A Simple Limit Cycle Negotiation Strategy for Robot Navigation

Camilo Ordonez*
camilor@eng.fsu.edu

Emmanuel G. Collins, Jr*
ecollins@eng.fsu.edu

Majura Seleka**
majura.selekwa@ndsu.edu

Damion Dunlap*
dunlap@eng.fsu.edu

* Center for Intelligent Systems, Controls and Robotics (CISCOR)
FAMU-FSU College of Engineering
2525 Pottsdamer Street Tallahassee
Florida 32310-6046, 1-850-410 6100

** Department of Mechanical Engineering and Applied Mechanics
North Dakota State University
Fargo, ND 58105

ABSTRACT
Limit cycles can occur when navigating unmanned ground vehicles (UGVs) using behavior-based algorithms. Limit cycles occur when the robot is navigating towards the goal but enters an enclosure that has its opening in a direction opposite to the goal. The robot then becomes effectively trapped in the enclosure. This paper reports a solution to the limit cycle problem for robot navigation in very cluttered environments, for example dense forests. These types of environments offer a challenge due to the diversity of shapes and sizes of deadlocks that are likely to appear. A simple deliberative algorithm for detecting and retracting from limit cycles is described. The algorithm uses spatial memory to detect the limit cycle. Once the limit cycle has been detected, a labeling operator is applied to a local map so that the obstacles that form the boundary of the deadlock enclosure are identified. Subsequently, the robot is directed outside the enclosure using a behavior based control system. Once it exits this region, the deadlocked area is designated as off-limits by means of a virtual wall. Finally, the robotic vehicle proceeds to its original target avoiding the virtual wall and the different obstacles that are found on its way. Simulation and experimental results demonstrate the effectiveness of the proposed method.

Keywords

1. INTRODUCTION
Navigation in unknown, very cluttered environments is a difficult task in mobile robotics. Behavioral approaches have been shown to be very successful in these environments [1], [2]. However, experimental results have shown that behavioral systems that are goal directed (i.e., systems in which one of the behaviors is goal seeking) tend to fail when the robot enters an enclosure that has its opening in a direction opposite to the goal. In scenarios like this, behavioral systems often get trapped in deadlocks known as limit cycles, where the robot retraces the same path indefinitely, thereby failing to accomplish its mission of reaching the goal.

One way of escaping these deadlock enclosures has been by inclusion of deliberative algorithms that use a stored map of the environment to detect and retract from the deadlock regions. While these strategies have been effective in some applications, many of them have been computationally demanding, resulting in a slow retraction from the deadlock enclosures. Other approaches add randomness to the robot behavior as a strategy to solve some limit cycles. However, these approaches are limited to simple scenarios.

This paper presents a relatively simple algorithm that uses local maps and virtual sensors to solve the problem of deadlock detection and avoidance. This algorithm is used in conjunction with the recent fuzzy preference-based behavioral system for navigation in cluttered environments [2].

The paper is organized as follows. Section 2 states the limit cycle problem. Section 3 presents the proposed strategy to detect and retract from limit cycles. Section 4 presents simulation results. Section 5 describes experimental results that show the performance of the proposed algorithm on a laboratory robot. Finally, Section 6 gives concluding remarks.

2. The Limit Cycle Problem
As pointed out earlier, since behavioral systems are goal directed and rely only on the current sensory information, they may...
sometimes lead the robot into limit cycles, i.e., infinite loops in which the robot continually traverses the same path. Figure 1 illustrates a typical example of a navigational limit cycle.

The robot's trajectory is shown and the target destination is represented by a circle outside the enclosure that is causing the limit cycle. In Figure 1, the limit cycle is described by:

\[ A \rightarrow B \rightarrow A \rightarrow C \rightarrow A. \]

This problem of limit cycles in navigation has continually attracted attention of researchers in the robotics community. Some important approaches to solve the limit cycle problem are presented in [3], [4], [5], [6], [7]. The strategy of [3] keeps track of visited cells and causes the robot to avoid going to cells that have already been visited. This approach is often successful. However, it lacks a time-decay mechanism, which can result in the robot "boxing itself in." The wall following behavior of [4] tracks the contours of the obstacle in order to skirt around it; this approach is presented as a possible solution for indoor environments. The virtual target approach of [5] defines a virtual target once a limit cycle is detected. The robot is navigated according to the virtual target, until a switch back condition is reached. The condition for switching back to the real target is based on the detection of an opening of the deadlock enclosure to the right or to the left of the robot. This switching back condition does not always occur in a cluttered environment where the enclosures are not continuous. The approach of [6] adds memory and memory related behaviors to basic reflexive systems as a possible solution to limit cycles. In addition, a virtual rectangular obstacle is placed over the area where the limit cycle was detected. However, in a cluttered environment designating this rectangular area as an obstacle will cause the robot to avoid navigable area. The research of [7] proposes a possible solution to the local minima problem by classifying the environment based on spatio-temporal sequences. This approach is particularly designed for indoor environments where walls form the enclosures.

