Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Control Allocation for Wheeled Mobile Robots Subject to Input Saturation

Jhomolos G. Alves * Fernando Lizarralde ** João C. Monteiro **

* Brazilian Air Force, Brazil
** Dept. of Electrical Eng., Federal University of Rio de Janeiro, Brazil

Abstract: This paper addresses the problem of input saturation in wheeled mobile robot control. Depending on the desired trajectory or on the terrain, wheel motors may be demanded to work beyond their limits. This situation can lead to undesired performance, and therefore, input saturation has to be properly managed. Although control allocation has been mainly employed in overactuated systems to enhance the control distribution, it can be also utilized for underactuated systems as well to grant performance. For this purpose, three different control allocation strategies, along with Lyapunov-based time-varying feedback controller, are applied to a differential drive mobile robot subject to input saturation. Experimental results illustrate the performance of the proposed strategies.

Keywords: Robotics, Nonholonomic Systems, Wheeled Mobile Robots, Control Allocation, Input Saturation

1. INTRODUCTION

Much attention has been given to the development of control strategies for wheeled mobile robots. The main difficulty in such systems arises from the fact that their motion is subject, in general, to nonholonomic constraints and, consequently their have more degrees of freedom than control inputs (underactuated systems). A nonholonomic system cannot be asymptotically stabilized by a time-invariant control law in the Lyapunov sense (Brockett, 1983).

Nonetheless, much of the largely known control strategies do not properly handle situations where vehicle actuators can be required to work beyond its limits, for instance, in a fast trajectory with elevated performance requirements or even in terrains with inclination, where additional torques can be required from the wheel motors.

In order to solve input saturation, a typical approach for linear systems is to add an anti-windup mechanism in the controller architecture (Galeani et al., 2009), e.g. PID control. For mobile robot control, typical control techniques must undergo ad-hoc modifications to handle actuator constraints, e.g. for backstepping control (Lee et al., 2001) and adaptive control (Huang et al., 2013). In Necsulescu et al. (1995) is proposed a predictive control to prevent torque saturation, wheel slippage and robot tip-over. In Chen (2014) is proposed a saturated controller combined with a dynamic time-varying controller with slope restrictions, obtained by applying a finite-time control strategy. In Huang et al. (2018), a robust neural network-based control scheme is utilized to perform stabilizing, tracking and to solve input saturation.

⋆ This study was financed in part by CNPq and the Coordenacao de Aperfeicoamento de Pessoal de Nivel Superior - Brasil (CAPES) - Finance Code 88887.136349/2017-00 and 001.

The contribution of this work is to pursue a different approach by adopting control allocation strategies. Control allocation has been mainly applied in overactuated systems, which have the advantage of providing fault tolerance and control reconfiguration (Johansen and Fossen, 2013) by combining multiple actuators to reproduce a given desired effort, according to an objective.

Here we propose to use several control allocation strategies to grant performance in the case of input saturation, e.g. in situations where the controller may demand an effort beyond the capabilities of the system, and hence, the actuators work saturated. In this sense, control allocation is useful for both over or underactuated.

The advantage of control allocation over typical approaches to solve input saturation lies on the fact that it can be split into two steps - a typical control law, called high-level controller, and the control allocation itself, typically a constrained optimization problem. Thus, the general strategy does not depend on high-level controller, and several high-level controller strategies could be considered to enhance control since the input saturation keeps being managed properly in the control allocation level. In Monteiro et al. (2016) and Kirchengast et al. (2018), the authors have adopted the backstepping and sliding mode techniques as high-level controllers with different control allocation strategies to deal with actuator constraints in quadrotors. In Tohidi et al. (2017), an adaptive control allocation method is proposed for overactuated systems.

The paper is organized as follows: Section II presents a general modelling of nonholonomic mobile robots, the input saturation problem and also important aspects concerning control allocation. In Section III, three different control allocation algorithms are presented: Direct Control Allocation, Weighted Least Squares with Active Set and Linear

Programming with Simplex. In Section IV, experiments in a differential drive mobile robot are undertaken and their results discussed.

