Chapter 4 – CAN-2: BEHAVIOUR-BASED COLLISION AVOIDANCE AND FORMATION CONTROL OF MULTIPLE UNMANNED VEHICLES

Dr. Bumsoo Kim
Defence R&D Canada – Ottawa
3701 Carling Avenue
Ottawa, Ontario K1A 0Z4
CANADA

Email: Bumsoo.kim@drdc-rddc.gc.ca

4.1 DATES

26-30 July 2010.

4.2 LOCATION

DRDC Ottawa.

4.3 SCENARIO/TASKS

Multiple UGVs (Unmanned Ground Vehicles: They also are considered as simulated UAVs) to reach to a destination without colliding each other while avoiding obstacles).

4.4 TECHNOLOGIES EXPLORED

In this study, AIC (Artificial Impedance Control) is applied for the generation of trajectory of UGVs instead of pre-planning the trajectory. AIC is a Cartesian space control and is one of the control techniques which can generate trajectories for both obstacle free and obstacle avoidance cases in real time. One of the advantages of artificial impedance control for UGVs motion control is the fact that it enables UGVs to perform obstacle avoidance tasks without knowing the full geometry of the obstacles and of the environment. [1]

In the present study, we started testing AIC algorithm for single vehicle trajectory generation and obstacle avoidance performance using simulation and experimentation.

Then, it was expanded to two vehicles reaching to the designated targets while avoiding collision with each other and avoiding obstacles in their ways to targets.

Thirdly, a five vehicle formation control and single target oriented behaviour-based control were tested in simulation.

4.5 HUMAN FACTORS ISSUES EXPLORED

4.5.1 Reference Frame

Figure 4-1 indicates the global reference frame (T,Q,S) that will be used. Since the X80 robot only supports planar motion, the S coordinate will often be ignored in this work. The global frame is fixed with heading defined as a counter-clockwise rotation about the S axis. The vehicle will use a body-frame coordinate scheme (Figure 4-2) where the X-axis is always pointing forward from the vehicle, and the Y-axis is pointing to the left.

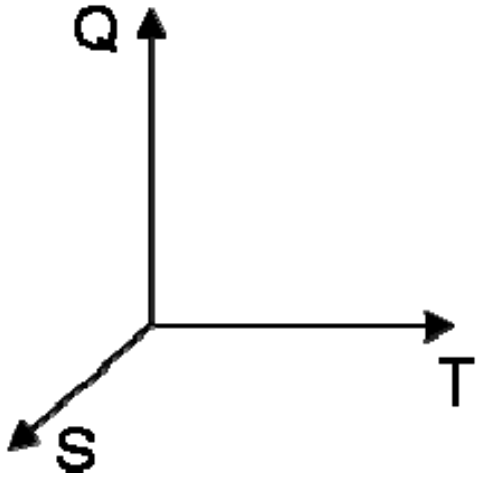

Figure 4-1: Global Coordinate System.

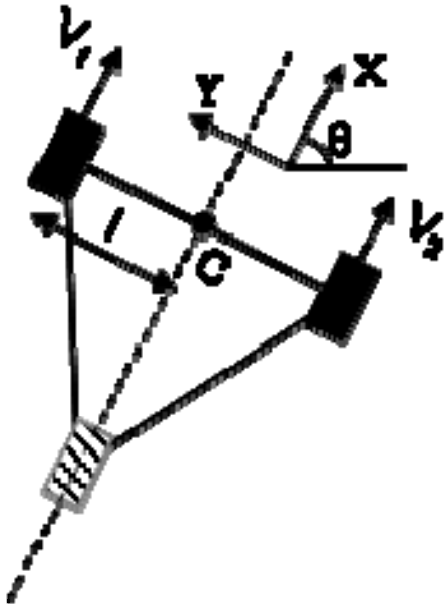

Figure 4-2: X80 Local Coordinate System.

Transformations from body coordinates to global coordinates can be done as follows:

$$\begin{bmatrix} t \\ q \\ \theta \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \tag{1}$$

4.5.2 Kinematic Model

The goal of the model will be to define the motion for all points on the platform, for a given set of known variables. In this case, the speed of the wheels (V_1 and V_2) and the length *l* from the wheels to the center of rotation *c*, are known. With this, we will describe the motion of the center of rotation, and can easily define the motion for all other points from there.

Figure 4-3 shows the axle, and center of rotation of the vehicle.

