



HAL
open science

Control framework for cooperative object transportation by two humanoid robots

Louis Hawley, Wael Suleiman

► **To cite this version:**

Louis Hawley, Wael Suleiman. Control framework for cooperative object transportation by two humanoid robots. *Robotics and Autonomous Systems*, 2019, 115, pp.1-16. hal-02047495

HAL Id: hal-02047495

<https://hal.science/hal-02047495>

Submitted on 25 Feb 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Control Framework for Cooperative Object Transportation by Two Humanoid Robots

Louis Hawley, Wael Suleiman

*Electrical and Computer Engineering Department,
Faculty of Engineering, Université de Sherbrooke, Quebec, Canada*

Abstract

This paper aims at proposing a comprehensive control framework designed for cooperative transportation of a heavy load by two humanoid robots. First, a simplified dynamic model of the cooperative task is developed and the system stability is rigorously analyzed. Next, a centralized controller is formulated, this formulation provides an optimal control of the system by considering the robots dynamical stability while satisfying the robot-object-robot constraints. Finally, the controller is integrated with an arm controller and a local planner module forming a complete framework for cooperative transportation tasks. The approach is thoroughly analyzed and validated in simulation, and experiments are carried out on a team of two Nao humanoid robots transporting a range of objects placed on a small table. The experimental results pointed out the robustness of the approach as the robots successfully accomplished the transportation tasks in a stable way, moreover the transported objects masses were up to half the mass of one of the robot. Besides increasing the robot payload, some of the transported objects are relatively large and it is simply impossible for a single robot to transport them.

Keywords: Humanoid robot, Control, Legged locomotion, Robot-Robot Cooperation

1. Introduction

The human-like structure of humanoid robots enables them to perform a variety of tasks, such as climbing a ladder and driving a car; tasks that cannot be achieved using traditional wheeled robots. However, this increased mobility comes at a price as some of the simplest chores such as object manipulation and transportation can become challenging tasks. The main limitations with respect to object handling and transportation tasks come from humanoid robots delicate dynamic balance and the usually low power-to-weight ratio of the humanoid robot actuators .

Since object transportation is one of the most common tasks for a robot, it is important for humanoids to be able to efficiently handle them in order to enact their capacity in real-life situations.

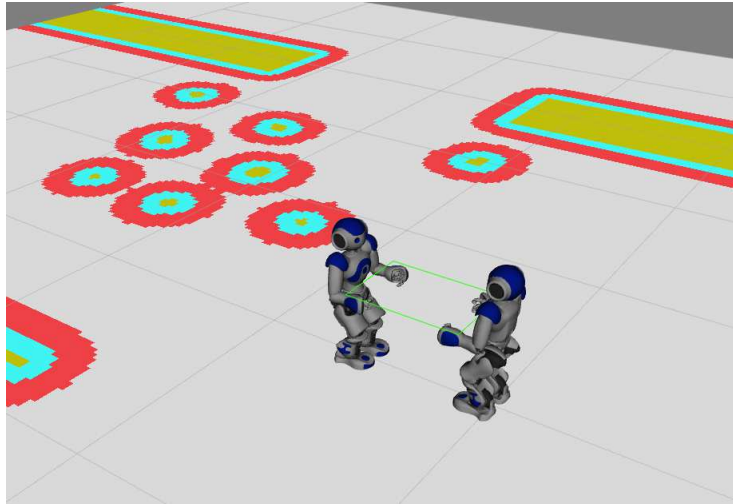


Figure 1: A team of two humanoid robots cooperating to transport an object among obstacles

To this end, one of the most interesting options is to have multiple robots cooperating in order to carry heavy and/or large object as shown in Fig. 1. Although simpler solutions such as using a wheeled-cart [15, 29] can achieve great results, cooperative transportation has the advantage of not requiring any external tool, thus it is a promising option in precarious and unknown environments such as the scene of a natural disaster. However, cooperation between robots requires particular considerations at both motion planning and control levels.

2. Related Work

There are two main challenges that arise when two robots are rigidly linked together :

1. The robots motion planning modules must be linked and synchronized in such a way that they move in a coherent manner.
2. The robots controllers must also be connected to make sure they don't compete with each other while trying to maximize their own balance instead of ensuring the whole system stability and overall performance.

Previous works in both robot-robot and robot-human interaction can provide insights into how to deal with both challenges.

2.1. Leader-Slave Cooperation

There are multiple ways of planning the displacement of a robot-object-robot (or a human) system in the environment. The simplest one requires defining only the trajectory of one of the agents that is considered the leader. It is

an approach that is often used in human-robot cooperation [1, 6, 35] where the human acts as the leader and the robot simply follows along with the carried object acting as a haptic interface between the human and the robot. A drawback is that complex coordinated motions that are usually required in a cluttered environment are not possible with this strategy. Furthermore, in a human-robot cooperation, the human can greatly reduce the imperfection in the haptic feedback by adapting his motions with the robot actions, however, in the case of a robot-robot scenario, the interaction force between the agents can increase rapidly. In [33], the Leader-Slave approach has been implemented on two HRP-2 robots with the slave robot using Force/Torque (F/T) sensors in the wrist to infer the intention/motion of the leader. Using this method, the leader robot velocity must be limited to consider the lag in the measurements of the force by the slave robot. Despite the relatively small velocity of the robots, strong interaction forces were measured during the experiments. Of course, it is expected that the results would be far worse if the robots were not equipped with F/T sensors as is the case for the majority of small or medium-sized humanoid robots.

2.2. Symmetry Control Framework

The authors in [34] proposed a control framework for multiple humanoids transporting an object where the input is a trajectory for the transported object and a desired internal force. Then, the trajectories for the hands of each robot are generated from the object desired trajectory and the robots footsteps are finally produced in order to keep the feet at a constant position from the center of the object.

From a motion planning point of view, it makes more sense to specify the trajectory of the object rather than the trajectory of one of the robots as done in the master-slave approach. However, keeping the robot always at a fixed position from each other severely reduce the set of feasible motions, some of which might be needed in complex environments.

An interesting aspect of the framework is the possibility of specifying an internal force as this force could be useful when transporting a deformable object. However, the authors only present the most practical case where the desired internal force is set to zero. The interaction force between a robot and the object is limited by moving the arms using a force control loop that uses measurements from F/T sensors in the robot wrists. As a consequence, the robustness of the framework highly depends on the performance of the force sensor akin to the master-slave approach. Furthermore, the pattern generator does not consider the whole system dynamic, but instead it is only treated as an inverse kinematics problem where constraints on the feet and the torso of a robot are computed by each robot pattern generator and the hand position constraints are supplied by a high-level motion planner. The approach exhibited a good performance in simulation with a transported object of negligible mass and the interaction force between the robot kept low by using the force control loop of the arms.

In [25], the authors used two Darwin-OP robots to transport a small stretcher. The proposed approach used a separate control schemes for each robot where the

footsteps were synchronized using communication over a wireless network. The distance between the robots was also monitored using the embedded camera of the robots. Therefore, the problem was treated more as a synchronization task in terms of footstep timing and relative displacement. However, the dynamic was simplified as the stretcher had a negligible mass, moreover, no interaction force was applied to the robots since the stretcher could easily slide into the robot grippers.

2.3. Centralized Controller

In [27, 28], a thorough strategy for planning is presented as the whole system footprint consisting of the two humanoid robots and of the transported object is considered in the motion planning phase. Using this approach, the robots can effectively use their arms to execute more complex motions such as sharp turns in tight corners. The biggest challenge with this strategy is ensuring that the robots are accurately following their respective trajectory and in a synchronized manner. The strategy used by the authors is to separate the navigation into two tasks: the first one is for each robot to follow its respective trajectory and the second is to respect the desired distance between the robots. Therefore, each robot must locate itself in the world frame as well as with respect to the other robot. A similar approach was investigated in [18], where all the possible positional errors (vertical misalignment, horizontal misalignment, etc.) between two humanoid robots are estimated and the optimal strategies to bring the system back to the desired state are found using Q-learning methods.

This sort of approach is efficient in cases where the agents are not directly linked together such as drone flying in formation, but as it has been shown by the authors of both papers, it is often not sufficient in the case of rigidly linked humanoid robots where the acceptable margin of error is really small.

Up to our knowledge, the most complete approach is probably the one presented in [5], where the authors propose a general framework for humanoid robots enabling them to execute any task by considering all the kinematic and dynamic constraints. Unfortunately, the computational requirements of this approach makes it currently unadapted for real-life scenarios.

3. Contributions

The main contributions in this work are:

- Designing a control framework allowing two humanoid robots to efficiently execute a cooperative transportation task of a heavy or large load. The stability of the closed-loop control system is also thoroughly analyzed.
- The implementation and validation of the framework on two Nao humanoid robots. Even though the proposed framework is general enough to be applied to any advanced high-end humanoid robot, such as HRP series, ATLAS or Toro robots, we show that it can also be implemented on small or medium-sized humanoid robots having limited hardware and computational capabilities, such as a Nao robot.

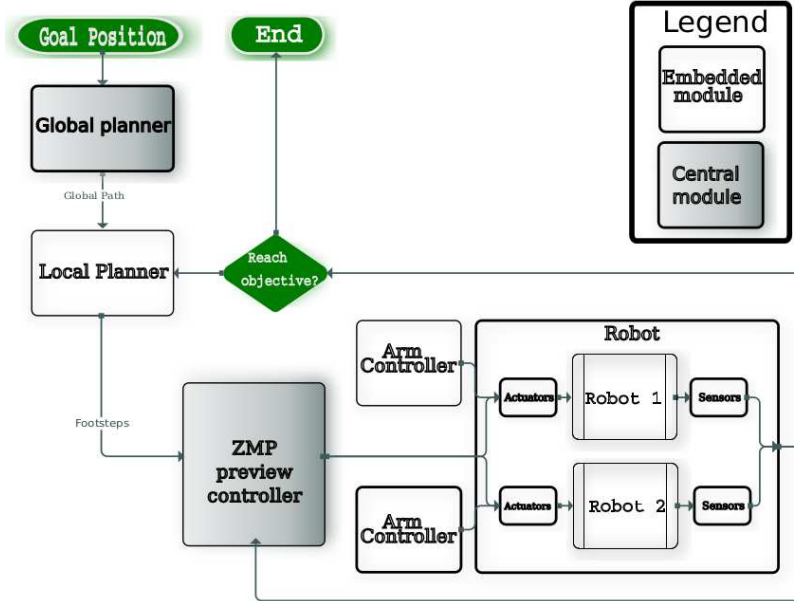


Figure 2: Global architecture: central modules are executed on an external PC and embedded modules are executed on the robot on-board computer.

The global architecture of the developed framework to carry out the cooperative transportation task is shown in Fig. 2. The main differences between our approach and those presented in Sec. 2:

1. From Fig. 2 and the connected centralized and embedded (decentralized) modules, one can figure out that our strategy clearly differs from the master-slave and centralized approaches.
2. It differs from the symmetry control strategy as the robots are given the possibility to have a variable distance between them. To cope with this variation, we propose a control strategy that is based on estimating the internal force, more details are in Sec. 4 and 5.

The next subsections briefly present each module separately.

