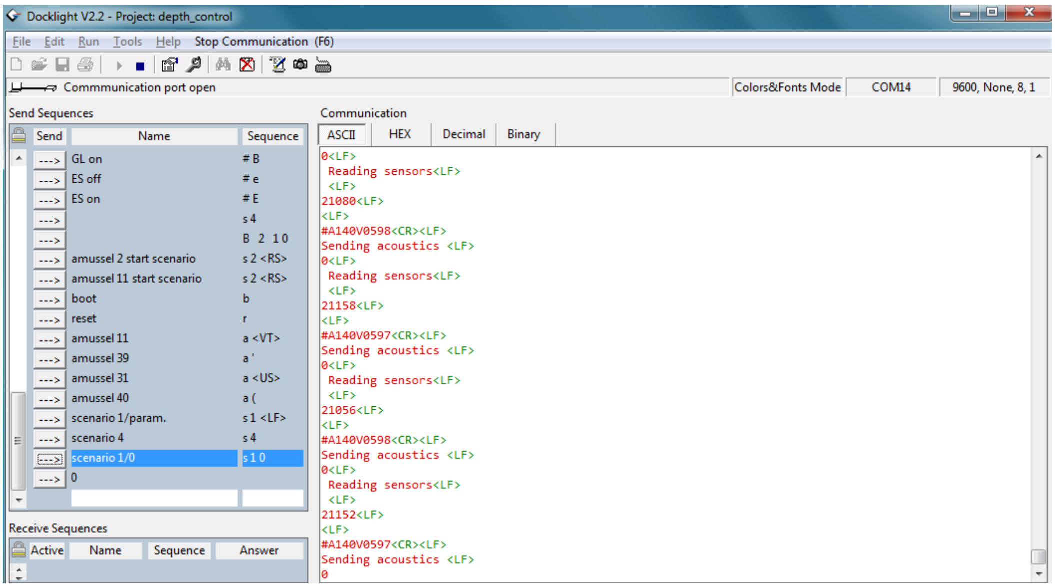

Figure 7.5: Scenario started for reading ambient light and sending the measurement over acoustics

There are few problems which have occurred during the testing phase. The first problem was difficulties in establishing the acoustic communication between the agents.

Second the round-robin mode could not be established as it is defined. Once the configuration and set up of the system was performed, round-robin scheduling mode was entered. Once the index of the aMussel is read and it is established that the agent is first for communication, it was only this agent transmitting on the channel and the other agents were not possible to receive the measurement value. For the next testing however, the agent was able to transmit on the channel and the next agent in line was able to receive the measurement value, but not the third agent in line for communication. Once the second agent in line for communication has received the measurement, next it reads its own ambient light measurement and sends it over acoustic. This is shown in Figure 7.6.

Again, the same problem as discussed above occurs.

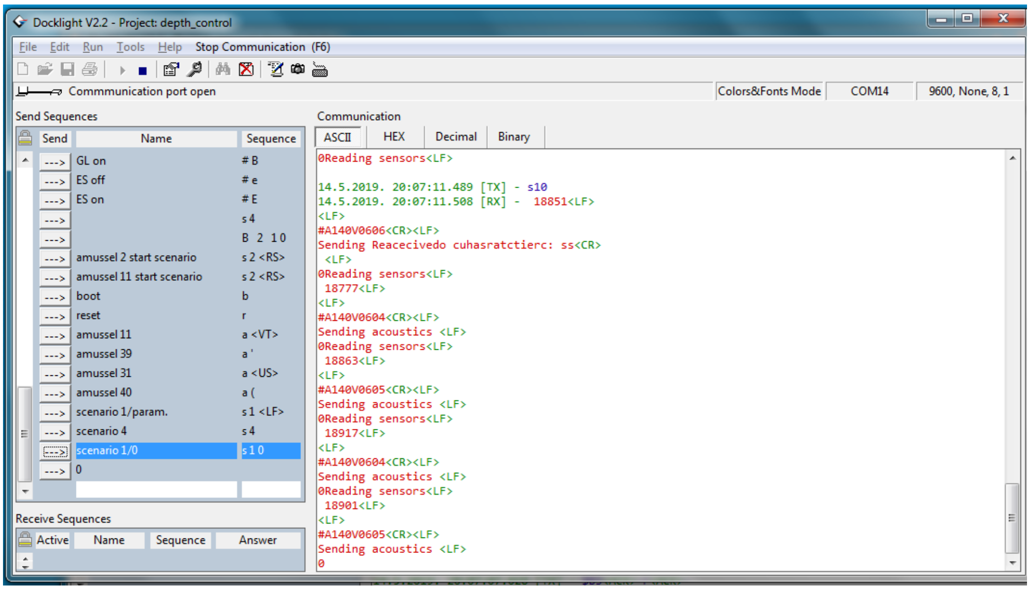

Figure 7.6: Received measurement on aMussel agent no. 40 and reading and sending the ambient light measurement over acoustics

At this point it was not possible to determine the cause of the problem, and therefore the acoustic communication could not be established fully. The acoustic communication however is main requirement for implementation of the already defined and discussed algorithm. Once the acoustic communication and the communication protocol is fully tested and established, only then the implementation of the algorithm can be performed.

As possible reasons for failure in establishing the acoustic communication the following can be listed: the acoustic testing in air, although the acoustic communication was functional when performing the configuration of the system; longer period of operation before the acoustic communication in the system is established and finally unnoticed bug in the tested algorithm.

Due to the limitation of time, it was not possible to proceed further with the testing and to determine the possible reason for the described problem in acoustic communication.

SUMMARY

The work presented in this thesis is associated with the subCULTron underwater swarm of robots. The subCULTron Project is envisioned to operate as an artificial swarm, composed of three types of autonomous agents, in order to perform long term monitoring of the underwater ecosystem in the Venice Lagoon, Italy. The three types of agents as part of the swarm are aMussel, aPad and aFish agents. Each of these agents has a certain role in the swarm towards achieving the end goal of the system. The work in this thesis was closely associated with the aMussel agents regarding their depth control in the underwater environment. The depth control of the aMussel agents has been introduced with the purpose of monitoring certain parameters at prespecified depth in the canals in Venice Lagoon. Therefore, depth control algorithm has been implemented on each aMussel agent by using pressure sensor measurement as reference signal. As the aMussel agents are envisioned to work in multi-agent system in order to provide long term monitoring of the environment, they should be able to exchange information and reach mutual agreement on the parameter which is being exchanged. For this purpose, all of the aMussel agents are equipped with nanomodem units and consensus and consensus-based algorithms are used and analysed, so the agents can reach mutual agreement on the measurement value, for example pressure measurement, and this value will then be used as an input to the depth control algorithm, in order to control the depth of the aMussel agents. One solution has already been implemented on the system which uses average consensus algorithm on scheduled acoustic communication defined. With this solution, the agents are able to reach consensus over the measurement value, however in case of incorrect sensor measurement, the sensor malfunction is not possible to be detected by the group and therefore this measurement is used by the average consensus algorithm, which result in achieving incorrect mutual value between the agents.

