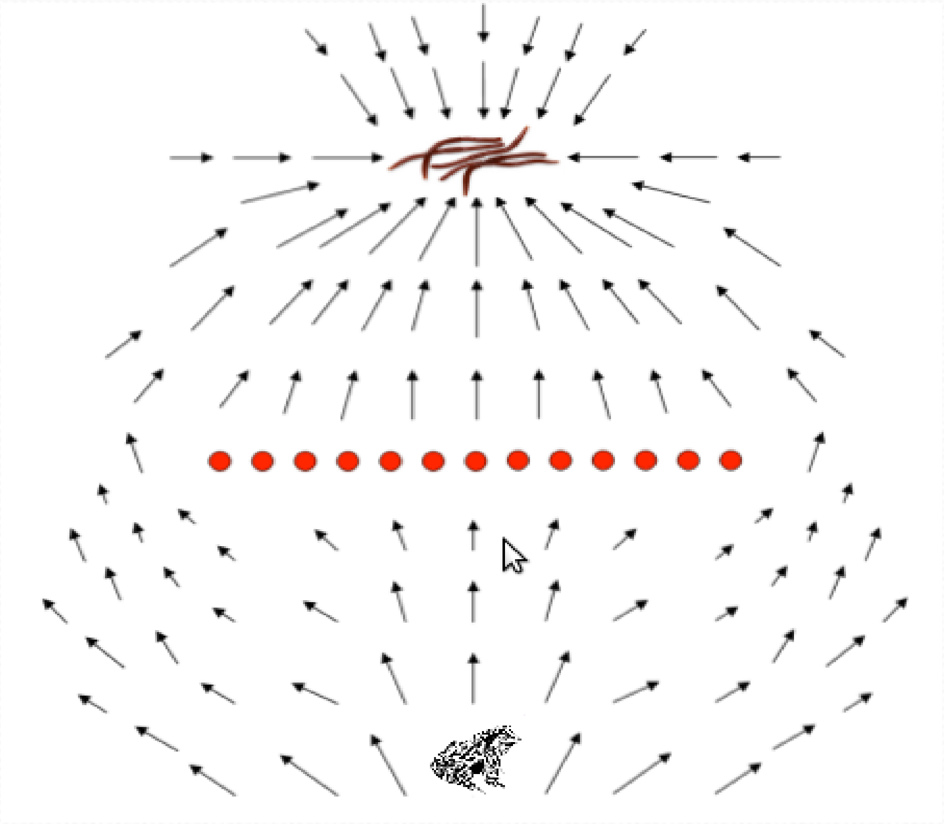
Behaviour Based Robot Control Architectures: Potential Fields

Introduction to Potential Fields

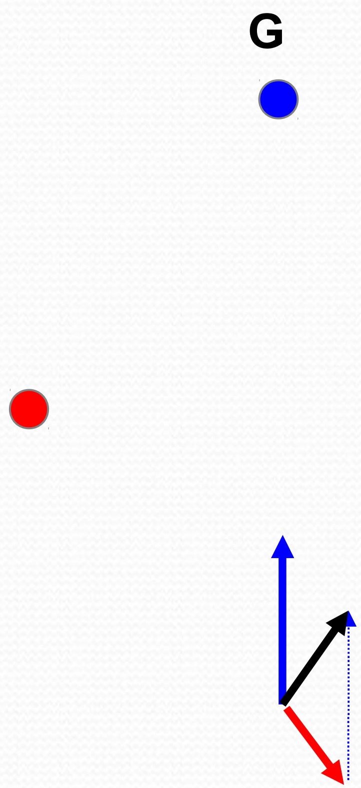


Arbib and House (1987) describe experiments with toads trying to reach some worms placed behind an obstacle fence. The observed motion of the toad is illustrated by the arrows in the figure above. The arrows indicate direction and speed of the toad when being placed at different locations. The motion can be described as a combination of attraction to the location of the worms and repulsion to the location of the fence.