3.1. Global Planner

The role of the global planner is to find a valid path for the system (robot-object-robot) to reach the desired goal (position and orientation). This can be a particularly complex task in the case of two rigidly linked humanoid robots since there are a lot of degrees of freedom. In our previous works [27, 28], we investigated the problem and came up with a lattice-based graph planner that uses motion primitives to reduce the set of possible motions. Using this approach, it is possible to easily tweak the planner to take into account the

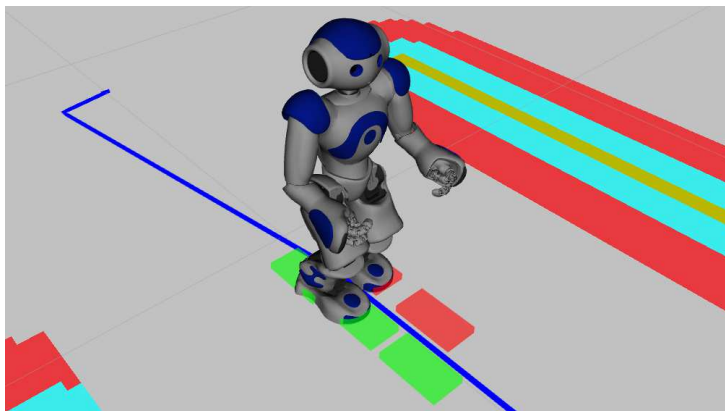


Figure 3: A footstep planner is used as a local planner to follow a global trajectory, which is in blue line, among obstacles

limitations and abilities of different robots. This paper does not focus on the global planner and the framework is independent of the actual implementation of this module as long as it provides a high-level command for the system whether it is an operator command or a global path provided by a planning algorithm.

3.2. Local Planner

The local planner module manages the navigation of the robots. In other words, the local planner provides a high-level command to the humanoid robots to follow the desired global plan as shown in Fig. 3. Typical high-level motion commands for humanoid robots are speed reference [16] or footsteps [10].

In the case of a cooperative transportation task, another role of the local planner is to ensure the synchronization of the robots displacements. Indeed, if one robot is progressing more rapidly than the other one for various reasons (slipping, external disturbance, etc.), the system will rapidly become unstable since the transported object ties them together. In this case, changing the footsteps is the preferable high-level control command as it gives an accurate control over the displacement of each robot. The local planner module is discussed in section 7.

3.3. ZMP Preview Controller

The objective of the ZMP preview controller is to plan the Center of Mass (CoM) trajectories in order to ensure the robots dynamical balance and providing the high-level command for the footstep module. A popular approach is the model predictive control [20] which uses a simplified model of the robot dynamics. This module is central since it has an immediate control over the robot motions contrarily to the local planner that can only modify the footsteps at discrete intervals. In our previous works [27, 28], we treated cooperative transportation as a kinematic problem since we didn't consider the particular

dynamic of this task. Hence, the approach was validated with the robots carrying a flexible table of negligible weight, thereby limiting the interaction force. As the motion of each robot directly impacts the other one via the transported object, a centralized controller is the preferable option in this case. In Section 5, the dynamic equations of the cooperative transportation task is analyzed. Section 6 presents our controller formulation.

3.4. Arm Controller

In human-robot and robot-robot cooperative system, the arms often play an important role since the interaction between the agents usually occurs through them. It is also the case for cooperative transportation, and as such special attention is given to the control of the arms during the task. In Section 4, the particular dynamic behavior and stability of the humanoid-object-humanoid system are presented and the relevance of having compliant arms during the task is discussed.

4. Compliance Control of the Robot Arms

One of the biggest challenges with cooperative transportation by two humanoid robots is that commonly used stabilization strategies [31] are no longer valid. This is because those approaches assume that all external forces applied to the humanoid robot are undesirable and must be compensated. However, when two robots are rigidly linked together, the main source of disturbance is the interaction between the robots. Therefore, it is no longer desirable to try to compensate for the force, but instead the robots must comply with it.

In such a situation, the controller must act rapidly to re-establish stability by reducing the perturbing interaction force. In a human-robot scenario, the human can quickly and actively rectify the situation and act accordingly. It is a more complex problem in a robot-robot collaborative transportation scenario. This is because the controller can either modify the desired ZMP trajectory or keep that trajectory and modify the CoM trajectory instead. The first solution is done by modifying the footsteps to bring the robot back to the desired relative position with respect to its partner. The second option can be achieved by moving the arms, thereby maintaining control of the robot torso/CoM.

Modifying the footsteps is the only solution that can really bring the system back to its desired state, but it is quite slow. That is why an important part of any humanoid robot interaction system is the implementation of a compliance control. Indeed, most work [8, 30, 33, 34] in robot-robot and human-robot cooperation have implemented some sort of compliance of the robot arms as a way to ensure the robot balance while executing the interaction task.

The most widely documented method in the literature for implementing arm compliance with position-controlled humanoid robot is based on an impedance control loop of the end-effector position. The control law is usually given by:

$$\begin{aligned} x_m &= x - x_r \\ F_c &= D \ddot{x}_m + B \dot{x}_m + K x_m \end{aligned} \tag{1}$$

where p_{x_1} and p_{x_2} are respectively the projections of the ZMP of the robots 1 and 2 on the x axis.

As shown in [2, 13], the ZMP is given by the following formula (for more details, refer to Appendix A):

$$\mathbf{p} = \begin{bmatrix} p_{x_1} \\ p_{x_2} \end{bmatrix} = \begin{bmatrix} x_{c_1} + \frac{M_{c_1} z_{c_1}}{f_{n_1}^o} \ddot{x}_{c_1} - \frac{z_{c_1}}{f_{n_1}^o} F_{x_1} - \frac{1}{f_n^o} \tau_{y_1}^{ext} \\ x_{c_2} + \frac{M_{c_2} z_{c_2}}{f_{n_2}^o} \ddot{x}_{c_2} - \frac{z_{c_2}}{f_{n_2}^o} F_{x_2} - \frac{1}{f_n^o} \tau_{y_2}^{ext} \end{bmatrix} \quad (2)$$

where:

- M_{c_i} and z_{c_i} are respectively the total mass and the height of the robot i CoM.
- g is the magnitude of gravity acceleration.
- x_{c_i} and \ddot{x}_{c_i} are the position and acceleration of the robot i CoM projection on the x axis.
- F_{x_i} is the external force applied to the CoM of robot i along x axis.
- $\tau_{y_i}^{ext}$ is the external moment applied to the CoM of robot i along y axis.
- $f_{n_i}^o$ is the total normal force applied to the robot i feet, it is given by the following expression:

$$f_{n_i}^o = -M_{c_i} g - M_{c_i} \ddot{z}_{c_i} + F_{z_i} \quad (3)$$

where F_{z_i} is the external force applied to the CoM of robot i along z axis.

Without loss of generality, we assume that the carried mass m_0 is located at the middle of the transported object and is therefore split uniformly between the two robots, i.e. a mass $\frac{m_0}{2}$ is carried by each robot. Note that if the object is not placed in the middle, each robot should estimate its carried mass separately as explained in Section 8.2.2.

By assuming that the CoM height, z_{c_i} , is constant during the locomotion, and replacing $F_{z_i} = -\frac{m_0}{2} g$, we obtain:

$$\begin{aligned} f_{n_i}^o &= -\left(M_{c_i} + \frac{m_0}{2}\right) g \\ &\triangleq -\widehat{M}_i g \end{aligned} \quad (4)$$

$\tau_{y_1}^{ext}$ is the moment applied to the CoM of robot 1 as a result of the force $\frac{m_0 g}{2}$ which is exerted on the robot hands:

$$\tau_{y_1}^{ext} = \frac{m_0 g}{2} x_1^h \quad (5)$$

where x_1^h is the position of the robot 1 hands with respect to its CoM.

Up to this point, the dynamic equations are equivalent to a robot carrying an object [12], however, the main difference is the additional external force caused by the interaction between the robots. For instance, the interaction force F_{x_1} applied to robot 1 CoM along x axis is expressed as:

$$F_{x_1} = \underbrace{-\frac{m_0}{2}\ddot{x}_0}_{F_0} + \underbrace{K_{eq}(x_{c_2} - x_{c_1} - L)}_{F_c} \quad (6)$$

where

- F_0 is the force caused by the transported mass and F_c is the resultant force of the compliance control.
- L is constant which is the initial/desired distance between the two robots.
- K_{eq} is the equivalent stiffness coefficient:

$$K_{eq} = \frac{K_1 K_2}{K_1 + K_2}$$

and K_i is the stiffness coefficient of the compliance controller for the robot i .

The position, velocity and acceleration, x_0 , \dot{x}_0 , \ddot{x}_0 , of the projection of the carried mass on the x axis are given by:

$$\begin{aligned} x_0 &= x_{c_1} + \frac{L}{2} + (x_{c_2} - x_{c_1} - L) \frac{K_{eq}}{K_1} \\ \dot{x}_0 &= \dot{x}_{c_1} + (\dot{x}_{c_2} - \dot{x}_{c_1}) \frac{K_{eq}}{K_1} \\ \ddot{x}_0 &= \ddot{x}_{c_1} + (\ddot{x}_{c_2} - \ddot{x}_{c_1}) \frac{K_{eq}}{K_1} \end{aligned}$$

The position of the robot 1 hands, x_1^h , can be obtained as follows:

$$F_c = K_1 (x_1^h - x_1^{href}) \Rightarrow x_1^h = x_1^{href} + (x_{c_2} - x_{c_1} - L) \frac{K_{eq}}{K_1}$$

x_1^{href} is constant which is the initial/reference value of the hands position with respect to the CoM.

Similarly, for robot 2 we obtain:

$$F_{x_2} = -\frac{m_0}{2}\ddot{x}_0 - K_{eq}(x_{c_2} - x_{c_1} - L)$$

and $\tau_{y_2}^{ext}$ is given by:

$$\tau_{y_2}^{ext} = \frac{m_0 g}{2} x_2^h \quad (7)$$

where x_2^h is the position of the robot 2 hands with respect to its CoM:

$$x_2^h = x_2^{h^{ref}} - (x_{c_2} - x_{c_1} - L) \frac{K_{eq}}{K_2}$$

$x_2^{h^{ref}}$ is defined in a similar way as $x_1^{h^{ref}}$. Note that $x_2^{h^{ref}}$ is negative, and as it is designed to be in the middle of the effective operational space of the robot hands, x_2^h is always negative, as a result $\tau_{y_2}^{ext}$ is also always negative.

As a result, p_{x_1} and p_{x_2} become:

$$\begin{aligned} p_{x_1} &= x_{c_1} - \frac{M_{c_1} z_{c_1}}{\widehat{M}_1 g} \ddot{x}_{c_1} - \frac{m_0 z_{c_1}}{2\widehat{M}_1 g} \ddot{x}_0 + \frac{K_{eq} z_{c_1}}{\widehat{M}_1 g} (x_{c_2} - x_{c_1} - L) \\ &\quad + \frac{m_0}{2\widehat{M}_1} x_1^{h^{ref}} + \frac{K_{eq} m_0}{2 K_1 \widehat{M}_1} (x_{c_2} - x_{c_1} - L) \\ &= x_{c_1} - \frac{z_{c_1}}{\widehat{M}_1 g} \left(M_{c_1} + \left(1 - \frac{K_{eq}}{K_1} \right) \frac{m_0}{2} \right) \ddot{x}_{c_1} - \frac{m_0 z_{c_1} K_{eq}}{2\widehat{M}_1 g K_1} \ddot{x}_{c_2} \\ &\quad + \frac{m_0}{2\widehat{M}_1} x_1^{h^{ref}} + \frac{K_{eq}}{\widehat{M}_1} \left(\frac{m_0}{2 K_1} + \frac{z_{c_1}}{g} \right) (x_{c_2} - x_{c_1} - L) \end{aligned} \quad (8)$$

and

$$\begin{aligned} p_{x_2} &= x_{c_2} - \frac{M_{c_2} z_{c_2}}{\widehat{M}_2 g} \ddot{x}_{c_2} - \frac{m_0 z_{c_2}}{2\widehat{M}_2 g} \ddot{x}_0 - \frac{K_{eq} z_{c_2}}{\widehat{M}_2 g} (x_{c_2} - x_{c_1} - L) \\ &\quad + \frac{m_0}{2\widehat{M}_2} x_2^{h^{ref}} - \frac{K_{eq} m_0}{2 K_2 \widehat{M}_2} (x_{c_2} - x_{c_1} - L) \\ &= x_{c_2} - \frac{z_{c_2}}{\widehat{M}_2 g} \left(M_{c_2} + \frac{K_{eq} m_0}{K_1} \right) \ddot{x}_{c_2} - \left(1 - \frac{K_{eq}}{K_1} \right) \frac{m_0 z_{c_2}}{2\widehat{M}_2 g} \ddot{x}_{c_1} \\ &\quad + \frac{m_0}{2\widehat{M}_2} x_2^{h^{ref}} - \frac{K_{eq}}{\widehat{M}_2} \left(\frac{m_0}{2 K_2} + \frac{z_{c_2}}{g} \right) (x_{c_2} - x_{c_1} - L) \end{aligned} \quad (9)$$

Similar equations can be found for the ZMP on the y axis.

