

Control Theory and Technology

Guaranteed Feasible Control Allocation using Model Predictive Control

Mehdi Naderi¹, Ali Khaki Sedigh^{1†}, Tor Arne Johansen²

1. *Center of Excellence in Industrial Control, Faculty of Electrical Engineering, K.N. Toosi University of Technology, Seyyed Khandan Bridge, Shariati Ave., Tehran, Iran.*

2. *Center for Autonomous Marine Operation and Systems, Department of Engineering Cybernetics, Norwegian University of Science and Technology, N7491, Trondheim, Norway.*

Received 2 May 2017; revised 25 October 2018; accepted 21 February 2019

Abstract:

This paper proposes a guaranteed feasible control allocation method based on the model predictive control. Feasible region is considered to guarantee the determination of the desired virtual control signal using the pseudo inverse methodology and is described as a set of constraints of an MPC problem. With linear models and the given constraints, feasible region defines a convex polyhedral in the virtual control space. In order to reduce the computational time, the polyhedral can be approximated by a few axis aligned hypercubes. Employing the MPC with rectangular constraints substantially reduces the computational complexity. In two dimensions, the feasible region can be approximated by a few rectangles of the maximum area using numerical geometry techniques which are considered as the constraints of the MPC problem. Also, an active MPC is defined and selected as the controller to minimize the cost function in the control horizon. Finally, several simulation examples are employed to illustrate the effectiveness of the proposed techniques.

Keywords: Control allocation, feasible region, actuator constraints, model predictive control

DOI https://doi.org/10.1007/s11768-018-****.*

1 Introduction

Many safety critical systems such as aircrafts [1], ships and underwater vehicles [2] are designed with redundant effectors and actuators. This redundancy could be employed in order to meet a secondary objective such as power consumption optimization, manoeuvrability and fault tolerance. Control allocation is an approach to manage actuator redundancy under various conditions. The prime objective in a control allocation technique is to produce a desired total control effect computed by the system controller from a set of redundant actuators such that the actuators' constraints are satisfied [3,4].

In the last two decades, several methods have been proposed for the control allocation problem. Most of the proposed schemes follow a modular control configuration pro-

gram which divides the control systems structure into the following two parts. The main controller that provides the specified desired total control effort (also called the virtual control signal) and the control allocator unit that maps the total control demand onto individual actuator settings. The simplest control allocation scheme is the pseudo inverse approach to distribute the virtual control signal among the available actuators [4]. The main drawback of the basic pseudo inverse approach is the fact that the actuators constraints can be easily violated.

The advanced techniques based on the pseudo inverse concept attempt to resolve this problem. These include the redistributed pseudo inverse method, the daisy chain scheme, the pseudo inverse correction along the null space, and the direct allocation method [4]. Basically, the pseudo inverse solution is calculated first and if this solution satis-

[†]Corresponding author.

E-mail: sedigh@kntu.ac.ir. Tel.: +982184062317.

This work was partly supported by Research Council of Norway through the Centres of Excellence funding scheme, grant number 223254 NTNU-AMOS.

fies the actuators constraints, no further steps are needed. Otherwise, different remedies are proposed to solve the constraints violation problem. In the redistributed pseudo inverse method, the control signal vector is partitioned into the saturated, and unsaturated elements and the unsaturated elements are recomputed by solving a reduced problem using the reduced pseudo inverse to compensate for the saturation effects on the virtual control signal. The redistribution procedure is repeated until either the desired virtual control signal is generated or no improvements are achieved in the process continuation [4, 5]. In the daisy chain scheme, actuators are divided into several groups. Initially, control signals of the first actuators group is calculated using the reduced pseudo inverse. If the first group of actuators could produce the desired virtual control, next groups remain unused. Otherwise, the second group of actuators is employed to generate the remaining virtual control demand. This procedure is repeated until either the virtual control signal is satisfied or all of the actuator groups have been employed [6, 7]. [8] proposes an algorithm to modify the pseudo inverse solution using the null space. This approach is based on the fact that each member of the null space of the control effectiveness matrix could not affect the virtual control signal whereas it could modify the control signal vector. In this method, a vector of the null space is added to the pseudo inverse solution if the solution does not satisfy the constraints. The corrective term is calculated such that elements of the control signal which violate the constraints are forced back to the admissible region. [9] presents a robust adaptive fault tolerant control using a weighted pseudo inverse control allocation in the presence of disturbance, unmodeled dynamics and actuator nonlinearity. The main controller is a combination of an adaptive control, radial basis function neural network and a robust controller. Although the above mentioned methods are fast and rather effective, they do not guarantee a feasible solution for the control allocation problem, even when one exists.

The direct allocation approach is a constrained control allocation approach based on scaling of the unconstrained problem solution. When the pseudo inverse solution violates the actuators' constraints, another control signal with the same direction and a smaller norm is applied to the system such that the control signal is located in the attainable space [10, 11]. This method changes the desired total virtual control which may lead to performance and stability degradations.

Iterative methods are also widely employed to solve the control allocation problem. Linear and quadratic pro-

gramming approaches are proposed as control allocation methodologies. These methods convert the control allocation problem to the standard optimization formats and consider actuators constraints. These methods minimize a selected linear or quadratic cost function to minimize the difference between the produced virtual control and its desired value [12–14]. Linear and quadratic programming techniques have been used to manage actuator constraints in many applications [15–17]. [18] considers interaction between actuators and proposes the control allocation problem as a mixed optimization problem.

In addition, some integrated methods are proposed for control allocation which concurrently analyse the main controller and the control allocator unit. Fault tolerant control scheme using on-line control allocation based on various types of sliding mode control concepts are presented in [19–24]. They develop a rigorous design procedure from a theoretical view and prove closed loop stability in the presence of some bounded uncertainties. [25] presents the standard H-infinity criterion to minimize the control effort and the virtual control error. The integrated methods do not consider actuators constraints explicitly in their design.

The Control allocation has a growing application in a wide range of systems. Recently, it has been employed in unmanned aerial vehicles (UAV) [26], neuroprosthesis [27], and electric vehicles [28].

In this paper, to achieve feasibility, actuator constraints are included in the control design through a feasible region determination procedure that simultaneously benefits from the modular and integrated control allocation strategies. The control allocator unit finds the feasible region in the virtual control space. The idea proposed in this paper is to consider the feasible region as the constraints of a set of MPC problems which produce the desired virtual controls. Generally, the feasible region could be considered as polyhedral constraints. The feasibility consideration guarantees generation of a desired virtual control which maintains systems stability and performance.

It is possible to replace a polyhedral by its approximation which includes a few axis aligned hypercubes. The set of employed model predictive controllers consider the control allocation feasible regions in the approximated hypercubical form as their control signal constraints, and produce the desired virtual control to satisfy the corresponding constraints. This reduces the computational complexity of the MPC problems, but may require the solution of several MPC problems to achieve an accurate approximation. In this paper, the two dimensional case is studied in detail and two algorithms are employed to find the largest rectan-

gle in the feasible region.

The paper is structured as follows. The problem statement is given in section 2. Section 3 presents the proposed methods to design the guaranteed feasible control allocation using the MPC approach. In section 4, simulation results are illustrated to show the performance of the proposed scheme. Concluding remarks are given in section 5. Also, two algorithms are outlined for finding the maximum area rectangle in a polygon in the appendix.

