Autonomous Systems LAB Report
Autonomous navigation

Name: Melaku Yemaneberhan MESIHU
ID No.: 42403002
MARS 2024/2025

To: *Dr. Lara BRIÑÓN*

1.1 Artificial Potential Field navigation strategy is influenced by two types of potentials: attractive and repulsive. **Attractive potential** that pulls the robot toward the goal or a target location. The strength of the attractive potential is higher when the robot is farther from the goal and decreases as it gets closer. So, this component encourages the robot to move towards the desired destination. The other is **repulsive potential**, which prevents the robot from colliding with obstacles in the environment. The strength of the repulsive potential is higher when the robot is close to an obstacle. So, this component discourages the robot from getting too close to obstacles, helping it navigate around the obstacle.

1.2 $U_{att}(P) = \|P - P_{goal}\|$

- This function linearly depends on the distance to the goal.
- Attractive force" pulling the robot toward p_{goal} does not change with distance. This may lead to less "fine" control when the robot is very close to the goal.
- The linearity may cause the robot to approach the goal more directly without slowing down as it gets closer.

$U_{att}(P) = \|P - P_{goal}\|^2$

- This function grows quadratically with the distance from the goal, creating a steeper increase in potential as the robot moves further from the target.
- The attractive force decreases as the robot nears the goal, allowing for a smoother approach.
- This function provides more control as the robot approaches the goal, causing it to slow down as it gets closer, reducing the risk of overshooting.

❖ The quadratic potential $U_{att}(P) = \|P - P_{goal}\|^2$ is generally the preferred choice for attractive navigation in robotics, as it provides a smoother approach and more stable control when nearing the goal.

1.3 The function $U_{rep}(p) = \frac{1}{\|P - P_{obs}\|}$ is suitable as a repulsive potential because:

- **Strong Repulsion Near Obstacles**: As the robot approaches the obstacle, $U_{rep}(p)$ increases rapidly, pushing it away with greater force to prevent collisions.
- **Minimal Effect at Distance**: When the robot is far from the obstacle, $U_{rep}(p)$ is small, allowing efficient movement without unnecessary deviation.

This ensures a safe yet efficient path, as the robot is repelled strongly near obstacles but moves freely when distant from them.

1.4 Then the gradient of the attraction potential is calculated as

$$\nabla U(P) = \nabla\left(\|P - P_{goal}\|^2\right) + \nabla\left(\frac{1}{\|P - P_{obs}\|}\right)$$

$$= \frac{2\left\|P - P_{goal}\right\|\left(P - P_{goal}\right)}{\left\|P - P_{goal}\right\|} + \frac{-1}{\|P - P_{obs}\|^2}\frac{(P - P_{obs})}{\|P - P_{obs}\|}$$

So,

$$\nabla U(P) = 2\left(P - P_{goal}\right) - \frac{1}{\|P - P_{obs}\|^2}\frac{(P - P_{obs})}{\|P - P_{obs}\|}$$

By taking the negative gradient $\left(-\nabla U(P)\right)$ i.e. the model dynamics, which points in the direction of the steepest decent of the total potential energy function the robot can move in the direction of the negative gradient to minimize the potential energy and approach the goal.

$$\dot{P} = -\nabla U(P)$$

When discretizing the above equation, we have $P_{t+1} = P_t - \nabla U(P_t)$, this means we are computing the gradient at every sample time t.

1.5 If the obstacle is a 2-dimensional circular shape centered at $P_{obs}$with a radius R_{obs}, the distance vector from the robot's position P to the nearest point on the obstacle's perimeter can be computed as follows:

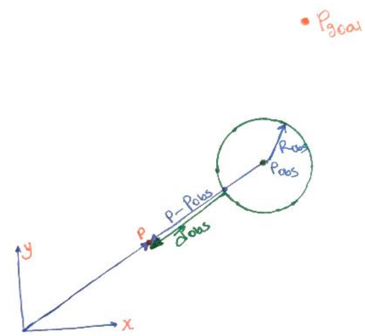

Figure 1 when the obstacle is considered as circular

First let's find a vector that points from the robot to the obstacle center

$$P_v = P - P_{obs}$$

Then in order to find the robots nearest point on the obstacle perimeter we have to use the unit vector direction of P_v so, the unit vector is given by $U_{center} = \frac{P - P_{obs}}{\|P - P_{obs}\|}$, then now the nearest point to the obstacle perimeter is in the direction of the unit vector U_{center} and is given by $d_{obs} = U_{center}(\|P - P_{obs}\| - R_{obs})$ there for this new vector have a magnitude of($\|P - P_{obs}\| - R_{obs}$) and the direction is always pointing towards the robot.

