On-line Computation of a Local Attainable Moment Set for Reusable Launch Vehicles*

A. S. Hodel[†] Yuri B. Shtessel[‡]

Submitted for possible publication in the 2002 AIAA GN&C Conference

Extended Abstract

Problem description Traditional attitude control design of reusable launch vehicles involves independent design of autopilot and control allocation modules [SBB99], [SK97], [SHJ00], [PS00], [Hod00]. Unfortunately, this results in the potential for overly aggressive commands in the autopilot resulting in a loss of performance due to actuator saturation, particularly if the autopilot may suffer from integrator wind-up [HH01]. This unfortunate situation can arise from, e.g., actuator position limits that require that the actuator command vector $\delta_c(t) \in \Delta$ where Δ is the set of feasible actuator commands defined by

$$\Delta \triangleq \{\delta : \delta_- \leq \delta \leq \delta_+\}$$

where δ_- and δ_+ are minimum and maximum position commands, respectively. Control allocation is limited by the *attainable moment set* $\mathcal{T}$ [BD95], [Dur93], [Dur99], defined as

$$\mathcal{T} = \{\tau : \exists \delta \in \Delta \text{ and } G\delta = \tau\}$$

where G is the current Jacobian (control derivatives) matrix from the vehicle actuator condition vectort δ to the vehicle body torques τ_b

$$G \triangleq \frac{\partial \tau_b}{\partial \delta}$$

On-line calculation of the entire attainable moment set $\mathcal{T}$ is not a practical option for the following reasons:

1. Actuator models (aerodynamic) are approximate at best.
2. Actuator failure will significantly modify the attainable moment set $\mathcal{T}$.

We propose instead to calculate a point-wise "snapshot" of $\mathcal{T}$ as shown in Figure 1. Given a torque command τ_c we compute a local attainable moment set $\mathcal{T}(\tau_c)$ determinining the maximum and minimum torque limits in each channel (x =roll, y =pitch, z =yaw) while holding the other torque

*This work was supported in part by NASA contract NAS8-01105.

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

[‡]Department of Electrical and Computer Engineering, University of Alabama at Huntsville.

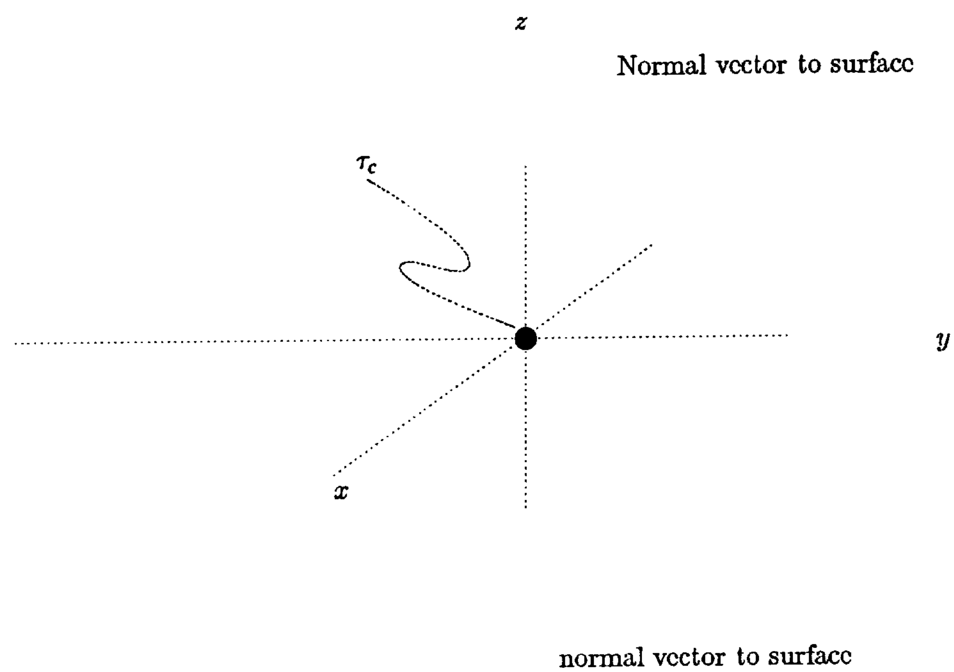

Figure 1: Attainable moment set snapshot

values constant. The dimensions of this local attainable moment set can be calculated by a linear programming problem, e.g., given the current system Jacobian matrix G,

$$\begin{aligned} \tau_{x,\max} &= \arg\max_{\delta} \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} G\delta \\ & \quad \text{subject to} \;\; \delta \in \Delta, \; \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} G\delta = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tau_c \end{aligned}$$

