Daisy Chaining Kalman Filter Control Allocation

Wissam Sayssouk, Rodolfo Orjuela, Mario Cassaro, Clement Roos, Michel Basset

Daisy Chaining Kalman Filter Control Allocation

Wissam Sayssouk[1,2], Rodolfo Orjuela[1], Mario Cassaro[2], Clement Roos[2] and Michel Basset[1]

***Abstract*— In this article, a novel Control Allocation (CA) approach based on Daisy Chaining and Kalman Filter CA (DCKFCA) approaches is presented. The proposed algorithm aims at overcoming the most common limitations of the existing algorithms: compensation of the different actuator dynamics and switching between different groups of actuators. These two limitations impact negatively the performance of the overall system and the closed loop stability. Daisy Chaining rearranges the actuators into groups, and then the CA problem is solved using the Kalman Filter. This approach has already shown promising results on a realistic simulator of the longitudinal control of an autonomous vehicle.**

I. INTRODUCTION

In recent years, control allocation (CA) has become an important topic in the field of control systems, as it plays a crucial role in the design of over-actuated systems. Control allocation refers to the process of distributing a limited control effort among multiple control inputs to achieve a desired system behavior [1]. It is commonly used in aeronautics and autonomous vehicles to assign control inputs to various actuators in order to achieve a desired performance or behavior [2], [3], [4]. The need to specify several goals has appeared with input-redundant systems, a system with more inputs than degrees of freedom (DOF).
A typical CA problem as seen in Fig. 1 consists of finding the optimum control input u_{cmd} to realize the desired virtual control input v_{desire}. Traditional methods, such as pseudoinverse and weighted pseudoinverse [5] are widely used when solving the control allocation problem. However, these methods have some limitations, such as poor performance in the presence of control constraints. To address this issue, most of the existing techniques try to solve a constrained optimization problem by taking into account the physical limitations of the actuators [1]. For example, [6] formulates the problem using linear and quadratic programming. However, among the various existing control allocation techniques, only few researchers take into account the actuator dynamics.

This latter is often neglected under the following two assumptions: actuators respond very quickly to a command and they all have the same dynamics. But when different dynamics are considered, the control strategy becomes challenging. It is because switching from one dynamics to another can negatively impact performance and therefore affect the closed loop stability. It is then a trade-off between performance and stability. In [7], the model predictive control is adopted to solve the control allocation problem, called MPCA. This method deals with different actuator dynamics. However, it requires a high computation effort. In [8] the Kalman Filter (KFCA) is used to solve the problem under the assumption that all actuators have the same dynamics.

To cope simultaneously with actuator saturations and different dynamics, this article proposes a novel control allocation approach, called Daisy Chaining Kalman Filter Control Allocation (DCKFCA). This approach allows the distribution of control commands among a group of actuators. It uses the Kalman Filter which facilitates the compensation of the actuator dynamics. The Daisy Chaining method is used to redistribute control commands in case of saturation and/or prioritization of actuators. This allows smooth transition between the different groups of actuators with the same dynamics. The effectiveness of the proposed approach is demonstrated through simulation results in a case study of longitudinal control for autonomous vehicles.

The paper is structured as follows. The actuator dynamics, control allocation background, as well as current control allocation approaches based on the Kalman Filter and the Daisy Chaining methodologies, are covered in Section II. In Section III, the new DCKFCA approach, including its pseudo-code, is described. Section IV proposes an application to the longitudinal control of autonomous vehicles. DCKFCA is compared first to the KFCA [8], then tested alone in both the nominal and saturated cases. Finally, Section V concludes the paper and suggests areas for future research.

II. PROBLEM FORMULATION

In this section, the actuator dynamics and the control allocation problem are presented first in subsections II-A and II-B. Subsection II-C then provides an overview of the Kalman Filter control allocator [8] and the Daisy Chaining method [9]. This sets the stage for the new approach presented in Section III.

A. Actuator Modeling

The actuator dynamics is represented as a first order system, described as follows:

$$\frac{u^i_{act}(t)}{u^i_{cmd}(t)} = \frac{K_i}{\tau_i s + 1} \tag{1}$$

where τ_i and K_i are the time constant and the gain of the actuator. u^i_{cmd} and u^i_{act} are the commanded and realized commands of the i^{th} actuator shown in Fig. 1.