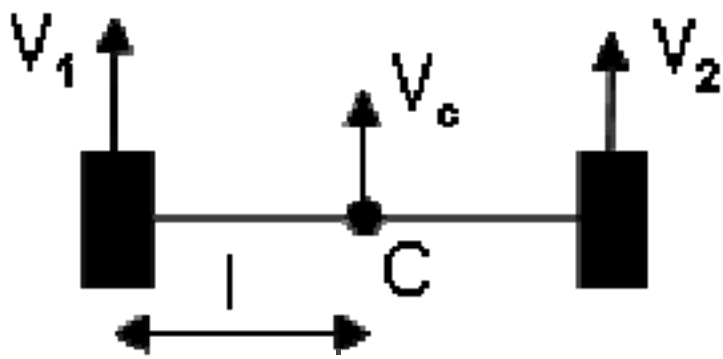

Figure 4-3: X80 Axle and Center of Rotation.

Since point C is directly in the middle of both wheels, its forward velocity will be defined by half of the velocity from each wheel. Thus:

$$V_{C_X} = \frac{1}{2}V_1 + \frac{1}{2}V_2 \tag{2}$$

Assuming there is no side-slip in the wheels, we can also assume that:

$$V_{C_Y} = 0 \tag{3}$$

For the vehicle's angular rotation, we can see that wheel 1 is going to affect the rotation negatively, while the 2nd wheel will have a positive affect. Fixing one of the wheels while driving the other will result in an angular rotation as follows:

$$\omega = \frac{V}{2l} \tag{4}$$

Summing the effects from both wheels will give the platform an angular rotation, about c, of:

$$\dot{\theta}_c = \frac{V_2 - V_1}{2l} \tag{5}$$

Combining equations 1 – 5, we get the following relationship for the motion at c:

$$\boxed{\begin{bmatrix} \dot{t} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1/2 & 1/2 \\ 0 & 0 \\ -1/2l & 1/2l \end{bmatrix} \cdot \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}} \tag{6}$$

4.5.3 Impedance Controller

Without the inherent vehicle dynamics, the impedance controller developed here is actually just a PD controller. However, some attempts have been made to simulate the dynamics of the vehicle and as such, we will continue to use the term ‘impedance’.

Figure 4-4 below depicts the attractive and repulsive forces presented on the vehicle during motion. We will denote the vehicle as *M* and the goal location as *T*. Let $\overline{R}_{mb}$ be the distance from the robot to the closest obstacle, $\overline{r}_b$ repulsive force field radius, $\overline{F}_a$ the attractive force, and $\overline{F}_r$ the obstacle’s repulsive force.

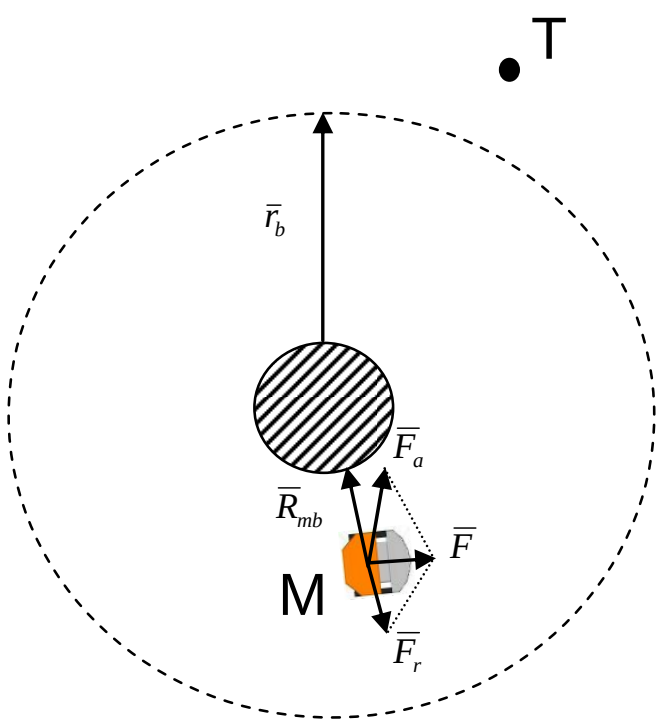

Figure 4-4: Impedance Control Force Diagram.

The repulsive and attractive forces are calculated as:

$$\overline{F}_r = K_r \cdot \frac{\overline{r}_b}{\overline{R}_{mb}} \cdot \left(\overline{R}_{mb} - \overline{r}_b\right)^2 - B_r \cdot V_{cur} \tag{7}$$

$$\overline{F}_a = \begin{cases} K_a \cdot (\Delta d) - B_a \cdot V_{cur} & \forall \Delta d \le S \\ K_a \cdot S & \forall \Delta d > S \end{cases} \tag{8}$$

where K and B are the controller constants, Δd is the distance to the goal, and S is a constant distance after which the force is constant.

