Sapienza
Università di Roma

Department of Computer, Control and Management Engineering

Real-Time Safe Bipedal Robot Navigation using Linear Discrete Control Barrier Functions

Autonomous and Mobile Robotics

Students:

Eugenio Bugli

Damiano Imola

Salvatore Michele Rago

Academic Year 2024/2025

Contents

1 Introduction

Eugenio:
Humanoid robots are inherently underactuated thanks to to unilateral ground contacts thus a strong coupling exists between path planning and gait control. A path is considered safe if the robot does not collide with any obstacle and its dynamics and physical limitations are respected. Due to the high complexity of humanoids, we cannot decouple path planning from motion control without taking into account the dynamics. We should solve gait optimization problems based on the robot's full order model or the reduced one. Due to computational complexity, reduced order models such the Linear Inverted Pendulum (LIP) are often employed. In our case, since Control Barrier Functions are employed to ensure safety in path planning, we will pre-compute heading angles ans use approximated linead DCBFs. This is needed since we may have problems at computation level due to the non linearity of kinematics and path constraints.

Salvatore:
The term *humanoid* refers to a robot with structure and kinematics similar to a human body. It is designed for locomanipulation and represent the best choice to navigate and interact in an environment that is structured for humans.
Real-time safe navigation is a crucial task for humanoid robots in real-world applications. A path is considered safe if it does not collide with any obstacle while fulfilling the robot's dynamics and physical constraints. In order to carry out such complex task in real-time, path planning is usually decoupled from gait control, resulting in a significant reduction of the computational load.
The aim of this work is to implement the solution proposed by Peng et al. in "Real-Time Safe Bipedal Robot Navigation using Linear Discrete Control Barrier Functions", which consists in a unified safe path and gait planning framework to be executed in real-time. It models the humanoid's walking dynamics by a Linear Inverted Pendulum, and leverages Model Predictive Control and Control Barrier Functions to deliver a collision-free path while satisfying specific constraints.
In the following chapters, we will delve into the details of this approach, discuss the results, and propose some improvements.

2 LIP

Eugenio:
This reduced model assumes that during the motion the Center of Mass (CoM) will have a constant height H.

$$\dot{v}_x = \frac{g}{H}(p_x - f_x) \qquad \dot{v}_y = \frac{g}{H}(p_y - f_y) \tag{1}$$

With (p_x, p_y) we denote the position of the CoM and with (v_x, v_y) its velocity with respect to the x-axis and y-axis. The stance foot position, which is the position in which both feets are in contact with the ground, is denoted with (f_x, f_y).

Given the position (p_{x_k}, p_{y_k}) and velocities (v_{x_k}, v_{y_k}) of the CoM at the k-th step, the closed-form solutions of the step-to-step discrete dynamics can be written as follows :

$$\begin{bmatrix} p_{x_{k+1}} \\ v_{x_{k+1}} \end{bmatrix} = A_d \begin{bmatrix} p_{x_k} \\ v_{x_k} \end{bmatrix} + B_d f_{x_k} \qquad \begin{bmatrix} p_{y_{k+1}} \\ v_{y_{k+1}} \end{bmatrix} = A_d \begin{bmatrix} p_{y_k} \\ v_{y_k} \end{bmatrix} + B_d f_{y_k} \tag{2}$$

Where $\beta = \sqrt{\frac{g}{H}}$ and the two matrices are:

$$A_d = \begin{bmatrix} cosh(\beta T) & \frac{sinh(\beta T)}{\beta} \\ \beta sinh(\beta T) & cosh(\beta T) \end{bmatrix} \qquad B_d = \begin{bmatrix} 1 - cosh(\beta T) & -\beta sinh(\beta T) \end{bmatrix} \tag{3}$$

