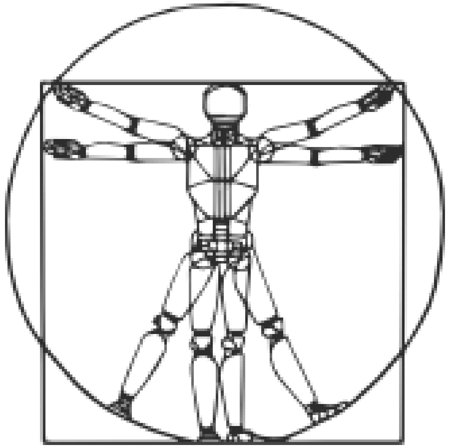
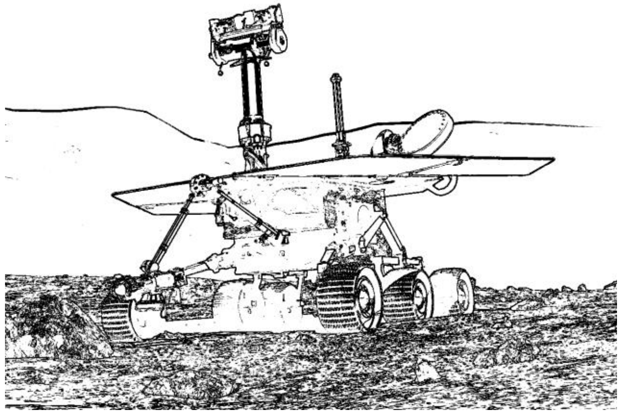




COMP417
Introduction to Robotics and Intelligent Systems

Lecture 6: Potential Fields & Obstacle Avoidance

Florian Shkurti

Computer Science Ph.D. student

florian@cim.mcgill.ca

McGill

MRL Mobile Robotics Lab at McGill University

Announcements

- A1 is up
- Linear algebra tutorial
- Probability tutorial

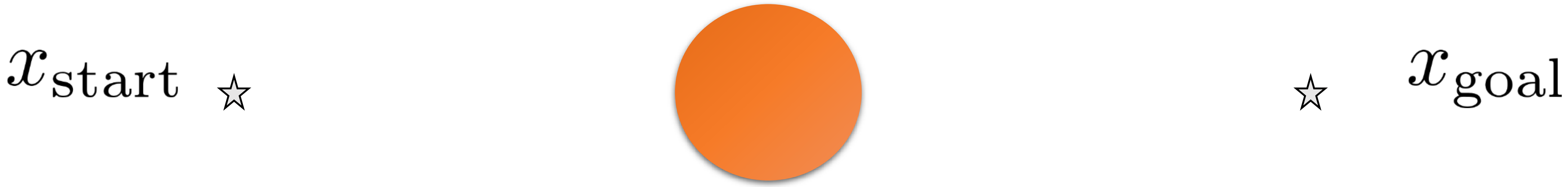
Recap: how to tune the PID

- After manual or Z-N tweaking you might want to use coordinate ascent to search for a better set of parameters automatically:

See Sebastian Thrun's online class "AI for robotics" on Udacity for more details on this. He calls the algorithm Twiddle and it is in Lesson 5.

```
# Choose an initialization parameter vector
p = [0, 0, 0] # [K_p, K_i, K_d]
# Define potential changes
dp = [1, 1, 1]
# Calculate the error
best_err = run_controller(p)

threshold = 0.001

while sum(dp) > threshold:
    for i in range(len(p)):
        p[i] += dp[i]
        err = run_controller(p)

        if err < best_err:  # There was some improvement
            best_err = err
            dp[i] *= 1.1
        else:  # There was no improvement
            p[i] -= 2*dp[i]  # Go into the other direction
            err = run_controller(p)

            if err < best_err:  # There was an improvement
                best_err = err
                dp[i] *= 1.05
            else  # There was no improvement
                p[i] += dp[i]
                # As there was no improvement, the step size in either
                # direction, the step size might simply be too big.
                dp[i] *= 0.95
```

This Week: Potential Fields
Main Motivation



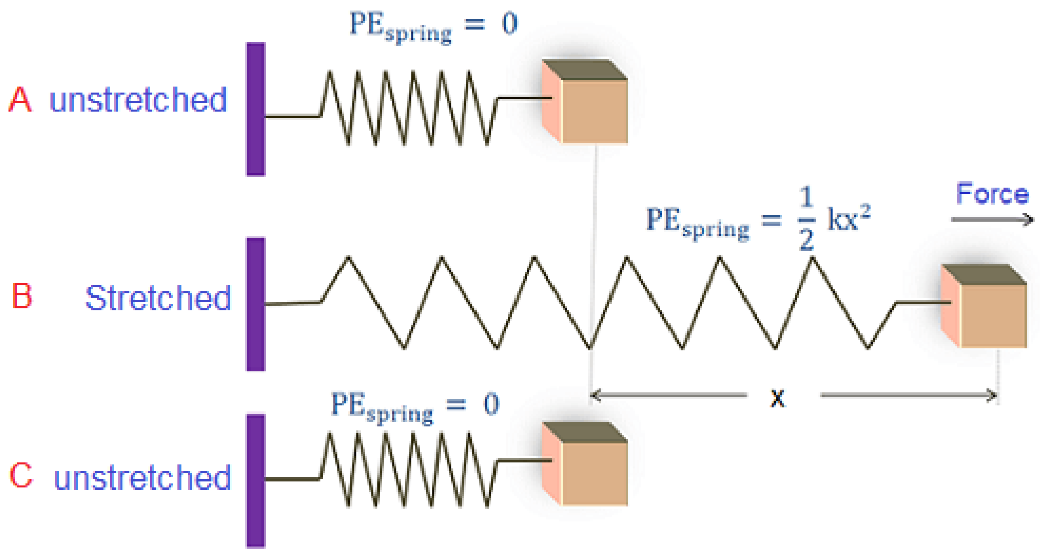
Q: How do you control the robot to reach the goal state while avoiding the obstacle?

Main Motivation

Q: How do you control the robot to reach the goal state while avoiding the obstacle?

Assume omnidirectional robot.

