Autonomous Reconfigurable Control Allocation (ARCA) for Reusable Launch Vehicles*

A. S. Hodel[†] Ronnie Callahan[‡]

Revision, January 7, 2002

1 Introduction

The role of control allocation (CA) in modern aerospace vehicles is to compute a command vector $\delta_c \in \mathbb{R}^{n_a}$ that corresponding to commanded or desired body-frame torques (moments) $\tau_c = \begin{bmatrix} L & M & N \end{bmatrix}^T$ to the vehicle, compensating for and/or responding to inaccuracies in off-line nominal control allocation calculations, actuator failures and/or degradations (reduced effectivness), or actuator limitations (rate/position saturation). The command vector δ_c may govern the behavior of, e.g., aerosurfaces, reaction thrusters, engine gimbals and/or thrust vectoring. Typically, the individual moments generated in response to each of the n_a commands does not lie strictly in the roll, pitch, or yaw axes, and so a common practice is to *group* or *gang* actuators so that a one-to-one mapping from torque commands τ_c to actuator commands δ_c may be achieved in an off-line computed CA function.

We shall assume the existence of an off-line computed nominal linear affine CA function

$$\delta_c = F(x)\tau_c + \delta_0(x) \tag{1.1}$$

where τ_c is the commanded torque vector, x is a vehicle state vector, δ_0 is a trim (neutral torque) vector and $F(x)$ is a matrix of nominal control allocation gains. One may interpret the columns of $F(x)$ as a set set of gains defining "ganged" actuators for each control axis. The resulting nominal autopilot/control allocation block diagram is shown in Figure 1. The vector τ_b in Figure 1 refers to the body torques induced on the vehicle by the actuators. Ideally, the control allocation matrix $F(x)$ would be chosen to be the pseudo-inverse $G(x)^\dagger$ of the Jacobian matrix

$$G(x) = \left[\frac{\partial \tau_{b,i}}{\partial \delta_j} \right]\Big|_x \in \mathbb{R}^{3 \times n_a}$$

where n_a is the number of actuators; that is, we wish to design the control allocation matrix $F(x)$

[*] This work was supported in part by NASA contract NAS8-01105 and in part by a NASA Graduate Student Research Fellowship.

[†] a.s.hodel@eng.auburn.edu, Dept. Elect. & Comp. Eng., 200 Broun Hall, Auburn University , Auburn, AL 36849-5201, corresponding author

[‡] a.s.hodel@eng.auburn.edu

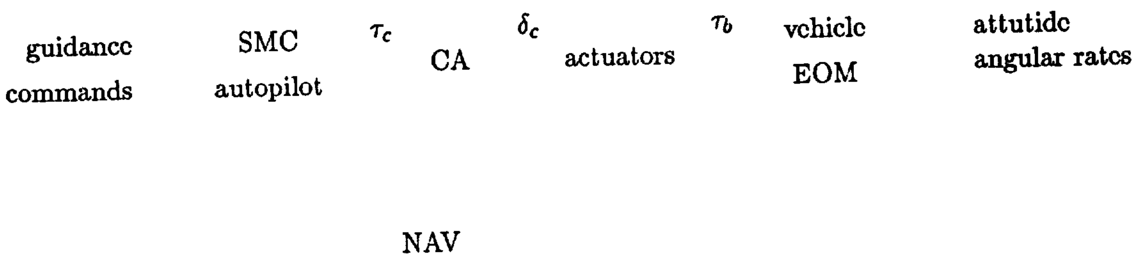

Figure 1: Nominal autopilot/control allocation block diagram.

such that

$$G(x)F(x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1.2}$$

so that the induced body-frame vehicle torques τ_b match the commanded body torques τ_c. However, due to sensor inaccuracy, modeling errors, and data compression in the allocation function, the ideal condition (1.2) cannot be achieved. As such, semidefinite programming techniques [HSZ01], [BEFB94] are used to design control allocation matrices $F(x)$ that achieve

$$\|G(x)F(x) - I\| < 1 - \gamma(x)$$

for some positive constant γ to achieve the "best possible" allocation given off-line data.