2. PROBLEM FORMULATION

The kinematic model of a nonholonomic mobile robot can be described (Siciliano et al., 2009) with respect to a given inertial frame as

$$\dot{q} = G(q)\ \tau \tag{1}$$

where $q \in \mathbb{R}^d$ is a vector of generalized coordinates which describes completely the configuration of the robot, $\tau \in \mathbb{R}^n$ corresponds to the virtual control input vector, n accounts for the amount of differential degrees of freedom of the system and $G(q) \in \mathbb{R}^{d\times n}$ is composed by the input vector field columns $g_j(q)$ with $j = 1, \ldots, n$.

Due to the rolling and sliding constraints in the wheels, such systems have more degrees of freedom d than the number of control inputs n, and therefore, the matrix $G(q)$ has more rows than columns.

On the other hand, the virtual control input vector τ is a result of the wheel velocities involved in the motion. Thus, let $B \in \mathbb{R}^{n\times m}$ be the input matrix, responsible for performing a mapping between τ and the control input vector $u \in \mathbb{R}^m$ applied to the wheel actuators, i.e.

$$\tau = B\ u \tag{2}$$

where the input matrix B can be seen as the contribution of each wheel to the differential degrees of freedom of the vehicle. In the control allocation context, B is denoted control effectiveness matrix, which is particular to the wheeled mobile robot under analysis.

2.1 Input Saturation

Actuators usually work in their linear region, whose range is bounded by upper and lower limits $\overline{u}$ and $\underline{u}$, respectively, given by

$$\underline{u} \le u_i \le \bar{u},\ \forall i \tag{3}$$

A desired virtual control input vector τ_d can be achieved only if the corresponding desired control input u_d lies within this range. Otherwise, a phenomena called saturation arises, which is a non-linearity present in every actuator, and therefore, it cannot be neglected.

With the inverse mapping of (2) and under the assumption that B is invertible, we can retrieve how much effort the virtual control input τ requires from the actuators, according to

$$u = B^{-1}\ \tau \tag{4}$$

Given a desired virtual control input τ_d and its corresponding desired control input vector u_d, if u_d does not lie in the range in (3), the actuators reproduce a saturated control effort u_{sat}, such that $u_{d,i} > u_{sat,i}$ for one or more actuators. Thus, τ_d cannot be attained and a deteriorated τ is delivered, such that

$$\tau = B\ (B^{-1}\ \tau_d)_{sat} \tag{5}$$

Hence, τ does not fulfill the control requirements since there exists one or more virtual control inputs τ_j, such that $\tau_j < \tau_{d,j}$. Due to saturation, a control input $u_{sat,i}$ may lie whether onto the limit $\underline{u}$ or $\overline{u}$ for one or more actuators, which consequently changes the direction of τ with respect to τ_d. Depending on the saturation level and on the actual state, the robot deviates from the desired trajectory and may be unable to recover itself, since increasing τ_d will be produced so as to take the robot to the desired state, although the actuators still remain saturated.

2.2 Control Allocation

The control allocation problem consists of finding a control input vector $u \in \mathbb{R}^m$, such that a virtual desired control input τ_d is produced by all the actuators at every instant t. If u satisfies the constraints in (3), then $u \in \mathbb{U}$, where $\mathbb{U}$ is the feasible set of control inputs (Johansen and Fossen, 2013). Similarly, let $\mathbb{A}$ be the feasible set of virtual control inputs τ, obtained from the inverse mapping in (4).

In case of saturation, τ_d cannot be attained, and therefore, the control allocation problem must primarily find a "best possible" solution, which consists basically of minimizing the virtual control input error. Hence, it is called error minimization and calculated by the optimization problem

$$\begin{aligned} \text{min: } & f(u) = \|Bu - \tau_d\| \\ \text{s.t.: } & \underline{u} \le u_i \le \overline{u},\ \forall i \end{aligned} \tag{6}$$

The most common norms utilized in the cost function are the l_1, l_2 and l_∞ ones (Bodson and Frost, 2011). There are situations where control minimization is desired, such as to reduce energy consumption or mechanical stresses (Oppenheimer et al., 2010), although it is mostly regarded as a secondary objective. Thus, the control allocation problem may also pose as a mixed optimization problem, in the form

$$\begin{aligned} \text{min: } & f(u) = \|u - u_p\| + \gamma\,\|Bu - \tau_d\| \\ \text{s.t.: } & \underline{u} \le u_i \le \overline{u},\ \forall i \end{aligned} \tag{7}$$

where u_p is a preferred control input vector and the weight adjust factor γ determines how much the error minimization must be prioritized over the control minimization.

