Step Toward Deploying the Torque-Controlled Robot TALOS on Industrial Operations

Côme Perrot, Olivier Stasse

Step Toward Deploying the Torque-Controlled Robot TALOS on Industrial Operations

Côme Perrot[1], Olivier Stasse[1,2]

***Abstract*— This paper tackles the use of torque controlled humanoid robot TALOS in the context of industrial manufacturing. It demonstrates that it is possible to use Whole Body Model Predictive Control (WBMPC) to reliably insert a tool in the holes of an aircraft structure with an accuracy of few millimeters. This result is based on the use of Crocoddyl, an optimal control library that exploits differential dynamic programming (DDP) to achieve high numerical efficiency. The focus of this article is put on the procedure that was undertaken to shape the cost function of the optimal controller. Our approach has first been validate in a low performance setting on the humanoid robot TALOS. Then, a strategy to improve the performances by reinjecting information about the posture of the robot from previous experiments is showed in simulation.**

I. Introduction

A. Context presentation

Robots are nowadays a standard tool in large-scale manufacturing [1]. They excel at performing repetitive tasks in very well-known environments. However, they are yet to reach a huge part of the wide variety of industrial work that exists in our society.

One of the major drawbacks most industrial robots suffer from is lack of mobility. Their design does not allow them to be a relevant solution for many low volume, high added value productions such as the one found in aeronautic manufacturing. According to [2], humanoids are a promising direction to overcome this weakness. However, the resort to humanoid robots induces a higher control complexity which is further heightened when dealing with variability in the environment.

In recent years, Reinforcement Learning (RL) has been successfully used to generate highly dynamical motions on quadruped robots, such as ANYMAL [3], as well as bipedal torque controlled walking robots such as CASSIE [4]. Still, in [5], a comparison with Model Predictive Control (MPC) shows that the latter has a higher rate of success in constrained environments. Despite both approaches being different, the definition of the cost function remains a central point for both RL and MPC. The increasing complexity of the system makes it difficult to properly design such a cost-function. The aim of this paper is, first, to experimentally find an initial feasible solution for a real situation. Then, design a simple strategy to modify the cost function in order to improve performances.

Côme Perrot and Olivier Stasse were supported by the cooperation agreement ROB4FAM and the regional Occitanie project MERLHBOT.

[1] Gepetto Team, LAAS-CNRS, Université de Toulouse, France.
[2] Artifical and Natural Intelligence Toulouse Institute, France

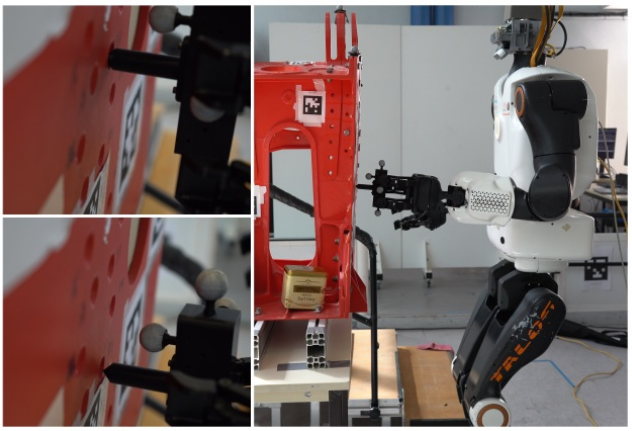


Fig. 1: Deburring task, high precision for a fine insertion into a hole using whole-body MPC on a torque controlled robot.

A widely used motion generation framework for humanoid robot is built upon a Model Predictive Controller for the centroidal dynamics in conjunction with an instantaneous QP-controller for the whole body [6]. A planner provides the reference trajectories to follow for tasks such as gaze direction, end-effectors placement and the overall direction of the robot.

If position control has been quite successful in generating a wide variety of robot behaviors, its capacity to react to external forces is limited to the end-effector, where a force sensor is typically incorporated for that purpose. In [2], [7] torque controlled robots appear as a potential solution for managing interactions with the environment as well as ensuring safe and compliant behavior in unplanned situations.

These situations can arise because of unforeseen events such as changes in the context, or human interaction. It opens up the way for more flexible use of robots than what was achieved with existing position control methods. Recent robots such as Digit [8] are using torque control and demonstrate impressive locomotion performances and robustness. It comes however at the cost of a more complex control architecture on robots with wave generators, and a lack of precision for positioning tasks.

