Dynamic Model Predictive Control Allocation using CVXGEN

Martin Hanger, Tor A. Johansen, Geir Kåre Mykland and Aage Skullestad

***Abstract*— Control allocation deals with the allocation of control among a redundant set of effectors, while taking into account the individual constraints. The use of model predictive control (MPC) for control allocation allows the response times of the actuators to be accounted for, and in particular to take advantage of predictions of the virtual control input as well as differences in dynamic control authority and cost of use among the actuators. The use of online quadratic programming (QP) is essential for implementation of the optimal constrained control allocation strategies. The main contributions of the present paper are the investigation of using the software system CVXGEN and the MPC-based control allocation method. CVXGEN synthesizes a customized portable and library-free C-source code QP solver for the specific QP problem resulting from the MPC formulation, exploiting structural properties of the QP and optimizing the source code for execution speed. Two case studies, one being a missile auto-pilot, illustrates the benefits of using the MPC formulation, and the efficiency of CVXGEN.**

I. INTRODUCTION

Some control systems are designed with redundant actuator and effectors, for reasons such as fault tolerance and design issues related to cost, response-time, size, and flexibility. Examples include flight control systems [2], dynamic positioning systems for ships with using thrusters [8], and airjet controlled paper motion in machines [4].

Control algorithm design for systems with input redundancy is challenging since the same control effect (like a generalized force) can be generated by a number of different actuator settings, and actuator constraints should be accounted for. In order to systematically manage such control design challenges, one may decompose the control problem into two parts - a controller that commands a virtual control input of minimal dimension (like the generalized force), and a control allocation module that maps the virtual control input into the redundant actuator settings. Since there are more degrees of freedom available in the actuator system than virtual control variables, the available degrees of freedom in the actuator system can be used to satisfy actuator constraints and to meet secondary objectives such as fault tolerance, power consumption minimization, and actuator wear minimization. In general, the control allocation problem can be formulated as an optimization problem where certain objectives are minimized subject to actuator and effector constraints, and the constraint that the resulting control effect fulfills the requirements of the virtual control command. The main difference between different control allocation methods are related to how the optimization problem is formulated, which models are used, and which numerical algorithms is employed to solve it. This is reviewed in the next paragraphs.

In the classic formulations of the constrained control allocation problem the actuator dynamics are neglected [2], under the assumption that all dynamic phenomena are accounted for by the controller that commands the virtual control to the control allocation module. This may in some cases be an unrealistic and inconvenient assumption when the actuator dynamics are limiting the control performance since response times and different dynamic authorities of the actuators are not taken into account. For systems where actuator dynamics are known, the interactions between the control allocation algorithm and the actuator dynamics working on the aircraft body become more complex, requiring a more sophisticated control allocation method. Actuators can have different response times, i.e. a fast actuator can be used to achieve fast transient response, while slow actuators can be used for steady state or trimmed flight, to improve power efficiency. A Model Predictive Control (MPC) allocation scheme will be able to optimally exploit such properties.

It is relatively straightforward to (re-)design a basic control allocation algorithm to comply with actuator rate constraints, e.g. [8], by incorporating this as a constraint on the change in control inputs from the previous sample to the current sample. More sophisticated dynamic actuator models may be incorporated by using the powerful MPC framework to solve the constrained control allocation problem [9], [10], [14], [20]. MPC is an optimization-based control algorithm which can be used in control allocation, beeing able to handle actuator dynamics as well as actuator saturation. MPC utilizes a model of the plant in predicting outputs and states, where in control allocation this model describes the actuator dynamics. Because of the predictive nature of the controller, the calculated control can pre-act to the actuator system dynamics to improve performance.

How to implement the numerical optimization for the optimal control allocaiton in real time, is a challenging task. Online optimization using off-the-shelf or customized quadratic programming (QP) solvers are studied in the context of linear actuator and effector models in [1],[15],[3], [16]. For nonlinear effector models, the use of sequential quadratic programming is proposed in [5]. Instead of demanding that the optimal control allocation is computed exactly at each sample, the dynamic online optimization appraoch in [6] will at each time instant move in the direction towards an optimal control allocation, but optimality is achieved only asymptotically. The method is extended to the case with actuator dynamics in [18]. While the dynamic online

M. Hanger and T. A. Johansen are with Department of Engineering Cyberentics, Norwegian University of Science and Technology, Trondheim, Norway. tor.arne.johansen@itk.ntnu.no

G. K. Mykland and A. Skullestad are with Kongsberg Defence Systems, Kongsberg, Norway

optimization approach reduces the online computational requirements, and at the same time guarantees that closed loop stability is not lost due to sub-optimality, one may also use multi-parametric programming to pre-compute an explicitly represented piecewise affine solution function. The remaining online computations corresponds to the evaluation of a piecewise linear function resulting from multi-parametric programming and explicit MPC [7],[19]. While this is highly attractive from the online processing point of view, its memory consumption and offline processing does not scale very well - in particular when considering control efficiency matrices that are time- or state-dependent due to nonlinear to time-varying characteristics like in fault tolerant control allocation [17].

