Decomposing Linear Control Allocation Problems

Jørgen Spjøtvold*‡, Petter Tøndel† and Tor A. Johansen*

***Abstract*— A strategy for decomposing a class of constrained linear control allocation problems is considered. The actuators are partitioned into groups of mutually non-interacting actuators. The problem is then divided into a set of sub-problems and a master problem, whose solution approximates the solution to the original problem. For some classes of allocation problems the method is extended such that it also yields an optimal solution. The motivation behind the approach is to reduce the computational effort needed to find explicit solutions by obtaining sub-optimal solutions, increase possibilities of reconfigurable control allocation, and to provide a scheme that allows for a tradeoff between the drawbacks and benefits of the explicit solution to control allocation problems.**

***Index Terms*— Parametric programming. Quadratic programming. Linear programming. Mixed integer programming. Constrained linear control allocation.**

I. INTRODUCTION

The task in control allocation is to determine how to generate a specified generalized force from a redundant set of actuators where the associated controls are constrained, see e.g. [1]–[6]. The main objective is to obtain the desired generalized force, however, it is also common to incorporate secondary objectives, such as minimizing energy consumption and limiting the rate of change for a control input. Several other factors, such as actuator dynamics [5] and power management, can also be incorporated. One way of achieving these secondary goals is to solve a constrained optimization problem online at every sampling instant.

Only recently, it has, in conformity with the explicit model predictive control approach [7], [8], been proposed to solve the optimization problem off-line [4] by utilizing parametric programming techniques [7], [9]–[12]. The online computational effort then reduces to evaluate a piecewise affine function, which can be formulated as a point location problem [13], [14]. The four main advantages of this approach are: i) removing the need for sophisticated optimization software on the microchip/proseccor, ii) the correctness of the solution can be verified off-line, which is a key issue in safety critical applications, iii) the worst case number of arithmetic operations needed to find the solution can easily be computed, and iv) for a large class of problems the average and worst case number of arithmetic operations needed to find the solution is greatly reduced. The main drawbacks, on the other hand, are that i) obtaining an explicit solution may be computationally intractable, ii) the storage space required to represent the solution may exceed the available memory, and iii) in the context of constrained control allocation, the method does not easily allow reconfigurable control without increasing problem complexity.

* Department of Engineering Cybernetics, Norwegian University of Science and Technology, N-7491 Trondheim, Norway. E-mail: {jorgesp,torj}@itk.ntnu.no

‡ Corresponding author.

† Research Scientist, SINTEF ICT, N-7491 Trondheim, Norway. E-mail: petter.tondel@sintef.no

In this paper we propose a decomposing strategy for obtaining feasible, sub-optimal solutions to constrained linear control allocation problems. The procedure is motivated by the observation that for practical problems not all the actuators interact directly, suggesting a division of the problem into a set of smaller problems. The actuators are partitioned such that each element of the partition does not interact with the other elements (note that in this paper, the term *actuators* also include effectors, for example, both the rudder, and the engine that drives it, are labelled actuators). A master- and a set of sub-problems are designed for the purpose of obtaining a feasible, but sub-optimal solution. The decomposing scheme is also extended to yield an optimal solution for a class of allocation problems. In the proposed scheme we can choose to solve some of the problems explicitly and some online, allowing the designer to choose an approach that is best suited for the hardware and software available. Another benefit of the procedure is that reconfigurable control is somewhat more computationally tractable.

II. PROBLEM SETUP

A. Basic definitions and nomenclature

If $\mathcal{I}$ is an index set, then $|\mathcal{I}|$ denotes the cardinality of $\mathcal{I}$ and $\mathcal{I}_i$ refers to the i^{th} element in $\mathcal{I}$. When referring to a set of indices $\mathcal{I}$, we assume that the set is ordered, i.e. for the i^{th} element in $\mathcal{I}$ we have $\mathcal{I}_i < \mathcal{I}_j$, $\forall j \in \{i+1, \ldots, |\mathcal{I}|\}$. Recall that a partition of a set S is a collection of sub-sets of S such that the sub-sets are mutually disjoint and their union is equal to S. Let $\mathbb{N}_q$ denote the set $\{1, 2, \ldots, q\}$. If $A \in \mathbb{R}^{n\times m}$ is a matrix or column vector, then $A_{(i,*)} \in \mathbb{R}^{1\times m}$ denotes the i^{th} row of A and $A_{(\mathcal{I},*)} \in \mathbb{R}^{|\mathcal{I}|\times m}$ denotes the matrix $[A^T_{(\mathcal{I}_1,*)}, \ldots, A^T_{(\mathcal{I}_{|\mathcal{I}|},*)}]^T$. Similarly, $A_{(*,i)} \in \mathbb{R}^{n\times 1}$ denotes the i^{th} column of A and $A_{(*,\mathcal{I})} \in \mathbb{R}^{n\times|\mathcal{I}|}$ denotes the matrix $[A_{(*,\mathcal{I}_1)}, \ldots, A_{(*,\mathcal{I}_{|\mathcal{I}|})}]$. If A is a column vector, i.e. $A \in \mathbb{R}^{n\times 1}$, then $A_{(\mathcal{I},*)} \in \mathbb{R}^{|\mathcal{I}|\times 1}$ is abbreviated $A_{\mathcal{I}}$. Finally, if $\{\mathcal{J}^i \mid i \in \mathcal{I}\}$ is a partition of the index set $\mathcal{J}$ and $u \in \mathbb{R}^{|\mathcal{J}|}$ is a vector, we define the operator $\operatorname{sort}(\cdot)$ as the operator that maps the set of sub-vectors $\{u_{\mathcal{J}^i} \mid i \in \mathcal{I}\}$ into $\mathbb{R}^{|\mathcal{J}|}$ and restores the original ordering of the vector, i.e. $u = \operatorname{sort}(\{u_{\mathcal{J}^i} \mid i \in \mathcal{I}\})$.