Precision is nonetheless of great importance when executing industrial tasks such as deburring. A simple way to handle this issue is, assuming you can measure it, to apply a strong feedback on the error between the desired and perceived position of the tool. But, on a torque controlled robot, this might lead to a diverging command [9]. [10]

has developed a passivity framework which is taking into account the energy of the system to maintain its stability. It was successfully tested on the TORO humanoid robot [2]. This approach assumes that either the desired position or trajectory of the end-effector is given. The passivity approach avoids injecting unsafe amounts of energy in the system if the environment differs too much from what was planned. The whole body instantaneous controller is in charge of absorbing model discrepancies and planner assumptions.

The success and the efficiency of the classical approach lie in the capabilities of the motion planner to generate a desired trajectory that is compatible with the whole body instantaneous controller. It can be done for instance by using a hybrid control approach and planning over a graph of motion primitives for quadrupeds [11]. It can also be done using A* through a discrete set of actions predefined according to the targeted tasks, see [12] for a locomotion example. In order to cope with the complexity of the problems most planners are using heuristics [12], or reason on low dimensional necessary conditions [13].

Still, no matter how advanced the planning heuristic is, it cannot entirely address the fundamental limitation of instantaneous whole body control. This limitation resides in the inability of this technique to account for whole-body related constraints within the MPC horizon. It means that potential conflicts with the constraints can only be detected by the whole-body controller when it is too late. For this reason, [14] proposed a whole-body model predictive control with state feedback at 100 Hz. It can perform trajectory optimization and provides reference torques to a low-level torque loop running at 2 KHz. An extension of this technique was introduced in [15], where the feedback gains of the DDP are directly sent to the low-level controller, improving meaningfully the quality of the generated motion. This approach has several advantages. A significant one is to include all the dynamical effects of the limbs on the balance criteria. This is particularly important with a robot having arms and legs that way more than 10 kg and 15 kg respectively, such as TALOS.

B. Contributions

WBMPC has successfully been applied to the humanoid robot TALOS to carry out an industrial deburring task similar to the one presented in [16]. As seen in Fig. 1, the objective of the task is to insert a 3d printed tool inside the holes of a mockup aircraft part. It simulates deburring, an operation that needs to be undertaken after drilling holes to clean up any material residues.

Experiments were conducted in a lab as well as on an Airbus site. The results that were obtained validate the relevance of using WBMPC for humanoid robots in an industrial context.

This article also tackles the issue of cost shaping. It is a challenging aspect for optimal control based approaches that needs to be resolved in order to unlock the full performances of the system. [17] uses a multi-objective optimization in conjunction with Bayesian optimization to find a suitable set of parameters. Because we expect optimal approaches to be too computationally intensive in our case, we choose to use a fixed cost function structure to experimentally explore the environment.

The contributions presented in this paper are twofold:

- Demonstration of the experimental use of a humanoid robot in an industrial setting;
- Use of advanced cost shaping solutions to enable better performances in this context.

The adopted control architecture is first presented in section II before going into more details about the structure of the cost function that was considered in section III. To finish, the most significant results are exposed in section IV.

II. Whole-body Model Predictive Control

A. Robot Modelling

The robot configuration $q \in \mathbb{R}^3 \times SO(3) \times \mathbb{R}^n$ defines the global position, orientation and posture that a mobile robot has at one given moment. Such configuration evolves under the action of internal and external forces as described by the rigid-body dynamics [18]:

$$\begin{bmatrix} M & J^T \\ J & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ -\lambda \end{bmatrix} = \begin{bmatrix} S\,\tau - b \\ -\dot{J}\,\dot{q} \end{bmatrix}, \tag{1}$$

where M is the inertia matrix, b stands for Coriolis, centrifugal and gravity forces, joint-motor torques $\tau \in \mathbb{R}^n$ affect only the actuated joint as indicated by the selection matrix $S \in \mathbb{R}^{n+6\times n}$, and all contact wrenches $\lambda_i \in \mathbb{R}^6$ are contained in $\lambda = \begin{bmatrix} \lambda_1 & \cdots & \lambda_i & \cdots \end{bmatrix}$ with the application points described respectively by the Jacobians $J_i \in \mathbb{R}^{6\times n+6}$ contained in $J = \begin{bmatrix} J_1 & \cdots & J_i & \cdots \end{bmatrix}$. The second line of Eq.(1) constraints the robot parts under contact to stay motion less during the contact.

