



Humanoid navigation and heavy load transportation in a cluttered environment

Antoine Rioux, Wael Suleiman

► To cite this version:

Antoine Rioux, Wael Suleiman. Humanoid navigation and heavy load transportation in a cluttered environment. 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sep 2015, Hambourg, Germany. pp.2180-2186, 10.1109/IROS.2015.7353669 . hal-01521584

HAL Id: hal-01521584

<https://hal.science/hal-01521584>

Submitted on 12 May 2017

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.

Humanoid Navigation and Heavy Load Transportation in a Cluttered Environment

Antoine Rioux and Wael Suleiman

Abstract—Although in recent years several studies aimed at the navigation of robots in cluttered environments, just a few have addressed the problem of robots navigating while moving a large or heavy object. This is especially useful when transporting loads with variable weights and shapes without having to change the robot hardware. On one hand, a major advantage of using a humanoid robot to move an object is that it has arms to firmly grasp it and control it. On the other hand, humanoid robots tend to have higher drift than their wheeled counterparts as well as having significant lateral swing while walking, which propagates to anything they carry. In this work, we present algorithms for a humanoid robot navigating in a cluttered environment while pushing a cart-like object. In addition, the algorithms make use of the hands and arms to articulate the cart when executing tight turns using whole body control scheme to reduce the lateral swing effect on the load and ensure a safe transport. Experiments conducted on a real Nao robot assessed the proposed approach and algorithms, they show that the payload of a humanoid robot can be significantly increased without changing the humanoid robot's hardware, and therefore enact the capacity of humanoid robots in real-life situations.

I. INTRODUCTION

One of the advantages of having arms on a robot is that it can carry a load. This capacity can be useful for a wide range of actions, including transporting objects from one place to another. However, the maximum payload is generally pretty low and generates a lot of instability if it is held at the arm's length. While it is possible to increase the strength of the motors in the legs and arms, it is not the best solution since a motor's power is proportional to its size, weight and price. Instead of putting the entire load directly on the robot, a cart-like object can be used to help supporting the weight. The cart can then be pushed by the robot and moved around more easily without having to modify the robot's hardware to fit the load.

Many researches have been done on navigating robots in a cluttered environment, but adding a controllable object supporting a heavy load and navigating it safely has not been widely examined. Furthermore, using a humanoid robot, which has unstable balance and great motion swinging, increases the difficulty of the task in hand. The objective is to plan a stable and safe trajectory that uses a whole body control scheme.

To address the problem of navigating a cart-like object supporting a load with a humanoid robot in a cluttered environment, the following sub-problems should be solved:

Antoine Rioux and Wael Suleiman are with Electrical and Computer Engineering Department, the Faculty of Engineering, University of Sherbrooke, Sherbrooke, Canada {antoine.rioux, wael.suleiman}@usherbrooke.ca

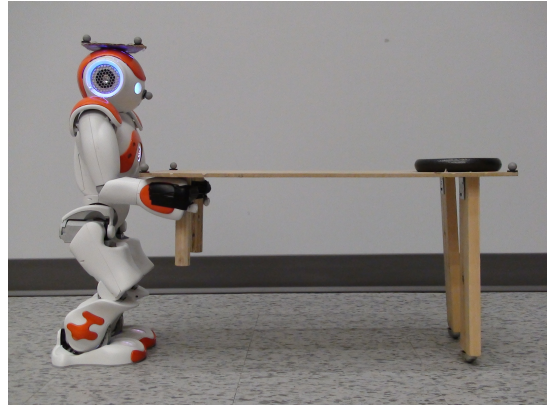


Fig. 1. The Nao robot holding the cart-like object

(I) planning, (II) sensing and (III) controlling the cart-like object. To plan a safe trajectory between obstacles, an anytime search-based planner exploits a given set of motion primitives that consider both the robot and the cart footprint. The second step is to find the humanoid robot's footprints, the humanoid's feet and hands trajectories are then computed in order to minimize the swing effect and follow the cart trajectory using a task priority whole body control scheme.

The main contribution of the paper is providing a framework for humanoid navigation in a cluttered environment while manipulating a cart-like object that carries a heavy load.

This paper is organized as follows. Section 2 presents an overview of related works. Section 3 describes the robot-cart model and the implemented algorithms. In Section 4, simulation and real results are presented and analyzed.

II. RELATED WORK

Many studies have been done on robots moving objects to a specific goal. Most of them though are executed with multiple wheeled robots that position themselves around the object to push it in the desired direction. One of the first examples of this behavior for a movable target tried to reproduce how ants move a prey bigger than them by trial and error [1]. Another added the possibility of pulling the object by using a rudimentary one degree of freedom arm [2], while [3] have the additional benefit of being able to manipulate an object in 3D when necessary, instead of 2D. In these researches, the manipulator comes in contact with the object at only one point, which barely allows any control over the object while moving and manipulating it. Holonomic wheeled robots are indeed less complex to control than humanoids, they are mainly useful at sliding box-like object

on the ground. Therefore, a robot with humanoid arms allow a better control of the structure of an object and is more suitable to control cart-like objects with loads.

