COORDINATION OF MULTIPLE AUTONOMOUS VEHICLES WITH DIRECTED COMMUNICATION GRAPHS

By

SHYAMPRASAD KONDURI

Bachelor of Engineering in Mechanical Engineering
Osmania University
Hyderabad, Andhra Pradesh, India
2007

Submitted to the Faculty of the
Graduate College of
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
July, 2012

COORDINATION OF MULTIPLE AUTONOMOUS VEHICLES
WITH DIRECTED COMMUNICATION GRAPHS

Thesis Approved:

Dr. Prabhakar R. Pagilla

Thesis Advisor

Dr. Gary E. Young

Committee Member

Dr. Andrew S. Arena Jr.

Committee Member

Dr. Sheryl A. Tucker

Dean of the Graduate College

ACKNOWLEDGMENTS

I would like to express my sincerest gratitude to my advisor, and thesis committee chairman Dr. Prabhakar R. Pagilla for his supervision, guidance and inspiration. He encouraged to work on the multi-vehicle coordinated control problem. Working under his guidance has been a wonderful learning experience both professionally and personally. I will definitely benefit from the knowledge I have accumulated for the rest of my life.

I would also like to thank Dr. Gary E. Young and Dr. Andrew S. Arena for being on my thesis committee and for the knowledge filled courses they taught during my years here in Oklahoma State University.

I would like to thank Dr. Mauro Cimino for his guidance during the initial stages of this project. I have learned a lot from his experience in designing and managing the project. His encouragement has been helpful during many setbacks.

I want to thank my colleagues at Oklahoma State University Emilio Gabino, Orlando Cobos, Andrew Burnap for their help and discussion all through the time of my work. I would also like to thank Pramod Raul, Aravind Seshadri and Carlo Branca for their intellectual inputs during various stages of the project.

This work was supported by the US National Science Foundation under Grant No. 0825937.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

CHAPTER 1

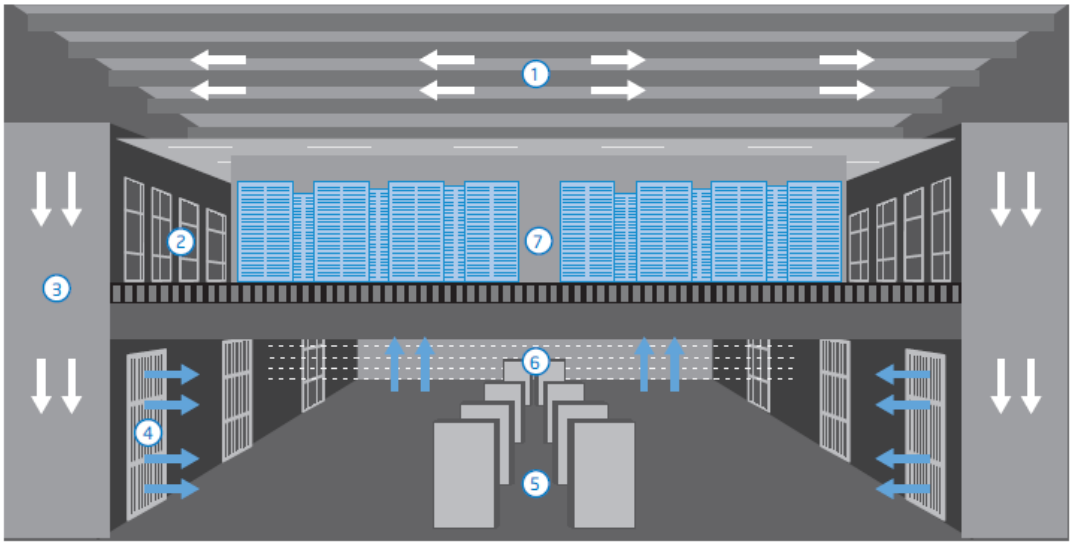
INTRODUCTION

The area of multiple autonomous vehicle systems has been one of the most studied areas of autonomous systems and robotics in the last decade. A system of multiple vehicles can accomplish tasks which are a combination of several subtasks or that they can span a large geographical area more effectively than any single vehicle whose capabilities are limited. Multiple autonomous robots find use in a vast number of areas. They can be used in defense applications like patrolling, search and rescue, disaster management, etc. Automated highway systems is another important application. Unmanned aerial vehicles, unmanned sea vehicle, vehicle platoons are some examples of multiple vehicles systems.

1.1 Coordinated Control

Coordinated control refers to the control of individual systems of a group to achieve a collective objective. It is widely used in many applications that involve controlling multiple interconnected systems. Coordinated control is used in maintaining computer servers, data centers, traffic signals, memory management in operating systems, vehicle control, etc., [3–6]. Of all the areas perhaps the most interesting is the coordination between autonomous or semi-autonomous vehicles in formations [7–9]. Figure 1.1 shows examples of various applications of coordination control.

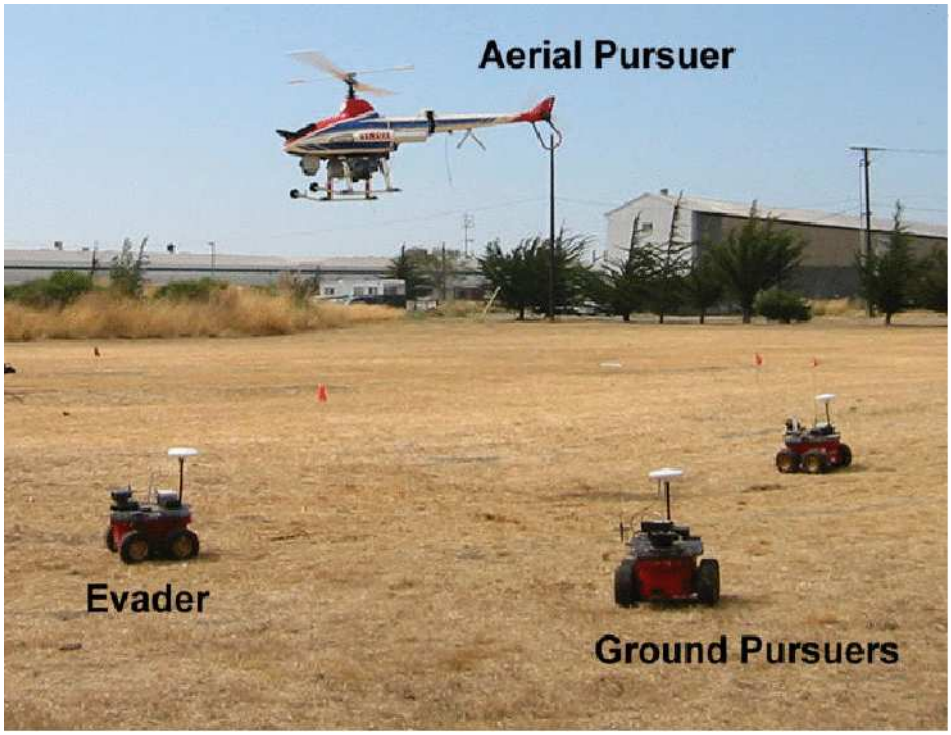
Coordination in vehicle formations is used to achieve collective objectives and to accomplish a wide variety of tasks. In many applications a group of vehicles is required to maintain the formation and follow a trajectory. The advantages include robustness,



(a) Support systems in data centers [10]

(b) Traffic signals based on traffic flow in different routes [11]

(c) Formation of air planes concept [12]



(d) Air and surface vehicle coordination [13]

Figure 1.1: Applications of coordinated control

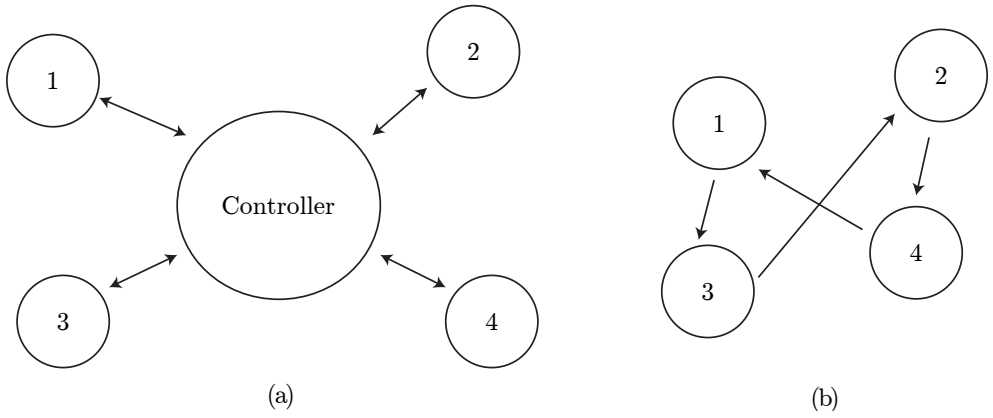


Figure 1.2: Controller architecture, (a) Centralized controller, (b) Decentralized controller

reliability and accuracy. But one has to deal with many issues in coordination of vehicles, such as formation stability, uncertainties in formation, controllability of formations and inter-vehicle communication [8].

The architecture of a controller in coordinated control is divided mainly into two types:

Centralized control: There is only one controller which generates the control inputs for all the vehicles in the group. The control input is then communicated to the individual vehicles. It is simple but as the number of vehicles increase, the number of computations required to control the formation increase proportionally.

Decentralized Control: Each vehicle is autonomous and has its own controller which generates the control input for the respective vehicle. Each controller acts based on a coordinated approach and takes as input among others the state of its own vehicle and the neighboring vehicles.

It should be clear that even for a small number of vehicles, as the complexity of the operations increases, a centralized controller fails to operate effectively. However, the decentralized (distributed) controller is much more easy to implement since each

vehicle is autonomous and control inputs are locally generated based on the information from the neighboring vehicles. Information flow between the vehicles is key in a decentralized controller, and this is referred to as the inter-vehicle communication. Information flow can occur in different ways. Each vehicle estimating the position of neighboring vehicles using on board sensors is one way the information can flow between vehicles. Another way is to have a communication link between vehicles which is used to pass information like position, heading, velocity, etc.

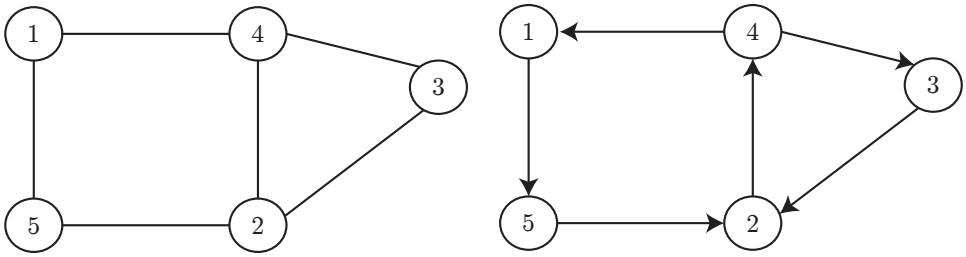
1.2 Information Flow Graphs

Inter-vehicle information flow is vital for controlling multiple vehicle formations. Graph theory is generally used to model the information flow between vehicles. A graph which represents the information flow of a vehicle formation is called an Information Flow Graph or Communication Graph. Information flow graph together with closed loop control structure is used to study the stability and controllability of formations [8, 14]. They are used to develop distributed controllers. Communication graphs are classified mainly into two categories:

1. Directed Graph: A graph is directed if the information flows from node i to node j but does not necessarily flow from node j to node i.

2. Undirected Graph: An undirected graph is a graph in which the information flows from node i to node j and also from node j to node i.

Figure 1.3 shows a formation of five vehicles with directed and undirected information flow graphs. Early coordination controllers were often based on undirected information flow graphs [15–18]. In [15] Barooah et al. used continuous approximation to convert the vehicle dynamics into a partial differential equation and explained the loss of stability as the number of vehicles increase and how mis-tuning the controller helps reduce this stability loss. However, with this approach the loss of stability



(a) (b)

Figure 1.3: Formation of five vehicles with (a) Undirected graph and (b) Directed information flow graph

is slowed down with increase in number of vehicles but not entirely eliminated. It has been shown that a controller based on an undirected graph is not scalable as the number of vehicles in the formation increases [15, 19] and the formation is not string stable. Hence, it is clear that one cannot use undirected information graphs for large vehicular formations.

One of the alternatives is to use a directed graph. In some early works like [9, 20] a type of directed graph is used where each vehicle receives communication from the preceding vehicle or vehicles. In this type of formation there is usually a leader vehicle which has all the information of the trajectory and multiple follower vehicles that follow the leader by maintaining a desired spacing. In [9] the author analyzed a longitudinal control law for a vehicle platoon of identical and non-identical vehicles. In this the information of first vehicle is communicated to all the other vehicles along with the information of respective preceding vehicle. Although a leader-follower model as in [9] is simple it is not string stable and the controllers are not scalable.

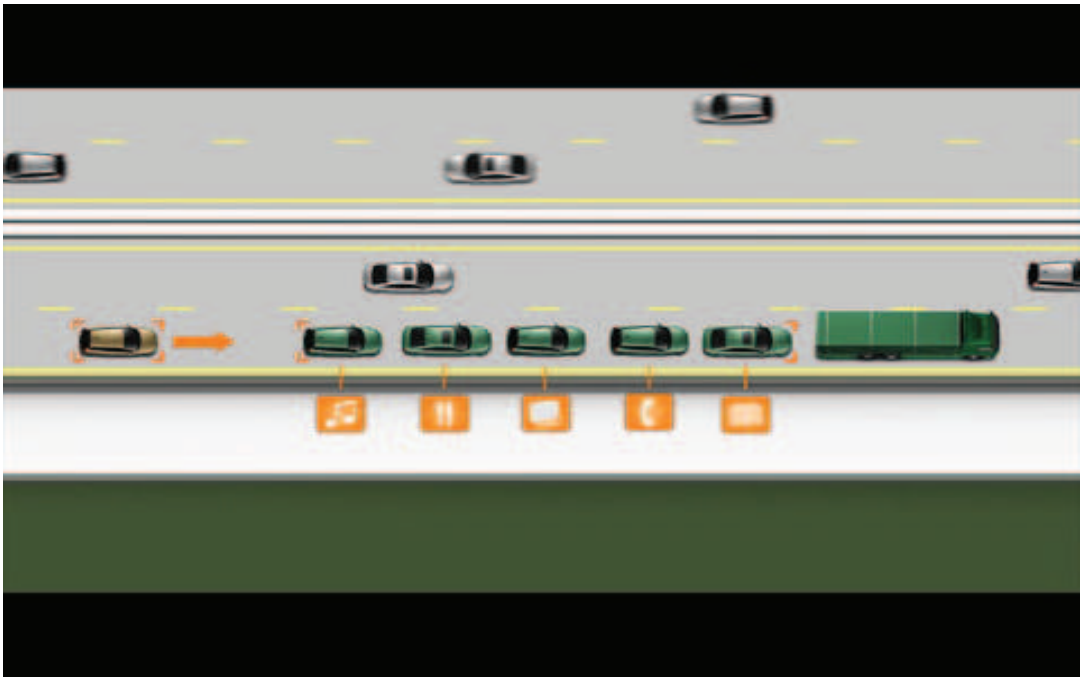
There has been a great deal of research in recent years on multi-vehicle formation control using other types of directed graphs [21–24]. One key question is how does one to generate a directed information flow graph for a given formation of N vehicles that will satisfy all the requirements. While this is still an unanswered question, there has been some work in literature in addressing this question for simple cases

like the 1D case [21–23]. In [22] Klinge and Middleton use L_2- and $L_\infty-$norms of the states to derive the conditions that guarantee string stability for a formation of vehicles with unidirectional nearest neighbor communication. References [21, 23] use the same unidirectional one to one communication but in the form of a ring. These controllers are based on a circulant graph. Any graph whose Laplacian matrix is a circulant matrix is called a circulant graph. Circulant matrices are a special case of Toeplitz matrices [25]. The unique properties of circulant graphs provide a simple directed communication topology for the entire platoon of vehicles.

1.3 Vehicle Platoons

A string of vehicles traveling together on a road is a called a vehicle platoon. Figure 1.4 shows examples of vehicle platoons. Vehicle platoons have been studied for many decades due to many reasons. As the number of vehicles on highways continues to increase the average travel time has also increased and so has the number of accidents due to human errors. It is expected that platoons of automated vehicles can increase highway capacity, reduce fuel consumption, reduce travel time, avoid collisions, etc., [15, 22]. Since the vehicles are all autonomous they are free of human error and thus are expected to perform better even in demanding conditions like smaller inter-vehicular distance, greater speeds, etc. The advantages of platoons of vehicles are listed below:

1. Increased highway capacity due to small inter-vehicle spacing;
2. reduction in accidents that are caused due to human error;
3. reduction in fuel consumption owing to reduced speed fluctuations;
4. decrease in traffic congestion on highways;



(a)

(b)

(c)

Figure 1.4: Examples of vehicle platoons. (a),(b) Conceptual sketch and actual experiment of road trains (SARTRE project [1]), (c) Vehicle platoon (PATH project [2])

5. Reduction in travel times due to increased speed and reduced traffic congestion.

These advantages have led to a considerable amount of research in vehicle platoons over the past two decades [21, 26–30]. Most of the earlier works are based on sensing the position of preceding vehicle and maintaining a desired inter-vehicle spacing. In [29] each vehicle is considered as a reactive agent whose behavior is based on the position of the preceding vehicle. In [27] a non-linear vehicle model comprising of the propulsion force and brake force is derived which is then used to calculate the acceleration and deceleration based on the preceding vehicle position. An adaptive controller which can deal with variations in model parameters such as mass, drag, etc., is developed in [28]. One thing that is common to the reported work in literature on vehicle platoons is the use of a coordinated decentralized or distributed controller. This current thesis also uses a distributed controller developed for vehicle platoons and is based on the communication graph being a ring graph.

1.4 Contributions

The contributions of this thesis are summarized as follows:

- A platoon controller from the literature that is based on double integrator vehicle dynamics is studied in detail. Scalability of the controller is analyzed.
- The limitations of vehicles coupled in the form of a ring and the possible modifications to the vehicle positions in the ring that would satisfy various physical limitations are investigated. It has been shown that the same controller from the literature is also stable for platoons with these modified platoons. Simulation results for both basic and modified platoons are presented.
- The effect of having multiple rings on a vehicle platoon are studied. Simulation results are presented which show that for a platoon, multiple rings provide faster convergence than a single ring.
- The question of creating a ring graph over a vehicle formation is answered. A method to create a ring graph for a linear vehicle platoon is presented and demonstrated.
- An investigative study into the TSP formulation and the solution algorithms available in the literature to solve the TSP which are suitable for forming a ring graph was undertaken. A modified TSP with a constraint on path length is formulated to satisfy all the problem needs of finding a ring graph. An algorithm based on the branch and bound technique is presented. The working of the algorithm is demonstrated for a five vehicle and an eight vehicle platoon.
- An experimental platform consisting of multiple mobile robots was developed to test various coordinated vehicle control algorithms. Each robot has an autonomous controller with a wireless transceiver to communicate with other robots and a central monitoring station.

- Experiments of coordination control using the path and ring graphs at slow speed are conducted to test the stability of the controllers on the platform developed.

CHAPTER 2

VEHICLE FORMATIONS AND RING GRAPH

The properties of controllability and stability of a vehicle formation depend on the flow of information between vehicles. Hence a study into the properties of the information flow graph is essential prior to any practical implementation. This chapter focuses on the properties of the ring graph over a vehicle platoon.

Section 2.1 provides an introduction to graph theory. The disadvantages of using undirected graph as information flow graph are presented. The basic idea behind a ring graph is also discussed in section 2.1. In section 2.2 the dynamics of vehicle platoons with a ring graph are derived. A few important properties of a circulant matrix are also discussed. The eigenvalues of the system are derived using the properties of circulant matrices and scalability of the platoon is explored. An important limitation for a vehicle platoon is the communication range of sensors transmitting information. Section 2.3 explains this limitation in detail and alternatives to the ring graph discussed in section 2.2 and their stability.

2.1 Graph Theory and Information Flow Graphs

Graph theory is used widely for modeling multiple vehicle systems. Graphs are used to describe the interconnection of vehicles and are also used to study various properties of vehicle formations. A graph is an abstract representation of a set of objects with some pairs of objects connected by links. It is defined below [31].

Definition 2.1 *A graph is a pair $G = (V; E)$ of sets satisfying $E \subseteq [V]^2$. Thus, the elements of E are 2-element subsets of V.*

The elements of set V are called the vertices (or nodes or points) of the graph and that of E are called the edges (or lines). The number of edges at a vertex $v \in V$ is called the degree (or valency) of the vertex. It is represented by $d_G(v)$ or $d(v)$. A graph is said to be connected if the degree of all the vertices is at least one ($d(v) \geq 1,\ \forall\, v \in V$). Graphs are classified into two broad types undirected (or symmetric) and directed (or asymmetric). Only a graph which is connected can be classified into undirected and directed graph. A spanning tree of a connected graph is defined as follows.

Definition 2.2 *Given a connected, undirected graph, a spanning tree of that graph is a subgraph that is a tree and connects all the vertices together.*

A special case of spanning tree called the minimum spanning tree is used in the next chapter to solve the Traveling Salesman Problem. We will now define two important properties which are used in later sections and chapters. Let in a graph G of size (no. of vertices) n, $x_i \in V$ represent the vertices and $x_i x_j \in E$ represent the edge between vertices i and j, then one can define a path as follows.

