Published in *Proceedings of the IASTED International Conference on Intelligent Systems and Control,* Honolulu, HI, Aug 2004.

MCRLS FOR ON-LINE SPACECRAFT MASS- AND THRUSTER-PROPERTY IDENTIFICATION

Edward Wilson[a], David W. Sutter[a], Robert W. Mah[b]
[a] – Intellization – 454 Barkentine Ln, Redwood Shores, CA 94065, USA
[b] – NASA Ames Research Center – MS 269-1, Moffett Field, CA 94035, USA
Ed.Wilson@intellization.com, Dave.Sutter@intellization.com, Robert.W.Mah@nasa.gov

ABSTRACT

Spacecraft control, state estimation, and fault-detection-and-isolation systems are affected by unknown variations in the vehicle mass and thruster properties. It is often difficult to accurately measure inertia terms on the ground, and mass properties can change on-orbit as fuel is expended, the configuration changes, or payloads are added or removed. Multiple concurrent recursive least squares identification (MCRLS ID) algorithms using gyros and accelerometers to monitor vehicle motions are used here to identify on-line the vehicle inertia, inverse-inertia, center of mass, thruster force, and total mass. Originally developed for application to the X-38 v.201 spacecraft, the algorithms have been extended and implemented on the MIT SPHERES experimental spacecraft, which are now awaiting launch to the ISS for space-based testing. The MCRLS ID algorithm is summarized briefly, and advances in the specific ID algorithms are presented. An accurate and efficient filtering method for estimating angular acceleration from raw gyro signals is presented.

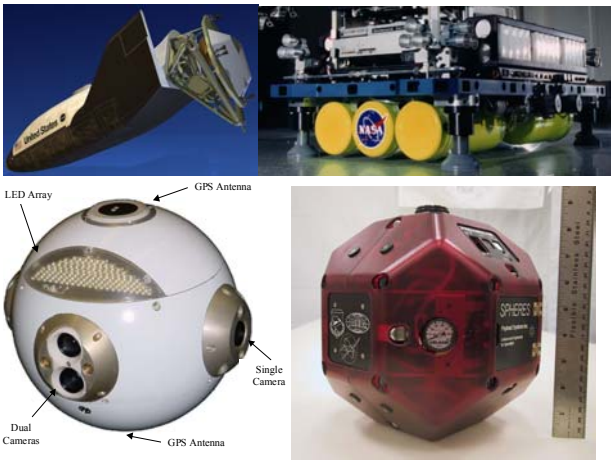
KEY WORDS
identification; spacecraft; mass; thruster; on-line; parameter estimation

1. Introduction

Due to the very small disturbance forces and torques present on spacecraft on orbit and in free space, accurate knowledge of their mass and thruster properties is important from a control and estimation standpoint. Fortunately, these features also make it feasible to identify these properties on-line using measurements of vehicle motions.

Spacecraft mass properties can be calibrated with limited accuracy during ground testing, and change further once on orbit due to expulsion of fuel mass, reconfiguration (of antennae, etc.), and for servicing robotic spacecraft, potentially variable payloads. An important difficulty encountered is that the unknown parameters do not appear linearly in the governing physical equations of motion. Accurate ID of mass properties has been studied extensively, but those methods have limitations as compared to the present research, and have not yet been implemented and tested on an actual spacecraft.

This paper will present recent advances in algorithms developed to achieve accurate on-line ID. These algorithms use and reference the MCRLS ID algorithm [1]. Initial mass-property ID algorithms as developed for the X-38 v.201 spacecraft and Mini-AERCam [2] are updated and extended here. Experimental results validating a subset of these algorithms on the MIT SPHERES spacecraft in zero-g aircraft flight tests are presented in [3]. For brevity, the reader is referred to those publications for supporting material. The above-mentioned spacecraft and the NASA Ames air-bearing simulator are shown below.



1.1 Related research

The use of linear least squares regression for the identification of unknown system parameters has been used and studied extensively, as by [4] and [5]. However, the requirement that a regression equation be formed with the unknown parameters linearly represented limits its direct applicability to many important problems, including the spacecraft application presented here.

