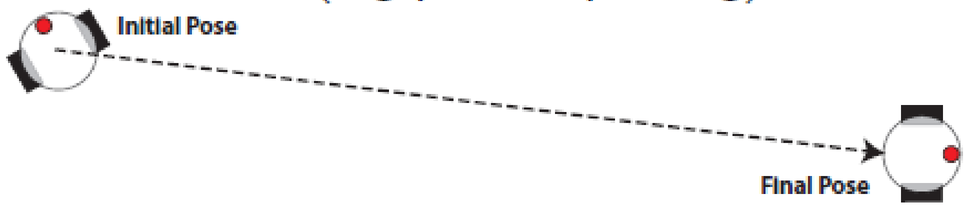
WMR control

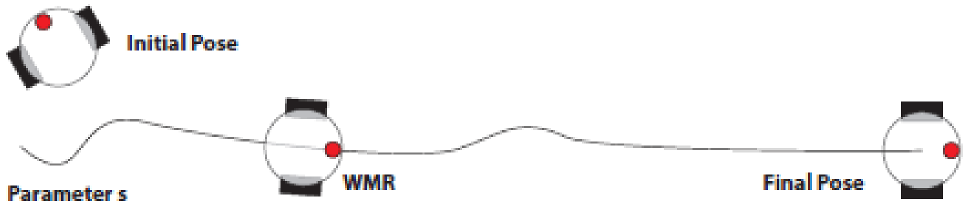
Basic motion planning

Elementary motion tasks

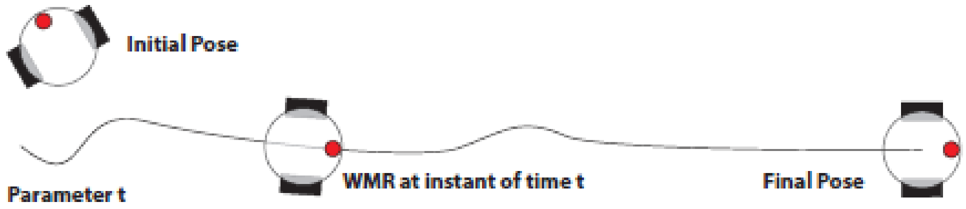
- Point-to-Point Transfer (e.g parallel parking)



- Trajectory Following (no time constraints)

