On the Implementation of Model Predictive Control for On-line Walking Pattern Generation

Dimitar Dimitrov[1] Pierre-Brice Wieber[2]
Hans Joachim Ferreau[3] Moritz Diehl[4]

[1]Örebro University, Sweden
[2]INRIA Rhône-Alpes, France
[3]K.U. Leuven, Belgium
[4]Univ. of Heidelberg, Germany

May 23, 2008

The aim

To generate a **stable** walking pattern for a humanoid robot **on-line**.

“stable walking” ?

ZMP within the convex hull formed by the foot/feet in contact with the ground.

Being able to generate a motion profile **on-line** is important, because:

- There is no guarantee that a motion profile generated off-line can be executed
- We might not know the final point

The problems

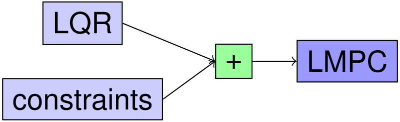
- Need to utilize a nonlinear optimization algorithm if the entire dynamic model of the system is considered
- In general it is not possible to model precisely the contact between the feet and the ground
- External disturbances can impose strong constraints on the system dynamics

“solutions”

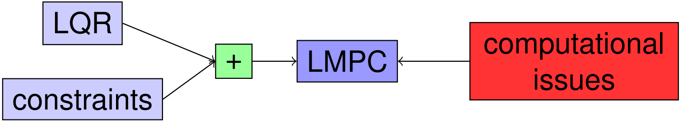
- Use a simplified model (linearize) in combination with a preview control scheme
- Rely on feedback
- Explicitly consider the constraints in the planning

- **Simplified model (3D-LIP)**
  (Kajita et al. “A realtime pattern generator for biped walking,” ICRA 2002)
- **Preview control scheme (infinite horizon LQR)**
  (Kajita et al. “Biped walking pattern generation by using preview control of zero-moment point,” ICRA 2003)
- **Explicitly consider constraints (LMPC)**
  (P.-B. Wieber “Trajectory free linear model predictive control for stable walking in the presence of strong perturbations,” ICHR 2006)

- **Simplified model (3D-LIP)**
  (Kajita et al. "A realtime pattern generator for biped walking," ICRA 2002)
- **Preview control scheme (infinite horizon LQR)**
  (Kajita et al. "Biped walking pattern generation by using preview control of zero-moment point," ICRA 2003)
- **Explicitly consider constraints (LMPC)**
  (P.-B. Wieber "Trajectory free linear model predictive control for stable walking in the presence of strong perturbations," ICHR 2006)

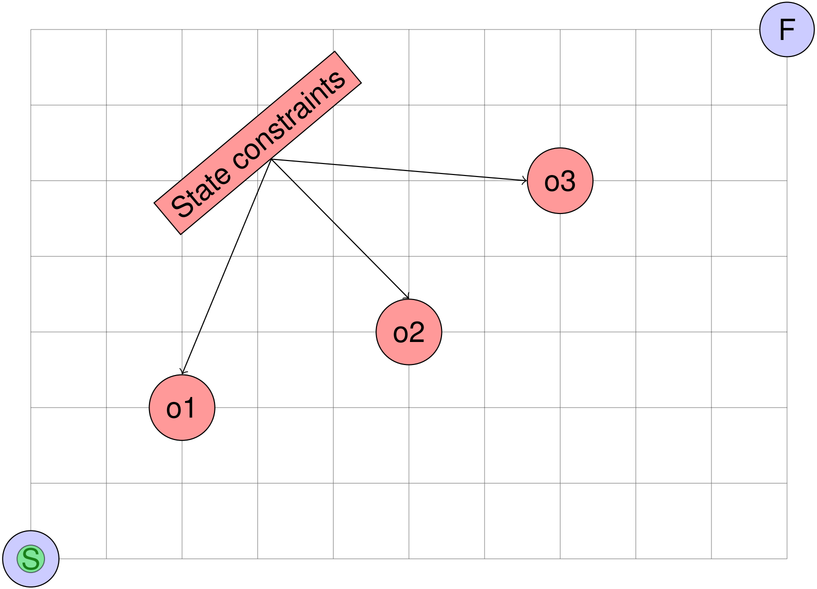


- **Simplified model (3D-LIP)**
  (Kajita et al. "A realtime pattern generator for biped walking," ICRA 2002)
- **Preview control scheme (infinite horizon LQR)**
  (Kajita et al. "Biped walking pattern generation by using preview control of zero-moment point," ICRA 2003)
- **Explicitly consider constraints (LMPC)**
  (P.-B. Wieber "Trajectory free linear model predictive control for stable walking in the presence of strong perturbations," ICHR 2006)

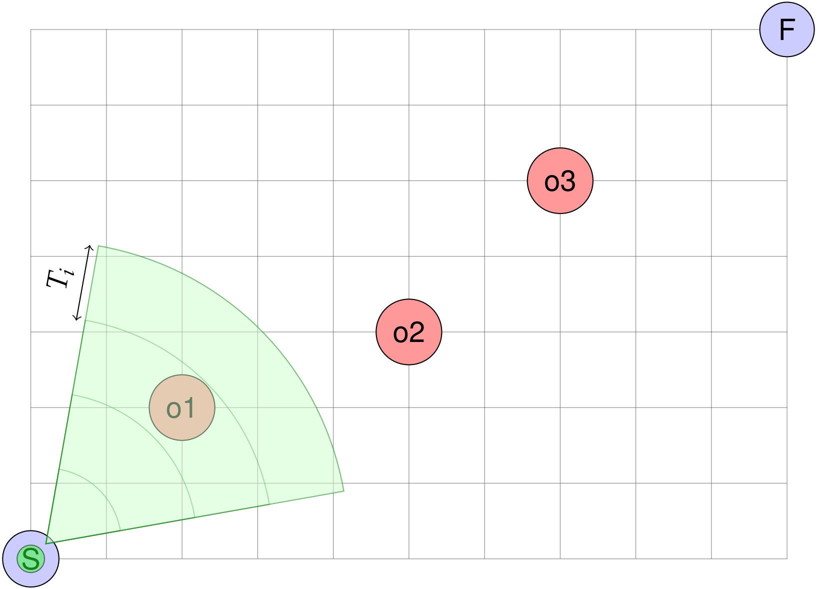


MPC is a general control scheme, that can deal with constraint dynamical systems with potential ability to react to unexpected situations. Its application is done through solving on-line a sequence of optimal control problems. The successful real time implementation of MPC with regard to complex robotic systems is sensitive to:

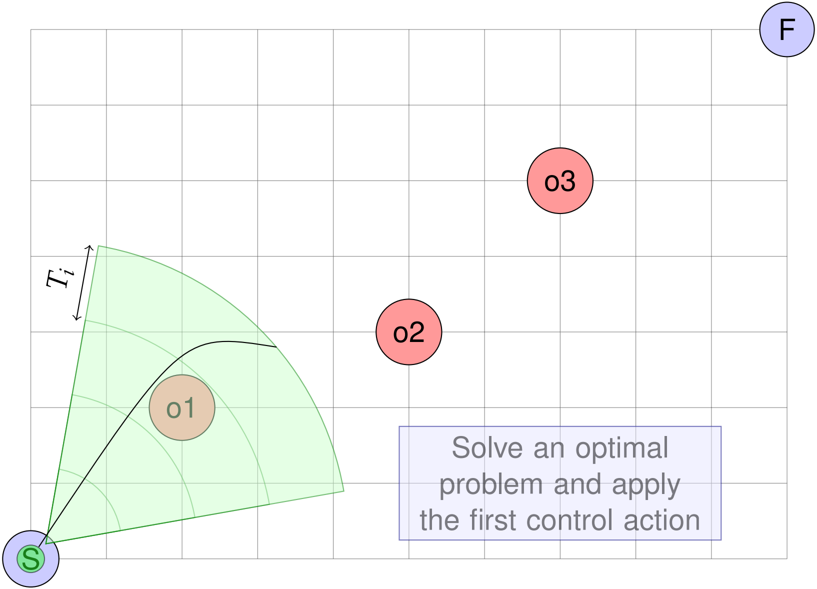
- The model to be used in order to describe the process of interest.
- The optimal control problem that should be considered and solved on-line, in order to achieve a desired system motion.