The transfer function (1) is then discretized using the first order Euler method to be used later in the CA formulation:

$$u^i_{act}(k+1) = u^i_{act}(k) + \frac{\Delta t}{\tau_i}\left(K_i u^i_{cmd}(k) - u^i_{act}(k)\right) \tag{2}$$

[1] IRIMAS EA 7499, Université de Haute-Alsace, 68093 Mulhouse, France, name.surname@uha.fr
[2] ONERA - The French Aerospace Lab, Information Processing and Systems Department, 31400 Toulouse, France, name.surname@onera.fr

where Δt is the sampling time, and:

$$A_{act}^{i} = \frac{\Delta t}{\tau_i} K_i \quad ; \quad B_{act}^{i} = 1 - \frac{\Delta t}{\tau_i} \tag{3}$$

will be used when formulating the KFCA.

B. Control Allocation Background

The control allocation layer is essential to split the control design into two parts: a control law for producing the desired virtual control input (i.e. total moments, total forces...), represented by v_{desire}, and a control allocation layer for distributing it on the available actuators as seen in Fig. 1. This approach has several advantages, such as ease of reconfiguration in the event of actuator dynamics changes, and decoupling of the control allocation from the high-level controller design.

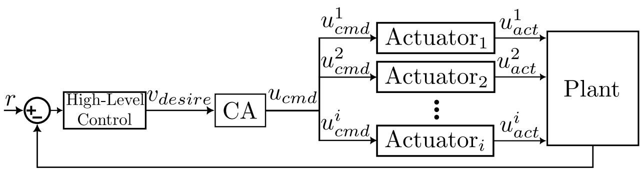

Fig. 1. Control allocation problem scheme.

Consider now the equation:

$$v_{desire} = B u_{cmd} \tag{4}$$

where $v_{desire} \in \mathbb{R}^k$ is the desired virtual control input, $B \in \mathbb{R}^{k \times m}$ the control effectiveness matrix and $u_{cmd} \in \mathbb{R}^m$ the control input. As the system is over-actuated, the number of actuators m is larger than the number of degrees of freedom to be controlled k. The control allocation problem can be described as finding a control input vector u_{cmd} satisfying equation (4). It is noteworthy that equation (4) is applied in situations where the actuator dynamics are not taken into account during the control allocation process.

Additionally, the control input vector u_{cmd} is usually bounded and subjected to position and rate limits as follows:

$$\begin{aligned} u_{cmd}^{p,min} \leq u_{cmd}(t) \leq u_{cmd}^{p,max} \\ u_{cmd}^{r,min} \leq \dot{u}_{cmd}(t) \leq u_{cmd}^{r,max} \end{aligned} \tag{5}$$

where the superscripts p and r denote the position and the rate limits, and *min* and *max* stand for the minimum and maximum.

In discrete time domain and using the first order Euler forward discretization, the position and rate limitations are formulated as follows:

$$\underline{u}_{cmd}(k) \leq u_{cmd}(k) \leq \overline{u}_{cmd}(k) \tag{6}$$

where:

$$\begin{aligned} \underline{u}_{cmd}(k) = \min\{u_{cmd}^{p,min}, u_{cmd}(k-\Delta t) + \Delta t \; u_{cmd}^{r,min}\} \\ \overline{u}_{cmd}(k) = \max\{u_{cmd}^{p,max}, u_{cmd}(k-\Delta t) + \Delta t \; u_{cmd}^{r,max}\} \end{aligned} \tag{7}$$

C. Existing Control Allocation Techniques

From an exhaustive control allocation literature review [1], [2], numerous control allocation techniques have been identified. Most of them consist on solving the control allocation problem neglecting the actuator dynamics under the assumption that actuators react instantaneously to command. Only those that will be useful in the rest of the article are described below.

Daisy Chaining:

Daisy chaining control allocation is a technique that organizes the actuators in a system into different priority levels and assigns control inputs to them in a sequential or "daisy chain" way [9]. This process is repeated until all actuators have been used. The aim of this method is to enhance the system performance by making the best use of the available actuators. This is because the most critical actuators are controlled first, which can help to stabilize the system and prevent it from reaching an unsafe or unstable state. This method is commonly used in aerospace and robotics applications, such as aircraft and robot control [10], [5].