Using these forces, we can then calculate the vehicle's desired heading by summing the forces:

$$\theta_{des} = \tan^{-1}\left(\frac{\overline{F}_a}{\overline{F}_r}\right) \tag{9}$$

To steer the vehicle to the desired heading, a proportional controller was used to determine the heading rate:

$$\dot{\theta} = K_\theta \cdot (\theta_d - \theta_c) \tag{10}$$

where θ_{cur} is the current vehicle heading. Using this value, along with the desired trajectory speed V_{set}, equations 2 and 5 can be used to calculate the individual wheel speeds to be sent to the platform.

This controller will cause the robot to move with constant velocity to point *T*, while avoiding any obstacle along its path.

4.5.4 Control Block Diagram

Autonomous navigation was implemented for a single robot using an AIC. An attractive virtual force pulls the robot to its goal, while a repulsive virtual force pushes the robot away from obstacles. The magnitude and direction of the vector sum of these attractive and repulsive forces is used to calculate an appropriate velocity and turning rate for the robot, so that no prior path planning is required. The block diagram showing the general flow of the impedance control program is shown in Figure 4-5 and Figure 4-6.

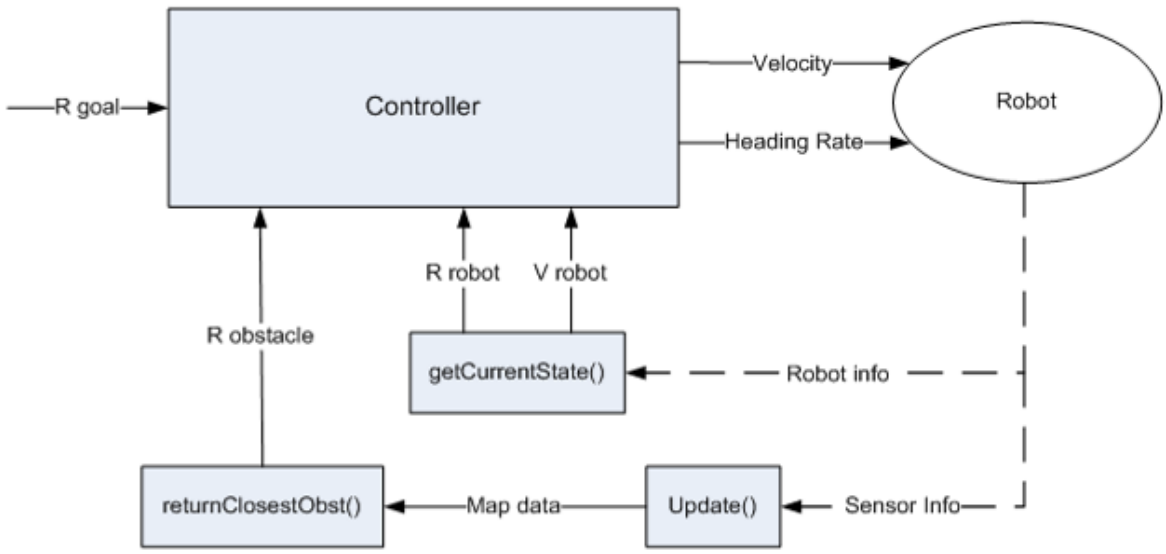

Figure 4-5: Block Diagram of the Impedance Control Program.

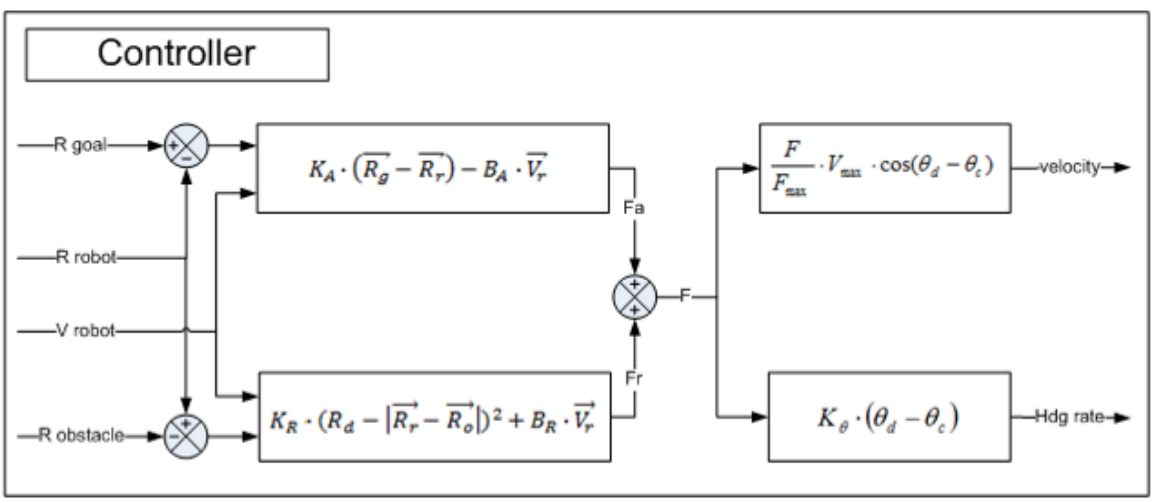

Figure 4-6: A More Detailed Block Diagram of the Controller.