F
State constraints
o3
o2
o1
S

MPC

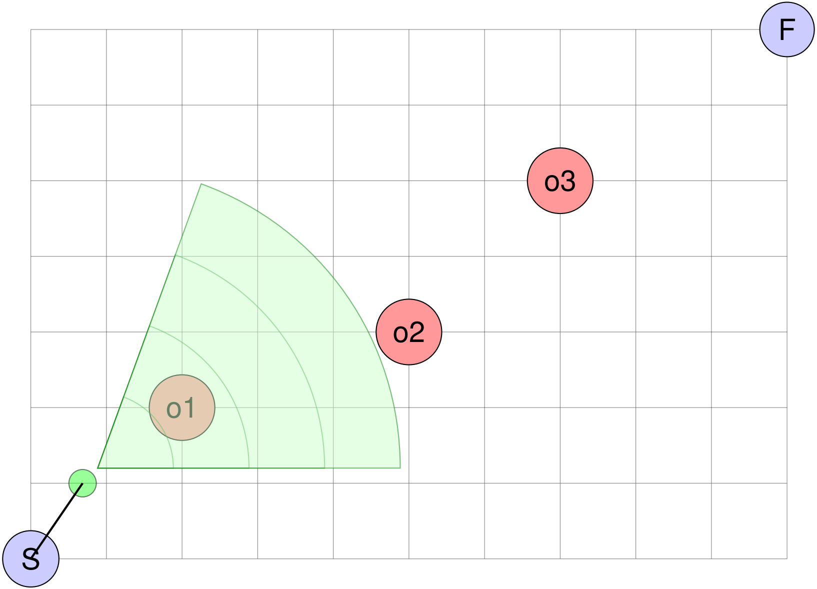


MPC

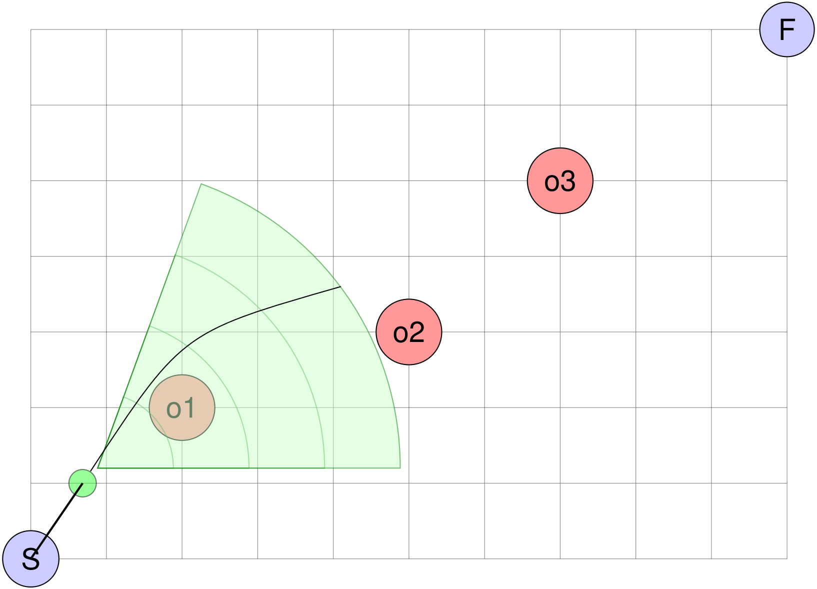


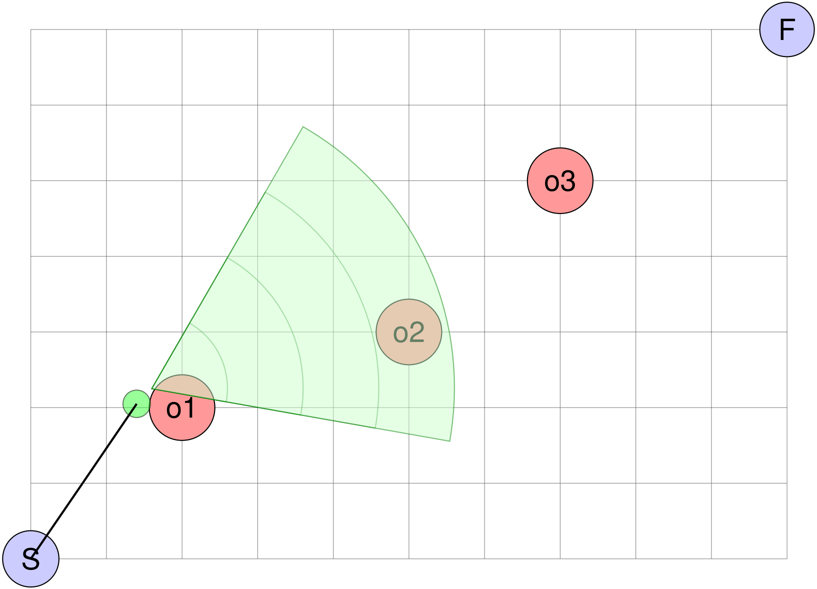
Solve an optimal problem and apply the first control action

MPC

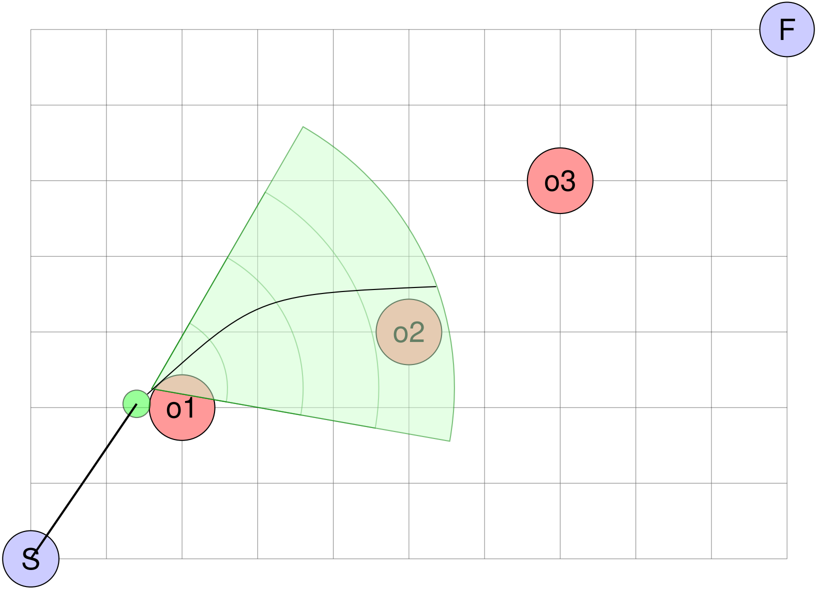


MPC

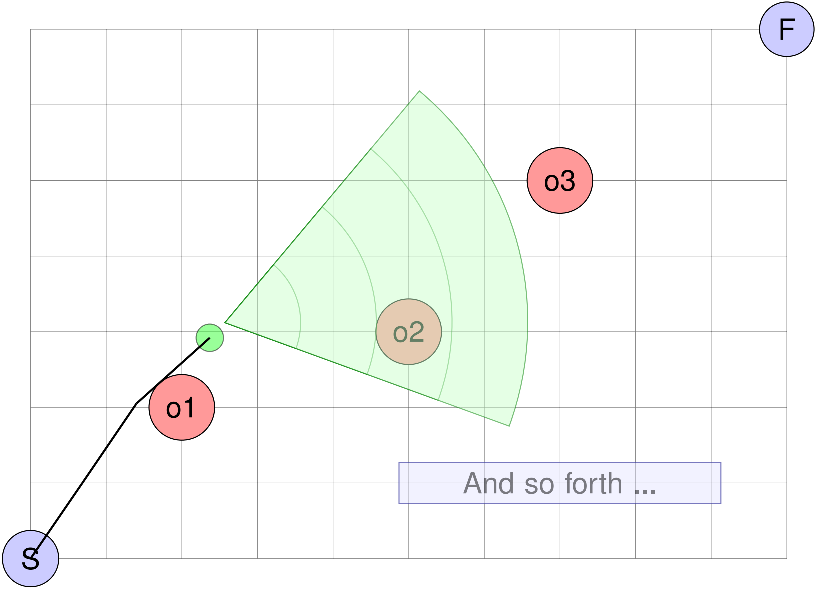



F
o3
o2
o1
S

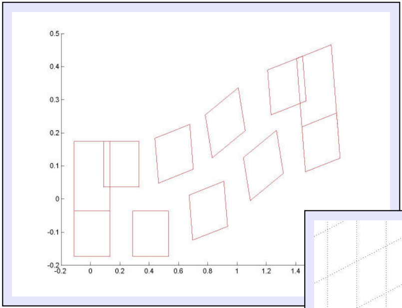
MPC



MPC

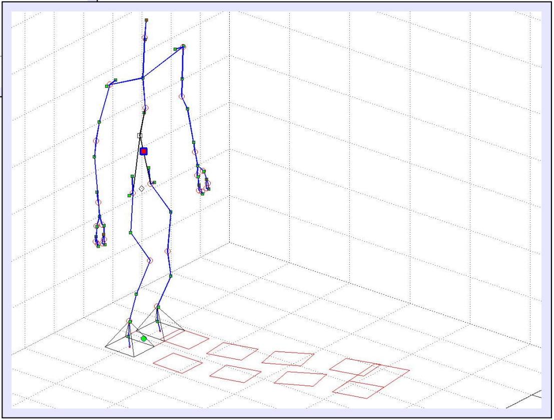


What is the relation to Humanoid walking?

