Abstract—Control algorithms of large Degrees Of Freedom (DOFs) robotic systems based on Inverse-Kinematic or Inverse-Dynamic approaches are two well known topics of research in control theory. In the last years those subjects never diminished their importance due to the progressively increase of the number of DOFs in robotic applications. Among the various possibilities, the design of tasks arranged in priorities is one of the prominent. Task-priority at the kinematic or dynamic level is witnessing an increase interest especially in the humanoid or floating-base (underwater or aerial) communities. This paper investigates the effects of modeling errors in control algorithms based on Inverse Dynamics with respect to the uncertain knowledge of the dynamic parameters of the model. In particular, it is of interest to understand the effect on the null-space projections and the sources of steady state errors.

I. INTRODUCTION

The use of large Degrees-Of-Freedom (DOFs) robotic structures is nowadays common in applications such as humanoid or aerial and underwater robots. Among the various approaches used to control all the DOFs, a popular one is to design several control variables and to arrange them in priority, in a sense that will be clarified later. Within the task-priority approach two frameworks are possible, the Inverse-Kinematic (IK) or the Inverse-Dynamic (ID). The former commands the robot at position or velocity level, while the latter at torque level.

This paper discusses modeling errors analysis of ID control algorithms with respect to the uncertain knowledge of the dynamic parameters of the mathematical model.

Due to the use of the dynamically consistent pseudoinverse of the Jacobian matrix within the control law, in fact, all the dynamic parameters, included the inertial ones, potentially affect the steady state. In this paper it will be mathematically proven that the steady state error is influenced only by the errors in the gravity terms compensation, i.e., the first moment of inertia, and the eventual conflict among the tasks but not by the error in the estimation of the inertial parameters as long as the mass matrix is positive definite.

The validity of the comments has been extensively tested by resorting to different robotic structures and by varying the number and kind of tasks. In this paper a 7DOF robot in two case studies, with compatible and conflicting tasks, is used as testbed.

The next Section discusses the relevant literature, Sec. III provides the necessary mathematical background, Sec. IV discusses modeling error issues followed by numerical analysis and finally by the Conclusions.

II. LITERATURE SURVEY

In [1], the operational space approach was first presented to the community. Starting from that seminal input during the years a huge research activity has been conducted up to its application to humanoids [2], [3], [4]. In [5] several task-priority control laws are presented within a unified approach. The authors show that they all minimize a proper index. The paper also shows an experimental comparison among various techniques in which the resolved rate appears to exhibit superior performances. This has been interpreted as a possible effect of the uncertainty in the knowledge of the dynamic parameters; in detail it is noticed that this might be amplified by the inversion of the inertia matrix.

In [6], [7] an interesting theoretical and experimental comparison among various kinds of task space control with redundancy resolution is presented. Resolution at velocity, acceleration and torque level is considered for control problems where the primary task is always the end-effector pose while the secondary is the optimization of a proper functional. In the numerical simulation, under ideal conditions, all the controllers exhibited the same performance (“in numerical simulation, all controllers achieved the same excellent performance, . . . ”). The following experimental differences are to be ascribed to the unmodelled dynamics, the uncertainty in the model knowledge, the sensor noise and quantization. In particular, the authors notice that acceleration and torque based control requires an accurate knowledge of the inertia matrix in order to achieve task decoupling; even if an accurate modelling and identification procedure were implemented, the secondary and primary tasks were coupled. The results reported in the present paper allow to better refine understanding of this, showing how the uncertainties on the inertia matrix used as weight in the pseudoinverse of the tasks Jacobian matrices do not couple tasks with different priority. Moreover, the authors refer of several cases of instability that, for the acceleration-based control laws, were ascribed to the difficulty of finding “a tradeoff between filtering of numeric derivative and choosing higher gain parameters”. However, the instability problems found by the authors could have been also related to the finite sampling time (unfortunately not reported in the paper) of the digital control implementation and, for the velocity-based controllers, to the use of actual joint states in the computation.
of joint references $\hat{q}_e$ in the velocity-based controllers. In fact, when actual joint states are used to compute task space errors, then the dynamics of the inner velocity loops play a key role in the stability of the overall system. Both issues, discrete time implementation and non ideal dynamics of joint velocity loops, are discussed in detail in [8] and [9], where limits to the task space gain are explicitly found related to both sampling time and bandwidth of inner motion control loops.