Hierarchically conceived, the control algorithm consists of two levels, as depicted in the fig. 1. It begins with the high-level controller, responsible for calculating τ_d, even though the controller does not know much about the system, given that τ_d is calculated based only on the desired state q_d and on the actual state q. The high-level controller can be any conventional control technique, such as PID, Lyapunov-based control (Aicardi et al., 1995), sliding mode (Young et al., 1999), adaptive, etc. In the second level, a control allocation algorithm maps τ_d into individual actuator positions u_i. If τ_d corresponds to a feasible virtual control, i.e. $\tau_d \in \mathbb{A}$, u is straightforwardly computed and the problem is solved. Otherwise, u is determined at the cost of degrading the system performance.

Prior to defining a control allocation technique, the primary and secondary (if applicable) objectives must be regarded, while taking into account the main characteristics of the technique to be employed and the benefits of the norm of the cost function adopted (Bodson and Frost, 2011). Other related aspects are the number of actuators and their layout, operating ranges and energy consumption (Johansen and Fossen, 2013). However, unlike other

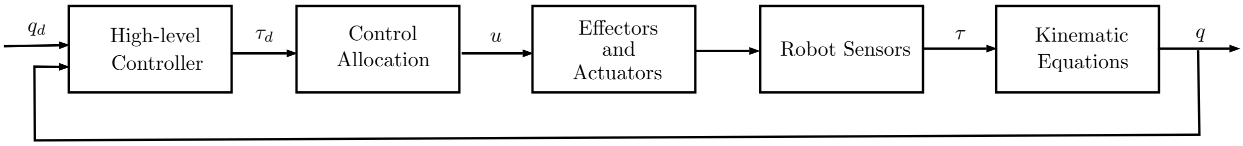

Fig. 1. Control system structure concerning the application of control allocation in wheeled mobile robots.

typical approaches, the input saturation problem is not directly managed by the control law adopted, but only in the second level. This approach allows to employ any high-level controller or even replace it by another one easily, while the second level keeps managing input saturation independently.

3. CONTROL ALLOCATION ALGORITHMS

In this section, three control allocation algorithms are outlined, which are the Direct Control Allocation, Linear Programming with Simplex and Weighted Least Squares with Active Set.

3.1 Direct Control Allocation

The definition of control allocation was firstly presented in Durham (1993), where the author proposed a method for allocating controls when $\tau_d \notin \mathbb{A}$. The objective of the Direct Control Allocation (DCA) is to find a feasible control input, such that the direction of the desired virtual control remains unchanged and a performance deterioration can be predicted when saturation occurs.

In Bodson (2002), the author reformulated the problem as a linear optimization problem with equality constraints to determine the scale factor α and u, in the form

$$\begin{aligned} \max : \; & f(\alpha) = \alpha \\ \text{s.t.} : \; & Bu = \alpha\tau_d, \;\; \alpha\tau_d \in \mathbb{A} \end{aligned} \tag{8}$$

where $\alpha < 1$ means that the control power demand cannot be met and τ_d must be scaled back to lie onto the boundary $\delta\mathbb{A}$. In case where $\alpha > 1$, τ_d can be attained since $u_d \in \mathbb{U}$. Although this problem poses as a linear program, its formulation is concerned with neither error minimization nor mixed optimization.

3.2 Linear Programming with Simplex

Let ξ be a basic feasible solution of a linear programming problem, such that $\xi = \begin{bmatrix} u^T & u_s^T \end{bmatrix}^T$ and $u_s \in \mathbb{R}^n$ is a vector of slack variables. For the error minimization problem in (6) with the l_1-norm, a conversion from a standard linear programming problem to the context of control allocation is proposed in Oppenheimer et al. (2006), which is [1]

$$\begin{aligned} \min : \; & f(u_s) = [0 \; \dots \; 0 \; 1 \; \dots \; 1] \begin{bmatrix} u \\ u_s \end{bmatrix} \\ \text{s.t.} : \; & \begin{bmatrix} u_s \\ -u \\ u \\ -Bu + u_s \\ Bu + u_s \end{bmatrix} \geq \begin{bmatrix} 0 \\ -\bar{u} \\ \underline{u} \\ -\tau_d \\ \tau_d \end{bmatrix} \end{aligned} \tag{9}$$

[1] Vector inequalities have to be considered as element-wise inequalities.

