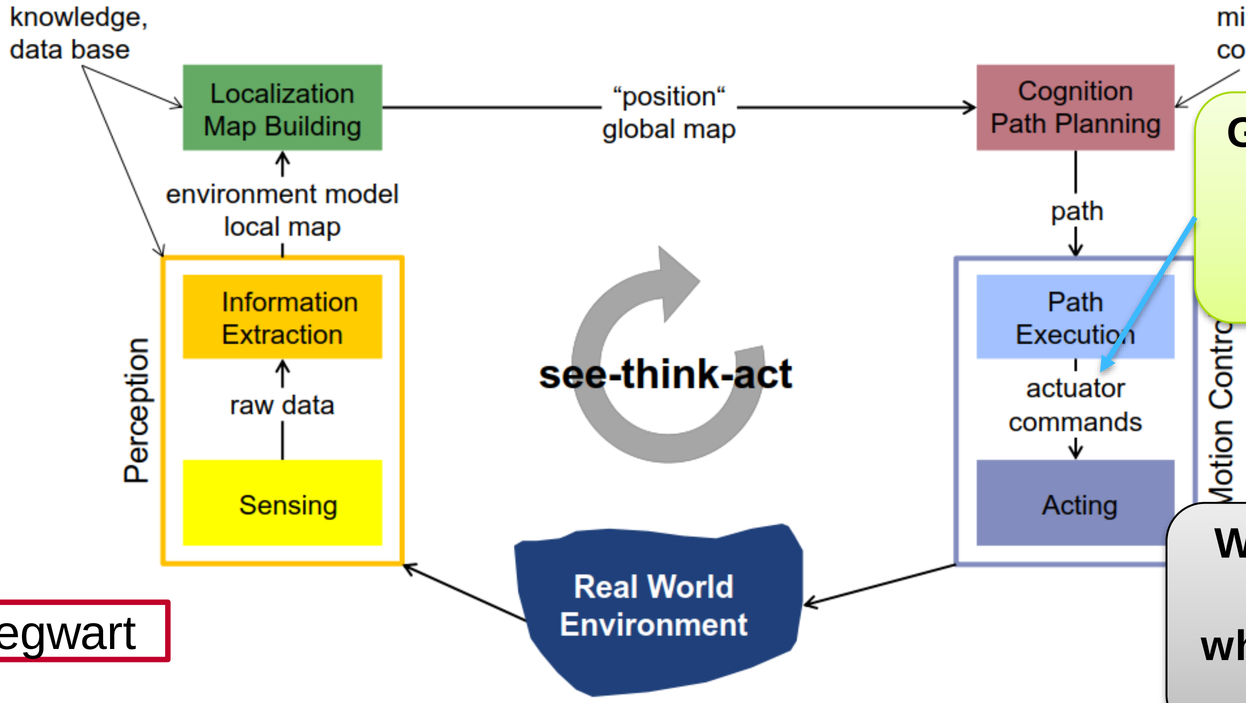
TAMPERE UNIVERSITY OF TECHNOLOGY

Reactive Obstacle avoidance

Reza Ghabcheloo

IHA 4506 Advanced Robotics

Control architecture
Autonomous robot



Generating feasible commands are usually done is a separate module

What if we cannot replan in time when encountering an obstacle!

