Operational Space Control of Constrained and Underactuated Systems

Michael Mistry
Disney Research Pittsburgh
4720 Forbes Ave., Suite 110
Pittsburgh, Pennsylvania 15213, USA
Email: mmistry@disneyresearch.com

Ludovic Righetti
University of Southern California
Los Angeles, California, USA
Email: ludovic.righetti@a3.epfl.ch

Abstract—The operational space formulation (Khatib, 1987), applied to rigid-body manipulators, describes how to decouple task-space and null space dynamics, and write control equations that correspond only to forces at the end-effector or, alternatively, only to motion within the null space. We would like to apply this useful theory to modern humanoids and other legged systems, for manipulation or similar tasks, however these systems present additional challenges due to their underactuated floating bases and contact states that can dynamically change. In recent work, Sentis et al. derived controllers for such systems by implementing a task Jacobian projected into a space consistent with the supporting constraints and underactuation (the so called support consistent reduced Jacobian). Here, we take a new approach to derive operational space controllers for constrained underactuated systems, by first considering the operational space dynamics within projected inverse-dynamics (Aghili, 2005), and subsequently resolving underactuation through the addition of dynamically consistent control torques. Doing so results in a simplified control solution compared with previous results, and importantly yields several new insights into the underlying problem of operational space control in constrained environments: 1) Underactuated systems, such as humanoid robots, cannot in general completely decouple task and null space dynamics. However, 2) there may exist an infinite number of control solutions to realize desired task-space dynamics, and 3) these solutions involve the addition of dynamically consistent null space motion or constraint forces (or combinations of both). In light of these findings, we present several possible control solutions, with varying optimization criteria, and highlight some of their practical consequences.

I. INTRODUCTION

The dynamics of a robot must be considered in order to plan and execute fast, dexterous, and compliant motion. More specifically, the inertial and energy characteristics of the end-effector, or operational degrees-of-freedom (DOFs), are critical for the ultimate success of the task. With this in mind, Khatib [3] presented the operational space formulation, and derived end-effector dynamics for rigid-body robot manipulators. The resulting control solutions are able to compensate for (linearize) operational space dynamics while decoupling tasks from redundant null space dynamics. Any additional forces applied within the null space remain dynamically consistent with the tasks.

We would like to apply this powerful technique for analysis and control of a wider range of robotic systems than classical manipulators. As an example, modern humanoid robots are typically complex, high DOF systems, where task dynamics need to be considered for manipulation, balance, locomotion, etc. Adding to their complexity, humanoid systems are generally both underactuated and constrained. The complete representation of their dynamics must consider the 6 DOF floating base link connected passively to an inertial reference frame. Additionally, contacts with the environment (e.g., feet, hands, etc.) input constraint forces into the system. We would like to fully understand how underactuation and constraints interact with the dynamics of the tasks we need to control.

Several contributions have addressed operational space control within constrained environments. Khatib [3] considered constraints applied at the end-effector as an additional task to be controlled, for example to realize a desired contact force. Russakow et al. [9] examined systems with more complex kinematic structures including internal closed kinematic chains. de Sapiro and Khatib [2] took a more general approach that included systems with holonomic constraints, and applied it to the operational space control of a parallel mechanism. Sentis et al. [10, 11, 13] developed operational space controllers to specifically address the underactuation and constrained dynamics of humanoid robots. Further works by Park, Sentis, and Khatib [6, 12] additionally consider the control of contact forces for humanoids.

A common theme among the above works is that the controllers are designed to remain dynamically consistent with the constraints (i.e., the controllers contribute no acceleration at the constraint locations). However, because constraints, by definition, are able to apply necessary forces to maintain their own consistency, controllers do not need to address the dynamic consistency explicitly. As a consequence, many of the above controllers can be significantly simplified by using orthogonal projections in place of the oblique projections containing the inertia matrix. For example, Aghili [1] was able to formulate both the direct and inverse dynamics of constrained systems using orthogonal projections derived from only kinematic parameters, which in practice, are easier to accurately determine than inertial parameters [5]. Mistry et al. [4] was also able to apply orthogonal projections for the inverse dynamics control of legged robots, without requiring measurement or estimation of contact forces, or oblique pro-