By defining the state of our system as $x = [p_x, v_x, p_y, v_y, \theta]^T \in \mathbb{R}^5$ and the control input as $u = [f_x, f_y, \omega]^T \in \mathbb{R}^3$, where θ is the heading angle and ω is its turning rate, the step-to-step dynamics of the 3D-LIP model is written as follows:

$$x_{k+1} = A_L x_k + B_L u_k \tag{4}$$

Where the two matrices are defined as follows:

$$A_L = \begin{bmatrix} A_d & 0 & 0 \\ 0 & A_d & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad B_d = \begin{bmatrix} B_d & 0 & 0 \\ 0 & B_d & 0 \\ 0 & 0 & T \end{bmatrix} \tag{5}$$

3 3D-LIP Model with Heading Angles

Salvatore:
If the full dynamic model of the humanoid is used to simulate its motion, it becomes computationally impossible to perform joint path and gait planning, due to its high dimensionality and non-linearity. Therefore, a simplifying model must be used. For this scope Peng et al. introduced the "3D-LIP Model with Heading Angle", which describes the discrete dynamics of the Center of Mass (CoM) similarly to the one of an inverted pendulum in three dimensions.

3.1 Local Robot Reference Frame

The state $\mathbf{x}$ and the input $\mathbf{u}$ of the dynamic model are defined as:

$$\mathbf{x} := \big(p_x,\ v_x,\ p_y,\ v_y,\ \theta\big)^T \in X \subset \mathbb{R}^5,$$
$$\mathbf{u} := \big(f_x,\ f_y,\ \omega\big)^T \in U \subset \mathbb{R}^3,$$

where (p_x, v_x) are the CoM position and translational velocity along the x-axis, f_x is the x-coordinate of the stance foot position, θ and ω are the humanoid's orientation and turning rate, respectively. X is the set of the allowed states, while U is the set of the admissible inputs.

Both the state and the input are expressed in the local coordinates of the robot. It means that (p_x, p_y) represents the position of the CoM in the reference frame (RF) that originates from the CoM position at the previous time step. The RF at the next time step will be rotated by an angle θ around the z-axis with respect to the previous frame. The relation between the vectors in different reference frames is represented in Figure 1.

The reference frame at time step 0 is considered the "inertial" or "global" frame. A transformation between the inertial and moving frames is necessary to obtain the position of the humanoid in the global map and to deal with obstacles. Transformations are performed as follows:

$$\mathbf{T_k} = \begin{pmatrix} \cos\theta_k & -\sin\theta_k & p_{x,\,k,\,\mathrm{glob}} \\ \sin\theta_k & \cos\theta_k & p_{y,\,k,\,\mathrm{glob}} \\ 0 & 0 & 1 \end{pmatrix},$$

$$\begin{pmatrix} f_{x,\,k,\,\mathrm{glob}} \\ f_{y,\,k,\,\mathrm{glob}} \\ 1 \end{pmatrix} = \mathbf{T_k} \begin{pmatrix} f_{x,\,k,\,\mathrm{loc}} \\ f_{y,\,k,\,\mathrm{loc}} \\ 1 \end{pmatrix},$$

$$\begin{pmatrix} p_{x,\,k+1,\,\mathrm{glob}} \\ p_{y,\,k+1,\,\mathrm{glob}} \\ 1 \end{pmatrix} = \mathbf{T_k} \begin{pmatrix} p_{x,\,k+1,\,\mathrm{loc}} \\ p_{y,\,k+1,\,\mathrm{loc}} \\ 1 \end{pmatrix},$$

$$\begin{pmatrix} v_{x,\,k+1,\,\mathrm{glob}} \\ v_{y,\,k+1,\,\mathrm{glob}} \end{pmatrix} = \begin{pmatrix} \cos\theta_k & -\sin\theta_k \\ \sin\theta_k & \cos\theta_k \end{pmatrix} \begin{pmatrix} v_{x,\,k+1,\,\mathrm{loc}} \\ v_{y,\,k+1,\,\mathrm{loc}} \end{pmatrix},$$

$$\theta_{k+1,\mathrm{glob}} = \theta_{k,\mathrm{glob}} + \theta_{k+1,\mathrm{loc}}, \qquad \omega_{\mathrm{glob}} = \omega_{\mathrm{loc}}.$$

The positional vectors are roto-translated using a 3 × 3 homogeneous matrix. The velocity vectors are only rotated around the z-axis (indeed, a translation would change their magnitude) using a rotation matrix. The global robot's orientation is obtained by summing the latest local variation to the previous global angle. The angular velocity does not need to be transformed because it is along the z-axis, which is fixed.

