CHAPTER 1

INTRODUCTION

1.1 TRUCK TRAILER ROBOT SYSTEMS

One of the interesting researches in the area of service robotics is hitching of passive trailers to a mobile robot (Fig.1.1). Such robot systems are more reliable than a single robot as it provides extra space which can be used for other applications. They can be used for material removal in industries or in load haulage or luggage cart (Land and King, 1994), military rescue operations like mobile first aid kit or in equipment operation such as road paving, agriculture, or lawn moving (Larsson et al, 1994). The operational costs of employing such robot systems are very cheaper compared to that of two individual mobile robots. Also, the trailers can anytime be detached from the robot to make use of the robot individually for other purposes. Hence, these robot systems have received much attention in the area of field robotics.

However, the motion control of a truck-trailer system is a complex problem, because of its highly nonlinear kinematics. This is because wheeled mobile robots are nonholonomic in nature and so by adding a passive trailer to a mobile robot adds more nonholonomic constraints to the system. DeSantis et al. in 1998 mentioned that if an n-trailer system has velocity inputs or actuators lesser than the degrees of freedom, it is underactuated. Control Engineers have shown enormous research interest in this area by working with motion control problems such as trajectory tracking, point stabilization and path planning for such nonlinear systems with or without external perturbations. These problems are comparatively easier when moving at a forward velocity. But when the robot is driven backwards, or at faster speeds or during sharp turns, the system becomes open loop unstable and pose great challenge to control. This is because the motion of the passive trailer connected to the robot by a physical link cannot be controlled by the driving robot. This results in skidding and ends up hitting the robot. This phenomenon is known as jack-knife shown in figure 1.2. Most of the research work on truck trailer systems is focused on the backward motion control. Moreover, an effective controller which addresses this problem can also be deployed to control heavy articulated vehicles. So, a significant development in this problem will also cause a milestone in transportation industry also. Though this work is not on truck-trailer systems, the focus is mainly on formation control of a multi robot system which imitates a real truck-trailer system.

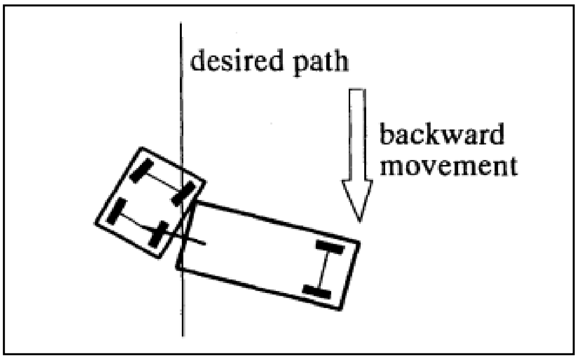


Fig 1.1. Radio controlled Truck-Trailer Robot (Altafini et al 1999)

Fig.1.2. Jack-Knife Phenomenon

1.2 MULTI -ROBOT SYSTEMS

A group of autonomous robots which operate cooperatively are termed as Multi-robot system. Advantage of employing such systems is that they can accomplish tasks far beyond the capabilities of a single robot (Kumar et al. 2008) which reduce the operational costs. Moreover, when two or more robots are used, the task is achieved much faster compared to employing a single robot. Figure 1.3 explains different set of multiple robot systems. Applications of multi robot systems are many (Yang et al 2008, Anderson et al. 2008, Hao & Agarwal 2005) and involve different fields such as industrial and military platoons, service robotics, etc. Challenging and difficult cooperative tasks, such as box pushing, rescue operation, navigation in unstructured environment, disaster monitoring and environmental surveillance etc. (Fabio Morbidi, 2008) can only be achieved effectively by use of robust multi-robot systems.

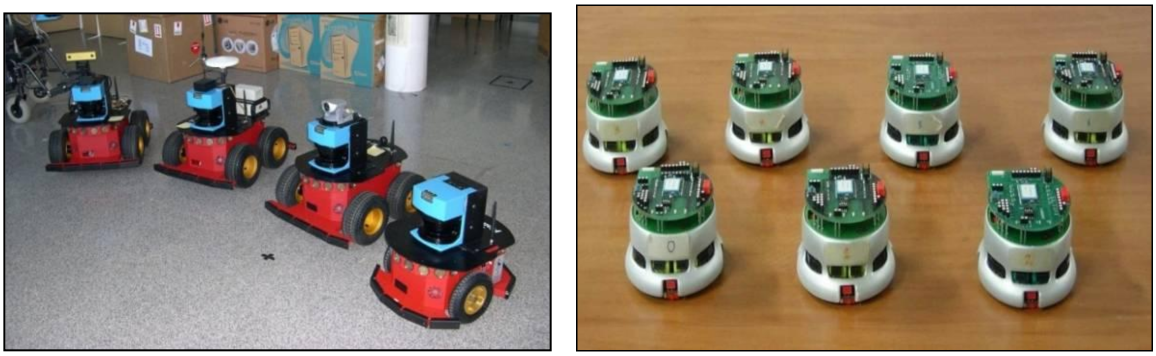


Fig. 1.3. Multi Robot System (Group of Pioneer P3ATs & Khepera II robots)

These systems are inspired by behavior of biological systems like swarm of bees, ants and flock of birds, etc., where each member interacts with the other member in the system to move effectively in a spatial formation. Similarly, each robot in the multi-robot system communicates their position, relative velocity or any other information with other robots to maintain the formation. Here, formation is defined as a specific arrangement of robots, which has to be maintained, when they move as a whole by controlling the position and orientation of individual robot to complete the specified task. So, in order to solve a formation problem, the group of mobile robots should have a proper formation model and a robust controller that is immune to changing environmental conditions. The formation control problem has been dealt by many researchers in recent years, because of its wide range of applications and that it is not restricted to the type of robots (like formation of a group of aerial vehicles, mobile robots, wall climbing robots etc.), where all together perform a specific task.

The different control strategies used for controlling the formation pattern are: Graph Theory (Barca et al., 2012), Vector Potential Field (Masoud 2007), Virtual Structure (Yuan 2010), Leader Follower (Dai and Lee 2012) and Behaviour based control (Arkin, 1998). Out of these methods, leader – follower control and behaviour-based control are the most widely used method. Active obstacle avoidance by follower robots is also an important challenge faced in formation control (Chaimowicz et al., 2004).

A reactive approach based formation control was developed by Chetty et al., 2010, which was specific to a single robot and cannot be employed for other robots. Many of the models and the controllers (Desai et al., 1998) in the literature used polar coordinates, which has potential singularities. Other type controllers used by Das et al. (2002) and Sanchez et al., (2003) were only locally stable and complicated to design. Hence, there is a need for a new control strategy that can address formation control problems in multi-robot system.

1.3 MOTIVATION AND RELATED WORK

There has been considerable amount of work done in addressing the problems of truck-trailer system (JL Martinez et al. 2002, Jin Cheng et al. 2009). But there are many applications where the problems like jack-knifing, multiple trailers stability, parking space constraints, etc. cannot be compromised at all. So, this work is focussed on addressing the problems of truck-trailer system in a new perspective by using multi robot system. Employing multi-robot system is costlier comparatively but all the problems of the truck-trailer system can be successfully addressed when two individual robots can be treated as a truck-trailer robot in a multi robot system. Hence, the focus on this research is to develop a rugged formation control for two mobile robots that can imitate a real truck-trailer system. The inline formation of two mobile robots can be treated as truck-trailer robot with a virtual hinge between them (Kuppan et al 2010). By treating this problem as a formation control problem, the problems of truck-trailer system can be avoided by adjusting length of the virtual hinge accordingly. Hence, this work combines the advantages of both the truck-trailer system and multi robot systems. This new approach cannot replace the conventional truck-trailer system and its stability problem completely because the multi robot system is costlier comparatively. Moreover, the kinematics of a conventional truck-trailer system is completely different than the kinematics of formation control model of two robots. But this work aims to suggest an alternative solution to the problems of truck-trailer system in certain applications where these problems cannot be compromised to the cost.

Moreover, one of the robots in the formation is made as leader and this robot commands the motion of the other robot in the group and they follow the leader robot's instructions. Similarly, the truck robot is treated as leader and the trailer robot is treated as follower. Hence, the position and orientation of the trailer robot is controlled by the truck robot. So the trailer must adopt a sense-think-act behaviour to deal with both dynamic obstacles as well as the spatial formation as commanded by the truck robot. To achieve this, a behaviour based formation approach is adapted by the trailer. Hence, a rugged formation framework is developed that can control the two robots which are coupled with a sense-think-act style reconfigurable layered architecture.

Another motivation for this work is to make the robots move in different shapes especially in parallel formation that can be used for agricultural purposes. Parallel movement of two robots has also been worked out by many authors (Kumar et al 2008) but many controllers did not adapt to different shape due to singularity problems. Also, there are space constraints to accommodate such large articulated robots. Application of proposed virtual link based control method in parallel and other formation shapes expands its use with many applications. The controller is expanded for multi-robot system also. Moreover, switching of formation shapes from one to other makes this a flexible system, where both the robots break the formation anytime and can park safely in the space available.

1.4 OBJECTIVES AND SCOPE OF WORK

Looking at the above problems and different approaches taken by various researches, there is a need for a suitable controller for a multi-robot system that can imitate a real truck-trailer robot and also can overcome the problems of conventional truck-trailer system. The main objective of this work is to develop a robust virtual link control strategy for the formation control of multi robot system. Added to that, a hybrid multi-layered approach is also developed to imitate a real truck-trailer system.

The scope of the work is as follows:

- To design a virtual link based control strategy for an inline formation of two mobile robots.

- To develop a hybrid multi-layered control architecture with layers of formation control, obstacle avoidance and collision avoidance with other robots.

- To extend the developed control strategy for multiple mobile robots for different formation shapes – parallel, random, and wedge shapes.

- To investigate the effectiveness of the developed control strategy using simulation tools.

- To experimentally investigate the developed control strategy using the available mobile robots and validate the results.

- To study the stability, robustness and disturbance rejection capability of the formation controller and also to improve the performance of complete system by optimizing the controller gains.

1.5 SUMMARY

The entire content of this work is organized into five chapters from literature survey to results and discussions. Basic ideas and definitions in the area of truck trailer system (TTMR), wheeled mobile robots (WMR) and Multi robot systems (MRS) are detailed in the Chapter 2. This chapter will give the reader better insight in the field of mobile multi robot systems for the correct comprehension for the following chapters. This chapter presents the survey on truck-trailer robots, their motion control problems, concept of non-holonomy, kinematic dynamic models of WMR and the basics of multi- robot formation, various control strategies, their advantages and disadvantages. It also summarizes all the research literature done in this area so far.

Chapter 3 describes the details on the control methodology of the multi-layered behaviour framework developed for dealing the virtually linked truck-trailer robot problem, with formation control and dynamic obstacle avoidance problem in an unstructured environment. It gives a detailed account on layered formation control approach, behavioural components used, formulation of behaviours and layers of the control architecture, kinematic model of the required system and mathematical formulation of tracking controller derived to keep the robots in closed defined formation in an unknown environment.

Chapter 4 details both the simulations and experimental studies done to evaluate the performance of this methodology. Detailed analysis of the linear and angular errors between the robots has been done. A brief description about the experimental setup comprising of robot research platforms, simulation environment, experimental architecture, implementation and description regarding the experiments are given. Further, the performance of the approach is measured through real experiments using commercially available robots and the results area analyzed and discussed.

Chapter 5 goes in to the detail of the formation controller. Optimization of the control gains is also done to get a better control. The robustness of the controller is studied by adding more number of trailers of different configuration to the present system. Studies are also done on the stability of the system by looking at different formation shapes. Attempts have also been done by use of localisation for a known environment to avoid initial input to the system. In Chapter 6 the concluding remarks on the performance of the approach are summarized and the main contributions of the thesis, possible directions of future research are highlighted.

CHAPTER 2

REVIEW OF LITERATURE

This chapter covers the literature survey conducted on both service robotics as well as multi-robot system. Since, the control strategy presented in the following chapters has been implemented with differential-drive wheeled mobile robots, this chapter gives a brief introduction to the concept of nonholonomy and how it gives rise to the problems in truck-trailer system. Finally, the problems and present issues on both these areas that has motivated to the current research work has been explained.

2.1 NON-HOLONOMIC CONSTRAINTS

Wheeled Mobile Robots (WMRs) are classified as non-holonomic systems according to non-linear systems. Due to this property, motion control of WMRs is a difficult task and for the past two decades many researchers have worked in this concept. Non-holonomic path planning is considered a research problem that includes research in the areas of: control theory, differential geometry and classical mechanics. In this section, we shall begin with understanding the basics of non-holonomic constraints and how it restricts the motion of the system. The practical problems caused due to this phenomenon have also been mentioned.

2.1.1 Nonholonomy

Non-linear systems are classified into holonomic and non-holonomic systems. A system is said to be holonomic, if it is possible to write all the constraints of the system as $f(q,t)=0$, where q is the system coordinate with respect to the reference frame and t is the time. But, if the constraints of a system are expressed as $f(q,\dot{q},\ddot{q},t)=0$ and cannot be reduced to *f(q,t)*=0, then the system is called as a non-holonomic system. By definition, it refers to the set of non-integrable differential equations, which describes the restriction on the motion of the system, or in simple words, the total degrees of freedom are greater than the controllable degrees of freedom.

A two-wheeled differential drive mobile robot is the best example for non-holonomic system. The kinematic model of this system gives the equation $\dot{x}\sin\theta - \dot{y}\cos\theta = 0$, which is a non-integrable equation. If the system is assumed to be having a rolling contact, then this equation denotes there is no lateral slip condition. But from experience, we know that the two wheeled robot or car can be controlled in a 3D space, but in reality there are only two inputs. According to Brockett theorem (Brockett 1981), nonholonomic systems cannot be stabilized to a desired posture via differential or state feedback. So, alternative approaches like discontinuous feedback, smooth time-varying feedbacks, etc. has been proposed.

Similarly, a multi body articulated vehicle (truck-trailer system) is also termed as a non-holonomic system. This is because of the rolling constraints of the wheel, which contributes more degrees of holonomy making the system underactuated. For a system with '*n*' trailers, there are two velocity inputs, but *n+3* generalized coordinates. Because of this, motion control of truck-trailer robots is also uncontrollable. Backward motion of this non-holonomic trailer system is an open loop unstable system. The trailer skids and goes off the track and ends up by jack-knifing with the WMR.

2.1.2 Jack-Knife Phenomenon

Truck with passive trailers or articulated vehicles experience a motion control problem during sharp turning, backward movements and at higher speeds. This is due to the passive trailer being pulled by the truck using a physical link, whose motion cannot be controlled by the truck. The trailers usually skid under these conditions and end up hitting with the truck. This phenomenon is called as Jack-knifing. The motion control problem is challenging, because it is a four degree non-holonomic system and the kinematics are hard to compute. As mentioned earlier, it is an unstable dynamic system and the input constraints drive the trailer to jack-knife with the truck, even if we make the system move in a straight line as shown in Fig.2.1. Though, many researchers are working in this area of developing control system for non-holonomic WMRs with trailers, an optimized controller have not been proposed yet.

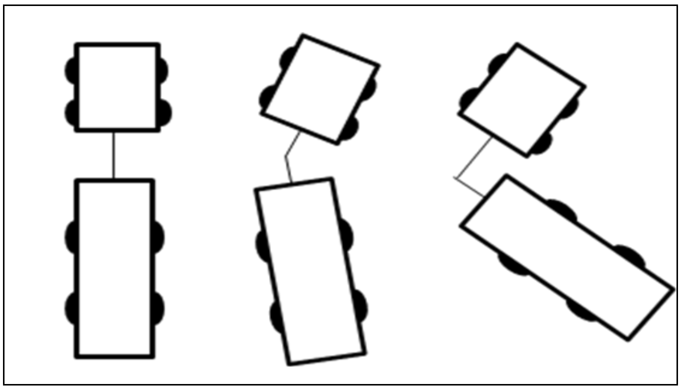


Fig.2.1 Jack-Knife Phenomenon

2.2 TRUCK-TRAILER MOBILE ROBOTS

A four-wheeled robot (car-like) with a trailer is treated as a non holonomic system with 4 dimensions and motion control for such system is very hard compared to a single robot. Backward movement of this system is a great challenge in the area of non-linear controls. Presently, there are many researchers working in this area and trying to propose effective controllers. Many practical problems such as parallel parking, jack-knife phenomenon, etc. have be addressed, if a robust motion control strategy to be effective in controlling these systems.

This section presents a review on the motion control strategies used for Truck-Trailer Mobile Robots (TTMR) to address problems like jack-knife and parallel parking. Most of the controllers designed so far have relied on non-linear control theory. Considerable amount of work is also done in this area using modern control strategies like neuro-fuzzy logy and control learning approaches. Studies are also done on articulated vehicles and their control methodologies.

2.2.1 Jack-knife problems in TTMR

Backward movement of the trailer is very important but at the same time, very complicated due to Jack-knife phenomenon. It occurs, when parking in constrained spaces, the sudden approach of dynamic obstacle obstructing all possible forward movements, loading docks, etc. In addition, this works involves design of a non-linear control system. Other solutions include new concepts in differential geometry, fuzzy logy, neural networks and linear algebra.

Non-linear Control Methods: Many researchers adopted general techniques of non-linear control methodologies since, 1990s like linear approximation, linearization, state feedback law, tracking control law, etc. This section gives a brief introduction of the various researches that has taken place over a decade in this area.

Path tracking of a TTMR was done as early as in 1991 by Sampei et al. A tracking controller is designed for the trailer using the exact linearization method and time scale transformation. To achieve this, a dynamic model of the robot is derived using state equation with the distance along the desired path as time scale. The state equation is then linearized with state transformation. A linear controller is designed for this linearized state equation system with a feedback controller of regulator and servo controller type. Further, time scaling is done to get a desired velocity-dependent controller. However, this method cannot track circular arc more accurately as the path is approximated as a sequence of straight lines. This was later carried out using radio controlled vehicle and CCD camera for backward driving of the trailer by Sampei et al. in 1995. However there was no path planning in this work.

Later, Divelbliss and Wen (1997) worked on path tracking using three step method: the trajectory is generated for satisfying the non holonomic constraints and the linear kinematic model of the car and this trajectory is stretched till the maximum velocity constraint is satisfied. In 1998, Doh-Hyun and Jun-Ho devised a globally asymptotically stable (GAS) tracking control law for the trajectory tracking of the trailer. This work was further modified by them in 1999 for backward movement of the trailer system. In 2002, JL Martinez et al. have used a pure geometric technique for path tracking of mobile robots. This technique is an extension of an efficient path tracking method employed for non-holonomic mobile robots to a TTMR. This avoids problem for cases where tracks are employed instead of wheels or when working with irregular terrain. This technique is simpler as it does not involve conversion of complex kinematic models into control laws. But the backward movement problem has been addressed only by limiting the demanded curvature and dynamic effects when working with heavy loads are also not considered.

Fuzzy and Neural Networks: Mostly, all the works which employed fuzzy and neural networks was analyzed in simulation and not through experimental work. In 1989, Nyugen and Widrow proposed a self-learning technique using neural networks to achieve non-linear controller design. Here, it addresses the backward movement of a truck with trailer by controlling the steering angle of the truck from an arbitrary initial position. This approach involves two stage learning process; first one is to train the emulator and the second one is to train the controller using emulator. Training of the two layered emulator is done to make it understand how the real truck-trailer behaves. A wide range of 'positional states' and 'steering angles' are given to the emulator, so that it generates the next positional state vector by learning. Later, training of the controller was done using a supervised back-propagation algorithm. However, the problem with this approach was it involved thousands of backups to train the network. Control decisions made earlier had significant effects upon final results.

Later in 1992, SG Kong and Kosko developed an adaptive fuzzy controller to address this problem. The supervised back-propagation algorithm trained the neural system. Error nulling intuitions and common-sense were used to generate Fuzzy-associate-memory (FAM) rules. The speed with which the DCL clustering technique which recovers the underlying FAM bank and the robust performance of the system even when 50% of the FAM rules removed, showed that this method can be used for controlling more complex higher dimensional problem.

In the same year, Tokunaga and Ichihashi proposed a neuro-fuzzy optimal control which is based on the mathematical model of the system. The trajectories of the truck-trailer system are represented by a fuzzy model with Gaussian Membership functions. An optimized controller is developed for backing up a trailer to a loading dock.

Later in 2009, Jin Cheng et al. designed a fuzzy control with a line of sight (LOS) guidance method to address this complex system. Here, the motion planning of the system is done by denoting the line segments (path) as way points. The "way point trajectory system" helps in driving the TTMR with reference speed, when the path cross track error is least. A LOS vector is taken along with the reference position. When the TTMR moves, these values are updated. For the trailer to track this vector, a fuzzy logic controller is derived, thus reducing the complexity of using a trajectory control.

Differentially Flat Systems: Fliess et al. in 1993 first studied about systems that exhibited the property of differential flatness or linearizing output. A control system is said to be differentially flat, if there exists a finite set of differentially independent variables, which are differential functions of the system variables and vice versa. Flatness can be considered as a non-linear extension of Kalman's controllability. These systems can be linearizable via a special type of feedback called 'endogenous'. In 1993, Rouchon et al devised an open loop generation strategy such that the general n-trailer system is flat. With the use of Frenet formula, this strategy was successful and the linearizing output is taken to the middle of the axle of the trailer. The path planning problem without obstacles is easy when the linearizing output is determined. This is because the reference trajectory and the open loop control are expressed in terms of linearizing output and its finite number of derivatives. Lamiraux and

Laumond in 2000 also proposed a new steering system for path planning that can be used with any differentially flat drift less system with two inputs. The method was devised such that the system satisfies topological property, which can account for small-time controllability of the system. By doing so and implementing on Hilare robot with trailer, it was found that unlike sinusoidal inputs, this method did not have unnecessary cusp points.

2.2.2 Multiple Trailers

As mentioned, the degree of the non-holonomy increases depending upon the number of trailers added to it. Hence, a mobile robot based truck-trailer system ends with stability problem, when added with additional number of trailers. Murray and Sastry (1990) proposed the chained form which was applied for a car with one trailer system and the convergence properties of the system were found to be satisfactory. But the algorithm failed when additional trailers were added. To overcome this problem, Sordalen et al. in 1993 shifted the coordinate frame to the last trailer and so the coordinates of the complete system is given by the position in the last trailer. Hence, the transformation of the coordinates enabled the chained form conversion as global in the position of the system and local in the orientations of the trailers. For stabilizing the chained form system, a time-varying feedback control law was introduced. This result proved positive as the system obtained asymptotic stability with exponential convergence to any desired configurations and the simulation results were provided for a car with 3 trailers.

Path planning of a Hilare (two-wheel drive) robot with a trailer was done by Lamiraux et al. in 1999. An integrated approach was developed with three tasks: 1) Computing a collision free feasible path–A random path planner which is probabilistic complete is used and new steering method based on differential flatness was proposed for the non holonomic systems. The planner mentioned in 1998 was used for approximation in any constrained environment. 2) Dynamical constraints of the system are included to transform the path into a trajectory - This path planning was done without any approximations on robot parameters. 3) Tracking of the trajectory is done - only in forward direction via geometric transformations. Later, Laumond et al. added non holonomic constraints step by step to for path planning of a TTMR in several levels. The algorithm was based on the concept of geometric approximation. But, the complexity of this planner increased, when the number of trailers was more.

Wen Li et al. in 1999 developed an index to estimate the manipulative difficulty of towing and pushing multiple trailers like the number of trailers, the length of each trailer or the relative position of the joints which is measured by applying an impulse disturbance to the system in a straight line motion.

In 1995, Tanaka and Yoshioka developed a fuzzy controller for backward trajectory tracking of a truck with five trailers. Depending upon the dynamics of the system, a fuzzy model was derived using fuzzy approximation method. Later, fuzzy rules were calculated by parallel distributed compensation (PDC) method. A trajectory fuzzy controller is designed using these rules and feedback gains. Tanaka et al. in 1998 further extended this work for a mobile robot with ten trailers. The stability condition is based on Lyapunov approach, which is expressed in terms of linear matrix inequalities (LMI's), so that the problem is just to find a common Lyapunov function for a set of Lyapunov inequalities. Stable feedback gains are found using convex optimization techniques. Simulation results showed that the fuzzy controllers successfully backed up the trailers, even in a difficult initial position and the fuzzy model developed imitated the dynamics of the original models. This work was the first one to work on more than three trailers.

In 1996, Hougen introduced ROLNNET (Rapid Output Learning Neural Network with Eligibility Traces) where the network partitions the input space by problem specification prior to learning. Results showed that the robot was able to back the truck successfully by learning quickly compared to other systems and with less computation time. Works were also carried out in 1997 with ROLNNET for backing up a robot with two trailers.

Myoungkuk et al in 2004 showed that a robot with n passive trailers can be controlled in backward direction by treating it as a trajectory following problem. The advantage of this method was that it can be employed for any multi-axle mobile robot also. Later, Matsushita and Murakami extended Saeki's work to double trailers in 2006. When the steering controller which was based on the observer and Lyapunov function of Saeki was used for two trailers, the tractor oscillated. So a new steering controller based on back-stepping was designed and the controller was verified using numerical results.

2.3 MULTIPLE MOBILE ROBOT SYSTEM

As seen in the last section, the major problems faced by truck-trailer robots have been tackled using various control strategies. This section will give a brief introduction on the control methodologies used for multi-robot formation systems. Further, it also summarizes the advantages and the disadvantages of the various control approaches found in the literature and the basis as which the scope of this thesis work is evolved.

Both these areas are analyzed closely so that the main scope of this thesis work can be brought up with respect to these systems.

The most important requirement in multi robot system is the motion coordination among the robots, so that all the robots keep a relative position with each other, as they move in a required spatial formation. Coordinating multiple robots requires a proper control design of an individual robot. This is because multi-robot control methods has grown out of individual robot control and is reflected in the control strategies at the collective group level. Thus, in order to achieve the coordination between multiple robots, it's important to understand the control motion of an individual robot. Hence, the motion control of a single robot which depends on the kinematics, dynamics of the formation shape is discussed here. But when multiple robots are used, the motion of the system depends on the kinematics of the whole system. Next section addresses the extensions of these approaches as applied to multi-robot systems.

2.3.1 Single Robot control

The robot control model explains the connection between the three basics of robotics – **Sense**: readings are taken from external sensors and are updated to the model, **Plan**: depending upon the sensor data from the world model and the required goal, the immediate actions are planned and **Act**: the planned actions are carried out by the planner. In short, the robot control model describes how the sensory data is processed and the decision is made based on it and is executed. There are four basic practical approaches which are as follows:

1. Reactive control
2. Deliberative/Hierarchical control

3. Behavior-based control and
4. Hybrid control

Reactive Control: It is a simple sense-act type of control. There are many occurrences of sense-act methods called as behaviors for various tasks of the robot. The robot has a combination of behaviors. All these methods/behaviors are run as simultaneous processes. The behaviors do no planning for the next action. They process the sensor data and estimate the required action to be executed irrespective of all other behaviors. Since there is a tight coupling between perception and actions by the use of behaviors, the robot can react very fast to any type of fast-changing or unknown environments (Agre & Chapman, 1987, Arkin, R.C., 1998). The major limitations to this approach are that such systems have no idea of the world model and do not store information which leads to its inability to learn dynamically (Mataric, 1992; Mataric, 1997).

Deliberative control: In this control, the robot work is concentrated more on planning. By sensing the world, plans the next step and reacts. So, for every move, the robot clearly plans about its next step and how it is to be executed. This is achieved by gathering all the sensing data obtained and are processed in to a single world model. This type of control strategy is good for situations with full knowledge about the environment like assembly line. However, this approach cannot react to fast changes and unexpected occurrences happen in real time (Brooks 1990b, Brooks 1991c).

Behavior based robotics: In this method, the robot has a group of processes or control laws that are designed for achieving certain subtasks/goals. Behaviors can be 'avoid-obstacles', which avoids collisions with obstacles or 'go-home' which sends the robot to the required home position or goal point. They are implemented as a procedure in software or as a processing element in hardware. Robot receives data from sensors like the camera, sonar, ultrasound, etc. for each behavior. They can also take sensor data from other behaviors in the system. Outputs to the robot's effectors like motors for wheels, grippers, etc are given by each behavior and they can give the output to other behaviors also. This makes the behavior-based control as a constituted network with many behaviors capable of interacting with each other. Moreover, each behavior can be treated as a ‘state’ and a group of behaviors can be treated as a ‘representation’ by networking. This is the advantage of using a behavior-based systems compared to a reactive system as the former have better learning capabilities. The only disadvantage of this system is the mathematical formulation of the different states and representations.

Hybrid control: This Robot Model is an advanced model compared to others and each primitive of the robot is a mixture of the different robot controls. During sensing, every sensor data is given to both the behavior and the planner, for building a global world model. Hence, this is a combination of reactive and hierarchical styles. Here, the robot plans to divide the required task into different subtasks known as “mission planning”. Then the robot plans (as in deliberative control) about the suitable behaviors required in achieving the individual subtask. After proper planning and allocating the behaviors, each sub task is executed according to the Reactive Paradigm. It attempts a compromise between purely reactive and deliberative approaches and provides an effective means of integrating world knowledge with reactive control (R.C. Arkin, 1994, Mataric, 1997).

The important difference between hybrid systems and behavior-based is in the representation of the control and how the different time-scales are handled. Hybrid systems are implemented with three layer architecture - the low-level reactive system working on a short time-scale, a high-level planner that functions on a long time-scale and the middle interaction layer which forms a bridge between both the high and low level layers. On the other hand, behavior-based systems group to form representations and have a uniform time-scale. This makes it easy to assist any other parallel demands on the system. These representations are implemented just like any other behavior in the system.

2.3.2 Multi Robot Control

For multi-robot control, formation problem requires multiple mobile robots to form up and move in a specific geometric pattern in a coordinated manner. They can be dealt using two approaches – centralized and decentralized. In a centralized system, a core unit serves the main purpose of the whole system by coordinating all the information received from other members with respect to the sensory information of the environment. It also takes all eventual decisions and communication with all other robots in its group.

The major problem with this approach is the failure of the core unit. In case of decentralized systems, the process computations are distributed among the robots since each robot is capable of sensing and communicating all the sensory information. This division of task among the robots makes the system more powerful and fault-tolerant as all the robots share information about the coordination. This also doesn’t involve huge computation capabilities for the core unit, since the overall task is distributed among all the robots. Moreover, the communication between the robots is robust and reliable in place of a core unit which transfers/communicates with other robots. (T. Balch and RC Arkin 1998). However, both these approaches are used depending upon the applications. Sometimes both the approaches are combined partially to exploit the advantages of both the system.

There are also different kinds of control strategies that are used for coordinated motion control of multiple mobile robots in the literature. Few of them are listed here:

1. Graph theory (Desai et al., 2001),

2. Vector potential field (Yamaguchi et al., 2001),

3. Virtual structure (Tucker Balch and Ronald C. Arkin, 1998),

4. Leader-follower (Alur R et al., 2002), and

5. Behaviour-based (Arkin, R., 1998)

Among all these approaches, the most commonly used ones are the virtual structure, behaviour based, and leader follower. The leader-following and virtual structure approaches are in centralized manner while, the behavior-based approach is a decentralized approach. This section details on each of these approaches.

Virtual Structure Approach: The idea of virtual structure was proposed by Tan, K.H. and Lewis (1997) and Tucker Balch and RC Arkin (1998). In this approach, the complete formation is treated as a single rigid body with fixed geometric relationship (Chena and Wang 2005). The disadvantage of this approach is that it is a centralized. So, if one of the robots in the system fails to manage the formation with other robots, then the whole system fails. Hence, this approach requires a large inter-robot communication bandwidth between the robots.

Leader Follower Approach: In this approach, one or more robots are treated as leaders in the group. The motion of the leader describes the motion of the entire system, and all other robots in the system follow the leader in a fixed spatial pattern (Xiaohai LI, Jixhong Xiao, 2005). Moreover, the robots which are designated as followers are to position themselves with respect to the leader and controller task is to maintain the required relative position of each follower robot with the leader robot. Different combinations of leader - followers and any other complex formations can be obtained by controlling the relative positions of the leader and follower robots.

The main disadvantage in this approach is in active obstacle avoidance on the follower robots. Moreover, the robots don't exhibit stable formation when the robots are asked to combine formation maintenance and to plan its paths. Another important limitation is that the population of the robots is not heterogeneous and different robots operate on the ground of different rules and the perturbation on the follower side is not considered in maintenance of formation.

One of the popular methods based on leader follower approach was proposed by J.P.Desai et al. (1998) called as the l-l controller and l-α controller. Each robot (but the leader) must maintain a given distance (separation – l) and angle (separation - α) with respect to an assigned target. In the l-α controller, the objective is to maintain a desired distance l^d and a desired angle α^d between the leader and the follower and the l-l controller deals with the relative positions of three mobile robots when there exists two leaders.

Behavior Based Control Approach: In this approach, the entire functionality of the system is divided in to a number of different behaviors. These different behaviors are integrated to obtain overall task in the coordinated fashion with cooperative action selection methods (Bailong Liu et al., 2006, Imen Ayari et al., 2007) The advantages of this method is the splitting up of the complete task of the entire system into different behaviours for achieving various sub-tasks. However, the disadvantage is that the system cannot be analyzed or implemented mathematically.

Arkin R (1998) had a detail review on behavior-based robotics that explains the performance of this approach in areas like path planning, navigation, obstacle avoidance and terrain mapping for both single robot and multi-robots. He addressed the problem of maintenance of group of mobile robots in a geometric formation and proposed a multi robot coordination system using behavior-based approach. In the proposed approach, the main objective of the entire system is decomposed into simpler tasks by the use of reactive methods. Different behaviors depending upon the requirement for each robot are designed. Movement of robots in the required formation is achieved by weighing the relative importance of each behavior with the other. These simple behaviors acts in parallel and are implemented concurrently where by each accepts input from the robot sensors and computes actuator commands based on the priority based arbitration mechanism using subsumption architecture to fulfill the overall objective of the robot system. The advantage of this parallel execution of different simple behaviors enables an online response of the entire system with low computational cost.

A dynamic weighted voting technique for the selection of active behaviours is proposed by Samshudin H.M. et al. (2005), where each behaviours concurrently shares control of the robot by generating votes for possible motor commands. However, this method of voting fails for the task sequencing problem, where the robot is not able to perform multiple task such as obstacle avoidance, formation maintenance and path planning together in advance. Imen Ayari and A. Chatti (2007) used both neural and fuzzy logic control methods to realize the task achieving behaviours for reaching goals while avoiding obstacles in the crowded environments. Zhang, Z et al. (2008) introduced feedback control laws to design the active behaviours for multi robot formation control.

The literatures on behavior based control approach have examined the constraints on the application of such approach to multi-robot formation control. Even though behaviors are easier to design than complete task strategy; a central issue is how to select right set of task achieving behaviors, behavioral components, its activation functions, its coordination mechanisms and its associated behavioral architecture for certain applications in a well understood way remains an open question. Further, the major limitation is that it is difficult to analyze the approach by theoretical formalization and hence as a result, it is very hard to assure the convergence of the robots to achieve the overall task.

Graph Theory: This is considered as a one of the key tool in the implementation and analysis of multi robot coordination systems. A graph is a diagrammatic-like representation of the interconnection of the formation robots for data exchange. In these systems, a node in the graph represents a robot, while an edge in the graph defines a dependency between two robots and only interconnected robots coordinate their actions at any particular instance.

Various formation maintenance algorithms based on graph theory have been suggested in the literature. In this method, each robot is assigned with a single or multiple neighboring robots termed as targets in its vicinity. These robots must monitor its target, while moving as a whole group to maintain the given geometric pattern. The robots, their assigned targets and the controller together are called as the control graph. Hence, the performance of the robot formations is measured and analyzed using the topology of this control. It is also used to select a suitable controller for a particular formation shape or to consider if such a controller can exist (Desai, J.P. et al. 2002). Thus the important aspect of this approach is in the selection or construction of the control graph that will optimize all other required properties than stability. Further, the major limitation is the rate of convergence from normal pose to the required formation. This is governed by the laplacian of the control graph on the size of its smallest positive eigen value.

