Whole body control using Robust & Online hierarchical quadratic optimization

Hilario Tomé[1], Luca Marchionni[1], and Adolfo Rodriguez Tsouroukdissian[1]

[1]PAL Robotics S.L.

***Abstract*— Recently, several formulations have been proposed to add inequality constraints to multi objective prioritized optimization problems. How to solve this problems with only equality constraints is a well known topic in robotics. Inequality constraints behave as equalities when they reach their bounds, so ideally we shouldn't bother about them before we reach them. How we take them into account and what to do when they are reached affects drastically the computational effort of solving the problem. We present and derive an efficient way to deal with these problems using an off the shelf Quadratic Programming solver and choosing an appropriate solving strategy. We finally apply the proposed method to do whole body control in real time on a humanoid robot with 44 dof using a hierarchy of objectives.**

I. INTRODUCTION

Control of humanoid robots is a challenging task due to their high number of degrees of freedom and their floating base underactuation. An intuitive way to program them is by combining different objectives and solve for the joint level commands that will minimise the error of the desired objectives. It often occurs that the different objectives compete against each other, and it is up to the user to decide which one should the optimization favour. There are two ways to do this: weighting and prioritization. Weighting suffers the problem that the associated gains with each level are not representative of the importance criterion, this is due to the fact that the matrix norms of each objective are different. The second way is to enforce a strict prioritization between the objectives, that we will call levels of the hierarchy. This approach is known as lexicographic optimization. In section II we will describe how to solve hierarchical quadratic problems with equality and inequality constraints. Section III reviews the basics of constraint optimization theory, with emphasis on problems that have quadratic form. Section IV introduces the active set algorithms and explains why we use the dual active set method to increase robustness. Finally in section V we will explain how we use the above introduced concepts to do whole body control with REEM-C biped humanoid.

II. HIERARCHICAL QUADRATIC PROGRAMMING

The k levels of the hierarchy want to minimize the squared norm of

$$\|\mathbf{A}_k\mathbf{x} - \mathbf{b}_k\|_2^2 \tag{1}$$

This cost function has a quadratic form, where $\mathbf{A}_k$ usually maps increments in the generalized coordinates of the robot to a task space increment $\mathbf{b}_k$. The objectives can be defined at

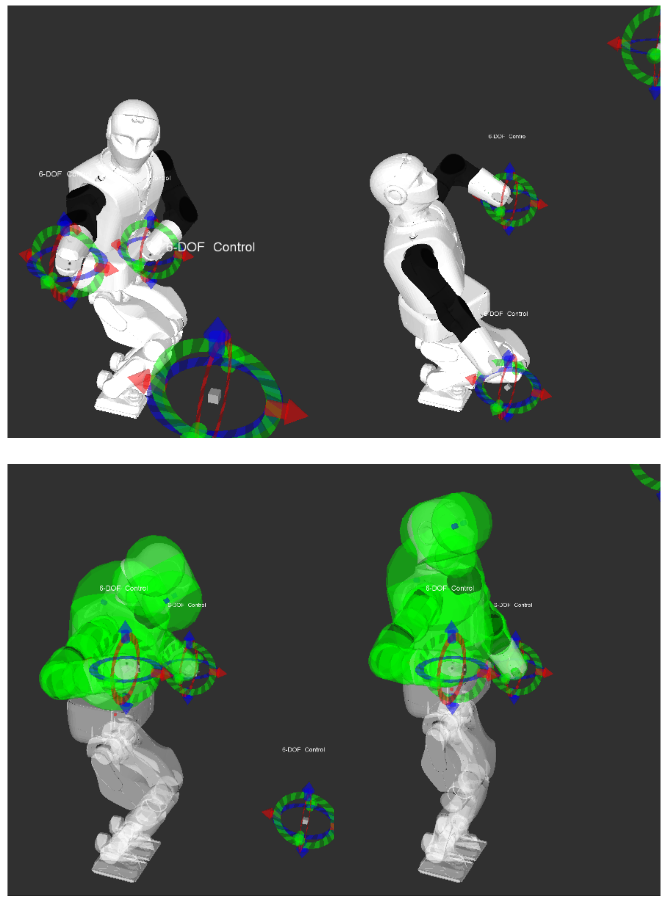

Fig. 1: REEM-C performing whole body control with a hierarchy of objectives: joint limits $\succ$ self collision avoidance $\succ$ fixed feet + com centered $\succ$ gaze $\succ$ hands position $\succ$ torso orientation upright.

the kinematic and dynamic level. The basis of the null space of $\mathbf{A}_k$ will be indicated with $\mathbf{Z}_k$ and the projection operator into the null space with $\mathbf{P}_k$. We will introduce different approaches to find the optimal $\mathbf{x}_p^\star$ between all the levels subject to prioritization, equality and inequality constraints.

