IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 25, 2022, accepted June 6, 2022, date of publication June 16, 2022, date of current version June 23, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3183609

Efficient Acceleration-Level Formula of Bias Acceleration Satisfying Time Precedence for Operational Space Formulation

SANG HYUN PARK[1,2], (Member, IEEE), MAOLIN JIN[2], (Senior Member, IEEE), AND SANG HOON KANG[1,3], (Member, IEEE)

[1]Department of Mechanical Engineering, Ulsan National Institute of Science and Technology (UNIST), Ulsan 44919, South Korea
[2]Korea Institute of Robotics and Technology Convergence (KIRO), Pohang 37666, South Korea
[3]Department of Physical Therapy and Rehabilitation Science, University of Maryland, Baltimore, MD 21201, USA

Corresponding author: Sang Hoon Kang (sanghkang@unist.ac.kr)

This work was supported in part by the Ministry of the Interior and Safety Research and Development Program under Grant 20014463; and in part by the Translational Research Program for Rehabilitation Robots, National Rehabilitation Center, Ministry of Health and Welfare, South Korea, under Grant NRCTR-EX21010.

ABSTRACT Operational space formulation has been used for robots to perform multi-tasks by taking advantage of dynamic consistency. Bias acceleration, which is an acceleration realized right after applying the control torque to a robot, compensates for the acceleration due to the higher priority tasks, thereby achieving the goal of the tasks with lower priority. Since the operational space formulation requires many computations during real-time control, there is a strong need to reduce the number of computations. We proposed an efficient acceleration-level bias acceleration formula while showing that the original bias acceleration formula could take 30-50% of the total computation for the operational space formulation. Moreover, we clarified the underlying time precedence of bias acceleration, which was implicitly provided in the original formula but was never clearly explained. Further, we have proved that the inefficiency of the original formula is due to the use of unnecessary full robot dynamics, the use of which is almost entirely eliminated in the proposed formula. Through extensive simulations and experiments, we have verified that the proposed formula needs 2-4% computations of the original, is independent of the Coriolis/centrifugal torque and the gravity torque, and is equivalent to the original formula. We expect that, with a deeper understanding of the bias acceleration, the proposed bias acceleration formula could promote the use of high degrees-of-freedom robots.

INDEX TERMS Bias acceleration, efficient formula, large degrees-of-freedom robot, operational space formulation.

I. INTRODUCTION

This paper concerns itself with implementing the operational space formulation (OSF) [1]–[5]. Specifically, we propose an acceleration-level bias acceleration formula (BAF), which is substantially more efficient than the original BAF [2], equivalent to the original BAF, and does not need the robot dynamics except for the joint space (JS) inertia matrix, unlike the original BAF [2]. Moreover, we discuss the time precedence of the bias acceleration [3], which needs to be well -understood for the implementation of OSF but seems not well -appreciated for some real implementations [6]. Further, we show that the significant computational effort to implement the original BAF is because of the use of unnecessary full robot dynamics, the use of which is eliminated in the proposed acceleration-level BAF.

The need to work, cooperate, assist, and interact with humans in unstructured environments implies that robots may need to have much larger degrees-of-freedom (DOFs) than the classical industrial robots [7]. The high DOFs robots could offer improved versatility by exploiting the kinematic redundancy allowing additional tasks besides the primary task.

The OSF [1]–[5] – a prominent force-based task-oriented framework for whole robot dynamics coordination and control [8] – allows the robots to perform multiple tasks in a

The associate editor coordinating the review of this manuscript and approving it for publication was Shihong Ding.

decoupled manner thanks to the dynamic consistency [9], [10] even for a robot with a large number of DOFs. Specifically, the joint control torques based on the OSF designed to achieve lower priority tasks do not affect the acceleration of higher priority tasks [6], [7]. This essential advantage allows the adoption of the OSF in wide robotic applications, including multi-arm systems [9], humanoid robots [2], [3], [5], [11], free-flying space robots [12], and underwater robots [13], even until recent years [14]–[24].

The OSF may be summarized as follows [1], [5]. First, one prioritizes desired tasks: constraints that should be satisfied, operational tasks, and postures (i.e., residual motion). Second, for each task, a prioritized Jacobian is defined by projecting the task Jacobian into the null space (NS) of all higher priority preceding tasks. The NS does not involve the motion of all the preceding tasks [25]. By utilizing the dynamically-consistent generalized inverse of the prioritized Jacobian, each task dynamics in the operational space (OS) can be obtained by projecting the robot's JS dynamics into the space associated with each task (i.e., the NS of all higher priority preceding tasks). Third, with the OS equation of motion obtained for each task, the control force of each task – which does not affect the acceleration of any of the preceding tasks with higher priority – is designed. In this process, bias acceleration is introduced to achieve the tasks by compensating for the acceleration due to the higher priority tasks [2], [3], [26]. The bias acceleration is the difference between the OS acceleration that does not consider higher priority tasks and the one that is consistent with higher priority tasks [3]. Lastly, the total torque to the robot can be obtained by simply summing all the torques obtained from the control forces of each task using the generalized torque/force relationship [1], [9] because of the torque-level decomposition that OSF provides using the property of dynamic consistency [1]–[5].

The original BAF [2] could pose a significant computational burden for real-time control, considering the original BAF needs full robot dynamics, including OS inertia matrix, Coriolis and centrifugal force, and gravity force for each task. A large number of computations needed to implement OSF have been a long-standing problem, which needs to be addressed for real-time control [6], [27]–[29]. However, it has rarely been pointed out that the original BAF [2] took a significant portion of the total computational load for implementation. The number of arithmetic computations needed to implement the original BAF could easily take up to 30-50% of the whole computation, as will be shown in this study. Thus, it is indispensable to reduce the arithmetic computation for bias acceleration. Further, for offline processing with recorded joint angle and its time derivatives, it is possible to obtain the two acceleration terms of the bias acceleration – the OS acceleration without considering higher priority tasks and the one consistent with higher priority tasks [3] – with the task Jacobian, prioritized Jacobian, and joint velocities and accelerations in a similar manner to [6]. This implies a high possibility that the bias acceleration may be obtained at acceleration-level without needing the Coriolis and centrifugal force, and gravity force. In other words, it may be possible to obtain a formula that can be used regardless of the environment, and that may be more efficient than the original one [2]. There is, however, a lack of an efficient BAF. Thus, there is a strong need for a more efficient BAF than the original.

It appears that clarifying the time precedence of the bias acceleration is still needed though the bias acceleration was introduced almost two decades ago. Although only implicitly implied in the original BAF [3], the bias acceleration is a prediction of future value that is supposed to be realized after applying the control torque designed based on the OSF [2], [3], and [26]. In [6], the bias acceleration was, however, obtained by using the backward Euler differentiation of OS velocities, which cannot, by any means, represent a future value, although the study [6] enhanced OSF by almost eliminating the requirement of the nonlinear time-varying large DOFs robot dynamics. This could be, at least partially, due to the lack of explicit explanation of the time precedence of the BAF. Thus, there is a strong need to clarify the underlying time precedence that any BAF should satisfy.

To address the unmet needs, we strive to make the following three contributions:

(1) to propose an efficient acceleration-level BAF, which reduces the computation needed for bias acceleration almost entirely thanks to the removal of the need for full dynamics, the reasons for the inefficiency of the original BAF;
(2) to clarify the underlying time precedence of bias acceleration; and
(3) to show that the proposed acceleration-level BAF meets the time precedence.

This paper is structured as follows. Section II briefly introduces the OSF. We, then, propose an efficient acceleration-level BAF in section III with an in-depth comparison of the original BAF and the proposed BAF. In section IV, simulations and experiments were performed to show the advantages of the proposed BAF and the equivalence between the proposed and the original.

II. REVIEW OF THE OPERATIONAL SPACE FORMULATION

A. ROBOT DYNAMICS IN JOINT SPACE

The JS dynamics of an n DOFs robot can be written as [1], [3], [5], and [6]

$$\mathbf{M}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{V}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) = \mathbf{\Gamma}, \tag{1}$$

where $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}} \in \Re^n$ denote the joint angle, and its first and second-time derivatives, respectively; $\mathbf{M}(\mathbf{q}) \in \Re^{n\times n}$ the inertia matrix; $\mathbf{V}(\mathbf{q}, \dot{\mathbf{q}}) \in \Re^n$ the Coriolis and centrifugal torque; $\mathbf{G}(\mathbf{q}) \in \Re^n$ the gravitational torque; and $\mathbf{\Gamma} \in \Re^n$ stands for the joint torque.

To implement OSF, one needs an estimated dynamics model of (1) as

$$\hat{\mathbf{M}}(\mathbf{q})\,\ddot{\mathbf{q}} + \hat{\mathbf{V}}(\mathbf{q}, \dot{\mathbf{q}}) + \hat{\mathbf{G}}(\mathbf{q}) = \mathbf{\Gamma}. \tag{2}$$

Hereafter, $\hat{\bullet}$ represents the estimate of $\bullet$ available. Since our concern is how to design a real-time OS robot controller using the OSF needing a full robot dynamics model, the estimated dynamics, (2), are regarded as the actual dynamics, (1), unless any distinction between the two is needed. There is always modeling error, usually degrading the control performance [6], [15], and [22]. Hereafter, for brevity, the arguments of $\mathbf{M}(\mathbf{q})$, $\mathbf{V}(\mathbf{q}, \dot{\mathbf{q}})$, $\mathbf{G}(\mathbf{q})$, $\hat{\mathbf{M}}(\mathbf{q})$, $\hat{\mathbf{V}}(\mathbf{q}, \dot{\mathbf{q}})$, and $\hat{\mathbf{G}}(\mathbf{q})$ will be omitted.

B. CONTROL OBJECTIVES

The robot needs to perform k (>1) tasks having assigned priority. The priority of the i^{th} task ($i = 1, 2, \ldots, k$) is defined to be lower than the j^{th} task if $i > j$ ($j = 1, 2, \ldots, k$). The i^{th} task is represented by a task coordinate vector $\mathbf{x}_i \in \Re^{m_i}$ with the corresponding task Jacobian

$$\mathbf{J}_i = \partial \mathbf{x}_i / \partial \mathbf{q} \quad (i = 1, 2, \ldots, k). \tag{3}$$

Here m_i denotes the required DOFs for the i^{th} task. For the i^{th} task, the control objective is to achieve

$$\ddot{\mathbf{x}}_i = \ddot{\mathbf{x}}_{i,ref}, \tag{4}$$

where $\ddot{\mathbf{x}}_{i,ref}$ represents the reference acceleration for the i^{th} task [5], [6]. Depending on what needs to be controlled for the i^{th} task (e.g., position, force, or impedance), $\ddot{\mathbf{x}}_{i,ref}$ can be designed in many different ways (for details, see [1], [5], and [6]).

C. TASK DYNAMICS IN OPERATIONAL SPACE

One can obtain OS task dynamics for the i^{th} task by projecting the JS dynamics, (2), into the associated task space. The prioritized Jacobian ($\mathbf{J}_{i|p}$) – the projection of $\mathbf{J}_i$ into the NS of all the tasks having higher priority than the i^{th} task – can be defined as [5]

$$\mathbf{J}_{i|p} = \begin{cases} \mathbf{J}_1 & (i = 1) \\ \mathbf{J}_i \mathbf{N}_i & (i = 2, 3, \ldots, k). \end{cases} \tag{5}$$

Here $\mathbf{N}_i$ denotes the dynamically-consistent NS projector [5], and can be obtained as follows:

$$\mathbf{N}_i = \mathbf{I} - \sum_{j=1}^{i-1} \left(\bar{\mathbf{J}}_{j|p} \mathbf{J}_{j|p} \right) \quad (i = 2, 3, \ldots, k), \tag{6}$$

where $\bar{\mathbf{J}}_{j|p}$ denotes the dynamically-consistent generalized inverse of $\mathbf{J}_{j|p}$ and is defined as

$$\bar{\mathbf{J}}_{j|p} = \hat{\mathbf{M}}^{-1} \mathbf{J}_{j|p}^T \hat{\mathbf{\Lambda}}_{j|p}, \tag{7}$$

with the prioritized inertia matrix [5]:

$$\hat{\mathbf{\Lambda}}_{j|p} = \left(\mathbf{J}_{j|p} \hat{\mathbf{M}}^{-1} \mathbf{J}_{j|p}^T \right)^{-1}. \tag{8}$$

Note that to obtain $\hat{\mathbf{\Lambda}}_{j|p}$, we need to have $\mathbf{J}_{j|p} \hat{\mathbf{M}}^{-1} \mathbf{J}_{j|p}^T$, which is essentially equivalent to $\hat{\mathbf{\Lambda}}_{j|p}^{-1}$, in advance [7] as shown in (8). Here, we focus on the case that $\mathbf{J}_{i|p}$ has a full rank for the i^{th} task to be fully controllable [2]. The case of $\mathbf{J}_{i|p}$ losing ranks can be derived similarly [2], [6]. $\mathbf{J}_{i|p}$ allows us to define a new velocity vector

$$\dot{\mathbf{x}}_{i|p} = \mathbf{J}_{i|p} \dot{\mathbf{q}}. \tag{9}$$

$\dot{\mathbf{x}}_{i|p}$ does not involve the motion of the previous $(i-1)$ tasks [25]. Multiplying $\bar{\mathbf{J}}_{i|p}^T$ on both sides of (2) provides us the (estimated) OS dynamics for the i^{th} task:

$$\hat{\mathbf{\Lambda}}_{i|p} \ddot{\mathbf{x}}_{i|p} + \hat{\boldsymbol{\mu}}_{i|p} + \hat{\mathbf{p}}_{i|p} = \mathbf{F}_{i|p}, \tag{10}$$

$$\hat{\boldsymbol{\mu}}_{i|p} = -\hat{\mathbf{\Lambda}}_{i|p} \dot{\mathbf{J}}_{i|p} \dot{\mathbf{q}} + \hat{\mathbf{\Lambda}}_{i|p} \mathbf{J}_{i|p} \hat{\mathbf{M}}^{-1} \hat{\mathbf{V}}, \text{ and} \tag{11}$$

$$\hat{\mathbf{p}}_{i|p} = \hat{\mathbf{\Lambda}}_{i|p} \mathbf{J}_{i|p} \hat{\mathbf{M}}^{-1} \hat{\mathbf{G}}, \tag{12}$$

where $\mathbf{F}_{i|p}$ denotes the prioritized OS force for the i^{th} task [1]–[5], and $\hat{\boldsymbol{\mu}}_{i|p}$ and $\hat{\mathbf{p}}_{i|p}$ represent the Coriolis/centrifugal force and gravity force, respectively.