Other Approaches: Other than the approaches mentioned in the previous sections various researchers has developed and used several approaches for multi robot formation control. The detailed literatures on those approaches are presented in this section.

A hierarchical control structure that enables the controllers to switch between themselves was proposed by Fierro et al. (2001, 2002). This switching is done to achieve a stable formation which is based on the sensing of their relative pose to other robots. The proposed multi robot formation framework consists of hierarchical and serial combination of control and estimation modes and parallel composition of agents, using the l-α controller and the l-l controller at the levels. The major limitation of the approach is that the investigations has been carried out concentrating only towards the maintenance of formation in linear separation rather than combined with angular separation. Further, the leader robots are made to follow a pre-planned path planned by the off line planner.

M.Fuji et al. (2003) proposed a hybrid formation control framework consists of the combination of dynamical control and reactive control to control the robots in a closed formation. A simple switching control strategy between the dynamical control approach and the reactive approach is also addressed to represent the stabilization of change in formation. The limited scope of the switching strategy with the occurrence of cut over steering angle is the major limitation of this framework, when there is a huge relative angle between the two robots.

V.T.Ngo et al. (2005) proposed a formation motion control, which is integrated to derive the velocity profiles for the robots in group taking into account the differential geometry of trajectories. A.Fujimori et al. (2008), proposed a control law for the follower robot, where the follower estimates its control input using the leader robot's state, which is calculated by the relative relationships between both the robots. The major drawback is that the leader robot doesn't have any onboard intelligence to navigate the environment; instead it tracks the user defined reference path.

Further, several other approaches such as S* approach (A.M.Rothenstein et al. 2003) based on behavior based control technique, shared potential functions for multi robot task assignment in soccer domain (D. Vail and M.Veloso, 2003), dynamic dead zone method, a combination of potential based avoidance and reactive approaches (Bialong Liu et al. 2001), Non linear Control theory based on linearization and gain scheduling (R. Gabcheloo et al. 2005), input to state stability combined with a dynamic window technique (Harry Chia et al. 2007) are also proposed for maintaining the formation of multiple mobile robots in the environment of interest filled with obstacles.

2.4. SUMMARY

Literature works on truck-trailer system explained the different methodologies used for addressing problems like jack-knife and multiple trailers stability. On the other hand, there are many formation control methods available for multi-robot control. The proposed work looks at developing a rugged formation control for mobile robots and also addressing the problems of the traditional truck-trailer system by treating two individual mobile robots in an in-line formation with each other.

A reactive approach based formation control was developed by Chetty et al., 2010, which was specific to a single robot and cannot be employed for other robots. Many of the models and the controllers (Desai et al., 1998) in the literature used polar coordinates, which has potential singularities. Other type controllers used by Das et al. (2002) and Sanchez et al., (2003) were only locally stable and complicated to design. Hence, there is a need for a new control strategy that can address formation control problems in multi-robot system in a new perspective.

This work takes the advantages of multi-robot system for addressing truck-trailer system problems where both robots can move independently without having any problems of collision with each other (jack-knife phenomenon) or multiple trailers stability. Active obstacle avoidance of the trailer or follower robot is also a major problem of the multi-robot formation control system (Chaimowicz et al., 2004). Hence, all these objectives are framed in to different behaviors/control laws as in the case of a behavior based robotics. These behaviours are added to make the system to imitate a truck-trailer robot.

The advantages of a leader-follower approach are used to make the truck as the leader and the trailer as the follower robot. Moreover, when the number of robot increases, the control methods such as the virtual structure, leader follower and graph theory fails due to their centralization and requirement of higher communication bandwidth. Hence, the advantages of behavior-based robotics are used here as all the objectives can be broken down in to small control laws/behaviors. These control laws are prioritized depending upon the importance of each behavior prior to implementation instead of weighing each state of the behaviors to implement the specific control law. By doing so, the implementation of the behaviors also becomes easy. Hence, a multi layered control architecture is designed with an inline formation controller with collision avoidance and obstacle avoidance features.

CHAPTER 3

HYBRID MULTI-LAYERED CONTROL ARCHITECTURE

3.1 INTRODUCTION

The main work of this thesis is on the development of an inline formation control of two individual robots which is considered to be virtually linked truck-trailer system. As mentioned in the last chapter, the real truck-trailer system suffers from many practical problems. The proposed control approach, wherein two individual robots imitate as a truck-trailer system can help in overcoming the problems of a conventional truck, hitched with passive trailer(s). The virtually linked truck-trailer formation control finds its use in an environment like a tightly packed shop-floor assembly line. Apart from maintaining the truck-trailer formation, the major challenges that need to be solved in this environment are: avoiding collision with dynamic obstacles like: other robots, workers, etc in narrow spaces, formation breakup during obstacle avoidance, and reformation after the breakup, etc.

The next section details on the selection of behaviors/control modules that forms up the complete architecture. This is followed by a detailed explanation of each of the behavior and its formulation starting with the formation controller for a pair of robots. Later, four more behaviors (control modules) are also explained that are added to imitate the system as a truck-trailer robot. All these behaviors, along with the formation controller are integrated into a multi-layered behavior based control, for controlling the virtual linked truck-trailer system. This chapter gives an insight to each of the behaviors developed and the architecture of the complete control.

3.2 SELECTION OF BEHAVIORS

Depending upon the objectives of the work, the behaviors to be designed must reflect these properties of the system. So a set of five behaviors are incorporated as control algorithm which will satisfy our requirement. The next five sections detail on the theoretical formulation of each of the five behavioral layers namely – formation control, dynamic obstacle avoidance, collision avoidance between the robots, formation breakup and switching formations. Formation control – incorporates truck-trailer behavior to both the individual robots; dynamic obstacle avoidance is a problem-of-study in multi robot system that helps in successful navigation of the trailer by avoiding obstacles; collision avoidance prevents the jack-knife phenomenon – one of the main reason for the failure of conventional truck-trailer robots; formation breakup / switching formations is useful in addressing the parking space constraints problem and can also make the robots adapt to different formation shape as and when required. Using the above behaviors, it is possible to add / remove multiple trailers to the existing control system.

In this work, the formation controller is treated as a tracking controller, which reduces the tracking error of the trailer robot to be asymptotically zero. Later, other behaviors are explained individually. Finally, the control architecture and setting up of each of the layers are presented, along with the contributions and advantages of the present work in the last section.

3.3 RESEARCH ASSUMPTIONS

For the simplicity of verification, the following research assumptions are considered:

3 Both the robots have the same physical configuration.

4 The truck robot moves with constant linear velocity v and angular velocity ω and the path/trajectory is pre-planned by an offline planner.

5 The control is not focused on the truck navigation or path planning rather on the formation control of the trailer with the truck.

6 Dynamics of the vehicle is not included.

7 Tyre parameters like slip and friction are neglected.

8 Both the robots exchange their sensory information with each other through explicit inter-robot communication.

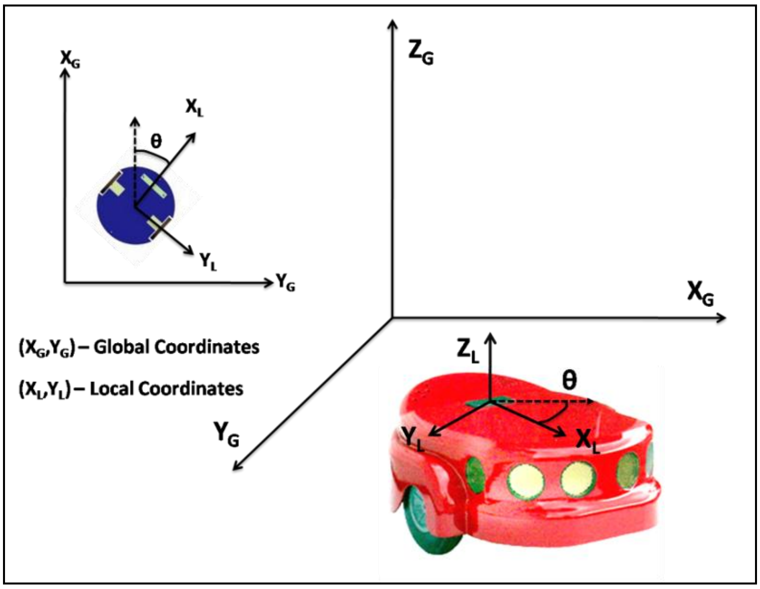


Fig 3.1 Coordinate System of a Mobile Robot

3.4 COORDINATE SYSTEM

Motion of rigid bodies is generally represented by a coordinate system. In the case of mobile robotics, two orthogonal coordinate frames are considered - the global axis coordinate frame and the robot's local axis coordinate frame. Any convenient point in the model environment can be taken as a reference point for the global coordinate system and it is a fixed point. Any moving object in the arena takes it reference from this non-moving point. The robot local axes coordinate frame is considered on the centre of mass of the robot; hence, it moves along with the robot. Figure 3.1 illustrates the local coordinate frame on the robot and the global coordinate frame in the corner of the room.

3.5 DESIGN OF VIRTUAL LINK AND VIRTUAL HINGE

In order to understand the formation parameters and virtual link, it is necessary to study the type of link to be designed and its configuration. The virtual link is the separation distance between the two robots and a point on this virtual link taken as the virtual hinge point. The complete control architecture is developed based on this point. Moreover, having a control point outside the leader robot's coordinate frame gives a better control over the entire system.

Generally, in conventional truck-trailer system, the trailer can be hooked to the truck mainly in two ways - directly hooked or off-hooked. Jaehyoung et al. in 2001 showed that under the steady state conditions, the trailer of an off-hooked system can successfully trace the trajectory of the truck compared to an on-hooked trailer. So, our virtual link is considered to be an off-hooked link connected by a virtual pin.

The virtual hinge is at a distance L_1 & L_2 from the truck and trailer respectively. So, the total link length (L) is equal the sum of L_1 and L_2 (i.e. $L=L_1+L_2$). For the purpose of simplifying the control problem, length of one of the link (L_1) is taken as constant value (L_1 = 150mm). This value is chosen according to the physical dimensions of the robot. Here, we used Pioneer P3DX and the Amigobots of Adept Mobile Robots Inc. The minimum turn radius of such a differential-drive robot is zero. In our experiments, L_1 value of 650mm is used between the two Pioneer P3DX robots to avoid collision when they rotate on their own axis; whereas a L_1 value of 400mm is used in the case of Amigobots.

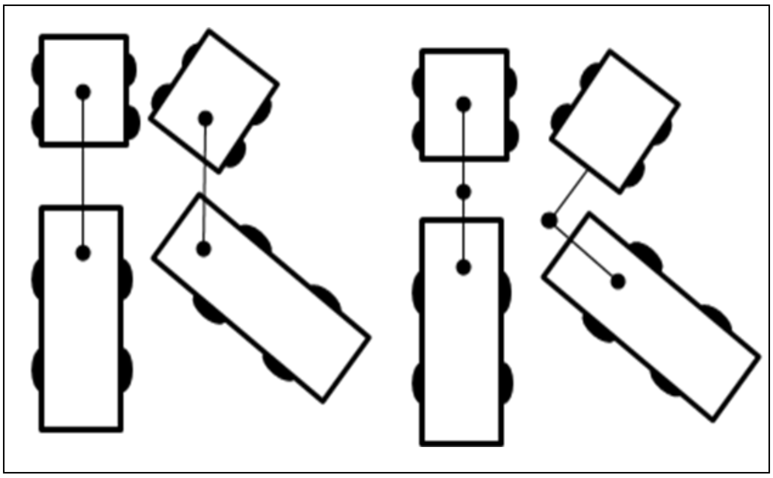


Fig 3.2 a) Directly-hooked and b) off-hooked trailer

The trailer computes its desired pose with respect to the truck and the virtual link. The angular orientation and the X coordinate of the hinge points are same as that of the truck. But, the separation distance is offset in the Y axis by a length equal to the upper half of the virtual link L_1 which is shown in fig 3.3. The total hinge length L value is the formation keeping distance between the robots but this value varies when the robots avoid collision between the two by means of jack-knife behaviour of the robot.

In case of Kuppan et al. (2008), the control point was taken to be the centre of axis of both the leader and follower robots whereas in case of the l-α controller by Kumar et al. (1998), there were two points taken on the truck robot and a single point on the trailer robot. Both these controllers had their own drawbacks. In case of the former, the controller failed for negative angular velocities whereas in the case of the latter, the controller went absurd when the all the robots are aligned in-line to each other. Hence, in this controller having a control point in space gives a better control on the entire system. Moreover, the location of this point is also important and has been assumed to be in line with the truck axis offset by some constant distance from the centre of axis of the truck. The exact hinge point has been found through experiments in the next chapter.

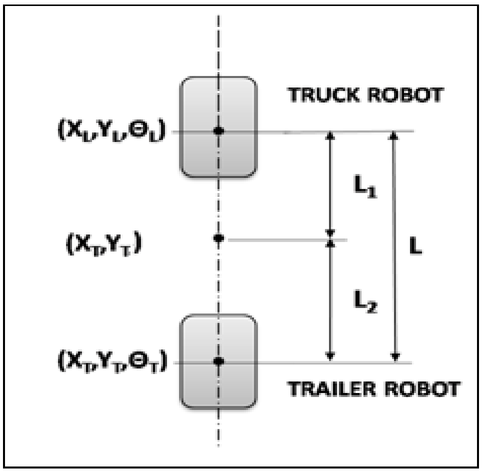


Fig 3.3 Virtual Hinge Coordinates

The equations for the virtual hinge in terms of global frame is given by,

$$X_H = X_L - (L_1 * \cos\theta_L) \tag{3.1}$$

$$Y_H = Y_L - (L_1 * \sin\theta_L) \tag{3.2}$$

where (X_H, Y_H) are the global coordinates at the hinge.

3.6 FORMATION CONTROLLER

The formation controller is derived here that requires both the robots to be in-line with each other as a truck-trailer. Hence, this is the most important behavior that will create a virtually linked truck-trailer system with respect to the off-hooked virtual control point. Since this problem can be treated as an inline formation of two robots, a tracking controller is derived for the trailer, such that it is easy for the trailer to trace the trajectory of the truck. However, the truck is given a pre-planned path as planned by an offline planner.

Taking two parameters, l and φ as the formation parameters in terms of linear and angular separation between the truck and trailer, a control law is derived for keeping up the formation. Once in motion, the relative l and φ values should converge to the formation parameters making the errors asymptotically to zero. So, a tracking controller is derived to compute the motion of the trailer in according to the truck trajectory. Since, the two robots act as truck and trailer, an off-hooked virtual hinge point is taken between the two in space. Hence, the tracking controller computes the linear velocity and angular velocity of the trailer robot to be in the required formation with the truck with respect to the off-hooked point. Having an off-hooked point in between the two robots, which acts as a control point for the tracking controller is the main feature of this work compared to other literature works. This is because the virtual point provides a better and robust control over the entire formation; in particular, when multiple trailers added into the existing formation.

3.6.1 Kinematic Model for the formation

The controller is derived by taking the kinematics of two robot system as shown in figure 3.4. A differential drive wheeled robot R_1 with Cartesian coordinates (X_L, Y_L, θ_L) with respect to the inertial axes X_IY_I is considered as the truck robot and it is in line with another robot R_2 of the same configuration - the follower trailer robot, with coordinates (X_F, Y_F, θ_F). The parameters '*d*' is the width of both the robots, '*D*' is the diameter of the robot wheel and '*h*' is the distance from the robot axis of rotation to the robot's front end (half of the robot's length). Similarly, L is the total length of the link (L=L_1+L_2), where $L = l_d$ is the formation distance and φ_d is the formation angle between the two robots respectively. The parameters (v_L, ω_L) and (v_F, ω_F) are the linear and angular velocities of the truck and trailer robots respectively. During turning, collision between the two robots occurs when the relative angle between the trailer and tractor is greater than 90° which is the hinge angle (β). So, it can also be defined as $\beta=\theta_L-\theta_F$.

3.6.2 Controller Block Diagram

As, mentioned earlier, formation controller is the main behaviour which governs the system. However, this is prioritised low under certain conditions of dynamic obstacle avoidances and jack-knife phenomenon, when the formation need to be broken or made flexible. The tracking controller is implemented on the trailer alone and the truck sends its pose and velocity data through explicit communication to the trailer.

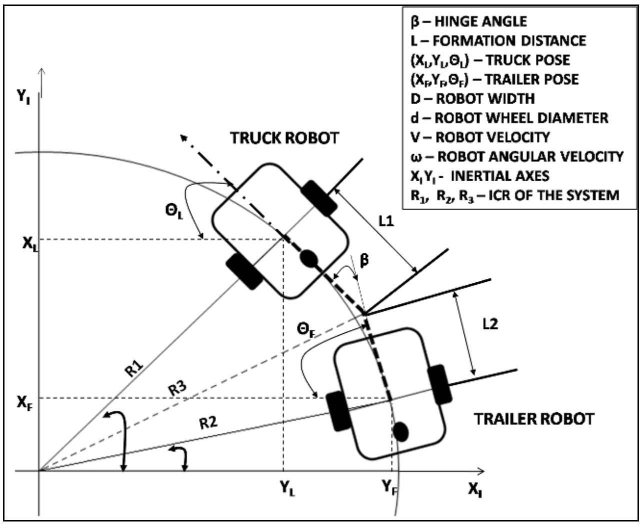


Fig 3.4 Off-Hooked Virtual Truck-Trailer Robot System

The tracking controller can be explained with the block diagram in figure 3.5 and its task is to reduce the tracking error of the trailer to zero, i.e. the trailer should move according to the linear and angular velocities v_F and ω_F computed from the truck robot's data. At every instant, the required trailer pose is computed using the current truck pose and the virtual link parameters. Tracking controller compares the error coordinates, truck robot pose and the virtual link parameters and using this data it computes the velocity of the trailer robot. This rugged control helps in providing tight-framework architecture and thus reduces the tracking error to a significant amount.

3.6.3 Control Law

Basic kinematic equations for the derivation of the tracking controller are taken from Kuppan et al. (2010) for the truck and trailer robot. Since our aim is to control the velocity of the trailer robot, we will derive equations in terms of velocity and not by position.

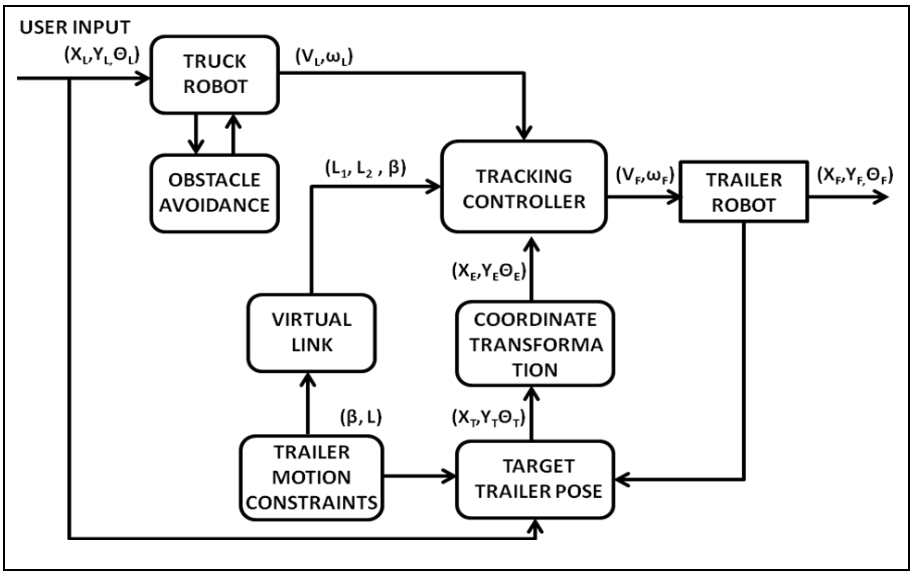


Fig.3.5 Block Diagram of the tracking controller

For a nonholonomic differential drive robot, the kinematic equations are given by,

$$\dot{X}_L = v_L \cos\theta_L \quad (3.3)$$

$$\dot{Y}_L = v_L \sin\theta_L \quad (3.4)$$

$$\dot{\theta}_L = \omega_L \quad (3.5)$$

These equations are taken to be the velocity equations of the truck robot. Same way, the velocity equations for the trailer robot is derived after mapping from the global coordinate frame using the orthogonal rotational transformation *R(θ).*

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

After mapping, the velocity equations for the trailer robot with respect to the global system are,

$$\dot{X}_F = v_F \cos\theta_F - \omega_F * h * \sin\theta_F \quad (3.7)$$

$$\dot{Y}_F = v_F \, s\,in\theta_F + \omega_F * h * \cos\theta_F \quad (3.8)$$

$$\dot{\theta}_F = \omega_F \quad (3.9)$$

The velocity equations of the hinge point can be found from equations 3.1 and 3.2 as

$$\dot{X}_H = v_L \cos\theta_L + \omega_L * l_1 * \sin\theta_L \quad (3.10)$$

$$\dot{Y}_H = v_L \sin\theta_L - \omega_L * l_1 * cos\theta_L \quad (3.11)$$

$$\dot{\theta}_H = \omega_L = \dot{\theta}_L \quad (3.12)$$

Now, the desired velocity equations for the trailer robot with respect to the hinge points are found. This will keep the velocity of the trailer to be in the required formation with the truck robot. From figure 3.4 and from the above equations, the desired velocity equations for the trailer robot can be given as follows,

$$\dot{X}_D = v_L \cos\theta_L + l_1 * \omega_L \sin\theta_L + l_2 * \sin(\phi_d + \theta_L)\dot{\theta}_L \quad (3.13)$$

$$\dot{Y}_D = v_L \sin\theta_L - l_1 * \omega_L \cos\theta_L - l_2 * \cos(\phi_d + \theta_L)\dot{\theta}_L \tag{3.14}$$

$$\dot{\theta}_D = \dot{\theta}_L = \omega_L \tag{3.15}$$

There are two set of equations available – one is the original trailer velocity equations and the other is the desired trailer velocity equations. In order to design the tracking controller, the velocity error equations for the desired trailer position has to be found; so that a tracking control law can be designed that will reduce these velocity errors to zero. This can be obtained by subtracting equations 3.13 to 3.15 from equations 3.7 to 3.9 as follows:

$$\begin{bmatrix} \dot{X}E \\ \dot{Y}E \\ \dot{\theta}E \end{bmatrix} = \dot{R}(\theta F) * \begin{bmatrix} \dot{X}D - \dot{X}F \\ \dot{Y}D - \dot{Y}F \\ \dot{X}D - \dot{\theta}F \end{bmatrix} \tag{3.16}$$

where,

$$\dot{R}(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.17}$$

Substituting all the values in Eqn 3.16, we get the velocity error equations of the desired trailer position as,

$$\dot{X}_E = v_L \cos\theta_E + l_1\omega_L \sin\theta_E - v_F + l_2\omega_L \sin(\phi_d + \theta_E) \tag{3.18}$$

$$\dot{Y}_E = v_L \sin\theta_E - l_1\omega_L \cos\theta_E - \omega_F h - l_2\omega_L \cos(\phi_d + \theta_E) \tag{3.19}$$

$$\dot{\theta}_{\mathrm{E}} = \omega_{\mathrm{L}} - \omega_{\mathrm{F}} \tag{3.20}$$

These are non-linear in nature and are to be linearized before applying control law. Later, a suitable tracking controller has to be designed such that the velocity error coordinates becomes asymptotically zero by making $l \rightarrow l_d$ and $\varphi \rightarrow \varphi_{\mathrm{d}}$. Hence, feedback linearization is applied for this system. This method is simpler, as the nonlinear system is transformed into an equivalent linear system by adding few variables and control inputs. This method is preferred over other linearization techniques, because in this linearization technique, both linearization as well as control law can be achieved in a single step. The effectiveness of this method depends upon the selection of suitable variables, which relates to the system output to the control input so that the system is made linear. Hence, taking the linearization variables as,

$$\dot{X}_E = -K_1 * X_E \tag{3.21}$$

$$\dot{Y}_E = -K_2 * Y_E \tag{3.22}$$

$$\dot{\theta}_E = -K_3 * \theta_E \tag{3.23}$$

Where, K_1 is linear control gain, K_2 and K_3 are angular control gains. These new variables are substituted in equations 3.18-3.20 to linearize the equations. Now rearranging the terms and writing in terms of the control input v_F and ω_F,

$$v_F = (v_L \cos\theta_E) + (L_1\omega_L * \sin\theta_E) + (L_2\omega_L * \sin(\phi_d + \theta_E)) + (K_1 X_E) \tag{3.24}$$

$$\omega_F = \frac{(v_L \sin\theta_E) + (L_1 K_3 \theta_E \cos(\phi_d + \theta_E)) + (L_2 K_3 \theta_E \cos\theta_E) - (K_2 Y_E)}{1 - [((1+\theta_E)L_2 \cos(\phi_d + \theta_E)) + ((1+\theta_E)L_1 \cos\theta_E)]} \tag{3.25}$$

where,

$$X_E = [(X_H - (L*\cos(\phi_d + \theta_L) - X_F))\cos\theta_F] + [(Y_H - (L*\sin(\phi_d + \theta_L) - Y_F))\sin\theta_F] \tag{3.26}$$

$$Y_E = -[(X_H - (L*\cos(\phi_d + \theta_L) - X_F))\sin\theta_F] + [(Y_H - (L*\sin(\phi_d + \theta_L) - Y_F))\cos\theta_F] \tag{3.27}$$

$$\theta_E = \theta_L - \theta_F \tag{3.28}$$

where, (X_E, Y_E, θ_E) = error coordinates of the desired trailer pose; (X_H, Y_H) = hinge pose and l_d and φ_d are the linear and angular formation parameters between the two robots respectively.

Equations 3.24 and 3.25 form the control law that governs the trailer robot to be in the required formation with the truck. The trailer leader and follower velocities are calculated by the tracking controller using these equations. These equations can also be written by taking only the linear separation L and replacing l_1 and l_2 as,

$$v_F = (v_H * \cos\theta_E) + (l_d * \omega_H * \sin(\phi_d + \theta_E)) + (K_1 * X_E) \tag{3.29}$$

$$\omega_F = \frac{(v_H * \sin\theta_E) - (l_d * K_3 * \theta_E * \cos(\phi_d + \theta_E) - (K_2 * Y_E)}{1 - ((1 + \theta_E) * l_d * \cos(\phi_d + \theta_E))} \tag{3.30}$$

3.7 OBSTACLE AVOIDANCE BY THE TRAILER

Obstacle avoidance is an important problem for the formation control of a group of mobile robots. The truck robot is assumed to navigate from obstacles on its own and this behaviour is implemented in the truck robot's controller. But, the in-built obstacle avoidance behaviour like that of the truck is not invoked in the trailer due to the formation controller imposed on it. However, new typical obstacle avoidance behaviour is implemented to the trailer which is discussed in detail in this section.

Since, the trailer is inline formation with the truck; the trailer can face only dynamic obstacles. Dynamic obstacle avoidance of the trailer robot is the major and the most prioritized behaviors of all. Obstacle avoidance of the follower robots in a formation has always been as one of the major challenges in formation planning.

There are many obstacle avoidance algorithms proposed by various researches. Summarizing few of them, Artificial Potential field is a technique inspired by physics, in which robots and obstacles are positive charge and hence repel each other, whereas goals are negative charge that attracts the robot. The complexity of the problem is determined by how the total forces are calculated by evaluating the force vector. Kuppan et al. (2005) has used dynamic role switching concept between the robots, where both them exchange each of their behaviour of navigation control and formation control respectively, so that the leader becomes follower and vice-versa.

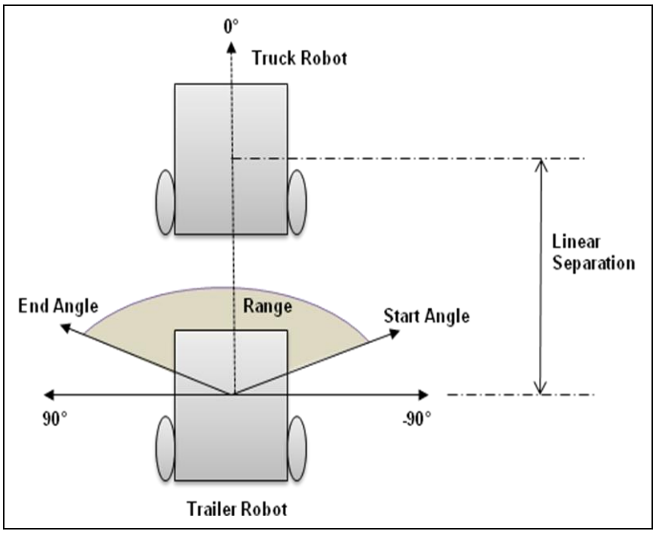


Fig 3.6 Sensing range of the trailer to avoid obstacles

The other most common method of obstacle avoidance is the decentralized reactive collision avoidance based on collision cones. It consists of a deconfliction maintenance controller, which checks for the robots to be out of conflict and a deconfliction manoeuvre that is executed to come out of the conflict. In this case, a sense-think-act like behaviour similar to this method is developed for the trailer. Though the algorithm is same for all types of formation shapes, the sensing range of the trailer robot is different for truck-trailer formation and other formation shapes.

This is because in case of truck-trailer formation, the truck robot is closer to the trailer robot and the sensing range is largely affected by the presence of the truck robot. Hence the sonar sensing range is set up for a new vicinity area around the trailer robot as shown in fig 3.6. The sensing range is usually taken to be l_2 so that the trailer can avoid any obstacle in the range distance equal to the minimum distance between the robots = l_1.

Truck Algorithm

```
if (trailer sent stop request?)
then
stop
release formation with trailer
wait for trailer
        if (trailer sends reformation
        wait for trailer to align
        goto  step 9
else navigate to the goal
        If (obstacle detected)
        avoid obstacles
        then navigate to goal
```

Trailer Algorithm

```
track the truck
If (obstacle detected && trailer track path
is blocked)
send stop and formation release
command to truck
then
avoid obstacle using sonars
then
calculate error coordinates
        if (time> 60000ms)
        declare lost
        else execute recovery move
then
repeat move till aligned with truck
formation realigned
repeat tracking
```

Fig 3.7 Truck-trailer obstacle avoidance algorithm

When the sensor of the trailer senses the presence of an obstacle in its vicinity and not able to move further without avoiding it, dynamic obstacle avoidance behaviour takes priority over the other behaviours. The trailer breaks the formation and executes the obstacle avoidance behaviour. Simultaneously, the trailer sends the formation breakup request packet to the truck. Immediately, truck releases the trailer from the formation and stops navigating. After avoiding the obstacle, the trailer computes the required pose to reform the formation with the truck. Finally, the trailer establishes the link with the truck robot and the formation is re-established. So the switching between these two behaviours does not allow any unnecessary movement of the robots and the robots will be able to reach the goal at a minimal time.

3.8 COLLISION AVOIDANCE BETWEEN THE ROBOTS (JACK-KNIFE PHENOMENON)

Control algorithms used for collision avoidance between robots is similar to obstacle avoidance algorithms found in literature. The differences being the obstacles are dynamic in the latter case and static in the former case. This behavior is implemented to overcome the existing jack-knife problem of the truck-trailer system. Collision between the two robots happens, when the navigation area is congested and the distance between the two robots is very less. Hence, it depends mainly on the dimension of the robots – length & width respectively. Collision between the robots can also happen when there is an initial delay for the leader to start / respond to the follower or a delay in communication between the robots or an unexpected sharp turn / jerk. This control module solves this problem by varying the length of the virtual link depending upon hinge angle, i.e. (v_F, ω_F) is adjusted depending upon the value of hinge angle ($\beta = \theta_L\text{-}\theta_F$). Since, the hinge angle is obtained directly from the angular orientation of truck and trailer, the angle of turn of the robot can be estimated from it. However, this is for differential drive robots. In case of a four-wheeled robot, the control behavior depends on the steering angle of the front wheels.

As mentioned earlier, L_1 is kept at a constant value (which is the minimum value required for avoiding collision), whereas L_2 is varied from the zero. This minimum collision distance (L_1) varies from robot to robot depending upon the physical dimensions. Also, this behavior has higher priority compared with the formation behavior. This behavior control makes use of a simple proportional controller, in which β is varied inversely with the linear velocity of the follower. Angular velocity

ω_F is obtained from $v = r*\omega$, where is r is obtained from ICR of both robots. The Proportional gain (K_p) value is taken as 0.75.

This behaviour adjusts the distance between the two robots as required to avoid collision between them. Another advantage is that the trailer can guide the truck using backward movement when the truck is caught in a narrow space and not able to turn back and return. Therefore, this formation control can be used for navigating narrow spaces. This is the mid-layer, which uses robot action states from both the high and low level.

Usually, the normal condition for jack-knife is: $\beta \geq 90°$. Since our formation parameters are made flexible, β can be varied without causing jack-knife by adjusting length of the virtual link $L_{2.}$ When moving in a straight line, i.e. when the turn angle $\theta_L = 0°$, the distance L and the angle β between the two robots is maintained as per the formation. However, when the truck-trailer system takes a turn or any other manoeuvre, especially during obstacle avoidance, the trailer velocity 'v_F' is reduced.

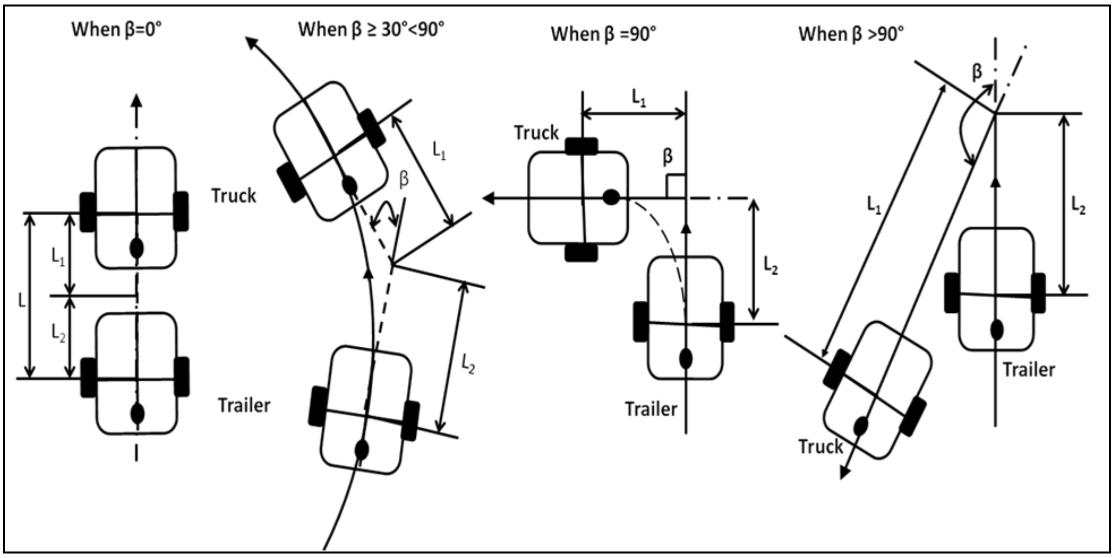


Fig 3.8 Truck-trailer formation during maneuvering

This results in new formation distance and angle depending on the hinge angle of the truck. In this case, β can take any value from 0° to 180° and L can vary from 400mm to 1200mm, both depending upon the physical dimensions of the robot. In this case, L_1 is kept constant which is the desired formation distance and L_2 is varied. Figure 3.8 illustrates this. The minimum collision distance L_1 between two amigobots is 400mm and so this behaviour is mostly used under special conditions only. This is because the optimum virtual hinge length has been found in experiments as 600mm and experiments are run at 600mm only.

3.9 FORMATION BREAK-UP

As mentioned earlier, another problem that is faced by truck-trailer robots is parking space constraints. During parking, a conventional truck-trailer robot occupies more space than the space required for the two individual robots, due to the connecting link between the robots. However, this issue will not be a problem, when both the robots are connected by virtual link. Unlike the conventional truck-trailer, both the robots can break the formation and can behave as individual robots to park separately in the space available. Formation breakup request is initiated by the trailer only depending upon the current sensory information under two conditions:

1. Obstacle Avoidance (Temporary) – when there is a dynamic obstacle in the path of the trailer. The trailer's first priority is to avoid the obstacle first and hence, it issues a temporary break-up request of 60000 ms to the truck robot. This is the maximum time that will be required to avoid an obstacle in a range of 2 meters which forms the basic elastic band for any obstacle avoidance algorithm. If the obstacle is not avoided in this time due to some complications, the trailer is declared lost by the truck and both the

robots get into a permanent break-up initiated by the trailer robot which is the second condition.