Definition 2.3 *A path is a non-empty graph* $P = (V, E)$ *of the form* $V = x_0, x_1, \cdots, x_k$ *and* $E = x_0x_1, x_1x_2, \cdots, x_{k-1}x_k$

If $P = x_0 \cdots x_{k-1}$ is a path and $k \geq 3$, then the path $C := P + x_{k-1}x_0$ is defined as a cycle. If a graph is connected then there exists at least one path between any two vertices. Using the concept of a cycle, we now define what is called a Hamiltonian cycle.

Definition 2.4 *A Hamiltonian cycle in a graph is a cycle passing through all the vertices of the graph. A graph is called Hamiltonian if it has at least one Hamiltonian cycle.*

Before the application of graphs to the control of formations is discussed, the concept of 'rigid' formation is introduced. A formation of vehicles is called a rigid formation

if the relative position of vehicles with respect to the neighbors is maintained constant. In other words, one desires to maintain the pairwise distances between vehicles constant throughout the motion of formation [19]. To maintain rigid formations one uses a combination of control, actuation, sensing and inter-vehicle communication all of which are part of coordinated control. Any typical coordinated controller must be able satisfy two properties:

Stability: It should be capable of achieving the formation and keeping it stable throughout the motion of the group of vehicles.

Scalability: The controller should remain stable for all sizes of formation. If the controller is not scalable then as the number of vehicles in the formation increase, the formation suffers from an effect called string instability.

String instability is defined below:

Definition 2.5 *A vehicle platoon is said to be string instable if disturbances are not attenuated as they propagate through the system [32].*

Stability and scalability are strongly dependent on the type of information flow graphs used in the formation [14, 19, 33]. It has been shown in [33] that for a rigid formation consisting of homogeneous vehicles undirected graphs cannot be used to design a scalable controller. The same has been proved to be valid for formation of heterogeneous vehicles in [19]. As a consequence of these results one can conclude that, it is not possible to design a scalable controller for a rigid vehicle formation using an undirected information flow graph.

A possible alternative is to use a directed information flow graph. In this work a type of directed graph called the ring graph is used where each vehicle in the formation is coupled to the other vehicles in the form of a directed ring. A ring graph is defined below.

Definition 2.6 *A graph where each node $i \in V$ is connected to only two other nodes such that information is received from one node and transmitted to the other is called a ring graph.*

As a consequence of Definition 2.6 every ring graph is a Hamiltonian cycle or every ring graph is a Hamiltonian graph with only one Hamiltonian cycle.

The basic idea of ring coupling for coordinated control has been used previously in [23,34]. The references [23,35,36] are based on cyclic pursuits in which each agent i starts from a point on the plane and captures its respective agent $i+1$ and pursues it in a cyclic fashion as shown in Figure 2.1. In [23] each vehicle is modeled as a kinematic unicycle. The system's equilibrium formations and the global behavior are studied. The control strategy for the pursuit is based on the concept of reducing the distance between the i^{th} vehicle and its neighbor. Reference [21] used a one dimensional second order dynamic model for the vehicle and developed a controller based on ring coupling for non-cyclic motion. In the present work the dynamics of the system are developed and the analysis is carried out similar to [21].

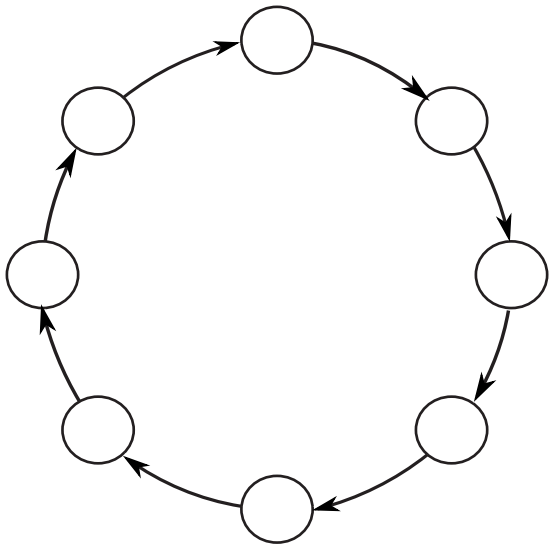


Figure 2.1: Vehicles in cyclic pursuit

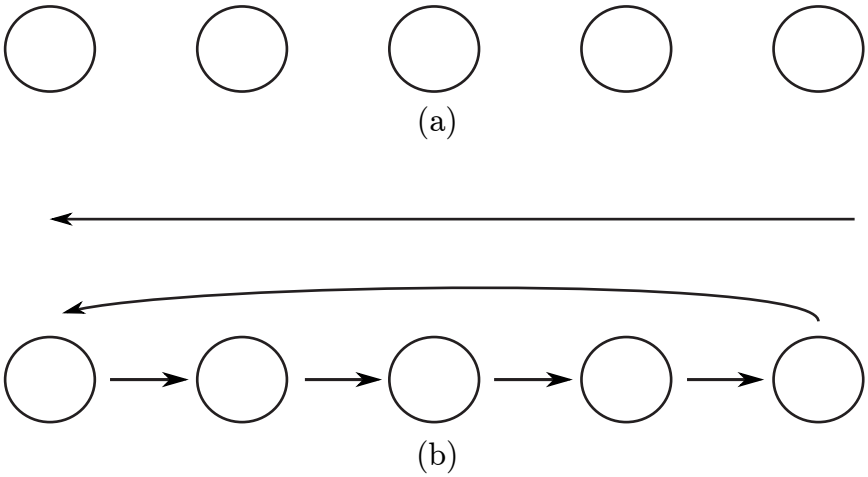


Figure 2.2: (a) Platoon of Vehicles, (b) Basic ring graph between the vehicles.

2.2 System Dynamics

Consider the platoon of vehicles shown in Figure 2.2(a). The closed loop dynamics of this vehicle platoon are derived in this section. Here we assume that the platoon moves only in the direction x. This is possible as the higher dimensions in some cases can be decoupled and the same derivation can be applied in all the directions (y and z). Considering second order dynamics, each vehicle $i \in \Phi$ in the platoon Φ is taken as a moving point mass with dynamic equation,

$$m_i \ddot{x}_i + p\dot{x}_i = u_i \tag{2.1}$$

where m_i is the mass and x_i is the position of the i^{th} vehicle, respectively, $p \geq 0$ is the friction coefficient and u_i is the input to i^{th} vehicle. If a constant input $u_i = w$ is given, the vehicle velocities will converge to the same value and an inter-vehicular distance depending on the initial conditions of the platoon Φ_0 will result.

All the vehicles are coupled in a unidirectional ring at communication level meaning that the i^{th} vehicle receives its information from the $(i-1)^{th}$ vehicle and the ring is closed with the last vehicle communicating with the first vehicle, i.e, $x_0 = x_N$ where N is the number of vehicles in Φ. Thus, the neighboring vehicle of each vehicle i is $(i+1)^{th}$ vehicle and the neighboring vehicle of the last vehicle in the ring is the first

vehicle. Figure 2.2(b) shows the vehicle platoon with the associated communication graph. The following control law and the platoon solution based on the ring coupling are used in [21]:

$$u_i = w_i + K(x_{i-1} - x_i - L_i - h\dot{x}_i) \tag{2.2}$$

where $K > 0$ is the coupling strength, $h > 0$ is the time headway and L_i ($i \in \Phi$) is a real constant. Using the control input (2.2) in (2.1) yields the following closed loop dynamics

$$\dot{x} = \frac{(Ax - Kb + w)}{m} \tag{2.3}$$

where

$$\begin{aligned}
x &= \begin{bmatrix} x_1 & \dot{x}_1 & x_2 & \dot{x}_2 & \cdots & x_N & \dot{x}_N \end{bmatrix}^T \\
A &= circ\left(\begin{bmatrix} 0 & 1 \\ -K & -(p+hK) \end{bmatrix}, O_2, O_2, \cdots, O_2, \begin{bmatrix} 0 & 0 \\ K & 0 \end{bmatrix} \right) \\
b &= \begin{bmatrix} 0 & L_1 & 0 & L_2 & \cdots & 0 & L_N \end{bmatrix}^T \\
w &= \begin{bmatrix} 0 & w_1 & 0 & w_2 & \cdots & 0 & w_n \end{bmatrix}^T
\end{aligned} \tag{2.4}$$

and O_2 denotes a 2×2 null matrix. The solution of Equation (2.3) is called the platoon solution and is given by

$$\varphi(t) = \alpha t + \beta_i, \tag{2.5}$$

where

$$\alpha = \frac{\omega_m - KL_m}{p + hK} \tag{2.6a}$$

$$\beta_{i-1} - \beta_i = \frac{\omega_i - \omega_m}{K} + L_m - L_i \tag{2.6b}$$

The value of the design parameters L_i, h and w_i are calculated based on the desired properties. The coupling strength (K) is a constant gain. For a desired inter-vehicular

distance (δ_d) and platoon velocity (υ_d), the parameter values are calculated using the relations

$$L_i = N\delta_d + L_1 \tag{2.7a}$$

$$h = -\frac{L_1 + (N-1)\delta_d}{\upsilon_d} - \frac{p}{mK} \tag{2.7b}$$

where L_1 is a parameter that is freely chosen. While deriving (2.7) it is assumed that $w_i = 0,\ \forall\ i \in \Phi$. Since we are primarily interested in a constant inter-vehicular spacing irrespective of the velocity of the platoon, h is a constant which changes only when the desired velocity (υ_d) of the platoon changes. Along with the position x_i, each vehicle should also communicate the values of N, L_1, δ_d and υ_d to the neighboring vehicle. The system matrix in Equation (2.6b) is a circulant matrix. Some important properties of the circulant matrix are discussed in the next section.

2.2.1 Circulant Matrices

Definition 2.7 *Any matrix where each successive row is obtained by cyclically right shifting the previous row by one step is called a Circulant matrix. An example of a circulant matrix is given below.*

$$C = \begin{bmatrix} c_1 & c_2 & c_3 & \cdots & c_N \\ c_N & c_1 & c_2 & \cdots & c_{N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_2 & c_3 & c_4 & \cdots & c_1 \end{bmatrix} \tag{2.8}$$

It is a special case of a broader class of matrices called the Toeplitz matrix. Appendix A provides a brief introduction to Toeplitz matrices. Circulant matrices are commonly used to approximate the behavior of Toeplitz matrices but they have a variety of uses themselves.

The most important and useful property of a circulant matrix is that only one row is enough to describe the entire matrix. This is of great significance in the analysis of systems which are described by circulant matrices. Another important property is

related to the eigenvalues. Let c_k be the sequence of the elements of the first row of the matrix C, then the eigenvalues of the matrix comprise of the coefficients of Discrete Fourier transform (DFT) of the sequence c_k. The converse is also true, i.e, the first row of the circulant matrix is the inverse DFT of its eigenvalues [25]. The following lemma summarizes the properties discussed above for a block circulant matrix.

Lemma 2.1 *Let $C \in \mathbb{R}^{Nm \times Nm}$ be a block circulant matrix*

$$C = circ\begin{pmatrix} C_1 & C_2 & C_3 & \cdots & C_N \end{pmatrix}$$

where $C_i \in \mathbb{R}^{m \times m}$ $\forall$ $i \in \Phi$. The matrix C can be block diagonalized into a matrix Λ

$$\Lambda = diag\begin{pmatrix} \Lambda_1, & \Lambda_2, & \cdots & \Lambda_N \end{pmatrix} \tag{2.9}$$

with the blocks $\Lambda_i \in \mathbb{C}^{m \times m}$ given by

$$\Lambda_i = C_1 + \omega^{i-1}C_2 + \omega^{2(i-1)}C_3 + \cdots + \omega^{(N-1)(i-1)}C_N \tag{2.10}$$

$\forall i \in \Phi$, with $\omega := e^{2\pi j/N}$.

2.2.2 Eigenvalues of the Closed-Loop System Matrix

Lemma 2.1 is used to find the eigenvalues of the system of vehicles in the platoon. Block diagonalizing the system matrix A in (2.4), each diagonal blocks is

$$A_i := \begin{bmatrix} 0 & 1 \\ K(\omega^{(N-1)(i-1)} - 1) & -(p + hK) \end{bmatrix}, \forall\ i \in \Phi \tag{2.11}$$

The eigenvalues of the matrix A are thus obtained from the matrices A_i as

$$\lambda_{i\pm} = -\frac{\gamma}{2} \pm \frac{1}{2}\sqrt{(\gamma^2 + 4K(\omega^{1-i} - 1))} \tag{2.12}$$

where $\gamma = (p + hK)$. When $i = 1$ in (2.12) the eigenvalues are zero and $-\gamma$. The zero eigenvalue corresponds to the rigid body mode and the system is thus spatially

invariant [37]. It was proved in [21] that the eigenvalues λ_i are all in the left half of the complex plane when

$$K < \frac{(p + hK)^2}{2\cos^2(\pi/N)} \tag{2.13}$$

The constant K can be selected so as to satisfy the inequality (2.13).

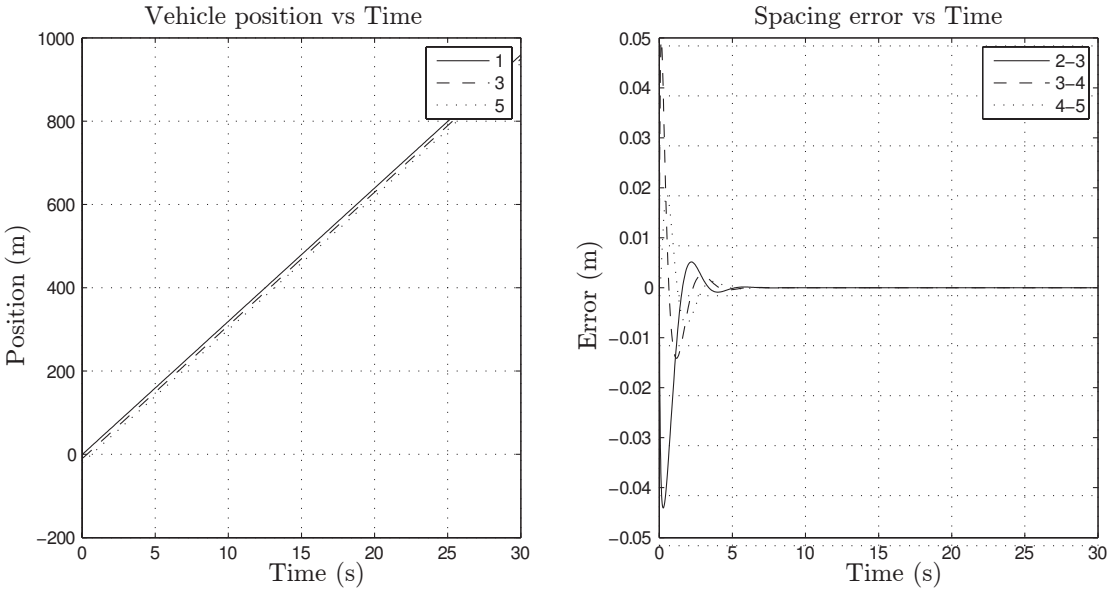


Figure 2.3: Position and spacing error of the platoon with ring graph of Figure 2.2(b)

A platoon of five vehicles is simulated with both zero and non-zero spacing errors. The desired inter-vehicular spacing of $\delta_d = 5$ m and platoon velocity $v_d = 25$ m/s are used in the simulations. The evolution of position of the odd numbered vehicles and their corresponding spacing errors is shown in Figure 2.3. It is important to choose the values of K and L_1 such that they satisfy the stability condition. To satisfy this, the coupling strength is set to a desired value first and the value of spacing constant (L_1) is varied such that time headway is positive (> 0) and the stability condition is satisfied. The parameter values for the current simulation are summarized in Table 2.1.

Variable	Value
Velocity (v_d)	25 m/s
Inter-vehicular spacing (δ_d)	5m
Coupling strength (K)	25
Spacing constant (L_1)	-40
Mass (m)	1kg
Friction coefficient (p)	0.5

Table 2.1: Parameter values used in simulation of basic ring graph

2.2.3 Scalability of Ring Graph

As explained in Section 2.1, scalability is one of the two important properties that a coordinated controller need to satisfy. In the following discussion it will be argued that the controller discussed in the beginning of this section may not be scalable. For this we show that as the size of the platoon increases, there is a group of eigenvalues whose real parts are negative but tend to the origin.

The eigenvalues of the system are given by Equation (2.12) where the complex number $\omega^{(1-i)}$ can be written as $\omega = \cos\left(\frac{2\pi(1-i)}{N}\right) + j\sin\left(\frac{2\pi(1-i)}{N}\right)$. Substituting the complex number and separating the eigenvalues into their respective real and complex parts we get

$$\lambda_{i,\pm} = -\frac{\gamma}{2} \pm \frac{1}{2}(a_i + jb_i)^{1/2} \tag{2.14}$$

where

$$a_i = \gamma^2 + 4K\cos\left(\frac{2\pi(1-i)}{N}\right) - 4K$$
$$b_i = 4K\sin\left(\frac{2\pi(1-i)}{N}\right)$$

Taking the square root of the complex number $a_i + jb_i$ and substituting it in Equation (2.14), the equation for real part of the eigenvalue is obtained as

$$\lambda_{\Re,i,\pm} = -\frac{\gamma}{2} \pm \frac{1}{2}\sqrt{\frac{|a_i + jb_i| + a_i}{2}} \tag{2.15}$$

Using the values of a_i and b_i in the above equation and simplification gives

$$\lambda_{\Re,i,\pm} = -\frac{\gamma}{2} \pm \frac{1}{2}\sqrt{\frac{\kappa + \gamma^2 + 4K(\cos(2\pi(1-i)/N) - 1)}{2}} \tag{2.16}$$

where

$\kappa = \sqrt{\gamma^4 - 16K\gamma^2 \sin^2(\pi(1-i)/N) + 64K^2 \sin^2(\pi(1-i)/N)}$.

Let the value of i be such that $\pi(1-i)/N \approx 0, \pi$. This approximation is reasonable if N is high and i is close to 0 or N. Using this approximation in the expression for κ and in Equation (2.16) the eigenvalue equation can be simplified as,

$$\begin{aligned} \lambda_{\Re,i,\pm} &= -\frac{\gamma}{2} \pm \frac{1}{2}\sqrt{\frac{\sqrt{\gamma^4} + \gamma^2}{2}} \\ &= -\frac{\gamma}{2} \pm \frac{\gamma}{2} \\ &= 0, \ -\gamma \end{aligned} \tag{2.17}$$

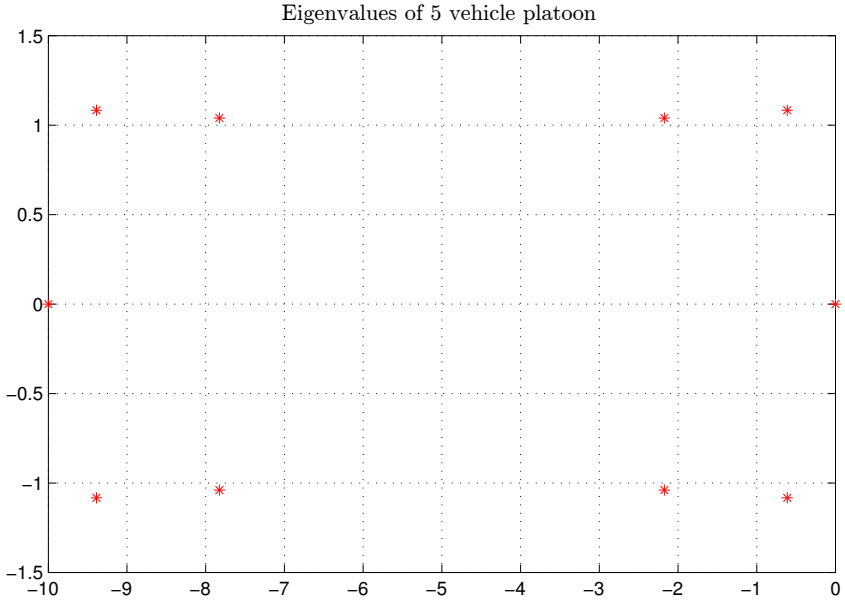
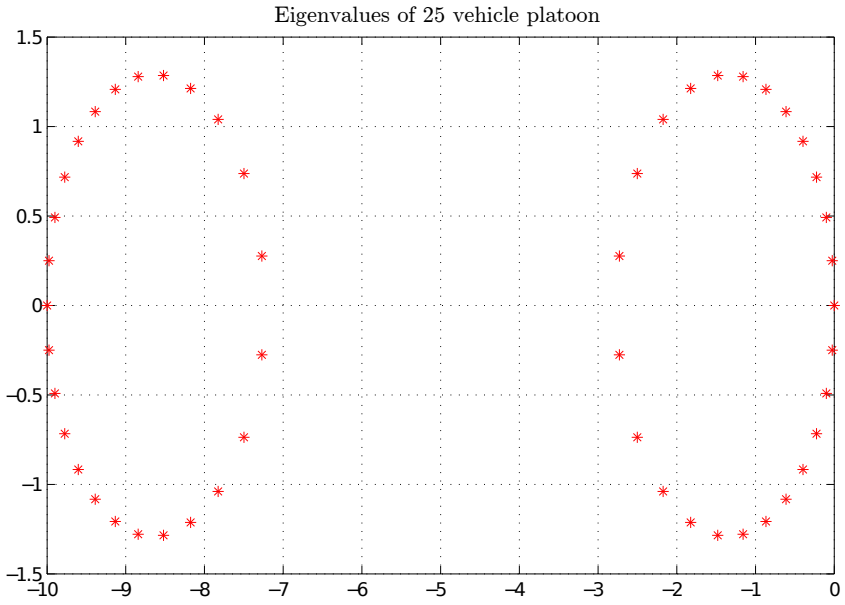
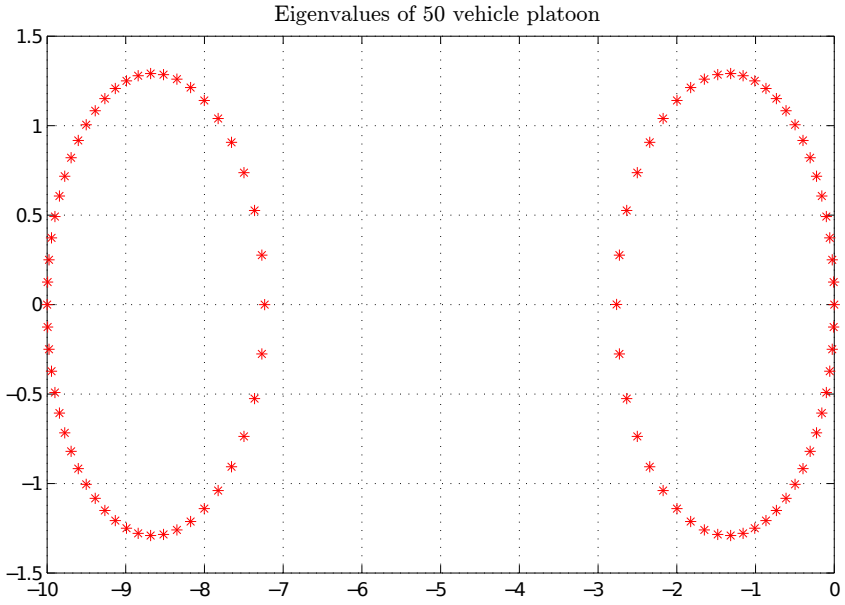


Figure 2.4: Eigenvalues of a five vehicle platoon

Thus with the increase in the number of vehicles the eigenvalues are concentrated near the points 0 and $-\gamma$. Figure 2.4 shows the eigenvalues for a platoon of five



(a)



(b)

Figure 2.5: Eigenvalues of platoons with (a) 25 and (b) 50 vehicles

vehicles with coupling strength $K = 10$ and the time headway $h = 1$. There is one eigenvalue at zero and the rest are in the left half plane. The farthest eigenvalue from the origin is at $-\gamma = -10$. It can be observed that there is a pair of complex conjugate pair above and below ≈ -0.7 on real axis.