D. OPERATIONAL SPACE CONTROL FORCE

Control force $\mathbf{F}_{i|p}$ for the i^{th} task can be designed using the prioritized robot dynamics in OS, (10). At the level of acceleration, the unit mass behavior of

$$\ddot{\mathbf{x}}_{i|p} = \ddot{\mathbf{x}}_{i|p,ref} \tag{13}$$

can be achieved with the following control force $\mathbf{F}_{i|p}$:

$$\mathbf{F}_{i|p} = \hat{\mathbf{\Lambda}}_{i|p} \ddot{\mathbf{x}}_{i|p,ref} + \hat{\boldsymbol{\mu}}_{i|p} + \hat{\mathbf{p}}_{i|p}, \tag{14}$$

where $\ddot{\mathbf{x}}_{i|p,ref}$ denotes the reference acceleration for $\ddot{\mathbf{x}}_{i|p}$.

E. TORQUE DECOMPOSITION

The OSF provides torque decomposition as follows:

$$\boldsymbol{\Gamma} = \boldsymbol{\Gamma}_1 + \boldsymbol{\Gamma}_2 + \ldots + \boldsymbol{\Gamma}_k, \tag{15}$$

where $\boldsymbol{\Gamma}_i$ denotes the prioritized torque for the i^{th} task. It was proved that $\boldsymbol{\Gamma}_i$ induces no acceleration of any of the $(i-1)$ tasks having higher priority than the i^{th} task [5] because $\boldsymbol{\Gamma}_i$ was designed in the NS of all preceding tasks [2] as follows:

$$\boldsymbol{\Gamma}_i = \mathbf{J}_{i|p}^T \mathbf{F}_{i|p}. \tag{16}$$

Substituting (14) into (16) provides us the prioritized control torque as follows

$$\boldsymbol{\Gamma}_i = \mathbf{J}_{i|p}^T \left(\hat{\mathbf{\Lambda}}_{i|p} \ddot{\mathbf{x}}_{i|p,ref} + \hat{\boldsymbol{\mu}}_{i|p} + \hat{\mathbf{p}}_{i|p} \right). \tag{17}$$

F. BIAS ACCELERATION

If $\hat{\mathbf{\Lambda}}_{i|p}$, $\hat{\boldsymbol{\mu}}_{i|p}$, and $\hat{\mathbf{p}}_{i|p}$ are identical to real ones, by substituting the control force (14) into (10), the resulting closed-loop dynamics becomes (13), which is different from the control objective ($\ddot{\mathbf{x}}_i = \ddot{\mathbf{x}}_{i,ref}$) given in (4). Thus, bias acceleration ($\ddot{\mathbf{x}}_{i|bias}$) was introduced to ensure that the resulting closed-loop dynamics meet the control objective [3].

The $\ddot{\mathbf{x}}_{i|p}$ as a result of applying $\boldsymbol{\Gamma}$ to the OS dynamics of i^{th} task is

$$\ddot{\mathbf{x}}_{i|p} = \hat{\mathbf{\Lambda}}_{i|p}^{-1} \mathbf{F}_{i|p} - \hat{\mathbf{\Lambda}}_{i|p}^{-1} \left(\hat{\boldsymbol{\mu}}_{i|p} + \hat{\mathbf{p}}_{i|p} \right). \tag{18}$$

On the other hand, $\ddot{\mathbf{x}}_i$ as a result of applying $\mathbf{\Gamma}$ is

$$\ddot{\mathbf{x}}_i = \mathbf{J}_i\hat{\mathbf{M}}^{-1}\sum_{j=1}^{i-1}\mathbf{\Gamma}_j + \hat{\mathbf{\Lambda}}_{i|p}^{-1}\mathbf{F}_{i|p} - \hat{\mathbf{\Lambda}}_i^{-1}\left(\hat{\boldsymbol{\mu}}_i + \hat{\mathbf{p}}_i\right), \tag{19}$$

$$\hat{\mathbf{\Lambda}}_i = \left(\mathbf{J}_i\hat{\mathbf{M}}^{-1}\mathbf{J}_i^T\right)^{-1}, \tag{20}$$

$$\hat{\boldsymbol{\mu}}_i = -\hat{\mathbf{\Lambda}}_i\dot{\mathbf{J}}_i\dot{\mathbf{q}} + \hat{\mathbf{\Lambda}}_i\mathbf{J}_i\hat{\mathbf{M}}^{-1}\hat{\mathbf{V}}, \text{ and} \tag{21}$$

$$\hat{\mathbf{p}}_i = \hat{\mathbf{\Lambda}}_i\mathbf{J}_i\hat{\mathbf{M}}^{-1}\hat{\mathbf{G}}, \tag{22}$$

where $\hat{\boldsymbol{\mu}}_i$ and $\hat{\mathbf{p}}_i$ denote Coriolis/centrifugal force and gravitational force, respectively. The proof of (19) is given in Appendix A. It is clear that the $\ddot{\mathbf{x}}_{i|p}$ in (18) and $\ddot{\mathbf{x}}_i$ in (19) are future values that can be realized after applying $\mathbf{\Gamma}$ to the robot and, thus, cannot be measured by any means during the real-time control. Further discussion is provided in section III.E.1).

From (18) and (19), the bias acceleration ($\ddot{\mathbf{x}}_{i|bias}$), defined as $\ddot{\mathbf{x}}_i - \ddot{\mathbf{x}}_{i|p}$ [3], can be obtained as

$$\ddot{\mathbf{x}}_{i|bias} = \underbrace{\mathbf{J}_i\hat{\mathbf{M}}^{-1}\sum_{j=1}^{i-1}\mathbf{\Gamma}_j}_{\substack{\text{Induced by}\\ \text{preceding}\\ \text{task torques}}} + \underbrace{\hat{\mathbf{\Lambda}}_{i|p}^{-1}\left(\hat{\boldsymbol{\mu}}_{i|p} + \hat{\mathbf{p}}_{i|p}\right) - \hat{\mathbf{\Lambda}}_i^{-1}\left(\hat{\boldsymbol{\mu}}_i + \hat{\mathbf{p}}_i\right)}_{\substack{\text{Correction for Coriolis/centrifugal}\\ \text{and gravity force}}}. \tag{23}$$

Thus, the OSF suggests using the following $\ddot{\mathbf{x}}_{i|p,ref}$ to achieve the control objective of (4) [2]:

$$\ddot{\mathbf{x}}_{i|p,ref} = \ddot{\mathbf{x}}_{i,ref} - \ddot{\mathbf{x}}_{i|bias}. \tag{24}$$

Indeed, the substitution of $\ddot{\mathbf{x}}_{i|p,ref}$ in the control force of (14) with (24) and applying the control force to the robot dynamics yield the desired closed-loop dynamics of (4) [2], [3]. Therefore, the bias acceleration is essential to achieve the control objective.

The original BAF, (23), needs a full dynamics model of the robot, including the Coriolis and centrifugal force ($\hat{\boldsymbol{\mu}}_{i|p}$ and $\hat{\boldsymbol{\mu}}_i$) and the gravity force ($\hat{\mathbf{p}}_{i|p}$ and $\hat{\mathbf{p}}_i$). Hereafter, the bias acceleration obtained from the original BAF, (23), is called $\ddot{\mathbf{x}}_{i|bias\_O}$ for brevity.

It is worth noting that $\ddot{\mathbf{x}}_{i|bias}$ is the difference between the two accelerations $\ddot{\mathbf{x}}_i$ and $\ddot{\mathbf{x}}_{i|p}$, implying a potential of obtaining $\ddot{\mathbf{x}}_{i|bias}$ at acceleration-level without a full robot dynamics model.

G. PHYSICAL MEANING OF ORIGINAL BIAS ACCELERATION FORMULA

The terms in (23) have physical meanings. The first term is the acceleration induced by the control torque for preceding tasks that have a priority higher than the i^{th} task. The second and third terms together represent the correction of acceleration induced by the Coriolis and centrifugal force and gravity force [2] (i.e., the difference between the one from the instantaneous space of the i^{th} task motion without considering the previous tasks, and another from the instantaneous space of the i^{th} task motion consistent with the previous $(i-1)$ tasks). Overall, all those terms are accelerations, implying there may be an acceleration-level formula equivalent to (23).

III. PROPOSED BIAS ACCELERATION FORMULA

We present an efficient acceleration-level BAF. The proposed acceleration-level BAF is equivalent to the original BAF, (23), does not require the dynamic model of the robot except for the JS inertia matrix, $\hat{\mathbf{M}}$, and is, hence, highly efficient and independent of Coriolis/centrifugal and gravitational torques.

A. EFFICIENT ACCELERATION-LEVEL FORMULA FOR BIAS ACCELERATION

From (5) and (6), $(\mathbf{J}_{i|p} - \mathbf{J}_i)$ becomes

$$\begin{aligned}\mathbf{J}_{i|p} - \mathbf{J}_i &= \mathbf{J}_i\left[\mathbf{I} - \sum_{j=1}^{i-1}\left(\bar{\mathbf{J}}_{j|p}\mathbf{J}_{j|p}\right)\right] - \mathbf{J}_i \\ &= -\mathbf{J}_i\sum_{j=1}^{i-1}\left(\bar{\mathbf{J}}_{j|p}\mathbf{J}_{j|p}\right).\end{aligned} \tag{25}$$

From (25), $\mathbf{J}_i$ can be written as follows

$$\mathbf{J}_i = \mathbf{J}_{i|p} + \mathbf{J}_i\sum_{j=1}^{i-1}\left(\bar{\mathbf{J}}_{j|p}\mathbf{J}_{j|p}\right). \tag{26}$$

Since $\dot{\mathbf{x}}_i = \mathbf{J}_i\dot{\mathbf{q}}$, with (26), we can obtain the following kinematic relation:

$$\ddot{\mathbf{x}}_i = \mathbf{J}_i\ddot{\mathbf{q}} + \dot{\mathbf{J}}_i\dot{\mathbf{q}} = \mathbf{J}_{i|p}\ddot{\mathbf{q}} + \mathbf{J}_i\sum_{j=1}^{i-1}\left(\bar{\mathbf{J}}_{j|p}\mathbf{J}_{j|p}\ddot{\mathbf{q}}\right) + \dot{\mathbf{J}}_i\dot{\mathbf{q}}. \tag{27}$$

From (9),

$$\ddot{\mathbf{x}}_{i|p} = \mathbf{J}_{i|p}\ddot{\mathbf{q}} + \dot{\mathbf{J}}_{i|p}\dot{\mathbf{q}}. \tag{28}$$

Since $\mathbf{J}_{i|p}\ddot{\mathbf{q}} = \ddot{\mathbf{x}}_{i|p} - \dot{\mathbf{J}}_{i|p}\dot{\mathbf{q}}$ from (28), (27) can be written as

$$\ddot{\mathbf{x}}_i = \mathbf{J}_{i|p}\ddot{\mathbf{q}} + \mathbf{J}_i\sum_{j=1}^{i-1}\bar{\mathbf{J}}_{j|p}\left(\ddot{\mathbf{x}}_{j|p} - \dot{\mathbf{J}}_{j|p}\dot{\mathbf{q}}\right) + \dot{\mathbf{J}}_i\dot{\mathbf{q}}. \tag{29}$$

Subtraction of (28) from (29) yields

$$\ddot{\mathbf{x}}_i - \ddot{\mathbf{x}}_{i|p} = \mathbf{J}_i\sum_{j=1}^{i-1}\bar{\mathbf{J}}_{j|p}\left(\ddot{\mathbf{x}}_{j|p} - \dot{\mathbf{J}}_{j|p}\dot{\mathbf{q}}\right) + \dot{\mathbf{J}}_i\dot{\mathbf{q}} - \dot{\mathbf{J}}_{i|p}\dot{\mathbf{q}}. \tag{30}$$

If $\mathbf{\Gamma}_j$ $(j < i)$ were properly designed from (17), $\ddot{\mathbf{x}}_{j|p,ref}$ is a future value that $\ddot{\mathbf{x}}_{j|p}$ will have $(\ddot{\mathbf{x}}_{j|p} = \ddot{\mathbf{x}}_{j|p,ref})$ immediately after applying the correctly designed control torque based on the OSF to the robot, as was proved in section II.D. Thus, the bias acceleration $\ddot{\mathbf{x}}_{i|bias}$, the value of $\ddot{\mathbf{x}}_i - \ddot{\mathbf{x}}_{i|p}$ right after applying the control torque, can be obtained by substituting $\ddot{\mathbf{x}}_{j|p}$ with $\ddot{\mathbf{x}}_{j|p,ref}$ in (30) as follows:

$$\ddot{\mathbf{x}}_{i|bias} = \underbrace{\mathbf{J}_i\sum_{j=1}^{i-1}\left[\bar{\mathbf{J}}_{j|p}\left(\ddot{\mathbf{x}}_{j|p,ref} - \dot{\mathbf{J}}_{j|p}\dot{\mathbf{q}}\right)\right]}_{\substack{\text{Induced by preceding}\\ \text{task accelerations}}} + \underbrace{\dot{\mathbf{J}}_i\dot{\mathbf{q}} - \dot{\mathbf{J}}_{i|p}\dot{\mathbf{q}}}_{\substack{\text{Correction for}\\ \text{velocity dependent}\\ \text{acceleration terms}}}. \tag{31}$$

Therefore, we have derived an acceleration-level BAF for $\ddot{\mathbf{x}}_{i|bias}$. The proposed acceleration-level BAF, (31), does not need $\hat{\boldsymbol{\mu}}_{i|p}$, $\hat{\boldsymbol{\mu}}_i$, $\hat{\mathbf{p}}_{i|p}$, and $\hat{\mathbf{p}}_i$, indicating $\hat{\mathbf{V}}$ and $\hat{\mathbf{G}}$ are not needed, compared with the original BAF, (23). Moreover, $\hat{\boldsymbol{\Lambda}}_i^{-1}$ and $\hat{\boldsymbol{\Lambda}}_i$ are unnecessary. For clarity, hereafter, $\ddot{\mathbf{x}}_{i|bias\_P}$ represents the bias acceleration obtained from the proposed BAF, (31).