Based on this dynamics, the robot configuration q and its time derivative $\dot{q}$, which are the state $x = (q, \dot{q})$, are controlled by inputting desired torques τ on joint motors during some discretization period dt to obtain the next state

$$x^+ = f(x, \tau), \tag{2}$$

which is predicted from numerical integration of Eq.(1).

B. Optimal Control

For a given initial state x_0, an optimal control sequence $U^* \triangleq \{\tau_0, \tau_1, \cdots, \tau_{N-1}\}$ is generated according to Eq.(2), along a horizon of N time-steps in the future by minimizing the cost function

$$V(x_0) = \sum_{i=0}^{N-1} l(x_i, \tau_i) + l_N(x_N), \tag{3}$$

that is designed to encode the desired robot behavior with a running cost $l(\cdot,\cdot)$ for each time-step, and a terminal cost $l_N(\cdot)$ guiding the robot to end into some safe set of states. This desired behavior is discussed more precisely in Section III.

The resulting optimal pair of control sequence and robot motion (U^*, X^*) is said *feasible* if it satisfies the dynamics

described in Eq.(1) [19]. Here, feasibility of the optimal controller is ensured by implicitly imposing the discrete form Eq.(2) in Eq.(3).

Following an MPC scheme, *i.e.:* at time j, the control sequence U_j^* is generated considering the initial state x_0^j, then only the first control τ_0^j of the sequence is executed during the discretization time dt arriving to a new state x_1^j, which is used as initial state $x_0^{j+1} = x_1^j$ to generate an entire new sequence U_{j+1}^* and this is repeated cyclically [20]. This procedure guarantees that the generated robot motion is part of a feasible path of at least N steps in the future. Feasibility beyond the horizon can also be ensured by making the robot reach some state where the robot can stay safely during indefinite time at the end of the horizon [21]. This property is enforced with the terminal cost $l_N(\cdot)$.

In particular, the DDP algorithm is used to minimize the cost function Eq.(3) at each iteration of the MPC. The computational efficiency of DDP allows controlling 31 degrees of freedom of the robot TALOS along a horizon of $N = 100$ time-steps with $dt = 10$ ms online (computed during the movement of the robot). DDP has the drawback of not accepting explicit constraints, though recent results suggest a forthcoming solution to this issue [22]. Here, however, the traditional solution is to consider Eq.(2) as an implicit constraint.

DDP produces Riccati gains

$$K_0 \triangleq \left.\frac{\partial \tau}{\partial x}\right|_{x_0} \tag{4}$$

evaluated at the initial state, as a partial result of the optimization. Control values are interpolated using these gains, as proposed in [15], to reach an updating period of 0.5 ms on the resulting control law:

$$\tau = \tau_0 + K_0(x - x_0), \tag{5}$$

with a feedback term based on the measured state x which is updated at every millisecond and a feedforward term τ_0 computed optimally from the measured initial state x_0 at each MPC iteration (every 10 ms). In order to further boost the DDP performance, the pair (U^*, X^*), obtained in the previous MPC iteration, is the warm start at each computation of the control sequence. For the first control sequence, since there is no previous solution to reuse, DDP is iterated starting from a constant trajectory until convergence.

III. Deburring Controller

Contrary to most solutions found in the litterature the WBMPC implemented on the robot does not rely on a reference trajectory. Instead, all the information about the task is encoded through the cost function and the robot's trajectory is implicitly generated. This reduces the overall complexity of the control structure because it does not require a higher level planner to be used. It however makes the design of the cost function for a single task much more challenging.

Shaping the cost function is made even more complex by the need to reconcile occasionally conflicting objectives in a single scalar function. Furthermore, the solver does not accept explicit constraints. So the cost function must incorporate relaxed safety constraints and address multiple objectives simultaneously. To simplify the process, a fixed structure is chosen where the cost is composed of sub-costs that incentivize or discourage specific robot behaviors.

A. Cost function structure

We reuse a general architecture that has already shown interesting results in [14].

