Control allocation—A survey

Tor A. Johansen, Thor I. Fossen
Department of Engineering Cybernetics, Center for Autonomous Marine Operations and Systems, Norwegian University of Science and Technology, Trondheim, Norway

ARTICLE INFO

Article history:
Received 23 November 2011
Received in revised form
4 July 2012
Accepted 2 November 2012
Available online 21 March 2013

Keywords:
Over-actuated mechanical systems
Effectors
Actuators
Redundant inputs
Optimization
Generalized inverse
Saturation
Constraints
Fault-tolerant control
Re-configurable control

ABSTRACT

The control algorithm hierarchy of motion control for over-actuated mechanical systems with a redundant set of effectors and actuators commonly includes three levels. First, a high-level motion control algorithm commands a vector of virtual control efforts (i.e. forces and moments) in order to meet the overall motion control objectives. Second, a control allocation algorithm coordinates the different effectors such that they together produce the desired virtual control efforts, if possible. Third, low-level control algorithms may be used to control each individual effector via its actuators. Control allocation offers the advantage of a modular design where the high-level motion control algorithm can be designed without detailed knowledge about the effectors and actuators. Important issues such as input saturation and rate constraints, actuator and effector fault tolerance, and meeting secondary objectives such as power efficiency and tear-and-wear minimization are handled within the control allocation algorithm.

The objective of the present paper is to survey control allocation algorithms, motivated by the rapidly growing range of applications that have expanded from the aerospace and maritime industries, where control allocation has its roots, to automotive, mechatronics, and other industries. The survey classifies the different algorithms according to two main classes based on the use of linear or nonlinear models, respectively. The presence of physical constraints (e.g. input saturation and rate constraints), operational constraints and secondary objectives makes optimization-based design a powerful approach. The simplest formulations allow explicit solutions to be computed using numerical linear algebra in combination with some logic and engineering solutions, while the more challenging formulations with nonlinear models or complex constraints and objectives call for iterative numerical optimization procedures. Experiences using the different methods in aerospace, maritime, automotive and other application areas are discussed. The paper ends with some perspectives on new applications and theoretical challenges.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

The objective of this survey is to give an overview of control allocation methods. It is not the objective of this paper to give a complete bibliography, but rather provide a subjective survey with an emphasis on recent developments within a common framework that is independent of the application domains where control allocation is conventionally used. The article is intended to encourage cross-disciplinary transfer of ideas and complement existing overview articles such as (Oppenheimer, Doman, & Bolender, 2010) that focus on aerospace applications of control allocation, and (Fossen & Johansen, 2006) that focus on marine applications. In particular, there has recently been increasing interest in control allocation in the automotive and other industries where mechatronics prevails, which has led to increased research on nonlinear approaches to control allocation. Optimization-based allocation methods are emphasized since their computational complexity is already within the capabilities of today’s off-the-shelf embedded computer technology, e.g. Bodson (2002), Härkegård (2002), Johansen, Fossen, and Berge (2004); Johansen, Fossen, and Tøndel (2005) and Petersen and Bodson (2006).

1.1. Over-actuated mechanical systems

Motion control systems are used to control the motion of mechanical systems such as vehicles and machines. Effectors are mechanical devices that can be used in order to generate time-varying mechanical forces and moments on the mechanical system, such as rudders, fins, propellers, jets, thrusters, and tires. Actuators are electromechanical devices that are used to control the magnitude and/or direction of forces generated by the individual effectors.
By mechanical design, there may be more effectors than strictly needed to meet the motion control objectives of a given application. Hence, in over-actuated mechanical systems, the controllability of the chosen states and outputs would also be achieved with less control inputs. An over-actuated mechanical design may be favorable due to several reasons:

- Need for effector redundancy in order to meet fault tolerance and control reconfiguration requirements.
- It may be desirable to choose a particular set of effectors rather than a smaller set of effectors for reasons such as cost, standardization, size, accuracy, dynamic response, flexibility, maintenance and mechanical design (see e.g. Huston, 2005).
- Certain effectors can be shared among several control systems with different objectives, and therefore be redundant for the given motion control system. For example, a lateral stability control system for a car may use the four individual wheel brake actuators in order to set up a yaw moment while these actuators are primarily designed for the car’s brake system to support the driver’s control of longitudinal motion, see also Valásék (2003).

The design of control algorithms for over-actuated mechanical systems is often divided into several levels. First, a high level control system structure including control allocation. The vector \( \tau \) denotes commanded virtual control effort (generalized forces), while \( \tau \) are the actual allocated control effort.

Third, there may be a separate low-level controller for each effector that controls its actuators in order to achieve its desired force and moment.

The modular structure of the control algorithm is illustrated in the block diagram in Fig. 1. This modularity allows the high-level motion control algorithm to be designed without detailed knowledge about the effector and actuator system. In addition to coordinating the effect of the different effectors in the system, issues such as effector/actuator fault tolerance, redundancy, and control constraints are typically handled within the control allocation module. Note that the effector model (3) is usually chosen to be static, so the low-level actuator control should handle the dynamic control of the actuators. Although the choice of a static effector model is common, it should be mentioned that it is also common that the control allocation algorithm is designed with actuator rate constraints in mind, and we will in later sections survey extensions where more sophisticated dynamic actuator models are integrated with the control allocation algorithm design.

It should also be mentioned that the design on the control allocation algorithm and the high-level motion control algorithm cannot always be independent. For example, it has been illustrated that the zero dynamics of the closed loop may depend on the control allocation design, such that a dynamic inversion type of control design approach may depend on the control allocation design to ensure stable zero dynamics (minimum phase response), see Buffington and Enns (1996) and Buffington, Enns, and Teel (1998). On the other hand, it is also proven that in the framework of optimal control (LQ or nonlinear methods) the high-level motion control and control allocation can be separated (by choosing weight matrices appropriately) with no loss of control performance (Härkégård & Glad, 2005). Moreover, as mentioned above and illustrated in Page and Steinberg (1999), lack of feasibility of the control allocation should be observed and handled by the high-level motion control algorithm in order to avoid unacceptable degradation of performance in such cases.

1.2. Control allocation introduction

The primary objective of a control allocation algorithm is to compute a control input \( u \in \mathbb{U} \) that ensures that the commanded virtual control \( \tau_c = h(u, x, t) \) is produced jointly by the effectors at all time \( t \). This objective could fail to be met if the command \( \tau_c \) require forces beyond the capabilities of the effectors due to saturation or other physical limitations. If a feasible \( u \in \mathbb{U} \) cannot be found, the control allocation algorithm is usually required to degrade its performance objectives and search for a control input \( u \in \mathbb{U} \) that minimizes the allocation error \( \tau - \tau_c = h(u, x, t) \), in some sense. Usually, some kind of priority is involved such that the primary objective can be represented as an optimization problem

\[
\min_{u \in \mathbb{U}, p \in \mathbb{R}^p} \| Q_s \| \text{ subject to } \tau - h(u, x, t) = s, \quad u \in \mathbb{U}
\]

where \( s \) is some slack variable, \( \| \cdot \| \) is some norm, and \( Q \) is some weight matrix that prioritizes the requirements that should be honored in case the commanded virtual input \( \tau_c \) cannot be achieved. With the linear effector model (4) the problem can be written

\[
\min_{u \in \mathbb{U}, p \in \mathbb{R}^m} \| Q_s \| \text{ subject to } \tau - B(x, t)u = s, \quad u \in \mathbb{U},
\]
Actuator rate constraints can be included in the formulation by limiting the change in control $\Delta u$ from the control $u_t$ from the last sampling instant to some set $\Delta u \in C$. This leads to the formulation

$$\min_{u \in \mathbb{R}^n, \tau \in \mathbb{R}^m} \|Qs\| \quad \text{subject to} \quad \tau_c - h(u, x, t) = s,$$

$$u \in U, \quad u = u_t + \Delta u, \quad \Delta u \in C. \quad (7)$$

In over-actuated mechanical systems, $\dim(u) > \dim(\tau)$, such that the solution to (5) or (7) is not unique. This is not a disadvantage, and offers the opportunity to introduce secondary objectives. Often, these are chosen from an operational perspective in order to minimize power or fuel consumption, minimization of actuator/effectror tear and wear, or other criteria. In an optimization framework, this can be formulated by including a secondary cost function $J$ into (5)

$$\min_{u \in \mathbb{R}^n, \tau \in \mathbb{R}^m} (\|Qs\| + J(x, u, t)) \quad \text{subject to} \quad \tau_c - h(u, x, t) = s, \quad u \in U, \quad u = u_t + \Delta u, \quad \Delta u \in C. \quad (8)$$

Some generic cost functions that are commonly used are

$$J(x, u, t) = \frac{1}{2} (u - \bar{u})^T W (u - \bar{u}) \quad (9)$$

$$J(x, u, t) = \|Wu\| \quad (10)$$

where $W \in \mathbb{R}^{p \times p}$ is a positive definite weighting matrix that prioritizes the different effectors, $\bar{u}_p$ is the preferred value of $u$ (like zero deflection of control surfaces, trimmed flight conditions, last value $u_p = u_t$, or zero thrust), and $\| \cdot \|$ usually denotes 1-norm or $\infty$-norm. Both $W$ and $u_p$ may be time-varying, and $W$ should normally be chosen small compared to $Q$ in order to reflect the fact that $J$ represents an objective that is secondary to the primary objective of minimizing the slack variables weighted by $Q$. In later sections we will also comment upon additional and more application specific objectives and constraints, as well as formulations that accommodate alternative models with discrete variables or unknown parameters that need adaptation.