Figure 1. Navigational Limit Cycle in a Box Canyon

3. PROPOSED APPROACH

This section describes a new simple deliberative approach for detection and avoidance of navigational limit cycles in cluttered indoor or outdoor environments with various sizes and shapes of potential deadlock regions. The basic flowchart of the proposed algorithm is given in Figure 2. The strategy was conceived trying to emulate the reasoning of a human being in a similar situation. Imagine a human in a very cluttered or maze-like environment seeking to reach a goal that is at a known location. As this person moves towards the goal (say using a compass), she may find herself in an enclosure that has only one exit, which is the original entrance. Hence, she would change her destination to a point outside this exit. Once outside the enclosure, she would remember not to enter that enclosure again, effectively creating a virtual wall. Then, she would proceed to her destination using a different path. Similarly to the "avoid the past" strategy of [3], this approach keeps track of places visited by the robot in a spatial memory. However, this new approach uses the spatial memory to detect limit cycles and not to compute repulsive forces. Furthermore, the avoidance of the limit cycle region is carried out by building a virtual wall to close the deadlocked area, instead of generating a repulsive force on each visited cell.

The algorithm is divided into three major components: Limit Cycle Detection, Obstacle Field Mapping, and Retraction from the Deadlock Enclosure. Each of these is described in some detail below.

3.1 Limit Cycle Detection

In order to detect navigational limit cycles, the robot must keep a memory of visited places. Dynamic spatial memory is employed to keep track of recently visited places. A virtual grid is defined over the robot workspace so that, as illustrated in Figure 3, a two dimensional array of integers corresponding to this virtual grid is stored in the computer memory to represent the memory of visited places. These integers serve as counters for the number of robot visits at the corresponding area in the workspace. The robot marks each cell in this virtual grid by increasing its corresponding integer whenever it passes over a point in the workspace that corresponds to the cell in the virtual grid. The grid size is given by \( l = w/(2\sqrt{2}) \) where \( w \) is the width of the robot, and \( l \) is the grid size.

Figure 2. Deadlock Detection and Avoidance Scheme
The Limit Cycle Detection stage is represented in Figure 2 by the block entitled Spatial Memory, which receives the robot position information from the localization unit.

A threshold number of visits is predefined such that whenever the number of visits to one cell in the spatial memory exceeds this threshold, a limit cycle is said to have been detected. Due to the size of the grid of the spatial memory, the position update rate, and the speed of the robot, the counter associated with a cell may be incremented more than once as the robot passes over that cell. Figure 3 illustrates the appearance of this array after a robot mission is accomplished. The values of the counter associated to each grid are shown. The threshold level was set to 26.

### Figure 3. Spatial Memory Appearance after the Threshold Level has been Reached

<table>
<thead>
<tr>
<th></th>
<th>5</th>
<th>19</th>
<th>20</th>
<th>16</th>
<th>16</th>
<th>1</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>23</td>
<td>17</td>
<td>23</td>
<td>24</td>
<td>26</td>
<td>21</td>
<td>19</td>
</tr>
<tr>
<td>7</td>
<td>15</td>
<td>5</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

3.2 Obstacle Field Mapping

As the robotic vehicle navigates through the environment, in addition to tracking visited grids, it also builds a record of obstacles observed in each grid. This record is stored in a grid as a virtual map of obstacles and is eventually used in determining a strategy for retracting from the limit cycle as described in the next subsection. A typical example of a map of an obstacle field is shown in Figure 4.

![Figure 4. Map of Obstacles in Gray Scale](image)

The gray level of each cell represents the number of times that an obstacle in that cell has been seen by the robot. The robot is represented by the square marked R.

![Figure 2 shows that the obstacle map construction requires both localization information and the relative position of the obstacles from the laser range finder.](image)

3.3 Retraction from the Deadlock Enclosure

Once the threshold level has been reached in the spatial memory, a way-point is defined outside the deadlock. The robot uses this way-point as a temporal goal so that it can safely leave the limit cycle enclosure using the behavioral control system. To define the way-point, the virtual map of the obstacle field and the spatial memory are processed by using the connected components algorithm, known as "The Classical Algorithm" [8]. This type of algorithm is commonly used in image processing applications. In this approach, the connected components algorithm is used to identify and label the connected components of the obstacle field.

As clarified in the next subsection, the Virtual Map is labeled using connectivity 8 and connectivity 4 criterions. Once the connected components have been identified, a way-point is chosen outside the deadlock and the robot is directed towards it.