2 Problem statement

Consider a linear system described by the following discrete time state-space equations:

$$\begin{aligned} \mathbf{x}(t+1) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}_u\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) \end{aligned} \quad (1)$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$, and $\mathbf{y} \in \mathbb{R}^k$ are the state variables, inputs and outputs of the system, respectively. Also, $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B}_u \in \mathbb{R}^{n \times m}$, and $\mathbf{C} \in \mathbb{R}^{k \times n}$ are the state, input and output matrices, respectively. It is assumed that the system has input redundancy and therefore the input matrix is rank deficient:

$$\text{rank}(\mathbf{B}_u) = d < m \quad (2)$$

The virtual control signal is the total effect of the inputs and it is introduced as follows:

$$\mathbf{v}(t) = \mathbf{B}\mathbf{u}(t) \quad (3)$$

where $\mathbf{B} \in \mathbb{R}^{d \times m}$ is called the control effectiveness matrix which is derived from the system structure and is related to the effectors type, size and location. Combining (1) and (3) yields the following state space representation:

$$\begin{aligned} \mathbf{x}(t+1) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}_v\mathbf{v}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) \end{aligned} \quad (4)$$

and \mathbf{B}_v satisfies

$$\mathbf{B}_u = \mathbf{B}_v\mathbf{B} \quad (5)$$

The admissible space can be defined as follows:

$$\mathbf{u}(t) \in \Omega \equiv \{\mathbf{u}_i \mid \underline{\mathbf{u}}_i \leq \mathbf{u}_i(t) \leq \bar{\mathbf{u}}_i; \forall 1 \leq i \leq m\} \quad (6)$$

where the constraints bounds $\underline{\mathbf{u}}_i$ and $\bar{\mathbf{u}}_i$ depend on the respective actuator health and status.

Control system is divided into two parts, the main controller and the control allocator unit. The main controller produces the desired virtual control signal in order to meet the primary closed loop specifications such as stability, set point tracking and disturbance rejection. The control allocator unit maps the desired virtual control to the actuators commands. It is assumed that the actuators are static and if their commands are admissible with respect to their constraints, $\mathbf{u}(t)$ will equal to its command.

3 The guaranteed feasible control allocation methodology

This paper proposes a control allocation scheme which employs the model predictive controllers to guarantee that the desired virtual controls are generated such that the actuators command are admissible.

3.1 Determining the feasible region

The first step of the method is to determine the feasible region in the virtual control space.

Definition 1. Feasible region is a subset of the virtual control space. If the desired virtual control signal is located there, it is guaranteed that the control allocation unit will map it into the control signal space such that actuator constraints are not violated.

Let \mathbf{B}^\dagger denote the pseudo inverse of the matrix \mathbf{B} :

$$\mathbf{F} = \mathbf{B}^\dagger = \mathbf{W}_I^{-1}\mathbf{B}^T(\mathbf{B}\mathbf{W}_I^{-1}\mathbf{B}^T)^{-1} \quad (7)$$

where \mathbf{W}_I is a weighting matrix which could be the identity matrix in particular. The solution of the control allocation problem using the pseudo inverse is:

$$\mathbf{u}(t) = \mathbf{F}\mathbf{v}_d(t) = \mathbf{B}^\dagger\mathbf{v}_d(t) \quad (8)$$

where \mathbf{v}_d is the desired virtual control. To determine the feasible region, the following inequalities, which define a convex polyhedron should be satisfied:

$$\underline{\mathbf{u}}_i \leq \mathbf{f}_i\mathbf{v}_d(t) \leq \bar{\mathbf{u}}_i; \forall i = 1, 2, \dots, m \quad (9)$$

where \mathbf{f}_i is the i th row of \mathbf{F} . There are numerous softwares to show the polyhedral in both hyper plane and vertex representations such as CDD and the multi parametric toolbox (MPT) [29]. For the two dimensional virtual control space, the admissible solution for each inequality is a space between two parallel lines denoted by s_{i1} and s_{i2} which is

defined as follows:

$$\begin{aligned} s_{i1} : \mathbf{f}_i \mathbf{v}_d &= c_{i1} \\ s_{i2} : \mathbf{f}_i \mathbf{v}_d &= c_{i2} \end{aligned} \quad (10)$$

where $c_{i1} = \underline{\mathbf{u}}_i$ and $c_{i2} = \bar{\mathbf{u}}_i$. The feasible region of the control allocation problem is a convex polygon whose vertices are among the intersection points of the lines which satisfy all the inequalities in (9). Intersection points could be determined as follows:

$$\mathbf{P}_{ik,jl} = \begin{bmatrix} \mathbf{f}_i \\ \mathbf{f}_j \end{bmatrix}^{-1} \begin{bmatrix} c_{ik} \\ c_{jl} \end{bmatrix} \quad (11)$$

where $i, j = 1, 2, \dots, m, i \neq j$ and $k, l = 1, 2$.

In what follows, the algorithm for determining the feasible polygon region for a two dimensional space is given.

Algorithm 1. The procedure of determining the feasible region could be expressed as follows:

- Consider the $2m$ lines given by equations s_{i1} and s_{i2} as (10).
- Determine the intersection points as (11).
- Find the intersection points satisfying (9) for all $1 \leq i \leq m$.
- The polygon whose vertices are the points found in the third step is the feasible region.

Fig.1 shows the feasible region of a given \mathbf{B} matrix and the actuator constraints.

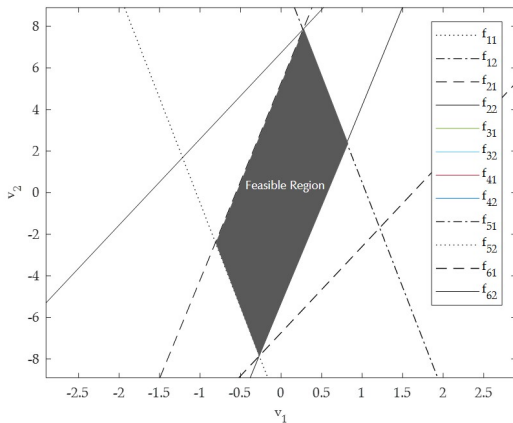


Fig. 1 The feasible region

3.2 Feasible control allocation using the MPC controller with the polyhedral constraints

If the desired virtual control vector is in the feasible region, the pseudo inverse solution directly gives the actuators commands. Therefore, the main controller should pro-

duce the desired value of the virtual control signal located in the feasible area. The first strategy is to employ an MPC controller which could consider inequalities shown in (9) as its input constraints. The MPC controller minimizes the following cost function [30]:

$$\begin{aligned} J(\mathbf{V}_d) &= \\ &= \sum_{j=1}^N ([\hat{\mathbf{y}}(t+j|t) - \mathbf{w}(t+j)]^T \mathbf{Q} [\hat{\mathbf{y}}(t+j|t) - \mathbf{w}(t+j)] \\ &+ [\mathbf{v}_d(t+j-1|t) - \mathbf{v}_d^*(t+j-1|t)]^T \mathbf{R} [\mathbf{v}_d(t+j-1|t) - \mathbf{v}_d^*(t+j-1|t)]) \end{aligned} \quad (12)$$

where $\hat{\mathbf{y}}(t+j|t)$ is an optimum j -step ahead prediction of the system output based on the data up to time t , N is the control horizon, \mathbf{Q} and \mathbf{R} are the weighting matrices, $\mathbf{w}(t+j)$ is the future reference trajectory, and \mathbf{v}_d^* is the ideal input value. To put the MPC problem in a suitable optimization form, stacked vectors with future states and control inputs are defined as follows [30]:

