AI Robotics

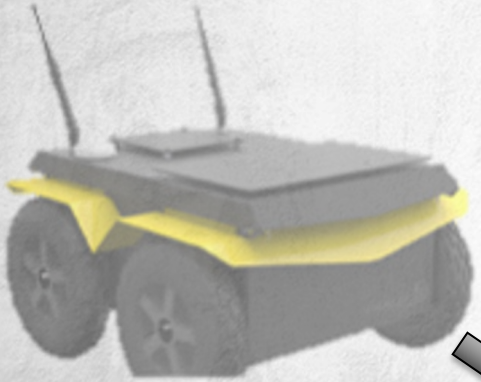
Artificial Potential Fields

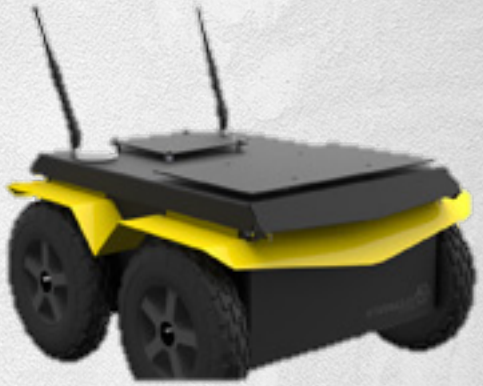
Overview

Control

Robot State (or pose):
$X_{0:t} = \{X_0, \ldots, X_t\}$

Control: An input action that causes the robot state to change

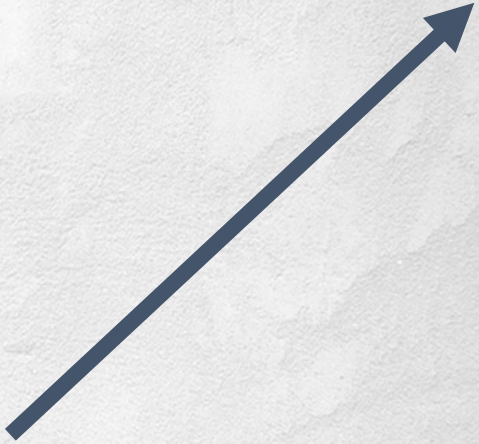




act

Robot Controls:
$u_{0:t} = \{u_0, u_1, \ldots, u_t\}$

Idea

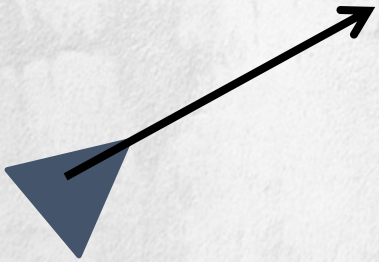


- Represent all actions as vectors
- A Vector is composed of:
  - Direction and magnitude
  - Or equivalently
  - X-component and Y-component
- What happens when multiple behaviors are active at the same time?
  - Simply add the vectors together
  - Emerging action is the sum of all the active behaviors

Potential Fields

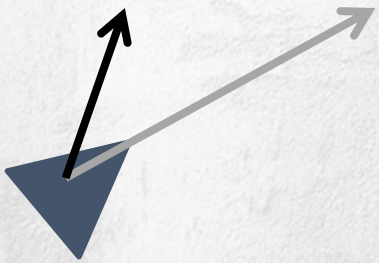
Example:

Example: towards goal



Example: avoid obstacle

Example: add them together



Example: final force

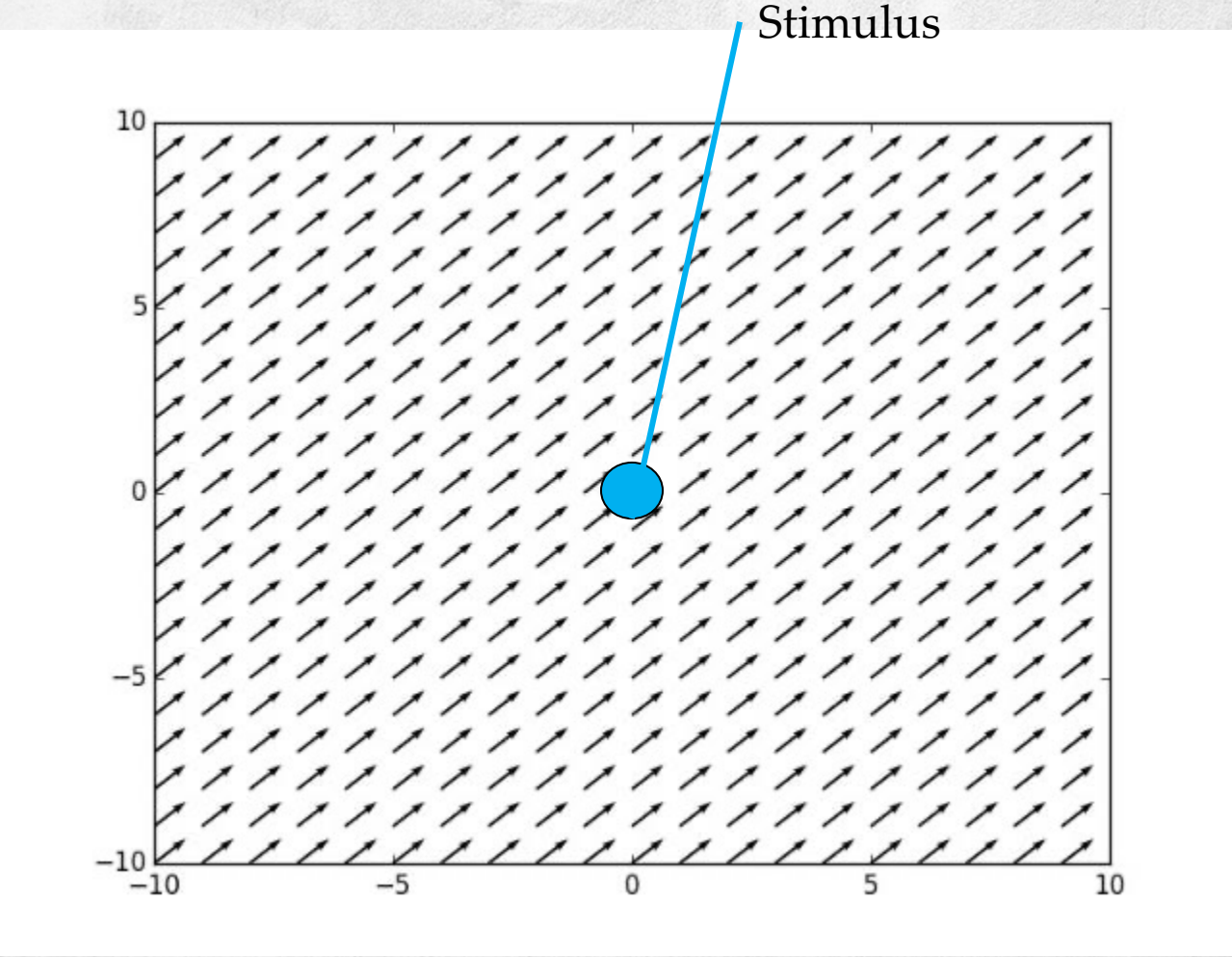
Potential Field Math

- (2D) Vector representations:
  - [x, y]
  - Direction (d) , Magnitude (m)
- Conversion (Magnitude/Direction -> [x , y])
  - x = m * cos(d) [python: `math.cos(a)` accepts radians]
  - y = m * sin(d) [python: `math.sin(a)` accepts radians]
- Conversion ([x,y] -> Magnitude/Direction)
  - Direction = atan2(y , x) [python: `math.atan2(y,x)` returns radians]
  - Magnitude = $\sqrt{x^2 + y^2}$ [python: `math.hypot(x,y)`]
- Summing/Adding Vectors:
  - Easiest using [x, y] representation
  - $[x_1, y_1] + [x_2, y_2] = [x_1 + x_2, y_1 + y_2]$

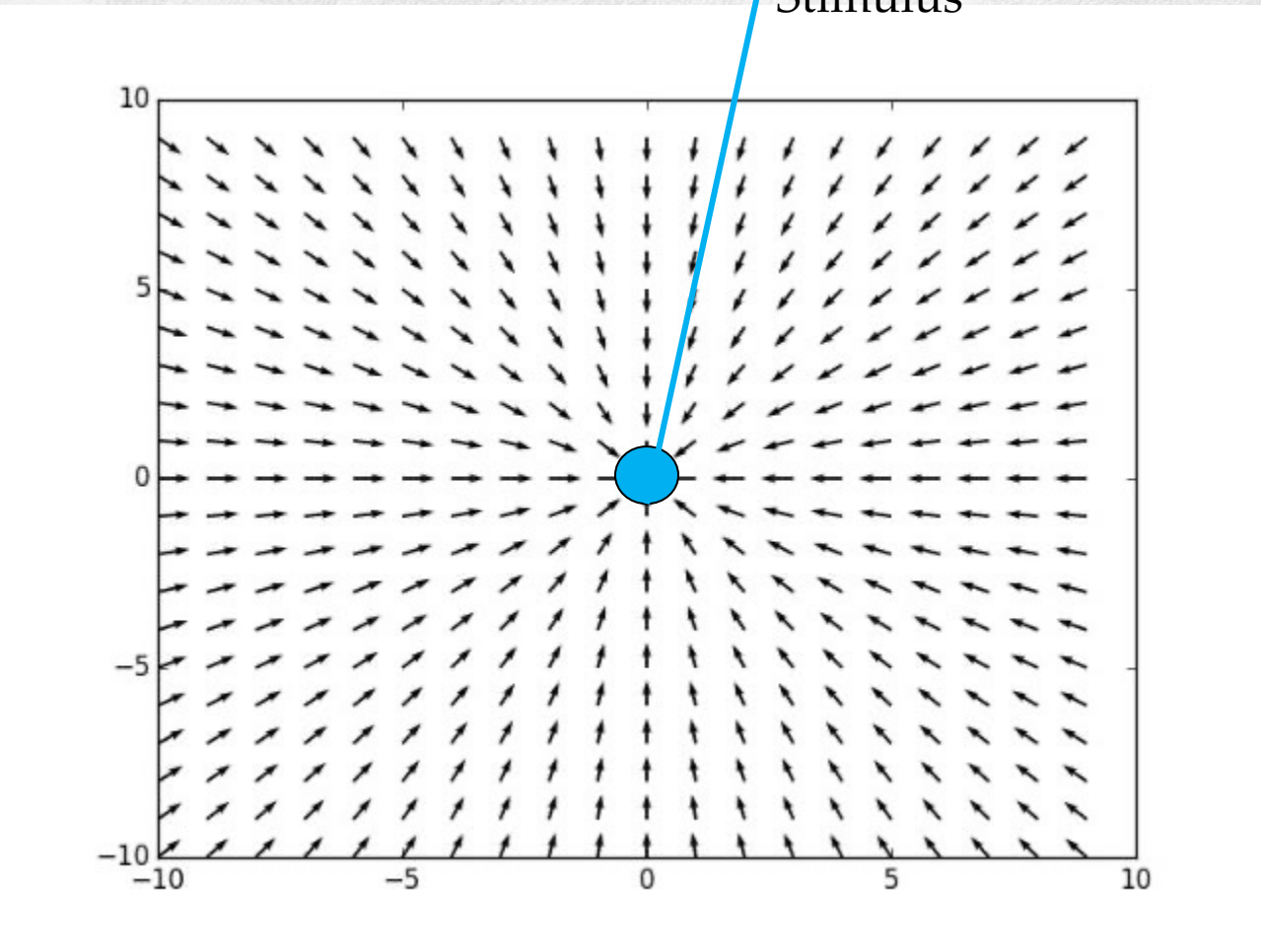
What type of behaviors?