Recall that the set of affine combinations of points in a set $S \subset \mathbb{R}^n$ is called the *affine hull* of S, and is denoted $\operatorname{aff}(S)$. The *dimension of a set* $S \subset \mathbb{R}^n$ is the dimension of $\operatorname{aff}(S)$, and is denoted $\dim(S)$; if $\dim(S) = n$, then S is said to be full-dimensional. The *closure* and *interior* of a set S is denoted $\operatorname{cl}(S)$ and $\operatorname{int}(S)$, respectively. A *polyhedron* is the intersection of a finite number of open and/or closed half-spaces. A *polygon* is a finite union of polyhedra. If $F : X \to Y$ is a mapping, then the restriction of F to the domain $D \subseteq X$ is written $F|_D : D \to Y$. If a mapping F is set-valued the notation $F : X \rightrightarrows Y$ specifies this. A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is said to be *piecewise affine* (PWA) on $D \subset \mathbb{R}^n$ if D can be represented as a finite union of polyhedra, relative to each of which $f(x)$ is given by an affine expression.

B. Static linear control allocation

Consider the system

$$\dot{x} = f(t, x, \bar{\tau}),$$
$$\bar{\tau} = Bu,$$

where $x \in \mathbb{R}^n$ is the state, t is time, $\bar{\tau} \in \mathcal{T} \subseteq \mathbb{R}^r$ are the generalized forces (virtual controls), $u \in \mathbb{R}^m$ are the controls, and the matrix $B \in \mathbb{R}^{r\times m}$ defines the (linear) relationship between the generalized forces and the controls. Assume further that a virtual controller $\tau := k(t, x)$ is given, i.e. τ is our desired generalized force (virtual control). The task in control allocation is to generate the force τ the controller specifies using the available controls $u \in U \subseteq \mathbb{R}^m$, where U is assumed to be full-dimensional and bounded. Since, in general, one cannot assume that it is possible to generate τ when u is constrained to U, slacks s are introduced in order to ensure that a solution is always obtained, i.e. $Bu+s = \tau$. Hence, the linear control allocation problem can be stated as:

$$\mathbb{P}(\tau) : \quad J^*(\tau) : = \inf_{(u,s)\in\mathcal{Y}(\tau)} J(u, s, \tau) \tag{1a}$$
$$: = \inf_{(u,s)\in\mathcal{Y}(\tau)} \|Qs\|_l + \|Ru\|_l, \tag{1b}$$
$$\mathcal{Y}(\tau) : = \left\{ (u, s) \in \mathbb{R}^m \times \mathbb{R}^p \;\middle|\; \begin{array}{l} Bu + s = \tau, \\ u \in U \end{array} \right\}, \tag{1c}$$

where $Q \in \mathbb{R}^{p\times p}$ and $R \in \mathbb{R}^{m\times m}$ are weight matrices, respectively penalizing use of controls and infeasibility, and $l \in \{1, 2, \infty\}$[1] denotes the weighting norm. We will assume that $\mathbb{P}(\tau)$ attains its minimum over $\mathcal{Y}(\tau)$, $\forall\tau \in \mathcal{T}$, where $\mathcal{T}$ is a full-dimensional polygon (where each polyhedron in $\mathcal{T}$ is also assumed to be full-dimensional). Henceforth, we write the problem as minimization. In the sequel let the set valued map $U^* : \mathbb{R}^r \rightrightarrows \mathbb{R}^m$ be defined as

$$U^*(\tau) := \left\{ u \;\middle|\; u \in \arg\min_{(u,s)\in\mathcal{Y}(\tau)} J(u, s, \tau) \right\}.$$

and let $u^*(\cdot) : \mathbb{R}^r \to \mathbb{R}^m$ denote a single-valued selection of $U^*(\cdot)$, i.e. $u^*(\tau) \in U^*(\tau)$ for all $\tau \in \mathcal{T}$. We also let $s^*(\cdot)$ denote a single-valued selection of $S^*(\cdot)$ where $S^*(\cdot)$ is defined by replacing u with s in the equation above.

In the sequel we distinguish between two types of linear allocation problems; *i)* where the set U is convex, and *ii)* when U is non-convex. We will also make use of the following assumption:

Assumption 1: Define:

$$\mathbb{P}_\varepsilon(\tau) : \quad J^*_\varepsilon(\tau) := \min_{(u,s)\in\mathcal{Y}_\varepsilon(\tau)} J(u, s, \tau),$$
$$\mathcal{Y}_\varepsilon(\tau) := \left\{ (u, s) \in \mathbb{R}^m \times \mathbb{R}^p \;\middle|\; \begin{array}{l} Bu + s = \tau, \\ u \in U_\varepsilon \end{array} \right\},$$

[1] $l = 2$ denotes, with some abuse of mathematical rigor, the quadratic norm, that is, $\|Qx\|_2 := x^TQx$.

and let the set U be full-dimensional and bounded. Given any $\varepsilon > 0$ we assume that there exists a polygon $U_\varepsilon := \cup_{i\in\mathcal{I}} U^i$ that inner approximates U in the sense that $U_\varepsilon \subseteq U$, where $\mathcal{I}$ contains a finite number of elements and each U^i is a full-dimensional polyhedron, and

$$\forall\tau \in \mathcal{T}\ J^*_\varepsilon(\tau) \le J^*(\tau) + \varepsilon \text{ and } \arg\min_{(u,s)\in\mathcal{Y}_\varepsilon(\tau)} J(u, s, \tau) \ne \emptyset$$

As a consequence of the above assumption, we will henceforth assume that the set U in $\mathbb{P}(\tau)$ is a polygon or a polyhedron, which will be clear from the context.

C. Reconfigurable control allocation

In many applications it is desirable to be able to switch on and off actuators or to change the constraints imposed on the control inputs to an actuator. The most straightforward way of achieving this is to define additional parameters ϕ, and rewrite (1) as

$$J^*(\tau, \phi) : = \min_{(u,s)\in\mathcal{Y}(\tau,\phi)} \|Qs\|_l + \|Ru\|_l,$$
$$\mathcal{Y}(\tau, \phi) : = \{(u, s) \in \mathbb{R}^m \times \mathbb{R}^p \mid Bu + s = \tau, u \in U(\phi)\}.$$

This approach does not complicate the online optimization problem. In addition, if the parametrization $U(\cdot)$ is linear, it is possible to solve the problem explicitly [4], however, with this approach the complexity of the optimal control $u^*(\cdot, \cdot)$ is often too high for the available memory, and, in some cases, it may even be computationally intractable to obtain the explicit solution.

