`Humoto`: manual

Contributors
Alexander Sherikov, 2014 – 2017
Jan Michalczyk, 2016 – 2017
Joven Agravante, 2014 – 2015

Generated on September 29, 2017

Contents

List of Acronyms

API	*Application Programming Interface*
CoM	*Center of Mass*
CoP	*Center of Pressure*
MPC	*Model Predictive Control*
QP	*Quadratic Program*
WPG	*Walking Pattern Generator*

Part I

Core

Chapter 1

Introduction

1.1 What is Humoto

`Humoto` – is a software framework for manipulation of linear least-squares problems with (in)equality constraints. It supports both weighting and lexicographic prioritization and can be characterized as a tool for goal programming [1]. However, the development was driven by works in other fields – robotics, control, and numerical optimization, *e.g.*, [3, 5, 4]; for this reason our terminology and interpretations are different.

The core functionalities of `Humoto` are formulation of least-squares problems and their resolution using various third-party software. Both of these operations are performed through unified *Application Programming Interface* (API). Moreover, due to our interest in robotic applications, the framework facilitates formulation and implementation of optimization problems for control of robots, in particular, *Model Predictive Control* (MPC) problems. For the same reason, we pay special attention to computational performance in order to be able to employ the framework in real-time scenarios.

In addition to the core components, the distribution of `Humoto` includes several modules – implementations of specific controllers. The modules serve as examples of using the framework, but can also be used in accordance with their primary purpose. You can learn more about the provided modules in part II and part III.

1.2 Outline

The present document introduces basic concepts behind the `Humoto` and contains various mathematical derivations which are used in the framework and the modules. However, this document cannot serve as a comprehensive guide – you have to refer to `Doxygen` documentation in order to learn how to use `Humoto` in your applications or learn more about implementation details.

1.3 Notation

Software names

Names of programs and software libraries, names of constants, variables and functions that are used in programs are typed in a monospaced font: `Eigen`.

Scalars, vectors, matrices

- Vectors and matrices are denoted by letters in a bold font: $\boldsymbol{v}$, $\boldsymbol{M}$, $\boldsymbol{\mathcal{A}}$.
- Scalars are denoted using the standard italic or calligraphic font: $N, n, \mathcal{K}$.

- $(\cdot)^\top$ – transpose of a matrix or a vector.
- $(\cdot)^\times$ – a skew-symmetric matrix used for representation of a cross product of two three dimensional vectors as a product of a matrix and a vector:

$$\boldsymbol{v} = \begin{bmatrix} v^x \\ v^y \\ v^z \end{bmatrix}, \quad \boldsymbol{v}^\times = \begin{bmatrix} 0 & -v^z & v^y \\ v^z & 0 & -v^x \\ -v^y & v^x & 0 \end{bmatrix}. \tag{1.1}$$

- Block diagonal matrices:

$$\operatorname*{diag}_2(\boldsymbol{M}) = \begin{bmatrix} \boldsymbol{M} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M} \end{bmatrix}, \quad \operatorname*{diag}_{k=1\ldots2}(\boldsymbol{M}_k) = \begin{bmatrix} \boldsymbol{M}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}_2 \end{bmatrix}, \\ \operatorname{diag}(\boldsymbol{M}, \boldsymbol{R}) = \begin{bmatrix} \boldsymbol{M} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R} \end{bmatrix}. \tag{1.2}$$

- Stacked vectors and matrices:

$$\boldsymbol{v} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n) = \begin{bmatrix} \boldsymbol{v}_1 \\ \vdots \\ \boldsymbol{v}_n \end{bmatrix}, \quad \boldsymbol{M} = (\boldsymbol{M}_1, \ldots, \boldsymbol{M}_n) = \begin{bmatrix} \boldsymbol{M}_1 \\ \vdots \\ \boldsymbol{M}_n \end{bmatrix}. \tag{1.3}$$

- Inequalities between vectors $\boldsymbol{v} \geq \boldsymbol{r}$ are interpreted component-wise.

Special matrices and vectors

- $\boldsymbol{I}$ – an identity matrix. $\boldsymbol{I}_n$ – $n \times n$ identity matrix.
- $\boldsymbol{I}_{(\cdot)}$ – a selection matrix.
- $\boldsymbol{0}$ – a matrix of zeros. $\boldsymbol{0}_{n,m}$ – $n \times m$ matrix of zeros.

Reference frames

- Frames are denoted using a sans-serif font: A. All considered frames are orthonormal.
- ${}^{\mathsf{A}}\boldsymbol{v}$ – vector expressed in frame A.
- ${}^{\mathsf{A}}\boldsymbol{R}_{\mathsf{B}}$ – rotation matrix from from frame B to frame A. For example, 2-d rotation matrix is defined as

$${}^{\mathsf{A}}\boldsymbol{R}_{\mathsf{B}} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}, \tag{1.4}$$

  where θ is the rotation angle of frame B with respect to A.
- The global frame is implicit and is not denoted by any letter, *e.g.*, $\boldsymbol{R}_{\mathsf{B}}$ rotates from frame B to the global frame.

Sets

- The sets are denoted using a blackboard bold font: $\mathbb{A}$.
- $\mathbb{R}$ is the set of real numbers.
- $\mathbb{R}_{\geq 0}, \mathbb{R}_{>0}$ are the sets of non-negative and positive real numbers.
- $\mathbb{R}^n$ is the set of real-valued vectors.
- $\mathbb{R}^{n\times m}$ is the set of real-valued matrices.

Other

- Function names in mathematical expressions are written in the regular font: $\operatorname{func}(\boldsymbol{x}, \boldsymbol{y})$.
- $\|\cdot\|_2$ denotes the Euclidean norm.

Chapter 2

Optimization concepts

This chapter introduces basic concepts and terms which commonly used in the framework. The presentation is terse, so you may be interested in reading more in other sources [2, 3, 5, 4].

2.1 Tasks

We define a *task* as set of logically related constraints of the form

$$\underline{\boldsymbol{b}} \leq \boldsymbol{\mathcal{A}}\boldsymbol{\chi} \leq \bar{\boldsymbol{b}}, \tag{2.1}$$

where $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n \times m}$ is a given matrix; $\underline{\boldsymbol{b}} \in \mathbb{R}^n$ and $\bar{\boldsymbol{b}} \in \mathbb{R}^n$ are given vectors of lower and upper bounds, which may include infinitely large values; and $\boldsymbol{\chi} \in \mathbb{R}^m$ is a vector of decision variables. We consider only well defined tasks where $\underline{\boldsymbol{b}} \leq \bar{\boldsymbol{b}}$ is always true.

Usually, control or optimization problems are composed of multiple tasks, which may not be achieved exactly. In order to account for this we assume that there exist implicit vector of violations $\boldsymbol{v} \in \mathbb{R}^n$ such that expression

$$\underline{\boldsymbol{b}} \leq \boldsymbol{\mathcal{A}}\boldsymbol{\chi} - \boldsymbol{v} \leq \bar{\boldsymbol{b}} \tag{2.2}$$

is always exactly satisfied. A value in vector $\boldsymbol{v}$ is negative when the corresponding lower bound is violated, positive when the upper bound is violated, and zero when the constraint is satisfied exactly. Here we aim at satisfaction of a task in the least-squares sense, which is equivalent to solving the following *Quadratic Program* (QP)

$$\begin{aligned} \underset{\boldsymbol{\chi}, \boldsymbol{v}}{\text{minimize}} \quad & \|\boldsymbol{v}\|_2^2 \\ \text{subject to} \quad & \underline{\boldsymbol{b}} \leq \boldsymbol{\mathcal{A}}\boldsymbol{\chi} - \boldsymbol{v} \leq \bar{\boldsymbol{b}} \end{aligned} \tag{2.3}$$

for optimal $\boldsymbol{v}^\star$. Note that due to semidefinite nature of this problem $\boldsymbol{\chi}^\star$ is not necessarily unique.

2.2 Objectives: weighting of tasks

Consider two tasks

$$\underline{\boldsymbol{b}}_1 \leq \boldsymbol{\mathcal{A}}_1\boldsymbol{\chi} \leq \bar{\boldsymbol{b}}_1 \quad \text{and} \quad \underline{\boldsymbol{b}}_2 \leq \boldsymbol{\mathcal{A}}_2\boldsymbol{\chi} \leq \bar{\boldsymbol{b}}_2, \tag{2.4}$$

which are known to be in a conflict, *i.e.*, cannot be satisfied exactly with zero $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ simultaneously. In this case, we may still want to satisfy the tasks simultaneously as much as possible with a certain trade-off, which can be achieved with minimizaion of a weighted sum of the norms of violations:

$$\gamma_1\|\boldsymbol{v}_1\|_2 + \gamma_2\|\boldsymbol{v}_2\|_2, \tag{2.5}$$

where $\gamma_1 \in \mathbb{R}_{\geq 0}$ and $\gamma_2 \in \mathbb{R}_{\geq 0}$. For example, $\gamma_1 > \gamma_2$ gives higher priority to the first task.

We call a group of weighted tasks an *objective*; note, however, that the difference between a "task" and "objective" is purely terminological. Usually, the weights are ommited since the necessary effect can be achieved by scaling the task.

2.3 Hierarchies: prioritization of tasks (objectives)

In the previous section we considered case when we want to satisfy conflicting tasks simultaneously, but what if one of the tasks has infinitely higher priority than another? In this case we resort to *hierarchies* of tasks or objectives:

Hierarchy (2.1)

1: $\underline{\boldsymbol{b}}_1 \leq \boldsymbol{\mathcal{A}}_1 \boldsymbol{\chi} \leq \bar{\boldsymbol{b}}_1$
2: $\underline{\boldsymbol{b}}_2 \leq \boldsymbol{\mathcal{A}}_2 \boldsymbol{\chi} \leq \bar{\boldsymbol{b}}_2$
3: $\underline{\boldsymbol{b}}_3 \leq \boldsymbol{\mathcal{A}}_3 \boldsymbol{\chi} \leq \bar{\boldsymbol{b}}_3$
... ...

Here task (objective) i is infinitely more important than task $i+1$, *i.e.*, satisfaction of task $i+1$ must never come at a price of increasing $\|\boldsymbol{v}_i\|_2^2$.

A hierarchy can be solved using a sequence of QP or with the help of a specialized algorithm [3].

2.4 Relation between hierarchies and QPs

Note that a QP can be represented as a hierarchy with two levels

Hierarchy (2.2)

1: $\underline{\boldsymbol{b}}_1 \leq \boldsymbol{\mathcal{A}}_1 \boldsymbol{\chi} \leq \bar{\boldsymbol{b}}_1$
2: $\boldsymbol{\mathcal{A}}_2 \boldsymbol{\chi} = \boldsymbol{b}_2$

where objective on the second level is an equality. The only difference is that QP requires the first inequality objective to be feasible. Since this condition is satisfied in many applications, the framework allows to cast and solve a hierarchy of two levels as a single QP, even though this is not strictly correct. This behavior can be suppressed by changing parameters of the solvers.

2.5 Computational performance

2.5.1 Hot-starting

Many solvers for optimization problems support hot-starting – they accept additional data, which may help to reduce computation time. Currently the framework allows hot-starting using

- a guess of the set of constraints, which are active at the solution;
- a guess of the solution.

2.5.2 Exploitation of the problem structure

One of the ways to improve performance of the solver is to shape the optimization problem in a beneficial manner and to inform the solver about the structure of the problem.

2.5.2.1 Two-sided inequalities

If a task is bounded from both sides it is beneficial to express it in the following form

$$\underline{\boldsymbol{b}} \leq \boldsymbol{\mathcal{A}}\boldsymbol{\chi} \leq \bar{\boldsymbol{b}} \tag{2.6}$$

instead of splitting it into two parts corresponding to lower and upper bounds as is common in practice:

$$\begin{bmatrix} \boldsymbol{\mathcal{A}} \\ -\boldsymbol{\mathcal{A}} \end{bmatrix} \boldsymbol{\chi} \leq \begin{bmatrix} \bar{\boldsymbol{b}} \\ -\underline{\boldsymbol{b}} \end{bmatrix} \tag{2.7}$$

The reason for this is that bounds $\underline{\boldsymbol{b}} < \bar{\boldsymbol{b}}$ cannot be violated simultaneously, which can be exploited by a solver to reduce computational load.

2.5.2.2 Sparsity

We call a task *sparse* if the corresponding matrices and vectors contain a large number of zeros. A task with simple bounds (box constraints) on the decision variables

$$\underline{\boldsymbol{b}} \leq \boldsymbol{\chi} \leq \bar{\boldsymbol{b}} \tag{2.8}$$

is a typical example of a sparse task. Handling of such constraints can be implemented in a very efficient way and is supported by many solvers. It is often beneficial to reformulate an optimization problem in order to express inequality tasks with simple bounds.

The framework does not support generic sparse matrices, but only specific sparsity types listed in the table below

Equality (zero)	Equality	Lower bounds	Upper bounds	Lower and upper bounds
$\boldsymbol{\mathcal{A}}\boldsymbol{\chi} = 0$	$\boldsymbol{\mathcal{A}}\boldsymbol{\chi} = \boldsymbol{b}$	$\underline{\boldsymbol{b}} \leq \boldsymbol{\mathcal{A}}\boldsymbol{\chi}$	$\boldsymbol{\mathcal{A}}\boldsymbol{\chi} \leq \bar{\boldsymbol{b}}$	$\underline{\boldsymbol{b}} \leq \boldsymbol{\mathcal{A}}\boldsymbol{\chi} \leq \bar{\boldsymbol{b}}$
$\boldsymbol{\mathcal{A}}\boldsymbol{S}\boldsymbol{\chi} = 0$	$\boldsymbol{\mathcal{A}}\boldsymbol{S}\boldsymbol{\chi} = \boldsymbol{b}$	$\underline{\boldsymbol{b}} \leq \boldsymbol{\mathcal{A}}\boldsymbol{S}\boldsymbol{\chi}$	$\boldsymbol{\mathcal{A}}\boldsymbol{S}\boldsymbol{\chi} \leq \bar{\boldsymbol{b}}$	$\underline{\boldsymbol{b}} \leq \boldsymbol{\mathcal{A}}\boldsymbol{S}\boldsymbol{\chi} \leq \bar{\boldsymbol{b}}$
$\boldsymbol{G}\boldsymbol{\chi} = 0$	$\boldsymbol{G}\boldsymbol{\chi} = \boldsymbol{b}$	$\underline{\boldsymbol{b}} \leq \boldsymbol{G}\boldsymbol{\chi}$	$\boldsymbol{G}\boldsymbol{\chi} \leq \bar{\boldsymbol{b}}$	$\underline{\boldsymbol{b}} \leq \boldsymbol{G}\boldsymbol{\chi} \leq \bar{\boldsymbol{b}}$
$\boldsymbol{I}\boldsymbol{\chi} = 0$	$\boldsymbol{I}\boldsymbol{\chi} = \boldsymbol{b}$	$\underline{\boldsymbol{b}} \leq \boldsymbol{I}\boldsymbol{\chi}$	$\boldsymbol{I}\boldsymbol{\chi} \leq \bar{\boldsymbol{b}}$	$\underline{\boldsymbol{b}} \leq \boldsymbol{I}\boldsymbol{\chi} \leq \bar{\boldsymbol{b}}$

Here $\boldsymbol{S}$ selects a continuous segment of $\boldsymbol{\chi}$; $\boldsymbol{G}$ is a weighted selection matrix; $\boldsymbol{I}$ is a simple selection matrix. Note that not all sparsity types are supported by the solvers supported by the framework.

2.5.3 Early termination

Some solvers support early termination by imposing a limit on the number of iterations or computation time. Early termination is potentially dangerous since the solution returned by the solver is suboptimal, *i.e.*, some feasible tasks may not be satisfied.

Chapter 3

Reference

This chapter contains various derivations and definitions which are used by multiple modules of `Humoto`.

3.1 Model Predictive Control

Model Predictive Control (MPC) is a branch of control theory, where the control inputs are generated by optimizing behavior of a system over certain preview horizon.

A linear model of the system has the following form:

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_k \boldsymbol{x}_k + \boldsymbol{B}_k \boldsymbol{u}_k, \quad k = 0, \ldots, N \tag{3.1}$$

where $\boldsymbol{x}_k$ and $\boldsymbol{u}_k$ are k-th state and control input respectively, while N is the length of preview (prediction) horizon.

Ouptput of the system can be defined in different ways, here we assume that it depends on the preceding state and control

$$\boldsymbol{y}_{k+1} = \boldsymbol{D}_k \boldsymbol{x}_k + \boldsymbol{E}_k \boldsymbol{u}_k, \quad k = 0, \ldots, N \tag{3.2}$$

3.1.1 Condensing

Condensing amounts to finding such matrices $\boldsymbol{U}_x$ and $\boldsymbol{U}_u$ that

$$\boldsymbol{v}_x = \boldsymbol{U}_x \boldsymbol{x}_0 + \boldsymbol{U}_u \boldsymbol{v}_u, \tag{3.3}$$

where

$$\begin{aligned} \boldsymbol{v}_x &= (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N), \\ \boldsymbol{v}_u &= (\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{N-1}). \end{aligned} \tag{3.4}$$

Similarly for the output

$$\boldsymbol{v}_y = \boldsymbol{O}_x \boldsymbol{x}_0 + \boldsymbol{O}_u \boldsymbol{v}_u. \tag{3.5}$$

Note that

$$\boldsymbol{O}_x = \begin{bmatrix} \operatorname*{diag}_{k=0\ldots N-1}(D_k) & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{U}_x \end{bmatrix} \qquad \boldsymbol{O}_u = \begin{bmatrix} \operatorname*{diag}_{k=0\ldots N-1}(D_k) & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{U}_u \end{bmatrix} + \operatorname*{diag}_{k=0\ldots N-1}(E_k) \boldsymbol{U}_u \tag{3.6}$$

3.1.1.1 Time-variant system

$$\boldsymbol{U}_x = \begin{bmatrix} \boldsymbol{A}_0 \\ \boldsymbol{A}_1\boldsymbol{A}_0 \\ \vdots \\ \boldsymbol{A}_{N-1}\dots\boldsymbol{A}_0 \end{bmatrix} \qquad \boldsymbol{U}_u = \begin{bmatrix} \boldsymbol{B}_0 & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{A}_1\boldsymbol{B}_0 & \boldsymbol{B}_1 & \dots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}_{N-1}\dots\boldsymbol{A}_1\boldsymbol{B}_0 & \boldsymbol{A}_{N-1}\dots\boldsymbol{A}_2\boldsymbol{B}_1 & \dots & \boldsymbol{B}_{N-1} \end{bmatrix} \tag{3.7}$$

$$\boldsymbol{O}_x = \begin{bmatrix} \boldsymbol{D}_0 \\ \boldsymbol{D}_1\boldsymbol{A}_0 \\ \boldsymbol{D}_2\boldsymbol{A}_1\boldsymbol{A}_0 \\ \vdots \\ \boldsymbol{D}_{N-1}\boldsymbol{A}_{N-2}\dots\boldsymbol{A}_0 \end{bmatrix} \qquad \boldsymbol{O}_u = \begin{bmatrix} \boldsymbol{E}_0 & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{D}_1\boldsymbol{B}_0 & \boldsymbol{E}_1 & \dots & \boldsymbol{0} \\ \boldsymbol{D}_2\boldsymbol{A}_1\boldsymbol{B}_0 & \boldsymbol{D}_2\boldsymbol{B}_1 & \dots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{D}_{N-1}\boldsymbol{A}_{N-2}\dots\boldsymbol{A}_1\boldsymbol{B}_0 & \boldsymbol{D}_{N-1}\boldsymbol{A}_{N-2}\dots\boldsymbol{A}_2\boldsymbol{B}_1 & \dots & \boldsymbol{E}_{N-1} \end{bmatrix} \tag{3.8}$$

3.1.1.2 System is varying in the first preview interval

$$\boldsymbol{U}_x = \begin{bmatrix} \boldsymbol{A}_0 \\ \boldsymbol{A}\boldsymbol{A}_0 \\ \vdots \\ \boldsymbol{A}^{N-1}\boldsymbol{A}_0 \end{bmatrix} \qquad \boldsymbol{U}_u = \begin{bmatrix} \boldsymbol{B}_0 & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{A}\boldsymbol{B}_0 & \boldsymbol{B} & \dots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}^{N-1}\boldsymbol{B}_0 & \boldsymbol{A}^{N-2}\boldsymbol{B} & \dots & \boldsymbol{B} \end{bmatrix} \tag{3.9}$$

$$\boldsymbol{O}_x = \begin{bmatrix} \boldsymbol{D}_0 \\ \boldsymbol{D}\boldsymbol{A}_0 \\ \boldsymbol{D}\boldsymbol{A}\boldsymbol{A}_0 \\ \vdots \\ \boldsymbol{D}\boldsymbol{A}^{N-2}\boldsymbol{A}_0 \end{bmatrix} \qquad \boldsymbol{O}_u = \begin{bmatrix} \boldsymbol{E}_0 & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{D}\boldsymbol{B}_0 & \boldsymbol{E} & \dots & \boldsymbol{0} \\ \boldsymbol{D}\boldsymbol{A}\boldsymbol{B}_0 & \boldsymbol{D}\boldsymbol{B} & \dots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{D}\boldsymbol{A}^{N-2}\boldsymbol{B}_0 & \boldsymbol{D}\boldsymbol{A}^{N-3}\boldsymbol{B} & \dots & \boldsymbol{E} \end{bmatrix} \tag{3.10}$$

3.1.1.3 Time-invariant system

$$\boldsymbol{U}_x = \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{A}^2 \\ \vdots \\ \boldsymbol{A}^N \end{bmatrix} \qquad \boldsymbol{U}_u = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{A}\boldsymbol{B} & \boldsymbol{B} & \dots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}^{N-1}\boldsymbol{B} & \boldsymbol{A}^{N-2}\boldsymbol{B} & \dots & \boldsymbol{B} \end{bmatrix} \tag{3.11}$$

$$\boldsymbol{O}_x = \begin{bmatrix} \boldsymbol{D} \\ \boldsymbol{D}\boldsymbol{A} \\ \boldsymbol{D}\boldsymbol{A}^2 \\ \vdots \\ \boldsymbol{D}\boldsymbol{A}^{N-1} \end{bmatrix} \qquad \boldsymbol{O}_u = \begin{bmatrix} \boldsymbol{E} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{D}\boldsymbol{B} & \boldsymbol{E} & \dots & \boldsymbol{0} \\ \boldsymbol{D}\boldsymbol{A}\boldsymbol{B} & \boldsymbol{D}\boldsymbol{B} & \dots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{D}\boldsymbol{A}^{N-2}\boldsymbol{B} & \boldsymbol{D}\boldsymbol{A}^{N-3}\boldsymbol{B} & \dots & \boldsymbol{E} \end{bmatrix} \tag{3.12}$$

3.2 Triple integrator (discrete-time)

In some cases it is convenient to represent model of the system with triple integrator

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_k\boldsymbol{x}_k + \boldsymbol{B}_k\dddot{x}_k, \quad \boldsymbol{x}_k = (x_k, \dot{x}_k, \ddot{x}_k) \tag{3.13}$$

$$\boldsymbol{A}_k = \begin{bmatrix} 1 & T_k & \frac{T_k^2}{2} \\ 0 & 1 & T_k \\ 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{B}_k = \begin{bmatrix} \frac{T_k^3}{6} \\ \frac{T_k^2}{2} \\ T_k \end{bmatrix} \tag{3.14}$$

The corresponding matrices are defined in `Maxima` as

Maxima

Listing 3.1

```
A: matrix([1, T_k, T_k^2/2], [0, 1, T_k], [0, 0, 1]);
B: matrix([T_k^3/6], [T_k^2/2], [T_k]);
X0: matrix([x0],[dx0],[ddx0]);
U: matrix([dddx0]);
X1: matrix([x1],[dx1],[ddx1]);
As: matrix([1, T_s, T_s^2/2], [0, 1, T_s], [0, 0, 1]);
Bs: matrix([T_s^3/6], [T_s^2/2], [T_s]);
```

It is possible to reformulate the triple integrator model in order to use acceleration, velocity, or position from the next state as control input instead of the jerk. This may be useful in the cases when constraints on the state variables can be represented with simple bounds. The adjusted model produces the same motions, but numerical properties of its matrices may be different.