R. Siegwart

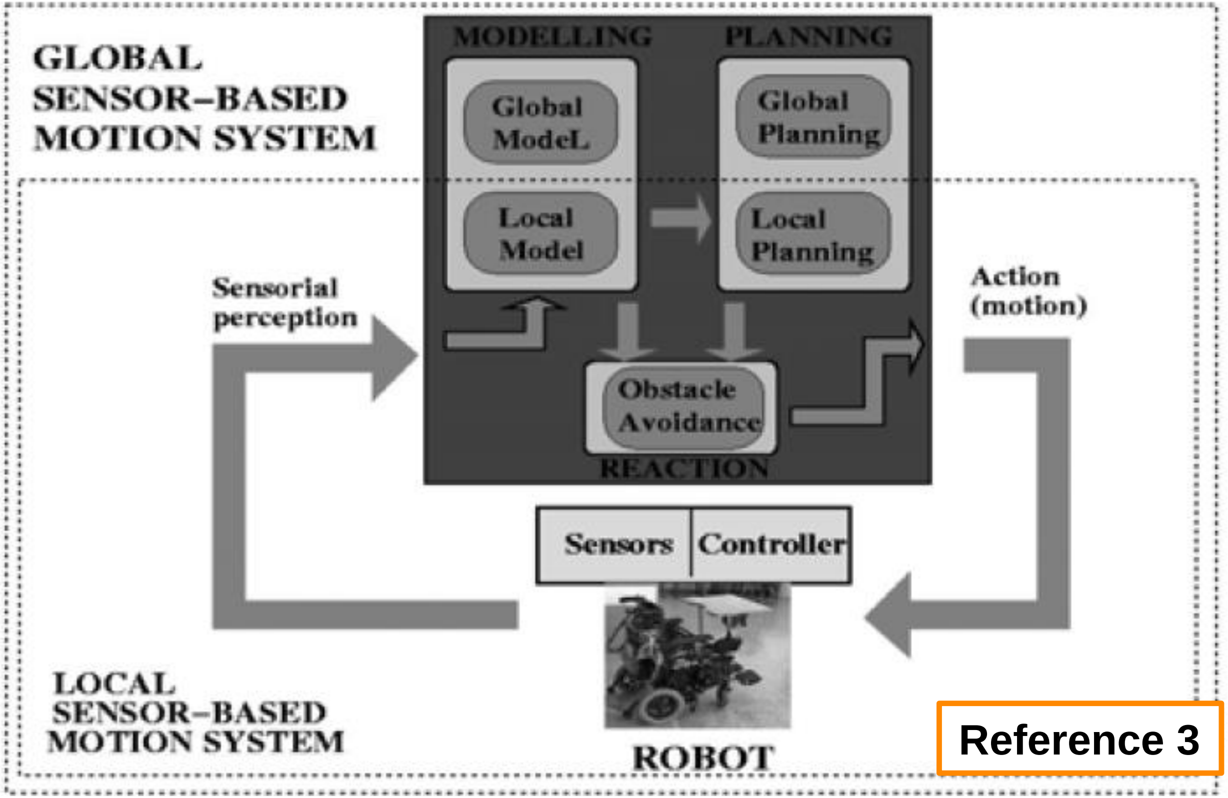
Topics

- Why we need reactive obstacle avoidance: Integration of deliberation and reactivity
- Potential fields
  - One of most popular methods for reactive obstacle avoidance

Local sensor based navigation: acts as safety module

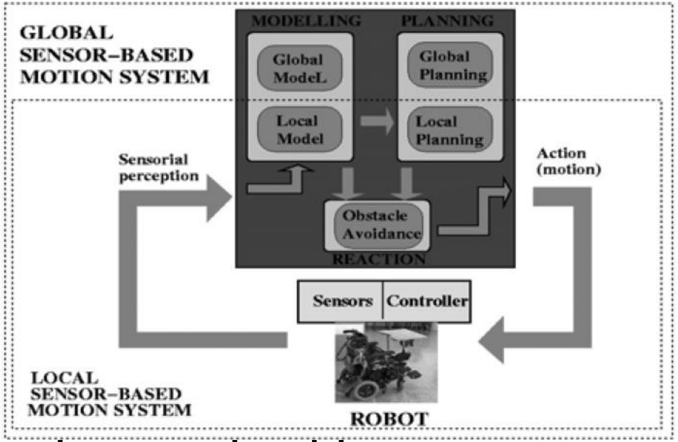
- Acts directly on sensor data or creates simple local models
- Real-time execution, robot shape and constraints are important for guaranteed safety

Global vs Local sensor based navigation



Reference 3

Architecture



- Build environment model
  - Although there are control architectures that work with no modeling, modeling increase spatial domain of the navigation problem, also used for local navigation
- Global planning or deliberation
  - to extract connectivity of the free space benefiting from the global model; thus avoid trap situations/local min
- Local or reactive control: collision free motion generation; allow global planner ignore “details”

Sensor based obstacle avoidance methods

- Virtual or Artificial Potential fields
- Tangent bug
- Vector field histogram
- Velocity obstacle
- Elastic band