The concept is as follows, consider m control inputs that are divided into M groups of decreasing priority:

$$u_{cmd} = \begin{bmatrix} u_{cmd}^1 & \dots & u_{cmd}^M \end{bmatrix}^T \tag{8}$$

Accordingly, the control effectiveness matrix is partitioned as follows:

$$B = \begin{bmatrix} B_1 & \dots & B_M \end{bmatrix} \tag{9}$$

The control allocation problem is now reformulated as follows:

$$v_{desire} = B_1 u_{cmd}^1 + \ldots + B_M u_{cmd}^M$$

The aim behind the daisy chain is to realize the virtual control v_{desire} with only the first group of actuators:

$$v_{desire} = B_1 u_{cmd}^1 \tag{10}$$

Problem (10) can be solved using any control allocation technique. Let's assume that the solution is given by:

$$u_{cmd}^1 = P_1 v_{desire} \tag{11}$$

where P_1 is any right (pseudo)inverse of B_1. When u_{cmd}^1 satisfies the actuator position and rate limits, the allocation is successful and the algorithm is interrupted. Otherwise, u_{cmd}^1 is saturated according to its position and rate constraints:

$$u_{cmd}^1 = \mathrm{SAT}_{\bar{u}_{cmd}^1}(P_1 v_{desire}) \tag{12}$$

and the unrealized virtual control is realized using the secondary actuator group:

$$B_2 u_{cmd}^2 = v_{desire} - B_1 u_{cmd}^1 \tag{13}$$

Again, when u_{cmd}^2 fails to satisfy (13) or violates some constraints, the solution is saturated and the same process is repeated until either the virtual control demand is met, or all actuator groups have been employed. The Daisy Chaining algorithm can be summarized as follows [9]:

$$\begin{cases} u^1_{cmd} = \mathrm{SAT}_{\bar{u}^1_{cmd}}(P_1 v_{desire}) \\ u^2_{cmd} = \mathrm{SAT}_{\bar{u}^2_{cmd}}(P_2(v_{desire} - B_1 u^1_{cmd})) \\ \quad\vdots \\ u^M_{cmd} = \mathrm{SAT}_{\bar{u}^M_{cmd}}(P_M(v_{desire} - \sum_{i=1}^{M-1} B_i u^i_{cmd})) \end{cases} \tag{14}$$

where P_i are the solutions of the CA problem.

Kalman Filter Control Allocation (KFCA):

It consists of solving the control allocation problem based on Kalman Filtering [8]. The main concept is to reformulate the CA as a state observer, in which the observer states are the commanded and realized actuator commands (u_{cmd} and u_{act}). The observer measurements are the desired virtual control input v_{desire} calculated by the high-level controller. The main idea is to estimate the state of the actuator dynamics in the process model and to employ this information for determining the optimal control input for each actuator. The control allocation is then performed by solving an optimization problem that considers the actuator dynamics and performance criteria.

Consider the following discrete-time model:

$$\begin{aligned} x(k+1) &= Fx(k) + n(k) \\ y(k) &= v_{desire} = Hx(k) + w(k) = Bu_{act} + w(k) \end{aligned} \tag{15}$$

where $x = \begin{bmatrix} u_{cmd} & u_{act} \end{bmatrix}^T \in \mathbb{R}^{2m}$ is the state vector, $y \in \mathbb{R}^k$ the output vector. $n(k)$ and $w(k)$ are uncorrelated white Gaussian noises with zero mean and diagonal covariance matrices $Q \in \mathbb{R}^{m\times m}$ and $R \in \mathbb{R}^{k\times k}$ respectively.

The process model for the design of the KFCA of equation (15):

$$\begin{aligned} F &= \begin{bmatrix} I^{m\times m} & 0^{m\times m} \\ A_{act} & B_{act} \end{bmatrix}; \quad H = \begin{bmatrix} 0^{k\times m} & B \end{bmatrix} \\ Q &= \begin{bmatrix} Q_1 & 0^{m\times m} \\ 0^{m\times m} & Q_2 \end{bmatrix}^T; \quad R \in \mathbf{R}^{k\times k} \end{aligned} \tag{16}$$

where the matrices $A_{act} = \mathrm{diag}[A^1_{act}, \ldots, A^m_{act}]$ and $B_{act} = \mathrm{diag}[B^1_{act}, \ldots, B^m_{act}]$ are the parameters of the different actuators calculated using equation (3).