6. Controller

Using the ZMP equations (8) and (9) of the robots, our objective is to define a state-space representation and formulate the control problem as an optimization problem.

6.1. State-Space Formulation

First, let us define p such that :

$$p(t) = \begin{bmatrix} p_{x_1} + \frac{K_{eq}}{\widehat{M}_1} \left(\frac{m_0}{2 K_1} + \frac{z_{c_1}}{g} \right) L - \frac{m_0}{2 \widehat{M}_1} x_1^{h^{ref}} \\ p_{x_2} - \frac{K_{eq}}{\widehat{M}_2} \left(\frac{m_0}{2 K_2} + \frac{z_{c_2}}{g} \right) L - \frac{m_0}{2 \widehat{M}_2} x_2^{h^{ref}} \end{bmatrix} \quad (10)$$

Then, it is possible to discretize the ZMP equations in (8) and (9) at a sampling period T , and express the system as a Linear Time Invariant (LTI) system as follows:

$$\begin{aligned}\mathcal{X}(k+1) &= A\mathcal{X}(k) + Bu(k) \\ p(k) &= C\mathcal{X}(k)\end{aligned}\quad (11)$$

where:

$$\mathcal{X}(k) = [x_{c_1}(kT) \quad \dot{x}_{c_1}(kT) \quad \ddot{x}_{c_1}(kT) \quad x_{c_2}(kT) \quad \dot{x}_{c_2}(kT) \quad \ddot{x}_{c_2}(kT)]^T \quad (12)$$

$$u(k) = u(kT) = [\ddot{x}_{c_1}(kT) \quad \ddot{x}_{c_2}(kT)]^T$$

$$p(k) = p(kT)$$

$$A = \begin{bmatrix} 1 & T & \frac{T^2}{2} & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & \frac{T^2}{2} \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} \frac{T^3}{6} & 0 \\ \frac{T^2}{2} & 0 \\ T & 0 \\ 0 & \frac{T^3}{6} \\ 0 & \frac{T^2}{2} \\ 0 & T \end{bmatrix} \quad (13)$$

$$C = \begin{bmatrix} 1 - \frac{K_{eq}}{M_1} \left(\frac{m_0}{2K_1} + \frac{z_{c_1}}{g} \right) & \frac{K_{eq}}{M_2} \left(\frac{m_0}{2K_2} + \frac{z_{c_2}}{g} \right) \\ 0 & 0 \\ -\frac{z_{c_1}}{M_1 g} \left(M_{c_1} + \left(1 - \frac{K_{eq}}{K_1} \right) \frac{m_0}{2} \right) & -\left(1 - \frac{K_{eq}}{K_1} \right) \frac{m_0 z_{c_2}}{2M_2 g} \\ \frac{K_{eq}}{M_1} \left(\frac{m_0}{2K_1} + \frac{z_{c_1}}{g} \right) & 1 - \frac{K_{eq}}{M_2} \left(\frac{m_0}{2K_2} + \frac{z_{c_2}}{g} \right) \\ 0 & 0 \\ -\frac{m_0 z_{c_1} K_{eq}}{2M_1 g K_1} & -\frac{z_{c_2}}{M_2 g} \left(M_{c_2} + \frac{K_{eq}}{K_1} \frac{m_0}{2} \right) \end{bmatrix}^T \quad (14)$$

Note that even though the state and input matrices, A and B , in (11) are the same as in [20], the coupling between the state variables occurs at the output level through the matrix C .

The main objective of the controller is then to maximize the system stability by maintaining the ZMP of each robot at the center of the supporting foot. This can be expressed as an optimization problem where the cost function to be minimized is expressed as:

$$J = e^T Q_e e + \mathcal{X}^T Q_x \mathcal{X} + u^T R u \quad (15)$$

where e is the error between the reference and the actual values of the vector p , it is defined as follows:

$$\begin{aligned}
e &= p^{ref} - p \\
&= \begin{bmatrix} p_{x_1}^{ref} + \frac{K_{eq}}{M_1} \left(\frac{m_0}{2K_1} + \frac{z_{c1}}{g} \right) L - \frac{m_0}{2M_1} x_1^{h^{ref}} \\ p_{x_2}^{ref} - \frac{K_{eq}}{M_2} \left(\frac{m_0}{2K_2} + \frac{z_{c2}}{g} \right) L - \frac{m_0}{2M_2} x_2^{h^{ref}} \end{bmatrix} - \begin{bmatrix} p_{x_1} + \frac{K_{eq}}{M_1} \left(\frac{m_0}{2K_1} + \frac{z_{c1}}{g} \right) L - \frac{m_0}{2M_1} x_1^{h^{ref}} \\ p_{x_2} - \frac{K_{eq}}{M_2} \left(\frac{m_0}{2K_2} + \frac{z_{c2}}{g} \right) L - \frac{m_0}{2M_2} x_2^{h^{ref}} \end{bmatrix} \\
&= \begin{bmatrix} p_{x_1}^{ref} - p_{x_1} \\ p_{x_2}^{ref} - p_{x_2} \end{bmatrix}
\end{aligned} \tag{16}$$

$p_{x_1}^{ref}$ and $p_{x_2}^{ref}$ are the ZMP reference trajectories of each robot.

The matrices Q_x , Q_e and R are positive definite. This cost function means that we wish to follow the reference ZMP trajectory without rapid change in the states and the inputs.

6.2. Optimal Control Approaches

Linear Quadratic Regulator (LQR) and Model Predictive Control (MPC) approaches are among of the most used methods in gait generation for humanoid robots. They use a model of the robot dynamic to predict the effect of current command on the robot future states. The objective of the controller is then to minimize the cost function over a preview horizon N instead of only the current state such as:

$$\begin{aligned}
\underset{u}{\text{minimize}} \quad & J = \sum_{i=k}^{N+k} \{e(i)^T Q_e e(i) + \mathcal{X}(i)^T Q_x \mathcal{X}(i) + u(i)^T R u(i)\} \\
\text{subject to} \quad & \mathcal{X}(i+1) = A\mathcal{X}(i) + Bu(i) \\
& p(i) = C\mathcal{X}(i)
\end{aligned} \tag{17}$$

The next subsections present how to design a controller that minimize such a cost function.

6.2.1. LQR Approach

LQR algorithm provides the optimal state-feedback, i.e. $u(k)$. Even though the formulation of (17) is under the finite-horizon form, many approaches, such as [20], solve it as an infinite-horizon problem by assuming that N is big enough. This is because the Discrete time Algebraic Riccati Equation (DARE), in the case of infinite-horizon, is solved only once offline and used online for computing the optimal feedback, therefore it is particularly interesting for small or medium-sized humanoids with limited on-board computational power. The controller can also be improved by integrating an integral term [19].

6.2.2. MPC Approach

Another possible way is the MPC approach, in this case the design of the controller is transformed into reformulating the optimization problem (17) as a Quadratic Programming (QP) problem of the following form :

$$\begin{aligned}
 & \underset{u}{\text{minimize}} && \frac{1}{2}u^T H u + g^T u \\
 & \text{subject to} && b^- \leq A u \leq b^+ \\
 & && u^- \leq u \leq u^+
 \end{aligned} \tag{18}$$

where H is a $n \times n$ positive semidefinite matrix. The objective is to find the input vector u that minimizes the cost function while also satisfying some constraints.

The authors in [16] present the detailed procedure to setup the QP problem in order to solve the cost function of (17). Evidently, the main advantage of using quadratic programming to solve this problem is the possibility to add constraints, such as constraining the ZMP within a certain area instead of following a predefined trajectory. However, this approach is computationally more expensive than the LQR algorithm even though it can usually be solved rapidly with an off-the-shelf solver such as [9]. Note that when no constraints are added, the QP can be solved analytically as shown in [32]. In our implementation, we used the MPC approach since it offers the possibility of adding constraints as discussed in the following subsection.

6.3. Controller Robustness

The primary objective of the previously presented controller formulation is to ensure the stability of the robots by ensuring that the ZMP is kept at the center of the feet. There is however another aspect that must be considered, the robots CoM motions are actually restricted since the arms motions are limited to a small maneuverability range. Once the limits are reached, the compliance is lost and the system of robot-object-robot becomes rigidly linked. In this state, the system can easily collapse as explained in Section 4.

To avoid this situation, there are two ways of limiting the distance between the robots:

- Option 1: Reducing the interaction force by implementing a fast and efficient compliance control of the arms.
- Option 2: Relaxing the constraint of precisely following the ZMP reference trajectory by reducing the norm of weighting matrix on the error, $e(k)$, in (17).

The best option is of course option 1, however, implementing such control is a challenging task for small humanoid robots since it requires a rapid and accurate estimation of the interaction force and a fast motion of the actuators.

To increase the overall robustness of the system to positional errors, we propose to integrate an additional constraint on the relative torso displacement,

hence allowing the ZMP to slightly move away from the reference position in order to reduce the torso displacement.

To this end, we add a new objective in the cost function, thereby obtaining:

$$J = e^T Q_e e + \mathcal{X}^T Q_x \mathcal{X} + u^T R u + \Delta x_c^T Q_d \Delta x_c \quad (19)$$

where $\Delta x_c = (x_{c2} - x_{c1} - L)$, and Q_d is the weighting matrix of the new objective that minimizes the relative distance between the robots torsos. However, each time a new objective function is added, the weighting matrices should be tuned to reflect the priority between those objectives, a way for tuning the weighting matrices is given in [21].

With this simple formulation, the limitations on the arm motions can be implicitly integrated into the controller. Additionally, the limitation can also be explicitly considered in the QP problem by adding the following constraint :

$$e^- \leq \Delta x_c \leq e^+ \quad (20)$$

where e^- and e^+ are respectively the minimum and maximum relative distance between the robots torsos.

6.4. Controller Stability

Another aspect to be taken into consideration during the design of the controller is the system stability.

Note that the stability of the closed-loop depends on the ratio R/Q and the length NT of the horizon over which the trajectories are pre-computed as mentioned in [32], using reasonable values similar to those in [32], $R/Q = 10^{-6}$ and $NT = 1.2s$, we only focus on the stability of the closed-loop of the optimization problem (17) as a function of K_{eq} and m_0 through numerical analysis. Fig. 5 shows the closed-loop system poles for a K_{eq} in the range of 10 to 300 Nm^{-1} and two different transported masses.

