Formation Control

of Nonholonomic Mobile Robots with Time Delay using Consensus Protocols

J.L.J. Peeters

D&C 2010.070

Report of Master traineeship

Supervisory committee:

prof.dr. H. Nijmeijer [1]
Dr. Eng. T. Oguchi [2]

[1] Eindhoven University of Technology
Department of Mechanical Engineering
Dynamics and Control Group

[2] Tokyo Metropolitan University
Department of Mechanical Engineering

Eindhoven, December 2010

Abstract

Over the years a significant amount of research has been done in the field of cooperative control for robotic systems. This is partly due to broad applications of multi-agent systems in many areas such as unmanned "aerial" vehicles, satellite clusters, automated highways, environmental monitoring, surveillance and mobile robots. In order for multi-agent systems to work together and execute prespecified tasks such as consensus and formation control, some form of cooperative control is required. In many applications groups of agents need to agree upon certain quantities of interests and reach consensus. The information flow between the agents plays a crucial role in order to achieve this. If there is a delay within the information flow between the mobile robots or if it is obstructed, the mobile robots will not cooperate with each other. In this project, consensus based algorithms and potential functions are used to derive a controller that will let a group of mobile robots converge and reach consensus. The main goal is to investigate the effect of time delay, applied on the communication network, on the formation control of the mobile robots. This is done by investigating the behavior of the mobile robots through simulations and experiments.

Contents

CHAPTER ONE

Introduction

1.1 Cooperative Control

Over the years a significant amount of research has been done in the field of cooperative control for robotic systems. This is partly due to broad applications of multi-agent systems in many areas such as unmanned "aerial" vehicles, satellite clusters, automated highways, environmental monitoring, surveillance and mobile robots. The goal of cooperative control is to develop a distributed control strategy for a group of agents such that the aggregated system attains a prespecified task including consensus, coverage, flocking or formation control [1], [2], [3].

In most previous work regarding cooperative control, agents are modeled as simple double integrators. This may be useful for fully actuated, unconstraint robotic systems but the agents that are used in the problems above typically have nonholonomic constraints restricting their instantaneous movement. Therefore the control laws that are derived for double integrator systems are generally not applicable in the case of systems with nonholonomic constraints.

In many applications groups of agents need to agree upon certain quantities of interests and reach consensus. In order to achieve consensus the information flow between the agents plays a crucial role. There are multiple studies that make use of the *graph Laplacian*, a special matrix used in algebraic graph theory [10], to achieve formation stabilization or to reach consensus for a group of agents [11], [12]. Using consensus based algorithms and potential functions it is possible to derive a controller that makes a group of mobile robots converge and reach consensus. As mentioned above, the communication network plays an important role in reaching consensus. If there is a delay within the information flow between the mobile robots or if it is obstructed, the mobile robots will not cooperate with each other.

The *Tokyo Metropolitan University* (*TMU*) in cooperation with *Eindhoven Technical University* (*TU/e*) already did a lot of research regarding control of mobile robots including time delay. In [4] remote tracking control of a mobile robot subject to a bilateral time-delay is investigated. In this case the delay affects the system since the controller and the robot are linked via a delay inducing communication channel. The bilateral time delay is compensated by means of a state estimator inspired on a predictor based on synchronization. In this paper a control scheme intended to control the mobile robot subject to a bilateral time delay was proposed and stability analysis was carried out. The proposed control strategies were validated by using two equivalent multi-robot platforms, one at the *TU/e* and the other at the *TMU*, that use the Internet as the communication channel. In [5] research was done on long distance master-slave and mutual synchronization of the mobile robots. In the case of master-slave synchronization, a master robot tracks its own reference, while its real trajectory constitutes the base for the reference of a slave robot while a constant time delay affects the system's unidirectional coupling. Here the control problem remains one of a tracking nature. In the case of mutual synchronization, all the robots receive a common signal known as the virtual center, which is used to define their individual reference trajectories. Additionally, all robots communicate their position and orientation errors to each other, forming a bidirectional coupling. There is a time delay in the robots' coupling induced by the communication channel linking them. A synchronization scheme was proposed for two mobile robots by using the virtual structure approach. For both synchronization methods experiments were done with the same experimental setup that was used in [4].

In [6] the effect of time-delays caused by the data transmission in a computer network on predictor-based tracking control of a mobile robot was investigated. A prediction control scheme using anticipating synchronization as a state predictor was extended for a nonholonomic system with input delay and saturation. The focus lies on the effect of time delays that arise in the system as a result of the use of communication networks. A control scheme was proposed that could compensate the negative effects of network induced delays on the system. A sufficient condition for the convergence of the prediction error was derived by using the Lyapunov-Razumikhin theorem and numerical simulation were done to illustrate the effectiveness of the proposed scheme and the derived condition.

In [7] tracking/consensus control laws are proposed to solve a problem where multiple unicycle agents are required to follow individual reference trajectories while maintaining a time-varying formation. The motions of the agents are coordinated by introducing coupling terms in their feedback control laws. The couplings are based on coordination errors that are defined as differential position errors between every pair of the interacting agents. Under saturation constraints on the control signals, this approach guarantees global asymptotic stability of the tracking error dynamics. The proposed synchronization approach is validated in experiments. In [8] a collision-free tracking control strategy is proposed for a group of unicycle mobile robots. A supervisory system assigns to each robot its reference path, together with the desired velocity profile as a function of the position along this path. The robot paths and velocity profiles do not necessarily prevent collisions of the robots. The reference velocities of individual robots are coordinated in real-time based on the actual robot coordinates to ensure collision-free movements. Feedback controllers of the individual robots realize globally asymptotic stable tracking of the reference trajectories under constraints on the actuator inputs. In [9] formation control of nonholonomic mobile robots was investigated. A known tracking controller was adapted to enforce mutual coupling between the robots. Formation control is achieved by using the master-slave approach and virtual structure approach and both methods are discussed. An experimental setup was designed to validate both methods.

1.2 Project goals

The main goal of this project is to investigate the effect of time delay, applied on the communication network, on the formation control of nonholonomic mobile robots. In order to investigate this effect a controller has to be designed which controls a group of nonholonomic mobile robots. With this controller, simulation and experiments have to be done to investigate the behavior of the mobile robots. A time delay will be applied on the communication network and the effect it has on the behavior of the mobile robots will be investigated by simulations and experiments.

An experimental setup was available at the Tokyo Metropolitan University, this made it possible to do experiments with mobile robots. At the end of the internship period this setup was completed and experiments regarding formation control could be done.

1.3 Organisation of this report

The layout of the report is a representation of all the steps that were done during the project. In chapter 2 relevant theory that is used for deriving the controller is explained. This chapter will also give some more insight in some of the methods that are used for formation control. In chapter 3 a controller is derived in order to reach consensus for any number of mobile robots. A *Simulink* model is created from this controller and simulation results will be discussed. This is done to investigate if the mobile robots indeed converge and reach consensus by using the derived controller. In chapter 4 the effect of time delay on the system will be investigated by modifying the derived controller from chapter 3. For two time delay cases a new *Simulink* model is created and simulations are done to determine the effect of weights on the communication links between the mobile robots and Time Delay.

The simulations from the previous chapters are repeated for three mobile robots in chapter 5 by doing experiments. First the experimental setup and mobile robots that are used will be explained in more detail. The remainder of chapter 5 is used to compare the experimental results with the theory and simulations from the previous chapters. Finally the conclusions and recommendations are given in chapter 6.

Chapter two

Robot Kinematics and Consensus in Communication Networks

This chapter discusses necessary and relevant theory that was used during research. First of all the kinematics of the mobile robots have to be described in order to derive a controller. Next some theoretical definitions about the communication network between the mobile robots will be given. The control laws that are derived in the next chapter use consensus protocols and the *Laplacian potential* function. In section 2.3 consensus protocols and the Laplacian potential will be treated in more detail. In section 2.4 the effect of time delay is introduced.

2.1 Kinematics of the Unicycle Mobile Robot

The mobile robots are nonholonomic *unicycles*, kinematic systems with two driven wheels in the middle. A graphical representation of the ith nonholonomic mobile robot can be seen in Figure 2.1. The position of the center of the ith mobile robot is described with coordinates x_i and y_i with respect to the fixed coordinate frame $\underline{\vec{e}}^{\,0} := [\vec{e}_1^{\,0} \;\; \vec{e}_2^{\,0}]^T$. The heading of the ith mobile robot with respect to the x-axis of the fixed coordinate frame $\underline{\vec{e}}^{\,0}$ is given by the angle θ_i [9].

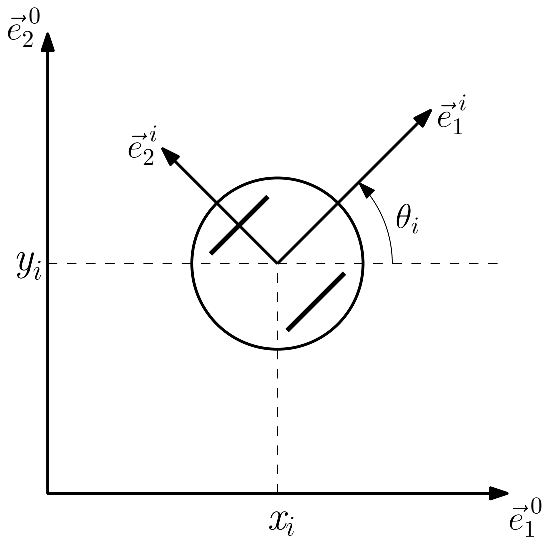


Figure 2.1 / Model of the ith nonholonomic unicycle mobile robot

The following differential equations describe the kinematics of the ith nonholonomic mobile robot:

$$\begin{aligned} \dot{x}_i &= u_i \cos\theta_i, \\ \dot{y}_i &= u_i \sin\theta_i, \\ \dot{\theta}_i &= \omega_i. \end{aligned} \tag{2.1}$$

In (2.1) u_i and ω_i are considered the control inputs of the system and denote the forward and angular velocity, respectively.

2.2 Communication Network

In order for the mobile robots to work together as a whole, some sort of communication network is needed. One way of representing a communication network is by using graphs. The advantage of this method is that it gives a complete overview of the entire system. This way the number of agents and their relations between one another can easily be distinguished. In this section the *algebraic graph theory* [10] and the information that can be derived from it will be explained in more detail.

2.2.1 Algebraic Graph Theory

The communication network is modeled through a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ defined by a set of vertices $\mathcal{V}$ and a set of edges $\mathcal{E}$. Each node is labeled by $v_i \in \mathcal{V}$, or $i \in \mathcal{I} := \{1, ..., n\}$ and each edge is denoted by $e = (v_i, v_j)$ or $e = ij$. In this case every node represents an agent and an edge (i, j) means that agents i and j can communicate with each other. Nodes that are linked by an edge are called *neighbors*. The neighboring relation is denoted with $i \sim j$ and it is assumed that $i \sim i$ always holds [1]. The communication structure is *bi-directional coupled* during simulations and experiments, this means that $(i, j) \in \mathcal{E} \Leftrightarrow (j, i) \in \mathcal{E}$. Also graph $\mathcal{G}$ is assumed to be *connected*, i.e. there is a path from any node i to any other node j. Weights on the communication links are defined by a weighting function w with properties:

$$\begin{aligned} &w(i, j) = w(j, i) \\ &w(i, j) > 0 \Leftrightarrow i \sim j. \end{aligned} \tag{2.2}$$

In Figure 2.2 a graph of the communication network between three agents can be seen. In addition there is a time delay variable that acts on the communication links. The effect of time delay will be discussed in section 2.4.

