frontiers

HUMANOIDS 2022

From Walking to Running: 3D Humanoid Gait Generation via MPC

F. M. Smaldone, N. Scianca, L. Lanari and G. Oriolo

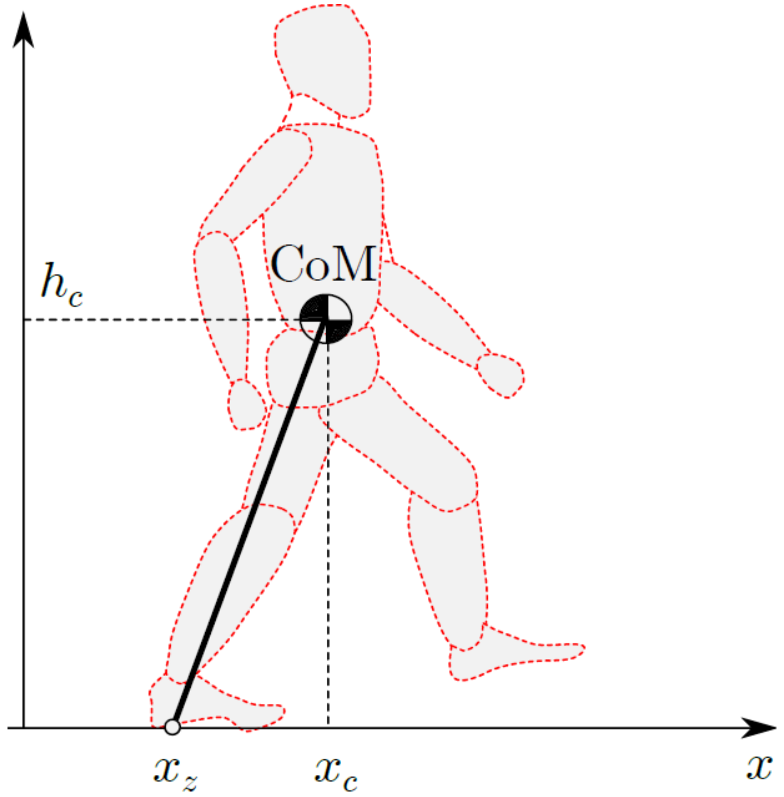
Dipartimento di Ingegneria Informatica Automatica e Gestionale Antonio Ruberti

Sapienza Università di Roma

dynamic balance

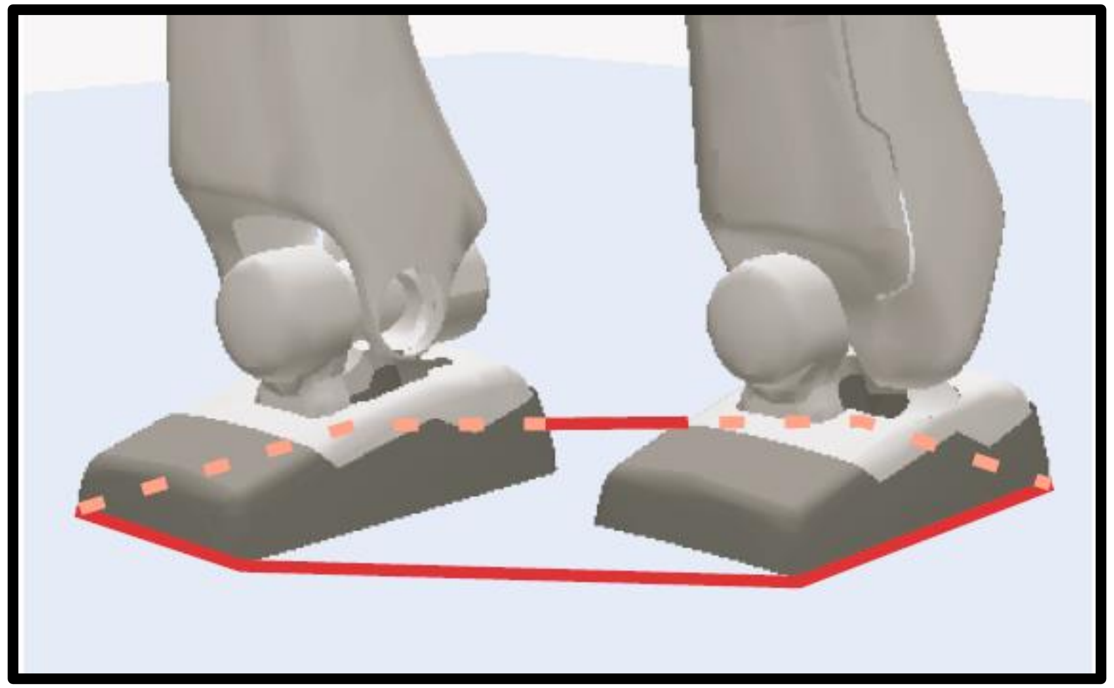
- Newton-Euler equations (wrt to the **ZMP**)

$$
\begin{aligned}
m(\ddot{\boldsymbol{p}}_c - \boldsymbol{g}) &= \boldsymbol{f} \\
(\boldsymbol{p}_z - \boldsymbol{p}_c) \times \boldsymbol{f} &= 0
\end{aligned}
$$



- on flat ground, the CoM height is usually kept constant to obtain the Linear Inverted Pendulum (**LIP**) model