B. EQUIVALENCE OF THE PROPOSED FORMULA AND THE ORIGINAL FORMULA

We prove the equivalence between the proposed BAF, (31), and the original BAF, (23). From (11), (12), (21), and (22), the Coriolis/centrifugal force and gravity force terms in (23), which are acting on the OS, can be expressed as

$$\hat{\boldsymbol{\mu}}_{i|p} + \hat{\mathbf{p}}_{i|p} = \hat{\boldsymbol{\Lambda}}_{i|p}\left(-\dot{\mathbf{J}}_{i|p}\dot{\mathbf{q}} + \mathbf{J}_{i|p}\hat{\mathbf{M}}^{-1}\hat{\mathbf{V}} + \mathbf{J}_{i|p}\hat{\mathbf{M}}^{-1}\hat{\mathbf{G}}\right), \text{ and} \tag{32}$$

$$\hat{\boldsymbol{\mu}}_i + \hat{\mathbf{p}}_i = \hat{\boldsymbol{\Lambda}}_i\left(-\dot{\mathbf{J}}_i\dot{\mathbf{q}} + \mathbf{J}_i\hat{\mathbf{M}}^{-1}\hat{\mathbf{V}} + \mathbf{J}_i\hat{\mathbf{M}}^{-1}\hat{\mathbf{G}}\right). \tag{33}$$

Substituting (32) and (33) into the original BAF of (23) and simple manipulation yield

$$\ddot{\mathbf{x}}_{i|bias\_O} = \mathbf{J}_i\hat{\mathbf{M}}^{-1}\sum_{j=1}^{i-1}\boldsymbol{\Gamma}_j + (\mathbf{J}_{i|p} - \mathbf{J}_i)\hat{\mathbf{M}}^{-1}(\hat{\mathbf{V}} + \hat{\mathbf{G}}) + \dot{\mathbf{J}}_i\dot{\mathbf{q}} - \dot{\mathbf{J}}_{i|p}\dot{\mathbf{q}}. \tag{34}$$

Substituting (25) into $(\mathbf{J}_{i|p} - \mathbf{J}_i)$ in the right-hand side (RHS) of (34) yields

$$\ddot{\mathbf{x}}_{i|bias\_O} = \mathbf{J}_i\hat{\mathbf{M}}^{-1}\sum_{j=1}^{i-1}\boldsymbol{\Gamma}_j - \left[\mathbf{J}_i\sum_{j=1}^{i-1}\left(\bar{\mathbf{J}}_{j|p}\mathbf{J}_{j|p}\right)\right]\hat{\mathbf{M}}^{-1}\left(\hat{\mathbf{V}} + \hat{\mathbf{G}}\right) + \dot{\mathbf{J}}_i\dot{\mathbf{q}} - \dot{\mathbf{J}}_{i|p}\dot{\mathbf{q}}. \tag{35}$$

Substituting (7) into (35) and simple manipulations provide us

$$\ddot{\mathbf{x}}_{i|bias\_O} = \mathbf{J}_i\hat{\mathbf{M}}^{-1}\sum_{j=1}^{i-1}\boldsymbol{\Gamma}_j - \mathbf{J}_i\hat{\mathbf{M}}^{-1}\sum_{j=1}^{i-1}\left[\mathbf{J}_{j|p}^T\hat{\boldsymbol{\Lambda}}_{j|p}\mathbf{J}_{j|p}\hat{\mathbf{M}}^{-1}\left(\hat{\mathbf{V}} + \hat{\mathbf{G}}\right)\right] + \dot{\mathbf{J}}_i\dot{\mathbf{q}} - \dot{\mathbf{J}}_{i|p}\dot{\mathbf{q}}. \tag{36}$$

One can rewrite (36) as follows:

$$\ddot{\mathbf{x}}_{i|bias\_O} = \mathbf{J}_i\hat{\mathbf{M}}^{-1}\sum_{j=1}^{i-1}\left[\boldsymbol{\Gamma}_j - \mathbf{J}_{j|p}^T\hat{\boldsymbol{\Lambda}}_{j|p}\mathbf{J}_{j|p}\hat{\mathbf{M}}^{-1}\left(\hat{\mathbf{V}} + \hat{\mathbf{G}}\right)\right] + \dot{\mathbf{J}}_i\dot{\mathbf{q}} - \dot{\mathbf{J}}_{i|p}\dot{\mathbf{q}}. \tag{37}$$

By substituting (11) and (12) into (17), the prioritized control torque $\boldsymbol{\Gamma}_i$ of (17) can be written as follows:

$$\boldsymbol{\Gamma}_i = \mathbf{J}_{i|p}^T\hat{\boldsymbol{\Lambda}}_{i|p}\left(\ddot{\mathbf{x}}_{i|p,ref} - \dot{\mathbf{J}}_{i|p}\dot{\mathbf{q}}\right) + \mathbf{J}_{i|p}^T\hat{\boldsymbol{\Lambda}}_{i|p}\mathbf{J}_{i|p}\hat{\mathbf{M}}^{-1}\left(\hat{\mathbf{V}} + \hat{\mathbf{G}}\right). \tag{38}$$

Substituting (38) into (37) yields

$$\ddot{\mathbf{x}}_{i|bias\_O} = \mathbf{J}_i\hat{\mathbf{M}}^{-1}\sum_{j=1}^{i-1}\left[\mathbf{J}_{j|p}^T\hat{\boldsymbol{\Lambda}}_{j|p}(\ddot{\mathbf{x}}_{j|p,ref} - \dot{\mathbf{J}}_{j|p}\dot{\mathbf{q}})\right] + \dot{\mathbf{J}}_i\dot{\mathbf{q}} - \dot{\mathbf{J}}_{i|p}\dot{\mathbf{q}}. \tag{39}$$

The control torque $\boldsymbol{\Gamma}_i$ canceled out the Coriolis/centrifugal and gravity torque terms originated from the correction for Coriolis/centrifugal and gravity force.

Considering the definition of $\bar{\mathbf{J}}_{j|p}$ in (7), one can easily obtain the proposed BAF of (31) from (39). Thus, the proposed BAF is equivalent to the original BAF (i.e., $\ddot{\mathbf{x}}_{i|bias\_P} = \ddot{\mathbf{x}}_{i|bias\_O}$).

C. PHYSICAL MEANING

As shown in the derivation of the equivalence between the original and the proposed BAFs (section III.B), the first term $(\mathbf{J}_i\hat{\mathbf{M}}^{-1}\sum_{j=1}^{i-1}\boldsymbol{\Gamma}_j)$ of the original BAF, (23) – which is induced by preceding task torques – canceled out the terms related to the Coriolis/centrifugal and gravity torque terms included in the correction for Coriolis/centrifugal and gravity forces. Consequently, the remainder of the first term of the original BAF became the first term $(\mathbf{J}_i\sum_{j=1}^{i-1}\left[\bar{\mathbf{J}}_{j|p}\left(\ddot{\mathbf{x}}_{j|p,ref} - \dot{\mathbf{J}}_{j|p}\dot{\mathbf{q}}\right)\right])$ of the proposed BAF, (31). Thus, the first term of (31) represents the acceleration induced by the acceleration of preceding tasks.

The second and third terms $(\hat{\boldsymbol{\Lambda}}_{i|p}^{-1}\left(\hat{\boldsymbol{\mu}}_{i|p} + \hat{\mathbf{p}}_{i|p}\right) - \hat{\boldsymbol{\Lambda}}_i^{-1}\left(\hat{\boldsymbol{\mu}}_i + \hat{\mathbf{p}}_i\right))$, representing the correction for Coriolis/centrifugal and gravity forces of the original BAF, (23), were mostly compensated for by the prioritized control torque. Subsequently, the remainder of the second and third terms of the original BAF became the second and third terms of the proposed BAF $(\dot{\mathbf{J}}_i\dot{\mathbf{q}} - \dot{\mathbf{J}}_{i|p}\dot{\mathbf{q}})$, respectively. More specifically, the second term $(\dot{\mathbf{J}}_i\dot{\mathbf{q}})$ originated from the acceleration induced by Coriolis/centrifugal and gravity forces in the instantaneous space of the i^{th} task motion without considering the previous tasks; and the third term $(\dot{\mathbf{J}}_{i|p}\dot{\mathbf{q}})$ originated from the acceleration induced by the Coriolis/centrifugal and gravity forces in the instantaneous space of the i^{th} task motion consistent with all the preceding tasks. Thus, the second and third terms of the proposed BAF together stand for the correction of the velocity-dependent acceleration terms induced by the Coriolis and centrifugal force, and gravity force.

D. INDEPENDENCE TO CORIOLIS/CENTRIFUGAL AND GRAVITY TORQUES

From (39), it is clear that the bias acceleration is independent of $\hat{\mathbf{V}}$ and $\hat{\mathbf{G}}$, meaning the original BAF (23) unnecessarily utilizes $\hat{\mathbf{V}}$ and $\hat{\mathbf{G}}$. It means that the computational efficiency of the original BAF may be unnecessarily larger than needed. Since we have used the estimated robot dynamics throughout the whole derivation of (23) and (39), the independence of

bias acceleration to $\hat{\mathbf{V}}$ and $\hat{\mathbf{G}}$ holds even with inevitable modeling errors.

The proposed acceleration-level BAF (31) is also independent of $\hat{\mathbf{V}}$ and $\hat{\mathbf{G}}$, and, indeed, does not require those terms. The independence mentioned above makes the proposed BAF highly efficient than the original BAF, which is a significant advantage for practical implementations (see section III.F. for details).

The independence does not mean robustness to modeling error but means that the two equivalent BAFs will be affected by the modeling error to the same degree. Strictly speaking, (18) and (19) only provide an estimate of $\ddot{\mathbf{x}}_{i|p}$ and $\ddot{\mathbf{x}}_i$, respectively, because of the use of the estimated dynamics. For real-time control purposes, however, those could be used instead of the real values of $\ddot{\mathbf{x}}_{i|p}$ and $\ddot{\mathbf{x}}_i$, which cannot be obtained without knowing the actual dynamics, (1).

E. TIME PRECEDENCE

We clarify the underlying time precedence of the BAF, which was implicitly implied in the original BAF and has never been explicitly explained. An example of incorrect implementation of bias acceleration, violating the time precedence, is also provided for clear understanding. Further, we prove that the proposed BAF satisfies the time precedence.

1) THE TIME PRECEDENCE OF THE ORIGINAL FORMULA

The time precedence that the bias acceleration is a future value that has not been realized at the moment of computing control torque using the OSF is implied in the original BAF though implicit. Specifically, it is clear from (18) and (19) that $\ddot{\mathbf{x}}_i$ and $\ddot{\mathbf{x}}_{i|p}$ are the accelerations resulting from applying the control torque, $\boldsymbol{\Gamma}$, to the robot. Thus, in (18) and (19), $\ddot{\mathbf{x}}_i$ and $\ddot{\mathbf{x}}_{i|p}$ are the accelerations realized at the very next moment $\boldsymbol{\Gamma}$ is applied to the robot. Consequently, $\ddot{\mathbf{x}}_{i|bias}$ is not an acceleration that has already been realized at the moment it is computed, but an acceleration that occurs at the very next moment $\boldsymbol{\Gamma}$ is applied to the robot. Therefore, at the moment of computing $\boldsymbol{\Gamma}$, $\ddot{\mathbf{x}}_{i|bias}$ can only be *predicted* through the formulae, including (23) and (31).