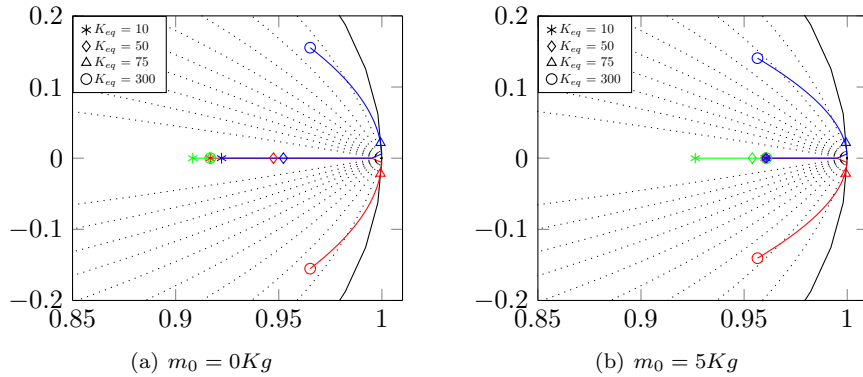


Figure 5: Closed-loop dominant pole positions in the Z-plane for K_{eq} varying from 10 to 300.

The following points need to be kept in mind when designing the controller:

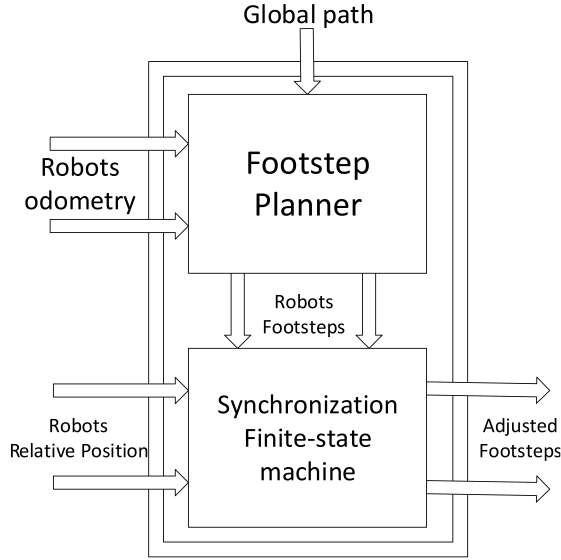


Figure 6: Block diagram of local planner

- The system is always stable. However, some values of K_{eq} are undesirable since they bring the poles really close to the instability region. In the case of small humanoid robots ($M_{c_1} = M_{c_2} = 5.1Kg$), the critical values are roughly between 70 and 80 Nm^{-1} .
- Fig. 5 points out that the oscillations in the closed-loop response can be avoided by keeping the interaction coefficient (K_{eq}) low.
- Transporting a heavier load has a stabilizing effect on the system response since it reduces the oscillations at high values of K_{eq} .

7. Local Planner for Cooperative Transportation

Although the previously presented architecture of ZMP preview controller can ensure the robots stability, it cannot bring the system back to the desired state contrarily to some other control schemes that enable reactive footstep modification [7, 11, 30]. Therefore, it is the role of the local planner to ensure that both robots follow their respective paths, it is decomposed into two modules as presented in Fig. 6.

The first module is a conventional footstep planner [10, 17] that, using the robots current position estimations and the global path, outputs high-level footsteps command to follow the global path. However, since it can be difficult for a robot to position itself precisely in the world, another module is necessary to ensure that the robots do not move too close or too far from each other. The approach used here is to modify the footstep length, but another solution could

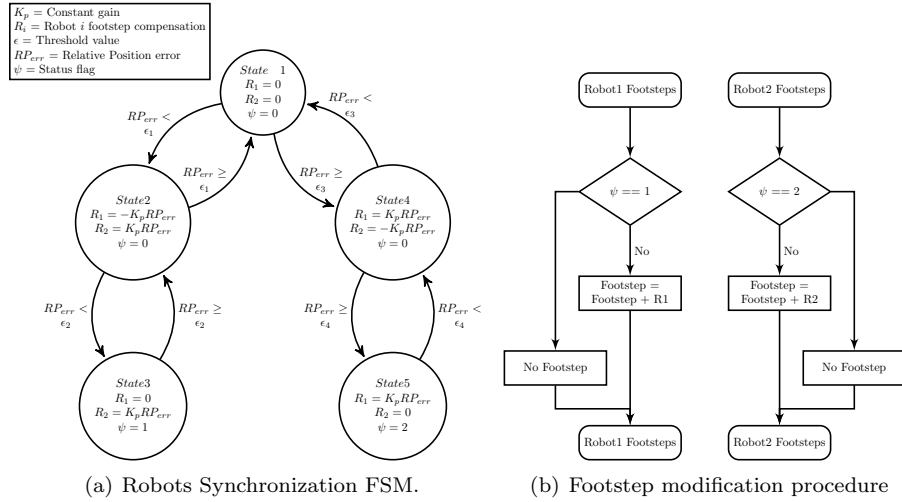


Figure 7: Block diagrams of footstep modification procedure in the synchronization module.

have been to modify the step frequency of the robots [26]. The used module is a finite-state machine (FSM) that modifies the footsteps command according to the robots relative position error.

As can be seen in Fig. 7(a), there are five possible states :

- State 1. The robots Relative Position error (RP_{err}) is small. No compensation is needed.
- State 2. The robots are too close to each other ($RP_{err} < \epsilon_1$). The leading robot (robot 2) needs to speed up by taking bigger steps while the other robot will slow down by taking smaller steps.
- State 3. The stability of the system is compromised as the arms are close to the limit ($RP_{err} < \epsilon_2$). The leading robot needs to speed up by taking bigger steps while the other robot should stop walking and wait for the system to come back to a more stable state.
- State 4. The robots are too far from each other ($RP_{err} > \epsilon_3$). The leading robot needs to slow down by taking smaller steps while the other robot will speed up.
- State 5. The stability of the system is compromised as the arms are extended to the maximum operational space ($RP_{err} > \epsilon_4$). The leading robot will stop walking while the other robot will walk faster until the system comes back to a more stable state.

The thresholds ϵ_1 and ϵ_3 are added so that compensation is done only when it is really needed. Whenever the footsteps are modified, the new resulting footprints must be checked for collision using a map of the environment. As for

the thresholds ϵ_4 and ϵ_5 , they were added because of the important slipping of the feet of Nao robot that was observed during the experimentations.

As shown in Fig. 6, the local planner needs both the estimated position of the robots in the world frame, using the robot odometry (visual or other), as well as an estimation of the robots relative position. An important assumption is that the robots are able to estimate their relative position more accurately than their global position.

In order to estimate the position between the two robots, two techniques were considered :

- Vision-based technique
- Closed-loop kinematic

In the previous version of our framework [27], the global position of the robots was monitored using a Simultaneous Localization and Mapping (SLAM) algorithm. The robots relative position was then computed and compared with the already known object dimension to compute the relative position error. The precision of the estimation was therefore highly dependent on the performance of the localization algorithm, which turns out not to be enough for our application.

The approach was refined by adding a locally running optical flow based module [28] that monitored the relative displacement of both robots using the robot embedded camera. A relatively good performance was obtained using this method but there were some drawbacks. First, the approach required some pre-processing to find a set of good features to track and to set the reference of the robots relative position. Second, an obvious requirement is that the robot can see each other. This requirement goes against the objective of cooperative transportation of a heavy or large object.

In the case of humanoid robots where most of the mass of the robot is located at the torso, the distance between the robots torsos is equivalent to the distance between their CoMs. As the transported table is rigid, the transformation between the robots hands and the center of the table is constant, as shown in Fig. 8. Therefore, using the knowledge of the object dimension, one can compute the transformation between the robot torso i and the center of the transported object as follows:

$$T_{ob}^{r_i} = \text{midpoint}(T_{r_i}^{hr} \times T_{hr}^{ob}, T_{r_i}^{hl} \times T_{hl}^{ob}) \quad (21)$$

where $T_{ob}^{r_i}$ is the homogeneous transformation matrix between the torso of robot i and the center of the object, $T_{r_i}^{h(l/r)}$ is the transformation between the robot torso and its left/right hand and $T_{h(l/r)}^{ob}$ is the transformation between the robot hand and the center of the object. However, this time we assume that $T_{h(l/r)}^{ob}$ is the actual transformation matrix and not simply a desired one. This assumption is valid because the table is rigid. Therefore, the relative position between the robots can be easily obtained with :

$$T_{r_2}^{r_1} = T_{ob}^{r_1} \times (T_{ob}^{r_2})^{-1} \quad (22)$$

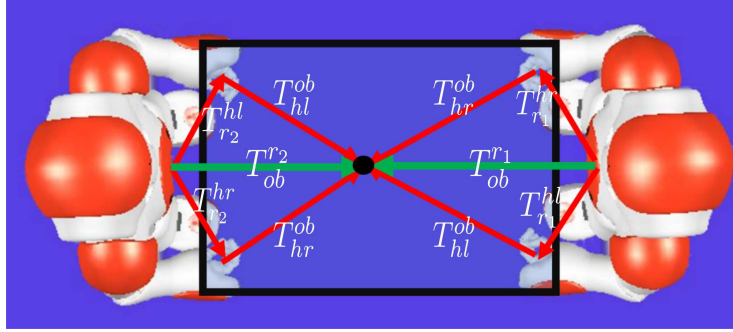


Figure 8: Transformation used to compute the distance between the robots torso

If the object dimensions are precisely known, the performance of the method is only limited by the precision of kinematic model of the robot arms and the joint encoders. In the case of Nao robot, the available magnetic rotary encoders have a 12 bit precision that provides precision around 0.1° . Considering the small size of Nao robot, a maximum error within a couple of mm is to be expected for the arms forward kinematics. Another advantage is that the method does not require a line of sight between the robots, thus allowing the robots to transport a huge object while allocating the cameras to other tasks, navigation for instance.

In the local control loop, it is not required to have an absolute transformation between the robots, but instead each robot only monitors its arms positions with respect to a reference/desirable position such as :

$$T_{r_i}^h = \text{midpoint}(T_{r_i}^{hr}, T_{r_i}^{hl}) \quad (23)$$

Then, the rotation and position errors between the actual hand position and the reference position for each robot can be found:

$$\begin{aligned} R_e &= R^{h^{ref}} \times R^h{}^T \\ P_e &= P^{h^{ref}} - P^h \end{aligned} \quad (24)$$

where R_e is a rotation matrix that expresses the orientation error, $R^{h^{ref}}$ is the desired orientation and R^h is obtained from the current transformation matrix $T_{r_i}^h$. P_e , $P^{h^{ref}}$ and P^h are respectively the positional error, reference position and actual position of the robot hands. All the rotation and position are expressed in the robot reference frame. These errors can then be used by the controller and by the local planner to adjust the footsteps and bring the system back to the desired state.

8. Experiments and Results

Before testing the system on real robots, the soundness of the controller was first asserted in simulation.