1 Unless control of a constraint force is considered part of the task.
jects involving inertia matrices.

One of the objectives of this paper is examine the role of orthogonal projections on constrained operational space dynamics. Working directly with the equations of projected inverse-dynamics [1], we are able to formulate operational space dynamics without consideration of constraint forces or constraint dynamic consistency, leading to a much simplified and manageable solution that retains a similar form to the original unconstrained formulation. Additionally we examine the consequences of passive DOFs and underactuation. Similarly as Aghili [1] used additional constraint forces to influence torque at passive joints, we use dynamically consistent null space motion to compensate for passive degrees of freedom. We attempt to formulate all our results in a unified manner, keeping the structure consistent regardless if the system is constrained or unconstrained, fully actuated or underactuated. We also present examples using a simple but insightful 3 DOF simulated arm, containing both constraints and underactuation.

II. OPERATIONAL SPACE DYNAMICS

We assume the robot with n degrees-of-freedom (DOFs) is represented by the configuration vector \( \mathbf{q} \in \mathbb{R}^n \). Additionally, the robot may have \( k \) linearly independent constraints such that locations in constraint undergo zero acceleration. We write this condition as \( \ddot{x}_C = J_C \dot{\mathbf{q}} + J_C \ddot{\mathbf{q}} = 0 \), where \( J_C \) is the constraint Jacobian. Note that constraints of this form can either be holonomic or nonholonomic [1]. The robot may also be underactuated, in which case we have \( p \) active joints and \( l = n - p \) passive joints. We write the rigid-body inverse-dynamics equation as:

\[
\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{B}\tau + J^T_C(\mathbf{q})\lambda
\]

where \( \mathbf{M}(\mathbf{q}) \in \mathbb{R}^{nxn} \) is the inertia matrix, \( \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^n \) is the vector of centripetal, Coriolis, and gravity forces, \( \tau \in \mathbb{R}^n \) is the vector of joint torques, \( \lambda \in \mathbb{R}^k \) is the vector of \( k \) linearly independent constraint forces, and

\[
\mathbf{B} = \begin{bmatrix} \mathbf{I}_p & 0 \\ 0 & 0 \end{bmatrix}
\]

is the projector into actuated joint space (\( \mathbf{I}_p \) is the p dimensional identity matrix). We assume the first \( p \) rows of \( \mathbf{I}_p \) represent the actuated DOFs.

We also have an \( m \) DOF task \( (m \leq n - k) \), e.g. control of an end-effector, represented by the operational space configuration vector \( \mathbf{x} \in \mathbb{R}^m \), and the task Jacobian \( \mathbf{J} \), defined by the relation \( \ddot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \). To summarize the contributions of Khatib [3] for unconstrained \( (k = 0) \) and fully actuated \( (l = 0) \) systems (e.g. a traditional robot manipulator), the dynamics in the operational space are derived as:

\[
\Lambda\ddot{\mathbf{x}} + \Lambda(\mathbf{JM}^{-1}\mathbf{h} - \dot{\mathbf{J}}\dot{\mathbf{q}}) = \mathbf{F},
\]

where \( \Lambda = (\mathbf{JM}^{-1}\mathbf{J}^T)^{-1} \) and \( \mathbf{F} \) is an external force applied at the end-effector. Then Khatib formulated the operational space control equation, for redundant manipulators \( (m < n) \), as:

\[
\tau = \mathbf{J}^T\mathbf{F} + \left(\mathbf{I} - \mathbf{J}^T\mathbf{J}^T\right)\tau_0
\]

where \( \mathbf{F} \) is defined by (3) using a desired task acceleration \( \ddot{\mathbf{x}}_{\text{des}} \) in place of \( \dot{\mathbf{x}} \), and \( \mathbf{J}^T\tau_0 \) is the following generalized inverse of \( \mathbf{J}^T \):

\[
\mathbf{J}^T\tau_0 = \left(\mathbf{JM}^{-1}\mathbf{J}^T\right)^{-1}\mathbf{JM}^{-1}
\]

As discussed by Khatib, this generalized inverse is defined to be dynamically consistent with the task: it is the only generalized inverse that results in zero end-effector acceleration for any \( \tau_0 \). This inverse also solves the equation \( \dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}} \) for the joint velocities that minimize the instantaneous kinetic energy of the system. By using (4) we are able to compensate for task space dynamics, such that \( \dot{\mathbf{x}} = \dot{\mathbf{x}}_{\text{des}} \), while decoupling motion generated by \( \tau_0 \) from affecting task-space dynamics.