- Basic types of motions in response to a stimulus -
  -- Uniform
    - Vector is the same, regardless of stimulus
  - Attractive
    - Vector points towards the stimulus
  - Repulsive
    - Vector points away from stimulus
  - Tangential
    - Vector points perpendicular to radial lines extending outward from the stimulus
  - Perpendicular
    - Vector points perpendicular to stimulus (used for linear stimuli)

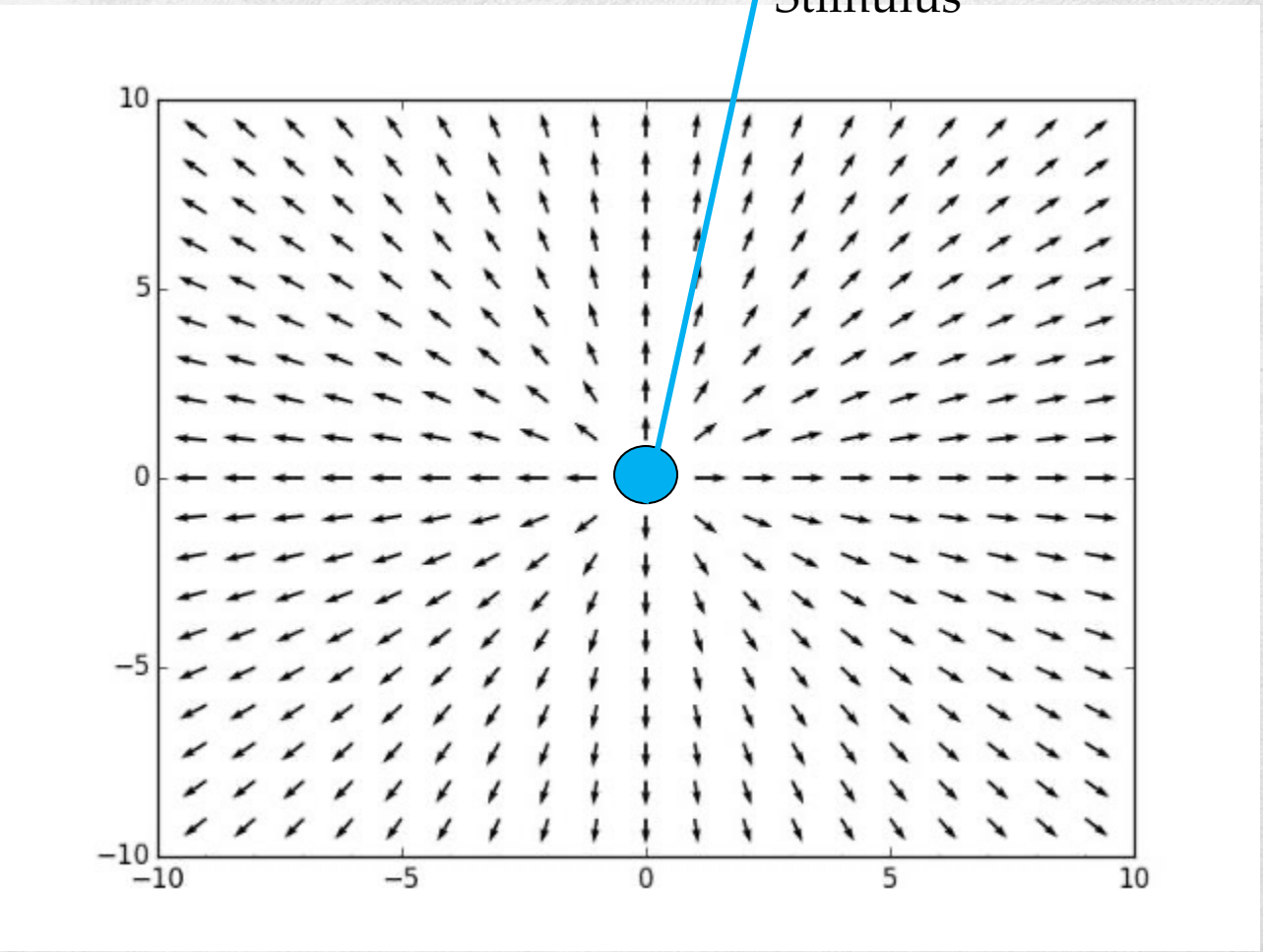
Uniform



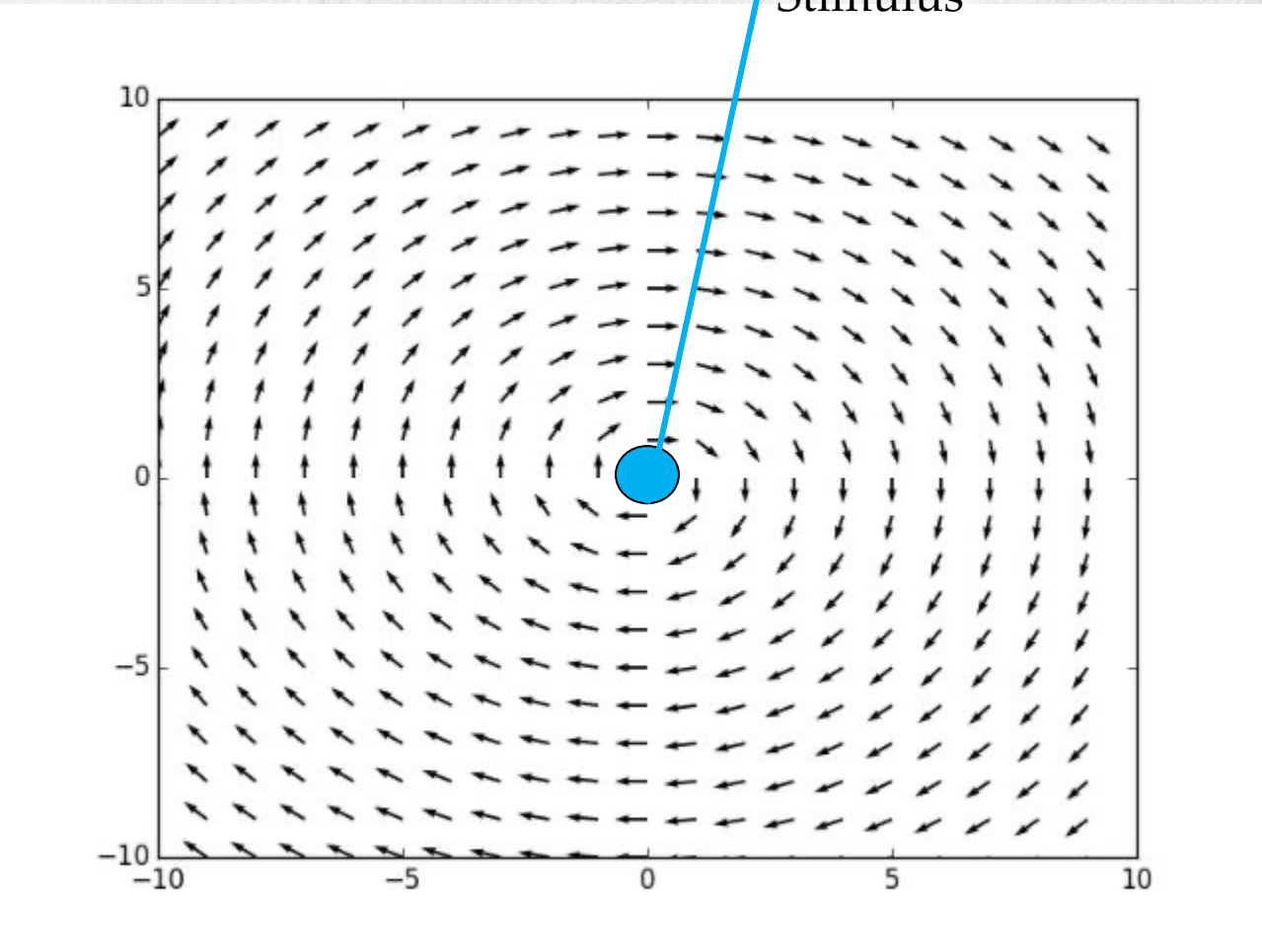
Attractive



Repulsive

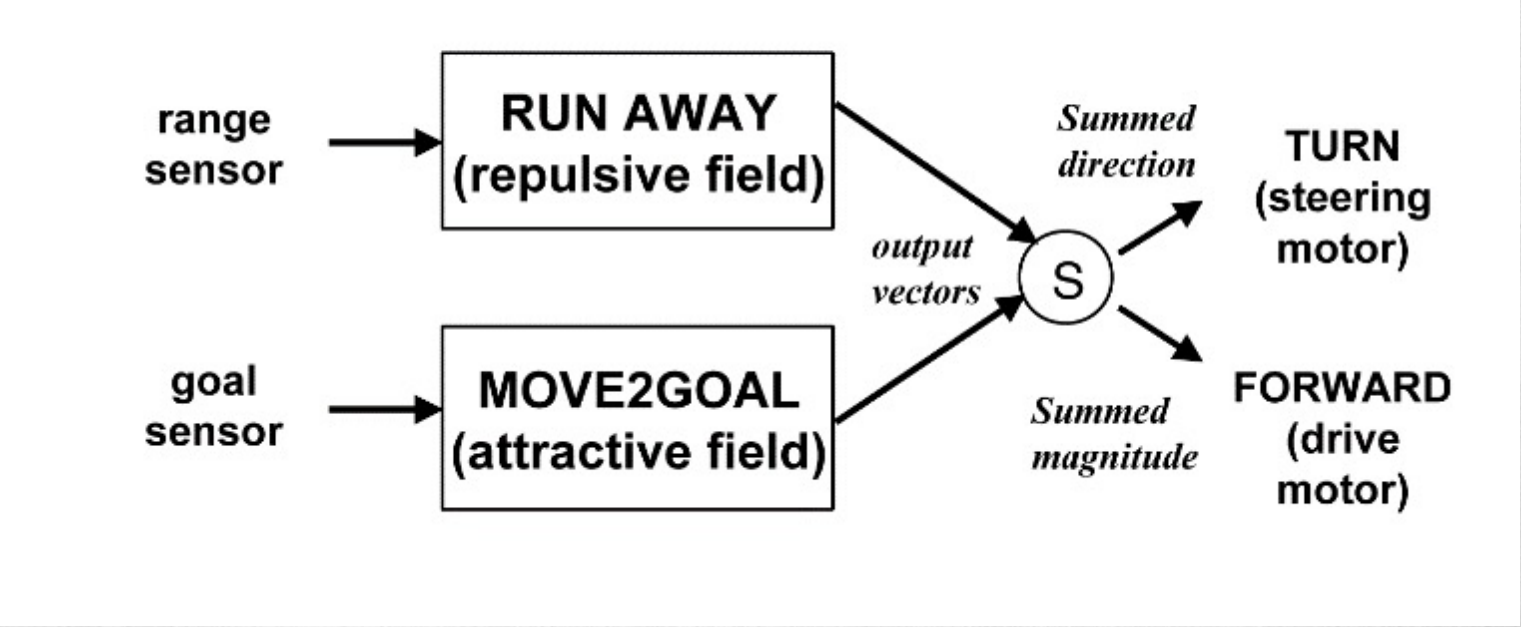


Tangential

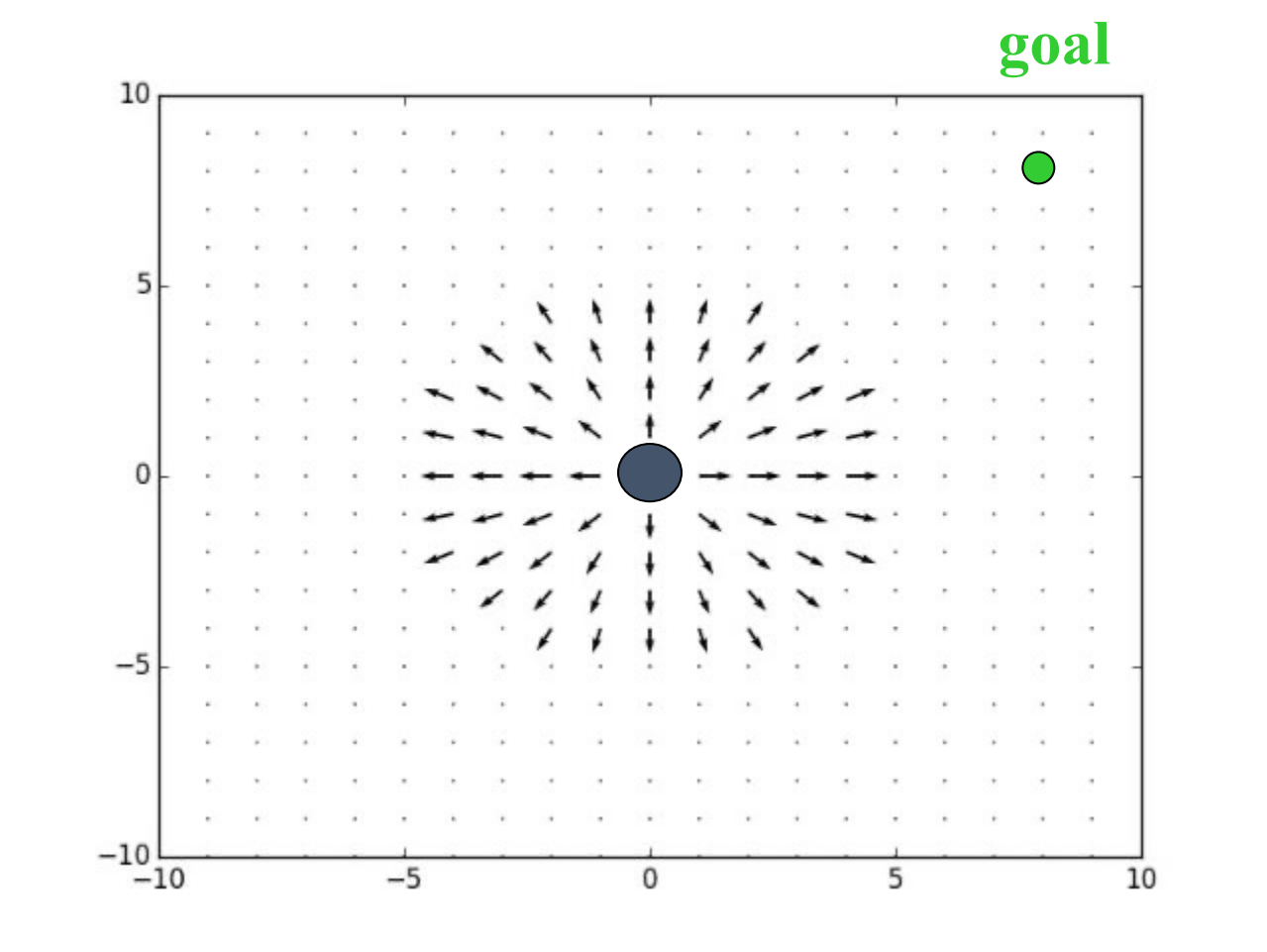


Combining Behaviors

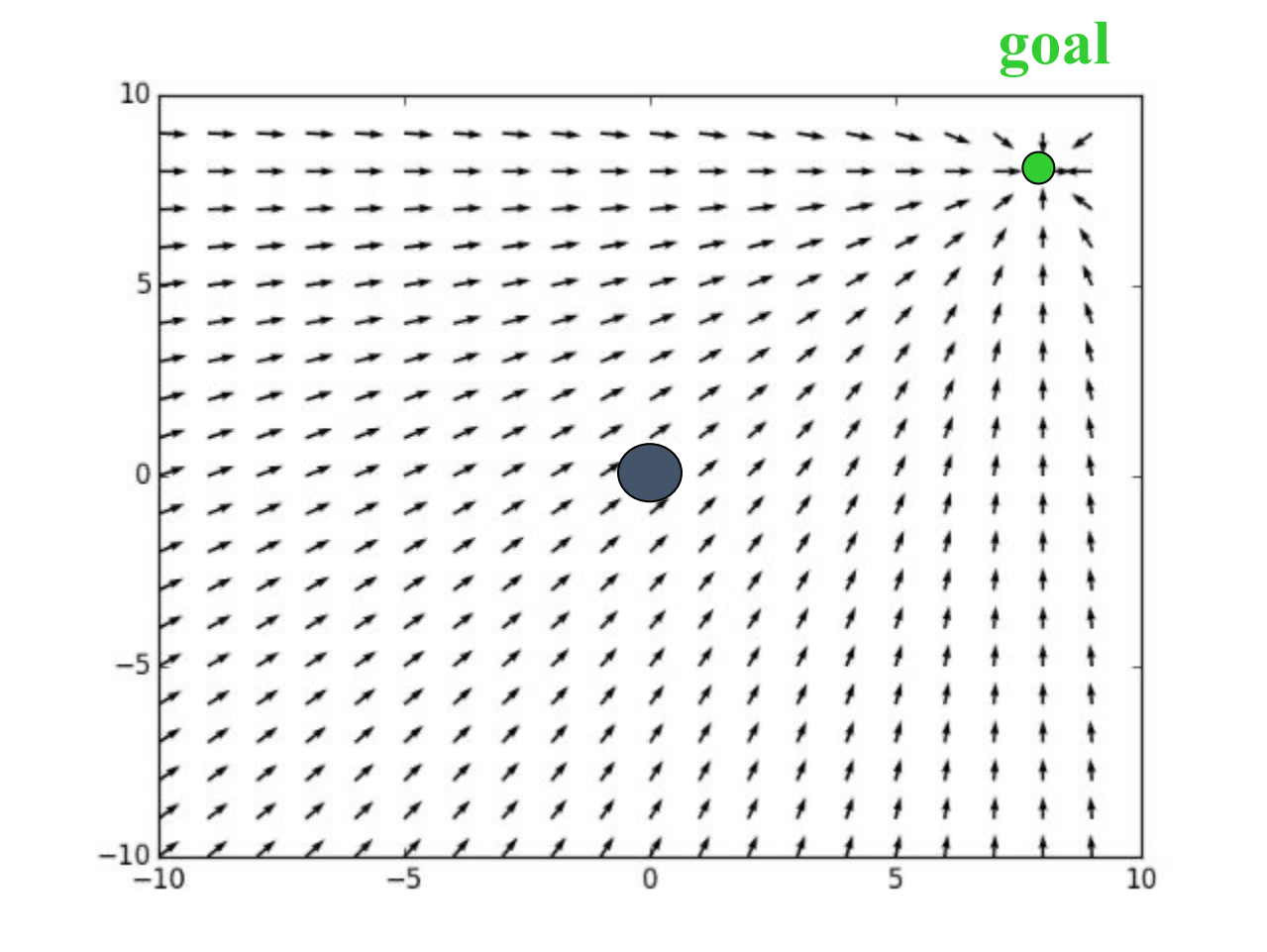
- Example – 2 behaviors
  - Go to goal (in top right corner)
  - Avoid obstacle (in center of grid)