2. Parking Requirement (Permanent) - In cases of parking space available to two robots in two different places, the two robots can break-up the formation which is a permanent request by the trailer. A special register is reserved in the memory which is shared between the truck and trailer. The truck checks up this register for every 100 ms (which is the regular update frequency for communication between the two robots) for the breakup request from the trailer. This helps both the robots to behave as individual robots that can be used for other purposes.

3.10 SWITCHING OF FORMATION SHAPE

Multi robot system comes up with different formation shapes for various applications. One of the other most important applications of multi robot system is when robots move in parallel. Parallel formation is very useful in agricultural purposes, where the truck does the cutting of weeds and the trailer collects the weeds. Instead of designing a new controller that satisfies this criterion, the feasibility of the existing controller to be used for this parallel formation is studied. In order to adapt to a new formation shape, the two additional temporal behaviors – formation breakup and switching to new shape that should be incorporated to the present architecture.

The existing formation controller is modified by changing the desired linear and angular separations so that both the robots make a parallel formation. The parameters l_d and φ_d are set to 300mm and 270°, which keeps both the robot in parallel formation. Also, any change in the formation parameters to adapt to a different formation shape did not alter the performance of the system. Another special register is updated during every cycle of the controller that will check for the user input or request given to the trailer (pre-programmed) for formation shape change.

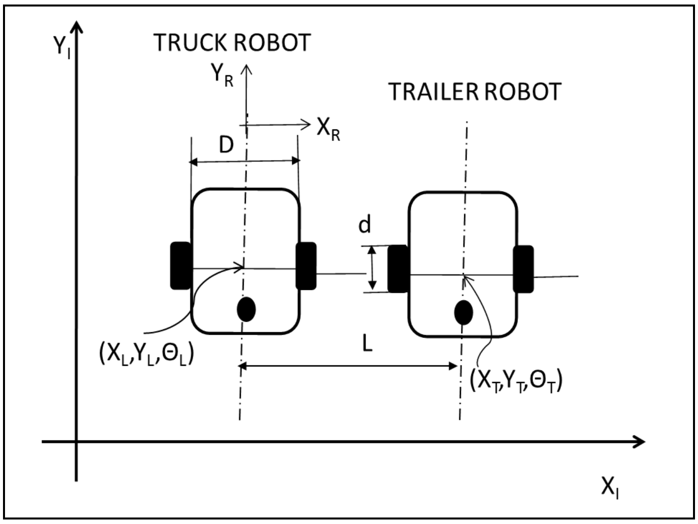


Fig 3.9 Truck-trailer in parallel formation

This is the least prioritized behavior of all to avoid deadlock conditions i.e. switching of formation shapes will work only when the higher level of control architecture especially, obstacle avoidance and collision avoidance behaviors are not active. This is because the higher level behaviors alter the formation shapes and they adapt to different requirements unlike this behavior which involves explicit formation change. The detailed explanation of the proposed control architecture is further explained with flowcharts and with implementation procedures in the next chapter.

3.11 MULTIPLE TRAILERS

Many algorithms as shown in the literature review have failed, when additional robots were brought in to the formation. Even in the case of a truck-trailer system, when additional trailers were added to the existing formation, the control algorithm collapses and the robots the formation breaks. Here, the proposed formation controller is tested by adding more number of trailers to the Truck-Trailer configuration and as well as in the parallel movement multiple robots. This is not a separate behavior, but the versatility of the formation control algorithm makes it possible to implement multiple trailers to the present system.

There are two methods to implement this algorithm for multiple trailers.

1. **One virtual link (with multiple hinge points)**: First, we can treat the first trailer as the leader for the second trailer, as shown in the Fig.3.10 (a). Each trailer has dual control algorithms running on it. It acts as a trailer to the robot preceding it as well as truck to the newly added trailer. It can be treated as a single lengthy link with multiple hinge points for attachment of each trailer. This gives a better rugged control on the whole system as there are multiple control points in each hinge and each trailer robot keeps track of the truck-cum-trailer robot in front of it. This forms a new type of leader-follower formation controller which overcomes the reliability problem on a single leader. As each robot behaves both leader and follower in the system like having dual architecture in both the same robots, even when one of the leaders fails, the next first robot preceding the main leader robot will still be able to lead the other robots in a tight- rugged formation. Hence, this architecture can be successfully employed in places where reliability on a single robot cannot help the formation.

2. **Multiple virtual links**: In the second method, there is only one truck and multiple trailers attached to it as shown in the Fig.3.10 (b). There are as many as number of virtual links as the number of trailers present with different

lengths depending upon the position of the individual trailer. This can be treated as multiple virtual links with different control points for different trailers. Here, each trailer maintains desired parameters with the truck and not with each other. In fact, each trailer is not aware of the other trailers present in the system. This is basic leader-follower formation control architecture with the introduction of virtual link between each of the truck and the trailer robots.

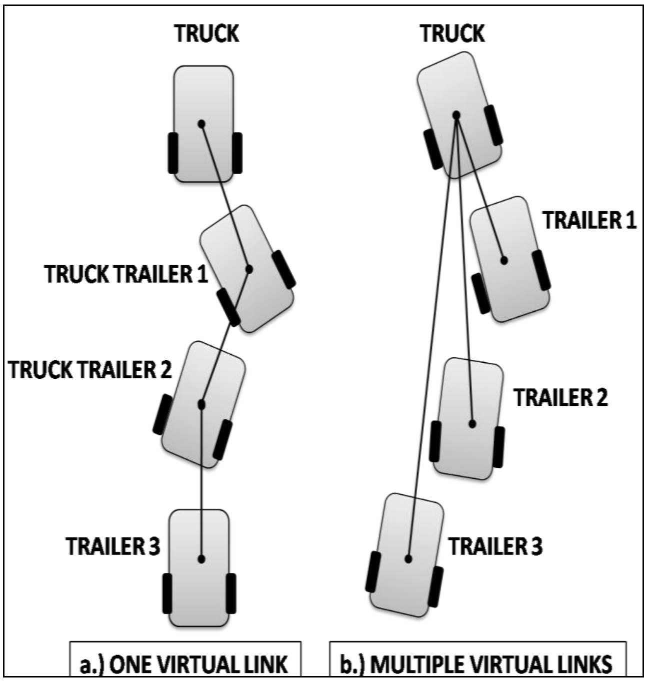


Fig 3.10 Illustration of the two methods for multiple trailers

Since, the focus on this work is formation control of multiple mobile robots; the formation control algorithm with multiple trailers is studied. The other four behaviors are not tested owing to dead-lock situations of the robot. It is a condition that happens when there are many trailers attached to the truck and all the behaviors are implemented to the system. In such cases, when there is a condition of obstacle avoidance, more than one trailer issues stop request to the truck and the other robots issue a go request. This causes confusion and dead-lock situation among the robots.

Since, the main scope of this work is on coordination of mobile robots, dead-lock situation is not tested for multiple mobile robots.

3.12 MULTI-LAYERED ARCHITECTURE

Keeping these requirements in mind, a new type of control architecture has been designed. This control architecture should be distributed in nature, so that there is a tight coupling between the perception and action of the robot like the reactive control approach. Moreover, the truck robot is treated as leader and all the other robots are treated as trailer or followers. So, the advantages of a leader-follower formation control are also used here. The different functions of the system are set at different priorities, so that the system properly imitates a truck-trailer, thus making this system as multi-layered architecture. Hence, a hybrid (leader-follower & behavior-based) multi layered (priorities set) architecture is designed to replace a truck with passive trailer system by two individual mobile robots that act a truck-trailer. The next section details on the selection of behaviors and type of layers and behaviors of the control architecture is presented.

The overall description of the proposed truck-trailer control architecture is discussed in this section. The multi-layered control architecture is shown in figure 3.11. The basic robot motion states /control laws used here are follow truck (set velocity, set angular velocity), stop, turn left, turn right, go back and ignore. These states are combined with the mathematical formulation of the control equations of each required function to give the required layers that can robots to behave as a truck-trailer.

The three main functional behaviors needed here are: formation planning, collision avoidance (jack-knife phenomenon) and obstacle avoidance. The first two functional behaviors makes use of follow truck and stop states, whereas obstacle avoidance makes use of other action states like stop, go, ignore, turn left and turn right. Since, the first two functional behaviors share the same motion states, the behavioral module sets priority for each one and classifies them as a layered architecture depending upon the requirement of truck-trailer formation and the environmental sensory information.

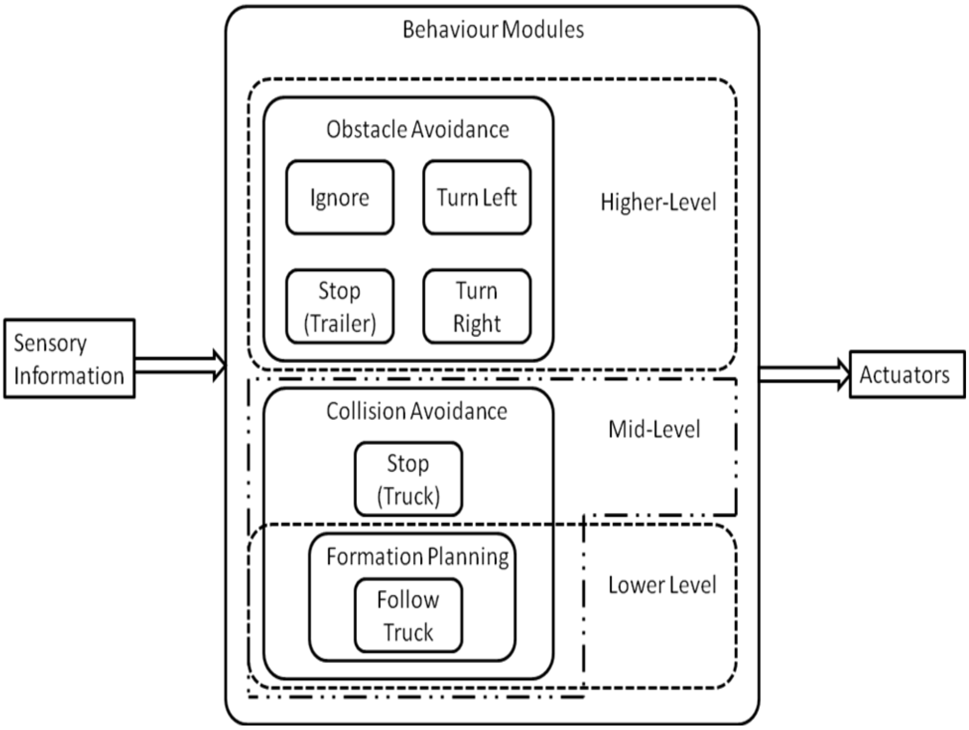


Fig 3.11 Multi-Layered Behaviour Module

Hence, the formation control forms the lower level in the behavioral architecture owing to its least priority and the higher level is the obstacle avoidance. It is similar to implementing two behaviors – pure go to goal and pure obstacle avoidance. The collision avoidance is given a blended behavior of both robot motion states and is sandwiched in between the other two as a mid-level layer. There are two more temporary functional behaviors used for complex movement – formation breakup and switching formation shapes. While, the formation breakup is used to completely dissolve the current formation of the robots and make them behave as individual ones, the latter modifies the current formation shape in to other formation shapes by modifying the tracking controller, depending upon the environmental conditions or user input. As mentioned earlier, this is a hybrid approach, which exploits the advantage of both behaviour based approach and the leader - follower formation approach.

CHAPTER 4

SIMULATION STUDIES AND EXPERIMENTAL INVESTIGATIONS

4.1 INTRODUCTION

This chapter presents the implementation part of the proposed controller in a simulation environment. It also gives a brief description of the mobile robot research platform used in this research work. While, the simulations are carried out in order to test the designed controller, real time experimental studies are also conducted to validate the controller effectiveness.

The first section explains on the hardware and software details of the research platforms used. This is followed by the simulations that are carried out using the Application Programming Interface (API) of the research platform. The simulation results are compared with the experimental results obtained by using the same set of parameters. A detailed study on the performance of the controller individually and as a whole architecture is done here.

4.2 RESEARCH PLATFORM

Before evaluating the performance of the proposed controller, it is important to understand the details of the research platform that is used. Experiments were carried out with Adept MobileRobots' Amigobot, which is a small cost-effective, differential drive mobile robot with two wheels and a passive castor for balance. The Pioneer P3DX mobile robot is also used in some experiments, where more than one trailer is needed for testing the effectiveness of the developed control system. Both the robots are developed by Adept MobileRobots Inc. and use the same application programming interface (API) to program them. The only difference being that Amigobot is comparatively smaller and less versatile compared with Pioneer P3DX, which has more sensors with better resolution, higher payload capacity, more operating range and rugged.

Fig 4.1 Pioneer P3DX & Amigobot

4.2.1 Hardware Details

Both the Pioneer P3DX and Amigobot robot research platforms used in this research work are the most widely used research platforms, due to their versatility. They are fully programmable using a dedicated application interface called as ARIA. Position and orientation information is estimated using dead reckoning method with the help of high resolution optical quadrature shaft encoders. These robots are also available with different set of expansion power, I/O ports, sensors and many other accessories. The data from the sensors and accessories can be accessible by the robot controller through an application interface known as ARCOS, which is the robot operating system. Apart from these, various other sensors and accessories can be integrated to customize these robots for specific applications.

Sensors: The Amigobot is equipped only with SONAR range finders. However, the Pioneer P3DX is supplied with sonar range finders and bumpers. These sensors are used for obstacle avoidance.

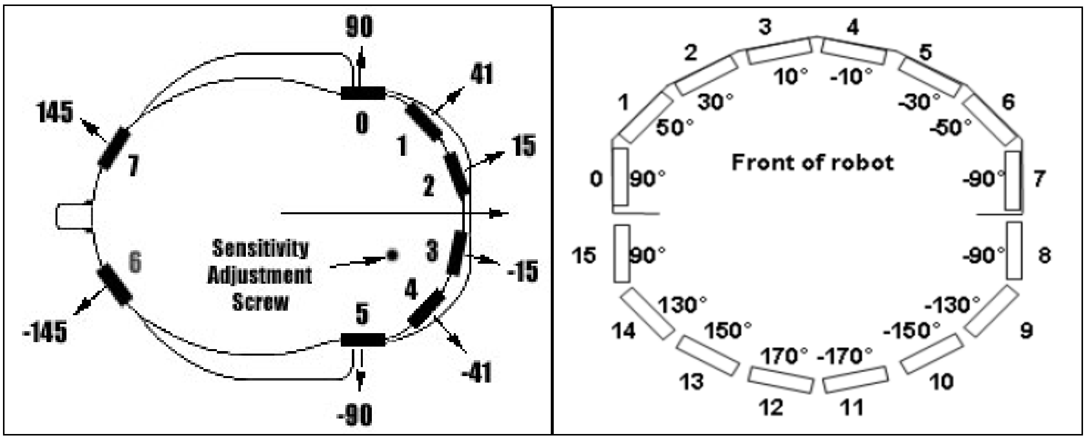


Fig 4.2 Arrangement of Sonar Range Finders in Amigobots and Pioneer P3DX

The AmigoBot robot has an array of eight Sonars, which provides 360 degrees of range sensing. There are four sonars in the front, one on each side and two at the rear. The sonars are numbered clockwise around the robot, beginning with the left side as shown in Fig.4.2. The sonar array's sensors are multiplexed: Only one sensor is active at a time. The sonar acquisition rate is adjustable and normally set to 25 Hertz (40 ms/transducer). Depending on the required detection range, the sensitivity of the individual sensor can be adjusted from 10cm up to 5 meters. The sonar's firing pattern is controlled via software.

The Pioneer P3DX robot consists of two sonar arrays and each array has eight transducers. These sensors can provide range information in the range of 120mm to 5000mm for detecting objects. Six sonars are positioned at 20 degree intervals on each side of the robot, while two more are placed on each side as shown in the Fig.4.2, thereby providing 360 degrees of range sensing. Similar to Amigobot robot, the sonar array's sensors are multiplexed and polled sequentially at the update rate of 25Hz (40ms per transducers per array). Hence for an eight sensor array, the data from any one sonar transducer would be read at every 320ms.

Motors (Encoders): The Amigobots and Pioneer mobile robots are equipped with high speed, high torque reversible-DC motors as prime movers. Each motor is fitted with a high-resolution optical quadrature shaft encoder. The encoder generates 123 ticks per mm of wheel rotation, thereby helps in accurate estimation of position and speed of the robot. The position and orientation are estimated by the advanced dead reckoning method, using encoders' readings. The AmigoBot maintains its internal coordinate position in platform-dependent units, but reports the values in platform-independent units (millimeters and degrees) in the standard Server Information Packets (SIP), which also contains robot's positional and Orientation details - Xpos, Ypos, and Thpos. On startup, the robot's X, Y and Orientation (θ) readings are set to zero (0, 0, 0). The robot points along the positive x-axis at 0 degrees and the orientation vary in absolute angles (between +180 to -179). Fig 4.3 illustrates the robot's internal coordinate system.

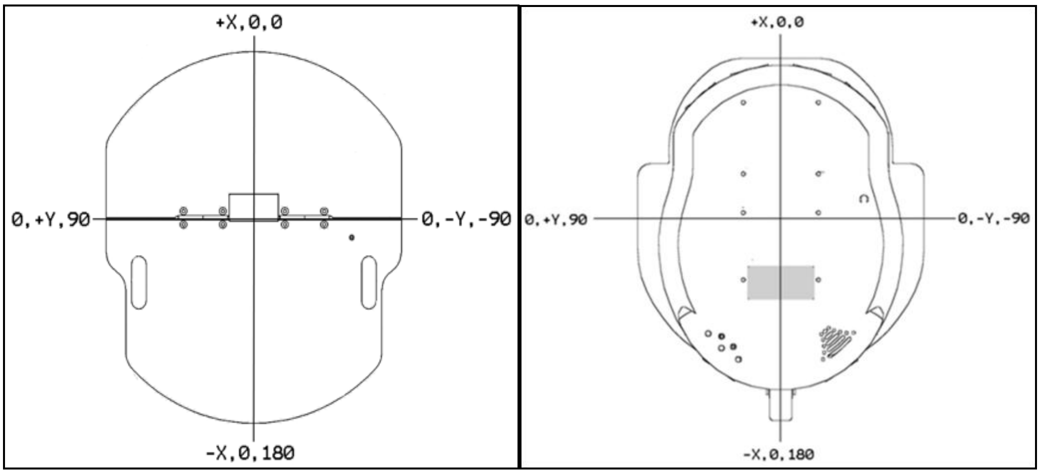


Fig 4.3 Internal coordinate System of Pioneer P3DX and Amigobot

The microcontroller based embedded controller of the robot controls the linear and angular velocity of the robot. It also estimates the robot's current state, control information, position data (x, y, and theta), battery charge status and range information based on the sonar sensor data. Apart from this, additional data can also be transferred between the two robots based on the control algorithm that is implemented on it.

4.2.2 Software Details:

Additionally, the SDK library supplied along with the ARIA software provide a native application programming interface in C++ (this feature is also available for the Python and Java languages using a wrapper interface layer). The package also contains shared libraries, source codes, DLLs, header files, documentation manual, example programs etc.

As mentioned earlier, the API provides a strong client-like interface to the robot's microcontroller and the other accessories attached to it. User can create higher level control programs like navigation, path planning and formation control using the API which is run on an onboard computer in the robot or offline through the simulator MobileSim. The API and the robot hardware is well supported by a number of other operating systems like Debian GNU, Linux 3.1, Windows 7, and Ubuntu 12.04.2

The Amigobots & P3DX robots are supplied with a set of advanced Server-client architecture based API library called ARIA-Advanced Robotics Application Interface that helps in easy interface of user developed control algorithms. The Pioneer and AmigoBot are controlled by Operating System firmwares called as ARCOS, AROS, P2OS, AmigOS, etc and they are treated as core "server" processes. These processes manage all the low-level tasks of the robot like: getting sensor data, making a motion, estimating position from odometry or driving other accessories. They do not perform any high level tasks like path planning, etc. The “robot platform” refers to its microcontroller, OS firmware, and any other integrated devices all together. Application-oriented high level complex robotic control tasks can be achieved by running an intelligent “client” program on a PC connected to the robot. Tasks like sensor fusion, obstacle avoidance, localization, features recognition, object detection, mapping, path planning, navigation, etc can be achieved using higher level programmes written in C++ / Java / Python. ARIA is the programming interface that supports these client applications and establishes communication with the robot firmware. It also helps in communicating to any other devices that is connected to the PC and to remote client software through the network. The additional Localization and Navigation package (ARNL & SONARNL) along with laser range finder makes the robot to map any indoor spaces, localize itself within that space, navigate through the map by avoiding obstacles and reach the goal.

4.3 IMPLEMENTATION:

This section details about the tasks involved in the implementation of the control architecture derived in the previous chapter. The controller is run on the simulator and then the results are verified on the experimental platforms (Amigobots), which contribute to the next two sections.

This section begins by explaining the control of a single robot and the in-built algorithms with respect to Amigobot. This is followed by the details on running multiple Amigobots together using communication module. This is extended to how the proposed control architecture makes both the Amigobots imitate as truck-trailer which is the main objective of this work.

4.3.1 Single Robot

A single robot is run using the application programming interface – ARIA, either through simulator or through the robot firmware. The API is an object oriented system and consists of number of classes that manages all the interface tasks. One of them is the 'ArRobot' class, which is the main class that provides: communication with the firmware, receives and transfers data of the robot's operating state, determine commands and trigger tasks within that cycle. Any client application program firstly establishes proper connection between an ArRobot object instance and the robot firmware followed by running the required task of the robot.

4.3.2 Multi-Robots

When multiple robots need to run at the same time by coordinating the movement of each other, an extensible networking protocol known as 'ArNetworking', is provided with ARIA. It is used to add networking services to a robot control program. It sets up client-server architecture, with bidirectional data communication for control and parameter request at a specified time interval. A client may be of operator's graphical interface or the other robots in the group constituting a multi robot system. It also handles the low level tasks of the client like interacting with the server that includes networking, command and server information, packet processing, serial communications, cycle timing and multi threading etc.

Figure 4.4 illustrates the client-server architecture of a single robot as well as the client-server architecture between the robots. This is achieved by means of different server and client programs using ArNetworking. ArNetworking server program is comprised of server base object to which the call back ‘functors’ are added for each request type to be handled. These call back functors are encapsulated with the several prewritten classes to for providing the robot information. When a call back functor for a request is invoked, it receives an ArNetpacket object containing the payload data and a pointer to an object used for sending as reply packet to the client. Further this server base creates the background thread to asynchronously accept client connections and receive client’s request through the default port opened at port number 7272 through wireless TCP/IP protocol. It also creates and retains an instance of the following classes as its services. A client program is comprised of an ‘ArClient’ Base object which is connected to a server via TCP/IP network socket. The server is identified by its network host name & port number and communicates with the server asynchronously by running a background thread. A closed, tight, reliable control loop is built between the robots using ARIA and ArNetworking.

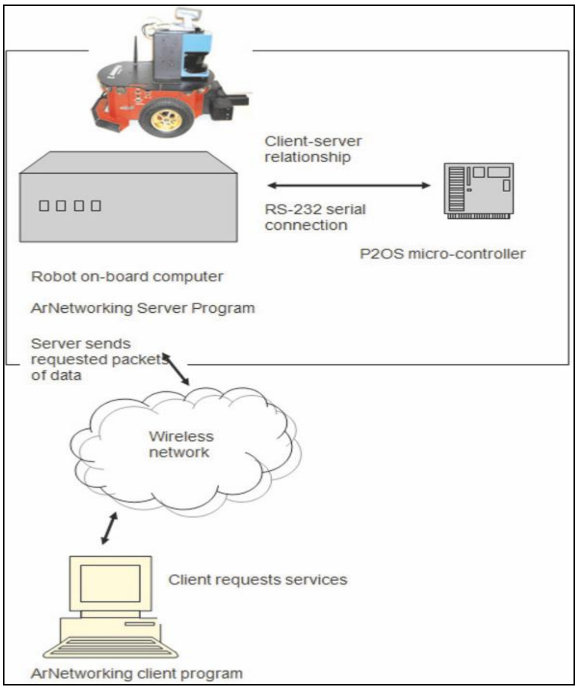


Fig 4.4 Server-client architecture in ARIA robots

4.3.3 Communication between robot and the computer

There are many ways to connect a client PC running an application program using ARIA with the robot or its offline simulator. Any application programs require 'ArSimpleConnector' class for establishing connection. This class will first try to establish connection with the simulator on a local TCP port by default. In case if there is no simulator running, then it will try to connect to any robot that is connected on a local serial port. This feature helps in debugging the program initially using the simulator and later directly on the robot's embedded controller without any modification needed. Command line arguments can also be parsed in this class to specify a specific remote hostname, or a TCP port, or an alternate local serial port that is used for robot connection. Fig 4.5 explains the different means of communication used for Pioneer & Amigobot in this work.

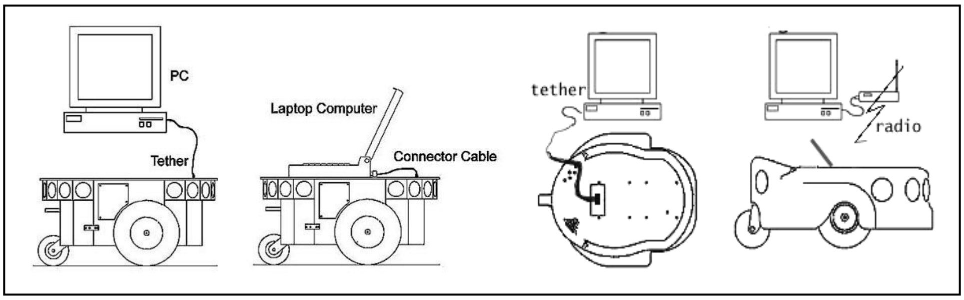


Fig 4.5 Communication with Pioneer P3DX and Amigobot from computer

Data Packets: Information about the robot and its accessories are updated using data packets called as Server Information Packets (SIPs) which are sent by the robot server. The standard SIP is sent automatically for every 100 ms, which is the update frequency. A SIP (also called as sockets) contains robot's current pose, translational and rotational speeds, sonar updates, battery charge data, analog and digital I/O states.

The size of a SIP is 512 bytes of which 506 bytes are about the robot information, 256 bytes is buffer, 64 bytes is for data status, 6 header byte and two footer bytes.

Table 4.2 describes the command line arguments used for connecting with the robot via serial port as well as with an Ethernet-to-serial converter using Wireless network.

Table 4.1 ARIA command line arguments for connection

-remoteHost <Host Name or IP> (or -rh)	Connect with robot through a remote host over the wireless network
-robotPort <Serial Port> (or -rp)	Connect with robot through serial port name; for ex: COM1 or /dev/ttyS0 is default

The truck robot sends SIPs with the information about the truck's current pose, translational and rotational speeds to the trailer robot for the formation controller. It also sends sonar readings, battery voltage and the type of formation shape (inline or parallel or other shapes). Since, the control architecture runs mainly on the trailer robot, this robot receives all the SIPs from the truck robot and sends back only one SIP which contains "stop / go" request to the truck robot.

4.3.4 Behaviors Algorithm

The five different behaviors mentioned in the last section are shown as simple pseudo codes, which will be implemented using C++.

1. Avoid Obstacle Behavior by Trailer:

If a dynamic obstacle in front and cannot move,

Then issue stop command to truck and avoid obstacle by estimating the wheel velocities as shown in 3.5

Else,

Proceed forward by piecewise constant velocity 'v'

2. Formation control by Trailer:

Track the truck in a defined geometric spatial pattern

Receive the Information from the truck

Estimate Wheel velocities using the tracking controller

3. Avoid collision between Truck-Trailer:

If truck is closer,

Wheel velocities are estimated according to the controller equations in 3.4

Else,

Proceed with forward by piecewise constant velocity 'v' using tracking controller equation.

4. Trailer Robot:

If truck robot is moving,

Track the truck using tracking controller

Else if dynamic obstacle,

Issue stop command to truck and avoid obstacle using relationship in 3.4

Else if truck is closer to trailer,

Avoid truck using relationship in 3.5

5. Truck Robot:

If no stop request from trailer,

Choose the trajectory to track and move. Keep updating information to trailer.

If obstacle,

Then perform avoid obstacle

Else,

Proceed with forward piecewise constant velocity 'v' by navigation

The application programming interface ARIA is an object oriented programming interface written in c++ language. However, ARIA also provides Java / Phython interface for the writing control algorithms in these languages. This is because each of these libraries is available with a wrapper that is automatically generated by SWIG. Hence, the behaviors mentioned above are implemented using a set of if-then-else statements using ARIA's different callback functions.

4.3.5 Priority-Based Architecture

Behavior-based controller involves comparison of individual behavior weights and depending upon the weight of each behavior, the controller tasks are prioritized. However, the main objective of this work is to imitate as real truck-trailer system. Hence, the priority is set according to the requirements of a real truck-trailer system that can overcome the common problems encountered.

The highest prioritized behavior is the obstacle avoidance of the trailer. As mentioned earlier, the problem of obstacle avoidance of the follower robots in a multi robot system has not been explored as much as it's been done for the truck/leader robot. Hence, it is important to prioritize the obstacle avoidance of the trailer due to the safety of the system. So, when the front sonar array of the trailer detects an obstacle then the trailer initiates this behavior and sends a stop request packet to the truck.

The next prioritized behavior is the collision avoidance between the truck and trailer also known as the jack-knife problem. Even in case of multi-robot system, when the robots are required to move in narrow spaces, collision avoidance of each other is a serious issue. This behavior is a complex one and is invoked only by the trailer robot depending upon the sensory information.

When the separation distance between the two robots is lesser than the half of the virtual link length, i.e. when both the robots are very close to each other, the front array sonar sensors of the trailer robot are blocked due to the vicinity of the truck robot. In such a condition, the chances of dynamic obstacles getting on the way of the trailer are greatly reduced. Hence, this behavior is not needed in such conditions. However, the behavior can also be invoked under normal conditions if necessary and in that case, obstacle avoidance is prioritized a level higher than this control.

Both these behaviors involve change in the geometrical pattern of the robots, thus disturbing the formation shape of the robots. So, formation controller is given priority next to these controls and will not work when other behaviors are executed. Though its prioritized low, this work is mostly concentrated on formation controller.

Next, the other two behaviors – Formation breakup and Change of formation shape are transitory ones that are used by other behavior controls. These behaviors will not work when other higher level behaviors are executed. And these are not explicit behaviors but rather temporal states that is required by other behaviors.

4.3.6 Overview of the Control Architecture

This section describes the overview of the complete control system in terms of variables, constants and parameters that are implemented as different callback functions in ARIA and the Fig.4.6 illustrates this. It also shows the total number of inputs and outputs of all the behaviors.

The constants and the variables used in this architecture are tabulated in table 4.3. The constant values are picked up by either using experiments or based on the robot parameters or the controller used with the robot.

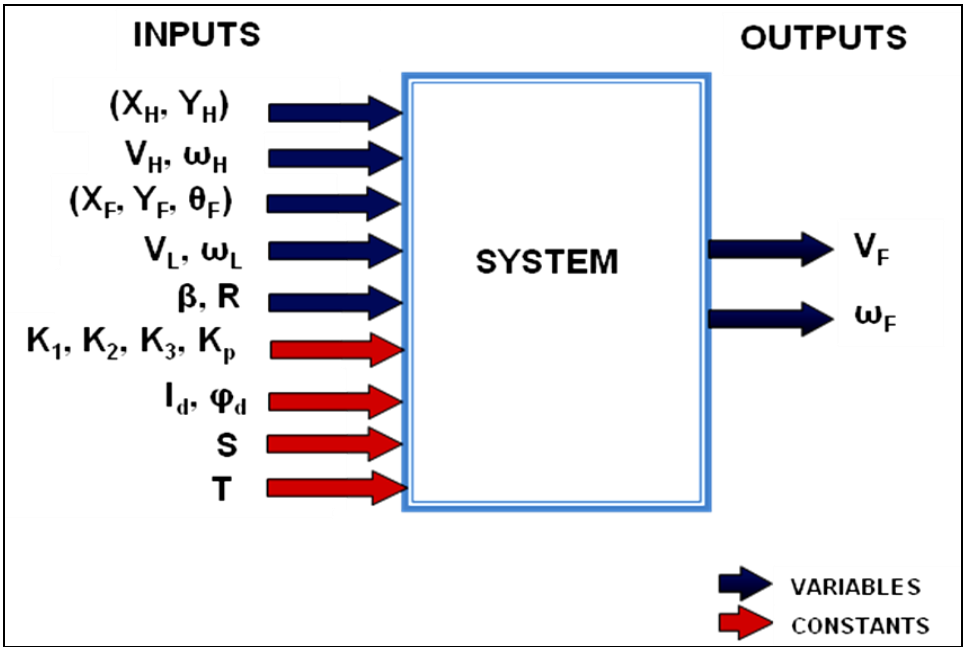


Fig 4.6 Inputs-Outputs of the System

Table 4.2 Constants and Variables used in the system

CONSTANTS	VARIABLES
K_1 - Controller gains = 1.1, 1.1	R - Radius of the ICR
K_2 ,K_3 - Controller gains = 1.1	(X_H, Y_H) - Hinge coordinates
S - Min separation dist = 600mm	v_H, ω_H - Hinge velocity components
T - Wait time for truck = 60000ms	(X_F, Y_F, θ_F) - Trailer Pose
l_d - Linear separation distance = 700mm	v_L, ω_L - Truck velocity components
φ_d - Angular separation distance = 0 deg	$\beta = \theta_L - \theta_F$ - Hinge angle
K_p - Proportional gain = 0.75	v_F, ω_F - Trailer velocity components

4.3.7 Multiple parallel threading

Being an object-oriented system, ARIA is highly multi-threaded and number of support classes like: ArAsyncTask, ArThread, ArCondition, ArMutex, etc. are available to write multi-threaded code. The objects like ArMutex and ArCondition helps in providing a wrapper around system calls that prevents and excludes any other threads from continuing till the mutex object is "locked". The wrapper is ‘pthreads’ functions when used in Linux and ‘CriticalSection’ functions for Windows.

In this work, the truck robot runs three parallel threads – robot, server and sonar. The trailer robot runs four threads – robot, client, sonar and additional packet info. Each of the thread is locked for protecting the data using ‘ArMutex’, thereby preventing other synchronization objects from accessing the same data.

4.3.8 Truck Robot & Trajectories

As mentioned in the research assumptions, this work is not concentrated on the truck navigation and path planning part. However, there are specific planned trajectories that are given as input to the truck and this section explains these paths and how it is implemented on the truck robots. By giving different trajectories, the performance of the individual behaviors can be tested and improved.

There are four types of trajectories provided to the truck such that all the individual behaviors can be tested separately as well as a complete system:

1. Safe Wander
2. Tele-operated
3. S-arc and deep s-arc
4. Circular and straight line

Safe wander:

This trajectory makes the truck to move freely avoiding obstacles. This helps in testing the complete control as well as individual behaviors. This works perfectly in known and unknown environment without running into obstacles. The piecewise constant velocity set for the robot is 40mm/s, however can be changed by editing the source code. The sensing range of each sonar sensor is up to 5 meters and each sonar are positioned with a specific sensing angle for both the robots as mentioned in figure 4.2 of section 4.2.1. Hence, when any obstacles is sensed in this range, it is avoided by the robot by calculating the avoid distance as shown in fig 4.7.

