Input Matrix Factorizations for Constrained Control Allocation

Martin Kirchengast, Martin Steinberger, *Member, IEEE,* and Martin Horn, *Member, IEEE*

***Abstract*—Control allocation is part of a hierarchical control architecture and distributes the desired control effort of a controller among a set of redundant actuators as it is used, for example, to meet high reliability requirements. Before applying control allocation, a factorization of a linear plant's input matrix must be carried out. This work investigates the influence of this factorization on two common algorithms for constrained control allocation: Direct Allocation and Redistributed Pseudoinverse. It is shown why the factorization does not affect the first method, whereas it influences the latter one as soon as the number of actuator saturations exceeds a certain threshold. On this basis a modification of Redistributed Pseudoinverse is proposed, which allows the prioritization of virtual control vector components. If actuator saturations prevent an exact solution, the errors of those components with high priority are preferentially forced to zero. Finally simulations demonstrate the main results.**

I. Introduction

There are several reasons for equipping systems with more actuators than necessary for ensuring controllability: fault tolerance, maintenance, control reconfiguration, and high requirements in accuracy and dynamic response [1]. Due to input redundancy in general infinitely many combinations of actuator actions result in the same effect on the plant. In case of strong input redundancy different input vectors result in the same dynamic behavior, whereas weak input redundancy means that various inputs yield the same steady state output [2]. An option to handle both of these types of plants[1] is the usage of control allocation (CA). Here a so-called high-level controller deals with a modified plant model with virtual inputs and the CA algorithm distributes them among the actuators ([1], [3]). This control architecture can be found for example in aerospace applications ([4], [5], and [6]), in marine vessel control ([7], [8]), and in automotive applications ([9], [10]). Also, in the fault-tolerance framework CA techniques are employed, e.g., in [11] and [12].

The first step to apply CA is transforming the model into a form that facilitates the separation of the regulation from the control allocation task ([1], [3]). For a strongly input-redundant plant model that is linear with respect to its inputs, an input matrix factorization has to be conducted, which is not unique for each plant. In [13] transformation matrices are introduced for describing the relation between factorizations and their influence on the most widely used CA-method, Generalized Inverse, is examined. Although factorization itself has no impact on CA via generalized inverses, it enables their efficient computation having regard to actuator constraints (see [13]).

Up to now, the influence of input matrix factorization on CA methods has not been examined to the author's knowledge. It is not obvious if the factorization influences the allocation results. The main contributions of the present work are:

(a) It is shown that Direct Allocation (DA) remains unaffected.

(b) The factorization impact on Redistributed Pseudoinverse (RPINV) is reduced to the number of active actuator constraints. It is proved that in general the rank-reduction of the pseudoinverse entails deviating allocation results for different factorizations.

(c) Above a certain number of active actuator constraints the desired control effort is generally not reached by RPINV. In order to address this issue an enhanced version of that algorithm is proposed. Therein one can specify allocation error components which should preferably vanish. This objective is met by a change of factorization during execution.

M. Kirchengast, M. Steinberger, and M. Horn are with the Institute of Automation and Control, Graz University of Technology, 8010 Graz, Austria, e-mail: martin.kirchengast@tugraz.at.

[1]However, the methods discussed in this work are intended for strongly input-redundant plants.

Consider the linear plant model

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B_u}\boldsymbol{u}, \tag{1}$$

with $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ being the system matrix, the state vector $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{B_u} \in \mathbb{R}^{n\times m}$ is the input matrix, and $\boldsymbol{u} \in \mathbb{R}^m$ being the input vector. Assume that $\boldsymbol{B_u}$ does not have full column rank, i.e.

$$k = \mathrm{rk}(\boldsymbol{B_u}) < m, \tag{2}$$

which indicates the input redundancy. The rank-nullity theorem [14] yields the dimension of the right nullspace of $\boldsymbol{B_u}$: $\dim[\mathcal{N}_\mathrm{r}(\boldsymbol{B_u})] = m - k$. Thus particular directions of control space $\mathbb{R}^m$ are mapped to zero or, in other words, infinitely many input vectors affect the dynamics of (1) in the same way. As a result of (2) the input matrix can be factorized into virtual input matrix $\boldsymbol{B_v} \in \mathbb{R}^{n\times k}$ and control effectivity matrix $\boldsymbol{B} \in \mathbb{R}^{k\times m}$, i.e.

$$\boldsymbol{B_u} = \boldsymbol{B_v}\boldsymbol{B} \tag{3}$$

and the ranks of all matrices are $\mathrm{rk}(\boldsymbol{B_v}) = \mathrm{rk}(\boldsymbol{B}) = k$. The factorization (3) of $\boldsymbol{B_u}$ induces the alternative plant model

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B_v}\boldsymbol{v} \tag{4a}$$

$$\boldsymbol{v} = \boldsymbol{B}\boldsymbol{u} \tag{4b}$$

where $\boldsymbol{v} \in \mathbb{R}^k$ is called virtual control vector and according to (2), it has less elements than $\boldsymbol{u}$. After designing a controller for redundancy-free (4a) one has to choose a suitable CA algorithm, which distributes the controller's output $\boldsymbol{v}$ among the available actuators $\boldsymbol{u}$ [3]. Typically actuators are constrained, i.e. only a subset

$$\Omega = \left\{\boldsymbol{u} \in \mathbb{R}^m \,\middle|\, u_{i,min} \le u_i \le u_{i,max} \;\; \forall i = 1,\dots,m\right\} \tag{5}$$

of the m-dimensional (m-D) control space is admissible. Consequently, a subset of k-D virtual control space is defined as

$$\Phi = \boldsymbol{B}\Omega := \left\{\boldsymbol{v} \in \mathbb{R}^k \middle| \exists \boldsymbol{u} \in \Omega : \boldsymbol{v} = \boldsymbol{B}\boldsymbol{u}\right\} \tag{6}$$

and called attainable moment set (AMS, see [4] and [5]). Whereas the mapping from Ω to Φ is unique, this is not true in the opposite direction due to Ω's higher dimension, i.e. for a given $\boldsymbol{v} \in \Phi$ there are infinitely many $\boldsymbol{u} \in \mathbb{R}^m$ (not only in Ω) that fulfill (4b). The main task of constrained CA is to solve the underdetermined system of equations (4b) while avoiding that $\boldsymbol{u} \notin \Omega$ although $\boldsymbol{v} \in \Phi$ [13].

II. Input matrix factorization

There are infinitely many choices of $\boldsymbol{B_v}$ and $\boldsymbol{B}$ that fulfill (3) for each input matrix $\boldsymbol{B_u}$. For $n = k$ the standard approach is selecting $\boldsymbol{B_v} = \boldsymbol{I_k}$ and $\boldsymbol{B} = \boldsymbol{B_u}$ where $\boldsymbol{I_k} \in \mathbb{R}^{k\times k}$ is the identity matrix. In case of $n > k$ it is not always so obvious how to accomplish (3). Singular value decomposition (SVD) and QR factorization, to name but a few, are possible options. Transformation matrices are the basis of the analysis of the factorization's impact on the CA methods. It is shown in [13] that for any pair of input matrix factorizations

$$\boldsymbol{B_u} = \boldsymbol{B_{v1}}\boldsymbol{B_1} = \boldsymbol{B_{v2}}\boldsymbol{B_2} \tag{7}$$

an invertible transformation matrix

$$\boldsymbol{T} = \left(\boldsymbol{B}_{\boldsymbol{v1}}^T \boldsymbol{B}_{\boldsymbol{v1}}\right)^{-1} \boldsymbol{B}_{\boldsymbol{v1}}^T \boldsymbol{B}_{\boldsymbol{v2}} \tag{8}$$

with $\boldsymbol{T} \in \mathbb{R}^{k \times k}$ exists, such that

$$\boldsymbol{B}_{\boldsymbol{v1}} \boldsymbol{T} = \boldsymbol{B}_{\boldsymbol{v2}} \tag{9}$$

and

$$\boldsymbol{T}^{-1} \boldsymbol{B_1} = \boldsymbol{B_2}. \tag{10}$$

High-level control has to ensure that the states of system (1) follow the desired trajectory no matter which factorization method has been chosen. It is shown in [13] that for different input matrix factorizations identical state vectors $\boldsymbol{x_1}(t) \equiv \boldsymbol{x_2}(t)$ are achieved if

$$\boldsymbol{v_2}(t) = \boldsymbol{T}^{-1} \boldsymbol{v_1}(t) \quad \forall t. \tag{11}$$

Each input matrix factorization has its own attainable moment set

$$\Phi_i = \left\{\boldsymbol{v_i} \in \mathbb{R}^k \middle| \exists \boldsymbol{u} \in \Omega : \boldsymbol{v_i} = \boldsymbol{B_i u}\right\} \;\; \forall i \in \{1, 2\}, \tag{12}$$

whereas the admissible control space Ω stays the same. These facts are exploited in the subsequent sections.