III. Explicit solutions to control allocation problems

Recently it has been proposed to solve (1) explicitly(see e.g. [4], [6]) and thereby avoid online optimization. The next three subsections summarizes the solution properties of parametric linear-, quadratic-, and mixed integer linear programs.

A. Parametric linear programs [9], [12], [15]

Consider the linear program with parameters on the right hand side of the constraints:

$$J^*(\theta) : = \min_{x\in P(\theta)} c^T x, \tag{2a}$$
$$P(\theta) : = \{x \in \mathbb{R}^n \mid Ax \le b + S\theta\} \tag{2b}$$

where c, A, b, and S are matrices with suitable dimensions, and (2) is to be solved for all values of $\theta \in \Theta \subseteq \mathbb{R}^s$, where Θ is the set of parameters in which the minimum in (2) exists.

Theorem 1: Consider (2).

1) The function $J^* : \Theta \to \mathbb{R}$ is continuous, convex and PWA on closed, full-dimensional, polyhedra.
2) There exists an optimizer function $x^* : \Theta \to \mathbb{R}^n$, $\theta \mapsto x^*(\theta) \in \arg\min_{x\in P(\theta)} c^T x$ that is continuous and PWA on closed, full-dimensional, polyhedra.

Obtaining a continuous selection $x^*(\cdot)$ can be done for instance via lexicographic perturbation of the pLP [16] or by choosing the minimum norm solution [17].

B. Parametric quadratic programs [7]

Consider the convex quadratic program with parameters on the right hand side of the constraints:

$$J^*(\theta) := \min_{x \in P(\theta)} \frac{1}{2} x^T H x + c^T x, \tag{3a}$$

$$P(\theta) := \{x \in \mathbb{R}^n \mid Ax \leq b + S\theta\} \tag{3b}$$

where H, c, A, b, and S are matrices with suitable dimensions, and $H = H^T \geq 0$.

Theorem 2: Consider (3)

1) The function $J^* : \Theta \to \mathbb{R}$ is continuous, convex and piecewise quadratic on closed, full-dimensional, polyhedra.
2) There exists an optimizer function $x^* : \Theta \to \mathbb{R}^n$, $\theta \mapsto x^*(\theta) \in \arg\min_{x \in P(\theta)} \frac{1}{2} x^T H x + c^T x$ that is continuous and PWA on closed, full-dimensional, polyhedra.

A continuous selection can be obtained by choosing the minimum norm solution [18]. Note that if $H > 0$, the solution $x^*(\cdot)$ to (3) is unique, and hence, also continuous.

C. Parametric mixed-integer linear programs [10], [12]

Consider the mixed integer linear program with parameters on the right hand side of the constraints:

$$J^*(\theta) := \min_{(x,y) \in P(\theta)} c^T x + d^T y, \tag{4a}$$

$$P(\theta) := \{(x, y) \in \mathbb{R}^n \times \{0, 1\}^p \mid Ax + Dy \leq b + S\theta\} \tag{4b}$$

where c, d, A, D, b, and S are matrices with suitable dimensions.

Theorem 3: Consider (3)

1) The function $J^* : \Theta \to \mathbb{R}$ is lower-semicontinuous and PWA.
2) There exists optimizer functions $x^* : \Theta \to \mathbb{R}^n$ and $y^* : \Theta \to \{0, 1\}^p$, $\theta \mapsto (x^*(\theta), y^*(\theta)) \in \arg\min_{(x,y) \in P(\theta)} c^T x + d^T y$ that are respectively PWA and piecewise constant.

D. Explicit solution to constrained linear control allocation

If we consider (1), then under our assumption on U we have that $\mathcal{T}$ is a polygon. Moreover, if $l \in \{1, \infty\}$ and U is a polyhedron, then (1) can be written as a pLP (2) by viewing τ as parameters. Similarly if $l = 2$ and U is a polyhedron we have a pQP (3). Finally, if U is a polygon, we have that (1) is a pMILP ($l \in \{1, \infty\}$) or pMIQP ($l = 2$).

IV. Decomposing allocation problems

In this section we propose the decomposing scheme for constrained linear control allocation for the case where U (or its inner approximation) is convex. In the sequel, if $u \in U \subseteq \mathbb{R}^n$ and $\mathcal{I} \subseteq \mathbb{N}_n$ is an index set, then $U_\mathcal{I}$ denotes the set $U_\mathcal{I} := \left\{u_\mathcal{I} \in \mathbb{R}^{|\mathcal{I}|} \,\middle|\, \exists u_{\mathbb{N}_n \setminus \mathcal{I}} : (u_\mathcal{I}, u_{\mathbb{N}_n \setminus \mathcal{I}}) \in U\right\}$. Moreover, if $\mathcal{J} \subseteq \mathbb{N}_n$ is another index set such that $\mathcal{I} \cap \mathcal{J} = \emptyset$, then with some abuse of notation, $U_\mathcal{I}(u_\mathcal{J})$ denotes the set

$$U_\mathcal{I}(u_\mathcal{J}) := \left\{ u_\mathcal{I} \in \mathbb{R}^{|\mathcal{I}|} \,\middle|\, \begin{array}{l} \exists u_{\mathbb{N}_n \setminus (\mathcal{I} \cup \mathcal{J})} : \\ (u_\mathcal{I}, u_\mathcal{J}, u_{\mathbb{N}_n \setminus (\mathcal{I} \cup \mathcal{J})}) \in U \end{array} \right\}$$

Definition 1 (Non-interacting actuators): Let the controls u be constrained to U. Given two actuators, A and B, and corresponding index sets $\mathcal{A}$ and $\mathcal{B}$ such that $u_\mathcal{A} \in \mathbb{R}^{|\mathcal{A}|}$ and $u_\mathcal{B} \in \mathbb{R}^{|\mathcal{B}|}$ are the control inputs to actuator A and B, respectively. The actuators A and B are said to be *non-interacting* if and only if

$$U_\mathcal{A}(u_\mathcal{B}) = U_\mathcal{A}, \ \forall u_\mathcal{B} \in U_\mathcal{B}, \quad \text{and}$$
$$U_\mathcal{B}(u_\mathcal{A}) = U_\mathcal{B}, \ \forall u_\mathcal{A} \in U_\mathcal{A}.$$