These torque limits may be of use in two scenarios:

1. in communicating overall actuator torque limits to the autopilot and autocommander so that autopilot and/or guidance commands may be appropriately adjusted, and
2. in flight scenarios where control allocation is required to divide torque commands a primary set of actuators (e.g, aerosurfaces) and a secondary backup set of actuators (e.g., reaction thrusters).

Torque limits computation and linear programming The computation of the local attainable moment set can be posed as a set of six linear programming problems

$$\begin{aligned} \max_{x} J(x) \quad & J(x) = c^T x \\ \text{subject to} \quad & Ax = b \\ & x^- \leq x \leq x^+ \end{aligned} \tag{0.1}$$

where c^T and A are construct from appropriate rows of the Jacobian matrix G and the vector b is the set of "pinned" torques from the current (feasible) torque command τ_c. (If τ_c is infeasible then torque limits max $J(x)$ can be computed by locating the vertex of the feasible set that minimizes the error $Ax = b$ [Luc84].) Standard codes [1] available for the linear programming problem such as `dsplp` and `lp_solve` are inappropriate for this problem, since

1. These codes are design for large sparse problems, where as our A matrix is dense and we have very few unknowns - at most 10 or 20.
2. These algorithms can require several iterations to converge to an optimal solution, where as we require fast operation (low computational overhead).
3. Our problem does not change too much from one time-step to the next (system Jacobians do not change much over one sampling period), so our method should make use of earlier solutions.
4. Further, our problem does not require an exact optimal solution - just a good approximation.

We therefore propose the use of a limited-iteration simplex method LP solution that uses results (active constraint set) of the previous iteration to compute initial values for the next iteration. Experience indicates that two simplex iterations is sufficient for adequate performance, and so our method requires the inversion of at most 24 2×2 matrices (2 for each limit, 2 limits per axis).

We show preliminary results of our method in Figure 2. The routine `lpsolve` is a full simplex LP solver that exits upon convergence or detection of an infeasible problem (notice time spike at $t \approx 13$sec). The routine `lpIter` is an implementation of our fast LP solver. Both `lpsolve` and `lpIter` have an outermost m-file script that is used to call a C-code implementation of our fast iteration. `dmTrqLim` is a full C-code translation of `lpIter` that is included for comparison. Observe that, except when the input torque command is infeasible, `lpIter` closely tracks that achievable torque limits even though it is limited to at most two simplex iterations per time step. This example was run with only 6 simulated actuators; similar results were obtained with much larger numbers of actuators in other simulation tests.

These torque limits can then be used to adjust autopilot control parameters on-line so that the autopilot can respond appropriately to either saturations or permanent failures. One potential method for using torque limits is presented explicitly in [SBB99]. We are investigating the use of this method in closed-loop control simulations of the X-33 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.

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

[1] search on www.netlib.org or on google for keywords "linear programming")