III. Direct Allocation

Direct allocation (DA) uses geometric principles to compute $\boldsymbol{u}$ from a given virtual control vector $\boldsymbol{v}$. Its main advantage is its ability to determine a $\boldsymbol{u} \in \Omega$ for every $\boldsymbol{v} \in \Phi$, i.e. it facilitates the usage of the entire AMS. Figures 1 and 2 give a visual interpretation of the DA problem in case of $m = 3$ and $k = 2$. The following requirements have to be met [5]:

- A half-line starting in the origin of Φ intersects its boundary $\partial(\Phi)$ in a single point, see Figure 2.
- Every point on $\partial(\Phi)$ uniquely maps to one point on $\partial(\Omega)$ (the boundary of Ω) in each case, i.e.

$$\boldsymbol{v}^* \in \partial\left(\Phi\right) \Rightarrow \exists! \boldsymbol{u}^* \in \partial\left(\Omega\right) \,:\, \boldsymbol{v}^* = B\boldsymbol{u}^*. \tag{13}$$

At first DA computes the intersection of vector $\boldsymbol{v}$ with the boundary $\partial(\Phi)$. This solution $\boldsymbol{v}^*$ is always located on a (k-1)-D object and is the image of one point $\boldsymbol{u}^*$ on $\partial(\Omega)$. Now $\boldsymbol{v}^*$ is scaled by a factor $0 < a \in \mathbb{R}$ to match $\boldsymbol{v}$. In order to get $\boldsymbol{u}$ the same scaling can be applied to $\boldsymbol{u}^*$ because of the linearity of the problem

$$\boldsymbol{v} = a\boldsymbol{v}^* = aB\boldsymbol{u}^* = B(a\boldsymbol{u}^*). \tag{14}$$

If $a > 1$, the desired virtual control lies outside of Φ. In this case the boundary-intersecting solution $\boldsymbol{u}^*$ is chosen, which preserves the desired virtual control direction ([4], [5])

$$\boldsymbol{u} = \alpha \boldsymbol{u}^* \quad \text{with} \quad \alpha = \begin{cases} a & \text{if } 0 < a \leq 1 \\ 1 & \text{else} \end{cases}. \tag{15}$$

Theorem 1: For any pair of input matrix factorizations (7) direct allocation yields the same input vectors $\boldsymbol{u_1} \equiv \boldsymbol{u_2}$ if the virtual control vectors $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$ fulfill (11).

Proof: Each factorization has its own boundary of the related attainable moment set

$$\partial(\Phi_i) = \{\boldsymbol{v_i} \in \Phi_i | (b\boldsymbol{v_i} \notin \Phi_i) \, \forall \, (b > 1)\} \;\; \forall i \in \{1, 2\} \tag{16}$$

with $b \in \mathbb{R}$. For one factorization the boundary-intersections $\boldsymbol{v_1^*}$ and $\boldsymbol{u_1^*}$ fulfill

$$\boldsymbol{v_1} = a_1 \boldsymbol{v_1^*} = a_1 \boldsymbol{B_1 u_1^*}. \tag{17}$$

Using (11) the virtual control vector $\boldsymbol{v_2}$ can be expressed as

$$\boldsymbol{v_2} = \boldsymbol{T}^{-1}\boldsymbol{v_1} = a_1 \boldsymbol{T}^{-1}\boldsymbol{v_1^*} = a_1 \underbrace{\boldsymbol{T}^{-1}\boldsymbol{B_1}}_{\boldsymbol{B_2}} \boldsymbol{u_1^*} \tag{18}$$

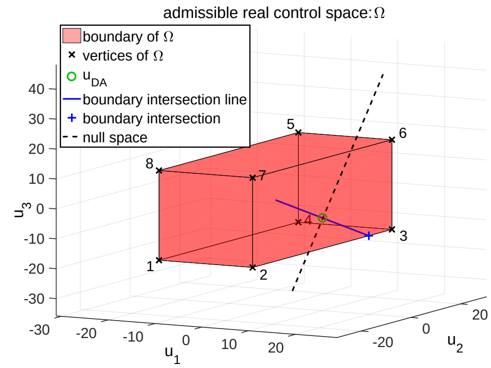

Fig. 1. DA computes the solution $\boldsymbol{u_{DA}}$ (green 'o', inside the box) for the desired virtual control $\boldsymbol{v_{des}}$ (see Figure 2) by scaling down the boundary intersection (blue '+'). Moving along the (m-k)-D nullspace direction (black dashed line) does not change the resulting virtual control.

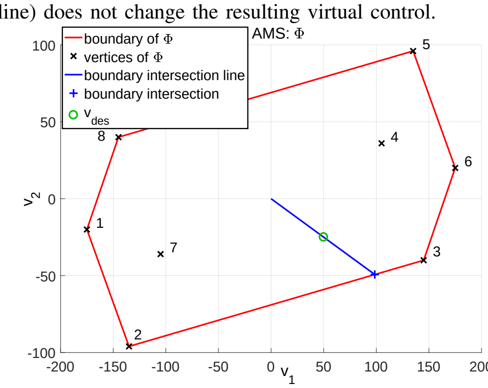

Fig. 2. 2-D virtual control space related to Figure 1: The numbered black 'x' represent the vertices of Ω. Some of them are not part of $\partial(\Phi)$. For $\boldsymbol{v_{des}} = [50 \; \text{-}25]^T$ DA computes the intersection with $\partial(\Phi)$ and only checks those (k-1)-D objects of $\partial(\Omega)$ for intersection which also belong to $\partial(\Phi)$.

in case of another factorization of the input matrix. Due to the dimension reduction which takes place by the mapping from Ω to Φ_2 not all elements on $\partial(\Omega)$ lie on $\partial(\Phi_2)$ but rather in the interior of Φ_2. However, only if $\boldsymbol{B_2 u_1^*} = \boldsymbol{T}^{-1}\boldsymbol{v_1^*} \in \partial(\Phi_2)$ DA will compute the same real control vector for both factorizations. Let $\boldsymbol{v_1^*}$ be located on $\partial(\Phi_1)$, i.e.

$$\boldsymbol{v_1^*} \in \partial(\Phi_1) \quad \Rightarrow \quad \forall \, b > 1 \,:\, b\boldsymbol{v_1^*} \notin \Phi_1. \tag{19}$$

Assuming that $\boldsymbol{T}^{-1}\boldsymbol{v_1^*}$ is in the interior of Φ_2 means that $\boldsymbol{T}^{-1}\boldsymbol{v_1^*} \in \Phi_2$ and $\boldsymbol{T}^{-1}\boldsymbol{v_1^*} \notin \partial(\Phi_2)$, which is equivalent to

$$\exists \, b > 1 \,:\, b\boldsymbol{T}^{-1}\boldsymbol{v_1^*} \in \Phi_2. \tag{20}$$

Utilizing (12) to rewrite (20) yields

$$\exists \, b > 1, \, \exists (b\boldsymbol{u}) \in \Omega \,:\, b\boldsymbol{T}^{-1}\boldsymbol{v_1^*} = b\boldsymbol{B_2 u}. \tag{21}$$

Now one can see from (10) that (21) is equivalent to $\exists \, b > 1, \, \exists (b\boldsymbol{u}) \in \Omega \,:\, b\boldsymbol{v_1^*} = b\boldsymbol{B_1 u}$, which means that $\boldsymbol{v_1^*}$ lies in the interior of Φ_1, i.e. $\exists \, b > 1 \,:\, b\boldsymbol{v_1^*} \in \Phi_1$ and this is contradictory to (19). Therefore $\boldsymbol{T}^{-1}\boldsymbol{v_1^*}$ is indeed on $\partial\left(\Phi_2\right)$ and so it follows from (13) that DA will yield the same $\boldsymbol{u}$ for all factorizations. ■

IV. Recap on Generalized Inverses

As generalized inverses provide the foundation for RPINV this section summarizes some of their properties. For every matrix $\boldsymbol{B} \in \mathbb{R}^{k \times m}$ there exist generalized inverses (also known as pseudoinverses) $\boldsymbol{P} \in \mathbb{R}^{m \times k}$ satisfying at least one of the four so-called Penrose-equations [15]. Since $\boldsymbol{B}$ is assumed to have full row rank such a matrix $\boldsymbol{P}$ is a right-inverse of $\boldsymbol{B}$, i.e. $\boldsymbol{BP} = \boldsymbol{I_k}$ [15]. A solution to the control allocation problem is

$$\boldsymbol{u} = \boldsymbol{Pv}, \tag{22}$$

whereby those elements of $\boldsymbol{u}$ which exceed Ω are set to their extremal values ([4] and [5]). Assuming two input matrix factorizations (7) and a generalized inverse $\boldsymbol{P_1}$ for factorization 1 are known, the corresponding generalized inverse for factorization 2 is given by [13]

$$\boldsymbol{P_2} = \boldsymbol{P_1 T} \tag{23}$$

with transformation matrix $\boldsymbol{T}$ according to (8). It is evident that if the virtual control vectors $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$ satisfy (11) and the generalized inverses fulfill (23) then (22) yields identical input vectors $\boldsymbol{u_1} \equiv \boldsymbol{u_2}$ for both factorizations [13]. Thus CA via generalized inverses is not affected by input matrix factorization.