$$d_{obs} = \frac{P - P_{obs}}{\|P - P_{obs}\|}\,(\|P - P_{obs}\| - R_{obs})$$

1.6 To update the equation of Artificial potential field $U(P)$ to consider this, we will need to modify the repulsive component of the total potential term, and it is given as:

$$\nabla U(P) = 2\left(P - P_{goal}\right) - \frac{1}{\|P - d_{obs}\|^2}\frac{(P - d_{obs})}{\|P - d_{obs}\|}$$

Autonomous Systems LAB Report
Autonomous navigation

Question 2: Autonomous navigation for a single integrator robot

First, we consider a robot with single integrator dynamics, i.e., the robot can move in any direction and we can control directly its velocity. The dynamic model of the robot is expressed as:

$$\dot{x} = u_1$$
$$\dot{y} = u_2$$

Where: $U = [u_1\ u_2]$ is the control input.
For this section the aim is to implement your navigation strategy and controller for the single integrator robot.

Question 2.1-2.2: Complete the code of the function robot dynamics.m to implement the single integrator dynamics of the robot.

To study and analyze the artificial potential field approach in MATLAB we have to implement the proposed navigation strategy in discrete time. To do so, the position of the robot at each step must be updated according to its dynamic model. If the sampling time is *dt*, the equation to update the position of the robot at each step *k* is implemented as:

```
%robot model: simple integrator
 x_dot=[0;0];
x_dot(1)=X_state(1)+dt*u(1);
x_dot(2)=X_state(2)+dt*u(2);
```

Question 2.3 To help you visualize the generated potential field the function draw field.m is provided. Explain how this function works and the relation with the artificial potential field you proposed in Question 1.
In general, this function visualizes the artificial potential field and its gradient vectors by providing insights into how the attractive and repulsive forces influence the motion of a robot in the specified environment.

The function initializes a grid of points within the specified limits using meshgrid. The attractive force towards the goal is calculated using the gradient of the potential field, and if there is an additional obstacle (p_obs2), its repulsive force is added to the gradient vectors. and after that, the gradient vectors are normalized to ensure consistent visualization. Finally, the quiver function is used to plot the vector field, where each arrow represents the direction and magnitude of the force at a particular point in the grid.

Question 2.4 In order to guide the robot to the goal while avoiding an obstacle, design a control law for the single integrator robot by defining a velocity vector reference to be followed, based on the artificial potential field that you proposed in Question 1.
Simulate your system with the following values:
– position of the static obstacle p obs=[4;4]
– position of the goal p goal=[6;6]
– robot’s state initial conditions x=[2;1]
– potential function parameters alpha=1 and beta=1

Autonomous Systems LAB Report
Autonomous navigation

Analyze the robot behavior when the position of the obstacle changes. Illustrate your results with a figure showing the initial and final state of the robot, the trajectory of the robot and the positions of both the goal and the obstacle.

```
% ponderation parameters
alpha=1;
beta=1;

% total gradient = desired velocity vector
grad_U=alpha*grad_Uattr+beta*grad_Urep;

%control law
u(:,k)=- grad_U;

% robot state update using its dynamics
xdot=[0;0]; % use the function robot_dynamics to compute f
X_state(:,k)=robot_dynamics(X_state(:,k-1),u(:,k),dt,'integrator');
```

The robot will accelerate faster at the start position, and since the obstacle has repulsive potential, it will be pushed away from it, and finally, it will decelerate when it is near the goal position.

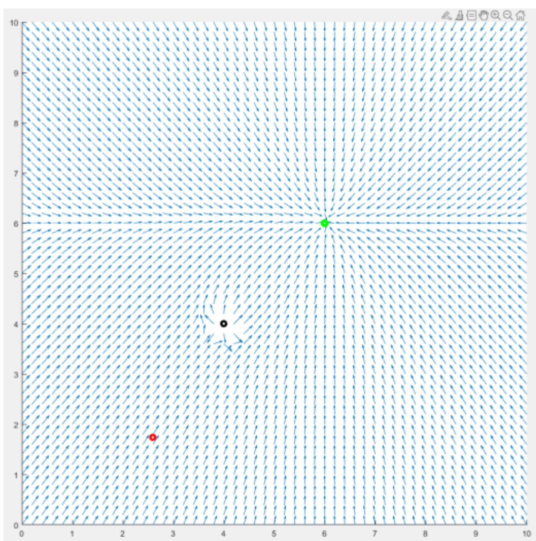

Figure 1: Mapping Robot's Position Relative to Obstacles, Goal and the gradient field

- Initial position of the robot with a red circle, the obstacle with black, the goal with green and the gradient vectors at each point.