8.1. Controller Validation (Simulation)

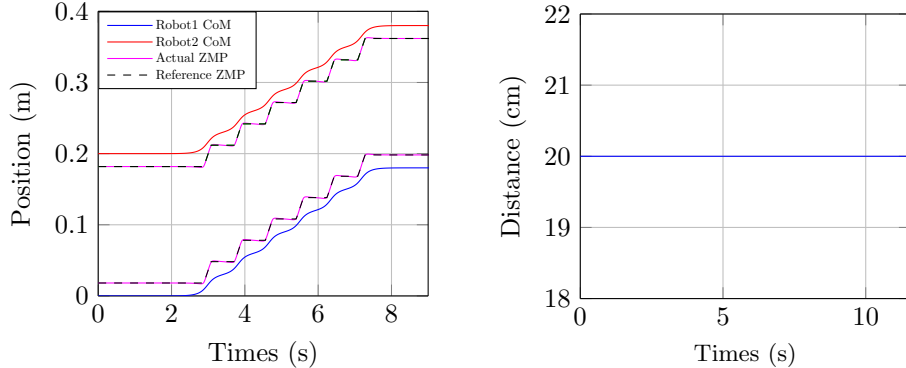
We have conducted two simulations:

- Case 1: The robots need to get closer or farther away from each other.
- Case 2: An external disturbance generates a rapid positional error between the robots.

The first case may be observed in a situation where the robot footprints are highly restricted, for instance in a cluttered environment. In this situation, the robots must remain stable despite the generated internal force. The second situation simply represents a case where one of the robot slips and an internal force is rapidly introduced in the system. The goal is then to bring the system back to the desired state while remaining stable.

The following parameters, which represent realistic values for our Nao robots, are used in simulation : $g = 9.81 \text{ m.s}^{-2}$, $T = 16.7 \text{ ms}$ ($F = 60 \text{ Hz}$ which is the frequency that we used in the real experiments on the Nao robots), $M_{c_{1,2}} = 5.1 \text{ kg}$, $K_{1,2} = 100 \text{ Nm}^{-1}$, $m_0 = 1 \text{ kg}$, $z_{c_{1,2}} = 0.315 \text{ m}$, $L = 0.20 \text{ m}$, $x_1^{href} = 0.1 \text{ m}$, $x_2^{href} = -0.1 \text{ m}$.

As for the controller, we used the MPC approach with the following parameters: $NT = 1.2 \text{ s}$, $Q_e = 1$, $R = 10^{-6}$, $Q_x = 0$ and $Q_d = 0$.



(a) Global ZMP and CoM trajectories in the sagittal plane (x -axis)

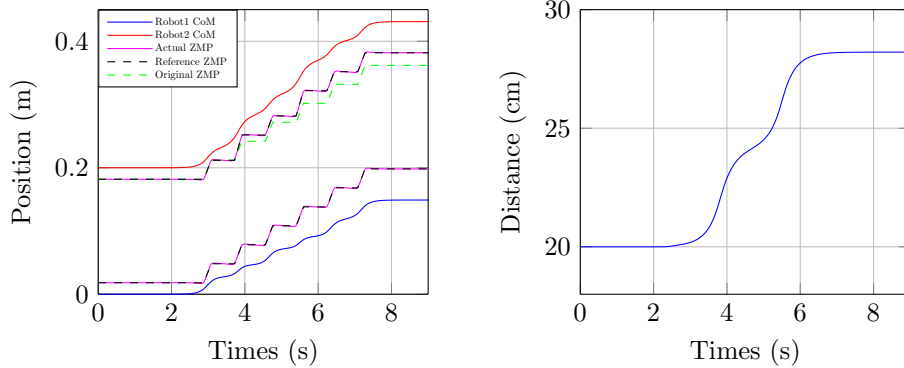
(b) The distance between the robots CoMs during the displacement

Figure 9: Simulation result for the optimal scenario where the robots are moving in a symmetrical way.

In all scenarios, the reference ZMP trajectories are generated so as to keep the ZMP in the middle of the support foot of each robot. Fig. 9 presents the ideal case where both robots are given symmetrical footstep commands at the same time. As can be seen, both robots are able to follow their respective reference ZMP trajectory and move in a perfectly symmetrical motion with their torsos remaining at the same relative distance during the whole trajectory.

8.1.1. Robustness to Planned Error

This situation represents the case where the robots cannot maintain a constant distance between them because of the environment. For example, the robot may need to get close or farther from each other to access a particularly restricted area. To simulate this case, the robot 2 executes bigger footsteps which results in an increased displacement of 2 cm in comparison to its partner as shown in Fig. 10. The relative displacement of robot 2 can be figured out by comparing its ZMP to the green dashed line which corresponds to the original position of the ZMP if it had executed symmetrical footsteps with respect to robot 1. Again, the results show that the robots are able to follow their respective reference ZMP trajectory despite the interaction force that is generated. However, in order to do so, both robots move their torsos far away from each other for a total displacement of 8 cm as shown in Fig. 10(b). As noted in Section 6.3, this behavior might generate problems since the robot arm movement is restricted.



(a) Global ZMP and CoM trajectories in the sagittal plane (x -axis). The dashed green line represents the position of the ZMP of robot 2 if it executed symmetrical footsteps.

(b) The distance between the robots CoMs during the displacement

Figure 10: Simulation where robot 2 executes two footsteps that are 1 cm longer than the footsteps of robot 1.

In order to reduce the torso displacement, we can increase the weight on the relative torso displacement in the minimization objective defined in section 6.3. For example, Fig. 11 presents the results of the same scenario as Fig. 10 except that Q_d was increased from 0 to 10^{-3} . As can be seen, the new objective function effectively reduced the relative torso displacement. An interesting observation is that the relative torso displacement was reduced by about 3 cm, or 1.5 cm per robot while introducing a ZMP tracking error of only 4 mm per robot.

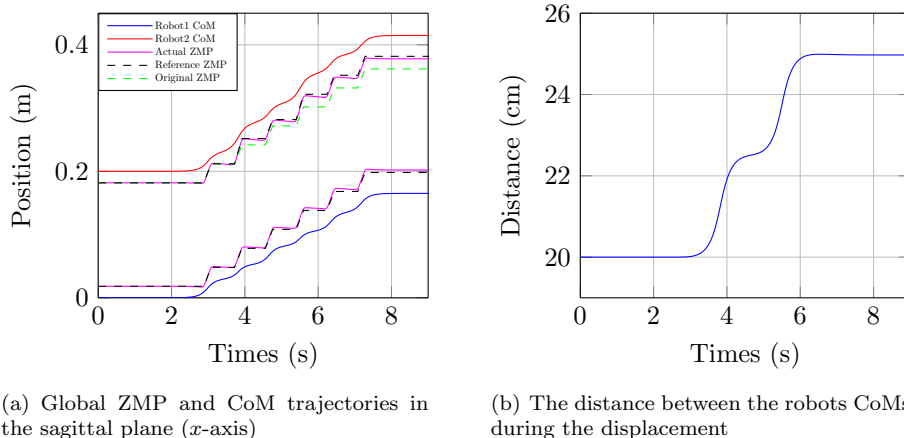
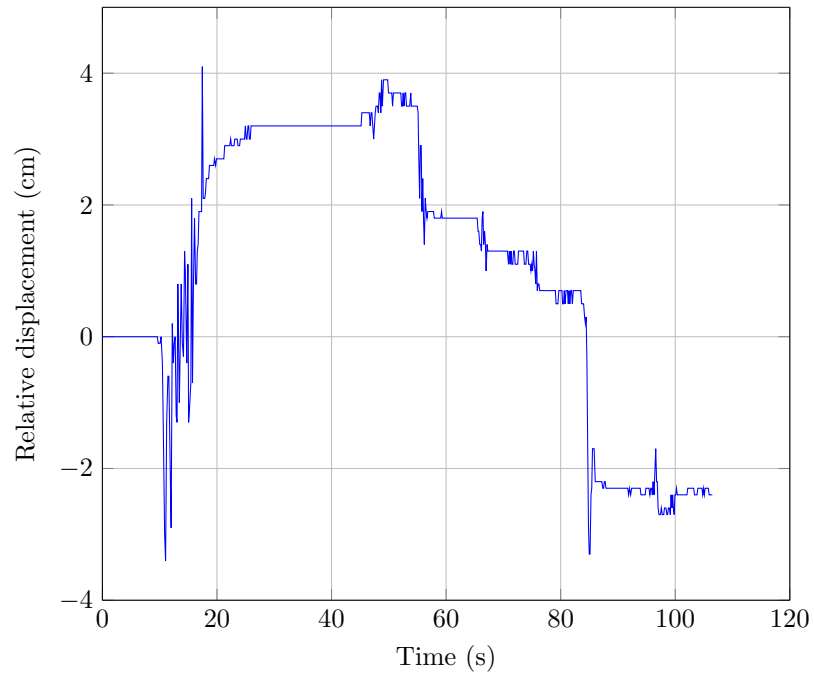


Figure 11: Same experiment as in Fig. 10 while minimizing the objective function (20) instead of (17).

8.1.2. Robustness to disturbance

A challenging problem with controlling the robot-load-robot system is the fact that positional errors are sometimes introduced rapidly in the system. For example, Fig. 12 shows the results of an experiment where two Nao robots were placed in front of each other and symmetrical footstep commands were sent to both of them. During the experiment, one of the robots was monitoring its relative distance from the other robot using an RGB-D Camera (Asus Xtion Pro-Live). Fig. 12 points out that the robots can't maintain a constant distance from each other even though they received identical commands. However, it is interesting to notice that the error increase incrementally in synchronization with the footsteps, this behavior is expected since the error mostly occurs when the robot foot lands on the floor. This kind of error is really problematic for the controller as instead of handling an error that increases gradually, it should handle relatively big and instantaneous variations of the error.

To simulate this case, let us analyze the controller behavior when an instantaneous error of 2 cm is added to the state of robot 2 (i.e. a 2 cm foot slip). As can be seen in Fig. 13, an undesirable motion of the robots CoMs is noticeable as the robots first move their torsos closer to each other, and then rapidly bring them backward. In this simulation, the reference ZMP trajectories are only moved to the new feet positions at the end of the preview window which is about two footsteps. A simple solution was to simply offset the reference ZMP trajectory by the error terms, thereby repositioning it at the current foot position. As can be seen in Fig. 14, using this simple solution greatly reduced the undesirable behavior in the CoM motion.

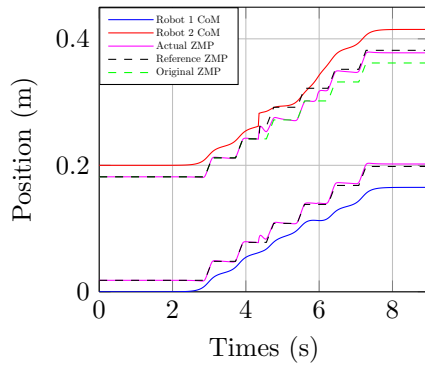


(a) Relative displacement between the robots

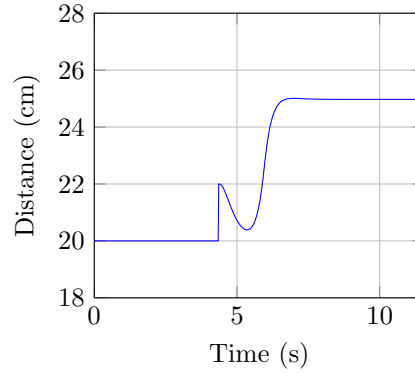


(b) Experimental setup

Figure 12: Relative displacement between the two robots while they walk freely and given the same commands.