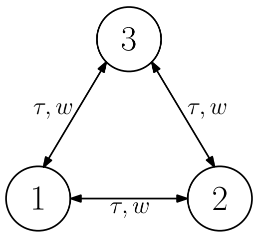


Figure 2.2 / Undirected connected graph $\mathcal{G}$.

2.2.2 Laplacian Matrix

The graph that describes the communication network can also be presented in a matrix form called the *Laplacian matrix*. The Laplacian matrix is used to find many properties of the graph and can be used as a basis for distributed consensus dynamics. Given a graph $\mathcal{G}$ the Laplacian matrix is defined in terms of the *degree matrix* Δ and the *adjacency matrix* $A(\mathcal{G})$, [1], [2], [11]. The degree of node v_i is the number of its neighbors and denoted as $deg(v_i)$. The degree matrix is an $N \times N$ matrix defined as $\Delta = \Delta(\mathcal{G}) = \{\Delta_{ij}\}$ and:

$$\Delta_{ij} := \begin{cases} deg(v_i) & ,i = j \\ 0 & ,i \neq j. \end{cases} \tag{2.3}$$

The adjacency matrix of a graph $\mathcal{G}$ with weights $w(i, j)$ is an $N \times N$ matrix with entries:

$$(A(\mathcal{G}))_{i,j} := \begin{cases} \omega(i, j) & \text{if } i \sim j \text{ and } i \neq j, \\ 0 & \text{otherwise.} \end{cases} \tag{2.4}$$

From the degree matrix (2.3) and the adjacency matrix (2.4) the Laplacian matrix of graph $\mathcal{G}$ is defined as:

$$L(\mathcal{G}) := \Delta(\mathcal{G}) - A(\mathcal{G}). \tag{2.5}$$

An important feature of the Laplacian is that all the row sums of L are zero and thus $e_0 = (1, 1, ..., 1)^T \in \mathbb{R}^n$ is an eigenvector of L associated with the eigenvalue $\lambda = 0$ [11]. As a result of (2.2) the Laplacian of a graph $\mathcal{G}$ is a symmetric positive-semidefinite matrix.

2.3 Consensus

Let $p_i \in \mathbb{R}$ denote the value of node v_i. The value of a node v_i might represent physical quantities including attitude, position, orientation, voltage, velocity, and so on. It is said that nodes v_i and v_j *agree* in a network if and only if $p_i = p_j$, otherwise they disagree. In the case of formation control the nodes will have to agree upon a common position, more precisely when they are aligned at a certain position with respect to each other. Consensus is obtained in a network when all the nodes agree on a value of a quantity of interest, in other words when $p_i = p_j$ for all $i, j \in \mathcal{I}, i \neq j$ [12].

In case the agents act as integrators the Laplacian can be used to reach consensus by the following differential equation:

$$\dot{\mathbf{p}} = -L\mathbf{p}. \tag{2.6}$$

For an individual agent the following dynamics hold:

$$\dot{p}_i = -\sum_{j \in \mathcal{N}_i} w_{ij}(p_i - p_j). \tag{2.7}$$

Here $\mathcal{N}_i$ denotes the set of neighbors agent i can communicate with and w_{ij} are the weights on the communication links. As mentioned in section 2.2.2 the row sums of the Laplacian are zero, $\sum_j L_{ij} = 0$. This implies that for any vector $\mathbf{p}$ with $p_i = p_j$ for all i and j, we have that (2.6) becomes $Lp = 0$. Thus, any consensus is an equilibrium, [2], [11]. There exists a derivation of the Laplacian which is called the *Laplacian potential* [12] and is a measure of total disagreement among all nodes:

$$\Phi_{\mathcal{G}}(p) = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij}(p_j - p_i)^2 \tag{2.8}$$

The Laplacian potential of a graph is also positive semi-definite and given a connected graph, $\Phi_{\mathcal{G}}(p) = 0$ if and only if $p_i = p_j$ for all i and j. If at least two neighboring nodes of $\mathcal{G}$ disagree then $\Phi_{\mathcal{G}}(p) > 0$, thus minimizing $\Phi_{\mathcal{G}}(p)$ is equivalent to reaching a consensus.

In case of integrator dynamics this results in the following dynamics for agent i:

$$\begin{aligned} \dot{p}_i &= u_i, \\ \dot{p}_i &= -\nabla\Phi_{\mathcal{G}}(p) = -\sum_{j\in\mathcal{N}_i} w_{ij}(p_i - p_j). \end{aligned} \tag{2.9}$$

Here the gradient of (2.8) is used as a gradient-based feedback and equals $u = -\nabla\Phi_{\mathcal{G}}(p)$.

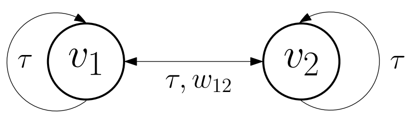
2.4 Time Delay

The main topic is the effect of time delay on control of a group of mobile robots. In this section two time delay cases that are investigated will be explained in more detail. In the experiment setup that is used, various time delays are present in the hardware and software components. These time delays will be discussed in section 5.2.1 but are not the topic of this research.

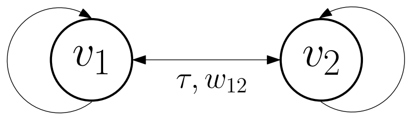
The mobile robots communicate with each other by transmitting data via *BlueTooth*, therefore reaching consensus is greatly influenced by time delay. As mentioned there are various time delays present in experiment setup. In order to investigate the effect of time delay, an additional time delay will be applied on the communication network. More specific, two different time delay cases are implemented in the simulations and experiments and are investigated. These two time delay cases are explained below for two mobile robots:

Time Delay case 1 In this case time delay τ acts on the total system. Meaning that the data mobile robot 1 sends to mobile robot 2 has a delay τ. Also the information that robot 1 uses to update its own position has the same delay. The same holds for mobile robot 2 and for any number of mobile robots. A graphical representation of this case can be seen in Figure 2.3(a). Here the two mobile robots are depicted by v_1 and v_2. The time delay is denoted by τ and the weight on the communication link by w_{12}.

Time Delay case 2 In this case time delay τ only acts on the communication link between the two mobile robots. The data that the mobile robots use to update their own positions is not influenced by time delay. A similar graphical representation as case 1 can be seen for this case in Figure 2.3(b).



(a) Case 1



(b) Case 2

Figure 2.3 / Both time delay cases for two mobile robots.

In chapter 4 these two time delay cases are implemented in a *Simulink* model and the dynamics of the system with time delay are given. Not only is the effect of time delay on the system investigated but also the relation between weights on the communication links and time delay.

Chapter three

Consensus Control for the Unicycle Mobile Robot

In this chapter the previous explained theory will be utilized to derive a controller for a group of nonholonomic unicycle mobile robots. In section 3.2 the *Simulink* model that was created will be explained. The simulations and results will be discussed in section 3.3.

3.1 Controller Design

Consider a system of n mobile robots with random orientations at arbitrary positions in a plane. The mobile robots can exchange information and will send their positions to each other. In this case the mobile robots will have to consent upon a common point and gather there. Using the kinematics of the mobile robot in combination with consensus algorithms and potential functions a controller is derived that will lead to consensus. Please note that the orientations and respective derivatives of the mobile robots all depend on time.

In [3] a controller is derived for unicycle mobile robots, based on consensus algorithms and potential functions to achieve rendezvous of a group of mobile robots. Furthermore stability analysis was done in [3] that determined consensus could be reached. In this section the controller design is explained and during the project this controller is used to investigate the effect of time delay on the system. In order to make the headings of the agents point to each other a consensus based control law for the angular velocity input of each vehicle is created. To let the heading of agent i point into the direction of the neighbor set $\mathcal{N}_i$ which it can communicate with is done by:

$$\phi_{ij} = arctan(\frac{y_j - y_i}{x_j - x_i}). \tag{3.1}$$

Using the following equation the headings of the agents will converge:

$$\omega_i = -\sum_{j \in \mathcal{N}_i} (\theta_i - \phi_{ij}). \tag{3.2}$$

(3.2) is similar to the Laplacian from (2.7). An attractive potential function (3.3) [12] is introduced which makes the agents converge towards one point until consensus is reached. For simulation purposes this is not a problem, however during experiments an offset is implemented to ensure that the mobile robots do not collide at the consensus point.

$$\begin{aligned} g_a(\|\mathbf{d}_i\|) &= \tfrac{1}{2}\|\mathbf{d}_i\|^2 = \tfrac{1}{2}\mathbf{d}_i^\top \mathbf{d}_i, \\ \text{with } \mathbf{d}_i &= \begin{pmatrix} x_i - x_j \\ y_i - y_j \end{pmatrix}. \end{aligned} \tag{3.3}$$

Using the following controller for each agent with kinematics given in (2.1) is an option to reach consensus:

$$\begin{aligned} u_i &= -\textstyle\sum_{j \in \mathcal{N}_i} w_{ij} \nabla g_a^\top \vec{e}_1^{\,i} = -\sum_{j \in \mathcal{N}_i} w_{ij}[(x_i - x_j)\cos\theta_i + (y_i - y_j)\sin\theta_i], \\ \omega_i &= -\textstyle\sum_{j \in \mathcal{N}_i} w_{ij}(\theta_i - \phi_{ij}) = -\sum_{j \in \mathcal{N}_i} w_{ij}(\theta_i - arctan(\frac{y_j - y_i}{x_j - x_i})). \end{aligned} \tag{3.4}$$

Here $\vec{e}_1^{\,i}$ denotes the unit heading vector of agent i as can be seen in Figure 2.1 and w_{ij} are weights that act on the set of edges $\mathcal{E}$. Operator ∇ computes the gradient of the function (see the Laplacian potential from (2.8)), hence the *transpose* of the attractive potential function. Using (3.4) as controller leads to the following dynamic system:

$$\begin{aligned} \dot{x}_i &= -\textstyle\sum_{j \in \mathcal{N}_i} w_{ij}[(x_i - x_j)\cos\theta_i + (y_i - y_j)\sin\theta_i]\cos\theta_i, \\ \dot{y}_i &= -\textstyle\sum_{j \in \mathcal{N}_i} w_{ij}[(x_i - x_j)\cos\theta_i + (y_i - y_j)\sin\theta_i]\sin\theta_i, \\ \dot{\theta}_i &= -\textstyle\sum_{j \in \mathcal{N}_i} w_{ij}(\theta_i - arctan(\frac{y_j - y_i}{x_j - x_i})). \end{aligned} \tag{3.5}$$

One can easily deduce from 3.5 that if the positions of the mobile robots are the same ($x_i = x_j$ and $y_i = y_j$), that the velocity input u_i is zero. Meaning that the mobile robots velocity will decrease when they approach each other. For angular velocity input w_i this means that $\theta_i = 0$. The headings of the mobile robots should thus consent and converge to zero when they reach consensus.

3.2 Simulation Model

A *Simulink* model is created from the total system that was derived in section 3.1. In this model the weights w_{ij} can be changed and *Time Delay* is not present. Using this model simulations are carried out to investigate the behavior of the agents using the controller proposed in (3.4).