The fusion of the spatial memory and the map of obstacles is represented in Figure 2 by the block named Fusion. The outputs of this block are the way point and the virtual wall coordinates. The way-point is simply used to navigate the robot to the outside of the enclosure; however the robot does not have to reach this point. The robot continues navigating towards the way-point until it realizes that its current position is outside the enclosure and it has enough room to make a turn without going back into the deadlock enclosure.

After traveling at a safe distance from the enclosure, the robot builds a virtual wall, essentially "closing the door" to the deadlocked area in the virtual obstacle map. A virtual laser range finder is used as an additional sensor to detect this virtual obstacle. This provides sensory information that enables the vehicle to avoid returning to the deadlock enclosure. Figure 2 shows that the virtual laser computes the distance to the virtual wall defined in the fusion block. The virtual laser is fused with the real laser range finder in the block named Sensor Fusion.

3.4 Illustration of the Calculation of the Virtual Wall and Way Point

The entire process is clarified below by using a simple example. For a more detailed explanation, refer to [9]. The numbers enclosed by squares in figures 5-8 correspond to the same spatial location.

Assume the matrix of Figure 5 represents the map of obstacles where 1 represents an obstacle that does not belong to the enclosure. This virtual map is treated as image data that is labeled using connectivity 8 and connectivity 4 criterions to identify different connected obstacle regions, [8]. For the obstacle field of Figure 5, the labeling process, using connectivity 8 criterion, identifies four obstacle regions as shown in Figure 6; these connected regions are marked by integers 1, 2, 3 and 4. The elements labeled as 1 belong to one connected region (Region 1), elements labeled as 2 belong to another connected region (Region 2), and so on. Now, based on the identified obstacle fields, a bounding box that contains the deadlock is determined.
This box is defined by the extreme points that make up the region identified as the deadlock, which happens to be the obstacle region that has the most elements. In this case the deadlock is defined by the extreme points of Region 2 in Figure 6.

Once the bounding box for the deadlock enclosure is created, the size of the virtual map of the obstacle field is trimmed down to remove any obstacle information that is outside the deadlock bounding box. This trimming is carried out in order to allow a short term navigation process whose only objective is to retract from the deadlock.

The trimmed virtual map of obstacles is then fused with the corresponding local spatial memory of visited places. This fusion helps to determine the location where the deadlock was first detected. The mark $\text{X}$ in Figure 7 indicates the location where the deadlock was identified.

In order to find an appropriate location for the virtual wall, a binary inverse of the trimmed image is computed and the resulting matrix is labeled using connectivity 4. The points on the sides of the bounding box that have the same label as the grid where the threshold was reached (label 1) are identified. These points correspond to the opening of the deadlock enclosure, and they are used to determine the location of the virtual wall. They are marked $\text{1}$ in Figure 8.

To choose the way point there are two possibilities, depending on whether the robot entered to the deadlock enclosure or whether it was already inside when the robot began its mission. In the first case, the way point is chosen at the location where the robot entered the enclosure. By doing this, it is possible to guarantee that the way point is a reachable target. In the second case, a way point is defined at a predefined distance from the virtual wall. However, the robot does not need to reach this way point. The robot switches back to its original target when it is at a safe distance from the virtual wall. In the current implementation, the second option is used. Once the way-point is found, a virtual wall is placed near the end points of the deadlock so that the robot does not go back into the limit cycle.

As previously discussed, this architecture is equipped with a virtual laser so that the robot can detect the position of the virtual walls and avoid them as if they were common obstacles. The actual input to the behavioral control algorithm is the minimum distance returned by the real and the virtual sensors.

4. SIMULATION RESULTS

The proposed method was simulated using cluttered environments that emulate dense forests. Two simulation scenarios are shown in Figures 9 through 11. Scenario 1 is shown in Figure 9, this scenario shows results when the limit cycle detection and avoidance ability is not engaged, while Figure 10 considers the case in which this ability is engaged. As expected, the robot in Figure 10 was able to successfully reach the goal while the robot in Figure 9 became trapped in a limit cycle. Scenario 2, shown in Figure 11, illustrates results obtained with the limit cycle negotiation ability when the robot encounters an enclosure with a totally different orientation.
5. EXPERIMENTAL RESULTS

After satisfactory simulation performance, the proposed strategy for limit cycle negotiation was implemented and tested in a laboratory environment on a Pioneer 2 robot equipped with a SICK laser range finder (Figure 12). This robot, which is manufactured by ActivMedia Robotics, is a differentially driven platform configured with two drive wheels and one swivel caster for balance. Each wheel is driven independently by a motor with 19.5:1 gear ratio, which enables the robot to drive at a maximum speed of 1.2 m/s and climb a 25 % grade [10].