A. Constraints

Now we extend the operational space formulation to constrained systems using orthogonal projections. Constraints are defined as in (1), and can include systems with closed kinematic chains, such as parallel mechanisms or humanoid/legged robots with two or more feet in contact with the environment. At first we address fully actuated systems, and later extend to the underactuated case. As outlined by Aghili [1], we can project (1) into the null space of the constraints, eliminating the constraint forces from the dynamics equation, and write the equation of projected inverse dynamics:

\[
\mathbf{P}\mathbf{M}\dot{\mathbf{q}} + \mathbf{Ph} = \mathbf{P}\tau
\]

where \( \mathbf{P} \) is an orthogonal projection operator, such that \( \mathbf{P}\mathbf{J}^T = 0 \) and \( \mathbf{P} = \mathbf{P}^2 = \mathbf{P}^T \). Also, \( \mathbf{P} \) is readily computable from the constraint Jacobian: \( \mathbf{P} = \mathbf{I} - \mathbf{J}_C^T\mathbf{J}_C \) (where \( + \) indicates the Moore-Penrose pseudoinverse). Note that \( \mathbf{P} \) depends only on kinematic parameters and unlike other approaches that attempt to maintain dynamic consistency with constraints, it does not require the inertia matrix.

Next, we wish to invert \( \mathbf{PM} \) to solve for \( \dot{\mathbf{q}} \), however because \( \mathbf{P} \) is generally rank deficient, this term will not be invertible. However, because our system is constrained, we have the following additional equations:

\[
(\mathbf{I} - \mathbf{P})\dot{\mathbf{q}} = 0
\]

\[
(\mathbf{I} - \mathbf{P})\ddot{\mathbf{q}} = \mathbf{C}\ddot{\mathbf{q}}
\]

where \( \mathbf{C} \) is \( (d/(dt))\mathbf{P} \), e.g. \( \mathbf{C} = -\mathbf{J}_C^T\mathbf{J}_C \). Employing a trick used by Aghili, recognizing that (8) is orthogonal to (6), we can add these two equations:

\[
\mathbf{M}\ddot{\mathbf{q}} + \mathbf{Ph} - \mathbf{C}\ddot{\mathbf{q}} = \mathbf{P}\tau,
\]

defining \( \mathbf{M}_c = \mathbf{PM} + \mathbf{I} - \mathbf{P} \). Note, as discussed by Aghili, the choice of \( \mathbf{M}_c \) for a given \( \mathbf{q} \) is not unique, but is always
invertible (provided \( \mathbf{M} \) is invertible). Next, we multiply by \( \mathbf{J}^{-1} \), and replace \( \mathbf{J} \dot{\mathbf{q}} \) with \( \ddot{x} - \mathbf{J} \dot{\mathbf{q}} \):

\[
\ddot{x} - \mathbf{J} \dot{\mathbf{q}} + \mathbf{J}^{-1} (\mathbf{P} \dot{\mathbf{q}} - \mathbf{C} \ddot{\mathbf{q}}) = \mathbf{J}^{-1} \mathbf{P} \tau
\]

A force at the end-effector \( \mathbf{F} \), is mapped into joint torques via:

\[
\tau = \mathbf{J}^T \mathbf{F}
\]

and therefore we derive the constrained operational space dynamics as:

\[
\mathbf{A}_c \ddot{x} + \mathbf{A}_c \left( \mathbf{J} \mathbf{M}_c^{-1} \mathbf{P} \dot{\mathbf{q}} - \left( \mathbf{J} + \mathbf{J} \mathbf{M}_c^{-1} \mathbf{C} \right) \ddot{\mathbf{q}} \right) = \mathbf{F}
\]

\[\text{(11)}\]

where \( \mathbf{A}_c = (\mathbf{J} \mathbf{M}_c^{-1} \mathbf{P} \mathbf{J}^T)^{-1} \). Although \( \mathbf{M}_c^{-1} \mathbf{P} \) is unique, the dynamically consistent inverse is also unique. Additionally, this generalized inverse minimizes the instantaneous kinetic energy in the constrained space (see Appendix B). For notational simplicity, we define \( \mathbf{N} = \mathbf{I} - \mathbf{J}^T \mathbf{J}^{T*} \) and write the control equation as:

\[
\tau = \mathbf{J}^T \mathbf{F} + \mathbf{N} \tau_0
\]

\[\text{(13)}\]

\[\text{C. Reducing control torques}\]