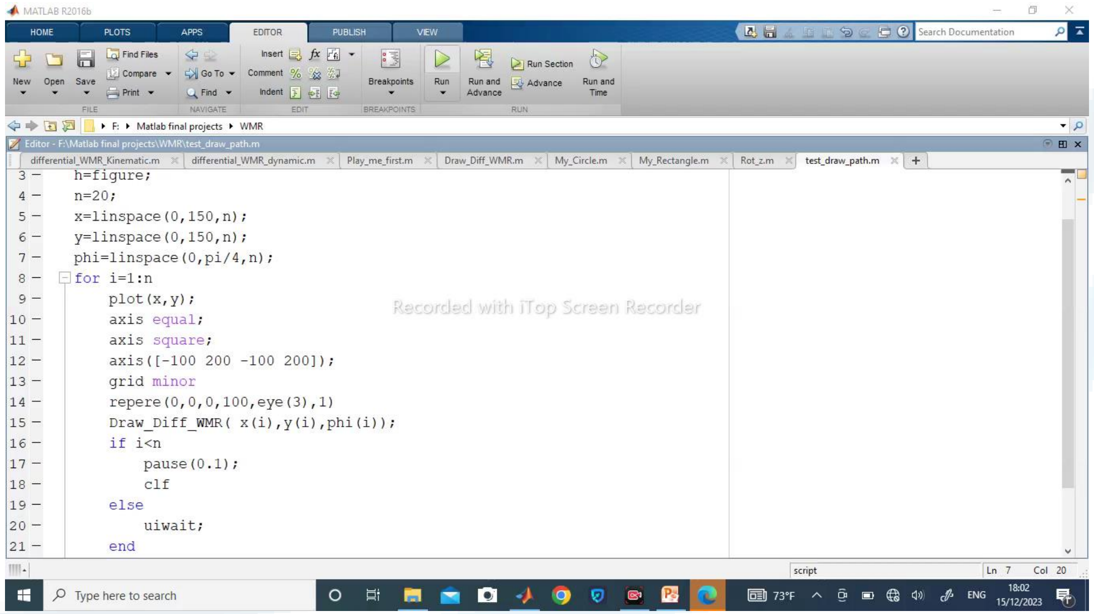


- Trajectory Tracking (with time constraints)