3.2.1 Controlled using acceleration

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_k \boldsymbol{x}_k + \boldsymbol{B}_k \ddot{x}_{k+1} \tag{3.15}$$

$$\dddot{x}_k = \boldsymbol{D}_k \boldsymbol{x}_k + \boldsymbol{E}_k \ddot{x}_{k+1} \tag{3.16}$$

$$\boldsymbol{A}_k = \begin{bmatrix} 1 & T_k & \frac{T_k^2}{3} \\ 0 & 1 & \frac{T_k}{2} \\ 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{B}_k = \begin{bmatrix} \frac{T_k^2}{6} \\ \frac{T_k}{2} \\ 1 \end{bmatrix}, \quad \boldsymbol{D}_k = \begin{bmatrix} 0 & 0 & -\frac{1}{T_k} \end{bmatrix}, \quad \boldsymbol{E}_k = \begin{bmatrix} \frac{1}{T_k} \end{bmatrix} \tag{3.17}$$

For $T_s \in [0, T_k]$

$$\boldsymbol{A}_{s,k} = \begin{bmatrix} 1 & T_s & -\frac{{T_s}^3 - 3 T_k {T_s}^2}{6 T_k} \\ 0 & 1 & -\frac{{T_s}^2 - 2 T_k T_s}{2 T_k} \\ 0 & 0 & -\frac{T_s - T_k}{T_k} \end{bmatrix}, \quad \boldsymbol{B}_{s,k} = \begin{bmatrix} \frac{{T_s}^3}{6 T_k} \\ \frac{{T_s}^2}{2 T_k} \\ \frac{T_s}{T_k} \end{bmatrix} \tag{3.18}$$

Maxima

Listing 3.2

```
e: solve([(A.X0 + B.U)[3][1] = (X1)[3][1]], [dddx0]);
Dnew: coefmatrix([rhs(e[1])], list_matrix_entries(X0));
Enew: coefmatrix([rhs(e[1])], [ddx1]);
e: subst(e, A.X0 + B.U);
Anew: coefmatrix(list_matrix_entries(e), list_matrix_entries(X0));
Bnew: coefmatrix(list_matrix_entries(e), [ddx1]);
tex(Anew);
tex(Bnew);
tex(Dnew);
tex(Enew);
/* subsampling */
e: solve([(A.X0 + B.U)[3][1] = (X1)[3][1]], [dddx0]);
e: subst(e, As.X0 + Bs.U);
Asnew: coefmatrix(list_matrix_entries(e), list_matrix_entries(X0));
Bsnew: coefmatrix(list_matrix_entries(e), [ddx1]);
tex(Asnew);
tex(Bsnew);
/* check */
subst([T_s = T_k], Asnew) - Anew;
subst([T_s = T_k], Bsnew) - Bnew;
```

3.2.2 Controlled using velocity

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_k \boldsymbol{x}_k + \boldsymbol{B}_k \dot{x}_{k+1} \tag{3.19}$$

$$\dddot{x}_k = \boldsymbol{D}_k \boldsymbol{x}_k + \boldsymbol{E}_k \dot{x}_{k+1} \tag{3.20}$$

$$\boldsymbol{A}_k = \begin{bmatrix} 1 & \frac{2T_k}{3} & \frac{T_k^2}{6} \\ 0 & 0 & 0 \\ 0 & -\frac{2}{T_k} & -1 \end{bmatrix}, \quad \boldsymbol{B}_k = \begin{bmatrix} \frac{T_k}{3} \\ 1 \\ \frac{2}{T_k} \end{bmatrix}, \quad \boldsymbol{D}_k = \begin{bmatrix} 0 & -\frac{2}{T_k^2} & -\frac{2}{T_k} \end{bmatrix}, \quad \boldsymbol{E}_k = \begin{bmatrix} \frac{2}{T_k^2} \end{bmatrix} \tag{3.21}$$

For $T_s \in [0, T_k]$

$$\boldsymbol{A}_{s,k} = \begin{bmatrix} 1 & -\frac{{T_s}^3 - 3{T_k}^2 T_s}{3{T_k}^2} & -\frac{2{T_s}^3 - 3T_k {T_s}^2}{6T_k} \\ 0 & -\frac{{T_s}^2 - {T_k}^2}{{T_k}^2} & -\frac{{T_s}^2 - T_k T_s}{T_k} \\ 0 & -\frac{2T_s}{{T_k}^2} & -\frac{2T_s - T_k}{T_k} \end{bmatrix}, \quad \boldsymbol{B}_{s,k} = \begin{bmatrix} \frac{{T_s}^3}{3{T_k}^2} \\ \frac{{T_s}^2}{{T_k}^2} \\ \frac{2T_s}{{T_k}^2} \end{bmatrix} \tag{3.22}$$

Maxima

Listing 3.3

```
e: solve([(A.X0 + B.U)[2][1] = (X1)[2][1]], [dddx0]);
Dnew: coefmatrix([rhs(e[1])], list_matrix_entries(X0));
Enew: coefmatrix([rhs(e[1])], [dx1]);
e: subst(e, A.X0 + B.U);
Anew: coefmatrix(list_matrix_entries(e), list_matrix_entries(X0));
Bnew: coefmatrix(list_matrix_entries(e), [dx1]);
tex(Anew);
tex(Bnew);
tex(Dnew);
tex(Enew);
/* subsampling */
e: solve([(A.X0 + B.U)[2][1] = (X1)[2][1]], [dddx0]);
e: subst(e, As.X0 + Bs.U);
Asnew: coefmatrix(list_matrix_entries(e), list_matrix_entries(X0));
Bsnew: coefmatrix(list_matrix_entries(e), [dx1]);
tex(Asnew);
tex(Bsnew);
/* check */
subst([T_s = T_k], Asnew) - Anew;
subst([T_s = T_k], Bsnew) - Bnew;
```

3.2.3 Controlled using position

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_k \boldsymbol{x}_k + \boldsymbol{B}_k x_{k+1} \tag{3.23}$$

$$\dddot{x}_k = \boldsymbol{D}_k \boldsymbol{x}_k + \boldsymbol{E}_k x_{k+1} \tag{3.24}$$

$$\boldsymbol{A}_k = \begin{bmatrix} 0 & 0 & 0 \\ -\frac{3}{T_k} & -2 & -\frac{T_k}{2} \\ -\frac{6}{T_k^2} & -\frac{6}{T_k} & -2 \end{bmatrix}, \quad \boldsymbol{B}_k = \begin{bmatrix} 1 \\ \frac{3}{T_k} \\ \frac{6}{T_k^2} \end{bmatrix}, \quad \boldsymbol{D}_k = \begin{bmatrix} -\frac{6}{T_k^3} & -\frac{6}{T_k^2} & -\frac{3}{T_k} \end{bmatrix}, \quad \boldsymbol{E}_k = \begin{bmatrix} \frac{6}{T_k^3} \end{bmatrix} \tag{3.25}$$

For $T_s \in [0, T_k]$

$$\boldsymbol{A}_{s,k} = \begin{bmatrix} -\frac{{T_s}^3 - {T_k}^3}{{T_k}^3} & -\frac{{T_s}^3 - {T_k}^2 T_s}{{T_k}^2} & -\frac{{T_s}^3 - T_k {T_s}^2}{2T_k} \\ -\frac{3{T_s}^2}{{T_k}^3} & -\frac{3{T_s}^2 - {T_k}^2}{{T_k}^2} & -\frac{3{T_s}^2 - 2T_k T_s}{2T_k} \\ -\frac{6T_s}{{T_k}^3} & -\frac{6T_s}{{T_k}^2} & -\frac{3T_s - T_k}{T_k} \end{bmatrix}, \quad \boldsymbol{B}_{s,k} = \begin{bmatrix} \frac{{T_s}^3}{{T_k}^3} \\ \frac{3{T_s}^2}{{T_k}^3} \\ \frac{6T_s}{{T_k}^3} \end{bmatrix} \tag{3.26}$$

Maxima

Listing 3.4

```
e: solve([(A.X0 + B.U)[1][1] = (X1)[1][1]], [dddx0]);
Dnew: coefmatrix([rhs(e[1])], list_matrix_entries(X0));
Enew: coefmatrix([rhs(e[1])], [x1]);
e: subst(e, A.X0 + B.U);
Anew: coefmatrix(list_matrix_entries(e), list_matrix_entries(X0));
Bnew: coefmatrix(list_matrix_entries(e), [x1]);
tex(Anew);
tex(Bnew);
tex(Dnew);
tex(Enew);
/* subsampling */
e: solve([(A.X0 + B.U)[1][1] = (X1)[1][1]], [dddx0]);
e: subst(e, As.X0 + Bs.U);
Asnew: coefmatrix(list_matrix_entries(e), list_matrix_entries(X0));
Bsnew: coefmatrix(list_matrix_entries(e), [x1]);
tex(Asnew);
tex(Bsnew);
/* check */
subst([T_s = T_k], Asnew) - Anew;
subst([T_s = T_k], Bsnew) - Bnew;
```

3.3 Kronecker and Block Kronecker products

The standard Kronecker product is defined as follows:

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} A_{1,1}\boldsymbol{B} & \dots & A_{1,m}\boldsymbol{B} \\ \vdots & \vdots & \vdots \\ A_{n,1}\boldsymbol{B} & \dots & A_{n,m}\boldsymbol{B} \end{bmatrix}, \tag{3.27}$$

where $\boldsymbol{A} \in \mathbb{R}^{n \times m}$ and $A_{i,j}$ denotes a scalar standing in the i-th row and j-th column.

Derivations presented in this manual can often be expressed using a slightly different operator:

$$\boldsymbol{A} \boxtimes \boldsymbol{B} = \begin{bmatrix} \boldsymbol{A} \otimes \boldsymbol{B}_{[1,1]} & \dots & \boldsymbol{A} \otimes \boldsymbol{B}_{[1,m]} \\ \vdots & \vdots & \vdots \\ \boldsymbol{A} \otimes \boldsymbol{B}_{[n,1]} & \dots & \boldsymbol{A} \otimes \boldsymbol{B}_{[n,m]} \end{bmatrix}, \tag{3.28}$$

where $\boldsymbol{B}_{[i,j]}$ denotes a block of matrix $\boldsymbol{B}$. The operator $\boxtimes$ is called block Kronecker product and is usually defined in a more general form, where matrix $\boldsymbol{A}$ is also partitioned [8, 7, 6]. For the purpose of the present work partitioning of $\boldsymbol{A}$ and nonuniform partitioning of $\boldsymbol{B}$ are not necessary and therefore not considered.

Block Kronecker product has several valuable properties, notably:

$$(\boldsymbol{I} \boxtimes \boldsymbol{A})(\boldsymbol{I} \boxtimes \boldsymbol{B}) = \boldsymbol{I} \boxtimes (\boldsymbol{A}\boldsymbol{B}), \tag{3.29}$$

which allows for more efficient computations. Another commonly used property is the following:

$$\operatorname*{diag}_{k=1\dots N} (\boldsymbol{I} \otimes \boldsymbol{A}_k) = \boldsymbol{I} \boxtimes \operatorname*{diag}_{k=1\dots N} (\boldsymbol{A}_k) \tag{3.30}$$

For example, consider a time-invariant system with the state transition matrix $\boldsymbol{A} = \boldsymbol{I} \otimes \bar{\boldsymbol{A}}$ and control matrix $\boldsymbol{B} = \boldsymbol{I} \otimes \bar{\boldsymbol{B}}$. Then the condensing matrices $\boldsymbol{U}_x$ and $\boldsymbol{U}_u$ can be expressed as

$$\boldsymbol{U}_x = \boldsymbol{I} \boxtimes \bar{\boldsymbol{U}}_x = \boldsymbol{I} \boxtimes \begin{bmatrix} \bar{\boldsymbol{A}} \\ \bar{\boldsymbol{A}}^2 \\ \vdots \\ \bar{\boldsymbol{A}}^N \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} \otimes \bar{\boldsymbol{A}} \\ \boldsymbol{I} \otimes \bar{\boldsymbol{A}}^2 \\ \vdots \\ \boldsymbol{I} \otimes \bar{\boldsymbol{A}}^N \end{bmatrix} \tag{3.31}$$

$$\boldsymbol{U}_u = \boldsymbol{I} \boxtimes \bar{\boldsymbol{U}}_u = \boldsymbol{I} \boxtimes \begin{bmatrix} \bar{\boldsymbol{B}} & \bar{\boldsymbol{0}} & \dots & \bar{\boldsymbol{0}} \\ \bar{\boldsymbol{A}}\bar{\boldsymbol{B}} & \bar{\boldsymbol{B}} & \dots & \bar{\boldsymbol{0}} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{\boldsymbol{A}}^{N-1}\bar{\boldsymbol{B}} & \bar{\boldsymbol{A}}^{N-2}\bar{\boldsymbol{B}} & \dots & \bar{\boldsymbol{B}} \end{bmatrix} \tag{3.32}$$

Bibliography

[1] Wikipedia. *Goal programming — Wikipedia, The Free Encyclopedia*. [Online; accessed May-2017]. 2017. URL: https://en.wikipedia.org/w/index.php?title=Goal_programming.

[2] A. Sherikov. "Balance preservation and task prioritization in whole body motion control of humanoid robots". PhD thesis. Communauté Université Grenoble Alpes, 2016.

[3] D. Dimitrov, A. Sherikov, and P.-B. Wieber. "Efficient resolution of potentially conflicting linear constraints in robotics". Submitted to IEEE TRO (05/August/2015). 2015. URL: https://hal.inria.fr/hal-01183003.

[4] A. Escande, N. Mansard, and P.-B. Wieber. "Hierarchical quadratic programming: Fast online humanoid-robot motion generation". In: *The International Journal of Robotics Research* (2014). DOI: 10.1177/0278364914521306.

[5] L. Saab, O. Ramos, F. Keith, N. Mansard, P. Soueres, and J. Fourquet. "Dynamic Whole-Body Motion Generation Under Rigid Contacts and Other Unilateral Constraints". In: *Robotics, IEEE Transactions on* 29.2 (2013), pp. 346–362. ISSN: 1552-3098. DOI: 10.1109/TRO.2012.2234351.

[6] R. H. Koning, H. Neudecker, and T. Wansbeek. "Block Kronecker products and the vecb operator". In: *Linear Algebra and its Applications* 149 (1991), pp. 165 – 184. ISSN: 0024-3795. DOI: http://dx.doi.org/10.1016/0024-3795(91)90332-Q. URL: http://www.sciencedirect.com/science/article/pii/002437959190332Q.

[7] D. C. Hyland and J. Emmanuel G. Collins. "Block Kronecker Products and Block Norm Matrices in Large-Scale Systems Analysis". In: *SIAM Journal on Matrix Analysis and Applications* 10.1 (1989), pp. 18–29. DOI: 10.1137/0610002. eprint: http://dx.doi.org/10.1137/0610002. URL: http://dx.doi.org/10.1137/0610002.

[8] D. S. Tracy and R. P. Singh. "A new matrix product and its applications in partitioned matrix differentiation". In: *Statistica Neerlandica* 26.4 (1972), pp. 143–157. ISSN: 1467-9574. DOI: 10.1111/j.1467-9574.1972.tb00199.x. URL: http://dx.doi.org/10.1111/j.1467-9574.1972.tb00199.x.

Part II

Walking Pattern Generators

Chapter 1

Introduction

1.1 Single point-mass models with coplanar contacts

1.1.1 Common variables

Position, velocity, acceleration, jerk of the CoM:

$$\boldsymbol{c} = \begin{bmatrix} c^x \\ c^y \end{bmatrix} \quad \boldsymbol{C} = \begin{bmatrix} \boldsymbol{c}_1 \\ \vdots \\ \boldsymbol{c}_N \end{bmatrix} \qquad \dot{\boldsymbol{c}} = \begin{bmatrix} \dot{c}^x \\ \dot{c}^y \end{bmatrix} \quad \dot{\boldsymbol{C}} = \begin{bmatrix} \dot{\boldsymbol{c}}_1 \\ \vdots \\ \dot{\boldsymbol{c}}_N \end{bmatrix} \tag{1.1}$$

$$\ddot{\boldsymbol{c}} = \begin{bmatrix} \ddot{c}^x \\ \ddot{c}^y \end{bmatrix} \quad \ddot{\boldsymbol{C}} = \begin{bmatrix} \ddot{\boldsymbol{c}}_1 \\ \vdots \\ \ddot{\boldsymbol{c}}_N \end{bmatrix} \qquad \dddot{\boldsymbol{c}} = \begin{bmatrix} \dddot{c}^x \\ \dddot{c}^y \end{bmatrix} \quad \dddot{\boldsymbol{C}} = \begin{bmatrix} \dddot{\boldsymbol{c}}_0 \\ \vdots \\ \dddot{\boldsymbol{c}}_{N-1} \end{bmatrix} \tag{1.2}$$

1.1.2 Discrete-time systems based on triple integrator

The state of the CoM:

$$\hat{\boldsymbol{c}} = \begin{bmatrix} \hat{\boldsymbol{c}}^x \\ \hat{\boldsymbol{c}}^y \end{bmatrix} = \begin{bmatrix} c^x \\ \dot{c}^x \\ \ddot{c}^x \\ c^y \\ \dot{c}^y \\ \ddot{c}^y \end{bmatrix} \qquad \hat{\boldsymbol{C}} = \begin{bmatrix} \hat{\boldsymbol{c}}_1 \\ \vdots \\ \hat{\boldsymbol{c}}_N \end{bmatrix} \tag{1.3}$$

Selection matrices:

$$\boldsymbol{I}_p = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad \boldsymbol{I}_v = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad \boldsymbol{I}_a = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.4}$$

$$\boldsymbol{I}_{pv} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{1.5}$$

1.1.2.1 Piece-wise constant jerk

The dynamic system:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \tag{1.6}$$

can be defined as:

$$\begin{aligned} \boldsymbol{x} &= \hat{\boldsymbol{c}} \\ \boldsymbol{u} &= \dddot{\boldsymbol{c}} \end{aligned} \tag{1.7}$$

resulting in:

$$\boldsymbol{A}_k = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \boldsymbol{B}_k = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{1.8}$$

