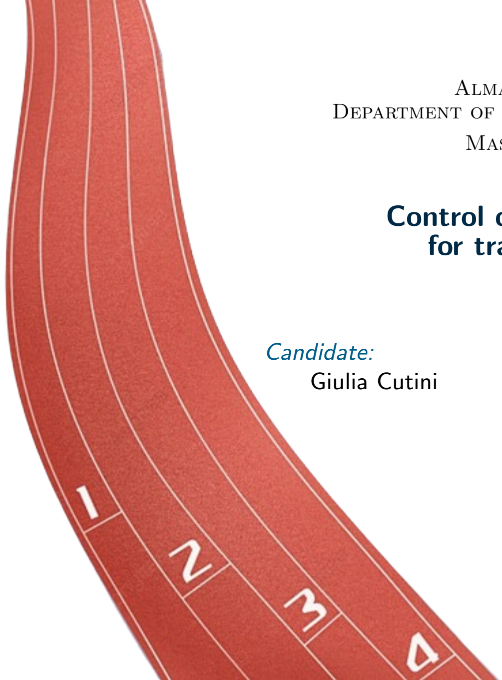


Alma Mater Studiorum Università di Bologna
Department of Electrical, Electronic and Information Engineering

Master's Degree in Automation Engineering

Control of an autonomous aerodynamic airshield for training Olympics 100m sprint athletes

Candidate:
Giulia Cutini

Advisor:
Prof. Giuseppe Notarstefano

Co-Advisors:
Prof. Melanie Zeilinger
Dr. Andrea Carron
Ing. Lorenzo Sforni

Bologna, 18 March 2024

Introduction

Motivations

In athletics, the **overspeed training** makes possible to enhance the competition performances.
It can be achieved by isolating the runner from the air resistance using an **airshield**.

Introduction

Motivations

In athletics, the **overspeed training** makes possible to enhance the competition performances.
It can be achieved by isolating the runner from the air resistance using an **airshield**.

The usage of an **autonomous go-kart** improves:

- the safety of the maneuver
- the reliability and the reusability

Motivations

In athletics, the **overspeed training** makes possible to enhance the competition performances.
It can be achieved by isolating the runner from the air resistance using an **airshield**.

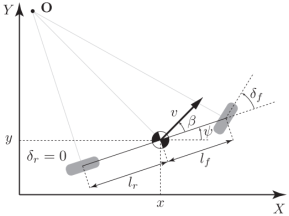
The usage of an **autonomous go-kart** improves:

- the safety of the maneuver
- the reliability and the reusability

Contributions

- Approximated model for the go-kart with the airshield attached
- Design a controller for to regulate the shield with respect to the runner
- Implement and test the controller in simulation and hardware-in-the-loop

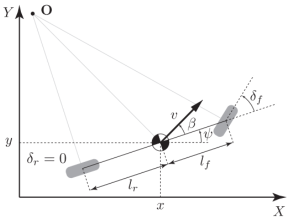
Modeling of the go-kart with the airshield attached



Nonlinear kinematic byclicle model

$$
\begin{cases}
\dot{x}(t) = v(t)\cos(\psi(t) + \beta(t)) \\
\dot{y}(t) = v(t)\sin(\psi(t) + \beta(t)) \\
\dot{\psi}(t) = \dfrac{v(t)}{l_r}\sin(\beta(t)) \\
\dot{v}(t) = \dfrac{F_x(t)}{m} = \dfrac{1}{m}(C_{m1}a(t) - C_f v(t) - C_d v^2(t) - C_{roll})
\end{cases}
$$

Modeling of the go-kart with the airshield attached

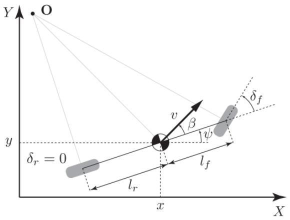


- High energy-demanding training
- Only 50-80 meters Executed

Nonlinear kinematic byclicle model

$$
\begin{cases}
\dot{x}(t) = v(t)\cos(\psi(t) + \beta(t)) \\
\dot{y}(t) = v(t)\sin(\psi(t) + \beta(t)) \\
\dot{\psi}(t) = \dfrac{v(t)}{l_r}\sin(\beta(t)) \\
\dot{v}(t) = \dfrac{F_x(t)}{m} = \dfrac{1}{m}(C_{m1}a(t) - C_f v(t) - C_d v^2(t) - C_{roll})
\end{cases}
$$

Modeling of the go-kart with the airshield attached



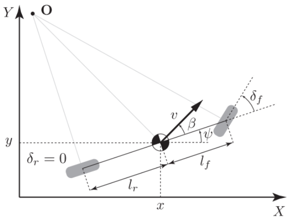
- High energy-demanding training
- Only 50-80 meters Executed

Nonlinear kinematic byclicle model

$$
\begin{cases}
\dot{x}(t) = v(t)\cos(\psi(t) + \beta(t)) \\
\dot{y}(t) = v(t)\sin(\psi(t) + \beta(t)) \\
\dot{\psi}(t) = \dfrac{v(t)}{l_r}\sin(\beta(t)) \\
\dot{v}(t) = \dfrac{F_x(t)}{m} = \dfrac{1}{m}(C_{m1}a(t) - C_f v(t) - C_d v^2(t) - C_{roll})
\end{cases}
$$

$\Rightarrow$

Modeling of the go-kart with the airshield attached



Nonlinear kinematic byclicle model

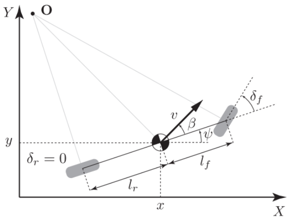
$$\begin{cases} \dot{x}(t) = v(t)\cos(\psi(t)+\beta(t)) \\ \dot{y}(t) = v(t)\sin(\psi(t)+\beta(t)) \\ \dot{\psi}(t) = \dfrac{v(t)}{l_r}\sin(\beta(t)) \\ \dot{v}(t) = \dfrac{F_x(t)}{m} = \dfrac{1}{m}(C_{m1}a(t) - C_f v(t) - C_d v^2(t) - C_{roll}) \end{cases}$$

- High energy-demanding training
- Only 50-80 meters Executed

$\Rightarrow$

- No steeering degree of freedom
- Model can be simplified

Modeling of the go-kart with the airshield attached



Nonlinear kinematic byclicle model

$$\begin{cases} \dot{x}(t) = v(t)\cos(\psi(t) + \beta(t)) \\ \dot{y}(t) = v(t)\sin(\psi(t) + \beta(t)) \\ \dot{\psi}(t) = \dfrac{v(t)}{l_r}\sin(\beta(t)) \\ \dot{v}(t) = \dfrac{F_x(t)}{m} = \dfrac{1}{m}(C_{m1}a(t) - C_f v(t) - C_d v^2(t) - C_{roll}) \end{cases}$$