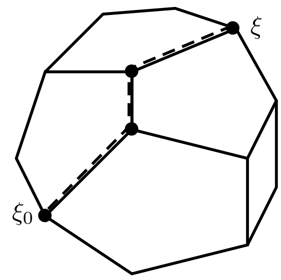

Fig. 2. The Simplex algorithm covers every adjacent vertex of a feasible polytope to find the solution that minimizes the cost function.

where the coefficient vector of the cost function is composed by as many zeros and ones as the size of the vectors u and u_s, respectively.

Assuming that the optimization problem in (9) underwent a pre-processing phase for removing redundancies and linearly dependent rows and columns, now it can be solved with any standard linear programming solver. A common solver is the Simplex algorithm, which consists of finding an initial basic feasible solution ξ_0 and covering adjacent vertices with successive lower function cost $f(u_s)$ in a polytope defined by its feasible set, as depicted in the fig. 2. The search for an optimal feasible basic solution ends when it is no longer possible to decrease the cost $f(u_s)$.

Given that a Simplex solution is always located at the vertex of a feasible polytope, some differential degrees of freedom are prioritized to the detriment of others. Another consequence is an unbalanced control distribution, since at every instant, one or more actuators may be required to operate at their limits, while the remaining ones work with a lower load.

3.3 Weighted Least Squares with Active Set

In Härkegård (2002), the mixed optimization problem is revisited as a quadratic programming problem, which can be written as

$$\begin{aligned} \min : \; & f(u) = \|W_u(u - u_p)\|_2^2 + \gamma \|W_\tau(Bu - \tau_d)\|_2^2 \\ \text{s.t.} : \; & \underline{u} \leq u_i \leq \bar{u}, \; \forall i \end{aligned} \tag{10}$$

where W_u and W_v are weighting matrices for setting priority to each actuator and to each degree of freedom, respectively. Its cost function can be rewritten as

$$f(u) = \left\| \underbrace{\begin{pmatrix} \gamma W_\tau B \\ W_u \end{pmatrix}}_{A} u - \underbrace{\begin{pmatrix} \gamma W_\tau \tau_d \\ W_u u_p \end{pmatrix}}_{b} \right\|_2^2 \tag{11}$$

Weighted Least Squares with Active Set (WLSSA) was proposed in Härkegård (2002) and demonstrated that an

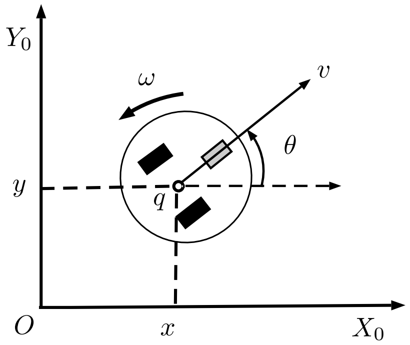

Fig. 3. A two-rear-drive-wheel mobile robot described by its position and orientation with respect to an inertial reference frame (X_0, Y_0).

attainable control input u can be found within a required time span, just like other control allocation algorithms, and hence, it can be utilized in practical situations. Unlike Simplex, WLSSA provides a better and simultaneous convergence of the differential degrees of freedom, since the cost function allows the solutions to be located anywhere in $\mathbb{U}$ and also allows a more balanced control distribution.

4. HIGH-LEVEL CONTROLLER

Consider a differential wheeled mobile robot, as depicted in the fig. 3, with two rear wheels with independent actuators, responsible for providing motion, and a front castor wheel for stabilization purposes only, such that its configuration can be fully described by $q = [\, x \ y \ \psi \,]^T$. The robot cannot move sideways, since it is subject to a nonholonomic constraint represented by the equation

$$[\sin\psi \quad -\cos\psi \quad 0]\,\dot{q} = 0 \tag{12}$$

Assuming no occurrence of slipping, the kinematic modelling of the vehicle can be described by

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = g_1(q)\tau_1 + g_2(q)\tau_2 = \begin{bmatrix} \cos\psi \\ \sin\psi \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega \tag{13}$$

where v is the linear velocity and ω represents the angular velocity.