The unconstrained control allocation problem (4b) has infinitely many solutions and a reasonable choice is to pick that with the lowest energy consumption (least-norm solution [16], [17]). Using a constant positive definite weighting matrix $\boldsymbol{W} \succ 0$ and a constant offset vector $\boldsymbol{c} \in \mathbb{R}^m$, an optimization problem can be formulated ([1], [18])

$$\min_{\boldsymbol{u}} \; (\boldsymbol{u}+\boldsymbol{c})^T \, \boldsymbol{W} \, (\boldsymbol{u}+\boldsymbol{c}) \, , \tag{24}$$

which is subject to equality constraint (4b). The closed-form solution is derived with the method of Lagrange multipliers and reads as

$$\boldsymbol{u} = -\boldsymbol{c} + \boldsymbol{B}^{\#} \left(\boldsymbol{v} + \boldsymbol{B c} \right) \tag{25a}$$

$$\boldsymbol{B}^{\#} = \boldsymbol{W}^{-1} \boldsymbol{B}^T \left(\boldsymbol{B W}^{-1} \boldsymbol{B}^T \right)^{-1} . \tag{25b}$$

$\boldsymbol{B}^{\#} \in \mathbb{R}^{m \times k}$ is called weighted pseudoinverse and if $\boldsymbol{W} = \boldsymbol{I_m}$ then $\boldsymbol{B}^{\#} = \boldsymbol{B}^{\ddagger}$ is the Moore-Penrose pseudoinverse (MPP) [16].

V. Redistributed Pseudoinverse

Redistributed Pseudoinverse (RPINV) is an iterative process that takes the actuator constraints into account. During the first step the unconstrained solution according to the conventional weighted pseudoinverse (25b) is calculated. If no controls exceed their limits, no further steps are required. Otherwise, the actuators that violate the constraints are fixed at their extremal values (saturated) and a reduced pseudoinverse is computed for the remaining free controls. This procedure is repeated until no new constraint violations occur or all actuators are at their limits ([1], [18]). Let $\boldsymbol{B_0}$ be the control effectivity matrix, $\boldsymbol{u}^N$ the result of the N-th iteration, and $\boldsymbol{u}^0 = \boldsymbol{0}$. Constraint violations are recorded in an offset vector

$$\boldsymbol{c}^N = [c_1^N \; ... \; c_m^N]^T \tag{26a}$$

$$c_i^N = \begin{cases} -u_{max,i} & \text{if } u_i^{N-1} \geq u_{max,i} \\ -u_{min,i} & \text{if } u_i^{N-1} \leq u_{min,i} \\ 0 & \text{else} \end{cases} \quad \text{with } i = 1, ..., m \tag{26b}$$

the ascendingly ordered set of free actuator indices reads as

$$J^N = \{ l \in \mathbb{N}^+ | c_l^N = 0 \} = \{ l_1, ..., l_j \} \tag{27}$$

and $j = |J^N|$ denotes its number. Using the i-th unit vector $\boldsymbol{e_i} \in \mathbb{R}^m$ one can define the column-selection matrix

$$\boldsymbol{R} = \left[\boldsymbol{e_{l_1}} \; ... \; \boldsymbol{e_{l_j}} \right] . \tag{28}$$

By means of (28) the modified control effectivity matrix with columns related to saturated controls set to zero is given by

$$\boldsymbol{B}^N = \boldsymbol{B_0 R R}^T . \tag{29}$$

The result of the N-th iteration of RPINV reads as

$$\boldsymbol{u}^N = -\boldsymbol{c}^N + \boldsymbol{B}^{\#} \left(\boldsymbol{v} + \boldsymbol{B_0 c}^N \right) . \tag{30}$$

Remark 1: The resulting virtual control after the application of RPINV and neglecting potentially exceeded constraints is denoted as $\boldsymbol{v_{act}}$. Considering $\boldsymbol{v} = \boldsymbol{B_0 u}^N$ and (30) one obtains

$$\boldsymbol{v_{act}} = -\boldsymbol{B_0 c}^N + \boldsymbol{B_0 B}^{\#} \boldsymbol{v_{des}} + \boldsymbol{B_0 B}^{\#} \boldsymbol{B_0 c}^N \tag{31}$$

where $\boldsymbol{v_{des}}$ is the desired value coming from the controller. One realizes from (31) that $\boldsymbol{v_{des}}$ can be reached in principle if $\boldsymbol{B_0 B}^{\#} = \boldsymbol{I_k}$. Inserting (29) into (25b) yields $\boldsymbol{B}^{\#} = \boldsymbol{W}^{-1} \boldsymbol{R R}^T \boldsymbol{B_0}^T \left(\boldsymbol{B_0 R R}^T \boldsymbol{W}^{-1} \boldsymbol{R R}^T \boldsymbol{B_0}^T \right)^{-1}$. In order to accomplish $\boldsymbol{B B}^{\#} = \boldsymbol{B_0 B}^{\#} = \boldsymbol{I_k}$ the computation is altered into

$$\boldsymbol{B}^{\#} = \boldsymbol{R R}^T \boldsymbol{W}^{-1} \boldsymbol{R R}^T \boldsymbol{B_0}^T \left(\boldsymbol{B_0 R R}^T \boldsymbol{W}^{-1} \boldsymbol{R R}^T \boldsymbol{B_0}^T \right)^{-1} . \tag{32}$$

A. Factorization influence

Whether the factorization affects RPINV is related with a possible decrease of the input matrix's rank.

Assumption 1: Every $k \times k$ submatrix of $\boldsymbol{B_0}$ is full rank.

Assumption 2: The number of free controls satisfies $j \geq k$.

Theorem 2: Under Assumptions 1 and 2 RPINV yields the same input vectors $\boldsymbol{u_1} \equiv \boldsymbol{u_2}$ for any pair of input matrix factorizations (7) if the virtual control vectors $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$ fulfill (11).

Proof: During the first iteration (unconstrained solution) $\boldsymbol{c}^1$ is zero for both factorizations. Considering (10), (29), and (32) the weighted pseudoinverse for factorization 2 reads as

$$\begin{aligned} \boldsymbol{B_2^{\#}} &= \boldsymbol{R R}^T \boldsymbol{W}^{-1} \boldsymbol{B_1}^{N,T} \boldsymbol{T}^{-T} \left[\boldsymbol{T}^{-1} \boldsymbol{B_1}^N \boldsymbol{W}^{-1} \boldsymbol{B_1}^{N,T} \boldsymbol{T}^{-T} \right]^{-1} \\ &= \boldsymbol{R R}^T \boldsymbol{W}^{-1} \boldsymbol{B_1}^{N,T} \left(\boldsymbol{B_1}^N \boldsymbol{W}^{-1} \boldsymbol{B_1}^{N,T} \right)^{-1} \boldsymbol{T} = \boldsymbol{B_1^{\#} T} \end{aligned} \tag{33}$$

and so $\boldsymbol{u_1}^1 \equiv \boldsymbol{u_2}^1$ follows from (11). Assume that $j \geq k$ actuators remain free in iteration $N \geq 1$. Because of $\boldsymbol{u_1}^{N-1} = \boldsymbol{u_2}^{N-1}$ this leads to the same changes in $\boldsymbol{c}^N$ and $\boldsymbol{B_i}^N$ for both factorizations $i \in \{1, 2\}$ and $k = \mathrm{rk}(\boldsymbol{B_i}^N) = \min(k, j)$ still holds. Hence $\boldsymbol{B_i}^N \boldsymbol{W}^{-1} \boldsymbol{B_i}^{N,T}$ is still invertible and (33) can be applied again to show that $\boldsymbol{u_1}^N = \boldsymbol{u_2}^N$. ■

Remark 2: Assumption 1 guarantees that $\mathrm{rk}(\boldsymbol{B}^N) = k$ as long as it contains k nonzero columns. If this assumption is not fulfilled, zeroing columns can lead to a rank-reduction of $\boldsymbol{B}^N$ although Assumption 2 holds. Whether Theorem 2 holds depends in this case not only on the number of saturated actuators but also on which of them saturate.

In order to analyze the case of $j < k$ assume that $\boldsymbol{W} = \boldsymbol{I_m}$ which causes $\boldsymbol{B}^{\#}$ to become the MPP[2], i.e. $\boldsymbol{B}^{\#} = \boldsymbol{B}^{\ddagger} = \boldsymbol{B}^T (\boldsymbol{B B}^T)^{-1}$. Due to numerical reasons this calculation is not carried out directly, but instead the SVD is used [16]. Consider matrix $\boldsymbol{B}$ as

$$\boldsymbol{B} = \begin{bmatrix} b_{11} & \cdots & b_{1j} & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & & \vdots \\ b_{k1} & \cdots & b_{kj} & 0 & \cdots & 0 \end{bmatrix} \tag{34}$$

with j being the number of nonzero columns[3] and singular values. The SVD yields $\boldsymbol{B} = \boldsymbol{U S V}^T$ with the orthonormal columns of $\boldsymbol{U} \in \mathbb{R}^{k \times k}$ being the eigenvectors of $\boldsymbol{B B}^T$,

$$\boldsymbol{S} = \begin{bmatrix} \sigma_1 & & & 0 & \cdots & 0 \\ & \ddots & & \vdots & & \vdots \\ & & \sigma_k & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{k \times m} \tag{35}$$

containing the singular values σ_1,...,σ_k of $\boldsymbol{B}$ (note that only the first j of them are nonzero), and the orthogonal matrix $\boldsymbol{V} \in \mathbb{R}^{m \times m}$. An important observation is that in the first j columns of $\boldsymbol{V}$ the last $m - j$ rows are zero, because these columns are the eigenvectors of the nonzero eigenvalues of $\boldsymbol{B}^T \boldsymbol{B}$, whose $m - j$ last rows are zero. The MPP of $\boldsymbol{B}$ reads as

$$\boldsymbol{B}^{\ddagger} = \boldsymbol{V \Sigma U}^T \tag{36}$$

[2] A generalization for arbitrary $\boldsymbol{W} \succ 0$ is presented later on. In order to keep notations concise superscript 'N' in $\boldsymbol{B}^N$ is omitted from now on.