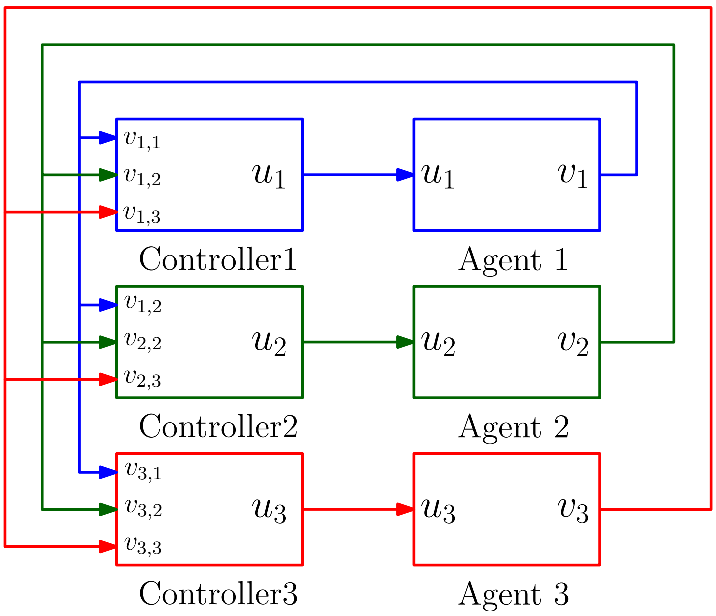


Figure 3.1 / Model consisting of three mobile robots without Time Delay.

In Figure 3.1 a schematic representation of the *Simulink* model for three mobile robots can be seen. Each mobile robot sends its position to itself and all other mobile robots. This means that the communication between the mobile robots is *bi-directional*. The controller then computes the new input variables u_i and ω_i using (3.4) which results in a new position for all the mobile robots. The main purpose of this model is to investigate the behavior of the mobile robots for various initial conditions. Also the effect of changing the weights on the communication links is examined. However, since there is no mathematical proof at this moment it is not certain that the proposed controller will lead to consensus. An M-file is created in *Matlab* that runs the simulation and allows the user to specify w_{ij} and initial conditions. These initial conditions are the orientations of each mobile robot with respect to the fixed coordinate frame.

The simulation parameters are unchanged throughout all the simulations and can be seen in Tabel 3.1. The simulation time is based on test simulations and is sufficient for the system to reach consensus. In order to use the same solver that was used in previous works at the *TMU* a fixed step size was used. The step size is equal to 0.001 seconds to achieve accurate simulation results.

Simulation Parameters		
Simulation time	300	[s]
Fixed step size	0.001	[s]
Solver	ode4 *Runga-Kutta*	[-]

Table 3.1 / *Simulink* model parameters

In order to check wether or not consensus is reached within the simulation time the *Euclidian Norm* is used as a criterion. The end positions of each mobile robot are computed and subtracted from each other via:

$$e_{i,j} = [x_i(t_{end}) - x_j(t_{end}) \;\; y_i(t_{end}) - y_j(t_{end})]. \tag{3.6}$$

Here $e_{i,j}$ is the difference in coordinates between mobile robot i and j. This results in the error coordinates for each mobile robot and gives already an indication for consensus. Using these error coordinates the Euclidian norm of the end error is calculated by:

$$\|\mathbf{e_{i,j}}\| = \sqrt{e(1)^2_{i,j} + e(2)^2_{i,j}}. \tag{3.7}$$

Since the mobile robots are bi-directional coupled and the weight on a communication link is $w(i,j) = w(j,i)$ it holds that $\|\mathbf{e_{i,j}}\| = \|\mathbf{e_{j,i}}\|$. In case of three mobile robots the criterion for each mobile robot is now defined by:

$$\begin{aligned} \|\mathbf{e_1}\| &= \|\mathbf{e_{1,2}}\| + \|\mathbf{e_{1,3}}\|, \\ \|\mathbf{e_2}\| &= \|\mathbf{e_{2,1}}\| + \|\mathbf{e_{2,3}}\|, \\ \|\mathbf{e_3}\| &= \|\mathbf{e_{3,1}}\| + \|\mathbf{e_{3,2}}\|. \end{aligned} \tag{3.8}$$

From simulation tests a constant was determined that weights the norm. In case $\|\mathbf{e_k}\| > 0.01$ for $k = 1, 2, ..., n$ the mobile robots have not reached consensus. This means that either the mobile robots did not converge fast enough to reach consensus within the simulation time or that they did not converge at all. In this case the simulation time is sufficient to reach consensus, thus meaning that for $\|\mathbf{e_k}\| > 0.01$ the agents diverge and consensus is not reached. This was deduced by looking at the simulation results for various initial conditions. In case $\|\mathbf{e_k}\| > 0.01$ the difference in end positions of the mobile robots is larger than approximately 0.007 $[m]$.

3.3 Simulations and Results

Using the *Simulink* model, explained in the previous section, a number of simulations are made for three mobile robots. The initial conditions of the mobile robots for these simulations are in SI-units and defined as:

$$\mathbf{x_{init}} = \begin{bmatrix} x0_1 & x0_2 & x0_3 \\ y0_1 & y0_2 & y0_3 \\ \theta0_1 & \theta0_2 & \theta0_3 \end{bmatrix} = \begin{bmatrix} -0.5 & 0.5 & 0 \\ -0.5 & -0.5 & 0.5 \\ \pi/2 & \frac{\pi}{2} & \frac{-\pi}{2} \end{bmatrix}. \tag{3.9}$$

The weights w_{ij} on the communication links are constant and equal to one. In Figure 3.2 the simulation results can be seen.

From Figures 3.2(b) and 3.2(c) one can easily see that the positions of the mobile robots indeed agree upon one common value. Looking at Figure 3.2(a) it can be seen that the consensus point is not equal to the average of the initial conditions. If the initial conditions are unchanged the consensus point will stay the same. However, even a small variation in the initial conditions will lead to a different consensus point. This will make it hard in practical situations to predict where consensus will be reached. The most remarkable result can be seen in Figure 3.2(d), here θ does not match the theoretical derived assumptions from (3.5). Depending on the choice of initial conditions one can distinguish basically two different behaviors: namely that θ is constant or is oscillating. Various combinations of these two behaviors occur given different initial conditions.

The end faults that are obtained from the simulation are given below and indeed indicate that consensus is reached:

$$\begin{aligned} \|\mathbf{e_1}\| &= 0.136 \times 10^{-14} \ [m], \\ \|\mathbf{e_2}\| &= 0.272 \times 10^{-14} \ [m], \\ \|\mathbf{e_3}\| &= 0.136 \times 10^{-14} \ [m]. \end{aligned}$$

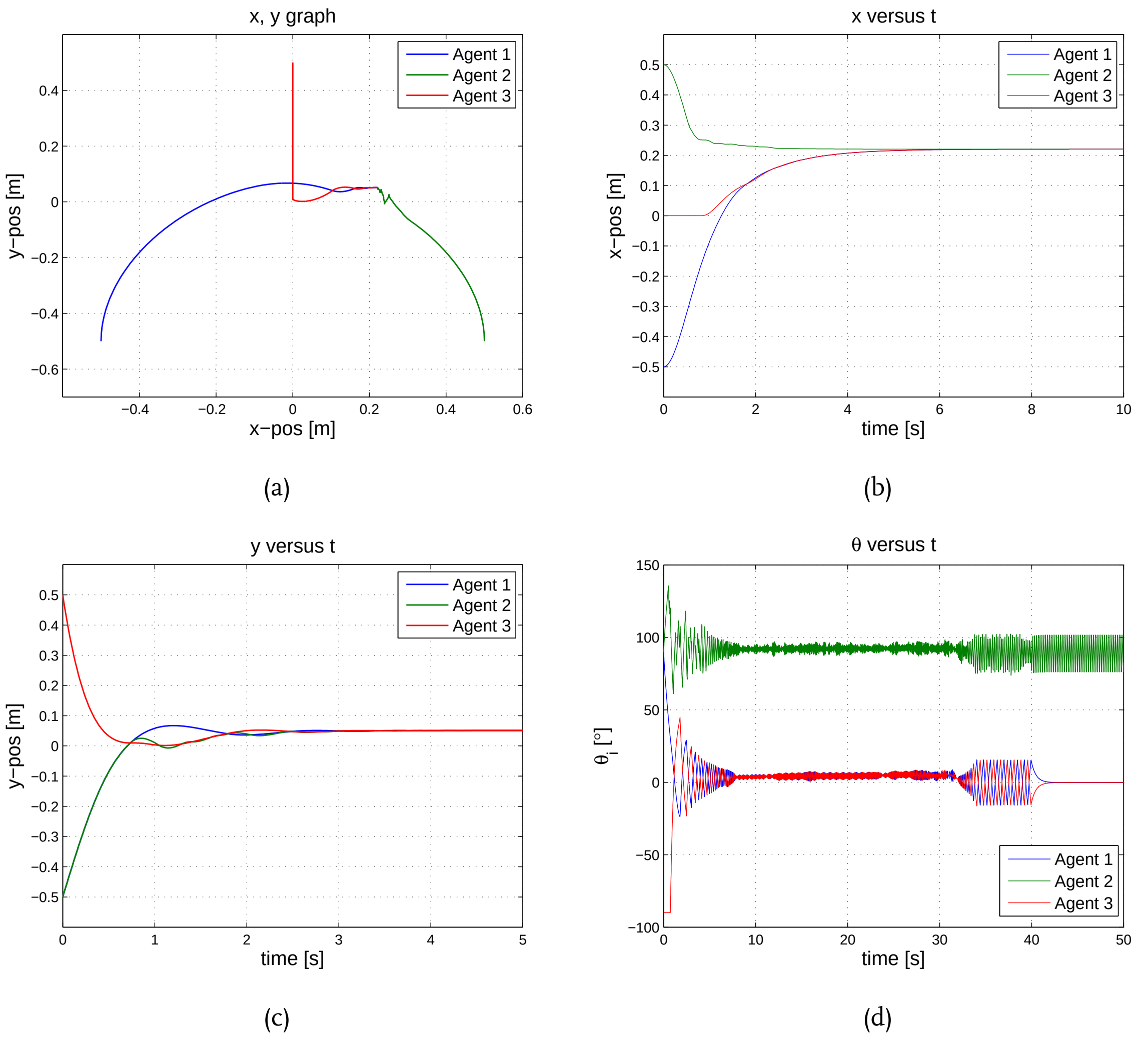


Figure 3.2 / Simulation results for three agents with $w_{ij} = 1$.

Next the effect of weights w_{ij} on the communication links will be investigated. The initial conditions from (3.9) will be used in order to compare the results. For this simulation the weight w_{12} on the communication link between mobile robot 1 and 2 is set to 7 and the other two weights are equal to 1 to exaggerate the effect. The results can be seen in Figure 3.3.

Using a higher weight on the communication links has both an advantage and disadvantage. The time that is needed to reach consensus has decreased, as can be seen in Figure 3.3(b) and 3.3(c). However, the drawback is that the mobile robots 1 and 2 involving w_{12} will react more towards each other resulting in overshoot near the consensus point. Looking close near the consensus point in Figure 3.4 this can be seen in more detail. It can be seen that only mobile robot 1 and 2 are affected by this weight and leads into a different consensus point compared to Figure 3.2(a).