In order to eliminate this problem, one solution was proposed in this work. The proposed solution is to implement trust algorithm instead, which will take into account the trust values of the agents in the multi-agent system and use these trust values in order to reach consensus on the measurement value, by using consensus algorithm at the same time. Moreover, as scheduled acoustic communication protocol was also defined for the system in order to avoid interference in underwater communication, the proposed solution is implemented on the established communication protocol. This will affect the algorithm in a way that the corresponding trust values of the agent which is currently transmitting on the channel are updated accordingly by the other agents in the group, while performing the trust algorithm, but also the consensus algorithm is performed by using these trust values. It is excepted the trust algorithm to reach convergence on

the trust values, and so it is also excepted the consensus algorithm to reach a convergence towards mutual value. Due to these reasons and due to the decentralized fashion of the system, the implementation of the solution was not straightforward. Understanding the theory behind these algorithms and their operation, as well as the troubleshooting for errors, was lengthily process. However, at the end of the work, the solution was fully implemented and tested through simulations, for various possible scenarios and for different parameters which can influence the outcome of the algorithm and the results were presented and discussed. As the algorithm was tested for specific cases, as such, the results for the tested scenarios are acceptable. However, to be certain in the validity of the results, and to know that this algorithm will work in every case, mathematical proof must be provided. The convergence of the trust algorithm alone is proven [8], however the convergence of the trust algorithm with the scheduled acoustic communication, still needs to be confirmed.

The testing of the solution on the aMussel agents however, was not fully implemented, mostly due to the problems which occurred during the testing of the acoustic communication between the agents. Establishing the acoustic communication was needed in order to implement and moreover to test the algorithm.

However, with the in-depth simulation analysis at this point, there is a solid ground for future work on the system. Detailing the behaviour of the algorithm and elaborating how different parameters influence the outcome, can be a solid reference for future work. The next step would be full implementation of the algorithm on the aMussel agents and confirming the theoretical aspects, as well as the simulation results obtained.

LIST OF REFERENCES

[1] Bazeille S., Lebastard V., Boyer F., "Underwater Robots Equipped with Artificial Electric Sense for the Exploration of Unconventional Aquatic Niches", Chapter 3 from book Marine Robotics and Applications, *Ocean Engineering & Oceanography*, Springer International Publishing AG 2018

[2] https://www.eea.europa.eu/themes/water/intro (20.05.2019)

[3] http://zool33.uni-graz.at/artlife/cocoro (20.05.2019)

[4] http://www.subcultron.eu/ (20.05.2019)

[5] Olfati-Saber R., Murray R. M., "Consensus Problems in Networks of agents With Switching Topology and Time-Delays", *IEEE Transactions on Automatic Control*, Vol.49, No.9, September 2004

[6] Olfati-Saber R., Fax J. A., Murray R. M., "Consensus and Cooperation in Networked Multi-Agent Systems", *IEEE,* Vol. 95 No.1, January 2007

[7] Haus T., Palunko I., Tolic D., Bogdan S., Lewis F. L., "Decentralized trust-based self-organizing cooperative control", *2014 European Control Conference (ECC),* Strasbourg, France, June 24-27 2014

[8] Haus T., Palunko I., Tolic D., Bogdan S., Lewis F. L., Mikulski D. G., "Trust-based self-organizing network control", *IET Control theory Appl.,* Vol. 8, Iss. 18, pp. 2126-2135, 2014

[9] Arbanas B., Petrovic T., Bogdan S., "Consensus Protocol for Underwater Multi-Robot System Using Scheduled Acoustic Communication", *OCEANS - MTS/IEEE Kobe Techno-Ocean (OTO),*, May 2018

[10] Mazdin P., Arbanas B., Haus T., Bogdan S., Petrovic T., Miskovic N., "Trust Consensus Protocol for Heterogeneous Underwater Robotic System", *2016 IFAC CAMS,* Trondheim, Norway, September 13-16 2016

[11] http://www.subcultron.eu/tag/afish/ (20.05.2019)

[12] http://www.subcultron.eu/wp-content/uploads/2015/10/subCultron_presserelease_en.pdf (20.05.2019)

[13] Thenius R., Moser D., Varughese J.C., Kernbach S., Kuksin I., Kernbach O., Kuksina E., Miskovic N., Bogdan S., Petrovic T., Babic A., Boyer F., Lebastard V., Bazeille S., Ferrari G.W., Donati E., Pelliccia R., Romano D., Jansen Van Vuuren G., Stefanini C., Morgantin M., Campo

A., Schmickl T., "subCULTron - Cultural Development as a Tool in Underwater Robotics", P.R Lewis et al. (Eds.): ALIA 2016, *CCIS 732*, pp. 27-41, 2018, Springer International Publishing AG, part of Springer Nature 2018

[14] H2020-FETPROACT-2014, "subCULTron (Submarine cultures perform long-term robotic exploration of unconventional environmental niches), Deliverable D1.2, Technical characteristic & performance specification of Hardware", EU-H2020 Project No.640967, July 2016

[15] Loncar I., Babic A., Arbanas B., Vaisljevic G., Petrovic T., Bogdan S., Miskovic N., "A Heterogeneous Robotic Swarm for Long-Term Monitoring of Marine Environments", *Applied Sciences 9(7):1388*, April 2019

[16] SubCULTron, First Year Summary, SubCULTron News Vol.1, 30th April 2016

[17] SubCULTron - Exploring and Monitoring the Venice Water Channels, First Year Summary, SubCULTron Flyer Vol.1, 30th April 2016

[18] Docklight V2.2 User Manual 07/2016, 2016 Flachmann und Heggelbacher GbR

[19] Stojanovic M., Beaujean P.P., "Acoustic Communication", Chapter in book Autonomous Sea Surface Vehicles (pp.359-386), Article, January 2016

[20] Neasham J., Nanomodem v2.1 User Guide, School of Electrical and Electronic Engineering, Newcastle University

[21] Vidal J. M., "Fundamentals of Multiagent Systems with NetLogo Examples", March 2010

[22] Vlassis N., "A Concise Introduction to Multiagent Systems and Distributed AI", Intelligent Autonomous Systems, Informatics Institute, University of Amsterdam, 2003

[23] CY5670 CYSmart USB Dongle Quick Start Guide, Cypress Semiconductor Corporation, updated June 2017

[24] Arbanas B., Babic A., Loncar I., Markovic M., Mandic F., Vasiljevic G., "subCULTron underwater measurements using aMussel - tutorial", 2018

[25] http://www.phillyseaperch.org/uploads/9/1/0/6/9106381/_buoyancy_for_hs.pdf (20.05.2019)

[26] https://ec.europa.eu/digital-single-market/en/news/cocoro-robot-swarms-use-collective-cognition-perform-tasks (20.05.2019)

[27] Ren W., Beard R., "Overview of Consensus Algorithms in Cooperative Control", Chapter 1 from Distributed Consensus in Multi-Vehicle Cooperative Control, Theory and Applications, XV, 319 p., 2008

[28] Graph Theory, MIT OpenCourseWare, 6.042J/18.062J Mathematics for Computer Science, Fall 2010

[29] Chung F., Richardson F., "Weighted Laplacians and the Sigma Function of a Graph", Contemporary Mathematics

[30] Farinelli A., Iocchi L., Nardi D., "Multirobot Systems: A Classification Focused on Coordination", *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics,* Vol.34, No.5, October 2004

[31] Babic A., Loncar I., Miskovic N., Vukic Z., "Energy-efficient environmentally adaptive consensus-based formation control with collision avoidance for multi-vehicle systems", *10th IFAC Conference on Control Applications in Marine Systems,* Trondheim, Norway, September 13-16, 2016

[32] Mohar B., "The Laplacian Spectrum of Graphs", Department of Mathematics, University of Ljubljana; Article, Ed. Y. Alavi, G. Chartrand, O. R. Oellermann, A. J. Schwenk, Wiley, "Graph Theory, Combinatorics, and Applications", Vol. 2, 1991, pp. 871–898

[33] Mikulski D. G., Lewis F., Gu E. Y., Hudas G., "Trust method for Multi-Agent Consensus", Conference Paper, *Proceedings of SPIE - The International Society for Optical Engineering,* April 2012

[34] Babic A., Mandic F., Vasiljevic G., Miskovic N., "Autonomous docking and energy sharing between two types of robotic agents", Conference paper, 11th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles – CAMS 2018, opatija, Croatia, September 2018

