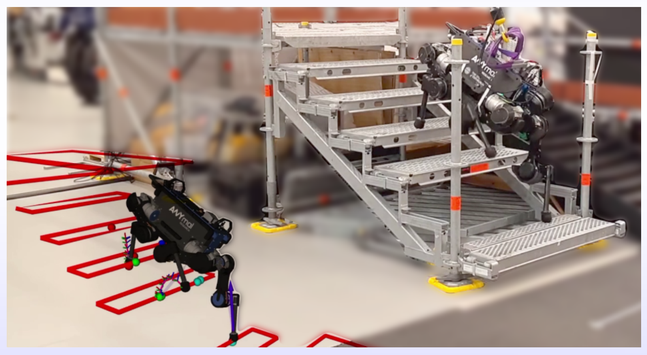


Inverse-Dynamics MPC via Nullspace Resolution

Carlos Mastalli[1†] Saroj Prasad Chhatoi[2†] Thomas Corbères[3] Steve Tonneau[3] Sethu Vijayakumar[3]

[1] *School of Engineering and Physical Sciences, Heriot-Watt University, U.K.*
[2] *Centro di Ricerca "Enrico Piaggio", Università di Pisa, Italy*
[3] *School of Informatics, University of Edinburgh, U.K.*

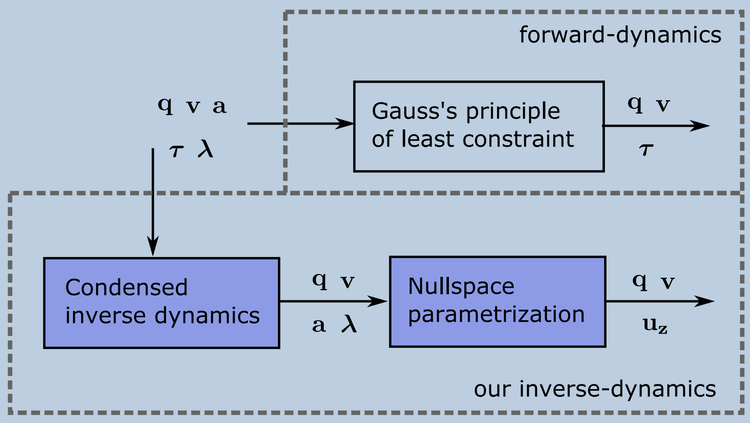
Introduction

Optimal control (OC) using inverse dynamics provides numerical benefits such as coarse optimization, cheaper computation of derivatives, and a high convergence rate. But to enjoy this in MPC settings we need to handle the inverse-dynamics equality constraints efficiently and exploit its structural and temporal structure. The **contribution of our work** are the follows

- A novel equality-constrained DDP that handle efficiently the inverse-dynamics constraints via nullspace parametrization.
- A combination of feasibility-driven search and merit function that increases the algorithm's basin of attraction.
- An original formulation of inverse dynamics that considers arbitrary actuator models.
- A unique feedback MPC based on inverse dynamics integrated into a perceptive locomotion pipeline.

Optimal control approaches

Thanks to Gauss's principle of least constraint, the forward dynamics fit naturally into classical optimal control.
To reduce the computational cost in inverse dynamics settings, our approach condenses the inverse dynamics and uses a nullspace parametrization:



where $\mathbf{q}$, $\mathbf{v}$, $\mathbf{a}$ are the generalized position, velocity and acceleration, respectively, $\boldsymbol{\tau}$ and $\boldsymbol{\lambda}$ are the joint torques and contact forces.
The problem of **optimal control with inverse dynamics** is described as

$$\min_{\mathbf{x}_s,\mathbf{u}_s} \ \ell_N(\mathbf{x}_N) + \sum_{k=0}^{N-1} \ell_k(\mathbf{x}_k,\mathbf{u}_k) \qquad (1)$$

$$\text{s.t.} \quad \underbrace{\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k,\mathbf{u}_k)}_{\text{kinematic evolution}}, \quad \underbrace{\mathbf{h}_k(\mathbf{x}_k,\mathbf{u}_k) = \mathbf{0}}_{\text{inverse dynamics}}.$$

To solve the problem efficiently we need to (i) **exploit the temporal/functional structure** and (ii) **handle equality constraints efficiently**.