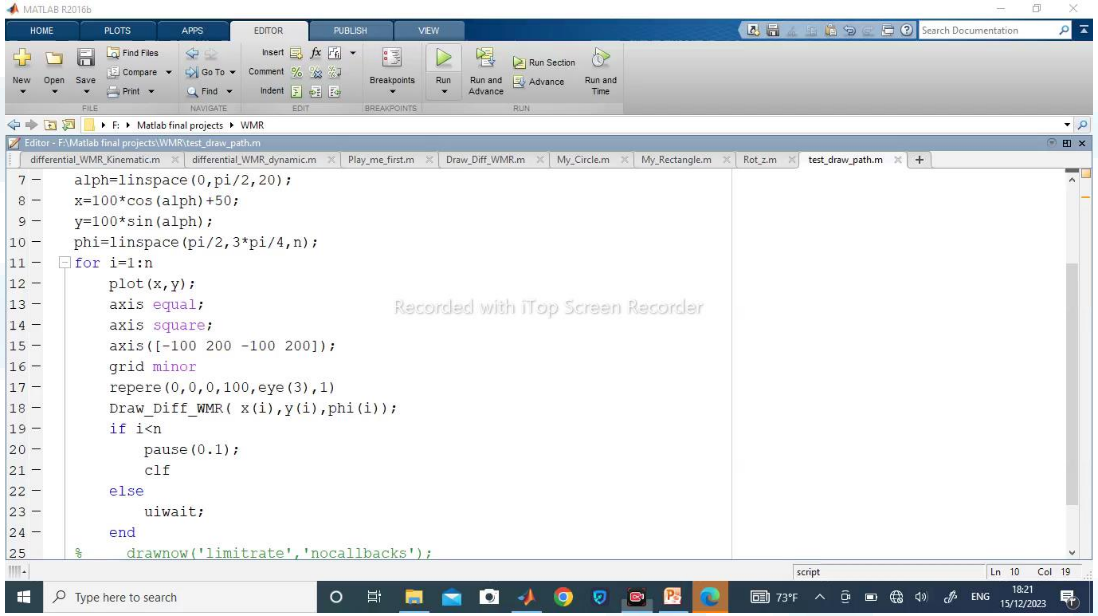


Elementary motion tasks

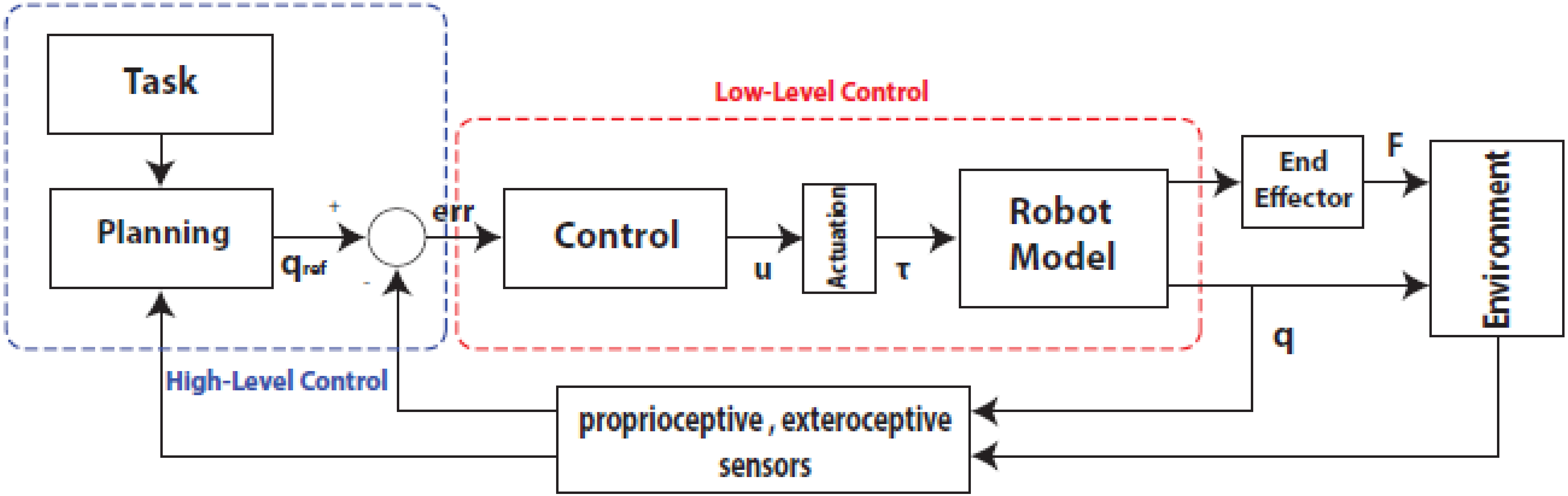
Point to point



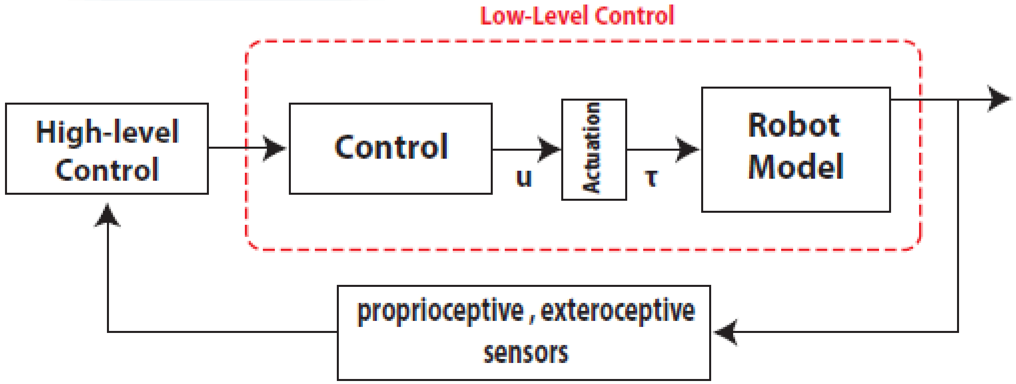
Trajectory following



Control Scheme



Low level control



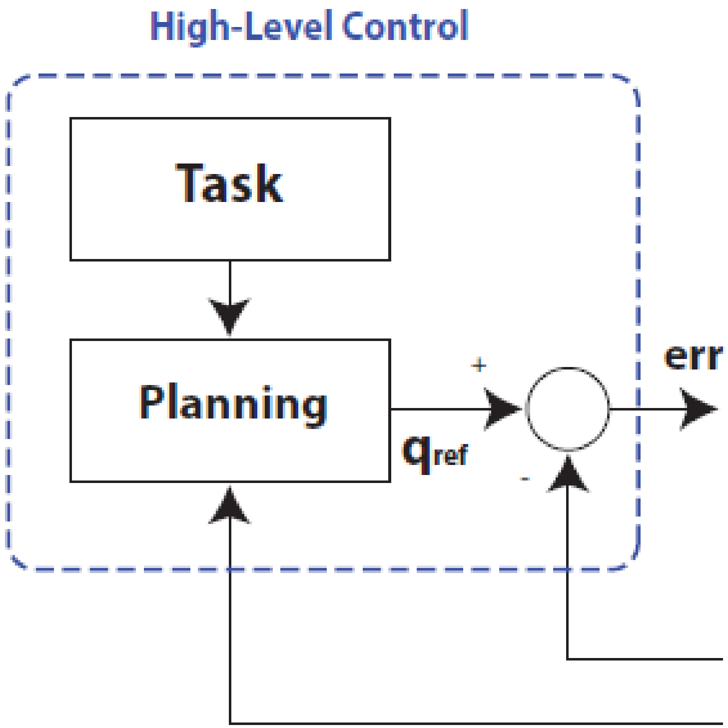
Core Concepts

- High-gain PI controllers control the robot's motors so that the robot moves according to the desired speed profile
- The low-level control deals only with the robot actuators control according to the high-level control instructions
- If the gains are high enough, the low-level control makes the robot a purely kinematic system

High-level control

Core Concepts

- It processes and computes the signals to send to the low-level controller using data coming from sensors
- From its point of view, the robot behaves as a purely kinematic system
- For mobile robots, speed control signals are used



Control of a WMR

Low-Level Control

- Internal loop on the motors side for controlling the robot actuation
- It is a simple PI for electric drives (linear systems)
- It is not affected by the non-holonomic constraints introduced by the wheels
- Known and solved issues

High-Level Control

- It defines the *motion* and the *behavior* of the robot based on the task to be performed
- It must consider the kinematic model
- Subject to the constraints of the wheels