Equation \( \text{(13)} \) is a joint space controller that emulates a force \( \mathbf{F} \), as if it were externally applied to the end-effector. This controller replicates both the joint motion and constraint forces that would result from the external force. However, constraint forces generate no robot motion, and ultimately no acceleration of the end-effector. Thus we can also consider a control equation that produces the joint acceleration we desire without adding extraneous constraint force. We rewrite the control equation as a sum of three components:

\[
\tau = \mathbf{P} \mathbf{J}^T \mathbf{F} + \mathbf{P} \mathbf{N} \tau_0 + (\mathbf{I} - \mathbf{P}) \tau_C
\]

\[\text{(14)}\]

where \( \mathbf{P} \mathbf{J}^T \mathbf{F} \) generates end-effector acceleration, \( \mathbf{P} \mathbf{N} \tau_0 \) generates acceleration only in the null space (with zero end-effector acceleration), and \( (\mathbf{I} - \mathbf{P}) \tau_C \) produces only constraint forces and no joint space motion. If we are not concerned about the constraint forces we generate, we can reduce the norm of our input torque by setting \( \tau_C = 0 \). This controller will produce motion identical to \( \text{(13)} \), but with different constraint forces and a smaller total control input (since \( \mathbf{P} \) is an orthogonal projection, we can verify that \( \|\mathbf{P} \tau\| \leq \|\tau\|, \forall \tau \)).

\[\text{III. UNDERACTUATION}\]

When the system contains passive degrees of freedom (\( l > 0 \)), we may no longer be able to generate desired end-effector forces using a direct mapping to joint torques via a Jacobian transpose. The torques we can generate are limited by the following constraint which always must be satisfied:

\[
\tau = \mathbf{B} \tau.
\]

\[\text{(15)}\]

Although in general \( \mathbf{J}^T \mathbf{F} \neq \mathbf{B}^T \mathbf{F} \), we may still be able to satisfy \( \text{(15)} \) by adding a null space component. When doing so, we can still guaranteed that the additional null space torque will not affect the task-space dynamics. Note, however, because we cannot compensate for task-space dynamics without the addition of null space torques, we are no longer able to decouple task and null space dynamics. Applying \( \text{(13)} \) to \( \text{(15)} \) results in:

\[
\mathbf{J}^T \mathbf{F} + \mathbf{N} \tau_0 = \mathbf{B}^T \mathbf{F} + \mathbf{B} \mathbf{N} \tau_0.
\]

\[\text{(16)}\]

and we can solve for \( \tau_0 \):

\[
(\mathbf{I} - \mathbf{B}) \mathbf{J}^T \mathbf{F} = - (\mathbf{I} - \mathbf{B}) \mathbf{N} \tau_0
\]

\[\text{(17)}\]

\[
\tau_0 = - (\mathbf{I} - \mathbf{B}) \mathbf{N} \tau_0
\]

\[\text{(18)}\]

\[\text{again using the Moore-Penrose pseudoinverse. Provided \( \text{(17)} \) has at least one valid solution for \( \tau_0 \), we can use \( \text{(18)} \) in \( \text{(15)} \) and write the control equation as:}

\[
\tau = \left( \mathbf{I} - \mathbf{N} \left[ (\mathbf{I} - \mathbf{B}) \mathbf{N} \right]^+ (\mathbf{I} - \mathbf{B}) \right) \mathbf{J}^T \mathbf{F}.
\]

\[\text{(19)}\]

\[\text{Also, because \( (\mathbf{I} - \mathbf{B}) \) is an orthogonal projection, the equation simplifies to (see Appendix C):}

\[
\tau = \left( \mathbf{I} - \mathbf{N} \left[ (\mathbf{I} - \mathbf{B}) \mathbf{N} \right]^+ \right) \mathbf{J}^T \mathbf{F}.
\]

\[\text{(20)}\]

\[\text{If the system has sufficient redundancy, there may be an infinite number of possible control solutions. However, by using the Moore-Penrose pseudoinverse in \( \text{(18)} \), we are computing the minimum possible \( \|\tau_0\| \). Thus \( \text{(20)} \) represents the operational space control solution with the minimum possible null space effect. We consider other possible solutions in the following sections.}