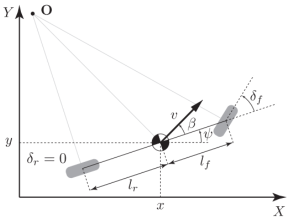
- High energy-demanding training
- Only 50-80 meters Executed

$\Rightarrow$

- No steeering degree of freedom
- Model can be simplified

$$\begin{cases} \dot{p}(t) = v(t) \\ \dot{v}(t) = \dfrac{F(t)}{m} = \dfrac{1}{m}(C_{m1}a(t) - C_f v(t) - C_d v^2(t) - C_{roll}) \end{cases}$$

Modeling of the go-kart with the airshield attached



Nonlinear kinematic byclicle model

$$\begin{cases} \dot{x}(t) = v(t)\cos(\psi(t) + \beta(t)) \\ \dot{y}(t) = v(t)\sin(\psi(t) + \beta(t)) \\ \dot{\psi}(t) = \dfrac{v(t)}{l_r}\sin(\beta(t)) \\ \dot{v}(t) = \dfrac{F_x(t)}{m} = \dfrac{1}{m}(C_{m1}a(t) - C_f v(t) - C_d v^2(t) - C_{roll}) \end{cases}$$

- High energy-demanding training
- Only 50-80 meters Executed

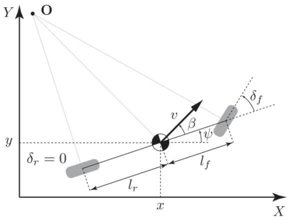
$\Rightarrow$

- No steeering degree of freedom
- Model can be simplified

$$\begin{cases} \dot{p}(t) = v(t) \\ \dot{v}(t) = \dfrac{F(t)}{m} = \dfrac{1}{m}(C_{m1}a(t) - C_f v(t) - C_d v^2(t) - C_{roll}) \end{cases}$$

- The nonlinear term can be simplified

Modeling of the go-kart with the airshield attached



Nonlinear kinematic byclicle model

$$\begin{cases} \dot{x}(t) = v(t)\cos(\psi(t) + \beta(t)) \\ \dot{y}(t) = v(t)\sin(\psi(t) + \beta(t)) \\ \dot{\psi}(t) = \dfrac{v(t)}{l_r}\sin(\beta(t)) \\ \dot{v}(t) = \dfrac{F_x(t)}{m} = \dfrac{1}{m}(C_{m1}a(t) - C_f v(t) - C_d v^2(t) - C_{roll}) \end{cases}$$

- High energy-demanding training
- Only 50-80 meters Executed

$\Rightarrow$

- No steeering degree of freedom
- Model can be simplified

$$\begin{cases} \dot{p}(t) = v(t) \\ \dot{v}(t) = \dfrac{F(t)}{m} = \dfrac{1}{m}(C_{m1}a(t) - C_f v(t)) \end{cases}$$

- The nonlinear term can be simplified

Linear time-invariant system

$$x_{k,t+1} = \begin{bmatrix} p_{k,t+1} \\ v_{k,t+1} \end{bmatrix} = \begin{bmatrix} 1 & dt \\ 0 & 1 - dt\frac{C_f}{m} \end{bmatrix} x_{k,t} + \begin{bmatrix} 0 \\ dt\frac{C_{m1}}{m} \end{bmatrix} u_t$$
$$= A\,x_{k,t} + B\,u_t$$

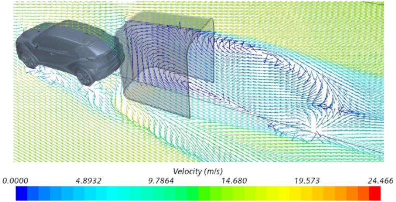
Application setup

Aerodynamics studies [1] have shown that, by running in the slipstream of a shield:

- Runner is isolated from the drag resistance
- A pushing force from behind enhances the speed

Reference distance: $d_{\text{des}} = 2.5$ meters

[1] Italian National Olympic Committee (CONI) Institute of Sports Science
"Aerodynamic Shield – new training support technologies"



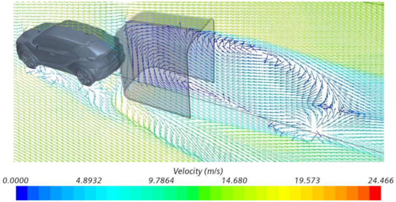
Application setup

Aerodynamics studies [1] have shown that, by running in the slipstream of a shield:

- Runner is isolated from the drag resistance
- A pushing force from behind enhances the speed

Reference distance: $d_{\text{des}} = 2.5$ meters

[1] Italian National Olympic Committee (CONI) Institute of Sports Science
"Aerodynamic Shield – new training support technologies"

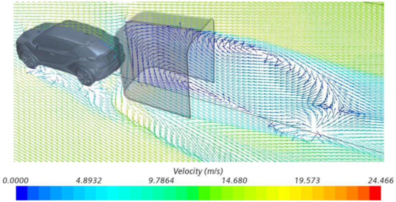


Controller objectives

Application setup

Aerodynamics studies [1] have shown that, by running in the slipstream of a shield:

- Runner is isolated from the drag resistance
- A pushing force from behind enhances the speed



Reference distance: $d_{\text{des}} = 2.5$ meters

[1] Italian National Olympic Committee (CONI) Institute of Sports Science
"Aerodynamic Shield – new training support technologies"

Controller objectives

- Maintain the desired reference distance between airshield and runner

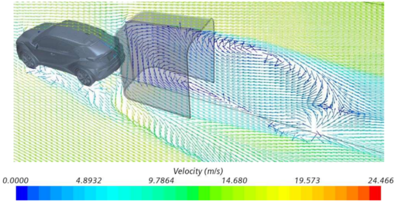
Application setup

Aerodynamics studies [1] have shown that, by running in the slipstream of a shield:

- Runner is isolated from the drag resistance
- A pushing force from behind enhances the speed

Reference distance: $d_{\text{des}} = 2.5$ meters

[1] Italian National Olympic Committee (CONI) Institute of Sports Science
"Aerodynamic Shield – new training support technologies"



Controller objectives

- Maintain the desired reference distance between airshield and runner
- Match, as closely as possible, kart velocity with the runner's one

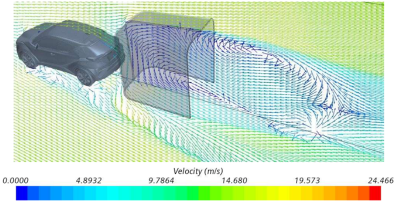
Application setup

Aerodynamics studies [1] have shown that, by running in the slipstream of a shield:

- Runner is isolated from the drag resistance
- A pushing force from behind enhances the speed