1) Recursive null space projection for equality constraint: Recursive null space projection was proposed in [Siciliano and Slotine, 1991]. At each level k we solve the associated least squares problem and project the solution in the null space of the previous one. This approach requires solving an $c \times m$ problem at each level, where c

are the number of rows of $\mathbf{A}_k$ and m the size of the state vector of the robot. $\mathbf{P}_k$ can be found recursively [Siciliano and Slotine, 1991].

$$\mathbf{x}_p^\star = \sum_{k=1}^{p} (\mathbf{A}_k \mathbf{P}_{k-1})^+ (\mathbf{b}_k - \mathbf{A}_k \mathbf{x}_{k-1}^\star) \tag{2}$$

2) Searching in the null space basis for equality constraints: [de Lasa and Hertzmann, 2009] introduced an optimization that allows to reduce the computation at each level. Instead of solving over the entire m dimensional state of the robot, the authors propose to search for the solution directly in the remaining null space of the previous solved levels. This formulation also lets us monitor the remaining degrees of freedom.

$$\begin{aligned} \min_{\mathbf{z}_k, \mathbf{w}_k} \quad & \|\mathbf{w}_k\| \\ \text{s.t.} \quad & \mathbf{A}_k(\mathbf{x}_{k-1}^\star + \tilde{\mathbf{Z}}_{k-1}\mathbf{z}_k) = \mathbf{b}_k + \mathbf{w}_k \end{aligned} \tag{3}$$

The optimal solution can be calculated recursively

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \tilde{\mathbf{Z}}_{k-1}(\mathbf{A}_k \tilde{\mathbf{Z}}_{k-1})^+ (\mathbf{b}_k - \mathbf{A}_k \mathbf{x}_{k-1}^\star) \tag{4}$$

Where $\tilde{\mathbf{Z}}_k$ is the the multiplication of null spaces of all the previous levels computed recursively as

$$\tilde{\mathbf{Z}}_k = \tilde{\mathbf{Z}}_{k-1}\mathbf{Z}_k \tag{5}$$

It is important to note how the size of the matrices $\tilde{\mathbf{Z}}_k$ decreases as we advance through the hierarchy while in (2) all the $\mathbf{P}_k$ remain constant in size through the hierarchy.

3) Cascade QP with inequality constraints: Given a single quadratic objective with associated inequality constraints on the decision variables we can use a QP solver to get the optimal solution that is feasible with the associated constraints. The question arises on how to solve a hierarchy of such problems, each with its own associated inequality constraints. [Kanoun et al., 2011] proposed to solve this problem using a cascade of QP's. Each level k is solved having as additional constraints that the solution cannot modify the optimal residual found in the previous $k-1$ levels.

$$\begin{aligned} \min_{\mathbf{x}_k, \mathbf{w}_k, \mathbf{v}_k} \quad & \|\mathbf{w}_k\| + \|\mathbf{v}_k\| \\ \text{s.t.} \quad & \mathbf{A}_k \mathbf{x}_k = \mathbf{b}_k + \mathbf{w}_k \\ & \mathbf{C}_k \mathbf{x}_k \leq \mathbf{d}_k + \mathbf{v}_k \\ & \underline{\mathbf{A}}_{k-1}\mathbf{x}_k = \underline{\mathbf{b}}_{k-1} + \underline{\mathbf{w}}_{k-1}^\star \\ & \underline{\mathbf{C}}_{k-1}\mathbf{x}_k \leq \underline{\mathbf{d}}_{k-1} + \underline{\mathbf{v}}_{k-1}^\star \end{aligned} \tag{6}$$

$\underline{\mathbf{A}}_{k-1}$, $\underline{\mathbf{b}}_{k-1}$, $\underline{\mathbf{w}}_{k-1}$, $\underline{\mathbf{v}}_{k-1}^\star$ is the concatenation of the corresponding $k-1$ matrices and vectors.