One of the few papers with an experimental implementation of multiple tasks at dynamic level is provided by [10] where 3 tasks are arranged in priority, namely the end-effector position, the orientation and the joint configuration. The experiments show steady state errors for both the first and second priority tasks which have been explained by the authors as caused by the presence of dry friction and uncertainty in the dynamic knowledge due to the absence of any integral action in their controller. Multiple-task priority is addressed also in [11], [12].

One of the main reasons why ID approaches are used is due to the need to implement impedance control, which is not possible in IK unless a force-torque sensor is used. In the latter case the interaction control loop is defined as admittance or as position-based impedance. An interesting comparison of impedance vs admittance is made in [13]. The strong dependency of the impedance control from the model knowledge is stressed. Due to the design, the position control loop of the admittance needs to be designed as to be faster than the impedance. However, when a disturbance is present out of the force sensor the system reacts with the inner loop dynamics and not with the desired impedance. The comparison, however, uses model-based control also for the admittance which is not mandatory. In the 1-DOF case of impedance control it is recognized that the sensitivity increases with a low desired bandwidth.

### III. Background

#### A. Pseudoinverse

A linear transformation of the kind

$$y = Jx$$

with $J \in \mathbb{R}^{m \times n}$, $m < n$  \hspace{1cm} (1)

admits infinite solutions in $x$.

A general solution for eq. (1) with full rank Jacobian exhibits the structure

$$x = J^\dagger y + \underbrace{(I_n - J^\dagger J)}_N x_0$$

\hspace{1cm} (2)

where $I_n$ is the $(n \times n)$ Identity matrix, with arbitrary $x_0 \in \mathbb{R}^n$ and being $J^\dagger$ the Moore-Penrose pseudo-inverse of $J$. The matrix $N \in \mathbb{R}^{n \times n}$ is the null-space projector associated to $J$, i.e., it projects any vector into the null space of $J$, in fact

$$JN = O_{m \times n}, \quad JN x_0 = 0_m.$$ \hspace{1cm} (3)

The matrix $N$ is a projection matrix, i.e., $N^2 = N$ (idempotent property), it is orthogonal (its image and null spaces are orthogonal), it holds $NJ^\dagger = O_{n \times m}$ and it is symmetric only for the non-weighted case.

#### B. Joint-space Dynamics

Equations of motion for serial-chain rigid-body dynamic systems can be written in matrix form as [14]:

$$M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + g(q) = \tau.$$ \hspace{1cm} (4)

Denoting with $n$ the number of joints, $q \in \mathbb{R}^n$ is the joint position vector, the dot operator represents the time derivative, thus, $\dot{q} \in \mathbb{R}^n$ and $\ddot{q} \in \mathbb{R}^n$ are the vectors of joint velocities and accelerations, respectively, $\tau \in \mathbb{R}^n$ is the joint torque vector, $M(q) \in \mathbb{R}^{n \times n}$ is the inertia matrix, $C(q, \dot{q}) \dot{q} \in \mathbb{R}^n$ is the Coriolis and centripetal vector and $g(q) \in \mathbb{R}^n$ is the gravity torque vector. By posing

$$n(q, \dot{q}) = C(q, \dot{q}) \dot{q} + g(q),$$

eq. (4) is often presented in the more compact form

$$M(q) \ddot{q} + n(q, \dot{q}) = \tau.$$ \hspace{1cm} (5)

#### C. Task-space Dynamics

The control objective is often the position and orientation of the end effector, however, in advanced robotics and with many DOFs available, several other tasks can be defined. It is then possible to introduce a generic task as

$$\sigma_x = \sigma(q),$$ \hspace{1cm} (7)

with $\sigma_x \in \mathbb{R}^{m_x}$ the task function. It holds:

$$\dot{\sigma_x} = J_x(q) \dot{q}$$ \hspace{1cm} (8)

$$\ddot{\sigma_x} = J_x(q) \ddot{q} + \dot{J}_x(q, \dot{q}) \dot{q}$$ \hspace{1cm} (9)

in which the matrix $J_x(q) \in \mathbb{R}^{m_x \times n}$ is lower rectangular, i.e., with more columns than rows.