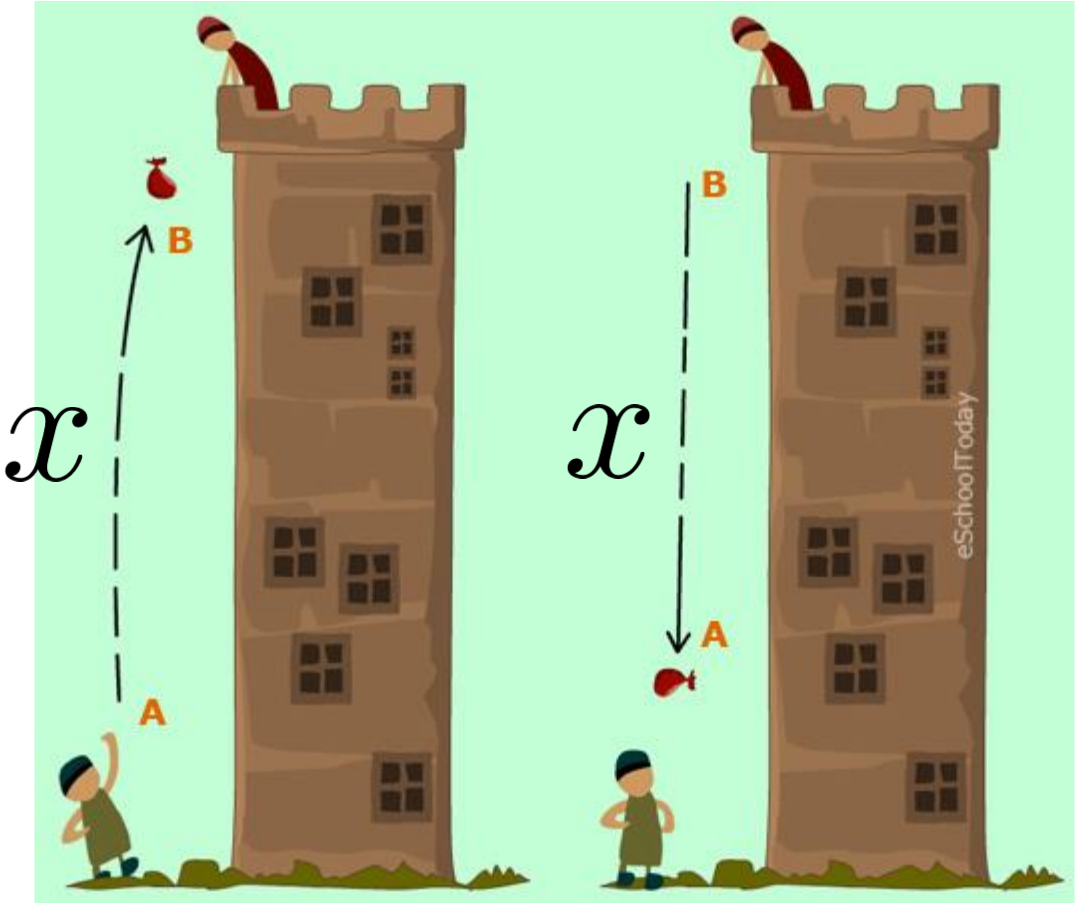
Background: potential energy and forces



$$U(x) = \frac{1}{2}kx^2$$

$$F(x) = -kx$$

Background: potential energy and forces



$$U(x) = mgx$$

$$F(x) = -mg$$

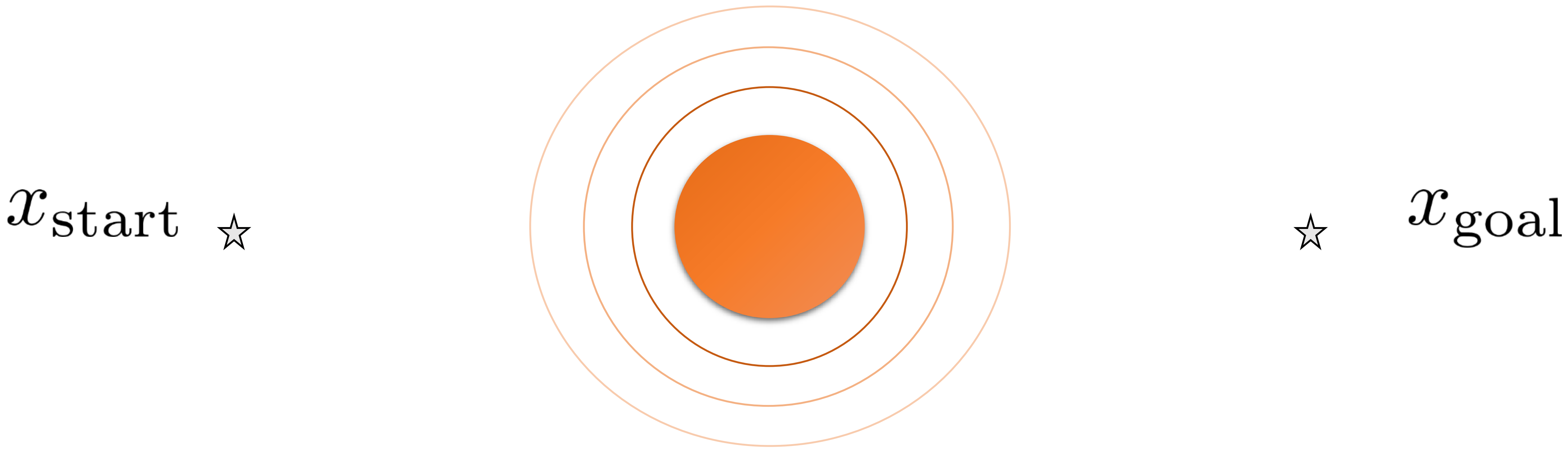
Background: potential energy and forces

In both cases we have conversion from kinetic energy to potential energy $U(x)$.

In both cases there is a force resulting from the potential field, and $F(x)=-dU(x)/dx$.

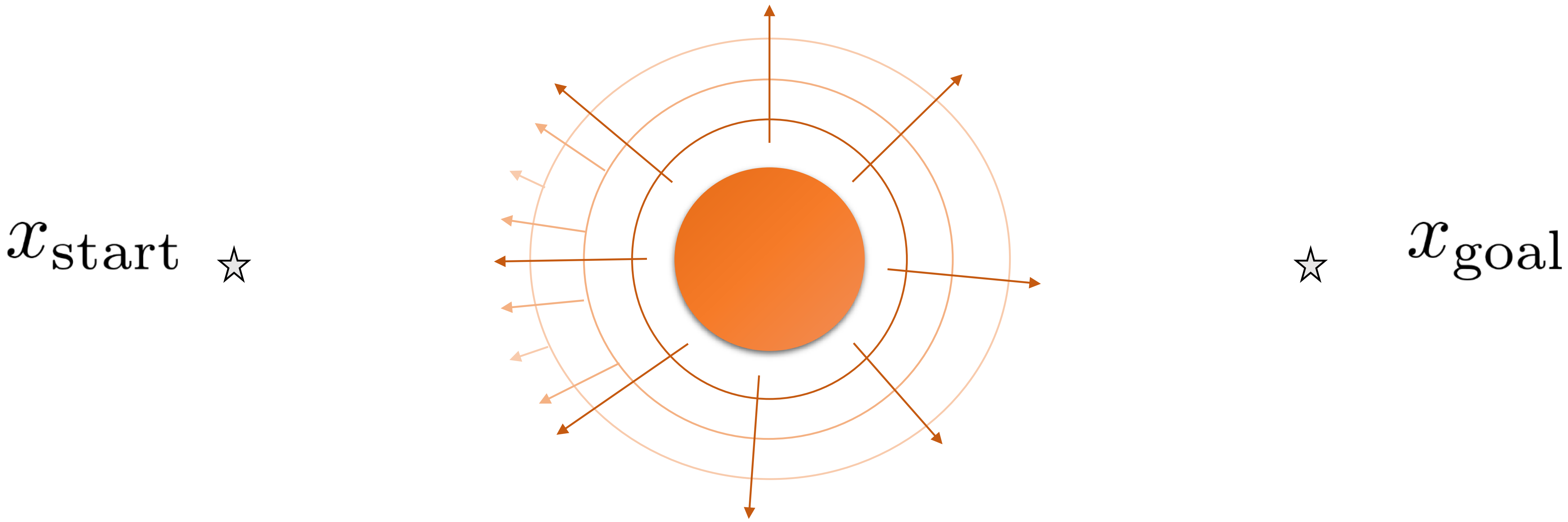
This is a general rule for conservative systems with no external forces.

Main Motivation



Q: How do you control the robot to reach the goal state while avoiding the obstacle?

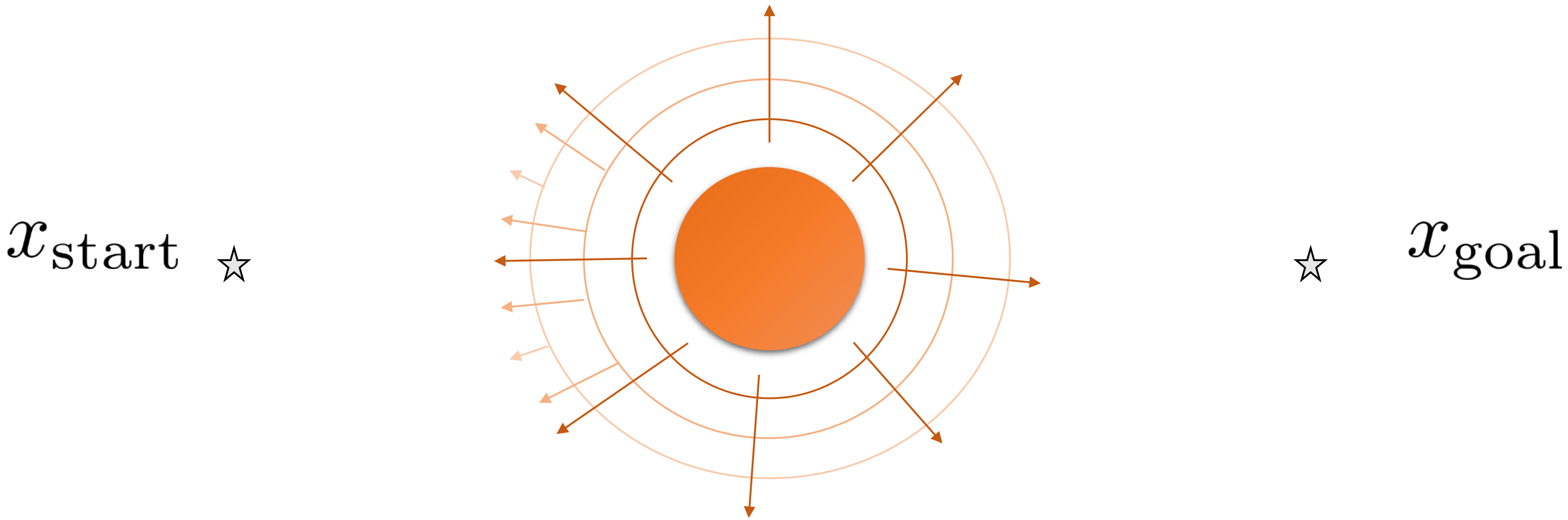
Artificial Potential Fields



Q: How do you control the robot to reach the goal state while avoiding the obstacle?