4) Cascade QP with reduction of the equality constraints: Following the idea in [de Lasa and Hertzmann, 2009] we can remove the degrees of freedom of the equality constraints on each level and only leave in the cascade the inequality constraints. If an inequality constraint is active, all the subsequent levels will loose a degree of freedom, just as if the constraint would had been an equality. We note that this is the intuition behind the active set method that we will introduce later to solve each QP. This formulation has been used in [Alexander Herzog, 2013] to solve prioritized inverse dynamics.

$$\begin{aligned} \min_{\mathbf{x}_k, \mathbf{w}_k, \mathbf{v}_k} \quad & \|\mathbf{w}_k\| + \|\mathbf{v}_k\| \\ \text{s.t.} \quad & \mathbf{A}_k(\mathbf{x}_{k-1}^\star + \tilde{\mathbf{Z}}_{k-1}\mathbf{z}_k) = \mathbf{b}_k + \mathbf{w}_k \\ & \mathbf{C}_k(\mathbf{x}_{k-1}^\star + \tilde{\mathbf{Z}}_{k-1}\mathbf{z}_k) \leq \mathbf{d}_k + \mathbf{v}_k \\ & \underline{\mathbf{C}}_{k-1}(\mathbf{x}_{k-1}^\star + \tilde{\mathbf{Z}}_{k-1}z_k) \leq \underline{\mathbf{d}}_{k-1} + \underline{\mathbf{v}}_{k-1}^\star \end{aligned} \tag{7}$$

Using the last formulation lets us use an off the shelf QP solver, and potentially if there are no active inequality constraints the computational cost will be the same as 3 (depending on the QP algorithm used).

We could further remove the degrees of freedom gained by the active inequality constraints to reduce the size of remaining null space. Because the inequalities become active and inactive during the optimization the $\tilde{\mathbf{Z}}_k$ will change correspondingly gaining or loosing a degree in its rank. To avoid recomputing the decompositions, in [Escande et al., 2014] the authors present an efficient way to update the null space of every level using a new decomposition called the HCOD and a custom active set QP optimizer.

III. OPTIMIZATION THEORY REVIEW

A. General optimization program

Given a nonlinear problem of the form

$$\begin{aligned} \min_{x} \quad & f_0(x) \\ \text{s.t} \quad & g_i(x) \leq 0, h_j(x) = 0 \end{aligned} \tag{8}$$

where $f_0(x)$ $g_i(x)$ and $h_i(x)$ are continuous differentiable functions, the necessary conditions for a local minimum $\mathbf{x}^\star$ to the problem are called the KKT (Karush-Kuhn-Tucker) conditions:

- Stationary.

$$\begin{aligned} \mathcal{L} = \nabla f(\mathbf{x}^*) + \sum_{i=1}^{m} \mu_i \nabla g_i(\mathbf{x}^*) \\ + \sum_{j=1}^{l} \lambda_j \nabla h_j(\mathbf{x}^*) = 0 \end{aligned} \tag{9}$$

- Primal feasibility: Satisfaction of the original problem constraints.

$$\begin{aligned} g_i(\mathbf{x}^*) \leq 0, \quad \forall i = 1, \ldots, m \\ h_j(\mathbf{x}^*) = 0, \quad \forall j = 1, \ldots, l \end{aligned} \tag{10}$$

- Dual feasibility: The slack variables associated with inequalities must non negative.

$$\mu_i \geq 0, \forall i = 1, \ldots, m \tag{11}$$

- Complementary slackness: If an inequality is not active its associated slack variable must be 0.

$$\mu_i g_i(\mathbf{x}^*) = 0, \quad \forall i = 1, \ldots, m \tag{12}$$

B. Quadratic program

We are interested in a particular form of problem where the objective is quadratic and the constraints are affine. This problem is convex, making the KKT conditions not only necessary but also sufficient, meaning that if we find a local solution it will also be the global solution.