The algorithm is as follows:

If sonar senses an obstacle within avoid distance (d_{avoid}),

Turn till the sonar readings are clear and move forward

Else if there are no obstacle for a period of time 't's

If previous turn was left, then take right turn

If previous turn was right, then take left turn

Else

After turning θ deg., proceed forward by piecewise constant velocity 'v'

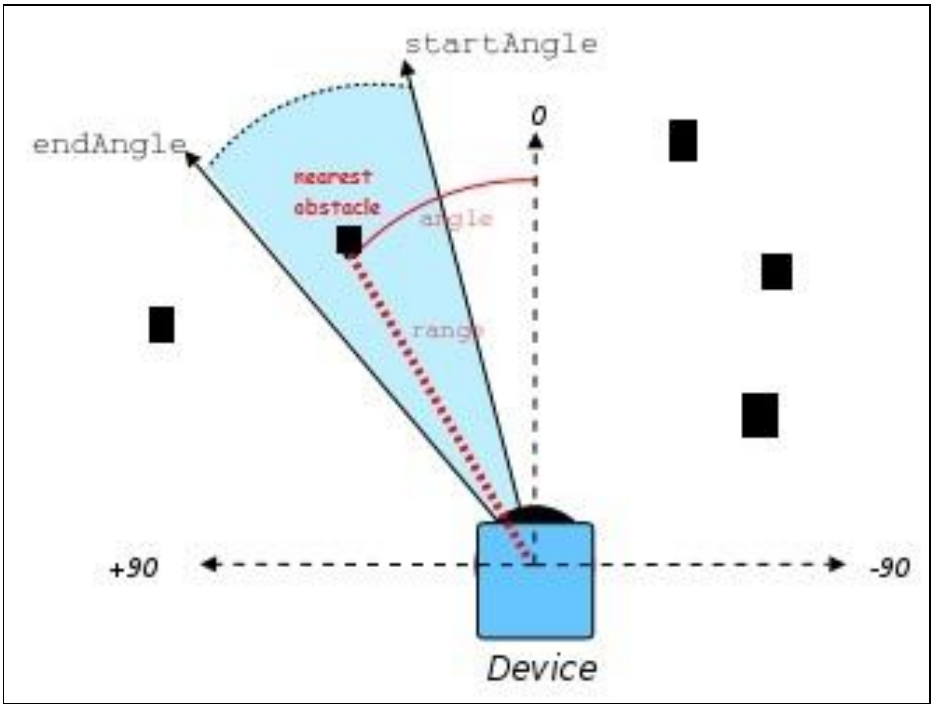


Figure 4.7 Avoiding obstacle using sonar for safe-wander

S-arc and Deep s-arc:

The truck moves in an s-arc trajectory defined by the user. The inputs to the trajectory are v – velocity of the truck and r – s-arc radius. The truck takes these inputs and calculates the corresponding angular velocity ω needed at the specific intervals of turn. Since the path is predefined, the higher level controls which needs flexible formation shape cannot be tested under this trajectory. However, the formation controller can be accurately tested for formation errors using this trajectory. Even other behaviors like formation shape change and formation break-up can also be tested. The truck robot runs continuously until a stop command is issued by the user. Since the velocity is set by the user, the robot will not able to avoid obstacles on its own.

Circular and Straight-line:

The truck moves in a circular or straight line trajectory defined by the user. The inputs to the circular arc are v – linear velocity of the truck and r – radius of the circular path to be taken by the truck. The truck takes these inputs and calculates the corresponding angular velocity (ω) using $\omega = v / r$ and sets ω accordingly for the circular arc. The truck robot runs until a stop command is issued by the user. Since the velocity is set by the user, the robot will not able to avoid obstacles on its own. So, only formation behavior with different formation shapes and formation break-up can be tested using these trajectories. Figure 4.8 is a screenshot of the different trajectories run in MobileSim.

Tele-operation:

The truck is operated in real time using a joystick or keyboard without any preplanned path by the user. This trajectory helps in testing the control under different conditions like – sudden change in velocity, a sharp turn, backward movement of the truck and sudden jerk in movement. The truck is capable of avoiding obstacle using its Sonars.

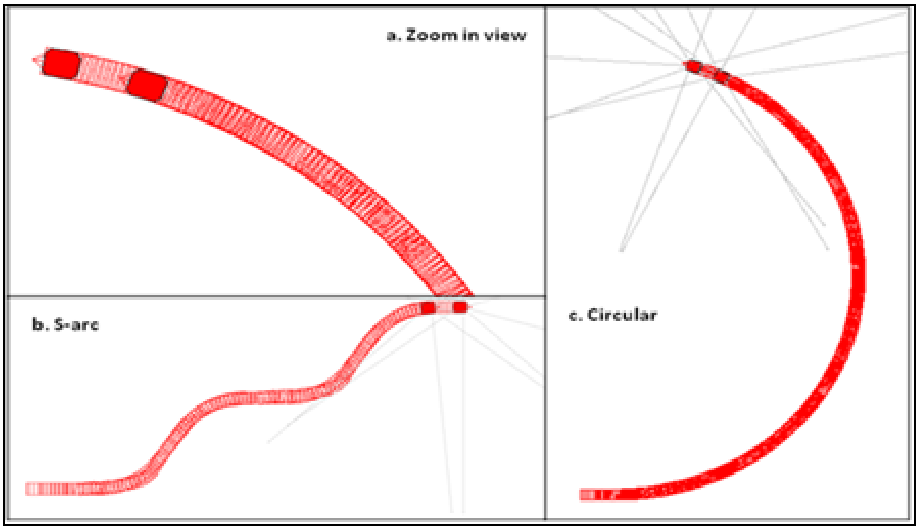


Fig 4.8 Screenshot of Simulator - Different Truck-Trajectories

4.4 SIMULATION STUDIES

Simulation studies are carried out to evaluate the different features of the proposed control algorithm by assessing the responses of all the five behaviours for various sensory inputs. Simulations are also carried out to evaluate the switching response of the behaviours to yield the desired output.

4.4.1 Simulator and Maps:

The MobileSim - Graphical User Interface (GUI) simulation environment generates a map automatically by driving the robot with a laser range finder (or) using the Mapper3 / Mapper3Basic (additional software supplied along with the ARIA for creating maps for the simulation environment). All the maps are saved as MobileRobots.map file (Same map format for ARNL and SONARNL). Mobilesim makes use of this map file to simulate walls and other objects and obstacles in the environment. MobileSim places the simulated robot model in that environment and provides a simulated control connection accessible through a TCP port. ARIA can be connected to TCP ports in the same way as serial ports. It is also used for debugging the code by running the simulation environment prior to experimentation.

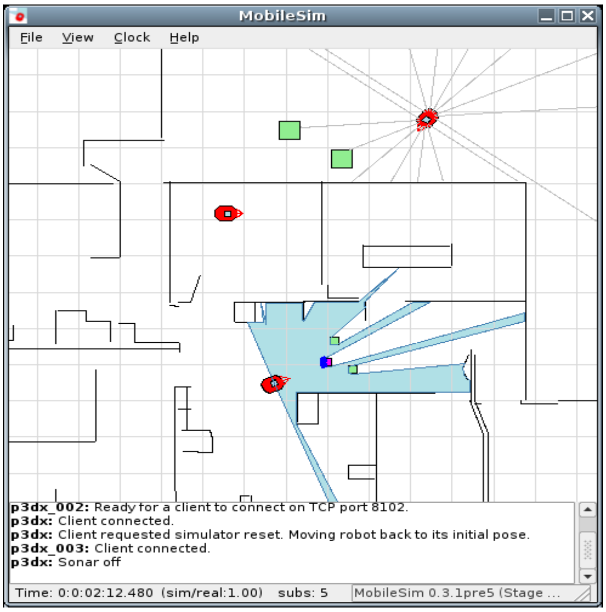


Fig 4.9 Mobilesim – Simulator for Amigobots and Pioneer P3DX

The Mapper3-Basic is an application program, which can be used along with SONARNL or ARNL and MobileSim for creating maps. It can also used to create the real environment by adding goals, docks, obstacles, forbidden lines, home points, etc into the generated map. These maps are used to localize the robot and navigate using MobileSim. Mapper3-Basic saves the map using bitmap formats and many other options like upload and download maps.

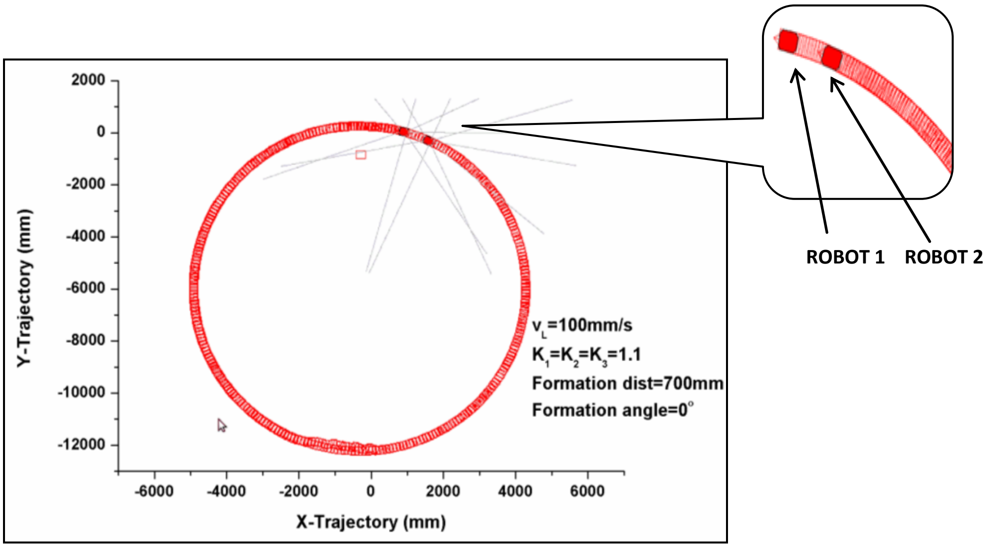
4.4.2 Multiple Robots in simulator

The goal of MobileSim is to supply an efficient simulation environment for multiple mobile robots. The MobileSim is capable of connecting upto 200 robots at the same time. However the the best performance in terms of response time can happen only with 25 robots and so its always good to restrict the number of robots to 25 or less. Since its inconvenient to create huge number of robots at a stretch during MobileSim startup, a command line argument -R option is provided, that will create a new robot on demand for each client that connects and gets destroyed when the client exits, which doesn’t preserve the robot's state.

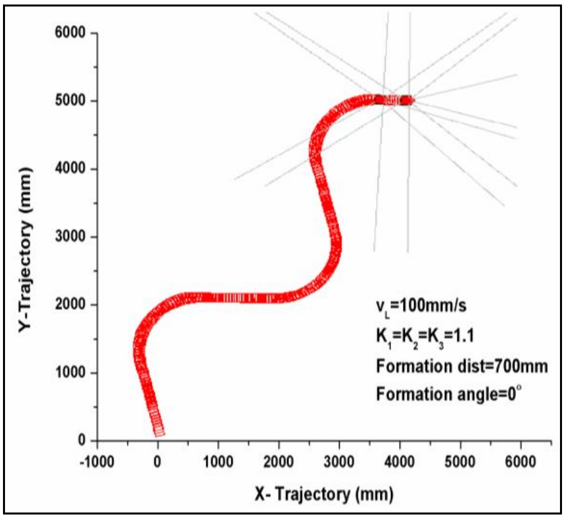
4.4.3 Performance of tracking controller

This section first runs the tracking controller with controller gain values taken as 1 similar to the tracking controller gain values by Kuppan et al. (2008). Later, the controller gains are optimized suitably. The performance of the controller is studied by running for different set of trajectories. S-arc, circular and straight line trajectories are given as input to the truck and error analysis done for each of the trajectory. This error analysis helps in finding the optimal linear velocity for the truck at which the trailer is able to track the truck with minimum linear and angular errors. In the second part of this section, the controller gains are optimized using Zeigler-Nicholas method. The control algorithm converted into ARIA C++ code is run in its simulator and the screenshot of the truck-trailer formation in different trajectories are shown in fig. 4.10. Each trajectory is discussed here in detail.

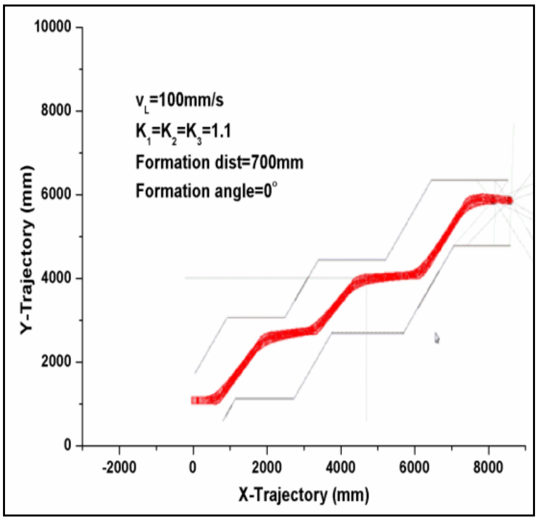
S-arc type trajectory: The truck is given an s-arc path with linear velocity as 165mm/s and the trailer is placed at a distance 660mm, making the hinge distance to be 330mm and the controller gains are to be 1. A total of 5 graphs are plotted to study the performance. Figure 4.11 shows the X-Y trajectory of the truck, trailer and the virtual hinge and figure 4.12 compares the orientation of both the robots.



a. Circular



b. S-arc



c. Safe-wander

Fig 4.10 Mobilesim – Screenshot showing different trajectories for Truck-Trailer

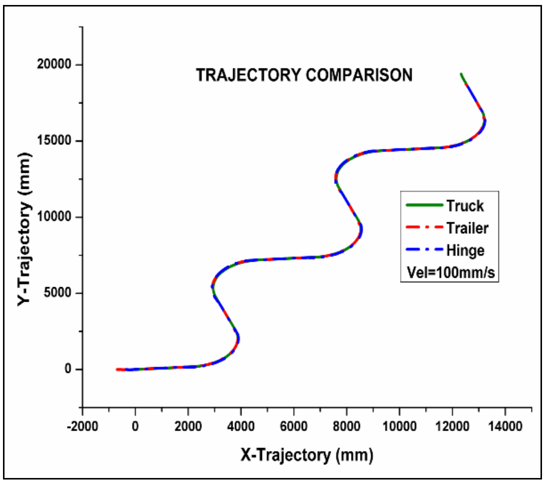


Fig 4.11 X-Y Trajectory of Truck, Trailer, Hinge

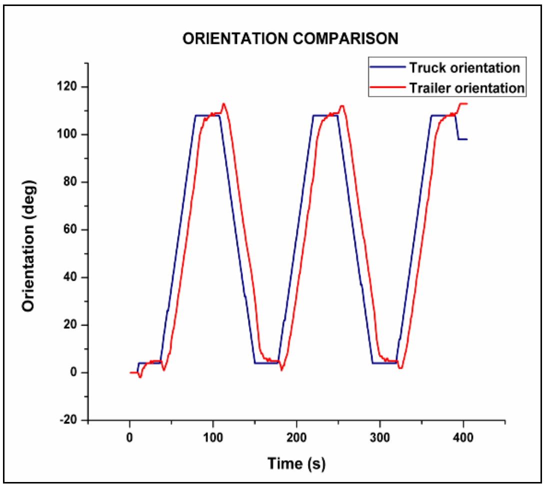


Fig 4.12 Comparison of truck-trailer orientation

Minor shoot-up of the orientation angle of the trailer is observed that is reflected only at the sharp turn angles in the s-arc trajectory as seen in figure 4.12. This shoot-up can be controlled, which will be discussed in the later part of this chapter.

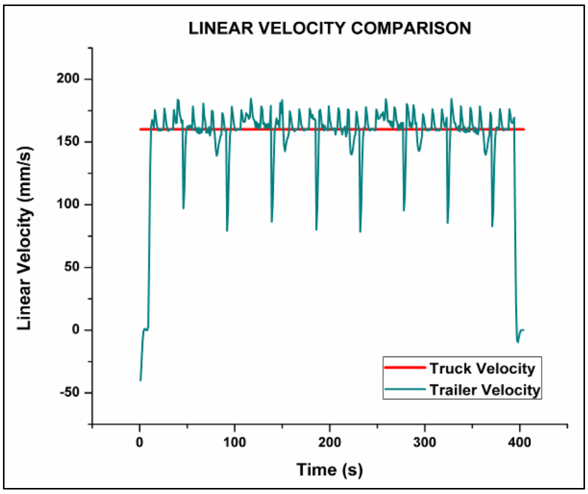


Fig 4.13 S-arc Linear velocity comparison

In the fig 4.13, it is seen that how the velocity of the trailer oscillates between 175mm/s and 130mm/s and there is a drop to 80mm/s in the velocity at regular intervals. These regular intervals are the bend point of the s-arc, where the change in sudden orientation of the truck causes this sudden surge in the trailer velocity.

Figure 4.14 and 4.15 explains the linear separation and angular separation errors between the two robots. The average error in terms of orientation is 1.529 rad and 170mm in terms of the position. These errors are large owing to the facts that the controller has not been optimized for the better performance.

Choosing Truck Velocity: The tracking controller is tested using circular and straight-line trajectories for different velocities. These sets of first experiments are run to find out the velocity at which the controller performs the best. A set of four test runs are carried out for each trajectory by varying the velocities. Table 4.3 illustrates the different configurations of the simulation tests run for evaluating the performance of the controller. All the desired values like linear separation, angular separation, linear and angular velocity and the controller gains K_1, K_2, K_3 are taken for this initial runs. The controller gains are optimized to specific values in the next section.

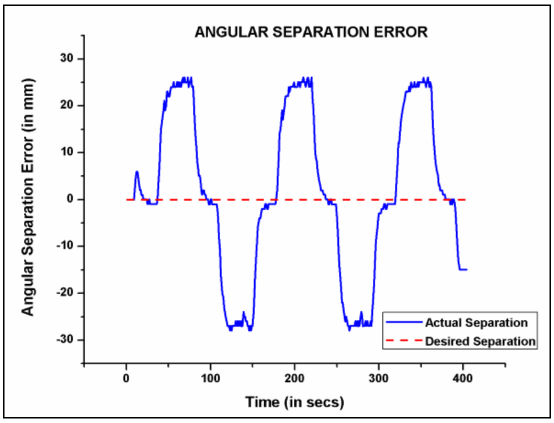


Fig 4.14 Angular Errors in S-arc

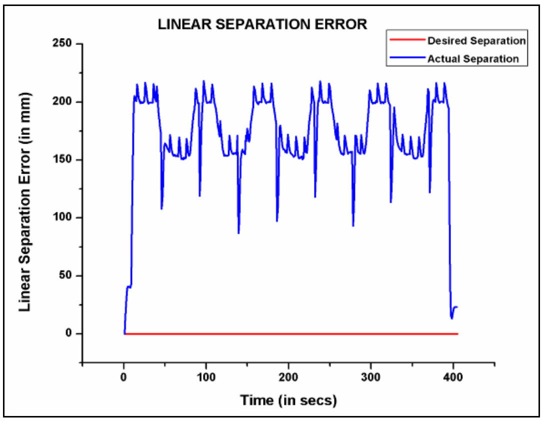
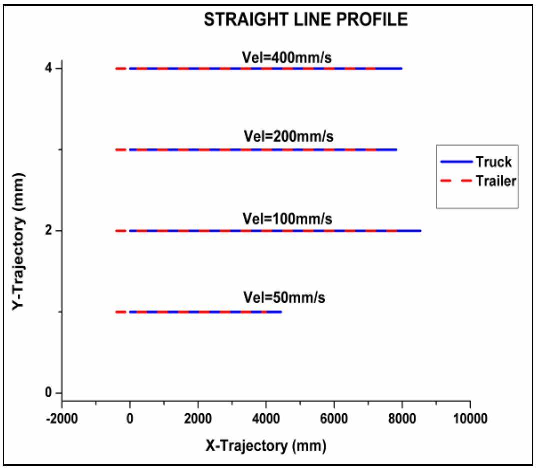


Fig 4.15 Linear Errors in S-arc

Table 4.3 Tracking Controller - Simulation test runs

Trajectory	Linear separation $l_d = 400$mm & Angular separation $\varphi_d = 0^\circ$ Controller Gains K_1=1, K_2=1, K_3=1 V_L = Truck linear velocity in mm/s ω_L = Truck angular velocity in rad/s Arc radius for truck = 100mm (for circular)			
	V_L=50	V_L=100	V_L=200	V_L=400
Straight-line	$\omega_L = 0$	$\omega_L = 0$	$\omega_L = 0$	$\omega_L = 0$
Circle	$\omega_L = 0.5$	$\omega_L = 1$	$\omega_L = 2$	$\omega_L = 4$

Straight line: Basic study begins with plotting all the four plots in a single graph showing the X-Y trajectories at different Y values. It is observed from the figure 4.16 that as the velocity of the truck increases, the distance between the truck and trailer also increases. This is because the trailer responds slower and so the error is more at higher velocities. Fig 4.17 supports this by clearly showing the dip in the trailer velocity when compared to truck velocity. The truck velocity is taken to be less than 100mm/s for all the hence forth simulation runs and 50mm/s or 75 mm/s are ideal for this controller.



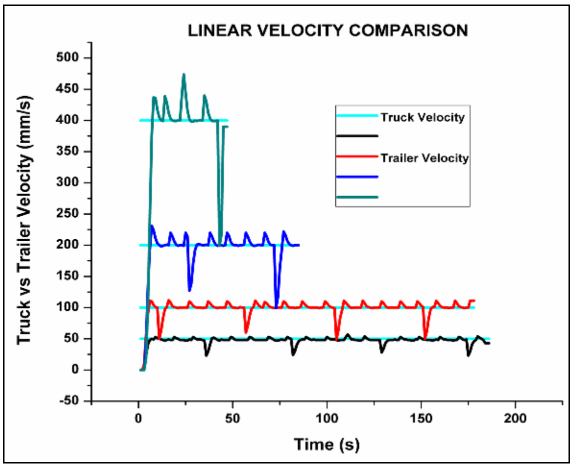


Fig 4.16 Straight Line X-Y Comparison Fig 4.17 Straight Line -Velocity Errors

Circular trajectory: The above result is confirmed with circular trajectories by running it at different velocities as mentioned in table 4.3. Since this involves displacement in X-Y axes, the dip in trailer velocity is found in figure 4.18 for all the velocities 100mm/s, 200mm/s and 400mm/s except with 75mm/s linear velocity of the truck. This also proves that this tracking control works well for lower velocities of the truck.

In straight line motion, for higher velocities, the linear error is more and linear oscillations are observed and at slower velocities, the error is less with no linear oscillations. Hence, 50-100mm/s are the optimum linear velocity range for all trajectories.

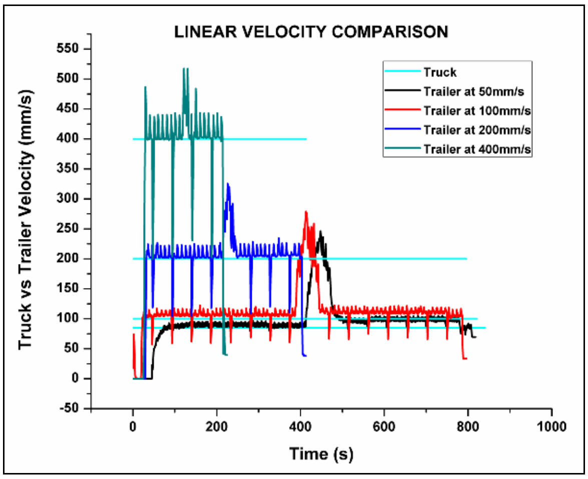


Fig 4.18 Linear Velocity Errors for circle

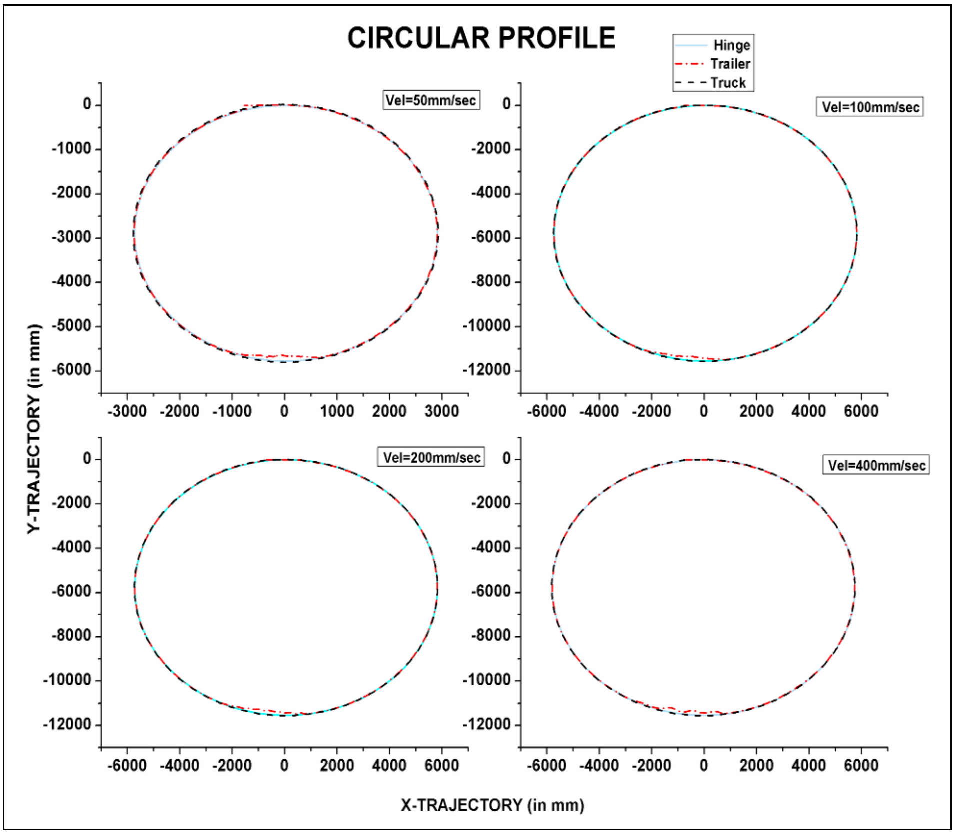


Fig 4.19 Circle - Trajectories with X-Y

Controller Gain Optimization: The simulation and experimental results obtained in the last section clearly shows that before formation convergence, there is an initial error shoot up. This error asymptotically goes to zero as the robots continue their motion. It also showed that if the linear velocity of the truck robot is increased, then there are a lot of oscillations in the trailer's tracking performance. This is because the controller gains K_1, K_2, K_3 were taken to be 1, 1 and 1 based on literatures and not specifically optimized for this controller. From equation 3.24 &3.25, it is evident that these gains depend on the linear and angular velocity of the trailer. It also shows that the controller gains have a large impact on proper tracking of the required trajectory and not all the values of the gain results in successful tracking of the trailer robot. Therefore, careful gain tuning is necessary to achieve the best performance of the formation controller.

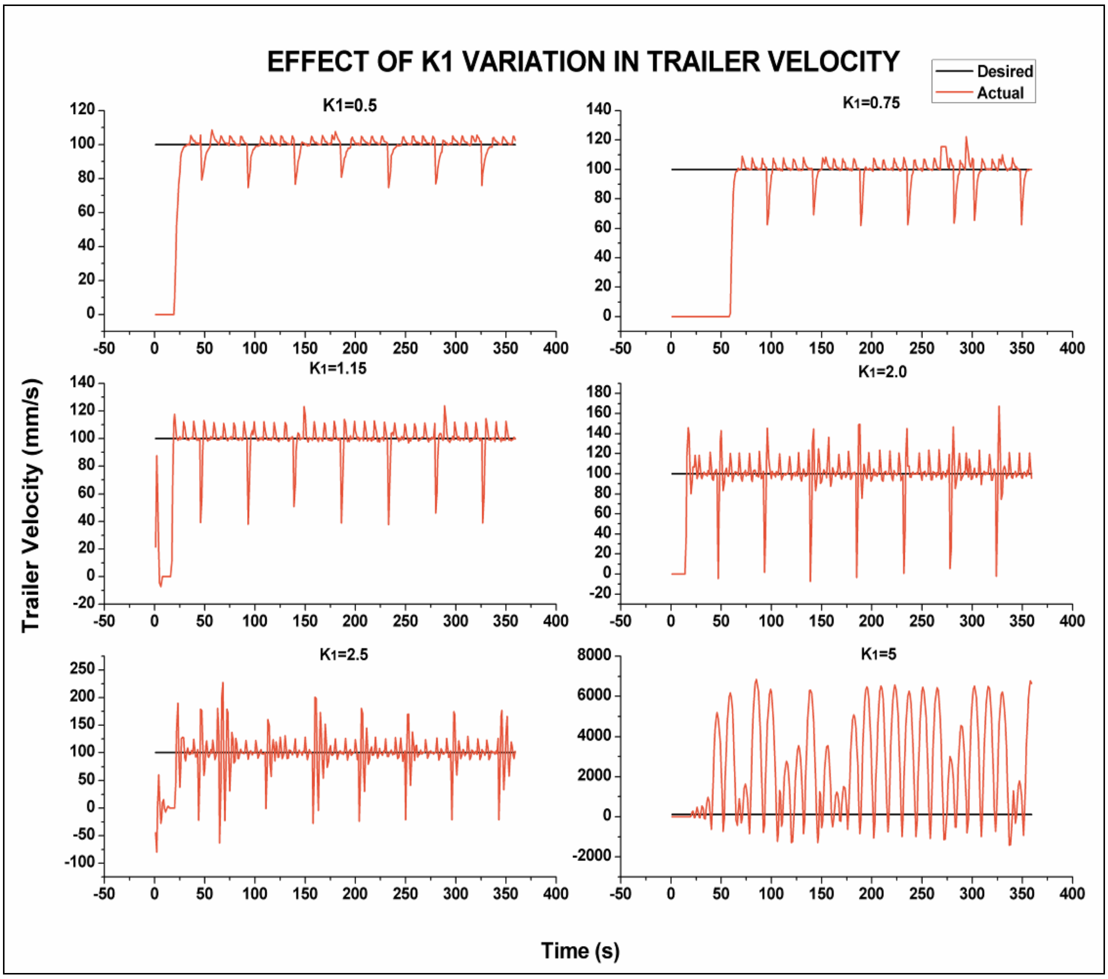


Fig 4.20 Circle Trajectory - Effect of Controller gain K_1on trailer velocity

Generally, controller gains can be optimized using theoretical or experimental way depending upon the controller. Since the proposed controller can be easily implemented in simulations and experiments, controller gain tuning is done by this method. A set of 20 simulations each are conducted to fine tune the values of the controller gains.

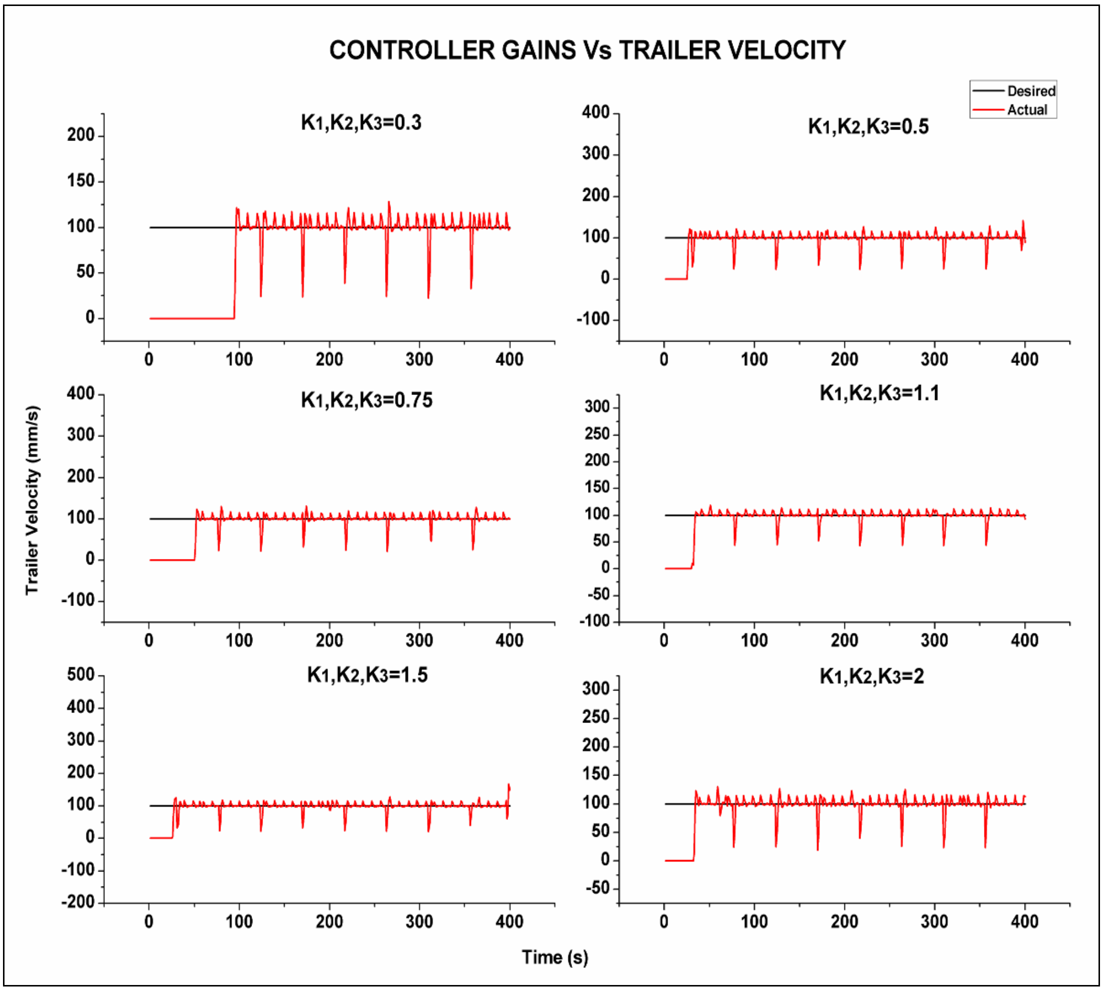


Fig 4.21 Circle Trajectory - Effect of all the three Controller gains on trailer velocity

It is found out that if the controller gain values are increased, it reduces oscillation and overshoot but at the cost of huge tracking errors. The results are verified by choosing the value of K_1 and K_2 by the Ziegler Nichols controller tuning method. It is based on the relationship provided by Ziegler Nichols for a proportional only controller as K_1 =

0.5 Ku_1, K_2 = 0.5 Ku_2 and K_3 = 0.5 Ku_3 where Ku_1, Ku_2 and Ku_3 are the ultimate gains of K_1, K_2 and K_3 at which the output of the control loop oscillates with a constant amplitude (Ku_1, Ku_2 and Ku_3 =2.2).

4.4.4 Presence of Virtual-Link

As mentioned, the presence of a hinge point on virtual link, which is nothing but a control point outside the robot's coordinate system gives a better control over the complete system. This section explores the importance of this control point. Table 4.4 shows four different hinge points moving from the truck robots' coordinate system to the trailer's coordinate system. The different plots obtained by running simulations in accordance to this table are also shown in figures 4.22, 4.23 and 4.24. The truck robot is given a circular trajectory with linear velocity 100mm/s and an arc radius of 100mm. Error analysis is done to check the best range of the hinge points in between the robots for a range of 700mm between them.

Table 4.4 Virtual Hinge - Simulation test runs

Trajectory	Angular separation $\varphi_d = 0^\circ$ Controller Gains K_1=1.1, K_2=1.1, K_3=1.1 Truck linear velocity V_L = 100 mm/s Truck angular velocity ω_L = 1 rad/s Arc radius for truck = 100mm (for circular, s-arc)			
	Hinge 1	Hinge 2	Hinge 3	Hinge 4
Circle	Hinge Pose = (350,0,0)	Hinge Pose = (500,0,0)	Hinge Pose = (600,0,0)	Hinge Pose = (700,0,0)

The average linear separation errors for each of the different poses are: 80.41mm, 178mm, 198mm and 547mm respectively. Similarly, the average angular errors are:

0.0584rad, 0.0864rad, 0.1346rad and 0.1174rad respectively. From this, it is evident that the controller has better control over the angular deviation of the system, due to two angular control gains. It is also found that as the distance between the two robots increases and the hinge point is at a greater distance from the truck; the formation errors also increase.