[3] W.l.o.g. one can assume the nonzero columns to be the first ones, because columns of $\boldsymbol{B}$ and related elements in $\boldsymbol{u}$ may be appropriately interchanged.

with

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^{-1} & & & \\ & \ddots & & \\ & & \sigma_j^{-1} & \\ & & & \ddots \\ 0 & & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{m\times k}. \tag{37}$$

One can see from (37) that $\boldsymbol{B}^\ddagger$ is no longer a right-inverse because $\text{rk}(\boldsymbol{\Sigma}) < k$. Matrix $\boldsymbol{\Sigma}$ has $m-j$ zero rows which means together with (36) that the last $m-j$ columns of $\boldsymbol{V}$ do not contribute to $\boldsymbol{B}^\ddagger$. Evaluating (36) and considering these insights yields

$$\boldsymbol{B}^\ddagger = \begin{bmatrix} \sigma_1^{-1}v_{11} & \cdots & \sigma_j^{-1}v_{1j} & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & & \vdots \\ \sigma_1^{-1}v_{j1} & \cdots & \sigma_j^{-1}v_{jj} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix} \cdot \boldsymbol{U}^T \tag{38}$$

with v_{pq} being the element from $\boldsymbol{V}$'s p-th row and q-th column. Equation (38) can be rewritten to

$$\boldsymbol{B}^\ddagger = \boldsymbol{R} \begin{bmatrix} v_{11} & \cdots & v_{1j} \\ \vdots & & \vdots \\ v_{j1} & \cdots & v_{jj} \end{bmatrix} \begin{bmatrix} \sigma_1^{-1} & & & 0 & \cdots & 0 \\ & \ddots & & \vdots & & \vdots \\ & & \sigma_j^{-1} & 0 & \cdots & 0 \end{bmatrix} \boldsymbol{U}^T \tag{39}$$

with $\boldsymbol{R} \in \mathbb{R}^{m\times j}$ being consistent with (28). Expression (39) can also be evaluated if $\text{rk}(\boldsymbol{B}) < k$, i.e. less than k actuators are free[4]. Assuming $j < k$ and taking all nonzero columns from (34), one obtains

$$\widetilde{\boldsymbol{B}} = \boldsymbol{B}\boldsymbol{R} = \begin{bmatrix} b_{11} & \cdots & b_{1j} \\ \vdots & & \vdots \\ b_{k1} & \cdots & b_{kj} \end{bmatrix} \in \mathbb{R}^{k\times j} \tag{40}$$

with $\text{rk}(\widetilde{\boldsymbol{B}}) = j$. Applying the SVD on $\widetilde{\boldsymbol{B}}$ results in $\widetilde{\boldsymbol{B}} = \widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{S}}\widetilde{\boldsymbol{V}}^T$ with $\widetilde{\boldsymbol{U}} \in \mathbb{R}^{k\times k}$, $\widetilde{\boldsymbol{S}} \in \mathbb{R}^{k\times j}$, and $\widetilde{\boldsymbol{V}} \in \mathbb{R}^{j\times j}$. Note that the zero columns of $\boldsymbol{B}$ do not affect $\boldsymbol{U}$, because $\boldsymbol{B}\boldsymbol{B}^T = \widetilde{\boldsymbol{B}}\widetilde{\boldsymbol{B}}^T$. This implies that $\boldsymbol{U} = \widetilde{\boldsymbol{U}}$ and the nonzero singular values of $\boldsymbol{B}$ and $\widetilde{\boldsymbol{B}}$ are the same. Furthermore, it can be shown that the elements of $\widetilde{\boldsymbol{V}}$ coincide with those from the top-left $j\times j$-submatrix of $\boldsymbol{V}$. The MPP of $\widetilde{\boldsymbol{B}}$ is now a left inverse and reads as $\widetilde{\boldsymbol{B}}^\dagger = (\widetilde{\boldsymbol{B}}^T\widetilde{\boldsymbol{B}})^{-1}\widetilde{\boldsymbol{B}}^T = \widetilde{\boldsymbol{V}}\widetilde{\boldsymbol{\Sigma}}\widetilde{\boldsymbol{U}}^T$ [16] or more specifically

$$\widetilde{\boldsymbol{B}}^\dagger = \begin{bmatrix} v_{11} & \cdots & v_{1j} \\ \vdots & & \vdots \\ v_{j1} & \cdots & v_{jj} \end{bmatrix} \begin{bmatrix} \sigma_1^{-1} & & & 0 & \cdots & 0 \\ & \ddots & & \vdots & & \vdots \\ & & \sigma_j^{-1} & 0 & \cdots & 0 \end{bmatrix} \boldsymbol{U}^T. \tag{41}$$

Comparing (39) and (41) one realizes that if $j < k$ then

$$\boldsymbol{B}^\ddagger = \boldsymbol{R}\widetilde{\boldsymbol{B}}^\dagger, \tag{42}$$

i.e. the pseudoinverse of $\boldsymbol{B}^N$ is the left inverse of its nonzero columns augmented with rows of zeros.

Now an arbitrary weighting matrix $\boldsymbol{W} \succ 0$ is considered. By means of Cholesky decomposition $\boldsymbol{W}$ can be factorized into $\boldsymbol{W} = \widehat{\boldsymbol{W}}^T\widehat{\boldsymbol{W}}$ and $\widehat{\boldsymbol{W}} \in \mathbb{R}^{m\times m}$ is an upper triangular matrix with positive diagonal entries [14]. Assuming $j \geq k$ an auxiliary matrix is introduced as

$$\boldsymbol{B_W} = \boldsymbol{B_0}\boldsymbol{R}\boldsymbol{R}^T\widehat{\boldsymbol{W}}^{-1} \tag{43}$$

[4]In the conventional formula $(\boldsymbol{B}\boldsymbol{B}^T)^{-1}$ does not exist any more.

with $\text{rk}(\boldsymbol{B_W}) = k$ following from (67) in Appendix A. Its MPP is

$$\boldsymbol{B}^\ddagger_{\boldsymbol{W}} = \widehat{\boldsymbol{W}}^{-T}\boldsymbol{R}\boldsymbol{R}^T\boldsymbol{B_0}^T\Big(\boldsymbol{B_0}\boldsymbol{R}\boldsymbol{R}^T\underbrace{\widehat{\boldsymbol{W}}^{-1}\widehat{\boldsymbol{W}}^{-T}}_{(\widehat{\boldsymbol{W}}^T\widehat{\boldsymbol{W}})^{-1}=\boldsymbol{W}^{-1}}\boldsymbol{R}\boldsymbol{R}^T\boldsymbol{B_0}^T\Big)^{-1}. \tag{44}$$

Comparing (32) and (44) it follows that the weighted pseudoinverse of $\boldsymbol{B}$ can be computed by

$$\boldsymbol{B}^\# = \boldsymbol{R}\boldsymbol{R}^T\widehat{\boldsymbol{W}}^{-1}\boldsymbol{B}^\ddagger_{\boldsymbol{W}}. \tag{45}$$

Lemma 1: In case of $j < k$ the weighting matrix has no influence on the resulting pseudoinverse.

Proof: Combining (40) and (43) yields another auxiliary matrix

$$\widetilde{\boldsymbol{B}}_{\boldsymbol{W}} = \boldsymbol{B_W}\boldsymbol{R} \tag{46}$$

with $\widetilde{\boldsymbol{B}}_{\boldsymbol{W}} \in \mathbb{R}^{k\times j}$. Its MPP reads as $\widetilde{\boldsymbol{B}}^\dagger_{\boldsymbol{W}} = \left(\boldsymbol{R}^T\widehat{\boldsymbol{W}}^{-T}\boldsymbol{R}\boldsymbol{R}^T\boldsymbol{B_0}^T\boldsymbol{B_0}\boldsymbol{R}\boldsymbol{R}^T\widehat{\boldsymbol{W}}^{-1}\boldsymbol{R}\right)^{-1}\boldsymbol{R}^T\widehat{\boldsymbol{W}}^{-T}\boldsymbol{R}\boldsymbol{R}^T\boldsymbol{B_0}^T$. Right-multiplying a matrix with $\boldsymbol{R}$ selects according to (27) columns with indices $\{l_1, ..., l_j\}$ and left-multiplying with $\boldsymbol{R}^T$ selects rows with identical indices. Therefore $\boldsymbol{R}^T\widehat{\boldsymbol{W}}^{-T}\boldsymbol{R}$ and $\boldsymbol{R}^T\widehat{\boldsymbol{W}}^{-1}\boldsymbol{R}$ are triangular matrices with positive diagonal entries which guarantees invertibility. It follows that

$$\widetilde{\boldsymbol{B}}^\dagger_{\boldsymbol{W}} = \left(\boldsymbol{R}^T\widehat{\boldsymbol{W}}^{-1}\boldsymbol{R}\right)^{-1}\left(\widetilde{\boldsymbol{B}}^T\widetilde{\boldsymbol{B}}\right)^{-1}\widetilde{\boldsymbol{B}}^T \tag{47}$$

and together with (42) and (45) one obtains

$$\begin{aligned} \boldsymbol{B}^\# &= \boldsymbol{R}\left(\boldsymbol{R}^T\widehat{\boldsymbol{W}}^{-1}\boldsymbol{R}\right)\left(\boldsymbol{R}^T\widehat{\boldsymbol{W}}^{-1}\boldsymbol{R}\right)^{-1}\left(\widetilde{\boldsymbol{B}}^T\widetilde{\boldsymbol{B}}\right)^{-1}\widetilde{\boldsymbol{B}}^T \\ &= \boldsymbol{R}\left(\widetilde{\boldsymbol{B}}^T\widetilde{\boldsymbol{B}}\right)^{-1}\widetilde{\boldsymbol{B}}^T = \boldsymbol{R}\widetilde{\boldsymbol{B}}^\dagger = \boldsymbol{B}^\ddagger \end{aligned}. \tag{48}$$

■

Assumption 3: Two input matrix factorizations (7) are related by means of a transformation matrix $\boldsymbol{T}$ with *orthogonal columns*, i.e.

$$\boldsymbol{T}\boldsymbol{T}^T = \boldsymbol{I_k}d \tag{49}$$

with $d \in \mathbb{R}\setminus\{0\}$.