Attractive and Repulsive potential

- $U_{att}(q)$ attractive
- $U_{rep}(q)$ repulsive
- $U(q) = U_{att}(q) + U_{rep}(q)$ → $F = -\frac{\partial U}{\partial q}$

One dimension robot:

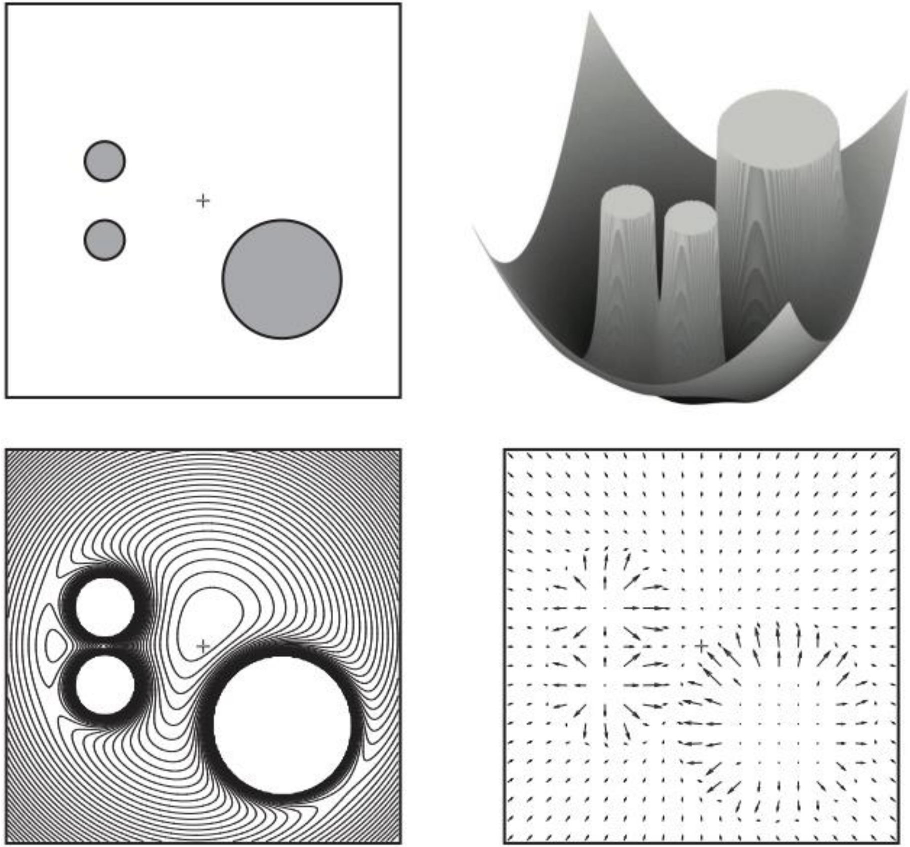
Gravity potential: $U(h) = mgh$

Gravity force: $F(h) = -\frac{\partial U}{\partial h} = -mg$

\+ the goal configuration

O obstacles

Simple idea, it is cost like, use gradient method: so it is used in many applications



Attractive potential

- $U_{att}(q) = \frac{1}{2}\left(q - q_{goal}\right)^T K\left(q - q_{goal}\right)$

with $K > 0$; $F_{att}(q) = -K\left(q - q_{goal}\right)$

or any distance function $d\left(q, q_{goal}\right)^2$

Repulsive potential

- Usually much more difficult to calculate
- Depends on
  - the shape of the obstacles
  - the shape of the robot
  - robot constraints

Repulsive potential

- $U_{rep}(q) = \begin{cases} \frac{1}{2}k\left(\frac{1}{D(q)} - \frac{1}{Q^*}\right)^2, & D(q) \leq Q^* \\ 0, & D(q) > Q^* \end{cases}$
- Q^*: ignore too far obstacles
- $D(q)$:
  - closest obstacles, or
  - closet point of each obstacle, or
  - distance to a “control point”