Although effector models must represent the actual physical forces and moments, they can be formulated and parameterized in many ways. Usually, one seeks effector models that are linear with respect to their inputs, leaving nonlinearities to be compensated for by nonlinear mappings, or in the low-level single effector/actuator control through inversion of monotone characteristics or linearizing feedback loops.

Optimization-based control allocation algorithms should be carefully chosen based on the properties of the effector model, objective function, constraints, and computational complexity. Explicit solutions can be found for linear effector models in combination with simple objective functions and constraints, while iterative numerical procedures are required for the more complicated problems. Particular attention will be needed for nonlinear and possibly non-convex optimization problems where issues such as numerical robustness, convergence to non-global minimums, computational complexity and reliability of the numerical software implementation should be addressed. These are key issues in the remaining sections of this survey.

We would like to emphasize that there exist several control allocation approaches that does not directly fit into the optimization formulation above. They will also be treated in this survey, and for consistency of the presentation they will still be presented in the context of optimization problems as far as possible.

### 1.3. Perspectives

This survey focuses on motion control for over-actuated mechanical systems, which is the conventional application area of control allocation. However, the principles of control allocation are general and not limited to motion control systems. Consequently, one does not need to restrict the virtual control input $\tau$ to be interpreted as generalized forces (forces and moments) and they may also represent quantities like energy and mass, for example. In particular, process plants are often characterized by excessive degrees of freedom for control. One example is allocation of gas lift rates in offshore oil production where the petroleum producing wells are coupled due to common pipelines and constraints on the available lift gas resources, (Bieker, Slupphaug, & Johansen, 2007; Camponogara & de Conto, 2005).

In process control, any excessive degrees of freedom are commonly exploited via model predictive control (MPC) and real-time optimization, (Garcia, Prett, & Morari, 1989; Qin & Badgwell, 2003), which are multi-variable optimization-based control strategies where the functionality of control allocation is inherently built into the optimal control formulation that is solved numerically online. Although control allocation usually has less ambitious objectives than MPC – recall the static effector model – we shall in later sections review how MPC can be used to solve the control allocation problem in motion control systems when actuator dynamics should be considered at the control allocation level. However, predictive control allocation has also been proposed in applications like engine management, (Vermillion, Sun, & Butts, 2011).

### 2. Control allocation for linear effector models

Most control allocation algorithms assume a linear effector model in the form

$$\tau = Bu \quad (11)$$

where $B \in \mathbb{R}^{m \times p}$ is a matrix, sometimes called the control effectiveness matrix, that describes the relationship between the control inputs $u$ and the virtual control inputs $\tau$ (forces and moments) produced jointly by the effectors. For many applications, the matrix $B$ will depend on the system state and time-varying parameters or inputs, cf. (4). For example, the aerodynamic lift and drag generated by an aircraft aileron, rudder, flap, spoiler, or other control surface, depends on vehicle velocity and angle of attack, in addition to nonlinearities due to the geometric shape of the control surface, e.g. Da Ronch et al. (2011). Fluid flow conditions of the effectors may depend on both the system state (e.g. velocity) and the effector state (e.g. slip angle or thrust vector direction), and lead to interactions between the effectors or between effectors and the body of the vehicle such as thruster–thruster interactions and thruster–hull interactions, (Fossen, 2011), or change of flow pattern over a control surface (Oppenheimer et al., 2010). For example, deflection of an aerodynamic surface that is upstream of another aerodynamic surface may cause the forces and moments produced by the downstream effector to differ from those produced when the upstream effector is not deployed, (Oppenheimer & Doman, 2007). Such issues will be further discussed in the Section 4 where application specific issues are surveyed. For now, we observe that since the control allocation problem is viewed as a static problem, the control allocation problem can utilize the time-frozen formulation (11) without any regard to how $B$ will change with time, states and inputs. Consequently, the scheduled control effectiveness matrix $B$ is updated at the next sampling instant and the control allocation problem is solved for the new scheduled matrix $B$. Hence, the formulation (11) does not exclude time-varying, parameter-varying, and linearized models. Robustness of the control allocation problem with respect to uncertainty in the scheduling of the $B$ matrix as a function of some operating point variables in discussed in Scottedward Hodel (2000).
The parameterization of the control input, i.e. the choice of elements of the \( u \) vectors in the model, is sometimes of great importance. In particular, it is usually desirable to choose the model such that is has the linear form (11) and thus avoiding more complex nonlinear control allocation methods. For vector thrust devices that can be commanded to produce thrust of varying amplitude and direction (in two or three degrees of freedom), the extended thrust formulation (Lindfors, 1993; Sørdalen, 1997) leads to a linear model. Rather than parameterizing the thrust vector through its amplitude and some angle(s), its thrust components are decomposed on the body axes and treated as elements of the control input. Depending on the actuator system, these components must usually be converted back to amplitude and angle(s) by suitable trigonometric transforms that describe the nonlinear relationships. Although this avoids a nonlinear model, the price to pay may be a more complicated formulation of amplitude and angular rate constraints in the control allocation formulation, e.g. Johansen, Fugleth, Tøndel, and Fossen (2008).

### 2.1. Unconstrained linear control allocation

The main challenge of inverting the model (11) is that \( B \) is not a square matrix. Usually, for an over-actuated system, \( B \) will have full row rank (equal to \( m < p \)) and we will in general assume it has a non-trivial null space. This means there is an infinite number of vectors \( u \in \mathbb{R}^p \) that satisfies (11) for any given \( \tau \in \mathbb{R}^m \). The common way to deal with such extra freedom is to use generalized inverses (or pseudo-inverses), e.g. Golub and Loan (1983) and Horn and Johnson (1985). Below, we present this approach in the context of minimizing a least-squares cost function.

Neglecting any saturation and rate constraints on the input \( u \), and choosing for convenience a quadratic cost function that measures the cost of control action, leads to the control allocation cost function formulation

\[
\min_{u \in \mathbb{R}^p} \frac{1}{2} W (u - u_p)^T W (u - u_p) \quad \text{subject to} \quad \tau_c = Bu \tag{12}
\]

where \( W \in \mathbb{R}^{m \times m} \) is a positive definite weighting matrix, and \( u_p \) is the preferred value of \( u \). When \( B \) has full rank, this weighted least-squares problem has the following explicit solution

\[
u = (I - CB) u_p + C \tau_c \tag{13}
\]

where

\[
C = W^{-1} B^T (BW^{-1} B^T)^{-1}
\]

is a generalized inverse that can be derived from optimality conditions of (12) using Lagrange multipliers, see e.g. Bordignon and Durham (1995), Durham (1993), Enns (1998), Fossen and Sagatun (1991), Oppenheimer et al. (2010), Snell, Enns, and Garrard (1990) and Virnig and Bodden (1994). For the special case \( W = I \) and \( u_p = 0 \), the solution \( u = B^T \tau_c \) is defined by the Moore–Penrose pseudo-inverse, (Golub & Loan, 1983; Horn & Johnson, 1985), given by \( C = B^T (BB^T)^{-1} \). Rank-deficiency of \( B \) means that no force or moment can be generated in certain direction of the space \( \mathbb{R}^m \) where \( \tau_c \) belongs. This means that all commands \( \tau_c \) cannot be achieved, even without considering saturation. Although the mechanical design of the actuators and actuators will normally avoid a rank-deficient \( B \)-matrix, it might appear in special cases like singularities, effector or actuator failures, so the control allocation algorithm should be able to handle it in some applications. Several regularization methods could be applied, like a damped least-squares inverse

\[
C_\epsilon = W^{-1} B^T (BW^{-1} B^T + \epsilon I)^{-1}
\]

where \( \epsilon \geq 0 \) is a small regularization parameter that must be strictly positive when \( B \) does not have full rank. Alternatively, a singular value decomposition (SVD) of the matrix \( BW^{-1} B^T = U \Sigma V^T \) will characterize the directions where no generalized force can be produced, (Golub & Loan, 1983). The matrix \( \Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r) \) contains the singular values. Inverting only the singular values that are non-zero (with some small margin \( \delta > 0 \)), leads to the reduced rank approximation

\[
\Sigma_\delta = \text{diag}(\sigma_1^{-1}, \sigma_2^{-1}, \ldots, \sigma_r^{-1}, 0, \ldots, 0)
\]

where \( r \) is the number of singular values larger than the regularization parameter \( \delta \), i.e. \( \sigma_i \geq \delta \). This leads to the approximate inverse

\[
C_\delta = W^{-1} B^T \Sigma_\delta^{-1} U^T
\]

to be used instead of \( C \) in (13). The SVD can also be used when \( B \) has full rank, e.g. Oppenheimer et al. (2010).