APPENDIX 1: ILUSTRATIONS OF THE SYSTEM’S FUNCTIONALITIES

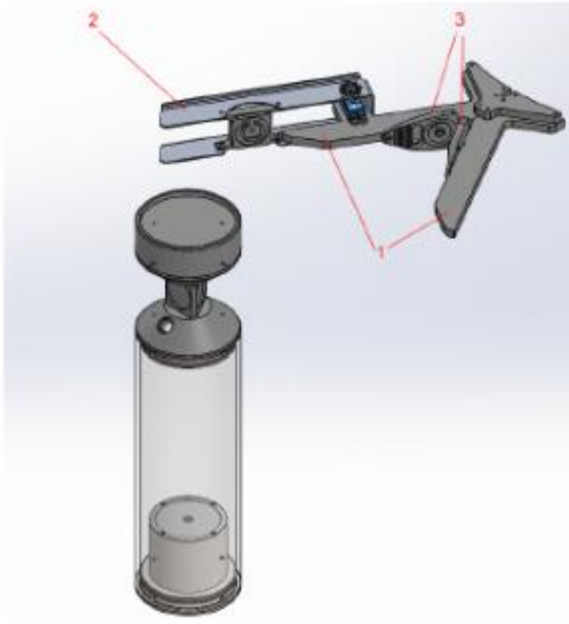

Figure 1: Docking mechanism of the aPad, for capturing the aMussel agent, 1) immovable delrin lever, 2) motorized aluminium shutter, 3) charging dock [34]

Figure 2: Sequence of docking of the aMussel agent [34]

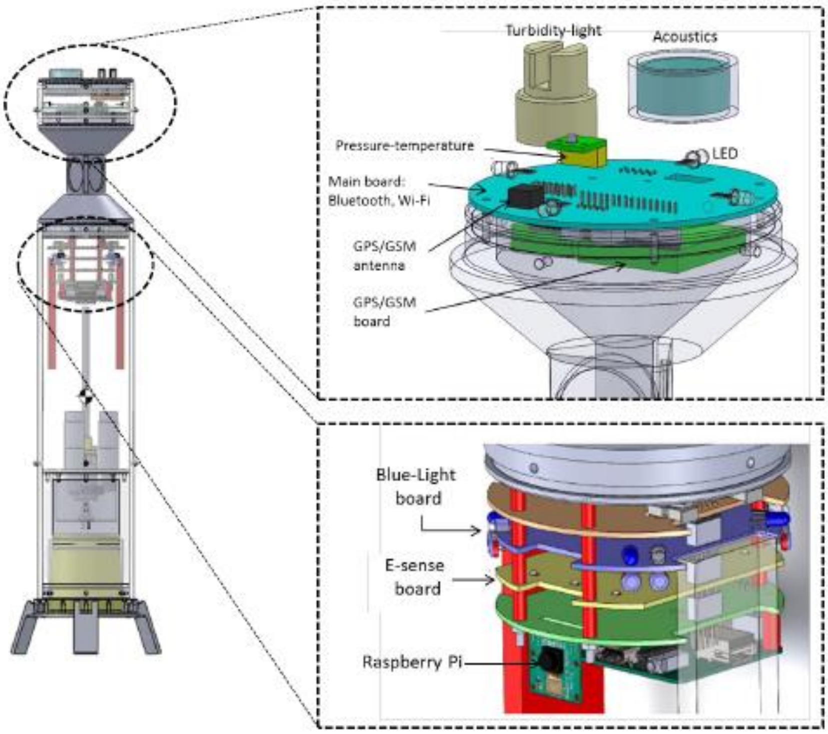

Figure 3: Sensors and communication modules placement on aMussel agents and electronics [14]