- It has to control a nonlinear and complex system

Planning

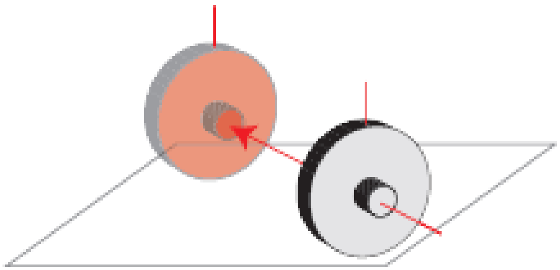
Planning for a WMR

- Problem: determining a trajectory in the configuration space to take the robot from a certain initial configuration to a final configuration, both feasible
- The initial and final configurations (boundary conditions) and *any* point of the trajectory must be compatible with the kinematic constraints of the robot

Note

Definition

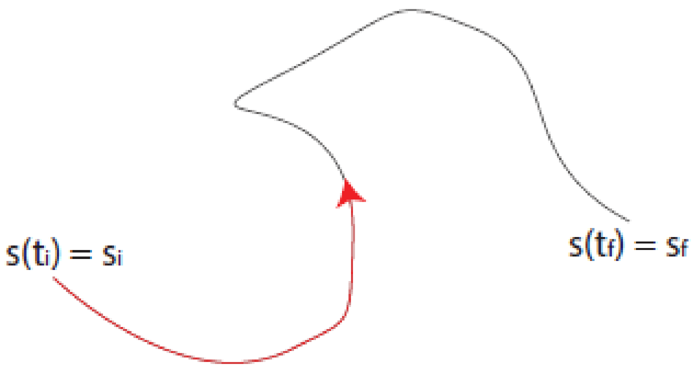
A trajectory is not feasible if it requires the robot to perform motion incompatible with its kinematic constraints.



Example: unicycle can not have lateral translational trajectories.

Space-time separation of the trajectory

- We want to plan a trajectory $q(t)$ for t belongs to $[t_i, t_f]$ that take the robot from an initial configuration $q(t_i) = q_i$ to a final configuration $q(t_f) = q_f$
- We assume no obstacles



The trajectory $q(t)$ can be decomposed in:

- a path $q(s)$, with $\dfrac{dq(s)}{ds} \neq 0, \forall s$
- a motion law $s = s(t)$, with $s_i \leq s \leq s_f$,
  with $\begin{cases} s(t_i) & = & s_i \\ s(t_f) & = & s_f \end{cases}$
  s monotonic, i.e. $\dot{s}(t) \geq 0$
- Typical choice for s is the *curvilinear abscissa* along the path: $\begin{cases} s_i & = & 0 \\ s_f & = & L \end{cases}$