4.6 UNMANNED SYSTEMS USED

4.6.1 Dr. Robot X80Pro

Modified X80Pro [2] is a WiFi enabled robot, and is designed for use as an autonomous navigation and control research platform. It comes equipped with multiple sensors, and low level motor controllers, enabling the user to focus solely on higher level algorithms. An SDK is also available for the windows operating system, simplifying access to the motor drivers, sensors, and communication system. However, for use on

different operating systems, the raw device protocols are given for direct integration. For Windows use, a detailed description of the SDK and how to get up and running with the X80Pro platform can be found in [2].

4.6.2 Player/Stage

Player [3] is a network server for robotic control. It provides access to a platform's sensors and actuators through well-defined interfaces over a TCP connection. As such, it is easy to set up any type of network topology provides that the robot and associated computers are connected over a TCP enabled network.

Stage [4] simulates a population of mobile robots and sensors. Supported sensors cover most areas that are used within the robotics community. Player can access the actuators and sensors in the Stage simulation environment in the same way that it would the actual hardware. As such, it is easy to simulate new algorithms and then transition to the hardware by simply changing the TCP address. Furthermore, it is also possible to mix both simulation and hardware environments. An experiment could be set up where the sensors are read for the simulated world but the actuators are commanded on the actual hardware, or vice-verse. The options are wide ranging.

4.7 SUMMARY OF ANY NATO COMMUNICATIONS/COLLABORATIONS/ INTERACTIONS

Canada (Bumsoo Kim) and France (Gilles Coppin) share interests in ideas and technologies with regard to the swarming concepts of multiple autonomous vehicles operation. Collaborating in this area of research is planned by establishing joint projects and by seeking opportunities to share within NATO Nations in the Task Group.

	Planning/Design	***Execution***	***Analysis***
Communication	*X*	*X*	*X*
Coordination			
Collaboration	*X*		

4.8 SUMMARY OF TD RESULTS

Figure 4-7 shows the results of a simulated obstacle avoidance situation, comparing the case when the robot is given a constant velocity and the case when the robot is allowed to vary its speed between a specified minimum and maximum. Note that the areas where the red data points are more densely concentrated indicate where the robot slowed down.

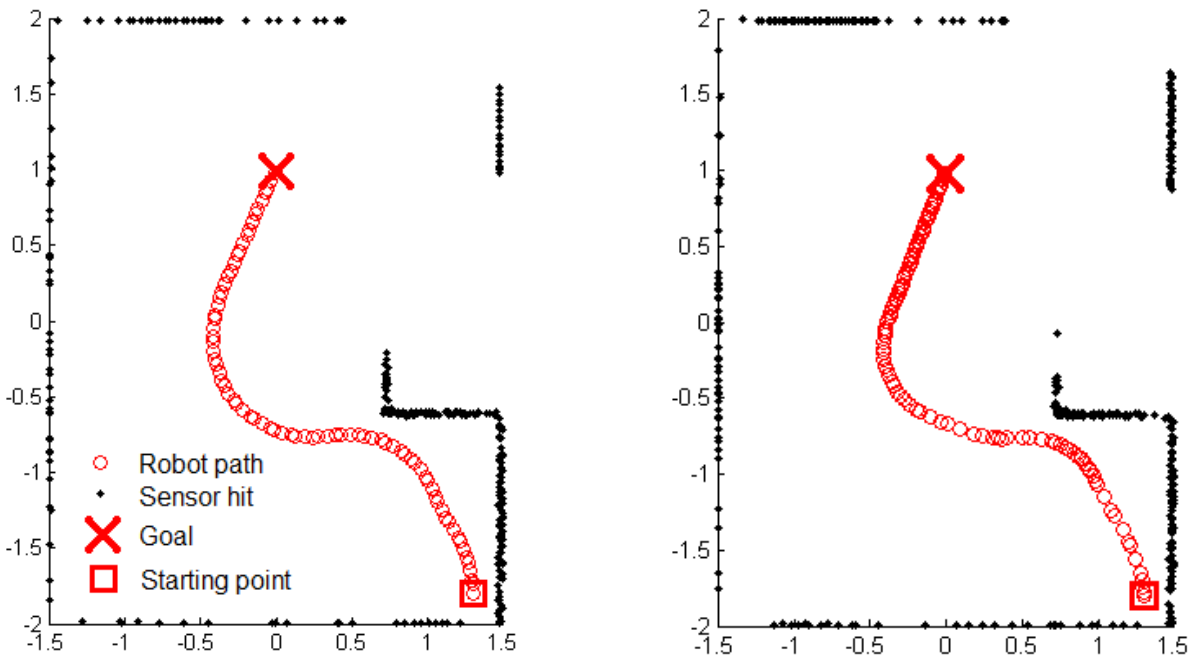

Figure 4-7: Stage Simulation Results
(Left: Constant Velocity (PD Control); Right: Variable Velocity (AIC)).