Other works have explored the usage of humanoid robots that push objects while keeping a firm grip on the handles. In theory, two arms are enough to fully constrain and control all the degrees of freedom of a cart. It was demonstrated that Honda's ASIMO is capable of moving a large cart in rooms and hallways [4]. However, the cart is mostly controlled as a Dubins car instead of taking advantage of the full possibility of the humanoid robot's holonomic movement. Also, the arms serve only as a mean of attaching the cart to the robot and are not taken into consideration to control the cart further.

Another project used a biped robot to push a person on a wheelchair [5]. To keep balance while performing the desired task, this HRP-2 and the ASIMO both implement an approach with a Zero-Moment Point (ZMP) offset. Although the HRP-2 has a heavy load to move on the wheelchair, the possible movements are akin to those of the ASIMO robot as the cart is controlled like a Dubins car. Furthermore, none of these examples minimize the footprint of the robot-cart contraption by turning the object because they are considered as one static bloc. In a tight and cluttered environment, taking advantage of the arms to manipulate the cart-like object is beneficial and possibly necessary.

In [6], a planning method for humanoids to navigate among movable obstacles has been proposed. The main purpose of that method is to find a path from a starting to a goal points in a complex environment where the robot can easily move objects to create a clear path, if it exists one. Our objective is however different, as we are interested in not only navigating in a cluttered environment, but also transporting a heavy load that is significantly bigger than the humanoid payload.

In the work of [7], a PR2 robot possessing a wheeled holonomic base and two 7 DoF (Degree of Freedom) arms is used to push a small cart and control its orientation. However, despite having humanoid arms to orient the cart, since the PR2 has wheels instead of legs, no lateral swing is transmitted to the transported object causing oscillations and instability, which is a problem with humanoid robots. Furthermore, the cart is very small and light weight with respect to the PR2, in contrast to our proportionally big cart that is able to carry a load heavier than the robot itself.

III. ROBOT-CART MODEL AND ALGORITHMS

A. Path finding

To be able to navigate through a cluttered environment, a path provided by a motion planning algorithm is essential. Among the different possibilities, we chose a lattice-based graph planning with an ARA* search [8]. This choice is mainly motivated by the use of motion primitives that assures feasible robot-cart configurations and transitions. The environment is modeled by a 2D grid costmap that discriminates obstacles from free space at a fixed threshold and allow obstacles inflations to increase the security margin.

Each node of the search graph needs a complete state representation of the robot-cart to properly operate. To achieve this, it is possible to model the state in $\mathbb{R}^2 \times \mathbb{S}^1 \times \mathbb{R}^2 \times \mathbb{S}^1$:

$$s = (x_r, y_r, \theta_r, x_{ca}, y_{ca}, \theta_{ca}) \quad (1)$$

Where x_r, y_r and θ_r are the positions and orientation of the robot, x_{ca}, y_{ca} and θ_{ca} are those of the cart. As the working space of our robot's arms is too small to fully take advantage of both rotation and translation, the problem can be simplified by setting a pivot point positioned in the middle of both hands to reduce the dimensionality of the search space, the chosen position of the pivot point also maximizes the rotation range within the robot's workspace (see **pivot point 1** in Fig. 5), resulting in a 4 dimensions state space $\mathbb{R}^2 \times \mathbb{S}^1 \times \mathbb{S}^1$:

$$s = (x_r, y_r, \theta_r, \theta_{ca}) \quad (2)$$

Even-though the above simplification removes the ability of the cart to translate on the plane, the robot retains enough manipulability to minimize the cart footprint on tight turns.

In a lattice-based graph planner, the transition between the nodes is a discrete action chosen within a fixed-set of possible actions called motion primitives. An important feature of the lattice representation is that each of those connections is a feasible path, in contrast to other forms of graph search, including "4-connected" or "8-connected" grid. This makes it really suitable for highly constrained systems, such as a robot moving a cart.

Because the cart can carry different loads, multiple sets of primitives are needed depending on the load's weight. Without load, the robot is holonomic and can move in any direction. A subset of movements composed of forward, backward, diagonal, rotate in place and turn while moving forward is used to reduce planning time while focusing on forward movements. With a heavy load though, moving sideways and rotating in place becomes really difficult because of the increased friction. For this reason, when the weight becomes too important, rotation occurs around a pivot point situated between the two table's legs touching the ground (see **pivot point 2** in Fig. 5). Thus, changing the feasible primitives is necessary for the lattice representation to remain coherent. An example of right turn for the omnidirectional and the heavy load sets of primitives are presented in Fig. 2.

The cost function of a transition from state s to s' is based on the time to execute that transition and is computed as follows:

$$Cost = \begin{cases} \frac{\sqrt{(\Delta x_r)^2 + (\Delta y_r)^2}}{\dot{r}^+} \times DF & \text{if } \Delta x_r \neq 0 \text{ or } \Delta y_r \neq 0 \\ \frac{\Delta \theta_r}{\dot{\theta}^+} \times DF & \text{otherwise} \end{cases} \quad (3)$$

where $\Delta x_r, \Delta y_r$ and $\Delta \theta_r$ are the differences between the x_r, y_r and θ_r , which are the coordinates of the robot's pelvis joint, between states s and s' , DF is a difficulty factor associated with each primitives, \dot{r}^+ is the maximal robot linear velocity and $\dot{\theta}^+$ is the maximal angular velocity for turning in place. The Euclidean distance between both states is computed and then divided by the maximum velocity