The linear affine nominal control allocation law (1.1) by itself is inadequate for the control control allocation problem for four reasons:

1. It fails to respond to torque allocation errors that can be detected on-line.
2. It fails to take into account saturation issues.
3. It fails to respond to on-line detected failures in actuators, and
4. It fails to provide a framework to work with discrete-valued (on-off) actuators such as reaction thrusters.

Dynamic control allocation [HC02] may be used to compensate for torque allocation errors detected on-line. DCA treats torque allocation error as a unknown additive uncertainty in the system Jacobians

$$G_{true}(x) = G(x) + \Delta_G(x)$$

where Δ_G is an unknown gain that respects the condition

$$\|G_{true}(x)F(x) - I\| < 1$$

over all operating conditions. If this condition is not met, then the CA problem is greatly complicated, requiring the use of on-line system identification [CPM95] [CBP98], [CG86], [HJN91]. Discrete-valued actuators may be dealt with by either using pulse-width modulation to emulate continuous valued actuators or, where actuator structure will not permit the use of PWM, one may use these actuators as "back-up" to the continuous valued actuators as in [HSZ01]. We address in this paper items 2 and 3 above by autonomous reconfigurable control allocation (ARCA).

An optimal solution of a constrained control allocation problem involves the solution of convex programming problems [Buf97], [Dur93], [Enn98]. Some heuristic approximate allocation solutions are presented in [BD95] such as computation of the attainable moment subset, generalized inverses, and daisy chaining. Alternatively, adaptive control may be used in tandem with linear system theory techniques in an attempt to avoid and/or compensate for actuator saturation [CGD+98]. In this paper we build on the work of [BD95], [Dur93], [BP98], [BD95], and [PB00] to present an online autonomous reconfigurable control allocation technique that is computationally tractable and practical for use in closed loop with a robust autopilot (attitude control law) such as sliding mode control [SMJ+98], [BLM99], [SHJ00]. Our technique makes use of a fast quadratic programming iteration step so that the actuator command δ_c results in a vehicle body torque τ_b that tracks the commanded torques τ_c when they lie within the attainable moment set or else approximates the commanded torques in a least-squares sense when they are not in the attainable moment set (the underlying quadratic programming problem is infeasible).

2 Autonomous Reconfigurable Control Allocation (ARCA)

Reconfigurable control refers to the ability of a control allocation law to continue to maintain tracking of the moment command τ_c in the face of actuator failures/degradation. In the short term, actuator saturation is indistinguishable from actuator failure, since in both cases an additional constraint is entered into the control allocation law. However, in the case of an actuator failure, the constraint is permanent and thus requires coordinated treatment between the control allocation law, the attitude control law, and the guidance law.

2.1 Problem statement

We formally describe the ARCA problem as follows. We shall denote the sequence of autopilot torque commands as $\tau_c(k)$ and the sequence of CA generated actuator command vectors as $\delta_c(k)$. In order to accomodate actuator rate limits, we shall compute the actuator command vectors recursively, i.e.,

$$\delta_c(k) = \delta_c(k-1) + \delta_e(k)$$

where $\delta_e(k)$ is an update to the previous actuator command vector $\delta_c(k-1)$. We shall omit the dependence on the time index k where it is clear by context. We shall make use of the following definitions:

Definition 2.1 The *actuator status vector* $\delta_{stat}(t)$ has entries in the range of $[0, 1]$ where $\delta_{stat,i} = 0$ reflects complete failure (actuator i has no impact on vehicle body torques) and $\delta_{stat,i} = 1$ reflects nonminal operation of actuator i. □

Remark 2.1 We shall assume that the the actuator status vector δ_{stat} is made available to the CA module by, e.g., a vehicle health monitoring system or an on-line system identification module.

Definition 2.2 The *effective actuator Jacobian* $\bar{G}(x)$ is the system Jacobian

$$\bar{G}(x) \triangleq \left[\frac{\partial \tau_b}{\partial \delta} \right]$$

evaluated under current operating conditions, including effects of actuator degradation and failure. □

Remark 2.2 Notice that the effective actuator Jacobian $\bar{G}(x)$ is unknown prior to flight time. If the nominal system Jacobians $G(x) \approx F(x)^{\dagger}$ are available, then $\bar{G}(x) = G(x)\text{diag}(\delta_{stat})$.