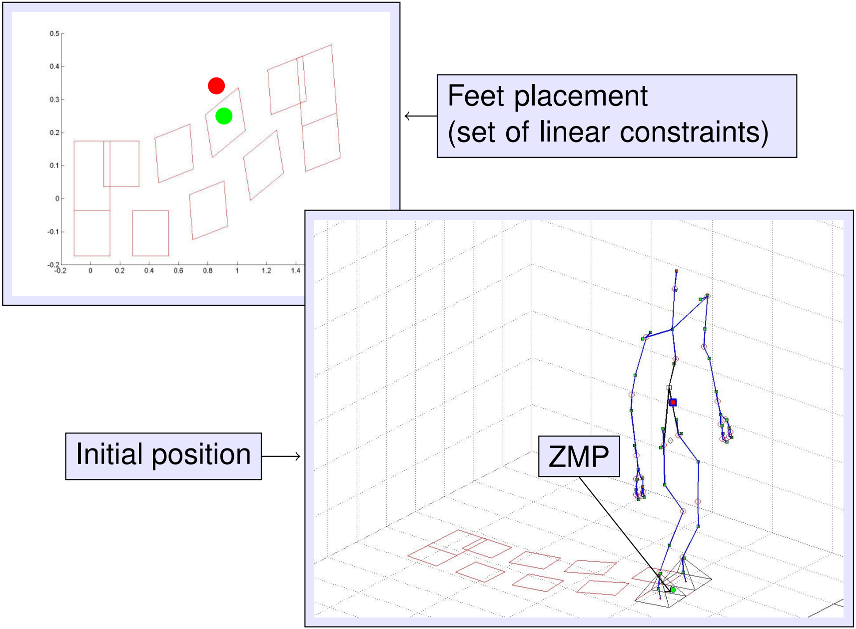


Feet placement
(set of linear constraints)

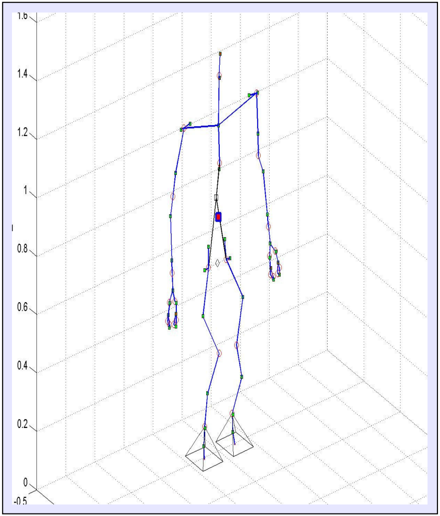
Initial position



What is the relation to Humanoid walking?



The setting



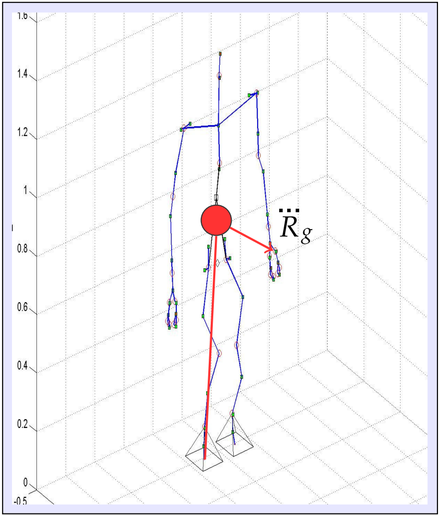
The model

The dynamics of each leg is approximated with a model of a 3D-LIP

The optimal problem

Minimize the jerk of the Center of Mass, while imposing constraints for the ZMP of the system

The setting



The model

The dynamics of each leg is approximated with a model of a 3D-LIP

The optimal problem

Minimize the jerk of the Center of Mass, while imposing constraints for the ZMP of the system

It is well-known that in the presence of linear constraints on the input and output, each "step" of a **LMPC** problem can be set up as a quadratic program (**QP**).

We define the state of the model at times t_k as:

$$\hat{x}_k = \left[\begin{array}{ccc} x(t_k) & \dot{x}(t_k) & \ddot{x}(t_k) \end{array} \right]^T$$

The dynamical system

$$\begin{array}{rcl} \hat{\boldsymbol{x}}_{k+1} & = & \boldsymbol{A}_k \hat{\boldsymbol{x}}_k + \boldsymbol{B}_k \dddot{\boldsymbol{x}}_k \\ z^x_{k+1} & = & \boldsymbol{C} \hat{\boldsymbol{x}}_{k+1} \end{array}$$

where C is a constant vector.

Background

Setting the optimization problem

$$\boxed{\hat{x}_{k+1} = A_k \hat{x}_k + B_k \dddot{x}_k}$$

$$\downarrow$$

$$\boxed{\hat{x}_{k+2} = A_k (A_k \hat{x}_k + B_k \dddot{x}_k) + B_{k+1} \dddot{x}_{k+1}}$$

$$\downarrow$$

$$\boxed{\hat{x}_{k+3} = A_k (A_k (A_k \hat{x}_k + B_k \dddot{x}_k) + B_{k+1} \dddot{x}_{k+1}) + B_{k+2} \dddot{x}_{k+2}}$$

$$\downarrow$$

$$\boxed{\cdots}$$

Multiply through with C we obtain the relation

$$Z^x = P_s \, \hat{x}_k + P_u \, U^x$$

$$Z^x = \begin{bmatrix} z^x_{k+1} \\ \vdots \\ z^x_{k+N} \end{bmatrix} \qquad U^x = \begin{bmatrix} \dddot{x}_k \\ \vdots \\ \dddot{x}_{k+N-1} \end{bmatrix}$$