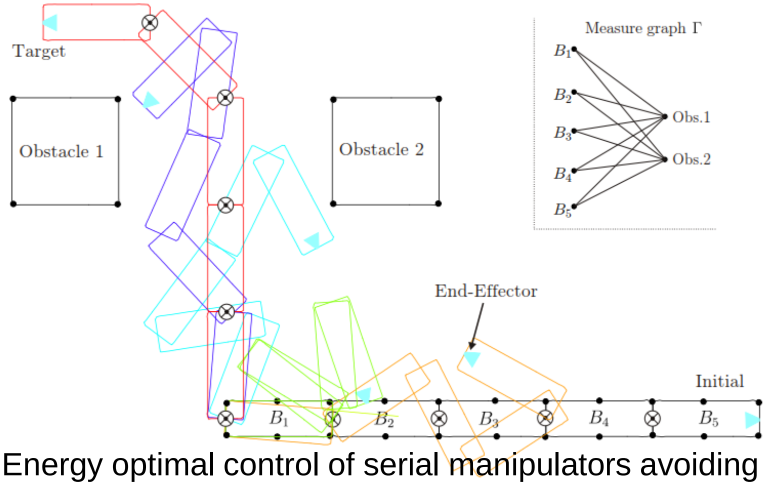
Repulsive potential

- $U_{rep}(q) = \begin{cases} \frac{1}{2}k\left(\frac{1}{D(q)} - \frac{1}{Q^*}\right)^2, & D(q) \leq Q^* \\ 0, & D(q) > Q^* \end{cases}$

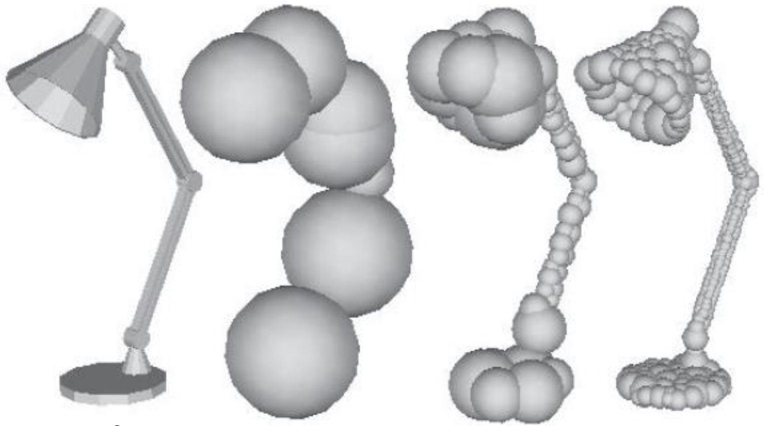
- $F(q) = \frac{\partial U_{rep}}{\partial q} = \begin{cases} k\left(\frac{1}{D(q)} - \frac{1}{Q^*}\right)\frac{1}{D(q)^2}\frac{\partial D}{\partial q}, & D(q) \leq Q^* \\ 0, & D(q) > Q^* \end{cases}$

Total potential

- $U(q) = U_{att}(q) + \sum_i U_{rep}(q)$



Energy optimal control of serial manipulators avoiding Collisions, Andreas Muller



Ref 1

Controlling robot different options (repulsive)

- Treat it as force: force + damping

$$u = F(q) + B\dot{q}, \qquad B > 0$$

- Treat it as velocity (no damping is needed)

$$\dot{q} = F(q)$$

Some points

- The repulsive force can be generated based on distance or time to collide
- Velocity is used to damp or reshape the field: for example to remove repulsive forces when moving away from obstacles *(Forklift example)*
- Advantage of force is that it takes the robot dynamic into account

Applications 1

Inverse kinematics w/ constraints &redundancy

$$e_P(t) = P\big(q(t)\big) - \rho\big(s(t)\big)$$

$$\ddot{q} = {K_q}^{-1}\left(-k_1\dot{q} - \beta_1 \frac{\partial P^T}{\partial q}\, e_P - \beta_3 \frac{\partial U}{\partial q}\right)$$

$$\frac{dU}{dq_i} = w^2 e + 2w\dot{q}_i, \text{ with } w = \frac{1}{l_{safe}}$$

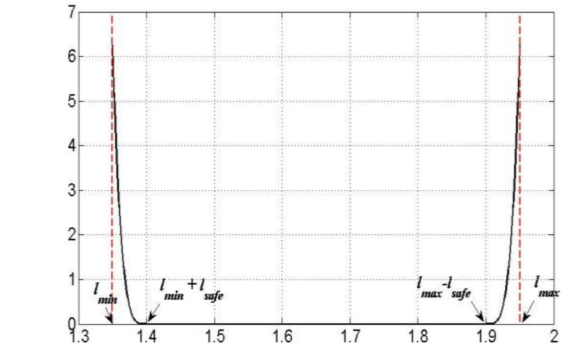
$$U = w\,(d + l\cos\theta_1 - 0.5)^2 \text{ with } w = 10$$