Reference distance: $d_{\text{des}} = 2.5$ meters

[1] Italian National Olympic Committee (CONI) Institute of Sports Science
"Aerodynamic Shield – new training support technologies"



Controller objectives

- Maintain the desired reference distance between airshield and runner
- Match, as closely as possible, kart velocity with the runner's one

Controllers formulations

Gain Scheduling Linear Quadratic Regulator

Linear Model Predictive Controller

Offset-free Model Predictive Controller

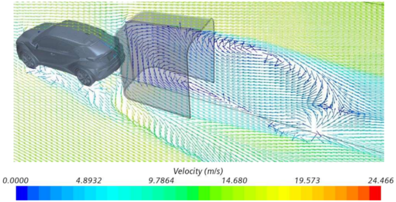
Application setup

Aerodynamics studies [1] have shown that, by running in the slipstream of a shield:

- Runner is isolated from the drag resistance
- A pushing force from behind enhances the speed

Reference distance: $d_{\text{des}} = 2.5$ meters

[1] Italian National Olympic Committee (CONI) Institute of Sports Science
"Aerodynamic Shield – new training support technologies"



Controller objectives

- Maintain the desired reference distance between airshield and runner
- Match, as closely as possible, kart velocity with the runner's one

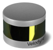
Controllers formulations

Gain Scheduling Linear Quadratic Regulator

Linear Model Predictive Controller

Offset-free Model Predictive Controller

Wheel Encoders
Absolute kart velocity v_k



LiDAR (Velodyne)
Relative distance $\Delta p = p_k - p_r$

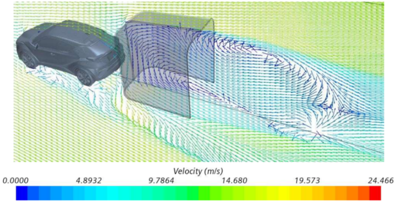
Application setup

Aerodynamics studies [1] have shown that, by running in the slipstream of a shield:

- Runner is isolated from the drag resistance
- A pushing force from behind enhances the speed

Reference distance: $d_{\text{des}} = 2.5$ meters

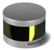
[1] Italian National Olympic Committee (CONI) Institute of Sports Science "Aerodynamic Shield – new training support technologies"



Controller objectives	▶ Maintain the desired reference distance between airshield and runner ▶ Match, as closely as possible, kart velocity with the runner's one

Controllers formulations	**Gain Scheduling Linear Quadratic Regulator**	**Linear Model Predictive Controller**	**Offset-free Model Predictive Controller**

Wheel Encoders
Absolute kart velocity v_k



LiDAR (Velodyne)
Relative distance $\Delta p = p_k - p_r$

Estimated:

- Relative velocity $\Delta v = v_k - v_r$
- Absolute runner acceleration a_r

Gain Scheduling Linear Quadratic Regulator

Linear Quadratic Regulator

$$h_t = \begin{bmatrix} \Delta p_t \\ \Delta v_t \end{bmatrix} = \begin{bmatrix} p_{k,t} - p_{r,t} \\ v_{k,t} - v_{r,t} \end{bmatrix}$$

$$h_{\text{des}} = \begin{bmatrix} d_{\text{des}} \\ 0 \end{bmatrix}$$

- Offline: Compute the Linear Quadratic Regulator: K
- Control the system using the state feedback $u^* = -K(h_t - h_{\text{des}})$

Gain Scheduling Linear Quadratic Regulator

Linear Quadratic Regulator

$$h_t = \begin{bmatrix} \Delta p_t \\ \Delta v_t \end{bmatrix} = \begin{bmatrix} p_{k,t} - p_{r,t} \\ v_{k,t} - v_{r,t} \end{bmatrix}$$

$$h_{\text{des}} = \begin{bmatrix} d_{\text{des}} \\ 0 \end{bmatrix}$$

- Offline: Compute the Linear Quadratic Regulator: K
- Control the system using the state feedback $u^* = -K(h_t - h_{\text{des}})$

Gain Scheduling Approach

Catch-up maneuver K_{catch}

Cruise control maneuver K_{cruise}

Gain Scheduling Linear Quadratic Regulator

Linear Quadratic Regulator

$$h_t = \begin{bmatrix} \Delta p_t \\ \Delta v_t \end{bmatrix} = \begin{bmatrix} p_{k,t} - p_{r,t} \\ v_{k,t} - v_{r,t} \end{bmatrix}$$

$$h_{\text{des}} = \begin{bmatrix} d_{\text{des}} \\ 0 \end{bmatrix}$$

- Offline: Compute the Linear Quadratic Regulator: K
- Control the system using the state feedback $u^* = -K(h_t - h_{\text{des}})$

Gain Scheduling Approach

Catch-up maneuver K_{catch}

In the first couple of seconds the runner has an acceleration exceeding the maximum go-kart capabilities.

- Ask the runner to start the sprint at $\Delta p_{\text{init}} = d_{\text{des}} + \bar{d}$
- Control using $u^* = -K_{\text{catch}}(h_t - h_{\text{des}})$
- Match the velocities with a safe distance bigger than d_{des}

Cruise control maneuver K_{cruise}

Gain Scheduling Linear Quadratic Regulator

Linear Quadratic Regulator

$$h_t = \begin{bmatrix} \Delta p_t \\ \Delta v_t \end{bmatrix} = \begin{bmatrix} p_{k,t} - p_{r,t} \\ v_{k,t} - v_{r,t} \end{bmatrix}$$

$$h_{\text{des}} = \begin{bmatrix} d_{\text{des}} \\ 0 \end{bmatrix}$$

- Offline: Compute the Linear Quadratic Regulator: K
- Control the system using the state feedback $u^* = -K(h_t - h_{\text{des}})$

Gain Scheduling Approach

Catch-up maneuver K_{catch}

In the first couple of seconds the runner has an acceleration exceeding the maximum go-kart capabilities.