Theorem 3: Under Assumption 3 RPINV yields identical input vectors $\boldsymbol{u_1} \equiv \boldsymbol{u_2}$ for any $j \geq 0$ if $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$ fulfill (11).

Proof: If $j \geq k$ Theorem 2 holds. In case of $j < k$ Lemma 1 enables to restrict considerations on $\boldsymbol{W} = \boldsymbol{I_m}$. Using (10) leads to $\widetilde{\boldsymbol{B}}_{\boldsymbol{2}} = \boldsymbol{T}^{-1}\widetilde{\boldsymbol{B}}_{\boldsymbol{1}}$ and according to (48) the rank deficient pseudoinverse for the second factorization is

$$\boldsymbol{B}^\#_{\boldsymbol{2}} = \boldsymbol{R}\left[\widetilde{\boldsymbol{B}}_{\boldsymbol{1}}^T\boldsymbol{T}^{-T}\boldsymbol{T}^{-1}\widetilde{\boldsymbol{B}}_{\boldsymbol{1}}\right]^{-1}\widetilde{\boldsymbol{B}}_{\boldsymbol{1}}^T\boldsymbol{T}^{-T}. \tag{50}$$

Considering (49) yields $\boldsymbol{T}^{-T}\boldsymbol{T}^{-1} = \boldsymbol{I_k}\frac{1}{d}$ and taking that out of the bracket one obtains $\boldsymbol{I_k}d\boldsymbol{T}^{-T} = \boldsymbol{T}$. Consequently $\boldsymbol{B}^\#_{\boldsymbol{2}} = \boldsymbol{B}^\#_{\boldsymbol{1}}\boldsymbol{T}$ and together with (11) $\boldsymbol{u_1} \equiv \boldsymbol{u_2}$ follows. ■

B. Optimal factorization

Suppose $j < k$ and $\boldsymbol{W} = \boldsymbol{I_m}$ (see Lemma 1). Due to $\boldsymbol{B}^\#$ lacking full column rank $\boldsymbol{v_{des}}$ will generally not be reached any more because $\boldsymbol{B_0}\boldsymbol{B}^\# \neq \boldsymbol{I_k}$. Of course this raises the question whether there is an optimal factorization of $\boldsymbol{B_u}$ which makes the error between $\boldsymbol{v_{des}}$ and $\boldsymbol{v_{act}}$ as small as possible. But actually it is more reasonable to minimize the deviation between desired and actual effect on the plant: $\boldsymbol{e_v} = \boldsymbol{B_v}\left(\boldsymbol{v_{des}} - \boldsymbol{v_{act}}\right)$. Using (10), (11), (31) and (50) one can formulate the optimization problem

$$\min_{\boldsymbol{T}} \left\|\boldsymbol{e_v}\right\| = \min_{\boldsymbol{T}} \big\|\boldsymbol{M_T}\underbrace{\left(\boldsymbol{v_{des}} + \boldsymbol{B_0}\boldsymbol{c}\right)}_{\tilde{\boldsymbol{v}}}\big\| \tag{51a}$$

$$\boldsymbol{M_T} = \boldsymbol{B_v}\boldsymbol{T}\left(\boldsymbol{I_k} - \boldsymbol{T}^{-1}\boldsymbol{B_0}\boldsymbol{B}^\#_{\boldsymbol{T}}\right)\boldsymbol{T}^{-1} \tag{51b}$$

$$\boldsymbol{B}^\#_{\boldsymbol{T}} = \boldsymbol{R}\left(\widetilde{\boldsymbol{B}}^T\boldsymbol{T}^{-T}\boldsymbol{T}^{-1}\widetilde{\boldsymbol{B}}\right)^{-1}\widetilde{\boldsymbol{B}}^T\boldsymbol{T}^{-T} \tag{51c}$$

$$\boldsymbol{u_{min}} \leq -\boldsymbol{c} + \boldsymbol{B}^\#_{\boldsymbol{T}}\boldsymbol{T}^{-1}\left(\boldsymbol{v_{des}} + \boldsymbol{B_0}\boldsymbol{c}\right) \leq \boldsymbol{u_{max}} \tag{51d}$$

If $\tilde{\boldsymbol{v}} \in \mathcal{N}_{\mathrm{r}}(\boldsymbol{M_T})$, the desired value $\boldsymbol{v_{des}}$ could be reached, although $\boldsymbol{B_0}\boldsymbol{B}^{\#} \neq \boldsymbol{I_k}$ provided that $\boldsymbol{v_{des}} \in \Phi$. It turns out that both $\boldsymbol{M_T}$ (because of $\widetilde{\boldsymbol{B}}$) and $\tilde{\boldsymbol{v}}$ depend on $\boldsymbol{v_{des}}$ and so there is no universally best factorization that could be computed offline. Problem (51) is a nonlinear constrained optimization problem which makes online solving quite sophisticated. RPINV is intended to be a low-effort method for considering actuator constraints and so this is not a reasonable option.

C. Effect prioritization

Instead of minimizing the total effect error $\boldsymbol{e_v}$ it is easier to focus on certain components. The basic idea of the approach is to exploit the rank deficiency of $\boldsymbol{M_T}$ by introducing zero rows. In this subsection $n = k$ and w.l.o.g. $\boldsymbol{B_v} = \boldsymbol{I_k}$ are assumed and therefore (51b) simplifies to

$$\boldsymbol{M_T} = \boldsymbol{I_k} - \boldsymbol{B_0}\boldsymbol{B_T^{\#}}\boldsymbol{T}^{-1}. \tag{52}$$

Lemma 2: For all $j < k$ and all invertible transformations $\boldsymbol{T} \in \mathbb{R}^{k\times k}$ the rank of (52) is $k - j$.

Proof: Define $\widetilde{\boldsymbol{B}}_{\boldsymbol{T}} = \boldsymbol{T}^{-1}\widetilde{\boldsymbol{B}}$ and (67) in Appendix A yields $\mathrm{rk}(\widetilde{\boldsymbol{B}}_{\boldsymbol{T}}) = j$. According to (48) one obtains $\boldsymbol{B_T^{\#}} = \boldsymbol{R}\widetilde{\boldsymbol{B}}_{\boldsymbol{T}}^{\dagger}$. From Lemma 1 of [15] it follows that $\mathrm{rk}(\widetilde{\boldsymbol{B}}_{\boldsymbol{T}}^{\dagger}) = \mathrm{rk}(\widetilde{\boldsymbol{B}}_{\boldsymbol{T}}) = j$ and repeatedly applying (67) results in

$$j = \mathrm{rk}(\boldsymbol{R}\widetilde{\boldsymbol{B}}_{\boldsymbol{T}}^{\dagger}) = \mathrm{rk}(\boldsymbol{B_T^{\#}}\boldsymbol{T}^{-1}). \tag{53}$$

Note that $\boldsymbol{B_0}\boldsymbol{B_T^{\#}}\boldsymbol{T}^{-1} = \widetilde{\boldsymbol{B}}\widetilde{\boldsymbol{B}}_{\boldsymbol{T}}^{\dagger}\boldsymbol{T}^{-1}$ and once again (67) reveals that

$$\mathrm{rk}(\boldsymbol{B_0}\boldsymbol{B_T^{\#}}\boldsymbol{T}^{-1}) = j. \tag{54}$$

Appendix B provides the rank of (52) by means of (68). Recall that $\boldsymbol{B_0}\boldsymbol{B_T^{\#}} = \widetilde{\boldsymbol{B}}\widetilde{\boldsymbol{B}}_{\boldsymbol{T}}^{\dagger}$ and $\boldsymbol{I_k^{\dagger}} = \boldsymbol{I_k}$. Thus $\boldsymbol{B_0}\boldsymbol{B_T^{\#}}\boldsymbol{T}^{-1}\boldsymbol{I_k^{\dagger}}\boldsymbol{B_0}\boldsymbol{B_T^{\#}}\boldsymbol{T}^{-1} = \widetilde{\boldsymbol{B}}\underbrace{\widetilde{\boldsymbol{B}}_{\boldsymbol{T}}^{\dagger}\boldsymbol{T}^{-1}\widetilde{\boldsymbol{B}}}_{\boldsymbol{I_j}}\widetilde{\boldsymbol{B}}_{\boldsymbol{T}}^{\dagger}\boldsymbol{T}^{-1} = \boldsymbol{B_0}\boldsymbol{B_T^{\#}}\boldsymbol{T}^{-1}$.