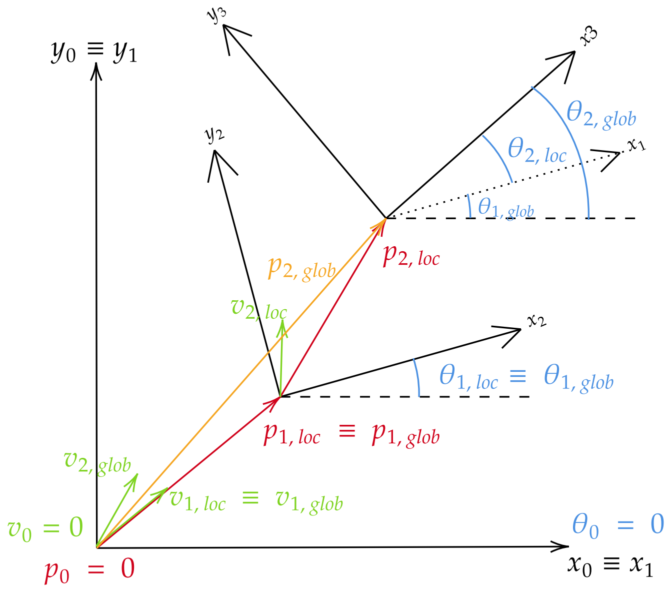

Figure 1: An example of the evolution of the 3D-LIP model's state, highlighting the relationship between local and global coordinates. The initial state is $\mathbf{0}$ and the RF $(x_0,\, y_0)$ is the inertial frame. The local RF translates and rotates: the position of the RF $(x_2,\, y_2)$ related to time step 1 is given by the position of $p_{1,\text{glob}}$, while its orientation is given by $\theta_{1,glob}$. The pose of the successive frames is computed analogously.

3.2 Model Definition

Assuming that the height H of the CoM is constant during the motion, according to [1] we can express the CoM velocity along the x-axis as:

$$\dot{v}_x = \frac{g}{H}\left(p_x - f_x\right),$$

with g as the magnitude of the gravitational acceleration. Since by definition $\dot{\theta} := \omega$, by Euler integration we obtain that:

$$\theta_{k+1} = \theta_k + \omega\, T,$$

where T is the duration of the sampling interval. By further assuming that the duration of a single humanoid's step is fixed and equal to T, we can derive the closed-form step-to-step discrete dynamics of the 3D-LIP model:

$$\mathbf{x}_{k+1} = \mathbf{A_L}\,\mathbf{x}_k + \mathbf{B_L}\,\mathbf{u}_k, \tag{6}$$

with:

$$\mathbf{A_L} := \begin{pmatrix} \mathbf{A_d} & \mathbf{0} & 0 \\ \mathbf{0} & \mathbf{A_d} & 0 \\ \mathbf{0} & \mathbf{0} & 1 \end{pmatrix}, \qquad \mathbf{B_L} := \begin{pmatrix} \mathbf{B_d} & \mathbf{0} & 0 \\ \mathbf{0} & \mathbf{B_d} & 0 \\ 0 & 0 & T \end{pmatrix},$$

$$\mathbf{A_d} := \begin{pmatrix} \cosh \beta T & \frac{\sinh \beta T}{\beta} \\ \beta \sinh \beta T & \cosh \beta T \end{pmatrix}, \qquad \mathbf{B_d} := \begin{pmatrix} 1 - \cosh \beta T \\ -\beta \sinh \beta T \end{pmatrix}, \qquad \beta := \sqrt{\frac{g}{H}}$$

In the following chapter, these dynamics will be used as the internal process representation of an MPC. Along with the constraints, they will grant that the found solution is meaningful to the humanoid motion.