In the example, the mapping from the right and left wheels velocity $u = [v_r \ v_l]^T$ to $\tau = [v \ \omega]^T$ is given by

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ \frac{1}{R} & -\frac{1}{R} \end{bmatrix} \begin{bmatrix} v_r \\ v_l \end{bmatrix} \tag{14}$$

A Lyapunov-based control law is utilized in Aicardi et al. (1995) to control a wheeled mobile robot in tasks such as navigation, path-following and steering, and for generating the desired virtual control input vector τ_d. However, it is necessary to convert the kinematic equation in (13) from Cartesian to polar coordinates. Consider that the robot, initially at pose q_0 with respect to an inertial frame, is required to travel to a desired pose q_d, as depicted in the fig. 4. The distance error r represents the distance between the poses q and q_d, such that $r > 0$; θ stands for the desired orientation angle and ψ is the actual orientation. The polar coordinates r, α and θ can be written as functions of the Cartesian coordinates, which gives

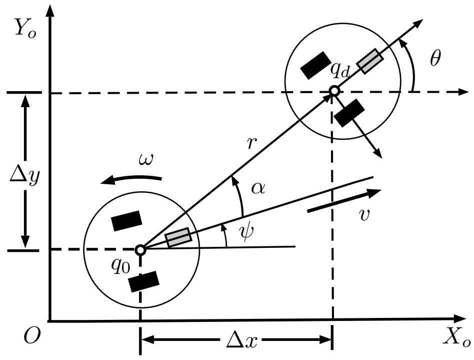

Fig. 4. Representation of the pose of the two-wheeled mobile robot in polar coordinates.

$$\begin{cases} r = \sqrt{\Delta x^2 + \Delta y^2} \\ \theta = \operatorname{atan2}(\Delta y, \Delta x) \\ \alpha = \theta - \psi \end{cases} \tag{15}$$

Then we search a control law that produces a desired virtual control input $\tau_d = [\, v_d \ \omega_d \,]^T$ and makes the state variables converge asymptotically to the limiting point $q = [\, 0 \ 0 \ 0 \,]^T$ in finite time while $r = 0$ is avoided, provided that it is a forbidden state. According to Aicardi et al. (1995), a control law that satisfies the conditions of the Lyapunov stability theory is given by

$$\tau_d = \begin{cases} v_d = (k_2 \cos\alpha) r; \\ \omega_d = k_1 \alpha + k_2 \dfrac{\cos\alpha \sin\alpha}{\alpha} (\alpha + k_3 \theta) \end{cases} \tag{16}$$

which provides global stability properties and drives the vehicle smoothly towards the desired pose q_d.

There are other types of wheeled mobile robots available, composed by castor, fixed or steering wheels, arranged in different layouts, which result in robots that possess two or three degrees of freedom and six different common wheel and axle configurations (Zhao and BeMent, 1992). Then, the Lyapunov stability theory or other control techniques can be utilized to find suitable control laws to be employed as high-level controllers combined with the control allocation algorithms here discussed.

5. EXPERIMENTAL RESULTS

Control allocation algorithms are validated in a *Roomba 621*, a differential drive vacuum robot manufactured by *iRobot* and depicted in the fig. 5. A USB to DIN cable establishes a physical link between the computer and *Roomba*, whereas the MATLAB Toolbox for the iRobot Create 2 (Esposito, 2015) is utilized to create a connection with the robot, to send commands and to read its sensors.

The wheel linear velocities bounds $\underline{u}$ and $\overline{u}$ are set as

$$-0.2 \text{ m/s} \le u_i \le 0.2 \text{ m/s}, \ i = 1, 2 \tag{17}$$

The *Roomba* robot is required to navigate through a preplanned path similar to a lane-change maneuver on a flat plane. It consists of waypoints $(x_{d,i}, y_{d,i})$, such that $x_d = [\, -4.5 \ -4 \ -3.5 \ \ldots \ 3.5 \ 4 \,]^T$ and the next desired coordinate $x_{d,i+1}$ is set when the distance error reaches $r \le 0.1$ m. The desired coordinate $y_{d,i}$ is a function of the actual desired coordinate $x_{d,i}$ and is given by

Fig. 5. The iRobot Roomba 621.