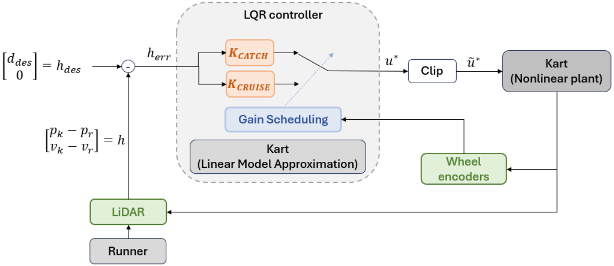
- Ask the runner to start the sprint at $\Delta p_{\text{init}} = d_{\text{des}} + \bar{d}$
- Control using $u^* = -K_{\text{catch}}(h_t - h_{\text{des}})$
- Match the velocities with a safe distance bigger than d_{des}

Cruise control maneuver K_{cruise}

Once the runner and go-kart velocities are matched at 80%

- Reduce the distance up to the reference d_{des}
- Control using $u^* = -K_{\text{cruise}}(h_t - h_{\text{des}})$
- Maintain the desired reference distance
- Maintain to zero the relative velocity

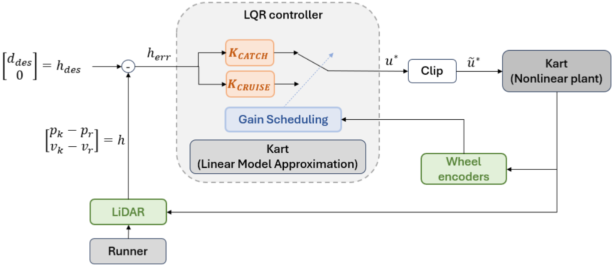
Gain Scheduling Linear Quadratic Regulator



Gain Scheduling

if $(v_{k,t} \geq 0.8\, v_{r,t})$ **and** $(\text{catched} = \text{false})$:
  catched = true;
end if
if (catched = false):
  $u^* = -K_{\text{catch}}(h_t - h_{\text{des}});$
else
  $u^* = -K_{\text{cruise}}(h_t - h_{\text{des}});$
end if

Gain Scheduling Linear Quadratic Regulator



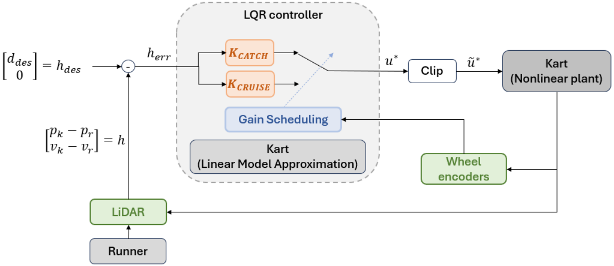
Gain Scheduling

if $(v_{k,t} \geq 0.8\, v_{r,t})$ **and** (catched = false) :
    catched = true;
end if
if (catched = false):
    $u^* = -K_{\text{catch}}(h_t - h_{\text{des}});$
else
    $u^* = -K_{\text{cruise}}(h_t - h_{\text{des}});$
end if

Pros of LQR controller

- Reduced computational effort

Gain Scheduling Linear Quadratic Regulator



Gain Scheduling

if $(v_{k,t} \geq 0.8\, v_{r,t})$ **and** (catched = false) :
  catched = true;
end if
if (catched = false):
  $u^* = -K_{\text{catch}}(h_t - h_{\text{des}});$
else
  $u^* = -K_{\text{cruise}}(h_t - h_{\text{des}});$
end if

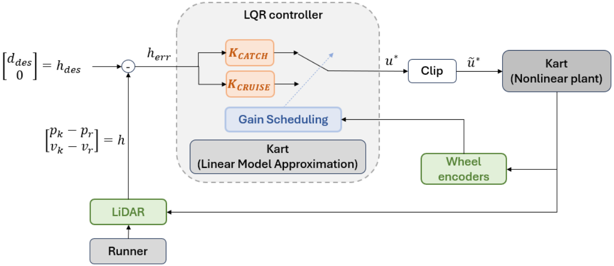
Pros of LQR controller

- Reduced computational effort

Cons of LQR controller

- No explicit constraints management

Gain Scheduling Linear Quadratic Regulator



Gain Scheduling

if $(v_{k,t} \geq 0.8\, v_{r,t})$ **and** (catched = false) :
  catched = true;
end if
if (catched = false):
  $u^* = -K_{\text{catch}}(h_t - h_{\text{des}});$
else
  $u^* = -K_{\text{cruise}}(h_t - h_{\text{des}});$
end if

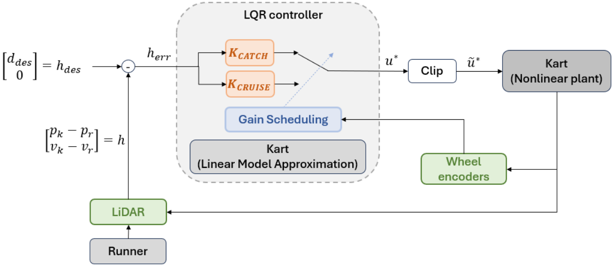
Pros of LQR controller

- Reduced computational effort

Cons of LQR controller

- No explicit constraints management
- Actuator limits: $a_{\text{min}} \leq u \leq a_{\text{max}}$

Gain Scheduling Linear Quadratic Regulator



Gain Scheduling

if $(v_{k,t} \geq 0.8\, v_{r,t})$ **and** (catched = false) :
  catched = true;
end if
if (catched = false):
  $u^* = -K_{\text{catch}}(h_t - h_{\text{des}});$
else
  $u^* = -K_{\text{cruise}}(h_t - h_{\text{des}});$
end if

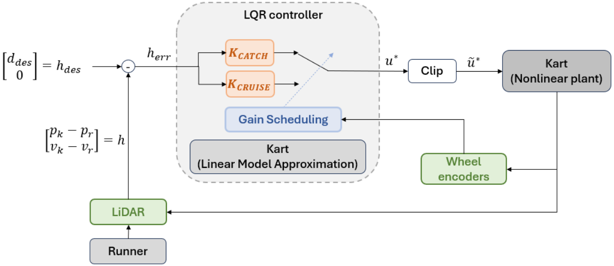
Pros of LQR controller

- Reduced computational effort

Cons of LQR controller

- No explicit constraints management
- Actuator limits: $a_{\min} \leq u \leq a_{\max}$
- No state constraints (i.e. $\Delta p \leq d_{\text{safe}}$)

Gain Scheduling Linear Quadratic Regulator



Gain Scheduling

if $(v_{k,t} \geq 0.8\, v_{r,t})$ **and** (catched = false) :
  catched = true;
end if
if (catched = false):
  $u^* = -K_{\text{catch}}(h_t - h_{\text{des}});$
else
  $u^* = -K_{\text{cruise}}(h_t - h_{\text{des}});$
end if

Pros of LQR controller

- Reduced computational effort

Cons of LQR controller

- No explicit constraints management
- Actuator limits: $a_{\text{min}} \leq u \leq a_{\text{max}}$
- No state constraints (i.e. $\Delta p \leq d_{\text{safe}}$)
- No assumption on runner expected behaviour

Linear Model Predictive Control

$$\begin{bmatrix} \Delta p_{t+1} \\ \Delta v_{t+1} \\ v_{k,t+1} \end{bmatrix} = \begin{bmatrix} 1 & dt & 0 \\ 0 & 1 & -dt\frac{C_f}{m} \\ 0 & 0 & 1 - dt\frac{C_f}{m} \end{bmatrix} x_{p,t} + \begin{bmatrix} 0 \\ dt\frac{C_{m1}}{m} \\ dt\frac{C_{m1}}{m} \end{bmatrix} u_t + \begin{bmatrix} 0 \\ -dt a_{r,t} \\ 0 \end{bmatrix}$$
$$= A_p\, x_{p,t} + B_p\, u_t + a_t$$