This key idea of the present paper is to employ a family of highly customized QP solvers that are automatically generated using CVXGEN [13],[11] to solve MPC-based dynamic control allocation problems. CVXGEN has the unique feature that the C code of the customized solvers is completely standard and standalone, i.e. portable, and extremely efficient since the key structural properties of the QP problem is exploited in the automatic code generation that leads to code with only static data structures and almost branch-free code where for-loops are rolled out for efficiency and deterministic execution on pipeline processor architectures. Performance improvement also comes for low software overhead as the CVXGEN targets small-scale problems, in some contrast to most off-the-shelf solvers that target large-scale problems. Orders of magnitude faster execution compared to state-of-the-art off-the-shelf solvers have been reported on test problems, including MPC problems [13],[11]. This makes it interesting to study CVXGEN's performance in challenging control allocation problems that are of relatively small scale compared to typical MPC problems.

The paper is organized as follows. First the dynamic MPC-based control allocation problem formulation is introduced. Then the use of CVXGEN to address this problem is described, before the computational performance is assessed in a simulation benchmark study.

II. DYNAMIC CONTROL ALLOCATION

A. Optimization problem formulation

It is assumed that all control actuators have dynamics which can be approximately modelled as second order systems,

$$\ddot{\delta} - 2\zeta\omega_0\dot{\delta} - \omega_0^2\delta = \omega_0^2\delta_{cmd} \tag{1}$$

where δ_{cmd} is the commanded control input, and δ is the actuator response. ζ and ω_0 are the actuators relative damping ratio and natural frequency, respectively. Rewritten in state-space form, the model for actuator i will be on the form

$$\dot{\boldsymbol{\delta}}_i = A_{\delta_i}\boldsymbol{\delta}_i + B_{\delta_i}\boldsymbol{\delta}_{cmd,i} \tag{2}$$

For a system with K actuators and effectors, the model will be

$$\begin{bmatrix}\dot{\boldsymbol{\delta}}_1\\ \vdots\\ \dot{\boldsymbol{\delta}}_K\end{bmatrix} = \begin{bmatrix}A_{\delta_1} & 0 & \cdots & 0\\ 0 & A_{\delta_2} & \cdots & 0\\ \vdots & & \ddots & \\ 0 & & \cdots & A_{\delta_K}\end{bmatrix}\begin{bmatrix}\boldsymbol{\delta}_1\\ \vdots\\ \boldsymbol{\delta}_K\end{bmatrix} + \begin{bmatrix}B_{\delta_1}\\ \vdots\\ B_{\delta_K}\end{bmatrix}\begin{bmatrix}\boldsymbol{\delta}_{cmd,1}\\ \vdots\\ \boldsymbol{\delta}_{cmd,K}\end{bmatrix} \tag{3}$$

In a more compact form, this can be written

$$\dot{\boldsymbol{\delta}} = A_\delta\boldsymbol{\delta} + B_\delta\boldsymbol{\delta}_{cmd} \tag{4}$$

The corresponding MPC control allocation problem is posed as follows: For the constrained system

$$\begin{aligned}\dot{\boldsymbol{\delta}}(t) &= A_\delta\boldsymbol{\delta}(t) + B_\delta\boldsymbol{\delta}_{cmd}(t)\\ \boldsymbol{\tau}(t) &= B\boldsymbol{\delta}(t)\\ \delta_{min} \le \boldsymbol{\delta} &\le \delta_{max}\end{aligned} \tag{5}$$

find $\boldsymbol{\delta}_{cmd}(t)$ such that $\boldsymbol{\tau}(t)$ tracks $\boldsymbol{\tau}^*(t)$ as closely as possible, where $\boldsymbol{\tau}^*(t)$ is the virtual control input vector, B is the control efficiency matrix and δ_{min}, δ_{max} are the upper and lower saturation limits of the effectors or actuators, respectively.