Introduction to Potential Fields

U_{REP}: 1/(distance to obstacle)

U_{ATT}: distance to Goal

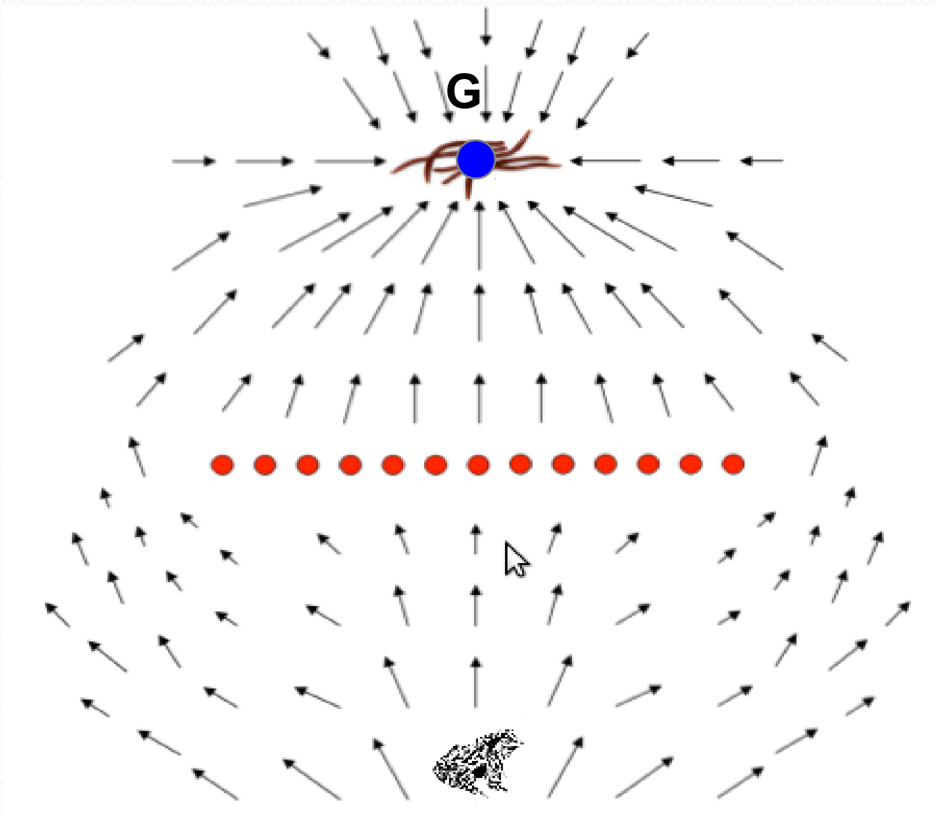


$$F_{REP} = - \nabla U_{REP}$$

$$F_{ATT} = - \nabla U_{ATT}$$

$$F = F_{REP} + F_{ATT}$$

Introduction to Potential Fields



U_{REP}: 1/(distance to obstacle)