Let $\overline{\boldsymbol{C}} = [\boldsymbol{I_k}\ \boldsymbol{B_0}\boldsymbol{B_T^{\#}}\boldsymbol{T}^{-1}] \in \mathbb{R}^{k\times 2k}$, $\underline{\boldsymbol{C}} = \begin{bmatrix}\boldsymbol{I_k}\\ \boldsymbol{B_0}\boldsymbol{B_T^{\#}}\boldsymbol{T}^{-1}\end{bmatrix} \in \mathbb{R}^{2k\times k}$. Due to $\mathrm{rk}(\boldsymbol{I_k}) = k$ it follows that $\overline{\boldsymbol{C}}$ has full row rank and $\underline{\boldsymbol{C}}$ full column rank. Consequently, (54) and (68) yields $\mathrm{rk}(\boldsymbol{I_k} - \boldsymbol{B_0}\boldsymbol{B_T^{\#}}\boldsymbol{T}^{-1}) = \mathrm{rk}(\boldsymbol{I_k}) - \mathrm{rk}(\boldsymbol{B_0}\boldsymbol{B_T^{\#}}\boldsymbol{T}^{-1}) = k - j$. ■

Theorem 4: Let $\widetilde{\boldsymbol{B}}_{\boldsymbol{j}} \in \mathbb{R}^{j\times j}$ be an invertible submatrix of $\widetilde{\boldsymbol{B}}$ consisting of rows with indices $\{r_1, ..., r_j\}$. Then there exist invertible matrices $\boldsymbol{T} \in \mathbb{R}^{k\times k}$ which transform (52) such that its rows with indices $\{r_1, ..., r_j\}$ contain only zeros.

Proof: $\mathrm{rk}(\boldsymbol{M_T}) = k - j$ due to Lemma 2 which means it can contain up to j zero rows. W.l.o.g. it is assumed that the first j rows of (52) should be zeroed. Using the partitioned matrices

$$\widetilde{\boldsymbol{B}} = \begin{bmatrix}\widetilde{\boldsymbol{B}}_{\boldsymbol{j}}\\ \widetilde{\boldsymbol{B}}_{\boldsymbol{k-j}}\end{bmatrix} \text{ with } \widetilde{\boldsymbol{B}}_{\boldsymbol{j}} \in \mathbb{R}^{j\times j} \text{ and } \widetilde{\boldsymbol{B}}_{\boldsymbol{k-j}} \in \mathbb{R}^{(k-j)\times j} \tag{55}$$

and

$$\widetilde{\boldsymbol{B}}_{\boldsymbol{T}}^{\dagger}\boldsymbol{T}^{-1} = \begin{bmatrix}\widetilde{\boldsymbol{A}}_{\boldsymbol{1}} & \widetilde{\boldsymbol{A}}_{\boldsymbol{2}}\end{bmatrix} \text{ with } \widetilde{\boldsymbol{A}}_{\boldsymbol{1}} \in \mathbb{R}^{j\times j} \text{ and } \widetilde{\boldsymbol{A}}_{\boldsymbol{2}} \in \mathbb{R}^{j\times(k-j)} \tag{56}$$

the first j rows of $\boldsymbol{M_T} = \boldsymbol{I_k} - \widetilde{\boldsymbol{B}}\widetilde{\boldsymbol{B}}_{\boldsymbol{T}}^{\dagger}\boldsymbol{T}^{-1}$ read as

$$\boldsymbol{0_{j\times k}} \overset{!}{=} \begin{bmatrix}\boldsymbol{I_j} - \widetilde{\boldsymbol{B}}_{\boldsymbol{j}}\widetilde{\boldsymbol{A}}_{\boldsymbol{1}} & \widetilde{\boldsymbol{B}}_{\boldsymbol{j}}\widetilde{\boldsymbol{A}}_{\boldsymbol{2}}\end{bmatrix} \tag{57}$$

where $\boldsymbol{0_{j\times k}} \in \mathbb{R}^{j\times k}$ is a zero-matrix. It follows that $\widetilde{\boldsymbol{A}}_{\boldsymbol{1}} = \widetilde{\boldsymbol{B}}_{\boldsymbol{j}}^{-1}$ and $\widetilde{\boldsymbol{A}}_{\boldsymbol{2}} = \boldsymbol{0_{j\times(k-j)}}$. Defining $\overline{\boldsymbol{T}} = \boldsymbol{T}^{-1}$ one obtains

$$\widetilde{\boldsymbol{B}}_{\boldsymbol{T}}^{\dagger}\overline{\boldsymbol{T}} = \begin{bmatrix}\widetilde{\boldsymbol{B}}_{\boldsymbol{j}}^{-1} & \boldsymbol{0_{j\times(k-j)}}\end{bmatrix} = \left(\widetilde{\boldsymbol{B}}^T\overline{\boldsymbol{T}}^T\overline{\boldsymbol{T}}\widetilde{\boldsymbol{B}}\right)^{-1}\widetilde{\boldsymbol{B}}^T\overline{\boldsymbol{T}}^T\overline{\boldsymbol{T}}. \tag{58}$$

After partitioning $\overline{\boldsymbol{T}} = [\overline{\boldsymbol{T}}_{\boldsymbol{j}}\ \ \overline{\boldsymbol{T}}_{\boldsymbol{k-j}}]$ with $\overline{\boldsymbol{T}}_{\boldsymbol{j}} \in \mathbb{R}^{k\times j}$ and $\overline{\boldsymbol{T}}_{\boldsymbol{k-j}} \in \mathbb{R}^{k\times(k-j)}$ one can split (58) into two equations. One of them is

$$\boldsymbol{0_{j\times(k-j)}} = \underbrace{\left(\widetilde{\boldsymbol{B}}^T\overline{\boldsymbol{T}}^T\overline{\boldsymbol{T}}\widetilde{\boldsymbol{B}}\right)^{-1}\widetilde{\boldsymbol{B}}^T\overline{\boldsymbol{T}}^T}_{\widetilde{\boldsymbol{B}}_{\boldsymbol{T}}^{\dagger}}\overline{\boldsymbol{T}}_{\boldsymbol{k-j}} \tag{59}$$

where $\mathrm{rk}(\widetilde{\boldsymbol{B}}_{\boldsymbol{T}}^{\dagger}) = j$ (see Lemma 2). Hence the bracket term in (59) is full rank. Considering the left nullspace $\mathcal{N}_{\mathrm{l}}(\widetilde{\boldsymbol{B}}^T) = \boldsymbol{0}^T$ and partitioning of $\widetilde{\boldsymbol{B}}^T$ and $\overline{\boldsymbol{T}}^T$, equation (59) simplifies to

$$\boldsymbol{0_{j\times(k-j)}} = \begin{bmatrix}\widetilde{\boldsymbol{B}}_{\boldsymbol{j}}^T & \widetilde{\boldsymbol{B}}_{\boldsymbol{k-j}}^T\end{bmatrix}\begin{bmatrix}\overline{\boldsymbol{T}}_{\boldsymbol{j}}^T\\ \overline{\boldsymbol{T}}_{\boldsymbol{k-j}}^T\end{bmatrix}\overline{\boldsymbol{T}}_{\boldsymbol{k-j}}. \tag{60}$$

Rewriting (60) yields

$$\boldsymbol{0_{j\times(k-j)}} = \underbrace{\left(\widetilde{\boldsymbol{B}}_{\boldsymbol{j}}^T\overline{\boldsymbol{T}}_{\boldsymbol{j}}^T + \widetilde{\boldsymbol{B}}_{\boldsymbol{k-j}}^T\overline{\boldsymbol{T}}_{\boldsymbol{k-j}}^T\right)}_{\in\mathbb{R}^{j\times k}}\overline{\boldsymbol{T}}_{\boldsymbol{k-j}} \tag{61}$$

Note that $\dim\left[\mathcal{N}_{\mathrm{l}}(\overline{\boldsymbol{T}}_{\boldsymbol{k-j}})\right] = j$, which coincides with the number of rows in the bracket term of (61). Thus one can choose an arbitrary $\overline{\boldsymbol{T}}_{\boldsymbol{k-j}}$ with $\mathrm{rk}(\overline{\boldsymbol{T}}_{\boldsymbol{k-j}}) = k - j$ and determine a matrix $\boldsymbol{N_T} \in \mathbb{R}^{j\times k}$ such that $\mathrm{span}(\boldsymbol{N_T}) = \mathcal{N}_{\mathrm{l}}(\overline{\boldsymbol{T}}_{\boldsymbol{k-j}})$. Finally, the remaining columns of $\overline{\boldsymbol{T}}$ are given by

$$\overline{\boldsymbol{T}}_{\boldsymbol{j}} = \left(\boldsymbol{N_T} - \widetilde{\boldsymbol{B}}_{\boldsymbol{k-j}}^T\overline{\boldsymbol{T}}_{\boldsymbol{k-j}}^T\right)^T\widetilde{\boldsymbol{B}}_{\boldsymbol{j}}^{-1}. \tag{62}$$