Tanygin and Williams [6] and Bergmann [7] have separately developed advanced estimation algorithms for mass-property ID. The capabilities and limitations are discussed in [2]. Significant limitations that the present research effort has sought to overcome are the increased computational complexity, rigid assumptions on thruster and dynamic properties, and lack of flexibility in adding or removing properties from the list of identified parameters.

Wilson and Rock developed an ID method based on exponentially weighted RLS using accelerometer and angular rate sensors [8] [9]. Driven by the requirements of real-time on-line implementation as part of a reconfigurable fault-tolerant control system, the problem was reformulated to ID the accelerations resulting from thruster firings. By combining the mass and thruster properties, the regression function was linearized.

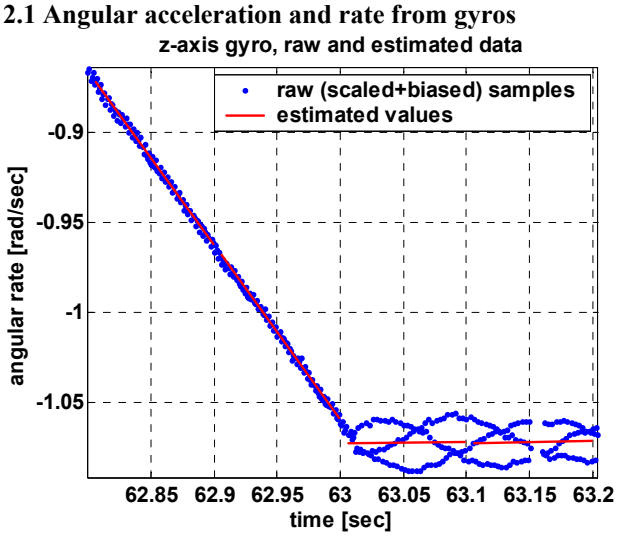
The MCRLS algorithm enables the application of linear recursive least squares (RLS) ID to certain nonlinear parameter estimation problems. Originally developed to address this spacecraft mass-property ID problem, it allows the nonlinear ID problem to be segmented into linear parts [1]. So the ID algorithms presented here (for inertia, thruster strength, etc.), may be considered without regard to the variability present in the other unknown parameters (an approximation).

2. Acceleration estimation

The ID algorithms operate by analyzing vehicle motions, during both coasting and thruster-firing periods. ID accuracy is dependent on the accuracy of the velocity and acceleration estimates. This section summarizes the algorithms developed to estimate angular rate, angular acceleration, and translational acceleration from the gyros and accelerometers (if available). These algorithms have been developed, tuned, implemented, and tested on the MIT SPHERES spacecraft, so they address the common issues of vibration disturbances and efficiency requirements for real-time code implementation.

Raw sensor (gyro or accelerometer) data is sampled at 1 kHz, so there are 100 samples per 100-ms control update period. Thrusters are held open or closed for the entire 100 ms, and ID processing is done based on each 100-ms cycle.

2.1 Angular acceleration and rate from gyros



This figure, using data from the SPHERES taken during flight tests aboard the NASA 0-g KC-135A in Nov. 2003, shows an example of the ringing induced whenever a thruster is opened or closed. In this case, the end of an 800-ms firing period occurs at 63.0 seconds. Prior to this, the signal is fairly clean, with a negative slope due to the applied torque. A latency of a few milliseconds follows 63.0 seconds, at which point the thrusters close for a 200 ms coasting period.

This 338 Hz ringing is close to the 500 Hz Nyquist frequency (a single-pole 300 Hz analog pre-filter is used, which does not do much for this ringing), with the near-aliasing resulting in the pattern shown. The ringing is handled very effectively, as seen by the relatively consistent acceleration estimates in the previous plot. The segments defined by the on-board-estimated rates and accelerations are overlaid on the raw gyro data to demonstrate the estimator's ability to filter out the gyro ringing while providing accurate and high bandwidth estimation.