- the ZMP must be at all times within the **support polygon** of the robot

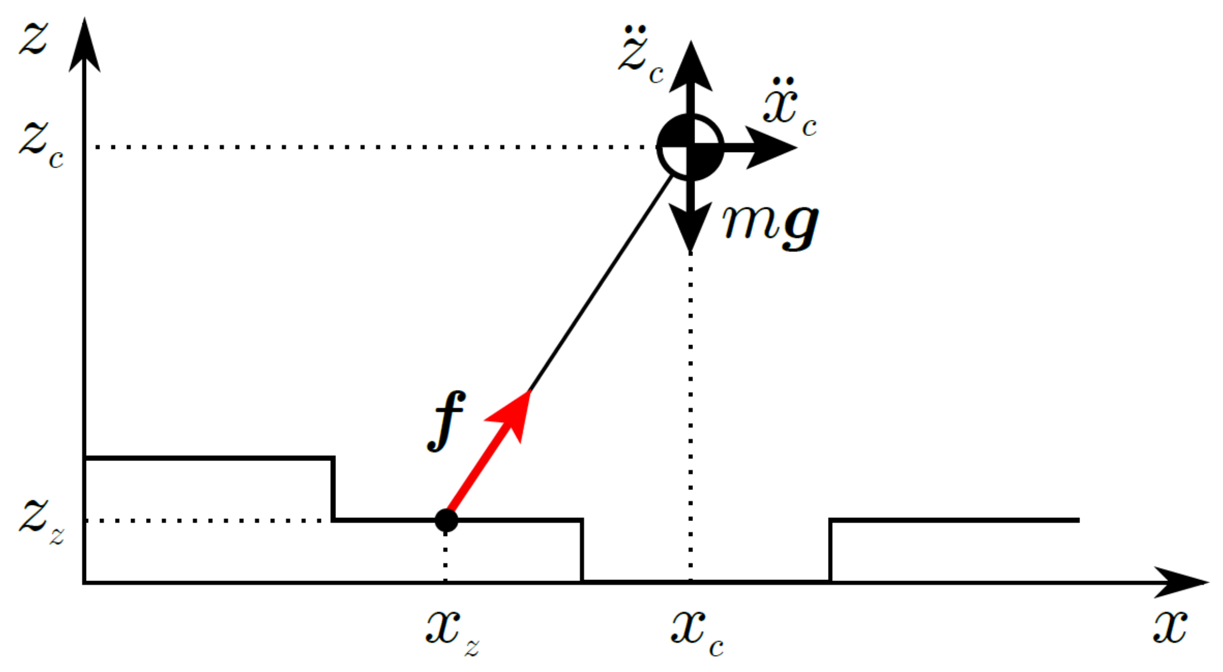


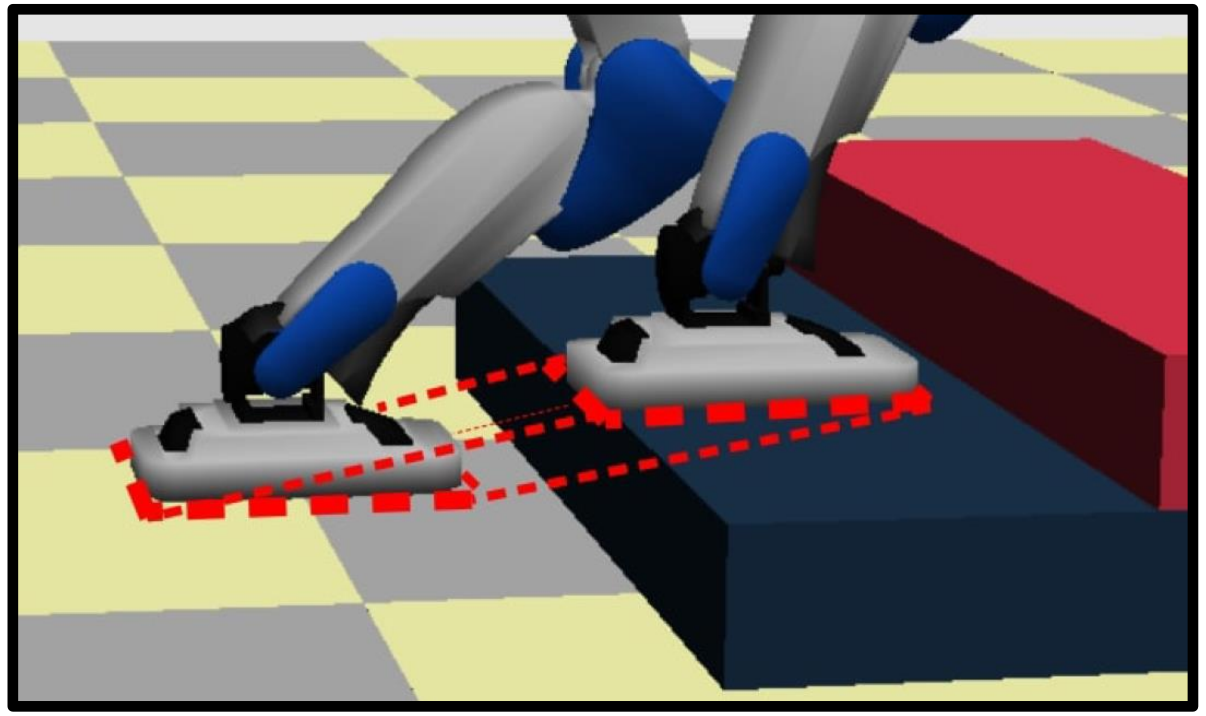
dynamic balance in 3D

- freeing the CoM height leads to the Variable-Height Inverted Pendulum (**VH-IP**) model, in which height variations modify the pendulum natural frequency λ