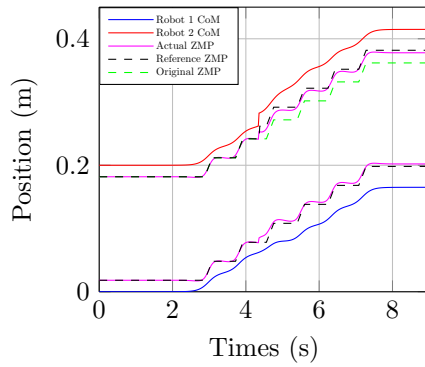


(a) Global ZMP and CoM trajectories in the sagittal plane (x -axis)

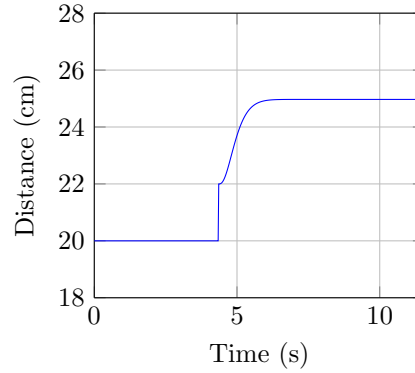


(b) The distance between the robots CoMs during the motion

Figure 13: Simulation result for a scenario where an instantaneous positioning error is added to robot 2



(a) Global ZMP and CoM trajectories in the sagittal plane (x -axis)



(b) The distance between the robots CoMs during the motion

Figure 14: Simulation result for a scenario where an instantaneous position error is added to robot 2 and the ZMP reference trajectory is offset in accordance with the detected error

8.2. Real Robot Experiments and Results

The framework was tested and validated on a team of two Nao humanoid robots transporting an object placed on a light and rigid table. The next subsections describe the details of the implementation on the Nao robot and the conducted experiments.

8.2.1. Compliance Control of Nao Robot Arms

As mentioned in Section 4, the arms compliance is a critical component of robot-robot cooperation. In that section, the most common active compliance implementation method relies on F/T sensors measurements was presented.

However, regarding the Nao robot, a major implementation challenge is the estimation of the external force applied to the robot using the available sensors. Although good results were obtained using the method presented in [13], the performance degrades during the cooperative transportation task. The main problem is that the force observer relies on the IMU, motor encoders and Force Sensitive Resistors (FSR) under the robot feet. However, during a cooperative transportation task, the interaction between the robots often causes a bad or lost contact of the feet on the floor which results in unreliable data from the FSR.

Thereby, to implement a compliance control of Nao arms, the force observer [13] was used in combination with a joint level compliance strategy. It is possible to estimate the magnitude and direction of an external force applied to a manipulator if an accurate dynamic model of the robot is available and if the torques applied by the motors are known [22, 23]. The approach can also be extended to humanoid robots [24], albeit it is more difficult since this type of robot is not rigidly attached to a base.

First, the torque due to the robot mass and transported object must be isolated. To do so, it is possible to use the following inverse dynamic equation:

$$\tau = B(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + J^T(q)F \quad (25)$$

where J is the Jacobian matrix of the considered chain, τ is the joint torques, $B(q)$ is the inertia matrix, $C(q, \dot{q})\dot{q}$ includes the Coriolis and centrifugal forces, $g(q)$ is the gravity terms, F is the external force applied at the end-effector, and q, \dot{q}, \ddot{q} are respectively the position, velocity and acceleration of the joints.

The equation can be simplified since we estimate the torque when the arms are not moving ($\dot{q} = \ddot{q} = 0$) and we do not consider external forces. Therefore, we obtain:

$$\hat{\tau} = g(q) \quad (26)$$

The external force can then be easily obtained by subtracting this torque from the measured one to get the external force applied to the humanoid arms:

$$\tau_m - \hat{\tau} = J^T(q)F \quad (27)$$

where τ_m is the measured torques.

In [24], the authors applied the above approach to a Nao robot to estimate a force applied anywhere on the body of the robot. Although great results were obtained in simulation, the implementation on a real Nao robot was rather inaccurate mostly because of the large static friction at the joints of the robot. However, in our case, we can greatly simplify the problem since we know the point of application of the interaction force. In a natural body configuration, in which the upper and lower arms are perpendicular as depicted in Fig. 1, an external force applied to the hands in the x -axis can be observed using only a single motor in the arm.

We decided to implement the arms compliance as an error minimization problem at the joint level. Referring to Nao robots official documentation [3],

we considered the shoulder pitch motor position error as an estimation of the interaction force. This is because the Nao arm motors are controlled through a PD controller, making it possible to determine the motor output torque with the following linear approximation [24]:

$$\tau_m = K (q_d - q) \quad (28)$$

where $(q_d - q)$ is the position error and K is a constant coefficient that depends on multiple parameters of the joint motor (e.g. reduction ratio). Compared to our observer [13], the motor error signal $(q_d - q)$ is less noisy, and more importantly only detects the forces that are applied to the robot hands. However, as pointed out in [24], the error signal is only meaningful with a large interaction force that produces a relatively high signal to noise ratio. Therefore, this strategy was used in conjunction with our force observer as depicted in Fig. 15.

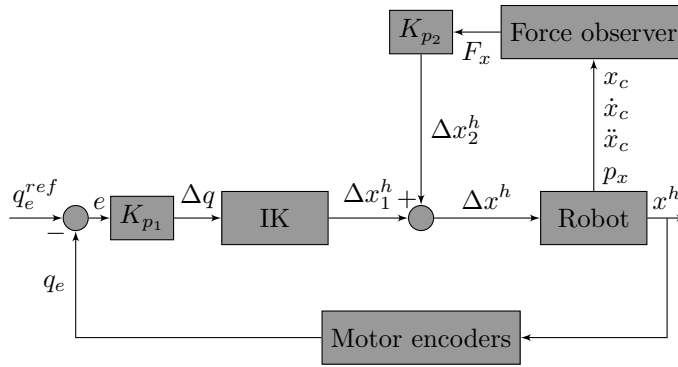


Figure 15: Arm Compliance Controller

Fig. 15 shows the control loop used to implement a compliant behavior of the Nao robot arms. The main component is a feedback loop that keeps the shoulder pitch motor position error (q_e) close to a reference value (q_e^{ref}) by moving the arms/hands (x^h) in the direction of the errors. K_{p1} and K_{p2} are proportional gains and IK stands for Inverse Kinematics algorithm that converts the shoulder displacement (q) into an equivalent hand displacement (Δx^h), the aim being to move the shoulders while keeping the hands in a parallel plane with respect to the floor. The secondary component is a feedforward compensation that is computed using the output from our external force observer [13], F_x . The estimated force F_x is converted into a relative hands displacement (Δx_2^h) using a proportional gain and injected into the main loop. This architecture increases the controller robustness as the arms will only move rapidly if a force is detected in both the joint and the operational spaces. However, because of the modeling errors and in order to have a stable compliance controller, we added threshold values to both e and F_x , that means if e and F_x are within a set of thresholds, their values are not used and simply considered as 0.

The performance of the compliance controller was validated by a spring-based force sensor as well as by our external force observer [13]. As shown

in Fig. 16, a stiffness coefficient of 50 Nm^{-1} seems to model the compliance mechanism reasonably well, therefore we used that value in our implementation.

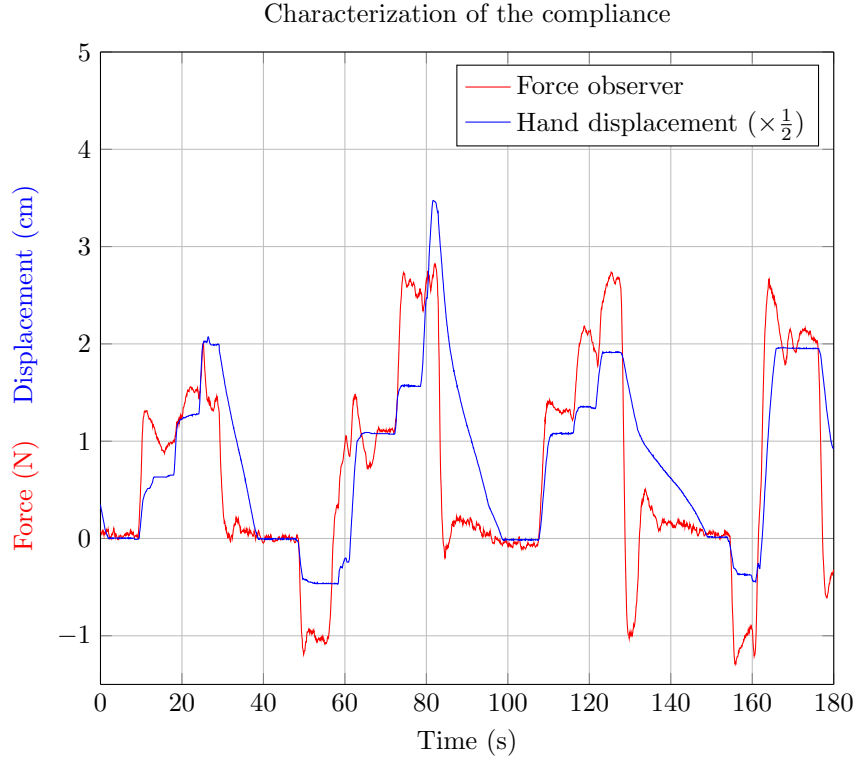


Figure 16: Displacement of the arms and the estimated external force using [13]

8.2.2. Estimating the mass

The limited sensor set of the Nao robot has motivated the development of a model-based controller instead of solely relying on sensor measurements to estimate the system state vector. However, similarly to the robots relative distance estimation during the execution of the task, the transported mass is also an important part of the model that must also be estimated.

To this end, we used the external force observer described in [13]. The transported mass could then be directly estimated using the observed force along the vertical axis, F_z , as follows:

$$m_0 = \frac{F_z}{g} \quad (29)$$

Fig. 17 presents the force observer outputs of one of the robots when they are immobile and a mass of 1.13 kg is added in the center of the transported

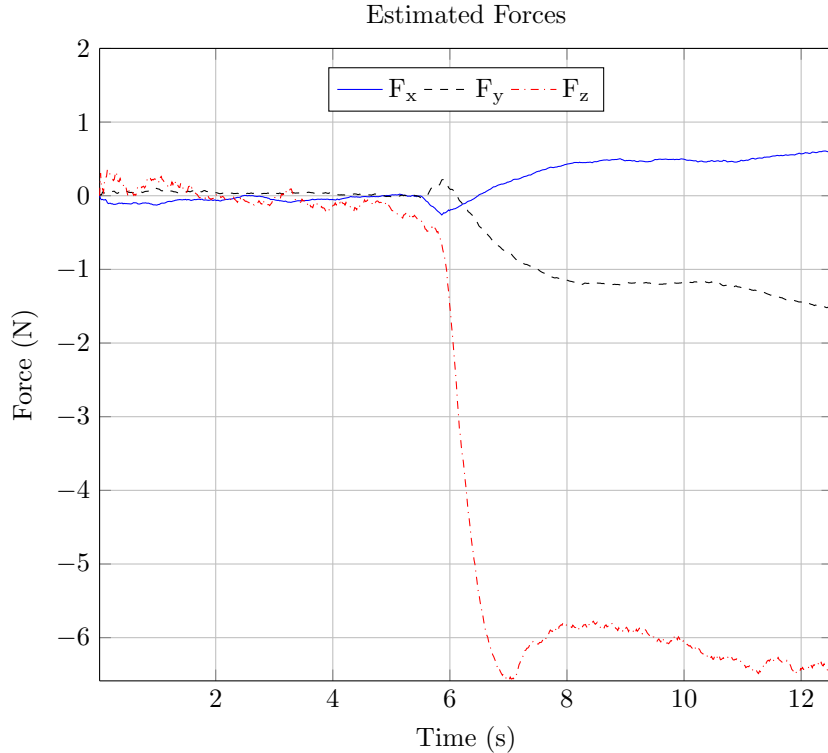


Figure 17: Estimated force by the force observer [13] when a mass of 1.13 kg is placed on the transported table.