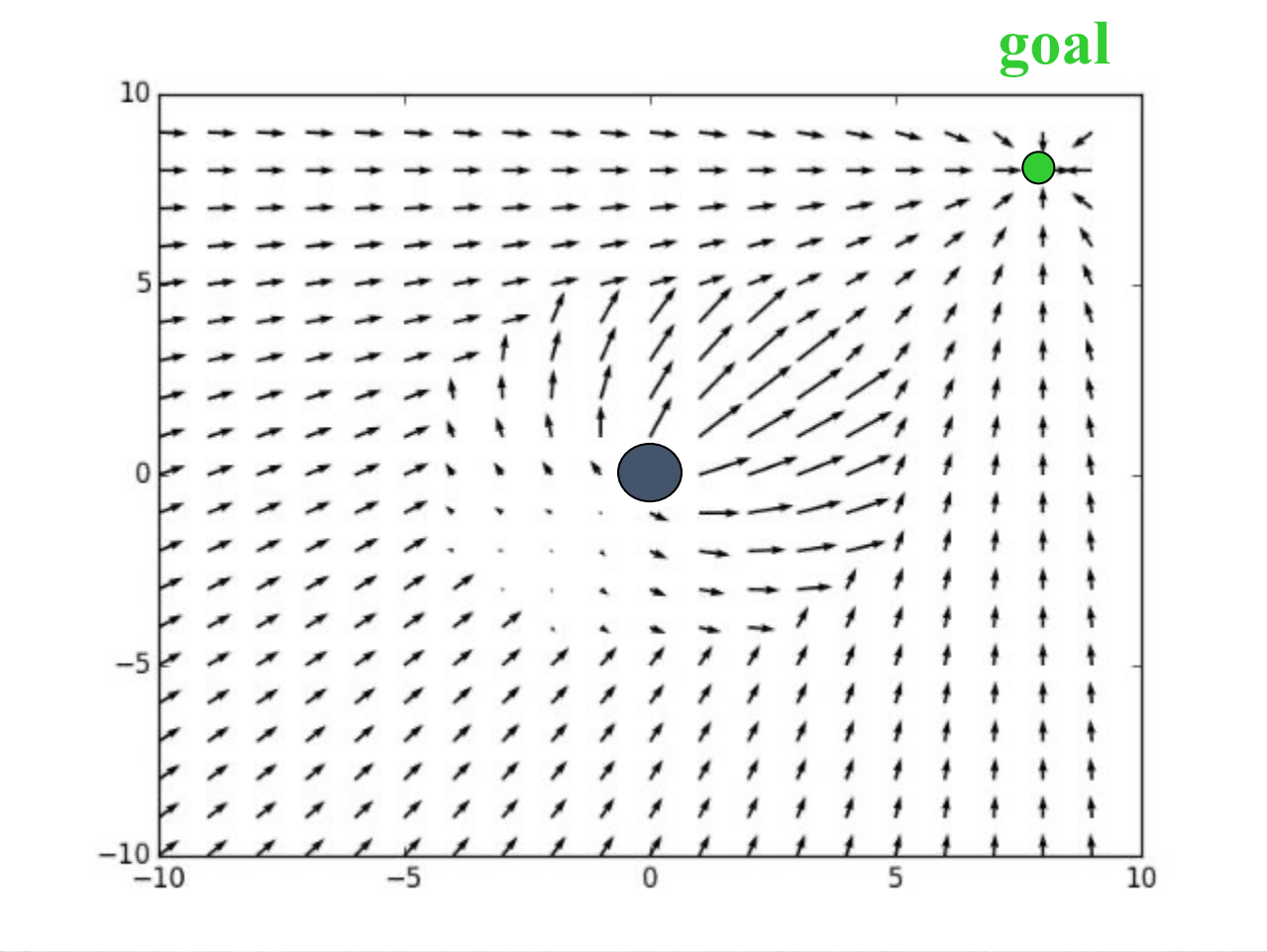
Behavior 1: Repulsive Field



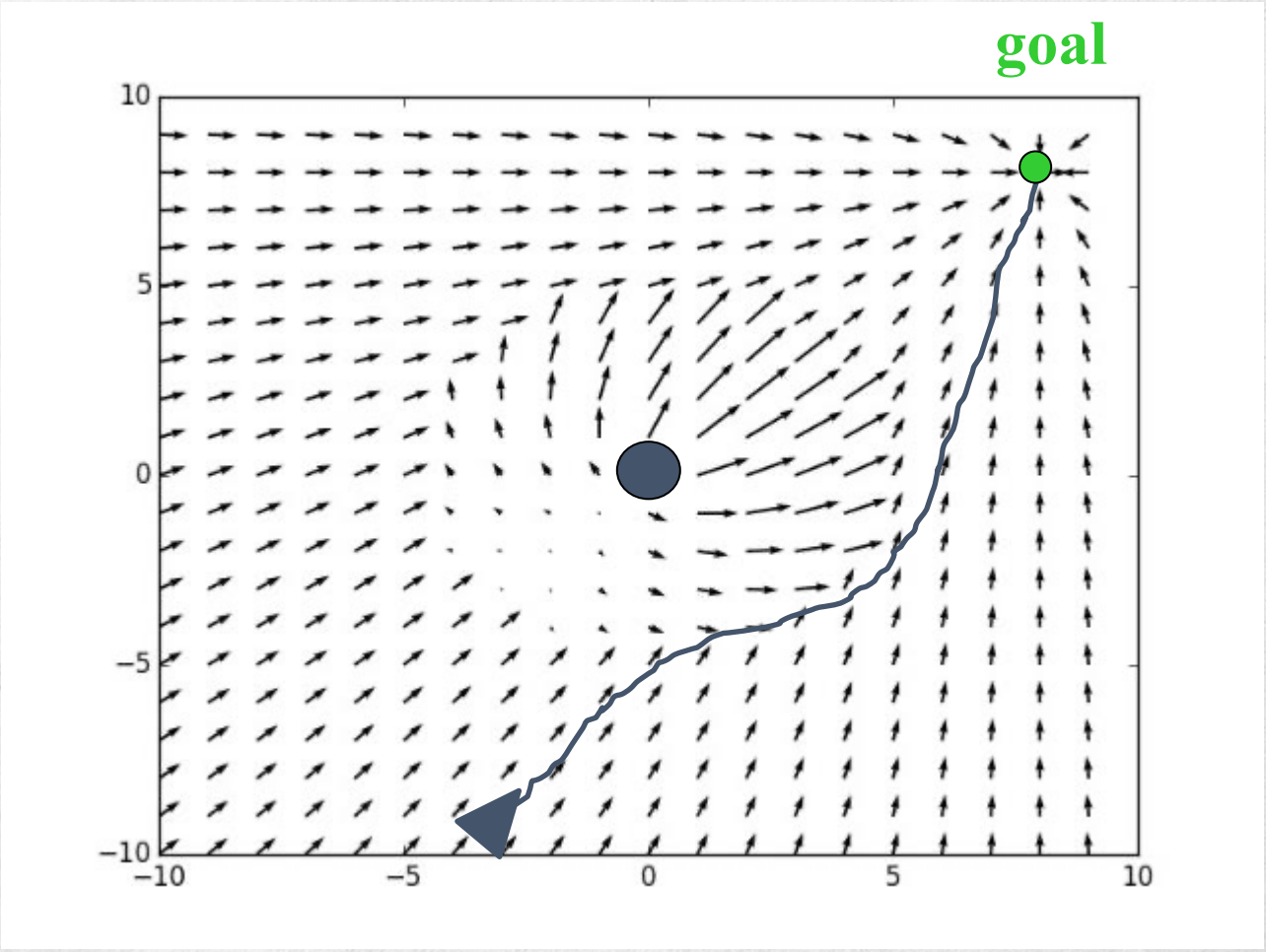
Behavior 2: Attractive Field



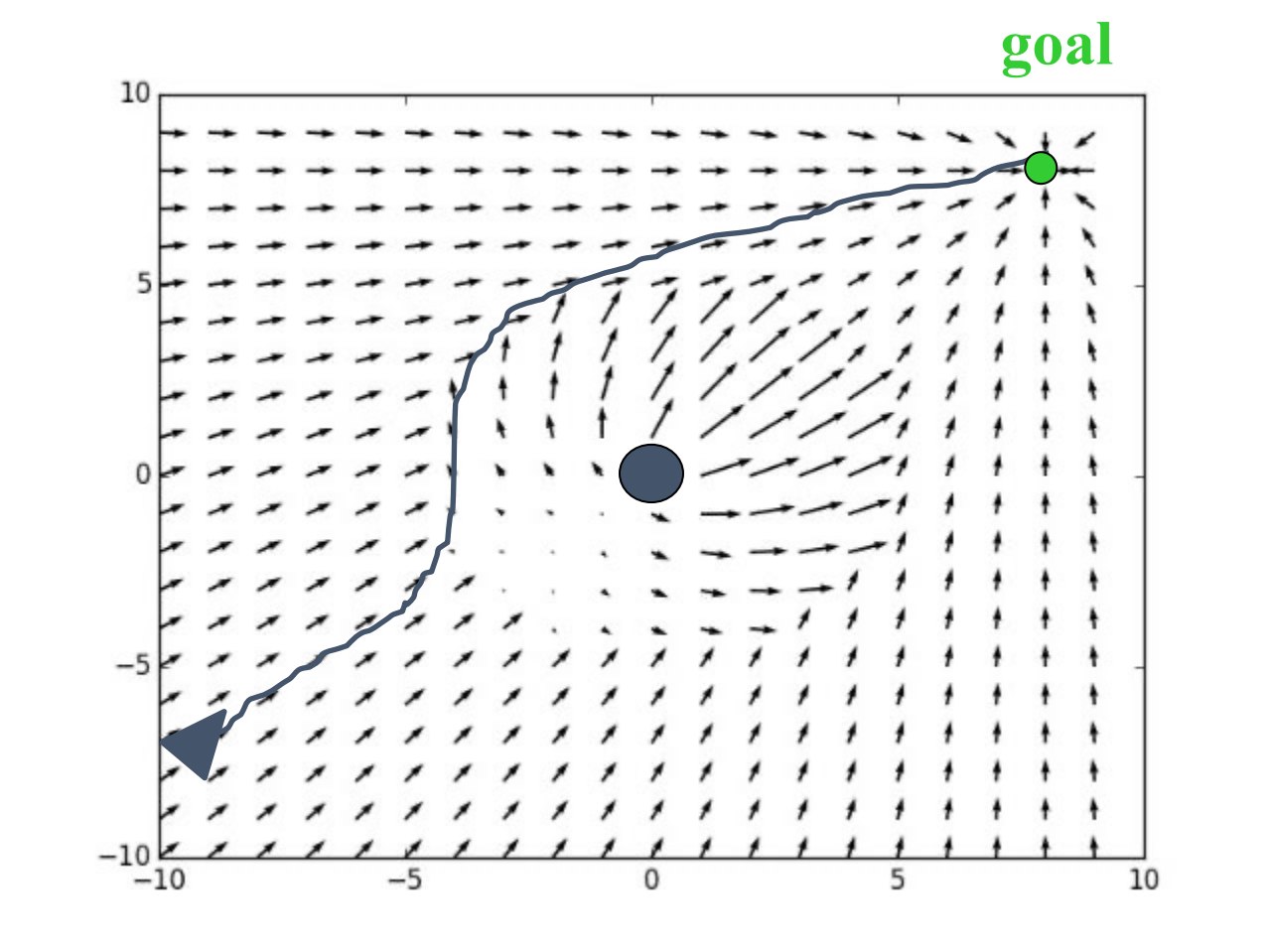
Both Behaviors Together



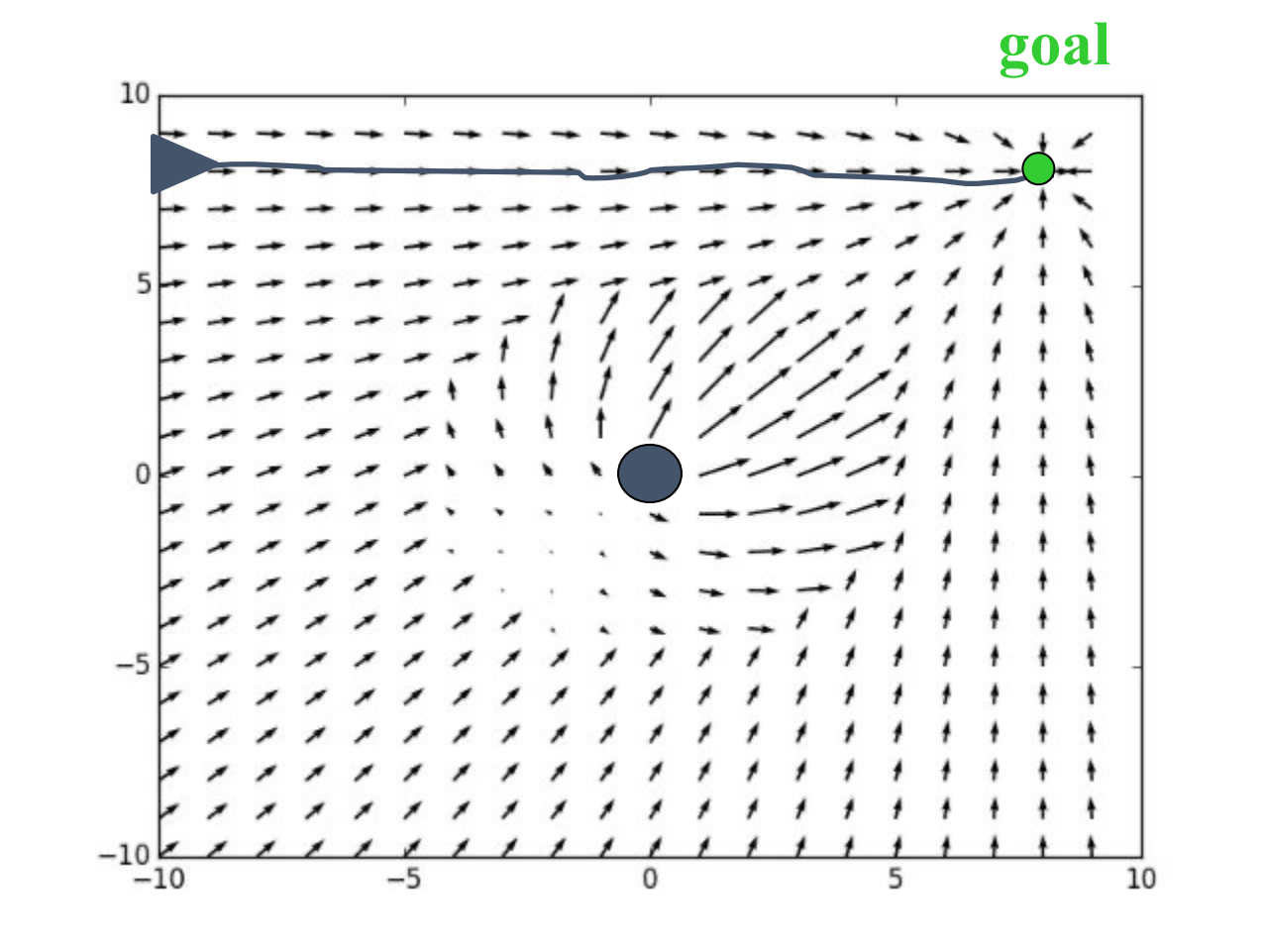
How would a robot move in Field?



How would a robot move in Field?



How would a robot move in Field?

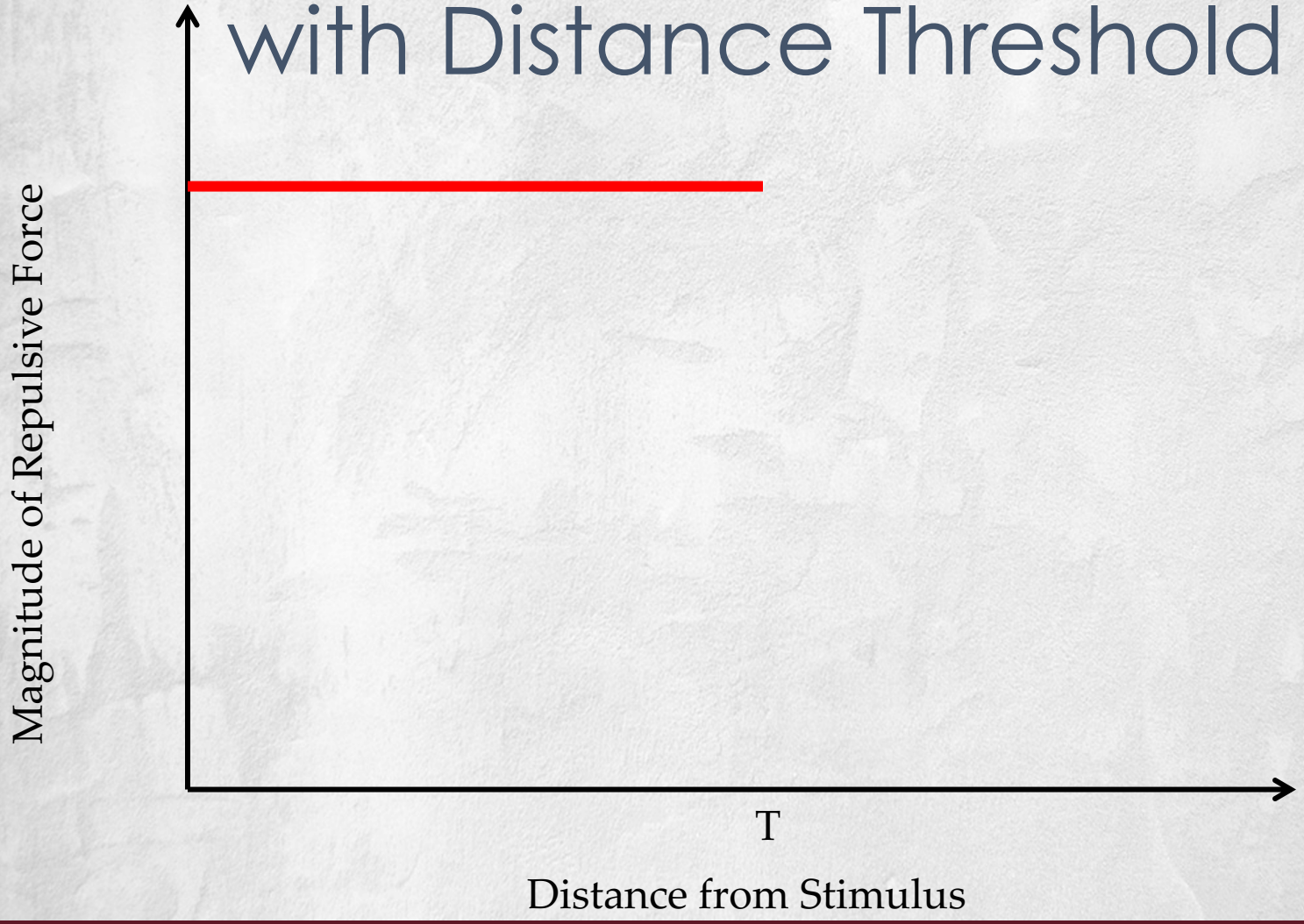


Magnitude Profile

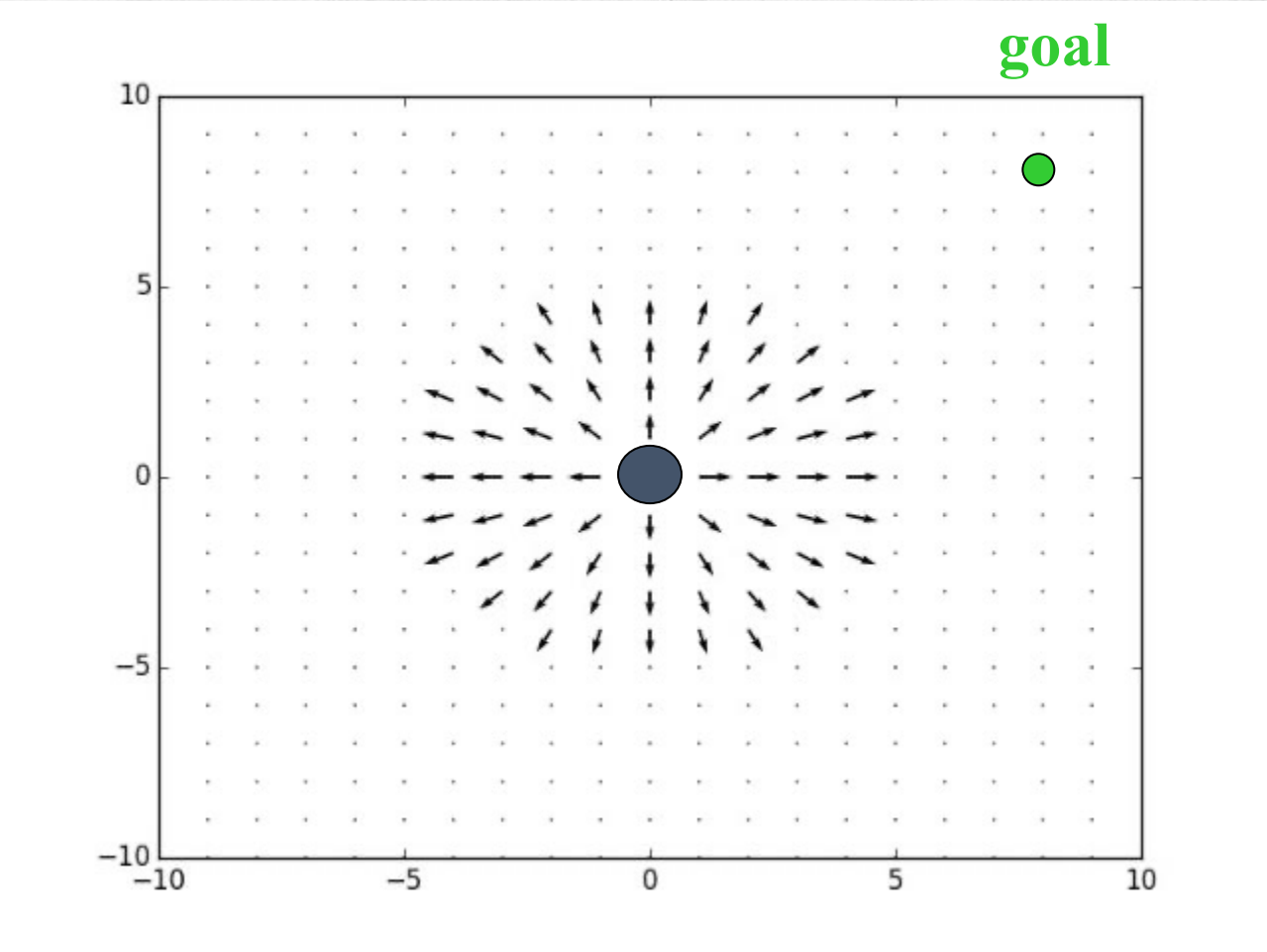
Magnitude Profiles

- Constant with distance threshold
  - If closer than distance, feel constant repulsive force
  - If further away, feel no force at all
- Linear drop-off
  - Maximum strength right next to the obstacle
  - Decreases linearly to 0 at the maximum distance
- Exponential drop-off
  - Maximum strength right next to the obstacle
  - Decreases exponentially as distance increases
  - Never actually 0, but usually ignored past some maximum distance from obstacle