- we consider as admissible region for the 3D ZMP the convex hull of active contact surfaces, which in a world of stairs becomes an **oblique prism**

- however, if λ is seen as an additional input of the model, it will introduce a **nonlinearity**





time-varying 3D model

- by solving the vertical dynamics first, the variable stiffness can be computed along the prediction and considered as a **time-varying** parameter, removing the nonlinearity

$$\ddot{z}_c = \frac{f_z}{m} - g$$

vertical dynamics

$$\lambda = \frac{\ddot{z}_c + g}{z_c - z_z}$$

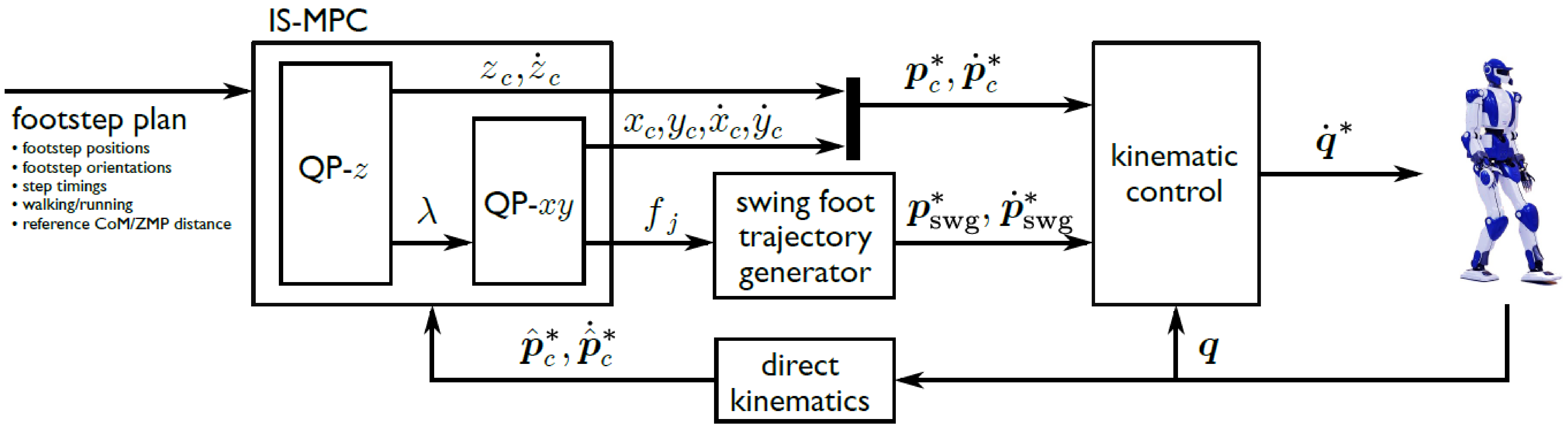
variable stiffness

$$\begin{aligned} \ddot{x}_c &= \lambda(x_c - x_z) \\ \ddot{y}_c &= \lambda(y_c - y_z) \end{aligned}$$

horizontal dynamics

- during flight phases, the dynamics are that of a **free-falling mass** → $\ddot{\boldsymbol{p}}_c = \boldsymbol{g}$