The cost function is split into four different sub-costs: constraints, equilibrium, regularization, and goal

$$l(x,u) = w_{cons}l_{cons} + w_{eq}l_{eq} + w_{reg}l_{reg} + w_{goal}l_{goal}. \tag{6}$$

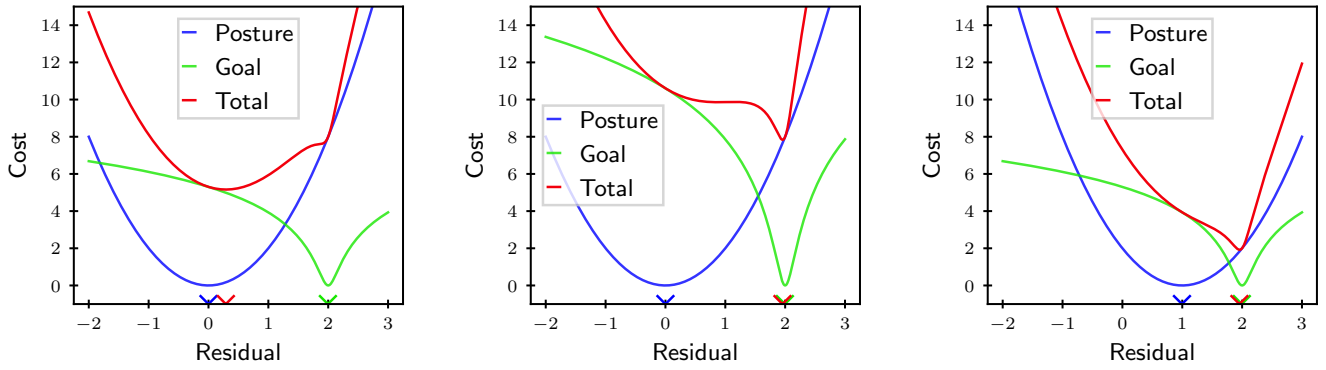
Each of the sub-costs has an associated weight, which can be adjusted to define the relative priority of each task.

1) Constraints cost: The first, and most highly weighted cost, aims at preserving the integrity of the robot. It is a barrier cost that greatly penalizes any configuration that does not respect the kinematic constraints of the robot: $l_{cons}(x) = \|max(x - x_u, 0) + min(x - x_l, 0)\|^2$. With x_u and x_l respectively being the upper and lower bounds of the admissible states.

2) Equilibrium cost: Balance is also a major concern when working with humanoid robots. The robot must stay on its feet, throughout the whole operation. It is achieved with an equilibrium cost: $l_{cons}(x) = \|c(x) - c_d\|^2$ with $c(x)$ and c_d the current and desired center of mass of the robot. For the deburring task, maintaining the center of mass of the full robot over its supporting feet is enough to penalize movements leading to losses of equilibrium. As this cost also preserves the robot integrity, it is set with the second highest relative weight.

3) Regulation cost: To guarantee the numerical stability of DDP, a regularization cost that ensures uniqueness of the optimal control is added: $l_{reg}(x,u) = (x - x_d)^T\mathbf{R}_x(x - x_d) + (u - u_d)^T\mathbf{R}_u(u - u_d)$. It prioritizes behaviors that are close to the desired state x_d built from the initial robot posture, with zero velocities. It also penalizes controls that are far from the torques u_d required to counteract the force of gravity in the desired position. $\mathbf{R}_x$ and $\mathbf{R}_u$ are positive definite matrices used to tune the relative impact of each joints on the regulation cost.

4) Goal related cost: While constraints, equilibrium and regulation are general enough to be widely used in humanoid robot applications, task-specific components are also required for the cost to be applied in a concrete experiment. For the deburring operation, the goal cost encourages the robot to position correctly its left end-effector and maintain zero velocity. It is designed as follows: $l_{goal} = log(1 + \frac{\|p - p_d\|}{\alpha}) + \|R - R_d\|^2 + \|v\|^2$ with p and p_d the actual and desired Cartesian position of the end-effector, R and R_d the actual and desired rotation (defined as elements of $SO(3)$), v its Cartesian velocity and $\alpha = 0.02$.



(a) Baseline (b) Variable Weight (c) Variable Posture

Fig. 2: Simplified illustration of the cost conflict. Values are just for scale and do not represent the actual value of the cost function for our application. Colored tick on the x-axis indicate the abscissa of the minimum of each functions. The distance between the red and green ticks represents the error associated to the cost function.

B. *Cost function shaping*

From the structure presented in Section III-A naturally arises a set of parameters that needs to be tuned in order to achieve a specific task. A common approach is to proceed via trial and error either in a simulator or directly on the real robot. For simplicity purposes, a single tuning is often chosen for the whole movement. However, even if it is theoretically possible to shape a cost function that exploits the full abilities of the robot in every situation, it is in practice very challenging.

Despite trying to set a clear hierarchy between tasks by choosing weights with different orders of magnitude, the problem of conflict between tasks still arises during the experiments. Indeed, the posture task reference is always the same for all the cases while the desired goal can vary in all of the robot workspace. It means that, most of the time, those two costs tend to attract the robot toward different equilibrium. This results in the optimal solution being a trade-off between both costs which leads to poor performances.