U_{ATT}: distance to G

$F_{REP} = -\nabla U_{REP}$

$F_{ATT} = -\nabla U_{ATT}$

$F = F_{REP} + F_{ATT}$

$F(x,y) = \Sigma F_i(x,y)$

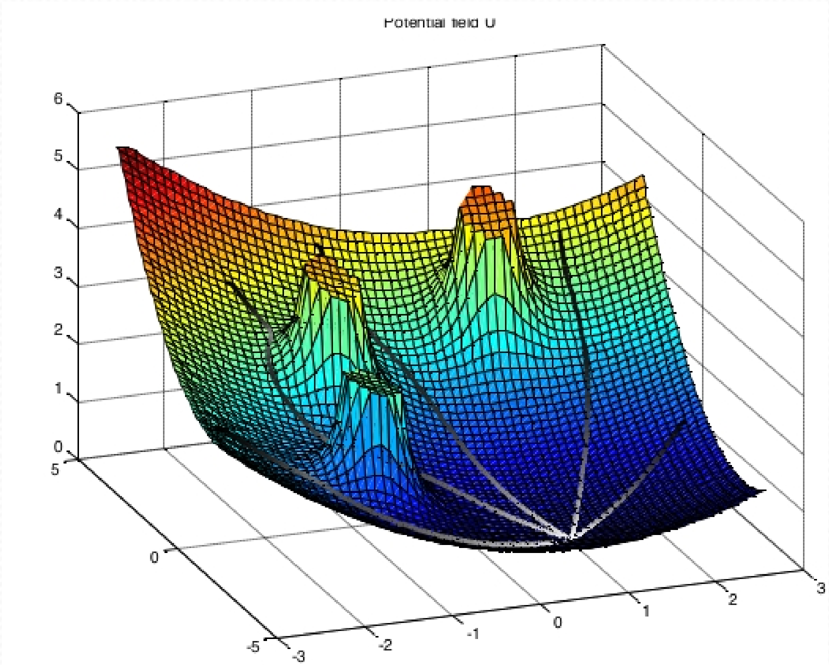
$F_i(x,y) = -\nabla u_i(x,y)$

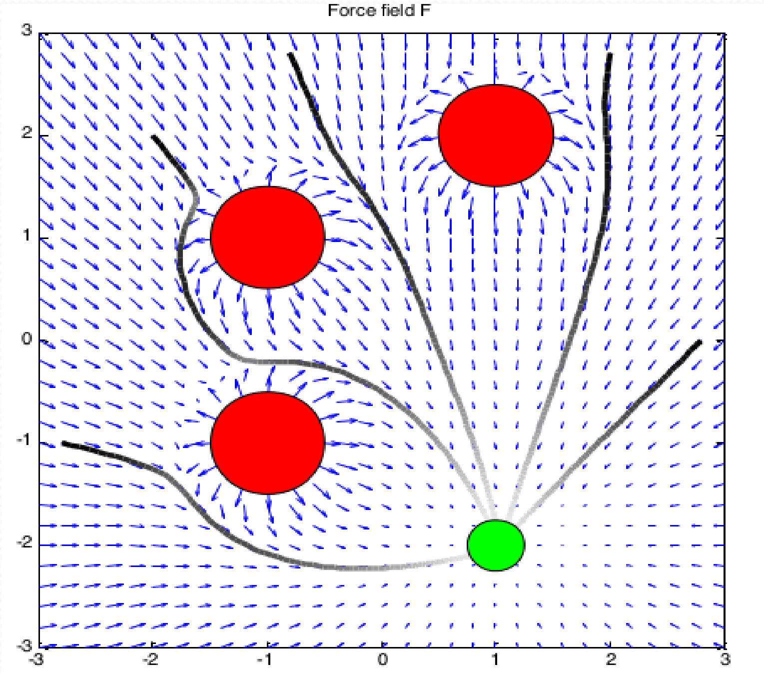
Potential Fields

- Vector space is 2D world, like bird's eye view of map
- Map divided into squares, creating (x,y) grid
- Each element represents square of space
- Perceivable objects in world exert a force field on surrounding space
- Algorithm

1. Compute force vectors Fi (x,y) for all potential fields
2. Combine all force vectors into F(x,y) using Equation 6.
3. Move along F(x,y) with speed proportional to |F(x,y)|.