$$y_{d,i} = \begin{cases} -0.9, & x_{d,i} < -1.25 \\ \sin(x_{d,i}), & -1.25 \leq x_{d,i} \leq 1.5 \\ 1, & x_{d,i} \geq 1.5 \end{cases} \tag{18}$$

Controller gains are defined as $k_1 = 3$, $k_2 = 1.5$ and $k_3 = 0.5$. For the WLSSA, the parameters are $W_u = I_2$, $W_\tau = I_2$, $u_p = [0\ 0]^T$ and $\gamma = 10^6$ for prioritizing the error minimization. In the experiment, the robot starts at pose $q(0) = [-5\ -1\ \pi/12]^T$ with respect to an inertial frame and the maneuver lasts $t_s = 50$ s, regardless of the control allocation technique employed.

5.1 *Discussion*

Fig. 6 displays the trajectory executed by the robot when saturation is not properly managed. Although some desired poses are reached during the experiment, the performance is poor, since input saturation leads the vehicle through an inefficient trajectory. When a new desired pose q_d is calculated, the high-level controller computes a desired virtual control τ_d that demands efforts beyond the actuator bounds. Since input saturation is not properly handled, the resulting τ produces large errors in the trajectory.

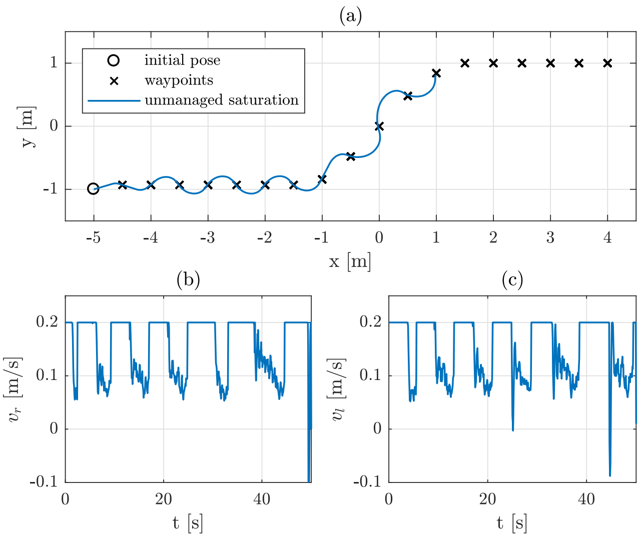

Fig. 6. Unmanaged input saturation. (a) trajectory performed and the desired poses q_d; (b) right wheel velocity; (c) left wheel velocity.

With the DCA algorithm, the direction of the desired virtual control τ_d is maintained, as depicted in the fig. 7, which results in an efficient trajectory towards the desired pose q_d, although not so fast as required.

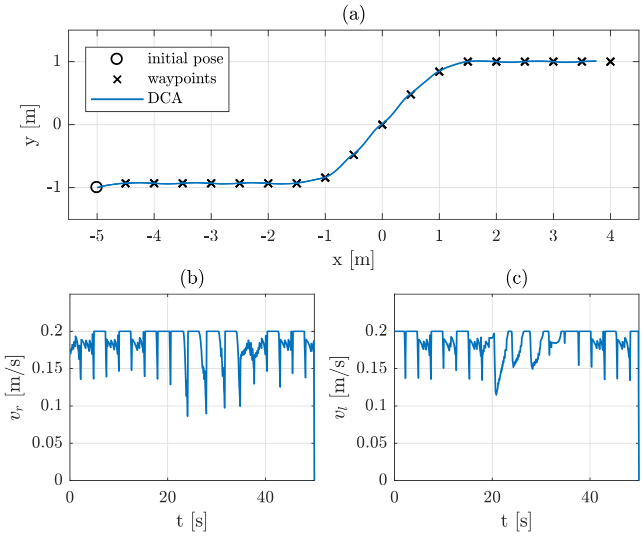

Fig. 7. Input saturation managed by the DCA algorithm. (a) trajectory traveled by the robot; (b) right wheel velocity; (c) left wheel velocity.

The WLSSA and Simplex also present efficient results, as depicted in the figures 8 and 9. Both present equivalent results in terms of tracking, although the dynamics of the linear and angular velocities present some remarkable differences. Since the solutions of the Simplex algorithm are always located on the vertex of a feasible polytope, they are prone to prioritize the angular velocity, whereas WLSSA present a more uniform convergence of the degrees of freedom.