$$\begin{aligned} \mathbf{Y} &= \begin{bmatrix} \hat{\mathbf{y}}(t+1|t) \\ \hat{\mathbf{y}}(t+2|t) \\ \vdots \\ \hat{\mathbf{y}}(t+N|t) \end{bmatrix}, \quad \mathbf{V}_d = \begin{bmatrix} \mathbf{v}_d(t|t) \\ \mathbf{v}_d(t+1|t) \\ \vdots \\ \mathbf{v}_d(t+N-1|t) \end{bmatrix}, \\ \mathbf{V}_d^* &= \begin{bmatrix} \mathbf{v}_d^*(t) \\ \mathbf{v}_d^*(t+1) \\ \vdots \\ \mathbf{v}_d^*(t+N-1) \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} \mathbf{w}(t+1) \\ \mathbf{w}(t+2) \\ \vdots \\ \mathbf{w}(t+N) \end{bmatrix} \end{aligned} \quad (13)$$

The predicted states can be rewritten as $\mathbf{Y} = \mathbf{H}\mathbf{x}(t|t) + \mathbf{S}\mathbf{V}_d$ where:

$$\begin{aligned} \mathbf{H} &= \begin{bmatrix} \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^N \end{bmatrix}, \\ \mathbf{S} &= \begin{bmatrix} \mathbf{CB} & 0 & \dots & 0 \\ \mathbf{CAB} & \mathbf{CB} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{CA}^{N-1}\mathbf{B} & \mathbf{CA}^{N-2}\mathbf{B} & \dots & \mathbf{CB} \end{bmatrix} \end{aligned} \quad (14)$$

Hence, the cost function can be written as follows:

$$\begin{aligned} J(\mathbf{V}_d) &= (\mathbf{H}\mathbf{x}(t|t) - \mathbf{W})^T \bar{\mathbf{Q}} (\mathbf{H}\mathbf{x}(t|t) - \mathbf{W}) \\ &+ \mathbf{V}_d^T (\mathbf{S}^T \bar{\mathbf{Q}} \mathbf{S} + \bar{\mathbf{R}}) \mathbf{V}_d + \mathbf{V}_d^*{}^T \bar{\mathbf{R}} \mathbf{V}_d^* \end{aligned}$$

$$+ 2 (\mathbf{H}\mathbf{x}(t|t) - \mathbf{W}) \mathbf{Q}\bar{\mathbf{S}}\mathbf{V}_d \quad (15)$$

where $\bar{\mathbf{Q}}$ and $\bar{\mathbf{R}}$ are the extended weighting matrices. Solving the following constrained convex quadratic optimization problem in the virtual control space yields the desired virtual control signal:

$$\begin{aligned} \min_{\mathbf{V}_d} J(\mathbf{V}_d) \\ \text{subject to } \underline{\mathbf{v}}_i \leq \mathbf{f}_i \mathbf{v}_d(t+j-1|t) \leq \bar{\mathbf{u}}_i; \quad (16) \\ \forall i = 1, 2, \dots, m, \forall j = 1, 2, \dots, N-1 \end{aligned}$$

3.3 Feasible control allocation using the MPC controller with rectangular constraints

Considering the constraints as (16) may lead to a large computational burden. Replacing the feasible region polyhedral by an approximated axis aligned hypercube can reduce the number of constraints and make them simpler and reduces the computational time. In the two dimensional virtual control space, the feasible region could be approximated by an axis aligned rectangle. It is possible to employ the algorithms of finding the maximum area rectangle in a polygon described in the appendix. The derived rectangles based on the algorithms mentioned in the appendix are shown in Fig.2. Note that the area of the second and fourth derived rectangles equal to zero.

One conservative strategy is to consider the largest rectangle among the four rectangles which are found based on the algorithms mentioned in the appendix as the control signal constraints of an MPC controller. The MPC controller solves the following optimization problem where the cost function is defined as (15):

$$\begin{aligned} \min_{\mathbf{V}_d} J(\mathbf{V}_d) \\ \text{subject to } \underline{\mathbf{v}}_i \leq \mathbf{v}_d(t+j-1|t) \leq \bar{\mathbf{v}}_i; \quad (17) \\ \forall i = 1, 2, \forall j = 1, 2, \dots, N-1 \end{aligned}$$

where $\underline{\mathbf{v}}_i$ and $\bar{\mathbf{v}}_i, i = 1, 2$ are the lower and upper bands of the x and y coordinates given by the largest rectangle. The advantage of this strategy is that it is computationally simpler than (16), but it is conservative and leads to reduced performance, since the rectangle is an inner approximation of the polygon.

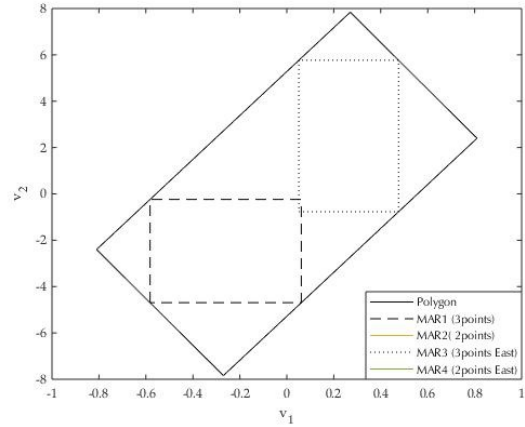


Fig. 2 Maximum area rectangles in the feasible region

Algorithm 2. The following steps are proposed to implement the feasible control allocation based on the MPC controller:

- Determine the feasible region of the control allocation problem.
- Find the largest axis aligned rectangle inscribed in the feasible polygon region.
- Design the MPC controller constrained by the control signal within a given rectangle.
- In each sample time:
 - Calculate the desired virtual control signal using the MPC strategy.
 - Map the desired virtual control to the actuator commands using the pseudo inverse (7)- (8).

It should be noted that the first three steps are executed only for initialization.

3.4 Feasible control allocation using the multiple MPC controllers

In Fig.2, it can be clearly observed that by selecting any rectangle, a significant part of the feasible polygon region will be neglected. This problem could decrease the controller performance due to the virtual control signal being too restricted. In order to enlarge the covered area of the feasible region approximation, a multiple MPC strategy is proposed, where each calculated rectangle is considered as the constraints of a separate MPC controller. Therefore, a larger area within the polygon region will finally be covered. Plant dynamics and cost function parameters remain unchanged while the control signal constraints are derived from the determined rectangles for $l = 1, 2, \dots, N_r$. Each controller solves an optimization problem as follows [30]:

$$\min_{\mathbf{V}_{dl}} J(\mathbf{V}_{dl})$$

$$\text{subject to } : \underline{v}_{il} \leq v_{dl}(t+j-1|t) \leq \bar{v}_{il}; \quad (18)$$

$$\forall i = 1, 2, \forall j = 1, 2, \dots, N-1$$

where \underline{v}_{il} and \bar{v}_{il} are specified by coordinates of the l th rectangle and N_r is the number of rectangles with non-zero areas. At each sample time, all controllers are evaluated and their corresponding cost functions are compared with the other cost functions. Then the MPC controller corresponding to the minimum cost function is selected as the active controller and its corresponding virtual control signals will be sent to the control allocator unit. The active controller index is denoted by $\sigma(t)$. The dwell time concept in switching between the controllers is considered for closed loop considerations, where the selected controller should be active for at least τ_d units of time and where τ_d is called the dwell time [31].