The segment of raw gyro readings covering the control segment of interest (during which thrust should be constant) is low-pass filtered with a forward-backward zero-phase smoothing filter to remove high frequency structural vibrations and vibration-induced ringing of the gyro's tuning fork. A line is then fit using a linear least squares regression to the smoothed result, with the slope of the line providing the angular acceleration estimate, and the mean offset providing the angular rate estimate. Since it used only samples taken during the thruster firing cycle of interest, the estimate is thus properly correlated with the thruster firing that caused it. No reliance on thruster properties was made in the estimate.

While the above describes the algorithmic functionality, due to the requirements of the real-time processing system, it is desirable to implement this derivative filter using as little code memory, processor cycles, and software development time as possible. An approach was developed that takes advantage of the fact that both low-pass filtering and least-squares line fitting act as linear operations on the data, meaning that the combination of the two is also linear. This allows the two-step procedure to be implemented as a single vector multiplication of a vector of filter coefficients with the data segment. The vector of filter coefficients is determined through a process whereby each point, one at a time, is set to one (with the rest zero) and the two-step filter process output then equals the value of the corresponding coefficient. This implementation approach has the additional benefit of allowing filter parameters to be tuned off-line in MATLAB, and requiring no adjustment to the real-time code other than to load a different vector of MATLAB-generated filter coefficients.

2.2 Translational acceleration from accelerometers

Translational acceleration estimation is simpler, since the sensors measure acceleration directly (vs. requiring differentiation), but complicated by the fact that correction for angular motions is required. The first step performs a vector-multiply averaging operation (re-using the code from the angular motion estimation, with different filter coefficients), returning the translational acceleration at the accelerometers.

Since the accelerometers can not be located exactly at the point at which acceleration measurement is desired (e.g., the center of mass or the geometric center), centripetal and angular accelerations will cause the accelerometers to return measurements that are a superposition of three effects:

1. Translational acceleration of the reference point.
2. Centripetal acceleration proportional to the angular rate and the distance of each accelerometer's effective center of mass for omega, r_ω, from the reference point.
3. Tangential acceleration proportional to the angular acceleration and the distance of each accelerometer's effective center of mass for alpha, r_α, from the reference point.

If the accelerometer proof mass were a point mass, and the suspension (typically a pendulum) were mass-less, then r_ω and r_α would be equal to the proof mass location. Equation 1 shows the correction of the acceleration measured at the accelerometer, $\hat{a}_a$, to acceleration at point c, $\hat{a}_c$, based on angular acceleration, α, mean angular rate, ω_0, and the segment duration, τ. The final term is an exact correction for the fact that ω varies linearly over the segment, resulting in a known distortion in $\omega_0 \times (\omega_0 \times r_\omega)$ [1]. Equation 1 must be calculated for each of the 3 accelerometers since they are at different locations.

$$\hat{a}_c = \hat{a}_a - \alpha \times r_\alpha - \omega_0 \times (\omega_0 \times r_\omega) - \frac{\tau^2}{12} \alpha \times (\alpha \times r_\omega) \quad (1)$$

3. Spacecraft mass- and thruster-property ID

This section summarizes the algorithms used for on-line ID of center of mass, inverse inertia, and inertia, using rotational measurements only, as would commonly be available with gyros. Angular acceleration is required as a minimum, and the use of angular rate as well allows accounting for the gyroscopic term, $\omega \times (I\omega)$, in the governing equations. Thruster strength ID can be performed using rotational measurements alone, or with the additional use of translational acceleration measurements. By using MCRLS, all that is required for each set of parameters is to put it in regression form. The inertial measurements used (as estimated in Section 2) are ω, $\dot{\omega}$, and $\ddot{x}^{body}$.

3.1 Equations of motion, nomenclature

The rotational and translational equations of motion (EOM) for thruster controlled spacecraft can be written as:

$$\dot{\omega} = I^{-1} \begin{pmatrix} (L \times D) S_k (F_{nom} + F_{bias} + F_{random,k}) T_k \\ + \tau_{disturb} - \omega \times (I\omega) \end{pmatrix}; \quad (2)$$

$$\ddot{x}^{body} = m^{-1} \begin{pmatrix} D S_k \left(F_{nom} + F_{bias} + F_{random,k} \right) T_k \\ + f_{disturb}^{body} \end{pmatrix}; \quad (3)$$