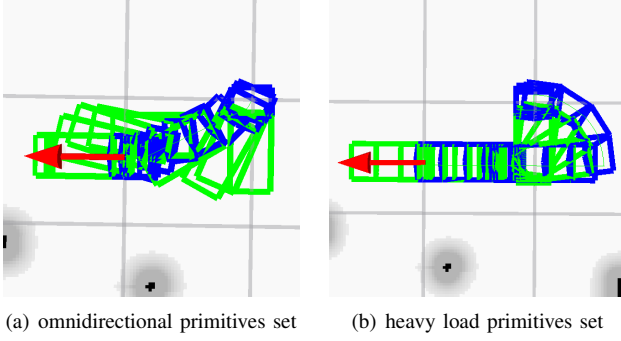


Fig. 2. A right turn executed by the robot (blue) pushing the cart (green) to the goal (red arrow) with both sets of primitives.

of the robot in the direction of the movement to give the approximate time to execute the primitive.

For both instances, the difficulty-factor associated with each primitive is then multiplied by the time cost. This DF is used to prioritize or penalize certain motions or directions, which result in a smoother and a more natural looking trajectory. For example, turning in place then moving forward takes a longer time than moving in diagonal. However, on a long distance, the former reduces the trajectory footprint and is therefore more natural looking while reducing the chances of drifts caused by the table movements. For those reasons, moving sideways has a higher DF than turning and moving forward.

A* is one of the most popular search method at finding a solution path using a cost function. In addition to the cost function, a heuristic bias the search towards the most promising states. In our case, our heuristic is a 2D grid containing all the Dijkstra distance costs from the start to the goal states. Even though A* is optimal when it finds a solution, that solution may not always exist or cannot be found within a certain time limit. The Anytime Repairing A* (ARA*) planner focuses on delivering a suboptimal solution as fast as possible, this solution is then optimized iteratively to obtain the optimal solution.

B. Humanoid footprints and whole body control scheme

Once a collision-free trajectory is found by the ARA* algorithm, a set of footprints are defined along the trajectory as it is shown in Fig. 3. Even-though, at first glance, the support polygon appears to be increased by adding the cart, the robot's support polygon is always defined by the contact between the feet and the ground. This is because the robot's arms are not fully bended, therefore the robot could fall forward or backward.

The second step is to define a Zero Moment Point (ZMP) trajectory. A trajectory of the Center of Mass (CoM) of the robot is then obtained using the preview control algorithm proposed in [9]. This algorithm has been widely used by researchers in humanoid robotics, it is simple to implement, yet efficient and yields a smooth CoM trajectory by minimizing the CoM jerk trajectory. The feet trajectories are obtained by spline interpolation between the footprints and

the hands trajectories and orientations are defined in order to minimize the walking swing effect as well as follow the cart orientation.

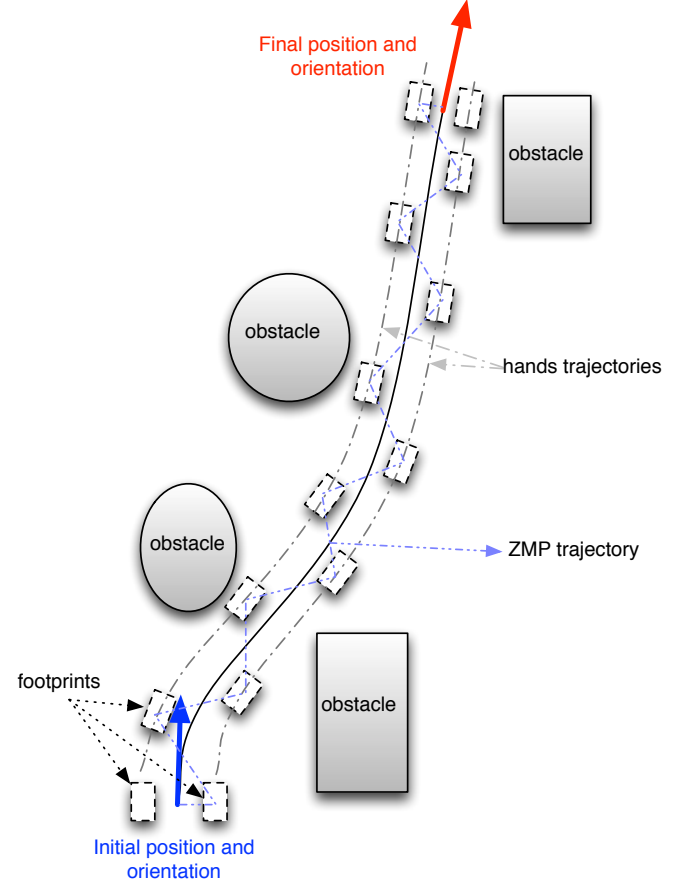


Fig. 3. Overview of the motion planning procedure

To obtain the humanoid robot's joint trajectories, a whole body control scheme with prioritized tasks is formulated as follows:

$$\begin{aligned}
 & \min_{\dot{\mathbf{q}}} \dot{\mathbf{q}}^T \mathbf{Q} \dot{\mathbf{q}} \\
 & \text{subject to} \\
 & \text{First priority} \quad \begin{cases} \mathbf{J}_c \dot{\mathbf{q}} = \dot{\mathbf{r}}_c \\ \mathbf{J}_{lf} \dot{\mathbf{q}} = \dot{\mathbf{r}}_{lf} \\ \mathbf{J}_{rf} \dot{\mathbf{q}} = \dot{\mathbf{r}}_{rf} \end{cases} \quad (4) \\
 & \text{Second priority} \quad \begin{cases} \mathbf{J}_{lh} \dot{\mathbf{q}} = \dot{\mathbf{r}}_{lh} \\ \mathbf{J}_{rh} \dot{\mathbf{q}} = \dot{\mathbf{r}}_{rh} \end{cases} \\
 & \text{Joint velocity limits} \quad \tilde{\mathbf{q}}^- \leq \dot{\mathbf{q}} \leq \tilde{\mathbf{q}}^+
 \end{aligned}$$

where $\dot{\mathbf{q}} \in \mathbb{R}^n$ is the joint velocity vector, \mathbf{Q} is a positive semi-definite matrix, $\mathbf{J}_c \in \mathbb{R}^{3 \times n}$, $\mathbf{J}_{lf} \in \mathbb{R}^{6 \times n}$, $\mathbf{J}_{rf} \in \mathbb{R}^{6 \times n}$, $\mathbf{J}_{lh} \in \mathbb{R}^{6 \times n}$, $\mathbf{J}_{rh} \in \mathbb{R}^{6 \times n}$ are the jacobian matrices of CoM, left foot, right foot, left hand and right hand respectively. $\dot{\mathbf{r}}_c$, $\dot{\mathbf{r}}_{lf}$, $\dot{\mathbf{r}}_{rf}$, $\dot{\mathbf{r}}_{lh}$, $\dot{\mathbf{r}}_{rh}$ are the linear and angular velocity of CoM, left foot, right foot, left hand and right hand respectively.

\tilde{q}^- and \tilde{q}^+ are generalized joint velocity limits defined as follows:

$$\begin{aligned}\tilde{q}_j^+ &= \begin{cases} \zeta \frac{(q_j^+ - q_j) - q_s}{q_i - q_s} & \text{if } q_j^+ - q_j \leq q_i \\ \dot{q}_j^+ & \text{otherwise} \end{cases} \\ \tilde{q}_j^- &= \begin{cases} -\zeta \frac{(q_j - q_j^-) - q_s}{q_i - q_s} & \text{if } q_j - q_j^- \leq q_i \\ \dot{q}_j^- & \text{otherwise} \end{cases}\end{aligned}\quad (5)$$

where \tilde{q}_j^* is the j element of the vector \tilde{q}^* , q_j is the value of joint j , q_j^+ and q_j^- are the upper and lower limits for the joint j , ζ , q_i and q_s are user-defined positive constants, q_i is usually called the interference distance. It can be easily proven that the equalities constraints in (5), not only yield a motion within the humanoid's velocity limits, but also the joints limits are respected as well with a safety margin equals to q_s :

$$q_j^- + q_s \leq q_j \leq q_j^+ - q_s$$

Eq. (5) provides a compact and efficient way for dealing with both of velocity and joint limits, it has been originally proposed in [10].

The optimization problem (4) can be efficiently approximated by the following standard Quadratic Programming (QP) problem:

$$\begin{aligned}\min_{\mathbf{X}} \quad & \mathbf{X}^T \mathbf{H} \mathbf{X} \\ \text{subject to} \quad & \mathbf{J}_1 \mathbf{X} = \dot{\mathbf{r}}_1 \\ & \mathbf{J}_2 \mathbf{X} = \dot{\mathbf{r}}_2 \\ & \mathbf{X}^- \leq \mathbf{X} \leq \mathbf{X}^+\end{aligned}\quad (6)$$

Where:

- $\mathbf{X} = \begin{bmatrix} \dot{\mathbf{q}} \\ \delta_1 \\ \delta_2 \end{bmatrix}$, $\delta_1 \in \mathbb{R}^{15}$ and $\delta_2 \in \mathbb{R}^{12}$ are slack variables.

- $\mathbf{H} = \begin{bmatrix} \mathbf{Q} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q}_2 \end{bmatrix}$, $\mathbf{Q}_1 \in \mathbb{R}^{15 \times 15}$ and $\mathbf{Q}_2 \in \mathbb{R}^{12 \times 12}$ are user-defined positive definite matrices. In order to respect the task priority, the following condition should be satisfied: $\|\mathbf{Q}_1\| \gg \|\mathbf{Q}_2\|$.

- $\mathbf{J}_1 = \begin{bmatrix} \mathbf{J}_c & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 6} & \mathbf{0}_{3 \times 6} & \mathbf{0}_{3 \times 12} \\ \mathbf{J}_{lf} & \mathbf{0}_{6 \times 3} & \mathbf{I}_{6 \times 6} & \mathbf{0}_{6 \times 6} & \mathbf{0}_{6 \times 12} \\ \mathbf{J}_{rf} & \mathbf{0}_{6 \times 3} & \mathbf{0}_{6 \times 6} & \mathbf{I}_{6 \times 6} & \mathbf{0}_{6 \times 12} \end{bmatrix}$, $\mathbf{I}_{n \times n}$ is the identity matrix.