Remark 1: For linear constrained control allocation problems non-interacting actuators means that by changing the control input for actuator A, the constraints on the control inputs to actuator B are unchanged. Note however, that the controls may still be coupled through the linear relationship $Bu = \bar{\tau}$. In addition, we would like to point out that the linear version of the control allocation problem is often an approximation to a non-linear relationship $\bar{\tau} = g(x, u, t)$. If this is the case, then one should add additional restrictions on the interactions between the actuators in the sense that: the contribution to the generalized forces from actuator A is unchanged for all possible contributions from actuator B. This captures non-linear interaction between the actuators, for example, for marine vessels it is not uncommon to loose effect from thruster A if thruster B affects the flow pattern around thruster A.

Definition 2 (Non-interacting actuator partition): Consider a set of actuators $\mathcal{P} := \{p_i \mid i \in \mathcal{I}\}$ and the partition $\{P_j \mid j \in \mathcal{J}\}$ of $\mathcal{P}$. If for every pair $(p_A, p_B) \in P_k \times P_j$, $\forall k \in \mathcal{J}$ and $\forall j \in \mathcal{J}$, $k \neq j$, (p_A, p_B) are non-interacting actuators, then $\{P_j \mid j \in \mathcal{J}\}$ is said to be a *non-interacting actuator partition* of $\mathcal{P}$.

In the sequel, let $\{P_j \mid j \in \mathcal{J}\}$ denote a non-interacting actuator partition of $\mathcal{P}$ and $\left\{\mathcal{J}^j \mid j \in \mathcal{J}\right\}$ be the corresponding collection of index sets, i.e. the control inputs to the actuators in P_j are $u_{\mathcal{J}^j}$. It is immediate that we can write $\mathbb{P}(\cdot)$ as

$$J^*(\tau) := \min_{(u,s) \in \mathcal{Y}(\tau)} \|Qs\|_l + \|Ru\|_l,$$

$$\mathcal{Y}(\tau) := \left\{ (u, s) \,\middle|\, \begin{array}{l} s + \sum_{j \in \mathcal{J}} B_{(*, \mathcal{J}^j)} u_{\mathcal{J}^j} = \tau \\ u_{\mathcal{J}^j} \in U_{\mathcal{J}^j}, \quad \forall j \in \mathcal{J} \end{array} \right\}.$$

In the next section we re-formulate the above problem to obtain a master- and a set of sub-problems.

A. Decomposing Constrained Linear Control Allocation over Convex Sets

In this section we propose the method for decomposing the allocation problem. $\mathbb{M}(\cdot)$ will denote the master problem and a sub-problem will be denoted $\mathbb{S}_j(\cdot)$ for $j \in \mathcal{J}$. The master problem is defined as

$$\mathbb{M}(\tau) : \quad V^*(\tau) := \min_{\{s, \tau^1, \ldots, \tau^{|\mathcal{J}|}\}} \|Qs\|_l + \sum_{j \in \mathcal{J}} \|H^j \tau^j\|_l \tag{5a}$$

$$\text{s.t.} \quad s + \sum_{j \in \mathcal{J}} \tau^j = \tau, \tag{5b}$$

$$\tau^j \in \mathcal{T}^j \subset \mathbb{R}^r, \ \forall j \in \mathcal{J}, \tag{5c}$$

where $H^j = (H^j)^T \geq 0 \in \mathbb{R}^{r \times r}$ are suitably defined weight matrices and

$$\mathcal{T}^j := \left\{\tau^j \in \mathcal{T} \subseteq \mathbb{R}^r \,\middle|\, \exists y_{\mathcal{J}^j} : y_{\mathcal{J}^j} \in \mathcal{N}_j(\tau^j)\right\} \tag{6a}$$

$$\mathcal{N}_j(\tau^j) := U_{\mathcal{J}^j} \cap \left\{ y_{\mathcal{J}^j} \in \mathbb{R}^{|\mathcal{J}^j|} \,\middle|\, B_{(*, \mathcal{J}^j)} y_{\mathcal{J}^j} = \tau^j \right\}. \tag{6b}$$

It is clear that $\mathcal{T}^j$ is the set of all possible generalized forces (virtual controls) that can be generated by the actuators in the j^{th} element of the actuator partition.

For a given $j \in \mathcal{J}$, the j^{th} sub-problem is defined as:

$$\mathbb{S}_j(\tau^j): \quad V_j^*(\tau^j) := \min_{y_{\mathcal{J}^j} \in \mathcal{N}_j(\tau^j)} \|R_{(\mathcal{J}^j,\mathcal{J}^j)} y_{\mathcal{J}^j}\|_l. \qquad (7)$$

For notational simplicity we let $\{s^{\mathbb{M}}(\cdot), \tau^1(\cdot), \ldots, \tau^{|\mathcal{J}|}(\cdot)\}$ denote a set of single valued, continuous, selections for $\mathbb{M}(\cdot)$. Moreover, $\{y_{\mathcal{J}^1}(\cdot), \ldots, y_{\mathcal{J}^{|\mathcal{J}|}}(\cdot)\}$ are single valued, continuous, selections for $\{\mathbb{S}_j(\cdot) \mid j \in \mathcal{J}\}$. By a solution to $\mathbb{M}(\cdot)$ and $\{\mathbb{S}_j(\cdot) \mid j \in \mathcal{J}\}$ we mean the function $y^* : \mathbb{R}^r \to \mathbb{R}^m$ defined as

$$y^*(\tau) := \operatorname{sort}\left(\{y_{\mathcal{J}^1}(\tau^1(\tau)), \ldots, y_{\mathcal{J}^{|\mathcal{J}|}}(\tau^{|\mathcal{J}|}(\tau))\}\right),$$

i.e. $y^*(\cdot)$ has the same dimension and ordering as $u^*(\cdot)$.