where ω is a *3*-by-*1* vector containing the angular velocity of the body-fixed frame with respect to an inertial reference frame; I is a *3*-by-*3* matrix containing the spacecraft inertia tensor (or dyadic, or matrix), measured about the true center of mass; L is a *3*-by-n matrix containing *x-y-z* location of each thruster in the body frame; n is the number of thrusters; D is a *3*-by-n matrix containing unit vectors indicating the direction of thrust in the body frame; the $L \times D$ cross product is taken column-by-column; S_k is a scalar containing thruster magnitude scale factor applied to all thrusters and includes effects of blowdown and the reduction in thrust when multiple thrusters are fired; F_{nom} is a n-by-n diagonal matrix containing nominal strength of each thruster at full tank pressure; $F_{bias}, \hat{F}_{bias}, \tilde{F}_{bias}$ (true, estimated, and error variants given here) are N-by-N diagonal matrices containing constant off-nominal strength of each thruster at full tank pressure, $F_{bias} = \hat{F}_{bias} + \tilde{F}_{bias}$; $F_{random,k}$ is a N-by-N diagonal matrix containing pulse-to-pulse strength variations of each thruster; T_k is a n-by-*1* vector of 1's and 0's containing effective values for which thrusters fire at time step k, accounting for transient effects; $\tau_{disturb}$ is a *3*-by-*1* vector containing the sum of all torques on the vehicle resulting from other sources (drag, gravity gradient, separately modeled known thruster anomalies, CMG, RWA, and other calculable dynamic effects, *etc.*); $\ddot{x}^{body}$ is a *3*-by-*1* vector containing vehicle translational acceleration, measured in the body frame; m is a scalar containing total vehicle mass; and $f_{disturb}^{body}$ is a *3*-by-*1* vector containing the net force on the vehicle through the center of mass, with the *body* superscript indicating measurement in the body frame.

Equation 2 contains all of the parameters we would like to identify, including CM location (contained in L), inertia (and its inverse), and F_{bias}. Unfortunately, the parameters multiply one another, and cannot all be manipulated into the desired linear regression form, $Ax \cong b$. The CM location, inverse-inertia, inertia, mass, and thruster strength will be identified individually by concurrently running RLS IDs according to the MCRLS ID approach. Note that depending on the application, not all of these values will be uncertain enough or critical enough to warrant ID.

[1] This term becomes important for longer sample periods, but for the SPHERES, it is negligible and not calculated.

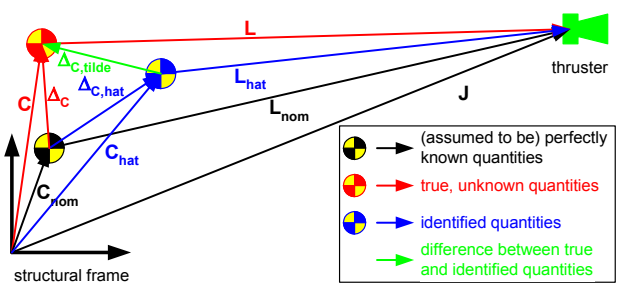
$\hat{I}$ and $\hat{I}^{-1}$ are symmetric, so instead of 9 free parameters in each, there are only 6 in each. The identifications are designed to directly identify these 6 parameters. $\hat{I}^{-1}$ will not exactly equal the inverse of $\hat{I}$.

3.2 MCRLS

Per the MCRLS algorithm, the IDs are initialized with the best prior parameter estimates (e.g., the nominal values). The respective estimate error covariance matrices are set according to the confidence in the initial nominal values. Then the RLS updates include weighting according to the sensor error covariance matrices. At each RLS update, a given ID will use the most recent estimates for other parameters being ID'ed (for example, CM ID would use the most recent estimates for inertia and thruster bias). The overall approach may be extended to other parameters in the governing equations (for example, the thruster directions) if they have sufficient variability and impact so as to justify ID.

3.3 ID of deviation from nominal

There is a component of the vehicle mass properties that can be calculated fairly well: the change in mass as fuel is depleted. The effect of this on the change in nominal mass properties is calculated and is referred to as burn-time-integration (BTI). For this reason, the mass ID is designed to ID the difference between true and nominal mass properties, for example, Δ_C rather than C. So the known part of the change is incorporated exactly and immediately, and the ID continues to ID the unknown deviation of the parameters from nominal. Inaccuracy in the BTI will be partially accounted for by the ID.