Definition 2.3 Given a sampling interval T, the current actuator position δ and vectors of actuator maximum position values δ^{+}, minimum actuator position values δ^{-}, and maximum actuator rates $\dot{\delta}_{\max}$, the *next-step feasible set* $\Delta_f(\delta_c)$ is defined as

$$\Delta_f(\delta_c) \triangleq \left\{ \delta_e : \delta^{-} \leq \delta_c + \delta_e \leq \delta^{+} \text{ and } |\delta_e - \delta| \leq T\dot{\delta}_{\max} \right\}$$

Remark 2.3 The set $\Delta_f(\delta(kT))$ is the set of legal next-step actuator commands $\delta_c((k+1)T)$ given current actuator position values $\delta(k)$.

Denoting $\delta_c = \delta_c(k-1)$, $\delta_e(k)$, the reconfigurable control allocation problem can thus be expressed as the quadratic programming problem [Luc84]

$$\begin{array}{rl} & \min\,(\delta_c + \delta_e)^T W_1\,(\delta_c + \delta_e) + {\delta_e}^T W_2 \delta_e \\ \text{subject to} & \bar{G}(x)\delta_e = \Delta\tau \\ \text{and} & \delta_e \in \Delta_f(\delta_c) \end{array} \qquad (2.1)$$

where W_1 is a position penalty on the command vector $\delta_c(k)$, W_2 is a rate penalty on the change in command vector δ_e, and $\Delta\tau$ is a torque command update, usually (but not always) defined as $\Delta\tau(k) = \tau_c(k) - \tau_c(k-1)$. Exceptions to this rule are discussed below.

In the case where the quadratic programming problem (2.1) is feasible, then the minimization searches for an update δ_e that matches the commanded torque while reducing the magnitude (cost) of the control command. Conversely, if the ideal torque command is *not* feasible, i.e., no solution $\delta_e \in \Delta_f(\delta_c)$ exists to the equality constraint $\bar{G}(x)\delta_e = \Delta\tau$, then it is necessary to relax the equality constraint and instead find a vertex (extreme point [Luc84]) of the feasible set $\Delta_f(\Delta_c)$ that minimizes the norm $\|\tau_{err}(k)\|$ where $\tau_e \triangleq \bar{G}(x)\delta_e - \Delta\tau$. In this case, the next value of $\Delta\tau$ is selected to reflect both the update to the torque command τ_c and the "unallocated" torque τ_e from the previous iteration.

Brute force application of standard quadratic programming techniques may not be desirable in the ARCA problem for the following reasons.

1. Computation time: solution of the quadratic programming problem can requires several iterations, each requiring solution of a linear system of equations $Ax = b$ of dimension up to $n_a \times n_a$.

2. Repetition: the quadratic programming problem to be solved in our application is solved repeatedly for problems and conditions that do not vary greatly from one sample time to the next. For example, during times of highly aggressive maneuvers, it is likely that the same actuators are saturated from one time step to the next. The similarity of these problems is not exploited by standard quadratic programming techniques.

3. Singular dual solution: because the number of constraints is always 4 times the number of unknowns, the dual problem [Luc69] used to solve for the corresponding Lagrange multiplier in a non-negative least squares problem is singular, which requires at the least an increase in the dimension of the problem solution via the use of "slack" variables.

2.2 Fast QP solution to ARCA

Given the concerns raised above, we propose the use of a fast QP solution method that differs from standard primal problem QP solvers in two ways:

1. The solution (active constraint set [Luc84]) of the previous problem is used as a starting point of the current problem.
2. The number of QP iterations is limited so that computational burden is reduced.