Let us recall the equations of motion in the task space. It holds:

$$\tau = J_x^T(q) f_x$$ \hspace{1cm} (10)

where $f_x \in \mathbb{R}^{m_x}$ is the task generalized force vector. The joint accelerations can then be found as

$$\ddot{q} = M^{-1}(q) \left[ J_x^T(q) f_x - n(q, \dot{q}) \right]$$ \hspace{1cm} (11)

that, substituted in (9), gives

$$\ddot{\sigma_x} = J_x(q) M^{-1}(q) J_x^T(q) f_x + \dot{J}_x(q) M^{-1}(q) n(q, \dot{q}) + \dot{J}_x(q, \dot{q}) \dot{q}.$$ \hspace{1cm} (12)

By defining

$$M_x(q) = \left( J_x(q) M^{-1}(q) J_x^T(q) \right)^{-1} \in \mathbb{R}^{m_x \times m_x}$$ \hspace{1cm} (13)

and

$$c_x(q, \dot{q}) = M_x(q) \left( J_x(q) M^{-1}(q) C(q, \dot{q}) \dot{q} - \dot{J}_x(q, \dot{q}) \dot{q} \right)$$

$$g_x(q) = M_x(q) J_x(q) M^{-1}(q) g(q)$$

the task space dynamics is described by:

$$M_x(q) \ddot{\sigma_x} + c_x(q, \dot{q}) + g_x(q) = f_x.$$ \hspace{1cm} (14)
Let us further define
\[ \mathbf{J}_x(q) = M^{-1}(q)J_x^T(q)M_x(q) = M^{-1}(q)J_x^T(q)J_x(q)M^{-1}(q)J_x^T(q) \in \mathbb{R}^{n \times m_x}, \]
where, being \( n > m \), eq. (21) holds for any weight matrix used to compute \( \mathbf{J}_x(q) \) no matter the value of the inertia matrix used. More precisely, the weighted pseudoinverse [1]. It holds
\[ \mathbf{J}_x(q)\mathbf{J}_x(q) = \mathbf{I}_{m_x}. \]
The relationship (10) is invertible in
\[ \mathbf{f}_x = \mathbf{J}_x^T(q)\tau \]
where, being \( n > m \), the null space of \( \mathbf{J}_x(q) \) needs to be taken into account and a more general joint torque \( \tau \) corresponding to the task space force \( \mathbf{f}_x \) is
\[ \tau = \mathbf{J}_x^T(q)\mathbf{f}_x + \mathbf{N}_x(q)\tau_0 \]
in which \( \tau_0 \) is a vector of arbitrary torques and the null-space projector is, e.g.,
\[ \mathbf{N}_x(q) = \mathbf{I}_n - \mathbf{J}_x^T(q)\mathbf{J}_x(q), \]
that gives
\[ \mathbf{N}_x(q)\mathbf{J}_x(q) = \mathbf{J}_x^T(q) - \mathbf{J}_x(q) = \mathbf{O}_{m_x \times n}. \]
Remarkably, from (13) and (15), by direct substitution it also results that
\[ \mathbf{N}_x(q)\mathbf{J}_x(q) = \mathbf{J}_x^T(q) = \mathbf{O}_{n \times m_x}, \]
no matter the value of the inertia matrix used. More precisely, eq. (21) holds for any weight matrix used to compute \( \mathbf{J}_x(q) \) in \( \mathbf{N}_x(q) \).

In the following, dependencies will be omitted to improve readability.