As shown in this figure, the center of mass, C, determines the origin of the body frame, and thereby determines the value of L, which contains the locations of each thruster in the body frame. Similarly, Δ_C, the difference between actual and nominal values of C determines L. Δ_C is the value that will be identified here. J is assumed to be known perfectly; C_{nom} is assumed to be known perfectly, including continual updates based on BTI; L_{nom} can then be calculated directly using these; C, and therefore L and Δ_C also, is the true quantity which cannot be known perfectly; $\hat{\Delta}_C$ is the identified quantity, leading directly to $\hat{C}$ and $\hat{L}$. Similarly for other mass properties:

$$\Delta_C \triangleq C - C_{nom}; \tag{4}$$

$$\Delta_I \triangleq I - I_{nom}; \tag{5}$$

$$\Delta_{I^{-1}} \triangleq I^{-1} - I_{nom}^{-1}; \tag{6}$$

$$\Delta_m \triangleq m - m_{nom}; \tag{7}$$

$$L = L_{nom} - \Delta_C \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}. \tag{8}$$

3.4 Variables common to multiple IDs

The rotational and translational EOM can be re-written after defining four 3-by-1 vectors to be used later, a_k, c_k, d_k, and e_k, where: a_k can be thought of as the nominal torque about the nominal CM; c_k is the nominal force on the vehicle due to thrusters; d_k is the torque resulting from the CM being off-nominal; and e_k is the accelerometer rotational correction term, correcting to the nominal CM. These four variables are defined as follows:

$$a_k \triangleq (L_{nom} \times D) S_k (F_{nom} + \hat{F}_{bias}) T_k + \hat{\tau}_{disturb}; \tag{9}$$

$$c_k \triangleq D S_k (F_{nom} + \hat{F}_{bias}) T_k; \tag{10}$$

$$d_k \triangleq a_k + c_k \times \hat{\Delta}_C - \hat{\omega} \times (\hat{I}\hat{\omega}); \tag{11}$$

$$e_k \triangleq \dot{\hat{\omega}} \times (r_\alpha - C_{nom}) - \hat{\omega} \times (\hat{\omega} \times (r_\omega - C_{nom})). \tag{12}$$

The rotational and translational EOM are given as:

$$I\dot{\hat{\omega}} + \hat{\omega} \times (I\hat{\omega}) = a_k + c_k \times \Delta_C; \tag{13}$$

$$\begin{aligned} &\dot{\hat{x}}_a^{body} - e_k + \dot{\hat{\omega}} \times \Delta_C + \hat{\omega} \times (\hat{\omega} \times \Delta_C) = \\ &m^{-1}\left(D S_k \left(F_{nom} + F_{bias} \right) T_k + \hat{f}_{disturb}^{body} \right) \end{aligned}. \tag{14}$$

Following the MCRLS method described earlier, the EOM will be manipulated into forms that enable the required IDs, individually. In the following sections, the resulting values of the A, x, and b variables from the general regression form ($Ax \cong b$) are listed for each individual ID. Unfortunately, due to the need for brevity, only the end results of the algebraic manipulations are presented.

3.5 CM ID using gyros

$$A_k = \hat{I}^{-1} \begin{bmatrix} 0 & -c_{k,3} & c_{k,2} \\ c_{k,3} & 0 & -c_{k,1} \\ -c_{k,2} & c_{k,1} & 0 \end{bmatrix}, \quad x = \begin{bmatrix} \Delta_{C,1} \\ \Delta_{C,2} \\ \Delta_{C,3} \end{bmatrix}, \tag{15}$$

$$b_k = \dot{\hat{\omega}} - \hat{I}^{-1}(a_k - \hat{\omega} \times (\hat{I}\hat{\omega})).$$