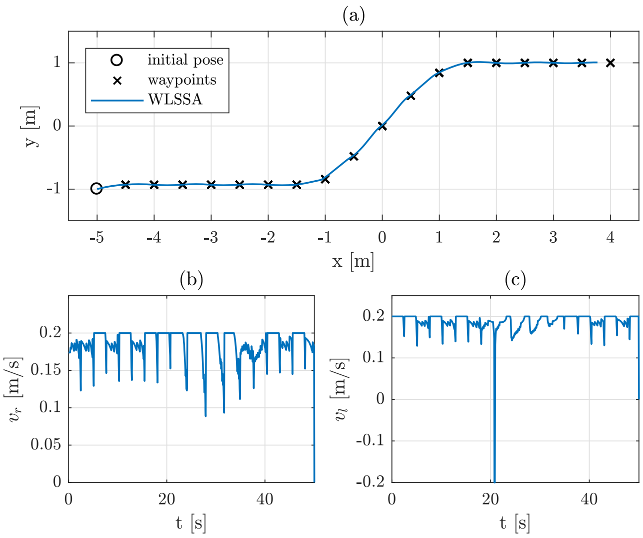

Fig. 8. Input saturation managed by the WLSSA algorithm. (a) trajectory traveled; (b) right wheel velocity; (c) left wheel velocity.

In table 1, some performance data concerning the control allocation algorithms revisited in this experiment are presented, which consists of the relative virtual control error (RVCE), given by

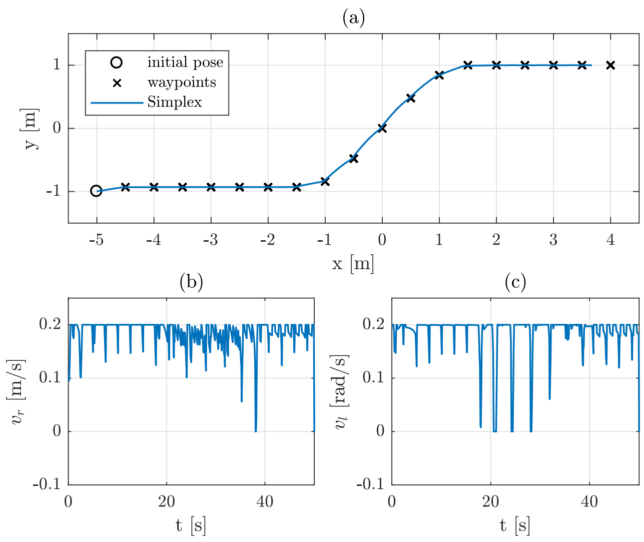

Fig. 9. Input saturation managed by the Simplex algorithm. (a) trajectory traveled; (b) right wheel velocity; (c) left wheel velocity.

$$\mathrm{RVCE} = \|\tau_d - \tau\|_2 \,/\, \|\tau_d\|_2 \tag{19}$$

and the relative control input error (RCIE)

$$\mathrm{RCIE} = \|u_d - u\|_2 \,/\, \|u_d\|_2 \tag{20}$$

The algorithms present no noticeable overall difference in the experiment, although the DCA has presented a higher RVCE, as expected, since it is not related to the error minimization but with the direction maintenance only.

Table 1. Performance Analysis

Algorithm	**DCA**	**WLSAS**	**Simplex**
RVCE (%)	74.8	73.4	73.7
RCIE (%)	72.2	72.3	73.6

6. CONCLUSION

Control allocation was applied to a multivariable underactuated system, which is a commercial differential drive two-wheeled mobile robot, and was subject to input saturation in the experiment, although other types of wheeled mobile robots could have been addressed as well, provided that the control effectiveness matrix is known. The experiment could also have been undertaken in an overactuated system to demonstrate equivalent results.

At the presence of saturation, the control allocation techniques granted performance in the robot and made it work within the linear range of its actuators along the trajectory. They also enforced a behavior according to the norms of the cost function adopted and the control allocation objectives, as detailed along the paper and presented in the table 1.

As topics for future research, control allocation promotes some changes in the direction and in the performance demanded by the high-level controller at the occurrence of saturation, which could be better visualized with a stability analysis. Moreover, other control allocation techniques available could be investigated to manage the saturation problem, just like adaptive control allocation and l_∞ norm algorithms.