A: Place a repulsive potential field around obstacles

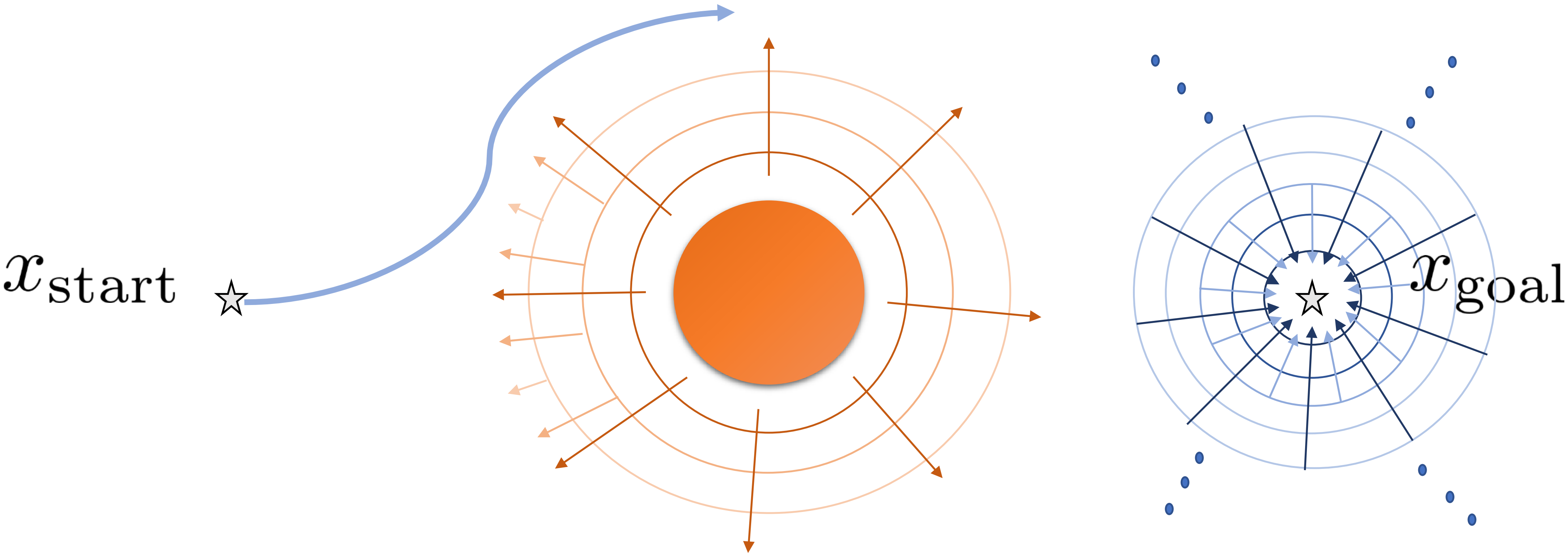
Artificial Potential Fields



Q: How do you control the robot to reach the goal state while avoiding the obstacle?

A: Place a repulsive potential field around obstacles

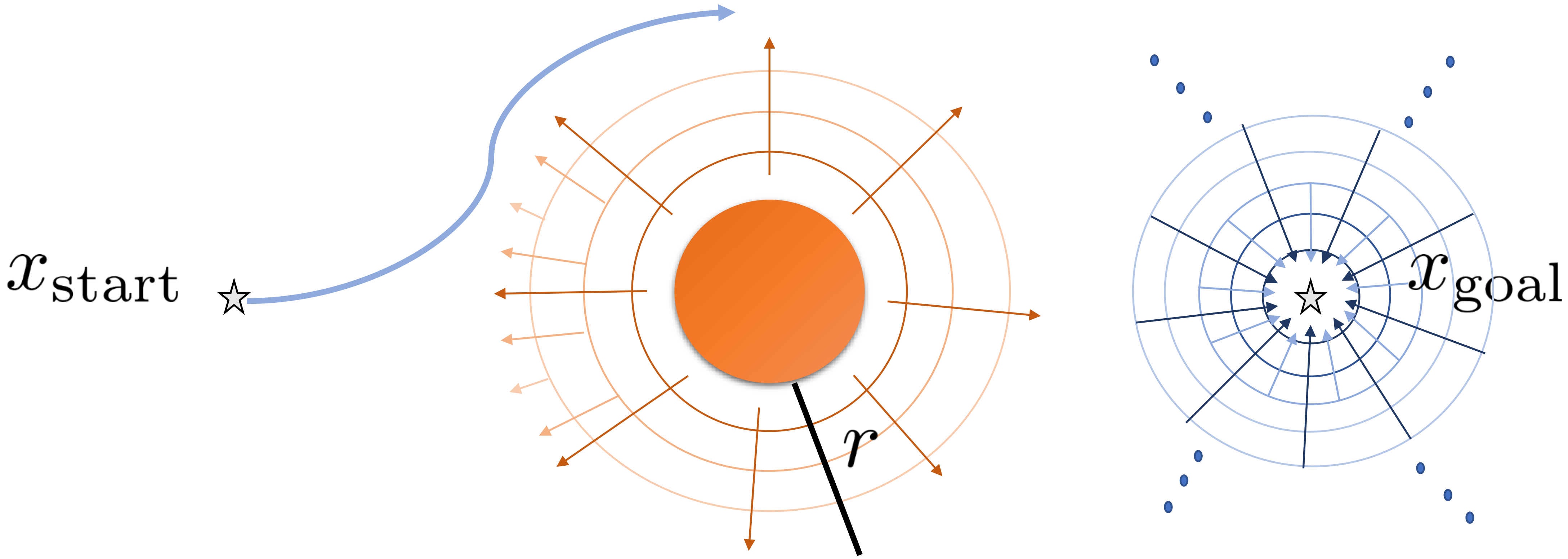
Artificial Potential Fields



Q: How do you control the robot to reach the goal state while avoiding the obstacle?

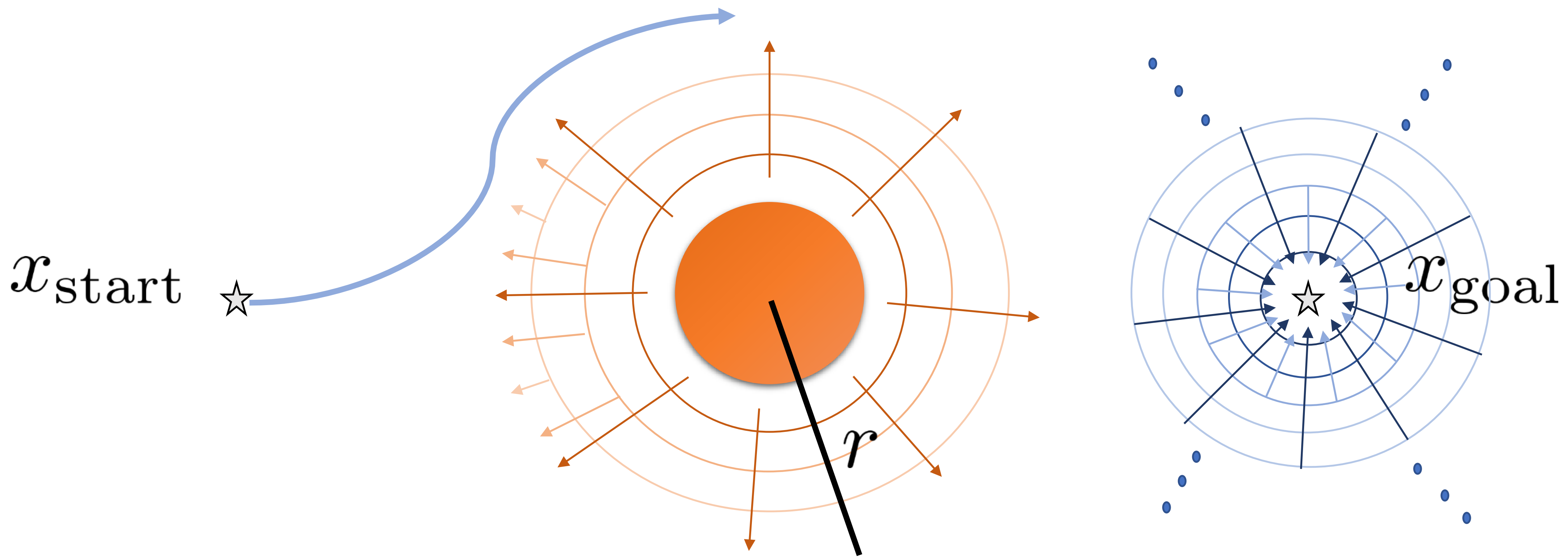
A: Place a repulsive potential field around obstacles
and an attractive potential field around the goal

Artificial Potential Fields



$$U_{\text{repulsive}}(x) = \begin{cases} \left(\frac{1}{d(x,\text{obs})} - \frac{1}{r}\right)^2 & \text{if} \quad d(x,\text{obs}) < r \\ 0 & \text{if} \quad d(x,\text{obs}) \geq r \end{cases}$$

Artificial Potential Fields



$$U_{\text{repulsive}}(x) = \begin{cases} \left(\frac{1}{d(x,\text{obs})} - \frac{1}{r}\right)^2 & \text{if} \quad d(x,\text{obs}) < r \\ 0 & \text{if} \quad d(x,\text{obs}) \geq r \end{cases}$$

$$U_{\text{attractive}}(x) = d(x, x_{\text{goal}})^2$$

How do we compute these distances from obstacles?