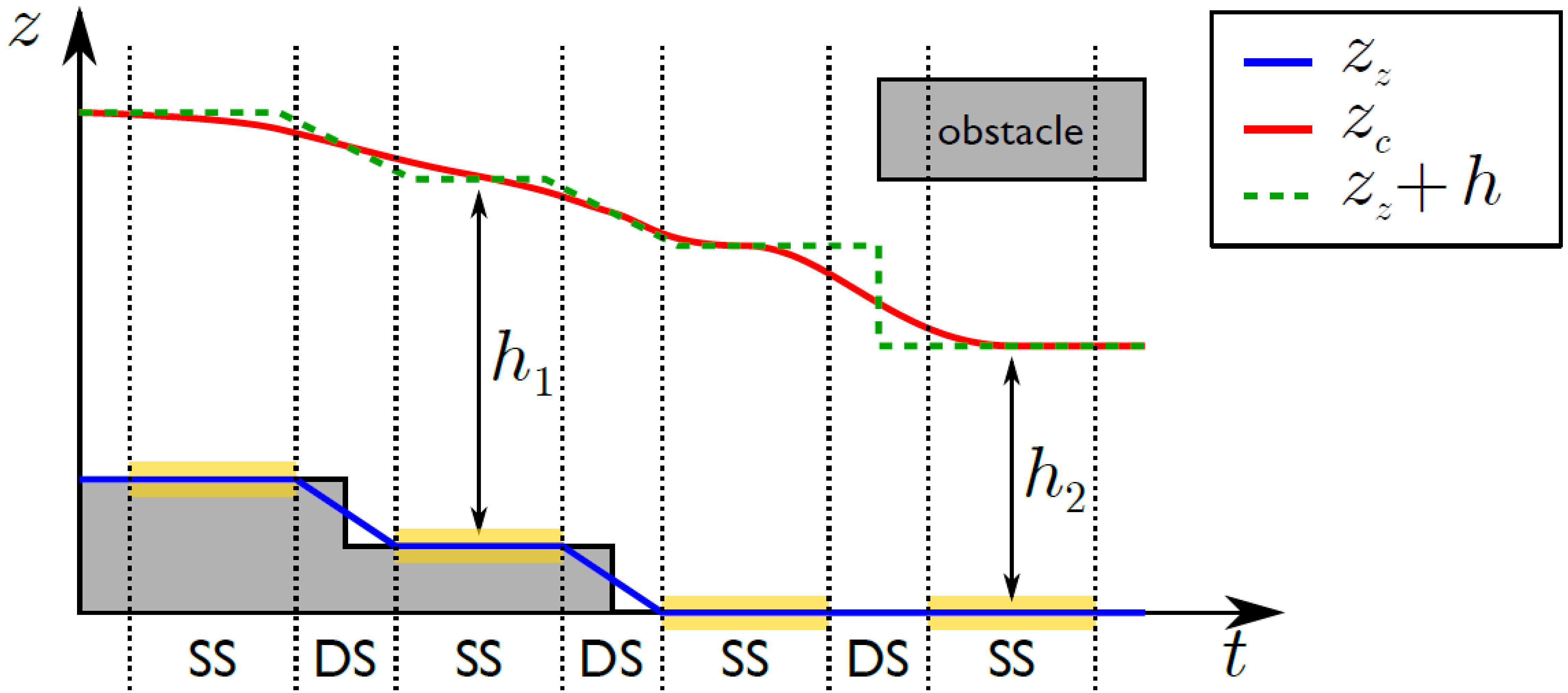
control architecture



- the controller uses a **two-stage MPC** scheme
  - input: candidate **footstep plan**
  - outputs: **CoM** position and velocity and **adapted footsteps**

- a swing foot trajectory reaching the adapted footstep is generated,
  then CoM and swing foot trajectories are tracked by the kinematic controller

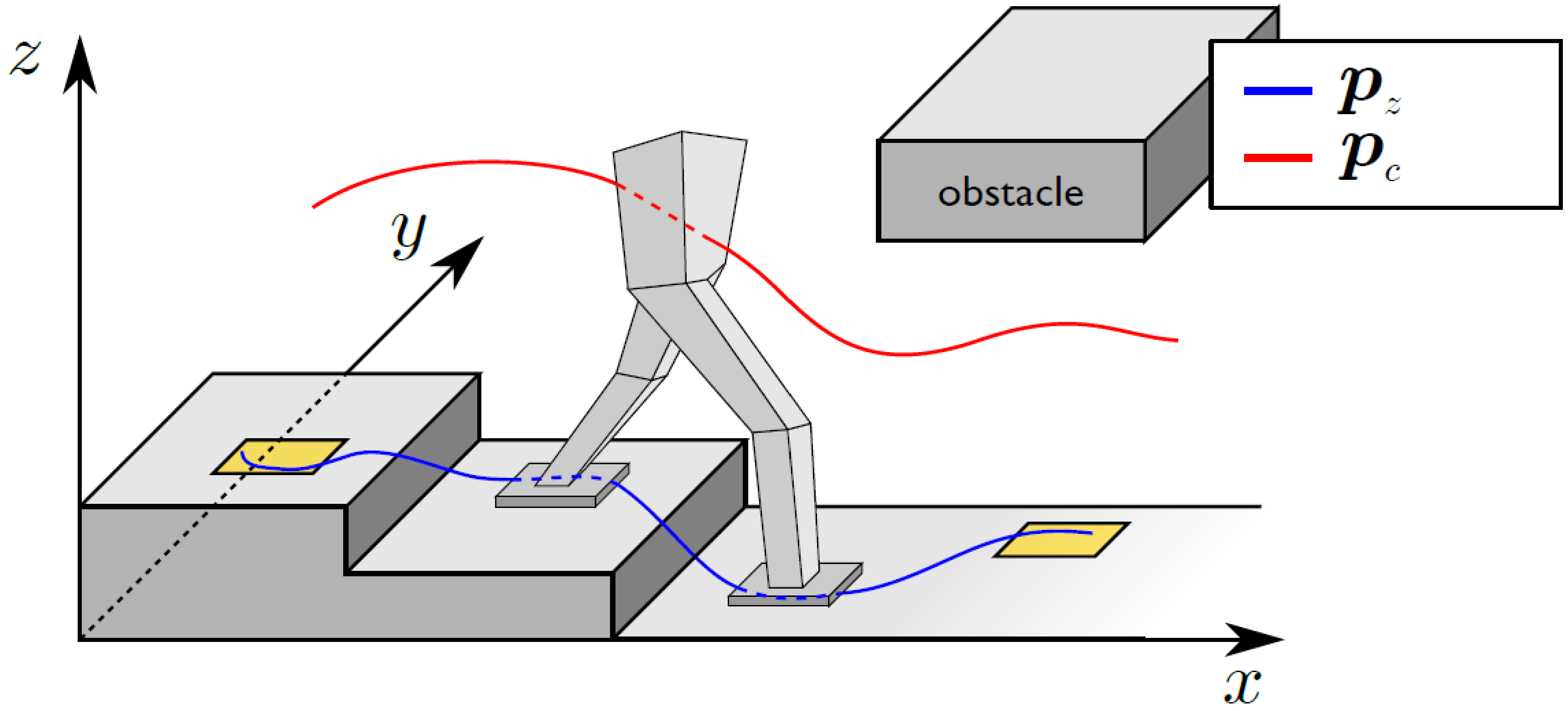
control architecture



- the first stage determines a Ground Reaction Force (**GRF**) profile over the control horizon to generate **a vertical CoM** trajectory which tracks the reference height as closely as possible