Algorithm 3. The following steps are performed to implement the feasible control allocation methodology based on the multiple MPC controller strategy:

- Determine the feasible region of the control allocation problem.
- Find N_r axis aligned rectangles in the feasible polygon region.
- Design N_r MPC controllers by the control signal constraints defined by the rectangles.
- In each sample time:
 - Calculate the desired virtual control signal and the cost function of each MPC controller.
 - Select the best virtual control which has the minimum cost function. The controller is switched only if the new active controller improves the cost function value with the hysteresis parameter h . Also, the selected controller remains active for at least τ_d time units.
 - Map the chosen desired virtual control to the actuator commands using the pseudo inverse (7)-(8).

Structure of the proposed algorithm is shown in Fig.3. The switching control supervisor unit selects the active controller by considering the dwell time and the hysteresis. Also, the feasible control allocation unit determines the feasible region and introduces the corresponding rectangles to the MPC controllers for constraints generation. The computation time can be reduced since the MPC controllers can be evaluated in parallel simple box constraints.

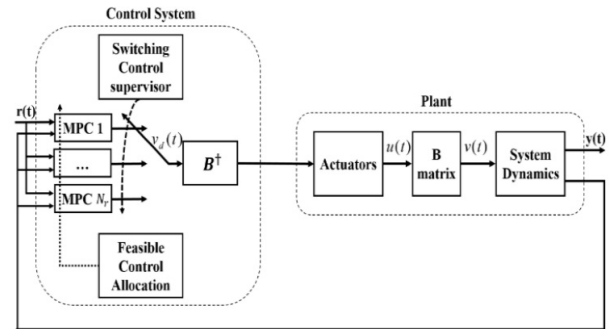


Fig. 3 Structure of the feasible control allocation using the multiple MPC strategy

4 Simulation results

In this section, simulation results are presented to show the effectiveness of the proposed methodologies. The CVX toolbox [32] is employed to solve the optimization problems.

4.1 Example 1

In this example, the following methods are employed for a comparative study:

- (1) **The feasible control allocation (FCA):** An MPC controller is employed as the main controller that considers the feasible region as the polyhedral input constraints.
- (2) **MPC with PAN:** An unconstrained MPC and the pseudo inverse along the null space (PAN) methodology [8] are used as the main controller and the control allocator unit, respectively.
- (3) **MPC with quadratic programming (QP):** An unconstrained MPC and the quadratic programming strategy are used as the main controller and the control allocator unit, respectively.
- (4) **LQ with quadratic programming:** A linear quadratic(LQ) optimal control and the quadratic programming strategy [13] are used as the main controller and the control allocator unit, respectively.
- (5) **LQ with RPI:** A linear quadratic(LQ) optimal control and the redistributed pseudo inverse (RPI) strategy [5] are used as the main controller and the control allocator roles, respectively.

Consider the following linearized state space model [33]:

$$\mathbf{A} = \begin{bmatrix} -0.543 & 0.013 & 0 & 0.978 & 0 \\ 0 & -0.12 & 0.221 & 0 & -0.9661 \\ 0 & -10.52 & -0.997 & 0 & 0.6176 \\ 2.62 & -0.003 & 0 & -0.506 & 0 \\ 0 & 0.708 & -0.0939 & 0 & -0.213 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0 & -4.24 & 4.24 & 1.487 \\ 1.653 & -1.27 & -1.27 & 0.0024 \\ 0 & -0.28 & 0.28 & -0.88 \end{bmatrix};$$

$$\mathbf{B}_v = \begin{bmatrix} 0_{2 \times 3} \\ I_3 \end{bmatrix}; \mathbf{C} = \begin{bmatrix} 0_{3 \times 2} & I_3 \end{bmatrix}$$

$$\underline{\mathbf{u}} = \begin{bmatrix} -5 & -3 & -3 & -3 \end{bmatrix}^T; \bar{\mathbf{u}} = \begin{bmatrix} 2 & 3 & 3 & 3 \end{bmatrix}^T \quad (19)$$

Under the given conditions, the different methods have a similar performance and they appropriately manage the actuators. As is shown in Table 1, employing the LQ controller instead of the MPC decreases the computational burden significantly. In order to quantitatively compare the results, Table 1 presents the computational time and the cost function value for the methods. The cost function is defined as follows:

$$J_s = \frac{\sum_{t=0}^{T_{end}} \left((\mathbf{y}(t) - \mathbf{w}(t))^T \mathbf{Q} (\mathbf{y}(t) - \mathbf{w}(t)) + \mathbf{u}(t)^T \mathbf{R} \mathbf{u}(t) \right)}{T_{end}} \quad (20)$$

Table 1. Example 1- Methods comparison

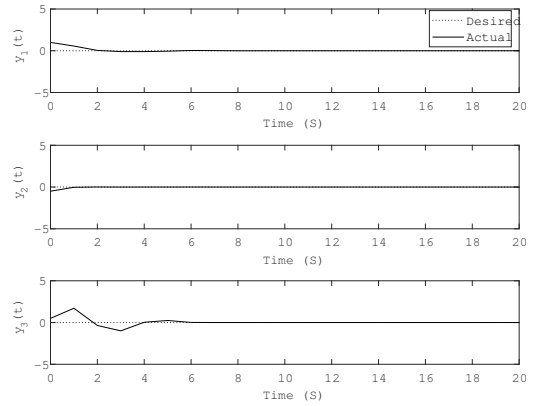
Method	Cost function	Computational time (s)
MPC with FCA	5.71	0.08
MPC with PAN	5.71	0.076
MPC with QP	5.71	0.081
LQ with QP	5.74	0.016
LQ with RPI	5.73	5.27×10^{-4}

In the next simulation, the upper and lower bounds of the 4th actuator are tightened (e.g. due to the actuator fault) as follows:

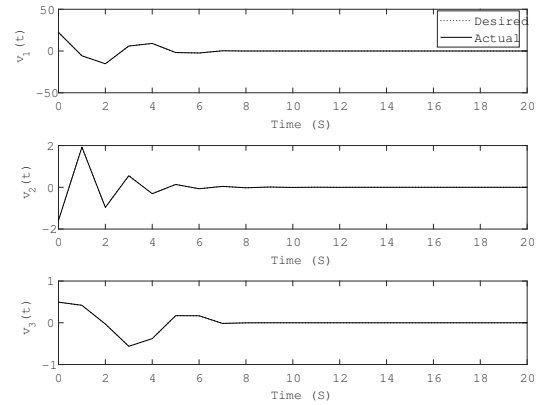
$$\underline{\mathbf{u}} = \begin{bmatrix} -5 & -3 & -3 & -1 \end{bmatrix}^T; \bar{\mathbf{u}} = \begin{bmatrix} 2 & 3 & 3 & 1 \end{bmatrix}^T \quad (21)$$

In this case, closed loop systems which do not consider the feasibility concept became unstable. Instability has occurred due to the discrepancy between the desired and the actual virtual control signals which appeared because of lack of feasibility consideration. As is shown in Fig.4, using the feasible control allocation preserves closed loop

stability and only the transient performance is degraded. There is no discrepancy between the desired and the actual virtual control signals as shown in Fig.4.b Hence, it is observed that using the feasible region information in the main controller could extend the stability region and achieve a better performance.