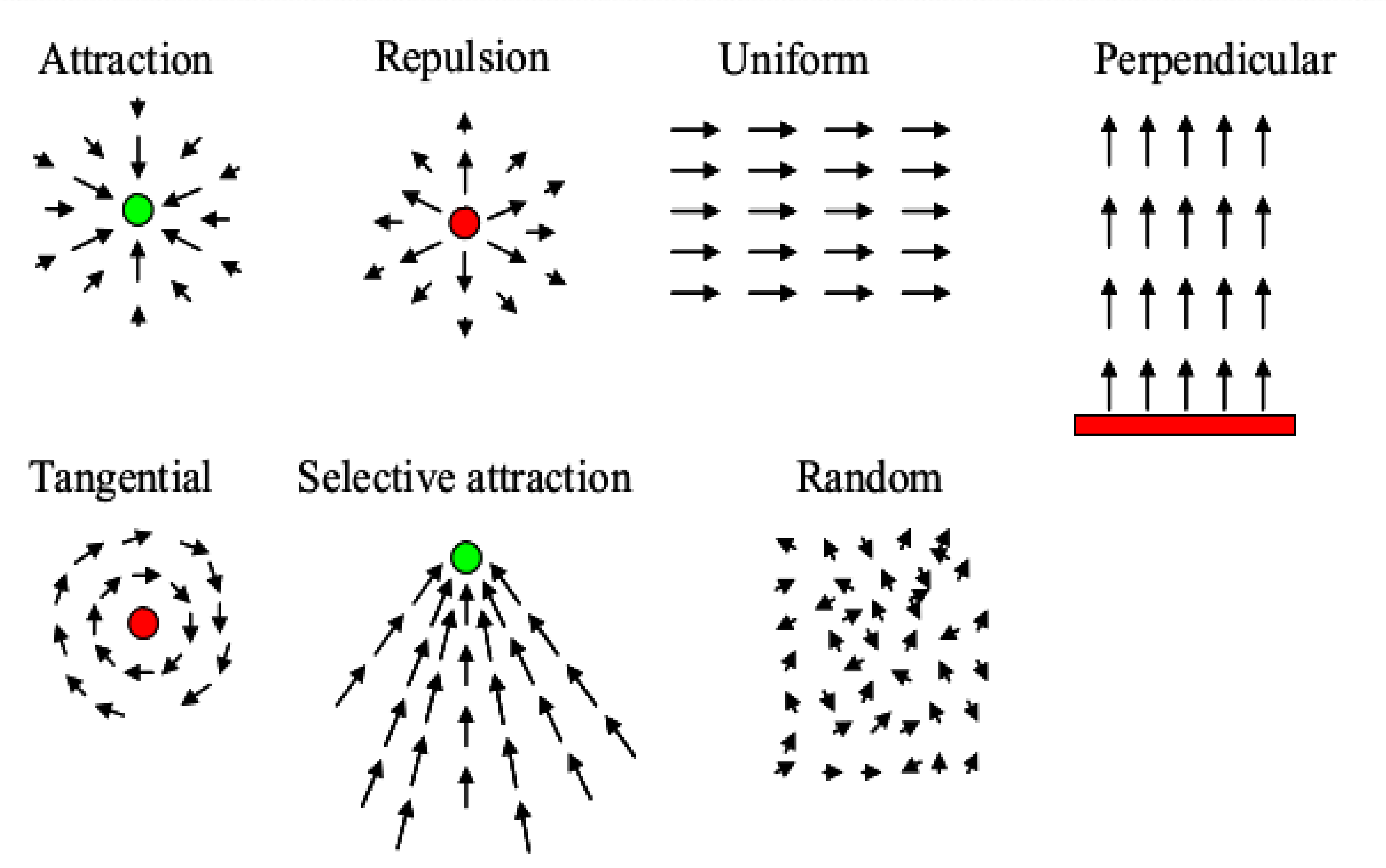
Potential Fields: example





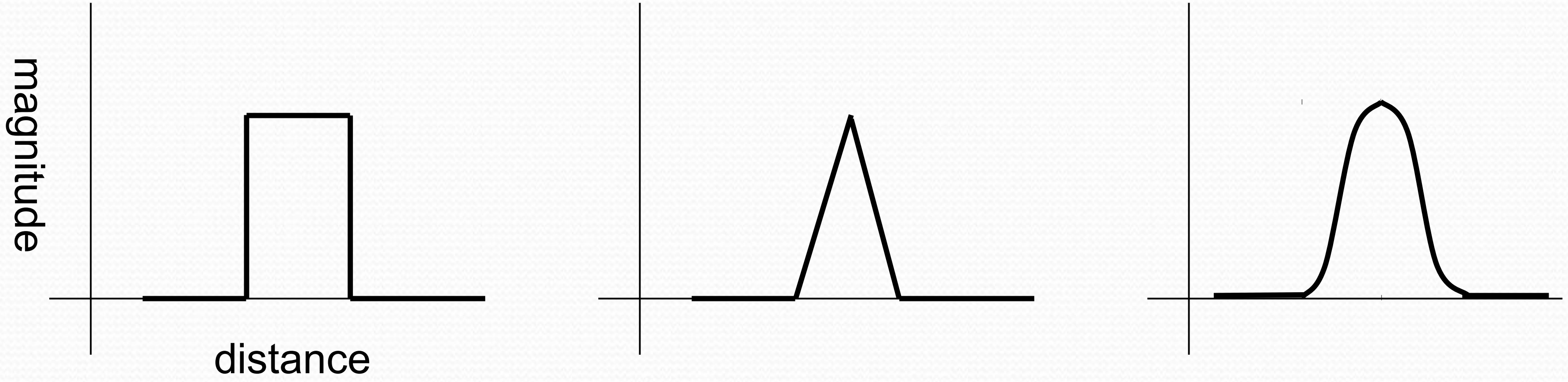
Example of one attraction goal (green, located at [1,-2])
and three obstacles