For simplicity, only one repulsive virtual force is applied to the robot at a time. In the initial implementation, this force was generated from the obstacle nearest the robot. The program used the map builder module to determine the coordinates of the obstacle closest to the coordinates of the robot, and the force was then calculated based on the distance between them. However, there are certain situations in which the repulsive force is ignored, namely when the obstacle is on the other side of the goal from the robot, or when the obstacle is behind the robot. In the latter situation, a problem would sometimes arise with this implementation. That is, even if there was an obstacle within sensor range in front of the robot, no repulsive force would be applied if a closer obstacle happens to be detected behind the robot. In such cases the robot sometimes had a delayed reaction to the obstacles in front of it; it would continue on a straight path until the distance to the obstacle in front was less than the distance to the obstacle left behind.

To fix this problem, the map builder function to return the closest obstacle was modified to also take into account the heading of the robot, so that it returns the closest obstacle *in front* of the robot (with a 180° perspective). After this change was implemented, the robot became more responsive to the objects in front of it, and it allowed for a smoother motion. A comparison of the results from before and after this change is shown in Figure 4-8 and Figure 4-9 below.

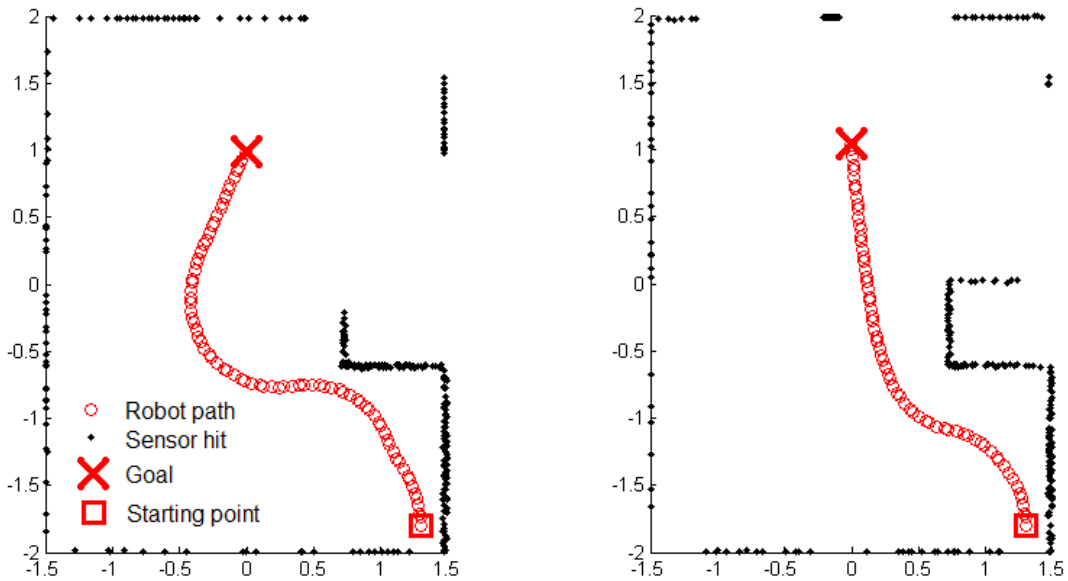

Figure 4-8: Simulation Results (Left: 360° Obstacle Return; Right: 180° Obstacle Return).