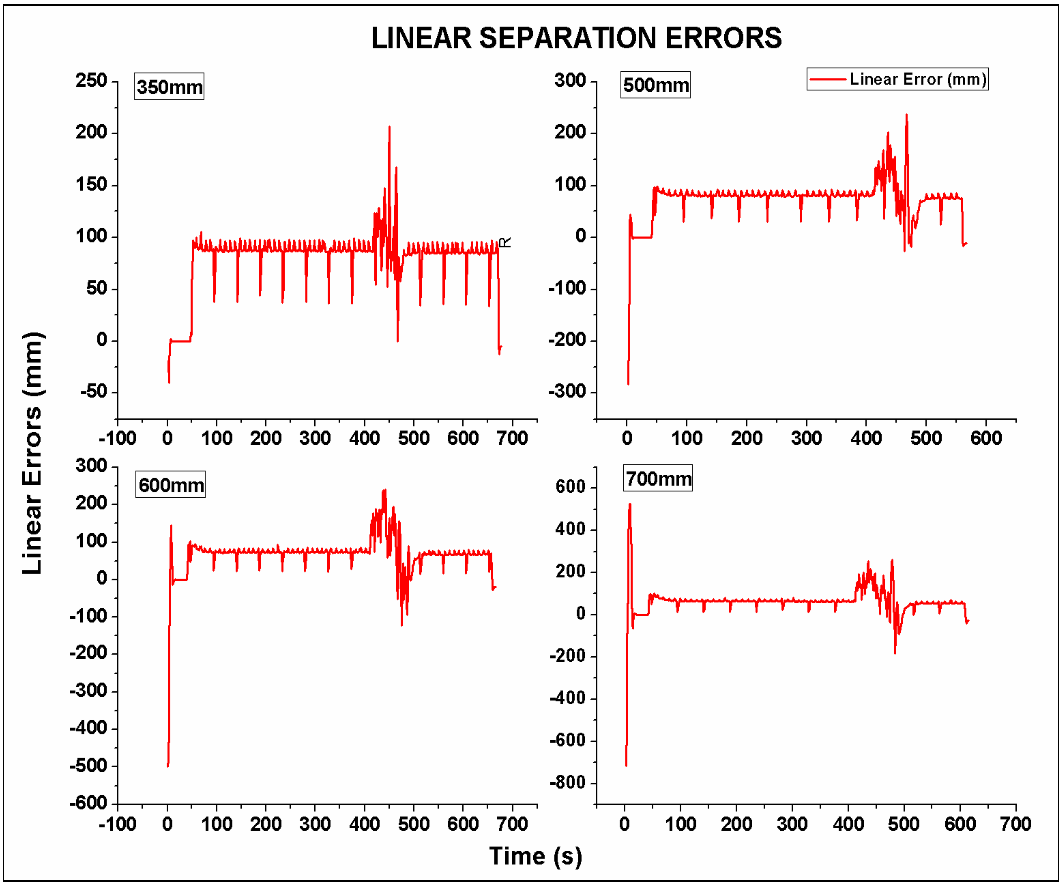


Fig 4.22 Circle Trajectory - Linear Errors of the controller for different hinge poses

Figure 4.24 clearly shows the variation in the trajectories of the truck and trailer. When a hinge point is taken closer to the truck robot, there is a better control over the system. So, if the hinge point is taken on the truck robot's centre of gravity as in Kuppan et al.(2008), the system yielded good results. But, having a virtual point as the control point has the advantage of having wide range of control area. Hence this experiment clearly shows the importance of the virtual hinge point that governs the controller that overcomes all the problems of the other controllers in the architecture.

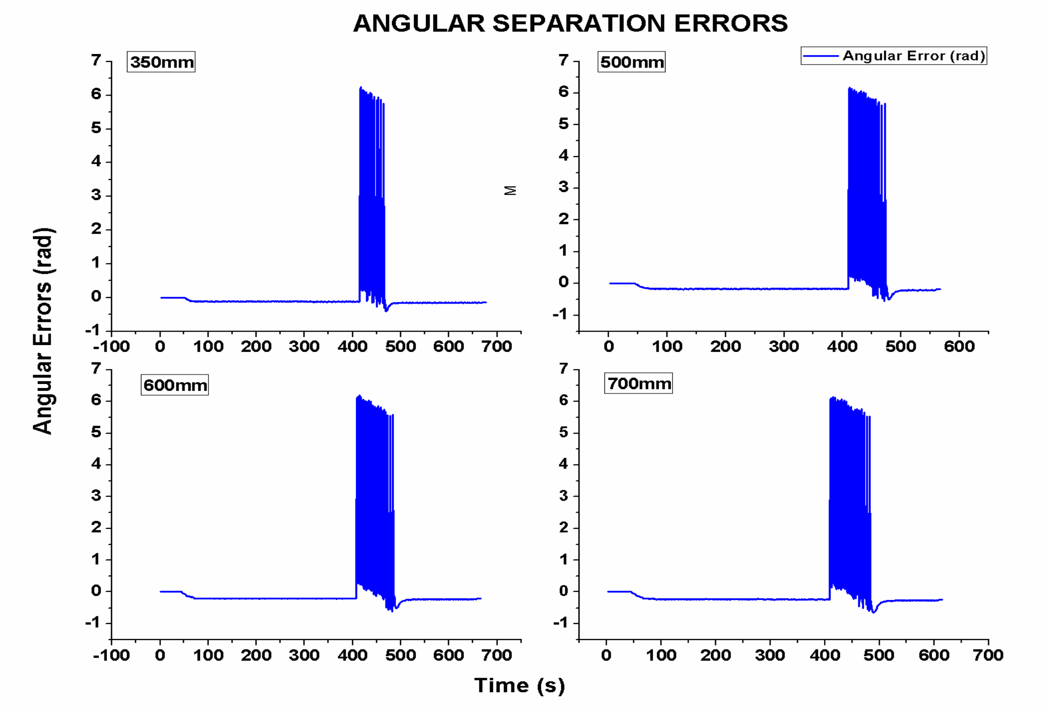


Fig 4.23 Circle Trajectory - Angular Errors of the controller for different hinge poses

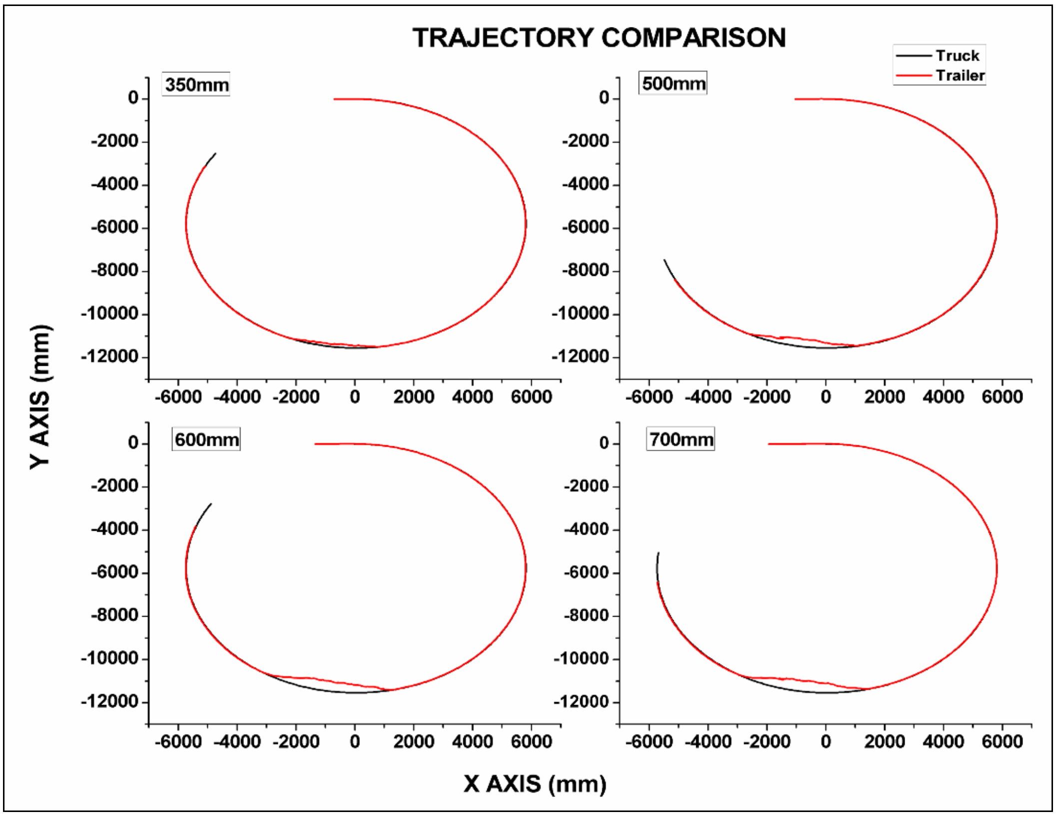


Fig 4.24 Circle - Trajectory Comparison for different hinge poses

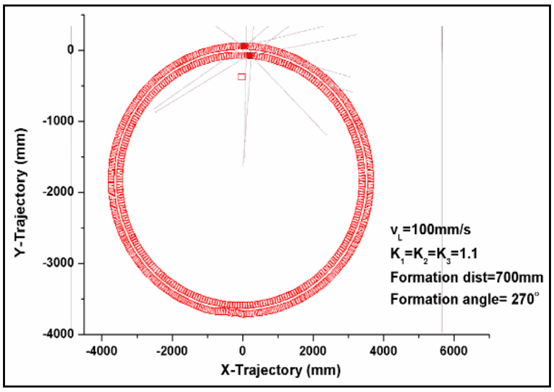
4.4.5 Parallel movement

The second important study on this controller is to detach both the robots to move in parallel to each other. The feasibility of the existing controller for a parallel formation is studied in this section. The same set of simulations that were run for a truck-trailer formation in section 4.4.3 is run for parallel movement conditions, in order to test the performance of the formation controller for this shape. This is because in the case of *l*-α controller by Desai et al. (1998), huge formation errors were observed when both the robots were kept one after the other as a truck-trailer system (in-line formation). This is because the controller has three control points - one in the leader robot and two in the follower robot. If any change in the formation of the system causes all the three points to be collinear, then the controller fails. Hence, it is important to validate the controller for different formation shapes starting with the parallel formation. Another set of simulations are also done to evaluate the performance of formation shape change from truck-trailer to parallel movement and vice versa.

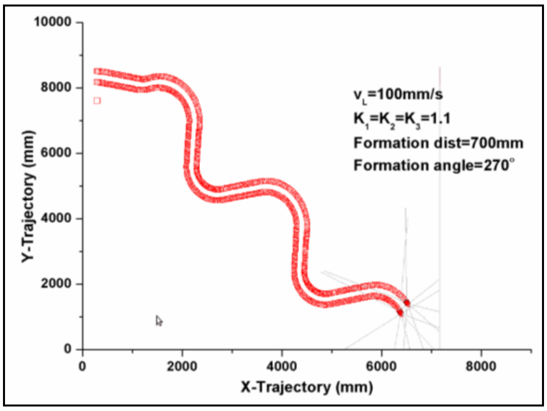
Table 4.5 Formation Control - Parallel Movement - Simulation test runs

Trajectory	Linear separation $l_d = 400$mm & Angular separation $\varphi_d = 270^\circ$ Controller Gains K_1=1.1, K_2=1.1, K_3=1.1 V_L = Truck linear velocity in mm/s ω_L = Truck angular velocity in rad/s Arc radius for truck = 100mm (for circular, s-arc)			
	V_L=50	V_L=100	V_L=200	V_L=400
Straight-line	$\omega_L = 0$	$\omega_L = 0$	$\omega_L = 0$	$\omega_L = 0$
Circle	$\omega_L = 0.5$	$\omega_L = 1$	$\omega_L = 2$	$\omega_L = 4$

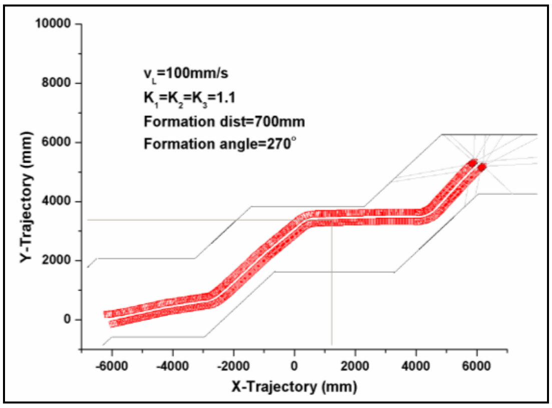
The proposed controller is tested for the truck-trailer to move in parallel by changing the angular separation between the robots φ_d to 270˚ and the linear separation of the two robots l_d to be 400mm. The controller gains are taken to be 1 as per literature and then optimized to suitable values. A set of four different velocities are given to the truck with two different trajectories as given by the table 4.5. The results are discussed in detail in this section. Figure 4.25 shows the screenshot of the Mobilesim simulator which runs the truck-trailer robots in parallel formation. While using a straight line trajectory, the angular errors cannot be calculated, since there is no angular deviation. In case of a circular trajectory, the switch over at 180° to -180° causes oscillations in the system and hence the s-arc trajectory is studied in detail here.



a. Circular



b. S-arc



c. Safe Wander

Fig 4.25 Mobilesim – Screenshot showing different trajectories for parallel-movement

From the straight line profile, the linear errors are found to be 1.797mm, 6.429mm, 23.2531mm, and 48.11mm for truck velocities of 50mm/s, 100mm/s, 200mm/s and

400mm/s respectively as shown in figure 4.26. Hence, as in the case of truck-trailer formation, the linear errors are least for slower velocities of the truck (50mm/s to 100mm/s) and this is the preferable range for the truck to have a smooth formation system. This result has been verified with the circular trajectories also. As mentioned earlier, the update frequency between the two robots about their present state is 100ms. But the trailer robot is not able to respond faster if the velocity of the truck is increased to larger value.

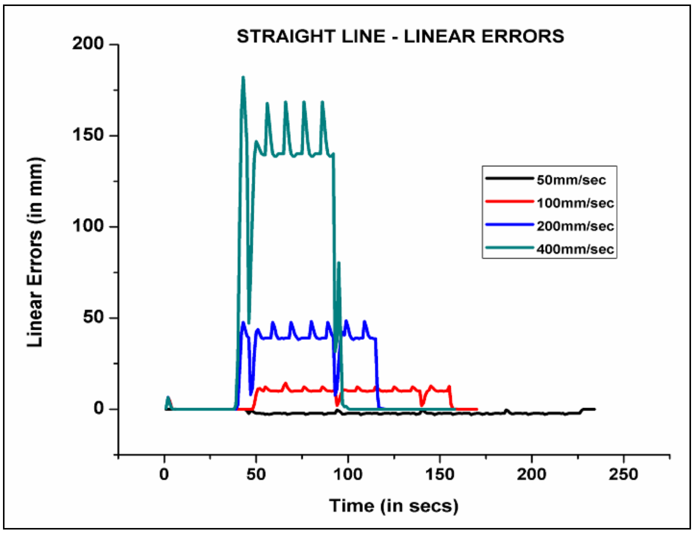


Fig 4.26 Straight line – Parallel Movement – Linear Errors

Similarly, for a circular trajectory, the linear separation errors are: 2.547mm, 7.955mm, 31.439mm and 75.952mm and the angular errors are: 0.0092rad, 0.04293rad, 0.06244rad and 0.01843rad for a truck velocity of: 50mm/s, 100mm/s, 200mm/s and 400mm/s respectively. The s-arc trajectory is run at a velocity range of 140-160mm/s during forward and sharp bends. The formation errors are 19.56mm in terms of position and -0.00538rad in terms of orientation respectively.

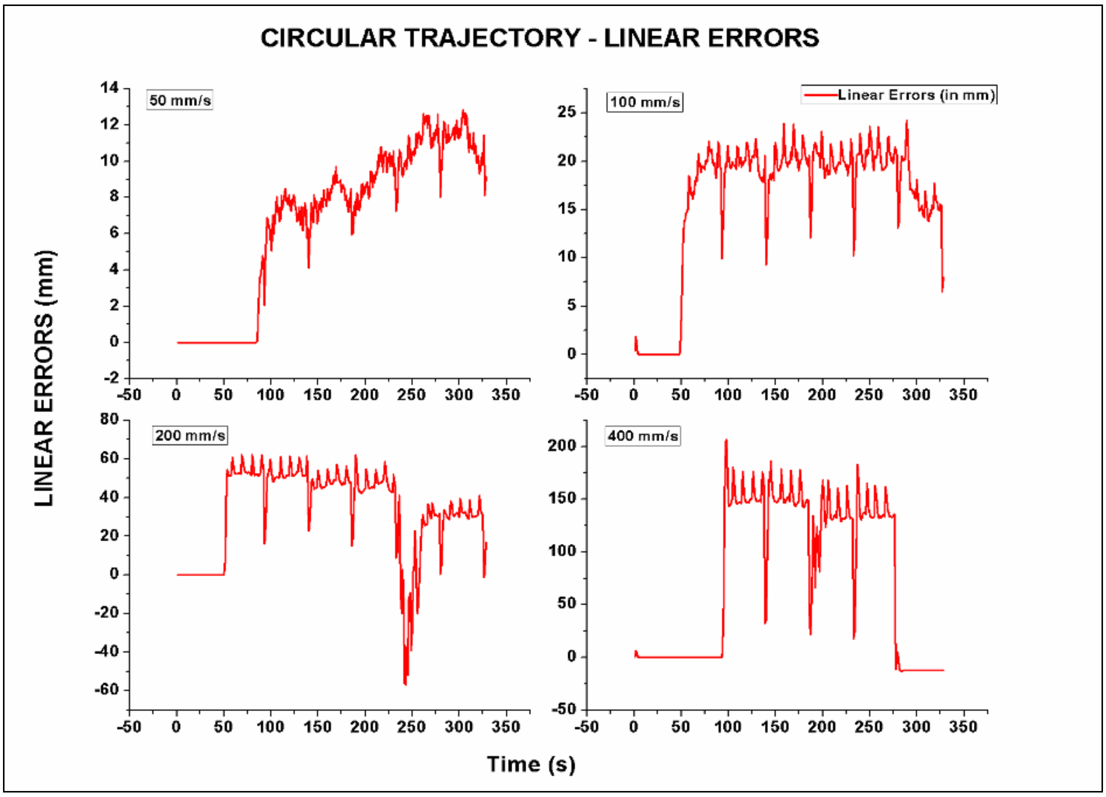


Fig 4.27 Circular Motion – Parallel Movement – Linear Errors

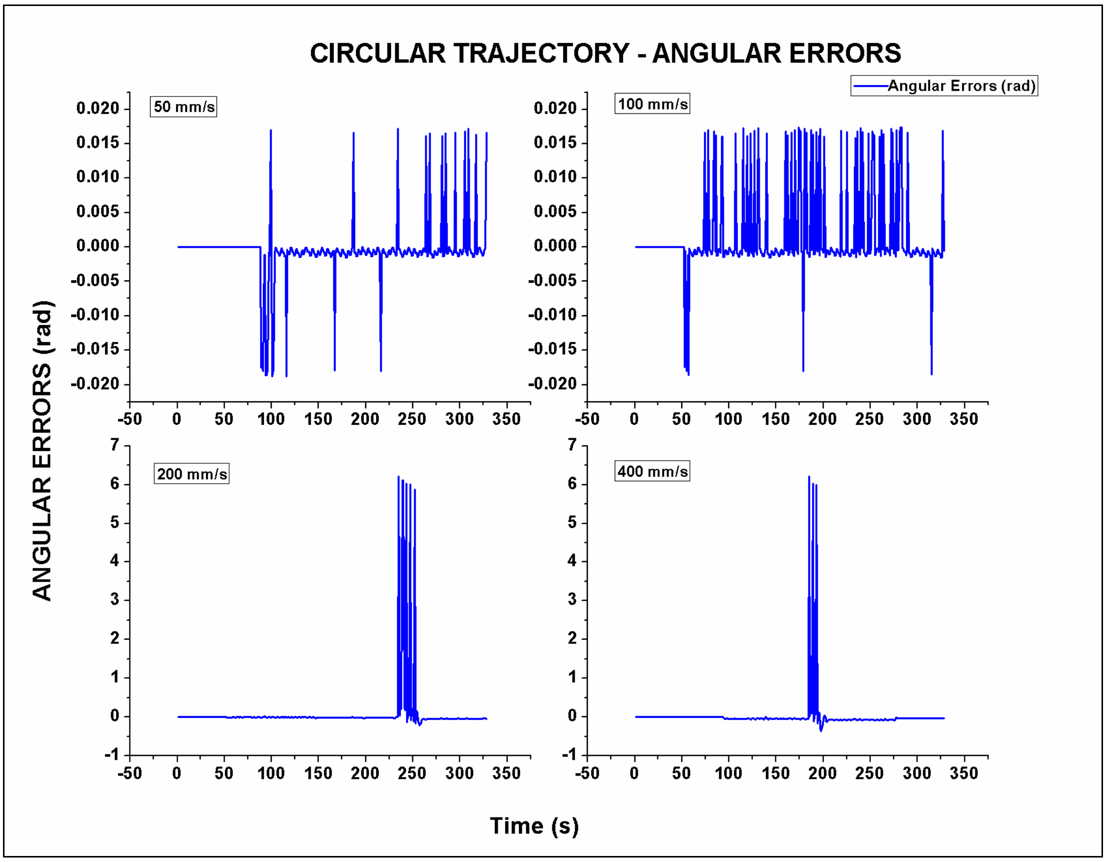


Fig 4.28 Circular Motion – Parallel Movement – Angular Errors

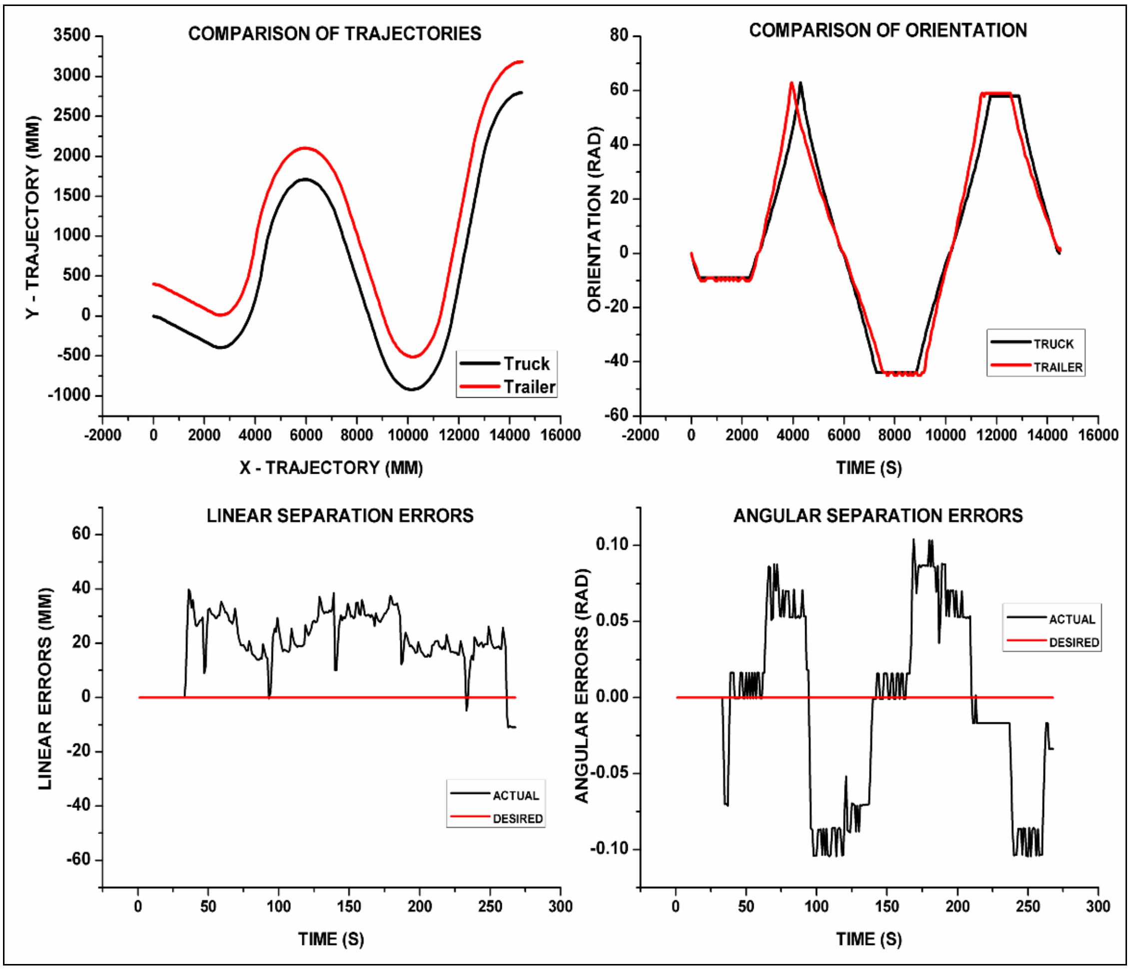
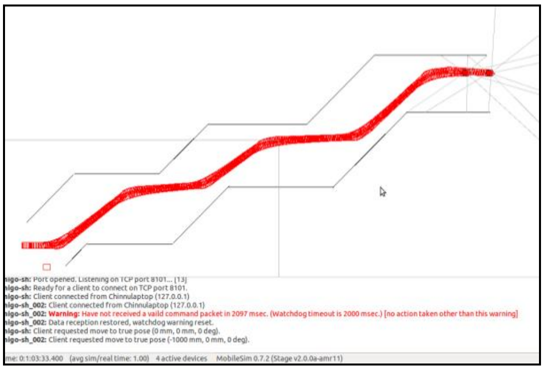


Fig 4.29 S-arc – Parallel Movement

4.4.6 Obstacle avoidance, formation shape change and Collision avoidance between robots

Though these behaviours can be run on any trajectory of the truck, the performance of the individual behaviour can only be measured when a specific trajectory is run that can bring out the features of this behaviour. Formation controller was tested with circular, s-arc and straight line trajectories which gave a proper insight to the controller performance. Here, obstacle avoidance behaviour is tested for the safe-wander trajectory of the truck in both truck-trailer and parallel movement mode separately. Whereas collision avoidance is tested for tele-operation, so that sharp turns and unexpected manoeuvres can be achieved through operating the truck robot remotely using joystick or keyboard.

Figure 4.32 shows the screenshot of the simulator MobileSim in safe-wander mode in both the formation pattern – truck trailer and parallel mode. Switching of formation shape is also dealt with obstacles and without obstacles also. As the controller is found to work satisfactorily for both the required formation shapes, simulations are also conducted to check for the performance of the other two least important behaviours – formation breakup and change in to other formation shape.



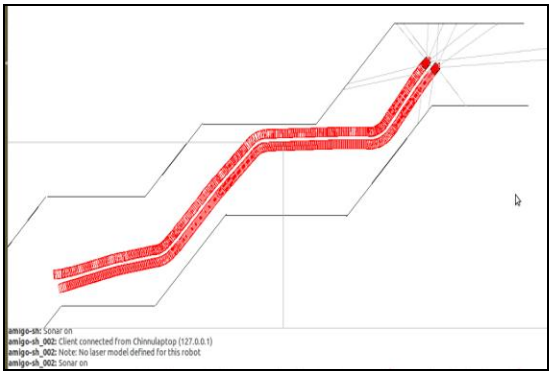


Fig 4.30 Screenshot of MobileSim – Safe Wander – TT Fig 4.31. Parallel Mode

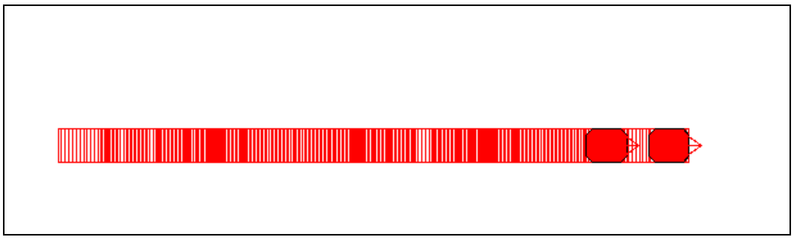


Fig. 4.32. Truck-Trailer movement

Obstacle Avoidance: As mentioned in chapter 3, one of the simplest obstacle avoidance behavior similar to decentralized reactive obstacle avoidance method is deployed here for the trailer. In case of truck-trailer formation, the truck occupies the front of the trailer and the chances of static obstacles are very less. Only dynamic obstacles will cause hurdles to the path of the trailer and hence this environment cannot be simulated rather only real-time experiments are conducted which will be seen in the next section. In case of parallel or any other formation shapes, the trailer faces all type of obstacles and hence it will involve deforamtion of the formation shape using the sense-think-act like behavior. Fig 4.33 gives the screenshot of the Mobilesim when navigating obstacles successfully in parallel formation. This also proves that the client can break the formation shape anytime to be independent and can navigate on its own.

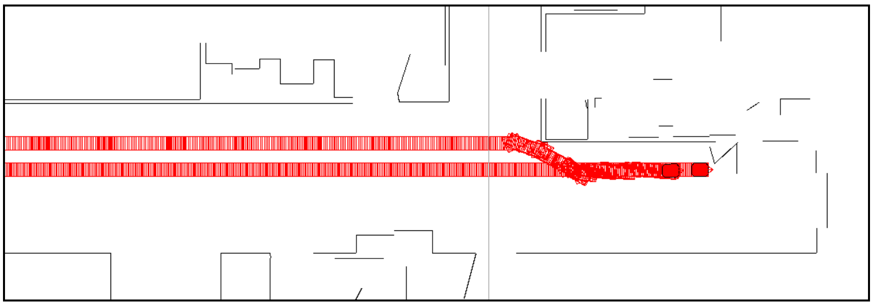


Fig 4.33 Truck-Trailer to Parallel Movement during obstacle in parallel

Switching of Formation Shape: In case of formation shape change, the trailer exhibits two types of change – one when the change is pre-planned or requested by the user and the second where the trailer avoids obstacles and hence changes the formation shape. The second case can be only tested using experiments and for truck-trailer conditions hence will be discussed in the next section. Moreover, switching of formation shape from truck-trailer to parallel mode and vice-versa are evaluated separately.

Figure 4.34 shows the X-Y trajectory plot of both the robots. This shows the performance of these behaviours. The waviness in the plot is due to constant change in the formation shape preplanned by the trailer to be in formation with the truck. This is also because the update frequency of the robot states is 100ms and the controller computes the trailer velocity every single second and hence the error is huge when the formation is continuously changed. However, this is not the problem in the case of formation change during obstacle avoidance

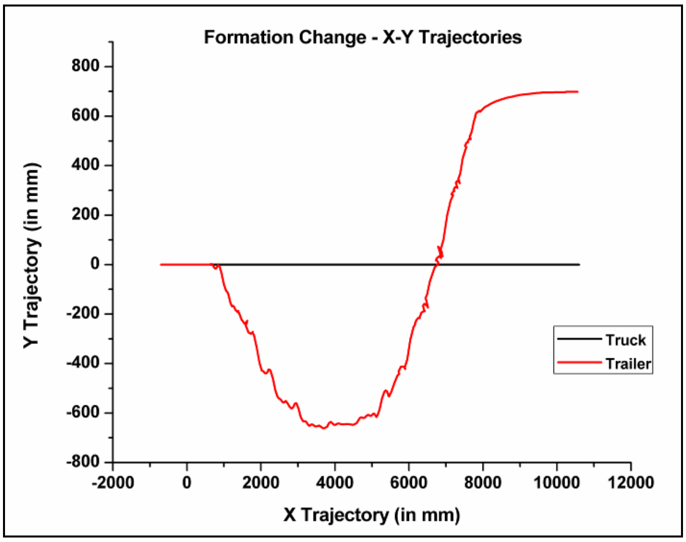


Fig 4.34 Truck-Trailer to Parallel at (45deg and 270deg)

Jack-knife phenomenon: As mentioned in the last chapter, this behaviour is basically a Proportional controller which controls the linear velocity of the follower robot depending upon the turn angle (β) of the leader robot. The turn angle of the truck is inversely proportional to the linear velocity of the trailer robot. A lookup table is derived which shows the range of change of the linear velocity with respect to the turn angle so that the minimum distance between the two Amigobots of 400mm is established. The proportional controller gain K is taken to be 0.75 in order to reach the required linear velocity and the angular velocity is calculated by taking the instantaneous centre of rotation which is the length at which the truck, trailer and hinge point coincide as they rotate about a centre point. The lookup table for controlling the linear velocity of the follower is shown in Appendix II.

4.4.7 Multiple Trailers

As seen in the literature, many controllers failed when more number of robots is added to the formation or the stability of the system is compromised. The stability for the system in terms of the number of trailers added to it is tested using Mobilesim. One by one each trailer is added to the system to analyze its performance in terms of linear and angular separation errors. The controller gains are adjusted to control the formation errors of each trailer with the truck robot. As already mentioned, there are two methods for deploying this control algorithm for connecting multiple trailer robots. One way is to treat the setup as one truck robot to define a particular trajectory with n number of trailer robots. Each of the trailer robots have a separate virtual link and hinge point attached to the truck/leader robot. And every trailer derives its linear velocity and angular velocity from the truck robot and its hinge pose. None of the trailer is aware of the position of each of the other trailer robots except of the truck robot. This is similar to the leader-follower architecture in multi-robot system but with virtual link between them.

The second method is in which there is one truck robot that defines the trajectory and n number of trailer robots which can also act as truck robot for the immediate robot next to it and so on. So if n trailer robots are used then, there are n-1 trailer robots that can act as both truck and trailer leaving the last trailer with just trailer algorithm. It also has n hinges and each trailer robot is aware of the truck and one more of its trailer only. The algorithm is implemented in each trailer as two parallel threads so that each serves as server-client architecture integrated to it. The formation errors are also very less as there is multiple control points for a single trailer control and hence the whole framework is tightly coupled with each other. Awareness about the members in the multi robot system is only partially known to all the members of the system. The server thread treats the robot as a truck robot for the next robot and the client architecture treats it to be a trailer robot for the previous robot. This is the modified leader-follower architecture which overcomes the reliability problem on a single leader.

Both the methods are studied here by implementing in Mobilesim and tested for simulation studies under three different trajectories – s-arc, circle and random for truck-trailer and parallel movement. Both these methods are illustrated in figure 3.10 in Chapter 3.

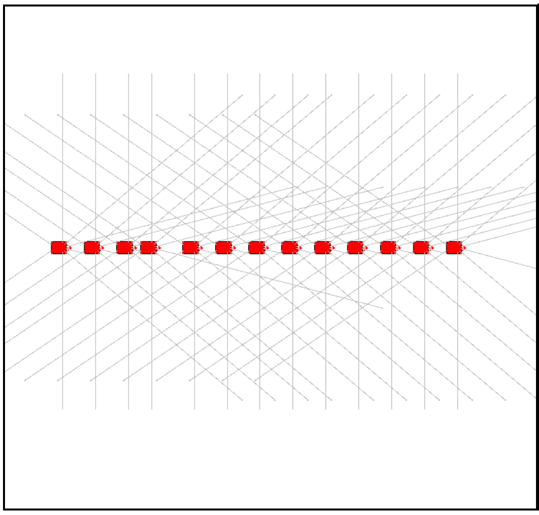
Truck Trailer: For the first method, it is found that the controller fails from the eighth trailer onwards or at a distance of 5600mm from the truck robot around it as such a long virtual link is not having a better control of the system. So the controller works for the maximum distance of 5600mm and hinge distance of 2800mm in truck-trailer formation for the second method. In terms of better formation errors, any number of trailers inside the radius of the truck robot at a distance of 2100mm yields the least linear and angular errors and hence the best tracking of the robot.

The first trailer robot is placed at a distance of 700mm with the hinge point at (-350, 0) from the truck robot. A total of 12 trailers are attached one after the other as shown in the table 4.6.