Now that $\overline{\boldsymbol{T}}$ is completely determined it must be shown that it satisfies

$$\widetilde{\boldsymbol{B}}_{\boldsymbol{j}}^{-1} = \left(\widetilde{\boldsymbol{B}}^T\overline{\boldsymbol{T}}^T\overline{\boldsymbol{T}}\widetilde{\boldsymbol{B}}\right)^{-1}\widetilde{\boldsymbol{B}}^T\overline{\boldsymbol{T}}^T\overline{\boldsymbol{T}}_{\boldsymbol{j}}, \tag{63}$$

i.e. the second matrix equation in (58). Observe that the last $k - j$ columns of $\widetilde{\boldsymbol{B}}^T\overline{\boldsymbol{T}}^T\left[\overline{\boldsymbol{T}}_{\boldsymbol{j}}\ \overline{\boldsymbol{T}}_{\boldsymbol{k-j}}\right]$ are all zeros due to (60). Hence it follows that $\widetilde{\boldsymbol{B}}^T\overline{\boldsymbol{T}}^T\overline{\boldsymbol{T}}\widetilde{\boldsymbol{B}} = \widetilde{\boldsymbol{B}}^T\overline{\boldsymbol{T}}^T\overline{\boldsymbol{T}}_{\boldsymbol{j}}\widetilde{\boldsymbol{B}}_{\boldsymbol{j}}$ and inserted into (63) results indeed in $\left(\widetilde{\boldsymbol{B}}^T\overline{\boldsymbol{T}}^T\overline{\boldsymbol{T}}_{\boldsymbol{j}}\widetilde{\boldsymbol{B}}_{\boldsymbol{j}}\right)^{-1}\widetilde{\boldsymbol{B}}^T\overline{\boldsymbol{T}}^T\overline{\boldsymbol{T}}_{\boldsymbol{j}} = \widetilde{\boldsymbol{B}}_{\boldsymbol{j}}^{-1}\left(\widetilde{\boldsymbol{B}}^T\overline{\boldsymbol{T}}^T\overline{\boldsymbol{T}}_{\boldsymbol{j}}\right)^{-1}\widetilde{\boldsymbol{B}}^T\overline{\boldsymbol{T}}^T\overline{\boldsymbol{T}}_{\boldsymbol{j}} = \widetilde{\boldsymbol{B}}_{\boldsymbol{j}}^{-1}$. ■

Based on Theorem 4 an Enhanced Redistributed Pseudoinverse (ERPINV) algorithm is proposed. This RPINV extension allows to specify a priority list containing the numbers of $k - 1$ components of $\boldsymbol{e_v}$ which should be made zero. As soon as $j < k$ it takes the following measures:

ERPINV:

1) priority list items $1, ..., j$ determine the rows which form $\widetilde{\boldsymbol{B}}_{\boldsymbol{j}}$
2) if $\det(\widetilde{\boldsymbol{B}}_{\boldsymbol{j}}) = 0$ then abort
3) choose an arbitrary $\overline{\boldsymbol{T}}_{\boldsymbol{k-j}}$ with $\mathrm{rk}(\overline{\boldsymbol{T}}_{\boldsymbol{k-j}}) = k-j$ and compute a basis of its left nullspace $\boldsymbol{N_T} \in \mathbb{R}^{j\times k}$
4) evaluate $\overline{\boldsymbol{T}}_{\boldsymbol{j}} = \left(\boldsymbol{N_T} - \widetilde{\boldsymbol{B}}_{\boldsymbol{k-j}}^T\overline{\boldsymbol{T}}_{k-j}^T\right)^T\widetilde{\boldsymbol{B}}_{\boldsymbol{j}}^{-1}$ to get the inverse transformation $\overline{\boldsymbol{T}} = \boldsymbol{T}^{-1}$
5) use (10) and (11) to transform $\boldsymbol{B}$, $\boldsymbol{B_0}$, and $\boldsymbol{v_{des}}$
6) continue according to ordinary RPINV to get $\boldsymbol{u}^N$

Note that these extensions do not require any modifications of the controller as the desired virtual control vector is transformed according to the new factorization.

Remark 3: Apparently zeroing components of (52) only works if the resulting $\boldsymbol{u}^N \in \Omega$ because otherwise (31) is not fulfilled which is the basis for the derivation of (52).

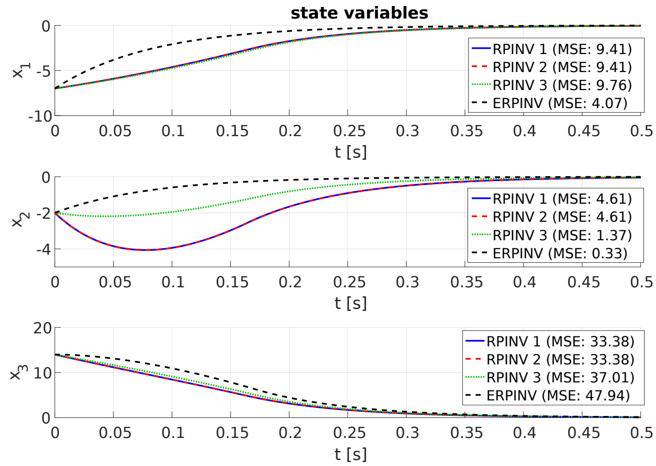

Fig. 3. States over time using RPINV for factorizations 1 - 3 and ERPINV.

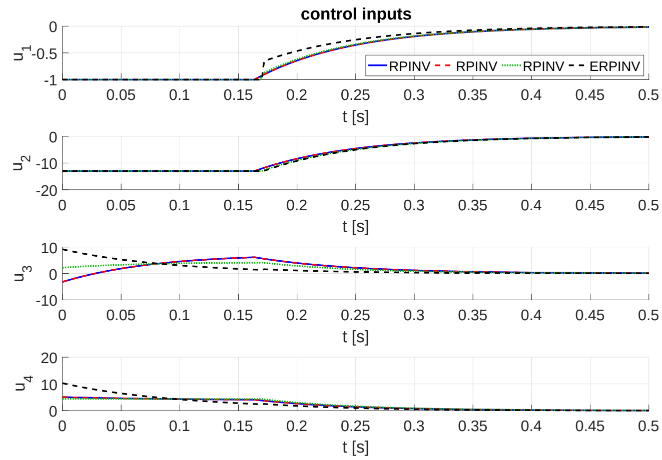

Fig. 4. Inputs over time using RPINV for factorizations 1 - 3 and ERPINV.

VI. Simulation Results

In this section the results for RPINV are demonstrated. The control goal is to stabilize the origin $\boldsymbol{x} = 0$. The state space representation of the plant reads as

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 2 & 0 \\ -1 & 1 & 2 \end{bmatrix} \boldsymbol{x} + \underbrace{\begin{bmatrix} -6 & 1 & 1 & 10 \\ -4 & 0 & 9 & -5 \\ -1 & 8 & 3 & 4 \end{bmatrix}}_{\boldsymbol{B_u}} \boldsymbol{u} \tag{64}$$

and the input constraints are $-\boldsymbol{u_{max}} \leq \boldsymbol{u} \leq \boldsymbol{u_{max}}$ with $\boldsymbol{u}_{\boldsymbol{max}}^T = [1\ 13\ 13\ 12]$. Three factorizations of $\boldsymbol{B_u}$ are tested:

Factorization 1:

$$\boldsymbol{B_{v1}} = \begin{bmatrix} 1 & 0 & 0 \\ \frac{2}{3} & -\frac{4}{47} & 1 \\ \frac{1}{6} & 1 & 0 \end{bmatrix} \qquad \boldsymbol{B_1} = \begin{bmatrix} -6 & 1 & 1 & 10 \\ 0 & \frac{47}{6} & \frac{17}{6} & \frac{7}{3} \\ 0 & 0 & \frac{403}{47} & -\frac{539}{47} \end{bmatrix}$$

Factorization 2:

$$\boldsymbol{B_{v2}} = \begin{bmatrix} -7 & 4 & 4 \\ -\frac{142}{141} & \frac{1516}{141} & \frac{139}{141} \\ \frac{17}{6} & -\frac{1}{3} & \frac{26}{3} \end{bmatrix} \qquad \boldsymbol{B_2} = \begin{bmatrix} \frac{14}{27} & \frac{73}{243} & \frac{5447}{11421} & -\frac{15022}{11421} \\ -\frac{8}{27} & -\frac{23}{486} & \frac{19673}{22842} & -\frac{7625}{11421} \\ -\frac{8}{27} & \frac{200}{243} & \frac{2551}{11421} & \frac{9889}{11421} \end{bmatrix}$$

Factorization 3:

$$\boldsymbol{B_{v3}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \boldsymbol{B_3} = \begin{bmatrix} -6 & 1 & 1 & 10 \\ -4 & 0 & 9 & -5 \\ -1 & 8 & 3 & 4 \end{bmatrix}$$

The transformation matrix describing the relationship between factorizations 1 and 2 has orthogonal columns and reads as

$$\boldsymbol{T_{12}} = \begin{bmatrix} -7 & 4 & 4 \\ 4 & -1 & 8 \\ 4 & 8 & -1 \end{bmatrix}. \tag{65}$$

The weighting matrix is $\boldsymbol{W} = \mathrm{diag}(\frac{1}{u_{max,1}}, ..., \frac{1}{u_{max,4}})$. A linear state-controller is chosen for stabilization. For each factorization the controller gain matrix $\boldsymbol{K_i}$ is chosen in order to place all eigenvalues of the closed-loop system matrices $(\boldsymbol{A} - \boldsymbol{B_{vi}}\boldsymbol{K_i})$ for $i = 1, 2, 3$ at -12. The initial state at $t = 0$ is $\boldsymbol{x}_{\boldsymbol{0}}^T = [-7 \quad -2 \quad 14]$.