As previously stated, the states of the Kalman Filter consist of the commanded and realized actuator commands, u_{cmd} and u_{act}, respectively. Hence, the following process equation is modeled as a process with zero derivative as follows:

$$u^i_{cmd}(k+1) = u^i_{cmd}(k) + n_1(k) \tag{17}$$

where $n_1(k)$ is the noise related to this process equation.

Furthermore, the realized actuator commands and referring to equation (2) is modeled as follows:

$$u^i_{act}(k+1) = u^i_{act}(k) + \frac{\Delta t}{\tau_i}(K_i u^i_{cmd} - u^i_{act}(k)) + n_2(k) \tag{18}$$

where $n_2(k)$ is the noise related to this process equation.

In the case of control allocation, the covariance matrices Q and R are a design parameters allowing the modification of the control command dynamics as it will be explained later.

The behavior of the CA is heavily dependent on the values of Q_1, Q_2 and R. These matrices represent the noise associated with actuator inputs and outputs, and measurements, respectively. Q_2 pertains to the noise in the actuator dynamics (18), which is assumed to be known. Therefore, the values of Q_2 is typically smaller than Q_1 to ensure that the actuator dynamics are respected when estimating the actuator inputs u_{cmd}. On the other hand, under the assumption that the virtual control inputs v_{desire} are calculated with high accuracy using the high-level controller, R is also relatively small compared to Q_1. The values Q_2 and R are assigned as design parameters to be tuned. They allow to handle the trade-off between minimizing the error between the v_{desire} and Bu_{act}, and increasing/reducing the response time of the system with respect to (18).

III. Daisy Chaining Kalman Filter Control Allocation

In this section, a new algorithm for dynamic state estimation for control allocation in the presence of different actuator dynamics and saturation is presented. The proposed algorithm, called Daisy Chaining Kalman Filter, combines the principles of Daisy Chaining and Kalman Filtering presented before.

The algorithm uses the Daisy Chaining technique to divide the actuators into different groups, with each group prioritized in a specific order. The Kalman Filter is then used to solve the control allocation problem for the highest priority group first. The parameters of the entire group are frozen when one or more actuators in that group saturate. The second group then allocates the gap between allocated and desired virtual control inputs. When a feasible solution is still not achieved and there are more than two groups, the process is repeated using Kalman for each sub-groups. By using this method, the algorithm prioritizes the allocation of control to the most important actuators while also taking into account actuator saturation. However, it should be noted that depending on the groups chosen, the solution may not fully employ all actuators, resulting in sub-optimal solutions due to physical limitations.

As previously stated, let's consider m actuators rearranged in M different groups of decreasing priority as follows:

$$u_{cmd} = \begin{bmatrix} u^1_{cmd} & \ldots & u^M_{cmd} \end{bmatrix}^T \tag{19}$$

where

$$\begin{aligned} u^1_{cmd} &= \begin{bmatrix} u^{1,1}_{cmd} & \ldots & u^{1,n_1}_{cmd} \end{bmatrix} \\ u^M_{cmd} &= \begin{bmatrix} u^{M,1}_{cmd} & \ldots & u^{M,n_M}_{cmd} \end{bmatrix} \end{aligned} \tag{20}$$

and $m = \Sigma^M_{i=1} n_i$. The first superscript of u_{cmd} indicates the group number, while the second superscript indicates the actuator index.

In line with the previous equation, the actuator outputs are also divided in the same manner, whereby:

$$u_{act} = \begin{bmatrix} u_{act}^1 & \dots & u_{act}^M \end{bmatrix}^T \tag{21}$$

where

$$\begin{aligned} u_{act}^1 &= \begin{bmatrix} u_{act}^{1,1} & \dots & u_{act}^{1,n_1} \end{bmatrix} \\ u_{act}^M &= \begin{bmatrix} u_{act}^{M,1} & \dots & u_{act}^{M,n_M} \end{bmatrix} \end{aligned} \tag{22}$$