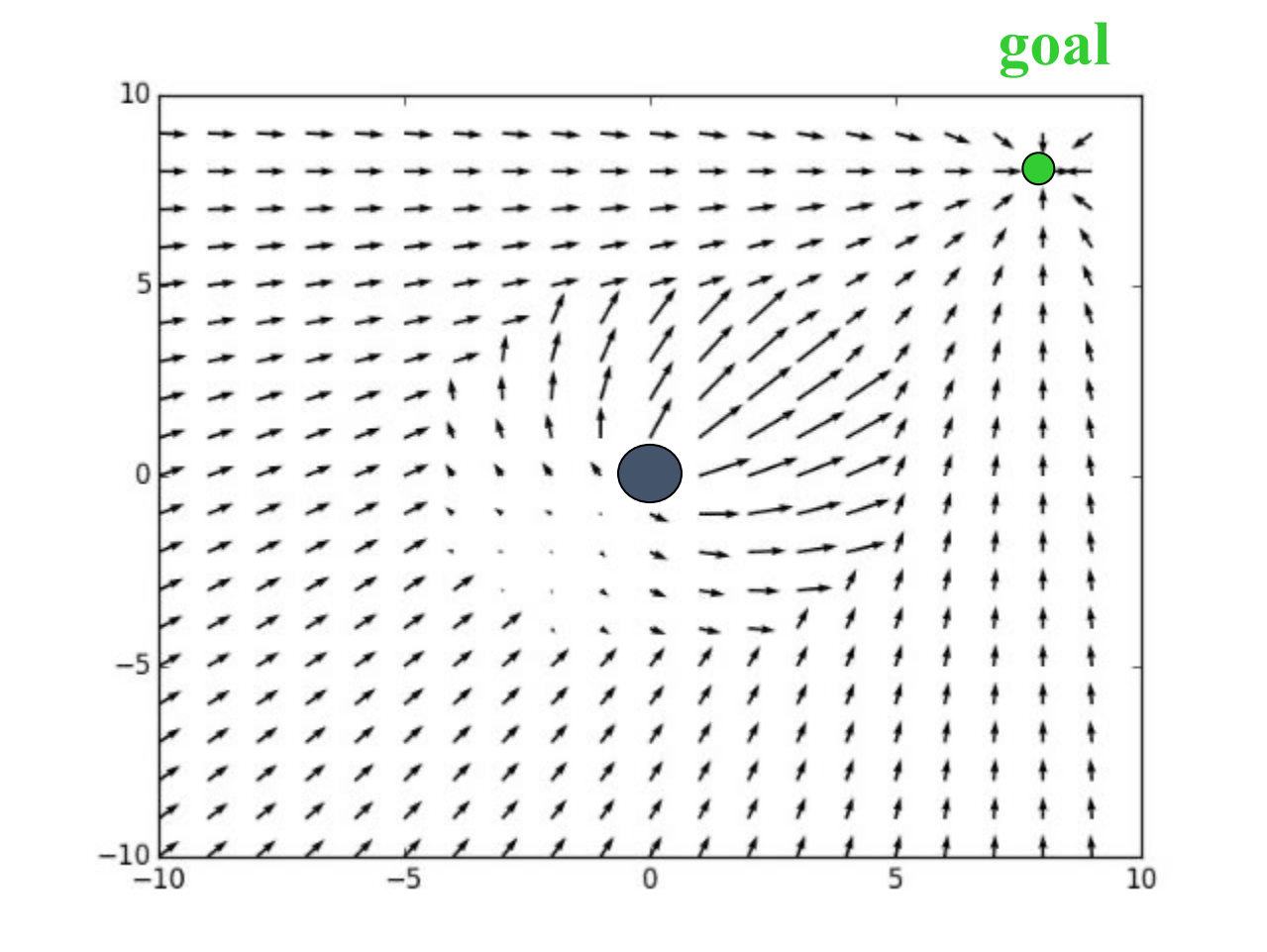
Magnitude Profile: Constant with Distance Threshold