APPENDIX 2: SIMULATION RESULTS FROM SECTION 6.1 - SCENARIO 1

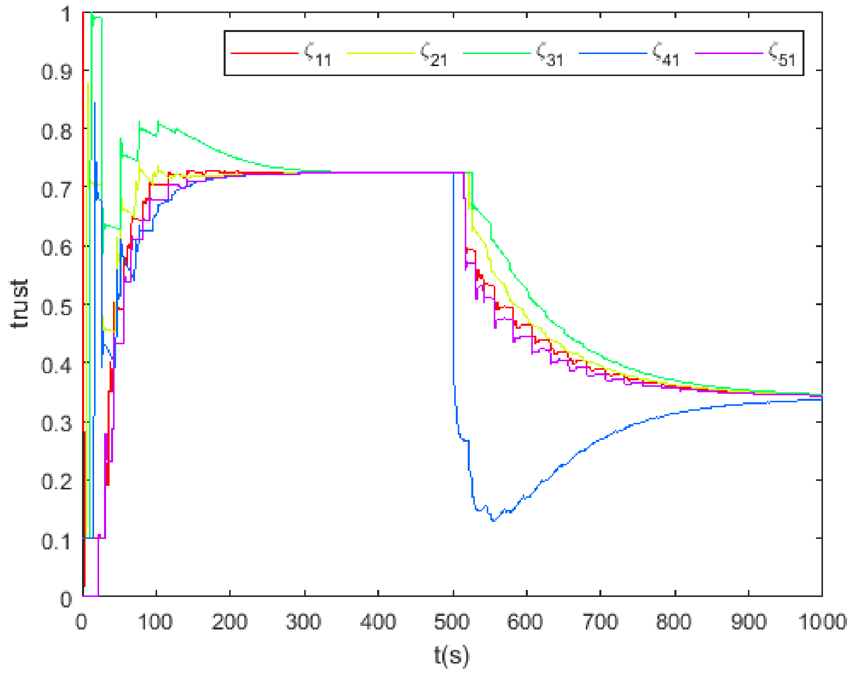

Figure 4: Trust values towards agent 1

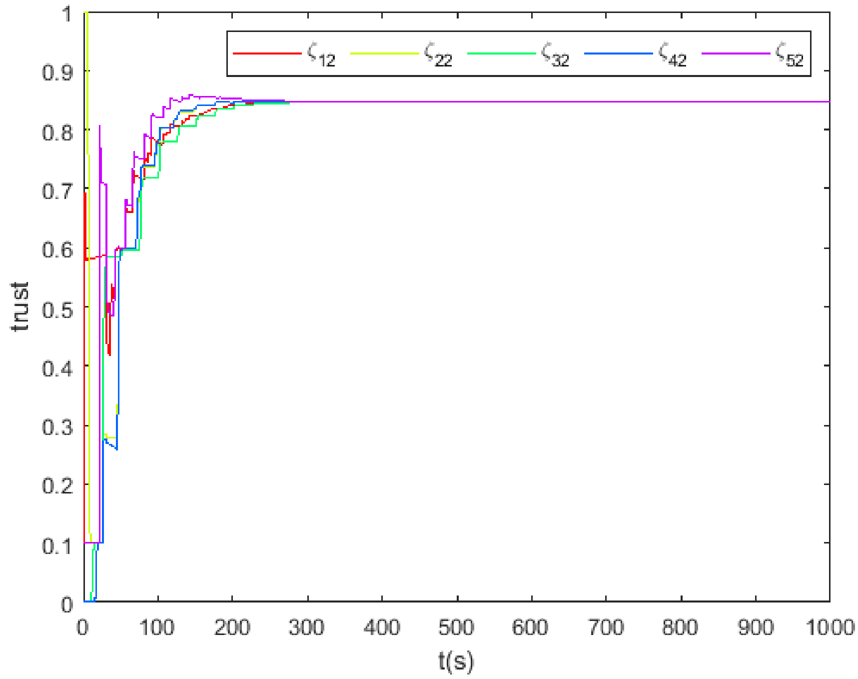

Figure 5: Trust values towards agent 2

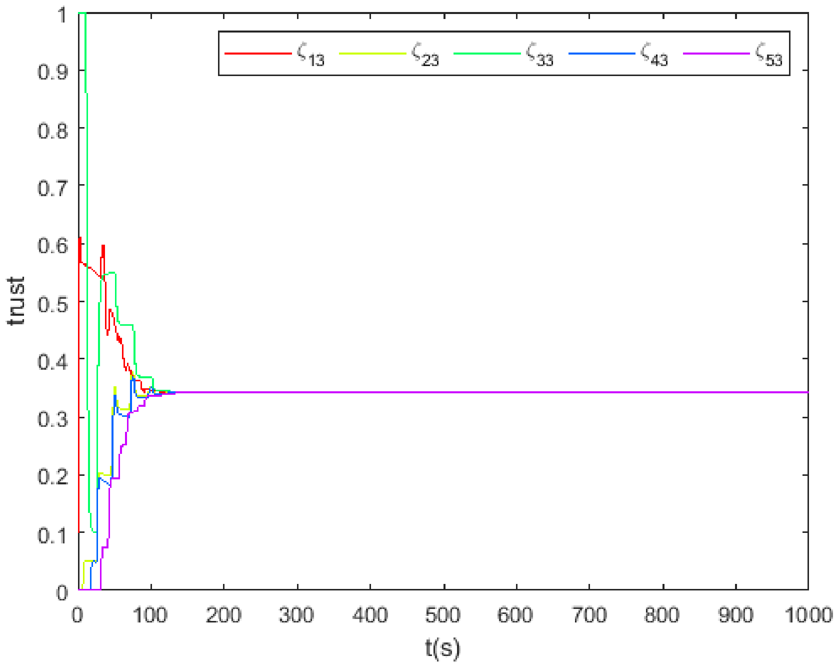

Figure 6: Trust values towards agent 3

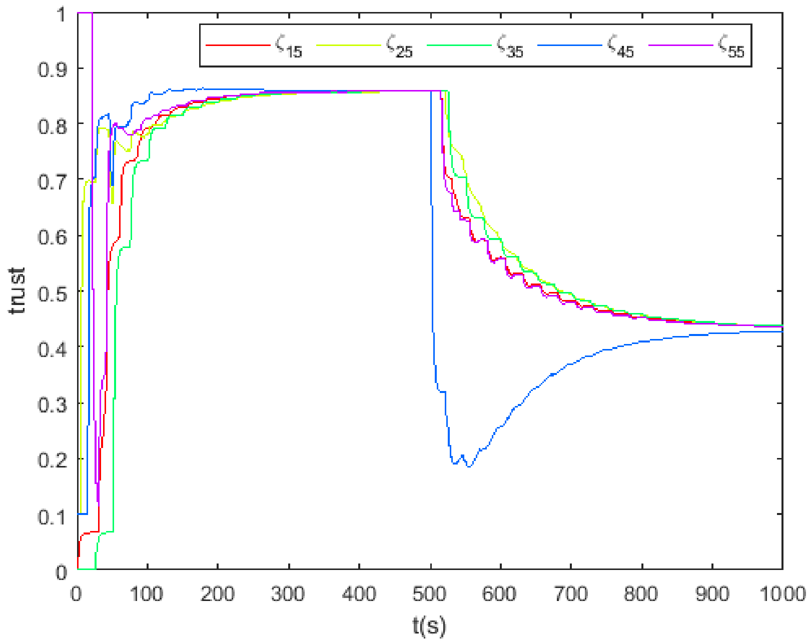

Figure 7: Trust values towards agent 5

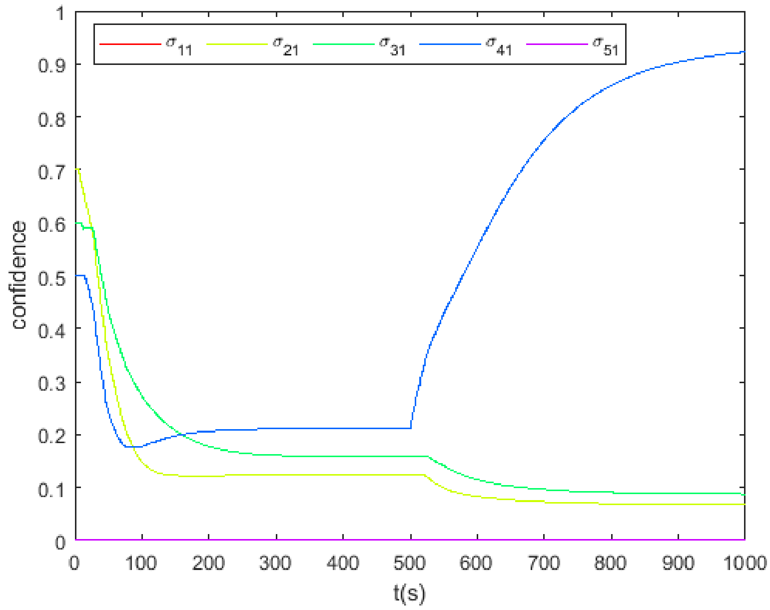

Figure 8: Confidence value regarding agent 1