7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
3	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

How do we compute these distances from obstacles?

7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
3	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	2
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

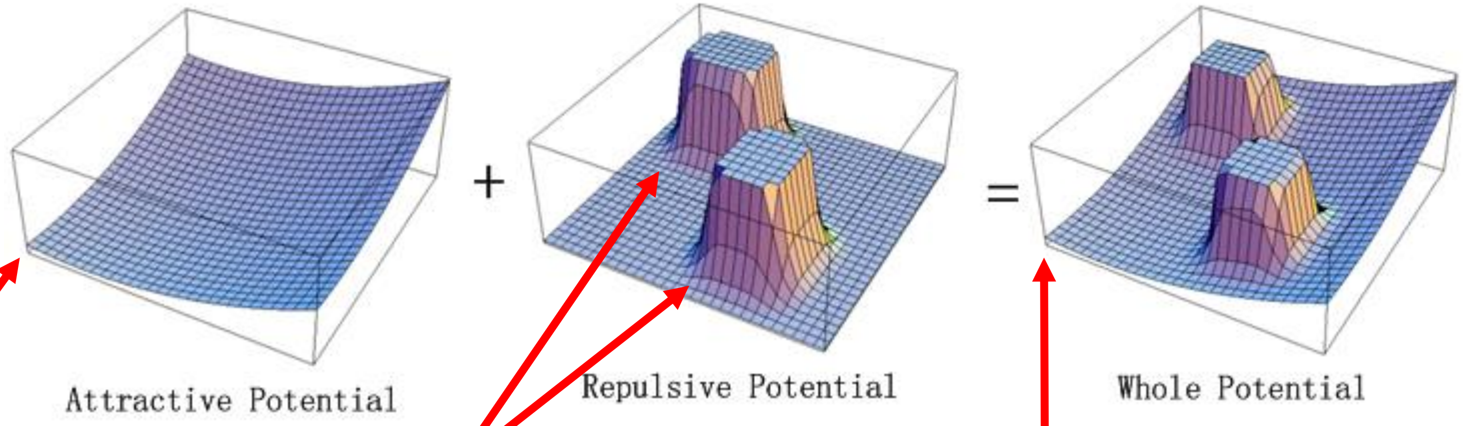
How do we compute these distances from obstacles?

7	**0**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	6	6	6	6
3	0	0	0	0	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	5	5	5	5
2	0	0	0	0	0	0	0	0	0	0	0	6	5	4	4	4
1	0	0	0	0	0	0	0	0	0	0	0	6	5	4	3	3
0	0	0	0	0	0	0	0	0	0	0	0	6	5	4	3	**2**
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

How do we compute these distances from obstacles?

7	**18**	17	16	15	14	13	12	11	10	9	9	9	9	9	9	9
6	17	17	16	15	14	13	12	11	10	9	8	8	8	8	8	8
5	17	16	16	15	14	13	12	11	10	9	8	7	7	7	7	7
4	17	16	15	15	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	6	6	6	6
3	17	16	15	14	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	5	5	5	5
2	17	16	15	14	13	12	11	10	9	8	7	6	5	4	4	4
1	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	3
0	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	**2**
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Attractive and Repulsive Potential Fields



Goal

Obstacles

$$U(x) = \alpha U_{\text{attractive}}(x) + \beta U_{\text{repulsive}}(x)$$

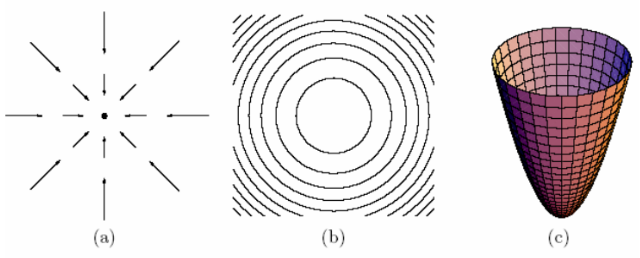
Q1: How do we reach the goal state from an arbitrary state?

Q2: In this example there is an unambiguous way to reach the goal from any state. Is this true in general?

From Potential Fields to Forces

Make the robot move by applying forces resulting from potential fields

$$U_{\text{attractive}}(x) = d(x, x_{\text{goal}})^2$$



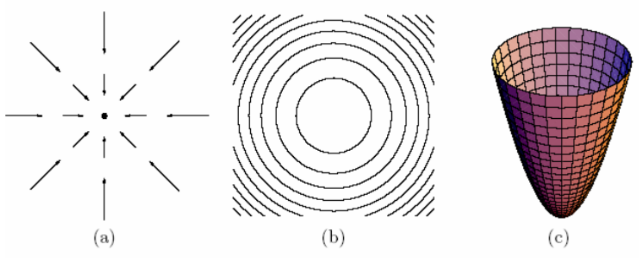
$$F_{\text{attractive}}(x) = -\nabla_x U_{\text{attractive}}(x) = -2(x - x_{\text{goal}})$$

Attractive force makes state x go to the bottom
of the potential energy bowl. Bottom=Goal = low-energy state.

From Potential Fields to Forces

Make the robot move by applying forces resulting from potential fields

$$U_{\text{attractive}}(x) = d(x, x_{\text{goal}})^2$$



$$F_{\text{attractive}}(x) = -\nabla_x U_{\text{attractive}}(x) = -2(x - x_{\text{goal}})$$

Attractive force makes state x go to the bottom
of the potential energy bowl. Bottom=Goal = low-energy state.

Move the robot using F=ma, for m=1:

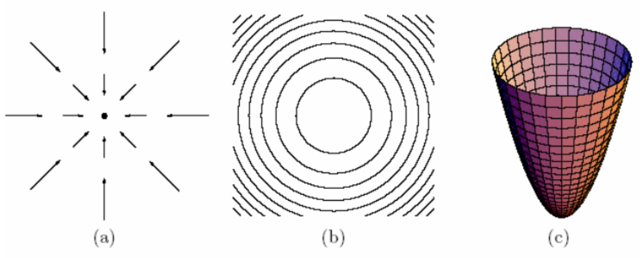
$$\dot{x}_{t+1} = \dot{x}_t + \delta t F(x_t)$$

Gradient descent down the potential bowl

From Potential Fields to Forces

Make the robot move by applying forces resulting from potential fields

$$U_{\text{attractive}}(x) = d(x, x_{\text{goal}})^2 \quad \Longrightarrow \quad F_{\text{attractive}}(x) = -\nabla_x U_{\text{attractive}}(x) = -2(x - x_{\text{goal}})$$



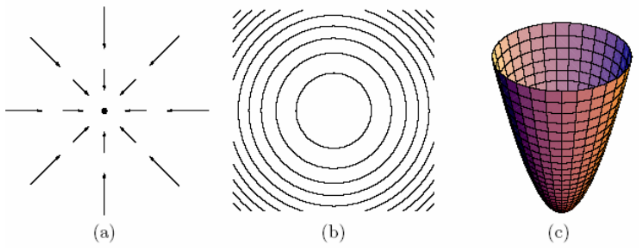
Attractive force makes state x go to the bottom
of the potential energy bowl. Bottom=Goal = low-energy state.

Q: Do you see any problems with this potential energy and force if x is far away from goal?

From Potential Fields to Forces

Make the robot move by applying forces resulting from potential fields

$$U_{\text{attractive}}(x) = d(x, x_{\text{goal}})^2 \Longrightarrow F_{\text{attractive}}(x) = -\nabla_x U_{\text{attractive}}(x) = -2(x - x_{\text{goal}})$$



Attractive force makes state x go to the bottom of the potential energy bowl. Bottom=Goal = low-energy state.

Q: Do you see any potential problems with this if x is far away from goal?

A: The farther the robot is the stronger the force. May need to normalize the force vector. Alternatively:

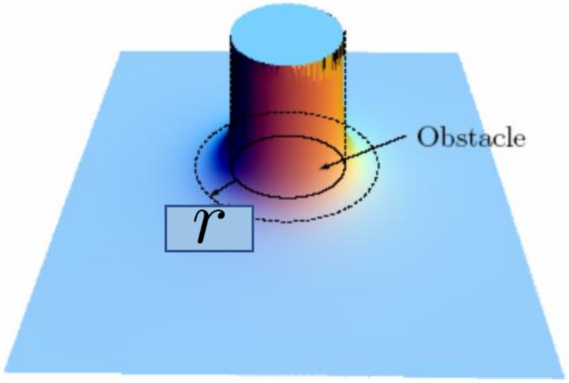
$$U_{\text{attractive}}(x) = d(x, x_{\text{goal}}) \Longrightarrow F_{\text{attractive}}(x) = -\frac{(x - x_{\text{goal}})}{d(x, x_{\text{goal}})}$$