Table 4.6 Truck-Trailer - Circle – First Method for Multiple Trailers

Trailer Number	**Separation Distance (in mm)**	**Trailer Initial Pose**	**Hinge Point (X,Y)**
1	700	(-700,0,0)	(-350,0)
2	1400	(-1400,0,0)	(-1050,0)

3	2100	(-2100,0,0)	(-1750,0)
4	2800	(-2800,0,0)	(-2450,0)
5	3500	(-3500,0,0)	(-3150,0)
6	4200	(-4200,0,0)	(-3850,0)
7	4900	(-4900,0,0)	(-2450,0)
8	5600	(-5600,0,0)	(-3150,0)
9	6300	(-6300,0,0)	(-3850,0)
10	7000	(-7000,0,0)	(-4550,0)
11	7700	(-7700,0,0)	(-5250,0)
12	8400	(-8400,0,0)	(-5950,0)



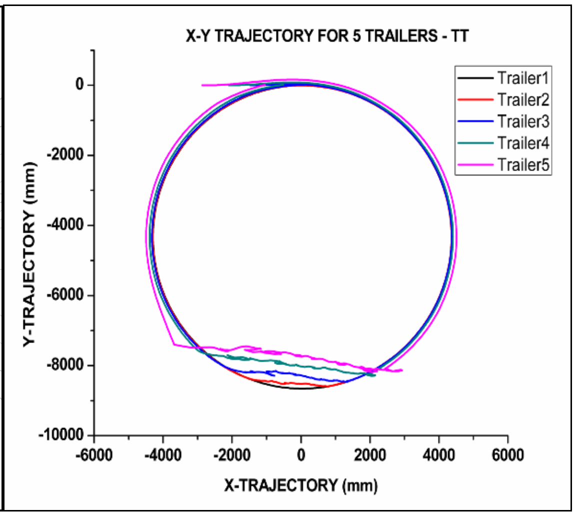
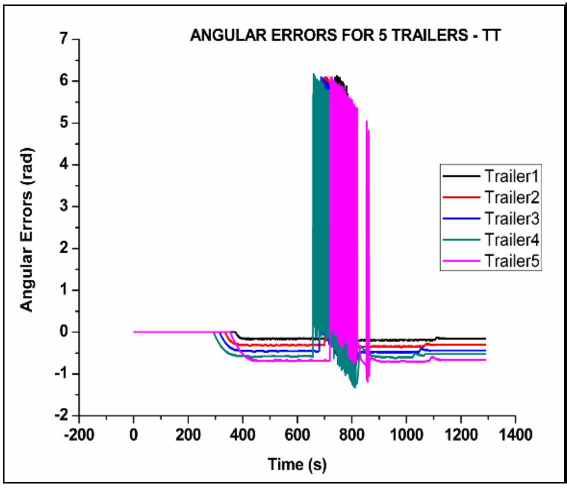


Fig.4.35 Simulation Screen shot of 12 trailers   Fig.4.36 Truck-trailer Circle XY Traj – 5 trailers



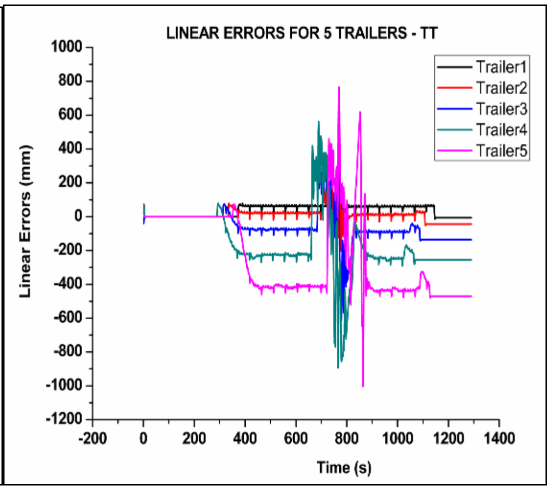


Fig.4.37 Truck-trailer Circle Angular Errors 5 trailers   Fig.4.38 Truck-trailer Circle Linear Errors – 5

The velocity set for the trailer is 75mm/s with 100mm arc radius and 0.75/s as angular velocity. As observed from the plots in figures 4.37 and 4.38, the linear and angular errors are less than 0.2% for the first two trailers. However, from the third trailer onwards, the tracking error increases till the 12th trailer. The plots show errors for first five trailers only for the clarity of the plots.

Parallel Movement: In case of truck trailer robots, it is clearly understood that all the trailer robots need to be aligned at the back side of the truck robot. But for truck-trailer robots to be in parallel, the trailer robots can be made parallel to the truck robot either in the positive y direction or in the negative direction. So, a set of three types of arrangement of the trailer robots are made. 1. All trailer robots to the left of the truck robot (in the positive y-direction) 2. All trailer robots to the right of the truck robot (in the negative y-direction) and 3. Trailer robots placed on both the sides of the truck robot.

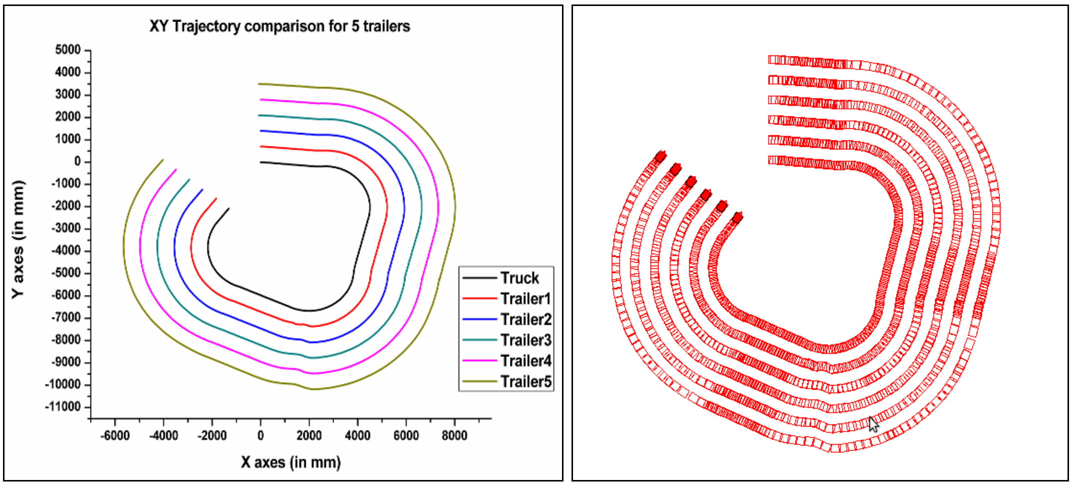


Fig.4.39 Random track-Parallel (5 trailers) – XY – Trajectory and Screenshot of Mobilesim

Figure 4.39 shows the trajectory of the robots with 5 trailers when run in a random pre-defined path. Figure 4.40 shows the different configurations run and for different placement of the trailer robots around the truck robot. The linear and angular errors are plotted for all the different configurations. It was found out that for trailer robots with negative coordinates where the switch over of 180° to -180° is frequent, the trailer robots went bizarre in unstable areas. Detailed study on the unstable areas will be done in the next chapter. So, it was found that at a distance of -2100mm (6^{th}, 7^{th} and 8^{th} trailers) on the negative coordinates, the controller failed. But, the trailer robots ran successfully for all 12 trailer robots in the positive direction of the truck robot. It was also found out that when the trailer and truck are very close to each other, the movement of the truck robot causes a negative velocity for the trailer robot making it to move backwards but still tracking the truck robot by satisfying the controller equations successfully. In parallel formation, the first method yielded good results when the initial velocities were positive and the second method ran successfully under all conditions. This is proved clearly by the plots obtained in figure 4.41 and 4.42

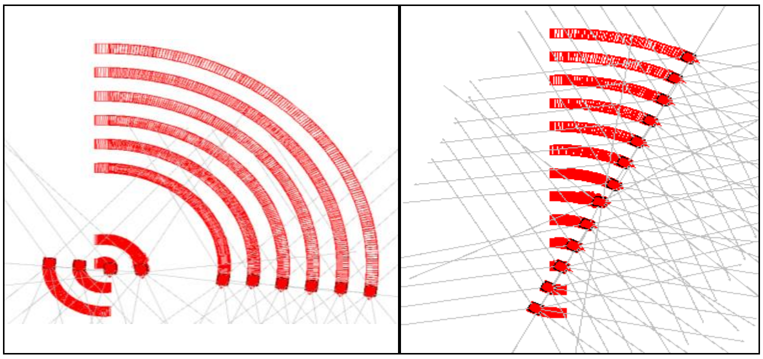


Fig.4.40 Parallel Mode – Circle Trajectory-Different screenshots for various trailers place

4.4.8 Overall System Performance

The simulation studies have yielded the following conclusions. The controller provides optimal performance, when it's gains are set to: K_1=1.1, K_2=1.1 and K_3=1.1. Also, the optimum virtual hinge point at a distance of 350mm from the centre of the axis of leader/truck robot yields better performance. Simulation studies yielded very good results, when the linear velocity of the truck is set to 50mm/s. Though the linear and angular separation errors varied with the linear velocity of the truck from 8.014mm to 101.35mm and ±0.0015 rad to ±6 rad respectively, the average separation errors were 12.2mm and ±0.0028 rad respectively. This also shows that the presence of two control gains for controlling the angular velocity of the trailer which yields better results compared with the literature (Kuppan et al. 2008).

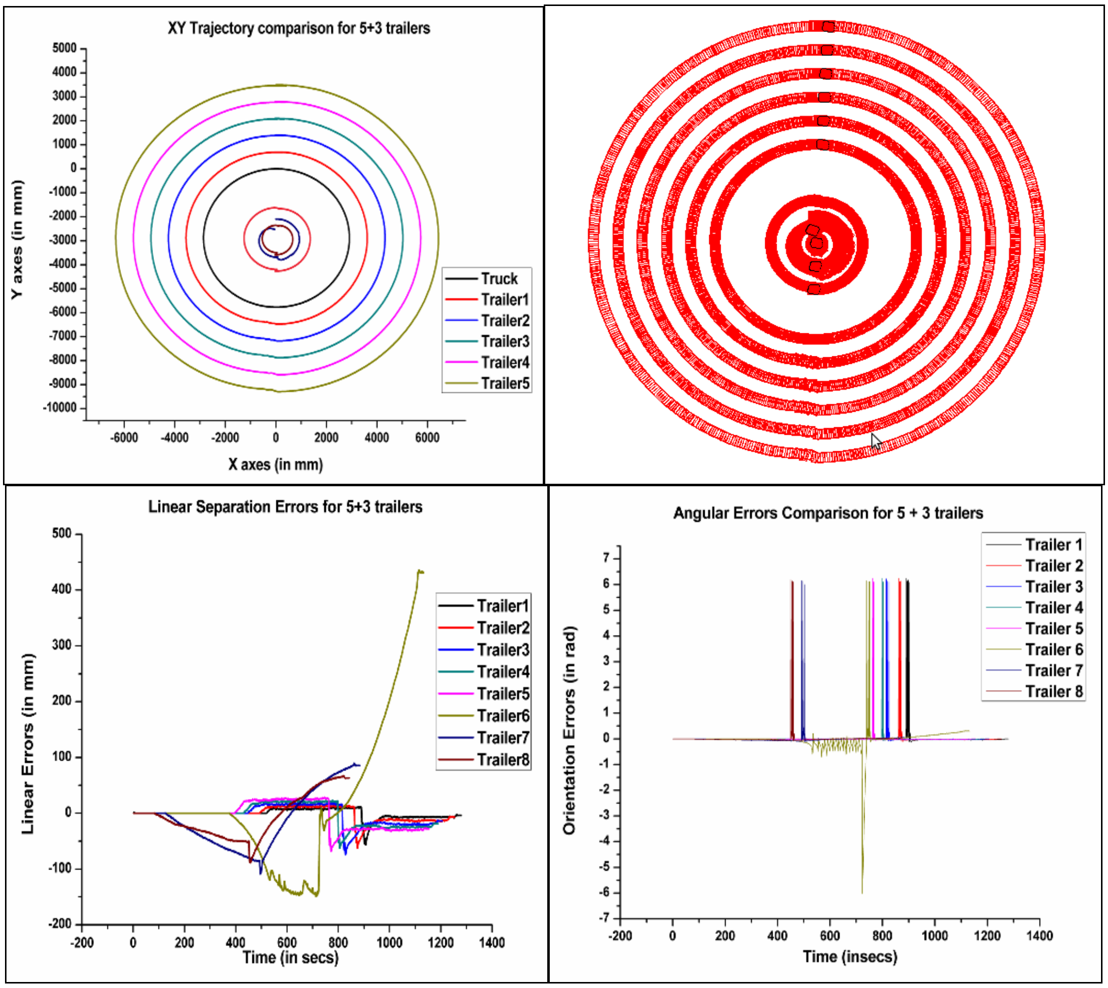


Fig.4.41 Parallel Movement of multiple trailers – 5 + 3 trailers

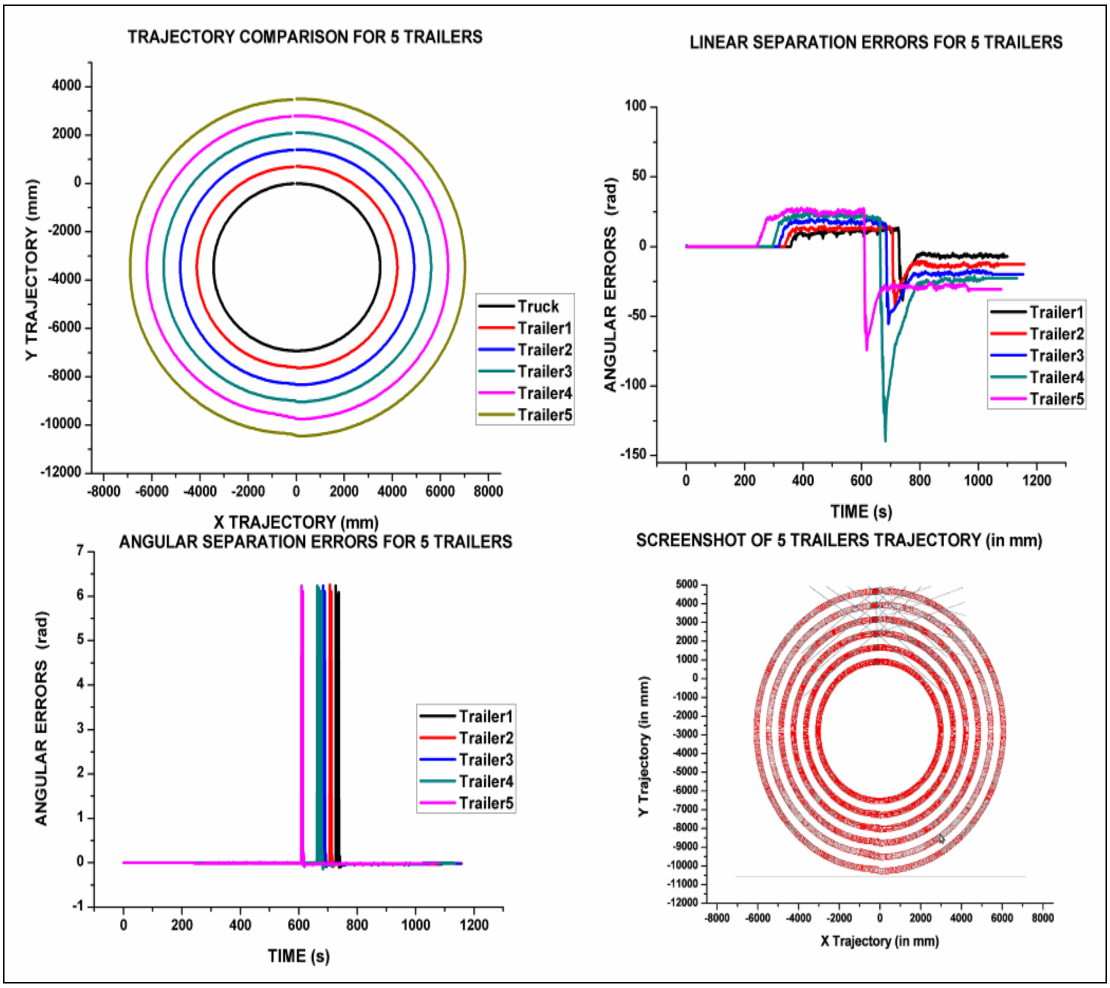


Fig.4.42 Parallel Movement of multiple trailers – 5 trailers

4.5 EXPERIMENTAL STUDIES

4.5.1 Truck-Trailer formation using Virtual link

The Truck Robot is considered as server/leader and performs the trajectory planning. and Trailer Robot as the client/follower which performs trajectory planning and formation respectively. As mentioned, each robot itself has server-client architecture in it, where the server is the robot and client is the control program running on the robot. The truck robot being the server can connect to multiple clients and the trailer connects to the truck via wireless network (TCP/IP communication protocol).

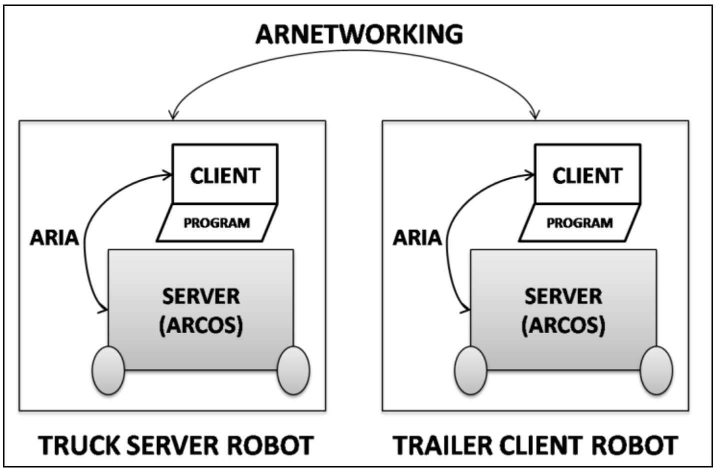


Figure 4.46 Multi Robot System Architecture

As mentioned in chapter 3, the truck and trailer robot has the same kinematic configuration and are identical in all aspects. The truck robot which is the server in the team commands the control signal and data packet request from its client to the server of the other robot with a 5Hz frequency synchronously.

Figure 4.47 Experimental Setup - Amigobots

ArNetworking and ARIA library helps in creating a firmware environment in the client and the server. All the five behaviours in the control algorithm are created as call back functors and are executed in parallel. Any other remote hosts can also be brought in to this setup to connect with the truck and trailer to receive all the information packets needed like wheel velocities, coordinate poses and logs it into the database. The trailer is made to synchronize with the truck such that the truck initiates the control algorithm for the trailer. This synchronous interaction will create a proper flow of the control between truck and trailer. Hence, the trailer will be able to maintain the desired position and the orientation required to be in the formation with the truck.

4.5.2 Simulation and Experimentation

This section lists out the main differences and similarities in implementation procedures for simulation and experimentation studies. In case of simulation, each robot in the simulator arena will have its odometry position to be (0,0,0) and are located in a default position specified by the simulator. Hence, initial pose of the trailer need to be set up so that each of the robots can identify each other.

On a real robot, ARIA may be communicating with different devices over separate communications channels. For example, ARIA communicates with the robot microcontroller over one RS-232 serial port, with a SICK LMS-200 laser rangefinder over a different RS-232 serial port, simultaneously using two different threads (one contained in ArRobot, the other in ArLMS200 which is a subclass of ArThreadedRangeDevice). Other additional devices may use additional parallel communications channels such as serial, USB and ethernet connections. However, ARIA communicates with MobileSim over only one TCP socket connection. Therefore all communications with both the simulated robot and simulated devices must occur on this one channel, the same as the robot connection ARIA's Connector classes detect MobileSim and use the appropriate code to handle data transfer with devices via the robot connection i.e. when structuring programs; for example, laser data will not be received unless the ArRobot connection is active and running.

4.5.3 Multiple robots communication

In simulator, it was simple to create multiple robots and establish connection between them due to Mobilesim, which is a dedicated GUI for multiple robots. However, in case of implementing the algorithm in real time, the following setup has to be done. Both the robots need to be communicated by a wireless TCP/IP network. Figure shows the communication setup of both Pioneer p3dx and Amigobot separately with the Wireless Ethernet access point, where each robot either has an onboard computer with a wireless Ethernet (like Amigobot) or a laptop connected with the robot that has wireless adapter (like the Pioneer). In both the cases, a small LAN is created with all the robots and the wireless access point hence is useful in setting up a closed network that communicates with every other node. For pioneer, the client programs for both the server robot and trailer robot are run on the respective laptops connected to the robots whereas for Amigobot, another computer which is in the LAN runs the program for both the robots.

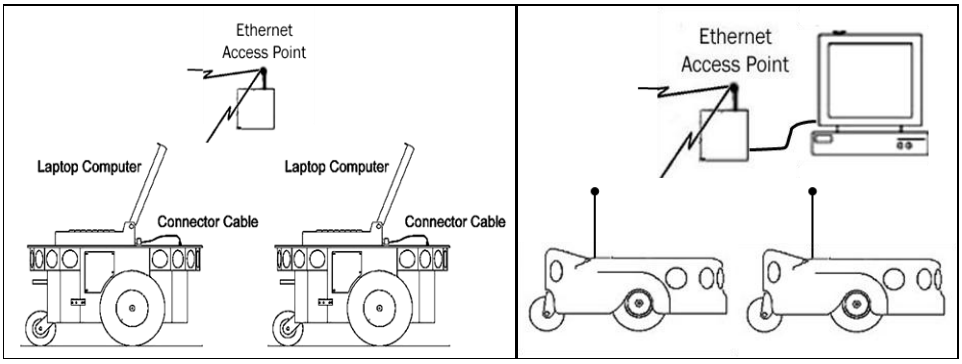


Figure 4.48 Intercommunication in Pioneer robots and Amigobots

4.5.4 Experimental Results

Simulation results are run based on the mathematical model of the robot defined by the simulator. However, when run on real time, there are so many other factors that affect the system performance like the dynamics of the system, traction force between the wheel and the floor, etc. Hence, the same set of tests that were run in the simulator is run in real time using the available research platforms (Amigobots). Each behavior is assessed individually and as well as a complete system. Finally this section compares the results obtained by simulation and experiments.

4.5.5 Performance of tracking controller

The tracking controller is tested using circular, straight-line, s-arc trajectories under different velocities as done for simulations. A set of four program runs are carried out for each trajectory by varying the velocities. The controller gains K_1, K_2, K_3 are taken to be 1, 1, and 1. Table 4.7 illustrates the different configurations of the experiment tests run for evaluating the performance of the controller.

Table 4.7 Formation Control – Experiment test runs

Trajectory	Linear separation $l_d = 600$mm & Angular separation $\varphi_d = 0^\circ$ Controller Gains K_1=1, K_2=1, K_3=1 V_L = Truck linear velocity in mm/s ω_L = Truck angular velocity in rad/s Arc radius for truck = 25mm (for circular, s-arc, deep s-arc)			
	V_L=25	V_L=30	V_L=50	V_L=100
Straight-line	$\omega_L = 0$	$\omega_L = 0$	$\omega_L = 0$	$\omega_L = 0$
Circle	$\omega_L = 1$	$\omega_L = 1.2$	$\omega_L = 2$	$\omega_L = 4$

The performance of the controller is tested by plotting varies graphs. Figure 4.46 shows linear and angular separation errors for a straight line trajectory of the truck for different velocities. X-Y Trajectory is not plotted as it just produces a straight line and the details of the figure are not visible. The average linear and angular errors were found to be 8.25mm and 0.001rad for a truck velocity of 25mm/s, 11.43mm and 0.00256rad for 30mm/s, 23.4mm and 0.00312rad for 50mm/s and 82.11mm and 0.004rad for 100mm/s. The results for s-arc with truck velocity of 165mm/s and circular trajectory with four different truck velocities (25, 30, and 50,100) also yields similar results. (figure 4.50)

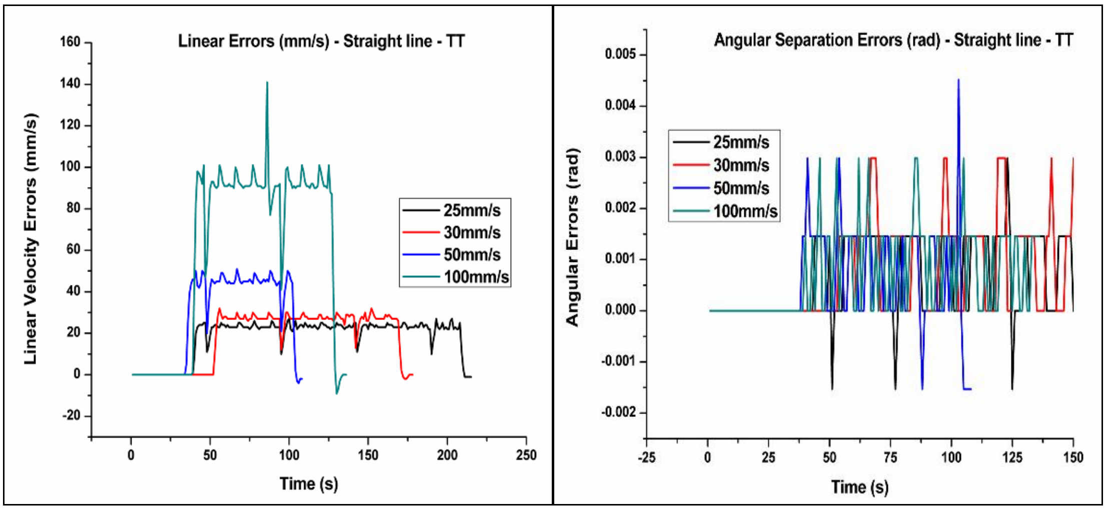


Figure 4.46 Formation parameters error – TT – Straight line

Controller Gain Optimization

Controller gains can be optimized using theoretical or experimental way depending upon the controller. Since the proposed controller can be easily implemented in experiments, controller gain tuning is done by this method. A set of 20 simulations each are conducted to fine tune the values of the controller gains and it was found out that under experimental conditions, the controller gains should be K_1=1.25, K_2=1.25, K_3=1.3. The results are also verified by Ziegler Nichols controller tuning method.

4.5.6 Presence of Virtual-Link

In case of simulation studies, 350mm is the optimum control hinge point, but with experiments it was found out that the optimum control point should be 600mm in order to make the robot rotate freely on its own axis. Hence, all the experiments henceforth are performed by taking the virtual hinge point to be 600mm.

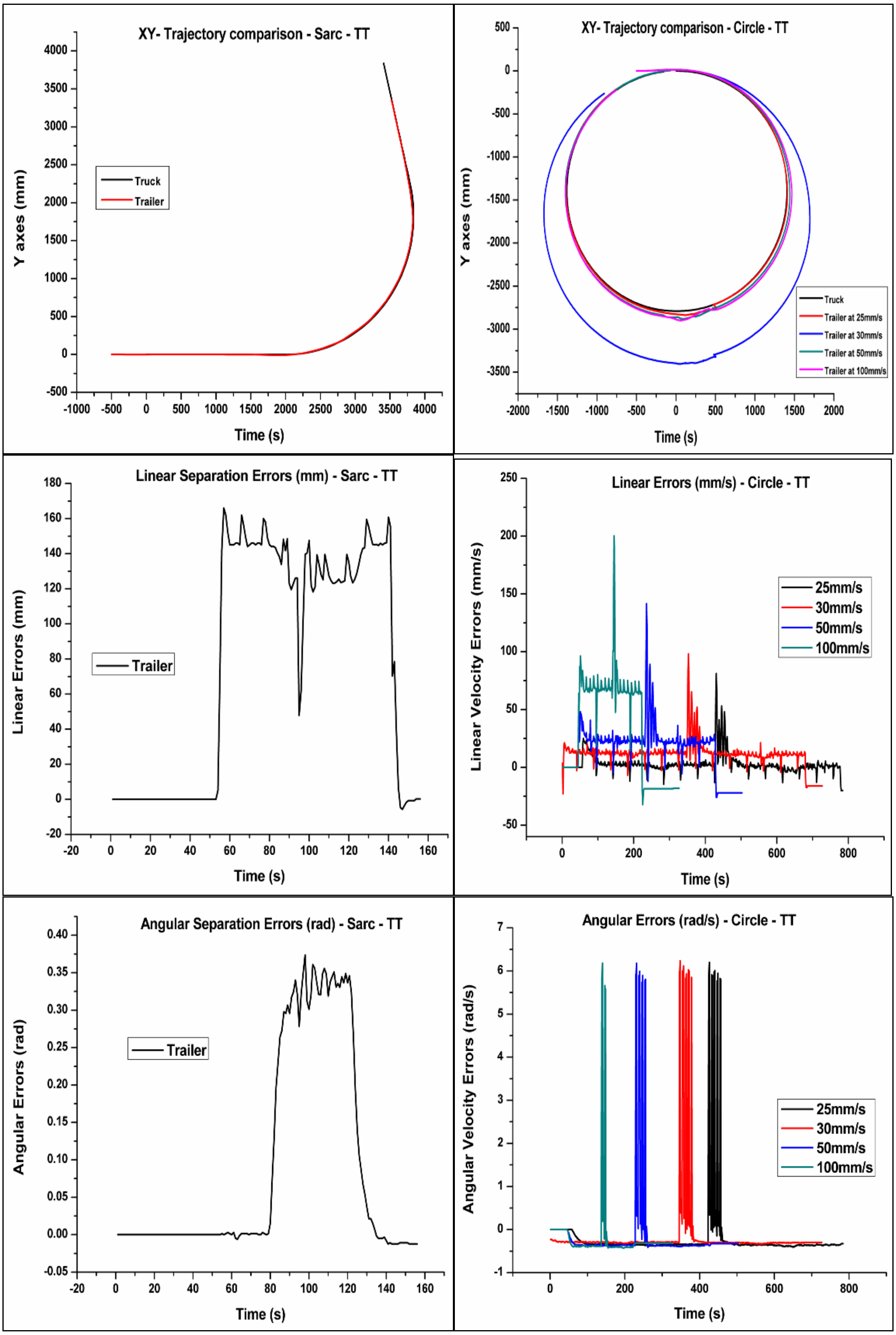


Figure 4.47 a) S-arc for 165mm/s b) Circular trajectory for different velocities

4.5.7 Parallel Movement

Experiments are conducted to check the feasibility of the controller for parallel formation shapes. The proposed controller is tested for the truck-trailer to move in parallel by changing the angular separation between the robots φ_d to 270° and the linear separation of the two robots l_d to be 600mm as shown in table 4.8. The results are discussed in detail in this section and are similar to truck-trailer formation between the robots. Figure 4.51 illustrates the linear and angular errors for straight line for different velocities whereas figure 4.52 explains linear and angular errors for s-arc for 50mm/s and circular trajectories for different set of velocities as in simulation.

Table 4.8 Formation Control - Parallel Movement - Experimental test runs

Trajectory	Linear separation $l_d = 600$mm & Angular separation $\varphi_d = 270°$ Controller Gains K_1=1.25, K_2=1.25, K_3=1.25 V_L = Truck linear velocity in mm/s ω_L = Truck angular velocity in rad/s Arc radius for truck = 25mm (for circular, s-arc, deep s-arc)			
	V_L=25	V_L=30	V_L=50	V_L=100
Straight-line	$\omega_L = 0$	$\omega_L = 0$	$\omega_L = 0$	$\omega_L = 0$
Circle	$\omega_L = 1$	$\omega_L = 1.2$	$\omega_L = 2$	$\omega_L = 4$

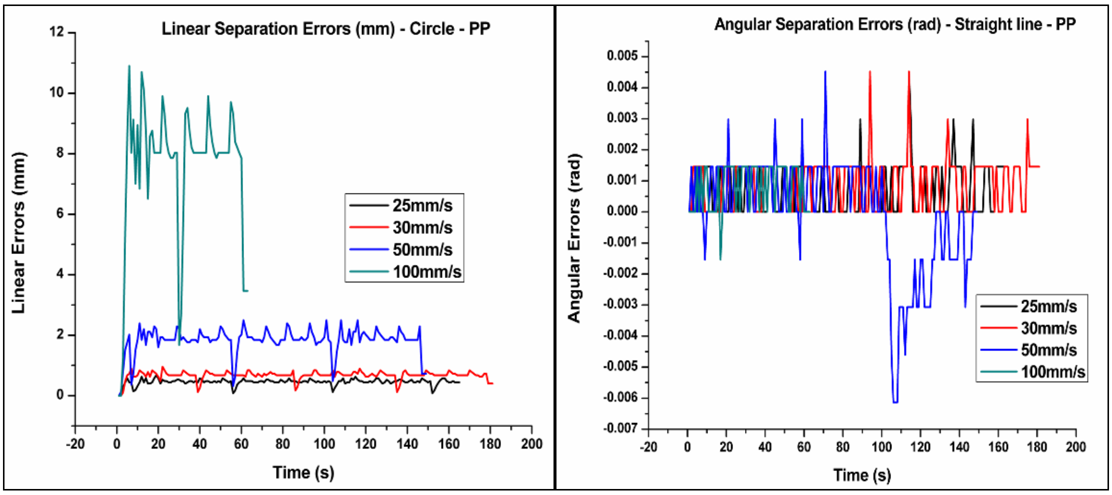


Figure 4.48 Linear and Angular Errors for Straight line - PP

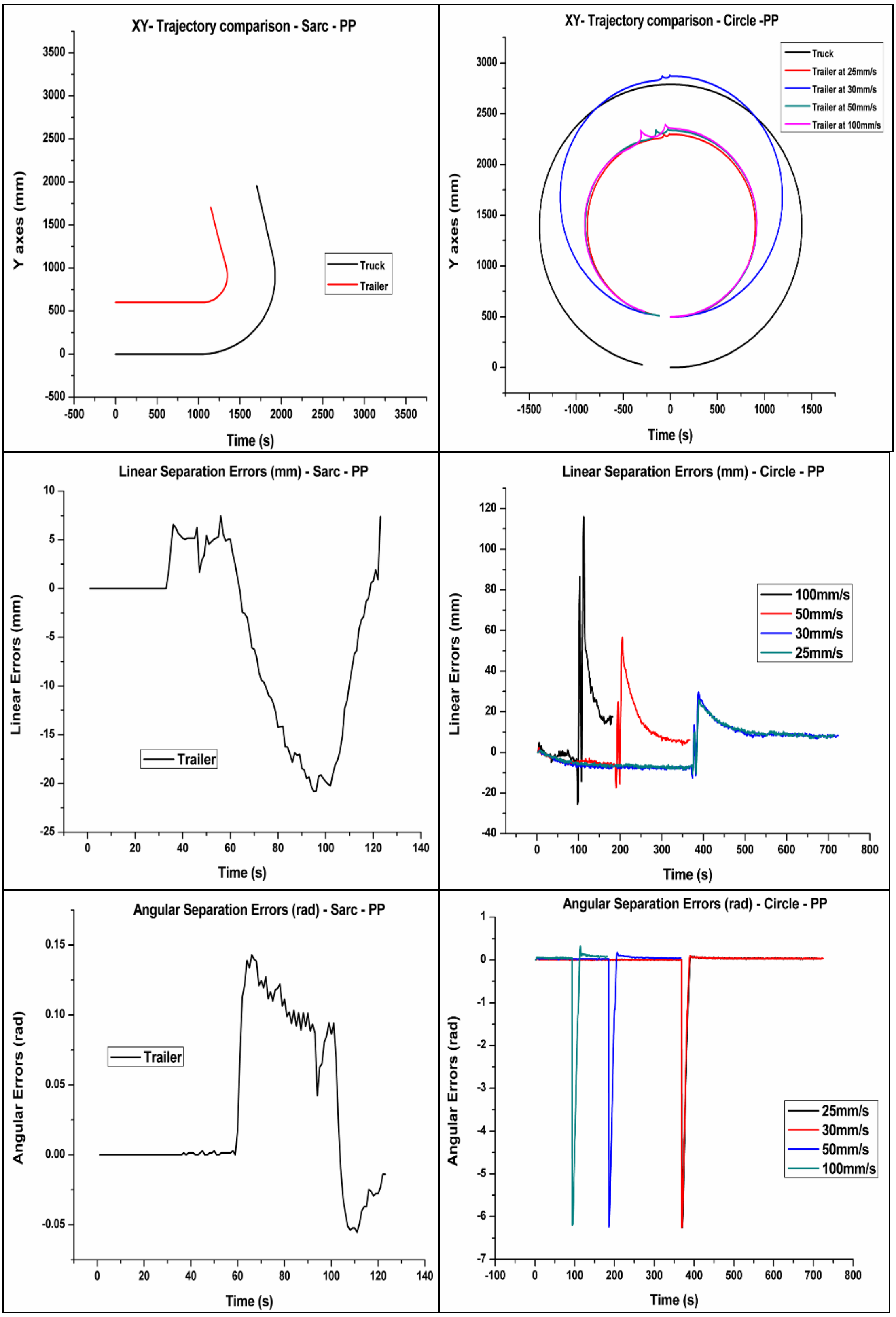
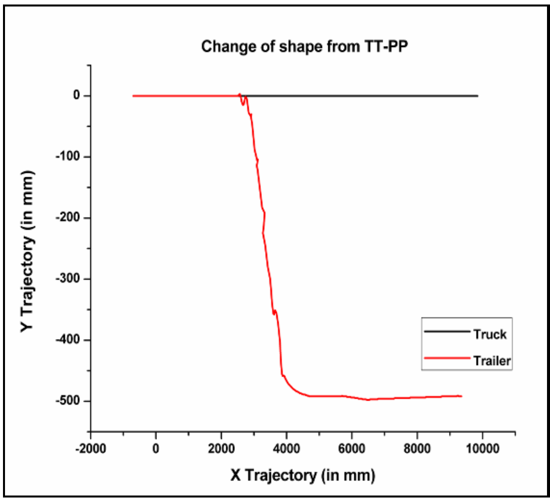


Figure 4.49 a) S-arc for 165mm/s b) Circular trajectory for different velocities

4.5.8 Other Behaviours:

As mentioned already, dynamic obstacle avoidance behaviour can only be measured using experiments, the following experiment was conducted which switches over from truck-trailer formation to parallel mode formation due to the presence of obstacles. Figure 4.53 gives the trajectory of the both the robots whereas figure 4.54 shows the variation in the orientation of the trailer with respect to the truck robot.