Equality-constrained DDP

To exploit the temporal structure of an inverse-dynamics optimal control problem, we apply the ***dynamic programming principle*** as in the DDP algorithm, i.e.,

$$\Delta\mathcal{V} = \min_{\delta\mathbf{u}} \ \frac{1}{2}\begin{bmatrix}\delta\mathbf{x}\\ \delta\mathbf{u}\end{bmatrix}^\top \begin{bmatrix}\mathbf{Q_{xx}} & \mathbf{Q_{ux}}^\top\\ \mathbf{Q_{ux}} & \mathbf{Q_{uu}}\end{bmatrix}\begin{bmatrix}\delta\mathbf{x}\\ \delta\mathbf{u}\end{bmatrix} + \begin{bmatrix}\delta\mathbf{x}\\ \delta\mathbf{u}\end{bmatrix}^\top \begin{bmatrix}\mathbf{Q_x}\\ \mathbf{Q_u}\end{bmatrix}$$
$$\text{s.t.} \quad \begin{bmatrix}\mathbf{h_x} & \mathbf{h_u}\end{bmatrix}\begin{bmatrix}\delta\mathbf{x}\\ \delta\mathbf{u}\end{bmatrix} + \bar{\mathbf{h}} = \mathbf{0}, \qquad (2)$$

where the $\mathbf{Q}$'s describe the local approximation of the *action-value function* in the free space, and $\mathbf{h_x}, \mathbf{h_u}$ are the Jacobians of the inverse-dynamics constraint.

This quadratic program has the following first-order necessary conditions (FONC) of optimality:

$$\begin{bmatrix}\mathbf{Q_{uu}} & \mathbf{h_u}^\top\\ \mathbf{h_u} & \end{bmatrix}\begin{bmatrix}\delta\mathbf{u}\\ \boldsymbol{\gamma}^+\end{bmatrix} = -\begin{bmatrix}\mathbf{Q_u} + \mathbf{Q_{ux}}\delta\mathbf{x}\\ \bar{\mathbf{h}} + \mathbf{h_x}\delta\mathbf{x}\end{bmatrix},$$

where $\boldsymbol{\gamma}^+ = \boldsymbol{\gamma} + \delta\boldsymbol{\gamma}$ is the next value of Lagrange multiplier associated to $\mathbf{h}$.

The FONC are often solved using the ***Schur-complement*** factorization, but it **increases computational complexity**. To address this, we propose a *nullspace* approach that decomposes $\delta\mathbf{u}$ into its null and range spaces.

Nullspace factorization

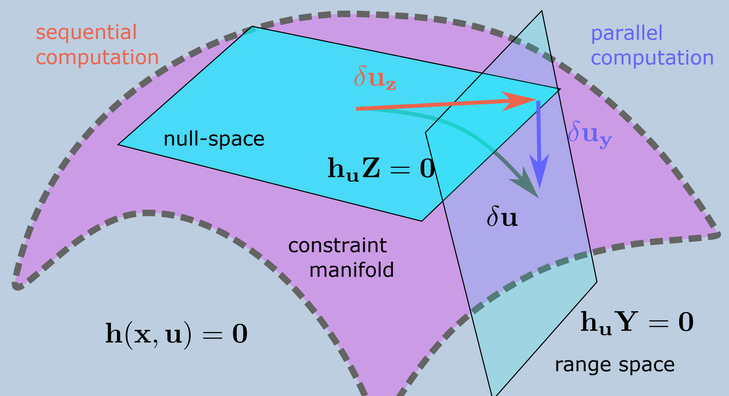
Our approach begins by parametrizing $\delta\mathbf{u}$ as

$$\delta\mathbf{u} = \mathbf{Y}\delta\mathbf{u_y} + \mathbf{Z}\delta\mathbf{u_z},$$

where $\mathbf{Z} \in \mathbb{R}^{n_u \times n_z}$ is the nullspace basis of $\mathbf{h_u}$, $[\mathbf{Y}\ \mathbf{Z}]$ spans $\mathbb{R}^{n_u}$. Using the condition $\mathbf{h_u Z} = \mathbf{0}$ inside optimality condition:

$$\delta\mathbf{u} = -\boldsymbol{\pi}_n - \boldsymbol{\Pi}_n\,\delta\mathbf{x},$$

with
$$\boldsymbol{\pi}_n \coloneqq \mathbf{Z}\hat{\mathbf{k}}_n + \hat{\mathbf{Q}}_{\mathbf{zz}}\boldsymbol{\Psi}_n\bar{\mathbf{h}} \quad \text{(feed-forward)},$$
$$\boldsymbol{\Pi}_n \coloneqq \mathbf{Z}\hat{\mathbf{K}}_n + \hat{\mathbf{Q}}_{\mathbf{zz}}\boldsymbol{\Psi}_n\mathbf{h_x} \quad \text{(feedback gain)}, \qquad (3)$$