From Potential Fields to Forces

Make the robot move by applying forces resulting from potential fields

$$U_{\text{repulsive}}(x) = \begin{cases} (\frac{1}{d(x,\text{obs})} - \frac{1}{r})^2 & \text{if} \quad d(x, \text{obs}) < r \\ 0 & \text{if} \quad d(x, \text{obs}) \geq r \end{cases}$$

$$F_{\text{repulsive}}(x) = \begin{cases} 2(\frac{1}{d(x,\text{obs})} - \frac{1}{r})\frac{\nabla_x d(x,\text{obs})}{d(x,\text{obs})^2} & \text{if} \quad d(x, \text{obs}) < r \\ 0 & \text{otherwise} \end{cases}$$



Repulsive force makes state x go away from the obstacle
to lower potential energy states. Free space = {low-energy states}

Move the robot using F=ma, for m=1:

$$\dot{x}_{t+1} = \dot{x}_t + \delta t F(x_t)$$

Gradient descent until obstacle is cleared

Combining Attractive and Repulsive Forces

Potential energy

$$U_{\text{total}}(x) = \alpha U_{\text{attractive}}(x) + \beta U_{\text{repulsive}}(x)$$

$$\Downarrow$$

results in forces

$$F_{\text{total}}(x) = \alpha F_{\text{attractive}}(x) + \beta F_{\text{repulsive}}(x)$$

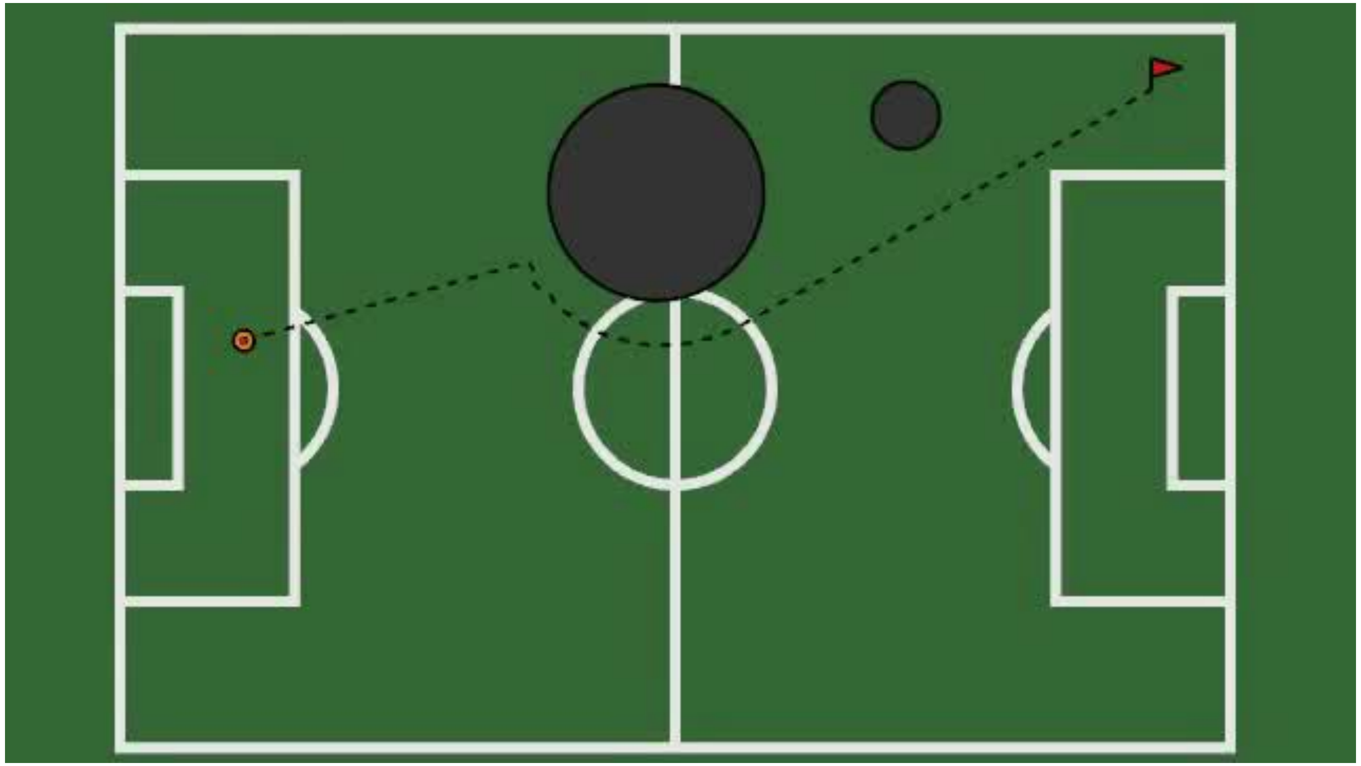
$$\Downarrow$$

makes robot accelerate

$$\dot{x}_{t+1} = \dot{x}_t + \delta t F(x_t)$$

Artificial Potential Fields: Example

Advantages of potential fields:
- Can handle moving obstacles
- Fast and easy to compute
- Fairly reactive



Combining Attractive and Repulsive Forces

Potential energy

$$U_{\text{total}}(x) = \alpha U_{\text{attractive}}(x) + \beta U_{\text{repulsive}}(x)$$

results in forces

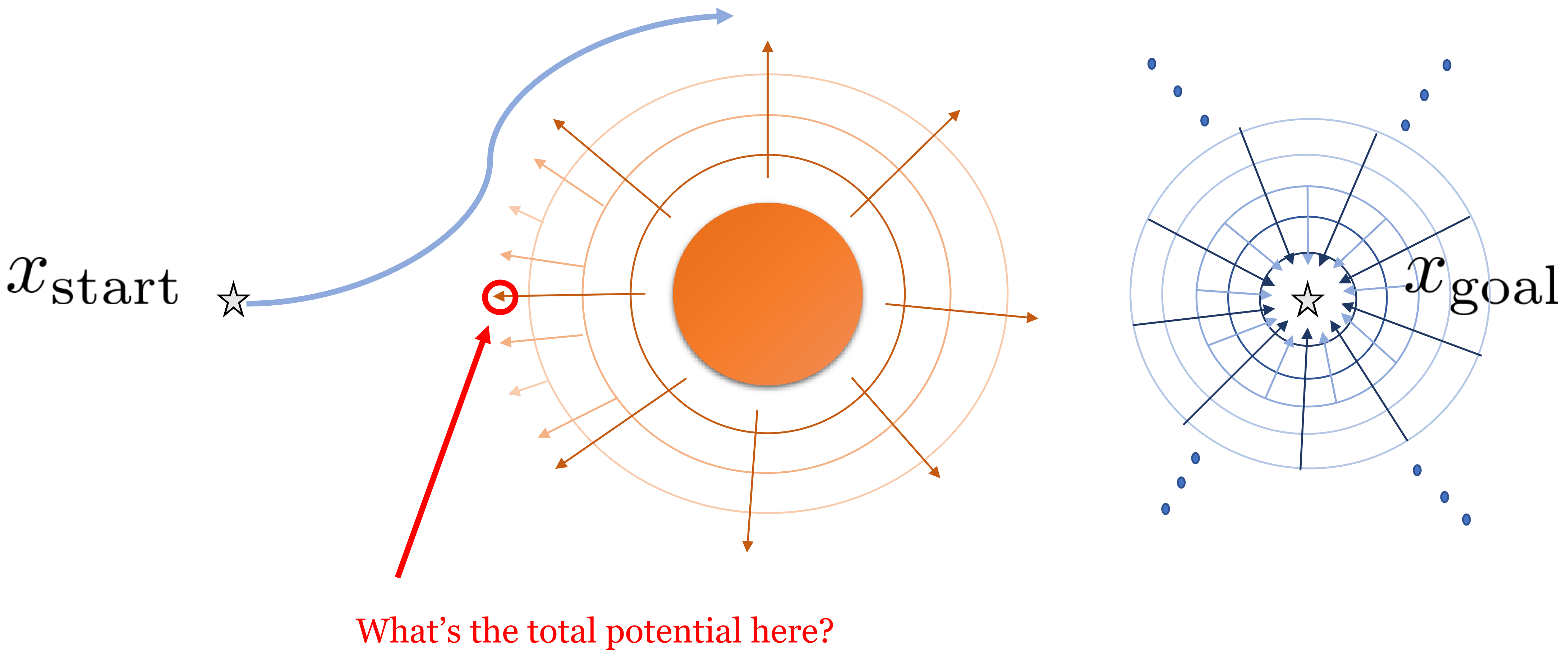
$$F_{\text{total}}(x) = \alpha F_{\text{attractive}}(x) + \beta F_{\text{repulsive}}(x)$$

Q: What's a possible problem with addition of forces?