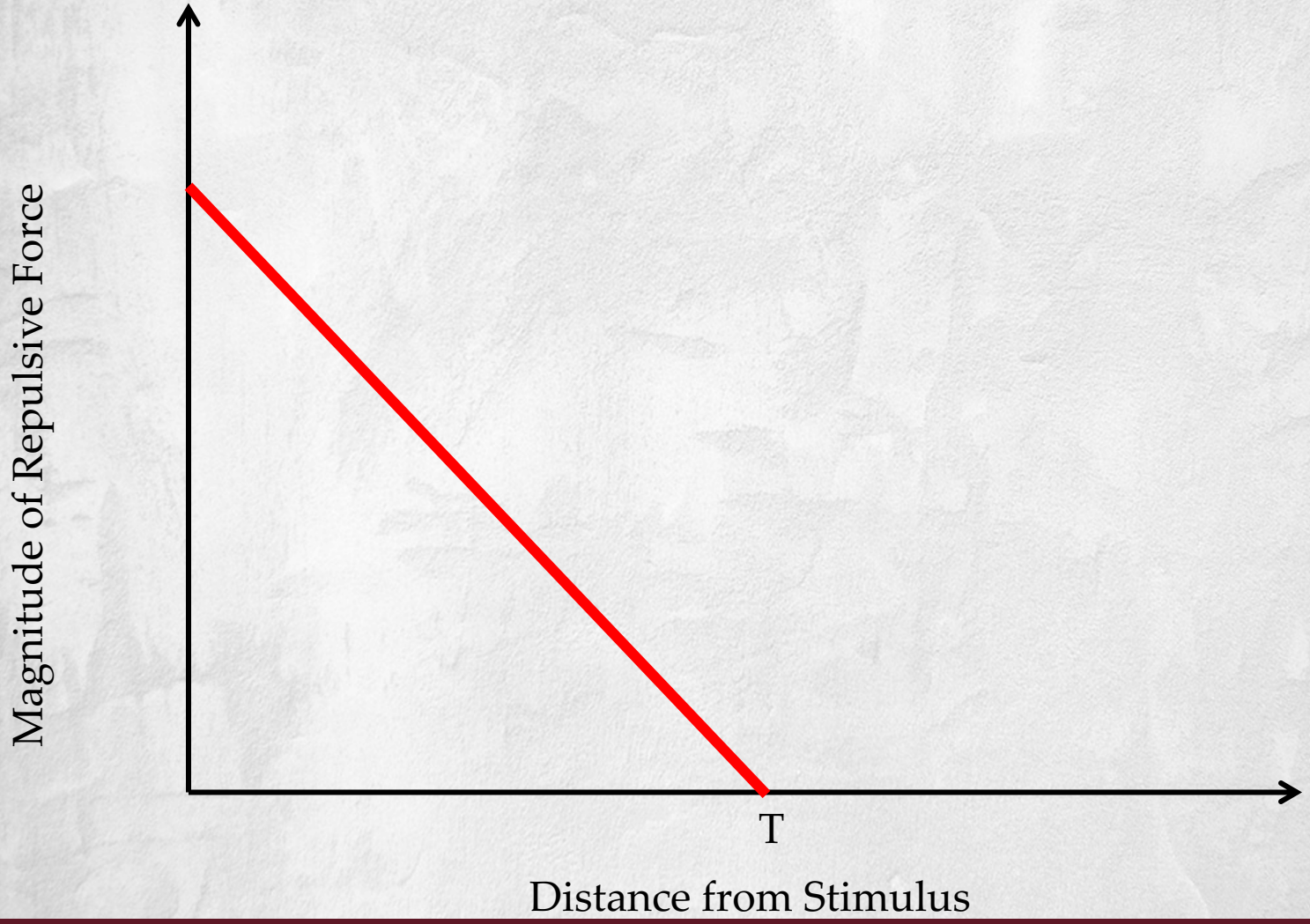
Constant with Distance Threshold



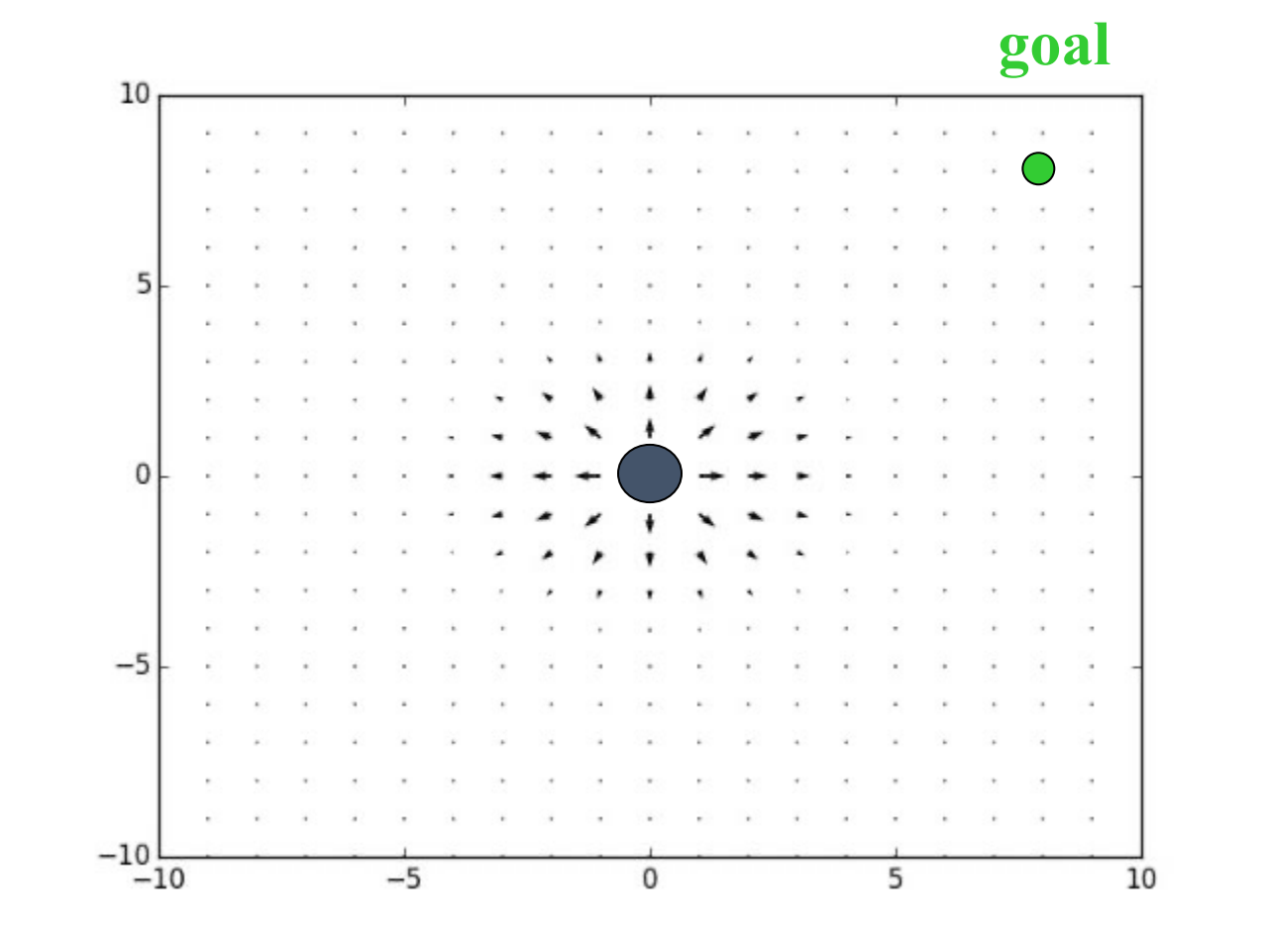
Constant + Attractive



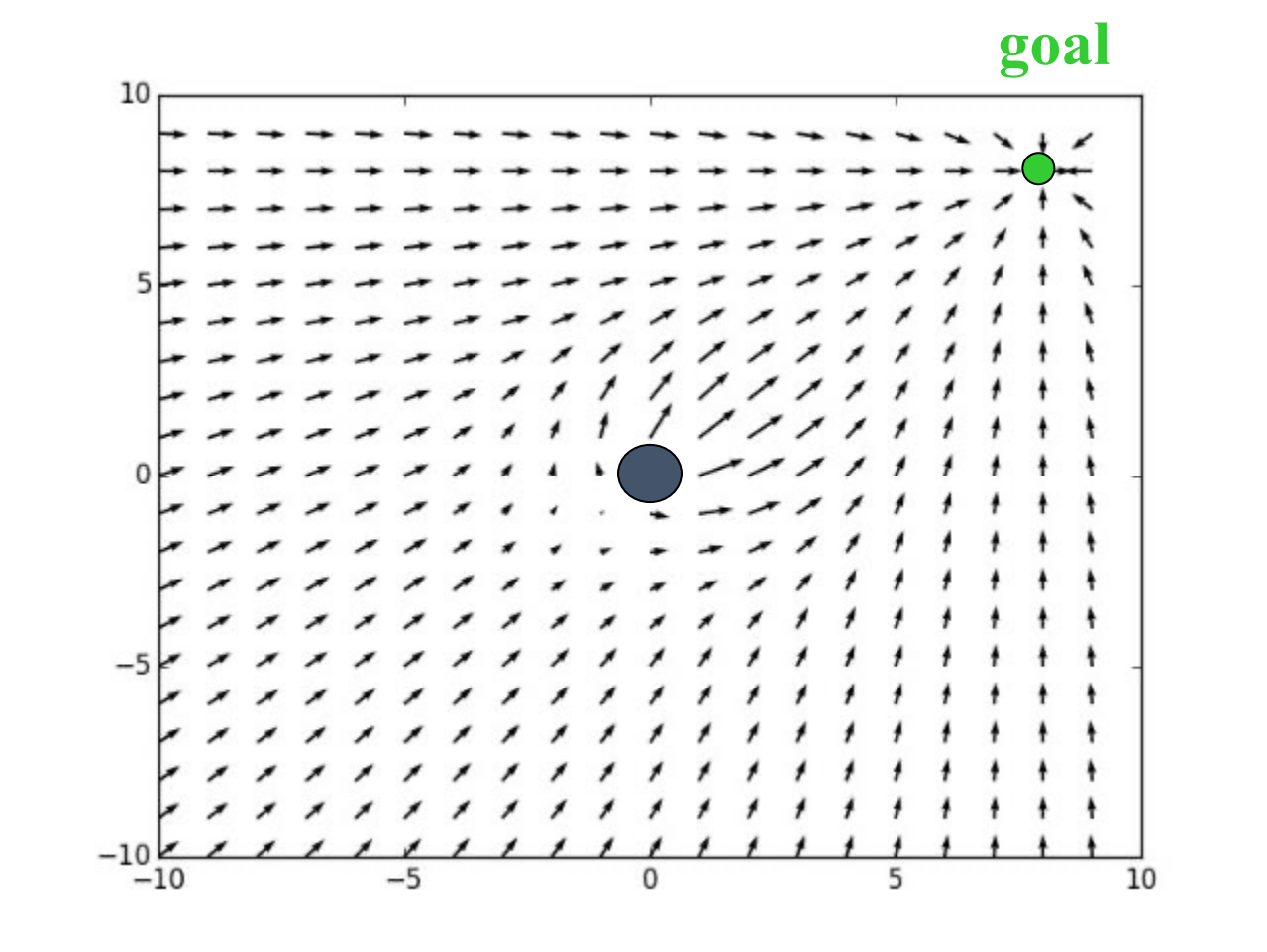
Magnitude Profile: Linear Drop-Off



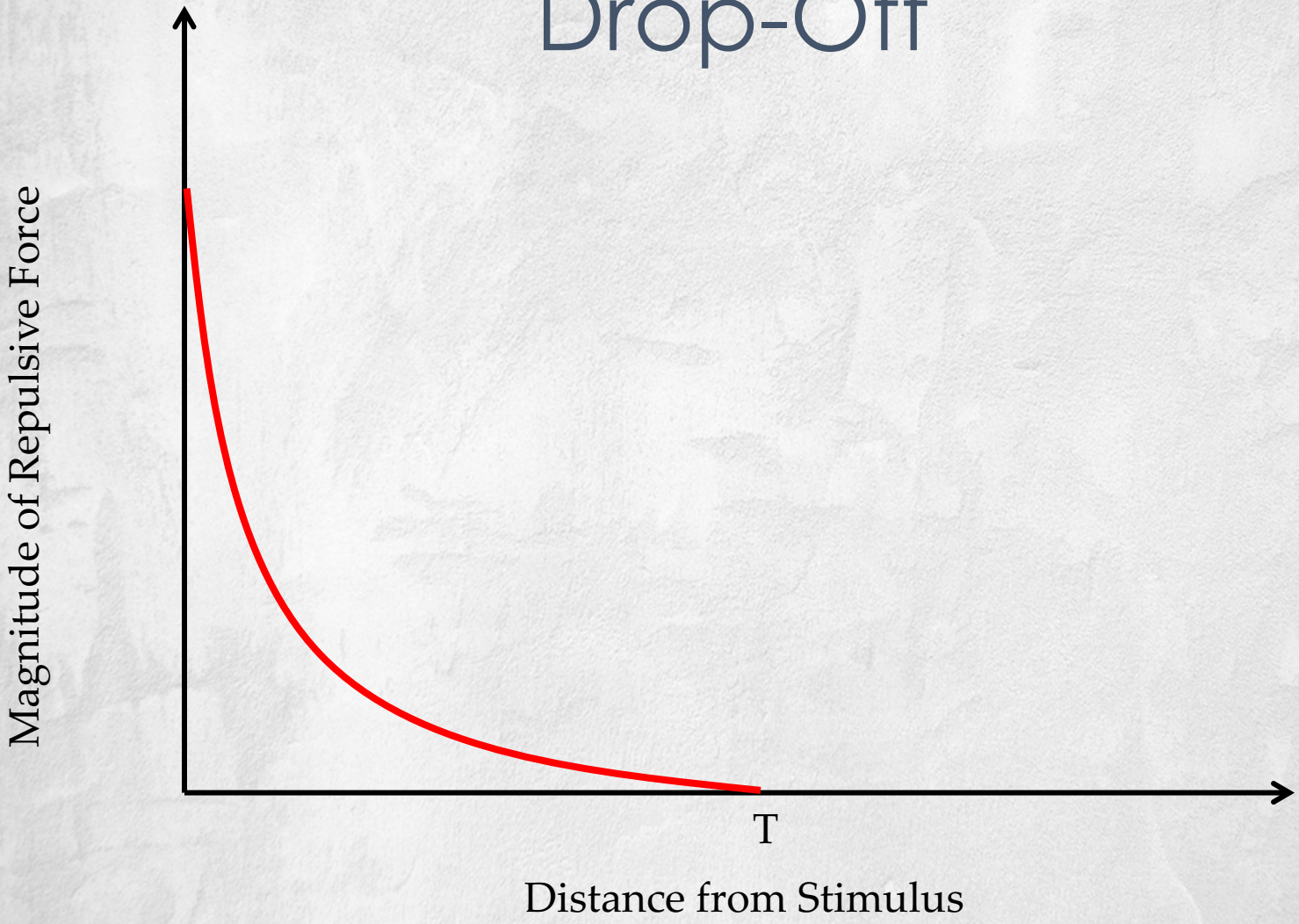
Linear Drop-off