### 2.2. Constrained linear control allocation

The methods based on generalized inverses do not guarantee that constraints on the input \( u \in \mathbb{U} \) are satisfied. A very simple solution to ensure satisfaction of saturation constraints (as well as rate constraints) is to saturate the \( u \) resulting from any of the unconstrained control allocation methods in Section 2.1. Obviously, this will normally mean that the allocated generalized force \( \tau = B \text{Proj}_I(u) \) may be different from the required/-commanded force \( \tau_c \). Furthermore, the method does not guarantee that the allocated generalized force equals the required force whenever possible, or that the error between allocated and required generalized force is minimized in some sense whenever an exact allocation is not possible. In Durham (1993) it is shown that no single generalized inverse (i.e. weight matrix \( W \)) can yield exact allocation whenever possible using simple saturation. Several constrained allocation methods are designed to give better solutions than simple saturation.

#### 2.2.1. Redistributed pseudo-inverse and daisy chaining

The first step of the redistributed pseudo-inverse method (see e.g. Shi, Zhang, Li, & Liu, 2010, Virnig & Bodden, 1994) is to solve the unconstrained control allocation problem, such as (12) (or a simpler version). If the solution satisfies the constraints, no further steps are needed. Otherwise, the unconstrained optimal vector \( u \) is projected onto the admissible set \( \mathbb{U} \) (i.e. saturated) to satisfy the constraints: \( \bar{u} = \text{Proj}_I(u) \). In order to reduce the gap between desired and allocated generalized forces, the unsaturated elements of the control vector \( \bar{u} \) are re-computed by solving a reduced problem using a reduced pseudo-inverse. More specifically, let \( \bar{u} = (\bar{u}_C^T, \bar{u}_U^T)^T \) be decomposed into the saturated elements \( \bar{u}_C \) and unsaturated elements \( \bar{u}_U \), and let \( B = (B_C, B_U) \) be the associated decomposition of the \( B \)-matrix. Then \( \tau_c = B_C \bar{u}_C \) is the allocated generalized force due to the saturated controls, and the remaining controls \( \bar{u}_U \) are redistributed by solving the redistribution equation

\[
B_C \bar{u}_U = \tau_c - \bar{\tau}_C
\]

using the pseudo-inverse method. Then new elements of the sub-vector \( \bar{u}_U \) may be saturated, and the redistribution procedure is repeated until either a feasible solution (that gives exact generalized force allocation) is found, or no further information improvement can be made. Although the method is simple, and often effective, it does neither guarantee that a feasible solution is found whenever possible, nor that the resulting control allocation minimizes the allocation error in some sense. There are examples (see e.g. Bodson, 2002) that demonstrate that clearly sub-optimal control allocation can result.

The daisy chaining method (Adams, Buffington, Sparks, & Banda, 1994; Buffington & Enns, 1996; Oppenheimer et al., 2010)
offers a very simple alternative, but is often less effective than the above mentioned methods. This method groups the effectors into two or more groups that are ranked such that first the control allocation problem is solved for the highest prioritized group. If one or more effectors in that group saturates, the settings of the whole group is frozen. The gap between allocated and required generalized forces is then allocated by the second group. This is then repeated if a feasible solution is still not found, and there are more than two groups. Depending on the selected groups, this may lead to solutions where several effectors may not be fully utilized to minimize the allocation error, and can be sub-optimal compared to the redistributed pseudo-inverse.

### 2.2.2. Direct allocation

Some constrained control allocation methods are based on some scaling of the unconstrained optimal control allocation, such that the resulting control allocation is projected onto the boundary of the set of attainable generalized forces. In aerospace applications this is commonly referred to as the attainable moment set (AMS) since moments in 3-DOF are normally allocated. Here, the set of attainable generalized forces is denoted $\mathcal{A}$, and is the set of vectors $\tau \in \mathbb{R}^n$ when the constrained optimization problem (e.g. (8)) has a feasible solution.

The direct allocation method (Durham, 1993) starts with the unconstrained control allocation computed using some pseudo-inverse, e.g. $\tilde{u} = B^+ \tau_c$. If $\tilde{u} \in U$ (i.e. satisfies the input constraints), no further steps are needed and we use $u = \tilde{u}$. Otherwise the method will search for another $u$ that preserves the direction of $\tau_c$ but leads to an allocated generalized force $Bu$ on the boundary of $\mathcal{A}$:

$$\max \alpha, \quad \text{subject to } Bu = \alpha \tau_c, \quad \alpha \tau_c \in \mathcal{A} \quad (19)$$

where $\alpha \in [0, 1]$ is a scalar. Notice that when the set $U$ is polyhedral, then also $\mathcal{A}$ is a polyhedral set. Solving the optimization problem (19) is not trivial for problems where the dimension of $u$ is large, since there will be a significant amount of facets and vertices and it is not straightforward to identify which facet is intersected by the straight line from $\tau_c$ to the origin. Different numerical algorithms have been suggested, with different computational complexities. Improvements over the original algorithm (Durham, 1993) are based on various data structures, enumerations and representations (Bordignon & Durham, 1995; Durham, 1994a,b, 1999; Petersen & Bodson, 2002) as well as linear programming (Bodson, 2002; Oppenheimer et al., 2010).

### 2.2.3. Error minimization using linear programming

A powerful approach is to explicitly minimize the weighted error between the allocated virtual control input and the desired one. Extending the unconstrained optimization problem formulation (12) with input constraints leads to formulations such as (28). The constraint set $U$ is usually polyhedral, i.e. for some appropriate matrix $A$ and vector $b$ it can be represented as

$$U = \{u \in \mathbb{R}^p \mid Au \leq b\}. \quad (20)$$

Rate constraints $C$ can be formulated as a polyhedral set, too.

With the cost function defined using either 1-norm or ∞-norm, this resulting problem is a linear program (LP) that can be solved using iterative numerical LP algorithms (e.g. Bodson, 2002, Bodson & Frost, 2011, Lindfors, 1993, Paradiso, 1991) by relatively straightforward reformulations into any of the standard LP forms via the introduction of additional variables. As an example, consider the 1-norm control allocation problem subject to

$$Bu = \tau_c + s, \quad u_{\min} \leq u \leq u_{\max}, \quad \delta_{\min} \leq u - u_\ell \leq \delta_{\max}. \quad (22)$$

With symmetric effectors and actuators, we have $u_{\min} = -u_{\max}$ and $\delta_{\min} = -\delta_{\max}$. Introducing auxiliary variables

$$s_i^+ = \begin{cases} s_i & s_i \geq 0 \\ 0 & s_i \leq 0 \end{cases}, \quad s_i^- = \begin{cases} -s_i & s_i \leq 0 \\ 0 & s_i \geq 0 \end{cases}, \quad u_i^+ = \begin{cases} u_i & u_i \geq 0 \\ 0 & s_i \leq 0 \end{cases}, \quad u_i^- = \begin{cases} -u_i & u_i \leq 0 \\ 0 & s_i \geq 0 \end{cases} \quad (23)$$

we have $s_i = s_i^+ - s_i^-, |s_i| = s_i^+ + s_i^-$, $u_i = u_i^+ - u_i^-$, and $|u_i| = u_i^+ + u_i^-$. Stacking these variables into vectors $s, s^-, u^+, u^-$ and defining $w = (w_1, \ldots, w_p)^T$ and $q = (q_1, \ldots, q_m)^T$ we get the following linear program

$$\min \begin{pmatrix} u^T, u^T, q^T, q^T \end{pmatrix} \begin{pmatrix} u^+ \\ u^- \\ s^- \\ s^+ \end{pmatrix} \quad (27)$$

subject to

$$(B, -B, -I, I) \begin{pmatrix} u^+ \\ u^- \\ s^- \\ s^+ \end{pmatrix} = \tau_c \quad (28)$$

$$\begin{pmatrix} I, -I, 0, 0 \\ -I, I, 0, 0 \end{pmatrix} \begin{pmatrix} u^+ \\ u^- \\ s^- \\ s^+ \end{pmatrix} \geq \begin{pmatrix} \max(u_{\min} - \delta_{\min} + u_\ell) \\ \min(u_{\max} - \delta_{\max} + u_\ell) \end{pmatrix} \quad (29)$$

Other LP standard forms exists, and similar reformulations can be made, e.g. Bodson (2002). The use of ∞-norm will minimize the maximum effector use and therefore lead to a balanced use of effectors, (Bodson & Frost, 2011; Frost & Bodson, 2010; Frost, Bodson, & Acosta, 2009), and can also be reformulated into linear programs using similar augmentations with auxiliary variables.

The use of slack variables in the above formulations ensures that a feasible solution always exists. It should be mentioned that infeasibility handling can also be included via a two-level approach (e.g. Bodson, 2002, Oppenheimer et al., 2010). While this might lead to reduced computational complexity on average, it may not contribute to reduced worst-case computational complexity that is usually the main concern in real-time implementations.