The system (5) is used to predict the commanded control inputs $\boldsymbol{\delta}_{cmd}$, the control commands $\boldsymbol{\delta}$ and outputs y throughout the prediction horizon,

$$\hat{\boldsymbol{\delta}}_{cmd} = [\ \hat{\boldsymbol{\delta}}_{cmd}(k|k),\ \cdots,\ \hat{\boldsymbol{\delta}}_{cmd}(k+N-1|k)\ ] \tag{6}$$

$$\hat{\boldsymbol{\delta}} = [\ \hat{\boldsymbol{\delta}}(k+1|k),\ \cdots,\ \hat{\boldsymbol{\delta}}(k+N|k)\ ] \tag{7}$$

$$\hat{\boldsymbol{\tau}} = [\ \hat{\boldsymbol{\tau}}(k+1|k),\ \cdots,\ \hat{\boldsymbol{\tau}}(k+N|k)\ ] \tag{8}$$

where N is the length of the prediction horizon, and k is the current time step. The MPC algorithm finds the optimal set of $\hat{\boldsymbol{\delta}}_{cmd}$ by minimizing a cost function on the form

$$\begin{aligned}J(\cdot) = &\sum_{j=1}^{N} W(j)[\ \hat{\boldsymbol{\tau}}(k+j\ |k) - \boldsymbol{\tau}^*(k+j)\ ]^2\\ &+\alpha\sum_{j=1}^{N-1}\sum_{i=1}^{K} W_a(i)[\ \boldsymbol{\delta}_{cmd,i}(k+j-1|k)\ ]^2\end{aligned} \tag{9}$$

subject to (5).

In the cost function, W is a weight matrix weighing the importance of tracking $\boldsymbol{\tau}^*$ at time j. W_a weighs the relative cost of use of effector $i \in \{1 \ldots K\}$. As before, K is the number of control actuators. $\alpha > 0$ weighs the relative importance between the tracking term and the effector penalty term, and is usually small. Only the first commanded control sample $\hat{\boldsymbol{\delta}}_{cmd}(k|k)$ is applied to the actuator. The whole algorithm is repeated when computing the consequtive $\hat{\boldsymbol{\delta}}_{cmd}(k+1|k+1)$.

B. CVXGEN Solver

The CVXGEN solver is currently available through a web interface `http://www.cvxgen.com`. An optimization problem specification can be entered through a MATLAB-like programming language. Syntax specifics can be found in CVXGEN's user manual [12]. The problem is entered in a fixed problem structure, specifying the problem's dimensions, parameters, variables, cost function and constraints.

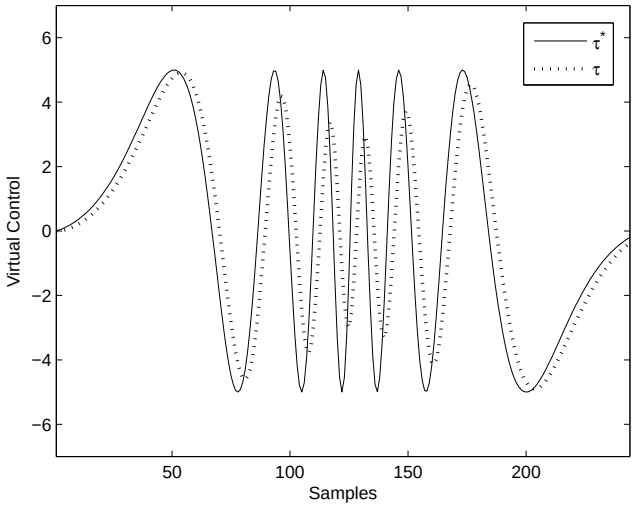

Fig. 1. QP CA Virtual Input Tracking

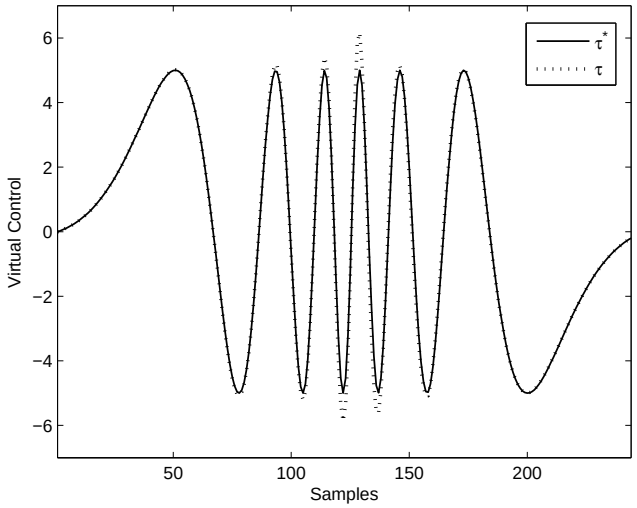

Fig. 2. MPCA Virtual Input Tracking

The library-free custom C solver is automatically generated. In addition to C code, a MATLAB interface is also available, making the custom solver available for e.g. prototyping and initial testing within the MATLAB environment.