The position of eigenvalues for vehicle platoons with 25 and 50 vehicles with same coupling strength and time headway as that of a five vehicle platoon are shown in Figures 2.5 (a) and (b). It can be seen that although the eigenvalue at 0 and -10 remain at the same location, there are multiple complex conjugate pairs of eigenvalues near zero instead of a single pair as in the five vehicles case. The two groups of eigenvalue that are symmetrical about the line passing through $-\gamma/2$. Changing the value of coupling strength merely moves the second group (since the group is formed near $-\gamma$ which changes with K and h). Therefore, as the size of the platoon increases, low frequency sinusoidal disturbances on the platoon are sustained and propagated. If the frequency of the sinusoidal disturbance is same as the resonant frequency due to a pair of complex conjugate eigenvalues whose damping ratio is less than 0.5, then these disturbances are amplified.

2.3 Limitations of Basic Ring Graph

The ring graph shown in Figure 2.2 suffers from feasibility issues such as the size (N) of the platoon which may be a major impediment not only because of the scalability but also because of the link between the first and the last vehicle. Communication may suffer if the length of the platoon is greater than the communication range of the sensing device, if there is an obstacle between the vehicles, etc. So an alternative ring graph needs to be in place which does not involve such long links. Development of such a graph, its behavior based on the controller parameters set in the previous section and stability are discussed in this section. It is beneficial to communicate with the nearest possible neighbor, that is to keep the edge length of the graph as

small as possible for all edges.

2.3.1 Design of Small Platoons with Ring Structure

One method of keeping the edge length small is by having small platoons. Any large platoon can be split into multiple smaller platoon which is equivalent of saying a large ring can be split into a number of small rings as shown in Figure 2.6.

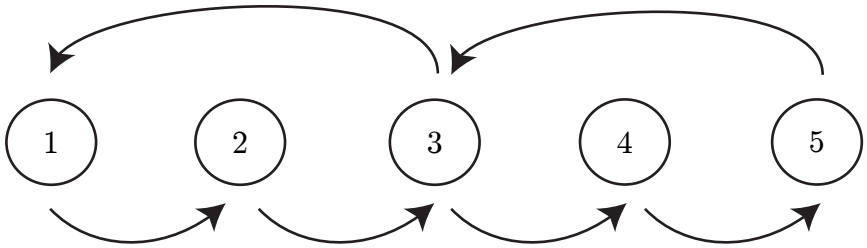


Figure 2.6: Platoon of five vehicles with two small rings. $1-2-3-1$ and $3-4-5-3$

In order to maintain the platoon objectives, there has to be information flow between each individual platoon. This can be achieved either by having a common vehicle between two successive platoons or by having one vehicle in each platoon communicating with any one vehicle in the neighboring platoon. However in both the cases the system matrix ceases to be circulant as the rows corresponding to the common vehicles have an additional non-zero block. The matrix in (2.18) represents the system matrix of a 5 vehicle platoon with two small rings with vehicle 3 as the common vehicle. It can be seen that row 3 has an additional non-zero element. Therefore, we cannot apply the properties of the circulant matrix for system analysis. The system may or may not be stable but its discussion is out of the scope of this

work.

$$A = \begin{bmatrix} M & O_2 & M_1 & O_2 & O_2 \\ M_1 & M & O_2 & O_2 & O_2 \\ O_2 & M_1 & M & O_2 & M_1 \\ O_2 & O_2 & M_1 & M & O_2 \\ O_2 & O_2 & O_2 & M_1 & M \end{bmatrix} \tag{2.18}$$

where

$$M = \begin{bmatrix} 0 & 1 \\ -K & -(p + hK) \end{bmatrix} \quad \text{and} \quad M_1 = \begin{bmatrix} 0 & 0 \\ K & 0 \end{bmatrix}$$

2.3.2 Alternative Ring Graphs

Alternative ring graphs are obtained by rearranging the position of the vehicles in the ring. It should be noted that only the information flow in the ring has been changed but not the physical position of the vehicle in the platoon. Figure 2.7 shows a possible alternative configuration of a platoon with five vehicles. It can be seen that instead of a link between i^{th} and $(i-1)^{th}$ vehicle, now the communication is between i^{th} and $(i-2)^{th}$ vehicle.

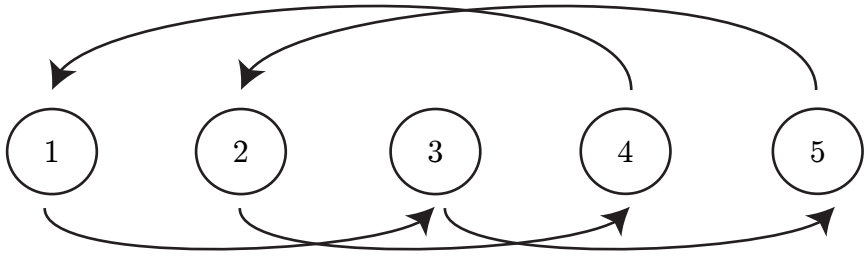


Figure 2.7: Alternate ring graph with $i-2$ to i i.e., 1-3-5-2-4-1

Consider the double integrator dynamics from (2.1). Using the ring graph from Figure 2.7 the control input u can be written as,

$$u_i = \omega_i + K(x_{i-2} - x_i - L_i - h\dot{x}_i) \tag{2.19}$$

This control input leads to the following closed loop dynamics

$$\dot{x} = Ax - Kb + \omega \tag{2.20}$$

where

$$\begin{aligned} x &= \begin{bmatrix} x_1 & \dot{x}_1 & x_2 & \dot{x}_2 & \cdots & x_N & \dot{x}_N \end{bmatrix}^T \\ A &= circ\left(\begin{bmatrix} 0 & 1 \\ -K & -(p+hK) \end{bmatrix}, O_2, O_2 \begin{bmatrix} 0 & 0 \\ K & 0 \end{bmatrix}, O_2, \right) \\ b &= \begin{bmatrix} 0 & L_1 & 0 & L_2 & \cdots & 0 & L_N \end{bmatrix}^T \\ w &= \begin{bmatrix} 0 & w_1 & 0 & w_2 & \cdots & 0 & w_N \end{bmatrix}^T \end{aligned} \tag{2.21}$$

and K, L_i and h all have the same definition as before. Let the solution of (2.20) is exactly the same as (2.5). Substituting the platoon solution in (2.1) and solving for α and β_i we get

$$\alpha = \frac{\omega_m - KL_m}{p + hK} \tag{2.22a}$$

$$\beta_{i-1} - \beta_i = \frac{\omega_i - \omega_m}{K} + L_m - L_i \tag{2.22b}$$

Comparing the above equations (2.22) with the corresponding α and β_i of the basic ring graph (2.6), it is clear that both are the same. In fact it can be shown that no matter how the vehicles are connected in a vehicle platoon as long as the information flow graph forms a ring, the resultant system matrix is always circulant and the platoon solution is given by (2.22). It is also noted that all the platoon solutions only differ in the absolute position of the platoon and hence the system is spatially invariant, and as a consequence of the spatial invariance, the system has one zero eigenvalue.

Stability of Alternate Ring Graphs

Eigenvalues of the alternative ring structures are analyzed to determine the condition for stability. Since the closed loop system matrix is circulant, we use Lemma 2.1 to

write it in the block diagonal form. Considering the ring from Figure 2.7 the system matrix of the platoon is given by (2.4). Using the Lemma on matrix A each block on the diagonal is given by

$$A_i = \begin{bmatrix} 0 & 1 \\ K(\omega^{2(1-i)} - 1) & -(p + hK) \end{bmatrix} \tag{2.23}$$

The eigenvalues of the system are the union of the eigenvalues of the matrices given by (2.23) for all i. The eigenvalues thus obtained are

$$\lambda_{i,\pm} = -\frac{p + hK}{2} \pm \frac{\sqrt{(p + hK)^2 + 4K(\omega^{2(1-i)} - 1)}}{2} \tag{2.24}$$

Substituting $i = 1$ yields the zero and the smallest eigenvalues of the system. The system is stable if the real part of the eigenvalues is negative, i.e,

$$\frac{\gamma}{2} \pm \frac{1}{2}\Re(a_i + jb_i)^{1/2} < 0 \tag{2.25}$$

where $\gamma = (p+hK)$, $a_i = \gamma^2 + 4K\cos(4\pi(i-1)/N) - 4K$ and $b_i = 4K\sin(4\pi(i-1)/N)$. From Equation (2.25) we conclude that the system is asymptotically stable if

$$K < \min \frac{\gamma^2}{2\cos^2\left(\frac{2\pi(i-1)}{N}\right)} = \frac{\gamma^2}{2\cos^2\left(\frac{2\pi}{N}\right)} \tag{2.26}$$

Comparing the above stability condition with the condition from previous section we see that both are similar but not exactly the same.

Thus for any number of vehicles N, a coupling strength K and given conditions of travel (velocity and inter-vehicular distance), the stability depends only on the value of 'h' which is the time headway. From the Equation (2.7) time headway depends on the structure of the ring, that is the inter-vehicular spacing between successive members of the ring. Since δ_m is always positive and is less than the magnitude of L_1 chosen, the value of h is always strictly positive. Therefore, the eigenvalues are always in the negative left half plane as long as the stability condition is satisfied. However, the problem of scalability still persists.

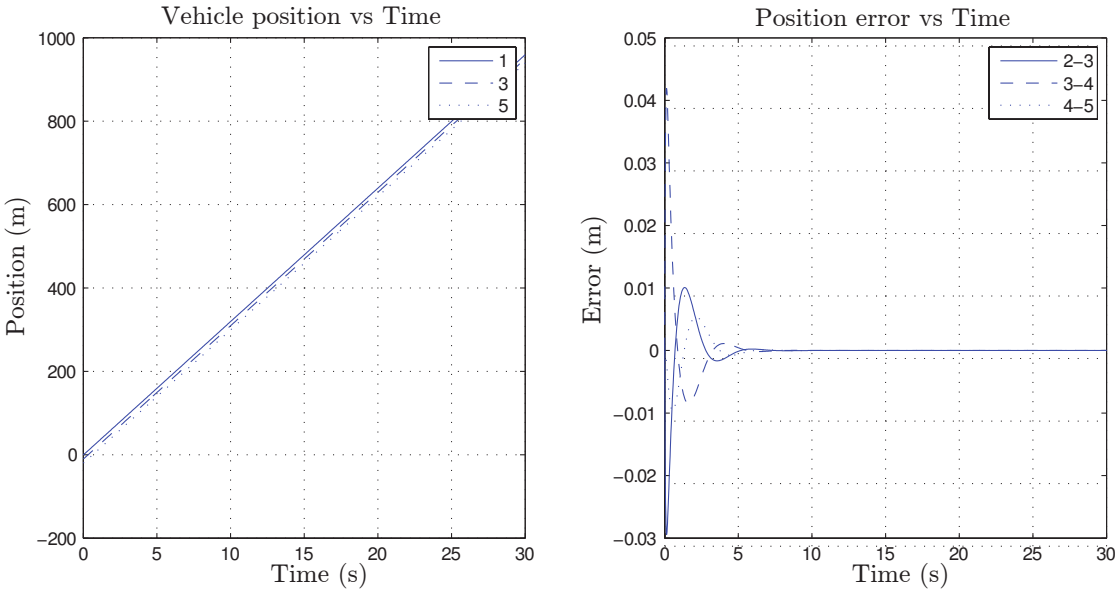


Figure 2.8: Position and spacing error of graph in Figure(2.7)

Simulation results for the five vehicle platoon with alternate graph configuration are shown in Figure 2.8. The values of coupling strength K and L_1 are chosen as in Table 2.1. The other conditions of simulation also remain the same. However since the position of vehicles with respect to the ring is different, inter-vehicular spacing of adjacent vehicles in ring is different and hence L_i and time headway values are different.

2.4 Multiple Vehicle Communication

The previous sections are based on communication from a single vehicle, i.e, each vehicle i receives the information of only one vehicle. But a controller based on communication from a single vehicle can lead to problems in some situations. For instance if one of the communication links is broken, then there is collision among the following vehicles or the entire platoons comes to a halt. This problem can be averted by using a two vehicle communication instead of one. Figure 2.9 shows a platoon where each vehicle communicates with two of its immediate neighbors downstream. Notice that if each vehicle receives information from the vehicle on either side then

the communication graph is symmetric.

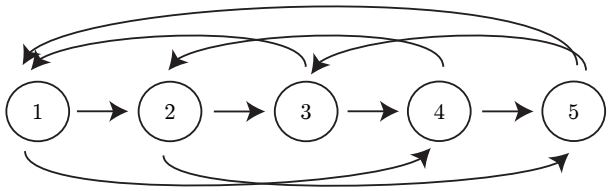


Figure 2.9: Vehicle platoon with each vehicle receiving communication from multiple vehicles. Ring1: 1-2-3-4-5-1, Ring2: 1-4-2-5-3-1

The platoon dynamics for this situation are,

$$\dot{x} = Ax - (K_1 b_1 + K_2 b_2) + \omega \tag{2.27}$$

$$A = circ\left(\begin{bmatrix} 0 & 1 \\ -(K_1+K_2) & -(p+h_1K_1+h_2K_2) \end{bmatrix}, O_2, O_2, \begin{bmatrix} 0 & 0 \\ K_1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ K_2 & 0 \end{bmatrix}\right)$$

$$b_1 = \begin{bmatrix} 0 & L_{11} & 0 & L_{12} & \cdots & 0 & L_{1N} \end{bmatrix}^T$$

$$b_2 = \begin{bmatrix} 0 & L_{21} & 0 & L_{22} & \cdots & 0 & L_{2N} \end{bmatrix}^T \tag{2.28}$$

Here K_1 and K_2 are the coupling strengths between the i^{th} vehicle and the two neighbors it is communicating with. With δ_{i1} and δ_{i2} being distance between i^{th} vehicle and its communication neighbors, the values of L_{i1} and L_{i2} and h_1, h_2 are calculated from (2.7).

Equation (2.27) is simulated for a platoon of five vehicles with gains $K_1 = 25$ and $K_2 = 5$. The resulting path and error of the each vehicle is plotted in Figure 2.10. Observing the error plots in Figures 2.3 and 2.8 it can be seen the settling time and overshoot of error have improved compared to that of the previous case with single neighbor communication. Hence, a two neighbor is better than a single neighbor communication in terms of error and collision avoidance; of course we have to keep in mind that this is true for only ring graphs.

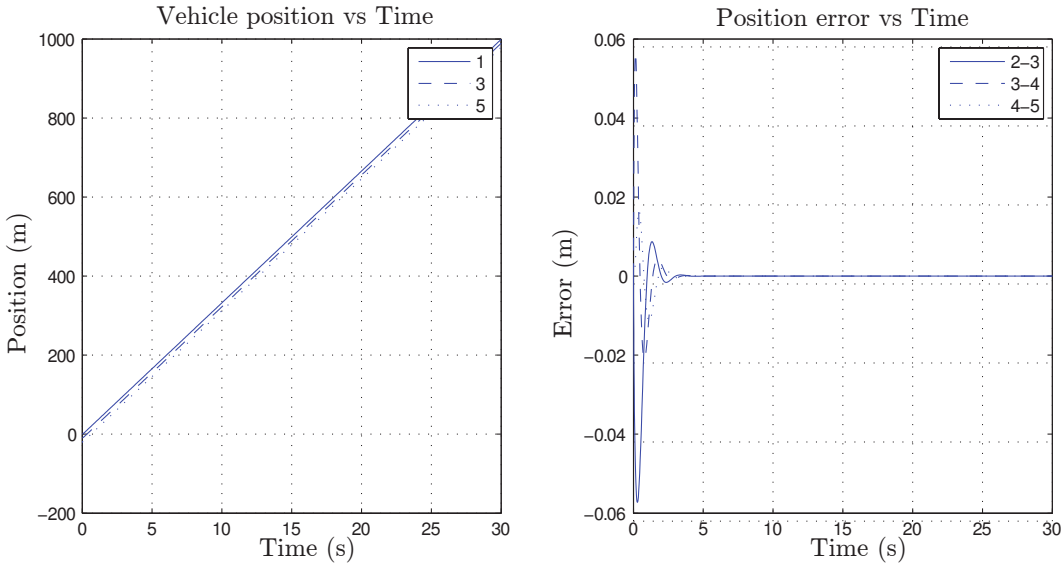


Figure 2.10: Position and error vs time for a platoon with each vehicle receiving information from two vehicles

2.4.1 Stability Analysis

Using the eigenvalue analysis discussed in Section 2.2 with the present case of two vehicle communication. Using the Lemma 2.1 on Equation (2.27) the diagonal matrices and the respective eigenvalues are

$$\Lambda_i = \begin{bmatrix} 0 & 1 \\ K_1(\omega^{2(1-i)} - 1) + K_2(\omega^{(1-i)} - 1) & -\gamma \end{bmatrix}$$

$$\lambda_{i,\pm} = -\frac{\gamma}{2} \pm \frac{1}{2}\sqrt{\gamma^2 + 4\left(K_1(\omega^{2(1-i)} - 1) + K_2(\omega^{(1-i)-1})\right)} \tag{2.29}$$

For stability the real part of the above eigenvalues should be in the left half of the complex plane. Solving the inequality we finally obtain the condition for stability with two vehicle feedback

$$(K_1 + K_2) > \max\left[\left(\frac{K_1 \sin\frac{4\pi(1-i)}{N} + K_2 \sin\frac{2\pi(1-i)}{N}}{4\gamma}\right)^2 + K_1 \cos\frac{4\pi(1-i)}{N} + K_2 \cos\frac{2\pi(1-i)}{N}\right] \tag{2.30}$$

Hence, as long as the sum of coupling strengths satisfies Equation (2.30) the platoon is stable. However, it is noted that this may not guarantee string stability. It should also be noted that there are symmetric links in the two vehicle communication graph shown in Figure 2.9.

2.5 Two Dimensional platoons

So far the characteristics of ring coupling for platoons moving in a single direction have been investigated. Single dimension travel would mean that the path has to be straight without any turns or curves. The path should be free of any obstacles because if there are any, the vehicles would either collide or would come to a stop (the later being possible if any avoidance detection system is in place). But this is practically not possible as any highway would have turns and there are also disturbances in the path. In order to overcome these issues the vehicle should be steerable which in turn would mean that the platoon should be steerable making the path two dimensional (2D). In some special cases, for example the differential drive model, the higher order dynamics of the vehicle can be decoupled into individual directions [38].

A simple 2D model is given by Equation (2.31) where directions x and y are decoupled. Although this is practically not possible always and the directions are coupled in many cases, the model can be used for a basic analysis to see if the ring coupling would actually work with 2D travel.

$$\begin{aligned} \ddot{x} + p_x \dot{x} &= u_x \\ \ddot{y} + p_y \dot{y} &= u_y \end{aligned} \tag{2.31}$$

where p_x, p_y is the friction in respective directions and u_x,u_y the inputs. Simulating the above dynamics using control input (2.33) for a square platoon of the form in Figure 2.11 with zero initial conditions gives the error plot shown in Figure 2.12.