The most common numerical methods for linear programming are the simplex method, active set methods and interior point methods, (Nocedal & Wright, 1999). The simplex method is studied for control allocation problems in Bodson (2002), where the main conclusions are that the computational complexity is clearly within the capabilities of current embedded computer hardware technology. The simplex method iterates between vertices of the polyhedral set describing the set of feasible solutions, where at each iteration a system of linear equations corresponding to a basic solution is solved using numerical linear algebra. Since there is a finite number of basic solutions, the simplex algorithm is a combinatorial approach that finds the optimal solution in a finite number of iterations. The simplex algorithm usually beats the combinatorial complexity of the problem by trying to reduce the cost function at each iteration. Many numerically robust
implementations of the simplex method exists, including portable C code, (Press, Flannery, Teukolsky, & Vetterling, 1992), which makes the approach fairly straightforward to apply in many embedded control platforms. However, there are some issues that may require special attention.

Although the simplex method tends to converge to the optimal solution within a number of iterations that is not bigger than the number of variables and constraints, (Press et al., 1992), it is hard to give a guaranteed limit on the number of iterations. Hence, the control allocation may have to accept some degree of sub-optimality since only a limited number of iterations may be allowed in a real-time implementation.

Degeneracies in the problem are characterized by constraints that are redundant, (Nocedal & Wright, 1999). They may lead to non-uniqueness and singular linear algebraic inversion problems that in combination with numerical inaccuracies may require some additional considerations. Due to degeneracies, a change in the basic solution during one iteration may lead to a new basic solution where the cost remains the same. If particular care is not taken, a phenomena called cycling may arise, where the solver jumps back and forth between the same set of basic solutions forever without making any progress towards the optimum. As observed in Bodson (2002), anti-cycling procedures are indeed needed since non-uniqueness and singular linear algebraic inversion problems that are redundant, (Nocedal & Wright, 1999). Theymay lead to back and forth between the same set of basic solutions forever, (Press et al., 1992), it is allowed in a real-time implementation.

Degeneracies in the problem are characterized by constraints that are redundant, (Nocedal & Wright, 1999). They may lead to non-uniqueness and singular linear algebraic inversion problems that in combination with numerical inaccuracies may require some additional considerations. Due to degeneracies, a change in the basic solution during one iteration may lead to a new basic solution where the cost remains the same. If particular care is not taken, a phenomena called cycling may arise, where the solver jumps back and forth between the same set of basic solutions forever without making any progress towards the optimum. As observed in Bodson (2002), anti-cycling procedures are indeed needed since non-uniqueness and singular linear algebraic inversion problems that are redundant, (Nocedal & Wright, 1999). Theymay lead to back and forth between the same set of basic solutions forever, (Press et al., 1992), it is allowed in a real-time implementation.

Degeneracies in the problem are characterized by constraints that are redundant, (Nocedal & Wright, 1999). They may lead to non-uniqueness and singular linear algebraic inversion problems that in combination with numerical inaccuracies may require some additional considerations. Due to degeneracies, a change in the basic solution during one iteration may lead to a new basic solution where the cost remains the same. If particular care is not taken, a phenomena called cycling may arise, where the solver jumps back and forth between the same set of basic solutions forever without making any progress towards the optimum. As observed in Bodson (2002), anti-cycling procedures are indeed needed since non-uniqueness and singular linear algebraic inversion problems that are redundant, (Nocedal & Wright, 1999). Theymay lead to back and forth between the same set of basic solutions forever, (Press et al., 1992), it is allowed in a real-time implementation.

Degeneracies in the problem are characterized by constraints that are redundant, (Nocedal & Wright, 1999). They may lead to non-uniqueness and singular linear algebraic inversion problems that in combination with numerical inaccuracies may require some additional considerations. Due to degeneracies, a change in the basic solution during one iteration may lead to a new basic solution where the cost remains the same. If particular care is not taken, a phenomena called cycling may arise, where the solver jumps back and forth between the same set of basic solutions forever without making any progress towards the optimum. As observed in Bodson (2002), anti-cycling procedures are indeed needed since non-uniqueness and singular linear algebraic inversion problems that are redundant, (Nocedal & Wright, 1999). Theymay lead to back and forth between the same set of basic solutions forever, (Press et al., 1992), it is allowed in a real-time implementation.

Degeneracies in the problem are characterized by constraints that are redundant, (Nocedal & Wright, 1999). They may lead to non-uniqueness and singular linear algebraic inversion problems that in combination with numerical inaccuracies may require some additional considerations. Due to degeneracies, a change in the basic solution during one iteration may lead to a new basic solution where the cost remains the same. If particular care is not taken, a phenomena called cycling may arise, where the solver jumps back and forth between the same set of basic solutions forever without making any progress towards the optimum. As observed in Bodson (2002), anti-cycling procedures are indeed needed since non-uniqueness and singular linear algebraic inversion problems that are redundant, (Nocedal & Wright, 1999). Theymay lead to back and forth between the same set of basic solutions forever, (Press et al., 1992), it is allowed in a real-time implementation.

Degeneracies in the problem are characterized by constraints that are redundant, (Nocedal & Wright, 1999). They may lead to non-uniqueness and singular linear algebraic inversion problems that in combination with numerical inaccuracies may require some additional considerations. Due to degeneracies, a change in the basic solution during one iteration may lead to a new basic solution where the cost remains the same. If particular care is not taken, a phenomena called cycling may arise, where the solver jumps back and forth between the same set of basic solutions forever without making any progress towards the optimum. As observed in Bodson (2002), anti-cycling procedures are indeed needed since non-uniqueness and singular linear algebraic inversion problems that are redundant, (Nocedal & Wright, 1999). Theymay lead to back and forth between the same set of basic solutions forever, (Press et al., 1992), it is allowed in a real-time implementation.
Interior-point methods, on the other hand, replaces the inequality constraints with a barrier function that prevents the solution for going into the infeasible region, (Nocedal & Wright, 1999). Newton’s method is then applied to search towards the optimum of the unconstrained optimization problem resulting from this reformulation. For each iteration, the barrier function is reduced in order to allow the solver to approach the boundary of the feasible region in case the optimal solution is located there.

The active set methods tend to perform well in control allocation problems, (Härkegård, 2002), while interior point methods have their advantage for larger-scale problems (Petersen & Bodson, 2005). Active set methods have the advantage that their initialization can take advantage of the solution from the previous sample (known as warm start), which is often a good guess for the optimal solution at the current sample. This may reduce the number of iterations needed to find the optimal solution in many cases. Interior-point methods are generally initialized with points near the center of the feasible region and will always need a minimum number of iterations in order to converge due to the need to reduce the barrier function penalty in several steps. Warm start procedures are therefore difficult to implement for interior-point methods.

Like in LP, it is hard to give guarantees on the maximum number of iterations and computation time needed to find the optimal solution. Hence, some degree of sub-optimality may need to be accepted in order to respect limitations on computational resources in order to meet real-time computation constraints. Numerical challenges with degeneracies and cycling must be addressed also in QPs.

Several implementations of active set QP solvers were studied for control allocation problems in Härkegård (2002), with fairly modest differences in computational complexity being observed. For non-real-time implementations, the Matlab toolbox QCAT (Härkegård, 2004b) is dedicated to QP-based control allocation and implements the methods of Härkegård (2002) and Petersen and Bodson (2005). For real-time implementation on embedded systems, there exists a few portable C code solvers such as the FORTRAN-to-C-converted active set solver QLD (Schittkowski, 1986), the interior-point method automatic code generation tool CVXGEN (Mattingley & Boyd, 2010), the active-set-like solver QPOASES (Potschka, Kirches, Bock, & Schlöder, 2010), and the conjugate gradient method, (Press et al., 1992). Recently, real-time certification and software for automated C-code generation of first-order (fast gradient) methods have become available, (Richter, Jones, & Morari, 2012).

2.2.5. Explicit approaches to constrained error minimization

Both linear and quadratic program formulations admit an explicit representation of the solution as a piecewise linear (PWL) solution function of $r$. This PWL function is determined using multi-parametric quadratic programming, (Johansen et al., 2005; Kvasnica, Grieder, & Baotić, 2004). Although the resulting PWL solution function may contain a large number of linear function pieces, each defined over a polyhedral region in $A$, the online computations required to evaluate the PWL function are dramatically reduced by data structures such as binary search trees, (Johansen et al., 2005), and lattice representations, (Bayat, Johansen, & Jalali, 2012). Still, the online computer memory requirements may limit the applicability of this approach to problems where the $B$-matrix and constraint sets $U$ and $C$ are time invariant, and the requirements for fault tolerance and reconfigurability are simple, see e.g. Spijtvold and Johansen (2009). A decomposition approach that groups effectors into non-interacting groups can sometimes reduced the complexity through master- and slave-problems (Spijtvold, Tønnel, & Johansen, 2006).

2.2.6. Dynamics and fault tolerance

An optimization-based control allocation method that is integrated with a parameter estimation scheme is described in Casavola and Garone (2010). It leads to an adaptive solution that can accommodate an unknown time-varying $B$-matrix due to losses and faults. Control allocation is an effective approach to implement fault tolerant control. When effector or actuator faults are identified, they can be modeled as changes in the $B$-matrix of the constraints, or other parameters in the optimization problem. For example, an actuator that is locked in a faulty position could be systematically treated by setting the lower and upper constraint limits to the locked value. Alternatively, the preferred control vector could also be set to locked actuator values as proposed in Schiermam et al. (2004). A systematic method that also changes the weights in the pseudo-inverse in order to ensure that faults are well distributed among the fault-free effectors without reconfiguring the high level controller is proposed in Atwi and Edwards (2008).