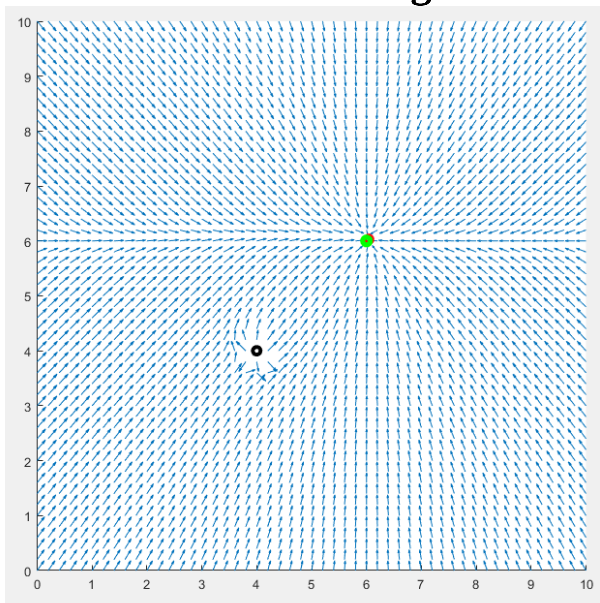

Figure 2: Mapping Robot's Position Relative to Obstacles, Goal and the gradient fie

- Shows final position of the robot after reached the goal

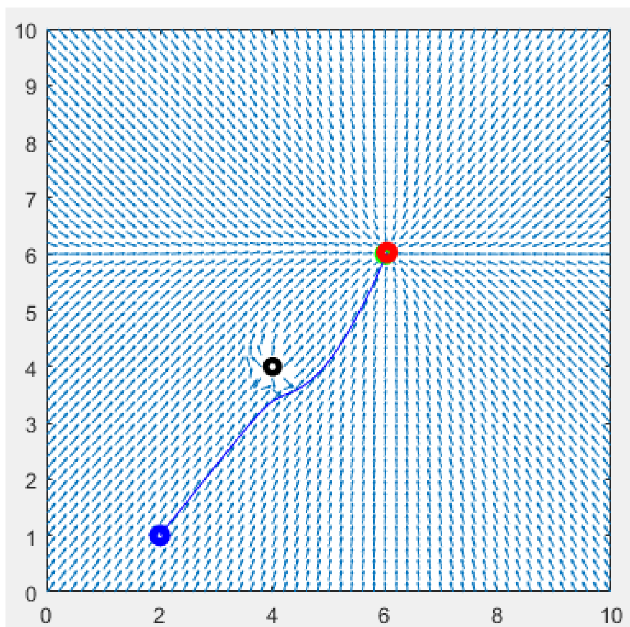

Figure 3: The trajectory of the robot when the obstacle is at (4,4)

The robot initially moves in a straight line toward the goal, which serves as the zone of attraction. When it approaches an obstacle and encounters a repulsive effect, it adjusts its path. Specifically, the robot veers to the right of the obstacle, selecting the shortest route to bypass it, before resuming a straight-line trajectory toward the goal.

Autonomous Systems LAB Report
Autonomous navigation

Question 3: Autonomous navigation for a non-holonomic robot

3.1 The state space of the robot is defined as $X_{state} = [x, y, v, \theta]^T$. Modify the code of the function robot dynamics.m to implement the unicycle dynamics of the robot.

I modify the code of the function robot_dynamics.m in the option ‘unicycle’ the robot’s state and control input.