- $\mathbf{J}_2 = \begin{bmatrix} \mathbf{J}_{lh} & \mathbf{0}_{6 \times 15} & \mathbf{I}_{6 \times 6} & \mathbf{0}_{6 \times 6} \\ \mathbf{J}_{rh} & \mathbf{0}_{6 \times 15} & \mathbf{0}_{6 \times 6} & \mathbf{I}_{6 \times 6} \end{bmatrix}$

- $\dot{\mathbf{r}}_1 = \begin{bmatrix} \dot{\mathbf{r}}_c \\ \dot{\mathbf{r}}_{lf} \\ \dot{\mathbf{r}}_{rf} \end{bmatrix}$ and $\dot{\mathbf{r}}_2 = \begin{bmatrix} \dot{\mathbf{r}}_{lh} \\ \dot{\mathbf{r}}_{rh} \end{bmatrix}$

- $\mathbf{X}^+ = \begin{bmatrix} \tilde{\mathbf{q}}^+ \\ \delta_1^+ \\ \delta_2^+ \end{bmatrix}$ and $\mathbf{X}^- = \begin{bmatrix} \tilde{\mathbf{q}}^- \\ \delta_1^- \\ \delta_2^- \end{bmatrix}$

The QP problem (6) can be solved in real-time using an appropriate QP solver such as uQuadProg solver [11] or qpOASES solver [12].

C. Mapping, localization and replanning

To move in a cluttered environment, a robust and precise sensing input is primordial to determine the position of obstacles, detect collisions and to plan valid long term and short term paths. Also, odometry drift must be constantly verified and corrected by a localization mechanism to ensure a close monitoring of the planned path. The human-size humanoid robots, such as HRP-2 or the humanoids robots which participated in DARPA Challenge, are able to build a 3D map on the fly using their very sophisticated proprioceptive and exteroceptive sensors. However, the Nao robot has only two cameras in the head for sensing and localization. A first approach would be using those cameras, however this has proven to be a very difficult task [13] [14]. Indeed, as explained previously, a humanoid robot swings laterally while walking, which leads to pictures of poor quality. Furthermore, the field of view of the Nao is greatly obstructed by the large table and load. As a result, it is hard to precisely determine the position of the environment and obstacles with respect to the robot.

A second approach would be adding a Kinect camera on the top of Nao's head for mapping [15], in our case a second Kinect camera placed on the front of the cart would probably improve the quality of the 3D mapping. This approach will be studied in a future work.

Our main purpose in this paper is to validate the motion planning approach and whole body control scheme, we therefore, as many related research [16], [17], [18], only to cite few, opted for a complete external sensing and localization. Our system is a Vicon motion capture system constituted of 8 MX20 and 4 T40 cameras. It runs at 100 Hz with a precision of 1 mm.

Markers are placed on the obstacles to construct the 2D costmap of the environment. A global costmap keep tracks of the initial obstacles positions for long term planning, while the local costmap is updated every time a marker moves to ensure safe short term planning and real-time obstacle tracking.

Four markers are placed on a trapezoid shape on the top of Nao's head, they are used to keep track of the robot position and orientation. Another four markers are placed on each corners of the cart and they are also used to track the cart position and orientation.

A collision might occur if an obstacle has been moved or a drift from the planned trajectory happened. When a collision is foreseen, a replanning is necessary as shown in Fig. 4. The new collision-free trajectory is found by the algorithm ARA* starting from the point at which the collision has been predicted. If the potential collision is due to drift, the Dijkstra grid does not need to be recalculated, accelerating therefore the replanning. As the walking pattern trajectory for a humanoid robot cannot be changed instantly, a time interval t_c is required to change the planned footprints. In

the implementation of ZMP preview control, a finite time horizon of 2 steps is used to compute the CoM trajectory. Therefore, if a collision is foreseen at instant t_c , the new collision-free trajectory provided by the algorithm ARA* is deformed to keep the next two footprints unchanged as shown in Fig. 4, the robot will however stop if the deformed trajectory is in collision.

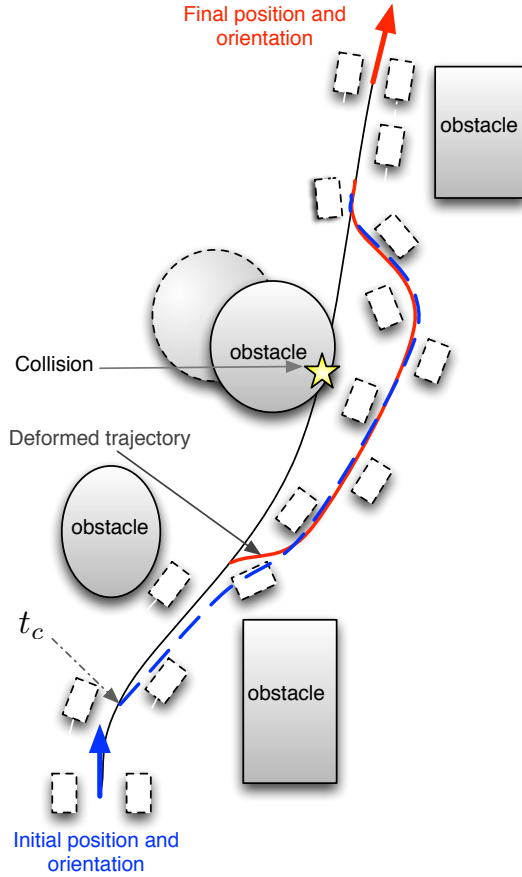


Fig. 4. Replanning in case of collision detection: t_c is the instant at which a collision is foreseen, the new collision-free trajectory is in dashed-blue line, the deformed trajectory is in red line.