$$\begin{aligned} \min_{x} \quad & \mathbf{J}(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x} \\ \text{s.t} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{I}\mathbf{x} \geq \mathbf{d} \end{aligned} \tag{13}$$

C. Dual Quadratic program

The above optimization problem can be solved from a different point of view called the Lagrange dual problem [Fletcher, 1987], which consists on the original problem with the constrains added to the objective function with an associated multiplier. This new function is called the Lagrangian (9). The problem now consists on optimizing over the dual variables λ that are formulated as functions of the primal variables. The solution to the dual problem provides a lower bound to the primal problem (weak duality). The difference between the optimal value of both problems is called the duality gap, which in the case of convex optimization problems is zero (strong duality), meaning that we will arrive to the same global optimum independently if we choose the primal or dual. We will exploit this fact to add robustness to the optimization algorithm that we will present in section IV. We drop the equality constraints for the sake of simplification in the following derivation for sake of brevity.

$$\begin{aligned} \max_{x,\lambda} \quad & \mathbf{J}(x) = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x} - \lambda(\mathbf{I}\mathbf{x} - \mathbf{d}) \\ \text{s.t} \quad & \mathbf{Q}\mathbf{x} + \mathbf{c} - \mathbf{I}^T\lambda = 0 \\ & \lambda \geq 0 \end{aligned} \tag{14}$$

We can use the constraints on the dual to reduce the maximization only on the dual variables

$$\begin{aligned} \max_{\lambda} \quad & \mathbf{J}(\mathbf{x}) = -\frac{1}{2}\lambda^T(\mathbf{I}\mathbf{Q}^{-1}\mathbf{I}^T)\lambda + \\ & \lambda^T(\mathbf{d} + \mathbf{I}\mathbf{Q}^{-1}\mathbf{c}) - \frac{1}{2}\mathbf{c}^T\mathbf{Q}^{-1}\mathbf{c} \\ \text{s.t} \quad & \lambda \geq 0 \end{aligned} \tag{15}$$

Once the $\lambda^\star$ are known we can get $\mathbf{x}^\star$ with $\mathbf{Q}\mathbf{x}^\star = \mathbf{A}^T\lambda^\star - \mathbf{c}$

IV. Active set algorithms

The primal active set method (Algorithm 2) belongs to the modified type simplex methods. This method starts with a feasible point and iterates searching for the optimal solution along the edges of the feasible set. The algorithm converts an inequality constraint to an equality one if it's violated, and solves the associated equality constraint problem. The indices of the activated inequalities are stored in the active set $\mathcal{A}$. Information from the Lagrange multipliers is used to add or remove constraints from $\mathcal{A}$. The main feature of this

Data: $\mathbf{A}$, $\mathbf{b}$, $\mathbf{x}$ is the current state, $\mathbf{p}$ is the step direction, $\mathcal{A}$ is the current active set
Result: α the step length in the step direction $\mathbf{p}$ that does not violate any primal constraint, cst is the identifier of the constraint with minimum α

$$\alpha = \min\left(1, \min_{\substack{i \notin \mathcal{A} \\ a_i^T\mathbf{p} < 0}} \frac{b_i - a_i^T\mathbf{x}}{a_i^T\mathbf{p}}\right) \tag{16}$$

Algorithm 1: Primal line search

Data: A QP of the form (13), a feasible starting point with its associated active set $\mathcal{A}$
Result: Optimal solution $\mathbf{x}^\star$, with optimal active set $\mathcal{A}$, if the maximum iterations is reached the algorithm has failed

```
iter = 0;
while iter < max_iterations do
    Compute the solution p to the equality QP;
    Compute step length α, cst using Algorithm 1;
    x* = x + αp;
    if α = 1 then
        Compute Lagrange multipliers λ_k to test optimality;
        {v, cst} = min{1, λ_i};
        if v >= 0 then
            TERMINATE Optimal solution found;
        else
            A := A\{v}
    else
        A := A ∪ {cst}
    iter = iter + 1
```

Algorithm 2: Primal active set algorithm

method is that it finds the optimal solution while staying primal feasible (10), due to this the optimization can be stopped at any point and the primal constraints will still be satisfied.