We consider here the equivalent QP problem using the notation of [Luc84].

$$\begin{array}{rl} & \min_x J(x) \triangleq x^TQx + c^Tx \\ \text{subject to} & Ax = b \\ & x^- \leq x \leq x^+ \end{array} \tag{2.2}$$

The unknowns x in problem (2.2) correspond to the actuator command deviations δ_e in the ARCA algorithm description. We shall proceed on the following assumptions:

1. $x^- \leq 0 \leq x^+$; i.e., the previous actuator command vector satisfies actuator constraint limits. This assumption may be violated in the case of, e.g., initial transients due to engine failure, where additional constraints on differential thrust can be imposed upon failure. Nevertheless, the above assumption will be reasonable in all cases where additional constraints are not suddenly imposed on actuator command behavior.
2. The torque command issued by the autocommander is feasible. This condition is can be met through the use of on-line computation of a local attainable moment set [?].

Based on the above assumptions, we may solve the QP problem (2.2) as illustrated in Figure 2. The initial point $x(0)$ by assumption satisfies the inequality constraints $x^- \leq x(0) \leq x^+$, but may not satisfy the equality constraint $Ax = b$ due to variations in the (effective) Jacobian and torque command from one time step to the next. The point $w(0)$ is computed that minimizes the optimization

$$\begin{array}{rl} \min\limits_{w(0)} & w(0)^TQw(0) + c^Tw(0) \\ \text{subject to} & \bar{A}w(0) = \bar{b} \end{array} \tag{2.3}$$

with $\bar{A} = A$, $\bar{b} = b$. As illustrated in the figure, $w(0)$ will satisfy the equality constraints, but may not satisfy the inequality constraints. We therefore choose our next solution value $x(1) = (1-\alpha)x(0) + \alpha w(0)$ where $w(0)$ is chosen so that $x(1)$ lies at the edge of the feasible set for the inequality constraints. We then append a row to $\bar{A}$ and $\bar{b}$ corresponding to the new "active constraint," $x_1 = x_1^-$ in the case of the figure, and solve the minimization (2.3) again, now constrained to the intersection of the hyperplanes $Ax = b$ and $x_1 = x_1^-$. This process continues until the iterate $w(k)$ is in the feasible set of problem (2.2). (Such a solution exists by assumption.)

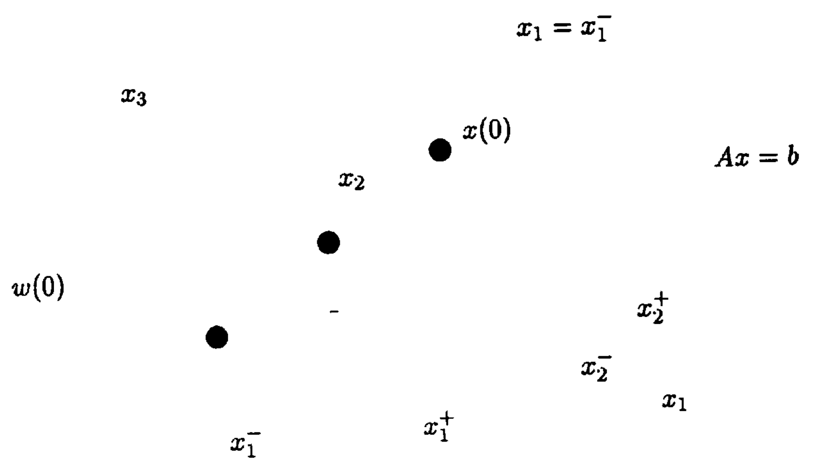

Figure 2: Quadratic programming problem (2.2) solution method for the ARCA problem. The feasible solution set is denoted by the rectangle on the interior of the hyperplane $Ax = b$.

Remark 2.4 Because the system Jacobians are continuous (except in the case of sudden failure), the active constraints from one iteration to the next will likely not change often. As a result, the above algorithm may be easily modified to solve the minimization (2.3) at most once per control step while evaluating revisions to the active constraint set by projecting the gradient

$$\frac{\partial J}{\partial x} = 2Qx + c$$

onto the null space of A

$$v = \left(I - A^{\dagger}A\right)(2Qx + c)$$

and comparing the signs of the resulting gradient search direction v with the active constraints imposed on x from the previous iteration.

3 Conclusions

Our initial tests of our fast QP solution method on artificially generated problems are very encouraging. For the final version of this paper we shall present closed-loop results using the ARCA algorithm in closed loop with a high-fidelity model of the X-33 experimental launch vehicle.