That is why we resort to have a cost function that varies in time and along the horizon of the MPC:

$$V(x_0, t) = \sum_{i=0}^{N-1} l_i^t(x_i, \tau_i) + l_N^t(x_N), \tag{7}$$

In order to guarantee the coherence of the problem between each iteration we update cyclically each node of the cost function so that only the last one contains new information:

$$\forall i \in [\![0 : N-2]\!], l_i^{t+1} = l_{i+1}^t, \tag{8}$$

All of the experiments used the same receding horizon approach where the first node of the horizon is discarded and a new custom one is added at the end. This approach permits to have a richer representation of the task while keeping a simple structure for the cost function even if it requires either to hard-code the time sequence or to resort to an external planner.

We tried several approaches to generate this new node in the trajectory as shown in Fig. 2.

1) Baseline: The baseline performances is computed using a mostly fixed cost function. The only parameter that changes over time is the desired Cartesian position of the end-effector. It allows the robot to reach several targets during one experiments.

Even if a more advanced tuning could lead to better results, any further improvement is made very challenging because of the sensitivity of the performances to the cost function.

2) Variable goal-cost weight: A straightforward way to solve the cost conflict is to increase the relative weight of the goal cost with respect to the posture cost. A linear scheduling of the weight is chosen so that the cost only increases at the end of the movement when high precision needs to be achieved:

$$w_{goal}(t) = at + a_0 \tag{9}$$

This strategy was successfully used to conduct the first set of validating experiments on the real robot.

3) Variable posture reference: Another solution to solve the conflict is to update the reference posture at the same time as the goal:

$$l_{reg} = \|x(t) - x_{reg}(t)\| \tag{10}$$

where $x(t)$ is the measured state and $x_{reg}(t)$ a variable reference state.

This allows to improve performances without tempering with the relative weight of each costs hence preserving the safety of the robot.

To do so we do not use an external logic but reuse solutions of previous experiments found using our control structure. In practice, we explore the environment of the task using a simple control structure and re-inject the reached posture as a reference for subsequent realisations. This approach is relevant when no expert data is available to guide the resolution.

IV. Application of the control structure

To validate the method presented in this paper, we study a task which consists in reaching a series of points in sequence while achieving a good accuracy (less than 5mm of error in our case). The accuracy threshold is chosen to match the radius of the hole in which the tool needs to be inserted.

We will explain the software architecture used during both the experiment and the simulation in Section IV-A before detailing the two phases of test that we carried out:

- First, an exploratory phase conducted on the robot. It aimed at validating that the presented method could reach a precision of 5mm.
- Then, a performance improvement phase focused on exploiting the full capabilities of the physical system.

A. Control setup

The control architecture is split into two levels. The computationally expensive optimal control resolution is done at 100Hz. In the case of experiments on the real robot this part is carried out by an external computer (fitted with an AMD Ryzen 5950X, 16 cores with 64 GB of RAM). A faster control, based on the gains computed by the MPC can then be run directly on the robot at 2kHz as shown in Eq.(5). The MPC implementation was based on Crocoddyl (Contact RObot COntrol by Differential DYnamic Library). The software architecture is summarized in Fig 3.

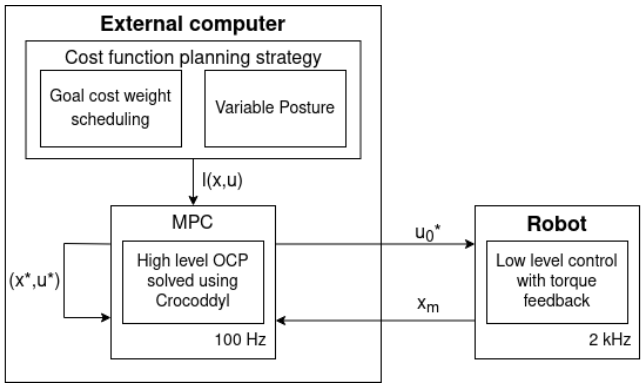


Fig. 3: Diagram of structure used to control the robot. $l(x, u)$ is the cost function optimized by the OCP, (x^*, u^*) are the current optimal state and control trajectory produced by the MPC, u_0^* is the control sent to the robot and x_m the state measured by the proprioceptive sensors of the robot.