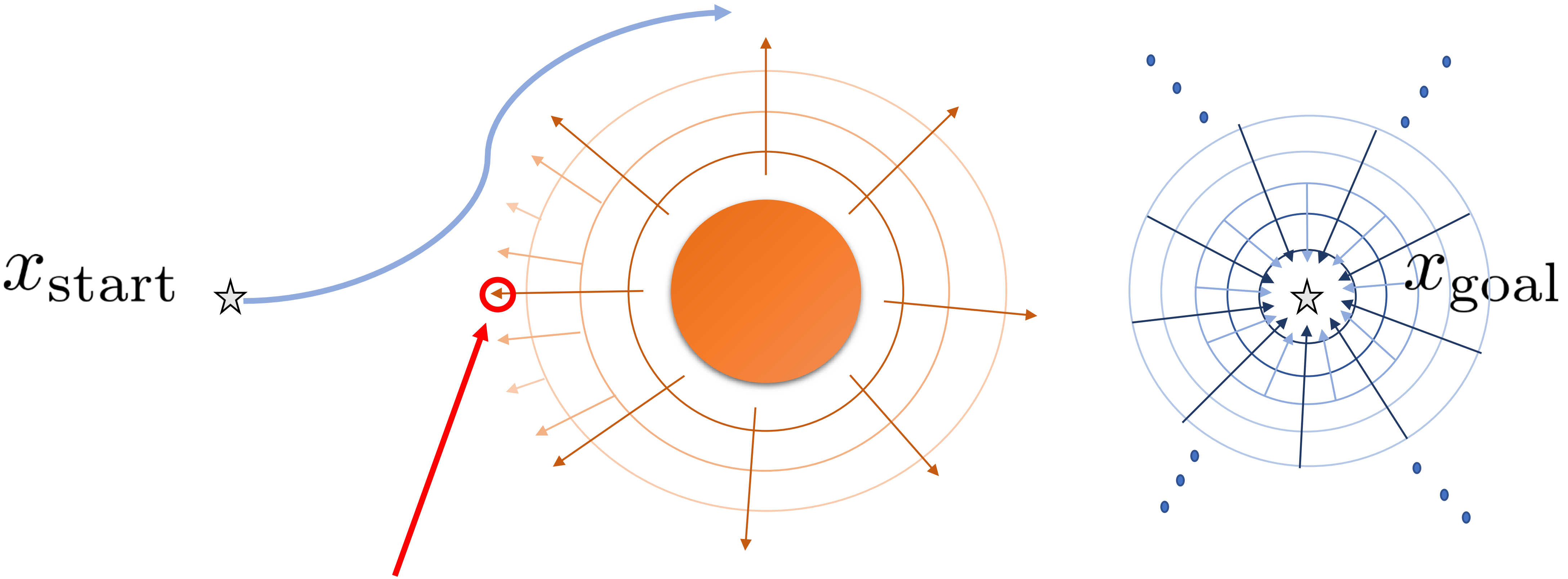
makes robot accelerate

$$\dot{x}_{t+1} = \dot{x}_t + \delta t F(x_t)$$

Artificial Potential Fields



Artificial Potential Fields



What's the total potential here?

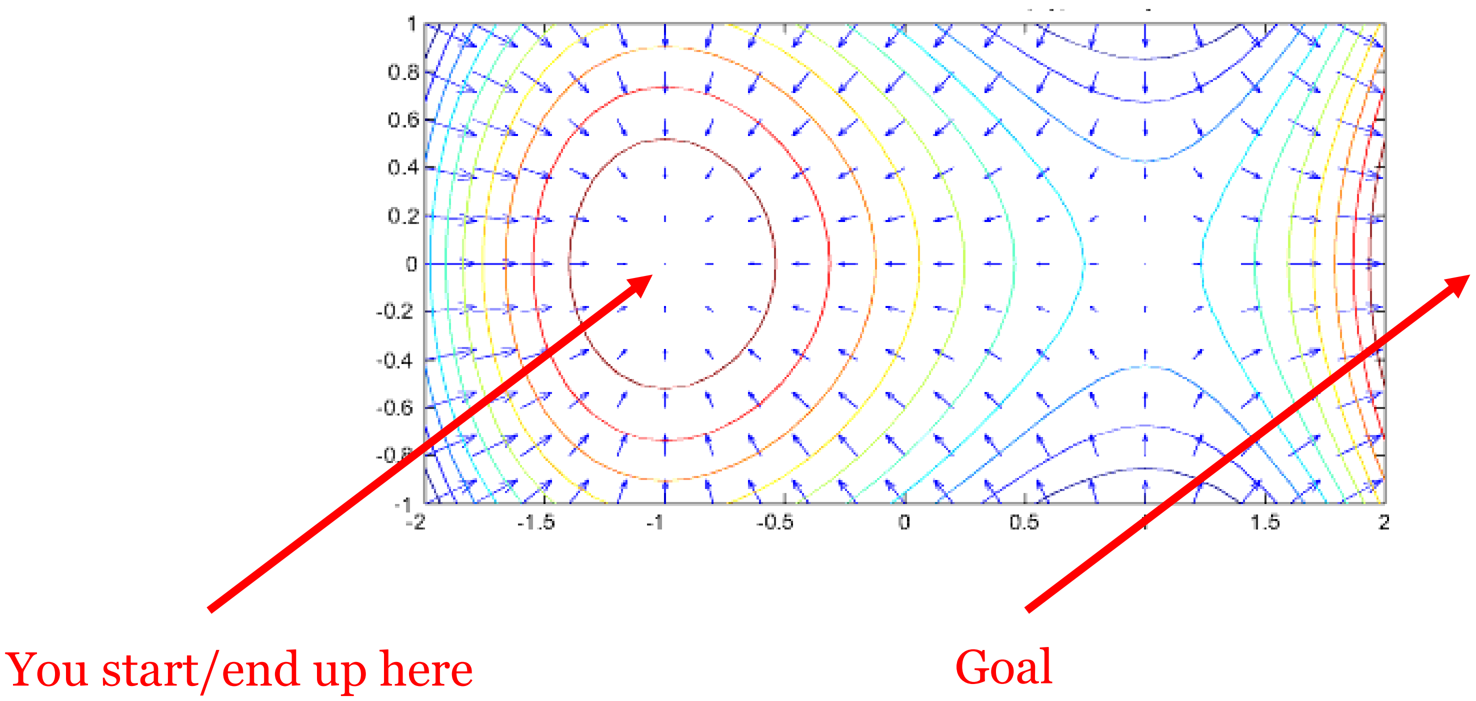
It's zero. The repulsive force is exactly the opposite of the attractive force (assuming alpha = beta)

$$F_{\text{total}}(x) = \alpha F_{\text{attractive}}(x) + \beta F_{\text{repulsive}}(x) = 0$$

Problem: gradient descent gets stuck

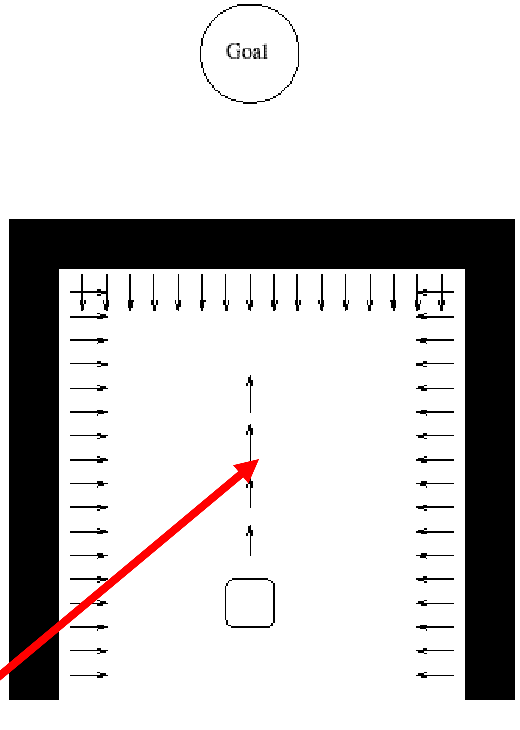
Local Minima on the Potential Field: Getting Stuck

States of zero total force correspond to local minima in the potential function:



Local Minima on the Potential Field: Getting Stuck

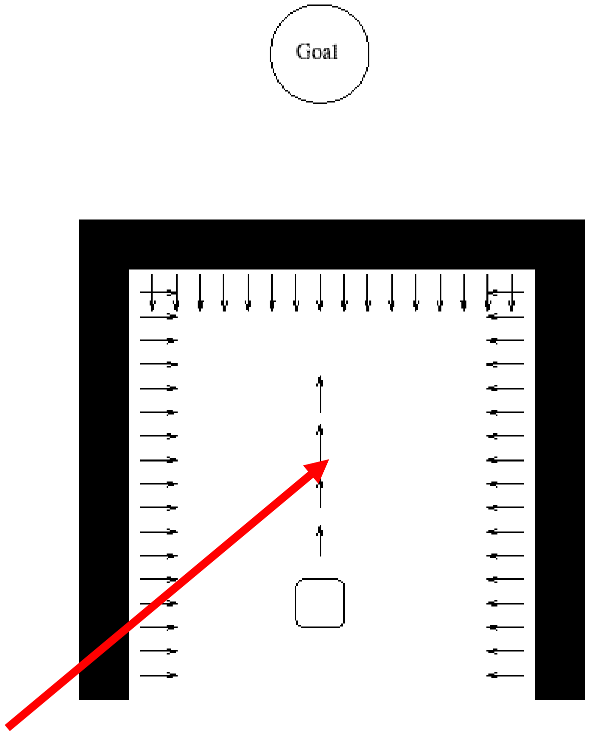
States of zero total force correspond to local minima in the potential function:



If you end up here gradient descent can't help you. All local moves seem identical in terms of value → local min

Local Minima on the Potential Field: Getting Unstuck Randomly

States of zero total force correspond to local minima in the potential function:

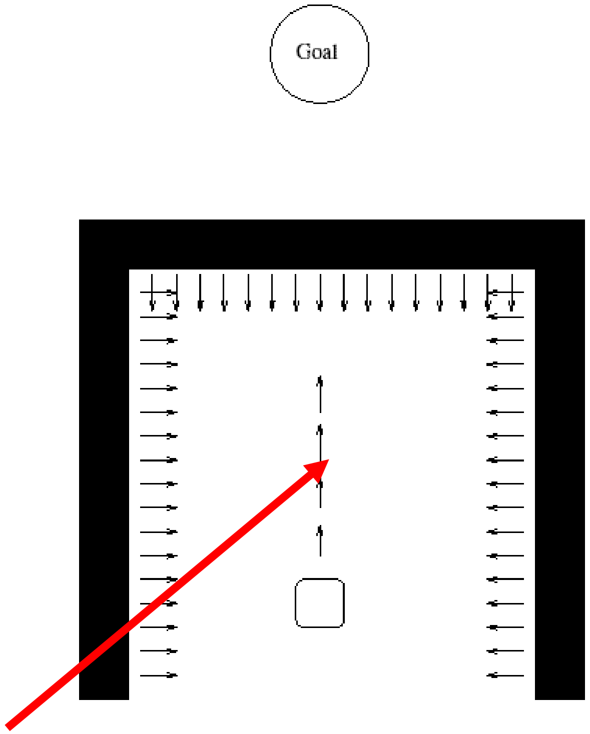


Solution: Do random move in case it helps you get unstuck.

Problem: If you end up here gradient descent can’t help you. All local moves seem identical in terms of value → local min

Local Minima on the Potential Field: Getting Unstuck By Backing Up

States of zero total force correspond to local minima in the potential function:

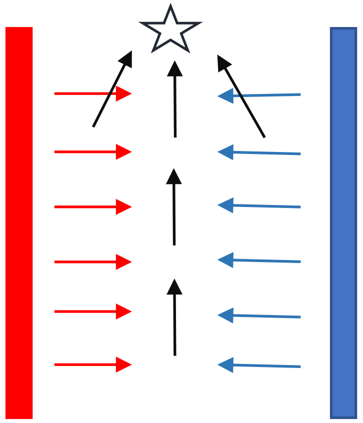


Solution: back up and get out from the dead end, just like you entered it.

Problem: If you end up here gradient descent can't help you. All local moves seem identical in terms of value → local min

Drawbacks of potential fields

- *Local minima*
  - Attractive and repulsive forces can balance, so robot makes no progress.
  - Closely spaced obstacles, or dead end.
- *Unstable oscillation*
  - The dynamics of the robot/environment system can become unstable.
  - High speeds, narrow corridors, sudden changes

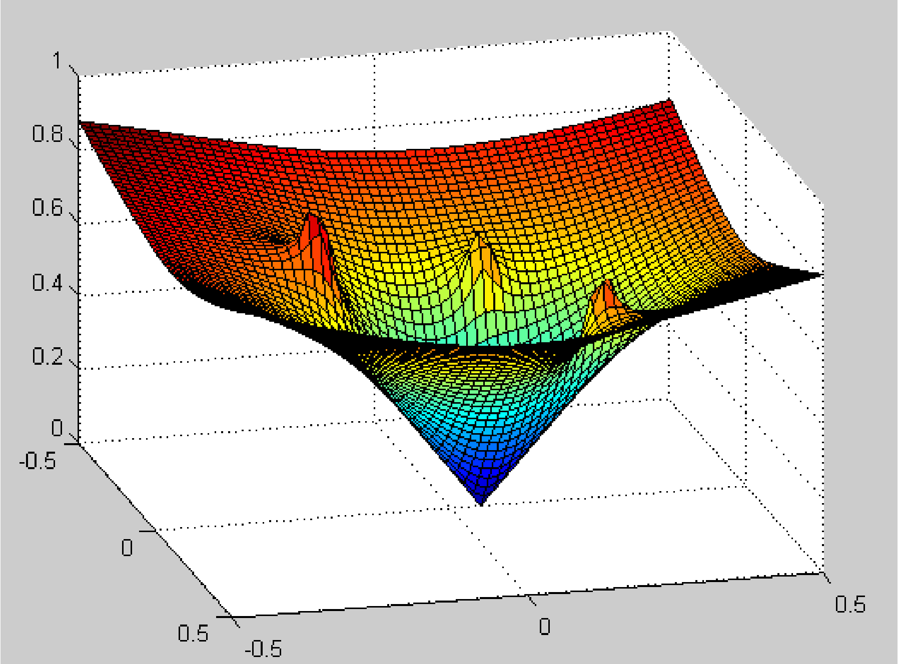


Avoiding Local Minima on the Potential Field: Navigation Functions

Potential energy function $\phi(x)$ with a **single global minimum** at the goal, and **no local minima.**

For any state x there exists a neighboring state x' such that $\phi(x') < \phi(x)$.

So far not used in practice very much because they are usually as hard to compute as a planned path from the current state to the goal.



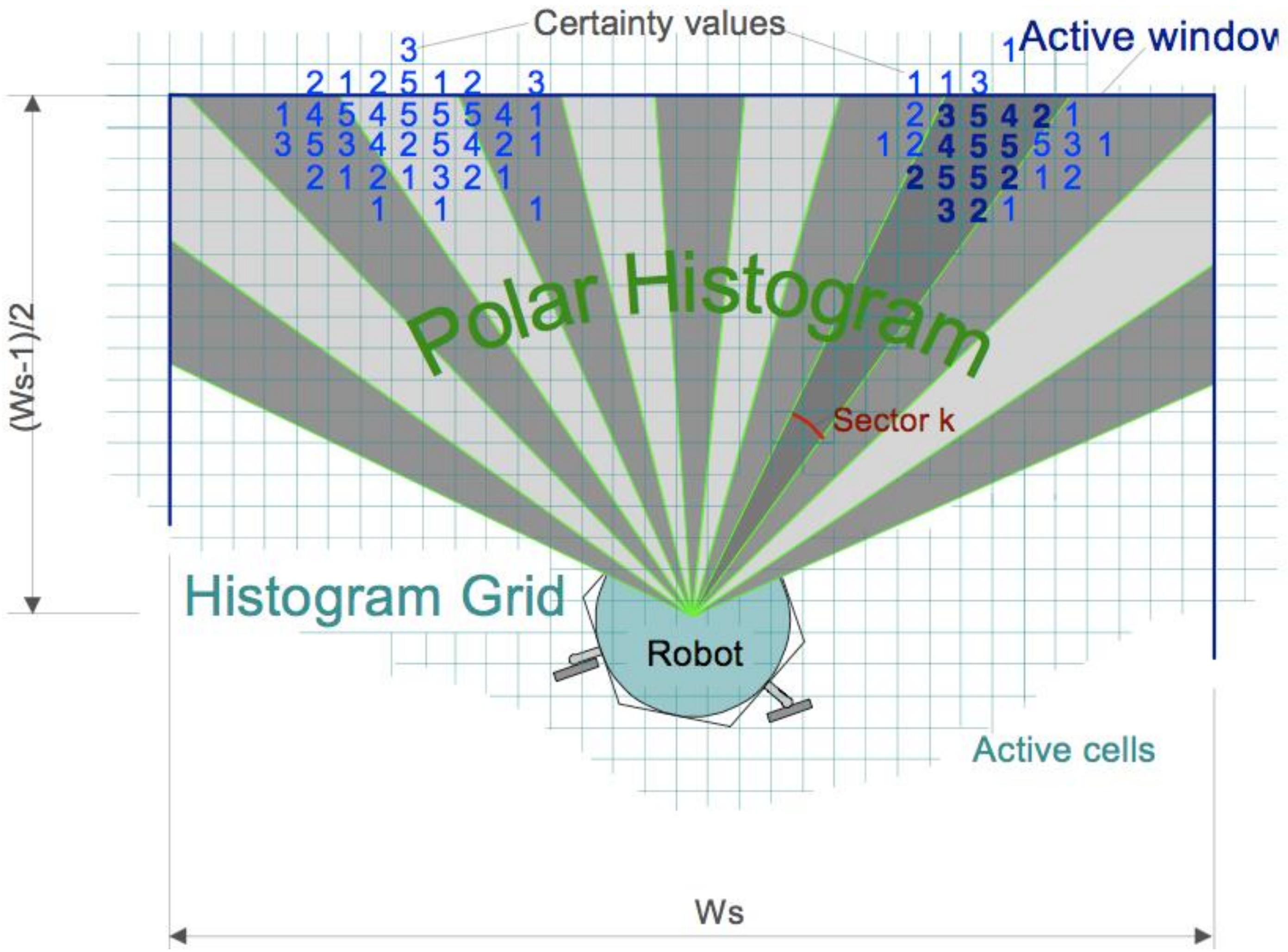
Addressing the Drawbacks of Potential Fields