In Figures 3 and 4 one can see a different behavior depending on the factorization because the number of free controls is smaller than $k = 3$. Due to the special transformation matrix $\boldsymbol{T_{12}}$ there is no deviation between factorizations 1 and 2. Figures 5 and 6 show desired and actually achieved effect on the plant as well as the effect error $\boldsymbol{e_v}$ for RPINV and ERPINV respectively. Initially ERPINV uses factorization 3 because $\boldsymbol{B_{v3}} = \boldsymbol{I_3}$ is required. It is configured to prefer vanishing error components $e_{v,1}$ and $e_{v,2}$ as recognizable in Figure 6. This also influences the performance in bringing the corresponding state variables to zero as illustrated by the mean squared errors (MSE) in Figure 3. The required transformation matrix is computed online and depends on which actuators saturate. In this example an approximation of one possible result is

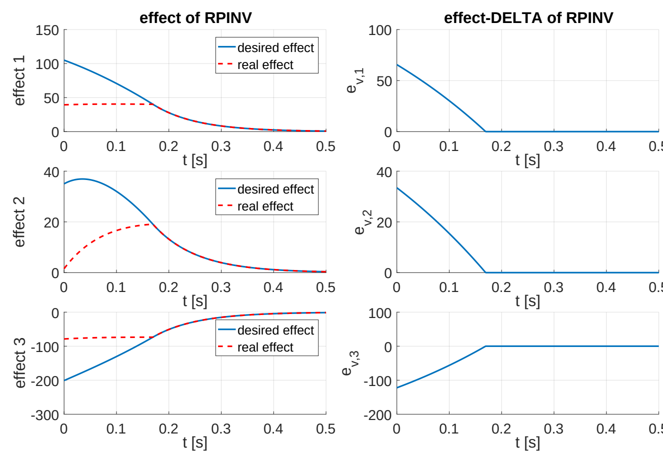

Fig. 5. Desired and achieved effect on plant for RPINV with factorization 3.

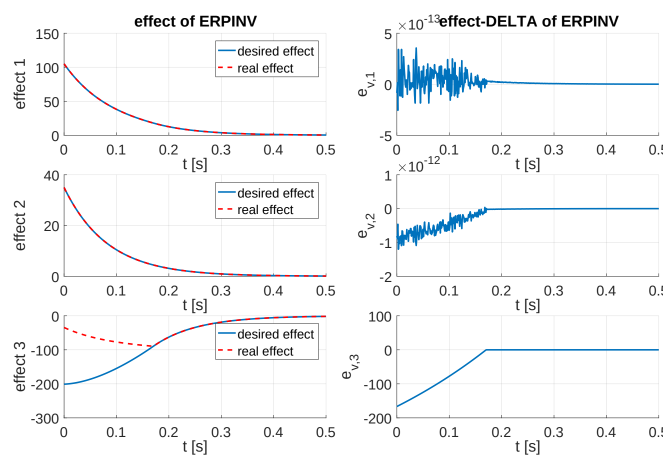

Fig. 6. Desired and achieved effect on plant for ERPINV with prioritization of $e_{v,1}$ and $e_{v,2}$.

$$\boldsymbol{T}^{-1} \approx \begin{bmatrix} -\frac{3155}{10844} & -\frac{5680}{33393} & \frac{1190}{3253} \\ -\frac{3397}{8324} & -\frac{1410}{9977} & \frac{16862}{22085} \\ -\frac{8543}{30064} & -\frac{4075}{19103} & \frac{9647}{15364} \end{bmatrix}. \tag{66}$$

VII. Conclusion

This work investigates the impact of input matrix factorization on two algorithms for constrained control allocation. The usage of transformation matrices allows to show why DA is invariant under factorization, while RPINV can be influenced depending on the number of actuator saturations. Only those factorizations which are connected by means of transformation matrices with orthogonal columns yield identical RPINV results under all circumstances. In other cases it is crucial when zeroing columns of the control effectivity matrix leads to its rank-reduction, because then the resulting virtual control vector starts to deviate from the desired one. The rank-reduction often prevents RPINV from achieving the desired effect on the plant. It is shown that by changing the factorization online depending on the desired virtual control one could reduce the deviation. But the complex structure of the optimization problem would stand in contrast to the simplicity of RPINV. If an electronic control unit is capable of handling optimization problems, it is more reasonable to solve the CA-problem with actuator constraints itself instead of using RPINV.

Rather than solving the optimization problem online it is possible to make a certain number of error components zero (ERPINV) depending on the rank of the reduced pseudoinverse. This is accomplished by a change of input matrix factorization, which is completely transparent to the controller. It allows a prioritization of error components in case of a rank deficient pseudoinverse originating from actuator saturations.

Appendix A
Rank of matrix product

Given $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{B} \in \mathbb{R}^{n\times o}$ their ranks satisfy (see [14])

$$\mathrm{rk}(\boldsymbol{A}) + \mathrm{rk}(\boldsymbol{B}) - n \leq \mathrm{rk}(\boldsymbol{A}\boldsymbol{B}) \leq \min\left[\mathrm{rk}(\boldsymbol{A}), \mathrm{rk}(\boldsymbol{B})\right]. \tag{67}$$

Appendix B
Rank of matrix difference

Assume two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ have the same size then one can conclude according to Theorem 17 in [19] that

$$\begin{gathered}\mathrm{rk}\left([\boldsymbol{A}\ \boldsymbol{B}]\right) = \mathrm{rk}(\boldsymbol{A}) = \mathrm{rk}\left(\begin{bmatrix}\boldsymbol{A}\\ \boldsymbol{B}\end{bmatrix}\right) \text{ and } \boldsymbol{B}\boldsymbol{A}^{\dagger}\boldsymbol{B} = \boldsymbol{B}\\ \Updownarrow\\ \mathrm{rk}(\boldsymbol{A}-\boldsymbol{B}) = \mathrm{rk}(\boldsymbol{A}) - \mathrm{rk}(\boldsymbol{B}).\end{gathered} \tag{68}$$

Acknowledgment

The project leading to this development has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 734832.

References

[1] T. A. Johansen and T. I. Fossen, "Control allocation - a survey," *Automatica*, vol. 49, pp. 1087–1103, May 2013.

[2] L. Zaccarian, "Dynamic allocation for input redundant control systems," *Automatica*, vol. 45, pp. 1431–1438, Mar. 2009.

[3] O. Härkegard and S. T. Glad, "Resolving actuator redundancy - optimal control vs. control allocation," *Automatica*, vol. 41, pp. 137–144, Jan. 2005.

[4] W. Durham, "Constrained control allocation," *Journal of Guidance, Control, and Dynamics*, vol. 17, pp. 717–725, Jul. 1993.

[5] K. A. Bordignon, "Constrained control allocation for systems with redundant control effectors," Ph.D. dissertation, Virginia Polytechnic Institute and State University, 1996.

[6] P. A. Servidia and R. S. Pena, "Spacecraft thruster control allocation problems," *IEEE Transactions on Automatic Control*, vol. 50, pp. 245–249, Feb. 2005.

[7] T. A. Johansen, T. P. Fuglseth, P. Tondel, and T. I. Fossen, "Optimal constrained control allocation in marine surface vessels with rudders," *Control Engineering Practice*, vol. 16, pp. 457–464, 2008.

[8] T. I. Fossen, T. A. Johansen, and T. Perez, "A survey of control allocation methods for underwater vehicles," in *Underwater Vehicles*. InTech, 2009.

[9] Y. Chen and J. Wang, "Fast and global optimal energy-efficient control allocation with applications to over-actuated electric ground vehicles," *IEEE Transactions on Control Systems Technology*, vol. 20, pp. 1202–1211, Sep. 2012.

[10] J. Brembeck and P. Ritzer, "Energy optimal control of an over actuated robotic electric vehicle using enhanced control allocation approaches," in *Proc. of 2012 Intelligent Vehicles Symposium*, Alcalá de Henares, Spain, Jun. 2012, pp. 322–327.

[11] H. Alwi and C. Edwards, "Sliding mode ftc with on–line control allocation," in *Proc. of 49th IEEE Conference on Decision and Control*, San Diego, USA, Dec. 2006, pp. 5579–5584.

[12] A. Cristofaro and T. A. Johansen, "Fault-tolerant control allocation with actuator dynamics: Finite-time control reconfiguration," in *Proc. of 53th IEEE Conference on Decision and Control*, Los Angeles, USA, dec 2014, pp. 4971–4976.

[13] M. Kirchengast, M. Steinberger, and M. Horn, "A new method to compute generalized inverses for control allocation," in *Proc. of the 55th IEEE Conference on Decision and Control*, Las Vegas, USA, dec 2016, pp. 5328–5334.

[14] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 1st ed. New York: Cambridge University Press, 2006.

[15] A. Ben-Israel and T. N. Greville, *Generalized Inverses: Theory and Applications*, 2nd ed., ser. CMS Series in Mathematics. Springer, 2003.

[16] W. Ford, *Numerical Linear Algebra with Application*, 1st ed. Elsevier Academic Press, 2015.

[17] S. Boyd and L. Vandenberghe, *Convex Optimization*, 7th ed. New York: Cambridge University Press, 2009.

[18] M. W. Oppenheimer, D. B. Doman, and M. A. Bolender, "Control allocation for over-actuated systems," in *Proc. of 2006 14th Mediterranean Conference on Control and Automation (MED'06)*, Ancona, Italy, Jun. 2006, pp. 1–6.

[19] G. Marsaglia and G. P. H. Styan, "Equalities and inequalities for ranks of matrices," *Linear and Multilinear Algebra*, vol. 2, no. 3, pp. 269–292, 1974.