B. Concept validation in the real world

As mentioned in Section I-B experiments were conducted both in our lab and directly on site at an Airbus plant. The obtained movement can be seen in the companion video.

Speed was not the focus of this stage, that is why we resorted to the gain scheduling technique to carry out the task. Indeed it was a straightforward way to achieve the desired result in a setting where stability was not a major concern because of the low movement speed involved.

The robot successfully managed to reliably insert the tool that was fitted on its end-effector in a sequence of 4 holes. In a separate experiment we checked that the robot remained compliant while it inserted the tool by having a human push its arm.

A motion capture system was used to calibrate the position of the aircraft piece with respect to the robot at the beginning of the experiment. Other than that, no visual feedback was required during the experiment and the proprioceptive based movement was precise enough to carry out the task.

C. Performance improvements

After validating the relevance of the chosen approach, work was done to improve the performances and the achieved movement speed using the PyBullet simulator. This simulator has been used in the past as a validation step before deploying new movements on the robot.

1) Benchmark: First, a benchmark of the three approaches presented in Section III-B is showed. The performances of the controllers are evaluated according to two metrics :

- The distance between the center of the hole and the tip of the end effector. The task is considered to be successful if this distance is below 5mm;
- The time to successfully carry out the task. Which is the time between the beginning of the movement and the moment where the tip of the tool is less than 5mm away from the hole and stays in this zone.

We setup the robot to reach a precise point in space starting from its default position using all three approaches. The results are compiled in the following table:

Method	Baseline	Variable weights	Variable posture
Accuracy (mm)	8.27	0.85	0.24
Time (s)	-	1.06	1.24

As can be seen in Fig. 4, the baseline is not precise enough to reach the desired threshold. This illustrates the limitation that we mentioned in Section III-B. The other two solutions can solve this issue if tuned properly.

However it is worth noting that this results comes from a simulation and cannot be directly translated to the real world because of unforecasted disturbances and discrepancies between the model and the real robot.

In particular, the variable weight approach suffers from a major weakness. Changing the relative weight of costs may reduce the significance of the safety related cost. This can lead to more dangerous movements if done recklessly. In addition to that, higher gains can hinder the stability of the control. We can see oscillations in the movement which indicates a less stable control.

On the other hand, updating the reference posture can solve the cost conflict without altering the relative weight of the tasks. Since the weights of the placement and posture task are relatively low with respect to the limits and stability cost in this setting, this approach is less dangerous for the robot.

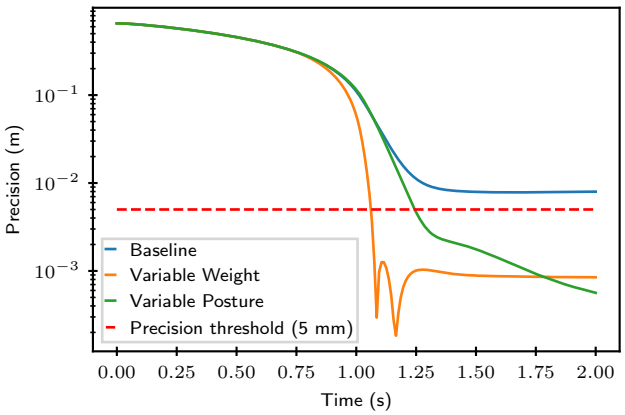


Fig. 4: Evolution of the cartesian position of the end effector with respect to time.

2) Performances of the variable posture approach: Because it is less dangerous for the robot while still being efficient, the variable posture approach is tested on a sequence of two holes. Fig 5 indicates the robot can precisely reach both holes with a transition time of 0.5 seconds. While we don't have precise data regarding the performances of a human operator for this specific experiment, it has been reported to us by Airbus employees that a worker would take around one second to transition between two holes. It means that the attained performances are in the same order of magnitude of what a human could achieve, which was not the case with the baseline solution.

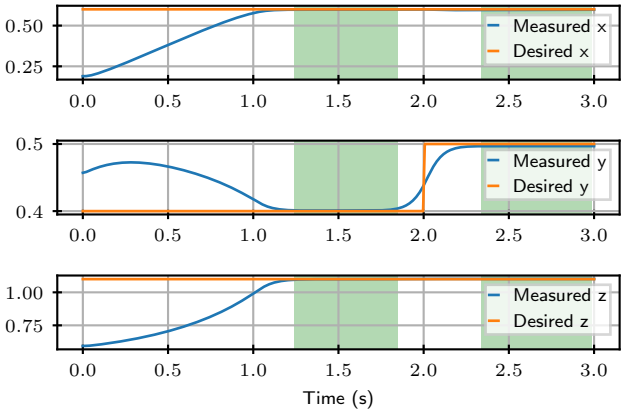


Fig. 5: Simulated evolution of the cartesian position (in meters) of the end effector with respect to time. The distances are given with respect to the center of mass of the robot. The x axis is oriented toward the front of the robot, the y axis to the left and the z axis is going up. Regions highlighted in green are when the end-effector is less than 5mm away from the target position.