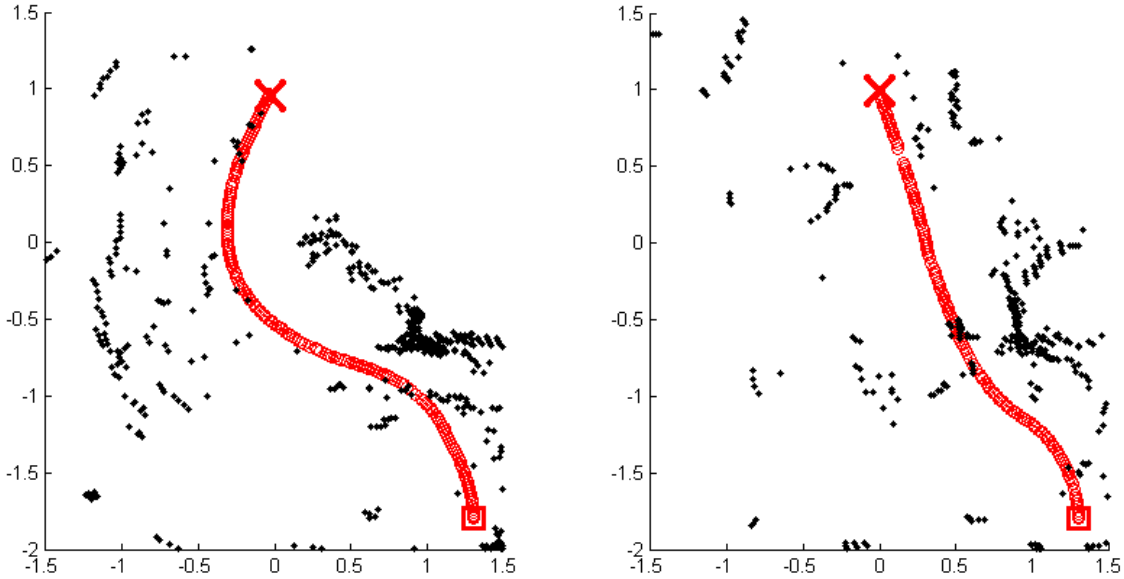

Figure 4-9: Experimental Results (Left: 360° Obstacle Return; Right: 180° Obstacle Return).

4.8.1 Singularities

There is a special case in which another problem arises with the impedance control program. It happens when the attractive and repulsive forces are perfectly lined up (for instance when the goal is on the opposite side of an obstacle from the robot). In the absence of a lateral repulsive force to tell the robot to try to go around the

obstacle, it will travel in a straight path until it gets stuck or crashes into the obstacle. This singularity problem has not yet been overcome.

4.8.2 Multiple Robots

Once the AIC for a single robot was developed and successfully demonstrated in both simulation and experiment, the next step was to extend that functionality to multiple robots. The first test that was done was a Stage simulation that was populated by three independent robots. Each robot used the AIC to navigate and each was given a separate goal point. There was no communication between robots; they could only detect each other as obstacles using their equipped sonar and infrared sensors. Since the robots were essentially moving obstacles, it was not only important for the robots to successfully detect each other, but also important for the robots to be able to detect when the others had moved out of the way. Without this, the map builder would continuously populate the occupancy grid with a streak of obstacles as the robots moved, and it would make navigating to the goals impossible. Figure 4-10 shows the results of two such simulations – one with constant velocity, and one with variable velocity.

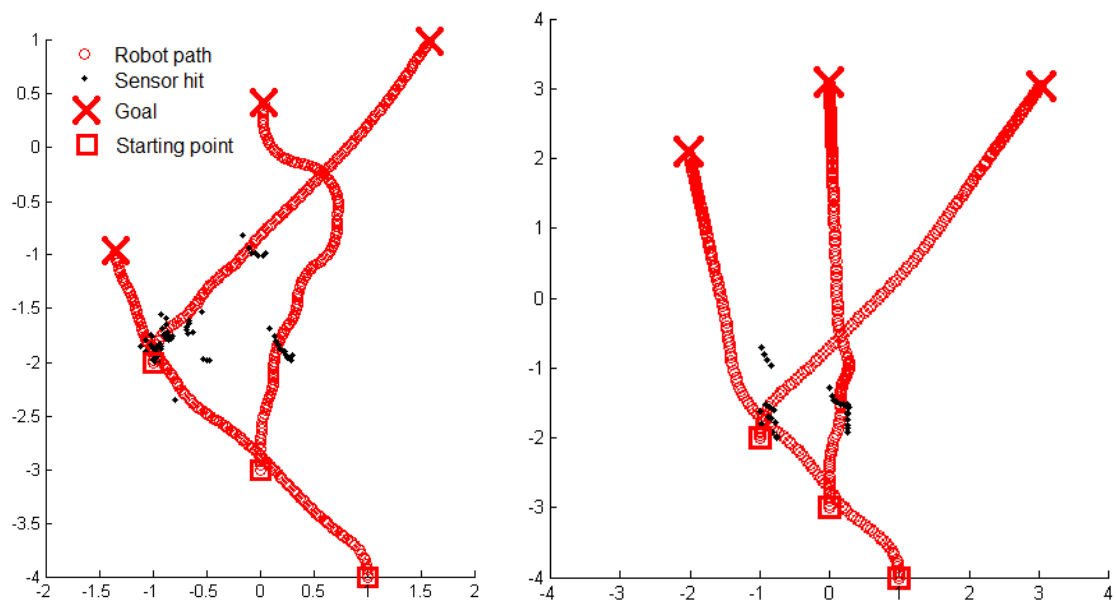

Figure 4-10: Stage Simulation Results (Left: Constant Velocity; Right: Variable Velocity).