3.3

I simulate the new system with the following values:

- ✓ position of the obstacle p_obs=[4;4]
- ✓ position of the goal p_goal=[6;6]
- ✓ robot’s state initial conditions X state=[2;1;1;π/4]
- ✓ potential functions parameters alpha=1 and beta=1

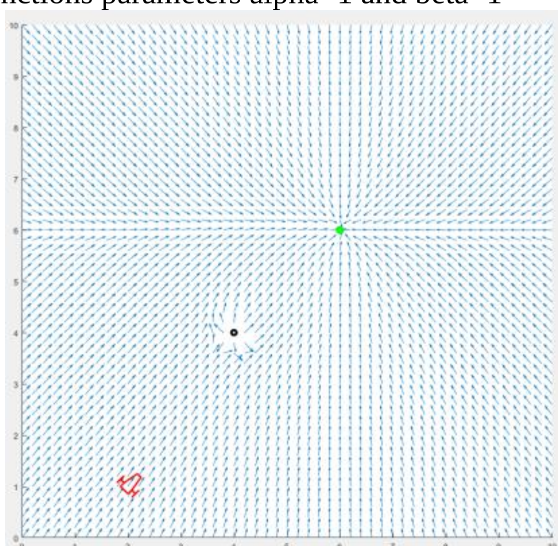

Figure 4: Drawing Robot's Position Relative to Obstacles, Goal and the gradient field

At the initial position

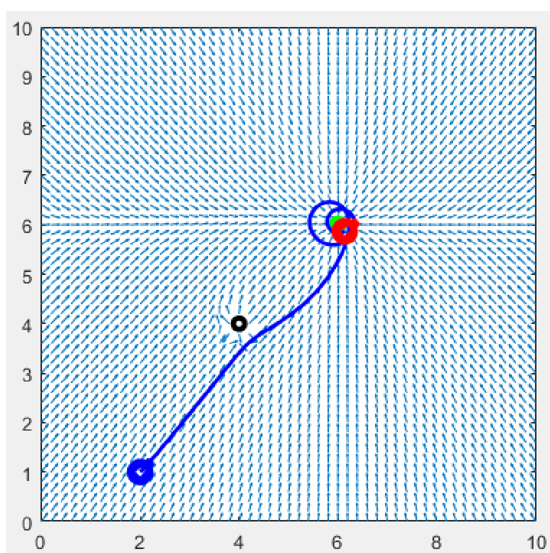

Figure 5: Robot Navigation Trajectory

- Illustration of the robot's path from its initial position to the final goal while avoiding obstacles.

The robot follows the same trajectory as before, but as it approaches the goal, it begins to circle around it before coming to a stop. This behavior arises from the rapid changes in the directional vector, which cause frequent variations in the angle theta. Consequently, the robot rotates while gradually converging toward the goal.

3.4

When the parameter β (representing the repulsive potential) is set to 20 times greater than α (representing the attractive potential), the robot prioritizes obstacle avoidance over reaching the goal. This results in the following behaviors:

- **Enhanced Obstacle Avoidance**: The robot significantly steers away from obstacles, often choosing a longer but safer route to the goal.
- **Weakened Goal Attraction**: The robot may face challenges in reaching the goal, especially if the obstacle is near the direct path.

The system is simulated using these parameter values:

– $\beta = 20 * \alpha$ with $\beta = 20$ and $\alpha = 1$
– position of the static obstacle p_obs=[5;5]
– position of the goal p_goal=[6;6]

– robot's state initial conditions X state=[2;1;1;π/4]

```
% Potential field parameters
alpha=1;
beta=20*alpha;
```

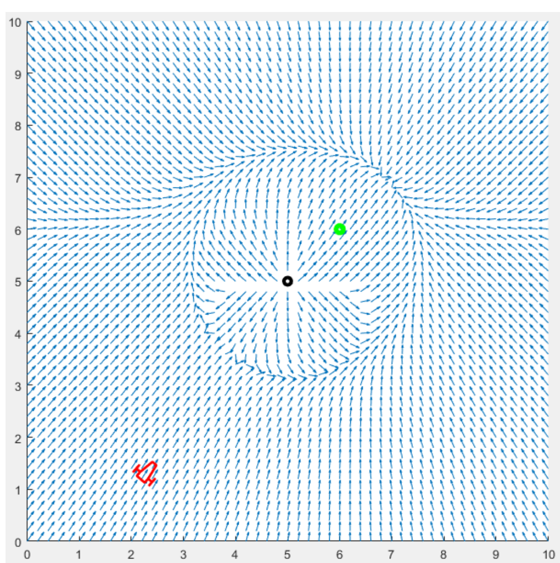

Figure 6: Robot's Position in Relation to Obstacles, Goal, and Gradient Field

Autonomous Systems LAB Report
Autonomous navigation

This figure illustrates the robot's initial position, the locations of the goal and obstacles, and the corresponding gradient field influencing the robot's movement at the specified time.