Lemma 1: Consider $\mathbb{M}(\cdot)$, $\{\mathbb{S}_j(\cdot) \mid j \in \mathcal{J}\}$ and (1). We have that if $(s^{\mathbb{M}}(\cdot), y^*(\cdot))$ is a feasible solution to $\mathbb{M}(\cdot)$ and $\{\mathbb{S}_j(\cdot) \mid j \in \mathcal{J}\}$, then $(s^{\mathbb{M}}(\cdot), y^*(\cdot))$ is feasible for (1). Moreover, if $R_{(\mathcal{J}^j,\mathcal{J}^j)} = R^T_{(\mathcal{J}^j,\mathcal{J}^j)} \geq 0$ for all $j \in \mathcal{J}$, then we have

1) if $l = 2$, then $V^* : \mathbb{R}^r \to \mathbb{R}$ and each $V_j^* : \mathbb{R}^r \to \mathbb{R}$, $j \in \mathcal{J}$ are piecewise quadratic, convex, and continuous.
2) if $l \in \{1, \infty\}$, then $V^* : \mathbb{R}^r \to \mathbb{R}$ and each $V_j^* : \mathbb{R}^r \to \mathbb{R}$, $j \in \mathcal{J}$ are PWA, convex, and continuous.

Proof: The feasible sets are equal by construction. The properties of $V^*(\cdot)$ and each $V_j^*(\cdot)$ follows from noting that the problems are pQPs for $l = 2$ and pLPs for $l \in \{1, \infty\}$ (Theorems 1 and 2). ■

How to choose the weight matrices H^j, $j \in \mathcal{J}$ such that the solution $(s^{\mathbb{M}}(\cdot), y^*(\cdot))$ is not only feasible for (1), but also as close to optimal as possible is non-trivial, however, we will not elaborate on this, since exact solutions can be obtained by imposing a natural assumption on R, which is stated below. In Section V we will show by example that if the problem has certain symmetry properties, the weight matrices $\{H^j \mid j \in \mathcal{J}\}$ are easy to choose.

Assumption 2: Consider (1). For the weighting matrix R, set of actuators $\mathcal{P} := \{p_1, \ldots, p_I\}$, and non-interacting actuator partition $\{P_j \mid j \in \mathcal{J}\}$ of $\mathcal{P}$, we assume that $R_{(\mathcal{J}^i,\mathcal{J}^j)} = R_{(\mathcal{J}^j,\mathcal{J}^i)} = 0$ if $i \neq j$. Moreover, we assume that for each $j \in \mathcal{J}$ we have that $R_{(\mathcal{J}^j,\mathcal{J}^j)} = R^T_{(\mathcal{J}^j,\mathcal{J}^j)} \geq 0$.

Lemma 2: Assumption 2 has the consequence that for $l \in \{1, 2\}$ we have:

$$\|Ru\|_l = \sum_{j \in \mathcal{J}} \|R_{(\mathcal{J}^j,\mathcal{J}^j)} u_{\mathcal{J}^j}\|_l, \qquad (8)$$

and for $l = \infty$ we have

$$\|Ru\|_\infty = \max_{j \in \mathcal{J}} \{\|R_{(\mathcal{J}^j,\mathcal{J}^j)} u_{\mathcal{J}^j}\|_\infty\}$$

Proof: For the quadratic norm we have

$$\begin{aligned}\|Ru\|_2 &:= u^T R u = \begin{bmatrix} u^T_{\mathcal{J}^1} & \ldots & u^T_{\mathcal{J}^{|\mathcal{J}|}} \end{bmatrix} \\ &\operatorname{diag}\left(R_{(\mathcal{J}^1,\mathcal{J}^1)}, \ldots, R_{(\mathcal{J}^{|\mathcal{J}|},\mathcal{J}^{|\mathcal{J}|})}\right) \begin{bmatrix} u^T_{\mathcal{J}^1} & \ldots & u^T_{\mathcal{J}^{|\mathcal{J}|}} \end{bmatrix}^T \\ &= \|R_{(\mathcal{J}^1,\mathcal{J}^1)} u_{\mathcal{J}^j}\|_2 + \cdots + \|R_{(\mathcal{J}^{|\mathcal{J}|},\mathcal{J}^{|\mathcal{J}|})} u_{\mathcal{J}^{|\mathcal{J}|}}\|_2,\end{aligned}$$

and for $l = 1$ we recall that if $i \notin \mathcal{J}^j$ then $R_{(i,\mathcal{J}^j)} u_{\mathcal{J}^j} = 0$, and hence

$$\begin{aligned}\|Ru\|_1 &= \sum_{p \in \mathbb{N}_m} \left| \sum_{q \in \mathbb{N}_m} R_{(p,q)} u_q \right| \\ &= \sum_{j \in \mathcal{J}} \sum_{p \in \mathcal{J}^j} \left| \sum_{q \in \mathcal{J}^j} R_{(p,q)} u_q \right| = \sum_{j \in \mathcal{J}} \|R_{(\mathcal{J}^j,\mathcal{J}^j)} u_{\mathcal{J}^j}\|_1.\end{aligned}$$

For $l = \infty$, Assumption 2 clearly leads to

$$\begin{aligned}\|Ru\|_\infty &= \max \left\{ \left| \sum_{p \in \mathbb{N}_m} R_{(1,p)} u_p \right|, \ldots, \left| \sum_{p \in \mathbb{N}_m} R_{(m,p)} u_p \right| \right\} \\ &= \max \left\{ \|R_{(\mathcal{J}^1,\mathcal{J}^1)} u_{\mathcal{J}^1}\|_\infty, \ldots, \|R_{(\mathcal{J}^{|\mathcal{J}|},\mathcal{J}^{|\mathcal{J}|})} u_{\mathcal{J}^{|\mathcal{J}|}}\|_\infty \right\}\end{aligned}$$

■

In the following lemma we are only concerned with the set $\mathcal{T}^* \subseteq \mathcal{T}$ in which $s^*(\tau) = 0$ i.e. we have also assumed that $\|Qs\|_l$ is an exact penalty function for (1) [4].