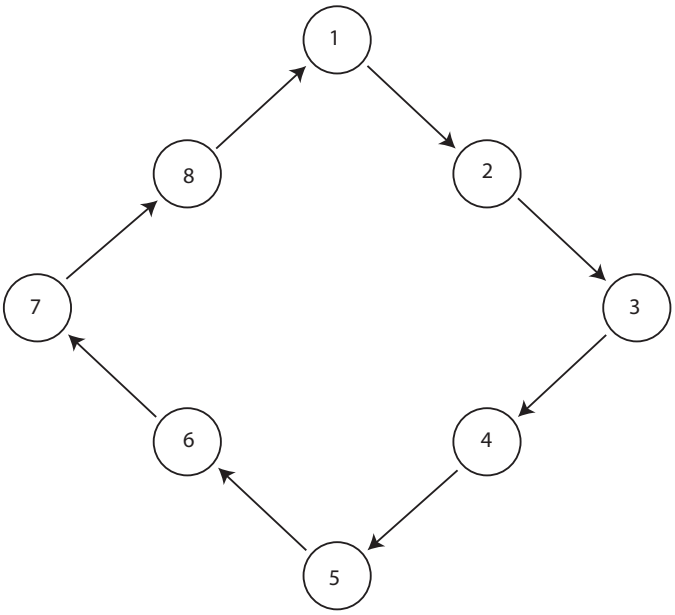


Figure 2.11: Two dimensional square formation

$$u_x = K_x(x_{i-1} - x_i - L_{x_i} - h_x\dot{x}_i) \tag{2.32}$$

$$u_x = K_y(y_{i-1} - y_i - L_{y_i} - h_y\dot{y}_i) \tag{2.33}$$

where, L, h, K represent the spacing constant, time headway and coupling strength in the respective dimension. This control input will result in the following closed loop system which is similar to the system in (2.20)

$$\dot{x} = Ax - Kb + \omega \tag{2.34}$$

where,

$$x = \begin{bmatrix} x_1 & y_1 & \dot{x}_1 & \dot{y}_1 & \cdots & x_N & y_N & \dot{x}_N & \dot{y}_N \end{bmatrix}^T$$

$$A = circ\left(\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -K_x & 0 & -(p_x + h_xK_x) & 0 \\ 0 & -K_y & 0 & -(p_y + h_yK_y) \end{bmatrix}, O_4, O_4 \cdots \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ K_x & 0 & 0 & 0 \\ 0 & K_y & 0 & 0 \end{bmatrix}\right)$$

$$b = \begin{bmatrix} 0 & 0 & L_{x_1} & L_{y_1} & \cdots & 0 & 0 & L_{N_x} & L_{N_y} \end{bmatrix}^T \tag{2.35}$$

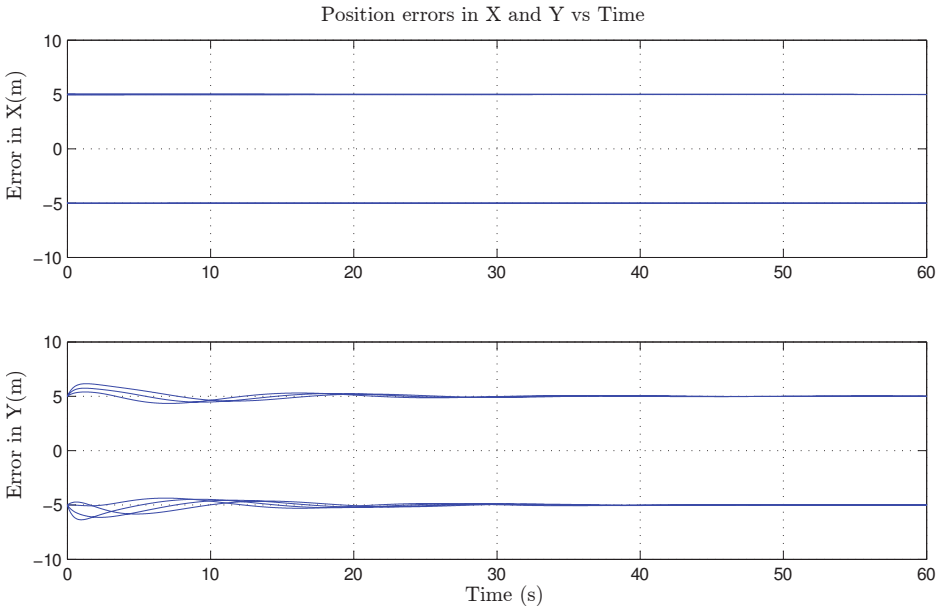


Figure 2.12: Error vs time for 2D square formation with decoupled dynamics

Since the dynamics are decoupled each direction is similar to 1D linear platoon model and hence is stable as long as the conditions in Equation (2.13) is satisfied.

CHAPTER 3

ALGORITHM TO FORM A RING GRAPH

The vehicle platoons discussed in the previous chapters are all small in size and may not be suitable for many practical applications. For a large formation, it is difficult to create a directed communication graph other than few simple graphs such as a path graph. Hence, it is important to have a strategy for creating a direct communication graph. In this chapter two possible algorithms to find a ring graph for a given formation are discussed. Section 3.1 discusses an algorithm to find a ring graph for a given platoon of vehicles. In section 3.2, the Traveling Salesman Problem (TSP) is introduced and the methods used to solve the TSP are explored. Section 3.3 the ring graph is formulated as a special case of TSP. An algorithm based on the branch and bound method is formulated to form a ring graph for a given vehicle formation. The details of this algorithm are presented in section 3.4.

3.1 Ring Graph for a Vehicle Platoon

As discussed in Chapter 2 a platoon is a linear formation of vehicles maintaining an inter-vehicular spacing and a constant speed. A platoon is one of the simplest formations even if it involves a large number of vehicles. The position of vehicles can be joined using a straight line or a curve. Forming a ring graph on a platoon of vehicles is also easy relative to other more complex formations, that require a lot of computations and time taking computer programs.

An algorithm to form a ring graph for a given vehicle platoon consisting of N vehicles, with a constant inter-vehicular spacing δ, and a communication range $\xi \ni$

$\xi \geq 2\delta$, is the following:

1. Denote the first vehicle by setting $i = 1$, the second vehicle $i = 2$, and so on till the last vehicle $i = N$.

2. Starting from the first vehicle, establish edges between successive odd numbered vehicles, that is one to three, three to five, five to seven and so on.

3. Repeat last step until N^{th} vehicle if N is odd or, $(N-1)^{th}$ vehicle if N is even.

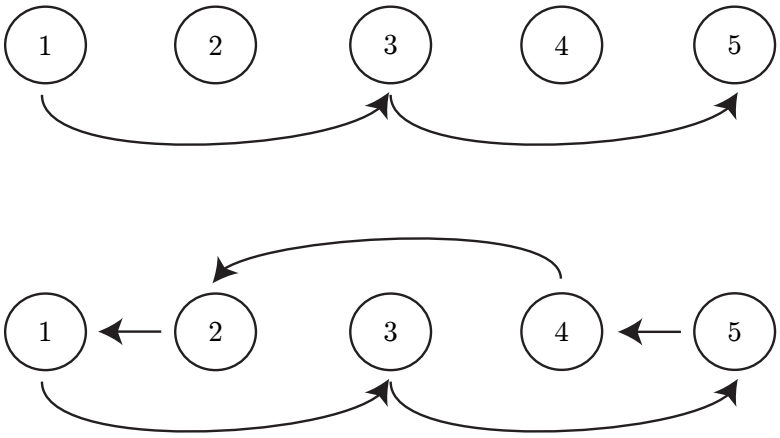
4. If N is odd, create an edge from N^{th} vehicle to $(N-1)^{th}$ vehicle or, if N is even create an edge from $(N-1)^{th}$ vehicle to the N^{th} vehicle.

5. Beginning from the vehicle in last step, move towards the front of platoon and create an edge between every successive even numbered vehicles until vehicle two.

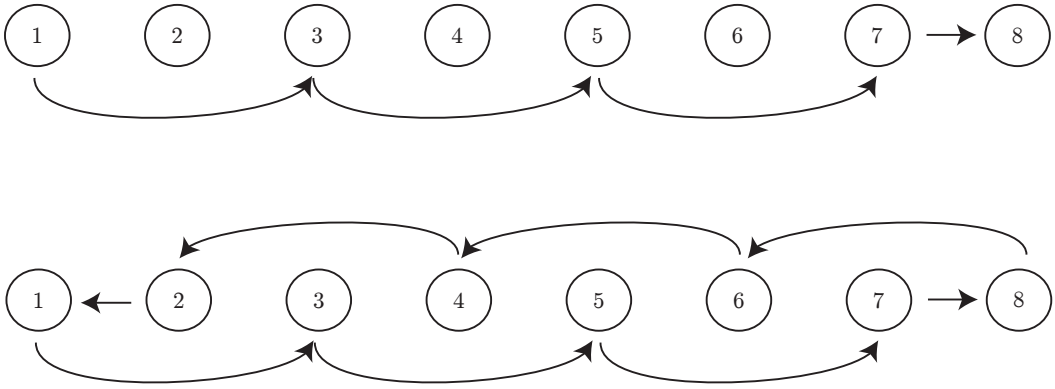
6. Finally create an edge from vehicle two to vehicle one.

The above algorithm gives a ring graph with the length of the longest edge less than or equal to the communication range since the longest distance two successive vehicles is less than or equal to twice the inter-vehicular spacing in the ideal situation where there are no spacing errors. Figure 3.1 shows this algorithm in action for a five vehicle and eight vehicle platoon.

Although this algorithm is simple to use, it is only suited for platoons, i.e., linear formations. It is unclear if the same approach will work with formations that are two- and three-dimensional. For instance, intuitively the algorithm does not yield a feasible solution with a two dimensional triangle formation or square formation. In the next section of this chapter the TSP formulation and related solution methods are used to create an algorithm which will work for any kind of formation.



(a)



(b)

Figure 3.1: Illustration of the algorithm for a platoon of (a) Five vehicles (b) Eight vehicles

3.2 Traveling Salesman Problem

Definition 3.1 *Given a list of n cities and their pairwise distances, find a tour, that is, a simple path visiting all cities, so that the total traversed distance is the least possible [39]*

The TSP has many application in planning, scheduling tasks, manufacture of integrated circuits, geographical mapping, etc. Considered one of the toughest problems to solve in combinatorial optimization and classified as NP-hard, it is also used for testing the efficiency of any algorithm developed to solve combinatorial optimization

problems. There is a vast amount of literature on TSP formulations, special forms of TSP, and polynomial-time algorithms to solve special cases. The reader is referred to [40, 41] for more information.

In graph theory given a completely weighted graph the problem of finding a Hamiltonian Cycle such that the total weight is the smallest is a TSP formulation. There are two types of weighted graphs: Asymmetric graphs, where the distance from node i to j is not equal to the distance from node j to i. For example, in traffic related problems when one way streets are encountered, distance traveled from point A to B is not the same as that from point B to A. Symmetric graphs, where the distance between any two nodes are equal both ways.

The formulation of a standard TSP problem is the following. We start with a cost matrix $C = c_{ij}$ where each element c_{ij} represents the distance between city i and j, the problem is to minimize

$$\sum_{i=0}^{n} \sum_{j=0}^{n} c_{ij} x_{ij} \tag{3.1}$$

where x_{ij} is how the traveler goes from city i to j. $x_{ij} = 1$ if person goes immediately from i to j. Equation (3.1) is minimized subjected to constraints

$$\sum_{i=0}^{n} x_{ij} = 1, \quad \sum_{j=0}^{n} x_{ij} = 1 \tag{3.2}$$

The constraints in the above equation limits the tours where traveler goes from city i to only one city and comes to city j from only one city. Thus the formulation so far gives us tours where the traveler visits all the cities while visiting each city only once. This is called an assignment problem or scheduling problem. But the assignment problem may include sub-tours. Figure 3.2 shows a possible solution of a five city assignment problem. There are two sub-tours in the solution, yet all the cities are assigned once. Hence for a TSP it is obvious that these sub-tours should be eliminated.

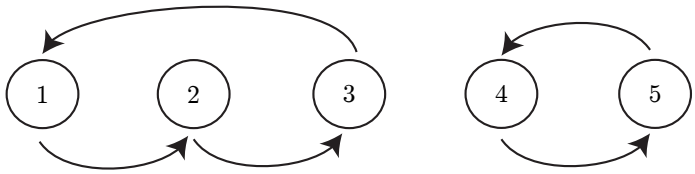


Figure 3.2: Possible solution of assignment problem with five cities

3.2.1 Sub-Tour Elimination Constraint

A tour of length less than the total number of nodes is called a sub-tour. To eliminate sub-tours we have to eliminate tours of length $1, 2, ...(n-1)$. The tours of length 1 are the tours which start and end at the same node or city. To eliminate this we can include the following condition, $x_{jj} = 0,\ \forall\ j = [1, n]$. This will give an extra n constraints. We can avoid this by including the constraint directly into the cost matrix by simply setting $c_{jj} = \infty$. By this the sub-tours of length 1 are automatically eliminated from any solution since they cost infinity.

Sub-tours of length 2 are eliminated by setting the constraint $x_{ij} + x_{ji} \leq 1$. This gives an additional ${}^nC_2$ constraints. Sub-tours of length 3 are eliminated by setting $x_{ij} + x_{jk} + x_{ki} \leq 2$ which gives a total of ${}^nC_3$ constraints. Writing the conditions for all the other sub-tours we will get

$$\begin{aligned} {}^nC_2 +^n C_3 + ... +^n C_{\frac{n(n-1)}{2}} \qquad & \text{constraints if, n is odd} \\ {}^nC_2 +^n C_3 + ... +^n C_{\frac{n}{2}} \qquad & \text{constraints if, n is even} \end{aligned} \tag{3.3}$$

One can see that many constraints need to be included in the problem. Another formulation of the sub-tour elimination constraints also known as Miller-Tucker-Zemlin (MTZ) formulation [42] is given by the inequality (3.4).

$$u_i - u_j + nx_{ij} \leq (n-1) \quad \forall i = [1, n-1], j = [2, n] \tag{3.4}$$

where u_i $(i = 1, 2, 3, \cdots, n)$ are arbitrary real numbers. Every tour satisfies this constraint including the complete tour of all n cities. This formulation reduces the total number of sub-tour constraints to n^2 from the order of $n!$. Thus the Equation

(3.1) is minimized subject to constraints (3.2) and (3.4). This kind of formulation is used in Integer Programming methods, which was popular during the early years of TSP literature [41, 42]. But TSP is seldom solved using integer programming today. However, it is discussed here because it is easy to understand how the problem is constructed. Many algorithms have been developed over the period to solve the TSP some of which are approximate algorithms and others are exact algorithms based on branch and bound techniques [40, 43, 44].

3.2.2 Exact Algorithms

These algorithms find the exact solution of a given TSP. Dynamic programming and branch and bound methods are commonly employed in early algorithms. These algorithms can be used to solve TSP's consisting of cities less than 100; [40,41,43] are good references for a variety of exact algorithms. Although dynamic programming is simpler of all the methods, it requires considerable computation and is very slow when the number of cities increases beyond 10. For this reason branch and bound methods are preferred to find an exact solution. The upper bounds are usually generated by solving a heuristic algorithm. There are many kinds of heuristic algorithms which generate an approximate solution. The cost of this solution is used as an upper bound.

Branch and Bound Algorithms

In a nutshell the branching starts with a lower bound and at each step select the minimum cost and branch further from that node. The essential parts of this method are a branching strategy and selection of bounds which vary with the algorithm. A rudimentary branch and bound uses the sum of row minimum to generate the bounds and the branching is done from the node which has the least cost.

Some algorithms use the cost of the tour generated by solving the assignment

problem. The assignment problem may be solved by any available algorithms like the Hungarian algorithm [45]. The cost of tour is used as the lower bound and branches are extended towards eliminating the sub-tours present in the assignment. The node which results in the minimum cost is selected and the assignment problem is solved again at that node.

Example 3.1 *Let C be a cost matrix of a five city problem and is given by*

$$\begin{bmatrix} \infty & 10 & 8 & 9 & 7 \\ 10 & \infty & 10 & 5 & 6 \\ 8 & 10 & \infty & 8 & 9 \\ 9 & 5 & 8 & \infty & 6 \\ 7 & 6 & 9 & 6 & \infty \end{bmatrix}$$

then the resultant tour and the optimum cost using the Eastman's branch and bound algorithm are $Tour = 1-3-4-2-5-1$, $Cost = 34$. The evolution of branches is shown in Figure 3.3. The number on the branch is the path and the value in the node is the minimum cost after reaching that node.

Advantages and Disadvantages

The advantages and disadvantages discussed are more suited for branch and bound methods and should not be taken for granted for the advanced methods. The biggest advantage of using the exact algorithms is of-course that they give the exact solution if it exists. The disadvantages are as follows:

1. The exact algorithms are usually very slow for large number of nodes.
2. Calculating lower bounds is difficult in many cases since at each instance the assignment problem has to be solved to get a lower bound.
3. These algorithms are usually not executed in polynomial times.

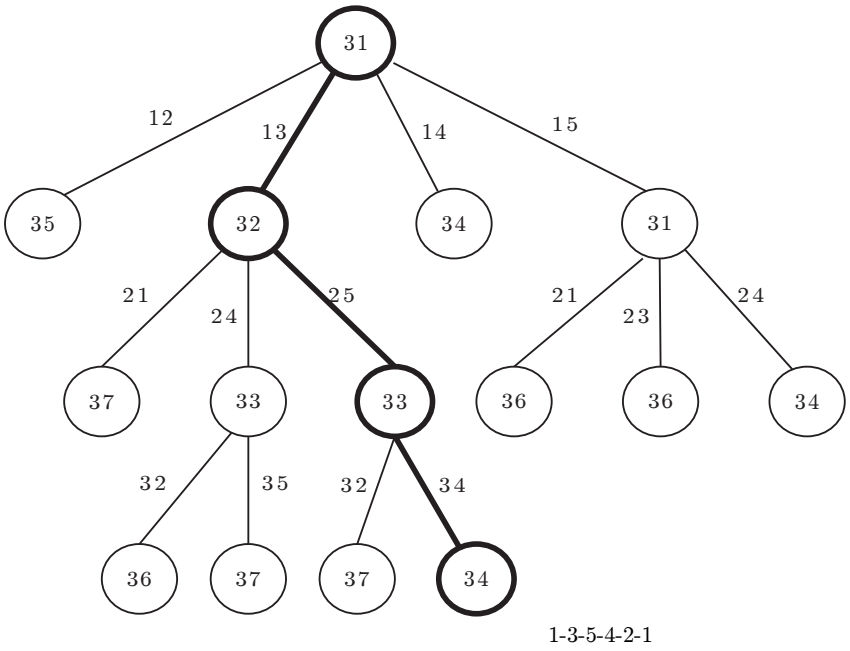


Figure 3.3: Working of branch and bound algorithm

4. They are associated with large computational costs.

Although these algorithms suffer from the above mentioned disadvantages, they are used widely if an exact solution is desired.

3.2.3 Approximation Algorithms

Approximation algorithms for solving the TSP are based on heuristic methods. They are used to solve a TSP in polynomial time. These algorithms do not give an exact solution but they give an upper bound on the solution. There are many approximation algorithms which are used depending on the quality of solution desired. References [40, 41, 46, 47] explain some of the heuristic methods, the quality of solution for each of the methods, and the approximate time required to run the algorithm for a given number of cities n. Some typical approximate algorithms are nearest neighbor algorithm or greedy algorithm, Twice Around the Tree algorithm, Christofide's algorithm, etc. Twice Around the Tree algorithm and Christofide's algorithm are both based on the concept of the minimum spanning tree (MST). MST algorithms can only

be used by cost matrices that obey triangle inequality. MST is defined as follows:

Definition 3.2 *Given a connected, undirected graph, a minimum spanning tree (MST) is a spanning tree with weight less than or equal to the weight of every other spanning tree.*

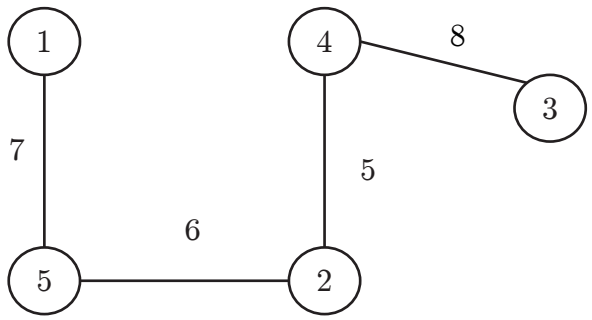


Figure 3.4: MST of cost matrix from Example 3.1

Given a connected and undirected graph the minimum spanning tree is the tree which connects all the nodes together. Figure 3.4 shows the minimum spanning tree of the cost matrix from Example 3.1. An algorithm based on the minimum spanning tree known as the Twice Around the Tree algorithm is

1. Get the minimum spanning tree (MST) of a given graph using any algorithm.
2. Starting from any arbitrary vertex or node, perform a nearest neighbor traversal of the MST and add each vertex visited to a list.
3. Start from the first vertex in the list and remove every vertex that has been previously visited until the first vertex is reached.

The step by step results from the algorithm are shown in Figure 3.5.

Advantages and Disadvantages

The approximation algorithms have several advantages over the exact algorithms.