Linear Model Predictive Control

$$\begin{bmatrix}\Delta p_{t+1}\\ \Delta v_{t+1}\\ v_{k,t+1}\end{bmatrix} = \begin{bmatrix}1 & dt & 0\\ 0 & 1 & -dt\frac{C_f}{m}\\ 0 & 0 & 1-dt\frac{C_f}{m}\end{bmatrix} x_{p,t} + \begin{bmatrix}0\\ dt\frac{C_{m1}}{m}\\ dt\frac{C_{m1}}{m}\end{bmatrix} u_t + \begin{bmatrix}0\\ -dt a_{r,t}\\ 0\end{bmatrix}$$

$$= A_p\, x_{p,t} + B_p\, u_t + a_t$$

Pros of MPC controller

- Incorporate a runner model in the predictions

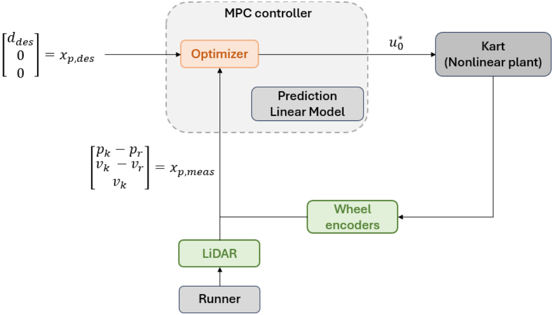
Linear Model Predictive Control

$$\begin{bmatrix} \Delta p_{t+1} \\ \Delta v_{t+1} \\ v_{k,t+1} \end{bmatrix} = \begin{bmatrix} 1 & dt & 0 \\ 0 & 1 & -dt\,\frac{C_f}{m} \\ 0 & 0 & 1 - dt\,\frac{C_f}{m} \end{bmatrix} x_{p,t} + \begin{bmatrix} 0 \\ dt\,\frac{C_{m1}}{m} \\ dt\,\frac{C_{m1}}{m} \end{bmatrix} u_t + \begin{bmatrix} 0 \\ -dta_{r,t} \\ 0 \end{bmatrix}$$

$$= A_p\, x_{p,t} + B_p\, u_t + a_t$$

Pros of MPC controller

- Incorporate a runner model in the predictions
- Semplification of the control architecture

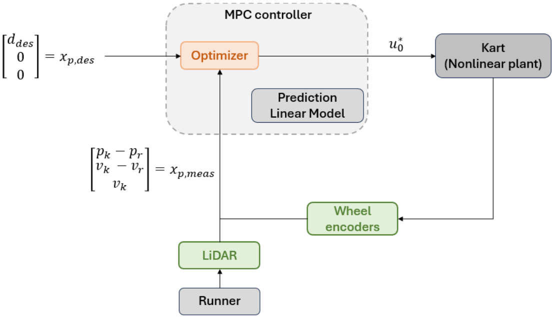


Linear Model Predictive Control

$$
\begin{bmatrix} \Delta p_{t+1} \\ \Delta v_{t+1} \\ v_{k,t+1} \end{bmatrix} = \begin{bmatrix} 1 & dt & 0 \\ 0 & 1 & -dt\,\frac{C_f}{m} \\ 0 & 0 & 1 - dt\,\frac{C_f}{m} \end{bmatrix} x_{p,t} + \begin{bmatrix} 0 \\ dt\,\frac{C_{m1}}{m} \\ dt\,\frac{C_{m1}}{m} \end{bmatrix} u_t + \begin{bmatrix} 0 \\ -dt a_{r,t} \\ 0 \end{bmatrix}
$$
$$
= A_p\, x_{p,t} + B_p\, u_t + a_t
$$

Pros of MPC controller

- Incorporate a runner model in the predictions
- Semplification of the control architecture
- Constraints explicitly included in the formulation



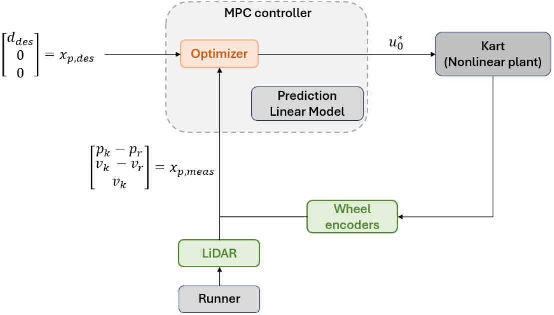
Optimizer

$$
\min_{\boldsymbol{x},\boldsymbol{u}} \sum_{i=t}^{t+N-1} \|(x_{p,i|t} - x_{p,\text{des}})\|_Q + \|u_{i|t}\|_R
$$

subj. to
$$
\begin{aligned}
& x_{p,i+1|t} = A_p\, x_{p,i|t} + B_p\, u_{i|t} + a_t \\
& a_{\min} \le u_{i|t} \le a_{\max} \\
& d_{\text{safe}} \le \Delta p_{i|t} \\
& x_{p,t|t} = x_{p,\text{meas}}
\end{aligned}
$$

Linear Model Predictive Control

$$\begin{bmatrix}\Delta p_{t+1}\\ \Delta v_{t+1}\\ v_{k,t+1}\end{bmatrix} = \begin{bmatrix}1 & dt & 0\\ 0 & 1 & -dt\,\frac{C_f}{m}\\ 0 & 0 & 1-dt\,\frac{C_f}{m}\end{bmatrix} x_{p,t} + \begin{bmatrix}0\\ dt\,\frac{C_{m1}}{m}\\ dt\,\frac{C_{m1}}{m}\end{bmatrix} u_t + \begin{bmatrix}0\\ -dta_{r,t}\\ 0\end{bmatrix}$$
$$= A_p\, x_{p,t} + B_p\, u_t + a_t$$



Pros of MPC controller

- Incorporate a runner model in the predictions
- Semplification of the control architecture
- Constraints explicitly included in the formulation
- Safety guarantees via state constraints

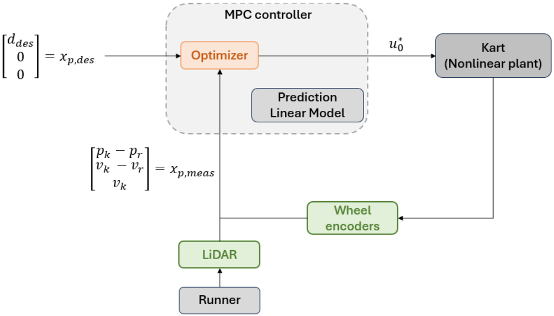
Optimizer

$$\min_{\boldsymbol{x},\boldsymbol{u}} \sum_{i=t}^{t+N-1} \|(x_{p,i|t} - x_{p,\text{des}})\|_Q + \|u_{i|t}\|_R$$

subj. to
$$x_{p,i+1|t} = A_p\, x_{p,i|t} + B_p\, u_{i|t} + a_t$$
$$a_{\min} \le u_{i|t} \le a_{\max}$$
$$d_{\text{safe}} \le \Delta p_{i|t}$$
$$x_{p,t|t} = x_{p,\text{meas}}$$