References

[BD95] Kenneth A. Bordignon and Wayne C. Durham. Closed-form solutions to constrained control allocation problem. *Journal of Guidance, Control, and Dynamics*, 18(5):1000–1007, September-October 1995.

[BEFB94] Stephen Boyd, Laurent El Ghaoui, Eric Feron, and Venkataramanan Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. SIAM, 1994.

[BLM99] Jovan D. Boskovic, Sai-Ming Li, and Raman K. Mehra. Globally stable adaptive tracking control design for spacecraft under input saturation. In *Conference on Decision and Control*, pages 1952–1957, December 1999.

[BP98] Marc Bodson and William A. Pohlchuck. Command limiting in reconfigurable flight control. *Journal of Guidance, Control, and Dynamics*, 21(4), 1998.

[Buf97] James M. Buffington. Tailless aircraft control allocation. In *Proceedings of the 1997 AIAA Guidance, Navigation, and Control Conference*, pages 737–747. AIAA, 1997.

[CBP98] P. Chandler, J. Buffington, and M. Pachter. Integration of on-line system identification and optimization-based control allocation. *1998 AIAA Guidance, Navigation, and Control Conference*, 1998.

[CG86] H. F. Chen and L. Guo. Convergence rate of least squares identification and adaptive control for stochastic systems. *International J. of Control*, 44:1459–1476, 1986.

[CGD+98] F. Z. Chaoui, F. Giri, L. Dugard, J. M. Dion, and M. M'saad. Adaptive tracking with saturating input and controller integral action. *IEEE Transactions on Automatic Control*, 43(11):1368–1643, November 1998.

[CPM95] P. R. Chandler, M. Pachter, and M. Mears. System identification for adaptive and reconfigurable control. *Journal of Guidance, Control, and Dynamics*, 18(3):516–524, May-June 1995.

[Dur93] Wayne C. Durham. Constrained control allocation. *Journal of Guidance, Control, and Dynamics*, 16(4):717–725, August 1993.

[Enn98] D. F. Enns. Control allocation approaches. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Boston, 1998. Paper No. 98-4108.

[HC02] A. S. Hodel and R. Callahan. Dynamic control allocation for reusable launch vehicles. Submitted to 2002 Southeastern Symposium on System Theory, Hunstville AL, March 2002.

[HJN91] A. J. Helmicki, C. A. Jacobson, and C. N. Nett. Control oriented system identification: A worst case/deterministic approach in ℓ_∞. *IEEE Transactions on Automatic Control*, 36(10):1163–1176, October 1991.

[HSZ01] A. Scottedward Hodel, Yuri Shtessel, and J. Jim Zhu. Robust inversion and data compression in control allocation for reusable launch vehicles. *J. Guidance, Control, and Dynamics*, 2001. submitted for publication.

REFERENCES

[Luc69] D. G. Luenberger. *Optimization by Vector Space Methods.* Wiley and Sons, Inc., New York, NY, 1969.

[Luc84] D. G. Luenberger. *Introduction to Linear and Nonlinear Programming.* Addison-Wesley, Reading, MA, 2nd edition, 1984.

[PB00] John A. M. Petersen and Marc Bodson. Control allocation for systems with coplanar controls. *AIAA Guidance, Navigation, and Control Conference and Exhibit*, A00-37182:1–11, August 2000.

[SHJ00] Y. Shtessel, C. E. Hall, and M. Jackson. Reusable launch vehicle control in multiple time scale sliding modes. In *Proceedings of AIAA Guidance, Navigation, and Control Conference*, number AIAA-2000-4155, Denver, Co, August 14-17 2000. AIAA.

[SMJ+98] Y. Shtessel, J. McDuffie, M. Jackson, C. Hall, D. Krupp, M. Gallaher, and N. Hentrix. Sliding mode control of the x33 vehicle in launch and re-entry modes. In *Proceedings of AIAA Guidance, Navigation, and Control Conference*, number 98-4414, pages 1352–1362, Boston, MA, August 10-12 1998. AIAA.