2) AN INCORRECT IMPLEMENTATION VIOLATING THE TIME PRECEDENCE

The time precedence is seemingly very obvious, but it has never been explicitly explained to the best of the authors' knowledge. At least partially, the lack of explicit explanation of the time precedence might have caused incorrect implementations such as [6] as follows. Seemingly, the idea to obtain $\ddot{\mathbf{x}}_{i|bias}$ from the kinematic relationship of

$$\dot{\mathbf{x}}_i - \dot{\mathbf{x}}_{i|p} = \left(\mathbf{J}_i - \mathbf{J}_{i|p}\right)\dot{\mathbf{q}} \tag{40}$$

by taking the time derivative of both sides of (40) instead of using the original BAF, (23), is feasible and may sound attractive due to the removal of the need for robot dynamics. This idea was implemented in [6] by taking the Euler backward numerical differentiation on the RHS of (40) as follows:

$$\begin{aligned}\ddot{\mathbf{x}}_{i|bias\_N}(t) &= \left\{\left[\mathbf{J}_i(t) - \mathbf{J}_{i|p}(t)\right]\dot{\mathbf{q}}(t)\right. \\ &\quad \left. - \left[\mathbf{J}_i(t-\Delta t) - \mathbf{J}_{i|p}(t-\Delta t)\right]\dot{\mathbf{q}}(t-\Delta t)\right\}/\Delta t,\end{aligned} \tag{41}$$

where $\ddot{\mathbf{x}}_{i|bias\_N}$ denotes the bias acceleration obtained from the Euler backward numerical differentiation, and Δt represents sampling time. $\dot{\mathbf{q}}$ was also obtained by taking the Euler backward numerical differentiation of measured $\mathbf{q}$. One may, however, notice that this idea is not right. To obtain $\ddot{\mathbf{x}}_{i|bias}$, again, the future values of $\ddot{\mathbf{x}}_i$ and $\ddot{\mathbf{x}}_{i|p}$ are needed. Thus, to truly utilize (40), we need the future value of $\dot{\mathbf{q}}$. However, during the real-time control of a robot, it is impossible to measure the future value of $\dot{\mathbf{q}}$ physically. Thus, for the case of [6], at best, the time-delayed value of $\ddot{\mathbf{x}}_{i|bias}$ might be obtained from (41). Further, the numerical differentiation can amplify accompanied noise and may significantly deteriorate the control performances. The acceleration signal obtained from the numerical differentiation may result in a large error and large and fluctuating control torque. In short, the idea of determining $\ddot{\mathbf{x}}_{i|bias}$ by taking the time derivative of (40) with current and past values of $\ddot{\mathbf{q}}$ and $\dot{\mathbf{q}}$ is neither suitable nor desirable for the OSF considering the underlying time precedence.

3) THE TIME PRECEDENCE OF THE PROPOSED FORMULA

In the process of deriving the proposed BAF, it was found that the bias acceleration $\ddot{\mathbf{x}}_{i|bias\_P}$, the value of $\ddot{\mathbf{x}}_i - \ddot{\mathbf{x}}_{i|p}$ right after applying the control torque, can be obtained by substituting $\ddot{\mathbf{x}}_{j|p}$ with $\ddot{\mathbf{x}}_{j|p,ref}$, the future value of $\ddot{\mathbf{x}}_{j|p}$, in an acceleration-level kinematic relation, (30), which always holds.

It implies that $\ddot{\mathbf{x}}_{i|bias\_P}$ is the future value realized immediately after applying the control torque to the robot, and it is a prediction and cannot be measured at the time of computing $\ddot{\mathbf{x}}_{i|bias\_P}$ to determine the control torque using (17) and (24). Therefore, the proposed BAF satisfies the time precedence that the original BAF met.

F. COMPARISON OF COMPUTATIONAL EFFICIENCY

Since the OSF was developed for the real-time control of robots with a large number of DOFs, the large number of computations is a problem for practical implementations [6], [27]–[29]. The computational efforts for the OSF using the original BAF and that using the proposed BAF were examined in terms of the number of DOFs of a robot (n), the DOFs required for a task (m), and the number of tasks (k) by analyzing the FLoating point Operations (FLOPs), representing the total number of four arithmetic operations [30]. We have assumed that each of the tasks needs m DOFs as [6], and all terms ($\mathbf{J}_{i|p}$, $\hat{\boldsymbol{\mu}}_{i|p} + \hat{\mathbf{p}}_{i|p}$, $\hat{\boldsymbol{\Lambda}}_{i|p}$, $\ddot{\mathbf{x}}_{i,ref}$) for computing the prioritized control torque, (17), are recycled for $\ddot{\mathbf{x}}_{i|bias}$.

TABLE 1. Computational effort to evaluate bias acceleration.

Calculation	Iteration	Computational Effort (FLOPs)	
		Original formula ($FLOP_{bias\_o}$)	Proposed formula ($FLOP_{bias\_p}$)
$\boldsymbol{\Lambda}_i$ [a]	k-1	$(1/3)m^3$	
$\boldsymbol{\Lambda}_i^{-1}$	k-1	$2mn^2 + m^2n - 0.5m^2 - 0.5m$	
$\dot{\mathbf{J}}_i\dot{\mathbf{q}}$	k-1	$4mn - m$	$4mn - m$
$\boldsymbol{\mu}_i + \mathbf{p}_i$	k-1	$2mn^2 + mn + 2m^2 - m$	
$\mathbf{J}_i\mathbf{M}^{-1}\sum_{j=1}^{i-1}\boldsymbol{\Gamma}_j$	k-1	$2mn^2 + mn - m$	
$\ddot{\mathbf{x}}_{i,bias\_O} = \mathbf{J}_i\mathbf{M}^{-1}\sum_{j=1}^{i-1}\boldsymbol{\Gamma}_j + \boldsymbol{\Lambda}_{i\|p}^{-1}\left(\boldsymbol{\mu}_{i\|p} + \mathbf{p}_{i\|p}\right) - \boldsymbol{\Lambda}_i^{-1}\left(\boldsymbol{\mu}_i + \mathbf{p}_i\right)$ [b]	k-1	$4m^2$	
$\mathbf{J}_i\sum_{j=1}^{i-1}\left[\bar{\mathbf{J}}_{j\|p}\left(\ddot{\mathbf{x}}_{j\|p,ref} - \dot{\mathbf{J}}_{j\|p}\dot{\mathbf{q}}\right)\right]$	k-1		$4mn + m$
$\ddot{\mathbf{x}}_{i,bias\_P} = \mathbf{J}_i\sum_{j=1}^{i-1}\left[\bar{\mathbf{J}}_{j\|p}\left(\ddot{\mathbf{x}}_{j\|p,ref} - \dot{\mathbf{J}}_{j\|p}\dot{\mathbf{q}}\right)\right] - \dot{\mathbf{J}}_{i\|p}\dot{\mathbf{q}} + \dot{\mathbf{J}}_i\dot{\mathbf{q}}$ [c]	k-1		$2m$
Summation		$[6mn^2 + (m^2 + 6m)n + (1/3)m^3 + 5.5m^2 - 3.5m](k\text{-}1)$	$2m(4n+1)(k-1)$

[a] The computation effort for an inverse of m-by-m positive definite matrix using Cholesky decomposition is $(1/3)m^3$ FLOPs.
[b] The final step of computing the original bias acceleration formula. There are some matrix multiplications, additions, and subtractions.
[c] The final step of computing the proposed acceleration-level bias acceleration formula. There are some additions and subtractions.

1) THE SIGNIFICANT COMPUTATIONAL EFFORT FOR IMPLEMENTING THE ORIGINAL FORMULA

The number of FLOPs for implementing OSF ($FLOP_{OSF/bias}$) except for the bias acceleration, which was reported in [6], was

$$\begin{aligned} FLOP_{OSF/bias} = {} & (1/3)\,n^3 + [(6k-4)\,m + 1]\,n^2 \\ & + \left[3km^2 + (5k+1)\,m - 1\right]n \\ & + k\left[(1/3)\,m^3 + 3.5m^2 - 1.5m\right]. \end{aligned} \tag{42}$$

$(k-1)$ iterations are needed to obtain the bias accelerations for the second to k^{th} tasks. The number of FLOPs for computing the bias acceleration using the original BAF ($FLOP_{bias\_o}$) was analyzed in the left column of Table 1 and given below

$$\begin{aligned} FLOP_{bias\_o} = {} & (k-1)\left[6mn^2 + \left(m^2 + 6m\right)n \right. \\ & \left. + (1/3)\,m^3 + 5.5m^2 - 3.5m\right]. \end{aligned} \tag{43}$$

The total number of FLOPs needed to implement OSF with the original BAF ($FLOP_{OSF\_o}$) becomes

$$\begin{aligned} FLOP_{OSF\_o} = {} & \left(1/3\right)n^3 + [(12k-10)\,m + 1]\,n^2 \\ & + \left[3km^2 + (k-1)m^2 + (11k+5)\,m - 1\right]n \\ & + (2k-1)\,m^3/3 + (9k-5.5)\,m^2 \\ & - (5k-3.5)\,m. \end{aligned} \tag{44}$$

Based on (43) and (44), $FLOP_{OSF\_o}$ and $FLOP_{bias\_o}$ were obtained for the case of increasing n with k and m fixed ($k = 2, m = 3$; Fig. 1(a)), for the case of increasing k with n and m fixed ($n = 30, m = 3$; Fig. 1(b)), and for the case of increasing m with n and k fixed ($n = 30, k = 2$; Fig. 1(c)). For the k and m fixed ($k = 2, m = 3$), and n is 50, $FLOP_{bias\_o}$ (46398 FLOPs) was 29.9% of $FLOP_{OSF\_o}$ (154937 FLOPs). Moreover, for $k = 10$ ($n = 30$, $m = 3$), $FLOP_{bias\_o}$ (153522 FLOPs) was 46.9% of $FLOP_{OSF\_o}$ (327642 FLOPs). Furthermore, for the n and k fixed ($n = 30$, $k = 2$), and m is 15, $FLOP_{bias\_o}$ (92760 FLOPs) was 35.7% of $FLOP_{OSF\_o}$ (259860 FLOPs).

It was found that the computation of $\hat{\boldsymbol{\Lambda}}_i^{-1}$, $\hat{\boldsymbol{\Lambda}}_i$, and $\hat{\boldsymbol{\mu}}_i + \hat{\mathbf{p}}_i$ took a significant portion of $FLOP_{bias\_o}$ (Table 1 and Fig. 2) as reported in [6] and [27]–[29]. For a 30 DOFs robot performing four tasks, each of which requires 3 DOFs ($n = 30$, $k = 4$, and $m = 3$), the computation of $\hat{\boldsymbol{\Lambda}}_i^{-1}$, $\hat{\boldsymbol{\Lambda}}_i$, and $\hat{\boldsymbol{\mu}}_i + \hat{\mathbf{p}}_i$ took 65.5% (33534 FLOPs) of $FLOP_{bias\_o}$. Moreover, the number of FLOPs for computing $\mathbf{J}_i\hat{\mathbf{M}}^{-1}\sum_{j=1}^{i-1}\boldsymbol{\Gamma}_j$ was 16461, another 32.2% of $FLOP_{bias\_o}$.

Thus, the computational effort for computing the original BAF took a substantial portion of the total computational effort needed for the OSF and would be desirable to be reduced considering the use of unnecessary $\hat{\mathbf{V}}$ and $\hat{\mathbf{G}}$.

2) SUBSTANTIAL IMPROVEMENT OF COMPUTATIONAL EFFICIENCY WITH THE PROPOSED FORMULA

The proposed BAF, (31), does not need $\hat{\mathbf{V}}$, $\hat{\mathbf{G}}$, $\hat{\boldsymbol{\Lambda}}_i^{-1}$, and $\hat{\boldsymbol{\Lambda}}_i$, thereby improving computational efficiency dramatically compared with the original BAF. The number of FLOPs for the proposed BAF ($FLOP_{bias\_p}$) was analyzed in the right column of Table 1 and given below

$$FLOP_{bias\_p} = 2m\,(4n+1)\,(k-1). \tag{45}$$

Details of obtaining $FLOP_{bias\_p}$ are given in Appendix B. For the case of using the proposed BAF, the number of

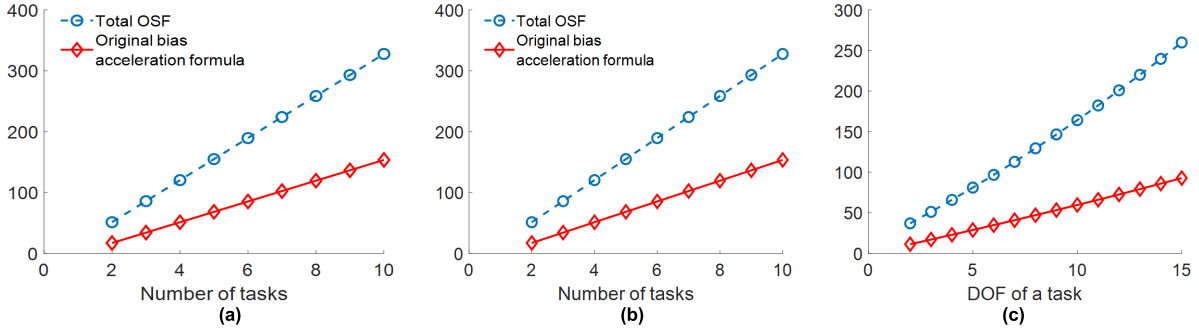


FIGURE 1. Computational cost needed for total the total OSF (blue circle) and that for original bias acceleration formula (red diamond). (a) As the DOFs of the robot increases from 6 to 50 (DOFs of a task: 3, the number of tasks: 2) (b) As the number of tasks increases from 2 to 10 (DOFs of a robot: 30, DOFs of a task: 3). (c) As the DOFs of a task increases from 2 to 15 (DOFs of a robot: 30, the number of tasks: 2).

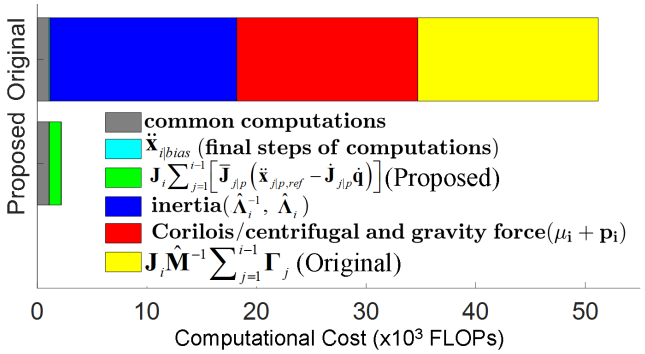


FIGURE 2. A representative case showing the computational cost for each term of the original and the proposed bias acceleration formulas (a 30 DOFs robot performing four tasks needing three DOFs each). Gray bar depicts the FLOPs equally required for both formulas. The light blue bar represents the final step of computing for both formulas (see Table 1). The blue ($\hat{\Lambda}_i^{-1}$ and its inverse, $\hat{\Lambda}_i$) and red ($\hat{\mu}_i+\hat{\mathbf{p}}_i$) color computations are not needed in the proposed formula, thus reducing a significant portion of computational effort. The accelerations induced by preceding tasks also contribute to the computational gain, which is depicted by the yellow bar for the original formula and the green bar for the proposed formula, respectively.