4 MPC

Eugenio:
The dynamic model is used to define a MPC problem to compute the optimal stepping positions for save nagivation and locomotion. The LIP-MPC is defined as:

$$J^* = \min_{u_{0:N-1}} \sum_{k=1}^{N} [(p_{x_k} - g_x)^2 + (p_{y_k} - g_y)^2]$$

$$\text{s. t} \quad x_{k+1} = A_L x_k + B_L u_k \qquad c_l \le c(x_k, u_k) \le c_u$$

Minimizing the cost function means that the robot is moving towards the goal position. This problem is subject to the satisfaction of the robot's dynamics and the kinematics and path constraints. The kinematics and path constraints are captured in $c(x_k, u_k)$ and they are often nonlinear. This nonlinearity is related to presence of the heading angle θ_k. These constraints are linearized by pre-computing the turning rates $\bar{\omega}_k$ for all the horizon length, in order to have them fixed in the MPC calculation.

5 LIP-MPC: Gait planning with Model Predictive Control

Salvatore:
The LIP dynamics defined in (6) is used as a model of the process inside a Model Predictive Control (MPC) scheme. The MPC controller uses that model to predict the future output along a *prediction horizon*, namely the predefined number of time steps to look out in the future. Based on those forecasts and the provided constraints, the MPC computes the sequence of control actions that optimize a given cost function during the prediction horizon. Then, only the first input of that sequence is taken and provided to the process. The real output will be used at the next time step to compute the new predictions and control actions.
In our case, the LIP-MPC is used to respond instantaneously to the changes in the humanoid's state, while providing optimal stepping positions for stable locomotion and safe navigation. It is formulated as follows:

$$
\begin{aligned}
J^* = \min_{\mathbf{u}_{0:N-1}} & \sum_{k=1}^{N} q(\mathbf{x}_k) \qquad (7)\\
\text{s.t.} \quad & \mathbf{x}_k \in X, \quad k \in [1, N]\\
& \mathbf{u}_k \in U, \quad k \in [0, N-1]\\
& \mathbf{x}_{k+1} = \mathbf{A}_\mathbf{L}\mathbf{x}_k + \mathbf{B}_\mathbf{L}\mathbf{u}_k, \quad k \in [0, N-1]\\
& \mathbf{c}_l \leq \mathbf{c}(\mathbf{x}_k, \mathbf{u}_k) \leq \mathbf{c}_u, \quad k \in [0, N-1],
\end{aligned}
$$

where $q(\mathbf{x}_k)$ is the cost function to minimize along the prediction horizon. It drives the humanoid toward the goal by minimizing the distance between its current position and the target position. It is defined as:

$$
q(\mathbf{x}_k) = \left(p_{x_k} - g_x\right)^2 + \left(p_{y_k} - g_y\right)^2 \qquad \forall k \in [1, N]\,,
$$

where the goal position (g_x, g_y) is expressed in the humanoid's local RF. The LIP dynamics is included in the MPC definition to specify how the future states are predicted. Whereas, all the constraints that the optimization problem is subject to are captured by $\mathbf{c}(\mathbf{x}_k, \mathbf{u}_k)$. In order to reduce the computational load, they are all expressed linearly, and they include the walking velocities, leg reachability, and maneuverability constraints, and the linear control barrier function, which will be described in details in the following sections.

6 Conclusion

This is the conclusion.

References

[1] Chengyang Peng, Victor Paredes, Guillermo A. Castillo, and Ayonga Hereid1. Real-time safe bipedal robot navigation using linear discrete control barrier functions. *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[2] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Inverse Differential Kinematics*, chapter 3.5, pages 123–128. Springer, 2009.