5.1 Pioneer 2 Sensors

The Pioneer 2 is equipped with several types of sensors. The navigation control system requires range sensors and localization sensors. This subsection will give a brief description of the range and localization sensors used in this implementation.

5.1.1 Range Sensors

Two types of range sensors were available: a bank of sonar sensors as well as a laser range finder. Because of its better accuracy and resolution, the laser range finder was used in this control system. The laser range finder used is a SICK LMS 200; it has a resolution of 10mm, a typical measurement accuracy of ± 15mm, a 180° scanning angle, and 10m typical measured distance range [11]. Measurements can be made for scan angles as small as 0.25° that can be composed into rectangular and cone shaped regions [11].

5.1.2 Localization Sensors

Typical localization sensors include Global Positioning Systems (GPS), Inertial Navigation Systems (INS) and electronic compasses. The Pioneer 2 robot used in these experiments had none of these localization sensors. Instead, localization information was achieved computationally by using wheel encoders. Each motor on the mobile robotic platform was equipped with a 500 tick encoder [10], which measures the change in orientation of the motors in increments of 1/500 of a rotation. This information along with the drive gear ratio provided the change in orientation of each wheel, which was differentiated to provide wheel velocities. The obtained wheel velocities were used in the calculation of the position of the vehicle relative to its initial position, and hence localization was achieved.

5.2 Observed Performance

This subsection presents a sample of the experimental results that show the performance of the proposed strategy used in conjunction with the recently developed preference-based fuzzy behavioral control system [2]. In each experiment, the threshold level was set to 6, and a grid of 30 cm by 30 cm was used for the local maps.

5.2.1 Obstacle Description and Configuration

A dense forest in which trees become obstacles to robot motion was chosen as the experimental environment. Such obstacles are very difficult to navigate through because they are relatively small with irregular spacing. Trees were simulated by 2' long by 2", 3", and 4" diameter PVC pipe sections. These pipe diameters scale appropriately to the vehicle size and accurately depict the trunks of trees. Figure 12 shows the Pioneer robot navigating through the experimental environment.
5.2.2 Results

For each experimental scenario the robot was set at a particular start point and the goal was defined in either Cartesian or polar coordinates with respect to the start position. The obstacle configurations were mapped and the localization data (X,Y) of the robot were plotted as shown in Figures 13 through 15. The path followed by the robot during the detection of the limit cycle is represented by a solid line. The retraction stage is represented by a dashed line. It is important to note that in order to record the path followed by the robot a marker was attached to its rear. Therefore, the (X,Y) data correspond to the position of the rear of the robot and not to its center. The mark X indicates the location of the way points and the final goal. The discrepancy between the goal and the actual final position of the robot is due to localization error, which can be improved by adding additional localization sensors to the robotic vehicle.

Scenario 1 (Figure 13) shows how the strategy works when the robot enters a deadlock enclosure in a cluttered environment. Notice that the starting position of the robot is outside the enclosure.

Scenario 2 (Figure 14) shows that the robot switches back to its original goal even if the way point is an unreachable target. The way point is classified as unreachable because it is very close to a group of obstacles, and in that situation the obstacle avoidance behavior would not allow the robot to reach it. However, at the point marked S, the robot was found to be at a safe distance from the virtual wall and therefore it switched back to its original goal.

Scenario 3 (Figure 15) shows that the proposed strategy also works when the robot starts its mission from within the deadlocked area. In this scenario, the importance of the virtual wall can be seen more clearly. When the robot switched back to its original target, the robot headed directly to the enclosure again but thanks to the virtual wall and the virtual laser, the robot was able to avoid the deadlock enclosure.

6. CONCLUSION

A method for detecting and avoiding navigational limit cycles, which is one of the major shortcomings of behavioral systems, has been proposed. This method uses dynamic local spatial memory to keep track of places visited in order to detect limit cycles. Whenever, a point in spatial memory records that one place has been repeatedly visited in excess of allowable levels, a limit cycle is said to have been detected. Once the limit cycle is detected, a virtual wall is built to close the deadlocked area so that the robot will not be able to enter that region again. The proposed method has been tested in very cluttered environments, both in simulation and experimentally, using a Pioneer 2 robot equipped with a SICK laser range finder.

Current research involves the inclusion of new features for this strategy. First, in order to improve the accuracy of the enclosure detection process, it is important to use information regarding the number of times the obstacles that form this enclosure have been scanned by the laser and not simply rely on the size of the enclosure. In addition, a dynamic threshold dependent on the speed of the vehicle is desired.
ACKNOWLEDGMENTS AND DISCLAIMER
This document was prepared through collaborative participation in the Robotics Consortium sponsored by the U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD 19-01-2-0012. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon. However, the views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U. S. Government.

7. REFERENCES