←Position error

← Inverse kinematic ODE

← Joint limits

← Redundancy resolution



Coordinated Control of Hydraulic Mobile Manipulators

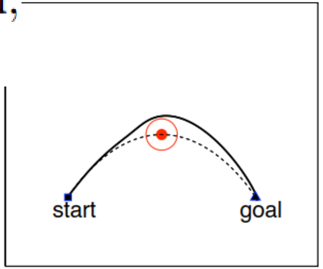
Ghabcheloo, R. & Huhtala, K.

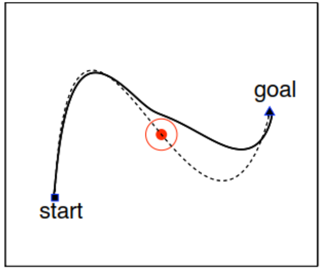
Applications 2
Dynamic movement primitive

$$\tau \dot{v} = K(g - x) - Dv - K(g - x_0)\theta + Kf(\theta) - \frac{\partial U}{\partial x}$$

$$U(\mathbf{x}_r, \mathbf{v}_r) = \begin{cases} \lambda(-\cos\gamma)^\beta \frac{||\mathbf{v}_r||}{||\mathbf{x}_r||} & : \quad \frac{\pi}{2} \le \gamma \le \frac{3\pi}{2} \\ 0 & : \quad \text{else} \end{cases} \quad (5)$$

where λ is a constant for the strength of the entire field, β another constant, and γ the angle between $\mathbf{x}_r$ and $\mathbf{v}_r$.