\[\text{Note that \( \text{(20)} \) is a general equation that can be applied to both unconstrained and constrained systems (by using either \( \text{(3)} \) or \( \text{(11,12)} \), respectively). In the unconstrained case, the controller generates dynamically consistent null space motion in order to compensate for lost torque at the passive joints. However, in the constrained case, the controller uses a combination of both motion and constraint forces to generate torque at the passive joints.}

\[\text{A. Multiple tasks}\]

\[\text{If we have more than one task, we can address under-}

\[\text{actuation using dynamically consistent torque as above, but the torque must be dynamically consistent with all tasks. As an example, consider two tasks with task 1 having a higher }

\[\text{priority than task 2. We define the augmented Jacobian of all tasks as}

\[
\mathbf{J}_a = \left[ \begin{array}{c} \mathbf{J}_1^T \mathbf{J}_2^T \end{array} \right],
\]

\[\text{and the null space projector that}\]

\[\text{Other choices of \( \mathbf{M}_c \) include (\( \mathbf{M} + \mathbf{P} \mathbf{M} - \mathbf{M} \)), (\( \mathbf{P} \mathbf{M} + (\mathbf{I} - \mathbf{P}) \mathbf{M} (\mathbf{I} - \mathbf{P}) \)), and (\( \mathbf{P} \mathbf{M} + \alpha (\mathbf{I} - \mathbf{P}) \)) for arbitrary scalar \( \alpha \). The definition of \( \mathbf{C} \) will change with choice of \( \mathbf{M}_c \).}
is dynamically consistent with all tasks as: \( N_a = I - J_a^T J_a^{\#} \).
Then we write the operational space control equation as:
\[
\tau = J_a^T F_1 + N_a J_a^T F_2 + N_a \tau_0.
\] (21)
Solving for \( \tau_0 \) as done previously, we can write the control solution as:
\[
\tau = \left( I - N_a [(I - B) N_a]^+ \right) (J_a^T F_1 + N_a J_a^T F_2) \] (22)

**B. Joint acceleration consistant controller**

The multi-task solution above assumes we have sufficient redundancy remaining after the assignment of all tasks. If all DOFs of the robot are accounted for in \( J_a \) (e.g. \( \text{Rank}(J_a) = n \)), then it is impossible to add any motion that will not conflict with at least one task. However, if the robot has constraints, we can still add constraint forces to resolve underactuation, without inducing any additional joint space motion. We use the equation:
\[
\tau = J_a^T F + (I - P) \tau_C,
\] (23)
where we have replaced \( N \tau_0 \) of \( \text{[13]} \) with \( (I - P) \tau_C \), the torques that induce only constraint forces and zero joint acceleration. The replacement is legitimate since \( (I - P) \) projects into a subspace of \( N \), i.e. \( (I - P) = N (I - P) \). Then we can satisfy \( \text{[15]} \) with the following control equation. This equation is derived similarly to \( \text{[19]} \), but replacing \( N \) with \( (I - P) \):
\[
\tau = \left( I - (I - P) [(I - B) (I - P)]^+ (I - B) \right) J_a^T F.
\] (24)
As both \( (I - P) \) and \( (I - B) \) are orthogonal projections, the equation simplifies to (see Appendix C):
\[
\tau = \left( I - [(I - B) (I - P)]^+ \right) J_a^T F,
\] (25)
and even further reduces to:
\[
\tau = [PB]^+ J_a^T F.
\] (26)

The above solution is significant for a number of reasons. First, it will produce identical joint acceleration as if the system were fully actuated (but constrained). The null space component, required to resolve underactuation, produces only constraint force and no additional motion. Therefore this controller is the solution that resolves underactuation with a minimal amount of kinetic energy. Secondly, the projector \([PB]^+\) is identical to the projector of constrained underactuated inverse dynamics controllers. For example, to achieve a desired joint space acceleration \( \dot{q}_{\text{des}} \), provided \( \dot{q}_{\text{des}} \) is constraint consistent, we can use the following controller independent of constraint forces \( \text{[4]} \):
\[
\tau = [PB]^+ (M \dot{q}_{\text{des}} + h).
\] (27)
Finally, \( \text{[26]} \) is mathematically identical to the operational space controller derived in works by Sentis et al. for task space motion control of humanoid and legged robots, but in a much reduced form. For example, the inertia matrix only appears in \( F \) and is not used in the projector or Jacobian as in \( \text{[13]} \). Note that this point is more thoroughly developed in \( \text{[8]} \) which specifically shows a proof of equivalence between orthogonal and oblique projector operators in the context of joint acceleration or inverse dynamics control.