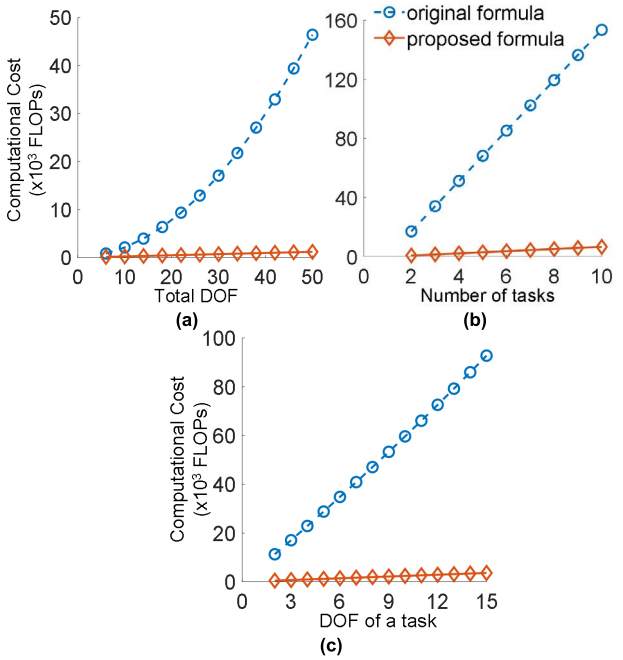


FIGURE 3. Computational effort to evaluate bias acceleration in one sampling time. (a) As the DOFs of the robot increases from 6 to 50 (DOFs of a task: 3, the number of tasks: 2) (b) As the number of tasks increases from 2 to 10 (DOFs of a robot: 30, DOFs of a task: 3). (c) As the DOFs of a task increases from 2 to 15 (DOFs of a robot: 30, the number of tasks: 2).

FLOPs needed for OSF except for the bias acceleration is the same as the case with the original BAF ($FLOP_{OSF/bias}$). Thus, the total number of FLOPS needed to implement OSF with the proposed BAF ($FLOP_{OSF\_p}$) is the summation of $FLOP_{OSF/bias}$ and $FLOP_{bias\_p}$ as follows:

$$\begin{aligned} FLOP_{OSF\_p} = &\left(1/3\right)n^3+\left[\left(6k-4\right)m+1\right]n^2 \\ &+\left[3km^2+\left(5k+1\right)m+8\left(k-1\right)m-1\right]n \\ &+\left(k/3\right)m^3+3.5km^2+\left(0.5k-2\right)m. \end{aligned} \tag{46}$$

$FLOP_{bias\_p}$ was obtained for the case of increasing n with m and k fixed ($m=3$, $k=2$; Fig. 3(a)), for the case of increasing k with n and m fixed ($n=30, m=3$; Fig. 3(b)) [6], and for the case of increasing m with n and k fixed ($n=30$, $k=2$; Fig. 3(c)). Consequently, $FLOP_{OSF\_p}$ was obtained.

With m and k fixed, for the original BAF, the FLOPs required was $O(n^2)$, and for the proposed BAF, it was reduced to $O(n)$ as shown in Table 1. For instance, when n is 50 ($k=2, m=3$), $FLOP_{bias\_p}$ (1206 FLOPs) was only 2.6% of $FLOP_{bias\_o}$ (46398 FLOPs; Fig. 3(a)), and, due to the reduction, $FLOP_{OSF\_p}$ was 29.2% smaller than $FLOP_{OSF\_o}$.

Moreover, for the same n and m, both $FLOP_{bias\_o}$ and $FLOP_{bias\_p}$ were linearly increased with k. There was, however, a huge reduction in the number of FLOPs with the proposed formula (Fig. 3(b)). For instance, for $k=10$ ($n=30$ and $m=3$), $FLOP_{bias\_p}$ (6534 FLOPs) was 4.3% of $FLOP_{bias\_o}$ (153522 FLOPs; Fig. 3(b)), and $FLOP_{OSF\_p}$ was 44.9% smaller than $FLOP_{OSF\_o}$.

Furthermore, for n and k fixed, $FLOP_{bias\_o}$ was $O(m^3)$, and $FLOP_{bias\_p}$ was $O(m)$, indicating the proposed BAF required much fewer FLOPs than the original BAF: when m is 15 ($k = 2$ and $n = 30$), $FLOP_{bias\_p}$ (3630 FLOPs) was 3.9% of $FLOP_{bias\_o}$ (92760 FLOPs; Fig. 3(c)), and $FLOP_{OSF\_p}$ was 34.3% smaller than $FLOP_{OSF\_o}$.

Overall, $FLOP_{bias\_p}$ was substantially smaller than $FLOP_{bias\_o}$, indicating the high efficiency of the proposed BAF. Further, $FLOP_{OSF\_p}$ was substantially smaller than $FLOP_{OSF\_o}$ with a 30-50% reduction in FLOPs, meaning the proposed BAF helped dramatically reduce the computational load in implementing the OSF.

It seems adequate to check where the efficiency of the proposed BAF comes from. The computation of $\hat{\mathbf{\Lambda}}_i^{-1}$, $\hat{\mathbf{\Lambda}}_i$, and $\hat{\boldsymbol{\mu}}_i + \hat{\mathbf{p}}_i$, which took almost two-thirds of $FLOP_{bias\_o}$, is not required in the proposed BAF (Fig. 2). Moreover, the number of FLOPs for the acceleration term induced by the preceding tasks in the original BAF ($\mathbf{J}_i\hat{\mathbf{M}}^{-1}\sum_{j=1}^{i-1}\mathbf{\Gamma}_j$), which could take another one-third of $FLOP_{bias\_o}$ (Fig. 2), is not required for the proposed BAF (Table 1). Therefore, the proposed BAF is a lot more efficient than the original BAF mainly because, in the proposed BAF, the computations to obtain the unnecessary $\hat{\boldsymbol{\mu}}_i + \hat{\mathbf{p}}_i$, needing $\hat{\mathbf{V}}$ and $\hat{\mathbf{G}}$, along with computing $\hat{\mathbf{\Lambda}}_i^{-1}$ and $\hat{\mathbf{\Lambda}}_i$, are eliminated.

G. ADVANTAGES OF THE PROPOSED ACCELERATION-LEVEL BIAS ACCELERATION FORMULA

The proposed acceleration-level BAF has the following advantages. First, the proposed acceleration-level BAF is highly efficient compared with that of the original BAF. The 30-50% of arithmetic operations needed to implement OSF are reduced. As revealed, the unnecessarily required $\hat{\mathbf{V}}$ and $\hat{\mathbf{G}}$ for the original BAF are no longer needed for the proposed BAF. The proposed formula only needs $\hat{\mathbf{M}}$ to compute $\bar{\mathbf{J}}_{j|p}$.

Second, the proposed BAF fully satisfies the underlying time precedence that should be met for any BAF.

Third, the bias acceleration's independence from $\hat{\mathbf{V}}$ and $\hat{\mathbf{G}}$, which was proved for the first time in this study, is fully reflected in the proposed BAF. That is, the proposed formula does not need those terms at all. Thus, the proposed BAF can be used anywhere, including space, ground, and underwater, without any modification depending on the environment regardless of the changes in gravity. For instance, when a robot, which was used on the Earth, is moved to Mars, the proposed BAF does not need to change the value of gravitational acceleration in contrast to the original BAF, which requires the change of the value of gravitational acceleration.

Fourth, if $\ddot{\mathbf{x}}_{i,ref}$ does not need the acceleration of the robot, the proposed BAF requires no acceleration information, except those related to the joint level velocity term, $\dot{\mathbf{q}}$, implying that $\ddot{\mathbf{x}}_{i|bias\_P}$ could be much less noisy than $\ddot{\mathbf{x}}_{i|bias\_N}$.

IV. SIMULATIONS AND EXPERIMENTS

Two studies were performed. The first study with a 6 DOFs robot manipulator was designed to present that the bias acceleration is independent of $\hat{\mathbf{V}}$ and $\hat{\mathbf{G}}$, $\ddot{\mathbf{x}}_{i|bias\_P}$ is equivalent to $\ddot{\mathbf{x}}_{i|bias\_O}$, and $\ddot{\mathbf{x}}_{i|bias\_N}$ does not correctly represent the bias acceleration. The second study with a 17 DOFs upper body of a humanoid was performed to verify the remarkable efficiency of the proposed BAF compared to the original BAF.

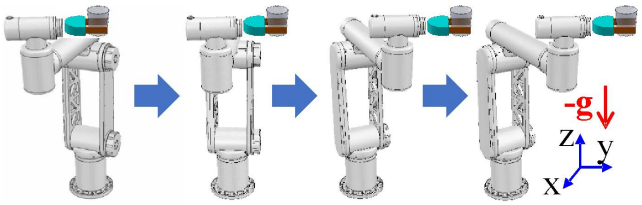


FIGURE 4. A 6 DOFs robot manipulator serving a cup of juice. First task was to move the cup in the y-direction, and the second task was to maintain the horizontal orientation of the cup.

A. FIRST STUDY: A 6 DEGREES-OF-FREEDOM ROBOT SIMULATIONS

1) THE 6 DEGREES-OF-FREEDOM ROBOT AND THE SIMULATION ENVIRONMENT

A 6 DOFs robot manipulator INDY RP (Neuromeka, South Korea) was used (Fig. 4). For all joints, the gear ratio was 100:1, and the encoder resolution was 17 bits/revolution. The rated torque of the first and the second joint starting from the base was 1.3 Nm each, of the third and fourth joints was 0.64 Nm, and of the fifth and sixth joints was 0.32 Nm. The robot was controlled by using the control torque, (17), for each task. The design of $\ddot{\mathbf{x}}_{i|p,ref}$ are given in Appendix C. For numerical integrations in the simulations, we have employed the fourth-order Runge-Kutta method with the time step of 0.1 ms, which was one-tenth of the sampling time (1 ms) for control and data acquisition.

2) TASKS

The goal of the robot was to serve a cup of juice without spilling it on the ground (Fig. 4). For the success of juice serving, two tasks were assigned to the robot: the first task having the highest priority was to move the cup on the end-effector horizontally in the y-direction, requiring three translational DOFs. The second task having a lower priority than the first one was to maintain the horizontal orientation of the cup needing three orientation DOFs. Consequently, the bias acceleration ($\ddot{\mathbf{x}}_{2|bias} = \begin{bmatrix} \ddot{x}_{2|bias-1} & \ddot{x}_{2|bias-2} & \ddot{x}_{2|bias-3} \end{bmatrix}^T$) to determine the prioritized control torque for the second task needed to be determined. For the first task, the x- and z-direction desired position trajectories were constant values of the initial position. The y-direction desired trajectory was designed to move 0.4 m starting from an initial position for 4.2 s using cubic polynomials, guaranteeing a smooth trajectory. For the second task, the desired orientation was set as constant values of the initial orientation.

3) SIMULATION CONDITIONS

Two sets of simulations were performed using the 6 DOFs robot. For the first set of simulations, we have assumed that

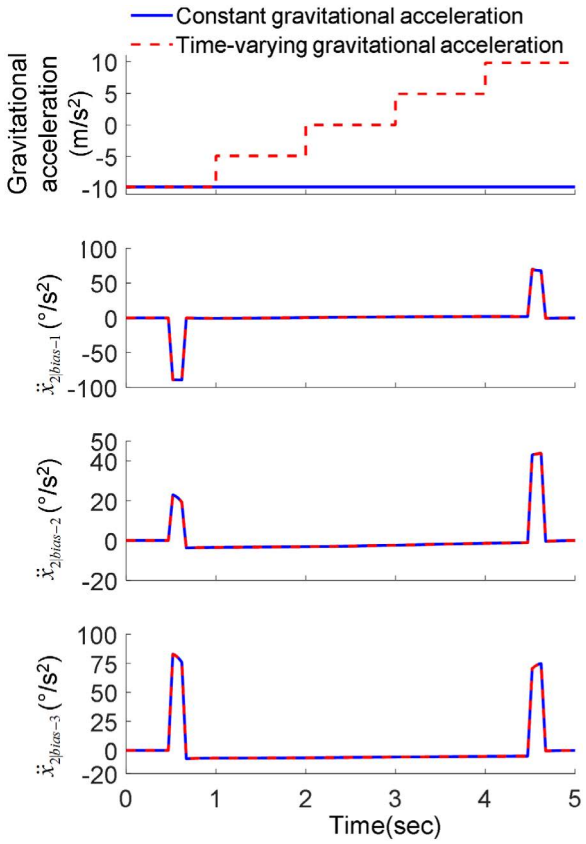


FIGURE 5. Bias accelerations obtained from the original formula under two different gravitational acceleration conditions. The top plot shows two different gravitational acceleration conditions (blue: downward constant gravity; red-dash: time-varying gravity). The three plots below the top plot show that the three components of the bias acceleration obtained under downward constant gravity was identical to those obtained under time-varying gravity, indicating the bias acceleration is independent of gravitational torque.

the exact values of all joint angles could be measured without any noise. The robot was controlled using the controller given in (17), which used the bias acceleration obtained. We obtained the bias acceleration ($\ddot{\mathbf{x}}_{2|bias\_O}$) from the original BAF, (23), utilizing $\hat{\mathbf{V}}$ and $\hat{\mathbf{G}}$, for the following two different cases. The first case is when there is a downward gravitational acceleration of -9.8 m/s^2; and the second case is when there is an artificially time-varying gravitational acceleration from -9.8m/s^2 to 9.8 m/s^2 in the vertical direction (Fig. 5).