A dynamic control allocation approach is presented in Zaccarian (2009). It is designed to allocate the required control effort, while allocating the excessive degrees of freedom through a dynamic system that can be tuned for optimizing secondary objectives and constraints.

It is relatively straightforward to design a basic control allocation algorithm to comply with actuator rate constraints by incorporating this as a constraint or penalty on the change in control inputs from the previous sample to the current sample, see (7) and Härkegård (2004a). More sophisticated dynamic actuator models may be incorporated by using the MPC framework to solve the constrained control allocation problem (Luo, Serrani, Yurkovich, Domant, & Oppenheimer, 2004, 2005; Vermillion, Sun, & Butts, 2007). MPC is an optimization-based control approach which can be used in control allocation, being able to handle actuator dynamics as well as actuator saturation. MPC is a systematic design method that utilizes a model of the plant for predicting outputs and states. In control allocation this model describes the actuator dynamics. Using MPC the control allocation problem is solved on a future horizon, and the optimal solution is a future trajectory. Because of the predictive nature of the controller, the calculated control can pre-act to the actuator system dynamics to improve dynamic performance. On the negative side, MPC allocation requires significantly more computations than the static control allocation formulation since the number of optimization variables and constraints is a multiple of the prediction horizon, which may be a factor of 10-20 larger compared to the static problem. Still, it has been demonstrated in Hanger, Johansen, Mykland, and Skullestad (2011) that with efficient numerical QP solver software (Mattingley & Boyd, 2010) that the real-time computations of MPC allocation with a linear dynamic actuator model can also be implemented with current off-the-shelf computer technology.

A similar strategy, that considers the current and past history of commanded virtual forces and moments, is given in Venkataraman, Oppenheimer, and Doman (2004). There, the control allocation problem is to solve for the control inputs so that some norm of the error between the achieved and desired moments is minimized. A computationally simpler strategy to include actuator constraints in the control allocation is via post-processing of the static control allocation, as proposed in Oppenheimer and Doman (2004). The post-processor will overdrive the actuator in order to compensate for the dynamics of a first or second order linear actuator model.

3. Control allocation for nonlinear effector models

The general control allocation formulations (8) allows for a nonlinear effector model $h$, a non-quadratic cost function $J$, and
constraint sets that are not polyhedral. Such nonlinear formulations may be necessary in order to achieve the desired performance, for example when time-varying linearization does not provide sufficiently accurate approximations, convex polyhedral sets do not accurately describe the effector limitations, or more complex cost functions are needed. Typical examples of nonlinearities are physically motivated cost functions, such as power consumption of a marine propeller that is a cubic function of its angular speed and a non-integer power function of its pitch angle, (Fossen, 2011). Other examples are rudders–propeller pairs of ships, where the rudder is effective only when the propeller produces thrust forward, leading to non-convex constraint sets, (Johansen et al., 2008). Tires have highly nonlinear characteristics with saturations at high longitudinal and lateral slips, which may lead to nonlinear effector models, e.g. Tjonnåsand Johansen (2010). Nonlinearities are discussed further in Section 4 when describing application domains.

3.1. Nonlinear programming methods

The use of nonlinear programming for control allocation was proposed in Johansen et al. (2004) and Poonamallee, Yurkovich, Serrani, Domanski, and Happenheimer (2004). In Johansen et al. (2004) it was shown how formulations similar to (8) could be addressed by locally approximating the cost function by a quadratic cost function, and linearizing the constraints. This leads to a numerical method similar to sequential quadratic programming (SQP), except that the linear/quadratic approximation was made only once per sample. More specifically, consider the following nonlinear program resulting from a control allocation formulation

\[ \min f(\xi), \quad \text{s.t.} \quad F(\xi) = 0, \quad G(\xi) \geq 0. \tag{36} \]

Algorithm: Sequential linearization and QP.

Initialize: Let \( \xi^0 \) be a feasible starting point for (36), possibly based on the solution from the previous sample. 

Main computations:

1. Compute a 2nd order truncated Taylor expansion of \( f(\xi) \) around \( \xi^0 \) using analytical or numerical differentiation.
2. Compute first order truncated Taylor expansions of \( F(\xi) \) and \( G(\xi) \) around \( \xi^0 \) using analytical or numerical differentiation.
3. Set \( z = \xi - \xi^0 \), and define the matrices of a QP approximation in the form (35) of the nonlinear program.
4. Solve the QP using an active set algorithm, and define an improved solution \( \xi^1 = \xi^0 + z \). 

In Johansen et al. (2004), this strategy was studied in an example and shown to produce adequate results, with only minor increase in computational complexity (due to linearization and quadratic approximation) compared to a quadratic programming approach to linear control allocation. However, this conclusion cannot be expected to generalize to arbitrary nonlinear control allocation problems. In particular, applications that require very large changes in allocated forces from one sample to the next may require several linear/quadratic approximations to be computed sequentially in order to achieve the necessary accuracy, see Poonamallee et al. (2004). In a full SQP implementation, e.g. Nocedal and Wright (1999), the linearization and QP steps in the above algorithm are repeated iteratively during one sampling instant until the optimality conditions are satisfied. With \( N \) iterations, the computation times would be roughly speaking \( N \) times the computation time of the sequential linearization and quadratic programming algorithm above.

Applications with strong nonlinearities may lead to non-convex cost or constraint functions such that the optimization may get stuck in local minimums that may severely degenerate performance, or require additional computational mechanisms in order to find close to global optimal solutions. Unlike the linear control allocation case, there is little hope to find a general-purpose nonlinear programming algorithm and numerical software implementation for general nonlinear allocation problems.

While a general nonlinear optimization framework can accommodate any cost function, any model and any type of constraints, it is of great interest to study control allocation for specific classes of nonlinearities and constraints. By exploiting structural properties one may pursue the analysis of theoretical properties such as guaranteed convergence to optimal solutions without excessive amount of computations.

3.2. Mixed-integer programming methods

One particular model class leading to control allocation problem formulations that can be solved using mixed-integer linear programming (MILP) are piecewise linear functions (Bolender & Doman, 2004). Generally, a control allocation problem based on a piecewise linear effector model, piecewise linear cost function, and a constraint set that can be described as the non-convex union of polyhedral sets can be formulation as an MILP, see Bemporad and Morari (1998) and Heemels, Schutter, and Bemporad (2001) for equivalence classes and how to formulate MILPs. While numerical MILP solvers are highly complex numerical software systems that may be difficult to verify and validate for use in a safety-critical real-time application, it can be noted that it has been demonstrated that simple enumeration methods in combination with numerical quadratic programming can be effective for solving practical non-convex control allocation problems where non-convex constraint sets are represented as the union of a small number of polyhedral sets, (Johansen et al., 2008; Ruth & Sørensen, 2009a).

3.3. Dynamic optimum-seeking methods

In Johansen (2004) it was proposed to re-formulate the static nonlinear optimization formulations of control allocation, i.e. (5), (7), or (8), as a control Lyapunov-function and use constructive Lyapunov-design methods. In particular, it was assumed that the cost function \( f(x, u, t) = J(x, u, t) + \rho(u) \) was augmented with a barrier or penalty-function \( \rho(\cdot) \) in order to enforce that input constraints \( u \in \mathbb{U} \) are satisfied. Then the corresponding Lagrange function is formulated, with \( \lambda \in \mathbb{R}^m \) being the vector of Lagrange multipliers

\[ L(x, u, t, \lambda) = f(x, u, t) + \lambda^T \left( t_e - H(u, x, t) \right). \tag{37} \]

Assuming that a Lyapunov function \( V_0(x, t) \) for the high level motion control algorithm exists, the following control Lyapunov function is defined for the control allocation design

\[ V(x, u, t, \lambda) = \sigma V_0(x, t) + \frac{1}{2} \left( \frac{\partial L^T}{\partial u} \frac{\partial L}{\partial u} + \frac{\partial L^T}{\partial \lambda} \frac{\partial L}{\partial \lambda} \right), \tag{38} \]

for some \( \sigma > 0 \). Requiring a negative time-derivative of \( V \) along trajectories of the system, one can derive a control allocation update algorithm on the form

\[ \dot{u} = -\Gamma \alpha + \xi, \quad \dot{\lambda} = -K \beta + \phi \tag{39} \]

where \( \Gamma \) and \( K \) are symmetric positive definite gain matrices, and \( \alpha, \xi, \beta \) and \( \phi \) are signals defined in Johansen (2004). The control allocation update law (39) will asymptotically track the optimal control allocation (assuming feasibility), while guaranteeing not to destabilize the closed-loop system. Notice that the latter is not an obvious feature due to the fact that this dynamic control allocation is only asymptotically optimal, and may deviate at every instant until the optimality conditions are satisfied. With \( N \) iterations, the computation times would be roughly speaking \( N \) times the computation time of the sequential linearization and quadratic programming algorithm above.
time-instant from the instantaneous optimal allocation of the corresponding static control allocation problem. This is leading to some loss of performance as shown in a case study in Tavasoli and Naraghi (2011). The main advantage of the method is that no direct numerical optimization is needed (optimality tracking is built into the dynamic update law (39)) leading to modest computational complexity. Disadvantages of the method include possible convergence problems in case of non-convex cost function and constraints, similar to the nonlinear programming approach.