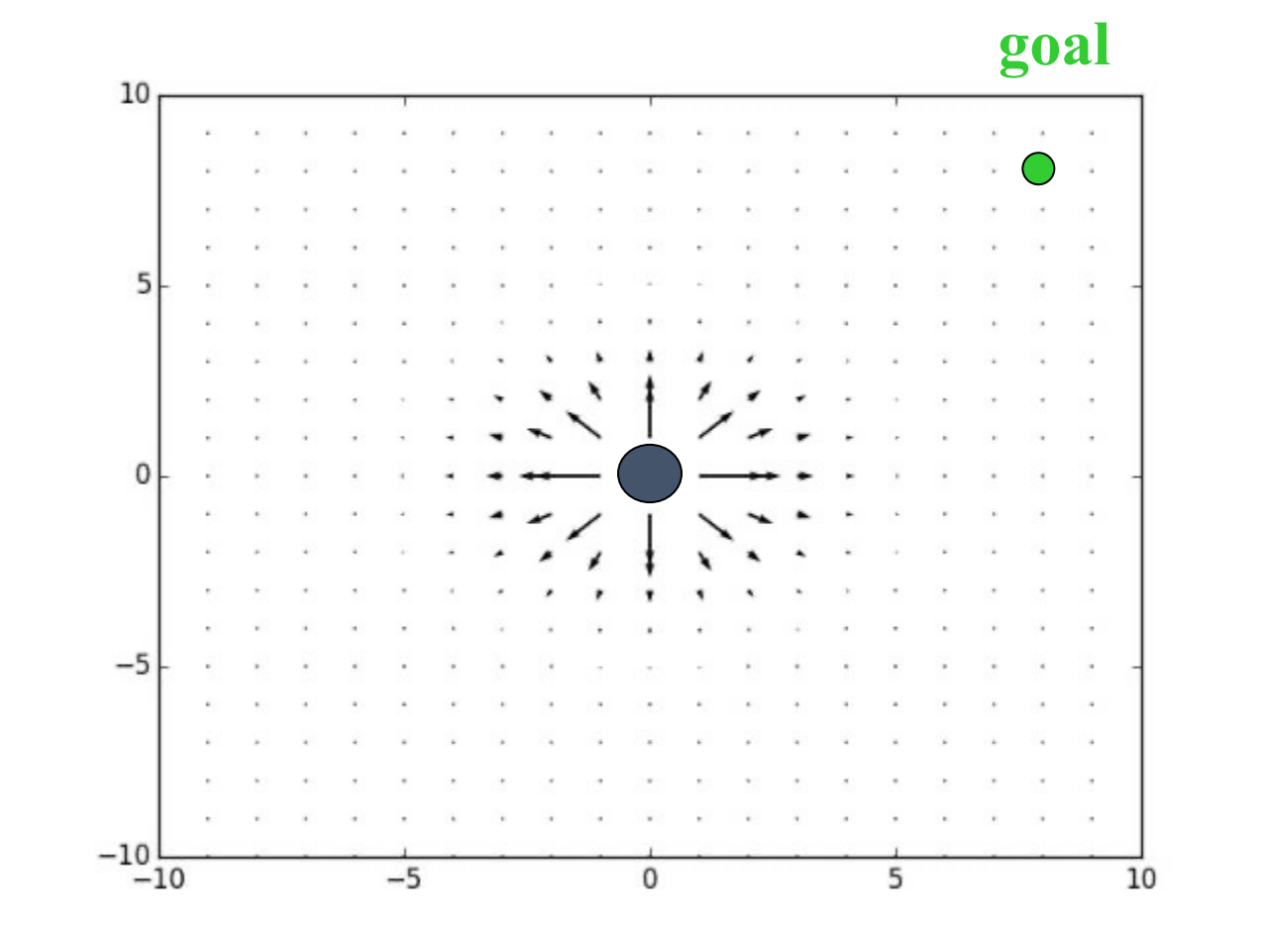
Linear + Attractive



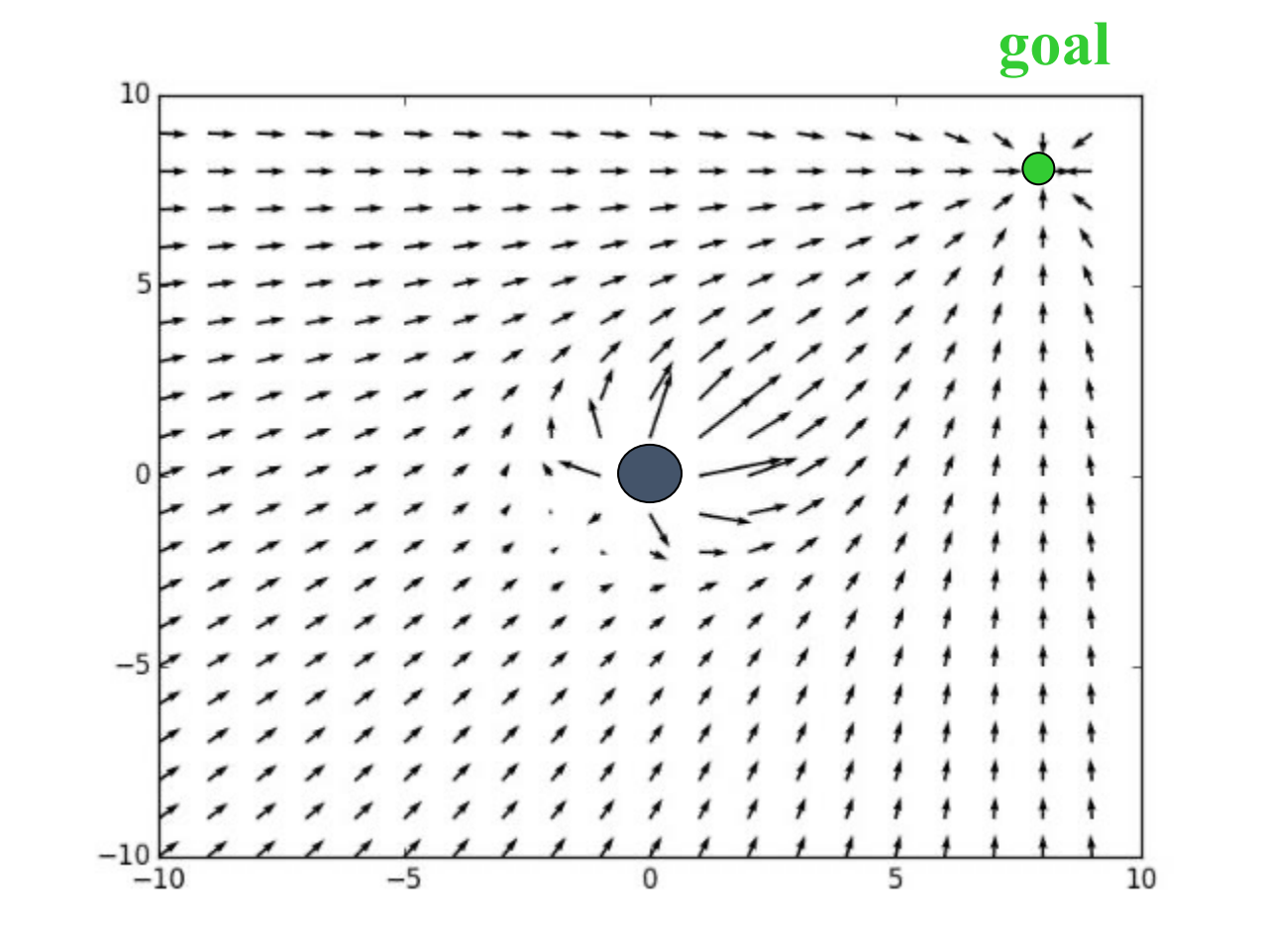
Magnitude Profile: Exponential Drop-Off



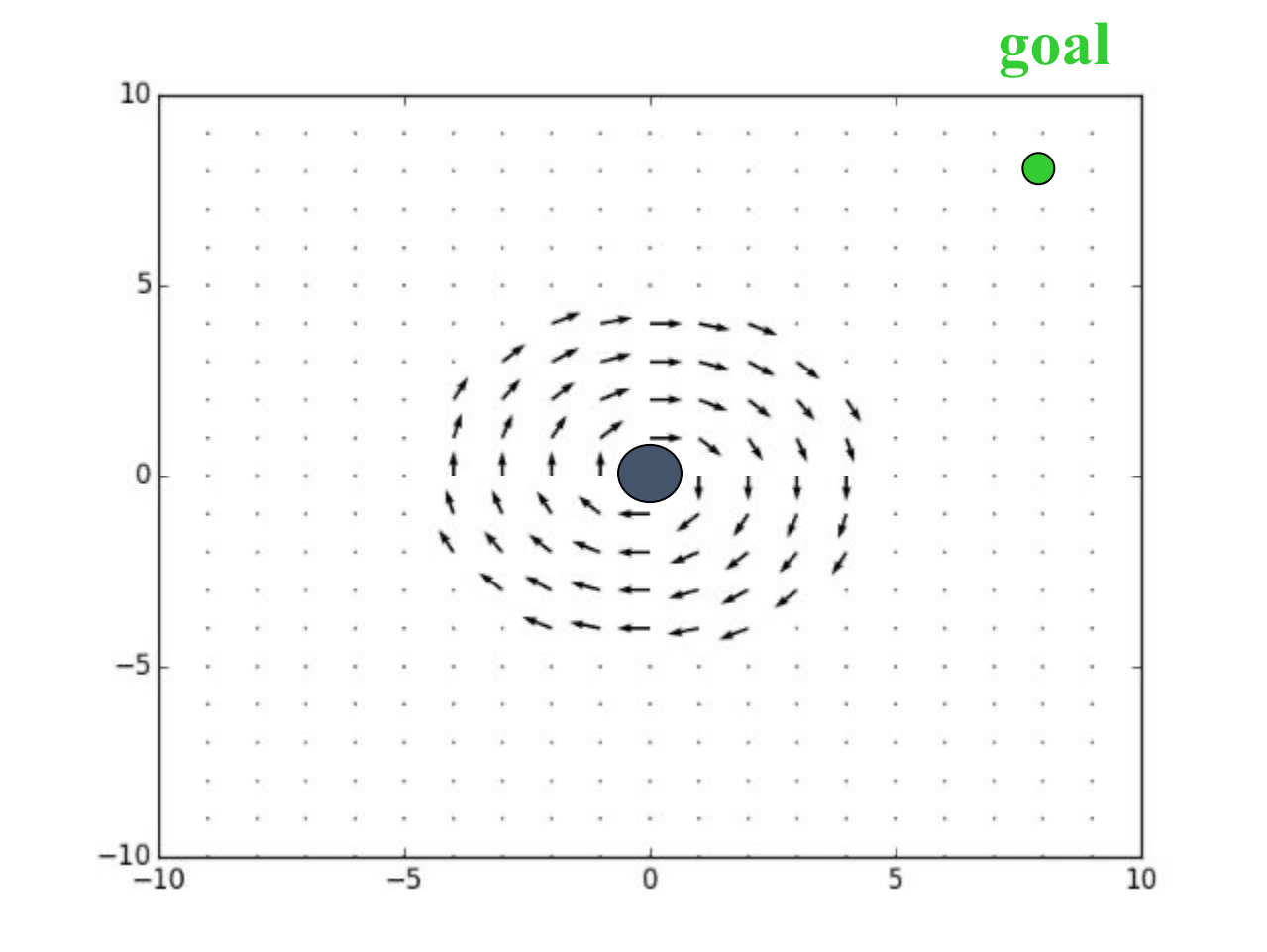
Exponential Drop-off



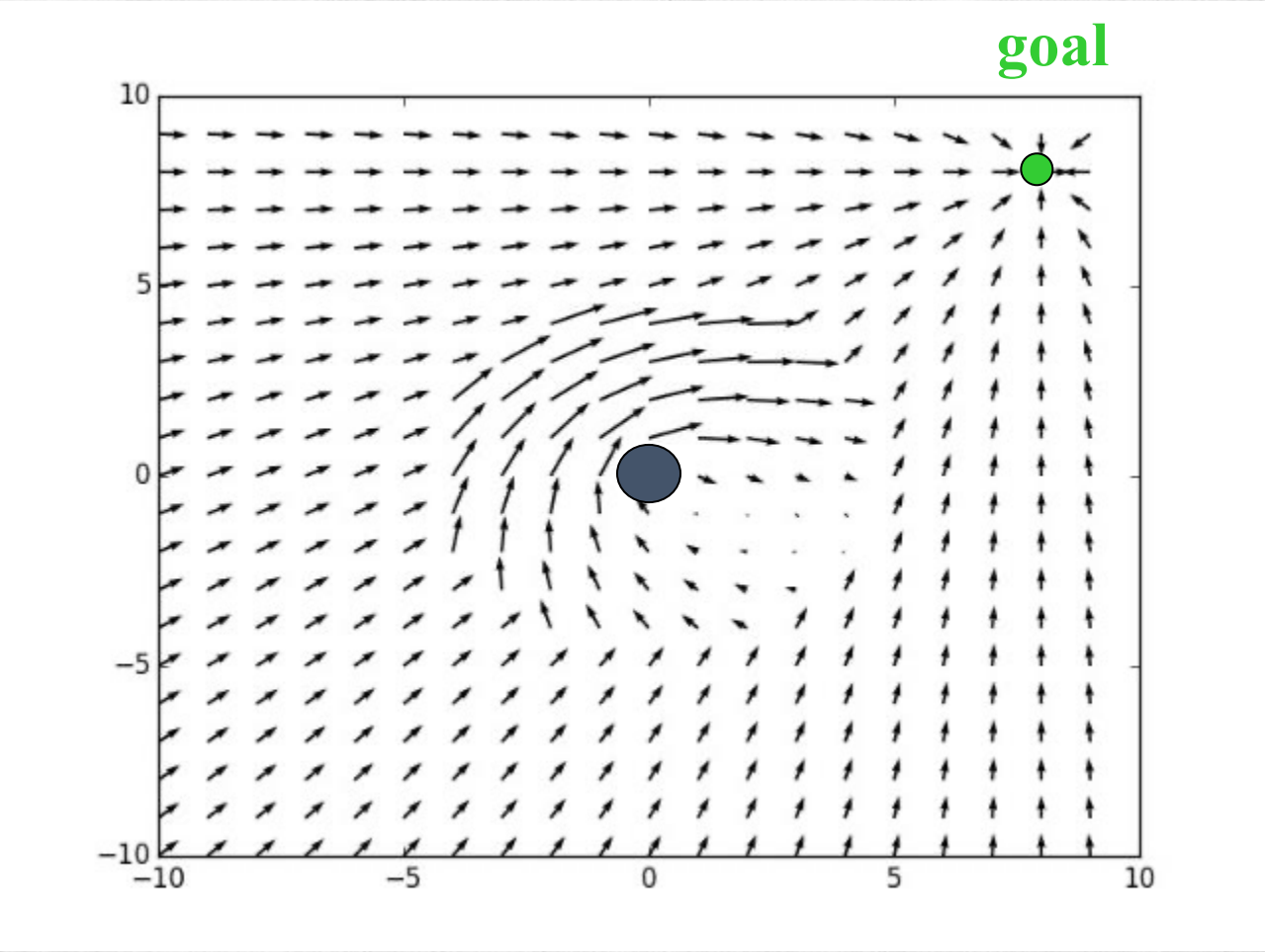
Exponential + Attractive



Tangential (Constant w/ threshold)