After discretization, the model is now:

$$\begin{aligned} \hat{\boldsymbol{c}}_{k+1} =& \boldsymbol{A}_k \hat{\boldsymbol{c}}_k + \boldsymbol{B}_k \dddot{\boldsymbol{c}}_k \\ \boldsymbol{z}_{k+1} =& \boldsymbol{D}_k \hat{\boldsymbol{c}}_{k+1} \end{aligned} \tag{1.9}$$

$$\boldsymbol{A}_k = \begin{bmatrix} 1 & T_k & T_k^2/2 & 0 & 0 & 0 \\ 0 & 1 & T_k & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T_k & T_k^2/2 \\ 0 & 0 & 0 & 0 & 1 & T_k \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \boldsymbol{B}_k = \begin{bmatrix} T_k^3/6 & 0 \\ T_k^2/2 & 0 \\ T & 0 \\ 0 & T_k^3/6 \\ 0 & T_k^2/2 \\ 0 & T \end{bmatrix} \tag{1.10}$$

$$\boldsymbol{D}_k = \begin{bmatrix} 1 & 0 & -\frac{1}{\omega_k^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -\frac{1}{\omega_k^2} \end{bmatrix} \quad \omega_k = \sqrt{\frac{g}{c_k^z}} \tag{1.11}$$

1.1.2.2 Piece-wise constant CoP velocity

The same dynamic system of Eq.(1.6) can be defined as:

$$\begin{aligned} \boldsymbol{x} &= \hat{\boldsymbol{c}} \\ \boldsymbol{u} &= \dot{\boldsymbol{z}} \end{aligned} \tag{1.12}$$

resulting in:

$$\boldsymbol{A}_k = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & \omega^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & \omega^2 & 0 \end{bmatrix} \quad \boldsymbol{B}_k = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -\omega^2 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & -\omega^2 \end{bmatrix} \tag{1.13}$$

Discretization in Maxima (see [1]):

Maxima

Listing 1.1

```
load("diag");
A: matrix([0, 1, 0], [0, 0, 1], [0, w^2, 0]);
B: matrix([0],[0],[-w^2]);
Ad: mat_function(exp, A*T);
Bd: expand(integrate(mat_function(exp, A*t), t, 0, T).B);
```

The resulting model is:

$$\begin{aligned} \hat{\boldsymbol{c}}_{k+1} =& \boldsymbol{A}_k \hat{\boldsymbol{c}}_k + \boldsymbol{B}_k \dot{\boldsymbol{z}}_k \\ \boldsymbol{z}_{k+1} =& \boldsymbol{D}_k \hat{\boldsymbol{c}}_{k+1} \end{aligned} \tag{1.14}$$

$$\boldsymbol{A}_k = \begin{bmatrix} 1 & \frac{\sinh(T_k\omega_k)}{\omega_k} & \frac{\cosh(T_k\omega_k)-1}{\omega_k^2} & 0 & 0 & 0 \\ 0 & \cosh(T_k\omega_k) & \frac{\sinh(T_k\omega_k)}{\omega_k} & 0 & 0 & 0 \\ 0 & \omega_k\sinh(T_k\omega_k) & \cosh(T_k\omega_k) & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \frac{\sinh(T_k\omega_k)}{\omega_k} & \frac{\cosh(T_k\omega_k)-1}{\omega_k^2} \\ 0 & 0 & 0 & 0 & \cosh(T_k\omega_k) & \frac{\sinh(T_k\omega_k)}{\omega_k} \\ 0 & 0 & 0 & 0 & \omega_k\sinh(T_k\omega_k) & \cosh(T_k\omega_k) \end{bmatrix} \tag{1.15}$$

$$\boldsymbol{B}_k = \begin{bmatrix} -\frac{\sinh(T_k\omega_k)}{\omega_k} + T_k & 0 \\ -\cosh(T_k\omega_k)+1 & 0 \\ -\omega\sinh(T_k\omega_k) & 0 \\ 0 & -\frac{\sinh(T_k\omega_k)}{\omega_k} + T_k \\ 0 & -\cosh(T_k\omega_k)+1 \\ 0 & -\omega\sinh(T_k\omega_k) \end{bmatrix} \tag{1.16}$$

The state of the system and the output matrix are the same as above.

1.1.2.3 Piece-wise constant CoP velocity (control is the CoP position)

Let the system controlled by piece-wise constant CoP velocity be defined as

$$\begin{aligned} \hat{\boldsymbol{c}}_{k+1} &= \tilde{\boldsymbol{A}}_k \hat{\boldsymbol{c}}_k + \tilde{\boldsymbol{B}}_k \dot{\boldsymbol{z}}_k \\ \boldsymbol{z}_{k+1} &= \tilde{\boldsymbol{D}}_k \hat{\boldsymbol{c}}_{k+1}, \end{aligned} \tag{1.17}$$

where matrices $\tilde{\boldsymbol{A}}_k, \tilde{\boldsymbol{B}}_k, \tilde{\boldsymbol{D}}_k$ are defined as above.

We can express $\boldsymbol{z}_{k+1}$ using $\hat{\boldsymbol{c}}_k$ and $\dot{\boldsymbol{z}}_k$ as

$$\boldsymbol{z}_{k+1} = \tilde{\boldsymbol{D}}_k \hat{\boldsymbol{c}}_k + T_k \dot{\boldsymbol{z}}_k. \tag{1.18}$$

Hence we express the CoP velocity through CoM and CoP positions

$$\dot{\boldsymbol{z}}_k = \frac{1}{T_k}\left(\boldsymbol{z}_{k+1} - \tilde{\boldsymbol{D}}_k \hat{\boldsymbol{c}}_k\right), \tag{1.19}$$

and substitute it back into our model:

$$\hat{\boldsymbol{c}}_{k+1} = \tilde{\boldsymbol{A}}_k \hat{\boldsymbol{c}}_k + \frac{1}{T_k}\tilde{\boldsymbol{B}}_k \boldsymbol{z}_{k+1} - \frac{1}{T_k}\tilde{\boldsymbol{B}}_k \tilde{\boldsymbol{D}}_k \hat{\boldsymbol{c}}_k = \left(\tilde{\boldsymbol{A}}_k - \frac{1}{T_k}\tilde{\boldsymbol{B}}_k \tilde{\boldsymbol{D}}_k\right)\hat{\boldsymbol{c}}_k + \frac{1}{T_k}\tilde{\boldsymbol{B}}_k \boldsymbol{z}_{k+1} \tag{1.20}$$

Thus we obtain a new model

$$\hat{\boldsymbol{c}}_{k+1} = \boldsymbol{A}\hat{\boldsymbol{c}}_k + \boldsymbol{B}\boldsymbol{z}_{k+1} \tag{1.21}$$

$$\dot{\boldsymbol{z}}_k = \boldsymbol{D}_k \hat{\boldsymbol{c}}_k + \boldsymbol{E}_k \boldsymbol{z}_{k+1}, \tag{1.22}$$

where

$$\boldsymbol{A}_k = \left(\tilde{\boldsymbol{A}}_k - \frac{1}{T_k}\tilde{\boldsymbol{B}}_k \tilde{\boldsymbol{D}}_k\right) \qquad \boldsymbol{B}_k = \frac{1}{T_k}\tilde{\boldsymbol{B}}_k \tag{1.23}$$

$$\boldsymbol{D}_k = -\frac{1}{T_k}\tilde{\boldsymbol{D}}_k \qquad \boldsymbol{E}_k = \operatorname*{diag}_2\left(\frac{1}{T_k}\right) \tag{1.24}$$

$$
\boldsymbol{A}_k = \begin{bmatrix}
\frac{\text{sh}}{\omega_k T_k} & \frac{\text{sh}}{\omega_k} & -\frac{\text{sh} - \omega_k T_k \text{ ch}}{\omega_k^3 T_k} & 0 & 0 & 0 \\
\frac{\text{ch} - 1}{T_k} & \text{ch} & \frac{\omega_k T_k \text{ sh} - \text{ch} + 1}{\omega_k^2 T_k} & 0 & 0 & 0 \\
\frac{\omega_k \text{ sh}}{T_k} & \omega_k \text{ sh} & -\frac{\text{sh} - \omega_k T_k \text{ ch}}{\omega_k T_k} & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{\text{sh}}{\omega_k T_k} & \frac{\text{sh}}{\omega_k} & -\frac{\text{sh} - \omega_k T_k \text{ ch}}{\omega_k^3 T_k} \\
0 & 0 & 0 & \frac{\text{ch} - 1}{T_k} & \text{ch} & \frac{\omega_k T_k \text{ sh} - \text{ch} + 1}{\omega_k^2 T_k} \\
0 & 0 & 0 & \frac{\omega_k \text{ sh}}{T_k} & \omega_k \text{ sh} & -\frac{\text{sh} - \omega_k T_k \text{ ch}}{\omega_k T_k}
\end{bmatrix}, \tag{1.25}
$$

$$
\boldsymbol{B}_k = \begin{bmatrix}
-\frac{\text{sh} - \omega_k T_k}{\omega_k T_k} & 0 \\
-\frac{\text{ch} - 1}{T_k} & 0 \\
-\frac{\omega_k \text{ sh}}{T_k} & 0 \\
0 & -\frac{\text{sh} - \omega_k T_k}{\omega_k T_k} \\
0 & -\frac{\text{ch} - 1}{T_k} \\
0 & -\frac{\omega_k \text{ sh}}{T_k}
\end{bmatrix}, \tag{1.26}
$$

where

$$
\begin{aligned}
\text{sh} &= \sinh(\omega_k T_k) \\
\text{ch} &= \cosh(\omega_k T_k)
\end{aligned} \tag{1.27}
$$

Obtaining the matrices in Maxima:

Maxima

Listing 1.2

```
load("diag");
A: matrix([0, 1, 0], [0, 0, 1], [0, w^2, 0]);
B: matrix([0],[0],[-w^2]);
Ad: mat_function(exp, A*T);
Bd: expand(integrate(mat_function(exp, A*t), t, 0, T).B);
D: matrix([1, 0, -1/w^2]);

Anew: (Ad - Bd.D/T);
Bnew: Bd/T;
Dnew: -D/T;
Enew: matrix([1/T, 0], [0, 1/T]);

Anew: hypsimp(w*T, Anew);
Bnew: hypsimp(w*T, Bnew);

tex(Anew);
tex(Bnew);
```

Chapter 2

WPG v.04

This version of the pattern generator is based on triple integrator with piece-wise constant CoP velocity. The control input is position of the CoP. Apart from that it is very similar to the WPG proposed in [3], for more information refer to [2].

2.1 Variables

2.1.1 Footstep positions

Any preview horizon contains 1 fixed and 0 or more variable footstep positions. Let M be the number of variable steps in the preview horizon, $\mathsf{p_j}$ – frame fixed to j-th footstep with $j = 0 \dots M$ (0-th footstep is fixed), $\boldsymbol{p}_j = (p_j^x, p_j^y)$ – position of the j-th footstep on the ground plane.

Coordinate transformation matrix from the frame fixed to the j-th footstep to the global frame is defined as

$$\boldsymbol{R}_{\mathsf{p_j}} = \begin{bmatrix} \cos\theta_j & -\sin\theta_j \\ \sin\theta_j & \cos\theta_j \end{bmatrix} \tag{2.1}$$

Orientations θ_j of the footsteps are predetermined to avoid nolinearity. Position of the j-th footstep in the global frame can be found as

$$\boldsymbol{p}_j = \boldsymbol{p}_0 + \sum_{i=1}^{j} \boldsymbol{R}_{\mathsf{p_{i-1}}} {}^{\mathsf{p_{i-1}}}\boldsymbol{p}_j, \tag{2.2}$$

which leads to

$$\begin{bmatrix} \boldsymbol{p}_0 \\ \boldsymbol{p}_1 \\ \vdots \\ \boldsymbol{p}_M \end{bmatrix} = \mathbf{1}_M \otimes \boldsymbol{p}_0 + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \boldsymbol{R}_{\mathsf{p_0}} & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \vdots & \dots & \vdots \\ \boldsymbol{R}_{\mathsf{p_0}} & \boldsymbol{R}_{\mathsf{p_1}} & \dots & \boldsymbol{R}_{\mathsf{p_{M-1}}} \end{bmatrix} \begin{bmatrix} {}^{\mathsf{p_0}}\boldsymbol{p}_1 \\ \vdots \\ {}^{\mathsf{p_{M-1}}}\boldsymbol{p}_M \end{bmatrix}. \tag{2.3}$$

It is preferable to express variable footstep positions in a frame fixed to the preceding footstep in order to impose simple bounds on variable footstep positions.

The number of sampling intervals in the preview horizon is denoted by $N \geq M$. The footstep position corresponding to k-th ($k = 1, \dots, N$) sampling interval is denoted as $\hat{\boldsymbol{p}}_k = (\hat{p}_k^x, \hat{p}_k^y)$.

These positions can be found using a selection matrices as

$$\underbrace{\begin{bmatrix}\hat{\boldsymbol{p}}_1\\ \vdots \\ \hat{\boldsymbol{p}}_N\end{bmatrix}}_{\hat{\boldsymbol{P}}} = \underbrace{\begin{bmatrix}\boldsymbol{I} & \boldsymbol{0} & \dots & \boldsymbol{0}\\ \boldsymbol{I} & \boldsymbol{0} & \dots & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I} & \dots & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I} & \dots & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I} & \dots & \boldsymbol{0}\\ \vdots & \vdots & \ddots & \vdots\\ \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{I}\end{bmatrix}}_{\boldsymbol{I}_{fps}} \begin{bmatrix}\boldsymbol{p}_0\\ \boldsymbol{p}_1\\ \vdots\\ \boldsymbol{p}_M\end{bmatrix} = \begin{bmatrix}\boldsymbol{I}\\ \boldsymbol{I}\\ \boldsymbol{0}\\ \boldsymbol{0}\\ \boldsymbol{0}\\ \vdots\\ \boldsymbol{0}\end{bmatrix}\boldsymbol{p}_0 + \begin{bmatrix}\boldsymbol{0} & \dots & \boldsymbol{0}\\ \boldsymbol{0} & \dots & \boldsymbol{0}\\ \boldsymbol{I} & \dots & \boldsymbol{0}\\ \boldsymbol{I} & \dots & \boldsymbol{0}\\ \boldsymbol{I} & \dots & \boldsymbol{0}\\ \vdots & \ddots & \vdots\\ \boldsymbol{0} & \dots & \boldsymbol{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{p}_1\\ \vdots\\ \boldsymbol{p}_M\end{bmatrix}$$

$$= \underbrace{\boldsymbol{1}_N \otimes \boldsymbol{p}_0}_{\boldsymbol{V}_0} + \underbrace{\begin{bmatrix}\boldsymbol{0} & \dots & \boldsymbol{0}\\ \boldsymbol{0} & \dots & \boldsymbol{0}\\ \boldsymbol{I} & \dots & \boldsymbol{0}\\ \boldsymbol{I} & \dots & \boldsymbol{0}\\ \boldsymbol{I} & \dots & \boldsymbol{0}\\ \vdots & \ddots & \vdots\\ \boldsymbol{0} & \dots & \boldsymbol{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{R}_{\mathsf{p}_0} & \boldsymbol{0} & \dots & \boldsymbol{0}\\ \vdots & \vdots & \dots & \vdots\\ \boldsymbol{R}_{\mathsf{p}_0} & \boldsymbol{R}_{\mathsf{p}_1} & \dots & \boldsymbol{R}_{\mathsf{p}_{M-1}}\end{bmatrix}}_{\boldsymbol{V}} \underbrace{\begin{bmatrix}{}^{\mathsf{p}_0}\boldsymbol{p}_1\\ \vdots\\ {}^{\mathsf{p}_{M-1}}\boldsymbol{p}_M\end{bmatrix}}_{\boldsymbol{P}}, \tag{2.4}$$

where $\boldsymbol{V}$ has the following structure

$$\boldsymbol{V} = \begin{bmatrix}\boldsymbol{0} & \dots & \boldsymbol{0}\\ \boldsymbol{0} & \dots & \boldsymbol{0}\\ \boldsymbol{R}_{\mathsf{p}_0} & \dots & \boldsymbol{0}\\ \boldsymbol{R}_{\mathsf{p}_0} & \dots & \boldsymbol{0}\\ \boldsymbol{R}_{\mathsf{p}_0} & \dots & \boldsymbol{0}\\ \vdots & \ddots & \vdots\\ \boldsymbol{R}_{\mathsf{p}_0} & \dots & \boldsymbol{R}_{\mathsf{p}_{M-1}}\end{bmatrix} = \underbrace{\begin{bmatrix}\boldsymbol{0} & \dots & \boldsymbol{0}\\ \boldsymbol{0} & \dots & \boldsymbol{0}\\ \boldsymbol{I} & \dots & \boldsymbol{0}\\ \boldsymbol{I} & \dots & \boldsymbol{0}\\ \boldsymbol{I} & \dots & \boldsymbol{0}\\ \vdots & \ddots & \vdots\\ \boldsymbol{I} & \dots & \boldsymbol{I}\end{bmatrix}}_{\boldsymbol{I}_R \otimes \boldsymbol{I}} \operatorname*{diag}_{k=0\dots M-1}(\boldsymbol{R}_{\mathsf{p}_k}) \tag{2.5}$$

where

$$\boldsymbol{I}_R = \begin{bmatrix}0 & \dots & 0\\ 0 & \dots & 0\\ 1 & \dots & 0\\ 1 & \dots & 0\\ 1 & \dots & 0\\ \vdots & \ddots & \vdots\\ 1 & \dots & 1\end{bmatrix}. \tag{2.6}$$

2.1.2 Positions of the CoP

Position of the center of pressure in the end of k-th ($k = 1, \dots, N$) sampling interval is denoted as $\boldsymbol{z}_k = (z_k^x, z_k^y)$. The current CoP position is $\boldsymbol{z}_0$. In order to obtain simple bounds on the CoP positions instead of general constraints, their positions are expressed in the local frames fixed to the respective feet, *i.e.*

$$\boldsymbol{z}_k = \hat{\boldsymbol{p}}_k + \boldsymbol{R}_{\hat{\mathsf{p}}_k}{}^{\hat{\mathsf{p}}_k}\boldsymbol{z}_k. \tag{2.7}$$

Orientation matrices $\boldsymbol{R}_{\hat{\mathsf{p}}_k}$ can also be selected with $\boldsymbol{I}_{fps}$

$$\begin{bmatrix}\boldsymbol{R}_{\hat{\mathsf{p}}_1}\\ \vdots\\ \boldsymbol{R}_{\hat{\mathsf{p}}_N}\end{bmatrix} = \boldsymbol{I}_{fps}\begin{bmatrix}\boldsymbol{R}_{\mathsf{p}_0}\\ \vdots\\ \boldsymbol{R}_{\mathsf{p}_M}\end{bmatrix} \tag{2.8}$$

All positions of the CoP within the preview horizon are

$$\begin{bmatrix} \boldsymbol{z}_1 \\ \vdots \\ \boldsymbol{z}_N \end{bmatrix} = \hat{\boldsymbol{P}} + \underset{k=1\dots N}{\text{diag}} (\boldsymbol{R}_{\hat{\mathsf{p}}_k}) \underbrace{\begin{bmatrix} {}^{\hat{\mathsf{p}}_1}\boldsymbol{z}_1 \\ \vdots \\ {}^{\hat{\mathsf{p}}_N}\boldsymbol{z}_N \end{bmatrix}}_{\boldsymbol{Z}} = \boldsymbol{V}_0 + \boldsymbol{V}\boldsymbol{P} + \underset{k=1\dots N}{\text{diag}} (\boldsymbol{R}_{\hat{\mathsf{p}}_k})\boldsymbol{Z} \\ = \boldsymbol{1}_N \otimes \boldsymbol{p}_0 + (\boldsymbol{I}_R \otimes \boldsymbol{I}) \underset{k=0\dots M-1}{\text{diag}} (\boldsymbol{R}_{\mathsf{p}_k})\boldsymbol{P} + \underset{k=1\dots N}{\text{diag}} (\boldsymbol{R}_{\hat{\mathsf{p}}_k})\boldsymbol{Z} \tag{2.9}$$

2.2 Model of the system

The model has the following form

$$\hat{\boldsymbol{c}}_{k+1} = \boldsymbol{A}\hat{\boldsymbol{c}}_k + \boldsymbol{B}\boldsymbol{z}_{k+1} \tag{2.10}$$

$$\dot{\boldsymbol{z}}_k = \boldsymbol{D}\hat{\boldsymbol{c}}_k + \boldsymbol{E}\boldsymbol{z}_{k+1}. \tag{2.11}$$