Lemma 3: Consider $\mathbb{M}(\cdot)$, $\{\mathbb{S}_j(\cdot) \mid j \in \mathcal{J}\}$ and (1), and let $l \in \{1, 2\}$. By changing the master problem to

$$\mathbb{M}_e(\tau): \quad V^*(\tau) := \min_{\{t, \tau^1, \ldots, \tau^{|\mathcal{J}|}\}} t \qquad (9a)$$

$$\text{s.t.} \quad t \geq \sum_{j \in \mathcal{J}} V_j^*(\tau^j) \qquad (9b)$$

$$\sum_{j \in \mathcal{J}} \tau^j = \tau, \qquad (9c)$$

$$\tau^j \in \mathcal{T}^j, \ \forall j \in \mathcal{J}, \qquad (9d)$$

we have that $J^*(\tau) = V^*(\tau)$, $\forall \tau \in \mathcal{T}^*$, $y^*(\cdot) \in U^*(\cdot)$ and $\mathbb{M}_e(\cdot)$ is a convex optimization problem.

Proof: Convexity of $\mathbb{M}_e(\cdot)$ follows easily by noting that all constraints are linear except (9b), which is also convex due to Theorem 1 and 2. Note first that by construction $y(\cdot) = u^*(\cdot)$ is feasible for (9) and $\{\mathbb{S}_j(\cdot) \mid j \in \mathcal{J}\}$ and that $J(u^*(\tau), 0, \tau) = J^*(\tau)$, $\forall \tau \in \mathcal{T}^*$. We also have

$$\begin{aligned}V^*(\tau) = t^*(\tau) &\geq \sum_{j \in \mathcal{J}} V_j^*(\tau^j) = \sum_{j \in \mathcal{J}} \|R_{(\mathcal{J}^j,\mathcal{J}^j)} y_{\mathcal{J}^j}(\tau^j)\|_l \\ &= \|Ry^*(\tau)\|_l = J(\tau, 0, y^*(\tau)).\end{aligned}$$

Hence, if $y^*(\tau) \notin U^*(\tau)$ then $J(y^*(\tau), 0\tau) > J(u^*(\tau), 0, \tau)$, which contradicts optimality since $u^*(\tau)$ is feasible for (9) and $\{\mathbb{S}_j(\cdot) \mid j \in \mathcal{J}\}$. ■

For $l = \infty$ the master problem has to be modified slightly as demonstrated by the following lemma:

Lemma 4: Consider $\mathbb{M}(\cdot)$, $\{\mathbb{S}_j(\cdot) \mid j \in \mathcal{J}\}$ and (1), and let $l = \infty$. By changing the master problem to

$$\begin{aligned}\mathbb{M}_e(\tau): \quad V^*(\tau) &:= \min_{\{t, \tau^1, \ldots, \tau^{|\mathcal{J}|}\}} t \\ \text{s.t.} \quad & t \geq V_j^*(\tau^j) \quad \forall j \in \mathcal{J} \\ & \sum_{j \in \mathcal{J}} \tau^j = \tau, \\ & \tau^j \in \mathcal{T}^j, \ \forall j \in \mathcal{J},\end{aligned}$$

we have that $J^*(\tau) = V^*(\tau)$, $\forall \tau \in \mathcal{T}^*$, $y^*(\cdot) \in U^*(\cdot)$ and $\mathbb{M}_e(\cdot)$ is a convex optimization problem.

Proof: The proof is identical to the proof of Lemma 3, except

$$V^*(\tau) = t^*(\tau) \geq \max\{\|R_{(\mathcal{J}^1,\mathcal{J}^1)} y_{\mathcal{J}^1}(\tau^1)\|_\infty, \dots,\\ \|R_{(\mathcal{J}^{|\mathcal{J}|},\mathcal{J}^{|\mathcal{J}|})} y_{\mathcal{J}^{|\mathcal{J}|}}(\tau^{|\mathcal{J}|})\|_\infty\} = \|Ry^*(\tau)\|_\infty.$$

■

Remark 2: For $l \in \{1, \infty\}$ it is straightforward to solve $\mathbb{M}_e(\cdot)$ explicitly since $\mathcal{T}^*$ can be expressed as a union of polyhedra, and in each of these $\mathbb{M}_e(\cdot)$ is a pLP. On the other hand, for $l = 2$, there is currently no available algorithm for obtaining an exact, explicit, solution of (9).

B. Decomposing Constrained Linear Control Allocation over Non-convex Sets

If the set of attainable forces U is a non-convex polygon, $U = \cup_{i \in \mathcal{I}} U^i$, the optimization problem (1) is no longer convex. However, the set $\mathcal{Y}(\cdot)$ can be written as

$$\mathcal{Y}(\tau) := \left\{(u, s) \,\middle|\, Bu + s = \tau, u \in U^1 \vee \cdots \vee U^{|\mathcal{I}|}\right\},$$

and (1) becomes a parametric mixed integer program. In this case the problem can also be decomposed into $\mathbb{M}(\cdot)$ and $\{\mathbb{S}_j(\cdot) \mid j \in \mathcal{J}\}$ for the purpose of obtaining sub-optimal solutions. The main difference being that the sets $\{\mathcal{T}^j \mid j \in \mathcal{J}\}$ are more computationally demanding to obtain, since $U_{\mathcal{J}^j} = \cup_{i \in \mathcal{I}} U^i_{\mathcal{J}^j}$. In the non-convex case, both the master- and sub-problems are parametric mixed integer programs. For brevity, we do not consider this case in detail.

C. Reconfigurable control allocation

If the master-problem is solved online, we can obtain a tradeoff between the benefits and drawbacks of the explicit solution when the scheme is applied to reconfigurable control allocation. By introducing extra parameters in the allocation problem, as described in Section II-C, the complexity of $u^*(\cdot)$ may increase to the level where the explicit scheme is rendered unusable. By solving the master problem online and the sub-problems explicitly, the control actions from actuator group P_j can be limited simply by changing the constraints on τ^j.