The solver is used by calling a pre-made function, with the problem instance's specific parameters as function input. Solver settings can also be entered when calling the solver. After the call, the solver solves the convex optimization problem with respect to the instance parameters, and outputs the globally optimal solution.

CVXGEN lends itself naturally to MPC problems, see [11] for a detailed overview.

III. CASE STUDIES

The examples will illustrate performance tradeoffs between control performance, accuracy and cost of actuation (power, wear,...) that can be systematically adressed with dynamic predictive control allocation. Furtermore, computational performance characteristics of the CVXGEN implementation are reported.

A. Simple test - actuators/effectors with different cost and dynamic response

First, a simple test is conducted, comparing the performance of similar MPCA and QP formulations. The virtual control command τ^* is one-dimensional, consisting of a sine with increasing and then decreasing frequency. There are two actuators δ_1 and δ_2, with associated effectors, both modeled as second order systems. Actuator 1 will be fast but expensive to use, while actuator 2 will be slow and inexpensive. The actuator coefficients and corresponding cost weight are

$$\omega_{0,1} = 150, \quad \zeta_1 = 0.7, \quad W_1 = 1$$
$$\omega_{0,2} = 10, \quad \zeta_2 = 0.9, \quad W_2 = 0.1$$

This means that the control allocation module should use actuator/effector 1 only when necessary. In addition, effector 2 will be more efficient than effector 1, reflected in the control efficiency matrix $B = [\ 0.3\ 0.8\ ]$. The virtual input prediction for the MPCA is done using a second order extrapolation based on the current and most recent samples.

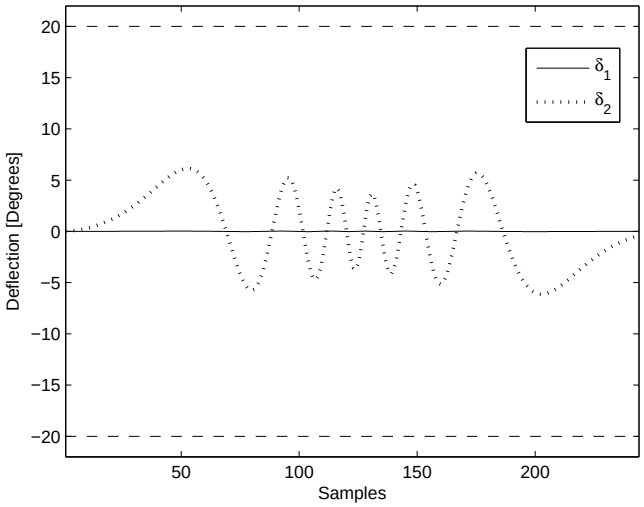

Fig. 3. QP CA Actuator Response

The virtual input tracking of the QP and MPCA methods can be seen in Figures 1 and 2 respectively. It is clear that the MPCA does a far better job than the similar QP formulation when it comes to tracking τ^*. This is because the QP CA ignores the actuator dynamics, leading to it commanding mostly the slow actuator δ_2 to deflect to track τ^*. As the frequency of the virtual input increases, actuator 2 can not follow, causing a larger tracking error. MPCA is aware of the actuator dynamics and optimally combines both actuators to meet the requirement of the virtual input. The actuator response δ_1 and δ_2 for the QP and MPCA methods can be seen in Figures 3 and 4, respectively. In these plots the actuator saturation limits are shown as dashed lines.

A comparison of the cost is shown in Figure 5, which summarizes the two methods' performance.