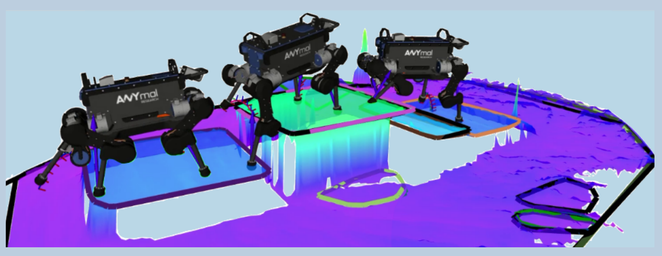
where $\hat{\mathbf{k}}_n = \mathbf{Q_{zz}}^{-1}\mathbf{Q_z}$, $\hat{\mathbf{K}}_n = \mathbf{Q_{zz}}^{-1}\mathbf{Q_{zx}}$, $\hat{\mathbf{Q}}_{\mathbf{zz}} = \mathbf{I} - \mathbf{Z}\mathbf{Q_{zz}}^{-1}\mathbf{Q_{zu}}$, $\boldsymbol{\Psi}_n = \mathbf{Y}(\mathbf{h_u Y})^{-1}$, $\mathbf{Q_z} = \mathbf{Z}^\top\mathbf{Q_u}$, $\mathbf{Q_{zx}} = \mathbf{Z}^\top\mathbf{Q_{ux}}$, $\mathbf{Q_{zu}} = \mathbf{Z}^\top\mathbf{Q_{uu}}$, $\mathbf{Q_{zz}} = \mathbf{Q_{zu}}\mathbf{Z}$.

When parametrizing the constraint, we search the direction along its null and range spaces (blue planes). This allows us to **perform parallel computations** for computing $\delta\mathbf{u_y}$ of each node, which reduces the algorithmic complexity of the Riccati recursion.

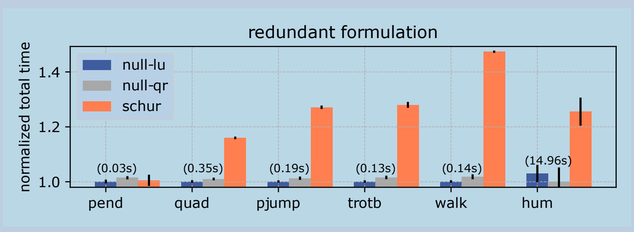
Inverse-dynamics MPC, feedback policy and results (https://youtu.be/NhvSUVopPCI)

MPC and pipeline

Our inverse-dynamics MPC computes whole-body motions, contact forces, and feedback policies for the generalized acceleration and contact forces at a fixed optimization horizon.



Our **nullspace factorization reduces the computation**, which is fast enough for MPC applications.



Feedback MPC in joint-torque space

Our inverse-dynamics MPC computes control policies for the generalized acceleration and contact forces, i.e.,

$$\begin{bmatrix}\delta\mathbf{a}^*\\ \delta\boldsymbol{\lambda}^*\end{bmatrix} = -\begin{bmatrix}\boldsymbol{\pi}_\mathbf{a}\\ \boldsymbol{\pi}_{\boldsymbol{\lambda}}\end{bmatrix} - \begin{bmatrix}\boldsymbol{\Pi}_\mathbf{a}\\ \boldsymbol{\Pi}_{\boldsymbol{\lambda}}\end{bmatrix}\delta\mathbf{x}, \qquad (4)$$

To map this policy into the joint-torque space, we use the inverse dynamics, i.e.,

$$\begin{aligned}\delta\boldsymbol{\tau}^* &= -\boldsymbol{\pi}_{\boldsymbol{\tau}} - \boldsymbol{\Pi}_{\boldsymbol{\tau}}\delta\mathbf{x},\\ &= (\mathbf{S}^*)^{-1}\mathrm{ID}(\mathbf{q}^*,\mathbf{v}^*,\delta\mathbf{a}^*,\delta\boldsymbol{\lambda}^*),\end{aligned}$$

where $\mathbf{S}^* = \mathbf{I} - \mathrm{A}^*_{\boldsymbol{\tau}}(\mathrm{A}^*_{\boldsymbol{\tau}})^{-1}$, with $\mathrm{A}^*_{\boldsymbol{\tau}} = \frac{\partial \mathrm{A}}{\partial\boldsymbol{\tau}}\Big|_{\mathbf{q}^*,\mathbf{v}^*}$, is the selection matrix linearized around the optimal state that is described by $\mathbf{q}^*$, $\mathbf{v}^*$. This boils down into two procedures, one for each term: feed-forward and feedback.

Our feedback MPC produces **motions that reach the limits** of our ANYmal robot.

Our inverse-dynamics MPC quickly **recovers against push forces** thanks to the adaptation of the *motion feasibility*.