Primitive Potential Fields



Magnitude Profiles

- Change in intensity in different parts of the field



Constant

Linear Dropoff

Exponential Dropoff

➔ *Field closest to an attractor/repellor will be stronger*

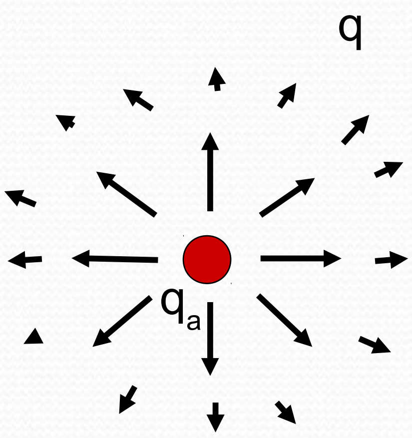
Programming a Single Potential Field

$U_{ATT} = \xi\, d^{2} / 2$

$\nabla U_{ATT} = \xi\,(q - q_a).$

$$U_{rep} = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{d} - \frac{1}{d_o}\right)^2 & d \le d_o \\ 0 & d > d_o \end{cases}$$

$$\nabla U_{rep} = \begin{cases} \eta\left(\frac{1}{d} - \frac{1}{d_o}\right)\frac{(\mathbf{q} - \mathbf{q}_o)}{d^3} & d \le d_o \\ 0 & d > d_o \end{cases}.$$

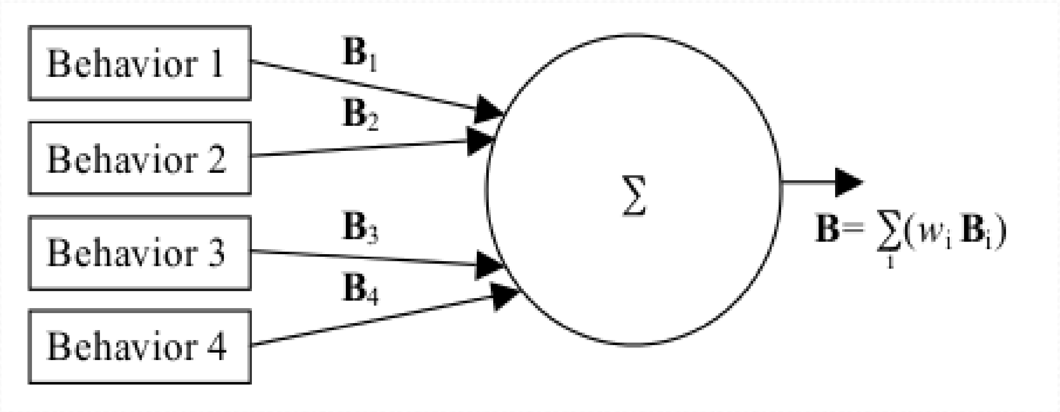


Obstacle centered at q_a

Robot currently located at q

Production of behaviour

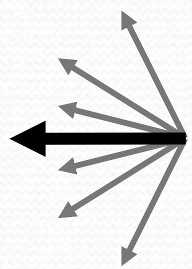
- Only portion of field affecting robot is computed (not the entire field)
- Robot uses functions defining potential fields at its position to calculate component vector
- Weighted sums can be used to fuse multiple primitive behaviour