Algorithm 1: DCKFCA Algorithm

Input: $x_k^1 \dots x_k^M, P_k^1 \dots P_k^M$, v_{desire}
Output: $u_{cmd}, P_{k+1}^1 \dots P_{k+1}^M$
Initalize: $F^1 \dots F^M$, $H^1 \dots H^M$, $Q^1 \dots Q^M$, $R^1 \dots R^M$
$v^1 \leftarrow v_{desire}$
$i \leftarrow 1$
while $v^i \neq 0$ **do**
  $[x_{k+1}^i, P_{k+1}^i]$ = KF $(x_k^i, P_k^i, v^i, Q^i, R^i, F^i, H^i)$
  $u_{act}^i \leftarrow x_{k+1}^i(\frac{n_i}{2}+1 : n_i)$
  if $B^i u_{act}^i > \mathrm{SAT}(B^i u_{act}^i)$ **then**
    $u_{cmd}^i \leftarrow \mathrm{SAT}[x_{k+1}^i(1 : \frac{n_i}{2})]$
    $v^{i+1} \leftarrow v^i - \mathrm{SAT}[B^i u_{act}^i]$
  else
    $u_{cmd}^i \leftarrow x_{k+1}^i(1 : \frac{n_i}{2})$
  end
  $i \leftarrow i+1$
end
return $u_{cmd} = \begin{bmatrix} u_{cmd}^1 & \dots & u_{cmd}^M \end{bmatrix}$

function KF $(x_k, P_k, y, Q, R, F, H)$
  % Time Update ('Prediction')
  $x_{k+1} = F x_k$
  $P_{k+1} = F P_k F^T + Q$
  % Measurement Update ('Correction')
  $K_k = P_{k+1} H^T (H P_k H^T + R)^{-1}$
  $x_{k+1} = x_{k+1} + K_k (y - H x_{k+1})$
  $P_{k+1} = P_{k+1} - K_k H P_{k+1}$
  return x_{k+1}, P_{k+1}
end function

The aim is to use the Kalman Filter to solve the control allocation problem for group 1. The filter uses the measurements provided by the high-level control v_{desire}. When any actuator in group 1 saturates, the value of u_{cmd}^1 is frozen and the algorithm proceeds to allocate the remaining necessary generalized desired virtual control input to group 2. Then, the Kalman Filter is used again to solve the control allocation for group 2. The process is repeated for all the groups until a feasible solution is achieved. The DCKFCA algorithm prioritizes the allocation of control to the most important actuators (group 1) while also taking into account the actuator's saturation constraints. This way, the algorithm improves the accuracy of the control allocation problem in the presence of different actuator dynamics and saturation. The proposed algorithm can be adapted to various actuator dynamics by rearranging the different dynamics into subgroups and considering the actuators' bandwidth during the implementation process. This makes the algorithm a versatile tool for different applications. Algorithm 1 summarizes the main steps of the proposed algorithm. The matrices $P_k^1 \dots P_k^M$ represent the covariance matrices, initialized to zero, and SAT the saturation function. As described in Section II-C, the KFCA method involves the rearranging of the input u_{cmd} and output u_{act} as state observer. In the case of DCKFCA, actuators with similar dynamics are grouped together. Consequently, the matrices F, H, Q, and R in equation (16) are partitioned into distinct groups of appropriate dimensions, resulting in the following:

$$\begin{aligned} F &\rightarrow \{F^1 \quad \dots \quad F^M\}; \quad H \rightarrow \{H^1 \quad \dots \quad H^M\} \\ Q &\rightarrow \{Q^1 \quad \dots \quad Q^M\}; \quad R \rightarrow \{R^1 \quad \dots \quad R^M\} \end{aligned} \tag{23}$$

By virtue of this partitioning, it becomes possible to construct different discrete-time models, denoted as $x_k^1 \dots x_k^M$, based on different dynamic actuator models, as elaborated in Section II-C.

Moreover, the Kalman Filter function consists of two main steps: the prediction step and the update step. During the prediction step, the filter uses the dynamical model to predict the system's state at the next time step, based on its current state. In the update step, the filter compares the predicted state x_{k+1} with the actual measurements $y = v^i$ of the system and adjusts the state estimate to account for any errors in the measurements. More details can be found in [11].

IV. A Case study: Application to Longitudinal Control for Autonomous Vehicle

In this section, the proposed algorithm is employed for the longitudinal control of a vehicle equipped with 4 in-wheel electric motors. A nonlinear vehicle model is introduced first, and then the high-level control and control allocation strategies are proposed. The suggested algorithm is then evaluated using MATLAB/Simulink on this benchmark.

A. Nonlinear Vehicle Model

In this study, the vehicle is modeled using a 7 degrees of freedom (DOF) nonlinear model to investigate the longitudinal control [12]. The impact of the suspension system is not considered in the model.