Linear Model Predictive Control

$$\begin{bmatrix} \Delta p_{t+1} \\ \Delta v_{t+1} \\ v_{k,t+1} \end{bmatrix} = \begin{bmatrix} 1 & dt & 0 \\ 0 & 1 & -dt\,\frac{C_f}{m} \\ 0 & 0 & 1-dt\,\frac{C_f}{m} \end{bmatrix} x_{p,t} + \begin{bmatrix} 0 \\ dt\,\frac{C_{m1}}{m} \\ dt\,\frac{C_{m1}}{m} \end{bmatrix} u_t + \begin{bmatrix} 0 \\ -dt a_{r,t} \\ 0 \end{bmatrix}$$
$$= A_p\, x_{p,t} + B_p\, u_t + a_t$$



Pros of MPC controller

- Incorporate a runner model in the predictions
- Semplification of the control architecture
- Constraints explicitly included in the formulation
- Safety guarantees via state constraints

Optimizer

$$\min_{\boldsymbol{x},\boldsymbol{u}} \quad \sum_{i=t}^{t+N-1} \|(x_{p,i|t} - x_{p,\text{des}})\|_Q + \|u_{i|t}\|_R$$

subj. to
$$x_{p,i+1|t} = A_p\, x_{p,i|t} + B_p\, u_{i|t} + a_t$$
$$a_{\text{min}} \leq u_{i|t} \leq a_{\text{max}}$$
$$d_{\text{safe}} \leq \Delta p_{i|t}$$
$$x_{p,t|t} = x_{p,\text{meas}}$$

Cons of MPC controller

- Higher computational effort

Offset-free Model Predictive Control [1]

Cons of MPC controller

- A nonlinear system is controlled using a linear prediction model

[1] U. Maeder, F. Borrelli, M. Morari
"Linear offset-free Model Predictive Control"

Offset-free Model Predictive Control [1]

Cons of MPC controller

- A nonlinear system is controlled using a linear prediction model
- **Model-plant mismatch** can lead to suboptimal performances

[1] U. Maeder, F. Borrelli, M. Morari
"Linear offset-free Model Predictive Control"

Offset-free Model Predictive Control [1]

Cons of MPC controller

- A nonlinear system is controlled using a linear prediction model
- **Model-plant mismatch** can lead to suboptimal performances

$\Rightarrow$

Pros of Offset-free MPC controller

- Estimate the non linearity via Luemberger observer

[1] U. Maeder, F. Borrelli, M. Morari
"Linear offset-free Model Predictive Control"

Offset-free Model Predictive Control [1]

Cons of MPC controller

- A nonlinear system is controlled using a linear prediction model
- **Model-plant mismatch** can lead to suboptimal performances

⇒

Pros of Offset-free MPC controller

- Estimate the non linearity via Luemberger observer
- Modify the MPC formulation to achieve offset-free

[1] U. Maeder, F. Borrelli, M. Morari
"Linear offset-free Model Predictive Control"

Offset-free Model Predictive Control [1]

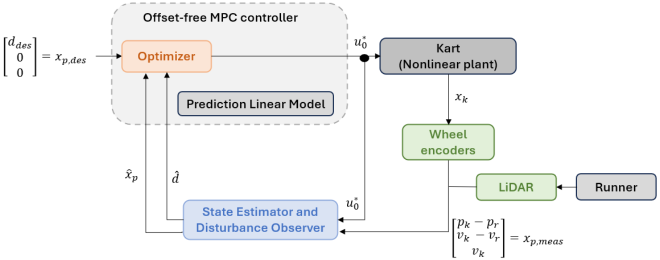
Cons of MPC controller

- A nonlinear system is controlled using a linear prediction model
- **Model-plant mismatch** can lead to suboptimal performances

$\Rightarrow$

Pros of Offset-free MPC controller

- Estimate the non linearity via Luemberger observer
- Modify the MPC formulation to achieve offset-free



[1] U. Maeder, F. Borrelli, M. Morari
"Linear offset-free Model Predictive Control"

Offset-free Model Predictive Control [1]

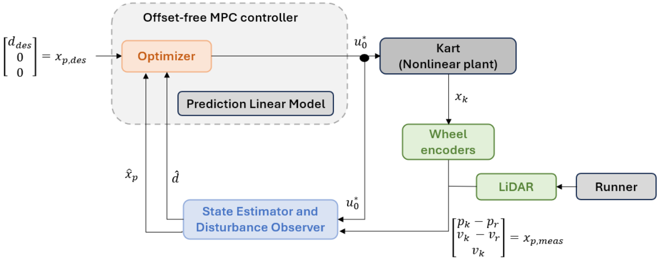
Cons of MPC controller

- A nonlinear system is controlled using a linear prediction model
- **Model-plant mismatch** can lead to suboptimal performances

$\Rightarrow$

Pros of Offset-free MPC controller

- Estimate the non linearity via Luemberger observer
- Modify the MPC formulation to achieve offset-free



Optimizer

$$\min_{\boldsymbol{x},\boldsymbol{u}} \sum_{i=t}^{t+N-1} \|x_{p,i|t} - \bar{x}_t\|_Q + \|u_{i|t} - \bar{u}_t\|_R$$

subj. to

$$x_{p,i+1|t} = A_p\, x_{p,i|t} + B_p\, u_{i|t} + a_t + B_d\, d_{i|t}$$
$$d_{i+1|t} = d_{i|t}$$
$$a_{\text{min}} \leq u_{i|t} \leq a_{\text{max}}$$
$$d_{\text{safe}} \leq \Delta p_{i|t}$$
$$x_{p,t|t} = \hat{x}_t$$
$$d_{t|t} = \hat{d}_t$$

[1] U. Maeder, F. Borrelli, M. Morari
"Linear offset-free Model Predictive Control"

Offset-free Model Predictive Control [1]

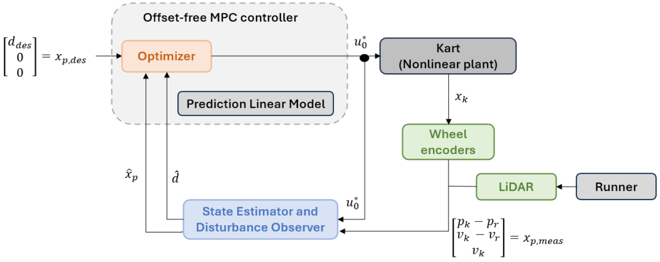
Cons of MPC controller

- A nonlinear system is controlled using a linear prediction model
- **Model-plant mismatch** can lead to suboptimal performances