Actuator rate constraints can in some case be enforced and implemented by choosing the gain \(Γ\) sufficiently small, although there is no guarantee that they can be met if \(Γ\) is not small when the high level motion control algorithm requires fast changes in the virtual control that cannot be implemented with the given actuator system. An extension that leads to convergence to optimal control allocation in finite-time was proposed in Liao, Lum, Wang, and Benosman (2007), and the effects of internal dynamics and minimum phase properties when using dynamic inversion high level motion controllers were studied in Benosman, Liao, Lum, and Wang (2009) and Liao et al. (2007).

The concept relies on a control Lyapunov-function, which allows for certain extensions to be made within the same framework. An adaptive approach where uncertain parameters \(θ\) in the effector model \(h(u, x, t, θ)\) are stably adapted using an adaptation law that is designed by augmenting the control Lyapunov-function in a standard way was proposed in Tjønnås and Johansen (2005, 2008). This framework was further extended to dynamically account for actuator dynamics within the control allocation in Tjønnås and Johansen (2007a,b), and internal dynamics in the context of model reference adaptive control (Liao, Lum, Wang, & Benosman, 2009a,b).

### 3.4. Direct nonlinear allocation

An extension of the method of attainable moment set computations and direct allocation for nonlinear effector models can be found in Bolender and Doman (2004b). The methods rely on nonlinear programming and the ideas in this paper can be traced back to Doman and Sparks (2002) and Venkataraman and Doman (2001).

### 4. Applications

#### 4.1. Aerospace

In this section we consider aircraft and spacecraft control allocation problems separately.

##### 4.1.1. Aircraft

In flight control applications the virtual control input \(τ\) is usually the moments about the roll, pitch and yaw axes. Conventional fixed-wing aircraft design and flight control systems are based on a relatively small number of control surfaces (effectors) that are dedicated to control each axis of the aircraft, like:

- ailerons for roll control,
- an elevator for pitch control, and
- a rudder for yaw control.

The grouping of two or more control surfaces into a single effector by constraining them to move together is common in flight control, and is often called ganging, e.g. Oppenheimer et al. (2010). Typically, left and right ailerons are constrained to deflect differentially, while right and left elevators are constrained to deflect equally. Assuming the actuators and effectors are fault-free and the above ganging scheme is used, even with five control surfaces there are only three effective effectors available to control the three axes and control allocation is not needed. However, many current aircraft designs have a larger number of control surfaces that can be used during normal or special conditions, such as vertical take-off-and-landing, or after failure of an actuator or effector. Depending on the type of aircraft, one may have many more effectors including:

- V-tails that give coupled lateral and longitudinal forces,
- control surfaces like flaps, spoilers, and slats,
- tiltable propellers, and thrust vector jets.

Control allocation is widely used with such designs in order to ensure optimal use of the effectors, including fault tolerant and robust control over a wide flight envelope, (Bolling & Durham, 1997; Burken et al., 2001; Davidson, Lallman, & Bundick, 2001; Durham, 1993; Huang, Liu, & Zhu, 2009). It is concluded in Burken et al. (2001) that although simulations demonstrate success of the conventional flight control approach in many cases, the control allocation approach appears to provide uniformly better performance in all cases.

Effector models for aerospace applications are usually assumed in the linear form (11). As discussed in Section 2, the control effectiveness matrix \(B\) may depend on slowly varying variables such as altitude and velocity, and is therefore scheduled as a function of these variables. It is also worthwhile to remark that nonlinear effector models tend to be better approximated using an affine model \(τ = Bu + b\) instead of the linear model (11), see Domon and Oppenheimer (2002). The extensions are straightforward, so most control allocation design work proceed without loss of generality with the model (11).

All the (constrained) linear control allocation methods described in Section 2 are commonly found in the flight control literature, e.g. Oppenheimer et al. (2010). The models and constraints are generally given by the physical characteristics of the effectors and actuators, while the choice of \(u_p\) and weighting matrices may reflect different objectives such as:

- minimum wing loading,
- minimum control surface deflection,
- minimum radar signature,
- minimum drag,
- maximum lift, and
- rapid reconfigurability for fault tolerance,

and others, e.g. Yang, Zhong, and Shen (2009). Using the pseudo-inverse solution as a preference vector \(u_p\) allows one to analytically represent the control allocator in a robustness analysis of the system that is valid as long as no single axis is saturated and the commanded accelerations are feasible, (Schierman et al., 2004).

This facilitates the verification and validation process that must be completed prior to flight testing when using optimization based control allocation methods.

A comprehensive comparison of performance of several state-of-the-art linear control allocation methods are provided by Bodson (2002). One main conclusion is that the optimization-based methods tend to outperform the alternative methods proposed in the literature both in terms of avoiding unnecessary infeasibility and minimizing the use of control effort. The 2-norm (quadratic) formulation seems to be favorable over the 1-norm (linear) formulation since the solution tends to combine the use of all control surfaces (rather than just a few), (Petersen & Bodson, 2006).

The use of \(∞\)-norm will minimize the maximum effector use and therefore lead to a balanced use of effectors, which also has advantages for robustness to failure and nonlinearities, (Bodson & Frost, 2011; Frost & Bodson, 2010; Frost et al., 2009).

The underlying motivation for the direct control allocation method, (Durham, 1993), is that in many applications (in particular
Aircraft) it is considered important to keep the direction of the allocated forces and moments equal to the command, in order to get graceful degradation of performance and handling qualities. Hence, it is practically motivated by a different objective than merely minimizing the error in allocated generalized forces, which may be important in some flight control applications.

Although the control allocation problem is in most cases decoupled from the high level motion control design, there may be cases when the interactions between control and control allocation should be studied or accounted for. This includes cases when actuator dynamics is significant, e.g. Oppenheimer et al. (2010), or the control allocation influences the zero-dynamics due to inversion-based control, e.g. Buffington and Enns (1996) and Buffaloing et al. (1998). An integrated approach to flight control and control allocation design is investigated in Jung, Lowenberg, and Jones (2006), while Boskovic and Mehra (2002) describes two control allocation methods that adapt the weight matrices of a pseudo-inverse like control allocation law in order to avoid saturation and rate constraints. Control allocation methods that explicitly take into account linear actuator dynamics have been proposed, using linear constrained MPC, (Luo et al., 2004, 2005; Luo, Serrani, Yurkovich, Oppenheimer, & Domani, 2007), or optimization in the framework of linear matrix inequalities (LMIs), (Kishore, Sen, Ray, & Ghoshal, 2008). In order to reduce effective time delay due to actuator rate limitations in case of quickly changing commanded moments, and thereby reduce the risk of pilot induced oscillations, Yildiz and Kolmanovsky (2011) proposed and studied a control allocation method that also penalizes the difference between the time-derivatives of the commanded and allocated moments.

A linear programming approach to control allocation that accounts for interaction between control effectors due to aerodynamic couplings are studied in Oppenheimer and Domani (2007). Constrained control allocation using nonlinear effector models have been studied using numerical nonlinear programming methods, (Davidson et al., 2001; Ma, Li, Zheng, & Hu, 2008).

Reconfiguration of the control allocation, via weight modifications (Zhou, Zhang, Rabbath, & Theilliol, 2010) or adaptation of effector model (Zhang, Suresh, Jiang, & Theilliol, 2007), was considered in order to manage faults. Methods for stable adaptation of parameter uncertainty in the effector models due to failures, and associated control allocation strategies that will group effectors into a smaller set equivalent effectors, similar to ganging or daisy chaining, is studied in Liu and Crespo (2010). Model predictive control is shown to be a powerful tool to model failures and a suitable basis for fault tolerant control allocation in Joosten, van den Boom, and Lombaerts (2008).

4.1.2. Spacecraft

Spacecraft may have other actuators and effectors, either instead of, or in addition to control surfaces. These include reaction control jets, and reaction wheels. In addition, the energy consumption for control is generally a high priority objective of the control system design of spacecraft and will often need to be strongly considered in the control allocation strategy as well.

Flight tests with the Boeing X-40A reusable launch vehicle using reconfigurable control allocation based linear programming are reported in Schierman et al. (2004). In case of faulty locked actuators, it is in Schierman et al. (2004) proposed to set the associated elements of the preferred control vector $u_i$ to the locked actuator positions.

Fault tolerant control allocation for a planetary entry vehicle was investigated in Marwaha and Valasek (2008), using mixed-integer linear programming of handle the quantized/discrete nature of pulsed reaction control jets. Domani, Gamble, and Ngo (2009) propose a control allocation approach to optimally combine the use of (discrete) pulsed reaction control jets with (continuous) control surfaces in spacecraft transitions from exo-atmospheric to endo-atmospheric flight, using mixed-integer linear programming.