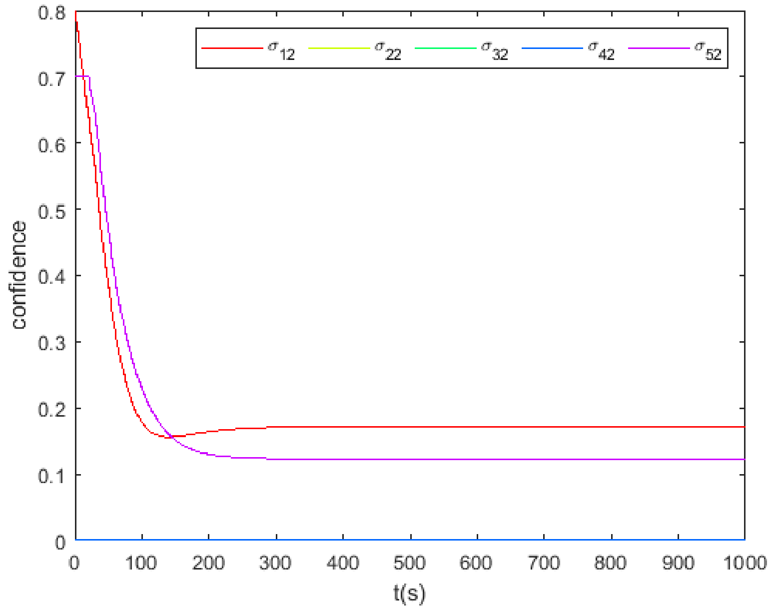

Figure 9: Confidence value regarding agent 2

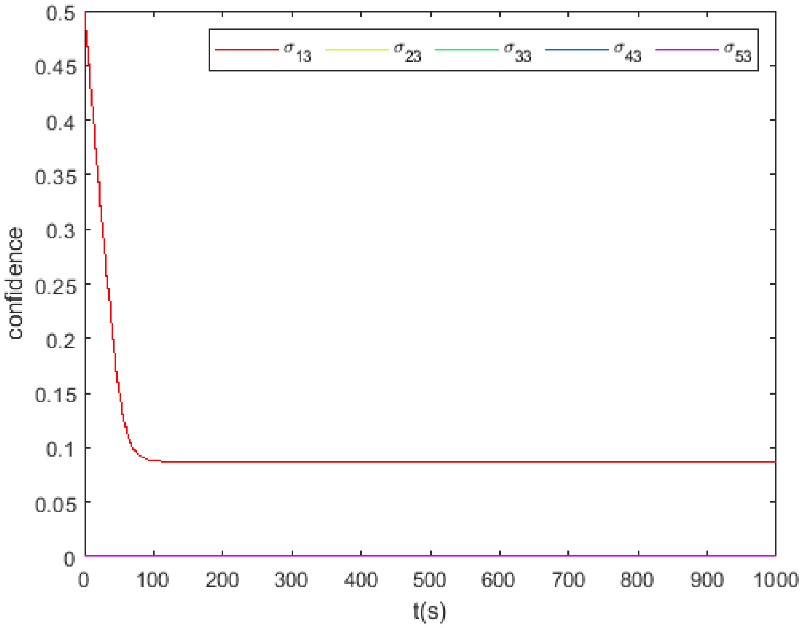

Figure 10: Confidence value regarding agent 3

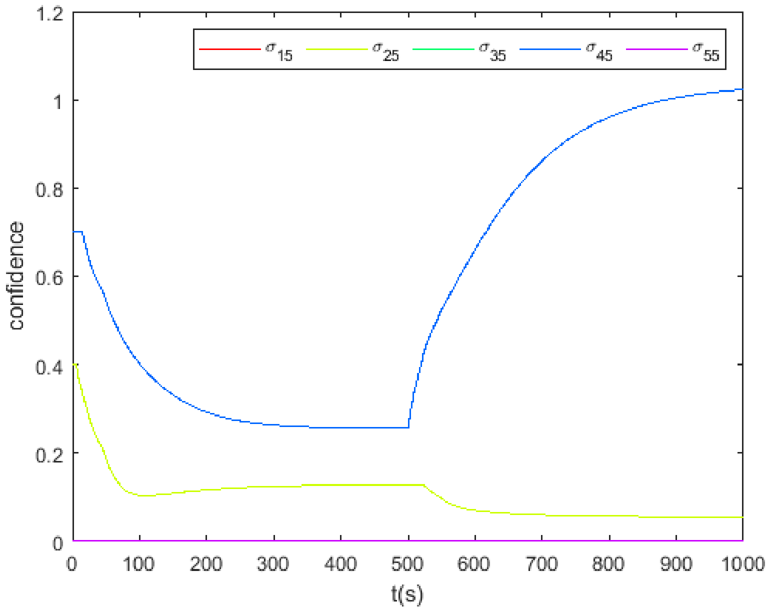

Figure 11: Confidence value regarding agent 5

APPENDIX 3: SIMULATION RESULTS FROM SECTION 6.1 - SCENARIO 2

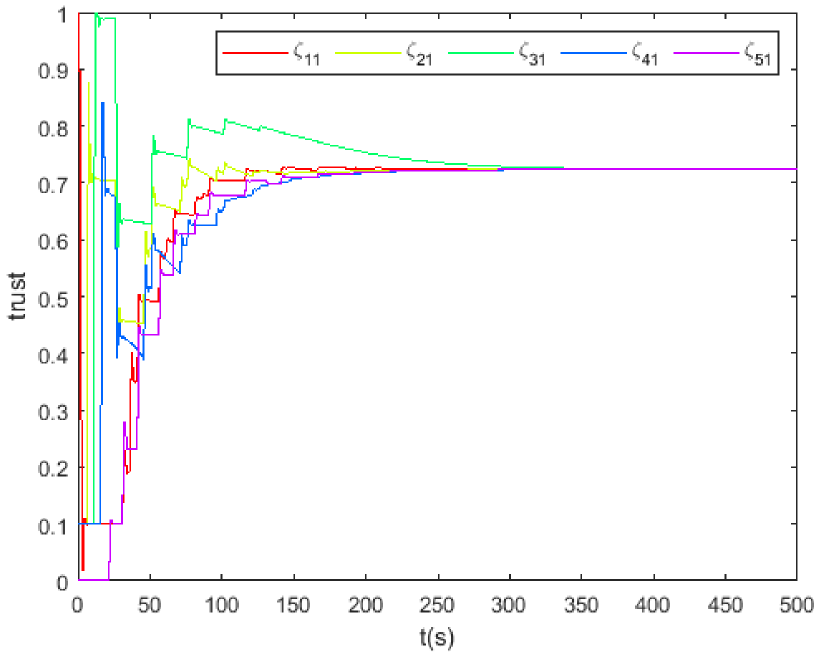

Figure 12: Trust value towards agent 1, with $K = 0{,}02$

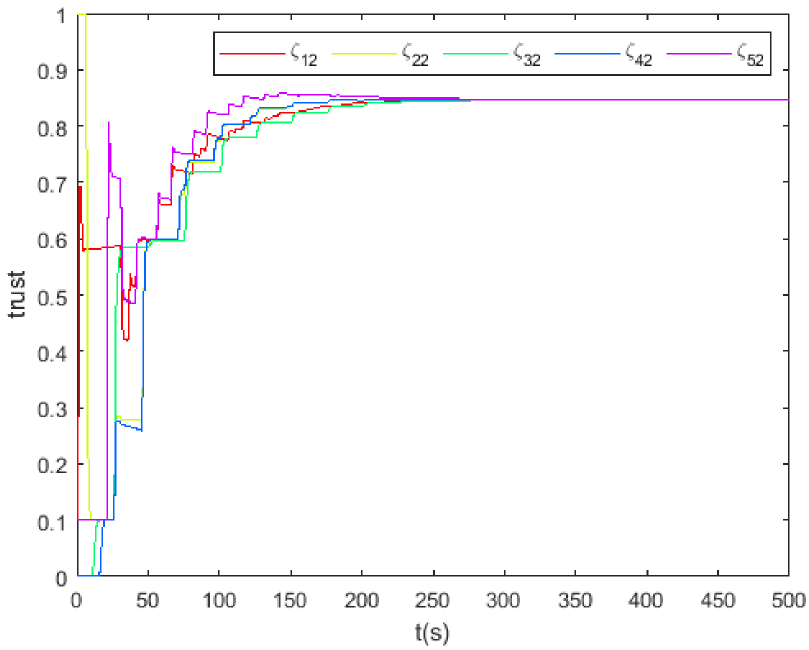

Figure 13: Trust value towards agent 2, with $K = 0{,}02$