Tangential + Attractive

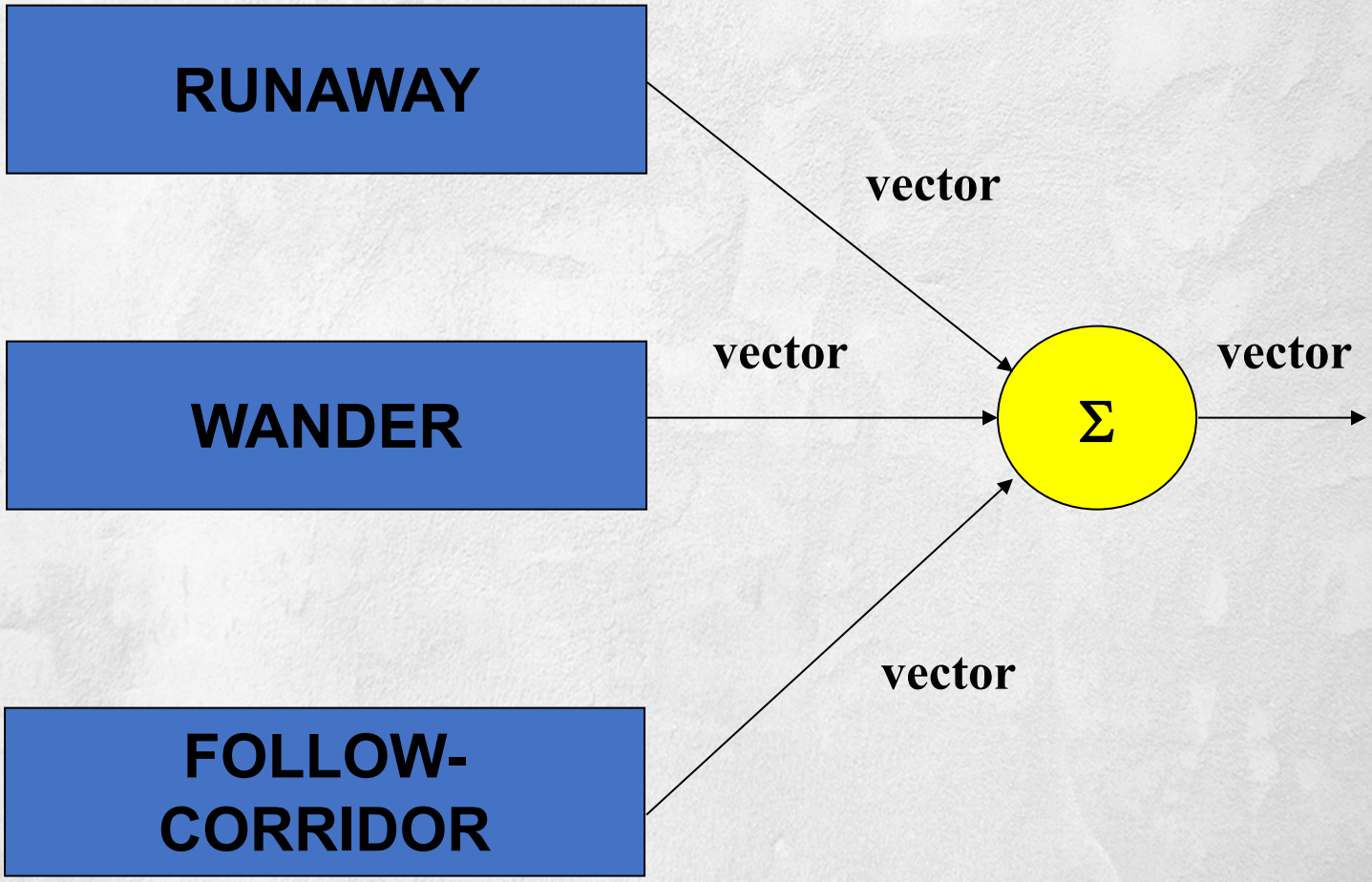


Behaviors

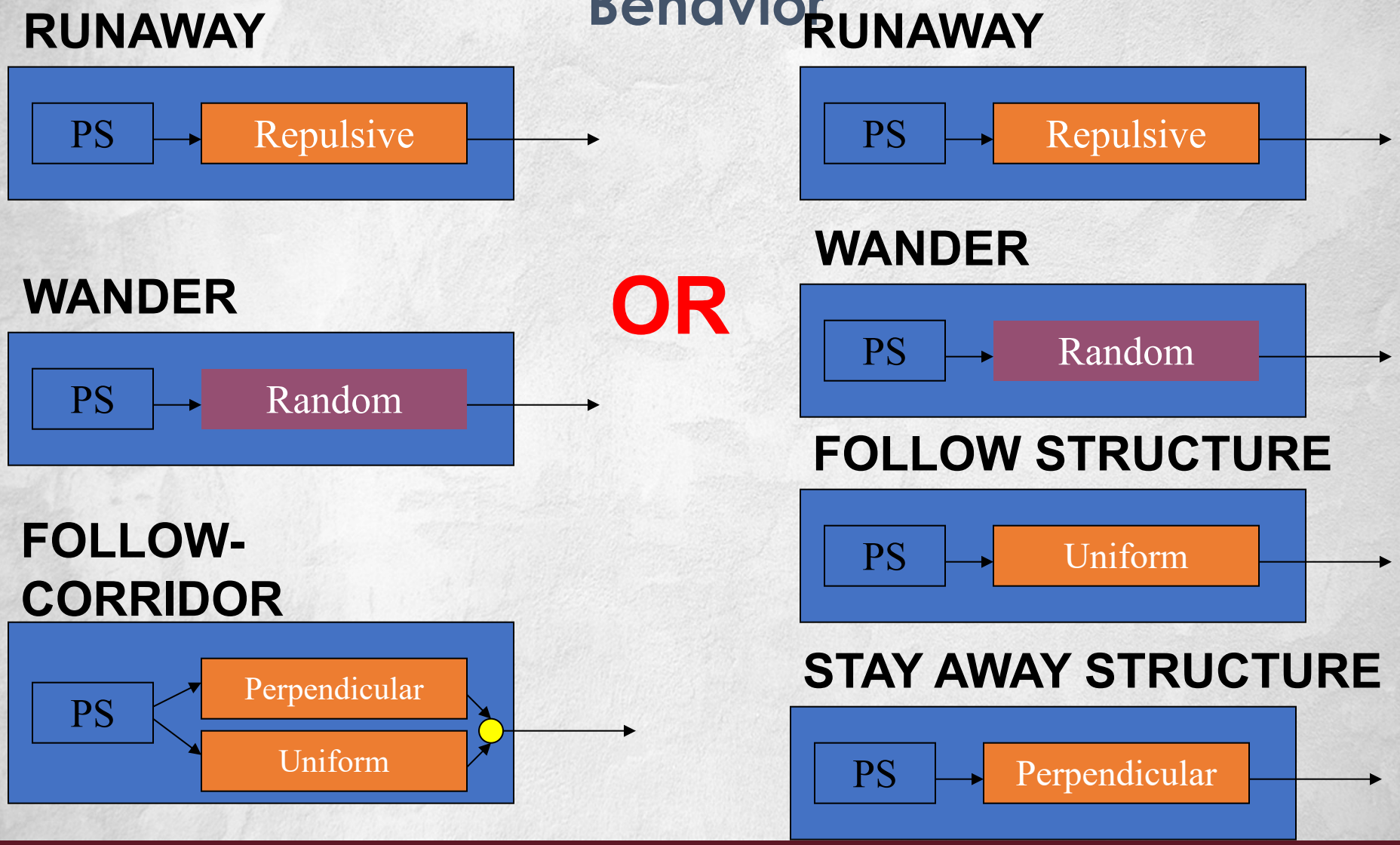
Example: A Corridor-Following Robot:

- It should WANDER until it finds a corridor
- When it finds a corridor it should FOLLOW-CORRIDOR
- If it encounters any obstacles, it should RUNAWAY

Example: Behaviors for This Robot

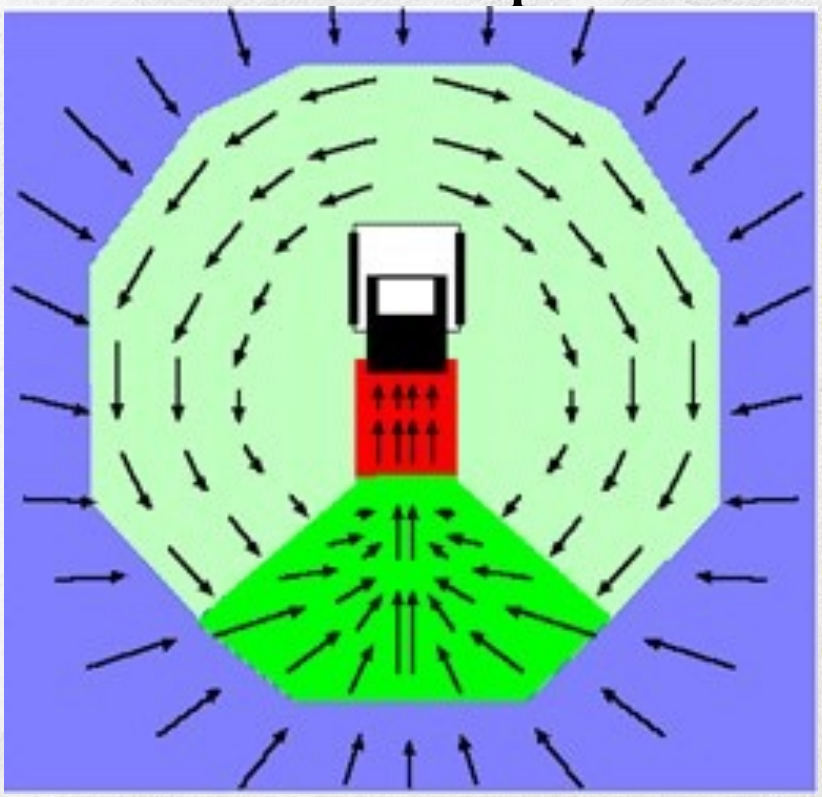


Example: Alternative Implementation: 1 per Behavior



Another Example: Docking Behavior

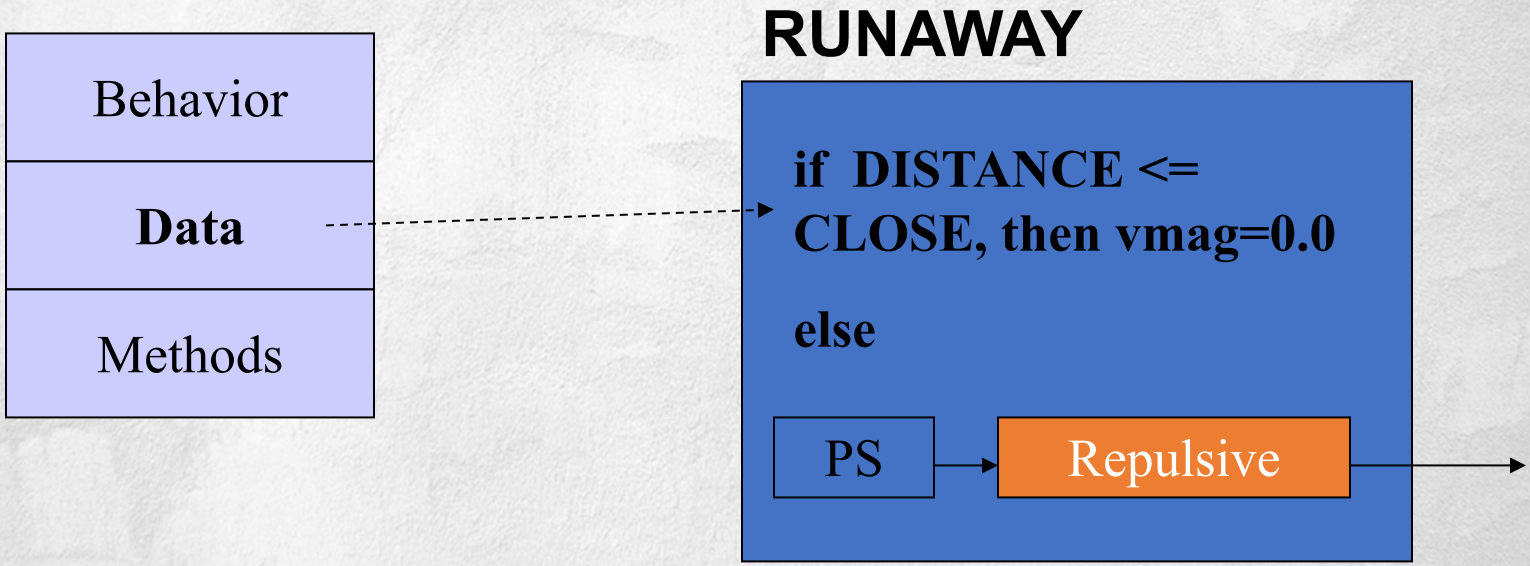
- Combination of multiple different fields:



Potential problems

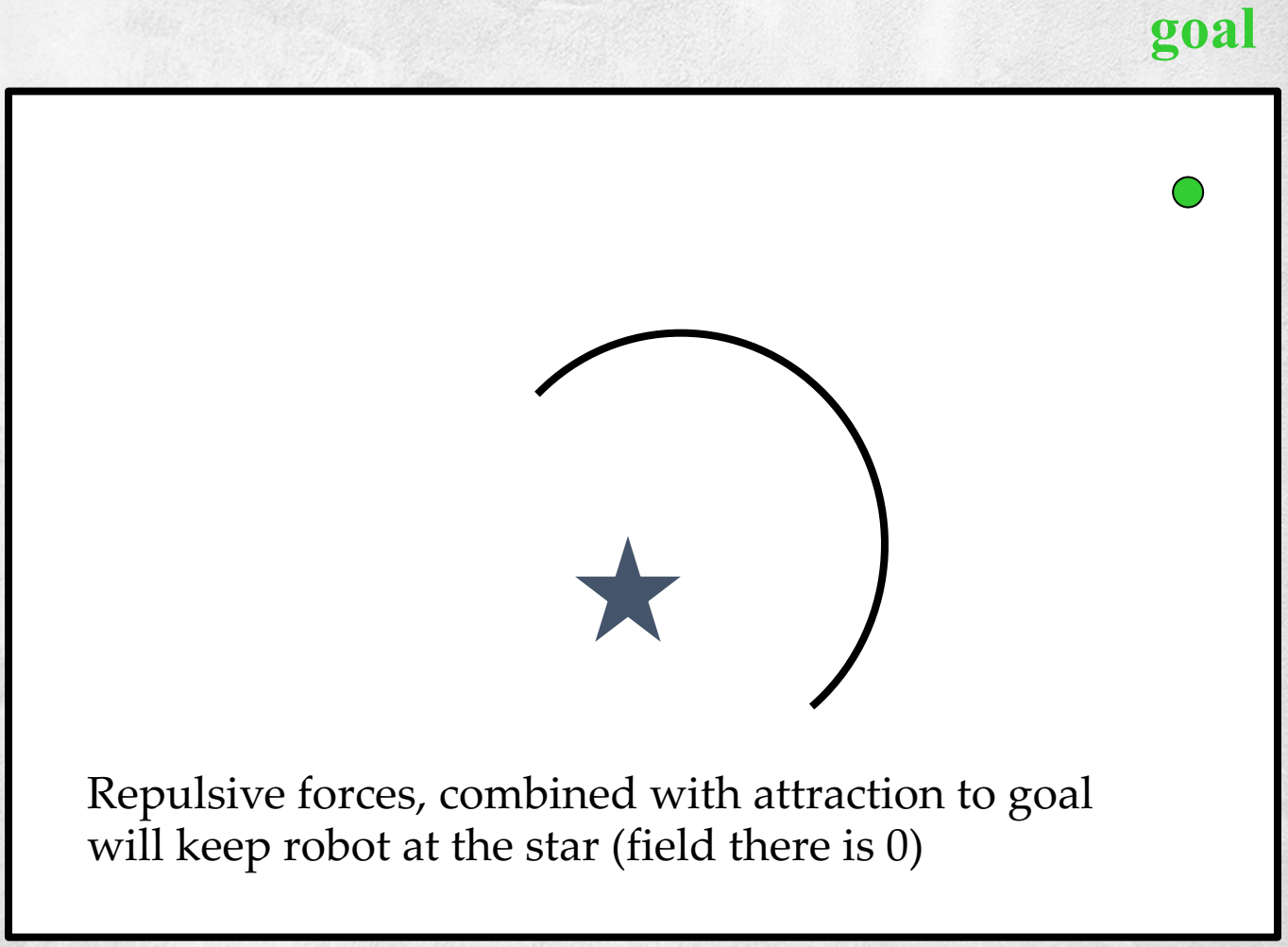
RUNAWAY behavior

- Most RUNAWAY behaviors have a "panic" aspect
  - If "really close" or "touching": STOP, rather than turn and risk damaging robot or environment by turning



  - But this will also produce a (0, 0) vector which won't stop

Local Minima



Local Minima - Solutions

- Use Tangential Fields around obstacles
- Apply random uniform field if stuck
- Remember where robot has been, make those waypoints obstacles that robot also avoids, try new paths
- Navigation templates:
  - Obstacle avoidance takes into account desired direction
- Harmonic functions

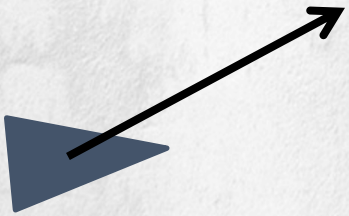
Summary

Potential Field

- Advantages
  - Easy to visualize
  - Easy to implemented
  - Fields can be parameterized and combined
- Disadvantages
  - Local minima
  - Jerky motion
  - Require hand-design and tuning of parameters

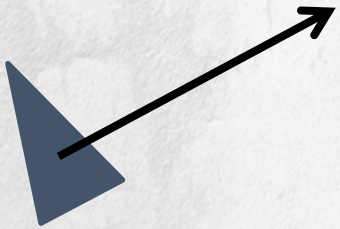
A technicality

- What is the angle of the vector from the robot to the goal?



A technicality

- What is the angle of the vector from the robot to the goal?



Note: Reference Frames

- Global Coordinates
  - +x = east, +y = north, etc.
  - angle is fixed direction on map (0 = east, π = west, etc)
- Robot/Local coordinates
  - +x = straight ahead, +y = turn left, etc.
  - angle is relative to robot's heading
- Make sure you know which you are using where and converting appropriately
  - To drive towards goal, you must use global/convert from global at some point
  - To control robot, you must use/convert to local at some point

References

1. https://medium.com/@rymshasiddiqui/path-planning-using-potential-field-algorithm-a30ad12bdb08