**C. Motion only controller**

Since the previous controller \( \text{[26]} \) only uses constraint forces to resolve underactuation, for comparison, we can consider a controller that only uses null space motion. The controller is written as:
\[
\tau = \left( I - PN [(I - B) PN]^{+} \right) J_a^T F,
\] (28)
where we have replaced \( N \) of \( \text{[20]} \) with \( PN \). Again, this is possible since \( PN = NPN \). As this controller only uses null space motion to resolve underactuation, it attempts not to change the constraint forces generated by task dynamics.

**IV. Examples**

In order to test the validity of our controllers, as well our understanding of constrained and underactuated operational space dynamics, we aim to use the simplest possible system that still makes sense to add constraints and/or underactuation. We use a simulated 3 DOF planar arm with rotary joints. Each link of the robot simulator has the same length and the same inertial parameters. We assume zero gravity and include a small amount of viscous friction at the joints. The system also has a 1 DOF bilateral constraint which fixes the \( x \) position of the end-effector, and a 1 DOF task: the \( y \) position of the end-effector. \( \text{Fig. 1} \) (upper left box) describes the setup and initial condition. To consider underactuation, we may turn off the actuator at joint 3 (green circle in \( \text{Fig. 1} \)). The task is to regulate to a new end-effector position \( x_{\text{target}} \), and we set our desired task acceleration as:
\[
\ddot{x}_{\text{des}} = K_F (x_{\text{target}} - x) - K_D \dot{x}
\] (29)
We test the 5 constrained operational space controllers labeled as follows.

FA : The fully actuated operational space controller of \( \text{[13]} \), with \( \tau_0 = 0 \)
FAr : The fully actuated controller with reduced torque \( \text{[14]} \), with \( \tau_0 = 0, \tau_C = 0 \)
UAN : The underactuated controller that minimizes the null space contribution \( \text{[20]} \)
UACF : The underactuated controller that uses only constraint forces (and therefore minimizes instantaneous kinetic energy) \( \text{[26]} \)
UAPN : The underactuated controller that uses only null space motion. \( \text{[28]} \)

We use the same control parameters (gains) for each case.

\( \text{Fig. 1} \) shows the resulting motion and constraint forces after simulating for 2 seconds. \( \text{Fig. 2} \) shows the task regulation performance. As expected, the fully actuated controllers FA and FAr produce identical joint motion, but very different constraint forces. We computed the sum of the norm torque over 2 seconds and found that indeed, FAr reduced this sum by 30.0%. As a consequence, contact forces are increased, and even flip sign. In the underactuated case, all three controllers achieve the task in spite of a passive joint. Even with zero torque at joint 3, UACF produces identical motion to the fully actuated controllers, however with a noticeable increase in

---

**Supplementary Notes**

1. \( \text{[4]} \) Sentis et al., 2015
2. \( \text{[8]} \) Rosu et al., 2017
3. \( \text{[13]} \) Choset et al., 2005
4. \( \text{[14]} \) Roy et al., 2016
5. \( \text{[20]} \) Shah et al., 2017
6. \( \text{[26]} \) Nakanishi et al., 2002
7. \( \text{[28]} \) Nakanishi et al., 2003
Fig. 1. Comparison of 5 different controllers for the constrained 3 DOF robot. Top left figure shows the initial condition for each case and the target 1.0 meters lower (horizontal dashed line). The vertical solid line represents a bilateral constraint with red lines indicating the magnitude and direction of constraint forces. The simulation is run for 2 seconds in each case. Top two results are fully actuated, with the top right using the reduced torque controller (FAr). Although the motion of FAr is identical to FA, torque input is reduced by 30.0%, but as a consequence the constraint force profile is significantly altered.

The bottom three simulations are underactuated with a passive joint 3 (green circle). UAPN (right most) only uses motion to resolve underactuation, and therefore has a force profile closest to FA. UACF (center) uses only constraint forces to resolve underactuation, and therefore has identical motion to FA, but with significantly larger constraint forces. The UAN controller (left most), minimizes the null space torque contribution, and strikes a balance between UACF and UAPN.

constraint force profile. UAPN, which uses purely null space motion to resolve underactuation, significantly increases the kinetic energy of the system (Fig. 3), however, it maintains the most similar constraint force profile to FA of the three underactuated controllers. By minimizing null space impact, UAN seems to reach a compromise between the two: a slight increase in kinetic energy in trade for a relatively lower contact force profile.