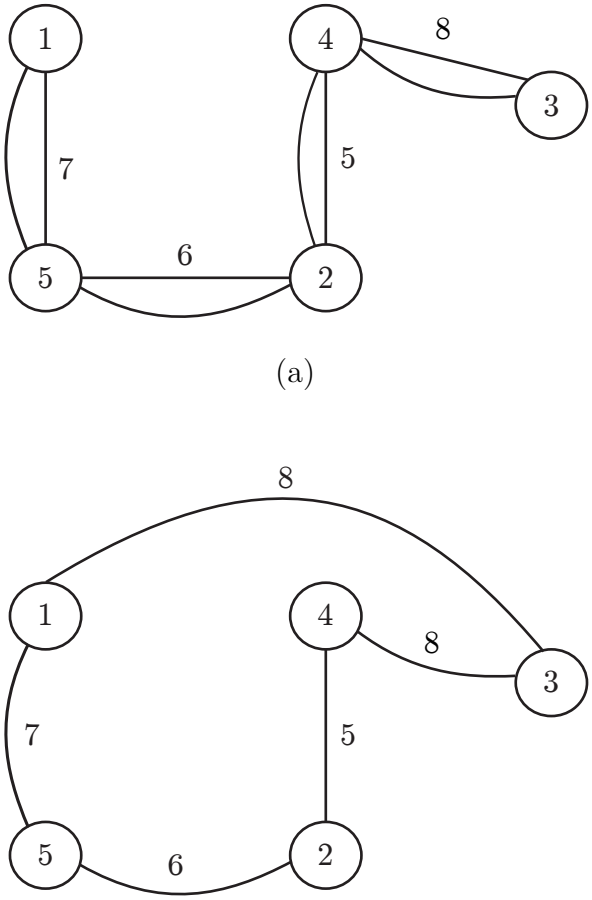


Figure 3.5: Steps in Twice Around the Tree algorithm. (a) Nearest neighbor traversal of MST, (b) The actual tour of cost 34 obtained from (a).

1. The algorithms are very fast and are executed in polynomial time.

2. A variety of heuristics exist for any type of problem.

3. In few special cases, the approximation algorithms may give the exact solution.

4. Computational costs are very small compared to exact algorithms.

5. These algorithms provide at least an upper bound to be used with exact algorithms.

Generally the chance of being close to the optimum varies based on a wide range of characteristics. The approximate algorithms are also widely used when a quick

solution is required.

3.3 Ring Graph as a TSP

As explained in Section 3.2 the TSP in graph theory can be formulated as finding the least Hamiltonian path. The Hamiltonian path is defined as follows

Definition 3.3 *Given a connected, undirected graph $G(V, E)$, a path which passes through all the vertices in the graph once and returns to the vertex from which it started is called a Hamiltonian path.*

Only connected graphs have a Hamiltonian path. If a graph is connected but not undirected there may not be a Hamiltonian path. Any algorithm that is used to solve a TSP can also be used to find a Hamiltonian path at no extra cost. Note that a Hamiltonian cycle defined in 2.4 is the same as in Definition 3.3.

Comparing Definition 2.6 of ring graph with that of Hamiltonian path in Definition 3.3 it is clear that finding a Hamiltonian path is equivalent to finding a ring graph. Thus finding a ring graph can be seen as a relaxed form of solving a TSP, i.e, finding a feasible solution. But a feasible solution may not satisfy the constraint on communication range of the transmitters. Thus there is a need for tighter bounds on this problem than the TSP itself.

3.3.1 Constraint Introduced by Transmitter Range

The range of a transmitter corresponds to the greatest distance it can transmit. Thus, a range of ξ will represent the greatest distance a tour can be between two cities or the tours where all the edges have a length less than ξ. Hence, we are looking for solutions where

$$c_{ij}x_{ij} < \xi, \quad \forall\ i, j \in [1, n] \tag{3.5}$$

This constraint makes sure that each traverse between two cities in the tour should be less than ξ. This is a total of n^2 additional constraints that need to be satisfied. Hence the problem can be formulated as minimizing Equation (3.1) subjected to constraints (3.2), (3.4) and (3.5), that is,

$$\begin{aligned} \min & \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \\ \text{subjected to} & \sum_{i=1}^{n} x_{ij} = 1, \\ & \sum_{j=1}^{n} x_{ij} = 1, \\ & u_i - u_j + nx_{ij} \leq (n-1), \\ & c_{ij}x_{ij} < \xi, \quad \forall i,j \in [1,n] \end{aligned} \tag{3.6}$$

It should be noted that even the optimum solution of TSP may not satisfy the range constraint. It can also be seen that for a given number of nodes or cities n, the time required to solve the problem is smaller than that of solving the TSP itself. This is due to two reasons:

- The communication range constraint (3.5) limits the number of branches at each step.
- The algorithm does not look for an optimum solution but stops as soon as a feasible solution is found since we do not need an optimum solution, i.e, we only need a Hamiltonian path but it need not be of minimum weight.

Finally, we make the following assumptions which are true for most of the formations

1. The cost matrix is symmetric
2. The problem is Eulerian, i.e, the cost of traveling between any three cities obey triangle inequality.

3.4 Algorithm to Form a Ring Graph Using TSP Formulation

An algorithm is developed using the branch and bound method described in Section 3.2.2. At every step branches extend only to the nodes which lie in the communication range. The steps involved in the algorithm are given in the following.

Given a cost matrix $C = c_{ij} \in \mathbb{R}^+$ and a communication range ξ, a Hamiltonian path is found by the following algorithm

1. The sum of row minima of the cost matrix C is taken as the initial lower bound.
2. Starting from a node with minimum cost, branches are extended to all possible cities which satisfy Equation (3.5)
3. For each branch the costs are calculated by fixing the cost of already traveled path and adding it to the sum of row minimum corresponding to the cities not traveled.
4. The node with minimum cost is selected and branching is continued.
5. Steps 2, 3 and 4 are repeated until the tour has $n-1$ cities.
6. At the $(n-1)^{th}$ city it is verified that the last two paths satisfy Equation (3.5). If it does, then the current tour is taken as the feasible solution and the algorithm stops. If the tour fails the constraint, then the current tour is discarded and branching starts from the node with the minimum cost.

The algorithm is implemented in MATLAB. The working of the algorithm can be seen from the Example 3.2.

Example 3.2 *Let C be the cost matrix of the distances between every vehicle pair i, j $(i, j = 1, 2, 3, ..., n)$ and $\xi = 2\delta$ be the communication range, where $\delta = 5$ is the inter-vehicular spacing and $n = 5$. The evolution of the branches is shown in Figure*

3.6 with the following as the cost matrix.

$$C = c_{ij} \quad where \tag{3.7}$$

$$c_{ij} = \infty, \quad \forall i = j$$

$$c_{ij} = c_{ji} = |(i-j)|\delta \quad \forall i, j = 0, 1, 2, \cdots n$$

As seen from the branches in Figure 3.6, at each node, the branches extend only to the vehicles satisfying the communication range constraint. The branches marked with a '$\times$' are terminated by the algorithm. This is because either the following branch or the terminal branch violates the constraint. The exit condition for the algorithm is the feasible solution. If the execution continues after a feasible solution is found, the algorithm returns the optimum solution which is not necessary since the main objective is to find a Hamiltonian path that satisfies the communication range constraint. Using the algorithm on the eight vehicle platoon, the solution is obtained as $1-2-4-6-8-7-5-3-1$. Coincidentally, the solution obtained for a linear vehicle platoon with $\delta = 2$ is also the optimum solution.

Using the results from both the algorithms, it can be summarized that for a linear vehicle platoon, the minimum cost ring graph can be formed as,

$$1-3-5-\cdots-(n-1)-n-(n-2)-\cdots-2-1, \quad \text{when n is even}$$

and

$$1-3-5-\cdots-(n-2)-n-(n-1)-\cdots-2-1, \quad \text{when n is odd}$$

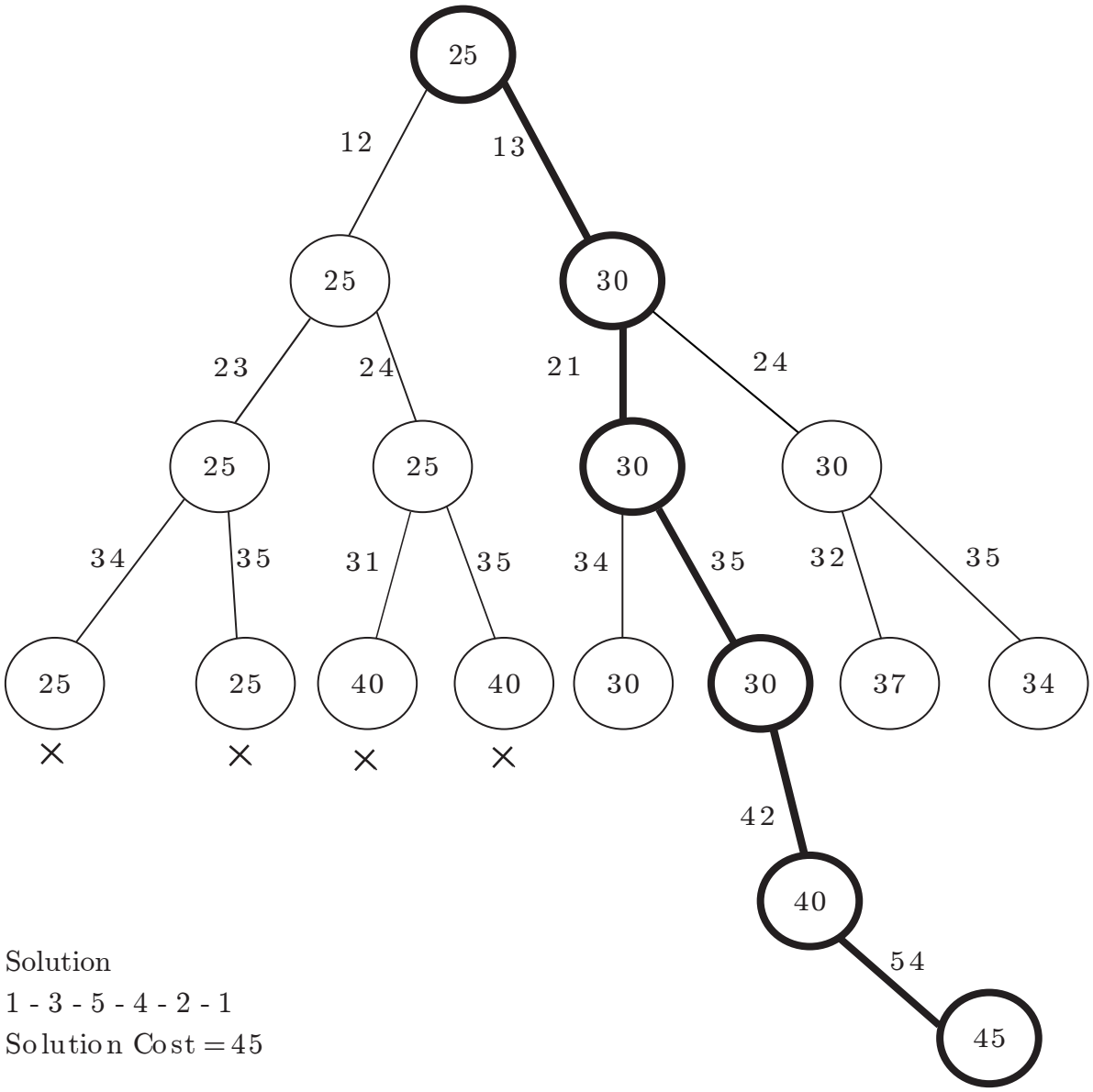


Figure 3.6: Evolution of branches for five vehicles in Example 3.2 using the algorithm for ring graph

CHAPTER 4

VEHICLE FORMATION EXPERIMENTAL SETUP

To verify the simulation results discussed in the previous sections and to verify the results of other coordinated control problems in hardware, a platoon of autonomous mobile robots is planned. A group of five differentially driven robots are built for this purpose. Each robot has two independent driven wheels and an on board autonomous controller. Encoders are used to measure the velocity and displacement of the robot. The development of this experimental setup, the details of the components, their specifications and the controller used are discussed in detail in this chapter. The mathematical model for a two wheeled differentially driven robot is derived in Section 4.1. A two stage controller for control of both velocity and position is explained in Section 4.2. The components of the robots and position sensors are discussed in Section 4.3.

A few basic requirements that the robot platoon has to satisfy in order to run the experiments are conceived. These are as follows:

1. The robots should be small and of compact design so that they can be used indoors.

2. All the robots should be autonomous. They should have their own on board controllers to control position and velocity.

3. All robots should have sensors to measure their own position accurately at any point of time.

4. There should be a central station to monitor the experiment, log data and to

change the trajectory or any other parameters online.

5. All robots should be equipped with wireless communication infrastructure to communicate with each other and with the central computer.

6. Lastly, each robot should be capable of handling any other extra accessories (sensors etc.,.) which may be required to carry out some experiments.

4.1 Robot Model

The differential drive robot model is selected because of its simplicity in deriving the dynamics. This model has been extensively used in the literature [48–53]. There are two independently actuated wheels on a common axis with a caster wheel for maintaining balance. The robot has two motors for actuating the two wheels. For linear motion perpendicular to the axle, both the motor shafts need to rotate in the same direction and with same velocity. To make a turn the wheels turn at different velocities, hence the name differential drive. Figure 4.1 illustrates a differential drive vehicle.

4.1.1 Robot Kinematics

To derive the kinematics of the robot, consider the simple illustration of a vehicle in Figure 4.1. The position and orientation of the robot at any instant of time are described by the vector $q = [x, y, \theta]^T$, where x, y denote the position of point C_R (the center of line joining both wheels) in the global coordinate frame and θ is the orientation of vehicle with respect to the global $x-$axis. The angular velocity of the left and right wheels are denoted by ω_l and ω_r respectively. There are three variables to control and only two control inputs (the wheel velocities) hence a transformation is used to reduce the number of variables to two namely the linear velocity v of the

point C_R and the angular velocity of the robot ω. This transformation is given by

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{4.1}$$

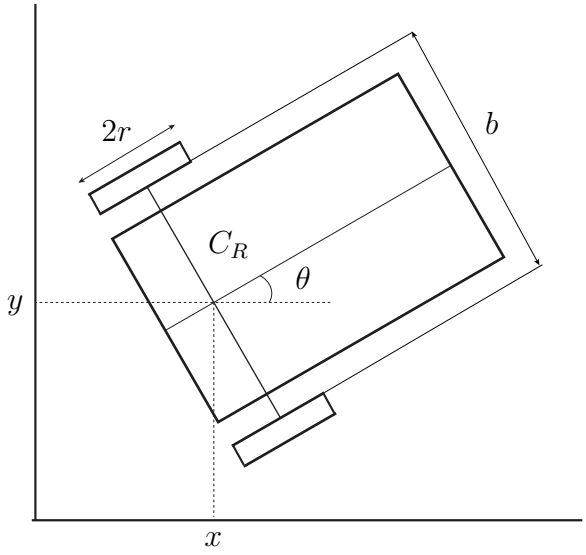


Figure 4.1: Two wheeled differential drive vehicle

The above equation represents the kinematics of the vehicle. It should be noted that if one point on the robot changes then all its points change owing to its rigidity. It is seen that the center of the robot is not necessarily the center of gravity or the center of rotation. The mid-point on the longitudinal axis of the robot, between the wheels is the reference point of rotation. Hence, the velocity at this point C_R is taken as the velocity of the robot. The rotation angle is not restricted to the robot's reference frame, because as a rigid body all points undergo the same change in orientation. When $\omega_r > \omega_l$ then the robot turns to the left of its current position and the orientation angle increases and if $\omega_l > \omega_r$ the robot turns right. The rate of change in orientation which is the angular velocity ω can be written as

$$\omega = \frac{d\theta}{dt} = \frac{r(\omega_r - \omega_l)}{b} \tag{4.2a}$$

$$v = \frac{r(\omega_r + \omega_l)}{2} \tag{4.2b}$$

where r is the radius of the wheel and b is the length of the wheel base, that is, the distance between the wheels measured along the rotation axis of the wheels. The linear velocity of the robot is the average of wheel velocities. Although it is not used in designing a kinematic controller it is worthy to note an important constraint on the robot's motion. This constraint is non-holonomic and is called the no-slip or no-skid constraint. This constraint restricts lateral slip of the wheels thus restricting any lateral motion and is given by the following equation.

$$\dot{x}_i \sin\theta_i = \dot{y}_i \cos\theta_i \tag{4.3}$$

4.1.2 Robot Dynamics

Equation (4.1) represent the kinematics of the vehicle and Equation (4.3) is the no-skid constraint. The dynamics are derived using the Euler-Lagrange equations. The Lagrangian is formed by the total energy of the system which is given by,

$$L = \frac{1}{2}m(\dot{x}^2 + \dot{y}^2) + J\dot{\theta}^2 \tag{4.4}$$

The Euler-Lagrange equation is

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = J^T(q)\lambda + B(q)\tau \tag{4.5}$$

where q is the generalized coordinate vector i.e, $q = [x, y, \theta]^T$, $J(q)$ is the non-holonomic constraint matrix, λ is the Lagrange multipliers of constraint forces, $B(q)$ is the input transformation matrix obtained from the kinematics of the robot and τ is the input torque vector. Substituting total energy (4.4), the transformation of coordinates (4.1) and no-skid constraint (4.3) in Equation (4.5) the dynamics are obtained as

$$\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \ddot{x}_i \\ \ddot{y}_i \\ \ddot{\theta}_i \end{bmatrix} = \frac{1}{r} \begin{bmatrix} \cos\theta_i & \cos\theta_i \\ \sin\theta_i & \sin\theta_i \\ b & -b \end{bmatrix} \begin{bmatrix} \tau_l \\ \tau_r \end{bmatrix} + \begin{bmatrix} \sin\theta_i \\ -\cos\theta_i \\ 0 \end{bmatrix} \lambda \tag{4.6}$$

where m is the mass and I the inertia of the robot about its rotational axis. There are several methods to eliminate the Lagrangian multiplier λ. One can use the vehicle kinematics to get the accelerations and substitute the accelerations in the above equation. Another method is to use Boltzmann - Hamel equations [48]. Here the kinematics based method is used since the equations are relatively simple.

Since there is no lateral torque on the vehicle, the total applied torque is the sum of linear (τ_L) and angular (τ_A) torques. The linear and angular torques are

$$\begin{aligned} \tau_L &= \frac{1}{r}(\tau_l + \tau_r) \\ \tau_A &= \frac{b}{r}(\tau_l - \tau_r) \end{aligned} \tag{4.7}$$

where τ_l and τ_r are the torques of left and right motor respectively. Using the linear and angular torques, Equation (4.6) can be written as

$$\begin{aligned} \ddot{x}_i &= \frac{1}{m}(\tau_{L_i}\cos\theta_i + \lambda\sin\theta_i) \\ \ddot{y}_i &= \frac{1}{m}(\tau_{L_i}\sin\theta_i - \lambda\cos\theta_i) \\ \ddot{\theta}_i &= \tau_{A_i}/I \end{aligned} \tag{4.8}$$

Differentiating the vehicle kinematics from Equation (4.1), substituting the vector $[\ddot{x}_i, \ddot{y}_i, \ddot{\theta}_i]^T$ in Equation (4.8) and solving for $[\dot{v}_i, \dot{\omega}_i]^T$ we get

$$\begin{bmatrix} \dot{v}_i \\ \dot{\omega}_i \end{bmatrix} = \begin{bmatrix} \tau_{L_i}/m \\ \tau_{A_i}/I \end{bmatrix} \tag{4.9}$$

Equation (4.9) is the dynamic equation of the individual vehicle. These dynamics are used in designing a torque based motor velocity controller.

4.2 Trajectory and Velocity Control

The trajectory controller used by every individual vehicle is derived in this section. There are different types of controllers available in literature for controlling the position and velocity of an autonomous robot. These controllers are based on the type of

application in which the robot is used; avoidance control, point to point control, etc. The closed loop controllers are designed based on the type of problem that needs to be solved [54].

1. Point-to-point control: The robot should reach the final point from a given starting point irrespective of the path it takes.

2. Path following control: The robot must reach a predefined path and follow the path starting on or off the path.

3. Trajectory tracking control: The robot must reach the trajectory (i.e, a path with associated timing law) and follow the trajectory starting on or off the trajectory.

The current work focuses on tracking a reference trajectory, hence, a trajectory tracking controller is used.

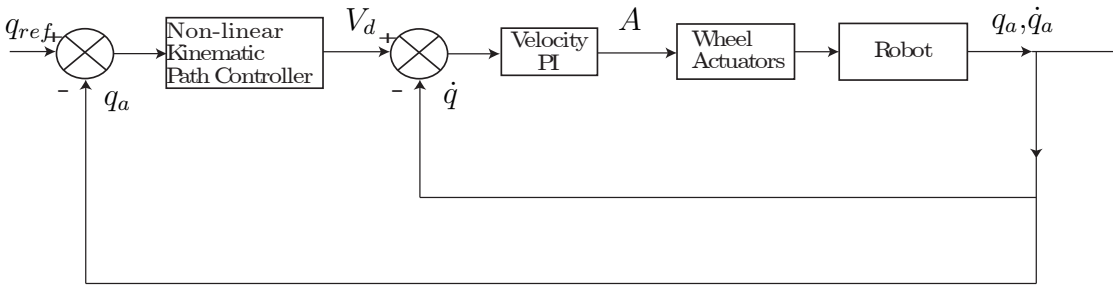


Figure 4.2: Two loop trajectory tracking kinematic controller

The strategy involves two control loops. The outer control loop is a path controller which controls the desired velocity. A non-linear kinematic path controller first developed by Kanayama et al. in [55], is used to calculate the desired velocities to track the reference trajectory. Different variations of this controller were used in the literature [48, 51]. The controller takes the position of the vehicle and the reference position as the inputs and, outputs the desired linear and angular velocities which are required to make the position error zero. The desired velocities are used as

inputs to the inner loop which controls the voltage to the individual motors. Since the dynamics of the robot are ignored, we assume that the motors are capable of generating the required velocity and accelerations. The working of the controller is better understood from the controller block diagram in Figure 4.2, where $V_d = [v_d, \omega_d]^T$ is the vector of desired velocities, $A = [V_l, V_r]^T$ is the vector of actuation voltages and q_a, $\dot{q}_a$ are the actual position and velocity.