The identified value for $\hat{\Delta}_C$ is then added to the nominal center of mass, to give the center of mass estimate:

$$\hat{C} = C_{nom} + \hat{\Delta}_C . \tag{16}$$

3.6 CM ID using accelerometers

$$A_k = \begin{bmatrix} -\hat{\omega}_2^2 - \hat{\omega}_3^2 & -\hat{\dot{\omega}}_3 + \hat{\omega}_1\hat{\omega}_2 & \hat{\dot{\omega}}_2 + \hat{\omega}_1\hat{\omega}_3 \\ \hat{\dot{\omega}}_3 + \hat{\omega}_1\hat{\omega}_2 & -\hat{\omega}_1^2 - \hat{\omega}_3^2 & -\hat{\dot{\omega}}_1 + \hat{\omega}_2\hat{\omega}_3 \\ -\hat{\dot{\omega}}_2 + \hat{\omega}_1\hat{\omega}_3 & \hat{\dot{\omega}}_1 + \hat{\omega}_2\hat{\omega}_3 & -\hat{\omega}_1^2 - \hat{\omega}_2^2 \end{bmatrix}_k , \tag{17}$$

$$x = \begin{bmatrix} \Delta_{C,1} \\ \Delta_{C,2} \\ \Delta_{C,3} \end{bmatrix}, b_k = m^{-1}\left(c_k + \hat{f}_{disturb}^{body}\right) - \hat{\ddot{x}}_a^{body} + e_k .$$

3.7 Inverse-inertia ID using gyros

The regression form equation for inverse-inertia ID using gyros, where the gyroscopic term is treated as a disturbance is:

$$A_k = \begin{bmatrix} d_{k,1} & & & d_{k,2} & d_{k,3} & \\ & d_{k,2} & & d_{k,1} & & d_{k,3} \\ & & d_{k,3} & & d_{k,1} & d_{k,2} \end{bmatrix},$$

$$x = \begin{bmatrix} \Delta_{I^{-1},11} \\ \Delta_{I^{-1},22} \\ \Delta_{I^{-1},33} \\ \Delta_{I^{-1},12} \\ \Delta_{I^{-1},13} \\ \Delta_{I^{-1},23} \end{bmatrix}, \quad b_k = \hat{\dot{\omega}}_k - I_{nom}^{-1} d_k . \tag{18}$$

$$\hat{I}^{-1} = I_{nom}^{-1} + \hat{\Delta}_{I^{-1}} . \tag{19}$$

By treating the gyroscopic term as a disturbance, we do not learn anything from it; we only try to keep it from negatively impacting the ID. This means that information is only present when thrusters are firing, and no ID updates occur during coasting periods. Depending on the spacecraft inertia properties and typical angular rates, this term may be sufficiently significant as compared to noise and disturbances to warrant direct treatment in the ID, as shown later. The magnitude of the gyroscopic term is greater for asymmetric spacecraft, and for high angular rates about more than one principal axis.

3.8 Inertia ID using gyros

The regression form equation for inertia-matrix ID using gyros only, where the gyroscopic term is treated as a disturbance is:

$$A_k = \begin{bmatrix} \hat{\dot{\omega}}_{k,1} & & & \hat{\dot{\omega}}_{k,2} & \hat{\dot{\omega}}_{k,3} & \\ & \hat{\dot{\omega}}_{k,2} & & \hat{\dot{\omega}}_{k,1} & & \hat{\dot{\omega}}_{k,3} \\ & & \hat{\dot{\omega}}_{k,3} & & \hat{\dot{\omega}}_{k,1} & \hat{\dot{\omega}}_{k,2} \end{bmatrix},$$

$$x = \begin{bmatrix} \Delta_{I,11} \\ \Delta_{I,22} \\ \Delta_{I,33} \\ \Delta_{I,12} \\ \Delta_{I,13} \\ \Delta_{I,23} \end{bmatrix}, \quad b_k = d_k - I_{nom}\hat{\dot{\omega}}_k . \tag{20}$$