\section*{D. Single task operational space control}
In [1], a seminal approach known as “operational space control” is proposed. By assuming that the subscript \( d \) means “desired” and the symbol \( \cdot \) denotes the error, by choosing
\[ \mathbf{f}_x = \mathbf{M}_x(\mathbf{\dot{\sigma}}_{x,d} + \mathbf{K}_v\mathbf{\dot{\sigma}}_x + \mathbf{K}_p\mathbf{\dot{\sigma}}_x) + \mathbf{c}_x + \mathbf{g}_x \]
in (10), with \( \mathbf{K}_v > \mathbf{O} \) and \( \mathbf{K}_p > \mathbf{O} \) design gains of proper dimensions, one obtains:
\[ \mathbf{M}_x\mathbf{\ddot{\sigma}}_x = \mathbf{M}_x(\mathbf{\dot{\sigma}}_{x,d} + \mathbf{K}_v\mathbf{\dot{\sigma}}_x + \mathbf{K}_p\mathbf{\dot{\sigma}}_x) + \mathbf{c}_x \]
and thus a second-order error dynamics:
\[ \mathbf{\ddot{\sigma}}_x + \mathbf{K}_v\mathbf{\dot{\sigma}}_x + \mathbf{K}_p\mathbf{\dot{\sigma}}_x = \mathbf{0}. \]
The controller at torque level is [1]:
\[ \tau = \mathbf{J}_x^T\mathbf{M}_x(\mathbf{\dot{\sigma}}_{x,d} + \mathbf{K}_v\mathbf{\dot{\sigma}}_x + \mathbf{K}_p\mathbf{\dot{\sigma}}_x) + \mathbf{c}'_x + \mathbf{g}'_x \]
in which \( \mathbf{g}'_x \) embeds gravity and the term \( \mathbf{c}'_x \) embeds the velocity terms. A damping term in the joint and task spaces are then necessary for the redundant case to achieve asymptotic stability [1].

It is worth noticing that also the compensation of the Coriolis and centripetal and gravity terms, i.e., the vector \( \mathbf{C}\dot{q} \) in eq. (4), may be also achieved more simply at torque level [6], [14].

\section*{E. Task priority operational space control}
In case of two tasks denoted as \( a \) and \( b \) the applied torque is [2], [3]:
\[ \tau = \tau_a + \mathbf{N}_a\tau_b \]
where both \( \tau_a \) and \( \tau_b \) are given by (25) properly modifying the subscripts and the null-space projector is given by (19).

To better appreciate the control laws, it is useful to rewrite the task dynamics of the generic task \( x \) by properly projecting it as
\[ \mathbf{J}_x^T[M\dot{q} + n = \tau] \]
\[ \downarrow \]
\[ \mathbf{M}_a\dot{\sigma}_a + \mathbf{c}_a + \mathbf{g}_a = \mathbf{J}_a^T\tau_a \]
while in the \( b \) task by
\[ \mathbf{M}_b\dot{\sigma}_b + \mathbf{c}_b + \mathbf{g}_b = \mathbf{J}_b^T(\tau_a + \mathbf{N}_a\tau_b). \]

Extensions to \( n \) tasks can be found in [2].

The work [5] proposes a formulation for redundant robots for several control laws. For two tasks the expression sounds like:
\[ \tau = \tau_b + \tau_a(\tau_b) \]
in which \( \tau_b \) is the action of the lower priority task and the higher priority ones are achieved by cancelling the lower priority ones.

Other task-priority ID-based approaches have been proposed in, e.g., [11], [12].

\section*{F. Stability analysis}
While the stability of the primary task is deeply discussed for all the techniques above, the same is not true for the lower priority tasks, even for the simple two-task case. As noticed in, e.g., [6], [7], until recently no analytic discussion existed. The stability of all the tasks for a generic number of them and for 4 different velocity-based IK algorithms has been solved in [15], [16], but only at kinematic level. In [17] it has been extended to the set-based case.

In [3] there are some reasonable considerations on the task functions and some textual descriptions of convergence “within the corresponding null space”.