$\Rightarrow$

Pros of Offset-free MPC controller

- Estimate the non linearity via Luemberger observer
- Modify the MPC formulation to achieve offset-free



Optimizer

$$\min_{\boldsymbol{x},\boldsymbol{u}} \sum_{i=t}^{t+N-1} \|x_{p,i|t} - \bar{x}_t\|_Q + \|u_{i|t} - \bar{u}_t\|_R$$

subj. to
$$x_{p,i+1|t} = A_p\, x_{p,i|t} + B_p\, u_{i|t} + a_t + B_d\, d_{i|t}$$
$$d_{i+1|t} = d_{i|t}$$
$$a_{\text{min}} \leq u_{i|t} \leq a_{\text{max}}$$
$$d_{\text{safe}} \leq \Delta p_{i|t}$$
$$x_{p,t|t} = \hat{x}_t$$
$$d_{t|t} = \hat{d}_t$$

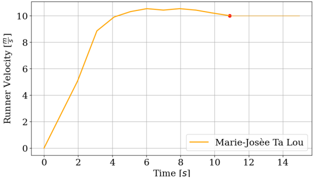
Disturbance Observer

$$\begin{bmatrix} \hat{x}_{p,t+1} \\ \hat{d}_{t+1} \end{bmatrix} = \begin{bmatrix} A_p & B_d \\ 0 & I \end{bmatrix} \begin{bmatrix} \hat{x}_{p,t} \\ \hat{d}_t \end{bmatrix} + \begin{bmatrix} B_p \\ 0 \end{bmatrix} u + \begin{bmatrix} L_x \\ L_d \end{bmatrix} \left(-y_{m,t} + \begin{bmatrix} C & C_d \end{bmatrix} \begin{bmatrix} \hat{x}_{p,t} \\ \hat{d}_t \end{bmatrix}\right)$$

[1] U. Maeder, F. Borrelli, M. Morari
"Linear offset-free Model Predictive Control"

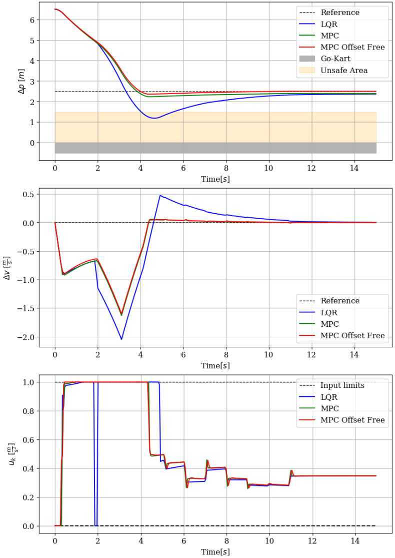
Simulation tests and Controllers comparison

- Simulation implementation of the controllers
- Tests on the same velocity profile (elite athlete)



Initial distance: $\Delta p_{\text{init}} = d_{\text{des}} + \bar{d}, \quad \bar{d} = 4\text{m}$

	LQR	**MPC**	**MPC Offset-free**
mean(Δp)	0.8892	0.6867	0.6168
mean(Δv)	0.3987	0.2611	0.2522
IAE on Δp	13.2352	10.1978	9.1515
IAE on Δv	5.9811	3.9176	3.7826
min(Δp)	1.1885	2.2371	2.3605
Energy consumption	0.5208	0.5200	0.5207
Steady state error	0.1354	0.1079	0.0009
"Rise" time to d_{des}	3.3	3.75	3.85



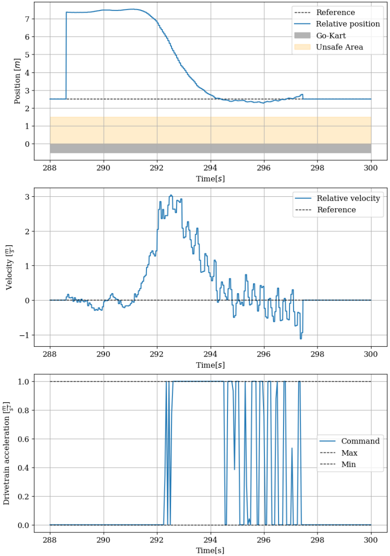
Hardware-in-the-loop tests

Focus on the catch-up maneuver

Hardware-in-the-loop tests

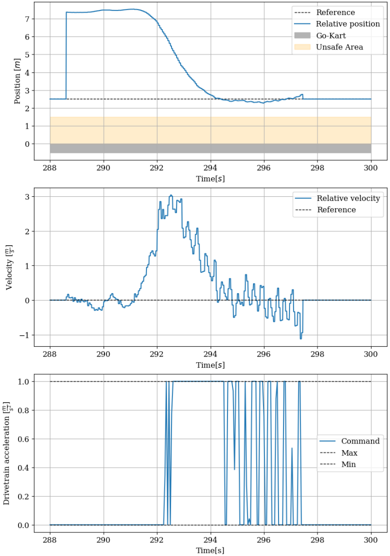
Focus on maintaining a constant distance with the runner having an almost constant velocity

Linear Model Predictive Controller Hardware-in-the-loop tests

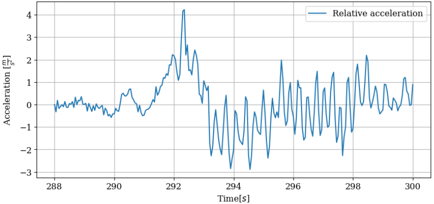


Oscillations around steady state condition especially in the input variable

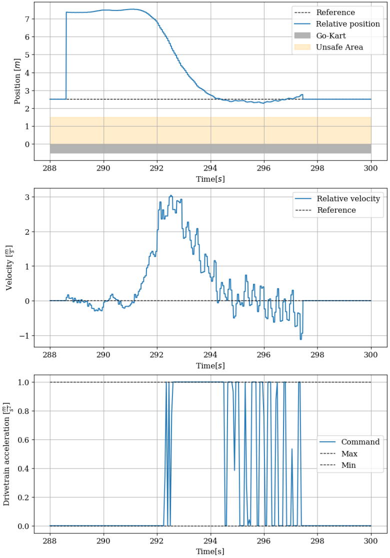
Linear Model Predictive Controller Hardware-in-the-loop tests



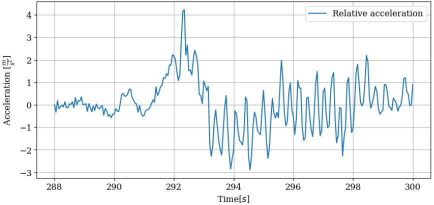
Oscillations around steady state condition especially in the input variable