V. Numerical Example

Note that in this section we use slightly different indexing of the variables. Consider the following allocation problem:

$$\min_{\{u,s\}} \left\{ u^T R u + s^T Q s \,\middle|\, \begin{array}{rl} s_x + \sum_{i=1}^4 u_{i,x} & = \tau_x \\ s_y + \sum_{i=1}^4 u_{i,y} & = \tau_y \\ |u_{i,x}| + |u_{i,y}| & \leq 2, \\ i = 1, 2 & \\ |u_{i,j}| & \leq 2, \\ i = 3, 4, j = x, y & \end{array} \right\} \tag{10}$$

where $R = \mathrm{diag}(1, 1, \dots, 1)$ and $Q = \mathrm{diag}(10^3, 10^3)$. In this problem we have four actuators $\mathcal{P} := \{p_1, \dots, p_4\}$, where the i^{th} actuator has two controls, $u_{i,x} \in \mathbb{R}$ and $u_{i,y} \in \mathbb{R}$, and we have two generalized forces, $\tau_x \in \mathbb{R}$ and $\tau_y \in \mathbb{R}$. Moreover, we define $u := [u_{1,x}, u_{1,y}, \dots, u_{4,x}, u_{4,y}]^T$ and $\tau := [\tau_x, \tau_y]^T$. Looking at the constraints it can be straightforwardly verified that all the actuators are non-interacting. In this example we show two different actuator partitions; first choose the following non-interacting actuator partition $\{P_1, P_2\}$, where $P_1 = \{p_1, p_3\}$ and $P_2 = \{p_2, p_4\}$, yielding $u^1 := [u_{1,x}, u_{1,y}, u_{3,x}, u_{3,y}]^T$, and $u^2 := [u_{2,x}, u_{2,y}, u_{4,x}, u_{4,y}]^T$. Following the proposed procedure we get the following master problem

$$\min_{\tau^1,\tau^2,s} \left\{ s^T Q s + \sum_{j=1}^{2} (\tau^j)^T H^j \tau^j \,\middle|\, \begin{array}{l} \tau = s + \tau^1 + \tau^2 \\ \tau^j \in \mathcal{T}^j, j = 1, 2 \end{array} \right\} \tag{11}$$

where

$$\mathcal{T}^1 = \left\{ \tau^1 \in \mathcal{T} \,\middle|\, \exists u^1 : \begin{array}{c} B^1 u^1 = \tau^1 \\ |u_{1,x}| + |u_{1,y}| \leq 2 \\ |u_{3,i}| \leq 2, i = x, y, \end{array} \right\}$$

where B^1 consists of the column in B that multiply with u^1, and $\mathcal{T}^2$ is found by replacing the appropriate indices, which yields an identical set, i.e. $\mathcal{T}^1 = \mathcal{T}^2$. Moreover, since we have a symmetrical problem, we choose $H^1 = H^2 = \mathrm{diag}(1, 1)$. The first sub-problem becomes

$$\min_{u^1} \left\{ (u^1)^T \mathrm{diag}(1, 1, 1, 1) u^1 \,\middle|\, \begin{array}{c} u_{1,x} + u_{3,x} = \tau^1_x \\ u_{1,y} + u_{3,y} = \tau^1_y \\ |u_{1,x}| + |u_{1,y}| \leq 2 \\ |u_{3,i}| \leq 2, \ i = x, y \end{array} \right\} \tag{12}$$

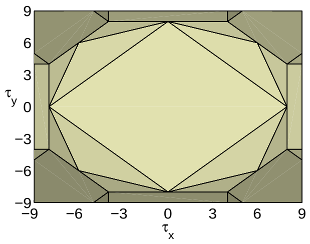
and the second sub-problem is found by replacing the appropriate indices. Let the function $z^*(\cdot) := [s^*(\cdot)^T \ (\tau^1(\cdot))^T \ (\tau^2(\cdot))^T]^T$ denote the PWA solution to the master problem, and $u^1(\cdot)$ and $u^2(\cdot)$ be the solutions to the two sub-problems. The polyhedra that $u^*(\cdot)$, $z^*(\cdot)$ and $u^1(\cdot)$ are defined on are depicted in Figures 1(a)-1(c), respectively. Note also that $u^2(\cdot) = u^1(\cdot)$. Figures 2(a)-2(c) depicts the solutions for the master and two subproblems for the the actuator partition $P_1 = \{p_1, p_2\}$ and $P_1 = \{p_3, p_4\}$. Considering the first actuator partition it is apparent that an explicit solution to the problem can be found by solving two smaller mpQPs (the two sub-problems are identical), but more importantly, one can choose to solve either of the problems on-line, allowing a tradeoff between the online computation time and the required storage space. From this example we see that the proposed strategy provides great flexibility. We have the following alternatives for the first actuator partition:

1) Solving master and sub-problems online.
2) Solving the master problem online and one subproblem explicitly, and since the sub-problems are identical this only yields 13 stored polyhedra.
3) Solving the master problem explicitly and one of the sub-problems online.
4) Solving both the master- and sub-problems explicitly.

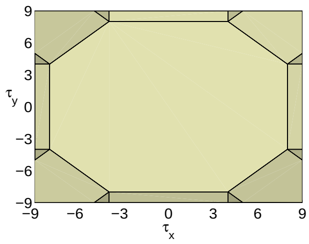
Obviously, we have similar alternatives for the second actuator partition. Finally, note that for the first actuator partition we have $u^1(\cdot) = u^2(\cdot)$ and that the solution to the original problem also has $u^1 = u^2$ (a strictly convex problem where the constraints and weights on u^1 and u^2 are identical.) Thus, if we choose $H^1 = H^2 = \mathrm{diag}(1, 1)$, we have a strictly convex master-problem whose solution is unique ($\tau^1(\cdot) = \tau^2(\cdot)$), hence, $(u^1(\tau^1(\tau)))^T u^1(\tau^1(\tau))) + (u^2(\tau^2(\tau)))^T u^2(\tau^2(\tau)) = (u^*(\tau))^T u^*(\tau)$, i.e. the solution is optimal also for the original problem.

VI. Conclusion

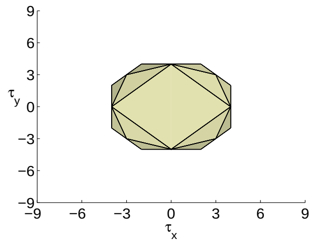
We have proposed a decomposing strategy for linear constrained control allocation problems. The actuators are



(a) The set of polyhedra representing the solution $v^*(\cdot) := (s^*(\cdot), u^*(\cdot))$ to (10).