IV. RESULTS

Experiments were conducted on a Nao humanoid robot (Fig. 1), manufactured by Aldebaran Robotics [19]. On top of its head, we added 4 motion capture markers in order to track the robot position and orientation.

The cart-like object, shown in Fig. 1, is a mini table 600mm long by 300mm wide. On one side, the two legs are 300mm high and are set on omnidirectional wheels. On the other side, the two legs are half the length so that the Nao can fully support this side of the table. Furthermore, these legs are round and small to fit more tightly in Nao's hands.

The primary objective of our approach is to increase the maximum load weight carried by a Nao humanoid robot, without destabilizing it, while maintaining sufficient flexibility and agility. For that purpose, an experiment aimed

at measuring the maximum carrying capacity of the Nao without any modifications has been carried out. Nao is placed in the same pose as in Fig. 1, then a small board (333g) is attached to both hands and is used to support various amount of calibration weights. Even with low weight, the robot is rapidly out of balance and has difficulty following planned trajectories, this is because stabilization algorithms are constantly prioritized to avoid falling. At 300g additional weight, however, Nao fell within the first steps nearly every run, which we determined to be its maximum carrying limit.

With the introduction of the cart, two sets of motion primitives are available, an omnidirectional and a heavy load sets. An example of a right turn for both set is shown in Fig. 2. The carry load at which the friction becomes too high to consider the heavy load set is 700g. In reality, the robot can push higher load, however the wood structure of the cart-like object cannot support a load higher than 7,000g. To summarize, the maximum carrying capacity of the Nao robot alone is 633g and by using the cart, it is 7,000g, which is 11 times its normal capacity.

In the case of omnidirectional set, the hands and arms are strong enough to articulate the cart while turning to obtain smooth trajectories. The maximum angle at which our robot can turn the cart is 30 degrees, as illustrated in Fig. 5. Over that limit, one hand is colliding with the torso while the other lies outside of the robot workspace.

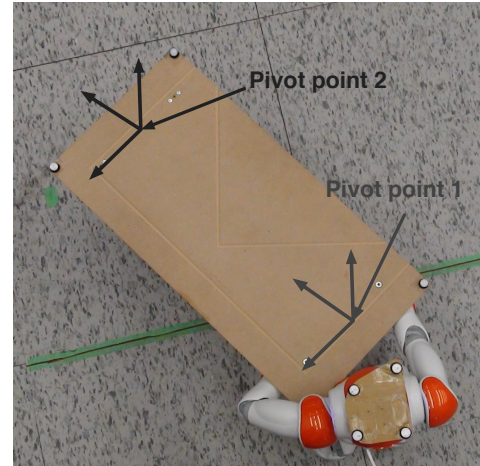


Fig. 5. The arms posture while turning the cart at maximum angle (30 degrees)

While walking in a straight line of 1 m, our Nao is affected by a drifting of an average of 10.5 cm. Without any corrections, this error would lead the robot to constantly diverge from the planned trajectories. The correct localization information provided by the external sensing allow us to modify the trajectory to cancel the drift.

While pushing heavy load, however, the robot cannot rotate in place or move laterally to cancel any drift errors, a quick replanning is therefore executed when the robot diverges too much from the planned trajectory.

As showed in Fig. 6, without any hand position correction, the lateral swing causes large oscillations that are transmitted

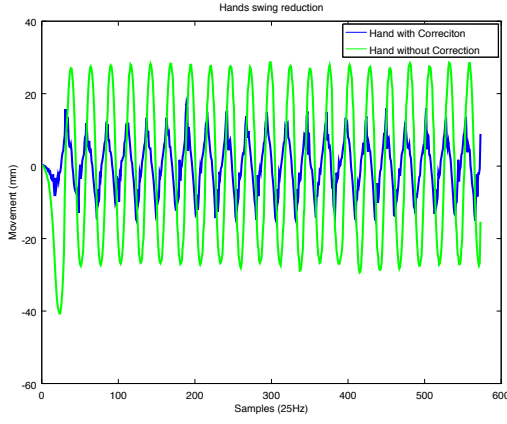


Fig. 6. The hands lateral movement while walking with and without corrections

to the table and the load. The average peak-to-peak position movement is 56.25 mm. However, with the proposed corrections, the hand distance from desired position has been reduced to 28.20 mm, reducing the hand error by 49.87%. As a result, significantly less oscillations are transmitted to the table, leading to a safer and enhanced carrying ability and load stability. However, note that the error cannot be completely cancelled because: I) the hand trajectories are second priority task, that means the robot will respect those trajectories as far as the trajectories of first priority are fully followed, II) the Nao robot has only 5 degrees of freedom in each arm, III) the joint backlash of the Nao robot.

To test the system as a whole and to validate the proposed algorithms, we conducted three series of 5 experiments. The omnidirectional set of primitive has 12 different primitives with θ_r sampling of 11.25° while the heavy set have only 5 possible primitives, but with a precision of 5.625° . In each experiment, the robot starting and goal positions were chosen in a way that the robot had to navigate through a field of motion capture markers on the ground. Every marker serves as an obstacle that must be avoided by the robot and the cart. They were placed to form various feasible paths and force tight turns in order to take advantage of the additional degree of freedom (the rotation of the cart θ_{ca}). The three series were composed of the same experiments containing the same initial configuration of the obstacles in the environment and using the same initial and goal positions and orientations, but with different transported objects.