- the values of λ can be computed using the vertical CoM trajectory generated at this stage

control architecture



- in the second stage, both ZMP trajectory and adapted footsteps are chosen in such a way to generate a horizontal **CoM trajectory**

- the resulting gait is guaranteed to be both dynamically **balanced** and internally **stable**

ground reaction force constraints

- to avoid slipping during contact phases, the GRF must satisfy a condition of **sufficient friction**

$$f_z^{k+i} \geq f_z^{\min}$$

where $f_z^{\min}$ is a minimum value computed based on a simple Coulomb friction model

- during **flight** phases, this constraint is replaced by

$$f_z^{k+i} = 0$$

vertical dynamics QP

- the optimization problem solved in the first stage is called QP-z

$$\min_{F_z^k} \underbrace{||Z_c^k - Z_c^{k,*}||^2}_{\text{CoM height tracking}} + \alpha_z \underbrace{||\dot{Z}_c^k||^2}_{\text{vertical CoM velocity}} + \beta_z \underbrace{||\Delta F_z^k||^2}_{\text{GRF variation}}$$

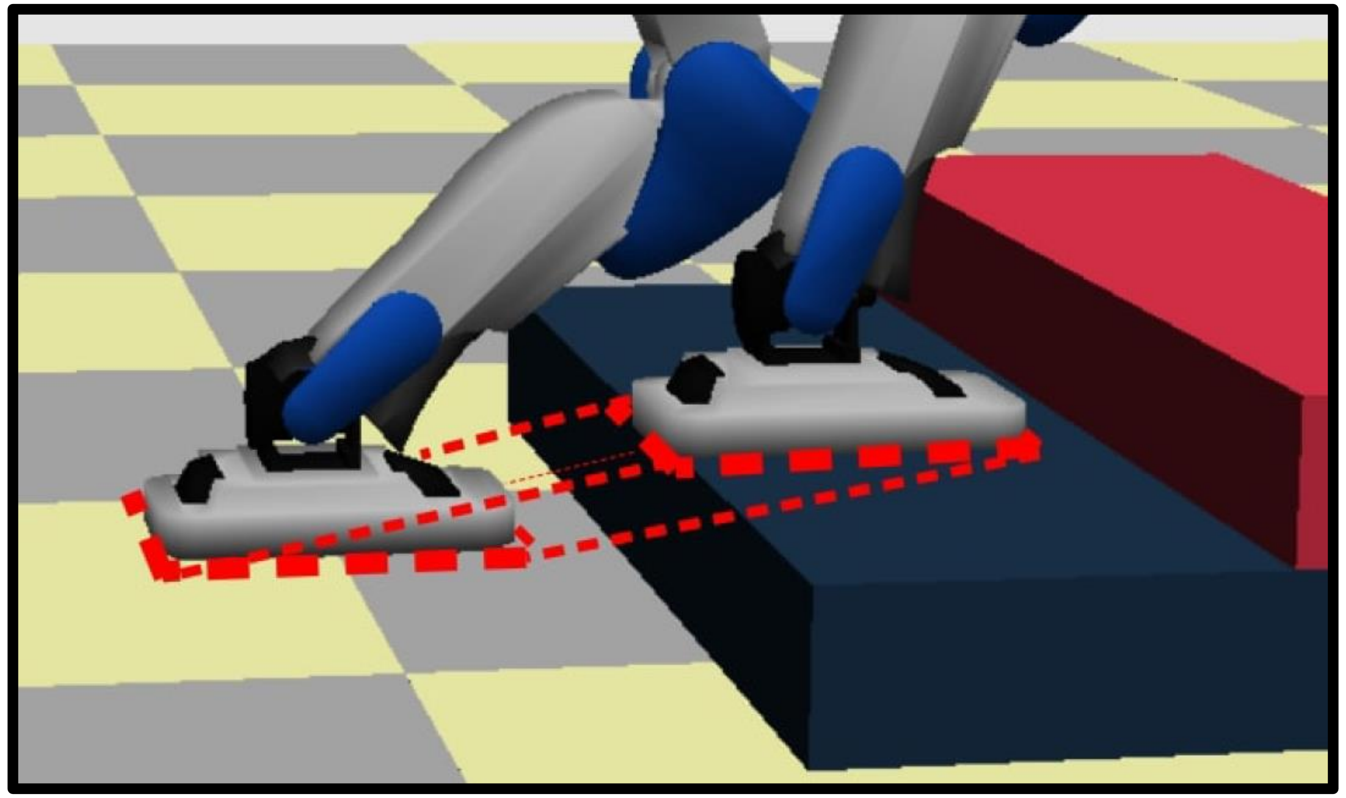
subject to:

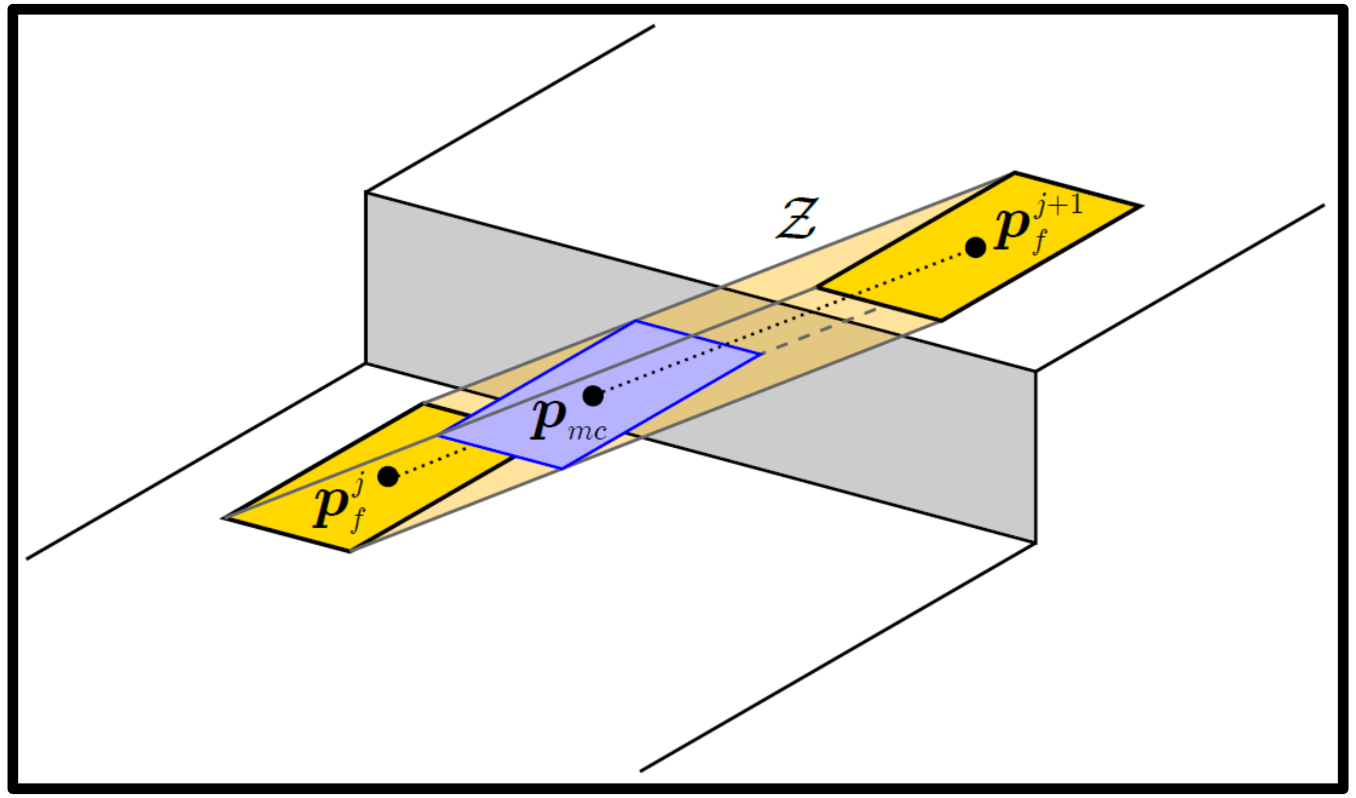
- minimum GRF constraint during support phases
- zero GRF constraint during flight phases

- the terms in the cost function perform the following tasks:
  - tracking the **reference** CoM height
  - penalizes **sudden** height variations
  - reduce GRF variation of consecutive samples, to generate a **smoother** profile

ZMP constraint

- we consider the convex hull of the active contact surfaces as admissible ZMP region, which in a world of stairs becomes an **oblique prism**





- to avoid nonlinearities, we employ a **moving constraint**: the ZMP must belong to a fixed-shape region (the footprint) which slides between consecutive footsteps during double support, and coincides with a footstep during single support

stability condition

- we use a **modified prediction model** which is identical to the VH-IP up to the end of the control horizon and becomes time-invariant after that

$$\ddot{x}_c = \begin{cases} \lambda^{k+i}(x_c - x_z) & \text{for} \quad t \in [t_{k+i}, t_{k+i+1}), \quad i = 0, \ldots, C-1 \\ \lambda_{\text{LIP}}(x_c - x_z) & \text{for} \quad t \geq t_{k+C} \end{cases}$$

- for this system we can write a stability condition at the **end of the control horizon**

$$x_u^{k+C} = \sqrt{\lambda_{\text{LIP}}} \int_{t_{k+C}}^{\infty} e^{-\sqrt{\lambda_{\text{LIP}}}(\tau - t_{k+C})} x_z(\tau) d\tau$$

- this condition can be expressed in terms of the current state and the decision variables, giving a stability constraint for the horizontal optimization problems QP-x and QP-y

stability constraint

- the **stability constraint** is expressed as

$$G \sum_{i=0}^{C-1} \prod_{j=i}^{C-1} \Phi_{k+j} B_{k+i} x_z^{k+i} = \sqrt{\lambda_{\text{LIP}}} \int_{t_{k+C}}^{\infty} e^{-\sqrt{\lambda_{\text{LIP}}}(\tau - t_{k+C})} \tilde{x}_z(\tau) d\tau - G \prod_{i=0}^{C-1} \Phi_{k+i} \begin{pmatrix} x_c^k \\ \dot{x}_c^k \end{pmatrix}$$