In the second set of simulations, to investigate the effect of $\ddot{\mathbf{x}}_{2|bias\_N}$ (the bias acceleration obtained from numerical differentiation), $\ddot{\mathbf{x}}_{2|bias\_O}$, $\ddot{\mathbf{x}}_{2|bias\_P}$, and $\ddot{\mathbf{x}}_{2|bias\_N}$ were obtained by considering the 17 bits/revolution encoder resolution. The three bias accelerations obtained were used for the OSF in each case.

4) DATA ANALYSIS

To investigate the difference between $\ddot{\mathbf{x}}_{2|bias\_P}$ and $\ddot{\mathbf{x}}_{2|bias\_O}$ and the difference between $\ddot{\mathbf{x}}_{2|bias\_N}$ and $\ddot{\mathbf{x}}_{2|bias\_O}$, the following measures (d_P and d_N) were defined.

$$d_P = \left\| \ddot{\mathbf{x}}_{2|bias\_O} - \ddot{\mathbf{x}}_{2|bias\_P} \right\|, \text{ and} \tag{47}$$

$$d_N = \left\| \ddot{\mathbf{x}}_{2|bias\_O} - \ddot{\mathbf{x}}_{2|bias\_N} \right\|. \tag{48}$$

Moreover, to investigate how well the control objective for the second task (i.e., $\ddot{\mathbf{x}}_2 = \ddot{\mathbf{x}}_{2,ref}$) was realized, we have defined a measure σ representing the degree of departure from desired closed-loop dynamics, (4), as follows:

$$\sigma = \left\| \ddot{\mathbf{x}}_{2,ref} - \ddot{\mathbf{x}}_2 \right\|. \tag{49}$$

Specifically, we have obtained three different values: σ_O, σ_P, and σ_N. σ_O denotes the degree of departure from the desired dynamics when using $\ddot{\mathbf{x}}_{2|bias\_O}$; σ_P the degree of departure from the desired dynamics when using $\ddot{\mathbf{x}}_{2|bias\_P}$; and σ_N the degree of departure from the desired dynamics when using $\ddot{\mathbf{x}}_{2|bias\_N}$.

Further, a measure showing the difference in orientation of the cup and the horizontal plane was defined as the angle ($\Delta\theta$) between the upward vertical vector normal to the horizontal plane and the normal vector to the cup. $\Delta\theta_O$, $\Delta\theta_P$, and $\Delta\theta_N$ represent the angular deviation when using $\ddot{\mathbf{x}}_{2|bias\_O}$, that when using $\ddot{\mathbf{x}}_{2|bias\_P}$, and that when $\ddot{\mathbf{x}}_{2|bias\_N}$, respectively.

TABLE 2. Maximum norm difference from the bias acceleration obtained from the original bias acceleration formula (°/s^2).

Proposed formula (maximum d_P)	Numerical differentiation method (maximum d_N)
0.00	246.94

There was no difference between the proposed formula and the original formula, indicating the proposed formula is identical to the original one. The bias acceleration obtained from the numerical differentiation was substantially different from that obtained from the original formula.

5) SIMULATION RESULTS

$\ddot{\mathbf{x}}_{2|bias\_O}$ under the constant downward gravitational acceleration was identical to that under time-varying gravitational acceleration (Fig. 5). $\ddot{\mathbf{x}}_{2|bias\_P}$, which clearly does not use $\hat{\mathbf{V}}$ and $\hat{\mathbf{G}}$, was identical to $\ddot{\mathbf{x}}_{2|bias\_O}$ as the maximum d_P is zero (Fig. 6 and Table 2).

$\ddot{\mathbf{x}}_{2|bias\_N}$ was significantly larger than $\ddot{\mathbf{x}}_{2|bias\_O}$ and $\ddot{\mathbf{x}}_{2|bias\_P}$ (Fig. 6 and Table 2). Moreover, clearly, $\ddot{\mathbf{x}}_{2|bias\_N}$ was highly noisy (Fig. 6). With the use of noisy and large $\ddot{\mathbf{x}}_{2|bias\_N}$, the resulting dynamics was significantly different from the desired dynamics: the maximum of σ_N was 211 times larger than the maximum of σ_O (Fig. 6 and Table 3). In contrast, σ_O and σ_P were identical at all time points, and the magnitude of the two was small (Fig. 6 and Table 3), indicating the desired dynamics were closely realized.

The angular deviation of the cup from the horizontal plane when using $\ddot{\mathbf{x}}_{2|bias\_P}$ was identical to that when using $\ddot{\mathbf{x}}_{2|bias\_O}$ (i.e., $\Delta\theta_P = \Delta\theta_O$), and, for both cases, the absolute maximum was 0.0008° (Fig. 6 and Table 3). In contrast, the angular deviation when using $\ddot{\mathbf{x}}_{2|bias\_N}(\Delta\theta_N)$ was significantly

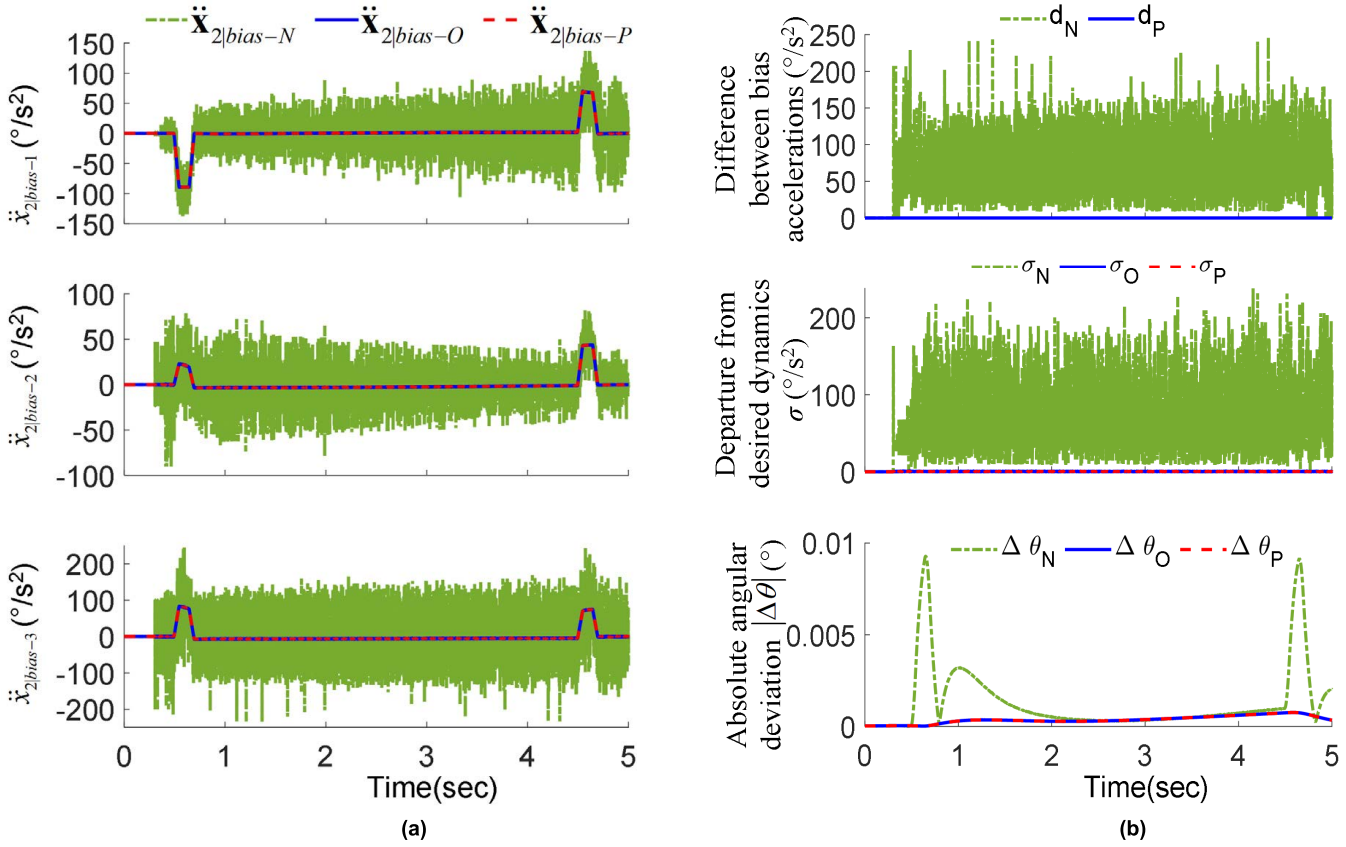


FIGURE 6. When the encoder resolution is considered, the results of a robot control using OSF with different bias accelerations ($\ddot{\mathbf{x}}_{2|bias\_O}$, $\ddot{\mathbf{x}}_{2|bias\_P}$, and $\ddot{\mathbf{x}}_{2|bias\_N}$) (a) Comparison of the bias accelerations $\ddot{\mathbf{x}}_{2|bias\_O}$, $\ddot{\mathbf{x}}_{2|bias\_P}$, and $\ddot{\mathbf{x}}_{2|bias\_N}$. From the top to the bottom, the three components of the bias accelerations are presented. The green dashed-dotted line denotes $\ddot{\mathbf{x}}_{2|bias\_N}$ from numerical differentiation; the blue solid line and the red dashed line depict $\ddot{\mathbf{x}}_{2|bias\_O}$ from the original formula, and $\ddot{\mathbf{x}}_{2|bias\_P}$ from the proposed formula, respectively. (b) The top plot shows the difference between $\ddot{\mathbf{x}}_{2|bias\_P}$ and $\ddot{\mathbf{x}}_{2|bias\_O}$, and that between $\ddot{\mathbf{x}}_{2|bias\_N}$ and $\ddot{\mathbf{x}}_{2|bias\_O}$. The middle plot shows the degree of departure from desired dynamics; the bottom plot shows the angular deviation of the cup from the horizontal plane. $\ddot{\mathbf{x}}_{2|bias\_O}$ and $\ddot{\mathbf{x}}_{2|bias\_P}$ were identical to each other, whereas $\ddot{\mathbf{x}}_{2|bias\_N}$ was significantly different from $\ddot{\mathbf{x}}_{2|bias\_O}$ with large and fluctuating noise. The fluctuation eventually deteriorated the control performance, which can be confirmed by large σ_N and $|\Delta\theta_N|$.

TABLE 3. Degree of realization of the control objective for the second task with different bias acceleration formulae.

	Proposed formula	Original formula	Numerical differentiation
Maximum σ (°/s²)	Max σ_P = 1.13	Max σ_O = 1.13	Max σ_N = 238.20
Maximum $\lvert\Delta\theta\rvert$ (°)	Max $\lvert\Delta\theta_P\rvert$ = 0.0008	Max $\lvert\Delta\theta_O\rvert$ = 0.0008	Max $\lvert\Delta\theta_N\rvert$ = 0.0093

Between the original and the proposed bias acceleration formulae, there was no difference in the degree of departure from the desired closed-loop dynamics and the difference in orientation of the cup and the horizontal plane. Using the bias acceleration from the numerical differentiation, the control objective for the second task was not well-realized. The poor realization of the desired closed-loop dynamics was reflected in the increase of difference in the orientation between the normal vector of the plate and that of the horizontal plane.

larger than both $\Delta\theta_O$ and $\Delta\theta_P$. The maximum of $|\Delta\theta_N|$ was 12 times larger than the maximum of $|\Delta\theta_O|$ and that of $|\Delta\theta_P|$ (Fig. 6 and Table 3).

6) DISCUSSION

First, we have confirmed that the bias acceleration is independent of $\hat{\mathbf{V}}$ and $\hat{\mathbf{G}}$ as proven mathematically with rigor. $\ddot{\mathbf{x}}_{2|bias\_O}$ did not change even though there was a change in gravity. Moreover, $\ddot{\mathbf{x}}_{2|bias\_O}$ was identical to $\ddot{\mathbf{x}}_{2|bias\_P}$, which do not require $\hat{\mathbf{V}}$ and $\hat{\mathbf{G}}$.

Second, the equivalence between $\ddot{\mathbf{x}}_{2|bias\_O}$ and $\ddot{\mathbf{x}}_{2|bias\_P}$ was verified. The equivalence holds even with the quantization error (e.g., the encoder resolution). Further, the resulting dynamics were identical: $\sigma_O = \sigma_P$. Consequently, the angular deviation of the cup from the horizontal plane was the same

Third, $\ddot{\mathbf{x}}_{2|bias\_N}$ did not correctly represent the bias acceleration, accompanied by large and fluctuating noise. The fluctuation in $\ddot{\mathbf{x}}_{2|bias\_N}$ caused $\ddot{\mathbf{x}}_{2|p,ref}$ to fluctuate, so did the control force $\mathbf{F}_{2|p}$. Eventually, control performance deteriorated significantly, as reflected in σ_N (Fig. 6). Consequently, even with the perfect robot dynamics model, which is an ideal condition that cannot be met in practice, the control objective was not realized well. These results emphasize the importance of obeying time precedence.

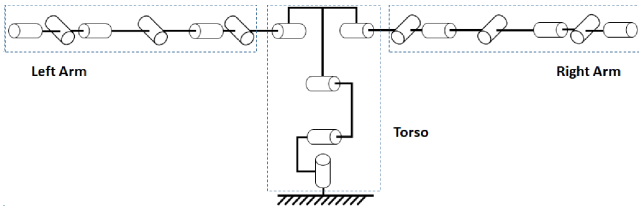


FIGURE 7. Kinematic structure of the upper body humanoid with 17 DOFs.