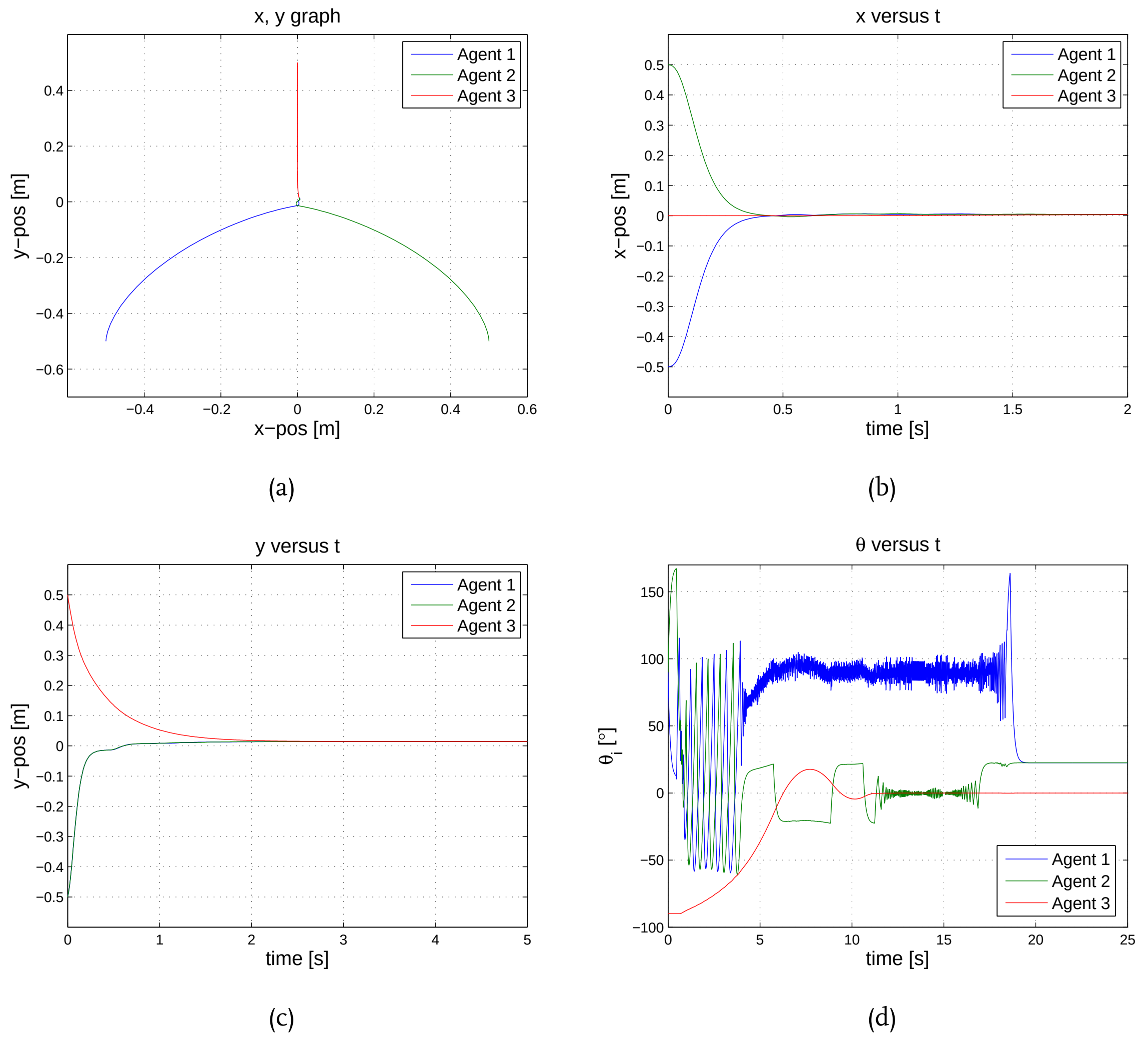


Figure 3.3 / Simulation results of three agents with $w_{12} = 7$ and $w_{23} = w_{31} = 1$.

3.4 Discussion

The behavior of the mobile robots was investigated by simulations that were made with the controller of (3.4). From the results it can be concluded that the mobile robots indeed converge to each other and eventually reach consensus, regardless of the initial conditions. It also followed that the consensus point changes if the initial conditions are changed. By changing the weights w_{ij} on the communication links the behavior of the system is also influenced. From the simulations it follows that the mobile robots, involving the communication weight, will react more towards each other. This results in overshoot near the consensus point but decreases the time it takes for the mobile robots to reach consensus.

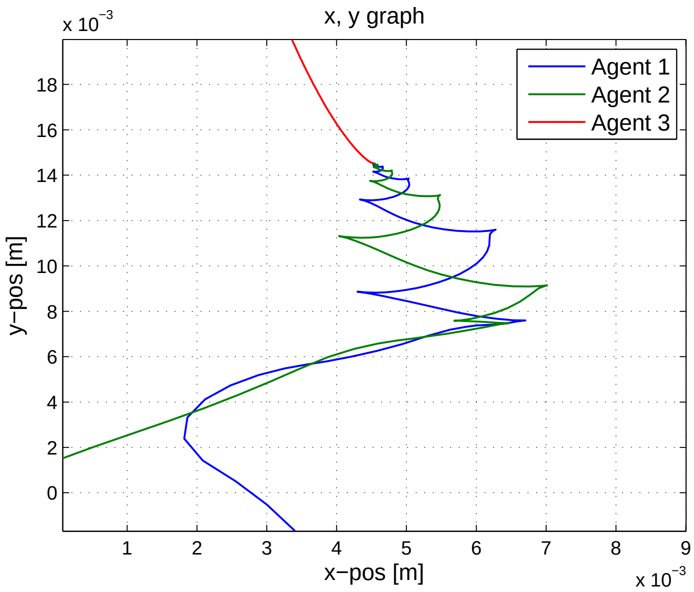


Figure 3.4 / Zoomed x versus y graph from Figure 3.3(a).

CHAPTER FOUR

Consensus Control for the Unicycle Mobile Robot with Time Delay

As mentioned in section 2.4 two time delay cases are investigated in the course of this research. In this chapter equations of the full system with time delay are derived. Using *Simulink* not only the effect of time delay on consensus for both cases is investigated, but also the relationship between weights on the communication links and both cases.

4.1 Time Delay case 1

In this section time delay that acts on the entire system is implemented on the dynamic system from (3.5). As can be seen, time delay τ is present in every term of the control inputs u_i and ω_i:

$$\begin{aligned}
\dot{x}_i(t) &= -\textstyle\sum_{j\in\mathcal{N}_i} w_{ij}[(x_i(t-\tau)-x_j(t-\tau))\cos\theta_i(t-\tau) \\
&+ (y_i(t-\tau)-y_j(t-\tau))\sin\theta_i(t-\tau)]\cos\theta_i(t), \\
\\
\dot{y}_i(t) &= -\textstyle\sum_{j\in\mathcal{N}_i} w_{ij}[(x_i(t-\tau)-x_j(t-\tau))\cos\theta_i(t-\tau) \\
&+ (y_i(t-\tau)-y_j(t-\tau))\sin\theta_i(t-\tau)]\sin\theta_i(t), \\
\\
\dot{\theta}_i(t) &= -\textstyle\sum_{j\in\mathcal{N}_i} w_{ij}(\theta_i(t-\tau)-arctan(\tfrac{y_j(t-\tau)-y_i(t-\tau)}{x_j(t-\tau)-x_i(t-\tau)})).
\end{aligned} \tag{4.1}$$

Time Delay system 1 is transformed into a *Simulink* model which schematic representation can be seen in Figure 4.1. The position and orientation parameters that are transmitted by all mobile robots undergo a time delay τ. As mentioned, this can be scaled up to any number of mobile robots.

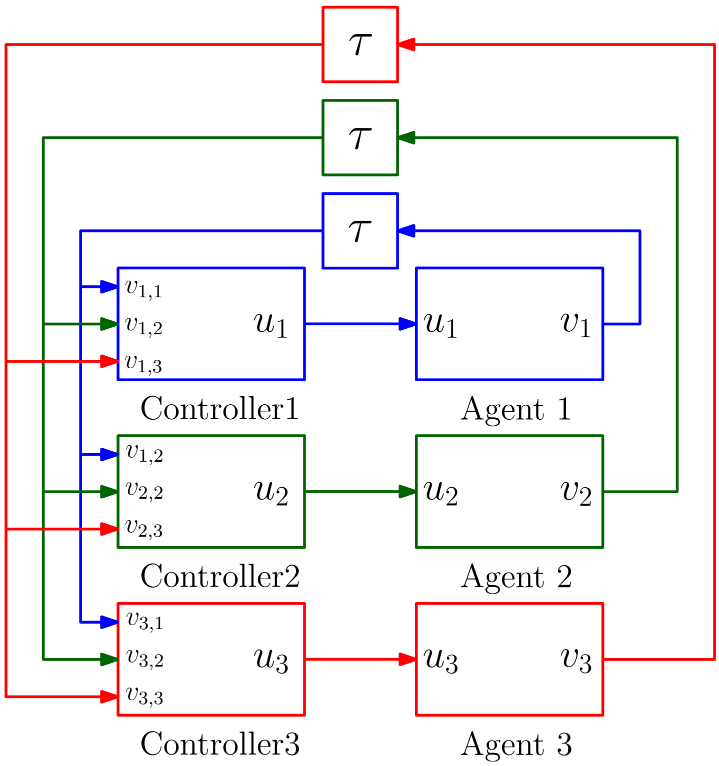


Figure 4.1 / Model of Time Delay case 1

4.2 Time Delay case 2

In case the time delay only acts on the communication links between the mobile robots, the dynamic system changes into:

$$
\begin{aligned}
\dot{x}_i(t) &= -\sum_{j\in\mathcal{N}_i} w_{ij}[(x_i(t)-x_j(t-\tau))\cos\theta_i(t)+(y_i(t)-y_j(t-\tau))\sin\theta_i(t)]\cos\theta_i(t), \\
\dot{y}_i(t) &= -\sum_{j\in\mathcal{N}_i} w_{ij}[(x_i(t)-x_j(t-\tau))\cos\theta_i(t)+(y_i(t)-y_j(t-\tau))\sin\theta_i(t)]\sin\theta_i(t), \\
\dot{\theta}_i(t) &= -\sum_{j\in\mathcal{N}_i} w_{ij}(\theta_i(t)-arctan(\tfrac{y_j(t-\tau)-y_i(t)}{x_j(t-\tau)-x_i(t)})).
\end{aligned}
\tag{4.2}
$$

Here mobile robot i receives the delayed position and orientation of mobile robot j.

A schematic representation of the *Simulink* model for *Time Delay system 2* can be seen in Figure 4.2. Here the position and orientation parameters that are transmitted by other mobile robots undergo a time delay τ.