After condensing we obtain

$$\begin{bmatrix} \hat{\boldsymbol{c}}_1 \\ \vdots \\ \hat{\boldsymbol{c}}_N \end{bmatrix} = \boldsymbol{U}_x\hat{\boldsymbol{c}}_0 + \boldsymbol{U}_u \begin{bmatrix} \boldsymbol{z}_1 \\ \vdots \\ \boldsymbol{z}_N \end{bmatrix} \tag{2.12}$$

$$\underbrace{\begin{bmatrix} \dot{\boldsymbol{z}}_0 \\ \vdots \\ \dot{\boldsymbol{z}}_{N-1} \end{bmatrix}}_{\dot{\boldsymbol{Z}}} = \boldsymbol{O}_x\hat{\boldsymbol{c}}_0 + \boldsymbol{O}_u \begin{bmatrix} \boldsymbol{z}_1 \\ \vdots \\ \boldsymbol{z}_N \end{bmatrix} \tag{2.13}$$

$$\tag{2.14}$$

or

$$\hat{\boldsymbol{C}} = \boldsymbol{U}_x\hat{\boldsymbol{c}}_0 + \boldsymbol{U}_u\left(\boldsymbol{V}_0 + \boldsymbol{V}\boldsymbol{P} + \underset{k=1\dots N}{\text{diag}} (\boldsymbol{R}_{\hat{\mathsf{p}}_k})\boldsymbol{Z}\right) \tag{2.15}$$

$$\dot{\boldsymbol{Z}} = \boldsymbol{O}_x\hat{\boldsymbol{c}}_0 + \boldsymbol{O}_u\left(\boldsymbol{V}_0 + \boldsymbol{V}\boldsymbol{P} + \underset{k=1\dots N}{\text{diag}} (\boldsymbol{R}_{\hat{\mathsf{p}}_k})\boldsymbol{Z}\right) \tag{2.16}$$

$$\tag{2.17}$$

The unknowns $\boldsymbol{Z}$ and $\boldsymbol{P}$ can be grouped together

$$\hat{\boldsymbol{C}} = \underbrace{\begin{bmatrix} \boldsymbol{U}_u \underset{k=1\dots N}{\text{diag}} (\boldsymbol{R}_{\hat{\mathsf{p}}_k}) & \boldsymbol{U}_u\boldsymbol{V} \end{bmatrix}}_{\boldsymbol{S}} \underbrace{\begin{bmatrix} \boldsymbol{Z} \\ \boldsymbol{P} \end{bmatrix}}_{\boldsymbol{X}} + \underbrace{\boldsymbol{U}_x\hat{\boldsymbol{c}}_0 + \boldsymbol{U}_u\boldsymbol{V}_0}_{\boldsymbol{s}} \tag{2.18}$$

$$\dot{\boldsymbol{Z}} = \underbrace{\begin{bmatrix} \boldsymbol{O}_u \underset{k=1\dots N}{\text{diag}} (\boldsymbol{R}_{\hat{\mathsf{p}}_k}) & \boldsymbol{O}_u\boldsymbol{V} \end{bmatrix}}_{\boldsymbol{S}_{\dot{z}}} \underbrace{\begin{bmatrix} \boldsymbol{Z} \\ \boldsymbol{P} \end{bmatrix}}_{\boldsymbol{X}} + \underbrace{\boldsymbol{O}_x\hat{\boldsymbol{c}}_0 + \boldsymbol{O}_u\boldsymbol{V}_0}_{\boldsymbol{s}_{\dot{z}}} \tag{2.19}$$

$$\tag{2.20}$$

or

$$\hat{\boldsymbol{C}} = \Big[(\boldsymbol{I} \boxtimes \boldsymbol{U}_u^x) \operatorname*{diag}_{k=1 \dots N} (\boldsymbol{R}_{\hat{\mathsf{p}}_k}) \quad (\boldsymbol{I} \boxtimes \boldsymbol{U}_u^x)(\boldsymbol{I}_R \otimes \boldsymbol{I}) \operatorname*{diag}_{k=0 \dots M-1} (\boldsymbol{R}_{\mathsf{p}_k}) \Big] \boldsymbol{X} \tag{2.21}$$

$$+ (\boldsymbol{I} \boxtimes \boldsymbol{U}_x^x) \hat{\boldsymbol{c}}_0 + (\boldsymbol{I} \boxtimes \boldsymbol{U}_u^x)(\boldsymbol{1}_N \otimes \boldsymbol{p}_0) \tag{2.22}$$

$$= \Big[(\boldsymbol{I} \boxtimes \boldsymbol{U}_u^x) \operatorname*{diag}_{k=1 \dots N} (\boldsymbol{R}_{\hat{\mathsf{p}}_k}) \quad (\boldsymbol{I} \boxtimes (\boldsymbol{U}_u^x \boldsymbol{I}_R)) \operatorname*{diag}_{k=0 \dots M-1} (\boldsymbol{R}_{\mathsf{p}_k}) \Big] \boldsymbol{X} \tag{2.23}$$

$$+ (\boldsymbol{I} \boxtimes \boldsymbol{U}_x^x) \hat{\boldsymbol{c}}_0 + (\boldsymbol{I} \boxtimes (\boldsymbol{U}_u^x \boldsymbol{1}_N)) \boldsymbol{p}_0 \tag{2.24}$$

$$\tag{2.25}$$

and

$$\dot{\boldsymbol{Z}} = \Big[(\boldsymbol{I} \boxtimes \boldsymbol{O}_u^x) \operatorname*{diag}_{k=1 \dots N} (\boldsymbol{R}_{\hat{\mathsf{p}}_k}) \quad (\boldsymbol{I} \boxtimes \boldsymbol{O}_u^x)(\boldsymbol{I}_R \otimes \boldsymbol{I}) \operatorname*{diag}_{k=0 \dots M-1} (\boldsymbol{R}_{\mathsf{p}_k}) \Big] \boldsymbol{X} \tag{2.26}$$

$$+ (\boldsymbol{I} \boxtimes \boldsymbol{U}_x^x) \hat{\boldsymbol{c}}_0 + (\boldsymbol{I} \boxtimes \boldsymbol{O}_u^x)(\boldsymbol{1}_N \otimes \boldsymbol{p}_0) \tag{2.27}$$

$$= \Big[(\boldsymbol{I} \boxtimes \boldsymbol{O}_u^x) \operatorname*{diag}_{k=1 \dots N} (\boldsymbol{R}_{\hat{\mathsf{p}}_k}) \quad (\boldsymbol{I} \boxtimes (\boldsymbol{O}_u^x \boldsymbol{I}_R)) \operatorname*{diag}_{k=0 \dots M-1} (\boldsymbol{R}_{\mathsf{p}_k}) \Big] \boldsymbol{X} \tag{2.28}$$

$$+ (\boldsymbol{I} \boxtimes \boldsymbol{U}_x^x) \hat{\boldsymbol{c}}_0 + (\boldsymbol{I} \boxtimes (\boldsymbol{O}_u^x \boldsymbol{1}_N)) \boldsymbol{p}_0 \tag{2.29}$$

$$\tag{2.30}$$

Velocity of the CoM can be expressed as

$$\dot{\boldsymbol{C}} = \operatorname*{diag}_N(\boldsymbol{I}_v) \hat{\boldsymbol{C}} = \operatorname*{diag}_N(\boldsymbol{I}_v)(\boldsymbol{S}\boldsymbol{X} + \boldsymbol{s}) = \boldsymbol{S}_v \boldsymbol{X} + \boldsymbol{s}_v \tag{2.31}$$

2.3 Constraints

2.3.1 CoP positions

Simple bounds in the case when supports are rectangular.

$$\underline{\boldsymbol{z}}_k \leq {}^{\hat{\mathsf{p}}_k}\boldsymbol{z}_k \leq \bar{\boldsymbol{z}}_k, \quad k = 1 \dots N \tag{2.32}$$

$$\underline{\boldsymbol{Z}} \leq \boldsymbol{Z} \leq \bar{\boldsymbol{Z}} \tag{2.33}$$

2.3.2 Foot positions

Simple bounds when feasible regions are rectangular.

$$\underline{\boldsymbol{p}}_j \leq {}^{\mathsf{p}_{j-1}}\boldsymbol{p}_j \leq \bar{\boldsymbol{p}}_j, \quad j = 1 \dots M \tag{2.34}$$

$$\underline{\boldsymbol{P}} \leq \boldsymbol{P} \leq \bar{\boldsymbol{P}} \tag{2.35}$$

Only initial and final double supports are handled, the respective constraints can also be represented as simple bounds provided that the feet are aligned.

2.4 Objectives

Three objectives are minimized: difference between the actual and reference CoM velocity

$$\left\| \dot{\boldsymbol{C}} - \dot{\boldsymbol{C}}_{ref} \right\|_2 = \left\| \boldsymbol{S}_v \boldsymbol{X} + \boldsymbol{s}_v - \dot{\boldsymbol{C}}_{ref} \right\|_2, \tag{2.36}$$

the CoP velocity

$$\left\|\dot{\boldsymbol{Z}}\right\|_2 = \|\boldsymbol{S}_{\dot{z}}\boldsymbol{X} + \boldsymbol{s}_{\dot{z}}\|_2, \tag{2.37}$$

distance between the CoP positions and the centers of the feet

$$\|\boldsymbol{Z}\|_2. \tag{2.38}$$

2.5 QP

$$\begin{array}{ll} \underset{\boldsymbol{Z},\boldsymbol{P}}{\text{minimize}} & \frac{\alpha}{2}\left\|\dot{\boldsymbol{C}} - \dot{\boldsymbol{C}}_{ref}\right\|^2 + \frac{\beta}{2}\left\|\dot{\boldsymbol{Z}}\right\|^2 + \frac{\gamma}{2}\|\boldsymbol{Z}\|^2 \\ \text{subject to} & \underline{\boldsymbol{Z}} \leq \boldsymbol{Z} \leq \bar{\boldsymbol{Z}} \\ & \underline{\boldsymbol{P}} \leq \boldsymbol{P} \leq \bar{\boldsymbol{P}} \end{array} \tag{2.39}$$

2.5.1 Reference velocity

$$\begin{aligned} &\left\|\dot{\boldsymbol{C}} - \dot{\boldsymbol{C}}_{ref}\right\|^2 = \left\|\boldsymbol{S}_v\boldsymbol{X} + \boldsymbol{s}_v - \dot{\boldsymbol{C}}_{ref}\right\|^2 = \\ &\boldsymbol{X}^T\boldsymbol{S}_v^T\boldsymbol{S}_v\boldsymbol{X} + \boldsymbol{X}^T\boldsymbol{S}_v^T\boldsymbol{s}_v - \boldsymbol{X}^T\boldsymbol{S}_v^T\dot{\boldsymbol{C}}_{ref} \\ &+ \boldsymbol{s}_v^T\boldsymbol{S}_v\boldsymbol{X} + \cancel{\boldsymbol{s}_v^T\boldsymbol{s}_v} - \cancel{\boldsymbol{s}_v^T\dot{\boldsymbol{C}}_{ref}} \\ &- \dot{\boldsymbol{C}}_{ref}^T\boldsymbol{S}_v\boldsymbol{X} - \cancel{\dot{\boldsymbol{C}}_{ref}^T\boldsymbol{s}_v} + \cancel{\dot{\boldsymbol{C}}_{ref}^T\dot{\boldsymbol{C}}_{ref}} \end{aligned} \tag{2.40}$$

Omitting the constant terms we obtain

$$\frac{\alpha}{2}\boldsymbol{X}^T\boldsymbol{S}_v^T\boldsymbol{S}_v\boldsymbol{X} - \alpha\dot{\boldsymbol{C}}_{ref}^T\boldsymbol{S}_v\boldsymbol{X} + \alpha\boldsymbol{s}_v^T\boldsymbol{S}_v\boldsymbol{X} \tag{2.41}$$

2.5.2 CoP velocity

$$\left\|\dot{\boldsymbol{Z}}\right\|^2 = \|\boldsymbol{S}_{\dot{z}}\boldsymbol{X} + \boldsymbol{s}_{\dot{z}}\|^2 = \boldsymbol{X}^T\boldsymbol{S}_{\dot{z}}^T\boldsymbol{S}_{\dot{z}}\boldsymbol{X} + \boldsymbol{X}^T\boldsymbol{S}_{\dot{z}}^T\boldsymbol{s}_{\dot{z}} + \boldsymbol{s}_{\dot{z}}^T\boldsymbol{S}_{\dot{z}}\boldsymbol{X} + \cancel{\boldsymbol{s}_{\dot{z}}^T\boldsymbol{s}_{\dot{z}}} \tag{2.42}$$

Omitting the constant terms we obtain

$$\frac{\beta}{2}\boldsymbol{X}^T\boldsymbol{S}_{\dot{z}}^T\boldsymbol{S}_{\dot{z}}\boldsymbol{X} + \beta\boldsymbol{s}_{\dot{z}}^T\boldsymbol{S}_{\dot{z}}\boldsymbol{X} \tag{2.43}$$

2.5.3 Displacement from the reference CoP

$$\frac{\gamma}{2}\|\boldsymbol{Z}\|^2 = \frac{\gamma}{2}\boldsymbol{Z}^T\boldsymbol{Z} \tag{2.44}$$

2.5.4 Objective in the matrix form

$$\underset{\boldsymbol{X}}{\text{minimize}} \quad \frac{1}{2}\boldsymbol{X}^T\boldsymbol{H}\boldsymbol{X} + \boldsymbol{h}^T\boldsymbol{X} \tag{2.45}$$

$$\boldsymbol{H} = \alpha\boldsymbol{S}_v^T\boldsymbol{S}_v + \beta\boldsymbol{S}_{\dot{z}}^T\boldsymbol{S}_{\dot{z}} + \gamma\begin{bmatrix} 0 & 0 & 0 \\ 0 & \boldsymbol{I} & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.46}$$

$$\boldsymbol{h} = -\alpha\dot{\boldsymbol{C}}_{ref}^T\boldsymbol{S}_v + \alpha\boldsymbol{s}_v^T\boldsymbol{S}_v + \beta\boldsymbol{s}_{\dot{z}}^T\boldsymbol{S}_{\dot{z}} \tag{2.47}$$

2.6 Hierarchical least squares problem

Level 1:

$$\begin{aligned} \underline{\boldsymbol{Z}} \leq \boldsymbol{Z} \leq \bar{\boldsymbol{Z}} \\ \underline{\boldsymbol{P}} \leq \boldsymbol{P} \leq \bar{\boldsymbol{P}} \end{aligned} \tag{2.48}$$

Level 2:

$$\begin{aligned} \sqrt{\frac{\alpha}{2}}\boldsymbol{S}_v\boldsymbol{X} &= \sqrt{\frac{\alpha}{2}}\left(\dot{\boldsymbol{C}}_{ref} - \boldsymbol{s}_v\right) \\ \sqrt{\frac{\beta}{2}}\boldsymbol{S}_{\dot{z}}\boldsymbol{X} &= \sqrt{\frac{\beta}{2}}\boldsymbol{s}_{\dot{z}} \\ \sqrt{\frac{\gamma}{2}}\boldsymbol{Z} &= 0 \end{aligned} \tag{2.49}$$

2.7 Swing foot trajectory

2.7.1 Polynomial and boundary conditions

Trajectory is generated using cubic polynomial of the form

$$at^3 + bt^2 + ct + d = y_{swing}, \tag{2.50}$$

where t is time instance; a, b, c, d are coefficients; and y_{swing} is position of the swing foot at time t.

Derivatives of the cubic polynomial are

$$\begin{aligned} &3at^2 + 2bt + c = \dot{y}_{swing}, \\ &6at + 2b = \ddot{y}_{swing}, \\ &6a = \dddot{y}_{swing}. \end{aligned} \tag{2.51}$$

There are four boundary conditions for the polynomial, the first two are defined for the current swing foot state at $t_i = 0$:

$$\begin{aligned} d &= y_{swing,i}, \\ c &= \dot{y}_{swing,i}, \end{aligned} \tag{2.52}$$

where $y_{swing,i}$ and $\dot{y}_{swing,i}$ are initial position and velocity; the other two conditions for landing time instance t_f are:

$$\begin{aligned} &at_f^3 + bt_f^2 + ct_f + d = y_{swing,f}, \\ &3at^2 + 2bt + c = \dot{y}_{swing,f}, \end{aligned} \tag{2.53}$$

where $y_{swing,f}$ and $\dot{y}_{swing,f}$ are final position and velocity. Trajectories along z axis and x, y axes are computed separately. Final position $y^z_{swing,f}$ for trajectory along z axis is set to the step height during the first half of the support and to 0 during the second half:

$$y^z_{swing,f} = \begin{cases} h_{step} & t \leq \frac{1}{2}T_{support}; \\ 0 & t > \frac{1}{2}T_{support}. \end{cases} \tag{2.54}$$

The final x, y positions are set to the next landing position computed as

$$\begin{bmatrix} y^x_{swing,f} \\ y^y_{swing,f} \end{bmatrix} = \boldsymbol{V}_{land}\boldsymbol{P}, \tag{2.55}$$

where

$$\boldsymbol{V}_{land} = \begin{bmatrix} \boldsymbol{R}_{\mathsf{p}_0} & \boldsymbol{0} \end{bmatrix}. \tag{2.56}$$

Velocity at the end of trajectory is set to zero.

2.7.2 Computation of the desired acceleration

Whole body control requires current acceleration (at time t_i) of the swing foot, which can be found as

$$2b = \ddot{y}_{swing,i}. \tag{2.57}$$

Hence it is necessary to find coefficient b from equations (2.52) and (2.53).

Substitution of (2.52) to (2.53) using $\dot{y}_{swing,f} = 0$ yields

$$\begin{aligned} &at_f^3 + bt_f^2 + \dot{y}_{swing,i}t_f + y_{swing,i} = y_{swing,f}, \\ &3at_f^2 + 2bt_f + \dot{y}_{swing,i} = 0. \end{aligned} \tag{2.58}$$

Trivial algebraic operations on (2.58) and (2.57) lead to the following equation:

$$\ddot{y}_{swing,i} = \frac{6(y_{swing,f} - y_{swing,i})}{t_f^2} - \frac{4\dot{y}_{swing,i}}{t_f} = \frac{6}{t_f^2}y_{swing,f} - \underbrace{\frac{6}{t_f^2}y_{swing,i} - \frac{4}{t_f}\dot{y}_{swing,i}}_{\text{constant}}. \tag{2.59}$$

Consequently

$$\ddot{\boldsymbol{y}}_{swing} = \begin{bmatrix} \ddot{y}^x_{swing,i} \\ \ddot{y}^y_{swing,i} \\ \ddot{y}^z_{swing,i} \end{bmatrix} = \underbrace{\frac{6}{t_f^2}\begin{bmatrix} \boldsymbol{V}_{land} \\ \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{V}_{sa}} \boldsymbol{P} + \underbrace{\frac{6}{t_f^2}\begin{bmatrix} \boldsymbol{0} \\ y^z_{swing,f} \end{bmatrix} - \frac{6}{t_f^2}\boldsymbol{y}_{swing,i} - \frac{4}{t_f}\dot{\boldsymbol{y}}_{swing,i}}_{\boldsymbol{b}_{sa}}, \tag{2.60}$$

2.7.3 Computation of the initial jerk

Jerk at the beginning of the swing foot trajectory characterizes behaviour of the polynomial near this point. High values of the jerk indicate rapid change of acceleration, which in turn increases error in trajectory tracking using piece-wise constant acceleration. Due to this reason we penalize the jerk at the starting point of trajectory. Obviously, only x, y components are penalized, since z component is directly computed.

The jerk can be computed as:

$$\dddot{y}_{swing,i} = -\frac{12(y_{swing,f} - y_{swing,i})}{t_f^3} + \frac{6\dot{y}_{swing,i}}{t_f^2} = -\frac{12}{t_f^3}y_{swing,f} + \underbrace{\frac{12}{t_f^3}y_{swing,i} + \frac{6}{t_f}\dot{y}_{swing,i}}_{\text{constant}}. \tag{2.61}$$

Therefore x, y components of the jerk are expressed as

$$\dddot{\boldsymbol{y}}^{x,y}_{swing} = \begin{bmatrix} \dddot{y}^x_{swing,i} \\ \dddot{y}^y_{swing,i} \end{bmatrix} = \underbrace{-\frac{12}{t_f^3}\boldsymbol{V}_{land}}_{\boldsymbol{V}_{sj}} \boldsymbol{P} + \underbrace{\frac{12}{t_f^3}\boldsymbol{y}_{swing,i} + \frac{6}{t_f^2}\dot{\boldsymbol{y}}_{swing,i}}_{\boldsymbol{b}_{sj}} = \underbrace{\begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{V}_{sj} \end{bmatrix}}_{\boldsymbol{A}_{sj}} \boldsymbol{X}_{mpc} + \boldsymbol{b}_{sj}. \tag{2.62}$$

Bibliography

[1] Wikipedia. *Discretization — Wikipedia, The Free Encyclopedia*. [Online; accessed May-2017]. 2017. URL: https://en.wikipedia.org/w/index.php?title=Discretization.

[2] A. Sherikov. "Balance preservation and task prioritization in whole body motion control of humanoid robots". PhD thesis. Communauté Université Grenoble Alpes, 2016.

[3] A. Herdt, H. Diedam, P. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl. "Online Walking Motion Generation with Automatic Footstep Placement". In: *Advanced Robotics, 24* 5.6 (2010), pp. 719–737.

Part III

Control of Pepper

Chapter 1

MPC controller for Pepper

This module implements MPC problem for Pepper locomotion and balancing as proposed in [1] with minor variations.

1.1 Notation

$\boldsymbol{c}_s$ – CoM position of the base;

$\boldsymbol{c}_d$ – CoM position of the upper body;

$\boldsymbol{p}$ – position of the CoP.

$\hat{\boldsymbol{c}}^x = (c_s^x, \dot{c}_s^x, \ddot{c}_s^x, c_d^x, \dot{c}_d^x, \ddot{c}_d^x)$;

$\hat{\boldsymbol{c}}^y = (c_s^y, \dot{c}_s^y, \ddot{c}_s^y, c_d^y, \dot{c}_d^y, \ddot{c}_d^y)$;

$\hat{\boldsymbol{c}} = (\hat{\boldsymbol{c}}^x, \hat{\boldsymbol{c}}^y)$;

T_k – sampling interval;

m_s – mass of the base;

m_d – mass of the body;

r_s – radius of the base support;

1.2 Support area

In accordance with Pepper's documentation, the distance between the front wheels is $b = 310[mm]$, while the distance distance between the back wheel and a line passing through the front wheels is $h = 260[mm]$.