Satellite systems often have redundant thrusters, where it is desirable to minimize energy consumption during a maneuver or attitude control. A linear programming control allocation approach is investigated in Jin, Park, Park, and Tahk (2006), where it is shown that it can reduce energy consumption compared to a simpler grouping strategy. A multi-saturation based model for a highly non-linear allocation function of micro-thrusters in a satellite is presented in Boada, Prieur, Tarbouriech, Pittet, and Charbonnel (2010). Quadratic programming is used for constrained thrust allocation of redundant satellites in Fu, Cheng, Jhang, and Yang (2011), with particular emphasis on fault tolerant reconfigurable control.

4.2. Ships, underwater vehicles and offshore vessels

In this section we first consider surface vessels in low-speed regimes where control surfaces are not effective unless actively excited by some generated flow. Next, we consider surface vessel maneuvering at high speed where control surface lift becomes significant, before we survey control allocation in multi vessel operations and underwater vehicles.

4.2.1. Station-keeping and low-speed maneuvering

Several types of ships and specialized vessels, such as semi-submersible platforms used in the petroleum industry, depend on thrust allocation control systems during certain modes of operation. This is in particular the case during dynamic positioning operations that include station keeping and low-speed maneuvering using joystick control or automatic tracking functionality, (Fossen, 2011; Sørensen, 2011). Such control systems often control the vessel in three degrees of freedom (surge, sway and yaw) and command the required surge and sway forces as well as yaw moment to the thrust allocation system. Dynamic positioning operations include drilling, offloading of cargo or petroleum at oilfield installations, pipe-laying, cable laying, seismic data acquisition, dredging, fire fighting and rescue, construction, diving support, and others. Thrust allocation for low-speed maneuvering is used for vessels ranging from cruise vessels, ferries, and tankers to smaller yachts, research and fishing vessels.

The thruster system can be implemented using various thrust producing devices that are effective in the low-speed regime:

- Main propellers gives positive or negative force in the longitudinal direction only, and possibly a small yaw moment if mounted off the longitudinal axis. The propeller thrust is usually controlled through its angular speed, a variable pitch angle, or both.
- Main propellers with rudders gives positive or negative force in the longitudinal direction. In addition, the rudder angle can be controlled to produce lateral forces and yaw moment when the propeller thrusts forward since the propeller slipstream is directed to flow at high speed past the rudder surface and can therefore produce a significant lateral force. When the propeller thrusts backwards, the rudder is not effective.
- Tunnel thrusters are propellers mounted in the lateral direction in tunnels through the ship hull. They produce lateral forces and yaw moments.
- Azimuth thrusters are propellers that can be turned to produce thrust in any direction in the horizontal plane. The propeller thrust is usually controlled through its angular speed, a variable pitch angle, or both. Since they are vector thrust devices, an azimuth thruster has two-degrees-of-freedom for the control allocation.
Thrusters are commonly powered by electricity distributed from a power plant that may comprise one or more diesel-engine or gas turbine electric generators. Main propellers are sometimes directly driven by the engine. Safety and operational requirements require a high degree of redundancy to achieve the necessary fault tolerance. Typical requirements is that operations can continue uninterrupted for some time to allow them to be aborted safely after major failures such as loss of a single thruster, single generator set, a single electric switchboard, or a single engine room due to fire or flooding in a single compartment. Often, the worst case single point failure is loss of half of thrust capacity due to a switchboard short circuit failure, or fire or flooding in a machine room. Advanced vessel design with high redundancy tend to have four to eight thrust producing devices, where some are azimuth thrusters with two independent degrees-of-freedom for control. The thrust allocation algorithm therefore has many degrees of freedom in order to be capable to handle critical failures. The thruster system capacity is usually designed based on vessel capability requirements to withstand environmental forces such as wind, waves and currents, (Ruth & Sørensen, 2009b). Usually, wind loads are dominating.

Thrust allocation objectives and constraints that are commonly accounted for, (Berge & Fossen, 1997; Fossen, 2011; Johansen et al., 2008; Lindfors, 1993; Ruth, Sørensen, & Perez, 2007; Sørdalen, 1997; Sørensen & Ædnanes, 1997; Webster & Sousa, 1999), include the following:

- Surge, sway and yaw control, usually with a priority on the yaw axis since loss of heading will usually imply loss of position under heavy wind conditions since ships are designed for minimum wind loads when heading up against the wind.
- Thrusters have individual capacity constraints due to their power rating, but may also have coupled constraints if limited by the electric power available on a shared power bus.
- Rate constraints are generally important for the turning of azimuth thrusters and rudders’ steering machine.
- Minimization of fuel consumption.
- Minimization of tear-and-wear on thrusters and generator sets due to time-varying control commands that must respond to the motion of the vessel caused by wind and waves.
- Avoiding too high variations in electric power consumption that may cause blackout due to over- or under-frequency protection of the weak electric power grid on an isolated ship or vessel.
- Sector constraints are sometimes imposed on azimuth thrusters in order to protect equipment (like subsea equipment lowered through a moon pool, or hydro-acoustic transceivers used for positioning), divers in the water, or to avoid thrust losses in nearby thrusters due to interactions caused when directing the slipstream of one thruster into the propeller disc of another thruster.
- Thrusters may be disabled and enabled dynamically in order to guarantee fault tolerance and operational flexibility.

Industrial solutions are described in Jenssen (1980), Jenssen and Realfsen (2006), Sørdalen (1997) and Sørensen and Ædnanes (1997). A static QP-based strategy is described in Jenssen (1980), while the method in Sørdalen (1997) utilizes pseudo-inverses, in combination with the extended thrust concept (Lindfors, 1993). The control vector $u$ consists of the horizontal plane thrust vector decomposed in the vessel xy-axes (horizontal plane) in order to allow linear models also with azimuth thrusters. In Sørødal (1997), constraints are handled by saturation strategies in combination with filtering of azimuth angle commands that also serves the secondary objective of reducing thruster tear and wear, and the singular value decomposition is used to handle cases when temporary controllability is weak due to all thruster being aligned and can produce thrust all in the same direction. The interactions between the thrust allocation and low level thruster control strategies are studied in Sørøsen and Ædnanes (1997), which is particularly important in extreme seas where thrust losses can be large when the propeller ventilates and in-and-out-of-water effects may lead to propeller spin if properly addressed, (Ruth, Smogeli, Perez, & Sørøsen, 2009; Smogeli & Sørøsen, 2009). Even at fairly low speeds, tunnel thrusters will significantly reduce their effect, which should be accounted for in scheduling of the control effectiveness matrix $B$, (Godhavn, Fossen, & Berge, 1998).

A practical strategy that explicitly optimize the thrust allocation in order to account for power generation constraints, variations in loads, and operational desires such as balancing the load on different electric bus segments and switchboards is described in Jenssen and Realfsen (2006). An integrated approach to dynamic control of power plant as part of the thrust allocation strategy is studied in Veksler, Johansen, and Skjetne (2012a,b), inspired by Radan, Sørensen, Ædnanes, and Johansen (2008) who studied the stabilizing effect on the electrical power plant. An industrial thruster allocation approach implementation with dynamic load control and prediction is described in Mathiesen, Realfsen, and Breivik (2012).

The allocation of control to rudders is particularly challenging due to their highly asymmetric characteristic (no effect when the propellers thrust backwards). Optimization-based approaches that consider the finite (usually small) number of combinations of propeller thrust directions have been proposed and successfully tested, (Johansen, Fuglseth, Tøndel, & Fossen, 2003; Johansen et al., 2008; Lindegaard & Fossen, 2003). Similar strategies can be used for general non-convex thruster constraints, (Ruth & Sørøsen, 2009a), e.g. due to forbidden sectors being less than 180 degrees. In special situations, e.g. when there are primarily azimuth thrusters in use, an additional objective of thruster configuration singularity avoidance might be useful in order to avoid temporary loss of controllability when all thrusters point in more or less the same direction (Johansen et al., 2004; Scibilia & Skjetne, 2012; Sørødal, 1997).

Station keeping of ships and semi-submersible platform for long periods of time are sometimes implemented using mooring lines with thruster assisted position and heading control. This is commonly used for drilling units and floating production, storage and offloading units (FPSOs) operating in water depths of less than 500 m. The thrust allocation must take into account the mooring line forces and provide assistance when needed to make corrections, for example in strong winds or after a mooring line break, e.g. Nguyen and Sørøsen (2009).

Control allocation for small-waterplane marine constructions such as semi-submersibles can obtain additional roll and pitch damping using a conventional thruster system. This is possible for constructions with large draft and beam relative the length since controllability depends on moment arms in roll and pitch (Sørøsen & Strand, 2000). In this case the thrust allocation scheme should not only allocate forces and moment in the horizontal plane (surge, sway and yaw), but also allocate moments in roll and pitch.

4.2.2. High-speed maneuvering and ship autopilots

Ship autopilots conventionally use rudders to meet heading control objectives, while they may also use additional control surfaces such as fins and azimuth propellers (azipods) which calls for control allocation solutions. It is also possible to use rudders for roll damping alone or in combinations with controllable fins (see Perez, 2005 and references therein). The penalties for the use of rudders and fins must be included in the control objective together with penalties and criteria for accurate steering and roll damping.
This is an over-actuated control allocation problem except for ships equipped with one single rudder for simultaneously heading and roll damping, that is a under-actuated rudder-roll damping system that depend on frequency separation of these functions with rudder-roll damping at high frequencies (see Fossen, 2011 and references therein).