4.2.1 Outer Path Control loop

In order to use the path controller, the position error should be calculated. Let q_r be the reference position of the robot and q_a be the actual position, that is calculated from the measured wheel displacement. Referring to Figure 4.3, the tracking error is written as

$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} \cos\theta_r & \sin\theta_r & 0 \\ -\sin\theta_r & \cos\theta_r & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - x_r \\ y - y_r \\ \theta - \theta_r \end{bmatrix} \tag{4.10}$$

$$\begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\theta \end{bmatrix} = \begin{bmatrix} \omega_d e_y - v_d + v_r \cos e_\theta \\ -\omega_d e_x + v_r \sin e_\theta \\ \omega_r - \omega_d \end{bmatrix} \tag{4.11}$$

where subscript d refers to the desired value of the corresponding variable. Equation (4.11) is the dynamics of the path error. The control input for the error dynamics $[v_d, w_d]^T$ are given by the following equations:

$$v_d = v_r \cos e_\theta + k_x e_x \tag{4.12a}$$

$$w_d = w_r + v_f(k_y e_y + k_\theta \sin e_\theta) \tag{4.12b}$$

where k_x, k_y and k_θ are positive gains. These desired velocities act as the reference to the inner loop controller which controls the motor velocities.

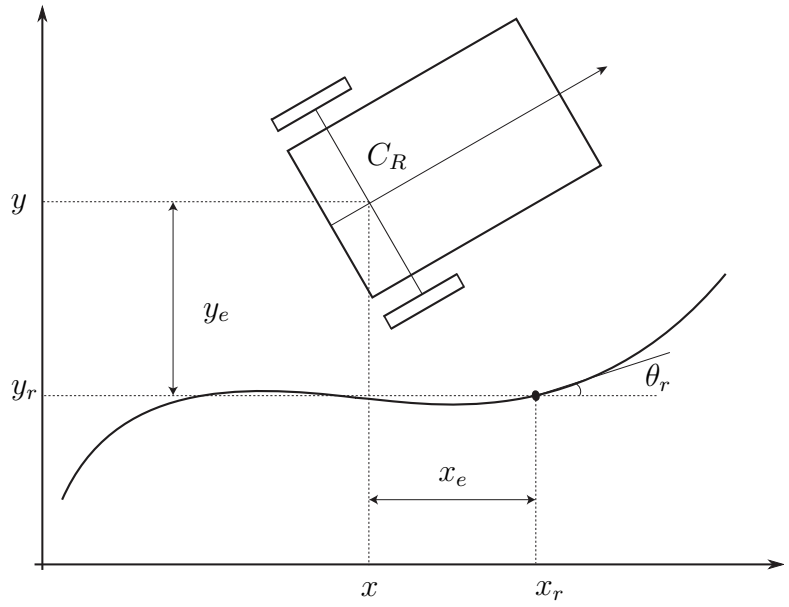


Figure 4.3: Trajectory tracking vehicle model

4.2.2 Inner Velocity Control loop

The inner loop uses the desired linear and angular velocities from (4.12) to generate the voltage inputs for the individual motors. The velocities of the individual motors are calculated using the Equation (4.2). These calculated velocities are used as the feedback for the inner velocity loop. A simple proportional integral (PI) controller is used for velocity control. It is given by

$$V = K_p W_e + K_i \int_0^t W_e(\tau) d\tau \tag{4.13}$$

where $V = [V_l, V_r]^T$ is the voltage input for each motor, $W_e = [\omega_{e_l}, \omega_{e_r}]^T$ is the error vector of the left and right motor velocities, K_p, K_i are the proportional and integral gains, t is the time.

4.3 Components of Robot

The robot consists of three main components. The robot base and actuators, controller and wireless modules, and position sensors. Each component is explained in detail in this section.

4.3.1 Robot Base and Actuators

A two wheeled differential drive model has been used to make the robot. For simplicity of dynamics and to avoid wheel lift off on uneven surfaces the two driven wheels are placed at the back of the base plate. There is a caster wheel in front to maintain balance. In order to keep the center of gravity of the entire robot close to the center of wheel axis the battery is placed between the wheels on the centerline of the base.

The base of the robot is machined from a sheet of aluminum of length 9 in, width 7 in, and thickness 0.1 in. The initial configuration of the base was made such that the wheels are mounted at the center of plate. There were two caster wheels, one in the front and the other in the rear for balance. The model was machined, assembled and tested. But, this model suffered from the wheel lift problem on uneven surfaces. The two caster wheels lifted the whole vehicle up when the surface got uneven, which resulted in wheel slippage. So, the wheel location was shifted to the rear of the base. A computer model of the base plate is create in Solidworks with the exact dimensions of the plate, the hole and cutout locations. The drawing of the base plate is shown in Figure 4.4.

The two motors to power the wheels are mounted under the base plate using aluminum brackets. The wheels are directly mounted on the motor shafts. A steel ball caster is used to maintain the balance and level of the vehicle. This is mounted on the front end of the base, close to the edge. A plastic box that houses the controller is mounted on the top of the base such that the plate and box centers are aligned. The battery is mounted under the plate between the motors along the longitudinal axis of the plate. The battery is held in place with a flange. This flange is also made of aluminum. This makes it easy to charge the battery without dissembling it from the vehicle. Finally, a switch is also assembled on the top of the plate for convenience.

Two DC motors each with a no-load speed of 350 rpm and a stall torque of 8 kg-cm are used to power the wheels of the robot. Each motor has an encoder attached to

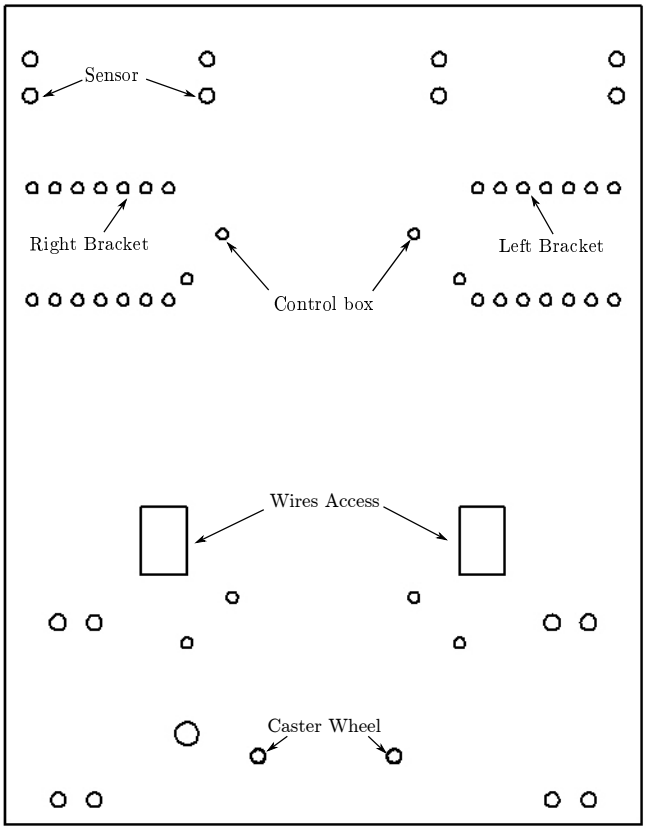


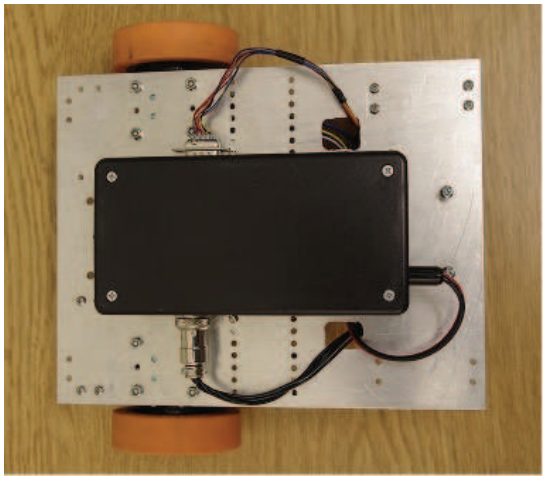
Figure 4.4: Computer generated drawing of the robot base

its back shaft which generate 1856 pulses for every revolution of the shaft. The axis of motor shafts are aligned and intersect the longitudinal axis of the robot. Other specifications of the robot are given in Appendix B. The entire robot weighs about 5 lbs and the base plate is 2.48 in high from the ground. Figure 4.5 shows the actual robot.

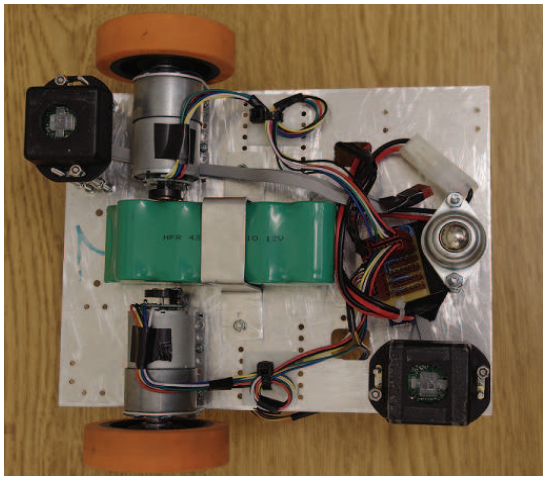
4.3.2 Controller and Wireless Module

The hardware used to implement the controller is saught based on the following conditions:

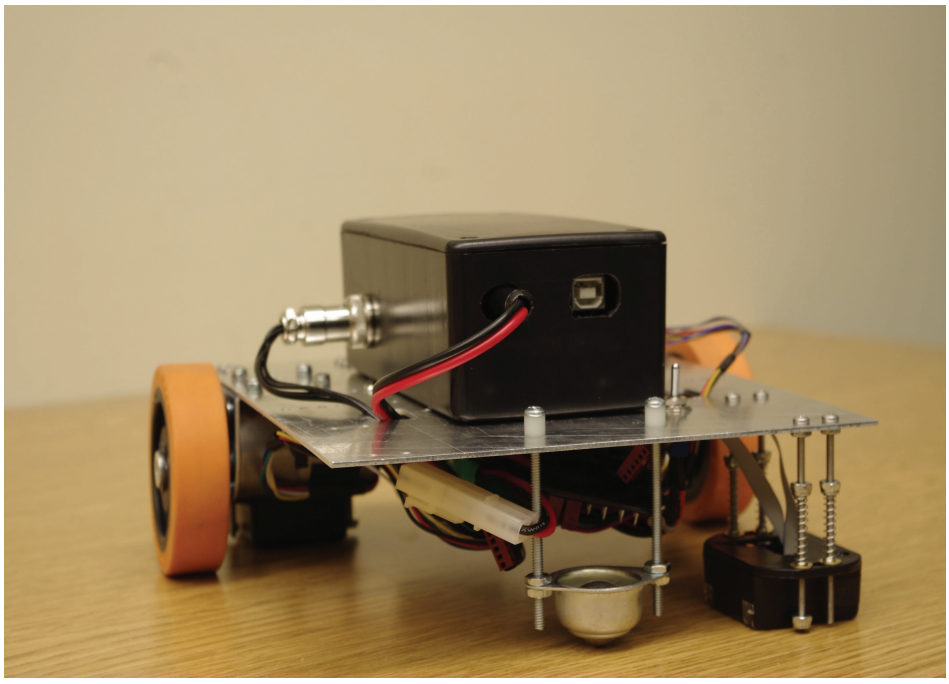
- The controller should be fast and should have sufficient memory to execute the control program in a given sampling period.



(a)



(b)



(c)

Figure 4.5: Robot view (a) Top, (b) Bottom, (c) Robot with sensors mounted

- It must support various communication interfaces including I^2C, it should be capable of actuating the motors and read the data from encoders.
- It should be programmable and must have necessary interfaces to connect with a computer.
- It should support all the extra hardware needed to run the motors, communicate via wireless, provide power to encoders.
- It must be cheap, reliable and should have necessary support.

The controller is implemented on a Arduino Mega 2560 development board. The board is based on the ATmega2560 micro controller. Some important features of the ATmega2560 include 256 KB of flash memory, 8 KB of SRAM and 4 KB of EEPROM, clock speed of 16MHz, 14 pins to provide 8-bit PWM output, 6 external interrupt pins and I^2C communication interfaces [56] . The board has an interface to connect directly to a PC via USB, internal voltage regulators of 5 V and 3.3 V. The board can operate from an external supply between 6-20 V and has a USB over-current protection. It can be programmed using Arduino Software Development Kit [57]. Figure 4.6(a) shows an Arduino Mega 2560 board.

The direction and velocity PWM signals are sent to the motor controller board which takes in the PWM and generates the voltages required to run the motors. An Arduino Motor shield shown in Figure 4.6(b) is used. It is based on the L298 dual full-bridge driver. The board can run two 12 V motors with a maximum of 2 A current. It also has an on board current sensor for each motor channel which is useful when a torque based motor velocity controller is used. A 2.4 GHz Xbee module from Digi is used for wireless communication. This module takes the 802.15.4 stack and wraps it into a simple to use serial command set [58, 59]. The Xbee module operates at 3.3 V @ 50 mA. It transmits data at a maximum rate of 250 kbps. It has a range

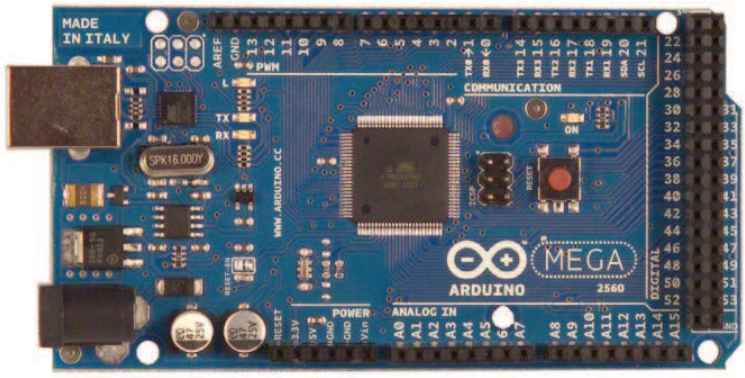
of 300 ft and has a 128-bit encryption. Figure 4.6(c) shows an XBee wireless module with an on-chip antenna.

The motor controller and the XBee wireless module are stacked on the top of the Arduino board as in Figure 4.6(d). The whole setup along with an input voltage regulator is enclosed in a box which is mounted on top of the base plate. This eliminates any interference to the wireless communication.

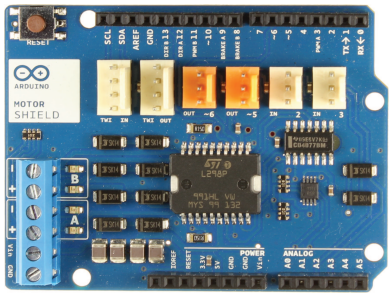
The stacked controller is capable handling all the required functionalities. It can execute the control algorithm, control the motors, communicate via wireless and obtain data from mouse sensors using I^2C bus. It can also handle the encoder pulses using four external interrupts.

4.4 Position Sensor

The whole experiment is based on accurately sensing the position of the robot by itself. Hence, a position sensor is of utmost importance. The encoders on the motor shafts act as the primary position sensors. As mentioned earlier they have a resolution of 1856 counts per revolution of output shaft. Each encoder is powered by a 5 V supply and has two data channels A and B. The voltage in the channels is either 5 V or 0 V and it pulsates as the motor shaft rotates. This is illustrated in Figure 4.7. Channels A and B of the left motor encoder are connected to pins 20, 21 and that of the right motor encoder are connected to pins 18 and 19. Each of these pins can function as external interrupt pins. The interrupt pins change their state when the voltage in the channel changes. An interrupt sub-routine is executed every time the pins change their states. This interrupt sub-routine counts the number of pulses in both the channels and the direction of the rotation, comparing the sequence in which the pins change their respective states. The binary coding for clock-wise and counter clock-wise rotation of shaft is shown in Tables 4.1 and 4.2. The bit 1 represents the high state or 5 V and 0 represents low state or 0 V.

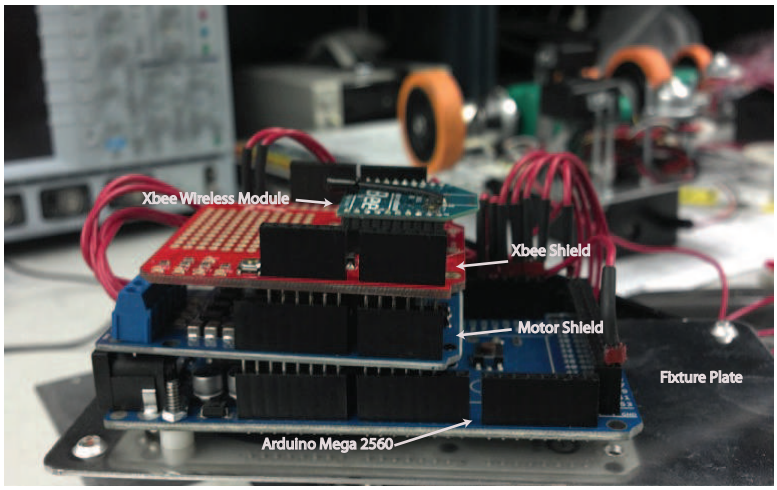


(a)



(b)

(c)



(d)

Figure 4.6: (a)Arduino Mega 2560 R3 Development board, (b) Arduino Motor Shield R3, (c)Xbee wireless communication module, (d) Arduino Motor Shield and Xbee Shield with Xbee module stacked on top of Arduino Mega 2560

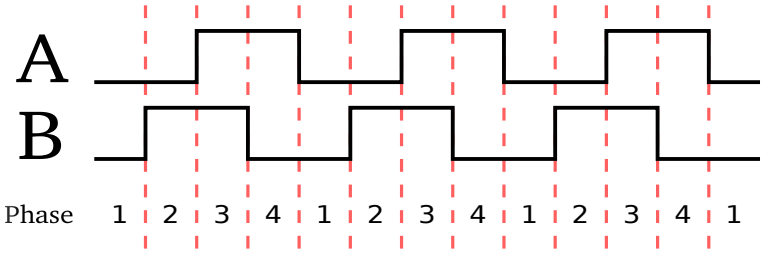


Figure 4.7: Pulses in channels A and B of quadrature encoder

Phase	A	B
1	0	0
2	0	1
3	1	1
4	1	0

Table 4.1: Clock-wise rotation

Phase	A	B
1	1	0
2	1	1
3	0	1
4	0	0

Table 4.2: Counter clock-wise rotation

The encoder output is measured for every time interval δt. This measurement gives the counts of each individual wheel which when divided by the time interval and the resolution give the velocity of each wheel.

$$W = \frac{EncoderCounts \times 2 \times \pi}{\delta t \times 1856} rad/s \tag{4.14}$$

The displacement of the robot is calculated from the measured counts during the time interval δt and is added to the position at the time sample $(t-1)$ gives the position at time t.

$$X = X_0 + \frac{b(v_r + v_l)}{2(v_r - v_l)} \begin{bmatrix} \sin(\theta) - \sin(\theta_0) \\ -(\cos(\theta) - \cos(\theta_0)) \end{bmatrix} \tag{4.15}$$

where $X = [x, y]^T$ and X_0 are the current and previous position of the robot. Although encoders are cheap, they are prone to errors which creep in due to the wheel slip. This error can quickly grow as the time increases and become a considerable part of measurement, if the trajectory involves rapid accelerations and decelerations.

CHAPTER 5

EXPERIMENTAL RESULTS

The experimental platform discussed in Chapter 4 was used to carry out a series of experiments using the path and ring graphs. Three robots are used in this series of experiments. Section 5.1 discusses the tests on individual robots. The tracking controller discussed in Section 4.2 is implemented on the robots and the gains k_x, k_y and k_θ of the outer path control loop and the PI gains K_p and K_i of the inner velocity control loop are tuned. Experiments with the path controller where robot 1 is completely independent and the remaining robots follow 1 are conducted. The results of these experiments are presented in Section 5.2. The experiments conducted with ring graph and the corresponding results are presented in Section 5.3.

5.1 Trajectory Tracking

The tracking controller derived in Section 4.2 is used on the robots 1 and 2 and the gains are tuned for each robot to follow a trajectory. Since, all the coordination experiments with path and ring graph involve tracking a trajectory, this set of tests are essential. The controller gains are tuned in two steps. First the inner motor control PI is tuned by setting the reference linear velocity of the robot to a constant value of 24 inches/sec and the angular velocity to 0 rad/s. The proportional and integral gains are tuned with the aforementioned values and the gains thus obtained are tested at different velocities. Once the inner loop is tuned, a straight line trajectory is used to tune the outer loop gains. The robot starts at zero velocity and reaches a steady velocity 24 inches/sec smoothly. The points on the trajectory are obtained from the

equations

$$v_r = \begin{cases} 24 \sin\left(\frac{\pi t}{t}\right) & 0 \leq t \leq 5 \\ 24 & t > 5 \end{cases}$$

$$x_r(t) = x_r(t-1) + v_r \delta t \cos(\theta_r)$$
$$y_r(t) = y_r(t-1) + v_r \delta t \sin(\theta_r) \tag{5.1}$$

where v_r is the reference linear velocity and t is the time interval. The tuned controller is used at different velocities and different trajectories. The gains thus obtained are summarized in Table 5.1.