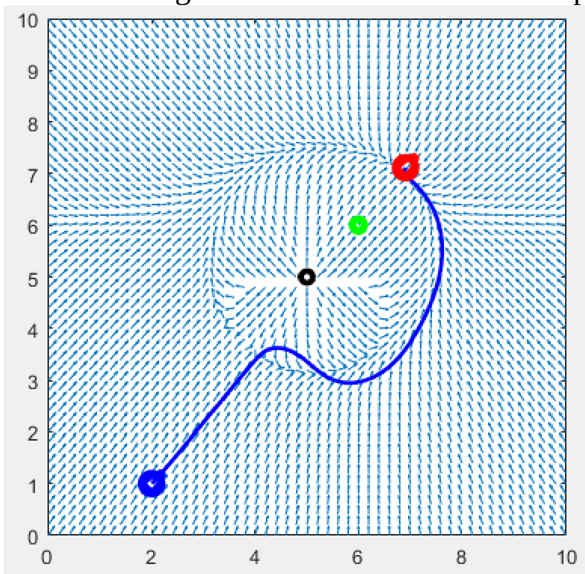

Figure 7: Robot Navigation Trajectory: Path from the initial position to the Final Goal avoiding the obstacles

This figure illustrates the robot's trajectory when β is increased to 20 times α. With the repulsive forces significantly stronger than the attractive forces, the robot avoids the obstacle from a much greater distance, resulting in a larger detour compared to **Figure 5**.

Due to the proximity of the obstacle to the goal, the repulsive forces remain dominant near the target, rendering the goal effectively inaccessible for the robot.

Question 4: More realistic scenario with circular shaped obstacles

4.1 To complete the function distance_obs.m, I calculate the distance vector from the robot's position to the nearest point on the obstacle. Here's a potential implementation, assuming the obstacle is represented by a circular region:

4.2 - Now I uses `distance_obs` function to compute vector to obstacle surface

- Repulsive gradient is based on distance to surface rather than center

- This creates a more accurate repulsive field around the circular obstacle

```
% scenario setup
p_goal=[6;6];
p_obs=[5;5];
R_obs=0.2;       % Radius of circular obstacle
p_obs2=[5;5];
R_obs2=0;
```

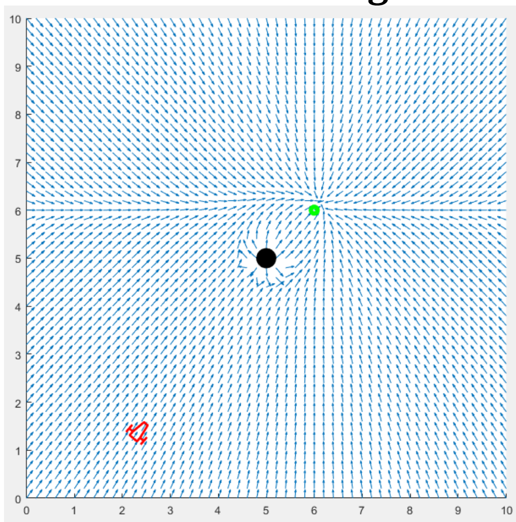

Figure 8: Robot's Position in Relation to Obstacles, Goal, and Gradient Field

– This figure depicts the robot's position along with the locations of the goal and obstacles, overlaid with the gradient field that dictates the robot's motion dynamics.

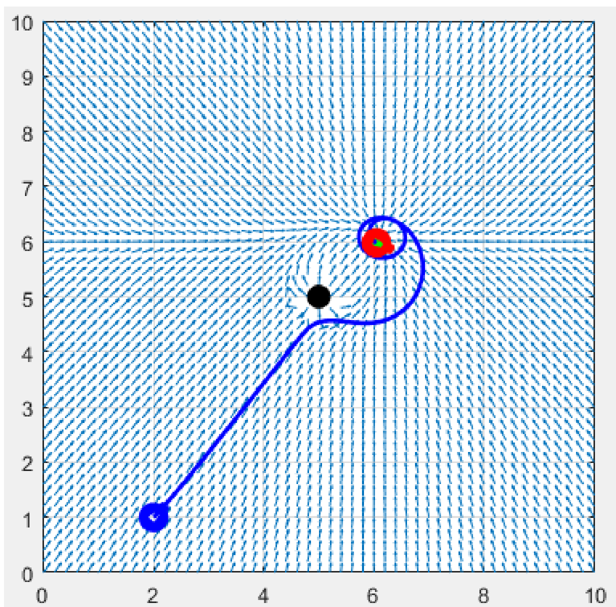

Figure 9: Robot Navigation Trajectory: Path from the initial position to the Final Goal avoiding the obstacles

This illustrates the trajectory of the robot as it moves toward its goal. Increasing the obstacle's radius leads to a corresponding increase in the avoidance radius, causing the robot to take a larger detour around the obstacle. Despite this adjustment, the starting and ending points of the trajectory remain unchanged.

