CHAPTER 6

DEVELOPMENT OF HEURISTIC ALGORITHMS FOR ROBOT PATH PLANNING

6.1 OVERVIEW

The basic principles of reinforcement learning algorithm, potential field and wave front algorithm is described briefly in the following sections. The box pushing with touch sensors, path planning with obstacle avoidance with IR and RF sensors and path planning with obstacle avoidance with computer vision are proposed to be performed using simplified reinforcement learning, potential field and wave front algorithm respectively using the mobile robots fabricated in-house.

6.2 REINFORCEMENT LEARNING ALGORITHM FOR BOX PUSHING TASK WITH TOUCH SENSORS

Reinforcement learning is one type of robot learning algorithm widely used to develop cooperation among the mobile robots to perform certain task. Reinforcement learning has been applied to learn new behaviours and to coordinate the existing ones. Reinforcement learning systems attempt to learn behaviour by exploring all the actions in all the available states and rank them in the order of appropriateness. It allows the robot to automatically determine the ideal behaviour within a specified context, in order to maximize its performance. Simple reward feedback is required for the mobile robot to learn its behaviour known as the reinforcement signal. Reinforcement learning systems attempt to learn behaviour by exploring all the actions in all
the available states through trial-and-error and rank them in the order of appropriateness. It uses only the reward function values in order to control the drive system of the robot by the controller. A component capable of evaluating the response is needed to send the necessary reinforcement signal to the control system. This component can be a human, observing the robot or a software module programmed to evaluate the robot’s actions. The first is called supervised learning, the latter is unsupervised learning. The feedback to the control system provides information about the quality of the behavioural response. Existence of mobile robot in a working environment can be summarized by a set of states. The robot can make observations from a set (O). Based on these observations, it can take any one of actions from a specified set of actions (A). It includes an additional input (R) from the environment to the robot through sensor signals. This is an immediate reward value from the environment and it represents a measure of the last action. The ultimate goal of reinforcement learning is to learn a policy based on the state of environment. Policy is mapping the observations to an action, which is to be performed by the mobile robot. The basic architecture of the proposed reinforcement learning algorithm is shown in Figure 6.1. An exclusive software code has been written in order to implement the reinforcement learning algorithm for the mobile robot.

![Figure 6.1 Basic architecture of reinforcement learning algorithm](image-url)
The flow chart for the box pushing task by reinforcement learning algorithm is presented in Figure 6.2. Initially, the first mobile robot will be started using move_forward function, immediately after receiving the target distance value from the user. The variable ‘a’ is used to store the observed value from the environment via three numbers of leaf switches provided on the front side of the mobile robot. Reward value of 1, i.e., R(1), will be assigned to the robot when it detects the box and reward value of zero, i.e., R(0), will be assigned in all other situations. The first mobile robot will stop immediately after receiving the reward value equal to 1. Then, the first mobile robot will send the signal to the second mobile robot. Immediately, the second mobile robot will started moving using move_forward function up to the location of the box similar to the first robot. Then, the reward function value of 1, i.e., R(1), for the second mobile robot is assigned. If the observed values of both mobile robots are equal to 1, it means the box is detected condition. Then, both the mobile robots will push the box simultaneously. This process is continued until the box reaches the target defined. Finally, the box pushing process will be stopped if the target distance is reached.
Figure 6.2 Flow chart for box pushing using reinforcement learning algorithm
6.3 POTENTIAL FIELD ALGORITHM FOR PATH PLANNING
WITH OBSTACLE AVOIDANCE TASK WITH IR AND RF SENSORS

Principally, the robot is represented as a point in space configuration as a moving particle under the influence of a signal produced by the sensors located on the target line (RF sensor) and located on the robot to detect the obstacles (IR sensor).

The goal always generates an attractive potential, which pulls the mobile robot towards the target and the obstacles always produces, a repulsive potential which pushes the robot away from them.