It is more useful, however, for robots to be able to communicate and work together to achieve a common goal. Two different approaches to multiple robot control were tried: a neighbour-follower approach, and a behaviour-based approach.

4.8.3 Neighbour-Follower Approach (Formation Control)

The main goal of the neighbour-follower approach is for the robots to achieve a specified formation on their way an end point. Initially each robot is assigned a 'neighbour' robot, and it is told to maintain a certain relative position with respect to its neighbour. This is done using another virtual force, called the formation force, which is added to the vector sum of the attractive force pulling the robot to the goal and the repulsive

force pushing it away from obstacles. In order to calculate this formation force, in the absence of more advanced sensor equipment, it is necessary for the robots to communicate their positions and velocities to their followers. The following equation shows how the formation force is calculated for each robot.

$$F_f = K_f \cdot \left(\Delta\overline{r_d} - \Delta\overline{r_c}\right) - B_f \cdot (\overline{V}_{cur} - \overline{V}_n) \tag{11}$$

where $\Delta\overline{r_d}$ is the desired position of the robot relative to its neighbour, $\Delta\overline{r_c}$ is the actual current position of the robot relative to its neighbour, $\overline{V}_{cur}$ is the current velocity of the robot, $\overline{V}_n$ is the current velocity of its neighbour, and K_f and B_f are controller constants.

This implementation was initially tested with two robots in a Stage simulation. The robots were instructed to form a horizontal line (1 m apart) and move to a goal several metres away, with no obstacles obstructing their path. The simulation is shown in Figure 4-11. The results show that the robots do indeed achieve the formation relatively quickly, but once they approach the end point, they get confused. The problem was that the robots were both given the exact same goal point, so while they were 'fighting' for it, they were unable to maintain the formation. This problem was solved by giving each robot a separate goal at a relative distance based on its relative position in the formation. Different gains Kf and Bf were tested to try to reduce the oscillations of the robots when they were getting in formation.

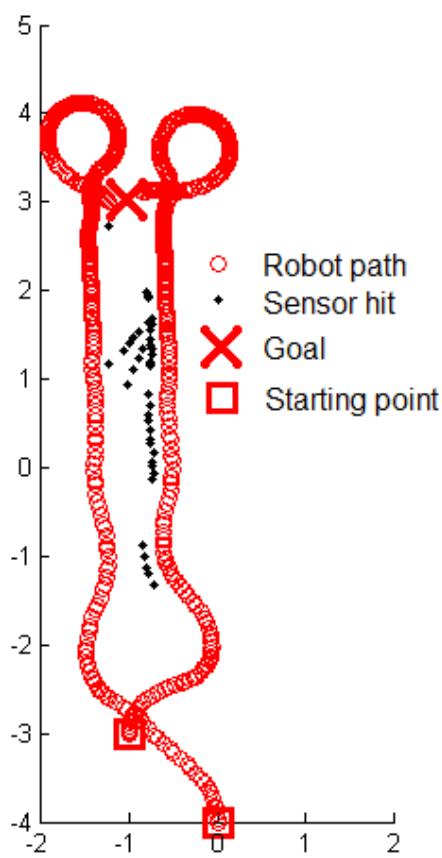

Figure 4-11: Simulation Results for Two Robots Attempting a Horizontal Formation.

4.8.4 Behaviour-Based Approach (Flocking Control)

A behavior-based approach to multi-robot control was also investigated. In this method, no specific formation is explicitly assigned to the robots. Instead, each robot tries to maintain a certain distance (although no specific orientation) with respect to the other robots in its vicinity. For example, if the desired distance between robots is 0.7 m, then robots under this distance from each other will experience a repulsive force, while robots above this distance will be attracted, up to a maximum distance of 1 m. Robots farther than 1 m apart will ignore each other.

When the X80 Pro robots only made use of their sonar and infrared sensors, communication of positions was required in order to distinguish robots from obstacles. In this case, however, instead of only needing to know the position of its one neighbour, each robot required the positions of every other robot. It then had to calculate its distance to every other robot, as well as a force for every robot in range. As more robots were added to the simulation, the computer would get increasingly bogged down, and as a result the sampling frequency of each robot diminished.

In order to determine the extent to which the sampling rate had an effect on the performance of the robots, a simulation was set up in stage involving five robots. The first run was done at normal simulation speed (real time), and it was determined that the frequency of each robot was approximately 1 Hz. Another run was done, this time at a slower simulation speed (0.3 times real time), which allowed more time for the computations to be completed, effectively increasing the frequency of the robots to 9 Hz. The results in Figure 4-12 clearly show that the sampling frequency plays an important role in the stability of the robots.