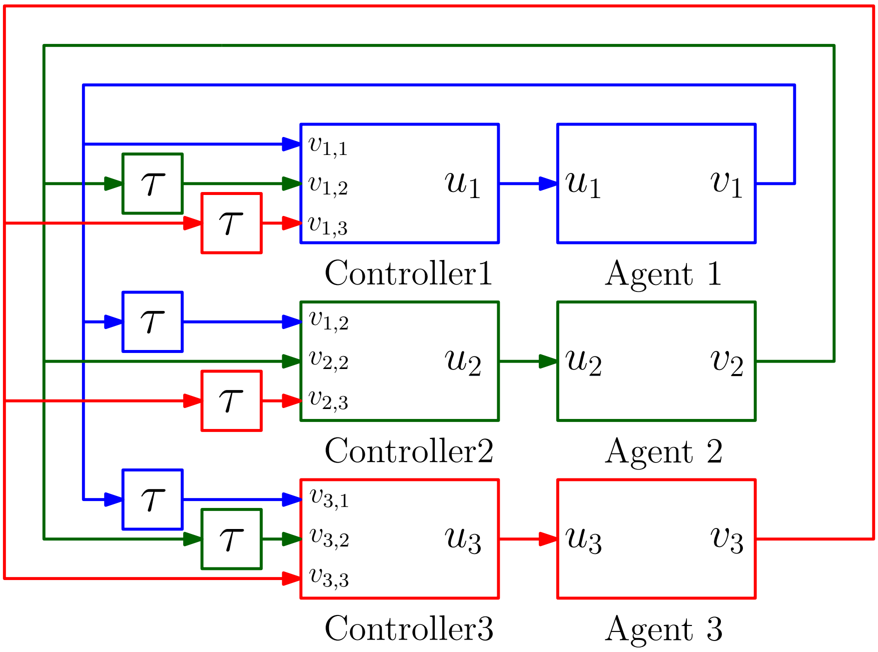


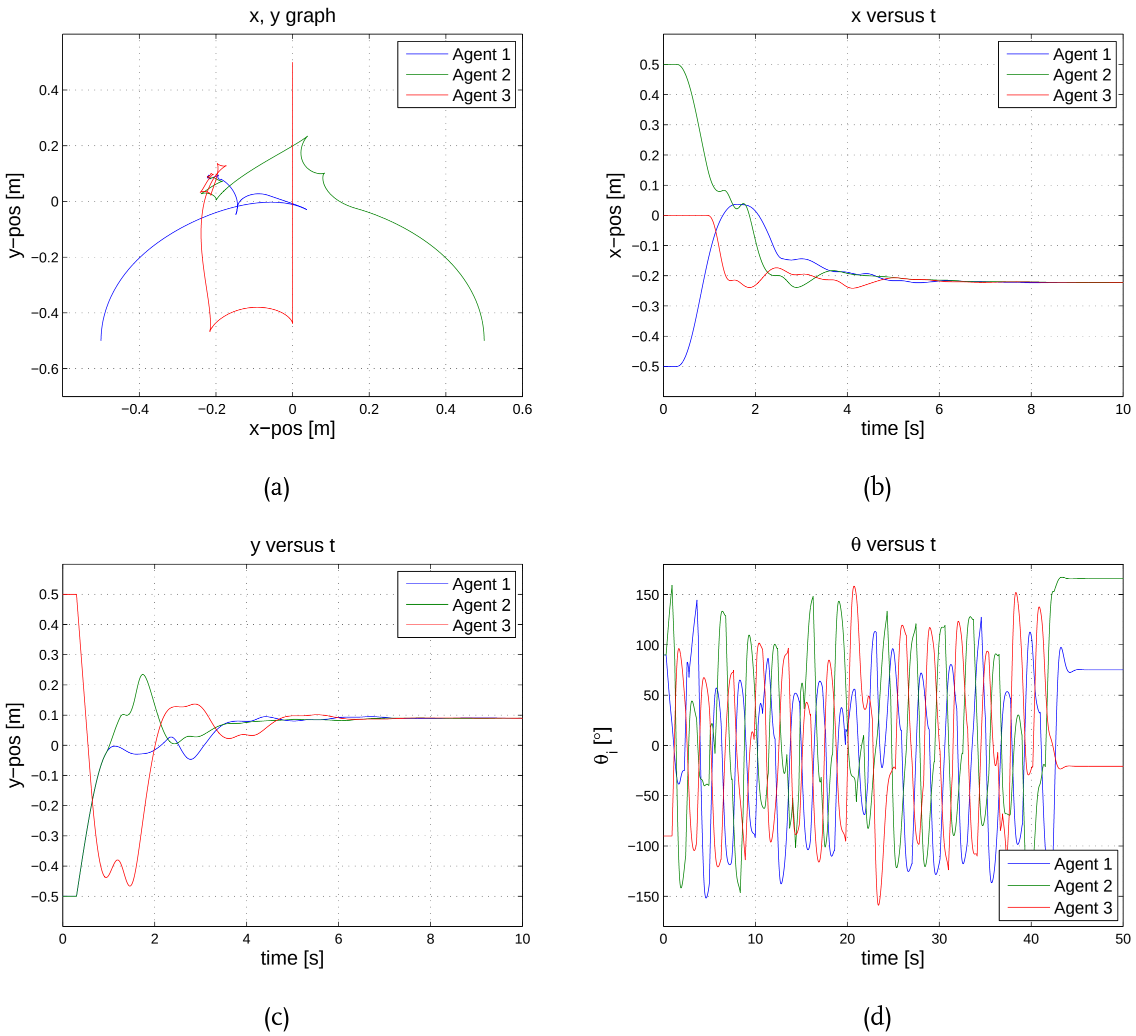
Figure 4.2 / Model of time delay case 2.

4.3 Simulation Results

Two *Simulink* models are created for both Time Delay systems. Considering simulation time and complexity of the full system only three agents are used for the remainder of this research. The fixed step size is also increased to 0.01 seconds in order to speed up simulations. The time it takes for the mobile robots to reach consensus is faster for a smaller step size. However the computation time to run simulation decreases tremendously and since the time it takes for the system to reach consensus is not the main subject of this research this forms no problems. From tests, including time delay, it follows that the mobile robots either reach consensus or that they completely diverge from each other. This makes it relatively easy to investigate the maximum time delay that is allowed for the system to be able to reach consensus.

4.3.1 The effect of Time Delay

The effect of time delay on the system is illustrated by a simulation for Time Delay case 1. The same parameters as in (3.9) were used for this simulation. The results can be seen in Figure 4.3.



(a) (b) (c) (d)

Figure 4.3 / Simulation results of three agents with $w_{ij} = 1$ and $\tau = 0.3s$.

The system reaches consensus with time delay $\tau = 0.3$ $[sec]$ but it takes a factor 2 longer compared to the system without time delay. Increasing the time delay or weights will also increase the time that is necessary to reach consensus.

The end faults that are obtained from the simulation are given below and indeed indicate that consensus is reached:

$$\|\mathbf{e_1}\| = 0.0773 \times 10^{-13}\ [m],$$
$$\|\mathbf{e_2}\| = 0.0847 \times 10^{-13}\ [m],$$
$$\|\mathbf{e_3}\| = 0.1469 \times 10^{-13}\ [m].$$

4.3.2 Relation between weights and Time Delay

The relationship between the communication weights and time delay is investigated by creating an *M-file* in combination with the mentioned *Simulink* models. For a given weight w_{ij} and time delay τ the simulation is carried out and in case $\|\mathbf{e_k}\| \leq 0.01$ for $k = 1, 2, ..., n$ consensus is reached. In that case τ is increased and the simulation is done again until $\|\mathbf{e_k}\| > 0.01$. When the norm criterion is greater than 0.01 consensus is not reached and therefore the previous time delay is the maximum allowed time delay τ_{max} for this weight in order to reach consensus. To save computing time in order to find τ_{max}, τ is increased in an iterative way with 0.01 $[sec]$ being the smallest increase of τ. Using this method a graph is created which gives the relationship between communication weight w_{12} and τ_{max}. This is done for both time delay cases and the results can be seen in Figures 4.4 and 4.5. During the simulations the initial conditions are unchanged and only weight $w_{12} = w_{21}$ is increased.

In both cases the maximum time delay decreases when the communication weight increases. This is not very hard to understand since the behavior of the system becomes more erratic for increasing weights. But because there is also a time delay this behavior is amplified and makes it harder to reach consensus. It is also clearly visible that the maximum time delay is much larger for time delay case 2 than for case 1. Looking at (4.2) the reason is probably due to the stabilizing effect that each mobile robot its own orientation is not influenced by time delay.

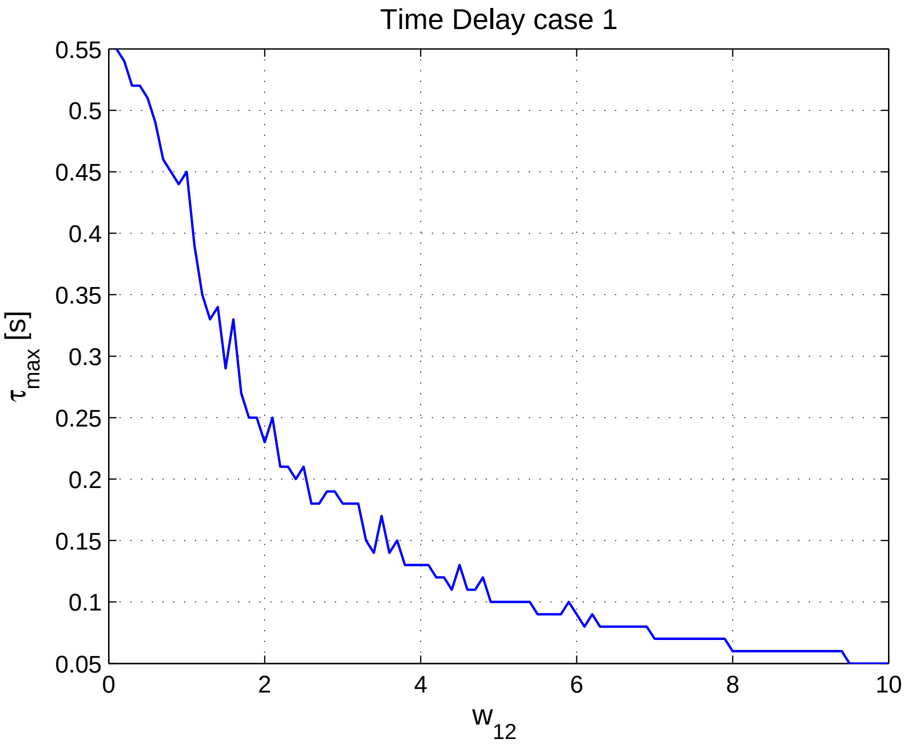


Figure 4.4 / Maximum Time Delay for increasing weight w_{12} "case 1".