Fig. 7 shows the start and end positions for each type of experiment. In this figure, the obstacles are in black, while the grey areas around them are an inflation zone where the cost is higher than free space to prevent the robot from passing too close to obstacles. The vertical and horizontal black lines are virtual walls to prevent the planner from detecting optimal trajectory passing around the experimental setup.

The first series consists of the Nao robot alone, without a cart or load. It uses the omnidirectional primitives set to navigate through the obstacles. The second one was

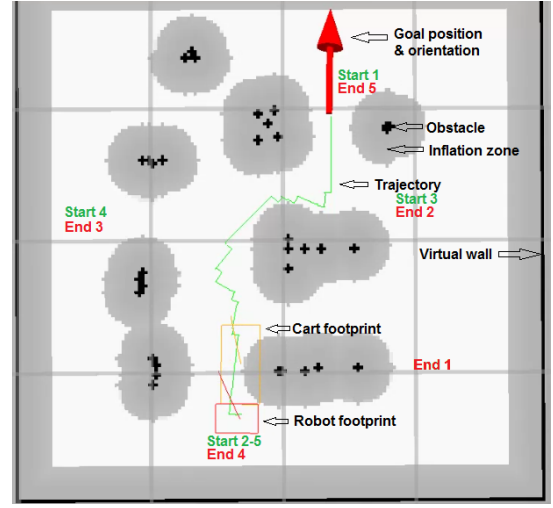


Fig. 7. Map of the obstacles (black), the inflation zone around them (grey) and experiments with the Nao (red square), the cart (orange rectangle) and goal position/orientation (red arrow). Note that the initial posture collides with the inflation zone, but it is not in collision with the obstacles.

conducted with the Nao holding the cart, which increased the navigation footprints significantly. These tests also used the omnidirectional set to construct the plan. For the third and final series, the robot is pushing the cart with an additional load of 2,300g, in this case the heavy load set of primitives have been used to allow the Nao to plan a trajectory in the cluttered environment.

The ARA* planner initial $\epsilon = 3$ means that the suboptimal solution cannot be worse than 3 times the cost of the optimal solution. A time limit of 10 seconds was chosen and within that time ϵ successfully decreased to 1 on every run, which corresponds to the optimal solution. For each generated path, we measured the total time to execute the trajectory, the trajectory length, the initial solution time, the optimal solution time, the initial expanded nodes and the final expanded nodes. The PC that has been used to generate these results has: a i7-3770 Processor with 8 cores at 3.4GHz and 8GB of RAM. These results are summarized in Table II.

	No Cart	Cart Omni	Cart Heavy Load
Total time (s)	69.75	104.90	134.71
Total time Std (s)	14.69	15.76	27.62
Trajectory length (m)	2.41	3.03	3.20
Trajectory length Std (m)	0.53	0.26	0.48
Average velocity (m/s)	0.035	0.029	0.024
Initial solution time ($\epsilon = 3$) (s)	0.020	0.026	0.012
Initial solution time Std (s)	0.020	0.024	0.008
Optimal solution time ($\epsilon = 1$) (s)	0.20	0.41	0.44
Optimal solution time Std (s)	0.11	0.46	0.38
Initial node expansions	541.6	769.8	892.4
Initial node expansions Std	492.5	497.0	508.7
Total node expansions	4775	10694	25165
Total node expansions Std	2760	11436	24105

TABLE I
EXPERIMENTAL STATISTICS

It can be observed that the average velocity is lower while using the cart. This is explained by the friction of the wheels of the cart causing slippage as the robot tries to move,

slowing its movements down. Every step in a direction results in a slippage in the opposite direction, thus progressing less distance with each step. This leads to a reduced speed of 17.14% for when pushing the empty cart and 31.43% speed reduction with the additional 2.3 kg load.

Even-though the primitives with the robot alone and with the cart are the same, it is hard to find a path as optimal with it. Since the length is only about 25% longer though, moving with the table does not impair too much the robot ability to travel the cluttered environment swiftly.

The weight primitives trajectory length however is higher in comparison, about 33% higher than the Nao alone. This is primarily due to the change of primitives. The movement is not as smooth, in particular, moving sideways or in diagonal become impossible. Also, turning in place versus turning around a pivot placed $r = 0.60m$ away increased the trajectory by at least $L = \frac{\theta\pi r}{180}$ every time the robot makes a turn.

The biggest problem we encountered is the very limited space where the Vicon is precise. Indeed, the precision is under 1 mm while correctly positioned, but during our experimentations, often when Nao got too close to the edges of the “sweet spot”, it starts to lose the markers and fails to follow correctly and consistently the localization. As a result the working space was limited, however our algorithm is more generic and already ready for large scale autonomous navigation. We plan changing this system for a more autonomous one as it will be explained in Section V.

Also, the friction of the wheels to the ground is not high enough to completely prevent them from sliding and thus to act as a perfect pivot while using heavy load. This causes additional errors when turning and walking that can accumulate and force a replanning. For this reason, with the heavy load, 1.8 replanning were needed on average, while only 0.4 for the cart alone and 0 without the cart when no obstacles are moved.