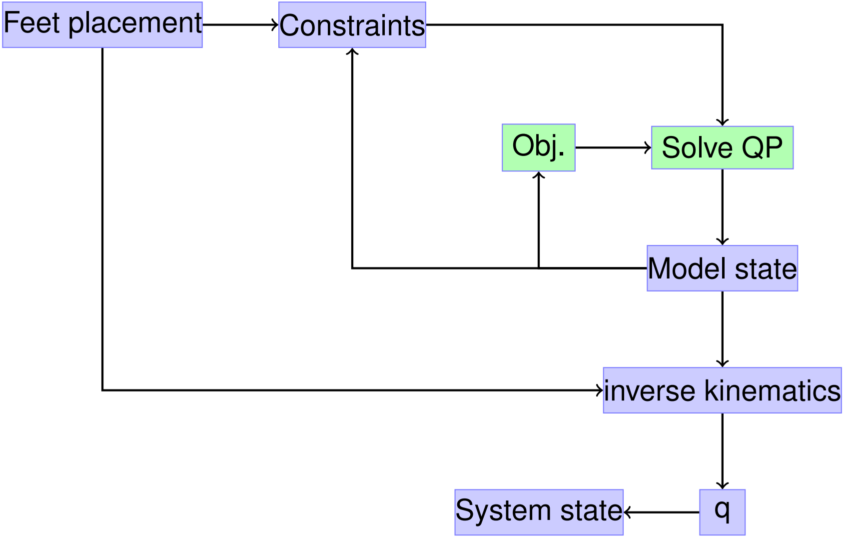
Objective function

$$\min_{\boldsymbol{U}} \frac{1}{2}(\boldsymbol{U}^2 + (...)^2)$$

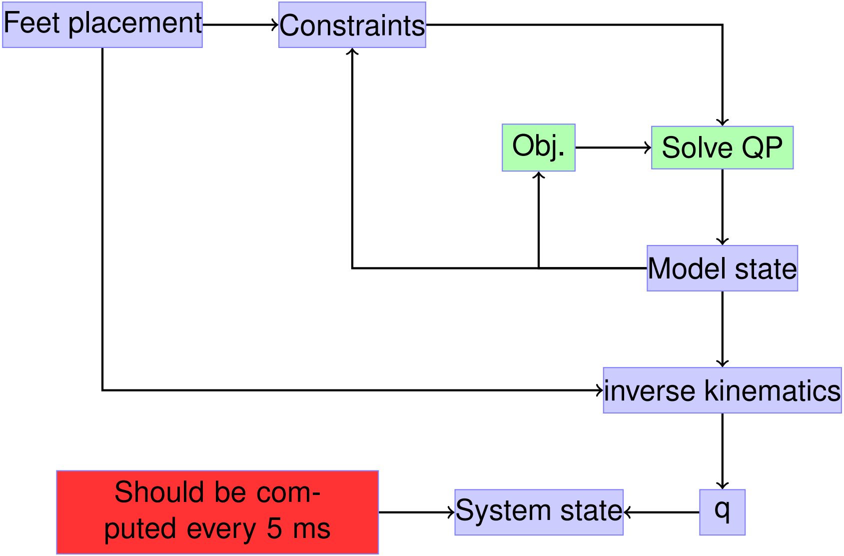
Standard notation

$$\min_{\boldsymbol{U}} \frac{1}{2}\boldsymbol{U}^T \boldsymbol{H} \boldsymbol{U} + \boldsymbol{U}^T \boldsymbol{g}$$

with $\boldsymbol{H} \in R^{N \times N}$ and $\boldsymbol{g} \in R^{N \times 1}$ being the Hessian matrix and gradient vector.



Feet placement
Constraints
Obj.
Solve QP
Model state
inverse kinematics
System state
q


Feet placement
Constraints
Obj.
Solve QP
Model state
inverse kinematics
q
System state
Should be computed every 5 ms

Contributions

Main contributions of the paper

- Forming a reliable initial guess for the optimization solver.
- A way to setup the optimal problem in a suitable form for on-line computation.

Size of the optimal problem

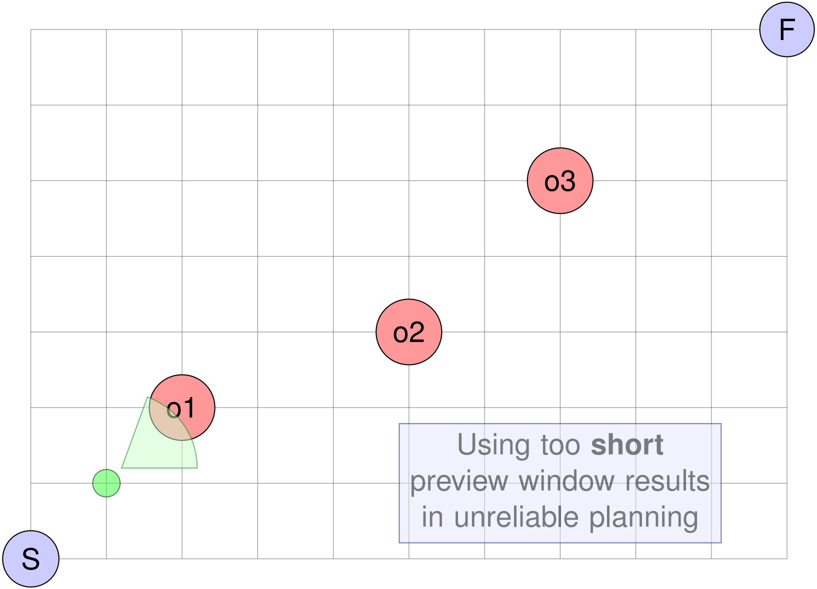
The size of the optimal problem is determined by:

- The length of the preview window (LPW).
- The sampling time (ST).