A nice property of some of the equations we derived, is that they retain the same form whether the system is constrained or unconstrained. Thus we may be able to use them to control operational space dynamics for underactuated systems without constraints. To test, we remove the constraint from the 3 DOF arm, and instead add a secondary task to control end-effector position in the x direction. We use a hierarchical multi-task controller such that the secondary task does not interfere with the primary. We validate with the following two controllers:

FAuc : A fully actuated, multi-task controller using (21) with $\tau = 0$

UANuc An underactuated multi-task controller using (22). Joint 3 is considered as the passive joint.

The resulting simulation is shown in Fig. 4 (the underactuated robot is on the bottom), with primary and secondary task performance shown in Fig. 5. We also plot the instantaneous kinetic energy (Fig. 6) which confirms that the underactuated robot will use more energy since it adds null space motion to compensate for torque at the passive joints. Note that this controller is similar to partial feedback linearization [14], however rather than linearizing the dynamics of active or passive joints, we are linearizing the operational space dynamics.

V. CONCLUSION

In this work, we have presented a new methodology for deriving operational space controllers for constrained and underactuated systems. In doing so, we have introduced two key ideas: First, for constrained systems, we are able to take advantage of orthogonal projections, and significantly simplify the formulation by deriving operational space dynamics without explicitly considering constraint forces or constraint consistency. Second, for underactuated systems, we compensate for passive DOFs by applying additional null space forces. As
a consequence, we show task and null space dynamics can no longer be decoupled.

Previous works have addressed constraints and underactuation within the context of operational space control (most specifically for humanoid robots in [10–13]). These works have relied on logic that the solution must remain dynamically consistent with the constraints. End-effector dynamics are then derived using the support consistent reduced Jacobian [13], which maps dynamics from a reduced constraint space into task space motion. The resulting controller does not affect dynamics of points in contact, as if these points were additional operational space tasks to achieve. However, constraints, by definition, are able to supply the necessary forces to maintain their own consistency, and thus do not need to be accounted for in the controller. While the derivation presented in these works is ultimately correct, it remains overly complicated with inertia matrices appearing unnecessarily in the projection operators. As an example, if we fully expand the operational space
control equation from \cite{13} (equation (2.63) in that work), we will see the inertia matrix appear nine times. Our equation \cite{26} turns out to be mathematically equivalent\footnote{Again, please see \cite{8} for a more rigorous explanation on the equivalence of these projection operators.}, however the inertia matrix appears only three times and only in the \( F \) term. As inertia matrices are often difficult to determine accurately for real robots, including them unnecessarily can potentially introduce errors into the controller.

Our second key insight is that null space forces are needed to compensate for underactuation. These forces come in the form of constraints or null space motion, and of course do not affect task dynamics. We showed how available redundancy in the null space can allow us to define various optimization criteria, such as the minimization kinetic energy or null space effect. It may be possible to exploit this redundancy to aid in task achievement. For example, certain configurations of the robot may result in singularities for some controllers but not others. If a constraint force vector becomes directly aligned with a passive joint, then it cannot contribute any torque to it. In this case, a control solution that relies on constraint forces may become singular. However, by introducing some null space motion, we may be able to avoid this singularity.

In the future we hope to have a deeper understanding of the role of constraint forces, in the context of operational space dynamics. For example, in our simple 3 DOF arm experiment, why does the reduced torque controller \cite{14} flip the sign of constraint force? How can we better use null space motion to achieve more desirable constraint forces? Additionally, we are interested in exploring the roll of null space motion for the control of whole-body underactuated tasks. For example, in platform diving, control of the trunk (for vertical entry into the water) must be regulated during the dive via redundant motion from the rest of the body. Finally, we would like to apply the methods and insights developed here for the control of real underactuated humanoid robots, particularly when they are in environmental contact.