(a)



(b)

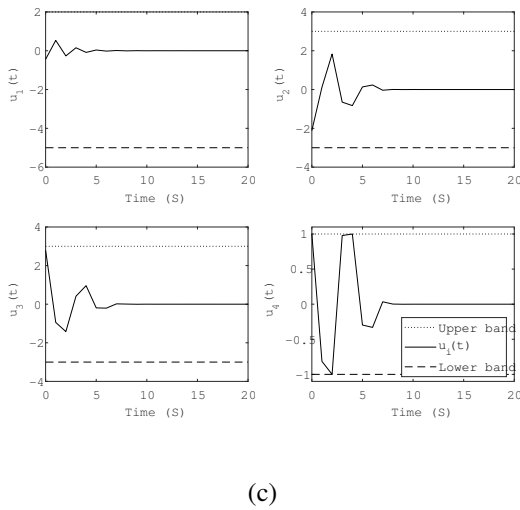


Fig. 4 Example 1- The feasible MPC method, (a) System outputs, (b) Virtual control signals, (c) Control signals

4.2 Example 2

In the second example three methods are employed for comparison study, 1)feasible control allocation using the MPC controller with polyhedral constraints, 2)feasible control allocation using the single MPC controller with rectangular constraints, and 3)feasible control allocation using the multiple MPC controllers with rectangular constraints. Consider the following linear model of an electric vehicle [34]:

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} 0.3348 & -0.0277 \\ 0.0927 & 0.2436 \end{bmatrix} \\ \mathbf{B}_v &= \begin{bmatrix} 0.0609 & -0.0022 \\ 0.0073 & 0.0537 \end{bmatrix} \\ \mathbf{C} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned} \quad (22)$$

Also, the control effectiveness matrix and the actuator upper and lower limits are as follows:

$$\begin{aligned} \mathbf{B} &= \begin{bmatrix} 5.4012 & 5.4012 & 0 & 0 & 0 & 0 \\ 51.2 & -54.4 & -0.0015 & -0.0015 & -0.0015 & -0.0015 \end{bmatrix} \\ \underline{\mathbf{u}} &= [-0.1 \ -0.05 \ -1 \ -1 \ -1 \ -0.1]^T \\ \bar{\mathbf{u}} &= [0.1 \ 0.05 \ 1 \ 1 \ 1 \ 0.1]^T \end{aligned} \quad (23)$$

Fig.5 depicts the results of employing the feasible control allocation using an MPC controller with the polyhedral constraints. As is shown in Fig.5(a), this scheme leads to

the desired transient and steady state responses. Also, there is no discrepancy between the desired virtual and actual control signals as shown in Fig.5(b).

For the feasible control allocation with a single MPC controller, the largest rectangle found in the feasible region of this example shown Fig.2 is considered as the constraints of an MPC controller. It is obvious that a large area of the feasible region is ignored by considering only the largest rectangle and it causes steady state error in the second output as is depicted in Fig. 6(a). A particular problem is that positive values of the desired virtual control signal are strongly restricted as is shown in 2. Note that it could be seen in Fig.6(b) that the generated and the desired virtual signals are identical which is due to applying a part of the feasible region as the input constraints of the MPC problem.

For the multiple MPC controller scheme both rectangles shown in Fig.2 are considered. Results of applying multiple control with the dwell time are shown in Fig.7. As is seen in Fig.7(a), enlarging the feasible region by considering two rectangles improves the control performance. Also, the difference between the virtual and actual control signals equals zero. The active controller is switched when it is necessary, as is shown in Fig.7(d). Table 4.2 shows the average running time of the proposed methods to compare their computational burden. Using the multiple MPC controller scheme with rectangular constraints which employs parallel processing decreases running time of the control procedure in each sample time while it does not cause an unacceptable performance. It should be mentioned that the computational times are presented only for comparison and they can be decreased substantially in real time implementations.

Table 2. Comparison of the running time of the presented methods

FCA Method	Computational time (s)
MPC with polyhedral constraints	0.38
MPC with rectangular constraints	0.21
Multiple MPC	0.23

5 Conclusion

This paper proposes a guaranteed feasible control allocation methodology. The feasible region is determined by characterizing the polyhedral feasible region of the pseudo inverse solution. Then, the feasible region is introduced to the main controller as the constraints of an MPC problem. Another approximated method is presented for decreasing computational time where several axis aligned hypercubes

are considered as an approximation of the feasible region. A set of MPC controllers solved using parallel processing are employed as the main controllers where each of the given hypercubes is considered as constraints in the corresponding MPC controller. Then, switching control is used to choose the most appropriate controller in each step having the lowest cost and satisfying dwell time specifications. In the case of two dimensional virtual control space, an algorithm is employed to find the maximum area rectangles inscribed in the feasible polygon region. Simulation results are used for comparison studies and to show that computational cost can be reduced with the approximate method.

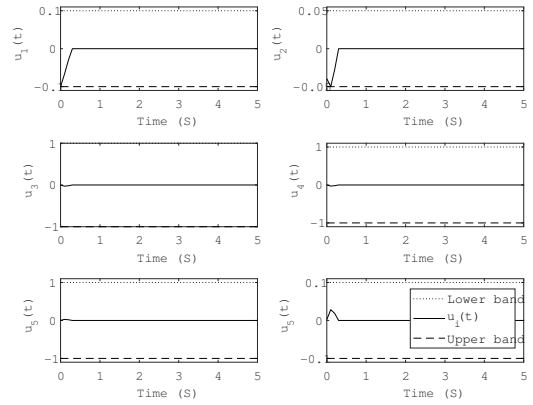
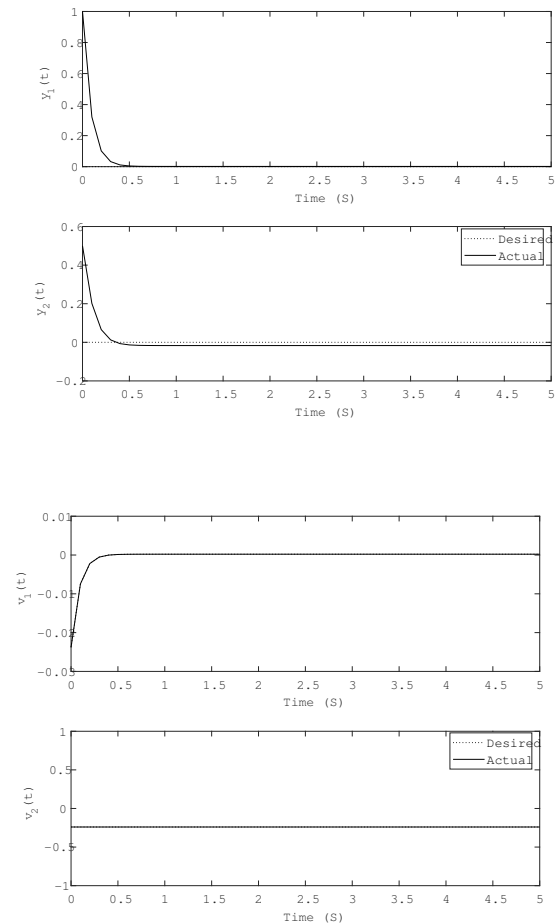
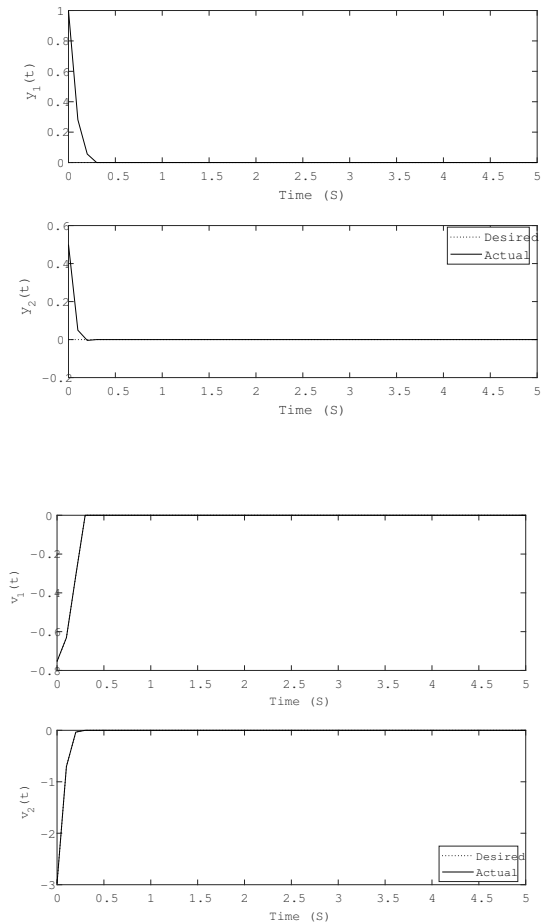