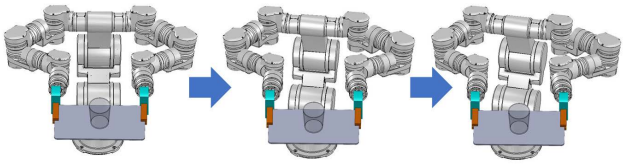


FIGURE 8. Commanded robot motion: The robot was holding a tray in the current position and orientation while rotating the torso. Three tasks were assigned for this. Task 1 was to control the five joints of the torso. Task 2 and task 3, each with 6 DOFs, were to control the position and orientation of the left and right end effector, respectively.

B. SECOND STUDY: COMPUTATIONAL EFFICIENCY EXPERIMENTS

The computational time for $\ddot{\mathbf{x}}_{i|bias\_O}$ and $\ddot{\mathbf{x}}_{i|bias\_P}$ were compared. A 17 DOFs upper body of a humanoid having a 5 DOFs torso and two 6 DOFs arms (Fig. 7) was controlled using the controller given in (17) based on the OSF.

The two BAFs with the controller were implemented on real-time Linux using a computer having a 1.6GHz CPU with a 1 ms sampling time. The fourth-order Runge-Kutta method with a 0.1 ms time step was used for numerical integration. The total computation time needed for each of the two BAFs ($\ddot{\mathbf{x}}_{i|bias\_O}$ and $\ddot{\mathbf{x}}_{i|bias\_P}$) at each time instance was measured.

The goal of the robot was to maintain a tray at a fixed position and orientation using the two arms while moving the torso, like serving a juice while listening to a guest's question (Fig. 8). To accomplish the goal, three tasks were needed. The first task having the highest priority was to control the position of the torso (joint level control). The second and third tasks were to maintain the position and orientation of the end-effectors of the left arm and the right arm. Thus, two bias accelerations ($\ddot{\mathbf{x}}_{2|bias}$ and $\ddot{\mathbf{x}}_{3|bias}$) were computed and used to implement OSF. The design of $\ddot{\mathbf{x}}_{i|p,ref}$ was the same as the first study (see Appendix C). For the first task, the desired trajectory for the first torso joint from the base was to rotate 15° from the initial posture (Fig. 8) in 0.55 s using cubic polynomials, guaranteeing a smooth trajectory. For all other torso joints, the desired trajectory was a constant value of an initial position throughout the task. The desired position and orientation trajectory for the second and third tasks were constant values of initial positions and orientations, respectively.

To demonstrate the efficiency of the proposed BAF compared to the original BAF, $\ddot{\mathbf{x}}_{2|bias}$ and $\ddot{\mathbf{x}}_{3|bias}$ were computed using the original BAF in one experiment and the proposed BAF in another experiment. For each experiment, the total computation time in computing the bias acceleration was obtained at each time instance. The mean, standard deviation, and maximum of the total computation time for each BAF were then obtained.

TABLE 4. Bias acceleration computational time, mean (SD).

Proposed formula	Original formula
0.011 (0.002) *ms*	0.102 (0.007) *ms*

The total computation time for the proposed BAF was only 11% of that for the original BAF (Table 4). These results strongly support the substantially improved computational efficiency of the proposed BAF as in section III.F. For this case, the total computation time for the proposed BAF for each time instance was 1% of 1 ms. 1 ms could be a sufficient sampling time even for force control [31].

V. CONCLUSION

In this study, we proposed a highly efficient acceleration-level BAF, which does not need the Coriolis/centrifugal and gravity torques, OS inertia, and its inverse. The proposed BAF can be used in any environment with no modification regardless of gravitational acceleration.

Moreover, we have clarified the time precedence in the implementation of the bias acceleration, which was only implied implicitly in the original BAF and rarely discussed. It was made clear that the bias acceleration is a future value that cannot be measured by any means during the real-time control, and the numerical differentiation method yielding $\ddot{\mathbf{x}}_{i|bias\_N}$ cannot provide us with future value during the real-time control. The rare discussion on the time precedence might have caused some misunderstanding on the bias acceleration, potentially leading to the physically incorrect implementation (e.g., the numerical differentiation method used in [6]).

Further, we proved that the inefficiency of the original BAF is due to the use of unnecessary Coriolis/centrifugal and gravitational torques, which are not needed for the proposed BAF. During the process, for the first time, we have proved that the bias acceleration is independent of Coriolis/ centrifugal and gravity torques.

It is expected that the investigation of the bias acceleration could provide a deeper understanding of it, and the proposed efficient BAF could promote the use of high DOFs robots, including multi-arm manipulators, humanoids, and service robots.

APPENDIX

A. PROOF OF (19)

The derivation of (19) is provided. Multiplying $\mathbf{J}_i\hat{\mathbf{M}}^{-1}$ on both sides of the JS dynamics (2) yields

$$\mathbf{J}_i\ddot{\mathbf{q}} + \mathbf{J}_i\hat{\mathbf{M}}^{-1}(\hat{\mathbf{V}} + \hat{\mathbf{G}}) = \mathbf{J}_i\hat{\mathbf{M}}^{-1}\boldsymbol{\Gamma}. \tag{50}$$

Since $\mathbf{J}_i\ddot{\mathbf{q}} + \dot{\mathbf{J}}_i\ddot{\mathbf{q}} = \ddot{\mathbf{x}}_i$, the left-hand side of (50) becomes

$$\mathbf{J}_i\ddot{\mathbf{q}} + \mathbf{J}_i\hat{\mathbf{M}}^{-1}\left(\hat{\mathbf{V}} + \hat{\mathbf{G}}\right) = \ddot{\mathbf{x}}_i - \dot{\mathbf{J}}_i\ddot{\mathbf{q}} + \mathbf{J}_i\hat{\mathbf{M}}^{-1}\left(\hat{\mathbf{V}} + \hat{\mathbf{G}}\right). \tag{51}$$

From (20), (21), and (22), the RHS of (51) can be written as follows:

$$\ddot{\mathbf{x}}_i - \dot{\mathbf{J}}_i\ddot{\mathbf{q}} + \mathbf{J}_i\hat{\mathbf{M}}^{-1}\left(\hat{\mathbf{V}} + \hat{\mathbf{G}}\right) = \ddot{\mathbf{x}}_i + \hat{\mathbf{\Lambda}}_i^{-1}\left(\hat{\boldsymbol{\mu}}_i + \hat{\mathbf{p}}_i\right). \tag{52}$$

Therefore,

$$\mathbf{J}_i\ddot{\mathbf{q}} + \mathbf{J}_i\hat{\mathbf{M}}^{-1}\left(\hat{\mathbf{V}} + \hat{\mathbf{G}}\right) = \ddot{\mathbf{x}}_i + \hat{\mathbf{\Lambda}}_i^{-1}\left(\hat{\boldsymbol{\mu}}_i + \hat{\mathbf{p}}_i\right). \tag{53}$$

Applying the control torque (15), the RHS of (50) becomes

$$\mathbf{J}_i\hat{\mathbf{M}}^{-1}\mathbf{\Gamma} = \mathbf{J}_i\hat{\mathbf{M}}^{-1}\sum_{j=1}^{i-1}\mathbf{\Gamma}_j + \mathbf{J}_i\hat{\mathbf{M}}^{-1}\mathbf{\Gamma}_i + \mathbf{J}_i\hat{\mathbf{M}}^{-1}\sum_{j=i+1}^{k}\mathbf{\Gamma}_j. \tag{54}$$

Substituting (16) into the second and third terms on the RHS of (54) yields

$$\begin{aligned}\mathbf{J}_i\hat{\mathbf{M}}^{-1}\mathbf{\Gamma} &= \mathbf{J}_i\hat{\mathbf{M}}^{-1}\sum_{j=1}^{i-1}\mathbf{\Gamma}_j + \mathbf{J}_i\hat{\mathbf{M}}^{-1}\mathbf{J}_{i|p}^T\mathbf{F}_{i|p} \\ &\quad + \mathbf{J}_i\hat{\mathbf{M}}^{-1}\sum_{j=i+1}^{k}\mathbf{J}_{j|p}^T\mathbf{F}_{j|p}.\end{aligned} \tag{55}$$

Since $\mathbf{J}_i\hat{\mathbf{M}}^{-1}\mathbf{N}_j^T = \mathbf{0}(\forall i < j)$ [2], [5], substituting (5) into the RHS of (55) reveals that the RHS last term of (55) is $\mathbf{0}$, and, consequently, (55) can be rewritten as follows:

$$\mathbf{J}_i\hat{\mathbf{M}}^{-1}\mathbf{\Gamma} = \mathbf{J}_i\hat{\mathbf{M}}^{-1}\sum_{j=1}^{i-1}\mathbf{\Gamma}_j + \mathbf{J}_i\hat{\mathbf{M}}^{-1}\mathbf{N}_i^T\mathbf{J}_i^T\mathbf{F}_{i|p}. \tag{56}$$

Because $\mathbf{N}_i^r = \mathbf{N}_i$ for any positive integer r [2], [5], (56) can be rewritten as follows:

$$\begin{aligned}&\mathbf{J}_i\hat{\mathbf{M}}^{-1}\sum_{j=1}^{i-1}\mathbf{\Gamma}_j + \mathbf{J}_i\hat{\mathbf{M}}^{-1}\mathbf{N}_i^T\,\mathbf{J}_i^T\mathbf{F}_{i|p} \\ &= \mathbf{J}_i\hat{\mathbf{M}}^{-1}\sum_{j=1}^{i-1}\mathbf{\Gamma}_j + \mathbf{J}_i\hat{\mathbf{M}}^{-1}\mathbf{N}_i^T\,\mathbf{N}_i^T\mathbf{J}_i^T\mathbf{F}_{i|p} \\ &= \mathbf{J}_i\hat{\mathbf{M}}^{-1}\sum_{j=1}^{i-1}\mathbf{\Gamma}_j + \mathbf{J}_i\hat{\mathbf{M}}^{-1}\mathbf{N}_i^T\,\mathbf{J}_{i|p}^T\mathbf{F}_{i|p}.\end{aligned} \tag{57}$$

Since $\hat{\mathbf{M}}^{-1}\mathbf{N}_i^T = \mathbf{N}_i\hat{\mathbf{M}}^{-1}$ [32], (57) becomes

$$\begin{aligned}&\mathbf{J}_i\hat{\mathbf{M}}^{-1}\sum_{j=1}^{i-1}\mathbf{\Gamma}_j + \mathbf{J}_i\hat{\mathbf{M}}^{-1}\mathbf{N}_i^T\,\mathbf{J}_{i|p}^T\mathbf{F}_{i|p} \\ &= \mathbf{J}_i\hat{\mathbf{M}}^{-1}\sum_{j=1}^{i-1}\mathbf{\Gamma}_j + \mathbf{J}_i\mathbf{N}_i\hat{\mathbf{M}}^{-1}\,\mathbf{J}_{i|p}^T\mathbf{F}_{i|p} \\ &= \mathbf{J}_i\hat{\mathbf{M}}^{-1}\sum_{j=1}^{i-1}\mathbf{\Gamma}_j + \mathbf{J}_{i|p}\hat{\mathbf{M}}^{-1}\mathbf{J}_{i|p}^T\,\mathbf{F}_{i|p} \\ &= \mathbf{J}_i\hat{\mathbf{M}}^{-1}\sum_{j=1}^{i-1}\mathbf{\Gamma}_j + \hat{\mathbf{\Lambda}}_{i|p}^{-1}\mathbf{F}_{i|p}.\end{aligned} \tag{58}$$

Thus,

$$\mathbf{J}_i\hat{\mathbf{M}}^{-1}\mathbf{\Gamma} = \mathbf{J}_i\hat{\mathbf{M}}^{-1}\sum_{j=1}^{i-1}\mathbf{\Gamma}_j + \hat{\mathbf{\Lambda}}_{i|p}^{-1}\mathbf{F}_{i|p}. \tag{59}$$

Substituting (53) and (59) into (50) and solving for $\ddot{\mathbf{x}}_i$ yield

$$\ddot{\mathbf{x}}_i = \mathbf{J}_i\hat{\mathbf{M}}^{-1}\sum_{j=1}^{i-1}\mathbf{\Gamma}_j + \hat{\mathbf{\Lambda}}_{i|p}^{-1}\mathbf{F}_{i|p} - \hat{\mathbf{\Lambda}}_i^{-1}\left(\hat{\boldsymbol{\mu}}_i + \hat{\mathbf{p}}_i\right), \tag{60}$$

which is identical to (19). Therefore, (19) is proved. □

B. COMPUTATIONAL EFFORT FOR THE PROPOSED BIAS ACCELERATION FORMULA

To compute $\ddot{\mathbf{x}}_{i,bias\_P}$ using (31), first, we need $\dot{\mathbf{J}}_i\dot{\mathbf{q}}$ in addition to all the other terms that are computed to obtain the prioritized control torque (17). $\dot{\mathbf{J}}_i$ can be obtained from the numerical differentiation of $\mathbf{J}_i$ needing $2mn$ FLOPs [6]. Multiplication of $\dot{\mathbf{J}}_i$ and $\dot{\mathbf{q}}$ needs $(2mn-m)$ FLOPs. Thus, $(4mn-m)$ FLOPs are needed for $\dot{\mathbf{J}}_i\dot{\mathbf{q}}$. Secondly, the first term of the proposed BAF, $\mathbf{J}_i\sum_{j=1}^{i-1}\left[\bar{\mathbf{J}}_{j|p}\left(\ddot{\mathbf{x}}_{j|p,ref} - \dot{\mathbf{J}}_{j|p}\dot{\mathbf{q}}\right)\right]$, needs to be computed. For $j = i-1$, $\ddot{\mathbf{x}}_{j,ref} - \ddot{\mathbf{x}}_{j|bias}$ providing $\ddot{\mathbf{x}}_{j|p,ref}$, subtraction of $\dot{\mathbf{J}}_{j|p}\dot{\mathbf{q}}$ from $\ddot{\mathbf{x}}_{j|p,ref}$, and multiplication of $\bar{\mathbf{J}}_{j|p}$ and $\left(\ddot{\mathbf{x}}_{j|p,ref} - \dot{\mathbf{J}}_{j|p}\dot{\mathbf{q}}\right)$ require m FLOPs, m FLOPs, and $(2mn-n)$ FLOPs, respectively. Adding $\left[\bar{\mathbf{J}}_{i-1|p}\left(\ddot{\mathbf{x}}_{i-1|p,ref} - \dot{\mathbf{J}}_{i-1|p}\dot{\mathbf{q}}\right)\right]$ to $\sum_{j=1}^{i-2}\left[\bar{\mathbf{J}}_{j|p}\left(\ddot{\mathbf{x}}_{j|p,ref} - \dot{\mathbf{J}}_{j|p}\dot{\mathbf{q}}\right)\right]$, and multiplication of $\mathbf{J}_i$ and $\sum_{j=1}^{i-1}\left[\bar{\mathbf{J}}_{j|p}\left(\ddot{\mathbf{x}}_{j|p,ref} - \dot{\mathbf{J}}_{j|p}\dot{\mathbf{q}}\right)\right]$ need n FLOPs and $(2mn-m)$ FLOPs, respectively. Thus, in total, $(4mn+m)$ FLOPs are needed for the first term. Finally, summing all the three terms of the proposed BAF, (31), needs $2m$ FLOPs.