V. Discussion

A. Difficulties to deploy an efficient motion

Even if the control scheme was successfully deployed on the robot, increasing the performances still represents a major challenge. Indeed, in [9] a similar performance improvement as the one described in paragraph IV-C was applied to the TALOS robot without any particular precaution. It caused the controller to inject a high quantity of energy in the system which, despite the safeties that are implemented, damaged the robot. This is not desirable and may imply to expand the solver and include a passivity constraint as proposed in [10], or a similar approach to prevent this type of behavior. Such an extension is beyond the scope of this paper.

To reduce the occurrence of such accident, the manufacturer of the robot, PAL-Robotics, provides a high fidelity simulator which includes a model of the actuators. It also warns the user of possible collision using an energy based criteria for each actuator separately. This is unfortunately not sufficient to guarantee the safety of the robot.

This means that the only tractable way to proceed is to gradually increase performances on the real robot. However, in the case of a complex system like a humanoid robot, this requires extensive manpower (at least three people are needed to operate the robot safely). It also subjects the hardware to high wear and tear.

B. Need for planning to achieve human-like performances

The proposition made in this paper to improve performances revolves around injecting relevant information inside of the system through the cost function. It differs from traditional motion planning approaches because it does not rely on an external heuristic to provide the necessary information. Instead it leverages data from previous experiments to achieve the desired performances.

This drives the intuition that work should be done to build a form of memory for the system. This memory would be queried in every situation to select the appropriate parameters of the cost function. It could be populated by exploring the environment using our control approach.

An hybrid MPC/RL approach could be used to achieve this goal. The RL Agent would be trained to maximize a higher level reward function that depends upon the performances (accuracy and speed). It would control the robot through the choice of the parameters of the cost function. This means that the MPC, with the structure presented in this article, would still be used on the robot. The reference posture would however be reactively picked by the RL algorithm. [3] successfully deploys Proximal Policy Optimization to control a quadruped robot. However, the approach we present would be much more computationally intensive because of the more advanced control strcuture that would need to be simulated. That is why off-policy algorithms, such as Soft Actor-Critic, that are knwown to be more sample efficient would be more appropriate.

VI. Conclusion

This paper demonstrates the use of high frequency MPC to carry out a position task with an accuracy of few millimeters with the humanoid robot TALOS, controlled in torque. Strategies regarding the shaping of the cost function are the main focus of this article. Simulations show that changing the reference posture during the movement can improve the speed of completion of the task to human like levels.

In the short term, we plan to demonstrate the shown results on the robot. We also plan to continue this work by leveraging machine learning as a planification tool to reactively choose the appropriate reference configuration for a wide range of situations.

VII. Acknowledgment

This work was supported by the cooperation agreement ROB4FAM and the MERLHBOT Région Occitanie project. The use of the experimental platform TALOS was supported by ROBOTEX 2.0 (Grants ROBOTEX ANR-10-EQPX-44-01 and TIRREX-ANR-21- ESRE-0015). The author would like to thank Nahuel Villa for sharing personal notes on whole-body dynamics. He would also like to thank Ewen Dantec, Pierre Ferenbach, Maximilien Naveau, and Guilhem Saurel for the help they provided during the experiments as well as Sébastien Boria and Noélie Ramuzat who made testing at Airbus site possible.

References

[1] M. Hägele, K. Nilsson, J. N. Pires, and R. Bischoff, "Industrial robotics," in *Springer handbook of robotics*, pp. 1385–1422, Springer, 2016.

[2] A. Kheddar, S. Caron, P. Gergondet, A. Comport, A. Tanguy, C. Ott, B. Henze, G. Mesesan, J. Englsberger, M. A. Roa, P.-B. Wieber, F. Chaumette, F. Spindler, G. Oriolo, L. Lanari, A. Escande, K. Chappellet, F. Kanehiro, and P. Rabaté, "Humanoid robots in aircraft manufacturing: The airbus use cases," *IEEE Robotics and Automation Magazine*, vol. 26, no. 4, pp. 30–45, 2022.