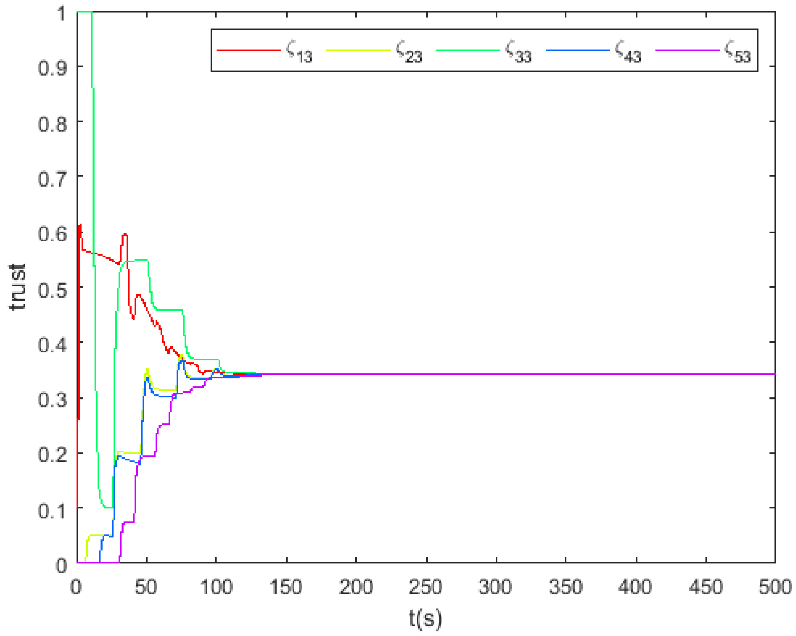

Figure 14: Trust value towards agent 3, with $K = 0{,}02$

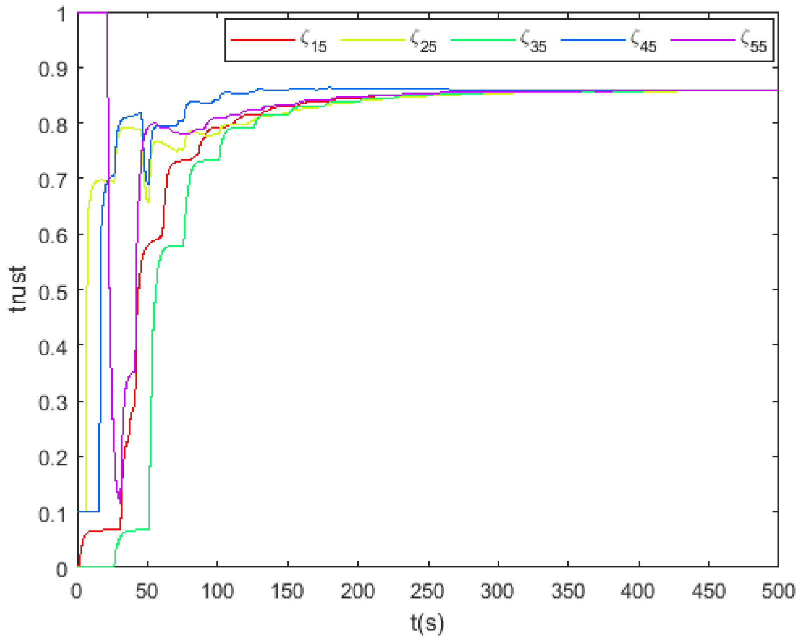

Figure 15: Trust value towards agent 5, with $K = 0{,}02$

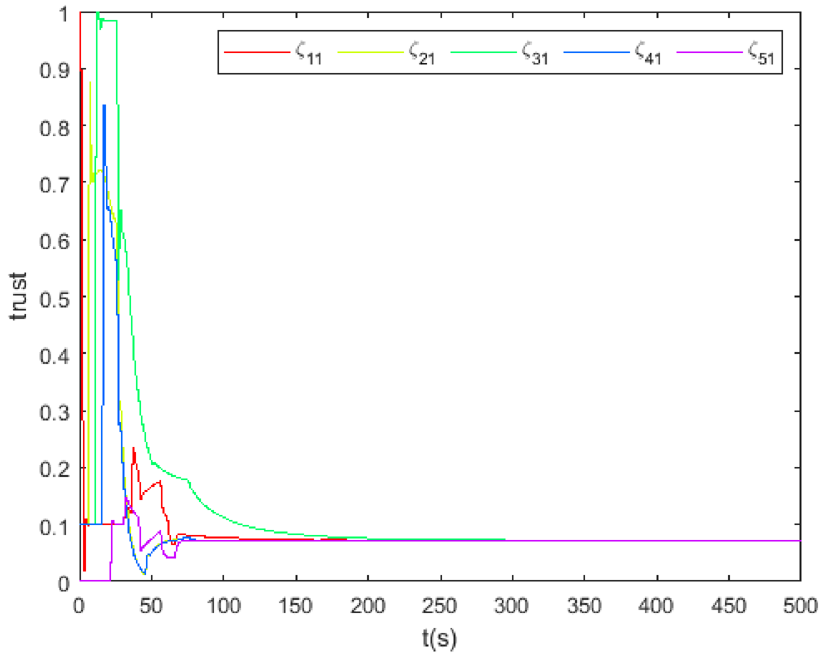

Figure 16: Trust value towards agent 1, with $K = 0{,}2$

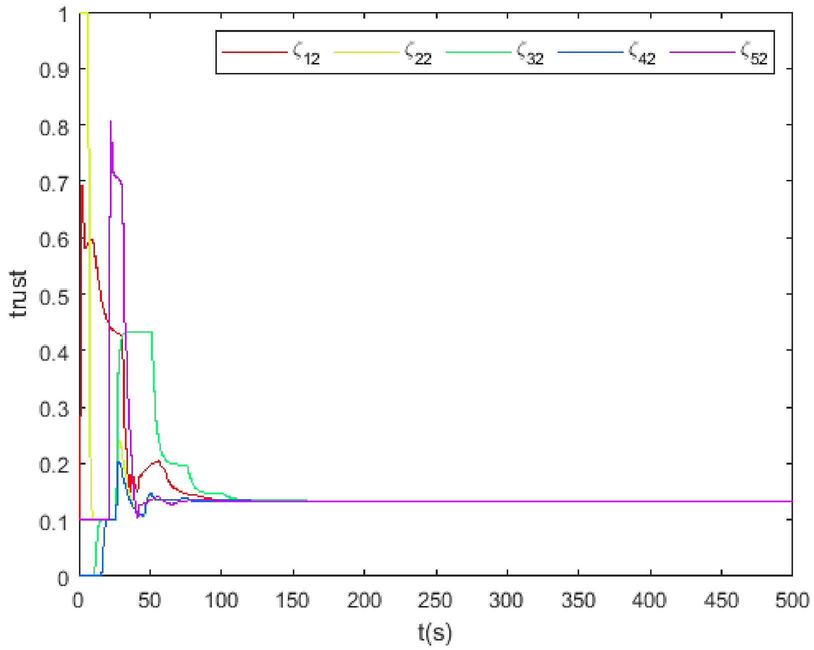

Figure 17: Trust value towards agent 2, with $K = 0{,}2$

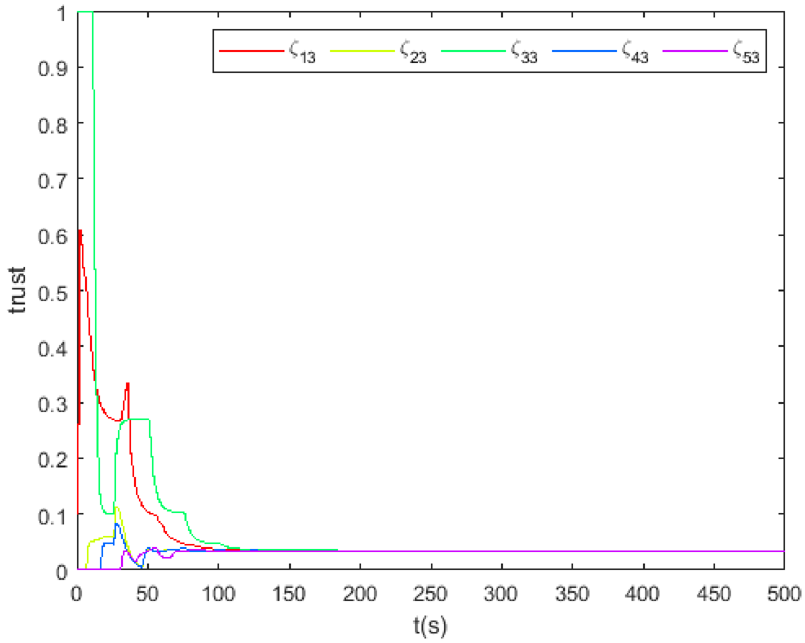