Now treating the gyroscopic term as significant:

$$A_k = \begin{bmatrix} \hat{\dot{\omega}}_1 & -\hat{\omega}_2\hat{\omega}_3 & \hat{\omega}_2\hat{\omega}_3 & \hat{\dot{\omega}}_2 - \hat{\omega}_1\hat{\omega}_3 & \hat{\dot{\omega}}_3 + \hat{\omega}_1\hat{\omega}_2 & \hat{\omega}_2^2 - \hat{\omega}_3^2 \\ \hat{\omega}_1\hat{\omega}_3 & \hat{\dot{\omega}}_2 & -\hat{\omega}_1\hat{\omega}_3 & \hat{\dot{\omega}}_1 + \hat{\omega}_2\hat{\omega}_3 & \hat{\omega}_3^2 - \hat{\omega}_1^2 & \hat{\dot{\omega}}_3 - \hat{\omega}_1\hat{\omega}_2 \\ -\hat{\omega}_1\hat{\omega}_2 & \hat{\omega}_1\hat{\omega}_2 & \hat{\dot{\omega}}_3 & \hat{\omega}_1^2 - \hat{\omega}_2^2 & \hat{\dot{\omega}}_1 - \hat{\omega}_2\hat{\omega}_3 & \hat{\dot{\omega}}_2 + \hat{\omega}_1\hat{\omega}_3 \end{bmatrix}_k \tag{21}$$

$$x = \begin{bmatrix} \Delta_{I,11} \\ \Delta_{I,22} \\ \Delta_{I,33} \\ \Delta_{I,12} \\ \Delta_{I,13} \\ \Delta_{I,23} \end{bmatrix}, \quad b_k = a_k + c_k \times \hat{\Delta}_C - A_k \begin{bmatrix} I_{nom,11} \\ I_{nom,22} \\ I_{nom,33} \\ I_{nom,12} \\ I_{nom,13} \\ I_{nom,23} \end{bmatrix}.$$

$$\hat{I} = I_{nom} + \hat{\Delta}_I . \tag{22}$$

Following updates during coasting periods, normalize $\hat{\Delta}_I$ so $\left\|\hat{I}\right\|$ does not change; otherwise during coasting, the solution would converge to $\hat{I} = 0$ since that solves Equation 13 exactly (with $T_k = 0$ and $\hat{\tau}_{disturb} = 0$, the right side of Equation 13 is zero).

This is the preferred approach for identification of inertia properties since it is the only one of the three to directly address the gyroscopic term, and can therefore be updated during coasting rotations. However, if the spacecraft symmetry and typical angular rates make the gyroscopic term negligible, there is no advantage over the other approaches.

3.9 Thruster-magnitude ID

$$A_k = \begin{bmatrix} (\hat{L} \times D)S_k \operatorname{diag}(T_k) \\ DS_k \operatorname{diag}(T_k) \end{bmatrix}, x = \operatorname{diag}(F_{bias}),$$

$$b_k = \begin{bmatrix} \hat{I}\dot{\omega} + \omega \times (\hat{I}\omega) \\ m\ddot{x}_C^{body} \end{bmatrix} - A_k \operatorname{diag}(F_{nom}). \tag{23}$$

$$\hat{L} = J - (C_{nom} + \hat{\Delta}_C)\begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}$$

$$\hat{\ddot{x}}_C^{body} = \hat{\ddot{x}}_a^{body} - \hat{\dot{\omega}} \times (r_\alpha - C_{nom} - \hat{\Delta}_C) - \hat{\omega} \times (\hat{\omega} \times (r_\omega - C_{nom} - \hat{\Delta}_C)) \tag{24}$$

Here the diag() operator is used, as in MATLAB, to convert the column vector, T_k, to a diagonal matrix, and conversely for F_{bias}.

3.10 Total mass ID

$$A_k = DS_k\left(F_{nom} + \hat{F}_{bias}\right)T_k + \hat{f}^{body}_{disturb}, \quad x = \Delta_{m^{-1}},$$
$$b_k = \hat{\ddot{x}}^{body}_a - e_k + \hat{\dot{\omega}} \times \hat{\Delta}_C + \hat{\omega} \times (\hat{\omega} \times \hat{\Delta}_C) - A_k m^{-1}_{nom}. \tag{25}$$

$$\hat{m} = (\hat{m}^{-1})^{-1} = (m^{-1}_{nom} + \hat{\Delta}_{m^{-1}})^{-1}. \tag{26}$$

3.11 RLS updating

The equations derived represent the relationship between variables at each sample update. So at each k update, and for each ID to be updated, the corresponding A_k matrix and b_k vector are created and run through the individual RLS ID [9].