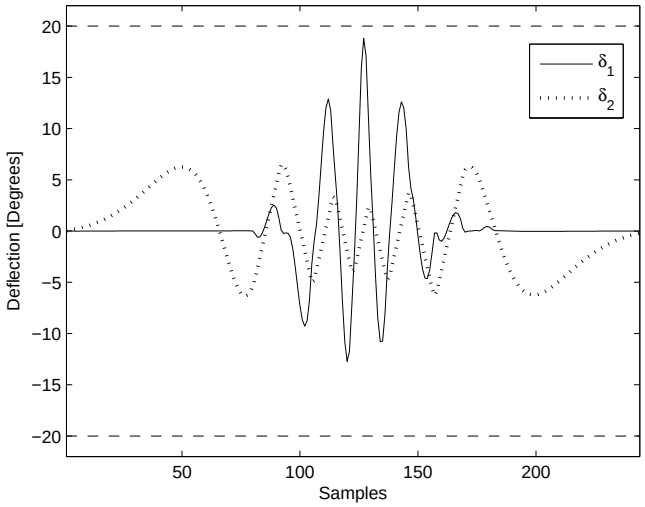

Fig. 4. MPCA Actuator Response

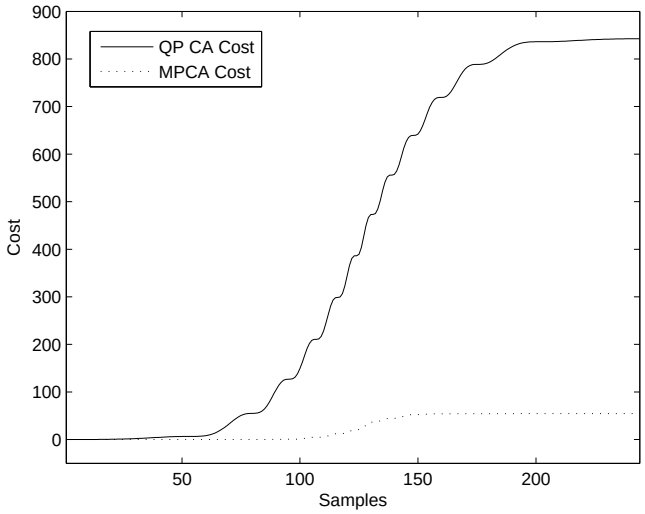

Fig. 5. Cumulative cost for QP CA and MPCA.

B. Missile auto-pilot

MPCA is also tested in a more realistic setup, as a part of a missile flight control system. The missile dynamics are approximated using decoupled longitudal and lateral models [22]. Such models are valid for small angles, but this is assumed to be sufficient for testing control allocation.

The models are on the form

$$\dot{x}_{long} = A_{long}x_{long} + B_{long}u_{long} \quad (10)$$
$$\dot{x}_{lat} = A_{lat}x_{lat} + B_{lat}u_{lat} \quad (11)$$

where

$$x_{long} = [\ \alpha\ q\ \theta\ ]^T, \quad u_{long} = \delta_P$$
$$x_{lat} = [\ \beta\ p\ r\ \phi\ \psi\ ]^T, \quad u_{lat} = [\delta_R\ \delta_Y\ ]^T$$

Subscripts denote longitudal and lateral models, and symbols are summarized in Table I.

The simulated missile has a mass of 200kg, flying at constant speed 300m/s, and has an inertia matrix

$$I = \begin{bmatrix} 5 & 1 & 1 \\ 0 & 150 & 0 \\ 0 & 0 & 150 \end{bmatrix} kgm^2$$

p	Roll Rate	α	Angle of Attack
q	Pitch Rate	β	Sideslip
r	Yaw Rate	δ_R	Roll Control Moment
θ	Pitch Angle	δ_P	Pitch Control Moment
ϕ	Roll Angle	δ_Y	Yaw Control Moment
ψ	Yaw Angle		

TABLE I
SYMBOLS IN MISSILE MODEL

The missile is tail controlled, where it has four fins placed in an x-configuration. The fins are controlled by four actuators $\delta_{1\ldots4}$, modeled as second order systems (1). The actuator characteristics and cost are summarized below.

$$\omega_{0,1} = 10, \quad \zeta_1 = 0.9, \quad W_1 = 0.1$$
$$\omega_{0,2} = 10, \quad \zeta_2 = 0.9, \quad W_2 = 0.1$$
$$\omega_{0,3} = 150, \quad \zeta_3 = 0.7, \quad W_3 = 1$$
$$\omega_{0,4} = 150, \quad \zeta_4 = 0.7, \quad W_4 = 1$$

The four missile fins are placed in pairs with one slow and one fast on each side. The slow and inexpensive pair are thought to be used while in trimmed flight, while the fast, expensive pair will be added on in agile flight.

The control allocation is part of a flight control system together with a bank-to-turn autopilot, designed to follow lateral and longitudal references. The autopilot design has two loops. The outer loop is controlling z and y position, while beeing fed back missile lateral and longitudal accelerations. This loop uses a bank-to-turn design to command the inner angular control loop. All controllers within these loops are PI- or P-controllers. The autopilot's virtual control output $\tau^* = [\ \delta_R^*\ \delta_P^*\ \delta_Y^*\ ]^T$ is the input to the control allocation module, which computes a commanded control $\delta_{cmd,i},\ i \in \{1,2,3,4\}$, which is applied to the actuators. The actual actuator response $\delta_i,\ i \in \{1,2,3,4\}$ is mapped with the control efficiency matrix B to form the control vector $\tau = [\ \delta_R\ \delta_P\ \delta_Y\ ]^T$. These are in turn inputs to the missile model.

An MPCA formulation like the one described in II-A is used, and a QP control allocation problem is used as a comparison. The prediction for the MPCA is created by holding the current value througout the prediction horizon, which spans five samples.

The lateral and longitudal references are steps of 50 m, and the missile step responses are seen in Figure 6. The responses for MPCA and QP CA are relatively similar, though the MPCA performs notably better in the longitudal step case.