The longitudinal dynamics equation, used to balance the forces acting in the longitudinal direction, is given as [12]:

$$\begin{aligned} m(\dot{v}_x - v_y \dot{\psi}) = (F_{x_{fl}} + F_{x_{fr}})\cos(\delta_f) + (F_{x_{rl}} + F_{x_{rr}}) \\ - (F_{y_{fl}} + F_{y_{fr}})\sin(\delta_f) \end{aligned} \tag{24}$$

v_x, v_y and $\dot{\psi}$ are the longitudinal and lateral speeds, and yaw rate respectively. F_{x_i} and F_{y_i} denote the longitudinal and lateral tire forces respectively. They are computed using the Magic Formula model [13]. The subscript $i \in \{fl, fr, rl, rr\}$ refers to the front, rear, left and right wheels. These subscripts will continue to be used in the subsequent equations. The vehicle front steering angle is represented by δ_f, and the vehicle mass by m.

The wheel dynamics is represented as follows:

$$I_\omega \dot{\omega}_i = T_i - F_{x_i} R_\omega \tag{25}$$

where $\dot{\omega}_i$ is the wheel speed and T_i is the drive or brake torque at wheel i, I_ω its rotational inertia, and R_ω the effective radius of the wheel.

B. High-Level Longitudinal Controller

A high-level controller is required to generate a virtual total torque v_{desire} which is then distributed using the CA among the available actuators.

The Lyapunov theory approach is employed to design a model-based longitudinal controller, which is synthesized using the longitudinal dynamics equation (24) and the wheel dynamics equation (25). Under the assumption of small wheel slip, negligible v_y, and $\delta_f = 0$ the following equation holds:

$$m_v \dot{v}_x = \frac{4T}{R_\omega} \tag{26}$$

where $m_v = m + \dfrac{4I_\omega}{R_\omega^2}$.

To ensure a good velocity reference tracking and the convergence towards zero, the following positive definite Lyapunov candidate is chosen:

$$\begin{aligned} V &= \frac{1}{2} e_v^2 \\ e_v &= v_{x_{ref}} - v_x \end{aligned} \tag{27}$$

To achieve exponential stability, it is necessary to fulfill the following condition [14]:

$$\dot{V} = -K_x V \tag{28}$$

where K_x is a strictly positive parameter representing the decay rate. With respect to the stability condition (28), the following longitudinal control law holds:

$$T^* = \frac{R_\omega}{4}(m_v(a_{x_{ref}} + K_x e_v)) = v_{desire} \tag{29}$$

where $a_{x_{ref}}$ and $v_{x_{ref}}$ are respectively the references acceleration and speed and T^* is the desired virtual control input v_{desire}.

C. Low-Level Control Allocation

The low-level control allocation refers to the process of distributing to the actuators the virtual control input $v_{desire} = T^*$ calculated using the high-level controller. The total longitudinal forces is described as follows:

$$F_x = F_{x_{fl}} + F_{x_{fr}} + F_{x_{rl}} + F_{x_{rr}} \tag{30}$$

At the steady state, equation (25) is written as:

$$F_{x_i} = \frac{T_i}{R_\omega} \tag{31}$$

The used in-wheel motor can operate in one mode at a time, torque vectoring or braking.

The desired virtual control input $v_{desire} = T^*$ is the total torques calculated using the high-level control (29) and the input vector $u_{cmd} = \begin{bmatrix} T_{fl} & T_{fr} & T_{rl} & T_{rr} \end{bmatrix}^T$. Hence, the control effectiveness matrix for this case is defined as

$$B = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \tag{32}$$

D. Results and Discussions

In this section, the efficiency of the proposed control allocation algorithm is evaluated by using the highly accurate 7 DOF vehicle model under MATLAB/Simulink. The performance of the Daisy Chaining Kalman Filter control allocation approach is evaluated using two scenarios. It is compared to a similar approach found in the literature, Kalman Filter CA, and then tested independently in normal and saturation conditions.

The application at hand involves different dynamics for the four in-wheel electric motors at the front and rear of the vehicle. The different actuator parameters are as follows: $K_{fl} = K_{fr} = 1$ and $K_{rl} = K_{rr} = 0.8$, $\tau_{fl} = \tau_{fr} = 40$ (ms), $\tau_{rl} = \tau_{rr} = 20$ (ms), and the maximum input torques $u_{cmd}^{max} = \begin{bmatrix} 350 & 350 & 380 & 380 \end{bmatrix}^T$ (Nm). The other parameters are given in Table I.