The dual active set method [Goldfarb and Idnani, 1983] is identical to the primal with the form of the objective changed to (15). The algorithm maintains dual feasibility (11) until primal feasibility is achieved (dual optimality). It has the benefit that it does not need an initial feasible point since the origin is always a dual feasible point. Once a primal constraint becomes feasible we can optionally leave it feasible by modifying the line search in Algorithm 1.

A. Degeneracy

Cycling is when a constraint is deactivated in one iteration and reactivated in the next one cyclically not allowing the solver to converge to the solution. One of the main sources of cycling in active set methods is degeneracy in $\mathcal{A}$. Degeneracy happens when the normal of the constraints gets close to be linearly dependent, for example when a kinematic chain becomes singular or two tasks have similar constraints. The dual active set solves this problem by not having degeneracy in the constraints by construction. The constrains are only the positiveness associated with the Lagrange multipliers, having a constraint matrix that is an identity. The caveat is that $\mathbf{Q}$ must be strictly positive definite. If this is not the case, small regularization can be added to its diagonal to enforce this property.

V. WHOLE BODY CONTROL

We will create a combination of tasks that is typical to control the behaviour of a humanoid robot. The objectives will be at kinematic velocity level with the following hierarchy: joint limits $\succ$ self collision avoidance$\succ$ fixed feet + com centered $\succ$ gaze $\succ$ hands position $\succ$ torso orientation upright. Fig. 1 shows the result of the optimization in different configurations. A small description of each task:

- Center of mass: This task relates joint velocity to the center of mass velocity of the robot. It is used to restrict the projection on the ground of the center of mass to stay in the middle of the ankles of the robot.
- Reach: This task relates joint velocity to the cartesian velocity of a link. Given the actual position of a link and its desired one, a velocity target is created with a proportional controller to reach the desired target.
- Fixed Constraint: This task has the same form as the previous one, but the target is zero, meaning that the link should have zero velocity an thus stay fixed.
- Gaze: This task maps the velocity of a 3d point to its 2d image velocity. Computing the error between the 2d projection of a 3d point and the origin of the camera frame we can make the robot look directly at the desired point with a target proportional to the error.
- Self collision: Given a simplification of the robot, where every link is fitted with a minimum volume capsule [Stasse et al., 2008], we can avoid self collision by adding an inequality to the relative velocity between the capsules of two links. The relative velocity has to be sufficiently small to avoid interpenetration.
- Joint limit: We can avoid joint limits by restricting the maximum allowed velocity proportional to how far we are from the limits.

VI. CONCLUSION

We have proposed a solution to resolve the redundancy of a humanoid robot in real time using a sequence of QP problems that decrease in size and the dual active set strategy to solve each QP. This allows us to generate feasible and safe kinematic configurations. We are currently working in adding dynamics into the optimization and testing the controller on the real robot.

REFERENCES

[Alexander Herzog, 2013] Alexander Herzog, Ludovic Righetti, F. G. P. P. S. S. (2013). Experiments with a hierarchical inverse dynamics controller on a torque-controlled humanoid.

[de Lasa and Hertzmann, 2009] de Lasa, M. and Hertzmann, A. (2009). Prioritized optimization for task-space control. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 5755–5762.

[Escande et al., 2014] Escande, A., Mansard, N., and Wieber, P.-B. (2014). Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *International Journal of Robotics Reasearch*. in press.

[Fletcher, 1987] Fletcher, R. (1987). *Practical Methods of Optimization; (2Nd Ed.)*. Wiley-Interscience, New York, NY, USA.

[Goldfarb and Idnani, 1983] Goldfarb, D. and Idnani, A. (1983). A numerically stable dual method for solving strictly convex quadratic programs. *Mathematical Programming*, 27(1):1–33.

[Kanoun et al., 2011] Kanoun, O., Lamiraux, F., and Wieber, P.-B. (2011). Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task. 27(4):785–792.

[Siciliano and Slotine, 1991] Siciliano, B. and Slotine, J.-J. (1991). A general framework for managing multiple tasks in highly redundant robotic systems. In *Advanced Robotics, 1991. 'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, pages 1211–1216.

[Stasse et al., 2008] Stasse, O., Escande, A., Mansard, N., Miossec, S., Evrard, P., and Kheddar, A. (2008). Real-time (self)-collision avoidance task on a hrp-2 humanoid robot. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 3200–3205.