Looking at the virtual input tracking, the MPCA and QP CA cases are shown in Figures 7 and 8, respectively. Also here it is clear that the MPCA, being aware of the actuator dynamics, provides significantly better tracking of τ^* than the QP CA case. The actuator response for MPCA and QP

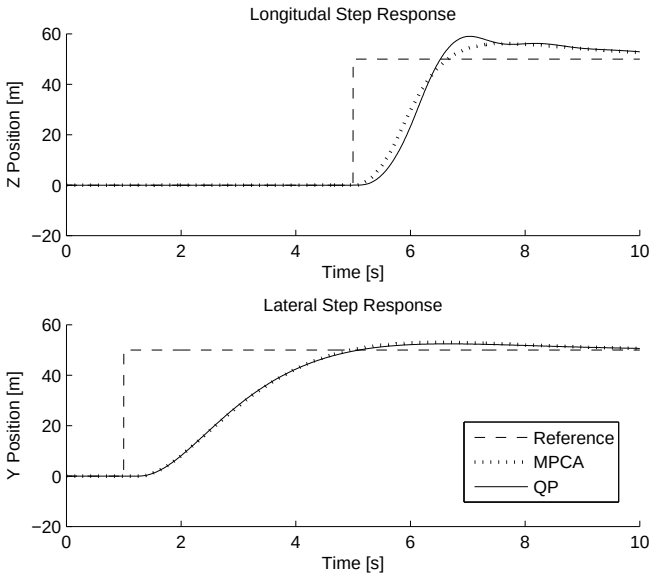

Fig. 6. Missile Step Response

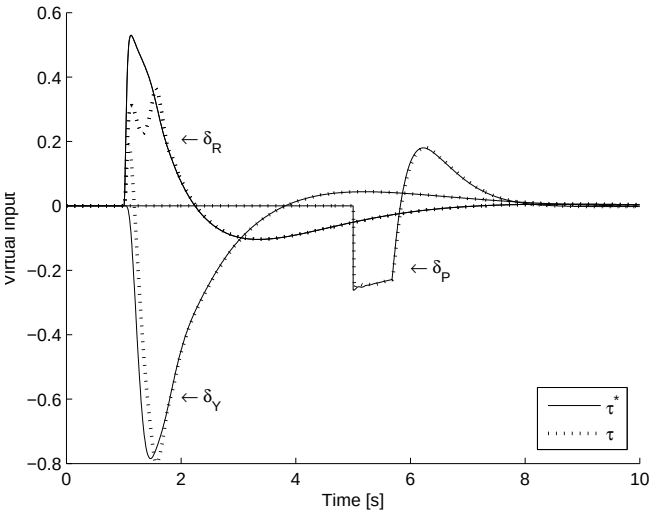

Fig. 7. MPCA Virtual Input Tracking

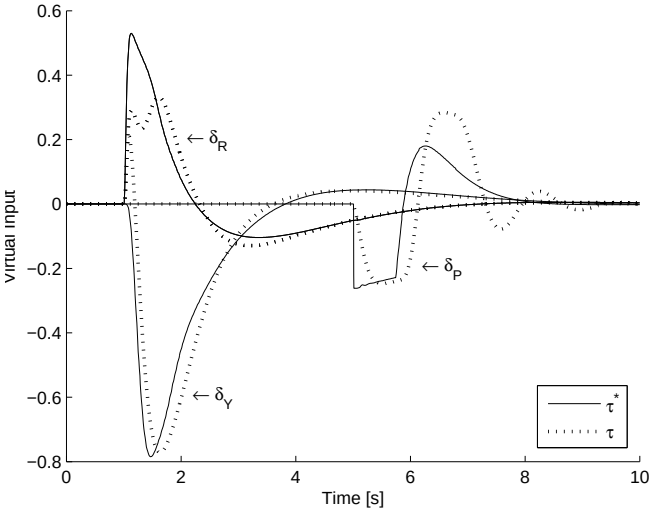

Fig. 8. QP Virtual Input Tracking

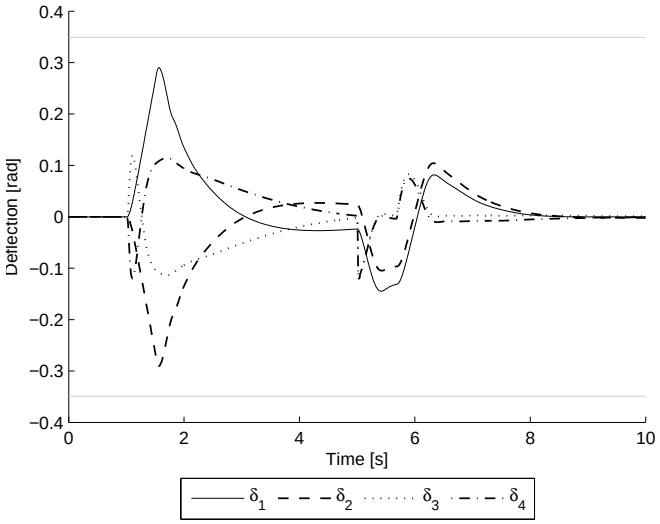

Fig. 9. MPCA Actuator Response

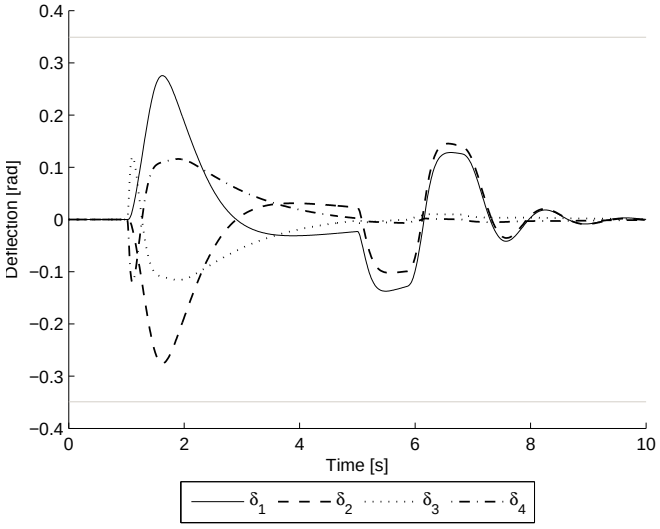

Fig. 10. QP Actuator Response

is shown in Figures 9 and 10. The MPCA clearly utilizes the fast actuators δ_3 and δ_4 more actively, leading to the previously mentioned improved virtual input tracking.

Lastly, the cost is compared. The cumulative cost is shown in Figure 11. As expected, the MPCA cost is well below that of QP CA.

CVXGEN is used during this simulation, and it is interesting to review the time consumption of the solver. During the 10-second simulation, 1045 calls are made to the model predictive control allocation function calculating the commanded control input δ_{cmd}. By isolating MATLAB on one CPU and using the program's *profiler* utility, it is found that these calls took a total of 0.434 seconds (CPU time), making each call on average consume 0.00041531 seconds CPU time. This is considered to be very fast, taking the large problem size into account. The MPCA problem size statistics are summarized in Table II.