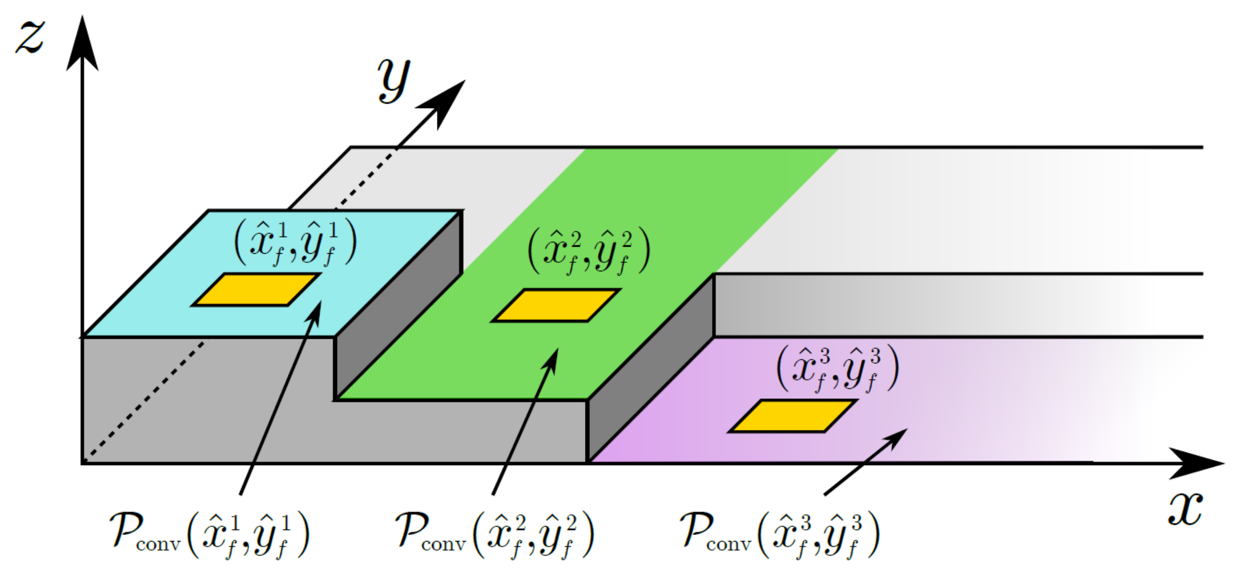
where:

- $\tilde{x}_z(t)$ is an **anticipative tail**, i.e., a ZMP trajectory after the end of the control horizon, conjectured on the basis of the available preview information from the footstep plan

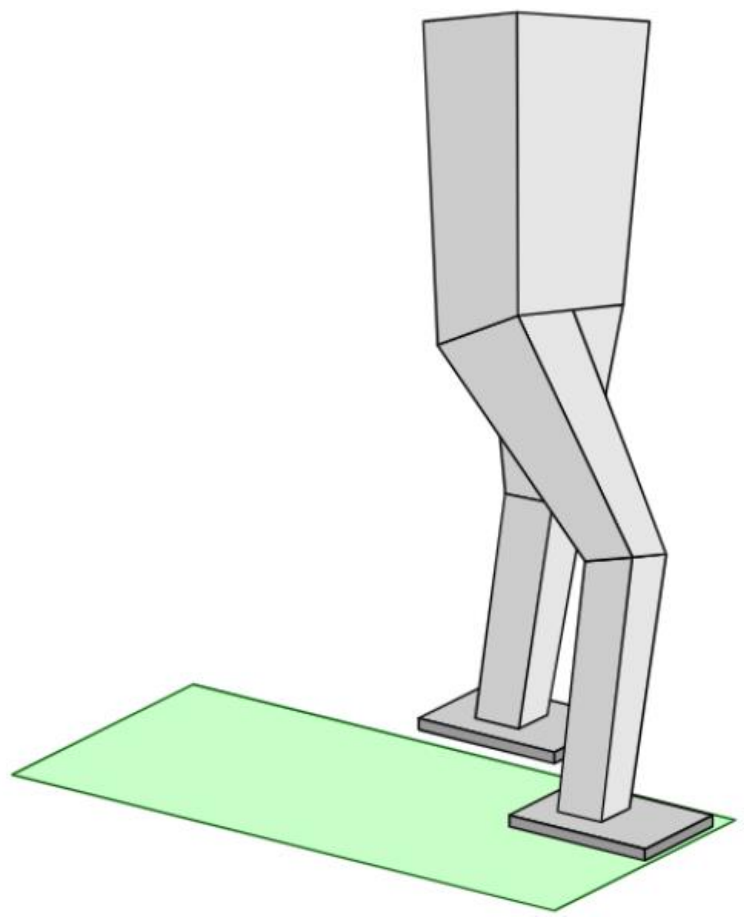
- the state transition matrices Φ and input matrices B of the time varying system are given by the appropriate expressions according to the alternation of contact/flight phases

constraints on the foot placement

- **ground patch constraint**: does not allow footstep adaptations that would move the footstep to a ground patch located at a different height



- **kinematic constraint**: ensures that footsteps are placed so as to be kinematically realizable by the robot



- **swing foot constraint**: prevents the leg joint speed from exceeding a feasible range

horizontal dynamics QP

- for constant footstep orientations, the second stage QP-xy can be split into QP-x and QP-y

$$\min_{X_z^k, X_f^k} \underbrace{\|X_z - X_{mc}^k\|^2}_{\text{ZMP tracking}} + \alpha_x \underbrace{\|\Delta X_z\|^2}_{\text{ZMP variation}} + \beta_x \underbrace{\|X_f^k - \hat{X}_f^k\|^2}_{\text{footstep tracking}}$$

subject to:

- ZMP constraints
- ground patch constraints
- kinematic constraints
- swing foot constraint
- stability constraint