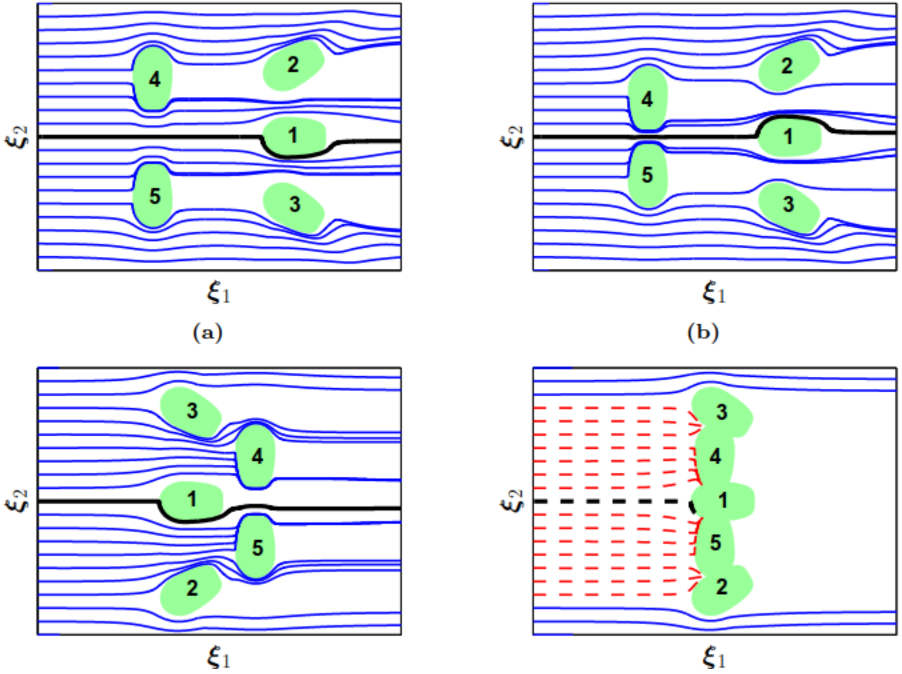


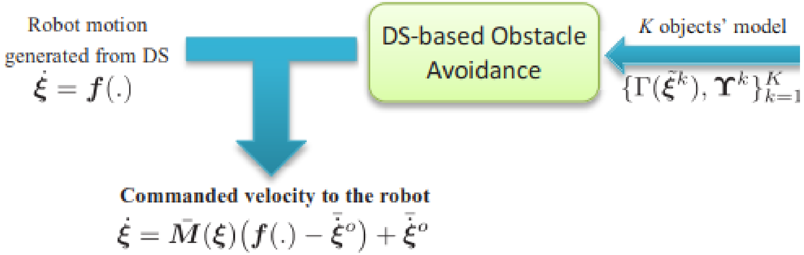


Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields, Park, Hoffmann, Schaal Humanoid Robots, 2008