Figure 18: Trust value towards agent 3, with $K = 0{,}2$

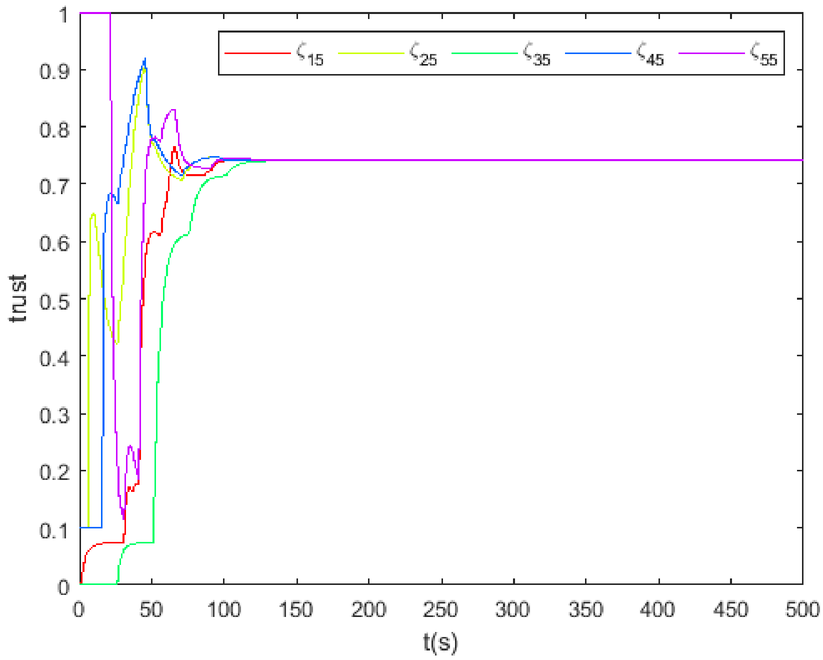

Figure 19: Trust value towards agent 5, with $K = 0{,}2$

APPENDIX 4: SIMULATION RESULTS FROM SECTION 6.1 - SCENARIO 3

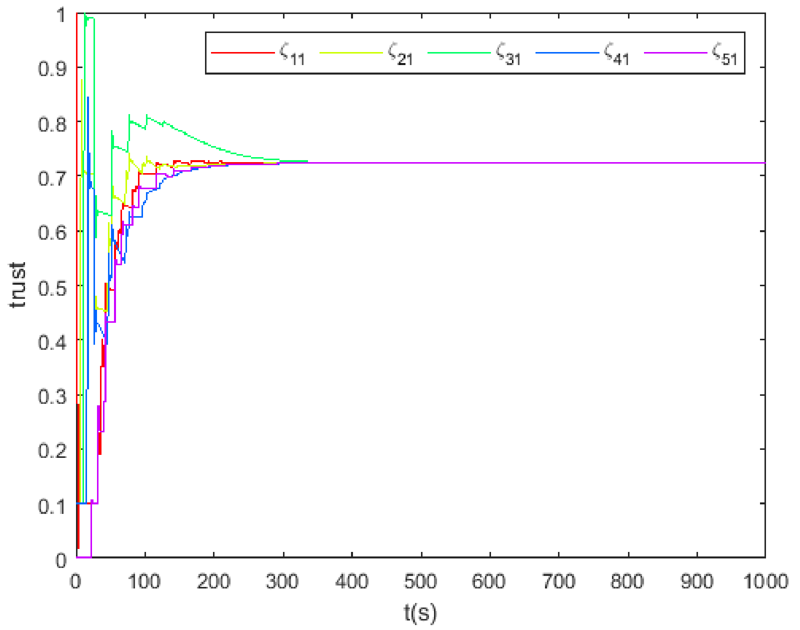

Figure 20: Trust value towards agent 1, with $T = 5$ s

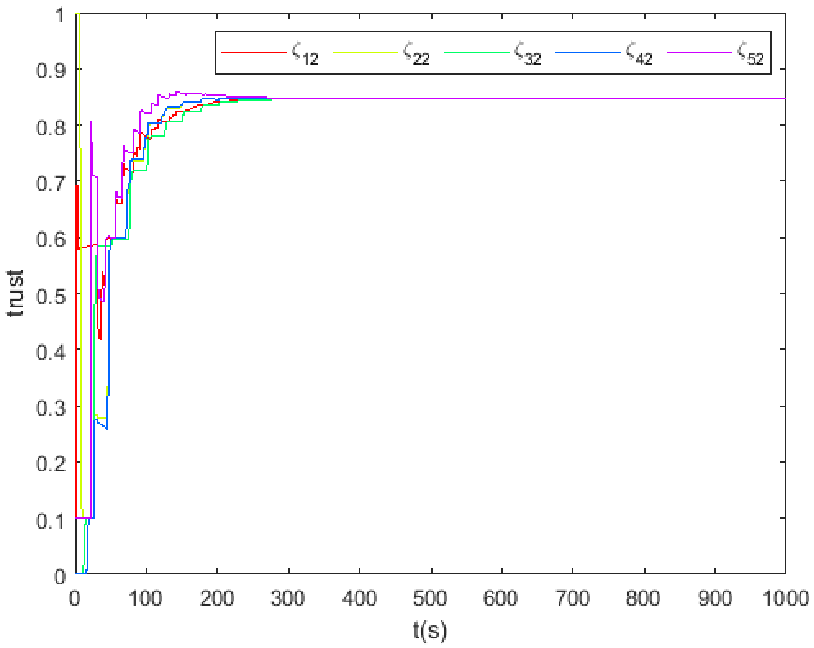

Figure 7: Trust value towards agent 2, with $T = 5$ s

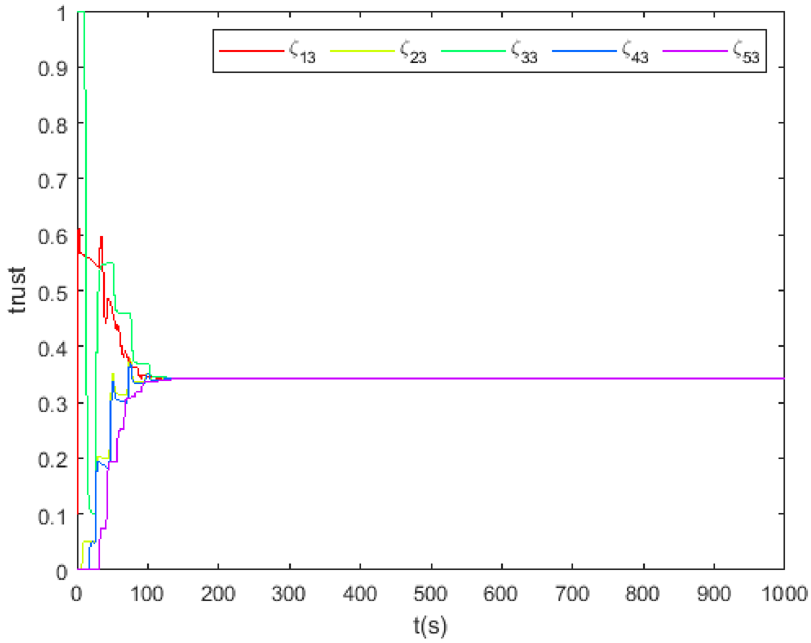

Figure 22: Trust value towards agent 3, with $T = 5$ s