K_p	0.5
K_i	0.01
k_x	3
k_y	0.001
k_θ	0.07

Table 5.1: Gains of trajectory controller

Linear Velocity v_r	18 in/s
Angular Velocity ω_r	0 rad/s
Controller Interval	150 ms
Velocity PI Interval	5 ms

Table 5.2: Experimental parameters for testing the tuned trajectory controller

The reference position and actual position of robot 1 and 2 are calculated using the encoders and are shown in Figure 5.2. Table 5.2 shows the parameters used for this test. Both the robots show good tracking capabilities. The error in $x-$ and $y-$ directions is of the order 0.1 in.

5.2 Path Graph

The path graph is defined in Definition 2.3 in Chapter 2. For a vehicle platoon, a path graph is obtained by simple connecting each vehicle to its following vehicle or by breaking the last link of a ring graph. Figure 5.1 illustrates an example of

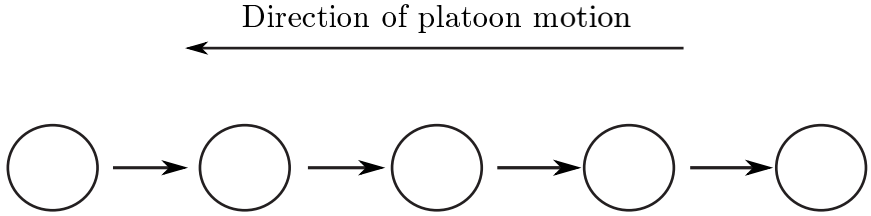


Figure 5.1: Vehicle platoon with a path graph

the path graph for a five vehicle linear platoon. The path graph is one of the most basic forms of a directed graph, hence, experiments using path graph will test the parameters required for coordination controllers. Limitations of the hardware like the wireless range, data transfer rates practically feasible, time interval between control loop execution, etc., can be better understood by testing the platoon with a path graph. The properties tested are essential for evaluation of a coordination controller. The conditions for the test are listed in Table 5.3. For tests involving the path graph and the ring graph in the later section, a start-up delay of five seconds was included before the coordination algorithm is turned on. During this delay the robots are controlled independently without coordinating with each other, and they are also accelerated to a starting velocity. This delay helps in reducing the accelerations and also sets non-zero initial conditions.

Experiments were conducted by considering two cases. In the first case, the robot motion is mimicked by simply running the wheels without touching the floor; this case is referred to as 'off the floor' setup. In this case the measurements are devoid of slip effects. The second is an 'on floor' test where the robots are placed on ground and are free to move on the floor. The first robot in the platoon, the leader uses

Property	**Value**
Linear Vvlocity v_r	12 in/s
Angular velocity ω_r	0 rad/s
Spacing δ_r	24 in
Control loop interval	150 ms
Velocity PI interval	5 ms
Communication interval	75 ms
Baud-rate	9600
Start-up delay	5 sec

Table 5.3: Test conditions for path graph on a platoon of three robots

an independent trajectory input similar to that discussed in Section 5.1 with an addition of a communication subroutine which is used to transmit its position and velocity data and to parse the received data. The communication follows the ring structure in the sense that the second robot transmits only when the first robot has finished transmitting and the third robot does not transmit until it receives from the second robot.

The results for a three robot platoon with a path graph in off the floor and on floor test settings are shown in Figure 5.3. The following are some observations from these experiments.

- The first and most important problem which surfaced during this experiment is related to the wireless communication. It was observed that the communication stops abruptly after a couple of seconds into the experiments. On further diagnosis, it was found that this is caused by the communication chain. Since each robot waits to receive information from the preceding robot to transmit its own information, any loss of communication will immediately break the chain and the communication loop comes to a halt. An additional variable in the form

of a tracker is added to the communication routine of the third robot. This tracker keeps track of the time elapsed since the last out-going transmission. If the value of the tracker goes beyond a set point, the program is forced to send out a transmission, thereby, restarting the communication loop. Figure 5.3 shows the results after implementing this strategy.

- From the plots it can be observed that in both on and off the floor settings, the robots 2 and 3 follow the first robot along the trajectory. The average inter-vehicular spacing between robots 2 and 3 is maintained close to the desired value of 24 inches with a maximum spacing error of 0.5 inches in off the floor setup. But, the spacing between robots 1 and 2 is 26 ± 2 inches, which is higher than the desired inter-vehicular spacing. This might be attributed to the path graph and the independent lead vehicle.

- Average velocity is maintained approximately at 12 ± 2 inches/sec. It was observed that the velocity of the follower robots is oscillatory due to the large delay of 150 ms between the successive control loops.

- It was observed at the end of the test that the actual position of robots 2 and 3 is different from the recorded position. This is due to wheel slip caused by rapid accelerations and decelerations along the path. The controller gain k_x is lowered to 2 which reduced the oscillations thereby reducing wheel slip. But the slip is not totally eliminated. Any further reduction in gains resulted in poor coordination and large spacing errors.

The following conclusions can be made from the results of the path graph.

- There are losses in communication which should be eliminated to have better coordination. A strategy needs to be implemented that is more effective than using the communication tracker.

- The controller gains which worked in the single robot case may not yield satisfactory results with coordination control.

- The path graph has the inherent problem of having one completely independent vehicle which results in higher inter-vehicular spacings and non-zero velocity errors.

5.3 Ring Graph

A series of experiments were conducted using the ring graph with different initial conditions. The experiments and their results are discussed in this section. Some parameters are fixed for these experiments. The reference velocity is chosen to be 12 in/s. The velocity value is set only in robot 1 which communicates it to the robot 3 which in turn communicates it to robot 2. The communication flows from robot to robot in the manner discussed in the previous section. The communication tracker discussed in the previous section is added to the communication routine of robot 3. The interval between successive control loop executions is set to 150 ms and communication loop is executed every 75 ms. Other test conditions are summarized in Table 5.4.

Three sets of experiments are conducted with three different initial conditions. In all the three cases, the vehicles start at zero velocity and accelerate for 5 seconds after which the coordination algorithm is turned on. The three sets of experiments are run at three different initial inter-vehicular spacing.

5.3.1 Initial spacing equal to the desired spacing

In the first set of experiments, the initial inter-vehicular spacing is made equal to the desired inter-vehicular spacing. The robots 1, 2 and 3 are initially positioned at 48, 24 and 0 inches respectively, from the base line. Several runs of the test are carried

Property	**Value**
Linear velocity v_r	12 in/s
Angular velocity ω_r	0 rad/s
Inter-vehicular spacing δ_r	24 in
Control loop interval	150 ms
Velocity PI interval	5 ms
Communication interval	75 ms
Baud-rate	9600
Start-up delay	5 sec

Table 5.4: Test conditions for ring graph on a platoon of three robots

out. The results of the tests are shown in Figure 5.4. The plot shows the evolution of positions of robots 1, 2 and 3 with respect to time.

It can be observed from the plot that all three vehicles maintain their respective positions in the workspace. After the start-up delay of 5 seconds there is a small adjustment where the robots 1 and 2 move back to drive the spacing error to zero. After this adjustment, the platoon moves smoothly along the trajectory. The average inter-vehicular spacing between robots 1 and 2 and robots 2 and 3 is maintained within ± 0.5 inches of the desired value. The velocity of the platoon is also maintained close to the desired value, 12 ± 0.2 in/s.

5.3.2 Initial spacing less than the desired spacing

The second set of experiments start with a smaller inter-vehicular spacing. The initial spacing is set to 12 in and the robots 1, 2 and 3 are positioned at 36, 24 and 12 inches from the base line. The results after several runs are represented in the robot position plots in Figure 5.5.

The plots shows a bigger bump right after the coordination algorithm starts work-

ing. This is due to the large magnitude of spacing error caused by the initial robot positions. Once the positions are adjusted, the rest of the plot is smooth with inter-vehicular spacing maintained within ± 0.2 inches as in the previous case. The platoon velocity is also maintained close to the desired value at 12 ± 0.1 in/s.

5.3.3 Initial spacing more than the desired spacing

In the last set of experiments, the initial spacing is set to 12 inches more than the desired spacing. The robots are positioned at 60, 24 and -12 inches from the base line. The results are plotted in Figure 5.6.

Although the plots shows a bigger bump similar to the previous case, the platoon does not settle. It can be observed from the plot that is the platoon is still trying to adjust the positions between 16 and 18 seconds into the experiment. The spacing error between robots 1 and 2 is found to be of the order 0.5 inches, while the error between robots 2 and 3 is maintained around 0.2 inches. The reason for this might be the time delay between successive control loop executions.

The following conclusions are made based on the results

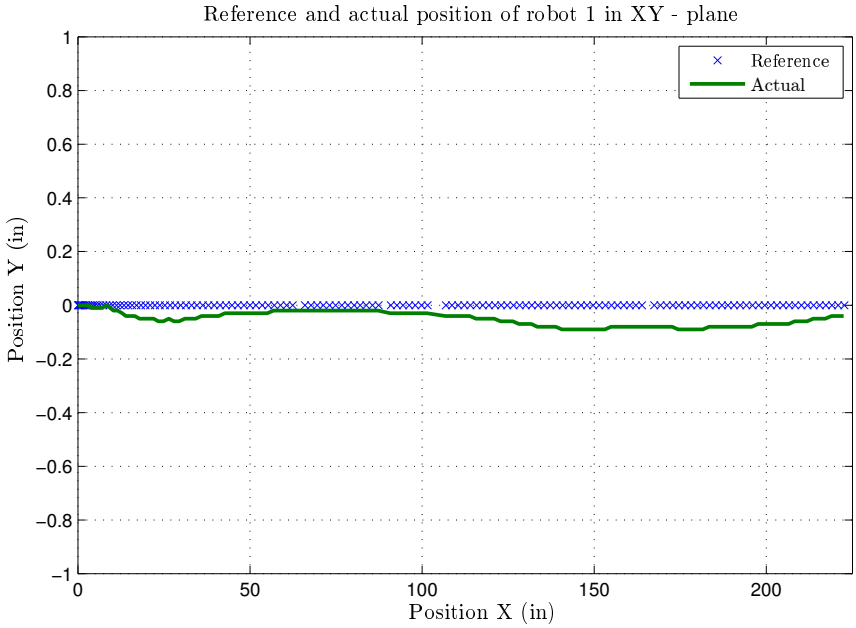
- The inter-vehicular spacing is maintained for all the three tests compared to that of the path graph. The inter-vehicular spacing is an average of 24 ± 0.5 in/s all through the test even with non-zero initial conditions.

- The platoon tracked the desired trajectories. The separation between robots is caused when the coordination algorithm tries to adjust the positions of the robots.

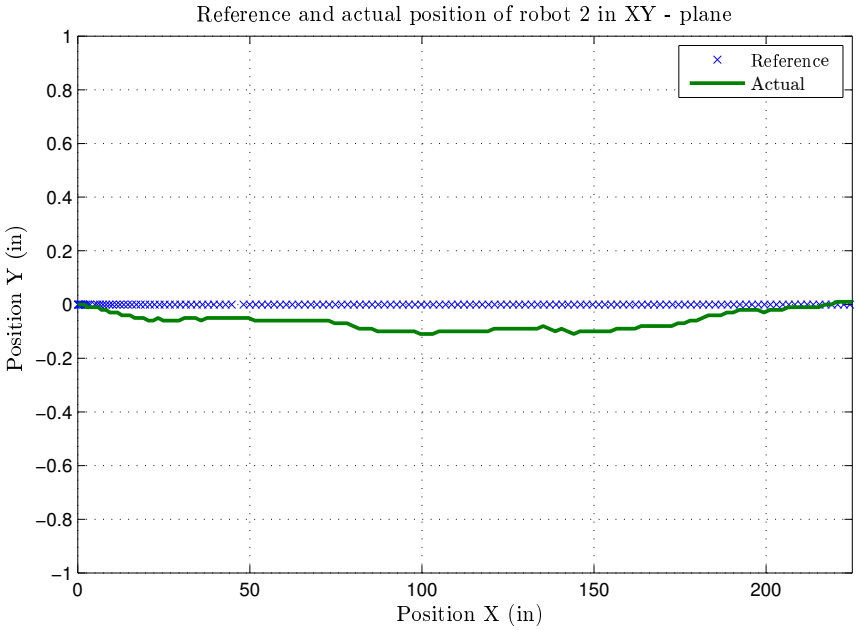
- The platoon velocity is maintained at approximately 12 ± 0.2 in/s.

- The effect of wheel slip is also observed with the ring graph. The actual displacement of the robots in the lateral direction is not captured in the measurements

from the encoders. In case of the ring graph, the first robot also slips since it is also a part of the coordination.

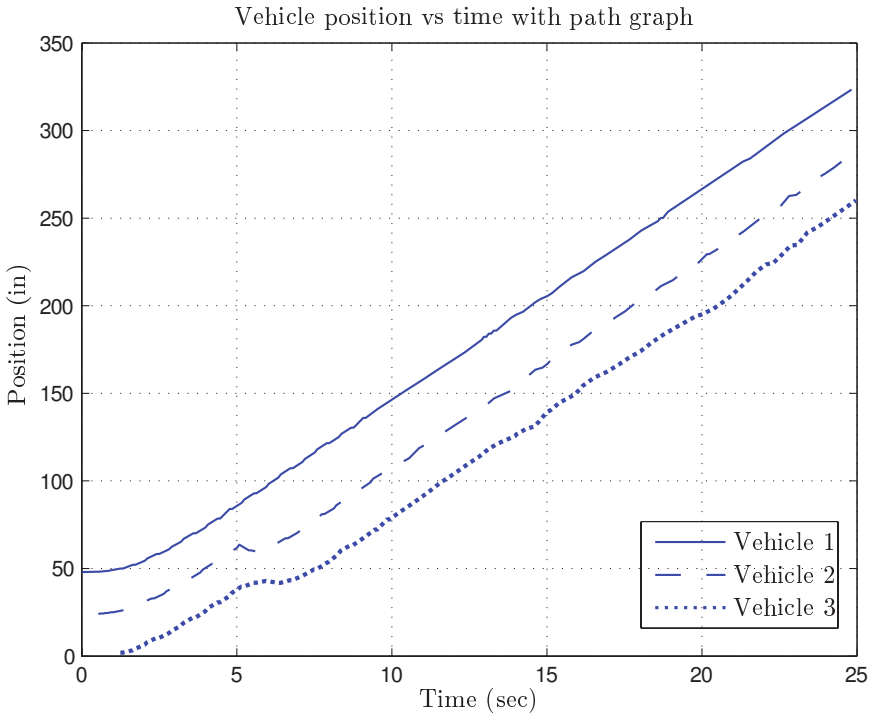


(a)

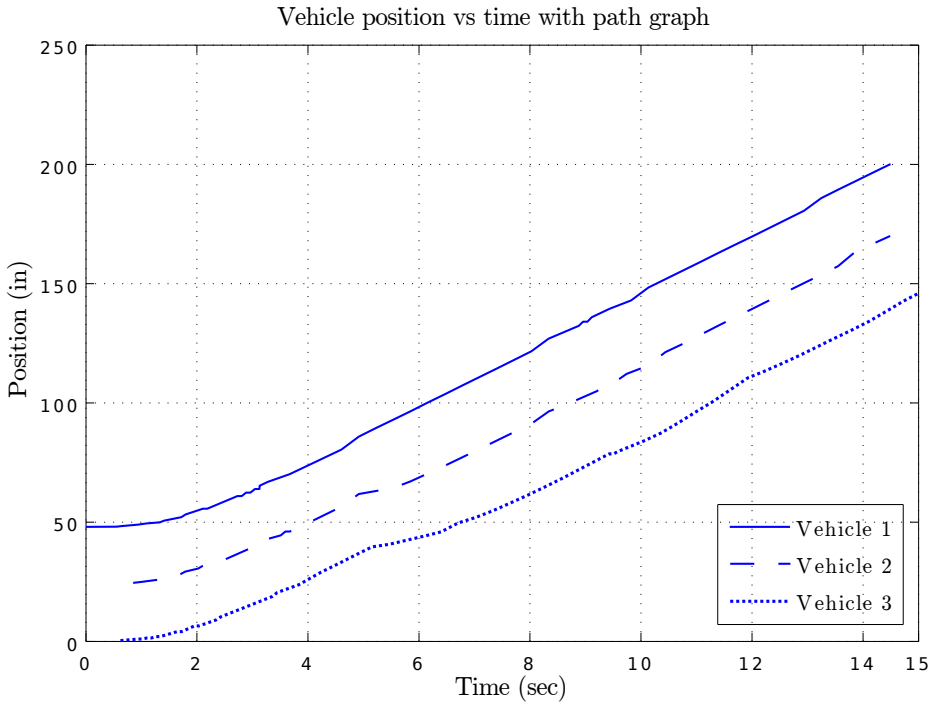


(b)

Figure 5.2: Reference and actual position of robots 1 and 2



(a)



(b)

Figure 5.3: Position plots for robots 1, 2 and 3 with path graph, (a) Off the floor test, (b) On floor test

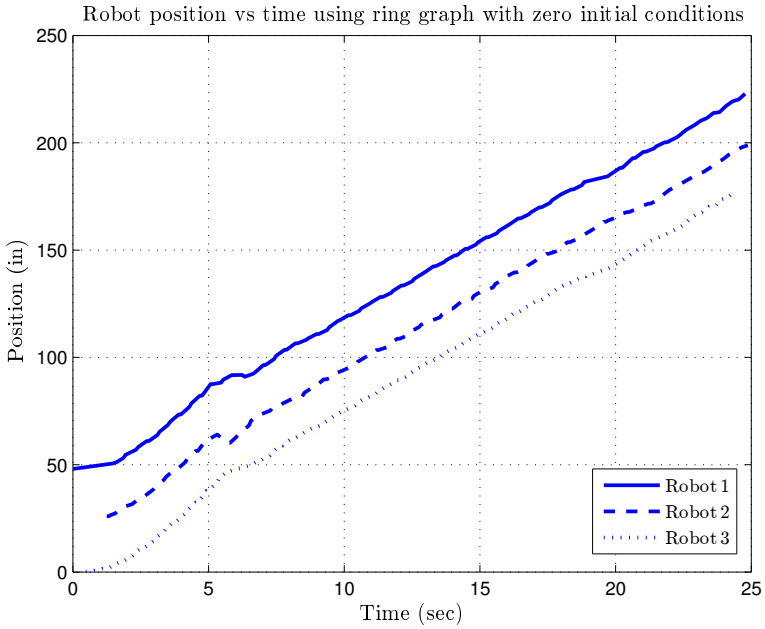


Figure 5.4: Position of robots vs time with initial spacing equal to desired spacing

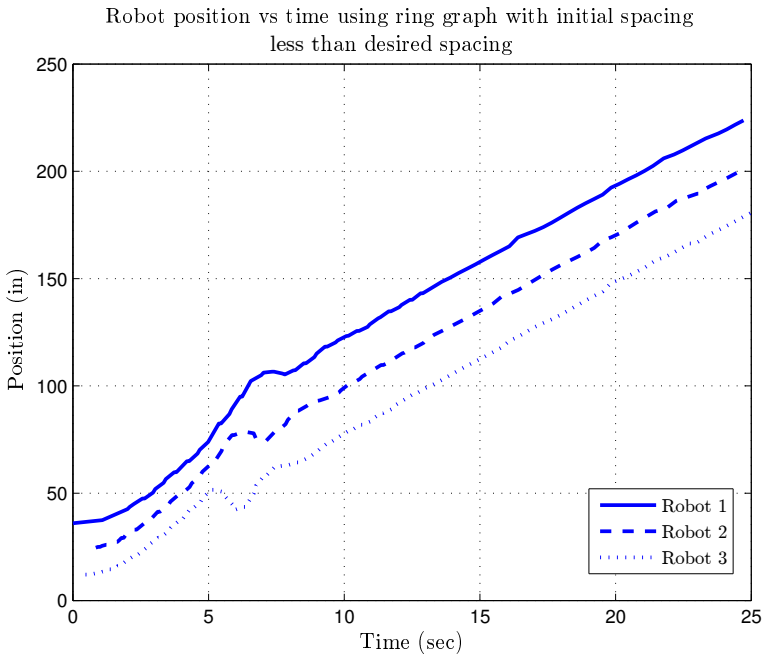


Figure 5.5: Position of robots vs time with initial spacing less than desired spacing

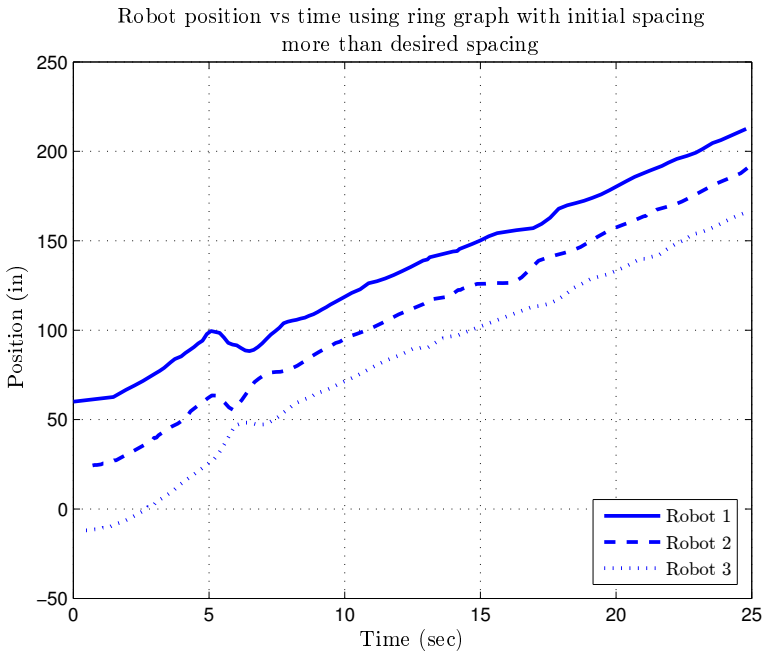


Figure 5.6: Position of robots vs time with initial spacing greater than desired spacing

CHAPTER 6

SUMMARY AND FUTURE WORK

6.1 Summary

The focus of this thesis was on coordination of multiple autonomous vehicles with the information flow between vehicles given by a special kind of directed graph called the ring graph. The effect of the ring graph based information flow structure was investigated in detail for coordination of multiple autonomous vehicles.