Severe instances of parametric rolling of ships can be avoided by specifying the control objectives of the speed and heading autopilots such that the frequency of excitation is changed via the Doppler-shift of the encounter frequency (Holden, Breu, & Fossen, 2012). The optimal frequency is found by using MPC or extremum seeking control, and nonlinear control allocation is used to compute the desired speed and heading angle based on a penalty function designed such that the encounter frequency never is equal to two times the natural frequency in roll, (Breu, Feng, & Fossen, 2012). This is the condition for parametric resonance.

4.3.2. Multi vessel operations

Control allocation strategies have also been proposed for multi-vessel operations, where several tug-boats cooperatively generate forces and moments in order to tow a floating structure. This is formulated in a straightforward manner in the control allocation framework by incorporating the constraints on the tug-boats capacity and direction, (Esposito, Feemster, & Smith, 2008; Feemster & Esposito, 2011). Its implementation requires a supervisory strategy that coordinates the tug-boats that operate as effectors/actuators in this framework.

4.2.4. Maneuvering of underwater vehicles

Highly maneuverable underwater vehicles, either ROVs (Remotely Operated Vehicles) or AUVs (Autonomous Underwater Vehicles), are often controlled using compact electrically driven thrusters and fins. The thrust allocation problem is similar to a dynamically positioned surface vessel, including cases when also vertical forces are controlled using thrusters in addition to buoyancy control. Commonly used methods include pseudo-inverses, re-distributed pseudo-inverses or simple optimization formulations, (Fossen & Sagatun, 1991; Indiveri & Parlangelo, 2006). Aspects of fault-tolerant control by saturation mechanisms and appropriate weighting of the pseudo-inverse is studied in Sarkar, Podder, and Antonelli (2002).

4.3. Automotive and ground vehicles

This section starts with an overview of yaw stability control allocation, before considering electric vehicles, rollover prevention systems, and mobile robots.

4.3.1. Yaw stability control

Active safety systems like the electronic stability control (ESC) are now common in production cars, and shown to have tremendous life-saving effect when skids may occur due to evasive maneuvers, slippery surface, or too high speed in curves, e.g. Ferguson (2007) and Lie, Tingvall, Krafft, and Kullgren (2005). The ESC detects deviation between the actual lateral motion and the driver’s command, (Ackermann, 1996; Ackermann & Bünte, 1996). There has also been proposed systems that have additional redundancy by combining active steering and active braking, (Guvenc, Acarman, & Guvenc, 2003; Wang & Longoria, 2006; Yu & Moskwa, 1994), which also achieves additional control authority and an opportunity to enhance the region of stability.

Due to the strong nonlinearities and dynamic constraints, the use of nonlinear constrained control allocation techniques will generally be desired for lateral vehicle control. However, in order to avoid the online computational burden of nonlinear programming, several simplified approaches have been proposed. The effector mapping from longitudinal tire slips and slip angles are linearized in Wang, Solis, and Longoria (2007) and an accelerated fixed-point iteration algorithm is studies as a computationally efficient alternative to quadratic programming, (Plumlee, Bevly, & Hodel, 2004). A commonly used control allocation objective is to minimize friction forces, for example the adhesion potential characterized using friction ellipse models for each individual tire, (Knobel, Pruckner, & Bünte, 2006). Using linearization of the model $\tau = G \varphi(u, x, \theta)$, where $\theta$ are time-varying parameters, the constrained least-squares problem of allocation error minimization can be solved using numerical online quadratic programming (Andreasson & Bünte, 2006; Tagesson, Sundström, Laine, & Dela, 2009). The effect of vehicle handling performance of weighting matrix coefficients on the control allocation performance is studied in Mokhiamar and Abe (2004, 2006), when using a control allocation method that assumes unconstrained optimization. Nonlinearities and uncertainty is with these approaches handled within low-level actuator/effector controllers that can also provide time-varying constraint limits (such as estimated maximum tire/road friction coefficient) to the control allocation.

A nonlinear programming approach to nonlinear constrained control allocation for yaw stabilization is taken in Tøndel and Johansen (2005), where computational efficiency is achieved through an approximative multi-parametric nonlinear programming algorithm that pre-computes a piecewise linear function that can be evaluated online using binary search tree data structures in order to approximate the optimal solution. The nonlinear optimizing control allocation method of Hattori, Koibuchi, and Yokoyama
4.3.4. Mobile robots

Traction control for mobile robots that operate off-road is considered in Waldron and Abdallah (2007) and Waldron and Hubert (1999). Based on the geometry of the problem, several simple (unconstrained) closed-form computationally efficient control allocation strategies for wheeled and legged mobile robots with active suspension are derived and compared in Waldron and Abdallah (2007), while pseudo-inverse type control allocation strategies are studied in An and Kwon (2010) and Waldron and Hubert (1999). A linear programming solution to control allocation for wheeled mobile robots is presented in Feng, Xu, Li, and Sun (2010). Nonlinearities, uncertainty and additional complexity in these approaches is to a large extent handled in the low-level controllers at each actuator/effecter.

4.4. Other application areas

Legged walking robots require coordination of the dynamic or periodic motion of each leg. The force distributing control allocation algorithm should take into account energy-efficiency and contact friction between the leg and ground which is non-zero only for a fraction of a cycle, e.g. Klein and Chung (1987) and Sreenivasan, Waldron, and Mukherjee (1996). A nonlinear programming approach, simplified by a pseudo-inverse calculations of initial solution guess, is presented in Jung and Baek (2000). The control effectiveness matrix is time-varying due to the cyclic contact pattern of the walk.

Control allocation has been used in the development flapping wing micro air vehicle control methods. In Oppenheimer, Doman, and Sighthorsson (2011) they describe a method to control 5 degrees of freedom using two physical actuators that drive flapping wings. The six variables parameterize the periodic motion of two independently flapping wings that in turn control five degrees of freedom.

Multi-agent swarms (like formations of mobile robots) are considered in a fairly general context in Pedrami, Wijenddra, Baxter, and Gordon (2009). Control allocation strategies based on pseudo-inverses and nonlinear programming are investigated.

The control allocation problem when using a large-scale distributed array of air-jet actuators in studied in Fromherz and Jackson (2003) and Jackson, Fromherz, Biegelsen, Reich, and Goldberg (2001). In order to achieve computational efficiency to allow real-time implementation at high update frequencies in case of thousands of independent actuators, they empirically compare optimal solutions with approximate solution that depend on hierarchical decomposition into actuator groups. A similar approach was taken in Singla and Junkins (2007), where a hierarchical decomposition and re-parameterization using basis-function leads to computational complexity reduction of the control allocation computations. These methods also allow parallelization so the algorithm can be distributed on multiple processors.

Over-actuated mechanical design are increasing in popularity in automotive, aerospace and maritime industries, and not only humanoid walking robots, but emerging concepts like robotic snakes (e.g. Liljebäck, Ståv Dahl, Pettersen, & Gravdahl, 2011), and robotic fish (e.g. Liang, Wang, & Wen, 2011) with highly redundant and over-actuated bio-inspired locomotion mechanisms will for sure benefit from further research on control allocation.

5. Conclusions

Control allocation is fairly well understood and established for linear models. Numerical optimization seems to offer many advantages, while their main challenge is the numerical implementation, in particular computational complexity, verification and validation.

For nonlinear models, the numerical optimization approach is also highly promising, and less alternatives exist. It is, however,
much more challenging than with linear models due to the possibility of local minima in addition to the even stronger challenges of computational complexity, numerical sensitivity, verification and validation. Alternative formulations, including asymptotically optimal methods based on Lyapunov design have been proposed, and the area of nonlinear control allocation is currently an active area of research.

The main driving force of nonlinear control allocation research seems to be applications in the automotive and aerospace industries, and other emerging areas where mechatronics is used.

Acknowledgments

The authors are grateful to Asgeir J. Sørensen, Andrea Cristofaro, and the anonymous reviewers, for comments and references that have contributed to improve the paper.

References


Thor I. Fossen received the M.Sc. degree in Naval Architecture and the Ph.D. in Engineering Cybernetics in 1987 and 1991 both from the Norwegian University of Science and Technology (NTNU). In the period 1989–1990 he pursued postgraduate studies as a Fulbright Scholar in aerodynamics and flight control at the Dept. of Aero- and Astronautics, University of Washington, Seattle. In 1993 Fossen was appointed as a Professor of Guidance, Navigation and Control at NTNU and he was elected into the Norwegian Academy of Technological Sciences in 1999. Fossen is teaching, mathematical modeling of aircraft, marine craft, unmanned vehicles and control theory. Fossen has authored approximately 300 scientific papers and 5 textbooks including the Wiley textbooks “Guidance and Control of Ocean Vehicles” and “Handbook of Marine Craft Hydrodynamics and Motion Control”. Fossen is one of the co-founders of Marine Cybernetics where he was Vice President R&D in the period 2002–2008. A patent for weather optimal positioning control of marine vessels was granted in 1998. This work received the Automatica Prize Paper Award in 2002. In 2008 he received the Arch T. Colwell Merit Award at the SAE 2008 World Congress for a paper entitled “Nonlinear Observer for Vehicle Velocity Estimation”.