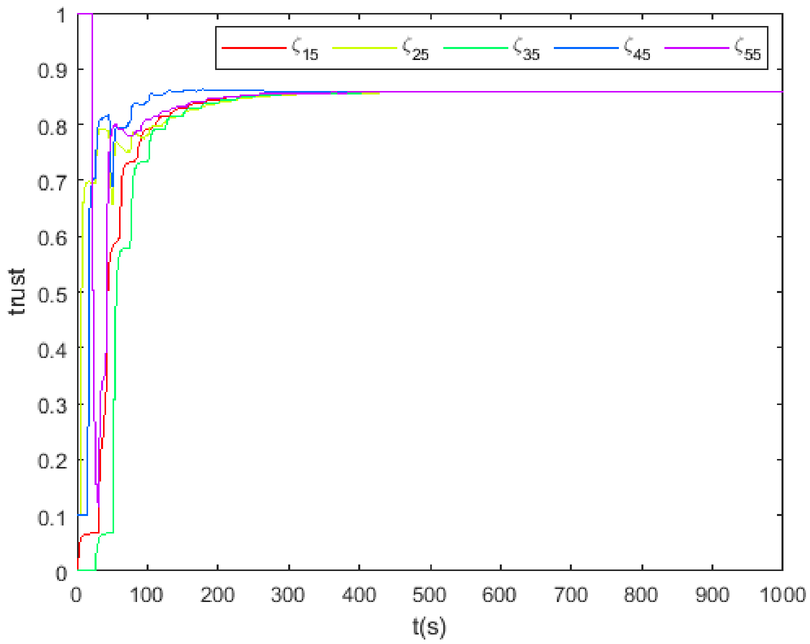

Figure 23: Trust value towards agent 5, with $T = 5$ s

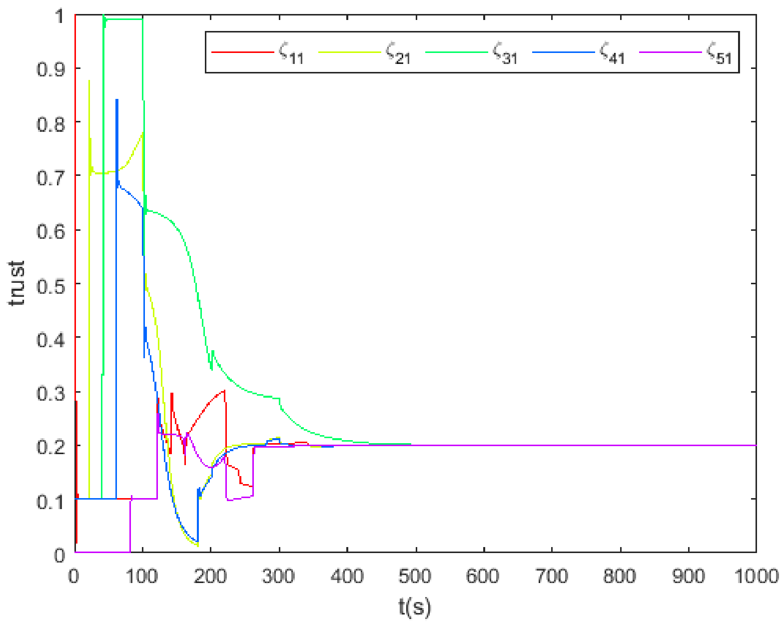

Figure 24: Trust value towards agent 1, with $T = 20$ s

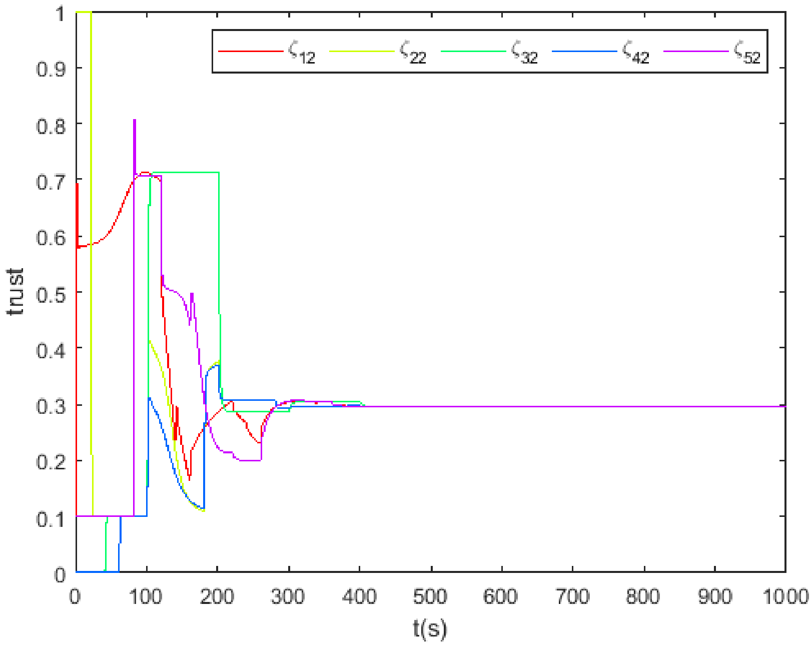

Figure 25: Trust value towards agent 2, with $T = 20$ s

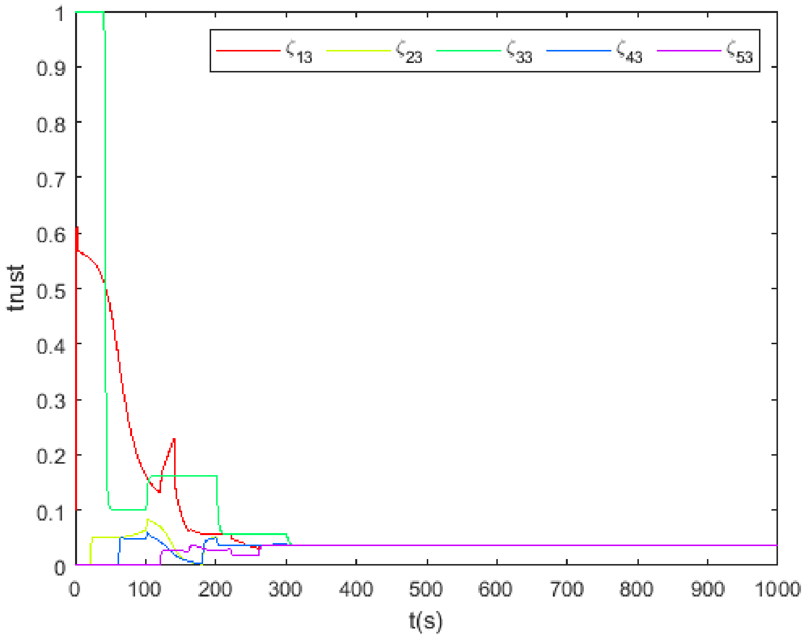

Figure 26: Trust value towards agent 3, with $T = 20$ s

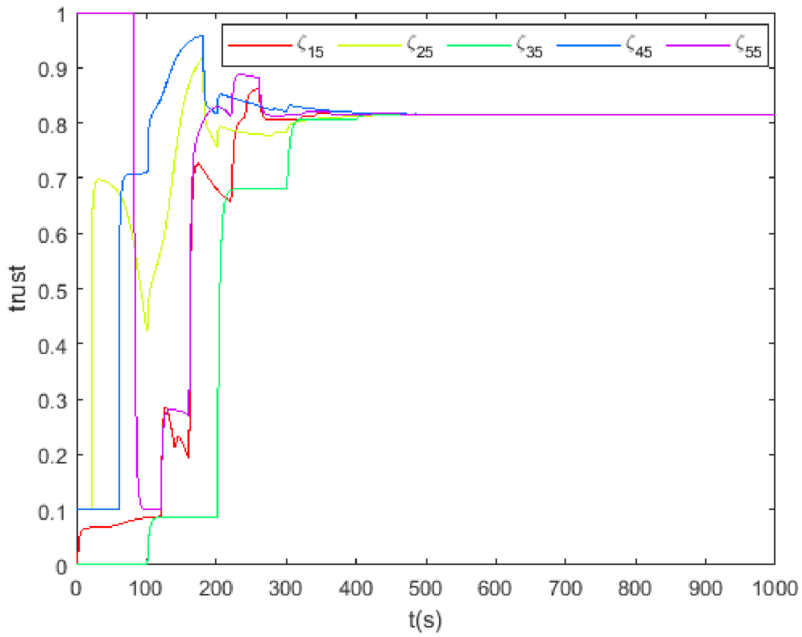

Figure 27: Trust value towards agent 5, with $T = 20$ s