In Chapter 2 the literature on ring graphs is reviewed in detail. It was shown that the ring graph may not be scalable due to the presence of multiple conjugate pairs of eigenvalues close to zero as the formation size increases. Limitations due to the basic ring structure are examined and possible alternative solutions are explored. It was shown that having multiple vehicle groups with inter-group communication will result in losing the circulant nature of the closed loop system matrix. We considered the double integrator dynamics of the vehicle and analyzed the eigenvalues of the platoon with a ring graph, to show that the platoon is stable as long as a closed ring is formed irrespective of actual vehicle positions. The change in the location of the eigenvalues due to change in the position of vehicles in ring is also studied. We also studied the effect of multiple rings graphs over a platoon. A controller similar to the controller based on single ring graph was used to show that the convergence rate is faster with multiple ring graphs than that of single ring graph. The condition for stability of closed loop system is derived using eigenvalue analysis.

In Chapter 3 we address the question of forming a ring graph for a given formation. Two algorithms were presented for this purpose. The first algorithm is a simple

assignment which is demonstrated for a linear vehicle platoon. An investigative study into the Traveling Salesman Problem and some algorithms to solve the TSP was undertaken. The problem of finding the ring graph is formulated as the problem of finding the Hamiltonian path of a given set of nodes. An additional constraint on the path length of TSP formulation was introduced which is due to the communication range of the vehicle transmitters. A solution algorithm based on the branch and bound methods for TSP is presented. The working of this algorithm for a linear vehicle platoon is demonstrated.

In Chapter 4, details of an experimental setup to test the coordination control algorithms are presented. The differential drive robot model is discussed. The kinematic model of the differential drive robot is derived and the dynamics are derived using Euler-Lagrange equation. The two-loop controller strategy that controls the position and velocity of each individual robot is presented. The non-linear kinematic path controller which forms the outer loop controls the robots linear and angular velocity and the Proportional Integrator (PI) controller of the inner loop which controls the individual motor velocities are presented. The components of each robot, that is, the mechanical base, wheel actuators, controller and other electronics, and motor shields are discussed in detail. The process of estimating the position from the wheel encoders and the associated limitations are discussed.

Chapter 5 discusses the experiments carried out using the robotic vehicles discussed in Chapter 4. Experiments are conducted to test the performance of a trajectory controller and wireless communication devices, effect of wheel slip, and sensor measurements. The multiple robot setup was also used to run experiments with a ring graph using different intial spacing conditions. The results of these experiments are presented and discussed. Some conclusions about the merits of the ring graph were drawn based on these results.

6.2 Future Work

There are some open problems resulting from this work which can be studied in the future.

The issue with the scalability of the controller using ring graph should be addressed. The string stability of the platoon with multiple communication rings needs to be investigated. The presence of some symmetric edges in the graph makes this problem more interesting. In Chapter 2 a ring graph is applied to 2D formations where the dynamics can be decoupled; however, this may not always be feasible. The effect of ring graph on stability of 2D formations need to be investigated when the dynamics cannot be decoupled.

In Chapter 4, the inner loop is a pure velocity controller, and it does not account for the robot and motor dynamics. This controller should be replaced with a dynamic torque based controller that accounts for the robot's mass and inertia.

More experiments should be conducted to quantitatively and qualitatively show the robustness properties of the ring graph. Experiments should be conducted with all the five robots that were built. So far the experiments are only concentrated on a single direction of transport with a linear vehicle platoon. Future experiments should focus on implementing ring graph on a two dimensional formation such as a delta formation or a square formation.

All the robots that are used in the current work are identical with same dynamics. Coordination control experiments should be carried out with vehicles that are not identical and have different dynamics.

BIBLIOGRAPHY

[1] A. Dávila and M. Nombela, "SARTRE: SAfe Road TRains for the Environment," September 2010.

[2] S. Shladover, "Path at 20-history and major milestones," *IEEE Trans. Intell. Transport. Sys.*, vol. 8, pp. 584 –592, dec. 2007.

[3] X. Wang and Y. Wang, "Coordinating power control and performance management for virtualized server clusters," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, pp. 245–259, Feb. 2011.

[4] J. Zhao and D. Tang, "Coordination traffic control under the framework of multi-agent technology," in *Proceedings of the 2009 Second International Workshop on Computer Science and Engineering - Volume 01*, pp. 219–222, 2009.

[5] J. Zhao, D. Tang, X. Geng, and L. Jia, "Urban arterial traffic coordination control system," in *Proceedings of the 2010 international conference on Artificial intelligence and computational intelligence: Part II*, pp. 275–283, 2010.

[6] E. Ebrahimi, O. Mutlu, C. J. Lee, and Y. N. Patt, "Coordinated control of multiple prefetchers in multi-core systems," in *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*, pp. 316–326, 2009.

[7] S. Melzer and B. Kuo, "A closed-form solution for the optimal error regulation of a string of moving vehicles," *IEEE Trans. on Automat. Contr.*, vol. 16, pp. 50–52, 1971.

[8] Y. Zou, *Distributed Control of Multiple Vehicle Systems using Constraint Forces*. PhD thesis, Oklahoma State University, Stillwater, OK, USA, 2008.

[9] S. E. Shladover, "Longitudinal control of automotive vehicles in close-formation platoons," *Journal of Dynamic Systems, Measurement, and Control*, vol. 113, no. 2, pp. 231–241, 1991.

[10] G. Doug and C. Daniel, "Air-cooled high-performance data centers: Case studies and best methods." White Paper, November 2006.

[11] Thanh, "Delayed Green Arrows." `http://hburgnews.com/2008/11/25/delayed-green-arrow/`, November 2008.

[12] A. Turner, "Tomorrow's engineers give air travel the bird." `http://www.flightglobal.com/blogs/future-proof/2009/05/-windowless-cabins-in-passenger.html`, May 2009.

[13] R. Vidal and S. Sastry, *Vision-Based Detection of Autonomous Vehicles for Pursuit-Evasion Games*. July 2002.

[14] S. Yadlapalli, S. Darbha, and K. Rajagopal, "Information flow and its relation to stability of the motion of vehicles in a rigid formation," *IEEE Trans. on Automat. Contr.*, vol. 51, pp. 1315 –1319, aug. 2006.

[15] P. Barooah, P. G. Mehta, and J. ao P. Hespanha, "Mistuning-based decentralized control of vehicular platoons for improved closed loop stability," *IEEE Trans. on Automat. Contr*, vol. 54, pp. 2100–2113, September 2009.

[16] H. Hao, P. Barooah, and P. G. Mehta, "Stability margin scaling laws for distributed formation control as a function of network structure," *IEEE Trans. on Automat. Contr.*, vol. 56, pp. 923–929, January 2011.

[17] R. Sepulchre, D. A. Paley, and N. E. Leonard, "Stabilization of planar collective motion with limited communication," *IEEE Trans. Automat. Contr.*, vol. 53, pp. 706–719, April 2008.

[18] D. A. Paley, N. E. Leonard, and R. Sepulchre, "Stabilization of symmetric formations to motion around convex loops," *Systems and Control Letters*, vol. 57, no. 3, pp. 209 – 215, 2008.

[19] S. Darbha and P. R. Pagilla, "Limitations of employing undirected information flow graphs for the maintenance of rigid formations for heterogeneous vehicles," *International Journal of Engineering Science*, vol. 48, no. 11, pp. 1164 – 1178, 2010.

[20] T. S. No, K.-T. Chong, and D.-H. Roh, "A lyapunov function approach to longitudinal control of vehicles in a platoon," in *Vehicular Technology Conference Proceedings, 2000. VTC 2000-Spring Tokyo. 2000 IEEE 51st*, vol. 1, pp. 336 –340 vol.1, 2000.

[21] J. A. Rogge and D. Aeyels, "Vehicle platoons through ring coupling," *IEEE Trans. on Automat. Contr.*, vol. 53, pp. 1370–1378, July 2008.

[22] S. Klinge and R. H. Middleton, "String stability analysis of homogeneous linear unidirectionally connected systems with nonzero initial conditions," in *Signals and Systems Conference (ISSC 2009), IET Irish*, pp. 1–6, December 2009.

[23] J. A. Marshall, M. E. Broucke, and B. A. Francis, "Formations of vehicles in cyclic pursuit," *IEEE Trans. on Automat. Contr.*, vol. 49, pp. 1963–1974, November 2004.

[24] H. Yu and Y. Wang, "Coordinated collective motion of groups of autonomous mobile robots with directed interconnected topology," *J. Intell. Robotics Syst.*, vol. 53, pp. 87–98, sep 2008.

[25] R. M. Gray, "Toeplitz and circulant matrices: A review," tech. rep., 2006.

[26] M. Fadali and Y. A. Ghoneim, ""2-d modeling, control, and simulation of vehicle platoons"," in *American Control Conference, 1993*, pp. 3187 –3191, june 1993.

[27] S. Seshagiri and H. Khalil, "Longitudinal adaptive control of a platoon of vehicles," in *American Control Conference, 1999. Proceedings of the 1999*, vol. 5, pp. 3681 –3685 vol.5, 1999.

[28] S. Darbha., J. Hedrick, and S. Choi, "Direct adaptive longitudinal control of vehicle platoons," *Vehicular Technology, IEEE Transactions on*, vol. 50, pp. 150 –161, jan 2001.

[29] J.-M. Contet, F. Gechter, P. Gruer, and A. Koukam, "Application of reactive multiagent system to linear vehicle platoon," in *Tools with Artificial Intelligence, 2007. ICTAI 2007. 19th IEEE International Conference on*, vol. 2, pp. 67 –70, Oct. 2007.

[30] J. Hederick, M. Tomizuka, and P. Varaiya, "Control issues in automated highway systems," *Control Systems, IEEE*, vol. 14, pp. 21–32, 1994.

[31] R. Diestel, *Graph Theory*, vol. 173. Springer-Verlag, Heidelberg, 3 ed., 2005.

[32] S. Darbha. and J. Hedrick, "String stability of interconnected systems," in *American Control Conference, 1995. Proceedings of the*, vol. 3, pp. 1806 –1810, jun 1995.

[33] S. Darbha and P. R. Pagilla, "Limitations of employing undirected information flow graphs for the maintenance of rigid vehicular formations," *ASME Conference Proceedings*, vol. 2009, no. 48937, pp. 731–738, 2009.

[34] J. A. Rogge, *Dynamic Behavior of Oscillator Networks*. PhD thesis, Universiteit Gent, Technologiepark Zwijnaarde 914 B-9052 Zwijnaarde België, 2006.

[35] T. Richardson, "Non-mutual captures in cyclic pursuit," *Annals of Mathematics and Artificial Intelligence*, vol. 31, pp. 127–146, 2001.

[36] K. Galloway, E. Justh, and P. Krishnaprasad, "Geometry of cyclic pursuit," in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pp. 7485 –7490, December 2009.

[37] B. Bamieh, F. Paganini, and M. Dahleh, "Distributed control of spatially invariant systems," *IEEE Trans. on Automat. Contr.*, vol. 47, pp. 1091 –1107, jul 2002.

[38] D. R. Isenberg and Y. P. Kakad, "A Decoupled Dynamical Model for Differentially Driven Mobile Robots," in *Proceedings of the IEEE SoutheastCon 2010*, pp. 428–432, 2010.

[39] C. H. Papadimitriou, "The euclidean travelling salesman problem is np-complete," *Theoretical Computer Science*, vol. 4, no. 3, pp. 237 – 244, 1977.

[40] E. L. Lawler, J. K. Lenstra, A. H. G. R. Kan, and D. B. Shmoys, *The Traveling Salesman Problem, A Guided Tour of Combinatorial Optimization*. New York, NY, USA: John Wiley and Sons Ltd, 1985.

[41] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1982.

[42] C. E. Miller, A. W. Tucker, and R. A. Zemlin, "Integer programming formulation of traveling salesman problems," *J. ACM*, vol. 7, pp. 326–329, Oct. 1960.

[43] E. L. Lawler and D. E. Wood, "Branch-and-bound methods: A survey," *Operational Research*, vol. 14, pp. 699–719, 1966.

[44] W. Eastman, *Linear programming with pattern constraints.* Harvard University, 1958.

[45] J. Munkres, "Algorithms for the assignment and transportation problems," *J. Soc. Indust. Appl. Math*, vol. 5, March 1957.

[46] M. E. Kurz, *Heuristics for the Traveling Salesman Problem.* John Wiley & Sons, Inc., 2010.

[47] N. Christofides, "Worst-case analysis of a new heuristic for the traveling salesman problem," *SIAM Journal on Computing*, vol. 6, no. 3, pp. 563–581, 1976.

[48] P. Petrov, "Modeling and adaptive path control of a differential drive mobile robot," in *Proceedings of the 12th WSEAS international conference on Automatic control, modelling & simulation*, pp. 403–408, 2010.

[49] H. Jia-Sheng, T. Mi-Ching, H. Feng-Reng, and H. Yoichi, "Robust control for coaxial two-wheeled electric vehicle," *Journal of Marine Science and Technology*, vol. 18, no. 2, pp. 172–180, 2010.

[50] G. Indiveri, A. Nuchter, and K. Lingemann, "High speed differential drive mobile robot path following control with bounded wheel speed commands," in *Robotics and Automation, 2007 IEEE International Conference on*, pp. 2202 –2207, april 2007.

[51] A. Albagul and Wahyudi, "Dynamic modeling and adaptive traction control for mobile robots," in *Industrial Electronics Society, 2004. IECON 2004. 30th Annual Conference of IEEE*, vol. 1, pp. 614 – 620, nov. 2004.

[52] V. V. Evgrafov, V. V. Pavlovsky, and V. E. Pavlovsky, "Dynamics, control, and simulation of robots with differential drive," *Journal of Computer and Systems Sciences International*, vol. 46, no. 5, pp. 836–841, 2007.

[53] G. Klančar, D. Matko, and S. Blažič, "Wheeled mobile robots control in a linear platoon," *J. Intell. Robotics Syst.*, vol. 54, no. 5, pp. 709–731, 2009.

[54] S. Ian, "Navigation and control of an autonomous vehicle," Master's thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 2005.

[55] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," in *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, pp. 384 –389 vol.1, may 1990.

[56] "8-bit atmel microcontroller with 256kbytes in-system programmable flash." `www.atmel.com/Images/doc2549.pdf`, 2011.

[57] "Arduino Mega 2560." `arduino.cc/en/Main/ArduinoBoardMega2560`.

[58] "XBee Multipoint RF Modules: Embedded RF Modules for OEMs." `www.digi.com/pdf/ds_xbeemultipointmodules.pdf`.

[59] "XBee/XBee-PRO RF Modules." `ftp1.digi.com/support/documentation/90000982_G.pdf`, 2012.

APPENDIX A

TOEPLITZ MATRICES

Toeplitz matrices are an important class of matrices in engineering applications. A Toeplitz matrix is defined as

Definition A.1 *Any matrix $T \in \mathbb{C}^{n\times n}$ which is diagonal constant i.e., the elements on any diagonal are equal is called a Toeplitz Matrix. It has the form*

$$\begin{bmatrix} t_0 & t_1 & t_2 & t_3 & \cdots & t_n \\ t_{-1} & t_0 & t_1 & t_2 & \cdots & t_{n-1} \\ t_{-2} & t_{-1} & t_0 & t_1 & \cdots & t_{n-2} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ t_{-n} & t_{-(n-1)} & t_{-(n-2)} & t_{-(n-3)} & \cdots & t_0 \end{bmatrix}$$

These kind of matrices are common in discrete time systems [25]. The discrete time causal time-invariant filter is an example application of the above matrix. A symmetric Tri-diagonal matrix, Identity matrix are some special examples of Toeplitz matrices.

There are several special cases of Toeplitz matrix the most important of which is a Circulant matrix presented in Definition 2.7 in Chapter 2. Circulant matrices can be used to analyze the behaviour of otherwise complex Toeplitz matrix. The following Lemma is from [25].

Lemma A.1 *Let* $T_n(f) = t_{k-j}$ *where*

$$\sum_{k=-\infty}^{\infty} |t_k| < \infty$$

and

$$f(\lambda) = \sum_{k=-\infty}^{\infty} t_k e^{ik\lambda}, \hat{f}_n(\lambda) = \sum_{k=-(n-1)}^{n-1} t_k e^{ik\lambda}$$

Define the circulant matrices $C_n(f)$ *and* $C_n(\hat{f}_n)$*. Then,*

$$C_n(f)\ C_n(\hat{f}_n)\ T_n \tag{A.1}$$

This means that there is a sequence of circulant matrices $C_n(f)$ that is asymptotically equivalent to the sequence $T_n(f)$. This is useful since we have eigenvalues, inverses and products of the circulant matrices that are known exactly and are easy to analyze than the Toeplitz matrices themselves.

APPENDIX B

ROBOT SPECIFICATIONS

The following tables gives all the specifications of various components of the robot.

Property	**Value**
Length of base plate	9 in
Width of base plate	7 in
Height of base plate from ground	2.48 in
Weight of robot	5 lbs
Speed	3.75 ft/s
Battery	12V @ 2800mAh
Wheel Diameter	2.75 in
Drive Motors	
Rated Voltage	12 V
Torque	110 oz-in (8 kg-cm)
Speed	350 rpm
Free run current	300mA
Gear ration	29:1
Stall current	5 A
Encoder Resolution	64 CPR
Output shaft resolution	1856 CPR

Table B.1: Specifications of Robot

Property	**Value**
Controller	
Board	Arduino Mega 2560 R1, R3
Micro controller	ATmega2560
Type	8-bit
Clock Speed	16MHz
Flash	256 KB
Voltage	5V
Motor Controller	DFrobot DRI00009
Amplifier	L298P
Voltage, Max. Current	7-12V, 2A
Channels	2
Wireless Communication	XBee Series 1
Range	100ft indoor
Transmission Rate	250kbps Max

Table B.2: Specifications of Controller

VITA

Shyamprasad Konduri

Candidate for the Degree of

Master of Science

Thesis: COORDINATION OF MULTIPLE AUTONOMOUS VEHICLES WITH DIRECTED COMMUNICATION GRAPHS

Major Field: Mechanical and Aerospace Engineering

Biographical:

Personal Data: Born in Hyderabad, Andhra Pradesh, India on December 15, 1985.

Education:

Received the B.Engg. degree from Osmania University, Hyderabad, Andhra Pradesh, India, 2007, in Mechanical (Production) Engineering
Completed the requirements for the degree of Master of Science with a major in Mechanical and Aerospace Engineering Oklahoma State University in July, 2012.

Experience:

Research Assistant at Oklahoma State University from June 2009 to present; Teaching Assistant at Oklahoma State University from August 2008 to May 2009 and January 2010 to May 2010; Programmar Analyst Trainee at Cognizant Technology Solutions India Ltd from October 2007 to July 2008.

Professional Memberships:

Student Member
American Society of Mechanical Engineers
International Society of Automation

Name: Shyamprasad Konduri Date of Degree: July, 2012

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: COORDINATION OF MULTIPLE AUTONOMOUS VEHICLES WITH DIRECTED COMMUNICATION GRAPHS

Pages in Study: 91 Candidate for the Degree of Master of Science

Major Field: Mechanical and Aerospace Engineering

Scope and Method: A considerable amount of research has been conducted on coordination control of vehicle formations in the last few decades due to its possible applications in numerous areas, such as patrolling, search and rescue, vehicle platoons, etc. Information flow between vehicles is vital for the formation to function properly. Controllability and stability of the formation depends on the type of information flow that is employed between vehicles. This work focuses on using a type of directed communication graph, called the ring graph, for coordination control of vehicle platoons. A detailed study of literature is undertaken to clearly understand the limitations of the various types of undirected graphs in terms of stability and scalability. The use of ring graph in vehicle platoons is investigated. The problems associated with the ring graph in its basic form related to communication sensor range and the scalability of ring graph are investigated. Methods to create a ring graph topology that requires minimal communication distance over a given formation are also investigated. A survey of literature was undertaken to investigate the formulation of the Traveling Salesman Problem (TSP), constraints involved in TSP, methods to solve TSP including exact and approximate algorithms. The problem of finding a ring graph for a given vehicle formation is addressed by formulating this problem as a special case of the TSP.

Findings and Conclusions: Properties such as stability and scalability of the formation with a controller based on the ring graph are studied. Lack of scalability of the controller that is based on the ring graph is discussed. Alternate ring graph topologies which address the practical issues such as communication range of transmitters are proposed. Simulations which show that the alternate ring graphs have similar properties as the basic ring graph are carried out. Controller based on multiple ring graphs on a vehicle platoon and its advantages over the single ring topologies are studied. An algorithm based on the branch and bound method is proposed to solve the modified TSP and to generate a ring graph for the given vehicle platoons, or two- and three-dimensional vehicle formations. An experimental platform consisting of multiple homogeneous autonomous robots is developed. A coordination controller using ring graph is implemented on the experimental platform and results for a number of platoon initial condition scenarios are presented.

ADVISOR'S APPROVAL: Dr. Prabhakar R Pagilla