In some cases equations have been presented for the rotational EOM (using gyros) as well as for the translational EOM (using accelerometers). The use of one or the other or both should be decided based on the relative accuracy of the acceleration estimates. If both are used, they should be weighted accordingly (e.g., based on the relative accuracy in translational vs. rotational accelerations). If one or the other is not available, the corresponding rows in A_k and b_k should be deleted.

As with any system ID, the result is only as good as the data used. In particular, these results are dependent on accurate values for angular acceleration, $\hat{\dot{\omega}}_k$. Special care should be taken to estimate that as accurately as possible, for example as described in Section 2. Similarly, several aspects of the thruster model are assumed to be perfectly known (directions, scale factor, locations)–they should be calibrated carefully since imprecision can lead to biased ID results.

4. Conclusions

Algorithms have been derived that identify, on-line, center of mass, inverse-inertia, inertia, thruster force, and total mass for thruster-controlled spacecraft. They have been used with a new algorithm, multiple concurrent recursive least squares (MCRLS ID) that enables linear RLS ID even though the unknown parameters appear nonlinearly in the system governing equations. ID algorithms have been developed and applied in simulation to the X-38, Mini-AERCam, S4, and MIT SPHERES thruster-controlled vehicles. Using gyro signals only, or with the additional capability provided with accelerometers, and based on recursive least squares, the algorithms reliably and accurately ID mass properties for these vehicles in the presence of several significant noise sources. The algorithms are computationally efficient and can be run either on-line for adaptive control (RLS) or off-line for post-flight analysis (batch LS).

Acknowledgements

Research funded by NASA Headquarters, HQ AA, PWC 349-00, and by the NASA Intelligent Systems Program 302-10-10 (part of the CICT Program). Thanks to Dustin Berkovitz, Dr. Edmund Kong, Prof. David Miller, and the rest of the MIT SPHERES team for assistance with the SPHERES integration. Thanks to Rodolfo Gonzalez and Dr. Steven Fredrickson of NASA JPL for assistance in the problem formulations for the X-38 and Mini-AERCam applications, as well as for highlighting on-line spacecraft ID as an important issue for advanced spacecraft dynamical analysis.

References

[1] E. Wilson, D.W. Sutter, and R.W. Mah, Multiple concurrent recursive least squares identification, *Proc. IASTED International Conference on Intelligent Systems and Control*, Honolulu, HI, Aug 2004.

[2] E. Wilson, C. Lages, and R.W. Mah, On-line, gyro-based, mass-property identification for thruster-controlled spacecraft, *Proc. 2002 IEEE International Midwest Symposium on Circuits and Systems*, Tulsa, OK, Aug 2002.

[3] E. Wilson, D. Sutter, et al., Motion-based system identification and fault detection and isolation technologies for thruster controlled spacecraft, *Proc. JANNAF 3rd Modeling and Simulation Joint Subcommittee Meeting*, Colorado Springs, CO, Dec 2003.

[4] C. L. Lawson and R. J. Hanson, *Solving least squares problems, Series in automatic computation* (Englewood Cliffs, NJ: Prentice Hall, 1974).

[5] L. Ljung, *System identification: theory for the user* (Upper Saddle River, NJ: Prentice Hall, 1999).

[6] S. Tanygin and T. Williams, Mass property estimation using coasting maneuvers, *Journal of Guidance, Control, and Dynamics, 20*(Jul-Aug) 1997, 625-632.

[7] E.V. Bergmann, B.K. Walker, and D.R. Levy, Mass property estimation for control of asymmetrical satellites, *Journal of Guidance, Control, and Dynamics*, *10*(Sep-Oct) 1987, 483-491.

[8] E. Wilson and S.M. Rock, Reconfigurable control of a free-flying space robot using neural networks, *Proc. 1995 American Control Conference*, Seattle WA, Jun 1995.

[9] G. Franklin, J.D. Powell, and M.L. Workman, *Digital control of dynamic systems, second edition* (Menlo Park, CA: Addison Wesley, 1990).