The authors of [5], even though they propose a nice unifying framework for control of redundant robots, they clearly admit that “the stability of this framework as most related approaches derivable from this framework cannot be shown conclusively but only in special cases”. They provide only necessary conditions for stability and prove that the hierarchical composition method of multiple tasks does not generate conflicts among tasks at different priorities.
IV. Modeling error analysis

The ID task-priority algorithm implemented in this paper is the one in eq. (26) with control torques for tasks a and b as in eq. (25) and here explicitly reported by compensating the nonlinear dynamics directly in the joint space

$$\tau = J_a^T \ddot{M}_a \left( \ddot{\sigma}_{a,d} + K_{a,v} \dot{\sigma}_a + K_{a,p} \sigma_a - \dot{J}_a(q, \dot{q}) \dot{q} \right) + \ddot{\hat{N}}_a J_b^T \ddot{M}_b \left( \ddot{\sigma}_{b,d} + K_{b,v} \dot{\sigma}_b + K_{b,p} \sigma_b - \dot{J}_b(q, \dot{q}) \dot{q} \right) + \hat{n}(q, \dot{q}),$$

(32)

where all the matrices depending on the dynamic parameters are considered imprecise (hence denoted with the symbol $\hat{}$) and the matrices depending on the sole kinematic parameters are considered perfectly known. Notice that the latter assumption reflects the state of the art in modeling knowledge, i.e., kinematic parameters are known with accuracy order of magnitudes larger than dynamic ones.

In addition, it is worth noticing the role of the inertial parameters. In a model-based controller those parameters affect the mass matrix and the Coriolis term, their effect, thus, vanishes at steady state, when $\ddot{q} = \dot{q} = 0$. For ID task priority algorithms, on the other hand, due to their use in the dynamically-consistent pseudo-inverse, the parameters potentially affect the steady state too in case of non-null errors. At steady state and assuming the higher priority task converged to zero, eq. (32) becomes:

$$\tau = \ddot{\hat{N}}_a \tau_b + \dot{\tilde{g}}(q).$$

(33)

However, by decomposing $\tau_b$ in the component lying in the image of $J_b^T$ and its orthogonal complement, i.e., $\tau_b = \tau_{b,\|} + \tau_{b,\perp}$, and by resorting to the property (21) it holds

$$\tau = \ddot{\hat{N}}_a \left( \tau_{b,\|} + \tau_{b,\perp} \right) + \dot{\tilde{g}}(q) = \ddot{\hat{N}}_a \tau_{b,\perp} + \dot{\tilde{g}}(q).$$

(34)

By further projecting this term within the higher priority task dynamics according to the (27), one finally notice that the term in $\tau_{b,\perp}$ also disappears which allows to conclude that the modeling errors affect the priority at steady state only for what the gravity terms are concerned, i.e., the first moment of inertia, and not with the inertia terms despite their presence in $\ddot{\hat{N}}_a$.

The conclusion is thus that modeling errors do not distort the projections as long as the mass still is positive definite while, however, they obviously affect the transient. All the steady state experimental errors remarked, e.g., in [6], [7], [10] are thus to be attributed, beyond the digitization issues discussed, to the term $\dot{\tilde{g}}$, i.e., the gravity compensation. It is worth recalling, in fact, that the controller is not endowed with an integral action.

In [6], [7] the secondary task is an optimization achieved via a gradient operation of a proper functional. This means that the secondary task control input is always non-null, except in a local minimum, and thus it always projects a value on the null of the higher-priority task. This case will be numerically simulated in the next section by defining a secondary task in conflict with the primary one reaching, thus, a non-null steady state error. The considerations above are confirmed.

V. Numerical case study

Numerical dynamic simulations of the ID control law in (32) have been run by considering a KUKA LBR III characterized by the Denavit-Hartenberg parameters reported in Table I.

Two case studies have been analyzed, with compatible and conflicting tasks respectively. For the first one the following task hierarchy has been chosen:

a) $\sigma_a = p_e(q)$

b) $\sigma_b = e_e(q)$,

where $e_e(q)$ and $p_e(q)$ are the vector part of the quaternion and the position vector expressing end-effector orientation and position with respect to the base frame, respectively. Note that the orientation the error is not computed as an algebraic difference but as the vector part of the quaternion expressing the orientation error, correspondingly, the time derivative is replaced by the angular velocity error as in the resolved acceleration algorithm in [18]. The gains of the ID control law, digitally implemented with a sampling time of 1 ms, are as follows:

$$K_{a,v} = 100I_3, K_{a,p} = 500I_3,$$

$$K_{b,v} = 100I_3, K_{b,p} = 500I_3.$$

First of all a simulation has been carried out supposing a perfect knowledge of all the dynamic parameters, thus completely compensating both the inertial and the nonlinear terms in (5). The desired position and orientation have been chosen within the dexterous workspace of the arm, thus the tasks are compatible. Figure 1 shows that both of them reach a null steady-state error, proving the effectiveness of the used control law.