Parameter entries	155
Original variables	78
Variables in solver	114
Equalities in solver	94
Inequalities in solver	80

TABLE II

MPCA OPTIMIZATION PROBLEM SIZE STATISTICS FOR MISSILE EXAMPLE

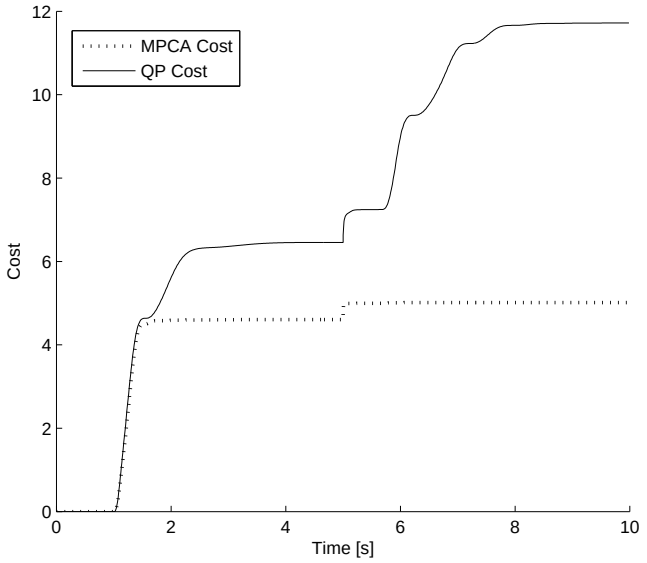

Fig. 11. Cumulative Cost, Step Response

IV. CONCLUSIONS

It is shown that the dynamic constrained allocation problem for typical configurations can be solved efficiently using MPC and CVXGEN.

The MPC formulation leads to improved overall control performance compared to a more conventional static control allocation method, and it is able to exploit an actuator configuration with different dynamic properties. Although the impact of actuator saturations are not illustrated very clearly in the examples, additional tests have shown that the MPCA and QP CA tend to degrade their performance in a similar manner when these constraints are activated.

The use of CVXGEN leads to a customized quadratic programming solver that typically require less than 1 millisecond computation time on a powerful processor. This may considered computationally feasible for implementation in a flight control system, although important aspects such as software code verifiability needs to be addressed carefully.