6.3.1 Attractive potential by target

The IR sensor is used to detect the obstacles and the RF sensor is used to identify the target. The attractive potential is the signal level from the RF sensor which logically attracts the mobile robot to the goal position and it has the relation given in Equation 6.1.

\[ \text{Attr} = \text{function of } \{\text{Ir, Rf}\} \]  \hspace{1cm} (6.1)

where,  \hspace{0.5cm} \text{Attr} = \text{attractor}

\( \text{Ir} \) = Signal from the IR sensor

\( \text{Rf} \) = Signal from the RF sensor

Example: (0, 1) and (1, 1)

6.3.2 Repulsive Potential by Obstacles

The repulsive potential is the potential which logically pushes the mobile robot away from them and it has the function shown in Equation 6.2.
Refl = function of \{Ir, Rf\} \hspace{1cm} (6.2)

where,

- **Refl** = Reflector
- **Ir** = Signal from the infrared sensor
- **Rf** = Signal from the RF sensor

Example: (0, 0) and (1, 0)

The above attractor and reflector functions are works based on the signals from the IR and RF sensors. The following combinations of functions are obtained using the attractive and repulsive potentials:

\[
\text{Attr} = \{Ir, Rf\} = [(0, 1) (1, 1)]
\]

\[
\text{Refl} = \{Ir, Rf\} = [(0, 0) (1, 0)]
\]

If,

- (0, 1) = Obstacle is not detected and target reached.
- (1, 0) = Obstacle is detected and target not reached.
- (1, 1) = Both obstacle and target are detected.
- (0, 0) = Obstacle and target are not detected.

The mobile robot can take decision based on the sensor input using above four possible combinations. If the repulsive potential is identified, the robot will take the move reverse function and moves in the reverse direction and then take turn left (or) turn right function based on random number generated. If the generated random number value is greater than 0.5, the robot will take forward right turn function, otherwise it will take forward left function. The same function is repeated until the mobile robot receives the signal from RF sensor, which is placed on the target line. An exclusive software code has been written in order to implement the potential field algorithm for the mobile robot.
The flowchart for the proposed simplified potential field algorithm for path planning with obstacle avoidance task is presented in Figure 6.3.
6.4 WAVE FRONT ALGORITHM FOR PATH PLANNING WITH OBSTACLE AVOIDANCE TASK WITH COMPUTER VISION

In the present work, the wavefront algorithm has been proposed to implement for path planning with obstacle avoidance task with computer vision. The wavefront algorithm involves breadth first search of an image of working environment beginning at the target position until it reaches the starting position. The image of working environment is captured using a camera available on the experimental setup and it is represented in a matrix form containing number of rows and columns wise cells shown schematically in Figure 6.4.

![Matrix Representation of Working Environment](image)

**Figure 6.4** Cell representation of a schematic image of working environment
All the cells on the image containing the obstacles are marked with number ‘1’ and the target point is marked with number ‘2’ excluding the starting position of the mobile robot. The adjacent cells with reference to the target cell in all four directions (left, right, up and down) will be assigned with a number 3. This number is called as distance value or wave number or potential value. Then a number ‘4’ is assigned to every cell adjacent to the cells with a distance value of ‘3’ and so on. Then the process of number assigning is continued until the starting position reached. During the process of assigning the wave numbers, those cells having value ‘1’ must be left as such. Once the wave is formed, the numbers must be read from the target position to the starting position. The mobile robot can move both diagonally or in any of the four directions (left, right, up and down). The fully wave formed representative model image of working environment is shown in Figure 6.5.

<table>
<thead>
<tr>
<th></th>
<th>2 Target</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td></td>
<td>1</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>7</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
</tbody>
</table>

**Figure 6.5**  Wave formed representation of schematic model image of working environment
6.5 STATISTICAL ANALYSIS

The positional deviation occurring during the box pushing or path planning tasks among the different runs are proposed to be checked using statistical methods like regression analysis. The reliability and stability of the proposed experimental procedures implementing through various tasks are proposed to check using analysis of variance (ANOVA) statistical test.

6.5.1 Regression Analysis

It studies the relationship among two or more variables. The relationships can be described and measured in a functional form. Generally, regression analyses are two types, simple and multiple regressions. The simple regression deals with only two variables and multiple regression deals with more than two variables.

a) Linear regression analysis

It is an estimation of the linear relationship between a dependent variable and one or more independent variables. If the relationship between the two variables is a linear function, then this linear function is called linear regression equation. The general linear regression relationship (Jerry Banks et al 2004) is given in Equation 6.3.

\[ Y = \beta X + \alpha \]  \hspace{1cm} (6.3)

where,  
\begin{align*}
X & \quad \text{Independent variable} \\
Y & \quad \text{Dependent variable} \\
\alpha & \quad \text{Standardized regression error} \\
\beta & \quad \text{Standardized regression coefficient}
\end{align*}
b) Non-linear regression analysis

It is a method of finding a non-linear model of the relationship between the dependent variable and a set of independent variables. Unlike traditional linear regression, which is restricted to estimating linear models, nonlinear regression can estimate models with arbitrary relationships between independent and dependent variables. This is accomplished using iterative estimation algorithms. This procedure is not necessary for simple polynomial models of the form \( Y = A + BX^2 \). By defining \( W = X^2 \), then it can be converted into simple linear model as \( Y = A + BW \), which can be estimated using traditional methods such as the linear regression analysis.