Autonomous Systems LAB Report
Autonomous navigation

Effects of Different Radii

For R_obs = 0.2:

- Robot maintains safe distance from obstacle surface
- Path curves smoothly around obstacle
- Reaches goal successfully

For larger radii (e.g., R_obs = 0.5):

- Robot takes wider path around obstacle
- May require more time to reach goal
- Greater minimum distance maintained from obstacle center

For smaller radii (e.g., R_obs = 0.1):

- Robot passes closer to obstacle
- More direct path to goal
- Less deviation from straight-line path

Safety Considerations

- Larger radius creates larger safety margin
- Smaller radius allows more efficient paths
- Need to balance safety vs efficiency based on application

4.3

I simulate the system with the following values:
– position of the centers of both obstacles p_obs=[4;4] and p_obs2=[6;4]
– radius of the circular shaped obstacles R_obs=0.2 and R_obs2=0.2
– position of the goal p_goal=[6;6]
– robot's state initial conditions X_state=[2;1;1;π/4]

Autonomous Systems LAB Report
Autonomous navigation

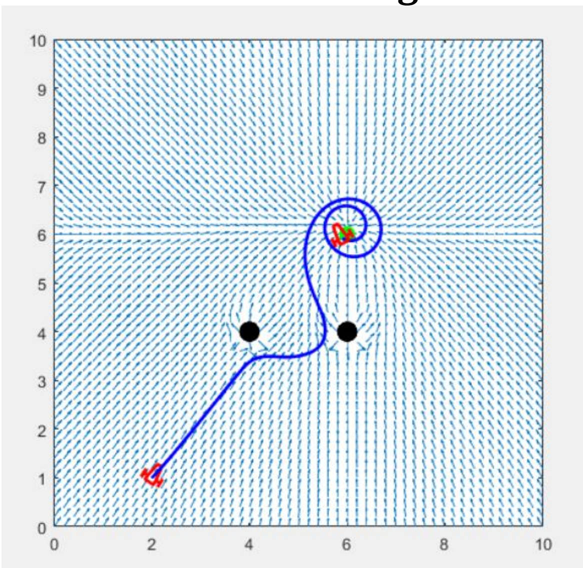

Figure 10: Robot Navigation Trajectory: Path from the initial position to the Final Goal avoiding the two obstacles

The robot's trajectory begins similarly to the previous case. However, upon approaching the first obstacle, it navigates around it by veering to the right. It then encounters the second obstacle, which it avoids by moving to the left. After bypassing both obstacles, the robot resumes its path toward the goal. As in earlier tests, it circles around the goal before finally stopping.

Question 5: Conclusion about the use of the Artificial Potential Field approach for autonomous navigation by analyzing its advantages and limitations.

The Artificial Potential Field (APF) approach is a popular method for autonomous navigation, as it provides a simple and intuitive way to guide a robot towards a goal while avoiding obstacles. The key advantages of this approach are:

1. Simplicity: The APF method is relatively straightforward to implement, as it only requires the definition of attractive and repulsive potential functions, which can then be used to compute the desired control inputs for the robot.

2. Real-time performance: The APF method is computationally efficient, allowing for real-time implementation and quick response to changes in the environment.

3. Obstacle avoidance: The repulsive potential function effectively pushes the robot away from obstacles, enabling it to navigate through cluttered environments.

4. Scalability: The APF approach can be extended to handle multiple obstacles and complex environments by simply adding more repulsive potential terms.

However, the APF approach also has some limitations:

Autonomous Systems LAB Report
Autonomous navigation

1. Local minima problem: The robot may get trapped in local minima of the potential field, where the net force is zero, preventing it from reaching the goal. This is a common issue that can be addressed using techniques like potential field shaping or the introduction of random noise.

2. Oscillations near obstacles: The robot may exhibit oscillatory behavior when navigating close to obstacles due to the strong repulsive forces. This can be mitigated by tuning the potential field parameters or using more sophisticated control strategies.

3. Difficulty in ensuring global optimality: The APF method does not guarantee the globally optimal path to the goal, as the robot's behavior is determined by the local gradient of the potential field.