- Vector Field Histogram (VFH)

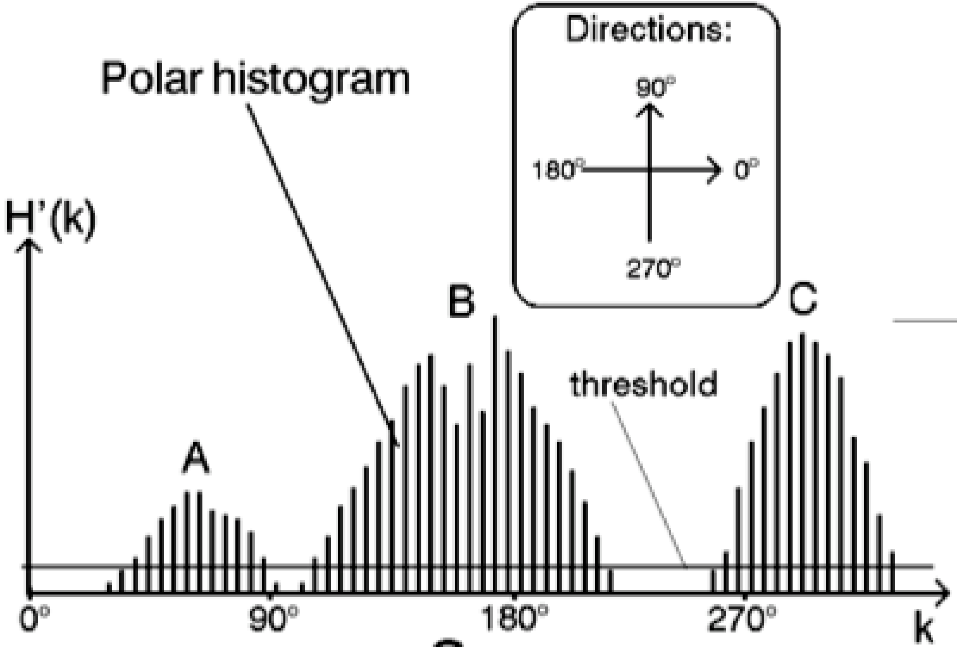
- Dynamic Window Approach

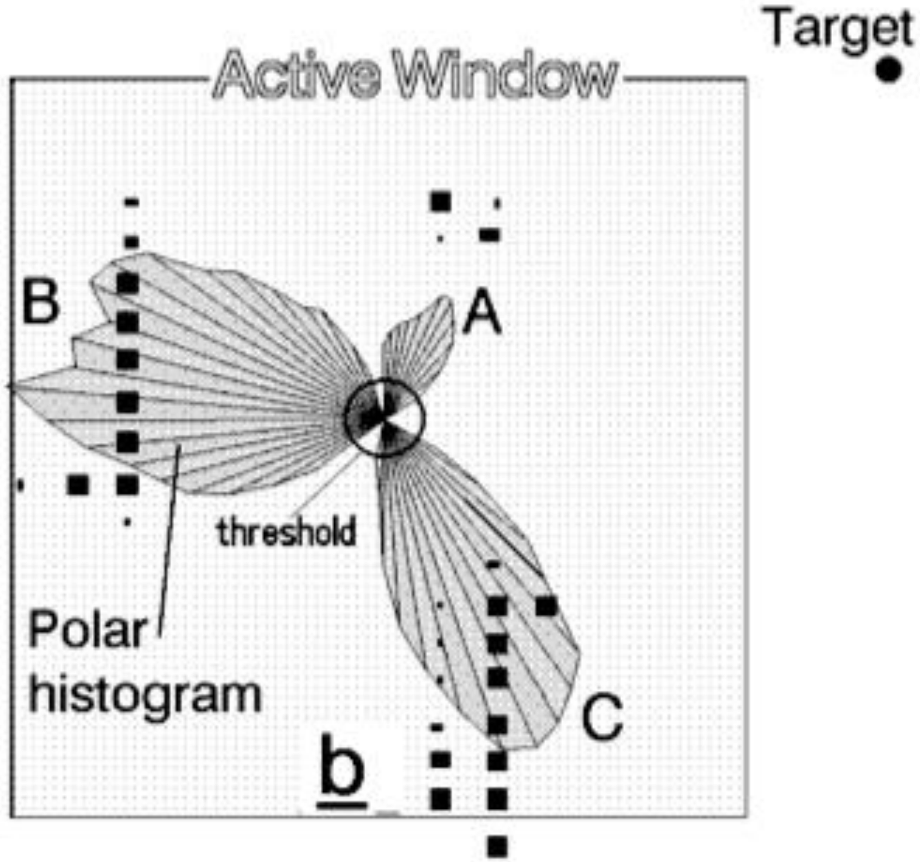
Both methods for local obstacle avoidance

VFH (Vector Field Histogram)



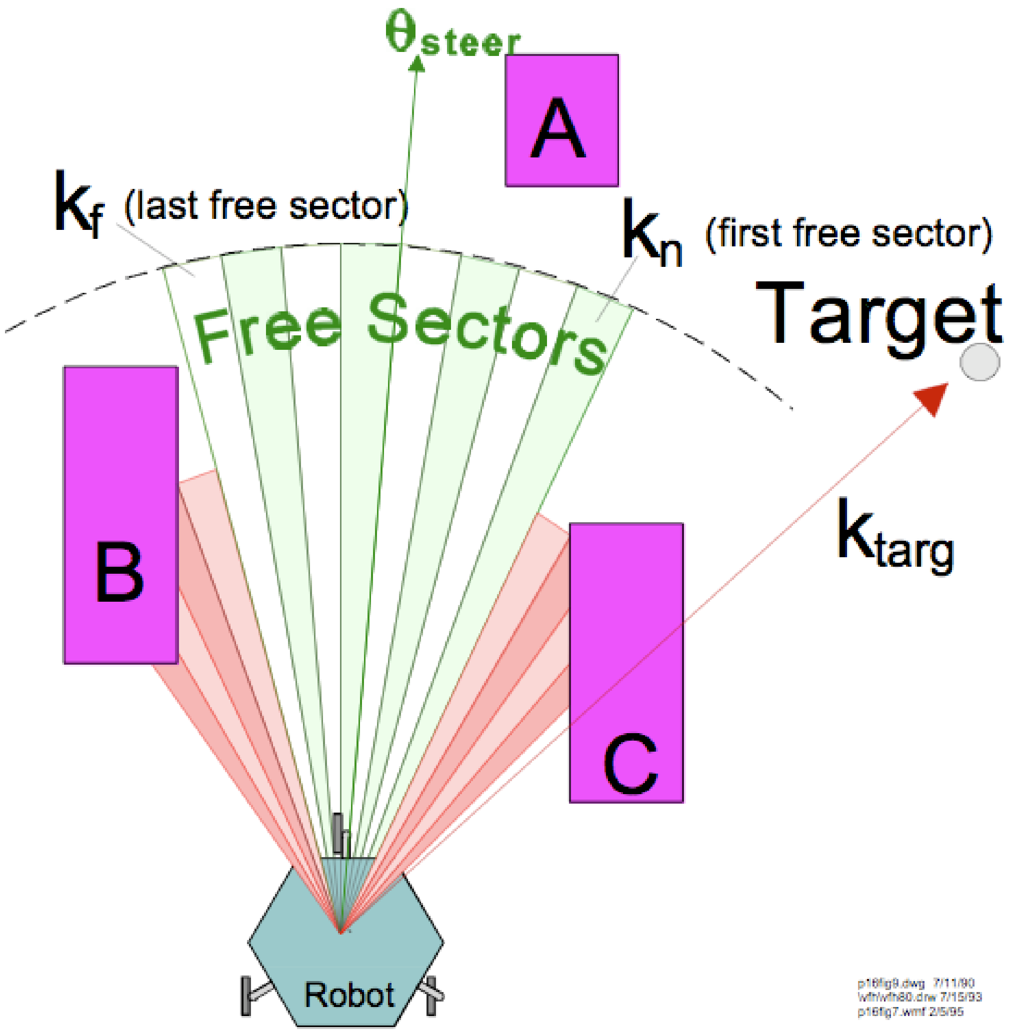
VFH (Vector Field Histogram)





High risk for cells with high probability of occupation.
Risk inversely proportional to distance.

VFH (Vector Field Histogram)



VFH (Vector Field Histogram)

VFH (Vector Field Histogram)

DWA (Dynamic Window Approach)

Local, reactive controller

1. Sample a set of controls for x,y,theta
2. Simulate where each control is going to take the robot
3. Eliminate those that lead to collisions.
4. Reward those that agree with a navigation plan.
5. Reward high-speeds
6. Reward proximity to goal.
7. Pick control with highest score that doesn't lead to collision.