When an obstacle is moved in the initial trajectory of the robot or cart, a replanning is essential to avoid collision. On all our experiments, there was no collision between neither the Nao or the cart with any obstacles or the virtual walls¹.

V. CONCLUSION AND FUTURE WORK

In future work, our priority will be to remove any dependency on the external localization system, that is the Vicon system, and instead use other techniques to make the system completely autonomous. As suggested in Section III-B, adding Kinects to the Nao and the table will be explored. Also, since we are using multiple sets of primitives depending on the load weight, an automatic estimation of the load weight, to decide which set to be used, would improve the system and make it more general and autonomous.

ACKNOWLEDGMENT

We thank the anonymous reviewers for their helpful comments. This research is supported by a discovery grant (Prof.

Wael Suleiman) from the Natural Sciences and Engineering Research Council of Canada (NSERC).

REFERENCES

- [1] C Ronald Kube and Eric Bonabeau. Cooperative transport by ants and robots. *Robotics and autonomous systems*, 30(1):85–101, 2000.
- [2] ZhiDong Wang, Majid Nili Admadabadi, Eiji Nakano, and Takayuki Takahashi. A multiple robot system for cooperative object transportation with various requirements on task performing. In *IEEE International Conference on Robotics and Automation, 1999*, volume 2, pages 1226–1233.
- [3] Atsushi Yamashita, Tamio Arai, Jun Ota, and Hajime Asama. Motion planning of multiple mobile robots for cooperative manipulation and transportation. *IEEE Transactions on Robotics and Automation*, 2003, 19(2):223–237.
- [4] Satoshi Shigemi, Yuichiro Kawaguchi, Takahide Yoshiike, Koji Kawabe, and Naohide Ogawa. Development of new ASIMO. *Honda R and D Technical Review*, 2006, 18(1), 2006.
- [5] Shunichi Nozawa, Toshiaki Maki, Mitsuharu Kojima, Shigeru Kan-zaki, Kei Okada, and Masayuki Inaba. Wheelchair support by a humanoid through integrating environment recognition, whole-body control and human-interface behind the user. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008*, pages 1558–1563.
- [6] Michael Stilman and James Kuffner. Navigation among movable obstacles: Real-time reasoning in complex environments. In *Proceedings of the 2004 IEEE International Conference on Humanoid Robotics (Humanoids’04)*, volume 1, pages 322 – 341, December 2004.
- [7] Jonathan Scholz, Sachin Chitta, Bhaskara Marthi, and Maxim Likhachev. Cart pushing with a mobile manipulation system: Towards navigation with moveable objects. In *IEEE International Conference on Robotics and Automation, 2011*, pages 6115–6120.
- [8] Maxim Likhachev, Geoffrey J Gordon, and Sebastian Thrun. ARA*: Anytime A* with provable bounds on sub-optimality. In *Advances in Neural Information Processing Systems*, page None, 2003.
- [9] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Biped Walking Pattern Generation by using Preview Control of Zero-Moment Point. In *Proc. IEEE International Conference on Robotics and Automation*, pages 1620–1626, Taipei, Taiwan, 2003.
- [10] F. Kanehiro, F. Lamiraux, O. Kanoun, E. Yoshida, and J.-P. Laumond. A Local Collision Avoidance Method for Non-strictly Convex Objects. In *2008 Robotics: Science and Systems Conference*, Zurich, Switzerland, June 2008.
- [11] Fumio Kanehiro, Mitsuharu Morisawa, Wael Suleiman, Kenji Kaneko, and Eiichi Yoshida. Integrating geometric constraints into reactive leg motion generation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4069–4076, 2010.
- [12] H.J. Ferreau, C. Kirches, A. Potschka, H.G. Bock, and M. Diehl. qpOASES: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, 2014. (in print).
- [13] Olivier Stasse, Andrew J Davison, Ramzi Sellaouti, and Kazuhito Yokoi. Real-time 3d slam for humanoid robot considering pattern generator information. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006*, pages 348–355.
- [14] Stefan Oßwald, Armin Hornung, and Maren Bennewitz. Learning reliable and efficient navigation with a humanoid. In *IEEE International Conference on Robotics and Automation, 2010*, pages 2375–2380.
- [15] D. Maier, A Hornung, and M. Bennewitz. Real-time navigation in 3d environments based on depth camera data. In *12th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 692–697, Nov 2012.
- [16] S. Hak, N. Mansard, O. Stasse, and J.P. Laumond. Reverse control for humanoid robot task recognition. *IEEE Transactions on Systems, Man, and Cybernetics—Part B : Cybernetics*, 42, 2012.
- [17] M. Levihn, K. Nishiwaki, S. Kagami, and M. Stilman. Autonomous environment manipulation to assist humanoid locomotion. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4633–4638, May 2014.
- [18] Seungsu Kim, A. Shukla, and A. Billard. Catching objects in flight. *IEEE Transactions on Robotics*, 30(5):1049–1065, Oct 2014.
- [19] David Gouaillier and et al. Mechatronic design of NAO humanoid. In *IEEE International Conference on Robotics and Automation, 2009*, pages 769–774.

¹A video of the experiments is available at <https://goo.gl/RUYAY7>