REFERENCES

Aicardi, M., Casalino, G., Bicchi, A., and Balestrino, A. (1995). Closed loop steering of unicycle like vehicles via Lyapunov techniques. *IEEE Robotics Automation Magazine*, 2(1), 27–35.

Bodson, M. (2002). Evaluation of optimization methods for control allocation. *Journal of Guidance, Cntrl, and Dyn.*, 25(4), 703–711.

Bodson, M. and Frost, S.A. (2011). Load balancing in control allocation. *Journal of Guidance, Cntrl, and Dyn.*, 34(2), 380–387.

Brockett, R.W. (1983). Asymptotic stability and feedback stabilization. In *Differential Geometric Control Theory*, 181–191. Birkhauser.

Chen, H. (2014). Robust stabilization for a class of dynamic feedback uncertain nonholonomic mobile robots with input saturation. *Int. Journal of Control, Automation and Systems*, 12(6), 1216–1224.

Durham, W.C. (1993). Constrained control allocation. *Journal of Guidance, Control, and Dynamics*, 16(4), 717–725.

Esposito, J.M. (2015). Matlab toolbox for the irobot create 2. www.usna.edu/Users/weapsys/esposito/.

Galeani, S., Tarbouriech, S., Turner, M., and Zaccarian, L. (2009). A tutorial on modern anti-windup design. *European Journal of Control*, 15(3–4), 418–440.

Härkegård, O. (2002). Efficient active set algorithms for solving constrained least squares problems in aircraft control allocation. In *Proc. of the 41st IEEE Conf. on Decision and Control*, 1295–1300.

Huang, H., Zhou, J., Di, Q., Zhou, J., and Li, J. (2018). Robust neural network–based tracking control and stabilization of a wheeled mobile robot with input saturation. *Int. Journal of Robust and Nonlinear Control*, 29(2), 375–392.

Huang, J., Wen, C., Wang, W., and Jiang, Z.P. (2013). Adaptive stabilization and tracking control of a nonholonomic mobile robot with input saturation and disturbance. *Systems & Control Letters*, 62(3), 234–241.

Johansen, T.A. and Fossen, T.I. (2013). Control allocation—a survey. *Automatica*, 49(5), 1087–1103.

Kirchengast, M., Steinberger, M., and Horn, M. (2018). Control allocation under actuator saturation: An experimental evaluation. In *9th IFAC Symposium on Robust Control Design*, 48–54.

Lee, T.C., Song, K.T., Lee, C.H., and Teng, C.C. (2001). Tracking control of unicycle-modeled mobile robots using a saturation feedback controller. *IEEE Trans. on Control Systems Technology*, 9(2), 305–318.

Monteiro, J.C., Lizarralde, F., and Hsu, L. (2016). Optimal control allocation of quadrotor uavs subject to actuator constraints. In *Proc. of the American Control Conference (ACC)*, 500–505.

Necsulescu, D., Lonmo, V., Kim, B., and Vukovich, G. (1995). Autonomous mobile robot control in operational space with torque saturation, slippage and tip-over avoidance. In *Proc. of the IEEE 21st Annual Conf. on Industrial Electronics (IECON)*, 1148–1153.

Oppenheimer, M.W., Doman, D.B., and Bolender, M.A. (2006). Control allocation for over-actuated systems. In *2006 IEEE 14th Mediterranean Conf. on Control and Automation.*

Oppenheimer, M.W., Doman, D.B., and Bolender, M.A. (2010). Control allocation. In *The Control Handbook - Control System Applications*, chapter 8 - Control Allocation. Springer, 2 edition.

Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. (2009). *Robotics*. Springer London.

Tohidi, S., Yildiz, Y., and Kolmanovsky, I. (2017). Adaptive control allocation for over-actuated systems with actuator saturation. *20th IFAC World Congress*, 50(1), 5492–5497.

Young, K.D., Utkin, V.I., and Ozguner, U. (1999). A control engineer's guide to sliding mode control. *IEEE Trans. on Control Systems Technology*, 7(3), 328–342.

Zhao, Y. and BeMent, S.L. (1992). Kinematics, dynamics and control of wheeled mobile robots. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 91–96.