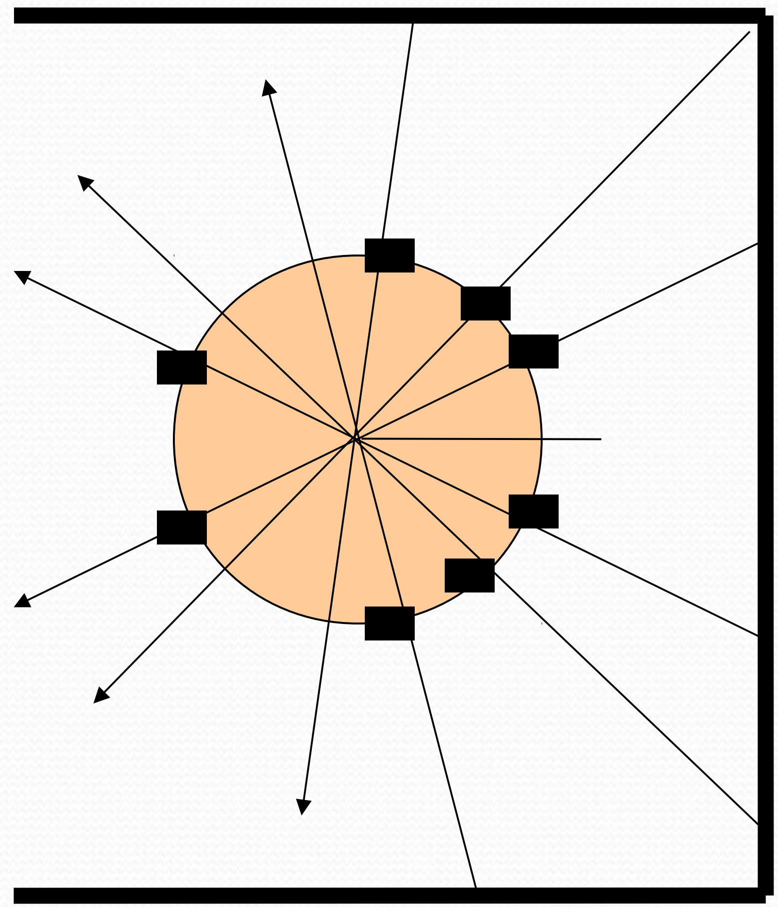


Consider “Box Canyon”

- Generate repulsive field for each sensor
- Sum the sensors to get the resultant direction

Summed output vector





- Result: Robot can back out of box canyon without needing “model” of walls

Issues with Combining Potential Fields

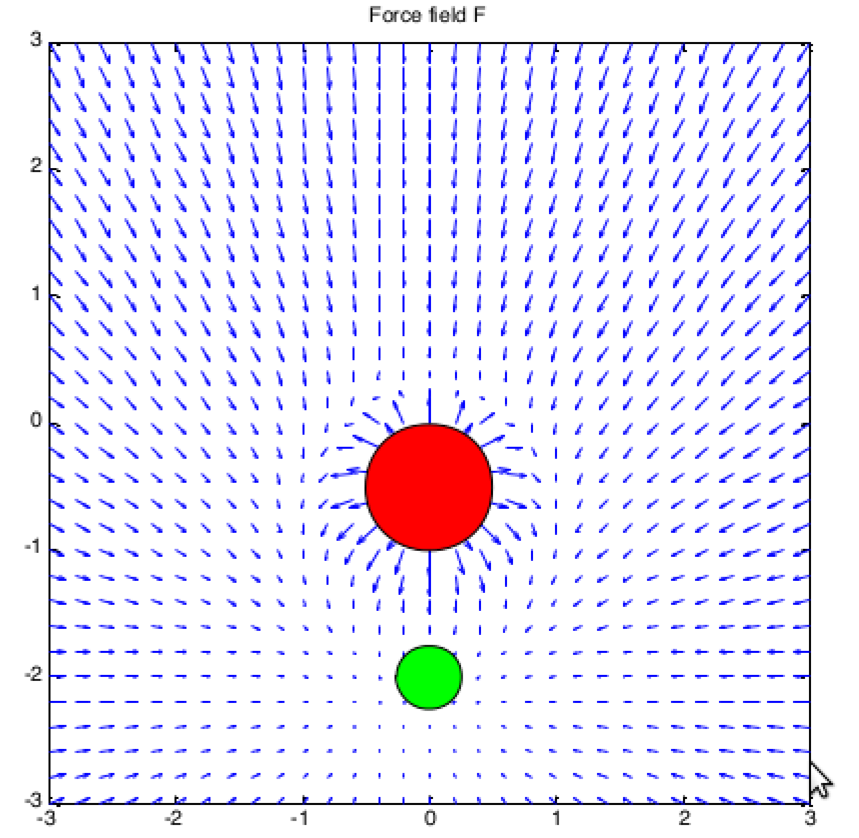
- Impact of update rates:
  - Lower update rates can lead to “jagged” paths

- Robot treated as point:
  - ➔ Expect robot to change velocity and direction instantaneously (can’t happen)

- Local minima:
  - Vectors may sum to 0.

The Problem of Local Minima

- If robot reaches local minima, it will just sit still



Solutions for Dealing with Local Minima

- Inject noise, randomness:
  - “Bumps” robot out of minima

- Include “avoid-past” behavior:
  - Remembers where robot has been and attracts the robot to other places

- Use “Navigation Templates” (NaTs):
  - The “avoid” behavior receives as input the vector summed from other behaviors
  - Gives “avoid” behavior a preferred direction

- Insert tangential fields around obstacles

Readings

Koren, Y.; Borenstein, J.; Dept. of Mech. Eng. & Appl. Mech., Michigan Univ., Ann Arbor, MI, **Potential field methods and their inherent limitations for mobile robot navigation,** in Proceedings IEEE International Conference on Robotics and Automation, 1991