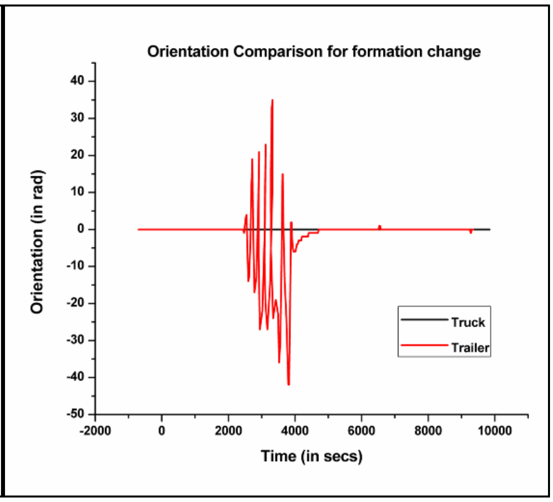


Fig 4.50 Trajectory comparison with obstacles   Fig. 4.51 Orientation comparison with obstacles

Fig 4.52 Experimental Setup

4.5.9 Multiple Trailers

In order to carry out this experiment, two trailer robots were linked with the truck robot. Nevertheless, the experiments also yielded the same results as the simulation studies, but at slower velocities like 50mm/s to 100mm/s. The results obtained for the first and second trailer were 0.0015rad and 0.002rad of angular errors and 7.91mm and 8.02mm of linear errors. However, the effect of more than two trailers in the system is not studied due to non-availability of extra research platforms.

4.5.10 Overall System Performance

The controller gains are optimized to K_1=1.25, K_2=1.25, K_3=1.3 and the optimum virtual hinge point at a distance of 600mm from the centre of the axis of leader/truck robot. The linear and angular separation error varies with change in truck velocity. Simulation experiments yielded very good results when the linear velocity of the truck is set to 25mm/s. In the experimental investigation, though the linear and angular separation errors varied from 6.83mm to 200.85mm and ±0.0015 rad to ±6 rad respectively, the average separation errors were 82.11mm and ±0.035rad respectively. This also shows that the presence of two control gains for angular control yields better results, compared to that of the available in the literature.

CHAPTER 5

CONTROLLER ANALYSIS &VALIDATION

5.1. INTRODUCTION

This chapter gives a detailed study on the analysis of the results obtained in the last section. The main objectives which define the effectiveness of the controller like stability, tracking and robustness are investigated using simulations and verified through experiments. Further, this chapter also details on the various other features that are added to the present control architecture to improve the versatility of the system like localization and different formation shapes by multiple robots. Finally, the modified complete algorithm is presented.

5.2. STABILITY ANALYSIS

From the last chapter, it is evident from the simulation and experimental results that the formation controller of virtual truck-trailer system is able to track the truck trajectory with errors less than ±1.0%. However, under some conditions, the trailer robot was found to lose the tracking ability. This can be seen in circular trajectories of the truck. Few oscillations were observed in the path of the trailer, when the truck had an orientation shift from 180° to -180°. By adjusting the controller gains, the trailer oscillations are reduced to a minimal level, but could not be eliminated completely. Hence, it was evident that there are some unstable areas and singularities, where the controller loses its tracking ability.

With reference to figure 5.1, it is found that when the trailer is placed in any of these unstable areas (red), it drifts away from the truck and the formation is lost. However, if the trailer is placed in any of the stable areas (blue), it can successfully track down and attach with the truck on its own. In order to find out these areas, simulations were conducted with gain values ranging from 0 to 5. The results were plotted for finding out the grey areas of the controller with respect to X-Y trajectories. This shows the importance of tuning the system gain to obtain the optimal performance and stability. It is also observed that the area of instability is greater near to the centre rather than the outer edge. This is proved in the last chapter on multiple trailers where three trailers are run in this area and results in instability.

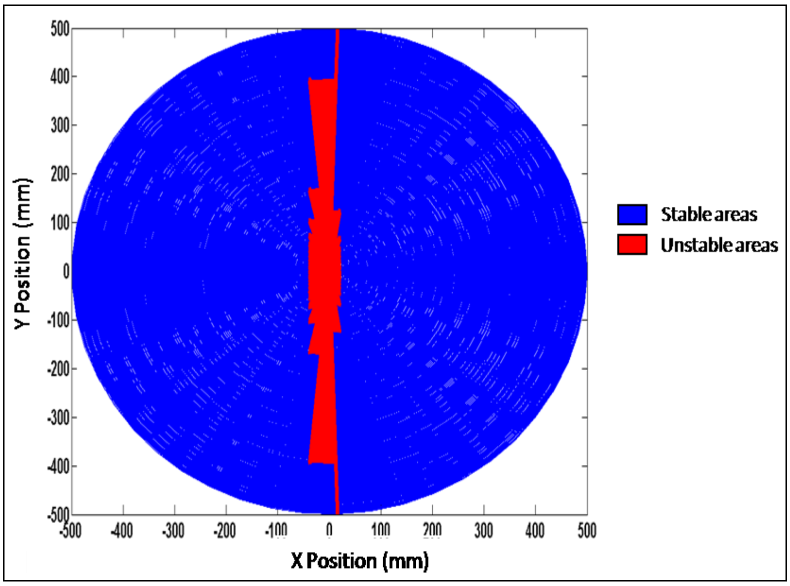
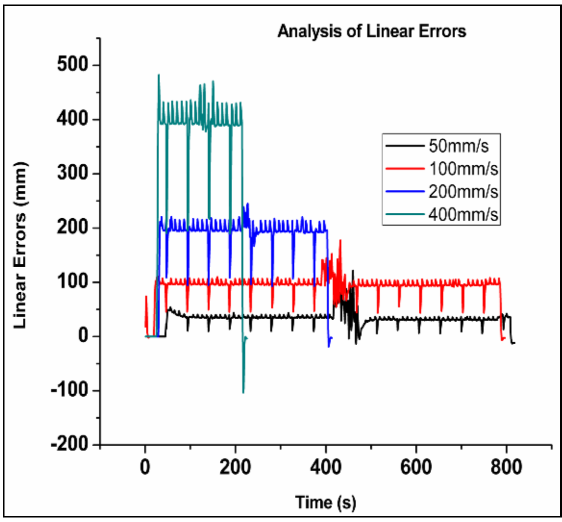


Fig 5.1 Stable and Unstable areas of the Controller

Moreover, another set of simulations are run for different velocities of the truck and the corresponding tracking errors are calculated at every instant (similar to the first set of experiments conducted for circular trajectory). The plot of this data shows the convergence of the tracking errors asymptotically to zero.

In case of the circular trajectories, the unstable areas are the shift from 180° to -180°. This singularity point is due to the complete change in the quadrant of the computed angle. In Matlab, python, c or c++, or any programming languages, the arctangent with two arguments (atan2) is used to implement the controller equation 3.30 such that the appropriate quadrant of the computed angle is obtained. Even though the magnitude of the errors is more in these unstable areas, the tracking errors ultimately converge to zero in all other areas. These plots are analyzed in terms of linear errors in figure 5.2 and angular errors in figure 5.3. Finally, the controller guarantees the stability of the controller with few singularity areas.



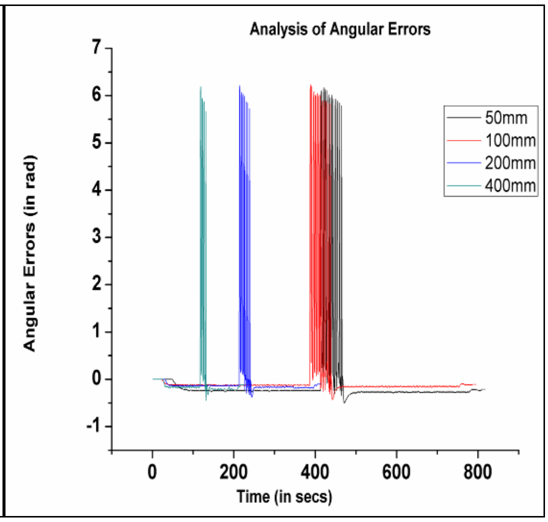


Fig 5.2 Linear Errors converging to zero

Fig 5.3 Angular Errors converging to zero

5.3. ROBUSTNESS TEST

Robustness of the system is defined as the performance changes of a control system with changing system parameters. The tracking controller designed by Kuppan et al. (2008) had many parameters of the robot that should be known before validating the controller. These parameters are the physical dimensions of the robot. Hence, the performance of the controller varied with the type of robot used. This means that the controller works only with the particular robot, for which it is designed and the performance varies for other type of robots. Also, it is very important for the controller to be behave in a robust manner. Though, there are many standard theoretical methods to check robust control like adaptive control, parameter estimation, $H_{infinity}$, etc, in our case the controller equations are implemented in real time and hence the robustness is tested with the research platform under experimental conditions.

By looking at the controller equations mentioned in 3.24 to 3.25, it is evident that these equations contain any of the robot's physical parameters and fully dependent on parameters like position, velocity, error coordinates, system output and control gains. It is evident that this controller is more robust compared to other controllers in the literature work. However, in order to explain this property experimentally, the control algorithm is tested for three set of experimental configurations: 1) Pioneer P3DX as a truck and Amigobot as the trailer, 2) Two Pioneer P3DX robots as truck and trailer and 3) Amigobot as truck and Pioneer P3DX as the trailer. It may also noted that the Pioneer P3DX is almost double the size of the Amigobots and all the physical and technical specifications of both the robots are different.

Since the 'Mobilesim' simulator does not have the option to run two different types of robots simultaneously (namely Pioneer P3DX and Amigobot), only real time experiments were conducted. Simulation is done only for the second set of experiments, which involves use of two Pioneer P3DX robots. Fig 5.4 and fig 5.5 shows the simulator screenshot and the experimental setup.

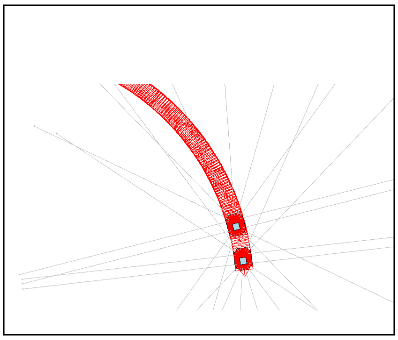


Fig 5.4 Two Pioneers in MobileSim Simulator   Fig 5.5 Pioneer - P3DX and Amigobots

During the experimental investigations, the control algorithm is run with a linear velocity of a 100mm/sec for circular trajectory and 150mm/sec for s-arc trajectories. Also, the controller gains were set at their optimal values (K_1, K_2, K_3=1.1).

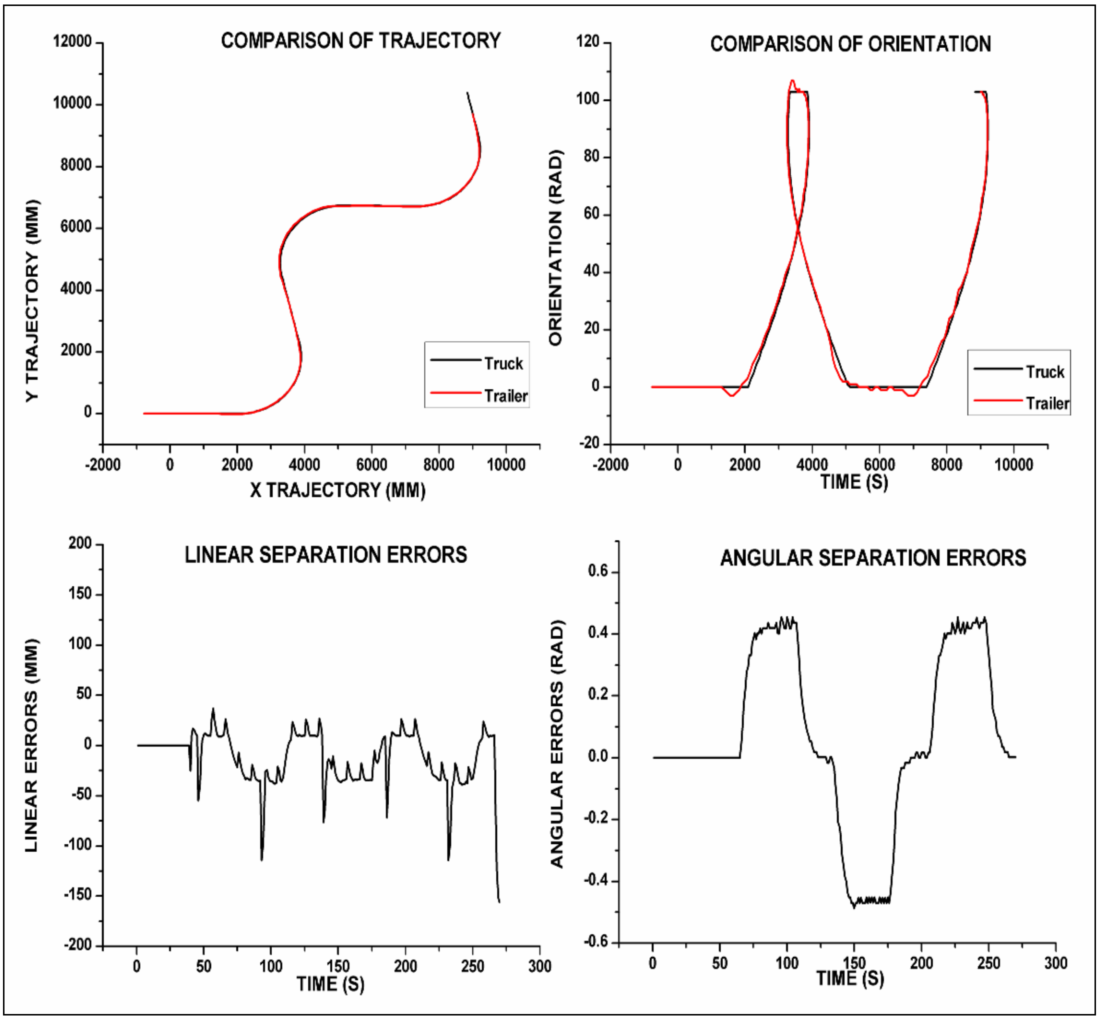


Fig 5.6 Simulation Results for S-arc Trajectory for Pioneer-Amigobot

Figure 5.5 illustrates the results obtained and it can be noted that the linear separation errors vary between 0 - 125mm during straight line motion and 0 - 155mm during the s-curve motion. As seen in chapter-4 on the experiments of formation controller, in these experiments also, the linear separation error is more and varies in the range of 30mm-150mm. However, angular errors are considerably less and it varies from 0.4 rad to -0.4 rad only.

When a different trajectory is given as an input, in this case, a circular trajectory, the maximum linear error found to be 101.25mm. Similarly, the angular error varies within 0.037rad, with an exception of 5.035rad at the instability areas. This is reflected in the orientation comparison plots between them.

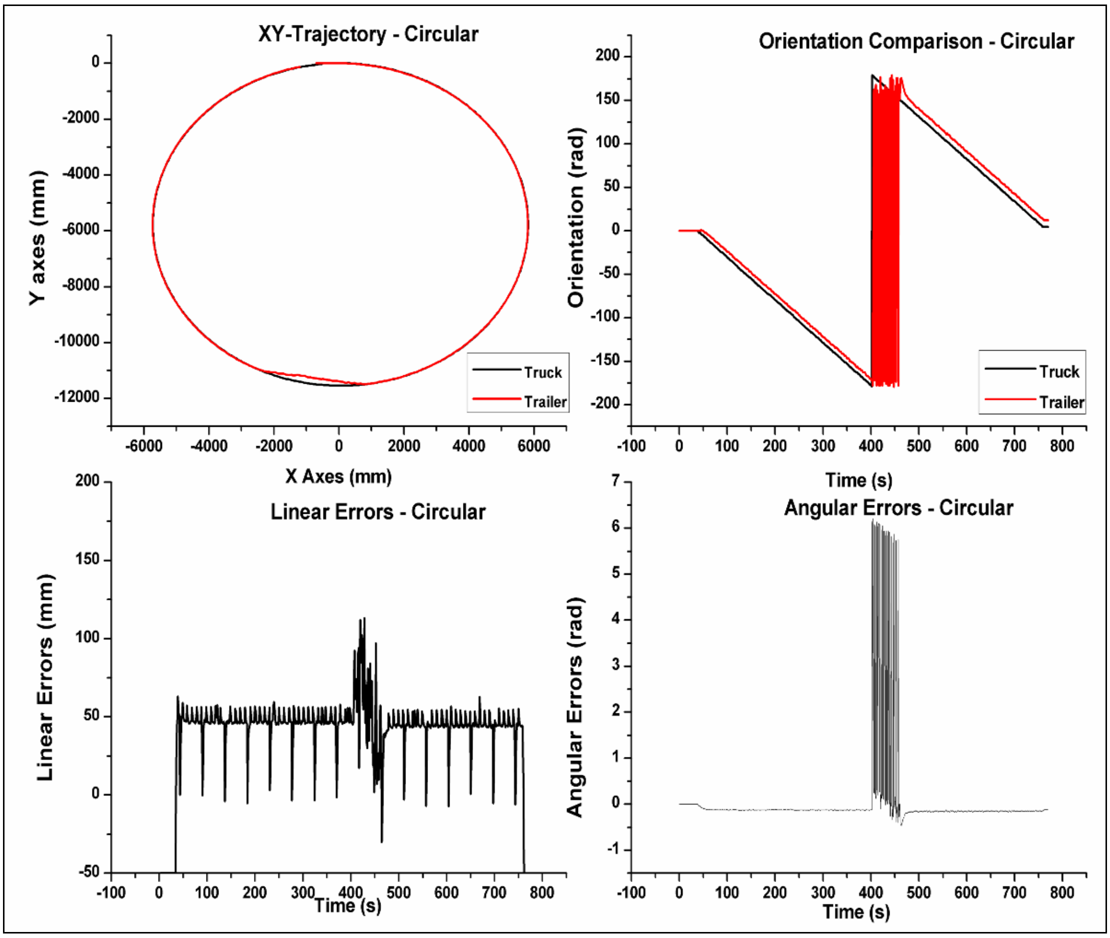


Fig 5.7 Simulation Results for Circular Trajectory for Pioneer-Amigobot

The controller was tested in all three configurations mentioned earlier, with the same trajectory as inputs and yields a comparables results. This proves that the controller is robust and can be implemented for any type of commercially available differential drive robots.

5.4. DISTURBANCE REJECTION

Many controllers are prone to noise and disturbance and hence, the proposed controller is also tested for such rigidity. Another set of experiments and simulations are run to check for disturbance rejection capability of the controller. This is done by abruptly changing the velocity of the truck in a rapid manner. This is achieved by providing two different trajectories to the truck – Teleoperation and keeping a pre-determined path with alternating velocities. The results of the complete plot are shown in the figures below.

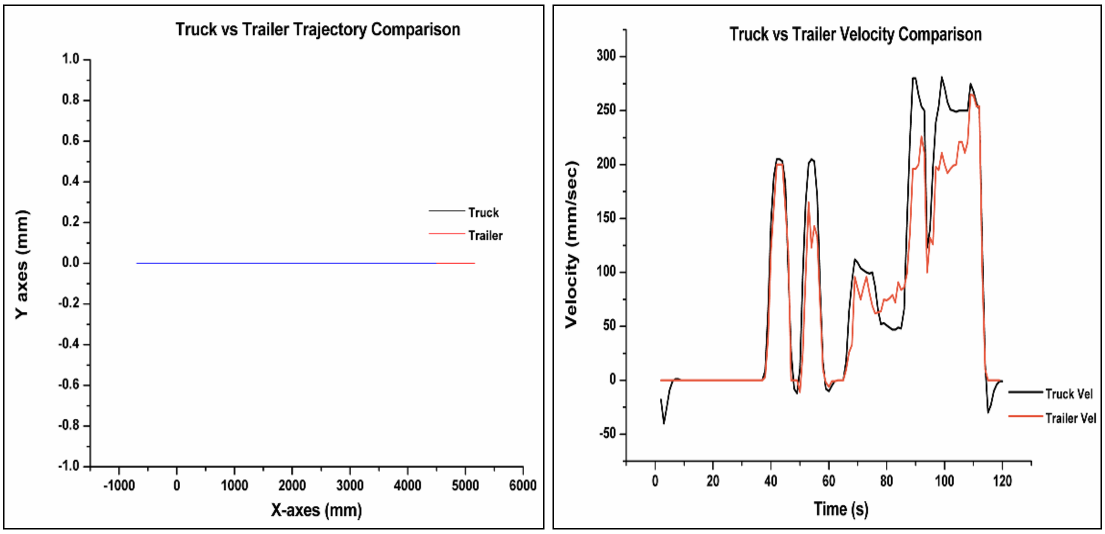


Fig 5.8 a.) X-Y Trajectory for Straight line b.) Velocity comparison with varying velocities

The reason for the delay of the response of the trailer robot is because the update frequency between the robots is 100ms and the controller computes the required follower/trailer velocity every 1000ms. Hence, the trailer robot takes 10ms to respond to the truck robot's input which causes the delay when the velocities or any other disturbances are stimulated to the existing system.

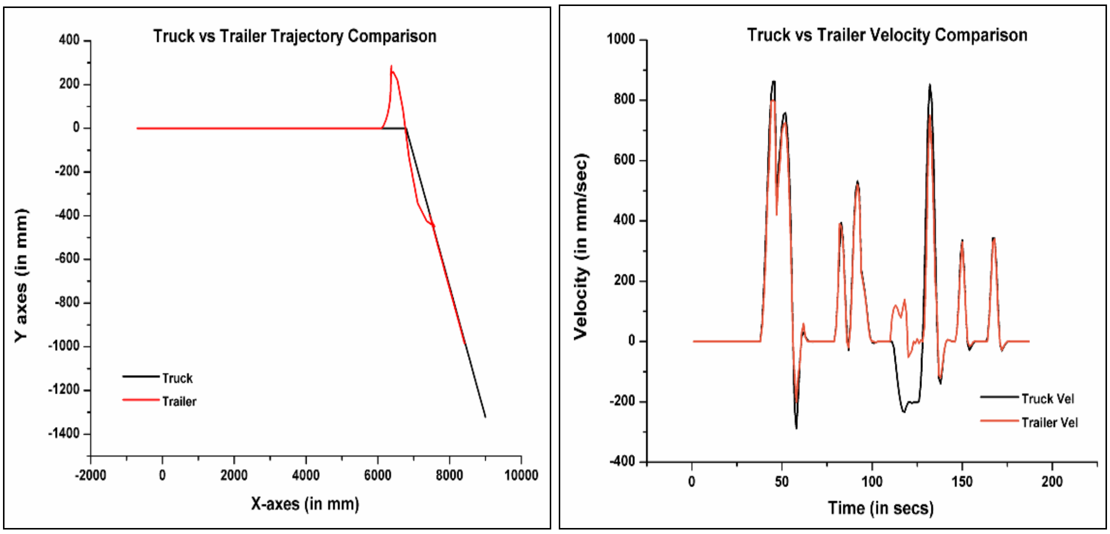


Fig 5.9 a.) X-Y Trajectory for Tele-operation b.) Velocity Comparison with varying velocities

5.5. LOCALIZATION OF THE SYSTEM

Localization feature is used in our present system for experimental conditions and can be useful in two different ways. Firstly, when the robots are switched on, the initial position of both the robots are read as (0,0,0) by the odometers during experiments. However, when the truck and trailer are placed in its initial position, the trailer should be at a distance equal to the virtual link. Hence, the odometer of the trailer should be reset to the required pose. This is done by the control algorithm manually. Localization of the complete system makes the trailer aware of its position and hence there is no need of setting up the odometer positions. Secondly, when a predefined trajectory is not given for the trailer robot or the truck does path planning of its trajectory or when there is a condition when the trailer gets lost during obstacle avoidance, this feature comes handy. The localization class helps both the robots to localize their position and have knowledge of each other positions.

SONARNL is used here for localization as they work on SONARs and are readily available with all the platforms. Sonar Localization localizes robot by using sonars and robot odometry information. Along with the SONAR sensor data and the prepared environment map, the ‘Mapper3Basic’ finds the most probable position of the robot within the map, and resets robot's pose to reflect this position within the map's coordinate system.

Table 5.1 Localization library of ARIA

Name	Sensors Used	Performance	Robots
ARNL	Laser Rangefinder	Accurate & precise	Robots with Laser
SONARNL	Sonar	Approximate ≈ 2 m	All Robots
MOGS	GPS receiver	Less than 5 m	P3AT and Seekur

5.5.1 Arnl – Localization Library of ARIA

ARIA has another set of package used for localization and navigation. The main purpose of this package is to track the robot’s position using any of the localization techniques (depends on the sensor available with the robot). It also helps in the navigation of the robot to the required goal or destination. ARNL libraries organize both the activities; in localization, the robot’s pose are corrected automatically and in navigation, the robot is navigated to the required goal given by the control algorithm or any other remote control client like ‘MobileEyes’. There are three different localization methods with ARNL depending upon the sensor & localization algorithm and each method has different libraries with the ARNL libraries.

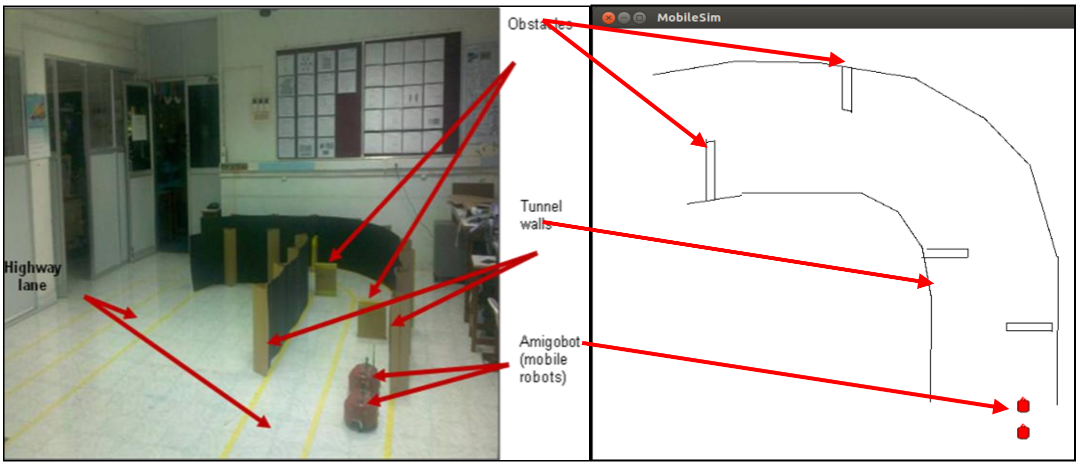


Fig 5.10 MobileSim Simulator Fig 5.11 Physical Setup of the map

5.5.2 Localization

Localization is achieved using Monte-Carlo Localization algorithm to place the robot within a given line map. Though, there are different localization techniques, all the methods required a known map of the environment with relevant environmental information. These maps can be generated by driving the robot in the given arena or developing a map with Mapper software, but to the scale. The data collected is converted into a map using the MobileRobot's MobilePlanner software with points and lines as the obstacles. SONARNL uses lines in the map, but MOGS uses the GPS coordinates with the origin point in terms of latitude longitude and altitude (LLA).

***Experiments**:*

Sonar Localization (using the built-in Sonar sensors of the P3DX) is not as accurate as compared to Laser sensor based location. Also, in order to localize the robots with better accuracy, more number of sonar readings and walls (reference marks) are needed for better accuracy. A physical experimental setup is made, so that the truck-trailer robots will be able to localize and navigate through the experimental setup. In this physical setup, walls are made closer to the robots and hence there is better localization of the robot.

Simulations:

A map is created using ‘Mapper3Basic’ which is an exact replication of the real setup (i.e. map is drawn to scale) and hence the robots can imitate the real environment and can localize itself in the map.

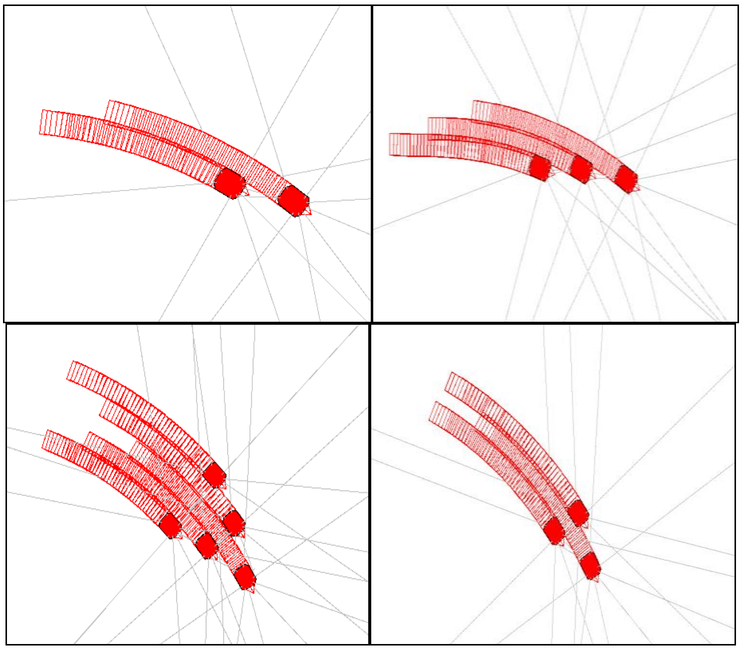


Fig 5.12 Screenshot of the Simulator for different formation shapes

5.6. DIFFERENT FORMATION SHAPES

Apart from the Truck-Trailer setup and parallel setup, the other formation shapes that can be used with the proposed control algorithm and its feasibility are checked. The proposed control algorithm is expanded for a set of three robots in a wedge shaped or v-shaped formation done by changing different desired formation parameters to 45° and 315° respectively with a separation of 700mm and an additional of 700mm for each of the robot from thereafter. Since multiple trailers stability has been established in the last chapter, more trailers are also added for getting into different formation shapes. The detailed analysis is done for wedge shaped with 4 trailers. The results on the linear and angular separation are plotted in the figures 5.14 and 5.15.

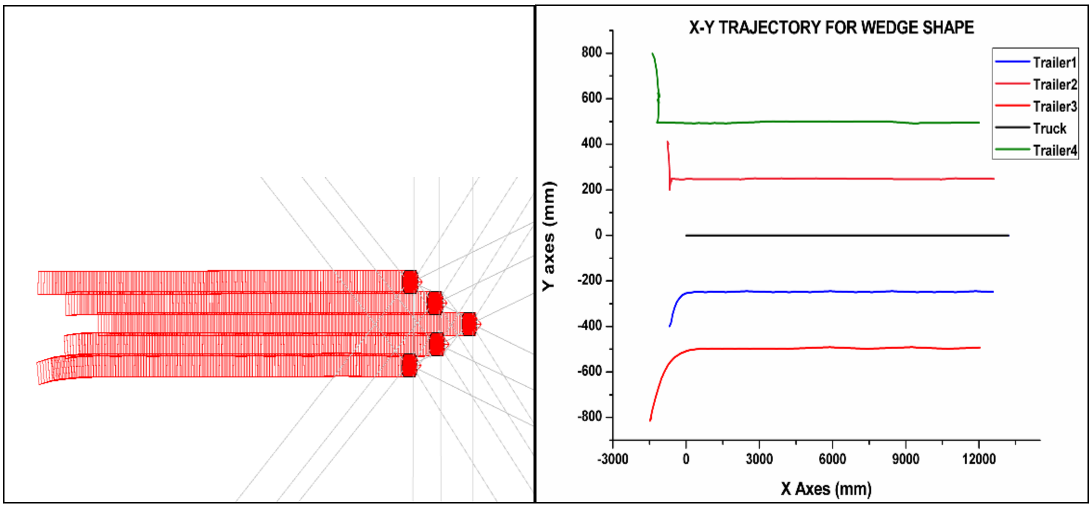


Fig 5.13 a.) Screenshot of the Simulator and b.) the corresponding plots in terms of X-Y Trajectory

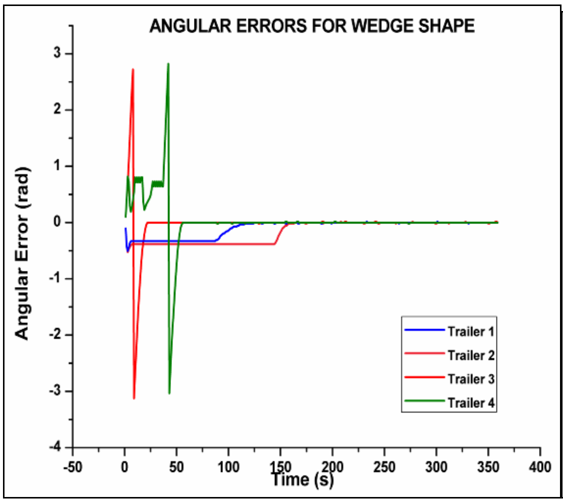


Fig 5.14 Angular Errors – Wedge – 5 trailers

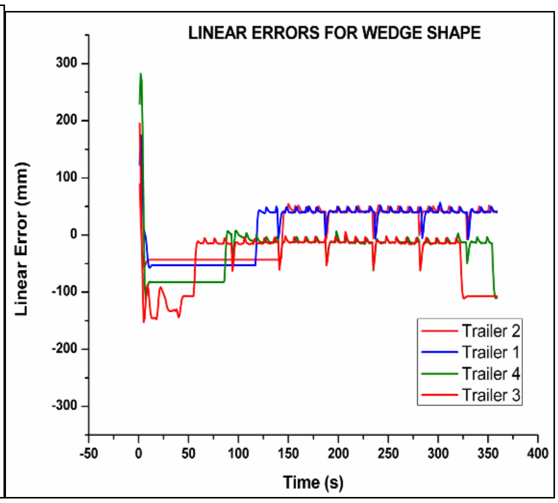


Fig 5.15 Linear Errors – Wedge – 5 trailers

The average tracking errors in terms of linear motion is -10.23mm for the nearest robots and 50.48mm for the third and fourth robots respectively. The angular tracking errors are found to be -0.032rad for the nearest robots and 1.05rad for the third and fourth robots respectively. There are huge tracking errors for robots from the 5th trailer compared to multiple trailers in truck-trailer and parallel arrangement which can add up to 7 trailers in the configuration. This is because of a different formation shape that results in change of both orientation and the position. Each robot is offset by 700mm along the X axes and 800mm along the Y axes with a 45° or 315° difference in orientation with the truck robot.