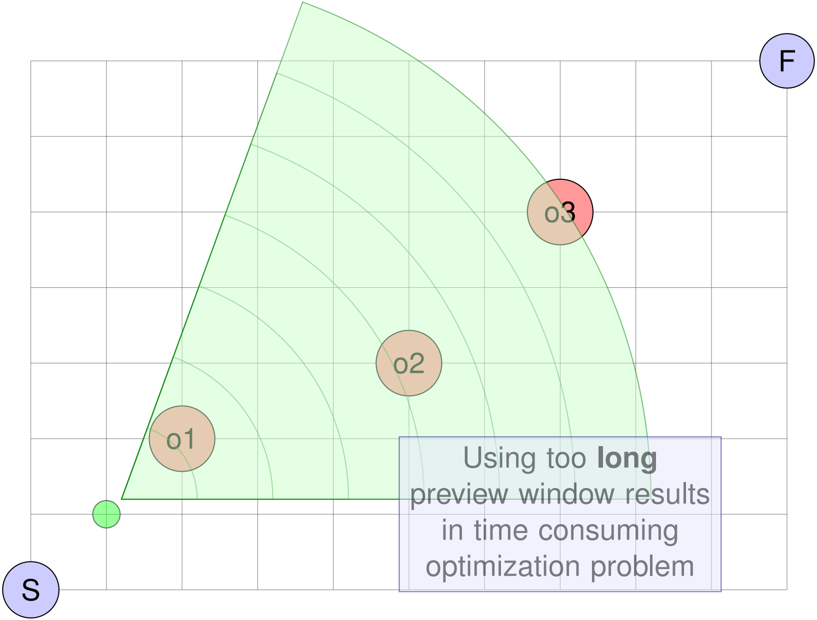
First consider a sequence of constant ST:

2ms	2ms	2ms	2ms	···	2ms	2ms	2ms	2ms

Short preview window



Long preview window



Using too **long** preview window results in time consuming optimization problem

In the case of the HRP2 robot for stability and robustness of the control algorithm one needs to use LPW = 1.5 sec.

Size of problem

sampling intervals = 75

state variables = 150

ineq. constraints = appr. 302

active constraints = appr. 6

Solver type

QP solvers based on active set strategy are well suited for the fast solution of such problem

Active constraints

Inequality constraints that hold as equalities.

Computation time

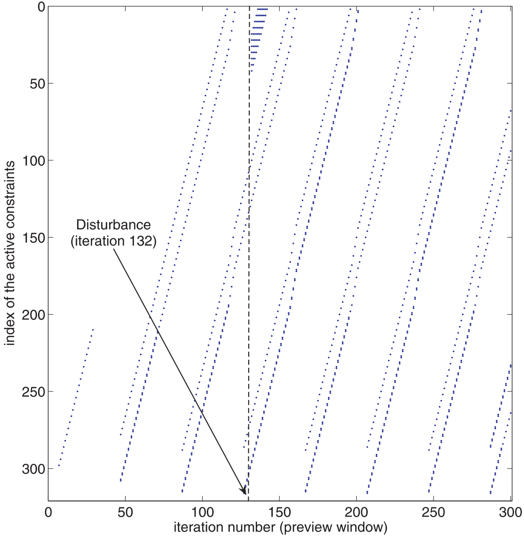
An active solver solves a QP by identifying the active constraints at optimality (by taking an iterative approach). For the case in the previous slide this results in computation time of about 6 ms with a “well implemented” dual QP solver.

Hence, considering each QP as a separate problem results in high computation time.

Contribution 1 (hot-start)

Form a reliable initial guess for the optimization solver at time t_k based on the solution of the QP at t_{k-1} and exploiting some of the **particular characteristics** of a humanoid walking process.

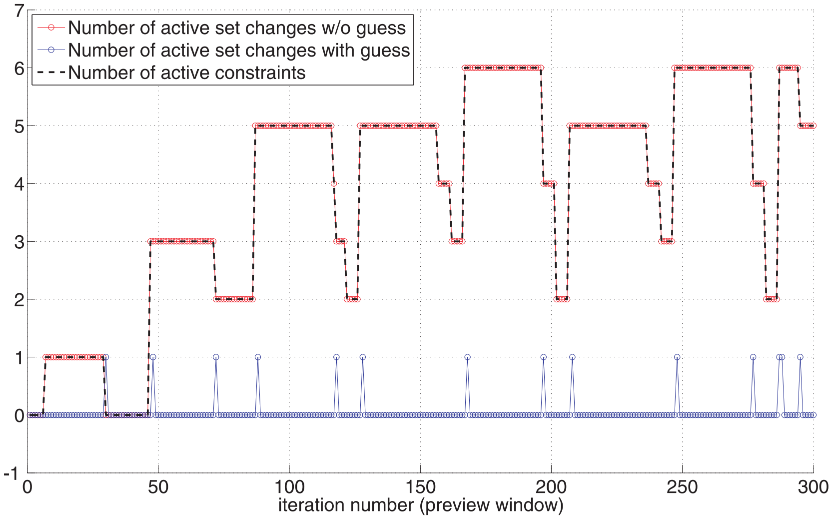
Why is forming a guess possible?



We developed a routine that makes a "perfect" guesses in average 94% of the time.

When the guess is not "perfect" usually only one active constraint is wrong.

Result 1 (using hot-start)



Setting up the optimization problem

Advantages of fixed sampling time

- Results in a constant Hessian matrix which can be formed and factorized off-line.

Disadvantages of fixed sampling time

- Using small sampling step results in large optimization problems.
- Puts limitations on the choice of Single- and Double-support time.

How can we use the advantages and avoid the dissadvantages of the fixed sampling time?

Problems when using large constant step size

Requirement

The time of transition between Single-support and Double-support phases is of great importance to the stable walking, hence solving a QP at the boundary is important.

S	S	S	S	S	S	D	S	S	S	S	S	S	S	D	S	S	S

Problems when using large constant step size

Requirement

The time of transition between Single-support and Double-support phases is of great importance to the stable walking, hence solving a QP at the boundary is important.

S	S	S	S	S	D	S	S	S	S	S	S	S	D	S	S	S	S

Problems when using large constant step size

Requirement

The time of transition between Single-support and Double-support phases is of great importance to the stable walking, hence solving a QP at the boundary is important.

S	S	S	S	D	S	S	S	S	S	S	S	D	S	S	S	S	S

Problems when using large constant step size

Requirement

The time of transition between Single-support and Double-support phases is of great importance to the stable walking, hence solving a QP at the boundary is important.

S	S	S	D	S	S	S	S	S	S	S	D	S	S	S	S	S	S

Problems when using large constant step size

Requirement

The time of transition between Single-support and Double-support phases is of great importance to the stable walking, hence solving a QP at the boundary is important.