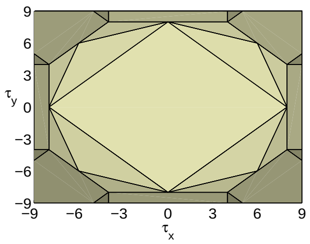


(b) The set of polyhedra representing the solution $z^*(\cdot)$ to (11) with the actuator partition $P_1 = \{p_1, p_3\}$ and $P_2 = \{p_2, p_4\}$.

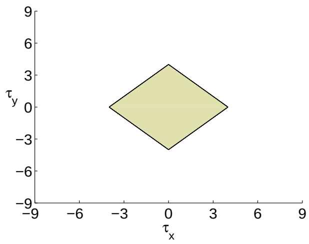


(c) Set of polyhedra representing the solutions $u^1(\cdot)$ and $u^2(\cdot)$ to the first sub-problem (12) and the second sub-problem.

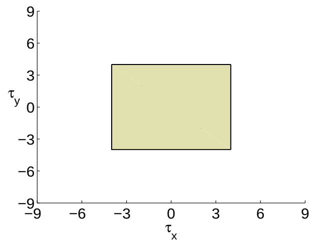
Fig. 1. Explicit solutions with the actuator partition $P_1 = \{p_1, p_3\}$ and $P_2 = \{p_2, p_4\}$.



(a) The set of polyhedra representing the solution $z^*(\cdot)$ to (11) with the actuator partition $P_1 = \{p_1, p_2\}$ and $P_2 = \{p_3, p_4\}$.



(b) Set of polyhedra representing the solution $u^1(\cdot)$ for the first sub-problem, defined by actuator group $P_1 = \{p_1, p_2\}$.



(c) Set of polyhedra representing the solution $u^2(\cdot)$ for the second sub-problem, defined by actuator group $P_2 = \{p_3, p_4\}$.

Fig. 2. Explicit solutions with the actuator partition $P_1 = \{p_1, p_2\}$ and $P_2 = \{p_3, p_4\}$.

partitioned such that a sub-optimal solution can be found be solving a master- and a set of sub-problems. It has also been shown that the decomposing strategy can provide an optimal solution to some classes of allocation problems if the master problem is modified appropriately. The advantages with the scheme is that it allows the designer to choose a mix of online optimization and explicit solutions of the allocation problem, providing a tradeoff between the benefits and drawbacks of the explicit approach.

VII. Acknowledgements

This research is part of the Strategic University Program on Computational Methods for Nonlinear Motion Control funded by the Research Council of Norway. The second author would like to thank the Research Council of Norway for their support. Finally, we would like to thank Dr. S. V. Raković at the Department of Electronic and Electrical Engineering at Imperial College for his comments.

References

[1] M. Bodson, "Evaluation of optimization methods for control allocation," *Journal of Guidance, Control, and Dynamics*, vol. 25, pp. 703–711, 2002.

[2] M. Bodson and W. A. Pohlchuck, "Command limiting in reconfigurable flight control," *Journal of Guidance, Control, and Dynamics*, vol. 21, pp. 639–646, 1998.

[3] J. M. Buffington, D. F. Enns, and A. R. Teel, "Control allocation and zero dynamics," *Journal of Guidance, Control, and Dynamics*, vol. 21, pp. 458–464, 1998.

[4] T. A. Johansen, T. I. Fossen, and P. Tøndel, "Efficient optimal constrained control allocation via multi-parametric programming," *AIAA J. Guidance, Control and Dynamics*, vol. 28, pp. 506–515, 2005.

[5] Y. Luo and D. B. Doman, "Model predictive dynamic control allocation with actuator dynamics," in *Proc. American Control Conference*, Boston, 2004, pp. 1695–1700.

[6] T. A. Johansen, T. P. Fuglseth, P. Tøndel, and T. I. Fossen, "Optimal constrained control allocation in marine surface vessels with rudders," in *IFAC 6th Conference on Manoeuvring and Control of Marine Crafts*, 2003, pp. 215–220.

[7] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

[8] A. Bemporad, F. Borrelli, and M. Morari, "Model predictive control based on linear programming - the explicit solution," *IEEE Trans. on Automatic Control*, vol. 47, no. 12, pp. 1974–1985, 2002.

[9] T. Gal and J. Nedoma, "Multiparametric linear programming," *Management Science*, vol. 18, pp. 406–442, 1972.

[10] V. Dua and E. N. Pistikopoulos, "An algorithm for the solution of multiparametric mixed integer linear programming problems," *Annals of operations research*, vol. 99, pp. 123–139, 2000.

[11] P. Tøndel, T. A. Johansen, and A. Bemporad, "An algorithm for multi-parametric quadratic programming and explicit MPC solutions," *Automatica*, vol. 39, no. 3, pp. 489–497, 2003.

[12] B. Bank, J. Guddat, D. Klatte, B. Kummer, and K. Tammer, *Non-linear Parametric Optimization*. Berlin: Birkhäuser, 1983.

[13] P. Tøndel, T. A. Johansen, and A. Bemporad, "Computation of piecewise affine control via binary search tree," *Automatica*, vol. 39, no. 5, pp. 945–950, 2003.

[14] C. N. Jones, P. Grieder, and S. V. Raković, "A logarithmic-time solution to the point location problem for parametric linear programming," submitted to *Automatica* September 2005.

[15] F. Borrelli, A. Bemporad, and M. Morari, "A geometric algorithm for multi-parametric linear programming," *Journal of Optimization Theory and Applications*, vol. 118, no. 3, pp. 515–540, 2003.

[16] C. N. Jones, E. C. Kerrigan, and J. M. Maciejowski, "Lexicographic perturbation for multiparametric linear programming with applications to control," submitted to *Automatica* September 2005.

[17] J. Spjøtvold, P. Tøndel, and T. A. Johansen, "A method for obtaining continuous solutions to multiparametric linear programs," in *Proc. IFAC World Congress*, Prague, July 2005.

[18] ——, "A continuous selection and unique polyhedral representation of solutions to convex parametric quadratic programs," *Journal of Optimization Theory and Applications*, 2006, accepted.