- the terms in the cost function perform the following tasks:
  - keeping the ZMP as much as possible close to the **center** of the constraint region
  - minimize ZMP variations to increase **smoothness** of the trajectory
  - realizing as close as possible the **candidate footstep** sequence

constraint activation

- constraints are activated or deactivated depending on the **gait type** (walking/running) and the specific **phase** (single/double support, support/flight)

velocity driven footstep planner

- we start from triplet of **cruise parameters** for the velocity, step length and timing such that

$$\bar{v} = \bar{L}/\bar{T}$$

- a change in velocity should be reflected in a change of both the spacing between the footsteps and their timing

$$v = \bar{v} + \Delta v = \frac{\bar{L} + \Delta L}{\bar{T} - \Delta T} \quad \xrightarrow{\Delta L = \alpha \Delta T} \quad T = \bar{T}\,\frac{\alpha + \bar{v}}{\alpha + v}$$

- we can derive an expression for the step duration realizing the velocity v, which we can then divide into phases (single/double or support/flight)

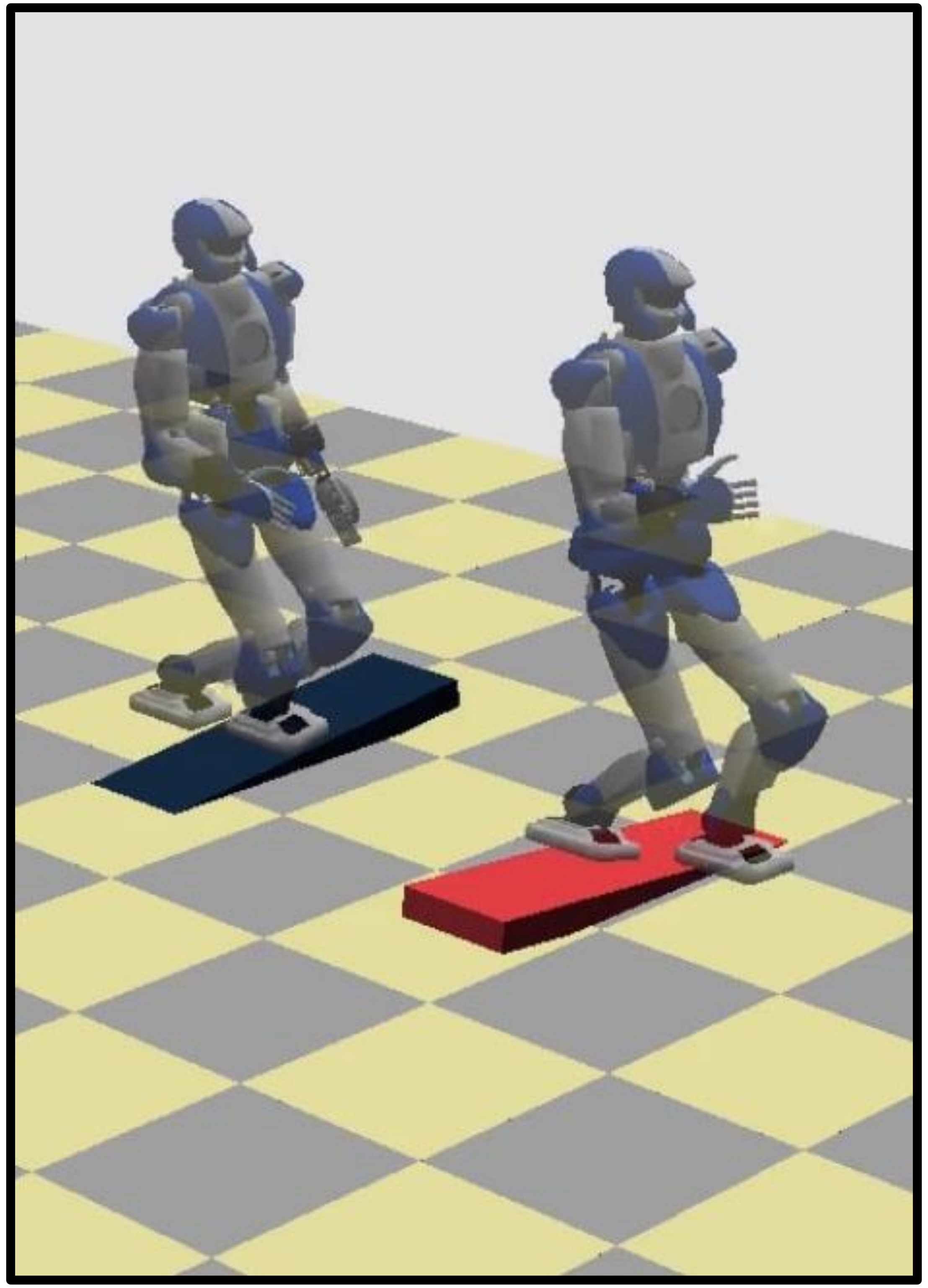
- we identify step length threshold $L_{\max}$ and command the scheme to generate a **running** step when $L > L_{\max}$, and a **walking** step otherwise

running over tilted surfaces

- the method can be easily extended to the case of running over **tilted** surfaces

- both walking and running are possible, but we only considered the case of running as it simplifies the problem as there are no double support phases

- we express CoM and ZMP in a frame that has the same orientation of the tilted patch and consider horizontal gravity terms as **known disturbances**



results