TABLE I

VEHICLE, CONTROLLER AND ALLOCATOR PARAMETERS

Vehicle parameters			
I_ω R_ω	0.99 (mKg^2) 0.313 (m)	m	1828 (Kg)
Controller parameters			
K_x	5		
Allocator weighting matrices			
R_{KFCA}	0.001	R_{DCKFCA}	0.001
Q_{KFCA}	$diag([10,10,10,10,0.001,0.001,0.001,0.001])$		
Q^1_{DCKFCA}	$diag([10,10,0.001,0.001])$		
Q^2_{DCKFCA}	$diag([10,10,0.001,0.001])$		

In the two presented scenarios, a linear acceleration profile is employed, starting with a rate of 1 m/s^2 for the first 20 seconds and subsequently increasing to 1.5 m/s^2 for the following 10 seconds.

<u>*Scenario 1:*</u> KFCA vs DCKFCA

In this scenario, the proposed DCKFCA approach is compared to the Kalman Filter Control Allocation (KFCA) approach previously presented in Section II-C. As the front and rear actuators have different dynamics, the actuators with similar dynamics are grouped together in the DCKFCA approach to form two distinct groups. The total torque is then divided equivalently between the two groups and distributed among the individual actuators within each group to allow the comparison with the KFCA. The weighting functions Q and R for KFCA and DCKFCA are given in Table I.

As shown in Fig. 2, the input and output torque distribution for the various actuators are depicted in order to evaluate and compare the performance of the proposed control allocation approach to the KFCA one. The objective of the control allocation is to compensate the actuator dynamics by ensuring similar actuators outputs to avoid asymmetric behavior. As depicted in Fig. 2, it is clear that the actuator outputs are not equal in the case of the KFCA approach, in contrast to the proposed approach. This enhances the overall performance of the system ensuring to have similar actuator outputs.

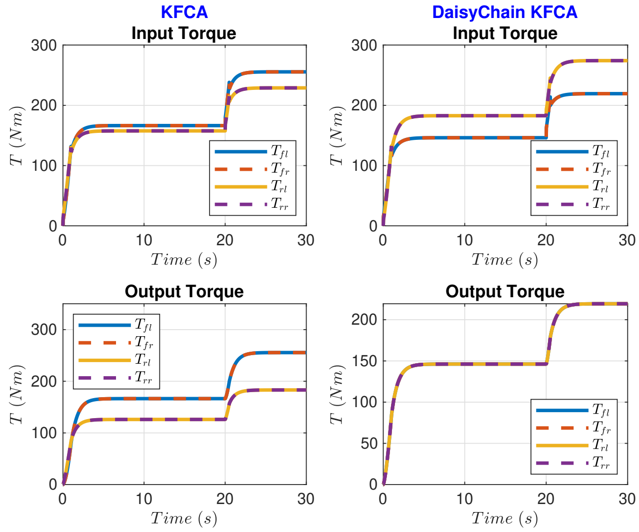

Fig. 2. Actuator Inputs/Outputs in Scenario1.

Scenario 2: DCKFCA in normal and saturated cases
In this scenario, the proposed DCKFCA approach is evaluated under both normal and saturated conditions. The front actuators are designated as high-priority group (group 1) when utilizing the proposed DCKFCA approach to solve the control allocation problem. The reason for this choice is that the rear actuators operate faster than the front ones, enabling more effective tracking in situations of saturation by utilizing the faster actuators in group 2. The weighting functions remain the same as in scenario 1. In Fig. 3, the total torque calculated using the high-level controller (29) is compared to the actuator outputs. It can be seen that there is a good tracking with a relative small error between what is demanded, v_{desire}, and what is realized. Furthermore, as shown in Fig. 4, during the initial period of $[0\,\text{s} - 20\,\text{s}]$, the total torque is compensated completely by the first group (front actuators). Once this group reaches saturation, between $[20\,\text{s} - 30\,\text{s}]$, the remaining torque is sent to the second group, thus allowing for accurate reference tracking. Although utilizing the same effectiveness matrix B, the approach allowed for the compensation of the different actuator gains. Thus, a greater input is anticipated for the rear actuators since they have a different gain from one ($K_{rl} = K_{rr} = 0.8$) as seen in Fig. 4.