Hence we can compute radius of the largest circle in the triangle formed by the wheels using

$$r = \frac{b\sqrt{h^2 - b^2/4} - b^2/2}{2h}, \tag{1.1}$$

which gives $r = 88.0496680375479[mm]$. The circle is shown in fig. 1.1.

The distance between "`KneePitch`" joint and the center of the circle is $90 - 88.0496680375479 = 1.95033196245210[mm]$

In the following we are going to assume $r = 70[mm]$, *i.e.*, the safety margin is equal to approximately $2[cm]$ ($18.0496680375479[mm]$).

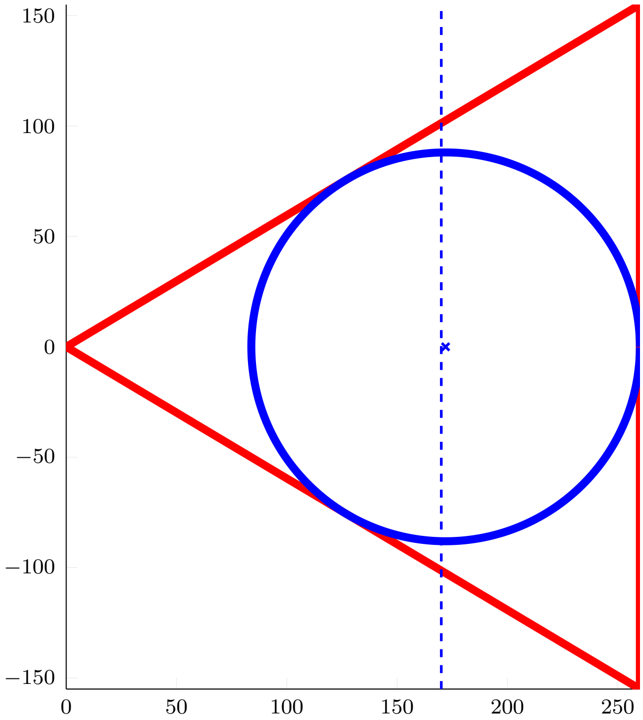

Figure 1.1: Circular support area in the triangle formed by the wheels of Pepper. Front wheels are on the right. Dashed line passes through `KneePitch` joint.

1.3 Base parameters

Parameters of the base can be determined using Pepper's model in `rmt`:

Octave

Listing 1.1

```
ids = [ ...
FRAME_Tibia, FRAME_WheelB_link, FRAME_WheelFL_link, FRAME_WheelFR_link];

bodies_mass = [];
bodies_com  = [];
for i = 1:numel(ids)
    bodies_mass = [bodies_mass, model.DP(ids(i)).m];
    bodies_com = [bodies_com, model.DP(ids(i)).c];
end

total_mass = sum(bodies_mass);
bodies_rel_mass = bodies_mass ./ total_mass;

com_global = bodies_com * bodies_rel_mass';

com_kneepitch = com_global - model.Frame(FRAME_Tibia).p;

com_height = (0.264+0.070) + com_kneepitch(3);
```

The results of the computations are:

$$m_s = 16.34234[kg] \tag{1.2}$$
$$c_s^z = 0.125564931735602[m] \tag{1.3}$$

and the distance from `KneePitch` joint is $0.002531976618098[m]$. Hence, the distance from the center of the circular support area to the base CoM along x axis is $0.002531976618098 * 1000 -$

$1.95033196245210 = 0.581644655645900[mm]$. In the following we assume that this difference is negligible, *i.e.*, the position of the base CoM always coincides with the center of the circular support area.

1.4 Upper body parameters

Parameters of the base can be determined using Pepper's model in `rmt`:

Octave

Listing 1.2

```
ids = [...
FRAME_torso, FRAME_Neck, FRAME_Head, ...
FRAME_Hip, FRAME_Pelvis, ...
FRAME_LShoulder, FRAME_LBicep, FRAME_LElbow, ...
FRAME_LForeArm, FRAME_l_wrist, FRAME_l_gripper, ...
FRAME_LFinger21_link, FRAME_LFinger22_link, FRAME_LFinger23_link, ...
FRAME_LFinger11_link, FRAME_LFinger12_link, FRAME_LFinger13_link, ...
FRAME_LFinger41_link, FRAME_LFinger42_link, FRAME_LFinger43_link, ...
FRAME_LFinger31_link, FRAME_LFinger32_link, ...
FRAME_LFinger33_link, ...
FRAME_LThumb1_link, FRAME_LThumb2_link, ...
FRAME_RShoulder, FRAME_RBicep, FRAME_RElbow, ...
FRAME_RForeArm, FRAME_r_wrist, FRAME_r_gripper, ...
FRAME_RFinger41_link, FRAME_RFinger42_link, FRAME_RFinger43_link, ...
FRAME_RFinger31_link, FRAME_RFinger32_link, FRAME_RFinger33_link, ...
FRAME_RFinger21_link, FRAME_RFinger22_link, FRAME_RFinger23_link, ...
FRAME_RFinger11_link, FRAME_RFinger12_link, FRAME_RFinger13_link, ...
FRAME_RThumb1_link, FRAME_RThumb2_link];


bodies_mass = [];
bodies_com  = [];
for i = 1:numel(ids)
    bodies_mass = [bodies_mass, model.DP(ids(i)).m];
    bodies_com = [bodies_com, model.DP(ids(i)).c];
end

total_mass = sum(bodies_mass);
bodies_rel_mass = bodies_mass ./ total_mass;

com_global = bodies_com * bodies_rel_mass';

com_kneepitch = com_global - model.Frame(FRAME_Tibia).p;

com_height = (0.264+0.070) + com_kneepitch(3);
```

The results of the computations are:

$$m_d = 12.3389[kg] \tag{1.4}$$

$$c_d^z = 0.763104597149514[m] \tag{1.5}$$

and the distance from `KneePitch` joint is $-4.18093905757085e{-}03[m]$. Hence, the distance from the CoM of the base is $0.002531976618098 + 4.18093905757085e{-}03 = 0.00671291567566885[m]$.

1.4.1 Kinematic feasibility

The default body CoM height of $0.763104597149514[m]$ results in a very small kinematic feasibility area of the CoM position. Hence, we set the height to $0.75[m]$.

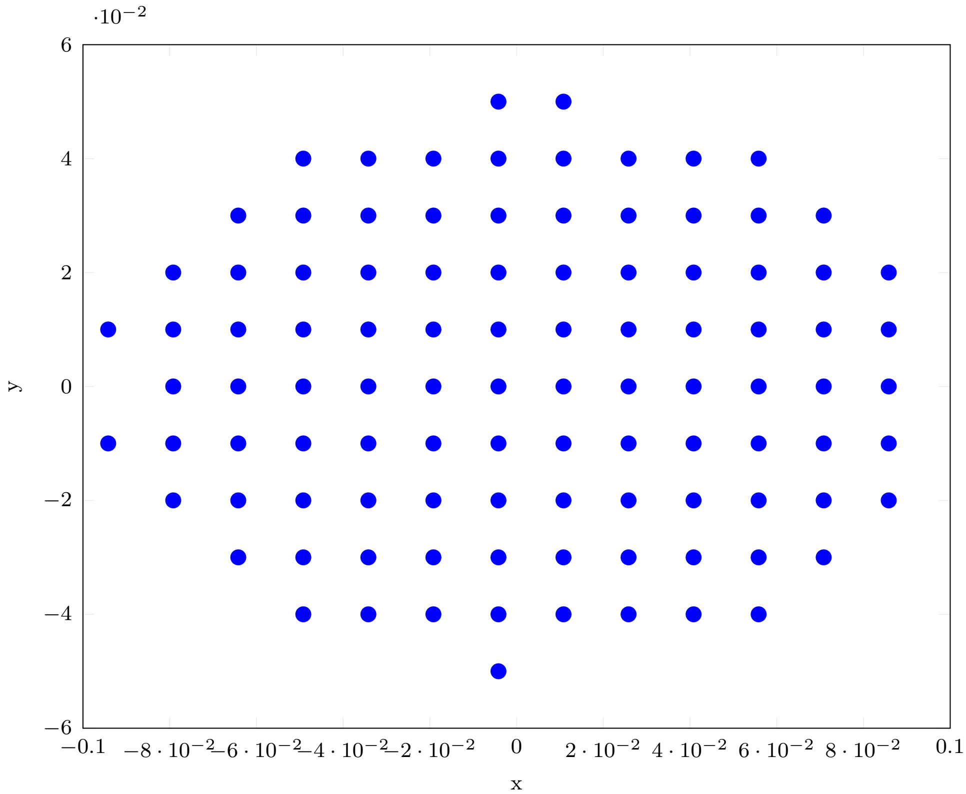

Figure 1.2: Feasible positions of the body CoM at $c_d^z = 0.75[m]$ with fixed arms and head.

1.5 Basic version

Control variables

$$\boldsymbol{u}_k = \dddot{\boldsymbol{c}}_k = (\dddot{\boldsymbol{c}}^x_{s,k}, \dddot{\boldsymbol{c}}^x_{d,k}, \dddot{\boldsymbol{c}}^y_{s,k}, \dddot{\boldsymbol{c}}^y_{d,k}) \tag{1.6}$$

1.5.1 Model

$$\hat{\boldsymbol{c}}_{k+1} = \boldsymbol{A}_k \hat{\boldsymbol{c}}_k + \boldsymbol{B}_k \boldsymbol{u}_k \tag{1.7}$$

$$\boldsymbol{p}_k = \boldsymbol{D}_{p,k} \hat{\boldsymbol{c}}_k \tag{1.8}$$

Four independent triple integrators (x, y motions of base and body)

$$\boldsymbol{A}_k = \operatorname*{diag}_4\left(\begin{bmatrix}1 & T_k & T_k^2/2\\ 0 & 1 & T_k\\ 0 & 0 & 1\end{bmatrix}\right), \quad \boldsymbol{B}_k = \operatorname*{diag}_4\left(\begin{bmatrix}T_k^3/6\\ T_k^2/2\\ T\end{bmatrix}\right) \tag{1.9}$$

$$\boldsymbol{D}_{p,k} = \frac{1}{m_s + m_d}\operatorname*{diag}_2\Big(\Big[m_s\Big[1 \quad 0 \quad -\tfrac{c^z_{s,k}}{g}\Big] \quad m_d\Big[1 \quad 0 \quad -\tfrac{c^z_{d,k}}{g}\Big]\Big]\Big) \tag{1.10}$$

1.5.2 Constraints

1.5.2.1 CoP

Position

$$\|\boldsymbol{p} - \boldsymbol{c}_s\|_2 \leq r_s \tag{1.11}$$

Approximated with a rectangle

$$\underbrace{-\sqrt{2}r_s\begin{bmatrix}1\\1\end{bmatrix}}_{\underline{\boldsymbol{p}}} \leq \boldsymbol{p} - \boldsymbol{c}_s \leq \underbrace{\sqrt{2}r_s\begin{bmatrix}1\\1\end{bmatrix}}_{\bar{\boldsymbol{p}}} \tag{1.12}$$

Over the preview horizon

$$\mathbf{1}_N \otimes \underline{\boldsymbol{p}} \leq \left(\operatorname*{diag}_{k=1\ldots N}(\boldsymbol{D}_{p,k}) - \boldsymbol{S}_{\boldsymbol{c}_s}\right)(\boldsymbol{U}_x\hat{\boldsymbol{c}}_0 + \boldsymbol{U}_u\boldsymbol{\Upsilon}) \leq \mathbf{1}_N \otimes \bar{\boldsymbol{p}} \tag{1.13}$$

1.5.2.2 Base velocity

$$\|\dot{\boldsymbol{c}}_s\|_2 \leq \bar{v} \tag{1.14}$$

Approximated with a rectangle

$$\underbrace{-\sqrt{2}\bar{v}\begin{bmatrix}1\\1\end{bmatrix}}_{\underline{\boldsymbol{v}}} \leq \dot{\boldsymbol{c}}_s \leq \underbrace{\sqrt{2}\bar{v}\begin{bmatrix}1\\1\end{bmatrix}}_{\bar{\boldsymbol{v}}} \tag{1.15}$$

Over the preview horizon

$$\mathbf{1}_N \otimes \underline{\boldsymbol{v}} \leq \boldsymbol{S}_{\dot{\boldsymbol{c}}_s}(\boldsymbol{U}_x\hat{\boldsymbol{c}}_0 + \boldsymbol{U}_u\boldsymbol{\Upsilon}) \leq \mathbf{1}_N \otimes \bar{\boldsymbol{v}} \tag{1.16}$$

1.5.2.3 Base acceleration

$$\|\ddot{\boldsymbol{c}}_s\|_2 \leq \bar{a} \tag{1.17}$$

Approximated with a rectangle

$$\underbrace{-\sqrt{2}\bar{a}\begin{bmatrix}1\\1\end{bmatrix}}_{\underline{\boldsymbol{a}}} \leq \ddot{\boldsymbol{c}}_s \leq \underbrace{\sqrt{2}\bar{a}\begin{bmatrix}1\\1\end{bmatrix}}_{\bar{\boldsymbol{a}}} \tag{1.18}$$

Over the preview horizon

$$\mathbf{1}_N \otimes \underline{\boldsymbol{a}} \leq \boldsymbol{S}_{\ddot{\boldsymbol{c}}_s}(\boldsymbol{U}_x\hat{\boldsymbol{c}}_0 + \boldsymbol{U}_u\boldsymbol{\Upsilon}) \leq \mathbf{1}_N \otimes \bar{\boldsymbol{a}} \tag{1.19}$$

1.5.2.4 Body position

Position with respect to the base

$$\underline{\boldsymbol{d}} \leq {}^{\mathsf{S}}\boldsymbol{R}(\boldsymbol{c}_s - \boldsymbol{c}_d) \leq \bar{\boldsymbol{d}} \tag{1.20}$$

Over the preview horizon

$$\mathbf{1}_N \otimes \underline{\boldsymbol{d}} \leq \operatorname*{diag}_{k=1\ldots N}\left({}^{\mathsf{S}_k}\boldsymbol{R}\right)(\boldsymbol{S}_{\boldsymbol{c}_s} - \boldsymbol{S}_{\boldsymbol{c}_d})(\boldsymbol{U}_x\hat{\boldsymbol{c}}_0 + \boldsymbol{U}_u\boldsymbol{\Upsilon}) \leq \mathbf{1}_N \otimes \bar{\boldsymbol{d}} \tag{1.21}$$

1.5.3 Objective function

1.5.3.1 Base position

$$\boldsymbol{S}_{\boldsymbol{c}_s}(\boldsymbol{U}_x\hat{\boldsymbol{c}}_0 + \boldsymbol{U}_u\boldsymbol{\Upsilon}) - (\boldsymbol{C}_s)_{ref} = \mathbf{0} \tag{1.22}$$

1.5.3.2 Velocity

$$\boldsymbol{S}_{\dot{\boldsymbol{c}}_s}(\boldsymbol{U}_x\hat{\boldsymbol{c}}_0 + \boldsymbol{U}_u\boldsymbol{\Upsilon}) - (\dot{\boldsymbol{C}}_s)_{ref} = \mathbf{0} \tag{1.23}$$

1.5.3.3 Jerk (simple)

$$\boldsymbol{\Upsilon} = \mathbf{0} \tag{1.24}$$

1.5.3.4 CoP

$$\left(\operatorname*{diag}_{k=1\ldots N}(\boldsymbol{D}_{p,k}) - \boldsymbol{S}_{\boldsymbol{c}_s}\right)(\boldsymbol{U}_x\hat{\boldsymbol{c}}_0 + \boldsymbol{U}_u\boldsymbol{\Upsilon}) = \mathbf{0} \tag{1.25}$$

1.5.3.5 Body position

$$(\boldsymbol{S}_{\boldsymbol{c}_s} - \boldsymbol{S}_{\boldsymbol{c}_d})(\boldsymbol{U}_x\hat{\boldsymbol{c}}_0 + \boldsymbol{U}_u\boldsymbol{\Upsilon}) = \mathbf{0} \tag{1.26}$$

1.5.4 Changing output variables

Instead of $\boldsymbol{p}_k$ we can directly compute

$$^{\mathsf{S}_\mathsf{k}}\boldsymbol{p} = \boldsymbol{p}_k - \boldsymbol{c}_{s,k} \tag{1.27}$$

as the output of the system. Since this difference is constrained to a circle (approximated), orientation of the base is not important.

$$\boldsymbol{D}_{p,k} = \frac{1}{m_s + m_d} \operatorname*{diag}_2\left(\begin{bmatrix} m_s \begin{bmatrix} 1 & 0 & -\frac{c^z_{s,k}}{g} \end{bmatrix} & m_d \begin{bmatrix} 1 & 0 & -\frac{c^z_{d,k}}{g} \end{bmatrix} \end{bmatrix}\right) \tag{1.28}$$

$$= \frac{1}{m_s + m_d} \operatorname*{diag}_2\left(\begin{bmatrix} m_s & 0 & -\frac{c^z_{s,k}}{g} m_s & m_d & 0 & -\frac{c^z_{d,k}}{g} m_d \end{bmatrix}\right) \tag{1.29}$$

$$\boldsymbol{S}_{\boldsymbol{c}_s,k} = \operatorname*{diag}_2\left(\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}\right) \tag{1.30}$$

$$= \frac{1}{m_s + m_d} \operatorname*{diag}_2\left(\begin{bmatrix} m_s + m_d & 0 & 0 & 0 & 0 & 0 \end{bmatrix}\right) \tag{1.31}$$

$$\boldsymbol{D}_{ps,k} = \boldsymbol{D}_{p,k} - \boldsymbol{S}_{\boldsymbol{c}_s,k} \tag{1.32}$$

$$= \frac{1}{m_s + m_d} \operatorname*{diag}_2\left(\begin{bmatrix} -m_d & 0 & -\frac{c^z_{s,k}}{g} m_s & m_d & 0 & -\frac{c^z_{d,k}}{g} m_d \end{bmatrix}\right) \tag{1.33}$$

1.6 With simple bounds (Version 1: base velocity)

Control variables

$$\boldsymbol{u}_k = (\dot{\boldsymbol{c}}^x_{s,k+1}, \dddot{\boldsymbol{c}}^x_{d,k}, \dot{\boldsymbol{c}}^y_{s,k+1}, \dddot{\boldsymbol{c}}^y_{d,k}) \tag{1.34}$$

1.6.1 Model

$$\hat{\boldsymbol{c}}_{k+1} = \boldsymbol{A}_k \hat{\boldsymbol{c}}_k + \boldsymbol{B}_k \boldsymbol{u}_k \tag{1.35}$$

$$^{\mathsf{S}_k}\boldsymbol{p}_k = \boldsymbol{D}_{ps,k} \hat{\boldsymbol{c}}_k \tag{1.36}$$

$$\ddot{\boldsymbol{c}}_{s,k} = \boldsymbol{D}_{\ddot{\boldsymbol{c}},k} \hat{\boldsymbol{c}}_k + \boldsymbol{E}_{\ddot{\boldsymbol{c}},k} \boldsymbol{u}_k \tag{1.37}$$

Four independent triple integrators (x, y motions of base and body), triple integrators corresponding to base motion are controlled with velocities.

$$\boldsymbol{A}_k = \operatorname*{diag}_2\left(\operatorname{diag}\left(\begin{bmatrix}1 & \frac{2T_k}{3} & \frac{T_k^2}{6}\\ 0 & 0 & 0\\ 0 & -\frac{2}{T_k} & -1\end{bmatrix}, \begin{bmatrix}1 & T_k & T_k^2/2\\ 0 & 1 & T_k\\ 0 & 0 & 1\end{bmatrix}\right)\right) \tag{1.38}$$

$$\boldsymbol{B}_k = \operatorname*{diag}_2\left(\operatorname{diag}\left(\begin{bmatrix}\frac{T_k}{3}\\ 1\\ \frac{2}{T_k}\end{bmatrix}, \begin{bmatrix}T_k^3/6\\ T_k^2/2\\ T\end{bmatrix}\right)\right) \tag{1.39}$$

$$\boldsymbol{D}_{ps,k} = \frac{1}{m_s + m_d} \operatorname*{diag}_2\left(\begin{bmatrix}-m_d & 0 & -\frac{c^z_{s,k}}{g} m_s & m_d & 0 & -\frac{c^z_{d,k}}{g} m_d\end{bmatrix}\right) \tag{1.40}$$

$$\boldsymbol{D}_{\ddot{\boldsymbol{c}},k} = \operatorname*{diag}_2\left(\begin{bmatrix}\begin{bmatrix}0 & -\frac{2}{T_k^2} & -\frac{2}{T_k}\end{bmatrix} & \begin{bmatrix}0 & 0 & 0\end{bmatrix}\end{bmatrix}\right), \quad \boldsymbol{E}_{\ddot{\boldsymbol{c}},k} = \operatorname*{diag}_2\left(\begin{bmatrix}\begin{bmatrix}\frac{2}{T_k^2}\end{bmatrix} & \begin{bmatrix}0\end{bmatrix}\end{bmatrix}\right), \tag{1.41}$$