S	S	D	S	S	S	S	S	S	S	D	S	S	S	S	S	S	S

Problems when using large constant step size

Requirement

The time of transition between Single-support and Double-support phases is of great importance to the stable walking, hence solving a QP at the boundary is important.

S	D	S	S	S	S	S	S	S	D	S	S	S	S	S	S	S	D

Problems when using large constant step size

Requirement

The time of transition between Single-support and Double-support phases is of great importance to the stable walking, hence solving a QP at the boundary is important.

D	S	S	S	S	S	S	S	D	S	S	S	S	S	S	S	D	S

Problems when using large constant step size

Requirement

The time of transition between Single-support and Double-support phases is of great importance to the stable walking, hence solving a QP at the boundary is important.

S	S	S	S	S	S	S	D	S	S	S	S	S	S	S	D	S	S

Problems when using large constant step size

Requirement

The time of transition between Single-support and Double-support phases is of great importance to the stable walking, hence solving a QP at the boundary is important.

S	S	S	S	S	S	S	D	S	S	S	S	S	S	S	D	S	S

Problems when using large constant step size

Requirement

The time of transition between Single-support and Double-support phases is of great importance to the stable walking, hence solving a QP at the boundary is important.

S	S	S	S	S	S	S	D	S	S	S	S	S	S	S	D	S	S

Problems when using large constant step size

Requirement

The time of transition between Single-support and Double-support phases is of great importance to the stable walking, hence solving a QP at the boundary is important.

S	S	S	S	S	S	S	D	S	S	S	S	S	S	S	D	S	S

Problems when using large constant step size

Requirement

The time of transition between Single-support and Double-support phases is of great importance to the stable walking, hence solving a QP at the boundary is important.

S	S	S	S	S	S	S	D	S	S	S	S	S	S	S	D	S	S

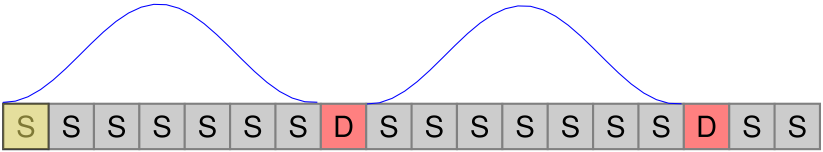
Problems when using large constant step size

Requirement

The time of transition between Single-support and Double-support phases is of great importance to the stable walking, hence solving a QP at the boundary is important.

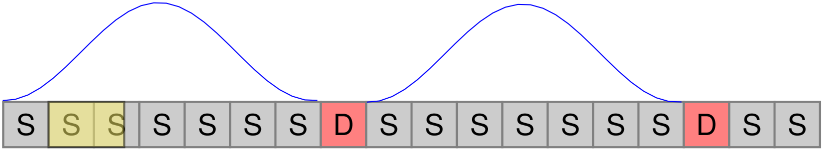
S	S	S	S	S	S	S	D	S	S	S	S	S	S	S	D	S	S

Using variable step size



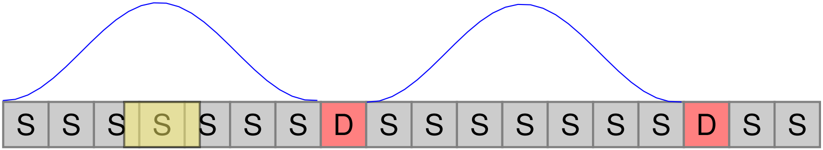
With variable sampling time, we can focus on the regions of interest while reducing the size of the optimization problem.

Using variable step size



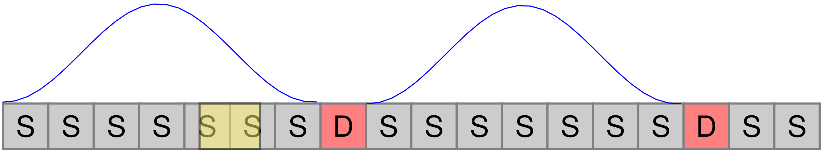
With variable sampling time, we can focus on the regions of interest while reducing the size of the optimization problem.

Using variable step size



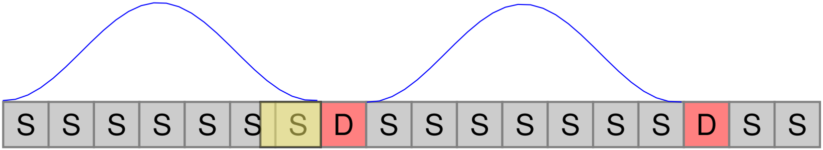
With variable sampling time, we can focus on the regions of interest while reducing the size of the optimization problem.

Using variable step size



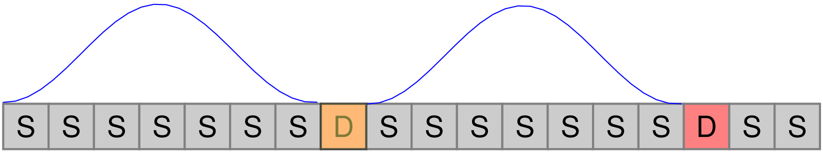
With variable sampling time, we can focus on the regions of interest while reducing the size of the optimization problem.

Using variable step size



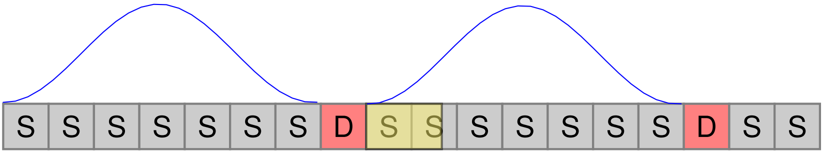
With variable sampling time, we can focus on the regions of interest while reducing the size of the optimization problem.

Using variable step size



With variable sampling time, we can focus on the regions of interest while reducing the size of the optimization problem.

Using variable step size



With variable sampling time, we can focus on the regions of interest while reducing the size of the optimization problem.