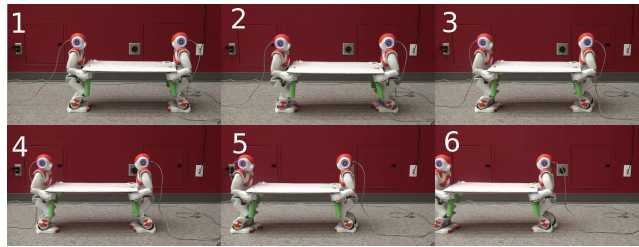
rigid table. As can be seen, F_z is estimated at around 6 N whereas the expected value is 5.54 N by assuming that the mass is uniformly split between the two robots.

Once the mass is found, the optimal feedback gains (or the matrices used in the QP formulation) are computed once and can then be used for the whole task. It is also possible to precompute a set of gains for different masses as shown in [15].

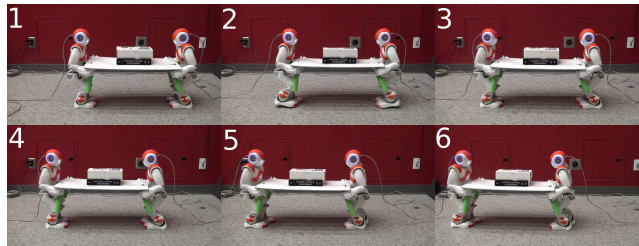
8.2.3. Experiments

We validated the proposed framework through several experiments of transporting a variety of objects ranging in size and in mass (0-2.13Kg). A selection of representative tests is shown in the video attached to this paper. During all experiments, the robots used a light cardboard table, of about 60 cm in length and 20 cm in width, and moved using a pace gait (legs on the same side move at the same time) to carry numerous objects as depicted in Fig. 18.

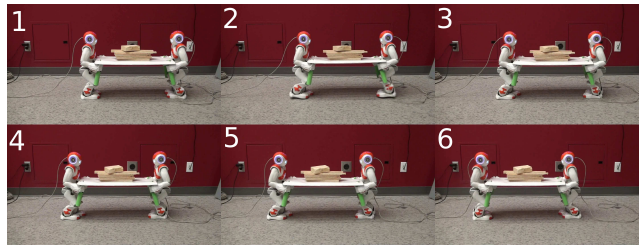
The experiments were carried out at low-speed ($\approx 2\text{cm} \cdot \text{s}^{-1}$) in large part because of the important oscillation in the joints of the robots that occurs



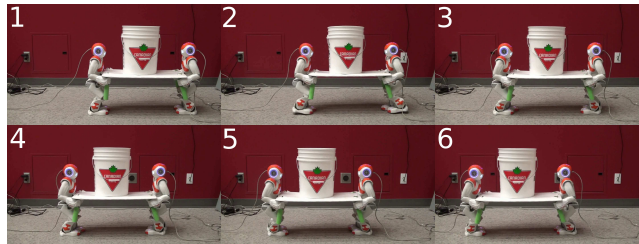
(a) Empty table



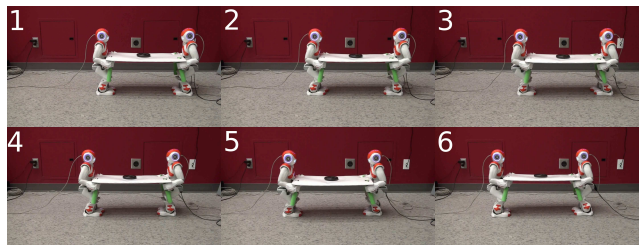
(b) Books



(c) Wood scraps



(d) Large bucket



(e) A mass of 2.23 Kg

Figure 18: Experiments with two Nao robots using a table to transport a variety of objects.

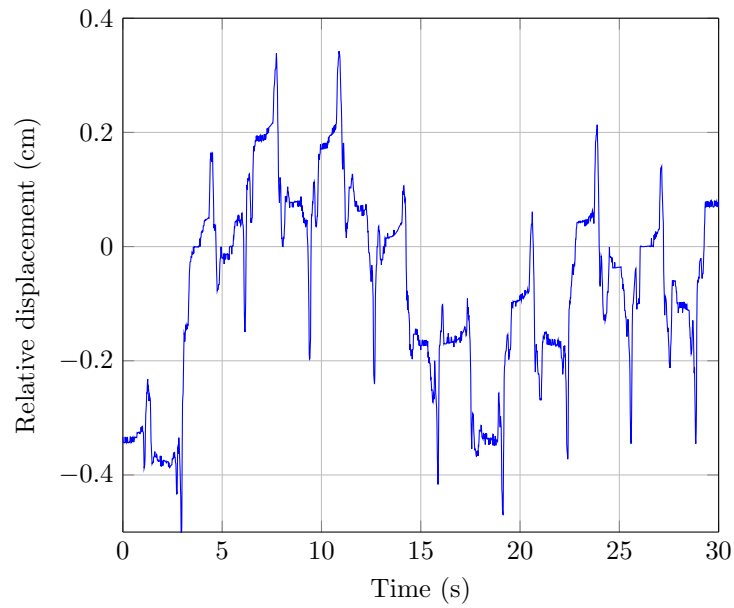
when transporting a heavy object. Fig. 19 and Fig. 20 respectively present the relative displacement between the robots and the robot 1 ZMP during two experiments. As can be seen, the robots displacement is synchronized in both cases as the distance between the robots do not vary much. Also, the walk is stable as the ZMP is kept well within the support region for most of the motions thereby validating our controller.

A problem that we have encountered during the experiments is that sometimes an important slipping of the robots feet occurred during the execution of the task. Although the stability of the system was always maintained, the speed and performance could certainly be enhanced by developing a stabilizer that is consistent with the controller proposed in this paper. Lastly, as can be seen in the screenshots of the experiments (Fig. 18), the table is not completely rigid and it sometimes bent because of the weight and its legs did not stay parallel. As a result, the performance of the arm controller was degraded, and the arms motion range was quite small ($\approx 5cm$).

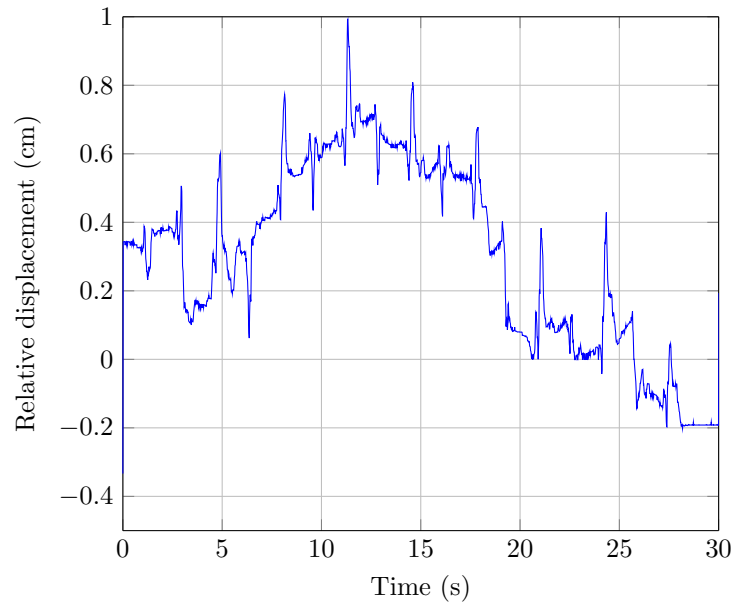
9. Conclusion

In this paper, we presented a control framework for the cooperative transportation of a heavy or large object by two humanoid robots. The main building blocks are a local footstep planner, a compliant control of the arms and a ZMP-based pattern generator. The ZMP controller uses a complete dynamic model of the task to ensure the stability of the system. This controller formulation is particularly interesting for smaller humanoid robots in comparison to sensor-based approaches that rely on measurements from force/torque sensors.

Cooperative transportation by humanoids is a topic that has received relatively few attention despite its relevance in multiple scenarios. This article explored both the theoretical and practical sides of the problem. Even though the focus was on the actual implementation on the Nao robotic platform, we also discussed how the mass of the transported object and the interaction force model impact the stability of the robots during the execution of the task. These results can be generalized to other humanoid robots and can help in designing a controller that makes the most out of the used platform. Notably, an interesting option would be to unify the arms control and the ZMP preview controller into a single module, thereby providing the ability to modify the interaction model (stiffness coefficient K) to stabilize the system. Another interesting option would be to give the controller the ability to modify the footsteps during execution to increase the responsiveness of the controller. Moreover, replacing the predicted state by the measured one as a feedback in the controller would improve its robustness. Finally, a limiting factor to the performance of the approach on the Nao was the important joint oscillation and foot slipping due to the physically challenging nature of the task. As mentioned briefly before, we believe the system would greatly benefit from the development of a low-level stabilizer for the Nao and it is a priority for future work.

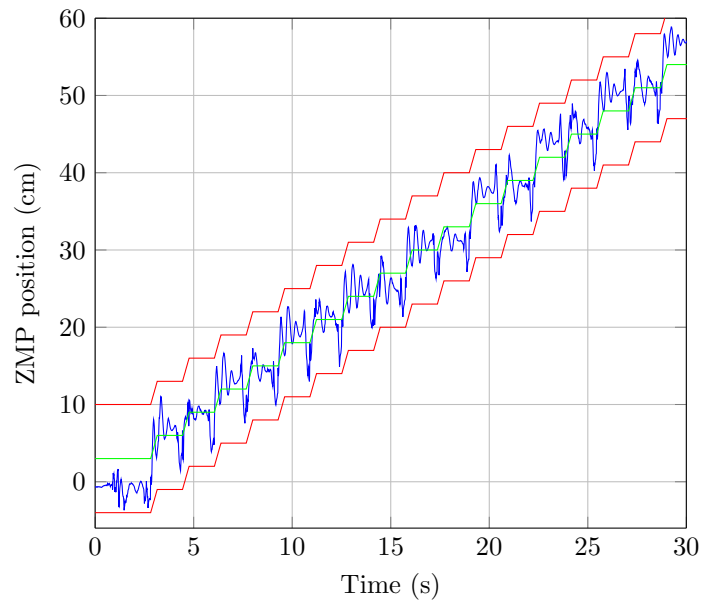


(a) An empty table

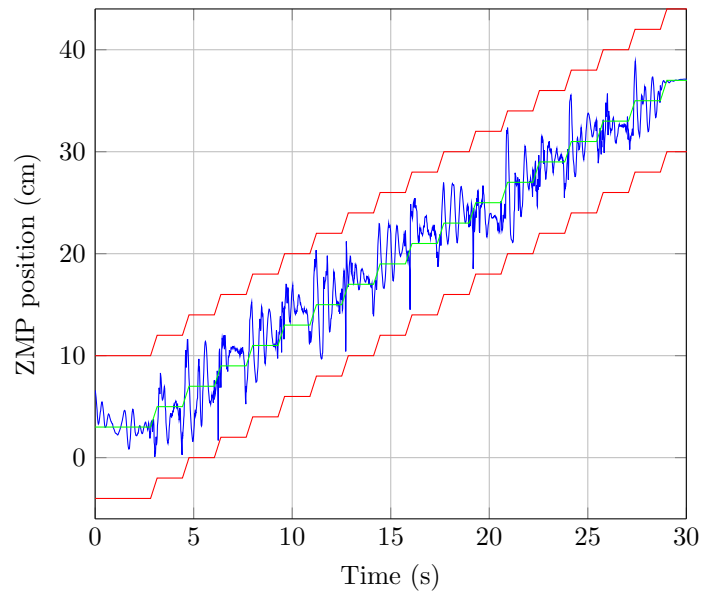


(b) A transported mass of 2.3 Kg

Figure 19: The relative displacement of the robots during two experiments



(a) Empty table



(b) A mass of 2.3 Kg

Figure 20: Robot 1 ZMP during two experiments. The stability limits (in red) represent the size of the robot foot, the desired ZMP position (in green) is located in the middle of the support foot.