1.6.2 Constraints

1.6.2.1 CoP

Position

$$\left\|^{\mathsf{S}}\boldsymbol{p}\right\|_2 \le r_s \tag{1.42}$$

Approximated with a rectangle

$$\underbrace{-\sqrt{2} r_s \begin{bmatrix}1\\1\end{bmatrix}}_{\underline{\boldsymbol{p}}} \le {}^{\mathsf{S}}\boldsymbol{p} \le \underbrace{\sqrt{2} r_s \begin{bmatrix}1\\1\end{bmatrix}}_{\bar{\boldsymbol{p}}} \tag{1.43}$$

Over the preview horizon

$$\boldsymbol{1}_N \otimes \underline{\boldsymbol{p}} \le \operatorname*{diag}_{k=1\ldots N}(\boldsymbol{D}_{ps,k})(\boldsymbol{U}_x \hat{\boldsymbol{c}}_0 + \boldsymbol{U}_u \boldsymbol{\Upsilon}) \le \boldsymbol{1}_N \otimes \bar{\boldsymbol{p}} \tag{1.44}$$

1.6.2.2 Base velocity (simple bounds)

$$\|\dot{\boldsymbol{c}}_s\|_2 \le \bar{v} \tag{1.45}$$

Approximated with a rectangle

$$\underbrace{-\sqrt{2}\bar{v} \begin{bmatrix}1\\1\end{bmatrix}}_{\underline{\boldsymbol{v}}} \le \dot{\boldsymbol{c}}_s \le \underbrace{\sqrt{2}\bar{v} \begin{bmatrix}1\\1\end{bmatrix}}_{\bar{\boldsymbol{v}}} \tag{1.46}$$

Over the preview horizon

$$\boldsymbol{1}_N \otimes \underline{\boldsymbol{v}} \le \boldsymbol{S}_{\dot{\boldsymbol{c}}_s} \boldsymbol{\Upsilon} \le \boldsymbol{1}_N \otimes \bar{\boldsymbol{v}} \tag{1.47}$$

1.6.2.3 Base acceleration

$$\|\ddot{\boldsymbol{c}}_s\|_2 \leq \bar{a} \tag{1.48}$$

Approximated with a rectangle

$$\underbrace{-\sqrt{2}\bar{a}\begin{bmatrix}1\\1\end{bmatrix}}_{\underline{\boldsymbol{a}}} \leq \ddot{\boldsymbol{c}}_s \leq \underbrace{\sqrt{2}\bar{a}\begin{bmatrix}1\\1\end{bmatrix}}_{\bar{\boldsymbol{a}}} \tag{1.49}$$

Over the preview horizon

$$\mathbf{1}_N \otimes \underline{\boldsymbol{a}} \leq \boldsymbol{S}_{\ddot{\boldsymbol{c}}_s}(\boldsymbol{U}_x\hat{\boldsymbol{c}}_0 + \boldsymbol{U}_u\boldsymbol{\Upsilon}) \leq \mathbf{1}_N \otimes \bar{\boldsymbol{a}} \tag{1.50}$$

1.6.2.4 Body position

Position with respect to the base

$$\underline{\boldsymbol{d}} \leq {}^{\mathsf{S}}\boldsymbol{R}(\boldsymbol{c}_s - \boldsymbol{c}_d) \leq \bar{\boldsymbol{d}} \tag{1.51}$$

Over the preview horizon

$$\mathbf{1}_N \otimes \underline{\boldsymbol{d}} \leq \operatorname*{diag}_{k=1\ldots N}\left({}^{\mathsf{S}_k}\boldsymbol{R}\right)(\boldsymbol{S}_{\boldsymbol{c}_s} - \boldsymbol{S}_{\boldsymbol{c}_d})(\boldsymbol{U}_x\hat{\boldsymbol{c}}_0 + \boldsymbol{U}_u\boldsymbol{\Upsilon}) \leq \mathbf{1}_N \otimes \bar{\boldsymbol{d}} \tag{1.52}$$

1.6.3 Objective function

1.6.3.1 Base position

$$\boldsymbol{S}_{\boldsymbol{c}_s}(\boldsymbol{U}_x\hat{\boldsymbol{c}}_0 + \boldsymbol{U}_u\boldsymbol{\Upsilon}) - (\boldsymbol{C}_s)_{ref} = \mathbf{0} \tag{1.53}$$

1.6.3.2 Velocity (simple)

$$\boldsymbol{S}_{\dot{\boldsymbol{c}}_s}\boldsymbol{\Upsilon} - (\dot{\boldsymbol{C}}_s)_{ref} = \mathbf{0} \tag{1.54}$$

1.6.3.3 Jerk (partially simple)

$$\boldsymbol{S}_{\dddot{\boldsymbol{c}}_d}\boldsymbol{\Upsilon} = \mathbf{0} \tag{1.55}$$

$$\boldsymbol{O}_{\dddot{\boldsymbol{c}},x}\hat{\boldsymbol{c}}_0 + \boldsymbol{O}_{\dddot{\boldsymbol{c}},u}\boldsymbol{\Upsilon} = \mathbf{0} \tag{1.56}$$

1.6.3.4 CoP

$$\operatorname*{diag}_{k=1\ldots N}(\boldsymbol{D}_{p,k})(\boldsymbol{U}_x\hat{\boldsymbol{c}}_0 + \boldsymbol{U}_u\boldsymbol{\Upsilon}) = \mathbf{0} \tag{1.57}$$

1.6.3.5 Body position

$$(\boldsymbol{S}_{\boldsymbol{c}_s} - \boldsymbol{S}_{\boldsymbol{c}_d})(\boldsymbol{U}_x\hat{\boldsymbol{c}}_0 + \boldsymbol{U}_u\boldsymbol{\Upsilon}) = \mathbf{0} \tag{1.58}$$

1.7 With simple bounds (Version 1 + sparsity and variable separation)

State variables

$$\hat{\boldsymbol{c}}_s = (c_s^x, \dot{c}_s^x, \ddot{c}_s^x, c_s^y, \dot{c}_s^y, \ddot{c}_s^y) \tag{1.59}$$
$$\hat{\boldsymbol{c}}_d = (c_d^x, \dot{c}_d^x, \ddot{c}_d^x, c_d^y, \dot{c}_d^y, \ddot{c}_d^y) \tag{1.60}$$

Control variables

$$\boldsymbol{u}_{s,k} = (\dot{\boldsymbol{c}}_{s,k+1}^x, \dot{\boldsymbol{c}}_{s,k+1}^y) \tag{1.61}$$
$$\boldsymbol{u}_{d,k} = (\dddot{\boldsymbol{c}}_{d,k}^x, \dddot{\boldsymbol{c}}_{d,k}^y) \tag{1.62}$$

1.7.1 Model

$$\hat{\boldsymbol{c}}_{s,k+1} = \boldsymbol{A}_{s,k}\hat{\boldsymbol{c}}_{s,k} + \boldsymbol{B}_{s,k}\boldsymbol{u}_{s,k} \tag{1.63}$$
$$\hat{\boldsymbol{c}}_{d,k+1} = \boldsymbol{A}_{d,k}\hat{\boldsymbol{c}}_{d,k} + \boldsymbol{B}_{d,k}\boldsymbol{u}_{d,k} \tag{1.64}$$
$$^{\mathsf{S}_k}\boldsymbol{p}_k = \boldsymbol{D}_{ps,s,k}\hat{\boldsymbol{c}}_{s,k} + \boldsymbol{D}_{ps,d,k}\hat{\boldsymbol{c}}_{d,k} \tag{1.65}$$
$$\dddot{\boldsymbol{c}}_{s,k} = \boldsymbol{D}_{\dddot{\boldsymbol{c}},s,k}\hat{\boldsymbol{c}}_k + \boldsymbol{E}_{\dddot{\boldsymbol{c}},s,k}\boldsymbol{u}_{s,k} \tag{1.66}$$

Four independent triple integrators (x, y motions of base and body), triple integrators corresponding to base motion are controlled with velocities.

$$\boldsymbol{A}_{s,k} = \boldsymbol{I}_2 \otimes \boldsymbol{A}_{s,k}^{x|y}, \quad \boldsymbol{A}_{d,k} = \boldsymbol{I}_2 \otimes \boldsymbol{A}_{d,k}^{x|y}, \quad \boldsymbol{B}_{s,k} = \boldsymbol{I}_2 \otimes \boldsymbol{B}_{s,k}^{x|y}, \quad \boldsymbol{B}_{d,k} = \boldsymbol{I}_2 \otimes \boldsymbol{B}_{d,k}^{x|y} \tag{1.67}$$

$$\boldsymbol{A}_{s,k}^{x|y} = \begin{bmatrix} 1 & \frac{2T_k}{3} & \frac{T_k^2}{6} \\ 0 & 0 & 0 \\ 0 & -\frac{2}{T_k} & -1 \end{bmatrix}, \quad \boldsymbol{A}_{d,k}^{x|y} = \begin{bmatrix} 1 & T_k & T_k^2/2 \\ 0 & 1 & T_k \\ 0 & 0 & 1 \end{bmatrix} \tag{1.68}$$

$$\boldsymbol{B}_{s,k}^{x|y} = \begin{bmatrix} \frac{T_k}{3} \\ 1 \\ \frac{2}{T_k} \end{bmatrix}, \quad \boldsymbol{B}_{d,k}^{x|y} = \begin{bmatrix} T_k^3/6 \\ T_k^2/2 \\ T \end{bmatrix} \tag{1.69}$$

$$\boldsymbol{D}_{ps,s,k} = \boldsymbol{I}_2 \otimes \boldsymbol{D}_{ps,s,k}^{x|y} = \boldsymbol{I}_2 \otimes \left(\frac{1}{m_s + m_d} \begin{bmatrix} -m_d & 0 & -\frac{c_{s,k}^z}{g} m_s \end{bmatrix} \right), \tag{1.70}$$
$$\boldsymbol{D}_{ps,d,k} = \boldsymbol{I}_2 \otimes \boldsymbol{D}_{ps,d,k}^{x|y} = \boldsymbol{I}_2 \otimes \left(\frac{1}{m_s + m_d} \begin{bmatrix} m_d & 0 & -\frac{c_{d,k}^z}{g} m_d \end{bmatrix} \right) \tag{1.71}$$

$$\boldsymbol{D}_{\dddot{\boldsymbol{c}},s,k} = \boldsymbol{I}_2 \otimes \boldsymbol{D}_{\dddot{\boldsymbol{c}},s,k}^{x|y} = \boldsymbol{I}_2 \otimes \begin{bmatrix} 0 & -\frac{2}{T_k^2} & -\frac{2}{T_k} \end{bmatrix}, \quad \boldsymbol{E}_{\dddot{\boldsymbol{c}},s,k} = \boldsymbol{I}_2 \otimes \boldsymbol{E}_{\dddot{\boldsymbol{c}},s,k}^{x|y} = \boldsymbol{I}_2 \otimes \begin{bmatrix} \frac{2}{T_k^2} \end{bmatrix} \tag{1.72}$$

1.7.2 Constraints

1.7.2.1 CoP

Position

$$\left\| {}^{\mathsf{S}}\boldsymbol{p} \right\|_2 \leq r_s \tag{1.73}$$

Approximated with a rectangle

$$\underbrace{-\sqrt{2}r_s\begin{bmatrix}1\\1\end{bmatrix}}_{\underline{\boldsymbol{p}}} \leq {}^{\mathsf{S}}\boldsymbol{p} \leq \underbrace{\sqrt{2}r_s\begin{bmatrix}1\\1\end{bmatrix}}_{\bar{\boldsymbol{p}}} \tag{1.74}$$

Over the preview horizon

$$\begin{aligned}\mathbf{1}_N \otimes \underline{\boldsymbol{p}} \leq \underset{k=1\dots N}{\operatorname{diag}}(\boldsymbol{I}_2 \otimes \boldsymbol{D}_{ps,s,k})\big((\boldsymbol{I}_2 \boxtimes \boldsymbol{U}_{s,x}^{x|y})\hat{\boldsymbol{c}}_{s,0} + (\boldsymbol{I}_2 \boxtimes \boldsymbol{U}_{s,u}^{x|y})\boldsymbol{\Upsilon}_s\big) \\ + \underset{k=1\dots N}{\operatorname{diag}}(\boldsymbol{I}_2 \otimes \boldsymbol{D}_{ps,d,k})\Big((\boldsymbol{I}_2 \boxtimes \boldsymbol{U}_{d,x}^{x|y})\hat{\boldsymbol{c}}_{d,0} + (\boldsymbol{I}_2 \boxtimes \boldsymbol{U}_{d,u}^{x|y})\boldsymbol{\Upsilon}_d\Big) \leq \mathbf{1}_N \otimes \bar{\boldsymbol{p}}\end{aligned} \tag{1.75}$$

$$\begin{aligned}\mathbf{1}_N \otimes \underline{\boldsymbol{p}} - \underset{k=1\dots N}{\operatorname{diag}}(\boldsymbol{I}_2 \otimes \boldsymbol{D}_{ps,s,k})(\boldsymbol{I}_2 \boxtimes \boldsymbol{U}_{s,x}^{x|y})\hat{\boldsymbol{c}}_{s,0} - \underset{k=1\dots N}{\operatorname{diag}}(\boldsymbol{I}_2 \otimes \boldsymbol{D}_{ps,d,k})(\boldsymbol{I}_2 \boxtimes \boldsymbol{U}_{d,x}^{x|y})\hat{\boldsymbol{c}}_{d,0} \\ \leq \underset{k=1\dots N}{\operatorname{diag}}(\boldsymbol{I}_2 \otimes \boldsymbol{D}_{ps,s,k})(\boldsymbol{I}_2 \boxtimes \boldsymbol{U}_{s,u}^{x|y})\boldsymbol{\Upsilon}_s + \underset{k=1\dots N}{\operatorname{diag}}(\boldsymbol{D}_{ps,d,k})(\boldsymbol{I}_2 \boxtimes \boldsymbol{U}_{d,u}^{x|y})\boldsymbol{\Upsilon}_d \leq \\ \mathbf{1}_N \otimes \bar{\boldsymbol{p}} - \underset{k=1\dots N}{\operatorname{diag}}(\boldsymbol{I}_2 \otimes \boldsymbol{D}_{ps,s,k})(\boldsymbol{I}_2 \boxtimes \boldsymbol{U}_{s,x}^{x|y})\hat{\boldsymbol{c}}_{s,0} - \underset{k=1\dots N}{\operatorname{diag}}(\boldsymbol{I}_2 \otimes \boldsymbol{D}_{ps,d,k})(\boldsymbol{I}_2 \boxtimes \boldsymbol{U}_{d,x}^{x|y})\hat{\boldsymbol{c}}_{d,0}\end{aligned} \tag{1.76}$$

$$\begin{aligned}\mathbf{1}_N \otimes \underline{\boldsymbol{p}} - \boldsymbol{I}_2 \boxtimes (\underset{k=1\dots N}{\operatorname{diag}}(\boldsymbol{D}_{ps,s,k})\boldsymbol{U}_{s,x}^{x|y})\hat{\boldsymbol{c}}_{s,0} - \boldsymbol{I}_2 \boxtimes (\underset{k=1\dots N}{\operatorname{diag}}(\boldsymbol{D}_{ps,d,k})\boldsymbol{U}_{d,x}^{x|y})\hat{\boldsymbol{c}}_{d,0} \\ \leq \Big[\boldsymbol{I}_2 \boxtimes (\underset{k=1\dots N}{\operatorname{diag}}(\boldsymbol{D}_{ps,s,k})\boldsymbol{U}_{s,u}^{x|y}) \quad \boldsymbol{I}_2 \boxtimes (\underset{k=1\dots N}{\operatorname{diag}}(\boldsymbol{D}_{ps,d,k})\boldsymbol{U}_{d,u}^{x|y})\Big]\boldsymbol{\Upsilon} \leq \\ \mathbf{1}_N \otimes \bar{\boldsymbol{p}} - \boldsymbol{I}_2 \boxtimes (\underset{k=1\dots N}{\operatorname{diag}}(\boldsymbol{D}_{ps,s,k})\boldsymbol{U}_{s,x}^{x|y})\hat{\boldsymbol{c}}_{s,0} - \boldsymbol{I}_2 \boxtimes (\underset{k=1\dots N}{\operatorname{diag}}(\boldsymbol{D}_{ps,d,k})\boldsymbol{U}_{d,x}^{x|y})\hat{\boldsymbol{c}}_{d,0}\end{aligned} \tag{1.77}$$

1.7.2.2 Base velocity (simple bounds)

$$\|\dot{\boldsymbol{c}}_s\|_2 \leq \bar{v} \tag{1.78}$$

Approximated with a rectangle

$$\underbrace{-\sqrt{2}\bar{v}\begin{bmatrix}1\\1\end{bmatrix}}_{\underline{\boldsymbol{v}}} \leq \dot{\boldsymbol{c}}_s \leq \underbrace{\sqrt{2}\bar{v}\begin{bmatrix}1\\1\end{bmatrix}}_{\bar{\boldsymbol{v}}} \tag{1.79}$$

Over the preview horizon

$$\mathbf{1}_N \otimes \underline{\boldsymbol{v}} \leq \boldsymbol{\Upsilon}_s \leq \mathbf{1}_N \otimes \bar{\boldsymbol{v}} \tag{1.80}$$

1.7.2.3 Base acceleration

$$\|\ddot{\boldsymbol{c}}_s\|_2 \leq \bar{a} \tag{1.81}$$

Approximated with a rectangle

$$\underbrace{-\sqrt{2}\bar{a}\begin{bmatrix}1\\1\end{bmatrix}}_{\underline{\boldsymbol{a}}} \leq \ddot{\boldsymbol{c}}_s \leq \underbrace{\sqrt{2}\bar{a}\begin{bmatrix}1\\1\end{bmatrix}}_{\bar{\boldsymbol{a}}} \tag{1.82}$$

Over the preview horizon

$$\mathbf{1}_N \otimes \underline{\boldsymbol{a}} \leq \boldsymbol{S}_{\ddot{c}_s}\big((\boldsymbol{I}_2 \boxtimes \boldsymbol{U}_{s,x}^{x|y})\hat{\boldsymbol{c}}_{s,0} + (\boldsymbol{I}_2 \boxtimes \boldsymbol{U}_{s,u}^{x|y})\boldsymbol{\Upsilon}_s\big) \leq \mathbf{1}_N \otimes \bar{\boldsymbol{a}} \tag{1.83}$$

$$\boldsymbol{S}_{\ddot{\boldsymbol{c}}_s} = \operatorname*{diag}_N\left(\boldsymbol{I}_2 \otimes \begin{bmatrix}0 & 0 & 1\end{bmatrix}\right) = \boldsymbol{I}_2 \boxtimes \boldsymbol{S}^{x|y}_{\ddot{\boldsymbol{c}}_s} \tag{1.84}$$

$$\boldsymbol{1}_N \otimes \underline{\boldsymbol{a}} \leq (\boldsymbol{I}_2 \boxtimes \boldsymbol{S}^{x|y}_{\ddot{\boldsymbol{c}}_s})\left((\boldsymbol{I}_2 \boxtimes \boldsymbol{U}^{x|y}_{s,x})\hat{\boldsymbol{c}}_{s,0} + (\boldsymbol{I}_2 \boxtimes \boldsymbol{U}^{x|y}_{s,u})\boldsymbol{\Upsilon}_s\right) \leq \boldsymbol{1}_N \otimes \bar{\boldsymbol{a}} \tag{1.85}$$

$$\begin{aligned} &\boldsymbol{1}_N \otimes \underline{\boldsymbol{a}} - (\boldsymbol{I}_2 \boxtimes \boldsymbol{S}^{x|y}_{\ddot{\boldsymbol{c}}_s})(\boldsymbol{I}_2 \boxtimes \boldsymbol{U}^{x|y}_{s,x})\hat{\boldsymbol{c}}_{s,0} \\ &\qquad \leq (\boldsymbol{I}_2 \boxtimes \boldsymbol{S}^{x|y}_{\ddot{\boldsymbol{c}}_s})(\boldsymbol{I}_2 \boxtimes \boldsymbol{U}^{x|y}_{s,u})\boldsymbol{\Upsilon}_s \leq \\ &\qquad\qquad \boldsymbol{1}_N \otimes \bar{\boldsymbol{a}} - (\boldsymbol{I}_2 \boxtimes \boldsymbol{S}^{x|y}_{\ddot{\boldsymbol{c}}_s})(\boldsymbol{I}_2 \boxtimes \boldsymbol{U}^{x|y}_{s,x})\hat{\boldsymbol{c}}_{s,0} \end{aligned} \tag{1.86}$$

$$\begin{aligned} &\boldsymbol{1}_N \otimes \underline{\boldsymbol{a}} - \boldsymbol{I}_2 \boxtimes (\boldsymbol{S}^{x|y}_{\ddot{\boldsymbol{c}}_s}\boldsymbol{U}^{x|y}_{s,x})\hat{\boldsymbol{c}}_{s,0} \\ &\qquad \leq \boldsymbol{I}_2 \boxtimes (\boldsymbol{S}^{x|y}_{\ddot{\boldsymbol{c}}_s}\boldsymbol{U}^{x|y}_{s,u})\boldsymbol{\Upsilon}_s \leq \\ &\qquad\qquad \boldsymbol{1}_N \otimes \bar{\boldsymbol{a}} - \boldsymbol{I}_2 \boxtimes (\boldsymbol{S}^{x|y}_{\ddot{\boldsymbol{c}}_s}\boldsymbol{U}^{x|y}_{s,x})\hat{\boldsymbol{c}}_{s,0} \end{aligned} \tag{1.87}$$