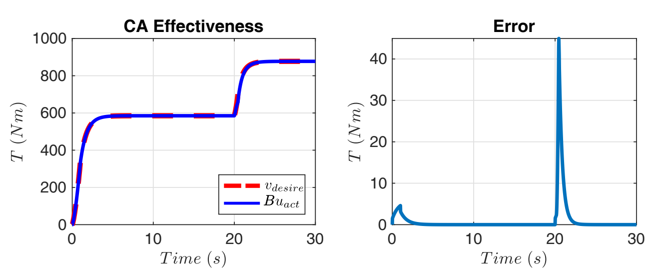

Fig. 3. Total torque commanded and its relative error in Scenario 2.

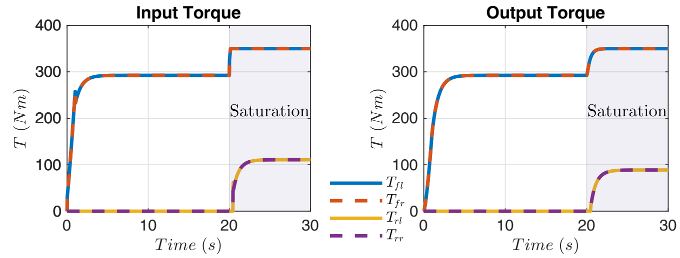

Fig. 4. Actuator Inputs/Outputs using DCKFCA in Scenario 2.

V. CONCLUSIONS

The approach presented in this article, the Daisy Chaining Kalman Filter Control Allocation (DCKFCA), has been shown to be more effective than another method found in the literature, the Kalman Filter Control Allocation (KFCA), when different actuator dynamics are involved. Through detailed analysis and simulation, it has been shown that the proposed approach offers improved performance and greater efficiency in the case of different actuator dynamics and saturation, allowing smooth transition between the different groups in the case of saturation. The proposed approach has the potential to be efficient and can be considered for future research and implementation.
As future work, the nonlinear actuator models will be investigated to improve the performance in more complex environment.

REFERENCES

[1] Tor A. Johansen, Thor I. Fossen, (2013), "Control allocation - A survey", Automatica, volume 49, issue 5:1087-1103.
[2] William S. Levine, (2011), "The Control Systems Handbook: Control System Advanced Methods", Second Edition. CRC Press.
[3] Edouard Sadien et al., (2020), "A simple and efficient control allocation scheme for on ground aircraft runway centerline tracking", Control Engineering Practice, volume 95, issue 2:104228.
[4] Faïza Khelladi et al., (2022), "An Emergency Hierarchical Guidance Control Strategy for Autonomous Vehicles", IEEE Transactions on Intelligent Transportation Systems, volume 23, issue 5:1-12.
[5] Michael W. Oppenheimer, David B. Doman and Michael A. Bolender, (2006), "Control Allocation for Over-actuated Systems", 14th Mediterranean Conference on Control and Automation, Ancona, Italy, 1-6.
[6] Mark Bodson, (2002). Evaluation of optimization methods for control allocation. Journal of Guidance, Control, and Dynamics, volume 25, issue 4:703–711.
[7] Moad Kissai et al., (2019), "Model Predictive Control Allocation of Systems with Different Dynamics", IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 4170–4177.
[8] Gianfranco Morani et al, (2019), "Dynamic Control Allocation through Kalman Filtering", American Journal of Engineering and Applied Sciences, volume 12 issue 1:46-56.
[9] Ola Härkegård, (2003), "Backstepping and control allocation with applications to flight control", PhD dissertation, Linköpings Universitet, Linköping.
[10] James Buffington and Dale Enns, (1996), "Lyapunov stability analysis of Daisy chain control allocation", Journal of Guidance Control and Dynamics, volume 19, issue 6.
[11] Mohinder Grewal and Angus Andrews, (2008) "Kalman theory, theory and practice using MATLAB", Third Edition. John Wiley Sons, Inc.
[12] Rajesh Rajamani, (2011), "Vehicle dynamics and control", Springer Science Business Media.
[13] Hans Bastiaan Pacejka, (2005), "Tire and vehicle dynamics", Elsevier.
[14] Rachid Attia et al., (2014), "Combined longitudinal and lateral control for automated vehicle guidance", Vehicle System Dynamics, volume 52, issue 2:261–279.