Fig. 5 Example 2- The feasible region method using an MPC controller with polyhedral constraints, (a) System outputs, (b) Virtual control signals, (c) Control signals



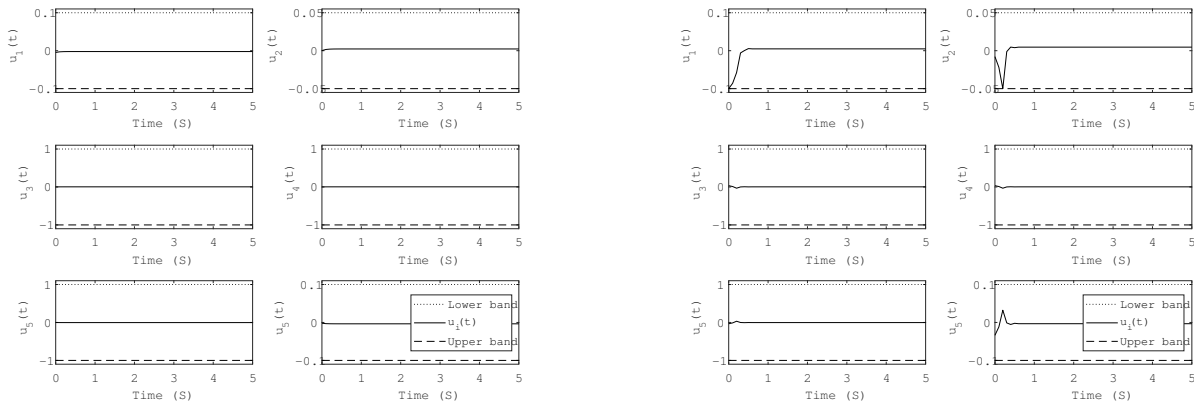


Fig. 6 Example 2- The MPC controller method, (a) System outputs, (b) Virtual control signals, (c) Control signals

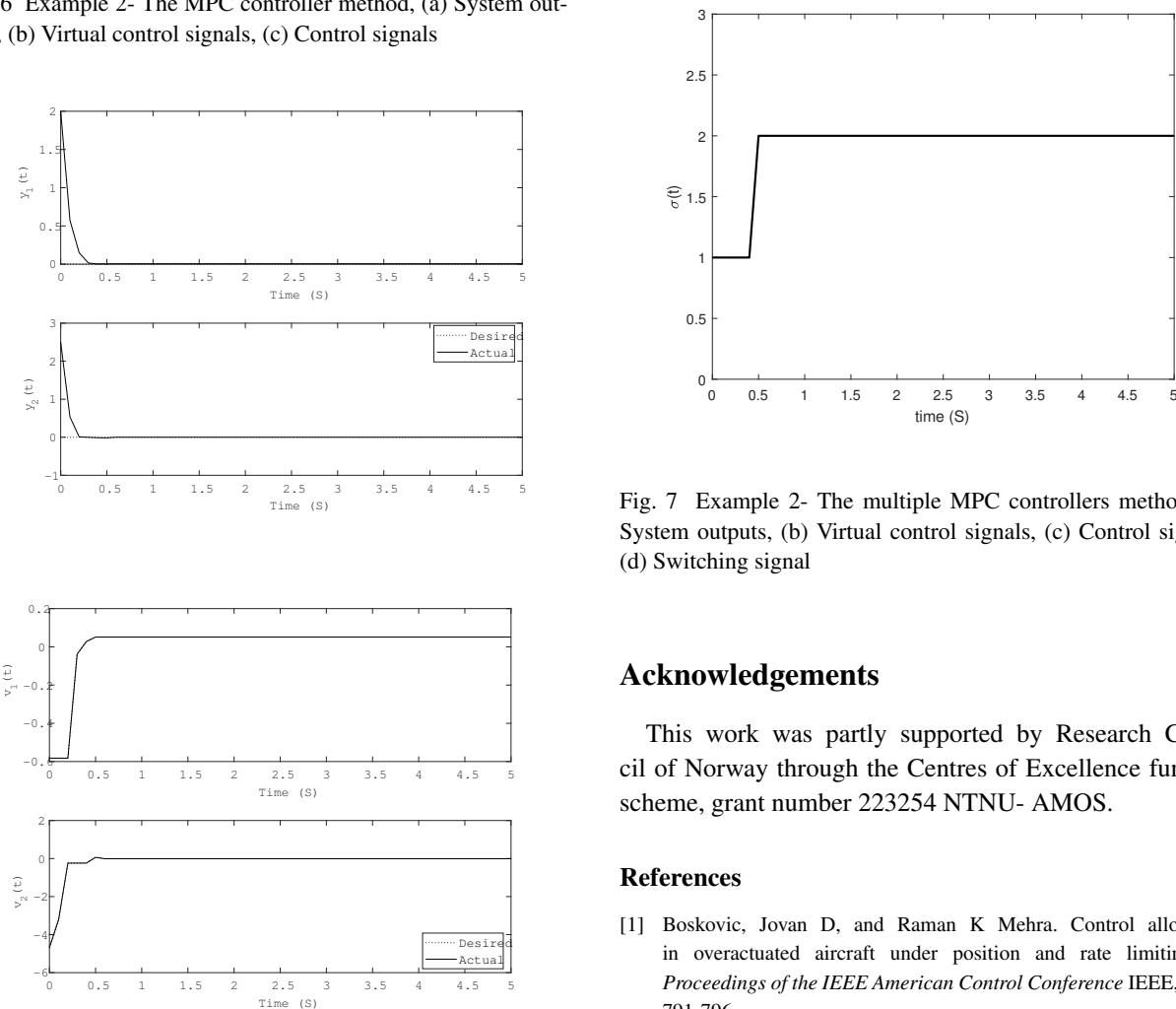


Fig. 7 Example 2- The multiple MPC controllers method, (a) System outputs, (b) Virtual control signals, (c) Control signals, (d) Switching signal

Acknowledgements

This work was partly supported by Research Council of Norway through the Centres of Excellence funding scheme, grant number 223254 NTNU- AMOS.