[3] N. Rudin, D. Hoeller, M. Bjelonic, and M. Hutter, "Advanced skills by learning locomotion and local navigation end-to-end," in *Int. Conf. on Intelligent Robots and Systems*, 2022.

[4] H. Duan, A. Malik, J. Dao, A. Saxena, K. Green, J. Siekmann, A. Fern, and J. W. Hurst, "Sim-to-real learning of footstep-constrained bipedal dynamic walking," in *ICRA*, pp. 10428–10434, 2022.

[5] R. Grandia, F. Jenelten, S. Yang, F. Farshidian, and M. Hutter, "Perceptive locomotion through nonlinear model predictive control," *Arxiv*, 2022.

[6] S. Caron, A. Kheddar, and O. Tempier, "Stair climbing stabilization of the hrp-4 humanoid robot using whole-body admittance control," in *IEEE Int. Conf. on Robotics and Automation*, pp. 277–283, 2019.

[7] J. Englsberger, A. Werner, C. Ott, B. Henze, M.-A. Roa, G. Garofalo, R. Burger, A. Beyer, O. Eiberger, K. Schmid, and A. Albu-Schäffer, "Overview of the torque-controlled humanoid robot toro," in *IEEE Int. Conf. on Humanoid Robots*, p. 916-923, 2014.

[8] J. Hurst, "Enable humans to be more human," 2022. https://youtu.be/qc8wwrdvHlc.

[9] N. Ramuzat, G. Buondonno, S. Boria, and O. Stasse, "Comparison of position and torque whole-body control schemes on the humanoid robot talos," *Frontiers in Robotics and AI*, vol. 9, 2022.

[10] J. Englesberger, A. Dietrich, G.-A. Mesesan, G. Garofalo, C. Ott, and A. Albu-Schäffer, "Mptc – modular passive tracking controller for stack of tasks based control frameworks," in *Robotics Science and Systems*, 2020.

[11] W. Ubellacker and A. Ames, "Robust locomotion on legged robots through planning on motion primitive graphs wyatt ubellacker, aaron ames," in *Int. Conf. on Robotics Automation*, 2023.

[12] R. J. Griffin, G. Wiedebach, S. McCrory, S. Bertrand, I. Lee, and J. Pratt, "Footstep planning for autonomous walking over rough terrain," in *Int. Conf. on Humanoid Robotics*, 2019.

[13] R. Tedrake, "Lqr-trees: Feedback motion planning on sparse randomized trees," in *Robotics Science and Systems*, 2009.

[14] E. Dantec, R. Budhiraja, A. Roig, T.-S. Lembono, G. Saurel, O. Stasse, P. Fernbach, S. Tonneau, S. Vijayakumar, S. Calinon, M. Taïx, and N. Mansard, "Whole body model predictive control with a memory of motion: Experiments on a torque-controlled talos," in *IEEE Int. Conf. on Robotics and Automation*, pp. 8202–8208, 2021.

[15] E. Dantec, M. Taix, and N. Mansard, "First order approximation of model predictive control solutions for high frequency feedback," *IEEE/RAS Robotics and Automation Letters*, vol. 7, no. 2, pp. 4448–4455, 2022.

[16] J. Mirabel, F. Lamiraux, T.-L. Ha, A. Nicolin, O. Stasse, and S. Boria, "Performing manufacturing tasks with a mobile manipulator: from motion planning to sensor based motion control," in *IEEE Int. Conf. on Automation Science and Engineering*, pp. 159–164, 2021.

[17] E. d'Elia, J.-B. Mouret, J. Kober, and S. Ivaldi, "Automatic tuning and selection of whole-body controllers," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 12935–12941, IEEE, 2022.

[18] P.-B. Wieber, R. Tedrake, and S. Kuindersma, "Modelling and control of legged robots," in *Springer Handbook of Robotics*, pp. 1203–1234, Springer, 2016.

[19] E. C. Kerrigan, *Robust constraint satisfaction: Invariant sets and predictive control*. PhD thesis, University of Cambridge UK, 2001.

[20] E. Fernandez-Camacho and C. Bordons-Alba, *Model predictive control in the process industry*. Springer, 1995.

[21] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[22] W. Jallet, A. Bambade, N. Mansard, and J. Carpentier, "Constrained differential dynamic programming: A primal-dual augmented lagrangian approach," in *IEEE Int. Conf. on Robotics and Automation*, 2022.