\section*{Appendix}

\subsection*{A. Proof of \( M_c^{-1} P = (PMP)^+ \)}

We write \( M_c \) in a form that is both symmetric and positive definite\footnote{Using this form of \( M_c \) in equations from previous sections requires setting \( C = -(I-2P)MJ_c^{-1}J_c \).}

\begin{equation}
M_c = PMP + (I - P)M(I - P),
\end{equation}

and take the singular value decomposition (SVD) of \( M_c = U\Sigma U^T \). Because \( PMP \) and \( (I - P)M(I - P) \) are both symmetric and mutually orthogonal, we can separate the singular values as:

\begin{equation}
M_c^{-1} = U\Sigma^{-1}U^T
\end{equation}

with \( PMP = U_1\Sigma_1U_1^T \) and \((I - P)M(I - P) = U_2\Sigma_2U_2^T \). Similarly for the inverse:

\begin{equation}
M_c^{-1} = U\Sigma^{-1}U^T
\end{equation}

\begin{equation}
= [ U_1 \ U_2 ] \left[ \begin{array}{cc}
\Sigma_1^{-1} & 0 \\
0 & \Sigma_2^{-1}
\end{array} \right] \left[ U_1 \ U_2 \right]^T
\end{equation}

\begin{equation}
= U_1\Sigma_1^{-1}U_1^T + U_2\Sigma_2^{-1}U_2^T
\end{equation}

with \((PMP)^+ = U_1\Sigma_1^{-1}U_1^T \) and \((I - P)M(I - P))^+ = U_2\Sigma_2^{-1}U_2^T \).

Thus we can write:

\begin{equation}
M_c^{-1} = (PMP)^+ + ((I - P)M(I - P))^+
\end{equation}

Finally, because \( P \) spans the null space of \((I - P)M(I - P)\), and likewise the null space of \((I - P)M(I - P))^+\), we can write:

\begin{equation}
M_c^{-1}P = (PMP)^+
\end{equation}

We can repeat a similar procedure for the other choices of \( M_c \), showing that \cite{37} always holds. Thus, \cite{37} is independent of choice of \( M_c \) and is unique.

\subsection*{B. Minimization of Kinetic Energy}

We prove that the following inverse kinematics solution:

\begin{equation}
\dot{q} = J^# \dot{x} = M_c^{-1}PJ^T(JM_c^{-1}PJT)^{-1}\dot{x}
\end{equation}

computes joint velocities that minimize the instantaneous kinetic energy of constrained motion:

\begin{equation}
\mathcal{T}_e = \frac{1}{2} \dot{q}^T PMP \dot{q},
\end{equation}
Using the result (37) and same SVD from Appendix A:
\[ q = M_c^{-1}P \left( J M_c^{-1} P J^T \right)^{-1} x \] (40)
\[ = U_1 \Sigma_{1}^{-1/2} \Sigma_{1}^{-1/2} U_1^T J^T \left( J U_1 \Sigma_{1}^{-1/2} \Sigma_{1}^{-1/2} U_1^T J^T \right)^{-1} x \] (41)
\[ = U_1 \Sigma_{1}^{-1/2} \left( J U_1 \Sigma_{1}^{-1/2} \right)^+ x \] (42)
and rearranging,
\[ \Sigma_{1}^{1/2} U_1^T q = \left( J U_1 \Sigma_{1}^{-1/2} \right)^+ x \] (43)
we see the RHS minimizes the quantity:
\[ q^T U_1 \Sigma_{1}^{1/2} \Sigma_{1}^{1/2} U_1^T q = q^T U_1 \Sigma_{1} U_1^T q = q^T P M P q \] (44)
and therefore minimizes (39).

C. Simplifications using orthogonal projections

We call \( \mathcal{R}(A) \) the range of the matrix \( A \). From the SVD of \( A = U \Sigma V^T \), and its pseudoinverse \( A^+ = V \Sigma^+ U^T \), we know that \( \mathcal{R}(A) = \mathcal{R} \left( A^+ \right) \) and \( \mathcal{R}(A^+) = \mathcal{R} \left( A^T \right) \). Take two projection operators \( X \) and \( Y \) and the expression:
\[ X \left[ Y X \right]^+ Y. \] Because \( \mathcal{R} \left( \left[ Y X \right]^+ \right) = \mathcal{R} \left( X^T Y^T \right) = \mathcal{R} \left( X^T \right) \), if \( X = X^T \) we can reduce our expression to \( \left[ Y X \right]^+ Y. \) By a similar argument, if \( Y = Y^T \) we can reduce it to \( X \left[ Y X \right]^+ \). If both \( X \) and \( Y \) are symmetric (orthogonal), then \( X \left[ Y X \right]^+ = \left[ Y X \right]^+ \).

REFERENCES