1.7.2.4 Body position

Position with respect to the base

$$\underline{\boldsymbol{d}} \leq {}^{\mathsf{S}}\boldsymbol{R}(\boldsymbol{c}_s - \boldsymbol{c}_d) \leq \bar{\boldsymbol{d}} \tag{1.88}$$

Over the preview horizon

$$\begin{aligned} \boldsymbol{1}_N \otimes \underline{\boldsymbol{d}} \leq \operatorname*{diag}_{k=1\ldots N}\left({}^{\mathsf{S_k}}\boldsymbol{R}\right)\Bigg(&\boldsymbol{S}_{\boldsymbol{c}}\left((\boldsymbol{I}_2 \boxtimes \boldsymbol{U}^{x|y}_{s,x})\hat{\boldsymbol{c}}_{s,0} + (\boldsymbol{I}_2 \boxtimes \boldsymbol{U}^{x|y}_{s,u})\boldsymbol{\Upsilon}_s\right) \\ &- \boldsymbol{S}_{\boldsymbol{c}}\Big((\boldsymbol{I}_2 \boxtimes \boldsymbol{U}^{x|y}_{d,x})\hat{\boldsymbol{c}}_{d,0} + (\boldsymbol{I}_2 \boxtimes \boldsymbol{U}^{x|y}_{d,u})\boldsymbol{\Upsilon}_d\Big)\Bigg) \leq \boldsymbol{1}_N \otimes \bar{\boldsymbol{d}} \end{aligned} \tag{1.89}$$

$$\boldsymbol{S}_{\boldsymbol{c}} = \operatorname*{diag}_N\left(\boldsymbol{I}_2 \otimes \begin{bmatrix}1 & 0 & 0\end{bmatrix}\right) = \boldsymbol{I}_2 \boxtimes \boldsymbol{S}^{x|y}_{\boldsymbol{c}} \tag{1.90}$$

$$\begin{aligned} \boldsymbol{1}_N \otimes \underline{\boldsymbol{d}} \leq \operatorname*{diag}_{k=1\ldots N}\left({}^{\mathsf{S_k}}\boldsymbol{R}\right)\Bigg(&\boldsymbol{I}_2 \boxtimes (\boldsymbol{S}^{x|y}_{\boldsymbol{c}}\boldsymbol{U}^{x|y}_{s,x})\hat{\boldsymbol{c}}_{s,0} + \boldsymbol{I}_2 \boxtimes (\boldsymbol{S}^{x|y}_{\boldsymbol{c}}\boldsymbol{U}^{x|y}_{s,u})\boldsymbol{\Upsilon}_s \\ &- \boldsymbol{I}_2 \boxtimes (\boldsymbol{S}^{x|y}_{\boldsymbol{c}}\boldsymbol{U}^{x|y}_{d,x})\hat{\boldsymbol{c}}_{d,0} - \boldsymbol{I}_2 \boxtimes (\boldsymbol{S}^{x|y}_{\boldsymbol{c}}\boldsymbol{U}^{x|y}_{d,u})\boldsymbol{\Upsilon}_d\Bigg) \leq \boldsymbol{1}_N \otimes \bar{\boldsymbol{d}} \end{aligned} \tag{1.91}$$

$$\begin{aligned} &\boldsymbol{1}_N \otimes \underline{\boldsymbol{d}} - \operatorname*{diag}_{k=1\ldots N}\left({}^{\mathsf{S_k}}\boldsymbol{R}\right)\Big(\boldsymbol{I}_2 \boxtimes (\boldsymbol{S}^{x|y}_{\boldsymbol{c}}\boldsymbol{U}^{x|y}_{s,x})\hat{\boldsymbol{c}}_{s,0} - \boldsymbol{I}_2 \boxtimes (\boldsymbol{S}^{x|y}_{\boldsymbol{c}}\boldsymbol{U}^{x|y}_{d,x})\hat{\boldsymbol{c}}_{d,0}\Big) \leq \\ &\quad \left[\operatorname*{diag}_{k=1\ldots N}\left({}^{\mathsf{S_k}}\boldsymbol{R}\right)\Big(\boldsymbol{I}_2 \boxtimes (\boldsymbol{S}^{x|y}_{\boldsymbol{c}}\boldsymbol{U}^{x|y}_{s,u})\Big) \quad - \operatorname*{diag}_{k=1\ldots N}\left({}^{\mathsf{S_k}}\boldsymbol{R}\right)\Big(\boldsymbol{I}_2 \boxtimes (\boldsymbol{S}^{x|y}_{\boldsymbol{c}}\boldsymbol{U}^{x|y}_{d,u})\Big)\right]\boldsymbol{\Upsilon} \\ &\quad\quad \leq \boldsymbol{1}_N \otimes \bar{\boldsymbol{d}} - \operatorname*{diag}_{k=1\ldots N}\left({}^{\mathsf{S_k}}\boldsymbol{R}\right)\Big(\boldsymbol{I}_2 \boxtimes (\boldsymbol{S}^{x|y}_{\boldsymbol{c}}\boldsymbol{U}^{x|y}_{s,x})\hat{\boldsymbol{c}}_{s,0} - \boldsymbol{I}_2 \boxtimes (\boldsymbol{S}^{x|y}_{\boldsymbol{c}}\boldsymbol{U}^{x|y}_{d,x})\hat{\boldsymbol{c}}_{d,0}\Big) \end{aligned} \tag{1.92}$$

1.7.3 Objective function

1.7.3.1 Base position

$$\boldsymbol{I}_2 \boxtimes (\boldsymbol{S}^{x|y}_{\boldsymbol{c}}\boldsymbol{U}^{x|y}_{s,x})\hat{\boldsymbol{c}}_{s,0} + \boldsymbol{I}_2 \boxtimes (\boldsymbol{S}^{x|y}_{\boldsymbol{c}}\boldsymbol{U}^{x|y}_{s,u})\boldsymbol{\Upsilon}_s - (\boldsymbol{C}_s)_{ref} = \boldsymbol{0} \tag{1.93}$$

$$\boldsymbol{I}_2 \boxtimes (\boldsymbol{S}^{x|y}_{\boldsymbol{c}}\boldsymbol{U}^{x|y}_{s,u})\boldsymbol{\Upsilon}_s = (\boldsymbol{C}_s)_{ref} - \boldsymbol{I}_2 \boxtimes (\boldsymbol{S}^{x|y}_{\boldsymbol{c}}\boldsymbol{U}^{x|y}_{s,x})\hat{\boldsymbol{c}}_{s,0} \tag{1.94}$$

1.7.3.2 Velocity (simple)

$$\boldsymbol{\Upsilon}_s - (\dot{\boldsymbol{C}}_s)_{ref} = \boldsymbol{0} \tag{1.95}$$

$$\boldsymbol{\Upsilon}_s = (\dot{\boldsymbol{C}}_s)_{ref} \tag{1.96}$$

1.7.3.3 Jerk (partially simple)

$$\boldsymbol{\Upsilon}_d = \boldsymbol{0} \tag{1.97}$$

$$\boldsymbol{I}_2\boxtimes\boldsymbol{O}_{\dddot{\boldsymbol{c}},s,x}\hat{\boldsymbol{c}}_{s,0} + \boldsymbol{I}_2\boxtimes\boldsymbol{O}_{\dddot{\boldsymbol{c}},s,u}\boldsymbol{\Upsilon}_s = \boldsymbol{0} \tag{1.98}$$

$$\boldsymbol{I}_2\boxtimes\boldsymbol{O}_{\dddot{\boldsymbol{c}},s,u}\boldsymbol{\Upsilon}_s = -\boldsymbol{I}_2\boxtimes\boldsymbol{O}_{\dddot{\boldsymbol{c}},s,x}\hat{\boldsymbol{c}}_{s,0} \tag{1.99}$$

1.7.3.4 CoP

$$\begin{aligned}&\left(\boldsymbol{I}_2\boxtimes \operatorname*{diag}_{k=1\dots N}\left(\boldsymbol{D}^{x|y}_{p,s,k}\right)\right)\left(\boldsymbol{I}_2\boxtimes\boldsymbol{U}^{x|y}_{s,x}\hat{\boldsymbol{c}}_{s,0} + \boldsymbol{I}_2\boxtimes\boldsymbol{U}^{x|y}_{s,u}\boldsymbol{\Upsilon}_s\right)\\ &\quad + \left(\boldsymbol{I}_2\boxtimes \operatorname*{diag}_{k=1\dots N}\left(\boldsymbol{D}^{x|y}_{p,d,k}\right)\right)\left(\boldsymbol{I}_2\boxtimes\boldsymbol{U}^{x|y}_{d,x}\hat{\boldsymbol{c}}_{d,0} + \boldsymbol{I}_2\boxtimes\boldsymbol{U}^{x|y}_{d,u}\boldsymbol{\Upsilon}_d\right) = \boldsymbol{0}\end{aligned} \tag{1.100}$$

$$\begin{aligned}&\boldsymbol{I}_2\boxtimes\left(\operatorname*{diag}_{k=1\dots N}\left(\boldsymbol{D}^{x|y}_{p,s,k}\right)\boldsymbol{U}^{x|y}_{s,x}\right)\hat{\boldsymbol{c}}_{s,0} + \boldsymbol{I}_2\boxtimes\left(\operatorname*{diag}_{k=1\dots N}\left(\boldsymbol{D}^{x|y}_{p,s,k}\right)\boldsymbol{U}^{x|y}_{s,u}\right)\boldsymbol{\Upsilon}_s\\ &\quad + \boldsymbol{I}_2\boxtimes\left(\operatorname*{diag}_{k=1\dots N}\left(\boldsymbol{D}^{x|y}_{p,d,k}\right)\boldsymbol{U}^{x|y}_{d,x}\right)\hat{\boldsymbol{c}}_{d,0} + \boldsymbol{I}_2\boxtimes\left(\operatorname*{diag}_{k=1\dots N}\left(\boldsymbol{D}^{x|y}_{p,d,k}\right)\boldsymbol{U}^{x|y}_{d,u}\right)\boldsymbol{\Upsilon}_d = \boldsymbol{0}\end{aligned} \tag{1.101}$$

$$\begin{aligned}&\left[\boldsymbol{I}_2\boxtimes\left(\operatorname*{diag}_{k=1\dots N}\left(\boldsymbol{D}^{x|y}_{p,s,k}\right)\boldsymbol{U}^{x|y}_{s,u}\right) \quad \boldsymbol{I}_2\boxtimes\left(\operatorname*{diag}_{k=1\dots N}\left(\boldsymbol{D}^{x|y}_{p,d,k}\right)\boldsymbol{U}^{x|y}_{d,u}\right)\right]\boldsymbol{\Upsilon} =\\ &\quad - \boldsymbol{I}_2\boxtimes\left(\operatorname*{diag}_{k=1\dots N}\left(\boldsymbol{D}^{x|y}_{p,s,k}\right)\boldsymbol{U}^{x|y}_{s,x}\right)\hat{\boldsymbol{c}}_{s,0} - \boldsymbol{I}_2\boxtimes\left(\operatorname*{diag}_{k=1\dots N}\left(\boldsymbol{D}^{x|y}_{p,d,k}\right)\boldsymbol{U}^{x|y}_{d,x}\right)\hat{\boldsymbol{c}}_{d,0}\end{aligned} \tag{1.102}$$

1.7.3.5 Body position

$$\begin{aligned}&\left[\left(\boldsymbol{I}_2\boxtimes(\boldsymbol{S}^{x|y}_{\boldsymbol{c}}\boldsymbol{U}^{x|y}_{s,u})\right) \quad -\left(\boldsymbol{I}_2\boxtimes(\boldsymbol{S}^{x|y}_{\boldsymbol{c}}\boldsymbol{U}^{x|y}_{d,u})\right)\right]\boldsymbol{\Upsilon}\\ &\quad = -\boldsymbol{I}_2\boxtimes(\boldsymbol{S}^{x|y}_{\boldsymbol{c}}\boldsymbol{U}^{x|y}_{s,x})\hat{\boldsymbol{c}}_{s,0} + \boldsymbol{I}_2\boxtimes(\boldsymbol{S}^{x|y}_{\boldsymbol{c}}\boldsymbol{U}^{x|y}_{d,x})\hat{\boldsymbol{c}}_{d,0}\end{aligned} \tag{1.103}$$

1.7.4 Optimization problem

Hierarchy (1.1)

1: General constraints

- CoP position

$$\underbrace{\mathbf{1}_N \otimes \underline{\boldsymbol{p}} - \boldsymbol{I}_2 \boxtimes (\operatorname*{diag}_{k=1\ldots N}(\boldsymbol{D}_{ps,s,k})\boldsymbol{U}_{s,x}^{x|y})\hat{\boldsymbol{c}}_{s,0} - \boldsymbol{I}_2 \boxtimes (\operatorname*{diag}_{k=1\ldots N}(\boldsymbol{D}_{ps,d,k})\boldsymbol{U}_{d,x}^{x|y})\hat{\boldsymbol{c}}_{d,0}}_{\underline{\boldsymbol{b}}_p}$$

$$\leq \underbrace{\left[\boldsymbol{I}_2 \boxtimes (\operatorname*{diag}_{k=1\ldots N}(\boldsymbol{D}_{ps,s,k})\boldsymbol{U}_{s,u}^{x|y}) \quad \boldsymbol{I}_2 \boxtimes (\operatorname*{diag}_{k=1\ldots N}(\boldsymbol{D}_{ps,d,k})\boldsymbol{U}_{d,u}^{x|y})\right]}_{\boldsymbol{A}_p} \boldsymbol{\Upsilon} \leq$$

$$\underbrace{\mathbf{1}_N \otimes \bar{\boldsymbol{p}} - \boldsymbol{I}_2 \boxtimes (\operatorname*{diag}_{k=1\ldots N}(\boldsymbol{D}_{ps,s,k})\boldsymbol{U}_{s,x}^{x|y})\hat{\boldsymbol{c}}_{s,0} - \boldsymbol{I}_2 \boxtimes (\operatorname*{diag}_{k=1\ldots N}(\boldsymbol{D}_{ps,d,k})\boldsymbol{U}_{d,x}^{x|y})\hat{\boldsymbol{c}}_{d,0}}_{\bar{\boldsymbol{b}}_p}$$

- Base acceleration

$$\underbrace{\mathbf{1}_N \otimes \underline{\boldsymbol{a}} - \boldsymbol{I}_2 \boxtimes (\boldsymbol{S}_{\ddot{\boldsymbol{c}}_s}^{x|y}\boldsymbol{U}_{s,x}^{x|y})\hat{\boldsymbol{c}}_{s,0}}_{\underline{\boldsymbol{b}}_{\ddot{\boldsymbol{c}}_s}}$$

$$\leq \underbrace{\boldsymbol{I}_2 \boxtimes (\boldsymbol{S}_{\ddot{\boldsymbol{c}}_s}^{x|y}\boldsymbol{U}_{s,u}^{x|y})}_{\boldsymbol{A}_{\ddot{\boldsymbol{c}}_s}} \boldsymbol{\Upsilon}_s \leq$$

$$\underbrace{\mathbf{1}_N \otimes \bar{\boldsymbol{a}} - \boldsymbol{I}_2 \boxtimes (\boldsymbol{S}_{\ddot{\boldsymbol{c}}_s}^{x|y}\boldsymbol{U}_{s,x}^{x|y})\hat{\boldsymbol{c}}_{s,0}}_{\bar{\boldsymbol{b}}_{\ddot{\boldsymbol{c}}_s}}$$

- Body position

$$\underbrace{\mathbf{1}_N \otimes \underline{\boldsymbol{d}} - \operatorname*{diag}_{k=1\ldots N}\left({}^{\mathsf{S_k}}\boldsymbol{R}\right)\left(\boldsymbol{I}_2 \boxtimes (\boldsymbol{S}_{\boldsymbol{c}}^{x|y}\boldsymbol{U}_{s,x}^{x|y})\hat{\boldsymbol{c}}_{s,0} - \boldsymbol{I}_2 \boxtimes (\boldsymbol{S}_{\boldsymbol{c}}^{x|y}\boldsymbol{U}_{d,x}^{x|y})\hat{\boldsymbol{c}}_{d,0}\right)}_{\underline{\boldsymbol{b}}_{\boldsymbol{c}_d}} \leq$$

$$\underbrace{\left[\operatorname*{diag}_{k=1\ldots N}\left({}^{\mathsf{S_k}}\boldsymbol{R}\right)\left(\boldsymbol{I}_2 \boxtimes (\boldsymbol{S}_{\boldsymbol{c}}^{x|y}\boldsymbol{U}_{s,u}^{x|y})\right) \quad -\operatorname*{diag}_{k=1\ldots N}\left({}^{\mathsf{S_k}}\boldsymbol{R}\right)\left(\boldsymbol{I}_2 \boxtimes (\boldsymbol{S}_{\boldsymbol{c}}^{x|y}\boldsymbol{U}_{d,u}^{x|y})\right)\right]}_{\boldsymbol{A}_{\boldsymbol{c}_d}} \boldsymbol{\Upsilon}$$

$$\leq \underbrace{\mathbf{1}_N \otimes \bar{\boldsymbol{d}} - \operatorname*{diag}_{k=1\ldots N}\left({}^{\mathsf{S_k}}\boldsymbol{R}\right)\left(\boldsymbol{I}_2 \boxtimes (\boldsymbol{S}_{\boldsymbol{c}}^{x|y}\boldsymbol{U}_{s,x}^{x|y})\hat{\boldsymbol{c}}_{s,0} - \boldsymbol{I}_2 \boxtimes (\boldsymbol{S}_{\boldsymbol{c}}^{x|y}\boldsymbol{U}_{d,x}^{x|y})\hat{\boldsymbol{c}}_{d,0}\right)}_{\bar{\boldsymbol{b}}_{\boldsymbol{c}_d}}$$

Simple bounds

- $\mathbf{1}_N \otimes \underline{\boldsymbol{v}} \leq \boldsymbol{\Upsilon}_s \leq \mathbf{1}_N \otimes \bar{\boldsymbol{v}}$ Base velocity

2: General objectives

- Base position
$$\underbrace{\boldsymbol{I}_2 \boxtimes (\boldsymbol{S}_{\boldsymbol{c}}^{x|y}\boldsymbol{U}_{s,u}^{x|y})}_{\boldsymbol{A}_{\boldsymbol{c}_s}} \boldsymbol{\Upsilon}_s = \underbrace{(\boldsymbol{C}_s)_{ref} - \boldsymbol{I}_2 \boxtimes (\boldsymbol{S}_{\boldsymbol{c}}^{x|y}\boldsymbol{U}_{s,x}^{x|y})\hat{\boldsymbol{c}}_{s,0}}_{\boldsymbol{b}_{\boldsymbol{c}_s}}$$

- Base jerk
$$\underbrace{\boldsymbol{I}_2 \boxtimes \boldsymbol{O}_{\dddot{\boldsymbol{c}},s,u}\boldsymbol{\Upsilon}_s}_{\boldsymbol{A}_{\dddot{\boldsymbol{c}}_s}} = \underbrace{-\boldsymbol{I}_2 \boxtimes \boldsymbol{O}_{\dddot{\boldsymbol{c}},s,x}\hat{\boldsymbol{c}}_{s,0}}_{\boldsymbol{b}_{\dddot{\boldsymbol{c}}_s}}$$

- CoP centering

$$\underbrace{\left[\boldsymbol{I}_2 \boxtimes \left(\operatorname*{diag}_{k=1\ldots N}\left(\boldsymbol{D}^{x|y}_{p,s,k}\right)\boldsymbol{U}^{x|y}_{s,u}\right) \quad \boldsymbol{I}_2 \boxtimes \left(\operatorname*{diag}_{k=1\ldots N}\left(\boldsymbol{D}^{x|y}_{p,d,k}\right)\boldsymbol{U}^{x|y}_{d,u}\right)\right]}_{\boldsymbol{A}_p} \boldsymbol{\Upsilon} =$$

$$\underbrace{-\boldsymbol{I}_2 \boxtimes \left(\operatorname*{diag}_{k=1\ldots N}\left(\boldsymbol{D}^{x|y}_{p,s,k}\right)\boldsymbol{U}^{x|y}_{s,x}\right)\hat{\boldsymbol{c}}_{s,0} - \boldsymbol{I}_2 \boxtimes \left(\operatorname*{diag}_{k=1\ldots N}\left(\boldsymbol{D}^{x|y}_{p,d,k}\right)\boldsymbol{U}^{x|y}_{d,x}\right)\hat{\boldsymbol{c}}_{d,0}}_{\boldsymbol{b}_p}$$

- Body position

$$\underbrace{\left[\left(\boldsymbol{I}_2 \boxtimes (\boldsymbol{S}^{x|y}_{\boldsymbol{c}} \boldsymbol{U}^{x|y}_{s,u})\right) \quad -\left(\boldsymbol{I}_2 \boxtimes (\boldsymbol{S}^{x|y}_{\boldsymbol{c}} \boldsymbol{U}^{x|y}_{d,u})\right)\right]}_{\boldsymbol{A}_{\boldsymbol{c}_d}} \boldsymbol{\Upsilon}$$

$$= \underbrace{-\boldsymbol{I}_2 \boxtimes (\boldsymbol{S}^{x|y}_{\boldsymbol{c}} \boldsymbol{U}^{x|y}_{s,x})\hat{\boldsymbol{c}}_{s,0} + \boldsymbol{I}_2 \boxtimes (\boldsymbol{S}^{x|y}_{\boldsymbol{c}} \boldsymbol{U}^{x|y}_{d,x})\hat{\boldsymbol{c}}_{d,0}}_{\boldsymbol{b}_{\boldsymbol{c}_d}}$$

Simple objectives

- $\boldsymbol{\Upsilon}_s = (\dot{\boldsymbol{C}}_s)_{ref}$ Base velocity
- $\boldsymbol{\Upsilon}_d = \boldsymbol{0}$ Body jerk

Decision variables: $\boldsymbol{\Upsilon} = (\boldsymbol{\Upsilon}_s, \boldsymbol{\Upsilon}_d)$

1.8 With simple bounds (Version 2: base velocity and body position)

Warning: This version was rejected due to poor conditioning.