References

- [1] Boskovic, Jovan D, and Raman K Mehra. Control allocation in overactuated aircraft under position and rate limiting, *In Proceedings of the IEEE American Control Conference* IEEE, 2002, 791-796.
- [2] Fossen, Thor I, and Tor A Johansen. A survey of control allocation methods for ships and underwater vehicles, *In proceeding of the 14th Mediterranean Conference on Control and Automation*, 2006, 1-6.
- [3] Cristofaro, Andrea and Johansen, Tor A. Fault tolerant control allocation using unknown input observers, *Automatica*, 2014, 50

- (7):1891-1897.
- [4] Johansen, Tor A, and Thor I Fossen. Control allocation: a survey, *Automatica*, 2013, 49 (5):1087-1103.
- [5] Virnig, John C, and David S Bodden. Multivariable control allocation and control law conditioning when control effectors limit (STOVL aircraft), *In proceeding of the AIAA Guidance, Navigation and Control Conference*, Scottsdale, 1994, 572-582.
- [6] Buffington, James M, and Dale F Enns. Lyapunov stability analysis of daisy chain control allocation, *Journal of Guidance, Control, and Dynamics*, 1996, 19 (6): 1226-1230.
- [7] Kim, Jiyeon, Inseok Yang, and Dongik Lee. Accommodation of actuator faults using control allocation with modified daisy chaining, *In proceeding of 11th International Conference on Control, Automation and Systems (ICCAS)*, 2011, 717-720.
- [8] Tohidi, SS, A Khaki Sedigh, and D Buzorgnia. Fault tolerant control design using adaptive control allocation based on the pseudo inverse along the null space, *International Journal of Robust and Nonlinear Control*, 2016, 26(16), 3541-3557.
- [9] Zhi, J., Chen, Y., Dong, X., Liu, Z., & Shi, C. Robust adaptive FTC allocation for over-actuated systems with uncertainties and unknown actuator non-linearity, *IET Control Theory & Applications*, 2017, 12(2), 273-281.
- [10] Durham, Wayne C. Constrained control allocation, *Journal of Guidance, Control, and Dynamics*, 1993, 16 (4): 717-725.
- [11] Durham, Wayne C. Constrained control allocation-Three-moment problem, *Journal of Guidance, Control, and Dynamics*, 1994, 17 (2): 330-336.
- [12] Bodson, Marc. Evaluation of optimization methods for control allocation, *Journal of Guidance, Control, and Dynamics*, 2002, 25 (4): 703-711.
- [13] Harkegard, Ola. Efficient active set algorithms for solving constrained least squares problems in aircraft control allocation, *In Proceedings of the 41st IEEE Conference on Decision and Control*, 2002, 1295-1300.
- [14] Harkegard, Ola. Efficient active set algorithms for solving constrained least squares problems in aircraft control allocation, *In Proceedings of the 41st IEEE Conference on Decision and Control*, 2002, 1295-1300.
- [15] Frost, S. A., & Bodson, M. Resource balancing control allocation, *In proceeding of the American Control Conference*, 2010, 1326-1331.
- [16] Johansen, T. A., Fossen, T. I., & Berge, S. P. Constrained nonlinear control allocation with singularity avoidance using sequential quadratic programming, *IEEE Transactions on Control Systems Technology*, 2004, 12 (1): 211-216.
- [17] Plumlee, J. H., Bevely, D. M., & Hodel, A. S. Control of a ground vehicle using quadratic programming based control allocation techniques, *in Proceedings of the American control conference*, 2004, 211-216.
- [18] Rajput, J., Xiaobo, Q., & Khan, H. Z. I. Nonlinear Control Allocation for Interacting Control Effectors using Bivariate Taylor Expansion, *In 5th International Conference on Control, Decision and Information Technologies*, 2018, 352-357.
- [19] Alwi, Halim, and Christopher Edwards. Sliding mode FTC with on-line control allocation, *In Proceedings of the 45th IEEE Conference on Decision and Control*, 2006, 5579-5584.
- [20] Alwi, Halim, and Christopher Edwards. Model-reference sliding mode FTC with on-line control allocation, *In Proceedings of the 46th IEEE Conference on Decision and Control*, 2007, 2639-2644.
- [21] Alwi, Halim, Christopher Edwards, Olaf Stroosma, and JA Mulder. Fault tolerant sliding mode control design with piloted simulator evaluation, *Journal of Guidance, Control, and Dynamics*, 2008, 31 (5): 1186-1201.
- [22] Hamayun, Mirza Tariq, Christopher Edwards, and Halim Alwi. 2012. "Design and analysis of an integral sliding mode fault-tolerant control scheme." *IEEE Transactions on Automatic Control* 57 (7):4704- 4709.
- [23] Hamayun, M. T., Edwards, C., & Alwi, H. *Fault tolerant control schemes using integral sliding modes*, Springer International Publishing, 2016.
- [24] Ros, Hector, Shyam Kamal, Leonid M Fridman, and Ali Zolghadri. Fault tolerant control allocation via continuous integral sliding-modes: a HOSM-observer approach, *Automatica*, 2015, 51: 318-325.
- [25] Ji, S. W., Bui, V. P., Balachandran, B., & Kim, Y. B. Robust control allocation design for marine vessel, *Ocean Engineering*, 2013, 63, 105-111.
- [26] , D. T., Saussi, D., & Saydy, L. Fault-Tolerant Control of a Hexacopter UAV based on Self-Scheduled Control Allocation, *In 2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2018, 385-393.
- [27] , Kirsch, N. A., Bao, X., Alibeji, N. A., Dicianno, B. E., & Sharma, N. Model-based dynamic control allocation in a hybrid neuroprosthesis, *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2018, 26(1), 224-232.
- [28] , Jing, H., Jia, F., & Liu, Z. Multi-Objective Optimal Control Allocation for an Over-Actuated Electric Vehicle. *IEEE Access*, 2018, 6, 4824-4833.
- [29] Hecceg, M., Kvasnica, M., Jones, C. N., & Morari, M. Multi-parametric toolbox 3.0, *In proceeding of European Control Conference*, 2013, 502-510.
- [30] Camacho, Eduardo F, and Carlos Bordons Alba. 2013. Model predictive control. *Springer Science & Business Media*
- [31] Liberzon, D. *Switching in systems and control*, Springer Science & Business Media, 2003.
- [32] Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.0 beta. <http://cvxr.com/cvx>, September 2013.
- [33] Harkegard, Ola. 2004. "Quadratic programming control allocation toolbox (qcat)." Mar. <http://research.harkegard.se/qcat>
- [34] Feng, C., Ding, N. G., He, Y. L., Xu, G. Y., & Gao, F. Control allocation algorithm for over-actuated electric vehicles, *Journal of Central South University*, 2014, 21(10), 3705-3712.
- [35] Fischer, Paul and Höffgen, Klaus-Uwe, Computing a maximum axis-aligned rectangle in a convex polygon, *Information Processing Letters*, 1994, 51(4), 189- 193
- [36] Yan, T., Gu, Y., He, T., & Stankovic, J. A. Design and optimization of distributed sensing coverage in wireless sensor networks, *ACM Transactions on Embedded Computing Systems (TECS)*, 2008, 7(3), 33.