5.7. FINAL ALGORITHM FOR TRUCK & TRAILER

By adding the localization feature to the existing algorithm, this section brings in the change in the control flow diagram for the truck and trailer.

Truck: The truck first localizes itself in the given environment i.e. the robot's odometers are updated to the required initial position by the controller depending upon the formation pattern. After initialization, the truck robot immediately checks for any stop request from the trailer. This checking cycle happens for every 100ms. The truck is given with a trajectory as required and the truck starts navigating by updating the current pose and the velocity vectors to a register for the computation of trailer velocity by the tracking controller. This cycle is repeated till the goal is reached or when the trailer issues a stop request to the truck. If a 'stop' command is requested, it can be temporary or permanent.

If it's temporary, the truck waits for 6000ms for the trailer to issue a 'go' command, which means that the reformation has occurred and the trailer is in place. Then, the truck starts driving itself to the required goal along with the trailer. However, if a permanent 'stop' command is given or the trailer fails to reform back within the 6000ms, then the truck breaks up the formation to be independent. In the former case, a new park goal is given to the truck and the truck parks itself in the given place by driving towards its new goal. In the latter case, the truck declares the trailer lost and stops navigating. The control flow chart of the truck is shown in the fig.5.15.

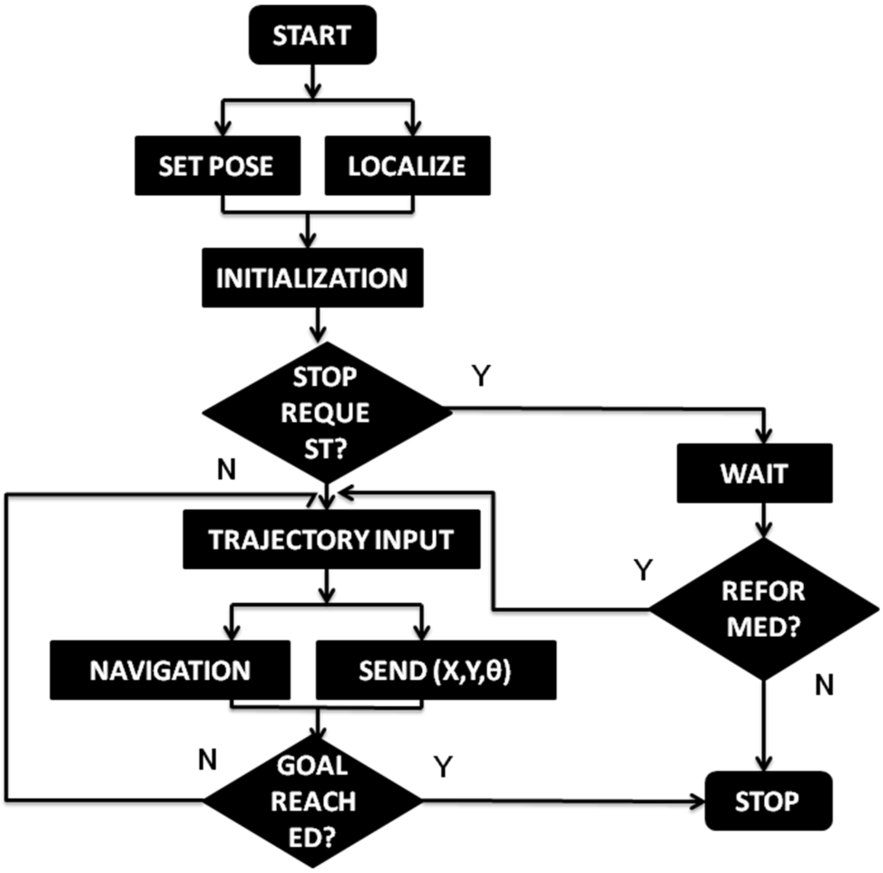


Fig 5.15 Control Flow for the Truck

Trailer: The control flow of the trailer begins by localizing itself in the given environment or resetting the robot's odometer to the required initial pose by the control algorithm. The trailer receives data packet from the truck every 100ms and computes the desired velocity parameters. It tracks the truck trajectory by setting the new velocity parameters, which are calculated at every instant. The trailer also looks for dynamic obstacle in the path. When the trailer detects an obstacle or the trailer cannot maintain formation with the truck without avoiding the obstacles, then it sends a stop request to the truck. Otherwise, the obstacle is avoided using the obstacle avoidance algorithm similar to the decentralized collision avoidance algorithm by means of a deconfliction manoeuvre which calculates the angle of deflection needed to avoid the obstacle. The trailer avoids the obstacle and then reforms with the truck by computing the new formation parameters. However, if the trailer robot takes long time (60000ms) (which is size for any elastic band around the robot so that the path can be deformed) to reform with the truck, then the trailer declares itself as lost. The Fig.5.17 shows the control flow chart for the trailer.

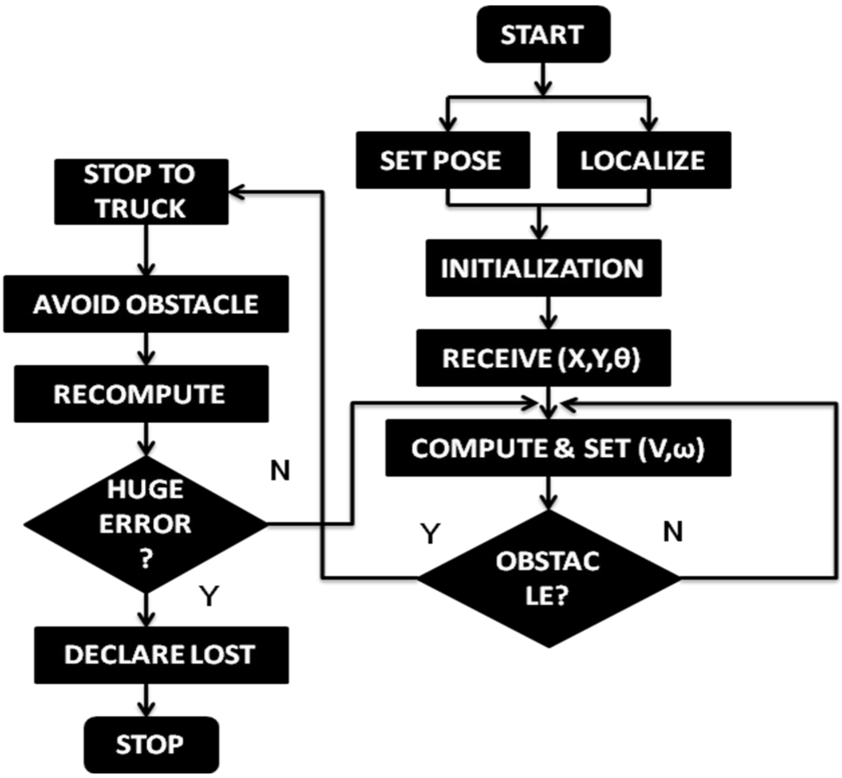


Fig 5.17 Control Flow for the Trailer

The ‘stop’ request issued to the truck could also be permanent, if the trailer is issued a park command by the user. Then both the robots break the formation and park themselves or move to any places as available.

CHAPTER 6

CONCLUSIONS AND SCOPE FOR FUTURE WORK

6.1 SUMMARY

In this research work, a new leader-follower formation framework for a couple of nonholonomic mobile robots is developed to imitate a real truck-trailer system. A virtual link and virtual hinge point are introduced between the two robots, which provide a better control over the system. Multi-layered control architecture with different behaviors has been developed to address the specific problems of multi-robot formation control and in particular truck-trailer system. Problems like: dynamic obstacle avoidance of the trailer, jack-knife phenomenon, parking space constraints and multiple trailers stability are addressed here.

The total functionality of the entire system is divided in to three different task-achieving behaviors. These control laws also known as behaviors are derived from the basic motion states of the robot – namely formation controller that will make the two robots to imitate a truck-trailer system, obstacle avoidance of the trailer – dynamic obstacles that come on the path of the follower robot in a multi-robot system and collision avoidance between the robots – jackknife phenomenon that happens during braking and sharp turning in truck-trailer system. These control laws are developed separately depending upon the kinematics of a set of differential drive robots. They are then prioritized based on the information from external sensors and the importance of each of the individual behavior as per the characteristics of a real truck-trailer system. This priority based control laws are formed in to multiple layers, as only few control laws share the same motion states. Additionally, there are two more temporary behaviors, which allow the controller to easily switch between the three main individual behaviors. These two behaviors also help in addressing the two other problems of a real truck-trailer system, parking space constraints and multiple trailer stability.

Hence, a multi-layered behavior based formation control framework is developed. This approach makes it easier to solve complex problems with theoretical formalization. It also provides a tight coupling between perception and action. Further, the usage of the SDK libraries available with the experimental robots was exploited to the fullest. Each of the individual behaviours was modelled successfully using the object oriented programming methods of the C++ library.

Series of simulation and experimental studies were carried out in an incremental manner to evaluate the performance of the proposed control architecture to maintain a well-defined formation of multiple robots and also to navigate in the unknown environment by avoiding obstacles on their paths. Simulation and experimental studies were carried out using the SDK libraries available with Amigobot and Pioneer P3DX robots. The typical problems that arose while implementing on the real robots are: position estimation / localization, errors in odometry, inter-robot communication delay, and sensory errors due to noise in signals are assumed at the ideal conditions. The effect of functional parameters of the formation approach such as linear separation & angular separation and a detailed error analysis are done using simulation and experimental studies. Moreover the controller is analyzed for its stability, robustness and disturbances rejection capabilities. The feasibility of using the developed controller to create and maintain different formations like parallel formation has been studied in this work. Moreover, stability problem of multiple robots in the formation is also investigated to show the versatility of the developed control. Implementation of this algorithm for multiple mobile robots give rise to a new leader-follower architecture which does not suffer from reliability problem on a single leader. Here, almost all the robots play a dual role of being the leader as well as the follower.

6.2 OBSERVATIONS

Based on the results, the following observations were made:

- The developed control approach exploits the advantages of both behavior based and leader-follower approaches by using priority based behaviors and one leader leading the follower robot.
- Division of the total functionality/requirement into different control laws based on the robot motion states makes the formation problem easier to solve.
- Simple decentralized reactive collision avoidance algorithm is used for avoiding dynamic obstacles in the path of the follower robot.
- Investigations of the controller performance through simulations and experimentation using Amigobot and Pioneer robot Platforms.
- A modified leader-follower control architecture is established that solves the reliability problem on one or more leaders in leader-follower architecture.
- At all instants, follower maintains the formation by adjusting its position on Y axis initially and then it mimics the behavior of the leader with desired separation and orientation, until the connection between them is lost.

- The introduction of coordinate transformation in which the error dynamics is matched robots local coordinate frame helps to maintain the desired formation irrespective of changes in the direction of robots in the work space
- Communication bandwidth and signal strength between the robots results in transitory errors. But the major delay is due to the difference in the update frequency and controller update frequency. This can cause a delay up to 300-400ms depending upon the signal strength.
- Introduction of virtual hinge point as the basic control point in the multi robot system enables effective control over the formation (minimum linear & angular errors are -1.108 mm and 0.0024 rad, at a linear velocity of 50mm/sec respectively).
- Experimental and simulation investigations on the proposed control approach showed that the linear and angular formation errors between the robots are found to be less than ±3.5% and ±0.2% respectively when compared to ±3% and ±0.81% reported in the literature (Rafel Fierro et. al., 2002 and Kuppan et al 2008).

6.3 CONCLUSIONS

This work helps in looking the classic truck-trailer problem in a different perspective. The problem is looked upon as an inline formation control with an intermediate virtual point taken in space that governs the control of the complete formation. This intermediate point makes this virtual truck-trailer system highly reliable compared to other formations. This virtual truck-trailer system can be successfully addressed in areas, where the problems of the truck-trailer systems cannot be avoided. For example, places as in a narrow shop-floor line; the jack-knifing of two robots results in failure of the two robots. In such places, this system can overcome such problems. The only disadvantage of the present system is that the usage of individual robots makes the system expensive and the presence of few singularity points where the controller fails.

6.4 CONTRIBUTIONS TO THE KNOWLEDGE BASE

- The three controller gains (1 for linear and 2 for angular control) introduced here has effective control over the formation.
- The controller does not depend on any of the parameters of the robot and hence can be used with any set of differential drive mobile robots.
- This control architecture can be extended for n – number of robots (multiple trailers) and irrespective of the formation shape.
- An improved leader-follower control approach that overcomes the reliability problem on a single leader is also proposed.
- The whole system is globally stable except at few singularity points (phase shift of 180˚ to -180˚).

6.5 SCOPE OF FUTURE WORK

The scope of future work is enormous due to the deadlock situations that occur in the developed approach during obstacle avoidance when working with multiple robots. A new approach can be developed rather than the decentralized reactive collision avoidance algorithm that allows only the feasibility with two robots.

Again, the switching from one formation to another is not allowed during the execution of high prioritized behaviours, due to the possibility of simultaneous request from multiple trailers and resulting dead-lock condition. This is another area that is open to discuss upon.

Moreover, the virtual truck-trailer system can be made in to a real truck-trailer system by attaching a serial manipulator arm on top of the truck robot and a trolley on top of the trailer robot. New behaviours can be incorporated for each of the robots separately to deal with the serial manipulator working by the truck and the docking of the trolley by the trailer.

The idea of the present multi-layer control architecture can be extended to vehicle operating in the uneven terrain, taking in to account the dynamics of the robot. This makes it possible to develop a robust control strategy for vehicles operating in the highly uneven terrain. This controller can also be extended to the formation of aerial robots with mobile robots where it involves six degrees of freedom.

REFERENCES

[1] **Anderi M. Rothenstein., Albert L. Rothenstein** and **John K. Tsotos** (2003) Mission Directed Behaviour Based Robots for Planetary Exploration, Proceedings of the *7th International Symposium on Artificial Intelligence, Robotics and Automation in Space: i-SAIRAS*, NARA, Japan.

[2] **Arkin R.** *Behavior-Based Robotics,* The MIT Press, Cambridge, Massachusetts, 1998.

[3] **Bailong Liu., Rubo Zhang** and **Changting Shi** (2006) Formation control of multiple behaviour based robots, Proceedings of the *IEEE International conference on computational intelligence and security*, Guangzhou, China, 1, 544-547.

[4] **Balch, T.** and **R. C. Arkin** (1998) Behavior-Based Formaion Control for Multirobot Teams, *IEEE Transactions on Robotics and Automation*, 14(6), 926-929.

[5] **Barca, J.C., A. Sekercioglu** and **A. Ford** (2012) Controlling Formations of Robots with Graph Theory, Proceedings of the 12th International Conference IAS-12, Jeju Island, Korea, 563-574.

[6] **Brockett RW**, (1982) Control theory and singular Riemannian geometry, *In New Directions in Applied Mathematics*, pages 11-27, Springer-Verlag.

[7] **Brooks, R.** (1986) A Robust layered control system for a mobile robot, *IEEE Journal on Robotics and Automation*, 2(1), 14-23.

[8] **Bushnell, L.G.,** (1995) An obstacle avoidance algorithm for a car pulling trailers with off-axle hitching, *Proceedings of the 34th IEEE Conference on Decision and Control*, 4, 3837-3842.

[9] **Chaimowicz, L., V. Kumar, R. Mario** and **F.M. Campos** (2004) A Mechanism for Dynamic Coordination of Multiple Robots, *International Journal of Autonomous Robots*, 17(1), 7-21.

[10] **Chen, Y.** and **Z. Wang** (2005) Formation control: A Review and Consideration, *Proceedings of the IEEE/RSJ international Conference on Intelligent Robots and Systems*, Alberta, Canada, August, 3181-3186.

[11] **Chetty, R.M.K., M. Singaperumal** and **T. Nagarajan** (2011) Distributed Formation Planning and Navigation Framework for Wheeled Mobile Robots", *Journal of Applied Sciences II*, 9, 1501-1509.

[12] **Dai, Y.** and **S.G. Lee** (2012), Leader-follower formation control of non-holonomic mobile robots, *International Journal of Control, Automation and Systems*, 10(2):350-36.

[13] **Das, A.K., R. Fierro, V. Kumar, J.P. Ostrowsky, J. Spletzer,** and **C. Taylor** (1998) A Vision-Based Formation Control Framework, *IEEE Transaction on Robotics and Automation*, 18(5), 813–825.

[14] **Desai, J. P.** (2002) A graph theoretic approach for modelling mobile robot team formations, *Journal of Robotic Systems*, 19 (11), 511-525.

[15] **Desai, J. P., J.P. Ostrowski,** and **V. Kumar** (2001) Modeling and control of formations of nonholonomic mobile robots, *IEEE Transactions on Robotics and Automation*, 17 (6), 905-908.

[16] **Desai, J.P., J. Ostrowski** and **V. Kumar** (1998) Controlling Formations of Multiple Mobile Robots, *Proceedings of IEEE Int. Robotics &Automation*, Leuven, Belgium, 2864-2869.

[17] **Divelbliss A.W.,** and **Wen J.T.**, (1997) Trajectory tracking control of a car-trailer system, *Proceedings of the IEEE Transactions on Control Systems Technology*, 5(3), 269-278.

[18] **Doh-Hyun Kim** and **Jun-Ho Oh**, (1998) Experiments of backward tracking control for trailer system, *Proceedings of IEEE International Conference on Robotics and Automation*, 1, 19-22.

[19] **Doh-Hyun Kim** and **Jun-Ho Oh**,(1999) Non-linear tracking control of trailer systems using Lyapunov Direct Method, *Journal of Robotics Systems*, 16(1), 1-8.

[20] **Domen Novak, Dejan Dovzan, Rok Grebensek, Simon Oblak**, (2007) Automated parking of a truck and trailer, *Proceedings of International Conference on Advances in Internet, Processing, Systems and Interdisciplinary Research*.

[21] **Douglas Vail** and **Manuela Veloso** (2003) Dynamic Multi-Robot Coordination, 87-100, In *Multi-Robot Systems: From Swarms to Intelligent Automata*, Vol. II, Kluwer Academic Publishers.

[22] **Dylan A Shell** and **Maja Matarić** (2005) Behaviour-Based Methods for Modelling and Structuring Control of Social Robots, 279-306, In *Cognition and Multi-Agent Interaction: From Cognitive Modelling to Social Simulation*, Cambridge University Press.

[23] **Fierro, R., A. Das, J. Spletzer, J. Exposito, V. Kumar, J.P. Ostrowski, C.J. Taylor, Y. Hur, R. Alur, I. Lee, G. Grudic** and **B. Southall** (2002) A framework and architecture for multiple-robot coordination, *International Journal of Robotics Research*, 21 (10-11), 977-995.

[24] **Fiorini,P.,** and **Shiller Z.**, (1998) Motion planning in dynamic environments using velocity obstacles, *International Journal of Robot Research*, 17, 760-772.

[25] **Fujimori, A., T. Fujimoto** and **G. Bochas** (2008) Mobile Robot Formation Control using a Modified Leader-Follower Technique, *International Journal of Integrated Computer Aided Engineering*, 15, 71-84.

[26] **Gonzalez-Cantos, A.** and **A. Ollero**, (2009) Backing-up Maneuvers of Autonomous Trailer-Trailer Vehicles using the qualitative theory of non-linear dynamical systems, *International Journal of Robotics Research*, 28(1), 49-65.

[27] **Harry Chia., Hung Hsu** and **Alan Liu** (2005) Multiagent-Based Multi team Formation Control for Mobile Robots, *International Journal of Intelligent and Robotic Systems*, 42(1), 337-360.

[28] **Hougen, D.F.; Fischer, J.; Gini, M.** and **Slagle, J.,** (1996) Fast connectionist learning for trailer backing using a real robot, *Proceedings of IEEE International Conference on Robotics and Automation*, 2, 1917-1922.

[29] **Hougen, D.F.; Fischer, J.; Gini, M.** and **Slagle, J.,** (1997) Rapid Unsupervised Connectionist learning for backing a robot with two trailers, *Proceedings of IEEE International Conference on Robotics and Automation*, 4, 2950-2955.

[30] **Imen Ayari** and **Abderrazak Chatti** (2007) Reactive Control using Behavior Modelling of a Mobile Robot, *International Journal of Computers, Communication and Control*, II(3), 217–228.

[31] **J.P. Laumond,** (1998) Robot Motion planning and Control, *Lecture Notes in Control and Information Sciences*, 229, Springer Verlag.

[32] **Jin Cheng; Yong Zhang** and **Zhonghua Wang,** (2009) Backward Tracking of Mobile Robot with One Trailer via Fuzzy Line-of-Sight Method, *Proceedings of 6th International Conference on Fuzzy Systems and Knowledge Discovery*, *FSKD 4*, 66-70.

[33] **Klančar., G, D. Matko** and **S. Balžič** (2009) Wheeled Mobile Robot in a Linear Platoon, *Journal of Intelligent Robot Systems*, 54, 709 – 731.

[34] **Kong, S.-G.** and **Kosko, B.,** (1992) Adaptive fuzzy systems for backing up a truck-and-trailer, *Proceedings of IEEE Transactions on Neural Networks*, 3(2), 211-223.

[35] **Lamiraux, F.** and **Laumond, J-P,** (1998) A practical approach to feedback control for a mobile robot with trailer," *Proceedings of IEEE International Conference on Robotics and Automation*, 4, 3291-3296.

[36] **Lamiraux, F.** and **Laumond, J-P**, (2000) Flatness and small-time controllability of multibody mobile robots: application to motion planning, *Proceedings of IEEE Transactions on Automatic Control*, 45(10), 1878-1881.

[37] **Lamiraux, F.; Sekhavat, S.** and **Laumond, J-P,** (1999) Motion planning and control for Hilare pulling a trailer, *Proceedings of IEEE Transactions on Robotics and Automation*, 15(4), 640-652.

[38] **Large, F.; Sekhavat, S.; Shiller, Z.** and **Laugier, C.,** (2002) Towards real-time global motion planning in a dynamic environment using the NLVO concept, *Proceedings of International Conference on Intelligent Robots and Systems*, 1, 607-612.

[39] **Laumond, J-P; Jacobs, P.E.; Taix, M.** and **Murray, R.M.,** (1994) A motion planner for non-holonomic mobile robots, *Proceedings of IEEE Transactions on Robotics and Automation*, 10(5), 577-593.

[40] **Li, W.; Tsubouchi, T.** and **Yuta, S.,** (2000) Manipulative difficulty index of a mobile robot with multiple trailers in pushing and towing with imperfect measurement, *Proceedings of IEEE International Conference on Robotics and Automation*, 3, 2264-2269.

[41] **Luiz Chaimowicz., V. Kumar, R. Mario** and **F.M. Campos** (2004) A mechanism for Dynamic coordination of Multiple Robots, *International Journal of Autonomous Robots*, 17(1), 7-21.

[42] **Maciej Michalek,** (2012) Tracking control strategy for the standard N-trailer mobile robot: geometrically motivated approach, *Robot Motion and Control 2011, Lecture Notes in Control and Information Sciences*, 422, 39-51.

[43] **Maja J Matarić** (1997) Behaviour based control: Examples from navigation, learning, and group behaviour, *Journal of Experimental and Theoretical Artificial Intelligence*, 9(2-3), 323-336.

[44] **Martinez, J.L.; M. Paz** and **Garcia-Cerezo, A.**, Path tracking for mobile robots with a trailer, *Proceedings of the 15th IFAC World Congress*, 15, 865-865, 2002.

[45] **Masanori Fuji., T. Yamamoto** and **T. Fujinami** Stable formation driving of Mobile Robots with hybrid strategy, pp. 238-242, In **K. Murase** and **T. Asakura** (eds.), *Dynamical systems approach to embodiment and sociality,* Advanced Knowledge International, Adelaide.

[46] **Masoud,** A.A. (2007) Decentralized, Self-organizing, Potential field-based Control for Individually-motivated, Mobile Agents in a Cluttered Environment: A Vector-Harmonic Potential Field Approach, *IEEE Transactions on Systems, Man and Cybernetics, Part-A: Systems and Humans*, 37(3), 372-390.

[47] **Matsushita, K.** and **Murakami, T.,** (2006) Backward Motion Control for Articulated Vehicles with Double Trailers Considering Driver's Input, *32nd Annual Conference on IEEE Industrial Electronics, IECON 2006*, 3052-3057.

[48] **Michel Fliess; Jean Levine; Philippe Martin** and **Pierre Rouchon,** (1993) Flatness and Defect of Non-linear Systems: Introductory theory and examples, *International Journal of Control* 61(6).

[49] **MortezaSharafi; A. Zare** and **Khalil Alizadeh,** (2011) Intelligent Parking method for truck and trailer in presence of fixed and moving obstacles, *Australian Journal of Basic and Applied Science*, 5(11), 398-416.

[50] **Murray, R.M.** and **Sastry, S.S.,** (1990) Steering non-holonomic systems using sinusoids, *Proceedings of the 29th IEEE Conference on Decision and Control*, 4, 2097-2101.

[51] **Myoungkuk Park., Woojin Chung, Munsang Kim** and **Jaebok Song** (2004) Control of a Mobile robot with passive multiple Trailers, Proceedings of the *IEEE International conference on Robotics and Automation*, 4369-4374, Barcelona, Spain.

[52] **Ngo, VT., AD. Nguyen** and **QP. Ha** (2005) Integration of Planning and Control in Robotic Formations, Proceedings of the *Australasian Conference on Robotics and Automation*, Sydney.

[53] **Nyugen, D.** and **Widrow, B.**, (1989) The truck backer-upper: an example of self-learning in neural networks, *International Joint Conference on Neural Networks*, 2, 357-363.

[54] **Roh, J.; Woojin Chung,** (2010) Reversing control of a car with a trailer using a Driver Assistance System, *Proceedings of IEEE Workshop on Advanced Robotics and its Social Impacts ARSO*, 99-104.

[55] **Rouchon, P.; Fliess, M.; Levine, J.**and **Martin, P.,** (1993) Flatness, motion planning and trailer systems, *Proceedings of the 32nd IEEE Conference on Decision and Control*, 3, 2700-2705.

[56] **Saeki, M.,** (2002) Path following control of articulated vehicle by backward driving, *Proceedings of the International Conference on Control Application*, 1, 421-426.

[57] **Sampei, M.; Tamura, T.; Itoh, T.** and **Nakamichi, M.,** (1991) Path tracking control of trailer-like mobile robot, *Proceedings of IROS Intelligent Robots and Systems - 'Intelligence for Mechanical Systems,*1, 193-198.

[58] **Sampei, M.; Tamura, T.; Kobayashi, T.** and **Shibui, N.,** (1995) Arbitrary path tracking control of articulated vehicles using non-linear control theory, *Proceedings of IEEE Transactions on Control Systems Technology*, 3(1), 125-131.

[59] **Samshudin H.M Amin., R. Mamat** and **T. C. Kwong** (2005) Behaviour Based Mobile Robot Navigation with Dynamic Voting Technique, 201-222, *Cutting Edge Robotics*, *Scientific Book on Advanced Robotics Research*, ARS Publishers, Germany.

[60] **Samson, C.,** (1995) Control of chained systems application to path following and time-varying point-stabilization of mobile robots, *Proceedings of IEEE Transactions on Automatic Control*, 40(1), 64-77.

[61] **Sanchez, J.,** and **F. Fierro** (2003) Sliding Mode Control for Robot Formations, Proceedings of IEEE International Symposium on Intelligent Control, Houston, Texas, USA, 438-443.

[62] **Sekhavat, S., P. Svetska, J. P. Laumond** and **M. Overmars,** (1998) Multi-level path planning for non-holonomic robots using semi-holonomic subsystems, *International Journal of Robotics Research*, 17(8), 840-857.

[63] **Sekhavat, S.; F. Lamiraux; JP Laumond; G. Bauzil** and **A. Ferrand,** (1999) Motion planning and control for Hillare pulling a trailer: experimental issues, *Proceedings of IEEE Transactions on Robotics and Automation*, 15(4).

[64] **Sordalen, O.J.,** (1993) Conversion of the kinematics of a car with n trailers into a chained form, *Proceedings of IEEE International Conference on Robotics and Automation*, 1, 382-387.

[65] **Tan, K. H.** and **M. A. Lewis** (1997) Virtual structures for high-precision cooperative mobile robot control, *Autonomous Robots*, 4, 387-403.

[66] **Tanaka, K.** and **Yoshioka, K.,** (1995) Fuzzy trajectory control and GA-based obstacle avoidance of a truck with five trailers, *Proceedings of IEEE International Conference on Systems, Man and Cybernetics, Intelligent Systems for the 21st Century*, 5, 4378-4382.

[67] **Tanaka, K.; Kosaki, T.** and **Wang, H.O.,** (1998) Backing control problem of a mobile robot with multiple trailers: fuzzy modeling and LMI-based design, *Proceedings of IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 28(3), 329-337.

[68] **Teck Chew Ng; Guzman, J.I.** and **Adams, M.D.,** (2005) Autonomous vehicle-following systems: a virtual trailer link model, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3057-3062.

[69] **Tilbury D.** and **S. Sastry,** (1995) The Multi-Steering N-trailer system: A case study of Goursat normal forms and prolongations, *International Journal of Robust and Non-linear Control*, 5(4), 343-364.

[70] **Tilbury, D.; Murray, R.M.** and **Shankar Sastry, S.,** (1995) Trajectory generation for the N-trailer problem using Goursat normal form, *Proceedings of IEEE Transactions on Automatic Control*, 40(5), 802-819.

[71] **Toibero, J M., F. Roberti, R. Carelli** and **P. Fiorini** (2008), Formation Control for Non Holonomic Mobile Robots: A Hybrid approach, *Recent Advances in Multi Robot Systems*, Springer verilag.

[72] **Tokunaga, M.** and **Ichihashi, H.,** (1992) Backer-upper control of a trailer truck by neuro-fuzzy optimal control", *Proceedings of 8th Fuzzy System Symposium*, 49-52.

[73] **Wang, A. P.** and **H. Wang** (2008) A generalized time-varying output feedback controller for nonlinear systems, *International Journal of Advanced Mechatronic Systems*, 1(1), 24-31.

[74] **Xiaohai Li** and **Jizhong XIAO** (2005) Robot formation control in Leader-Follower Motion Using Direct Lyapunov Method, International Journal of Intelligent Control and Systems, 10(2), 244-259.

[75] **Yamaguchi, H., T. Arai** and **G. Beni** (2001) A distributed control scheme for multiple mobile robotic vehicles to make group formations, *International Journal of Robotics and Autonomous systems*, 3(4), 125-147.

[76] **Yamamoto, M.; Hayashi, Y.; Mohri, Akira**, (2005) Garage parking planning and control of car-like robot using a real time optimization method, *The 6th IEEE International Symposium on Assembly and Task Planning: From Nano to Macro Assembly and Manufacturing*, 248-253.

[77] **Yongxing Hao** and **Sunil K Agarwal,** (2004) Formation planning and Control of UGVs with trailer, *Autonomous Robots*, 19, 257–270.

[78] **Yongxing Hao** and **Sunil K Agrawal,** (2005) Planning and control of UGV formations in a dynamic environment: A practical framework with experiments, *International Journal of Robotics and Autonomous Systems*, 51, 101-110.

[79] **Yuan, J.** (2010) Formation Control for Mobile Multiple Robots based on Hierarchical Virtual Structures, *8th IEEE International Conference on Control and Automation (ICCA)*, Xiamen, China June, 393-398.

[80] **Zhong-Ping Jiang** and **Nijmeijer, H.,** (1999) A recursive technique for tracking control of non-holonomic systems in chained form, *Proceedings of IEEE Transactions on Automatic Control*, 44(2), 265-279.

LIST OF PAPERS

SUBMITTED ON THE BASIS OF THIS THESIS

I. REFEREED JOURNALS:

[1] **Jennifer David, P.V. Manivannan,** Control of Truck-Trailer Mobile Robots: A Survey - *Int. Journal of Intelligent Service Robotics, Springer Publication (In Press)*

II. PRESENTATION IN CONFERENCES:

[1] **Jennifer David, P.V. Manivannan,** Behavior based Control of Two Mobile Robots in Parallel Formation, *Proceedings of the 9th International Conference on Intelligent and Unmanned Systems, ICIUS,* Jaipur, India, No. 526, September 2013.

[2] **Jennifer David, P.V. Manivannan,** Behavior based Formation Control of Mobile Robots as Truck Trailer using Virtual Link, *Second International Conference on Robotics, Automation and Manufacturing IRAM,* IIT Indore, Dec 16-18, 2013.

III. POSTER PRESENTATION:

[1] **Jennifer David, P.V. Manivannan,** Formation Control of Mobile Robots, Indian Institute of Technology, Madras – Research Scholars' day, March, 2013.

APPENDIX I

β increase in rad	Increase in L_2 (in mm)
0	50
0.909091	49.09091
1.818182	48.18182
2.727273	47.27273
2.727273	47.27273
2.727273	47.27273
3.636364	46.36364
4.545455	45.45455
5.454545	44.54545
7.272727	42.72727
8.181818	41.81818
9.090909	40.90909
9.090909	40.90909
9.090909	40.90909
10	40
10.90909	39.09091
11.81818	38.18182
13.63636	36.36364
14.54545	35.45455
15.45455	34.54545
17.27273	32.72727
18.18182	31.81818
20	30
20.90909	29.09091
21.81818	28.18182
23.63636	26.36364
23.63636	26.36364
24.54545	25.45455
25.45455	24.54545
24.54545	25.45455
23.63636	26.36364
22.72727	27.27273
22.72727	27.27273
21.81818	28.18182
20.90909	29.09091
20	30
18.18182	31.81818
18.18182	31.81818
16.36364	33.63636
16.36364	33.63636

15.45455	34.54545
14.54545	35.45455
12.72727	37.27273
12.72727	37.27273
11.81818	38.18182
10.90909	39.09091
10.90909	39.09091
10	40
9.090909	40.90909
9.090909	40.90909
8.181818	41.81818
7.272727	42.72727
7.272727	42.72727
7.272727	42.72727
5.454545	44.54545
5.454545	44.54545
5.454545	44.54545
4.545455	45.45455
4.545455	45.45455
3.636364	46.36364
2.727273	47.27273
1.818182	48.18182
0.909091	49.09091
0.909091	49.09091
0	50

APPENDIX II

Specifications	Pioneer P3DX	Amigobot
Wheel Drive Movement	Differential Drive 2 foam-filled wheels with caster Wheel Dia: 195.3mm Wheel Width: 47.4mm Gear Ratio: 38.3:1 Turn Radius: Zero Swing Radius: 267 mm	Differential Drive 2 solid rubber wheels with caster Wheel Dia: 102mm Wheel Width: 30mm Gear Ratio: 19.5:1 Turn Radius: Zero Swing Radius: 165 mm
Construction & Operation	Body: High Impact Plastic Shell Frame: Aluminum Robot Weight: 9 Kg	Body: High Impact Plastic Shell Frame: Aluminum Robot Weight: 3.6 Kg
L * b * h (mm)	445 * 393 *237	330 * 280 *150
Payload (kg)	25	1
Sensors • SONARS • Encoders • Bumpers	16 Sonar's (Front array - 1 each side, 6 on front and rear array - 1 each side, 6 on rear @ 20° intervals), 76,600 counts / wheel rev; 128 ticks per mm 10	8 Sonar's (2 on each side, 4 on front and 2 on rear), 39000 ticks / wheel rev; 124 ticks per mm N/A

Run time	8-10 hrs	2-3 hrs
Max Speed	1400 mm/s	1000 mm/s
Rotational Speed	300 deg/s	100 deg/s
Sonar Range (mm)	100 - 5000	
Sonar acquisition rate	25 Hz; 40 ms per sensor per array; 320 ms/array	
Operating power	12V DC	
Electronics	Processor: 44 Mhz Renesas SH2-7144 Data bus: 8 bit R/W; Flash: 128 KB; RAM: 32 KB	
Communication	2 x RS-232 and Wireless Ethernet Controller (TCP – IP)	