REFERENCES

[1] M. Bodson, Evaluation of optimization methods for control allocation, *J. Guidance Control and Dynamics*, Vol. 25, 2002

[2] W. C. Durham, Constrained control allocation, *J. Guidance, Control, and Dynamics*, Vol. 16, pp. 717-725, 1993

[3] O. Härkegård, Dynamic control allocation using constrained quadratic programming *Journal of Guidance, Control, and Dynamics*, Vol. 27, pp. 1028-1034, 2004.

[4] W. Jackson, M. P. J. Fromherz, D. K. Biegelsen, J. Reisch, and D. Goldberg, Constrained optimization based control of real time largescale systems: Airjet object movement system, in Proc. IEEE Conf. Decision and Control, Orlando, 2001.

[5] T. A. Johansen, T. I. Fossen, Svein P. Berge, Constrained Nonlinear Control Allocation with Singularity Avoidance using Sequential Quadratic Programming, *IEEE Trans. Control Systems Technology*, Vol. 12, pp. 211-216, 2004

[6] T. A. Johansen, Optimizing nonlinear control allocation, IEEE Conf. Decision and Control, Nassau, Bahamas, pp. 3435-3440, 2004

[7] T. A. Johansen, T. I. Fossen, P. Tøndel, Efficient Optimal Constrained Control Allocation via Multi-Parametric Programming, *J. Guidance, Control and Dynamics*, Vol. 28, pp. 506-515, 2005

[8] T. A. Johansen, T. P. Fuglseth, P. Tøndel, T. I. Fossen, Optimal constrained control allocation in marine surface vessels with rudders, *Control Engineering Practise*, Vol. 16, pp. 457-464, 2008

[9] Y. Luo, A. Serrani, S. Yurkovich, D. B. Doman, M. W. Oppenheimer, Model predictive dynamic control allocation with actuator dynamics, American Control Conference, Boston, pp. 1695-1700, 2004

[10] Y. Luo, A. Serrani, S. Yurkovich, D. B. Doman, M. W. Oppenheimer, Dynamic control allocation with asymptotic tracking of time-varying control input commands, American Control Conference, Portland, pp. 2098-2103, 2005

[11] J. Mattingley, Y. Wang and S. Boyd, Code Generation for Receding Horizon Control, Proceedings IEEE Multi-Conference on Systems and Control, pages 985992, Yokohama, Japan, September 2010

[12] J. Mattingley and S. Boyd, CVXGEN: A Code Generator for Embedded Convex Optimization, CVXGEN Website http://www.cvxgen.com, 2010

[13] J. Mattingley and S. Boyd. Automatic Code Generation for Real-Time Convex Optimization, chapter in Convex Optimization in Signal Processing and Communications, Y. Eldar and D. Palomar, Eds., Cambridge University Press, 2009

[14] M. W. Oppenheimer and D. B. Doman, Control allocation for overactuated systems, Proc. Mediterranean Conf. Control and Automation, 2006

[15] J. A. M. Petersen and M. Bodson, Constrained Quadratic Programming tecniques for Control Allocation, Proc. IEEE Conf. Decision and Control, Maui, Hawaii, 2003

[16] E. Ruth and A. J. Sørensen, A solution to the Nonconvex Linearly Constrained Quadrating Thrust Allocation Problem. In Proceedings of 8th Conference on Manoeuvring and Control of Marine Craft (MCMC2009), pp. 195-200, Guaruj, Brazil, 2009

[17] J. Spjøtvold, T. A. Johansen, Fault Tolerant Control Allocation for a Thruster-Controlled Floating Platform using Parametric Programming, IEEE Conf. Decision and Control, Shanghai, 2009

[18] J. Tjønnås and T. A. Johansen, Optimizing Adaptive Control Allocation with Actuator Dynamics, Proc. IEEE Conf. Decision and Control, New Orleans, 2007

[19] P. Tøndel, and T. A. Johansen, Control allocation for Yaw Stabilization in Automotive Vehicles using Multiparametric Nonlinear Programming, American Control Conference, Portland, 2005

[20] C. Vermillion, J. Sun and K. Butts, Model predictive control allocation for overactuated systems - stability and performance, IEEE Conf. Decision and Control, New Orleans, pp. 1251 - 1256, 2007.

[21] K. A. Bordignon, Constrained Control Allocation for Systems with Redundant Control Effectors, Virginia Polytechnic Institute and State University, 1996.

[22] T. I. Fossen, Mathematical Models for Control of Aircraft and Satellites, Technical Report, Department of Engineering Cynernetics, Norwegian University of Science and Technology, 2011.