Therefore, for a total of $k-1$ iterations, $2m(4n+1)(k-1)$ FLOPs are needed for the proposed BAF.

C. THE DESIGN OF $\ddot{\mathbf{x}}_{i|p,ref}$

For position control tasks, the reference acceleration, $\ddot{\mathbf{x}}_{i|p,ref}$, for the i^{th} task was designed as follows [6]:

$$\ddot{\mathbf{x}}_{i|p,ref} = \ddot{\mathbf{x}}_{i,des} + \mathbf{K}_{Di}\dot{\mathbf{e}}_i + \mathbf{K}_{Pi}\mathbf{e}_i, \text{ and} \tag{61}$$

$$\mathbf{e}_i = \mathbf{x}_{i,des} - \mathbf{x}_i, \tag{62}$$

where $\mathbf{x}_{i,des}$, $\dot{\mathbf{x}}_{i,des}$, and $\ddot{\mathbf{x}}_{i,des}$ denote the desired position, velocity, and acceleration for the i^{th} task, respectively; $\mathbf{e}_i$ and $\dot{\mathbf{e}}_i$ position error and its first-time derivative, respectively; and $\mathbf{K}_{Di}$ and $\mathbf{K}_{Pi}$ the feedback gain matrices for the i^{th} task.

For orientation control tasks, assuming a small orientation error, the orientation error – used for constructing $\ddot{\mathbf{x}}_{i|p,ref}$

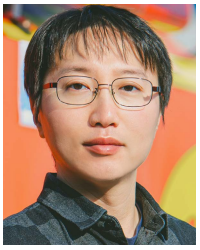
in the place of position error – may be described using the angle/axis representation [1], [33], and [34]:

$$\mathbf{e}_o = \mathbf{r}_o \sin\varphi, \tag{63}$$

where $\mathbf{e}_o$ denotes the orientation error, and $\mathbf{r}_o$ and φ represent the rotation axis and angle of an equivalent angle/axis representation between the desired orientation and the actual orientation, respectively.

REFERENCES

[1] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE J. Robot. Autom.*, vol. RA-3, no. 1, pp. 43–53, Feb. 1987.
[2] L. Sentis and O. Khatib, "Prioritized multi-objective dynamics and control of robots in human environments," in *Proc. IEEE-RAS Int. Conf. Hum. Robots*, Santa Monica, CA, USA, vol. 2, Nov. 2004, pp. 764–780.
[3] O. Khatib, L. Sentis, J. Park, and J. Warren, "Whole-body dynamic behavior and control of human-like robots," *Int. J. Hum. Robot.*, vol. 1, no. 1, pp. 29–43, Mar. 2004.
[4] L. Sentis and O. Khatib, "Task-oriented control of humanoid robots through prioritization," in *Proc. IEEE-RAS Int. Conf. Hum. Robots*, Los Angeles, CA, USA, Nov. 2004, pp. 475–480.
[5] L. Sentis and O. Khatib, "Synthesis of whole-body behaviors through hierarchical control of behavioral primitives," *Int. J. Hum. Robot.*, vol. 2, no. 4, pp. 505–518, Dec. 2005.
[6] P. H. Chang and J. W. Jeong, "Enhanced operational space formulation for multiple tasks by using time-delay estimation," *IEEE Trans. Robot.*, vol. 28, no. 4, pp. 773–786, Aug. 2012.
[7] O. Khatib, O. Brock, K.-S. Chang, D. Ruspini, L. Sentis, and S. Viji, "Human-centered robotics and interactive haptic simulation," *Int. J. Robot. Res.*, vol. 23, no. 2, pp. 167–178, Feb. 2004.
[8] J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal, "Operational space control: A theoretical and empirical comparison," *Int. J. Robot. Res.*, vol. 27, no. 6, pp. 737–757, Jun. 2008.
[9] O. Khatib, "Inertial properties in robotic manipulation: An object-level framework," *Int. J. Robot. Res.*, vol. 14, no. 1, pp. 19–36, Feb. 1995.
[10] R. Featherstone and O. Khatib, "Load independence of the dynamically consistent inverse of the Jacobian matrix," *Int. J. Robot. Res.*, vol. 16, no. 2, pp. 168–170, Apr. 1997.
[11] L. Sentis, J. Park, and O. Khatib, "Compliant control of multicontact and center-of-mass behaviors in humanoid robots," *IEEE Trans. Robot.*, vol. 26, no. 3, pp. 483–501, Jun. 2010.
[12] J. Russakow, S. M. Rock, and O. Khatib, "An operational space formulation for a free-flying, multi-arm space robot," in *Experimental Robotics IV*. Berlin, Germany: Springer, 1997, pp. 448–457.
[13] O. Khatib, X. Yeh, G. Brantner, B. Soe, B. Kim, S. Ganguly, H. Stuart, S. Wang, M. Cutkosky, A. Edsinger, and P. Mullins, "Ocean one: A robotic avatar for oceanic discovery," *IEEE Robot. Autom. Mag.*, vol. 23, no. 4, pp. 20–29, Dec. 2016.
[14] A. Tulbure and O. Khatib, "Closing the loop: Real-time perception and control for robust collision avoidance with occluded obstacles," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Las Vegas, NV, USA, Oct. 2020, pp. 5700–5707.
[15] J. Lee, H. Dallali, M. Jin, D. G. Caldwell, and N. G. Tsagarakis, "Robust and adaptive dynamic controller for fully-actuated robots in operational space under uncertainties," *Auto. Robots*, vol. 43, no. 4, pp. 1023–1040, Apr. 2019.
[16] M. Jorda, E. G. Herrero, and O. Khatib, "Contact-driven posture behavior for safe and interactive robot operation," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Montreal, QC, Canada, May 2019, pp. 9243–9249.
[17] M. Kaspar, J. D. M. Osorio, and J. Bock, "Sim2Real transfer for reinforcement learning without dynamics randomization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Las Vegas, NV, USA, Oct. 2020, pp. 4383–4388.
[18] J. Jung, S. You, and J. Park, "The operational space formulation on humanoids considering joint stiffness and bandwidth limit," *IEEE Access*, vol. 9, pp. 121883–121893, 2021.
[19] M. A. Lee, Y. Zhu, K. Srinivasan, P. Shah, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg, "Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Montreal, QC, Canada, May 2019, pp. 8943–8950.
[20] M. A. Lee, Y. Zhu, P. Zachares, M. Tan, K. Srinivasan, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg, "Making sense of vision and touch: Learning multimodal representations for contact-rich tasks," *IEEE Trans. Robot.*, vol. 36, no. 3, pp. 582–596, Jun. 2020.
[21] V. Morlando, A. Teimoorzadeh, and F. Ruggiero, "Whole-body control with disturbance rejection through a momentum-based observer for quadruped robots?" *Mech. Mach. Theory*, vol. 164, Oct. 2021, Art. no. 104412.
[22] P. Di Lillo, G. Antonelli, and C. Natale, "Effects of dynamic model errors in task-priority operational space control," *Robotica*, vol. 39, no. 9, pp. 1642–1653, Sep. 2021.
[23] J. Obregon-Flores, G. Arechavaleta, H. M. Becerra, and A. Morales-Diaz, "Predefined-time robust hierarchical inverse dynamics on torque-controlled redundant manipulators," *IEEE Trans. Robot.*, vol. 37, no. 3, pp. 962–978, Jun. 2021.
[24] Y. Lee, S. Kim, J. Park, N. Tsagarakis, and J. Lee, "A whole-body control framework based on the operational space formulation under inequality constraints via task-oriented optimization," *IEEE Access*, vol. 9, pp. 39813–39826, 2021.
[25] O. Khatib, L. Sentis, and J.-H. Park, "A unified framework for whole-body humanoid robot control with multiple constraints and contacts," in *Proc. Eur. Robot. Symp.*, H. Bruyninckx, L. Přeučil, and M. Kulich, Eds. Berlin, Germany: Springer, 2008, pp. 303–312.
[26] H. Sadeghian, L. Villani, M. Keshmiri, and B. Siciliano, "Dynamic multi-priority control in redundant robotic systems," *Robotica*, vol. 31, no. 7, pp. 1155–1167, Oct. 2013.
[27] R. Featherstone, "Exploiting sparsity in operational-space dynamics," *Int. J. Robot. Res.*, vol. 29, no. 10, pp. 1353–1368, Sep. 2010.
[28] K. D. Bhalerao, J. Critchley, D. Oetomo, R. Featherstone, and O. Khatib, "Distributed operational space formulation of serial manipulators," *J. Comput. Nonlinear Dyn.*, vol. 9, no. 2, pp. 021012:1–021012:10, Apr. 2014.
[29] P. M. Wensing, L. R. Palmer, and D. E. Orin, "Efficient recursive dynamics algorithms for operational-space control with application to legged locomotion," *Auto. Robots*, vol. 38, no. 4, pp. 363–381, Apr. 2015.
[30] G. Golub and C. V. Loan, *Matrix Computations*. Baltimore, MD, USA: Johns Hopkins Univ. Press, 1996.
[31] S. H. Kang, M. Jin, and P. H. Chang, "A solution to the accuracy/robustness dilemma in impedance control," *IEEE/ASME Trans. Mechatronics*, vol. 14, no. 3, pp. 282–294, Jun. 2009.
[32] L. Sentis, "Synthesis and control of whole-body behaviors in humanoid systems," Ph.D. dissertation, Dept. Elect. Eng., Standford Univ., Stanford, CA, USA, 2007.
[33] F. Caccavale, C. Natale, B. Siciliano, and L. Villani, "Resolved-acceleration control of robot manipulators: A critical review with experiments," *Robotica*, vol. 16, no. 5, pp. 565–573, Sep. 1998.
[34] J. Y. S. Luh, M. W. Walker, and R. P. C. Paul, "Resolved-acceleration control of mechanical manipulators," *IEEE Trans. Autom. Control*, vol. AC-25, no. 3, pp. 468–474, Jun. 1980.



SANG HYUN PARK (Member, IEEE) received the B.S. and M.S. degrees in mechanical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2002 and 2004, respectively. He is currently pursuing the Ph.D. degree in mechanical engineering with the Ulsan National Institute of Science and Technology (UNIST), Ulsan, South Korea. From December 2011 to March 2018, he was a Senior Researcher with Robostar Company Ltd., Ansan, South Korea. He is also a Senior Researcher with the Korea Institute of Robotics and Technology Convergence (KIRO), Pohang, South Korea. His research interests include redundant robotics, robot motion control, robust control of nonlinear plants, time-delay control, and disaster robotics.

MAOLIN JIN (Senior Member, IEEE) received the B.S. degree in material science and mechanical engineering from the Yanbian University of Science and Technology, Jilin, China, in 1999, and the M.S. and Ph.D. degrees in mechanical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2004 and 2008, respectively. He was the Postdoctoral Researcher at the Mechanical Engineering Research Institute, KAIST, in 2008, and also a Senior Researcher with the Research Institute of Industrial Science and Technology, Pohang, South Korea, from November 2008 to February 2016. He is currently the Director and the Chief Researcher of the Human-Centered Robotics Center of KIRO, Pohang. His research interests include robust control of nonlinear plants, time-delay control, robot motion control, electro-hydraulic actuators, winding machines, disaster robotics, and factory automation. He serves as an Associate Editor for the *International Journal of Control, Automation, and Systems* (IJCAS), *Journal of Drive and Control*, and *Journal of the Korean Society for Precision Engineering*.

SANG HOON KANG (Member, IEEE) received the B.S. *(cum laude)*, M.S., and Ph.D. degrees in mechanical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea, in 2000, 2002, and 2009, respectively. He was a Research Associate at the Sensory Motor Performance Program (SMPP) of the Shirley Ryan AbilityLab (formerly Rehabilitation Institute of Chicago, RIC), Chicago, IL, USA, also the Postdoctoral Research Fellow with the Department of Physical Medicine and Rehabilitation, Northwestern University, Chicago, IL, from 2010 to 2015, and also an Instructor at the Department of Biomedical Engineering, Northwestern University, Evanston, IL, in 2012 and 2013. He is currently as an Associate Professor with the Department of Mechanical Engineering, UNIST, Ulsan, Republic of Korea, and also an Adjunct Assistant Professor with the Department of Physical Therapy and Rehabilitation Science, School of Medicine, University of Maryland, Baltimore, MA, USA. His current research interests include rehabilitation robotics and biomechanics of human movement, with an emphasis on rehabilitation medicine.