In the second simulation uncertainties have been introduced on the sole inertia matrix by considering uncertainty factors of 20% of the nominal value affecting link masses, center of gravity positions as well as inertia tensors. This choice has been made to ensure a perfect compensation of the nonlinear terms in (5) so as to leave only the effects of an uncertain inertia matrix. The results are shown in Fig. 2.

Both the tasks can be accomplished simultaneously, thus they reach a null steady-state error but the uncertainty on the inertia matrix computation introduces an higher error during

<table>
<thead>
<tr>
<th>link</th>
<th>$a_i$ [mm]</th>
<th>$\alpha_i$ [rad]</th>
<th>$d_i$ [mm]</th>
<th>$\theta_i$ [rad]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>$\pi/2$</td>
<td>310</td>
<td>$q_1$</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>$-\pi/2$</td>
<td>0</td>
<td>$q_2$</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>$-\pi/2$</td>
<td>400</td>
<td>$q_3$</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>$\pi/2$</td>
<td>0</td>
<td>$q_4$</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>$\pi/2$</td>
<td>390</td>
<td>$q_5$</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>$-\pi/2$</td>
<td>0</td>
<td>$q_6$</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$q_7$</td>
</tr>
</tbody>
</table>
the transient with respect to the first simulation. For the third simulation we consider an uncertainty factor on both the inertial and the nonlinear terms, keeping the tasks references within the dexterous workspace of the arm. Figure 3 shows that both the tasks have a non-null steady-state error due to the imperfect compensation of the terms in (5), even if they are still compatible.

In the second case study, two incompatible tasks have been chosen:

a) \( \sigma_a = d_w(q) \)

b) \( \sigma_b = p_e(q) \),

where \( p_e(q) \) is the end-effector position and \( d_w(q) \) is the distance from a vertical virtual wall placed at \(-0.5\) m along the \( x \) axis of the arm base frame. The reference position for the secondary task has been chosen beyond the virtual wall while the reference distance for the primary task has been set at \(10\) cm, thus the tasks are conflicting. Figure 4 shows a graphical representation of the starting configuration of the manipulator, the virtual wall and the desired final end-effector position. The gains of the ID control law, are as follows:

\[
K_{a,v} = 10, \quad K_{a,p} = 50, \\
K_{b,v} = 20I_3, \quad K_{b,p} = 100I_3.
\]

The first simulation has been run assuming the perfect knowledge of all the dynamic parameters, Fig. 5 shows the task errors during the simulation. When they are conflicting, the task space error on the higher priority task still converge to zero, while, of course, the error of the lower priority task reaches a non null steady-state value. For the last simulation an uncertainty factor of \(0.2\) has been introduced in the terms needed for inertia matrix computation, while the nonlinear and gravity terms are perfectly compensated. Figure 6 shows that the primary task still converges to a null error despite the uncertainty on the compensation of the inertial terms in \(\hat{N}_{ia}\), following the consideration made in Sec. IV.

Finally, it is clear that the uncertainty on the inertia matrix does affect the error on the higher priority task during the
transient, which is three orders of magnitude higher with respect to the previous simulation.

VI. CONCLUSIONS

Inspired by controversial results in the literature of ID-based task priority control algorithms, in this paper we investigated the effect of modeling uncertainty. Interesting enough, even in presence of errors in the estimates of the inertial parameters and despite their presence in the task projection, the dynamics of the higher-priority task error would always converge to the null value. This is due to the property of the decomposition of the weighted pseudo-inverse used in the controller. Future research will concern a comparison with the IK approach, it is known that evaluation in robotics is not an easy job as validation is, per-se, a research domain [19], however, further insights, including also the presence of physical interaction with the environment, seem to be necessary to appreciate the differences between the two approaches.

REFERENCES

This work was supported by the European Commission (projects ROBUST (H2020-690416), EuRoC (FP7-608849), DexROV (H2020-635491), AEROARMS (H2020-644271), EUMR (H2020-731103-2)).