جَـامعة
المَـنارة
MANARA UNIVERSITY

Planning

Space-time separation of the trajectory $\dot{q} = \frac{dq}{dt} = \frac{dq}{ds}\dot{s} = q'\dot{s}$

- q' has the direction of the tangent to the path in the configurations space oriented for growing s
- $\dot{s}$ is a scalar which modulates the intensity

Form the Pfaffian form of the nonholonomic constraints we get the **feasability condition** of the geometric path:

$$\begin{cases} A(q)\dot{q} & = & A(q)q'\dot{s} = 0 \\ \dot{s} & > & 0, \forall t \in [t_i, t_f] \end{cases}$$

$$\Downarrow$$

$$A(q)q'$$

Planning

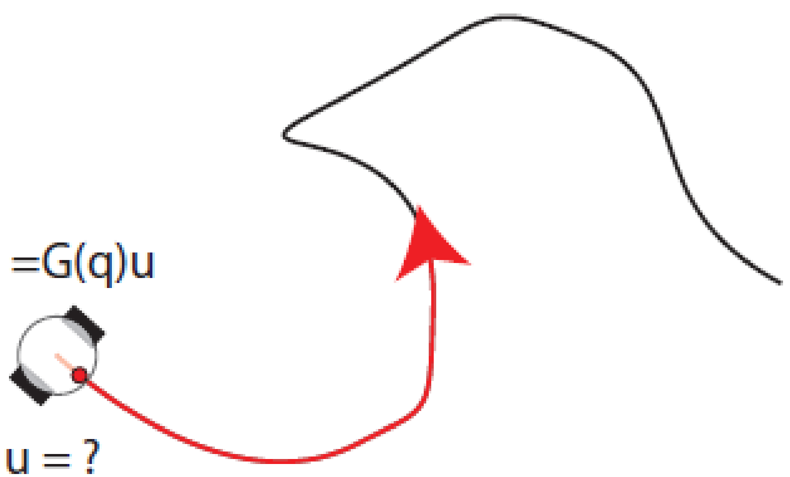
A feasable path is given by: $q' = G(q)\tilde{u}$

- *Geometric inputs* $\tilde{u}$: determine the geometry path
- Chosen $\tilde{u}$ feasable, we define the *motion law* $s(t)$ to define how fast the robot run across the path

Problem

How to combine the geometric path with known inputs $\tilde{u}$ and the motion law in order to obtain the control inputs for the robot?

Planning



$$q' = G(q)\tilde{u}(s)$$

$$\Downarrow$$

$$\frac{dq}{ds}\dot{s} = G(q)\tilde{u}(s)\dot{s}$$

$$\Downarrow$$

$$\begin{cases} \dot{q} & = & G(q)\tilde{u}(s)\dot{s} \\ \dot{q} & = & G(q)u(t) \end{cases}$$

$$\Downarrow$$

$$\tilde{u}(s)\dot{s} = u(t)$$

Example: unicycle

For the unicycle, the wheel's nonholonomic constraints imply the following feasibility condition for the geometric path:

$$[\sin\theta, -\cos\theta, 0]\, q' = x' \sin\theta - y' \cos\theta = 0$$

- The condition highlights the fact that the Cartesian speed must be oriented along the direction of motion (no lateral slip)
- The feasible paths for the unicycle are given by:

$$\begin{cases} x' &= \tilde{v}\cos\theta \\ y' &= \tilde{v}\sin\theta \\ \theta' &= \tilde{\omega} \end{cases}$$

- The kinematic inputs are obtained from the geometric ones:

$$\begin{cases} v(t) &= \tilde{v}\dot{s} \\ \omega(t) &= \tilde{\omega}\dot{s} \end{cases}$$

Thanks

Think about MATLAB SIMULINK to validate the resultant model