Application 3
Locally Modulating Dynamic Systems

- modulating/Rotating the original vector field



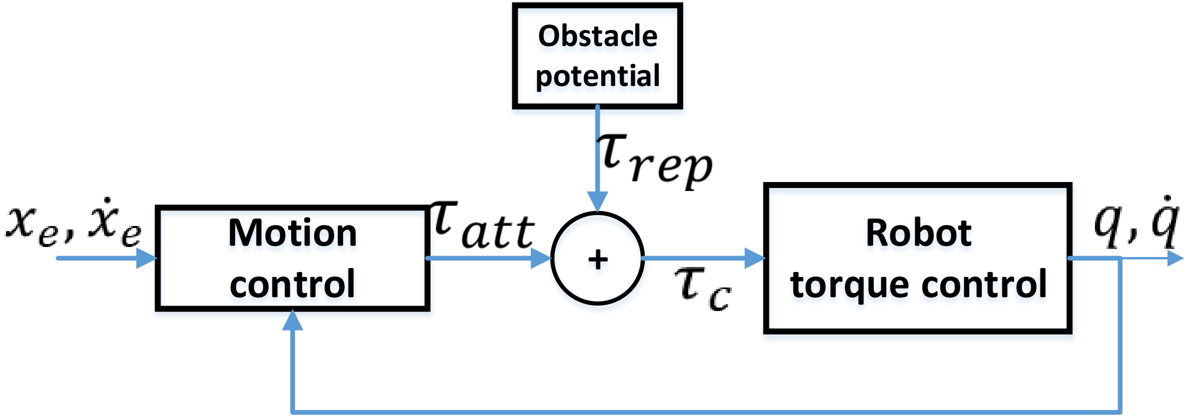


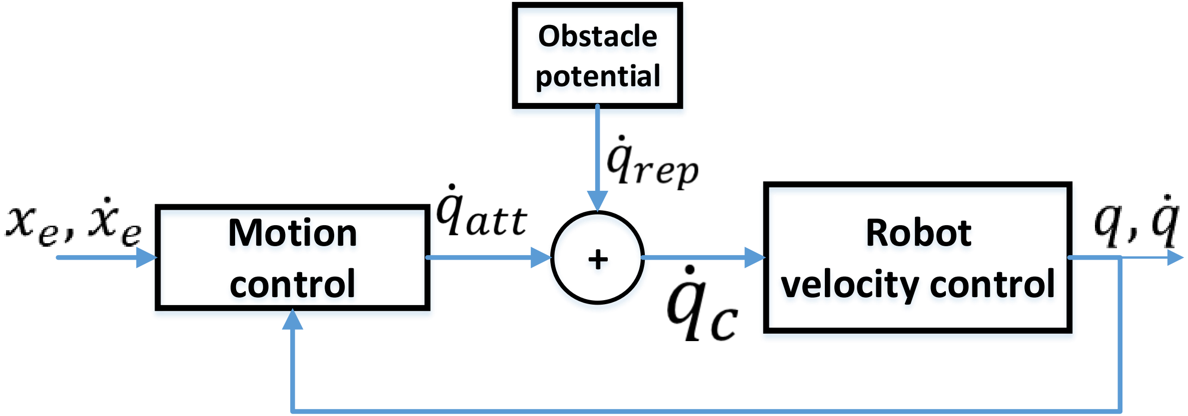
Implementation

$$u = u_{att} + u_{rep}$$

Don’t forget the damping

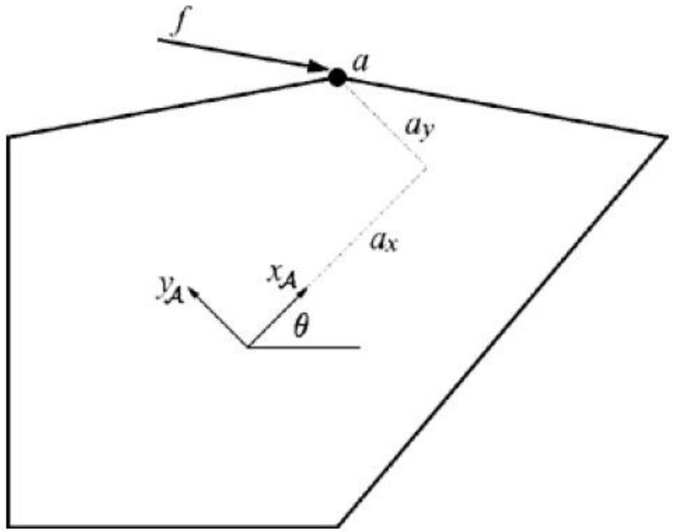
- Option 1: torque/acceleration control
  - u_{rep} is repulsive torque/acceleration
  - u_{att} is the torque/acceleration generated by the motion controller
- Option 2: velocity (or rate) control
  - u_{rep} is repulsive velocity vector
  - u_{att} is the velocity generated by the motion controller


Obstacle potential
τ_{rep}
$x_e, \dot{x}_e$
Motion control
τ_{att}
+
τ_c
Robot torque control
$q, \dot{q}$


Obstacle potential
$\dot{q}_{rep}$
$x_e, \dot{x}_e$
Motion control
$\dot{q}_{att}$
+
$\dot{q}_c$
Robot velocity control
$q, \dot{q}$

Potential field in non-Euclidian space

- $\begin{bmatrix} u_x \\ u_y \\ u_\theta \end{bmatrix} = J^T(q) \begin{bmatrix} f_x \\ f_y \end{bmatrix}$



Wheeled robot, differential constraint

- $A(q)\dot{q} = 0$ non-holonomic constraints
- Controls must be projected tangent to the constraints

$$\dot{q} = F_{proj}(q)$$

$$F_{proj}(q) = (I - A^T(AA^T)^{-1}A)F(q)$$

This will guarantee that the rolling constraints are satisfied

Further reading

Concept of multiple control points and flooring point

- Potential field for rigid body robots (Ref 2 Sec 4.7.2; and Ref 1 Sec 10.6.3.)
- Potential field and articulated robots (Ref 2 Sec 4.7.3)

Main references

1. **MODERN ROBOTICS: MECHANICS, PLANNING, AND CONTROL** by Kevin M. Lynch and Frank C. Park, 2017 (Sec 10.6)
2. **Principles of Robot Motion-Theory, Algorithms, and Implementation**, by Howie Choset, Kevin Lynch, Seth Hutchinson, George Kantor, Wolfram Burgard, Lydia Kavraki, Sebastian Thrun, 2005 (Chapter 4; minus Sec 4.6)
3. **Handbook of robotics, Second edition, Chapter 47 (**Sections 47.7 – 47.9.1, and 47.11 intro and 47.11.2**), Motion Planning and Obstacle Avoidance**, By Javier Minguez, Florant Lamiraux, Jean-Paul Laumond
4. Further reading: A Depth Space Approach for Evaluating Distance to Objects with Application to Human-Robot Collision Avoidance; F. Flacco, T. Kroeger, A. De Luca, O. Khatib
5. Nirmal Giftsun (PhD Thesis, 2017) Handling Uncertainty and Variability in Robot Control; Chapter 2- a very good read about state of art

Exercise

1. Develop a control architecture to implement a reactive obstacle avoidance can be implemented on Franka arm control system
2. Implement classic artificial potential field for point end-effector, point obstacle
3. Implement a code to keep the end-effector inside a given bounding box
4. Avoid hitting the table while moving from A to B
5. Bonus1: implement 3 for whole robot body
6. Bonus2: implement 1 for a moving obstacle