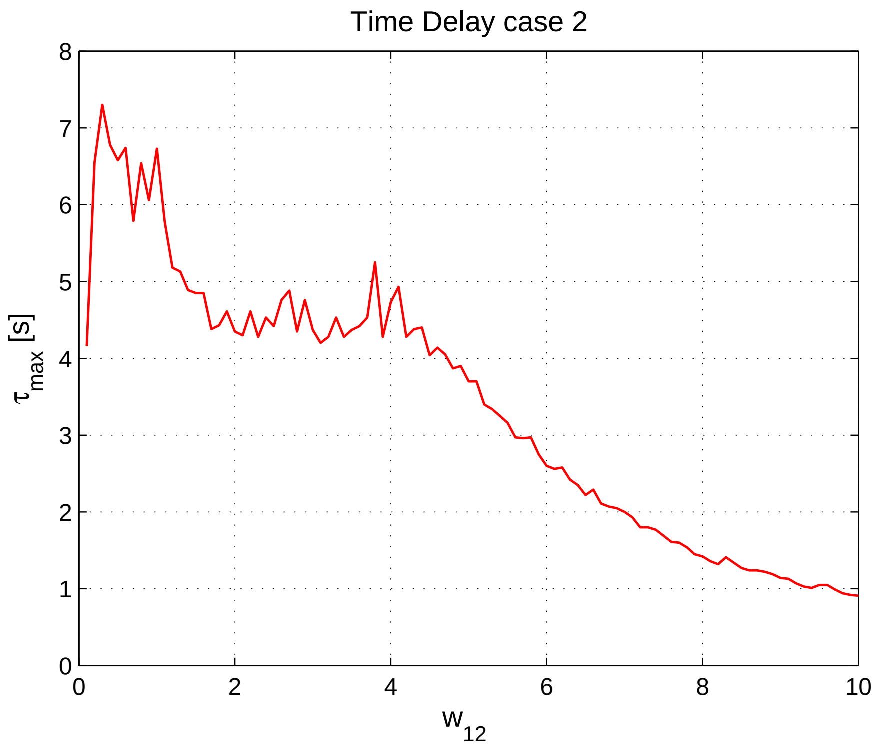


Figure 4.5 / Maximum Time Delay for increasing weight w_{12} "case 2".

4.4 Discussion

For two different cases the effect of time delay on the behavior of the mobile robots was investigated. From simulation results it can be seen that even with a time delay the mobile robots converge and reach consensus. It takes however more time for the system to reach consensus compared to the system without time delay. The behavior of the mobile robots is clearly more erratic for both time delay cases, this is amplified when the communication weights are increased. Given the same initial conditions the consensus point remains the same. However, a variation of the initial conditions in these two cases will also lead to a different consensus point.

In both time delay cases the maximum allowable time delay decreases when the communication weight increases because the erratic behavior of the mobile robots makes it harder to reach consensus. The maximum allowable time delay is much larger for case 2 since the time delay does not act on the mobile robot its own position and orientation.

CHAPTER FIVE

Experiments

In this chapter the experimental part of the research will be explained in detail. Before discussing the experiments, information regarding the mobile robots that were used and the experiment setup will be given. Most of the experimental equipment was already available, because this research subject is closely related to previous research about formation control. This saved a lot of research time since not everything had to be calibrated, assembled and a lot of practical information was already available.

5.1 Mobile Robot

For the experiments a mini unicycle mobile robot called *e-puck* was used. The e-puck is a widely used mini mobile robot that was designed for educational purposes in engineering, see Figure 5.1 and 5.2. The e-puck has various sensors and actuators and is equipped with a *dsPic* microcontroller. The main actuators that are used in the project are the two stepper motors which control the movements of the wheels. This is done with a resolution of 1000 steps per wheel revolution [13]. The e-puck has a diameter of 75 $[mm]$ and the height is 55 $[mm]$. Tests on the e-pucks were already done in previous projects by students from the *Tokyo Metropolitan University*. From these tests the maximum velocity of the e-puck was measured and is 0.146 $[m/s]$. The maximum angular velocity of the e-puck is 5.43 $[rad/s]$. The e-puck catalog states that the maximum velocity is 0.150 $[m/s]$, but this is without any additions to the e-puck (in this case special markers that were used by the vision software).

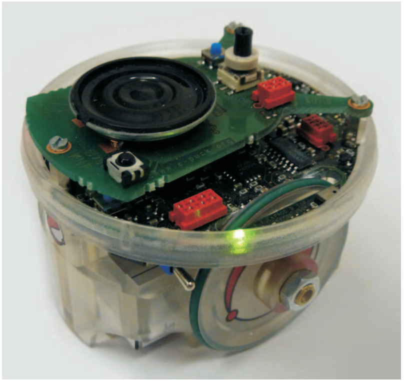


Figure 5.1 / The nonholonomic mobile robot: e-puck.

The e-puck also contains a *Bluetooth* radio link to connect to a desktop computer or to communicate with up to 7 other e-pucks. A bootloader program is used to program the e-puck over BlueTooth. Using this bootloader in combination with a remote control, called the "BTcom protocol", the e-puck can be controlled by setting the speed of the motors among other things [13]. This BTcom protocol allows the user full remote control of the e-puck from a desktop computer.

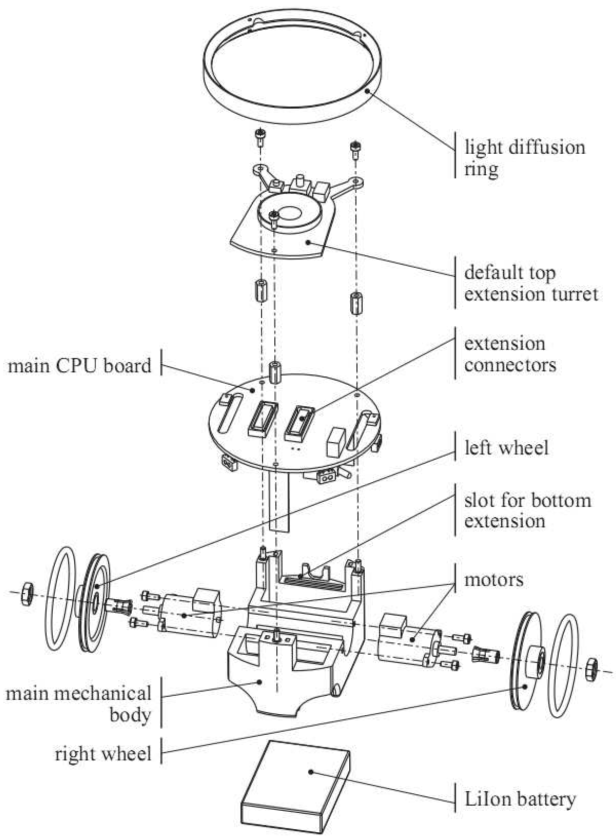


Figure 5.2 / The mechanical structure of the e-puck in an exploded view [13].

5.2 Experiment Setup

The experiment setup consists of a camera, platform, two desktop computers, BlueTooth device and e-pucks. The setup was already present at the start of this project and in the course of it completed by a *TMU* student. In Figure 5.3 a graphical representation of the experimental setup can be seen. The e-pucks are denoted with v_1, v_2, v_3 respectively. The typical operations that are carried out are as follow: the *camera* sends an image at a frame rate of 30 $[FPS]$ to desktop computer *PC 1*. Using *ARToolKit* software *PC 1* determines the orientation of the e-pucks and sends this information to desktop computer *PC 2*. This computer runs a program that, using the proposed controller from (3.4), determines the new inputs for the e-pucks. Via BlueTooth these values are send to the e-pucks. In Appendix A the hardware and software specifications of the experimental setup are given.

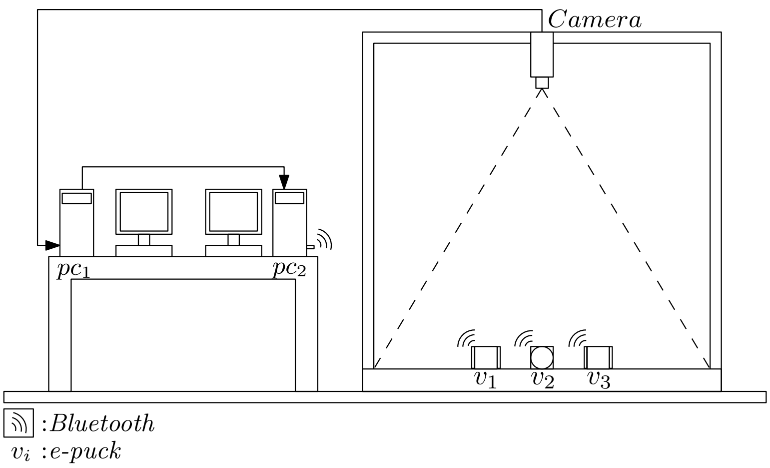


Figure 5.3 / A schematic representation of the experiment setup.

5.2.1 Time Delay in the Experiment Setup

In the various communications within the experiment setup time delay is also present. This time delay is divided over all the various operations that were explained above. It is interesting to know how much the total time delay of the experiment setup is, since it can have an effect on the experiments. Moreover it can be used in further research for improvements. The total time delay τ_{total} is subdivided into the following contributors:

- *Camera* τ_{camera}: The maximum frame rate of the camera is set at 30 $[Hz]$, resulting in a time delay for the camera of $\tau_{camera} = \frac{1}{30}$ $[sec]$. During the experiments the sampling rate of the camera is set to $\frac{1}{30}$ $[sec]$.

- *ARToolKit* τ_{ARTool}: The visual software determines the position of the mobile robots from the image received from the camera. The time delay is the average between receiving the image and sending it to *PC 2* by approximating 300 times this equals $\tau_{ARTool} = 0.006047\ [sec]$.

- *UDP* τ_{UDP}: The position of the mobile robots is send from *PC 1* to *PC 2* by means of UDP-protocol. With an average of the round-trip time for 500 times, the time delay is $\tau_{UDP} = 0.000281\ [sec]$.

- *The new input* τ_{input}: *PC 2* computes the new inputs for the individual robots using the controller and data send by *PC 1*. The sampling rate of the controller was set to $0.033\ [sec]$. There are three cases that are investigated, namely no time delay, time delay 1 and 2 which have their own time delays and is averaged for 500 times. The corresponding delays are, in case of:

  no Time Delay $\tau_{input} = 0.001758\ [sec]$,

  Time Delay case 1 $\tau_{input} = 0.001888\ [sec]$,

  Time Delay case 2 $\tau_{input} = 0.001843\ [sec]$.

- *BlueTooth* τ_{BT}: This is the time delay that occurs by sending the new velocities over the wireless communication link including the reaction time of the e-pucks. According to the developers at *Cyberotics*, e-puck can receive data by BlueTooth and execute theoretically 30 actions per second. This means that the time delay is theoretically $\tau_{BT} = \frac{1}{30}\ [sec]$. This was checked by executing a series of actions at $0.033334\ [sec/action]$ on the e-puck and it can be seen that there is only slightly loss of data.

The maximum time delay that occurs in the experiment setup is for *Time Delay case 1* and equals $\tau_{total} = 0.074883\ [sec]$. The most dominant contributors of the total time delay τ_{total} are the camera and BlueTooth time delays.

5.3 Experimental Results

Before the actual experiments various test runs were done to check what would be the best method to do the experiment. In the simulations the mobile robots agree on one point, converge to that position and eventually reach consensus. As mentioned in section 3.1 this is not possible in reality since it would mean that the mobile robots would collide into each other at the consensus point. A strategy was implemented in the program which made it possible to do experiments and avoid collisions at the consensus point. This was done as followed: for two mobile robots an offset in the position from the camera was given. Doing this, the mobile robots will think that they reach consensus but their real position will be at a specific distance from the consensus point. The reason this was not implemented in the consensus algorithm is that the method explained above seemed to be the easiest at the time being. In Figure 5.4 the end positions of the mobile robots can be seen, here the offset for $v_1 = [0.15\ 0.15]$ and for $v_3 = [-0.15\ \ -0.15]$. Thus meaning that mobile robot 2 is exactly at the consensus point.