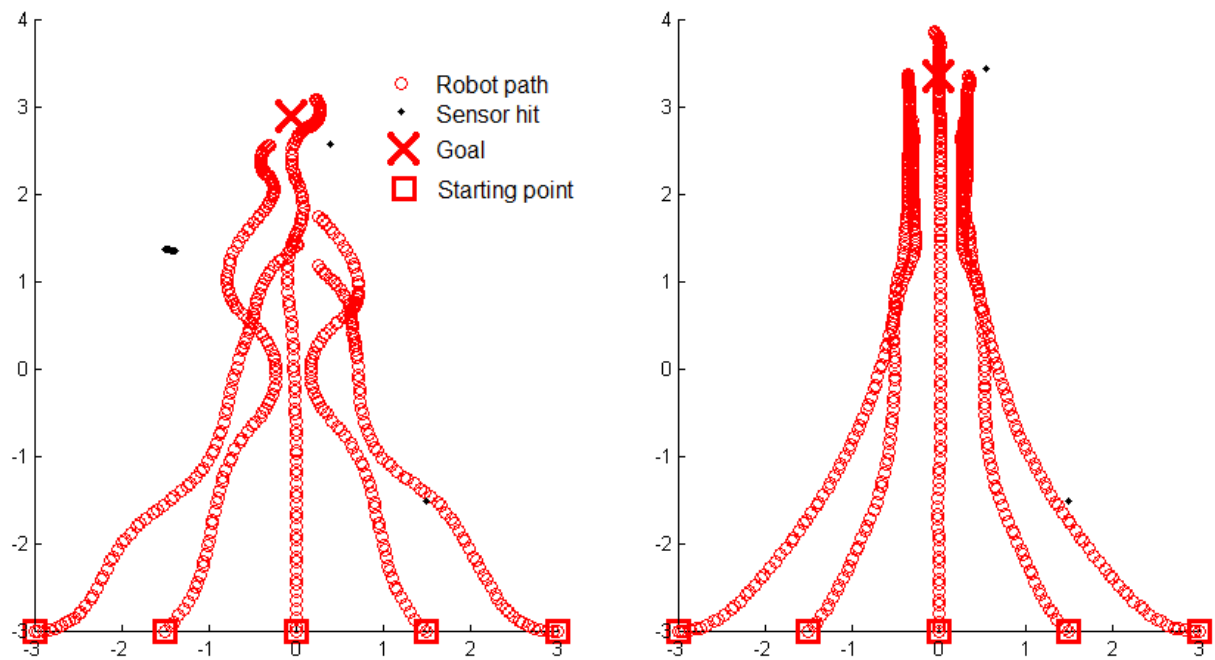

Figure 4-12: Simulation of Behaviour-Based Control (Left: Frequency 1 Hz; Right: Frequency 9 Hz).

4.9 LESSONS LEARNED

Communication link is very important to the control of multiple unmanned vehicles. Distributed computing power is essential for the system stability. AIC proved to be an effective method to generate trajectory, avoid obstacle, and avoid collision with other Unmanned Vehicles (UVs).

The AIC enables UVs to avoid obstacles without knowing the full geometric description which usually requires a complex vision system. The only information needed is the closest point of surface of an obstacle from the vehicle at each time provided by simple range sensors.

4.10 STUDY CONSTRAINTS/LIMITATIONS

Continuation of the research after the completion of present project is in question. Pending funding opportunities proposals are submitted for further investigation. The technology is very high.

4.11 CONCLUSIONS

Advancements of the state of the art in supervisory control of multiple autonomous vehicles can be pursued by studying human interface aspects and the basic self-organizing and protecting autonomous control. We studied and demonstrated the self-organization and protection capabilities of multiple autonomous ground vehicles simulating air vehicles using computer simulations and verifying the results with experimental platforms. The Artificial Impedance Control for local autonomy including collision avoidance and trajectory generation shows excellent results. It is also expanded to study formation control and flocking control. The computer simulation results are really promising.

4.12 FUTURE RESEARCH NEEDS AND PLANS IN THIS AREA

The operator friendly and robust ground control station interface should be researched and developed for the operational capability of the developed technology. And autonomous mission management and more robust flocking control algorithms development should be pursued.

4.13 ACKNOWLEDGEMENTS

Mr. Luc Brunet and Mr. Robert Yandon, Contractors for the project.

4.14 REFERENCES

[1] Kim, B., “Free Motion and Obstacle Avoidance Control for Mobile Robots”, Master’s Thesis, University of Ottawa, August 1992.

[2] Dr. Robot, “Dr. Robot X80 User Manual”, Version 1.0.7, August 2008.

[3] Gerkey, B., “Player 2.1.1 User Manual”, http://playerstage.sourceforge.net/index.php?src=doc.

[4] Vaughan, R., “Stage 3.2.1 User Manual”, http://playerstage.sourceforge.net/index.php?src=doc.