Appendix

A The maximum area rectangle principle

In the two dimensional virtual control space, the feasible region can be generally characterized by a convex polygon. To develop a guaranteed feasible control allocation strategy with lower computational complexity, a maximum area rectangle (MAR) which lies in the convex polygon of the feasible region is identified. To derive this MAR the algorithm developed in [35] is employed. This mathematical algorithm has been used in several engineering applications such as the wireless sensor network [36]. The algorithm takes the polygon vertices and yields an axis-aligned rectangle in the polygon which has the maximum area. To briefly present the MAR algorithm, some basic definitions are given below. Consider $\mathbf{Q}, \mathbf{Q}' \in \mathbb{R}^2$, $\mathbf{Q} = (x, y)$, $\mathbf{Q}' = (u, v)$ with $x < u$ and $y < v$. Then $R = \text{Rect}(\mathbf{Q}; \mathbf{Q}')$ is the rectangle with the lower left corner given by \mathbf{Q} and the upper right corner given by \mathbf{Q}' . Area of the rectangle is denoted by $\text{area}(R) = (u - x)(v - y)$. Let P be a bounded convex polygon with n vertices and denote its boundary by δP . Also, the corners of a rectangle indicated by the lower left, lower right, upper left and upper right are denoted by $\text{LL}(R)$, $\text{LR}(R)$, $\text{UL}(R)$ and $\text{UR}(R)$, respectively. Also, the southeast, southwest, northeast and northwest parts of the polygon are denoted by SE, SW, NE, and NW, respectively. Note that the four parts are not necessarily disjoint. The maximum area rectangle could have two or three corners on δP , and two separate algorithms are presented in [35] to find the above two possible rectangles.

A.1 MAR with at least three vertices on the boundary

Let x_l be the minimum x -coordinate of a vertex of P . Consider $x_{\text{NW}}(x_{\text{SW}})$ as the maximum x -coordinate of a vertex on NW (SW), and $x_r = \min(x_{\text{NW}}, x_{\text{SW}})$. Let $S(x)$ be the vertical line at x , $x_l \leq x \leq x_r$, and let y_1 and y_2 be the y -coordinates of the intersections of $S(x)$ and P . Consider u_i as the x -coordinate of the horizontal projection of y_i onto $\text{NE} \cup \text{SE}$, $i = 1, 2$. Define $x' = \min(u_1, u_2)$. Also, consider the rectangle $R(x) = \text{Rect}((x, y_1), (x', y_2))$.

Definition A.1 [35]. A function $f(x)$ is called almost strictly bitonic on $[x_i, x_j]$, if there exist $x_k, x_m \in [x_i, x_j]$ such that $f(x)$ is strictly increasing on $[x_i, x_k]$, strictly decreasing on $[x_m, x_j]$ and constant on $[x_k, x_m]$.

Theorem A.1 [35]. $\text{area}(R(x))$ is continuous and almost

strictly bitonic on $[x_l, x_r]$.

Algorithm A.1 [35]. The procedure of finding the maximum area rectangle with at least 3 corners on the boundary of the polygon is summarized as follows:

- Find that vertex (x_1, y_1) of SW which halves SW with respect to the number of vertices and its right neighbour (x_2, y_2) .
- Compute the corresponding rectangle $R(x_1)$, $R(x_2)$ and determine the following alternatives:
 - If $\text{area}(R(x_1)) > \text{area}(R(x_2))$, then search should be followed in $[x_l, x_2]$.
 - if $\text{area}(R(x_1)) < \text{area}(R(x_2))$, then search should be followed in $[x_l, x_r]$.
 - if $\text{area}(R(x_1)) = \text{area}(R(x_2))$, then iteration should be stopped.
- If only two adjacent vertices of SW are left, a binary search should be done.

A.2 MAR with two vertices on the boundary

Consider the maximum area rectangle M that has exactly two diagonally opposed corners on δP . Only the cases that $\text{LL}(M) \in \text{SW}$ and $\text{UR}(M) \in \text{NE}$ will be discussed here. Let x_l and x_r be defined as above, $x \in [x_l, x_r]$ and (x, y) be the projection of x onto SE. Then, $\text{MR}(x)$ is the maximum area rectangle (not necessary included in P) with lower left corner (x, y) and upper right corner on NE.

Theorem A.2 [35]. $\text{area}(\text{MR}(x))$ is continuous and almost strictly bitonic on $[x_l, x_r]$.

Algorithm A.2 [35]. The procedure of finding the maximum area rectangle with exactly two corners on the boundary of the polygon is summarized as follows:

- Find x as the x -coordinate of that vertex which halves $[x_l, x_r]$ with respect to the number of vertices on SW.
- Compute $\text{MR}(x)$ by a binary search on NE.
- If both $\text{UL}(\text{MR}(x))$ and $\text{LR}(\text{MR}(x))$ are outside of P , stop the algorithm. Because it is proved that no $\text{MR}(x')$ could be found such that it is totally contained in P , for any $x' \in [x_l, x_r]$.

As indicated in the algorithms A.1 and A.2, two rectangles could be obtained which have a corner on SW. According to the algorithms, it is possible to determine another two rectangles that their lower right corners are located on SE. It should be noted that some of these rectangles could have zero area.

Mehdi NADERI received his M.Sc. degree in electrical engineering from University of Tehran, Tehran, Iran in 2013. From 2013, he is pursuing a Ph.D. degree in the department of electrical engineering at K.N.

Toosi University of Technology on fault tolerant control systems using control allocation. His research interests are control allocation, fault tolerant systems, and adaptive control. E-mail: mnaderi@ee.kntu.ac.ir.

Ali KHAKI SEDIGH received the honors degree in mathematics from the University of Newcastle upon Tyne, Newcastle upon Tyne, U.K., in 1983, the Masters degree in control systems from the University of Manchester Institute of Science and Technology, Manchester, U.K., in 1985, and the Ph.D. degree in control systems from the University of Salford, Salford, U.K., in 1988. He is currently a Professor of control systems with the Department of Electrical and Electronics Engineering, K. N. Toosi University of Technology, Tehran, Iran. He is the author or a coauthor of about 110 journal papers and fourteen books in the area of control systems. His main research interests are adaptive and robust multivariable control systems, complex systems control, and the history of control. E-mail: sedigh@kntu.ac.ir.

Tor Arne JOHANSEN received the MSc degree in 1989 and the PhD degree in 1994, both in electrical and computer engineering, from the Norwegian University of Science and Technology, Trondheim, Norway. From 1995 to 1997, he worked at SINTEF as a researcher before he was appointed Associated Professor at the Norwegian University of Science and Technology in Trondheim in 1997 and Professor in 2001. He has published several hundred articles in the areas of control, estimation and optimization with applications in the marine, aerospace, automotive, biomedical and process industries. In 2002 Johansen co-founded the company Marine Cybernetics AS where he was Vice President until 2008. Prof. Johansen received the 2006 Arch T. Colwell Merit Award of the SAE, and is currently a principal researcher within the Center of Excellence on Autonomous Marine Operations and Systems (NTNU-AMOS) and director of the Unmanned Aerial Vehicle Laboratory at NTNU. In 2017 he co-founded the spin-off companies Scout Drone Inspection AS and UBIQ Aerospace AS. He is currently the leader of the SmallSat Lab at NTNU. E-mail: tor.arne.johansen@itk.ntnu.no.