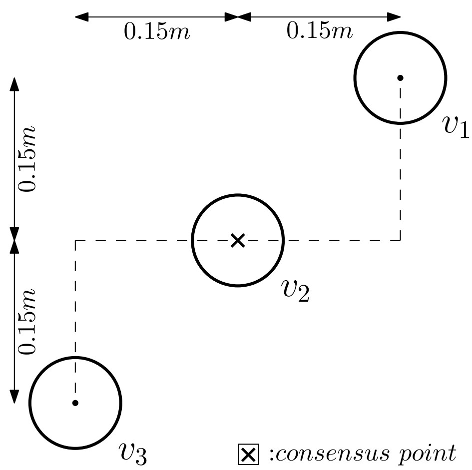


Figure 5.4 / Consensus for three mobile robots with offset.

5.3.1 Simulation and Experiment comparison

It is interesting to see wether or not the real mobile robots will behave in the same way as the simulations suggest. Therefore some experiments were done without time delay to verify if the mobile robots indeed reach consensus, using controller (3.4). It is especially interesting to see if the behavior of θ has a similar result as for the simulations. The step-size for the experiments is 0.033 $[sec]$, this is limited by the frame rate of the camera and ability of the e-pucks to execute actions. The simulation time that was used during the experiments is 60 seconds. The results can be seen in Figure 5.5

Looking at Figure 5.5(a) one can see that the mobile robots indeed converge towards each other and reach consensus. Just like in simulation the behavior of θ has some sort of oscillating behavior at the consensus point. Because this behavior continues through the entire simulation time, the mobile robots translate slowly in the y-direction, see Figure 5.5(c). The norms that are obtained from this experiment are given below and indeed indicates that consensus is reached.

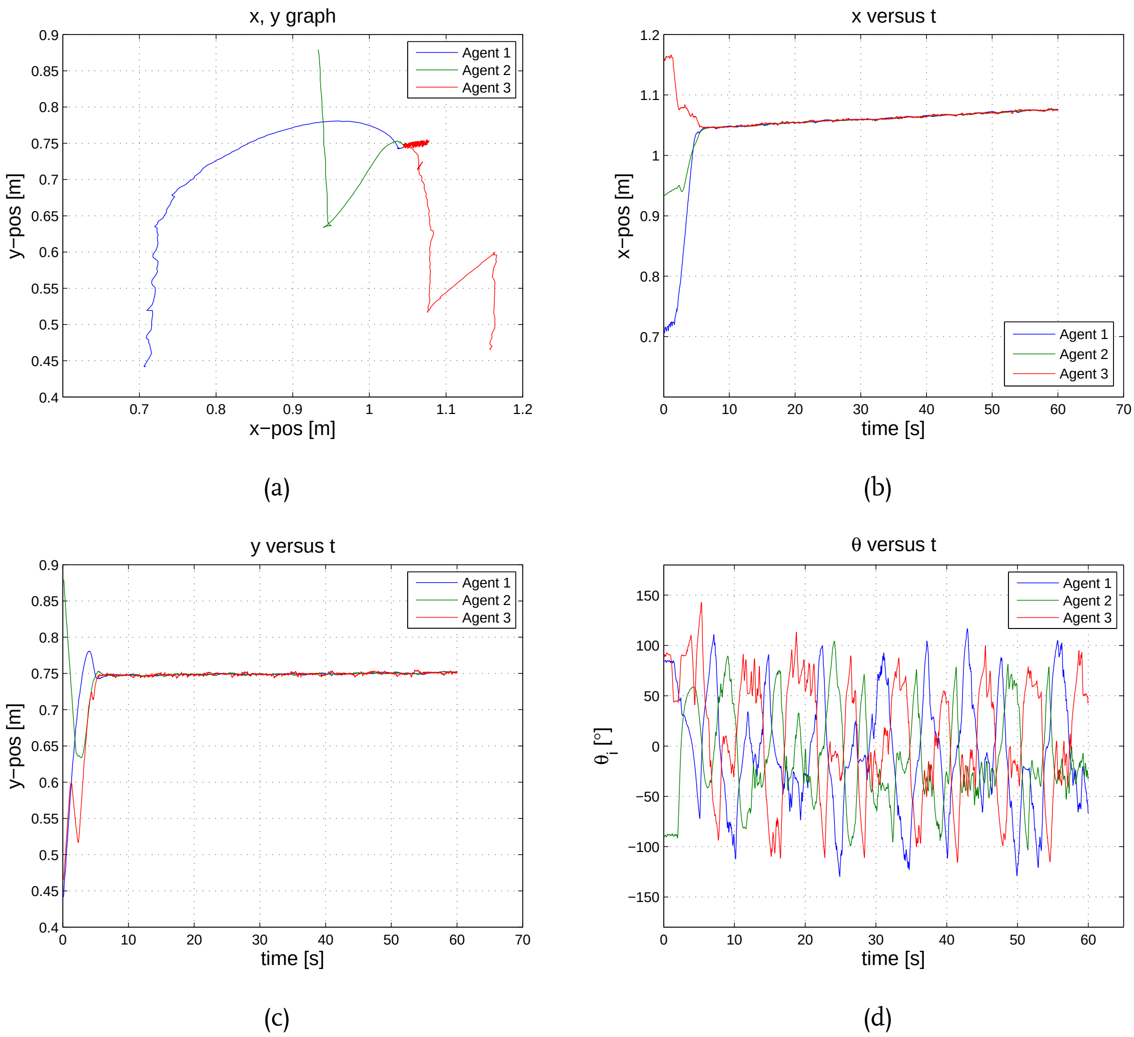


Figure 5.5 / Experiment results of three agents with $w_{ij} = 1$ and $\tau = 0$.

However the end faults for the experimental results are much higher compared to the simulation results:

$$
\begin{aligned}
\|\mathbf{e_1}\| &= 1.3111 \times 10^{-3}\ [m], \\
\|\mathbf{e_2}\| &= 0.7817 \times 10^{-3}\ [m], \\
\|\mathbf{e_3}\| &= 0.8122 \times 10^{-3}\ [m].
\end{aligned}
$$

5.3.2 Weight and Time Delay Experiments

A number of experiments were carried out to determine if the relationship between the communication weights and time delay from the simulations gives an accurate representation. The e-pucks where positioned in the same start positions for every experiment and notes were made concerning their behavior. For a number of weights the time delay was varied during experiments to investigate what the maximum allowable time delay is corresponding to that weight. The results are plotted as markers within the simulation Figures 5.6 and 5.7.

Every green marker indicates that the experiment proceeded without any errors and that consensus is reached. The red markers indicate that the e-pucks were not able to reach a consensus. For red markers near the simulation results line the e-pucks would converge to each other but eventually not reach consensus. For an even higher time delay the e-pucks would not even converge anymore. This is verified by analyzing the experiment data with *Matlab* and by using the same norm criterion as in section 3.2. The distance the e-pucks traveled before reacting was even larger than the initial spacing of the mobile robots for those time delays. During experiments the velocity of the e-pucks was closely monitored. Especially when weight $w_{12} > 4$ or when the Time Delay was too large the velocity of the e-pucks turned out to be the limiting factor. In this case the necessary velocity, in order to reach consensus, was higher than the maximum achievable velocity of the e-pucks. This is the reason why τ_{max} is lower for $w_{12} = [0.1 : 0.1 : 10]$ compared to the simulation results. The maximum allowable time delay for each weight is given by the maximum green markers in Figures 5.6 and 5.7.

By comparing Figures 5.6 and 5.7, one can immediately see that the maximum allowable Time Delay τ_{max} is much higher for Time Delay case 2. In both time delay cases the maximum velocity of the e-pucks is the limiting factor for reaching consensus. This was not yet taken in consideration during the simulation phase. This explains why the simulation results indicate a higher maximum allowable time delay.

5.4 Discussion

During the experiments the behavior of the e-pucks was also monitored. Since there was no collision avoidance control law implemented, situations occurred where the e-pucks would touch each other. Depending on their orientation this could occasionally lead to the e-pucks getting stuck against each other. In other situations they would touch each other briefly and due to their orientations could still reach consensus. The limiting factor in reaching consensus turned out to be the maximum velocity of the e-pucks. Due to this the e-pucks can not always move as fast as is required to reach consensus. This effect occurs when the weights on the communication links are increased or when the time delay becomes to large. Eventually the e-pucks will always operate at maximum velocity but will never be able to reach consensus.

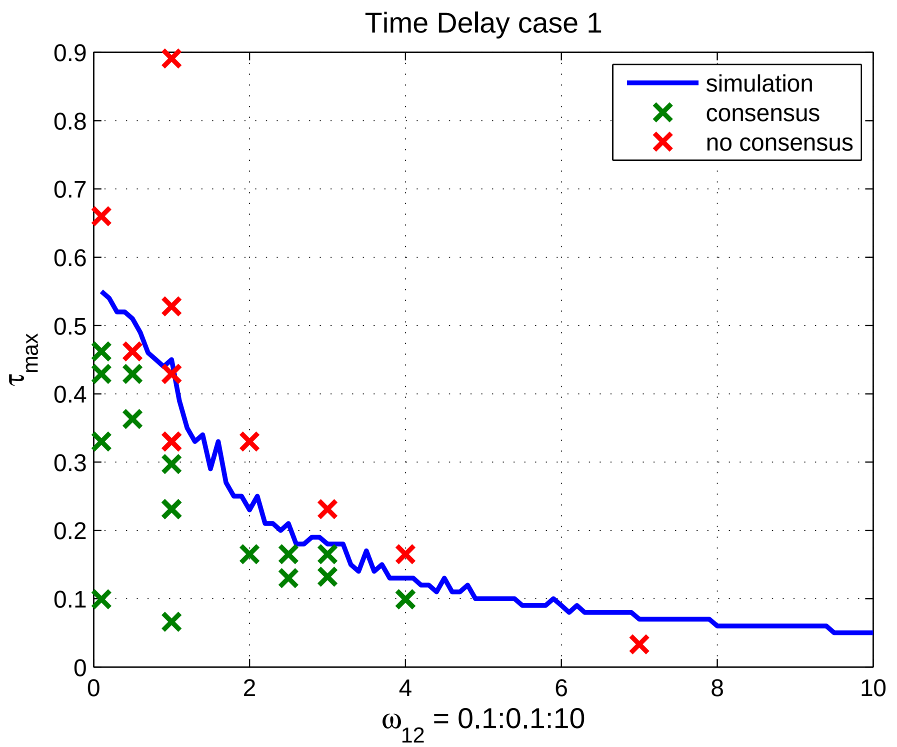


Figure 5.6 / Maximum Time Delay for increasing weight w_{12} "Experiments case 1".