One cycle

S	S	S	S	S	S	S	D	S	S	S	S	S	S	S	D	S	S

One cycle

S	S	S	S	S	S	S	D	S	S	S	S	S	S	S	D	S	S
S	S	S	S	S	S	D	S	S	S	S	S	S	S	D	S	S	S

One cycle

S	S	S	S	S	S	S	D	S	S	S	S	S	S	S	D	S	S
S	S	S	S	S	S	D	S	S	S	S	S	S	S	D	S	S	S
S	S	S	S	S	D	S	S	S	S	S	S	S	D	S	S	S	S

One cycle

S	S	S	S	S	S	S	D	S	S	S	S	S	S	S	D	S	S
S	S	S	S	S	S	D	S	S	S	S	S	S	S	D	S	S	S
S	S	S	S	S	D	S	S	S	S	S	S	S	D	S	S	S	S
S	S	S	S	D	S	S	S	S	S	S	S	D	S	S	S	S	S

One cycle

S	S	S	S	S	S	S	D	S	S	S	S	S	S	S	D	S	S
S	S	S	S	S	S	D	S	S	S	S	S	S	S	D	S	S	S
S	S	S	S	S	D	S	S	S	S	S	S	S	D	S	S	S	S
S	S	S	S	D	S	S	S	S	S	S	S	D	S	S	S	S	S
S	S	S	D	S	S	S	S	S	S	S	D	S	S	S	S	S	S

One cycle

S	S	S	S	S	S	S	D	S	S	S	S	S	S	S	D	S	S
S	S	S	S	S	S	D	S	S	S	S	S	S	S	D	S	S	S
S	S	S	S	S	D	S	S	S	S	S	S	S	D	S	S	S	S
S	S	S	S	D	S	S	S	S	S	S	S	D	S	S	S	S	S
S	S	S	D	S	S	S	S	S	S	S	D	S	S	S	S	S	S
S	S	D	S	S	S	S	S	S	S	D	S	S	S	S	S	S	S

One cycle

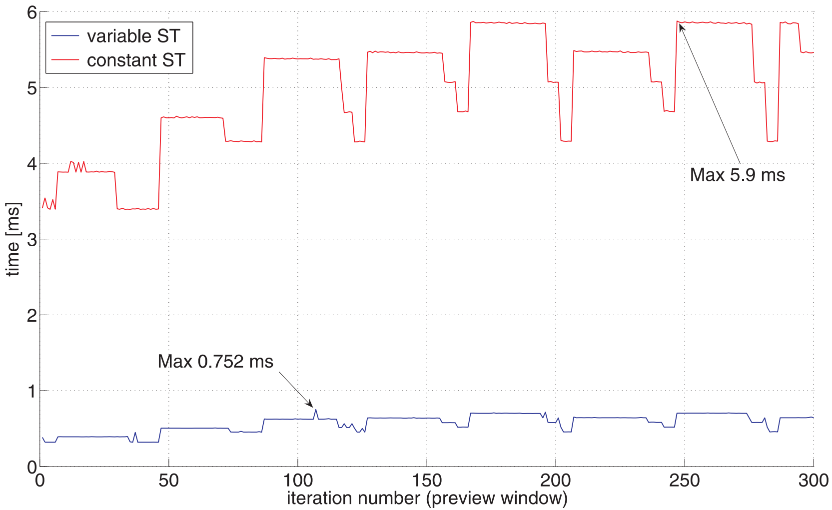
S	S	S	S	S	S	S	D	S	S	S	S	S	S	S	D	S	S
S	S	S	S	S	S	D	S	S	S	S	S	S	S	D	S	S	S
S	S	S	S	S	D	S	S	S	S	S	S	S	D	S	S	S	S
S	S	S	S	D	S	S	S	S	S	S	S	D	S	S	S	S	S
S	S	S	D	S	S	S	S	S	S	S	D	S	S	S	S	S	S
S	S	D	S	S	S	S	S	S	S	D	S	S	S	S	S	S	S
S	D	S	S	S	S	S	S	S	D	S	S	S	S	S	S	S	D

One cycle

S	S	S	S	S	S	S	D	S	S	S	S	S	S	S	D	S	S
S	S	S	S	S	S	D	S	S	S	S	S	S	S	D	S	S	S
S	S	S	S	S	D	S	S	S	S	S	S	S	D	S	S	S	S
S	S	S	S	D	S	S	S	S	S	S	S	D	S	S	S	S	S
S	S	S	D	S	S	S	S	S	S	S	D	S	S	S	S	S	S
S	S	D	S	S	S	S	S	S	S	D	S	S	S	S	S	S	S
S	D	S	S	S	S	S	S	S	D	S	S	S	S	S	S	S	D
D	S	S	S	S	S	S	S	D	S	S	S	S	S	S	S	D	S

One cycle

S	S	S	S	S	S	S	D	S	S	S	S	S	S	S	D	S	S
S	S	S	S	S	S	D	S	S	S	S	S	S	S	D	S	S	S
S	S	S	S	S	D	S	S	S	S	S	S	S	D	S	S	S	S
S	S	S	S	D	S	S	S	S	S	S	S	D	S	S	S	S	S
S	S	S	D	S	S	S	S	S	S	S	D	S	S	S	S	S	S
S	S	D	S	S	S	S	S	S	S	D	S	S	S	S	S	S	S
S	D	S	S	S	S	S	S	S	D	S	S	S	S	S	S	S	D
D	S	S	S	S	S	S	S	D	S	S	S	S	S	S	S	D	S
S	S	S	S	S	S	S	D	S	S	S	S	S	S	S	D	S	S



QP computation time for the cases with constant (150 states, red line) and variable sampling times (64 states, blue line).

Conclusions

This article addresses the on-line implementation of MPC for walking pattern generation.

I presented

- Development of a reliable guess for the active constraints at optimality.
- Utilization of variable sampling time for decreasing the size of the optimal problem.

Additional issues addressed in the paper

- Develop a feasible initial point (with respect to the constraints). Useful for **primal QP solvers**.
- Addresses issues related to difference between preview sampling time and control sampling time.

Thank you