1.8.1 Model

1.8.1.1 Intermediate step 1

Control variables

$$\boldsymbol{u}_k = (\dot{\boldsymbol{c}}^x_{s,k+1}, \boldsymbol{c}^x_{d,k+1}, \dot{\boldsymbol{c}}^y_{s,k+1}, \boldsymbol{c}^y_{d,k+1}) \tag{1.104}$$

$$\hat{\boldsymbol{c}}_{k+1} = \boldsymbol{A}_k \hat{\boldsymbol{c}}_k + \boldsymbol{B}_k \boldsymbol{u}_k \tag{1.105}$$

$${}^{\mathsf{S}_k}\boldsymbol{p}_k = \boldsymbol{D}_{ps,k} \hat{\boldsymbol{c}}_k \tag{1.106}$$

$$\dddot{\boldsymbol{c}}_k = \boldsymbol{D}_{\dddot{c},k} \hat{\boldsymbol{c}}_k + \boldsymbol{E}_{\dddot{c},k} \boldsymbol{u}_k \tag{1.107}$$

$$\boldsymbol{A}_k = \operatorname*{diag}_2\left(\operatorname{diag}\left(\begin{bmatrix} 1 & \frac{2T_k}{3} & \frac{T_k^2}{6} \\ 0 & 0 & 0 \\ 0 & -\frac{2}{T_k} & -1 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ -\frac{3}{T_k} & -2 & -\frac{T_k}{2} \\ -\frac{6}{T_k^2} & -\frac{6}{T_k} & -2 \end{bmatrix}\right)\right) \tag{1.108}$$

$$\boldsymbol{B}_k = \operatorname*{diag}_2\left(\operatorname{diag}\left(\begin{bmatrix} \frac{T_k}{3} \\ 1 \\ \frac{2}{T_k} \end{bmatrix}, \begin{bmatrix} 1 \\ \frac{3}{T_k} \\ \frac{6}{T_k^2} \end{bmatrix}\right)\right) \tag{1.109}$$

$$\boldsymbol{D}_{\dddot{c},k} = \operatorname*{diag}_2\left(\operatorname{diag}\left(\begin{bmatrix} 0 & -\frac{2}{T_k^2} & -\frac{2}{T_k} \end{bmatrix}, \begin{bmatrix} -\frac{6}{T_k^3} & -\frac{6}{T_k^2} & -\frac{3}{T_k} \end{bmatrix}\right)\right) \tag{1.110}$$

$$\boldsymbol{E}_{\dddot{c},k} = \operatorname*{diag}_2\left(\operatorname{diag}\left(\begin{bmatrix} \frac{2}{T_k^2} \end{bmatrix}, \begin{bmatrix} \frac{6}{T_k^3} \end{bmatrix}\right)\right), \tag{1.111}$$

1.8.1.2 Intermediate step 2

Control variables

$$\boldsymbol{u}_k = (\dot{\boldsymbol{c}}^x_{s,k+1}, {}^{\mathsf{S}}\boldsymbol{c}^x_{d,k+1}, \dot{\boldsymbol{c}}^y_{s,k+1}, {}^{\mathsf{S}}\boldsymbol{c}^y_{d,k+1}), \tag{1.112}$$

where

$$\begin{bmatrix} {}^{\mathsf{S}}\boldsymbol{c}^x_{d,k+1} \\ {}^{\mathsf{S}}\boldsymbol{c}^y_{d,k+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{c}^x_{d,k+1} \\ \boldsymbol{c}^y_{d,k+1} \end{bmatrix} - \begin{bmatrix} \boldsymbol{c}^x_{s,k+1} \\ \boldsymbol{c}^y_{s,k+1} \end{bmatrix} \tag{1.113}$$

Maxima

Listing 1.3

```
A: matrix([1, T, T^2/2], [0, 1, T], [0, 0, 1]);
B: matrix([T^3/6], [T^2/2], [T]);
X: matrix([x],[dx],[ddx]);
U: matrix([dddx]);
Xp: matrix([xp],[dxp],[ddxp]);


e: solve([(A.X + B.U)[2][1] = (Xp)[2][1]], [dddx]);
Ds: coefmatrix([rhs(e[1])], list_matrix_entries(X));
```

```
Es: coefmatrix([rhs(e[1])], [dxp]);
e: subst(e, A.X + B.U);
As: coefmatrix(list_matrix_entries(e), list_matrix_entries(X));
Bs: coefmatrix(list_matrix_entries(e), [dxp]);

e: solve([(A.X + B.U)[1][1] = (Xp)[1][1]], [dddx]);
Dd: coefmatrix([rhs(e[1])], list_matrix_entries(X));
Ed: coefmatrix([rhs(e[1])], [xp]);
e: subst(e, A.X + B.U);
Ad: coefmatrix(list_matrix_entries(e), list_matrix_entries(X));
Bd: coefmatrix(list_matrix_entries(e), [xp]);

Xs: matrix([xs],[dxs],[ddxs]);
Xd: matrix([xd],[dxd],[ddxd]);
Xsp: matrix([xsp],[dxsp],[ddxsp]);
Xdp: matrix([xdp],[dxdp],[ddxdp]);

e: xsp = (As[1].Xs + Bs[1]*dxsp)[1];
e1: subst(e, Ad.Xd + Bd*(xsp + xdp));
e2: subst(e, Dd.Xd + Ed*(xsp + xdp));
Anew: coefmatrix(list_matrix_entries(e1),
    append(list_matrix_entries(Xs),list_matrix_entries(Xd)));
Bnew: coefmatrix(list_matrix_entries(e1), [dxsp, xdp]);
Dnew: coefmatrix(list_matrix_entries(e2),
    append(list_matrix_entries(Xs),list_matrix_entries(Xd)));
Enew: coefmatrix(list_matrix_entries(e2), [dxsp, xdp]);

tex(Anew);
tex(Bnew);
tex(Dnew);
tex(Enew);
```

$$\hat{\boldsymbol{c}}_{k+1} = \boldsymbol{A}_k \hat{\boldsymbol{c}}_k + \boldsymbol{B}_k \boldsymbol{u}_k \tag{1.114}$$

$$^{\mathsf{S}_k}\boldsymbol{p}_k = \boldsymbol{D}_{ps,k} \hat{\boldsymbol{c}}_k \tag{1.115}$$

$$\dddot{\boldsymbol{c}}_k = \boldsymbol{D}_{\dddot{\boldsymbol{c}},k} \hat{\boldsymbol{c}}_k + \boldsymbol{E}_{\dddot{\boldsymbol{c}},k} \boldsymbol{u}_k \tag{1.116}$$

$$\boldsymbol{A}_k = \underset{2}{\operatorname{diag}}\left(\begin{bmatrix} 1 & \frac{2T_k}{3} & \frac{T_k^2}{6} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{2}{T_k} & -1 & 0 & 0 & 0 \\ 1 & \frac{2T_k}{3} & \frac{T_k^2}{6} & 0 & 0 & 0 \\ \frac{3}{T_k} & 2 & \frac{T_k}{2} & -\frac{3}{T_k} & -2 & -\frac{T_k}{2} \\ \frac{6}{T_k^2} & \frac{4}{T_k} & 1 & -\frac{6}{T_k^2} & -\frac{6}{T_k} & -2 \end{bmatrix}\right), \quad \boldsymbol{B}_k = \underset{2}{\operatorname{diag}}\left(\begin{bmatrix} \frac{T_k}{3} & 0 \\ 1 & 0 \\ \frac{2}{T_k} & 0 \\ \frac{T_k}{3} & 1 \\ 1 & \frac{3}{T_k} \\ \frac{2}{T_k} & \frac{6}{T_k^2} \end{bmatrix}\right) \tag{1.117}$$

$$\boldsymbol{D}_{\dddot{\boldsymbol{c}},k} = \underset{2}{\operatorname{diag}}\left(\begin{bmatrix} 0 & -\frac{2}{T_k^2} & -\frac{2}{T_k} & 0 & 0 & 0 \\ \frac{6}{T_k^3} & \frac{4}{T_k^2} & \frac{1}{T_k} & -\frac{6}{T_k^3} & -\frac{6}{T_k^2} & -\frac{3}{T_k} \end{bmatrix}\right), \quad \boldsymbol{E}_{\dddot{\boldsymbol{c}},k} = \underset{2}{\operatorname{diag}}\left(\begin{bmatrix} \frac{2}{T_k^2} & 0 \\ \frac{2}{T_k^2} & \frac{6}{T_k^3} \end{bmatrix}\right) \tag{1.118}$$

1.8.1.3 Final

Control variables

$$\boldsymbol{u}_k = (\dot{\boldsymbol{c}}^x_{s,k+1}, {}^{\mathsf{S}}\boldsymbol{c}^x_{d,k+1}, \dot{\boldsymbol{c}}^y_{s,k+1}, {}^{\mathsf{S}}\boldsymbol{c}^y_{d,k+1}), \tag{1.119}$$

where

$$\begin{bmatrix} {}^{\mathsf{S}}\boldsymbol{c}^x_{d,k+1} \\ {}^{\mathsf{S}}\boldsymbol{c}^y_{d,k+1} \end{bmatrix} = {}^{\mathsf{S}_{k+1}}\boldsymbol{R}\left(\begin{bmatrix} \boldsymbol{c}^x_{d,k+1} \\ \boldsymbol{c}^y_{d,k+1} \end{bmatrix} - \begin{bmatrix} \boldsymbol{c}^x_{s,k+1} \\ \boldsymbol{c}^y_{s,k+1} \end{bmatrix}\right) \tag{1.120}$$

Maxima

Listing 1.4

```
SR: zeromatrix (4, 4);
SR[1][2]: 1;
SR[2][4]: 1;
SR[3][1]: 1;
SR[4][3]: 1;
R: matrix([cos(t), -sin(t)], [sin(t), cos(t)]);
SR: transpose(SR).diag([R,ident(2)]).SR;
tex(SR);
```

$$\hat{\boldsymbol{c}}_{k+1} = \boldsymbol{A}_k\hat{\boldsymbol{c}}_k + \boldsymbol{B}_k\boldsymbol{S}_{\boldsymbol{R}_{\mathsf{S}_{k+1}}}\boldsymbol{u}_k \tag{1.121}$$

$${}^{\mathsf{S}_k}\boldsymbol{p}_k = \boldsymbol{D}_{ps,k}\hat{\boldsymbol{c}}_k \tag{1.122}$$

$$\dddot{\boldsymbol{c}}_k = \boldsymbol{D}_{\dddot{\boldsymbol{c}},k}\hat{\boldsymbol{c}}_k + \boldsymbol{E}_{\dddot{\boldsymbol{c}},k}\boldsymbol{S}_{\boldsymbol{R}_{\mathsf{S}_{k+1}}}\boldsymbol{u}_k \tag{1.123}$$

$$\boldsymbol{A}_k = \operatorname*{diag}_2\left(\begin{bmatrix} 1 & \frac{2T_k}{3} & \frac{T_k^2}{6} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{2}{T_k} & -1 & 0 & 0 & 0 \\ 1 & \frac{2T_k}{3} & \frac{T_k^2}{6} & 0 & 0 & 0 \\ \frac{3}{T_k} & 2 & \frac{T_k}{2} & -\frac{3}{T_k} & -2 & -\frac{T_k}{2} \\ \frac{6}{T_k^2} & \frac{4}{T_k} & 1 & -\frac{6}{T_k^2} & -\frac{6}{T_k} & -2 \end{bmatrix}\right), \quad \boldsymbol{B}_k = \operatorname*{diag}_2\left(\begin{bmatrix} \frac{T_k}{3} & 0 \\ 1 & 0 \\ \frac{2}{T_k} & 0 \\ \frac{T_k}{3} & 1 \\ 1 & \frac{3}{T_k} \\ \frac{2}{T_k} & \frac{6}{T_k^2} \end{bmatrix}\right) \tag{1.124}$$

$$\boldsymbol{D}_{ps,k} = \frac{1}{m_s + m_d}\operatorname*{diag}_2\left(\begin{bmatrix} -m_d & 0 & -\frac{c^z_{s,k}}{g}m_s & m_d & 0 & -\frac{c^z_{d,k}}{g}m_d \end{bmatrix}\right) \tag{1.125}$$

$$\boldsymbol{D}_{\dddot{\boldsymbol{c}},k} = \operatorname*{diag}_2\left(\begin{bmatrix} 0 & -\frac{2}{T_k^2} & -\frac{2}{T_k} & 0 & 0 & 0 \\ \frac{6}{T_k^3} & \frac{4}{T_k^2} & \frac{1}{T_k} & -\frac{6}{T_k^3} & -\frac{6}{T_k^2} & -\frac{3}{T_k} \end{bmatrix}\right), \quad \boldsymbol{E}_{\dddot{\boldsymbol{c}},k} = \operatorname*{diag}_2\left(\begin{bmatrix} \frac{2}{T_k^2} & 0 \\ \frac{2}{T_k^2} & \frac{6}{T_k^3} \end{bmatrix}\right) \tag{1.126}$$

$$\boldsymbol{S}_{\boldsymbol{R}_{\mathsf{S}_{k+1}}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta_{k+1}) & 0 & -\sin(\theta_{k+1}) \\ 0 & 0 & 1 & 0 \\ 0 & \sin(\theta_{k+1}) & 0 & \cos(\theta_{k+1}) \end{bmatrix} \tag{1.127}$$

1.9 With simple bounds (Version 3: base velocity and CoP)

Warning: This version was rejected due to poor conditioning.

1.9.1 Model

1.9.1.1 Intermediate step 1

Control variables

$$\boldsymbol{u}_k = (\dot{\boldsymbol{c}}^x_{s,k+1}, {}^{\mathsf{S}_k}\boldsymbol{p}^x_{k+1}, \dot{\boldsymbol{c}}^y_{s,k+1}, {}^{\mathsf{S}_k}\boldsymbol{p}^y_{k+1}) \tag{1.128}$$

$$\hat{\boldsymbol{c}}_{k+1} = \boldsymbol{A}_k \hat{\boldsymbol{c}}_k + \boldsymbol{B}_k \boldsymbol{u}_k \tag{1.129}$$

$$\dddot{\boldsymbol{c}}_{s,k} = \boldsymbol{D}_{\dddot{c},k} \hat{\boldsymbol{c}}_k + \boldsymbol{E}_{\dddot{c},k} \boldsymbol{u}_k \tag{1.130}$$

Maxima

Listing 1.5

```
A: matrix([1, T, T^2/2], [0, 1, T], [0, 0, 1]);
B: matrix([T^3/6], [T^2/2], [T]);
X: matrix([x],[dx],[ddx]);
U: matrix([dddx]);
Xp: matrix([xp],[dxp],[ddxp]);

e: solve([(A.X + B.U)[2][1] = (Xp)[2][1]], [dddx]);
Ds: coefmatrix([rhs(e[1])], list_matrix_entries(X));
Es: coefmatrix([rhs(e[1])], [dxp]);
e: subst(e, A.X + B.U);
As: coefmatrix(list_matrix_entries(e), list_matrix_entries(X));
Bs: coefmatrix(list_matrix_entries(e), [dxp]);

Ad: A;
Bd: B;

Dpss: 1/(ms+md) * matrix([-md, 0, -czs/g*ms]);
Dpsd: 1/(ms+md) * matrix([md, 0, -czd/g*md]);

Xs: matrix([xs],[dxs],[ddxs]);
Xd: matrix([xd],[dxd],[ddxd]);
Us: matrix([dxps]);
Ud: matrix([dddxd]);
e: solve([z = Dpss.(As.Xs + Bs.Us) + Dpsd.(Ad.Xd + Bd.Ud)], [dddxd]);
e: subst(e, Ad.Xd + Bd.Ud);
Adnew: coefmatrix(list_matrix_entries(e),
    append(list_matrix_entries(Xs), list_matrix_entries(Xd)));
Bdnew: coefmatrix(list_matrix_entries(e), [dxps, z]);
AA: addrow(addcol(As, zeromatrix (3,3)), Adnew);
AAA: subst([T=0.01, g=9.8, czd=0.8, czs=0.1, md=30, ms=30], AA);
```

Chapter 2

Inverse kinematics controller for Pepper

2.1 Computation of wheel velocities

Motion of the base in the inverse kinematics controller is represented with translational and angular velocities. In order to execute a motion it is necessary to map the corresponding velocities to wheel velocities, which are the inputs of the wheel controllers of Pepper.

We assume that the z axis of the global frame is normal to the contact surface, and there is a frame s fixed to the base so that its x and y axes span the contact surface and its origin is the center of rotation. Then we can represent rotation in the x-y plane from the global frame to the base frame with ${}^{\mathsf{s}}\boldsymbol{R} \in \mathbb{R}^{2\times 2}$.

Let $\boldsymbol{v}_s = (\dot{\boldsymbol{c}}_s^{xy}, w_s^z)$ be the three-dimensional vector of base velocity composed of translational velocity in x-y plane and angular velocity about axis z. In the following we derive matrix $\boldsymbol{T}$ such that $\boldsymbol{v}_w = \boldsymbol{T}\,{}^{\mathsf{s}}\boldsymbol{v}_s$, where ${}^{\mathsf{s}}\boldsymbol{v}_s = ({}^{\mathsf{s}}\boldsymbol{R}\dot{\boldsymbol{c}}_s^{xy}, w_s^z)$ and $\boldsymbol{v}_w$ is a three-dimensional vector composed of scalar angular velocities of the wheels.

In order to compute $\boldsymbol{T}$ we need to know the following parameters

${}^{\mathsf{s}}\boldsymbol{d}_i$	$\in \mathbb{R}^3$	distance to the ground contact point of the i-th wheel in the base frame ($d_i^z = 0$)
r_i	$\in \mathbb{R}$	radius of the i-th wheel
θ_i	$\in \mathbb{R}$	angle about z axis representing orientation of the i-th wheel in the base frame

Also, by definition, the translational velocity $\boldsymbol{v}$ at distance $\boldsymbol{r}$ from the center of rotation is computed as $\boldsymbol{v} = \boldsymbol{w} \times \boldsymbol{r}$, which corresponds to scalar relation $v = wr$ when $\boldsymbol{w} \perp \boldsymbol{r}$.

Translational velocity at the wheel contact point in the base frame can be computed as

$$
{}^{\mathsf{s}}\boldsymbol{v}_i = \begin{bmatrix} {}^{\mathsf{s}}\boldsymbol{R}\dot{\boldsymbol{c}}_s^{xy} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ w_s^z \end{bmatrix} \times {}^{\mathsf{s}}\boldsymbol{d}_i = \begin{bmatrix} \begin{matrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{matrix} & -{}^{\mathsf{s}}\boldsymbol{d}_i \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{bmatrix} \begin{bmatrix} {}^{\mathsf{s}}\boldsymbol{R}\dot{\boldsymbol{c}}_s^{xy} \\ w_s^z \end{bmatrix}. \tag{2.1}
$$

We have to project ${}^{\mathsf{s}}\boldsymbol{v}_i$ on the normal of this axis ${}^{\mathsf{s}}\boldsymbol{n}_i$, which is equal to the negative second column of rotation matrix computed based on θ_i. Hence, the velocity of the i-th wheel is

$$
v_i = w_i r_i = {}^{\mathsf{s}}\boldsymbol{n}_i^\top \begin{bmatrix} \begin{matrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{matrix} & -{}^{\mathsf{s}}\boldsymbol{d}_i \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{bmatrix} {}^{\mathsf{s}}\boldsymbol{v}_s = \begin{bmatrix} ({}^{\mathsf{s}}\boldsymbol{n}_i^{xy})^\top & -{}^{\mathsf{s}}\boldsymbol{n}_i^\top \left({}^{\mathsf{s}}\boldsymbol{d}_i \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right) \end{bmatrix} {}^{\mathsf{s}}\boldsymbol{v}_s, \tag{2.2}
$$

and the i-th row of matrix $\boldsymbol{T}$ can be deduced from equation

$$
w_i = \frac{1}{r_i} \begin{bmatrix} ({}^{\mathsf{s}}\boldsymbol{n}_i^{xy})^\top & -{}^{\mathsf{s}}\boldsymbol{n}_i^\top \left({}^{\mathsf{s}}\boldsymbol{d}_i \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right) \end{bmatrix} {}^{\mathsf{s}}\boldsymbol{v}_s \tag{2.3}
$$

Bibliography

[1] J. Lafaye, D. Gouaillier, and P.-B. Wieber. “Linear model predictive control of the locomotion of Pepper, a humanoid robot with omnidirectional wheels”. In: *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*. 2014, pp. 336–341.