6.5.2 Analysis of Variance (ANOVA)

It is one of the statistical tools for testing the equality of more than two data sets of population (Srivastava and Shailaja Rego 2008, Arora and Arora 2008). The problem can be reduced to test the following null hypothesis:

\[
H_0 : \text{No Significance difference} \\
H_1 : \text{Significance difference}
\]

The standard ANOVA procedure can be used to perform ANOVA statistical test using Equations 6.4 to 6.13. The calculated \( F \) value will be obtained in this standard procedure and the corresponding critical \( F \) value will be found out using the standard \( F \) table. These two values are to be compared for making decision, whether the null hypothesis will be rejected or not rejected.

Degrees of Freedom (DF) of run = Total number of runs – 1 \hspace{1cm} (6.4)

Degrees of Freedom (DF) of total = Total number of observation – 1 \hspace{1cm} (6.5)
Degrees of Freedom (DF) of error = DF of total – DF of run \hspace{1cm} (6.6)

Correction factor (CF) = \( \frac{(\text{Sum of all observations})^2}{N} \) \hspace{1cm} (6.7)

Sum of squares of Run = \( \left[ \frac{\text{(Total of run-1 values})^2}{n} + \left[ \frac{\text{(Total of run-2 values})^2}{n} + \ldots + \left[ \frac{\text{(Total of run-n values})^2}{n} \right] \right] \right) \) – CF \hspace{1cm} (6.8)

Sum of squares of total = Sum of (squares of individual values) – CF \hspace{1cm} (6.9)

Sum of squares of Error = Sum of square total– Sum of square of run \hspace{1cm} (6.10)

Mean squares of Run = Sum of square of Run \( \div \) DF of run \hspace{1cm} (6.11)

Mean squares of Error = Sum of square of Error \( \div \) DF of error \hspace{1cm} (6.12)

Calculated F value = Mean square of Run \( \div \) Mean square of Error \hspace{1cm} (6.13)

where, \hspace{1cm} N \hspace{0.5cm} - \hspace{0.5cm} Total numbers of observation.

\hspace{1cm} n \hspace{0.5cm} - \hspace{0.5cm} Total numbers of observation in a single run.

\hspace{1cm} CF \hspace{0.5cm} - \hspace{0.5cm} correction factor.

\hspace{1cm} DF \hspace{0.5cm} - \hspace{0.5cm} degrees of freedom.

\hspace{1cm} R_i \hspace{0.5cm} - \hspace{0.5cm} Run number. \hspace{0.5cm} i = 1, 2, 3, \ldots

6.5.3 Mean, Standard Deviation and Standard Error

Apart from regression analysis, it is worth to find out other statistics like mean, standard deviation and standard error for the generated in each task performed in order to study the variations in an observed data. Mean (M) is the arithmetic average of the data observed. It will act as reference value for finding other statistics. Standard deviation (SD) is an index of how closely the individual data points cluster around the mean value of the data set. Standard error (SE) is the variability of the means (David 1996, Averill and David Kelton 2000). The mean, SD, SE and 95% CI relations are given in the Equations 6.14 to 6.17.
Mean (M) for X coordinate = Arithmetic average of observed Xi value (6.14)
Mean (M) for Y coordinate = Arithmetic average of observed Yi value (6.15)

Standard Deviation (SD) for X coordinate = \( \sqrt{\frac{\sum (Xi-M)^2}{N-1}} \) (6.16)

Standard Deviation (SD) for Y coordinate = \( \sqrt{\frac{\sum (Yi-M)^2}{N-1}} \) (6.17)

Standard Error (SE) = \( \frac{SD}{\sqrt{N}} \) (6.18)

95% Confidence Interval (CI) = M ± (1.96 * SE) (6.19)

where, Xi - Observed data point in a run. i – 1, 2, 3,....
M - Mean of a data set.
N - Total number of data or observation.
SD - Standard Deviation
SE - Standard Error
CI - Confidence Interval