Linear Model Predictive Controller Hardware-in-the-loop tests



Oscillations around steady state condition especially in the input variable



The system results reliable in the usage on the track and field racetrack

Runner absolute acceleration estimation can be improved by:

- adding an IMU on the runner's body
- introducing a learning approach

Conclusion

Contributions

- Analysis of the application and requirements definition
- Control design
  - Gain Scheduling Linear Quadratic Regulator
  - Linear Model Predictive Control
  - Offset-free Model Predictive Control
- Simulation test and numerical comparison of performances
- Hardware-in-the-loop implementation and tests

ROS 2 docker GitHub

Conclusion

Contributions

- Analysis of the application and requirements definition
- Control design
  - Gain Scheduling Linear Quadratic Regulator
  - Linear Model Predictive Control
  - Offset-free Model Predictive Control
- Simulation test and numerical comparison of performances
- Hardware-in-the-loop implementation and tests

Future developments

- Robust MPC formulation using the estimated disturbance
- Learning MPC formulation for enhance the runner future behaviour prediction
- Improve the runner acceleration estimation

Conclusion

Contributions

- Analysis of the application and requirements definition
- Control design
  - Gain Scheduling Linear Quadratic Regulator
  - Linear Model Predictive Control
  - Offset-free Model Predictive Control
- Simulation test and numerical comparison of performances
- Hardware-in-the-loop implementation and tests

ROS 2 docker GitHub

Future developments

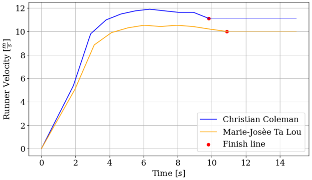
- Robust MPC formulation using the estimated disturbance
- Learning MPC formulation for enhance the runner future behaviour prediction
- Improve the runner acceleration estimation

Thanks for your attention

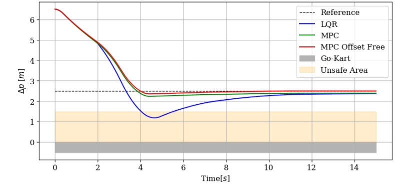
Female athlete simulation results - Controllers comparison

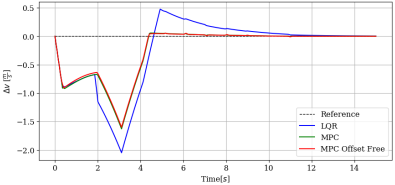
Test the three controllers on the same velocity profile
mimic the one of a female professional athlete during a competition.

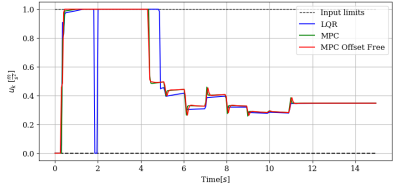
Initial distance: $\Delta p_{\text{init}} = d_{\text{des}} + \bar{d}, \quad \bar{d} = 4\text{m}$



	LQR	MPC	MPC Offset-free
mean(Δp)	0.8892	0.6867	0.6168
mean(Δv)	0.3987	0.2611	0.2522
IAE on Δp	13.2352	10.1978	9.1515
IAE on Δv	5.9811	3.9176	3.7826
min(Δp)	1.1885	2.2371	2.3605
Energy consumption	0.5208	0.5200	0.5207
Steady state error	0.1354	0.1079	0.0009
"Rise" time to d_{des}	3.3	3.75	3.85



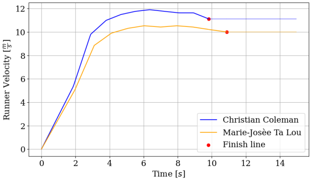




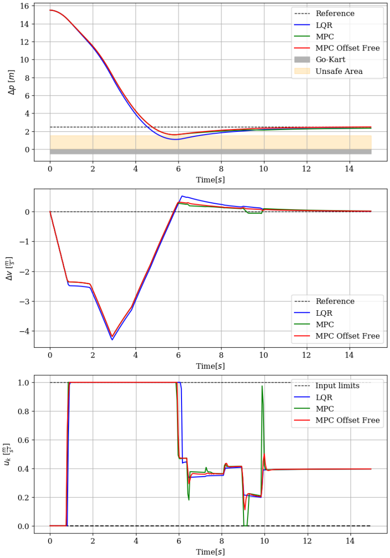
Male athlete simulation results - Controllers comparison

Test the three controllers on the same velocity profile
mimic the one of a male professional athlete during a competition.

Initial distance: $\Delta p_{\text{init}} = d_{\text{des}} + \bar{d}, \quad \bar{d} = 13\text{m}$



	LQR	MPC	MPC Offset-free
mean(Δp)	2.5984	2.5636	2.4848
mean(Δv)	1.0037	0.9392	0.9419
IAE on Δp	38.6465	38.1247	36.9459
IAE on Δv	15.0578	14.0901	14.1308
min(Δp)	1.0802	1.6036	1.6105
Energy consumption	0.5694	0.5694	0.5697
Steady state error	0.1689	0.1784	0.0350
"Rise" time to d_{des}	4.5	4.7	4.7



Controllers computational effort

Wheel Encoders
Frequency: 100 Hz
Sampling time: 10 ms

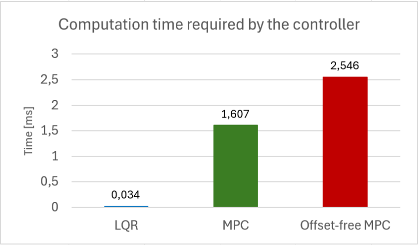
LiDAR (Velodyne)
Frequency: 20 Hz
Sampling time: 50 ms

Controller
Frequency: 20 Hz
Sampling time: 50 ms

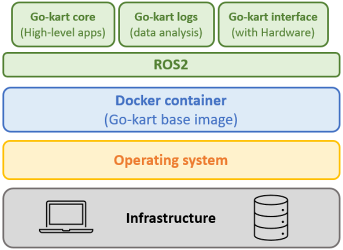
Computer board
AMD Ryzen 7 5700G
Frequency: 3.8 GHz

Controllers computational required efforts
computed in the Python Simulation Environment

	LQR	MPC	Offset-free MPC
Average Computation Time [ms]	0.034	1.607	2.546
Sampling Time Usage [%]	0.07	3.21	5.09



Go-kart repository for hardware implementation



The repo is composed of four different docker images

- **Base image**: Ubuntu 20.04, ROS 2 and communication with other images
- **Go-kart logs**: Data analysis (from binary data to csv files)
- **Go-kart interface**: Interface with hardware components (actuators and sensors)
- **Go-kart core**: High level applications (estimation, control, visualization, ecc)