Acknowledgment

This research is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under the grant RGPIN-2012-419406.

10. References

- [1] Agravante, D.J., Cherubini, A., Bussy, A., Gergondet, P., Kheddar, A., 2014. Collaborative human-humanoid carrying using vision and haptic sensing. *IEEE International Conference on Robotics and Automation (ICRA)*, 607–612.
- [2] Agravante, D.J., Sherikov, A., Wieber, P.B., Cherubini, A., Kheddar, A., 2016. Walking pattern generators designed for physical collaboration. *IEEE International Conference on Robotics and Automation (ICRA)*, 1573–1578.
- [3] Aldebaran, 2016. Nao official documentation. URL: http://doc.aldebaran.com/2-1/home_nao.html.
- [4] Berger, E., Vogt, D., Haji-Ghassemi, N., Jung, B., Amor, H.B., 2015. Inferring guidance information in cooperative human-robot tasks, in: *IEEE-RAS International Conference on Humanoid Robots*, pp. 124–129.
- [5] Bouyarmane, K., Kheddar, A., 2012. Humanoid Robot Locomotion and Manipulation Step Planning. *Advanced Robotics* 26.
- [6] Bussy, A., Gergondet, P., Kheddar, A., Keith, F., Crosnier, A., 2012. Proactive behavior of a humanoid robot in a haptic transportation task with a human partner, in: *IEEE International Workshop on Robot and Human Interactive Communication*, pp. 962–967.
- [7] Choi, H., Lee, S., Jin, T., Lee, S.H., 2015. Trajectory-free reactive stepping of humanoid robots using momentum control, in: *IEEE-RAS International Conference on Humanoid Robots*, pp. 1173–1178.
- [8] Don Joven Agravante, Andrea Cherubini, Alexander Sherikov, Pierre-Brice Wieber, A.K., 2017. Human-Humanoid Collaborative Carrying. Submitted to *IEEE Transactions on Robotics*.
- [9] Ferreau, H., Kirches, C., Potschka, A., Bock, H., Diehl, M., 2014. qpOASES: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation* 6, 327–363.
- [10] Garimort, J., Hornung, A., Bennewitz, M., 2011. Humanoid navigation with dynamic footstep plans, in: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3982–3987.

- [11] Griffin, R.J., Wiedebach, G., Bertrand, S., Leonessa, A., Pratt, J., 2017. Walking Stabilization Using Step Timing and Location Adjustment on the Humanoid Robot, Atlas, in: IEEE International Conference on Intelligent Robots and Systems (IROS), pp. 667–673. [arXiv:1703.00477](https://arxiv.org/abs/1703.00477).
- [12] Harada, K., Kajita, S., Saito, H., Morisawa, M., Kanehiro, F., Fujiwara, K., Kaneko, K., Hirukawa, H., 2005. A humanoid robot carrying a heavy object, in: IEEE International Conference on Robotics and Automation (ICRA), pp. 1712–1717.
- [13] Hawley, L., Rahem, R., Suleiman, W., 2018. Kalman filter based observer for an external force applied to medium-sized humanoid robots, in: To appear in IEEE International Conference on Intelligent Robots and Systems. URL: <https://goo.gl/yW3K52>.
- [14] Hawley, L., Suleiman, W., 2016. External Force Observer for Medium-sized Humanoid Robots, in: IEEE-RAS International Conference on Humanoid Robots, pp. 366–371.
- [15] Hawley, L., Suleiman, W., 2017. Control Strategy and Implementation for a Humanoid Robot Pushing a Heavy Load on a Rolling Cart, in: IEEE International Conference on Intelligent Robots and Systems (IROS), pp. 4997–5002.
- [16] Herdt, A., Diedam, H., Wieber, P.B., Dimitrov, D., Mombaur, K., Diehl, M., 2010. Online Walking Motion Generation with Automatic Footstep Placement. *Advanced Robotics* 24.
- [17] Hornung, A., Dornbush, A., Likhachev, M., Bennewitz, M., 2012. Anytime search-based footstep planning with suboptimality bounds, in: IEEE-RAS International Conference on Humanoid Robots, pp. 674–679.
- [18] Inoue, Y., Tohge, T., Iba, H., 2003. Cooperative Transportation by Humanoid Robots: Learning to Correct Positioning. *Design and Application of Hybrid Intelligent Systems* , 1124–1134.
- [19] Kajita, S., Hirukawa, H., Harada, K., Yokoi, K., 2014. Introduction to Humanoid Robotics. *Springer Tracts in Advanced Robotics*. chapter Biped Walking. pp. 142 – 147.
- [20] Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., Hirukawa, H., 2003. Biped Walking Pattern Generation by using Preview Control of Zero-Moment Point, in: IEEE International Conference on Robotics and Automation (ICRA), pp. 1620–1626.
- [21] Keith, F., Wieber, P., Mansard, N., Kheddar, A., 2011. Analysis of the discontinuities in prioritized tasks-space control under discreet task scheduling operations, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3887–3892.

- [22] Magrini, E., Flacco, F., De Luca, A., 2014. Estimation of contact forces using a virtual force sensor, in: IEEE International Conference on Intelligent Robots and Systems (IROS), pp. 2126–2133.
- [23] Magrini, E., Flacco, F., De Luca, A., 2015. Control of generalized contact motion and force in physical human-robot interaction, in: IEEE International Conference on Robotics and Automation (ICRA), pp. 2298–2304.
- [24] Mattioli, T., Vendittelli, M., 2017. Interaction Force Reconstruction for Humanoid Robots. IEEE Robotics and Automation Letters 2, 282–289.
- [25] McGill, S.G., Lee, D.D., 2011. Cooperative humanoid stretcher manipulation and locomotion. IEEE-RAS International Conference on Humanoid Robots ,
- [26] Motahar, M.S., Veer, S., Huang, J., Poulakakis, I., 2015. Integrating dynamic walking and arm impedance control for cooperative transportation, in: IEEE International Conference on Intelligent Robots and Systems (IROS), pp. 1004–1010.
- [27] Rioux, A., Esteves, C., Hayet, J.B., Suleiman, W., 2015. Cooperative SLAM-Based Object Transportation by Two Humanoid Robots in a Cluttered Environment, in: IEEE-RAS International Conference on Humanoid Robots, pp. 331–337.
- [28] Rioux, A., Esteves, C., Hayet, J.B., Suleiman, W., 2017. Cooperative Vision-Based Object Transportation by Two Humanoid Robots in a Cluttered Environment. International Journal of Humanoid Robotics 14, 1–30.
- [29] Rioux, A., Suleiman, W., 2018. Autonomous SLAM based humanoid navigation in a cluttered environment while transporting a heavy load. Robotics and Autonomous Systems 99, 50–62.
- [30] Stasse, O., Evrard, P., Perrin, N., Mansard, N., Kheddar, A., 2009. Fast foot prints re-planning and motion generation during walking in physical human-humanoid interaction, in: IEEE-RAS International Conference on Humanoid Robots, pp. 284–289.
- [31] Stephens, B., 2008. Humanoid push recovery, in: IEEE-RAS International Conference on Humanoid Robots, pp. 589–595.
- [32] Wieber, P.B., 2006. Trajectory free linear model predictive control for stable walking in the presence of strong perturbations, in: IEEE-RAS International Conference on Humanoid Robots, pp. 137–142.
- [33] Wu, M.H., Konno, A., Ogawa, S., Komizunai, S., 2014. Symmetry cooperative object transportation by multiple humanoid robots. IEEE International Conference on Robotics and Automation (ICRA) ,

- [34] Wu, M.H., Ogawa, S., Konno, A., 2016. Symmetry position/force hybrid control for cooperative object transportation using multiple humanoid robots. *Advanced Robotics* 30, 131–149.
- [35] Yokoyama, K., Handa, H., Isozumi, T., Fukase, Y., Kaneko, K., Kanehiro, F., Kawai, Y., Tomita, F., Hirukawa, H., 2003. Cooperative works by a human and a humanoid robot, in: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2985–2991.

Appendix A. Proof of Eq (2):

Let the ZMP on the horizontal ground be given by the following vector:

$$\mathbf{p} = [p_x \quad p_y]^T \quad (\text{A.1})$$

where p_x and p_y are respectively the projections of the ZMP on the x and y axis. Let us suppose that the robot CoM is subject to external force \mathbf{F}^{ext} and external moment $\boldsymbol{\tau}^{ext}$.

To compute \mathbf{p} , one can use the following formula:

$$\mathbf{p} = \mathbf{N} \frac{\mathbf{n} \times \boldsymbol{\tau}^o}{(\mathbf{f}^o | \mathbf{n})} \quad (\text{A.2})$$

where the operators \times and $(\cdot | \cdot)$ design respectively the cross and scalar products, and

- \mathbf{N} is a constant matrix

$$\mathbf{N} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (\text{A.3})$$

- the vector \mathbf{n} is the normal vector on the horizontal ground ($\mathbf{n} = [0 \quad 0 \quad 1]^T$).
- The vector \mathbf{f}^o is the result of the gravity, inertial and external forces:

$$\mathbf{f}^o = M_c \mathbf{g} - M_c \ddot{\mathbf{X}}_c + \mathbf{F}^{ext} \quad (\text{A.4})$$

where \mathbf{g} denotes the gravity acceleration ($\mathbf{g} = -g\mathbf{n}$), M_c is the total mass of the humanoid robot, and $\ddot{\mathbf{X}}_c = [\ddot{x}_c \quad \ddot{y}_c \quad \ddot{z}_c]^T$ is the acceleration of the CoM. $\mathbf{F}^{ext} = [F_x \quad F_y \quad F_z]^T$ is the external force applied to the humanoid robot.

The normal force is given by the following expression:

$$(\mathbf{f}^o | \mathbf{n}) = f_n^o = -M_c g - M_c \ddot{z}_c + F_z \quad (\text{A.5})$$

- $\boldsymbol{\tau}^o$ denotes the moment of the external force and the moment of the force \boldsymbol{f}^o about the origin of the fixed world frame in addition to the external moment $\boldsymbol{\tau}^{ext}$, it is given by the following expression:

$$\boldsymbol{\tau}^o = \boldsymbol{X}_c \times \boldsymbol{f}^o + \boldsymbol{\tau}^{ext} \quad (\text{A.6})$$

where $\boldsymbol{X}_c = [x_c \quad y_c \quad z_c]^T$ is the Cartesian position of CoM.

Thus, we obtain:

$$\begin{aligned} \boldsymbol{p} &= \begin{bmatrix} p_x \\ p_y \end{bmatrix} \\ &= \begin{bmatrix} x_c + \frac{M_c z_c}{f_n^o} \ddot{x}_c - \frac{z_c}{f_n^o} F_x - \frac{1}{f_n^o} \tau_y^{ext} \\ y_c + \frac{M_c z_c}{f_n^o} \ddot{y}_c - \frac{z_c}{f_n^o} F_y + \frac{1}{f_n^o} \tau_x^{ext} \end{bmatrix} \end{aligned} \quad (\text{A.7})$$