REFERENCES

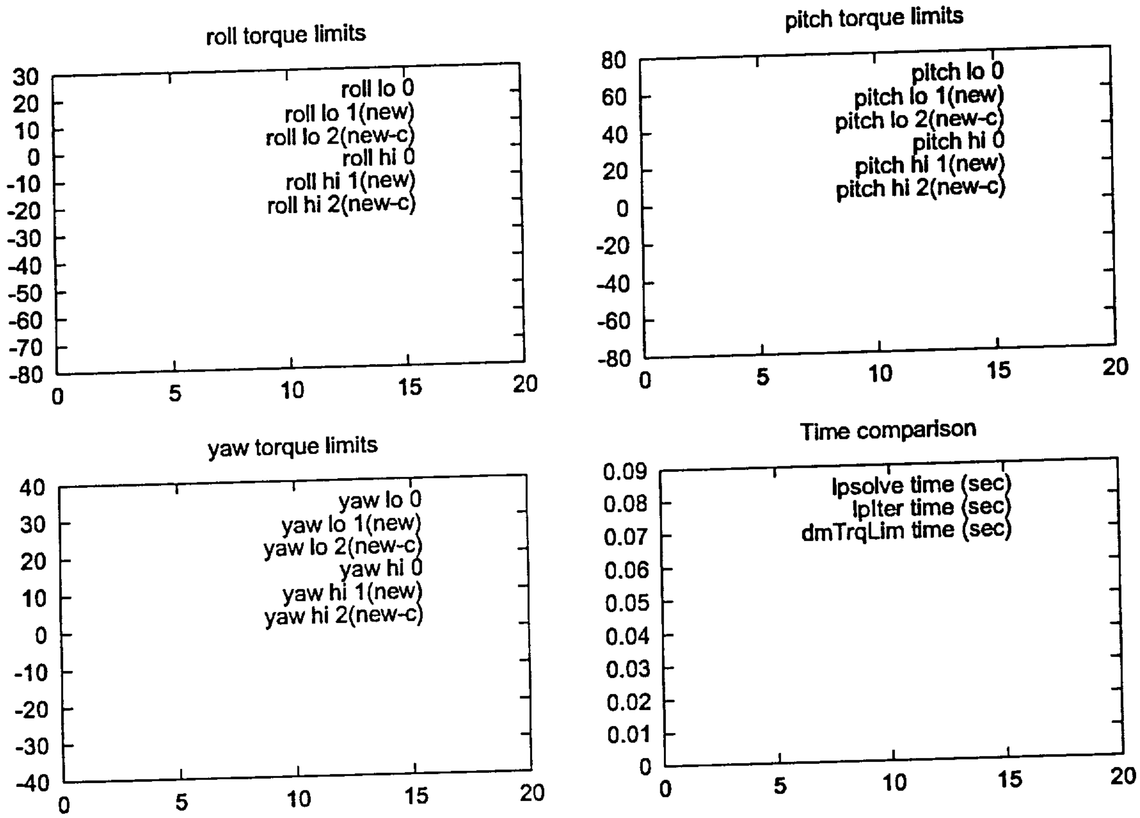

Figure 2: Preliminary test results for fast computation of the local attainable moment set. Plant Jacobians were made to vary sinusoidally. An artificial failure (zero effectiveness) was simulated at $t = 10$sec. Simulation experiments were performed with C-code integrated into Octave on a 450 MHz Macintosh running Yellow Dog Linux v 2.0.

[Dur99] Wayne C. Durham. Efficient, near-optimal control allocation. *Journal of Guidance*, 22(2):369–372, 1999.

[HH01] A. Scottedward Hodel and Charles E. Hall. Variable structure pid control to prevent integrator windup. *IEEE Trans. Indust. Elect.*, 48(2):442–451, April 2001.

[Hod00] A. S. Hodel. Robust inversion and data compression in control allocation. In *Proceedings of AIAA Guidance, Navigation, and Control Conference and Exhibit*, Denver, Colorado, August 14-17 2000. American Institute of Aeronautics and Astronautics. Paper AIAA-2000-4154.

[Lue84] D. G. Luenberger. *Introduction to Linear and Nonlinear Programming*. Addison-Wesley, Reading, MA, 2nd edition, 1984.

REFERENCES

[PS00] Anthony B. Page and Marc L. Stenberg. A closed-loop comparison of control allocation methods. In *Proceedings of AIAA Guidance, Navigation, and Control Conference and Exhibit*, Denver, Colorado, August 14-17 2000. American Institute of Aeronautics and Astronautics. Paper AIAA-2000-4538.

[SBB99] Y. Shtessel, J. Buffington, and S. Banda. Multiple time scale flight control using reconfigurable sliding modes. *AIAA Journal on Guidance, Control, and Dynamics*, 22(6):873–883, 1999.

[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.

[SK97] Yuri Shtessel and Don Krupp. Reusable launch vehicle trajectory control in sliding modes. In *Proceedings of the American Control Conference*, pages 2557–2561, Albuquerque, New Mexico, June 1997.