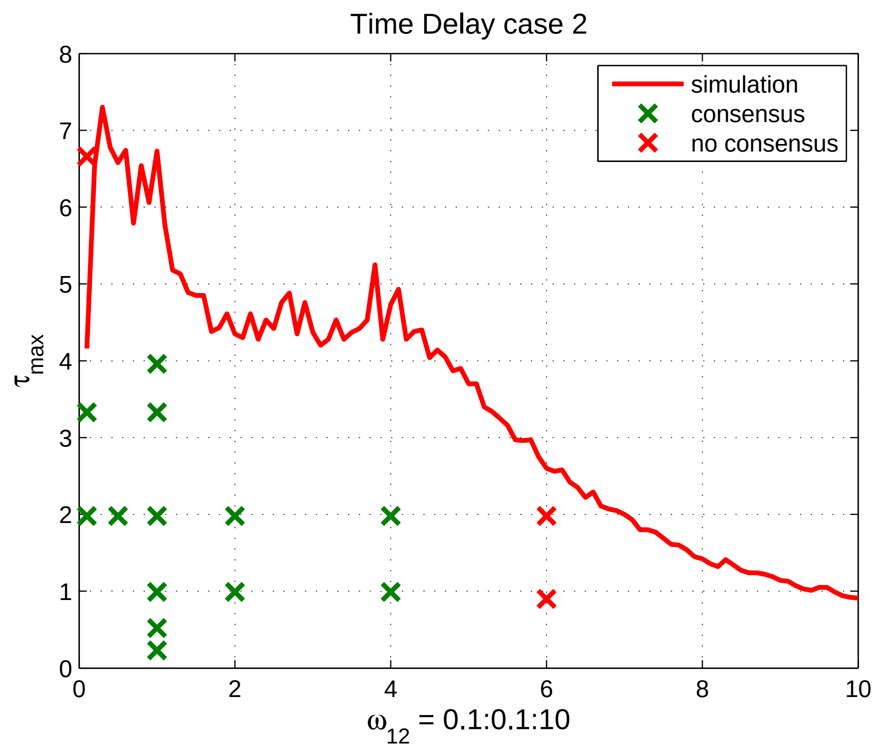


Figure 5.7 / Maximum Time Delay for increasing weight w_{12} "Experiments case 2".

Chapter six

Conclusions and Recommendations

6.1 Conclusions

The main goal of this project was to investigate the effect of time delay on the formation control of nonholonomic mobile robots. In order to investigate this effect a controller was designed which controls a group of mobile robots. With this controller, simulations and experiments were done to investigate the behavior of the mobile robots. A time delay was applied on the communication network and the effect it has on the behavior of the mobile robots was investigated by simulations and experiments.

The conclusions in this section regarding the topics that have been investigated, mentioned above, will follow the outline of this report. From the simulations and experiments it can be concluded that, using the proposed controller, the mobile robots converge to each other and eventually reach consensus. It also followed that the consensus point changes if the initial conditions are changed. If there is no time delay present then, regardless of the mobile robots initial conditions, the mobile robots will converge and reach consensus. The behavior of θ at the consensus point does not influence the position of the consensus point during simulations, however during experiments it could cause the consensus point to shift very slowly.

The effect of increasing the weights w_{ij} on the communication links between the mobile robots is also evidently present. In case there is no time delay present, increasing the weights in the simulations will still lead to consensus. But the path that the mobile robots take while converging changes because of the weights. In the experiments increasing the weights will make it harder for the mobile robots to reach consensus due to the erratic behavior that occurs. When the weight is too large the mobile robots won't be able to reach consensus. In this case the desired velocity, in order to reach consensus, will be larger than the maximum velocity of the mobile robots. This is due to the physical limitations of the mobile robots.

The previous conclusion is similar in case only time delay is increased. If the time delay is too large the mobile robots will have to respond much faster then they can in order to reach consensus. In the simulations this leads to a maximum time delay for the mobile robots in order to reach consensus. In the experiments increasing the time delay will eventually lead to a desired velocity that is larger than the maximum velocity of the mobile robots. This explains that the maximum allowable time delay decreases for increasing weights. Due to the physical limitations of the mobile robots this maximum allowable time delay will be lower for the experiments than the simulations. Basically one can regard the simulation results as a boundary for reaching consensus in case of time delay. It is very clear by comparing the two time delay cases that the influence of time delay is much greater for time delay case 1. This is due to the absence of time delay in the mobile robots itself for time delay case 2.

By investigating the behavior of the mobile robots via simulations and experiments it can be concluded that they converge to each other and eventually reach consensus. In case there is a disturbance or an error occurs in one of the mobile robots the system will still reach consensus. Using this as a base for formation control a new control law could be created what ensures that a group of mobile robots will stay in formation regardless of disturbances. However, due to the complexity of this system the effect of time delay on such a system will probably by larger.

6.2 Recommendations

During the course of this internship choices had to be made regarding the focus of this research. By doing so, some topics could be investigated more in depth while less attention was paid for other topics. As can be seen from the simulations and experimental results the behavior of θ at the consensus point does not agree with the theory. Some more in depth research could be done to analyse this behavior. Also more in depth analysis of the proposed controller regarding stability would be desirable since there is no mathematical proof at this moment that it leads to consensus. Experiments were only carried out for three mobile robots but this can be upgraded to any number of mobile robots. This makes it interesting to investigate if there is a relationship between the maximum time delay and the number of mobile robots.

During this research the main topic was about reaching a consensus point, but in many situations a specific formation is required. The proposed controller in this project could be modified by implementing a *repulsive potential function* [14][15]. In [3] a controller is presented that utilizes such potential functions and can be used to let a group of mobile robots reach a formation. After achieving a formation the effects of time delay on the communication network can be investigated again. A method for collision-free motion coordination of a group of unicycle agents is proposed in [8]. Collision-free motion coordination is very appealing for a project like this since the mobile robots can collide during experiments. It is also possible to investigate if the mobile robots converge and reach consensus when the controller and robot are linked via a delay inducing communication channel as in [4].

Appendix A

This Appendix covers the hardware and software that is used on the experiment setup.

Camera

The camera that is used during experiments is a *Point Grey Research, Inc.* $Flea^{®}2$ camera. The specifications of this camera can be seen in Table A.1. For more specifications please refer to [16].

FL2-08S2M/C Model	
Image Sensor Model	Sony ICX204 1/3"
Dimensions	$29mm \times 29mm \times 30mm$ (excluding lens holder)
Maximum Resolution	1032×776 [pixel]
Experiment Resolution	1024×768 [pixel]
Pixel Size	4.65×4.65 [μm]
Maximum Frame Rate	30 *FPS*
Lens Mount	C

Table A.1 / Camera specifications.

Lens

The lens that is mounted on the camera is a *Fujinon Lens, Inc* with specifications given in Table A.2. This lens is specially designed for optimizing the image quality of *3CCD* cameras. For more specifications please refer to [17].

TF2.8DA-8 Model	
Focal Length	$2.8mm$
Iris Range	F2.2$\sim$F16·Close
Angle of View (H×V)	$1/3$" $89°08' \times 69°20'$
Focusing Range	$\infty \sim 0.1$
Lens Mount	C

Table A.2 / Lens specifications.

Operating System

During experiments two desktop computers are used:

- *PC 1*: This computer is used for detecting markers (e-pucks) and calculating the x, y and θ orientation of them. This is done by using the *ARToolKit* software [18]. The Operating System for *PC 1* is *Windows XP*.

- *PC 2*: This computer is used for receiving position data from *PC 1* via *UDP* (User Datagram Protocol), calculating the input values for the e-pucks using the controller and transmitting input values for the e-pucks via a *BlueTooth* adaptor. The Operating System for *PC 2* is *Linux* with *Ubuntu 10.04 LTS* Version.

BlueTooth

For communicating with the e-pucks a *BlueTooth* device is used with the specifications given in Table A.3. For more specifications please refer to [19].

BSHSBD02BK Model	
USB Interface	USB2.0
Transmit Frequency Range	$2.4GHz \sim 2.4835GHz$
Communication Output	Max $100mW$ (Class I)
Communication Distance	about $100m$

Table A.3 / BlueTooth adapter specifications.

Bibliography

[1] P.-A. Bliman and G. Ferrari-Trecate, "Average consensus problems in networks of agents with delayed communications," *Automatica*, vol. 44, pp. 1985–1995, 2008.

[2] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, "Dynamic consensus for mobile networks," *International Federation of Automatic Control world congress*, 2005.

[3] K. Listmann, M. Masalawala, and J. Adamy, "Consensus for Formation Control of Nonholonomic Mobile Robots," *ICRA. IEEE International Conference on Robotics and Automation*, vol. 12-17, pp. 3886 – 3891, May 2009.

[4] A. A. Aguirre, H. Nijmeijer, T. Oguchi, and K. Kojima, "Remote Control of a Mobile Robot Subject to a Communication Delay," *Proceedings of the 7th International Conference on Informatics in Control, Automation and Robotics (ICINCO), Funchal, Madeira*, 2010.

[5] A. A. Aguirre, H. Nijmeijer, and T. Oguchi, "Long Distance Synchronization of Mobile Robots," *Proceedings of the 2010 American Control Conference, Baltimore, MD*, 2010.

[6] K. Kojima, T. Oguchi, A. A. Aguirre, and H. Nijmeijer, "Predictor-Based Tracking Control of a Mobile Robot with Time-Delays," *8th IFAC Symposium on Nonlinear Control Systems (NOLCOS), Bologna*, 2010.

[7] D. Kostic, S. Adinandra, J. Caarls, N. van de Wouw, and H. Nijmeijer, "Saturated Control of Time-Varying Formations and Trajectory Tracking for Unicycle Multi-agent Systems," *IEEE Conf. on Decision and Control, Atlanta*, 2010.

[8] D. Kostic, S. Adinandra, J. Caarls, and H. Nijmeijer, "Collision-free Motion Coordination of Unicycle Multi-agent Systems," *American Control Conference, Baltimore, Maryland*, 2010.

[9] T. van den Broek, N. van de Wouw, and H. Nijmeijer, "Formation Control of Unicycle Mobile Robots: a virtual structure approach," *Conference on Decision and Control*, vol. 15-18, pp. 8328–8333, December 2009.

[10] C. Godsil and G. Royle, “Algebraic Graph Theory,” *Graduate Texts in Mathematics, Springer*, vol. 207, 2001.

[11] R. Olfati-Saber and R. M. Murray, “Consensus Protocols for Networks of Dynamic Agents,” *American Control Conference*, vol. 2, pp. 951–956, June 2003.

[12] R. M. Murray and R. Olfati-Saber, “Consensus Problems in Networks of Agents with Switching Topology and Time-Delays,” *Automatic Control, IEEE Transactions on*, vol. 49, no. 9, pp. 1520 – 1533, September 2004.

[13] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.-C. Zuffery, D. Floreano, and A. Martinoli, “The e-puck, a Robot Designed for Education in Engineering,” *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, pp. 59–65, 2009.

[14] V. Gazi and K. Passino, “A Class of Attraction/Repulsion Functions for Stable Swarm Aggregations,” *Decision and Control, Proceedings of the 41st IEEE Conference on*, vol. 3, pp. 2842 – 2847, December 2002.

[15] V. Gazi, B. Fidan, Y. S. Hanay, and M. I. Köksal, “Aggregation, Foraging and Formation Control of Swarms with Non-Holonomic Agents using Potential Functions and Sliding Mode Techniques,” *Tübitak*, vol. 15, no. 2, 2007.

[16] *Flea®2 Camera Specifications.* http://www.ptgrey.com/products/flea2/index.asp: Point Grey Research Inc, Oktober 2010.

[17] *3CCD Camera Lenses Specifications.* http://www.fujinon.com/ProductDocs /TF2.8DA8.pdf: Fujinon Lens Inc, Oktober 2010.

[18] *ARToolKit Website.* http://www.hitl.washington.edu/artoolkit/: ARToolKit, Oktober 2010.

[19] *BlueTooth Specifications.* http://buffalo-kokuyo.jp/products/bluetooth/bluetooth /adapter/bshsbd02bk/: Buffalo Inc, Oktober 2010.