Chapter | eleven

Implementation of Δv Commands

It will have outlet, brave and not so brave.
weapons of war and implements of peace
Are but the points at which it finds release.

Robert Frost (1874–1963)

Historically, sensor noise has been a key factor in the analysis of system performance for formation flying spacecraft [39,172]. Since formation flying missions require coordination between multiple spacecraft, knowledge of the relative states must be as accurate as possible. This knowledge is typically used when planning trajectories to meet specific criteria, such as minimizing fuel use or maintaining a desired formation geometry. The planning process depends on knowledge of the initial conditions of the spacecraft [173], and degrades with increasing error. Specifically, Tillerson and How [38] showed that velocity estimation error was the primary cause of poor performance, and that an error of just 2 mm/s can result in errors of about 30 m after just one orbit.[1]

During the late 1990's and early 2000's the best navigation filters, using Carrier-Phase Differential GPS (CDGPS) signals, achieved velocity accuracy on the order of 0.5 mm/s [174]. Since then, the state-of-the-art has improved significantly, with Leung and Montenbruck [175] demonstrating a filter that can estimate relative position and velocity to within 1.5 mm and 5 µm/s, respectively. While those numbers represent a best-case performance for real-world operation, it is nonetheless important to understand how other sources of noise can affect formation flying. The PRISMA formation flying demonstration mission aims to validate sensor and actuator technologies for formation flight and rendezvous and docking. Consisting of a two-spacecraft formation deployed in a sun-synchronous 700 km orbit, detailed simulations of the estimation and

[1] This error estimate is consistent with the along-track drift term in Eq. (5.18). For $x(0) = 0$, the drift becomes $y_{\text{drift}}(t) = -3\dot{y}(0)t$. For an initial velocity error of 2 mm/s and a 90-minute orbit, the resulting drift is $y_{\text{drift}} = 32.4$ m.

Spacecraft Formation Flying; ISBN: 9780750685337

control processes have indicated that a driver of system performance is thruster performance [176]. Specifically, the minimum impulse bit of 0.7 mm/s is only adequate for controlling the relative mean along-track formation to within 60 m when using the impulsive control scheme presented in Ref. [3], indicating that incorrect implementation of a thruster burn is a significant source of error.

Thruster burns are commonly specified by the spacecraft's desired change in velocity (Δv), and the propulsion subsystem is then responsible for applying this Δv to the spacecraft. Mission-critical maneuvers require precise implementation of thruster burns. For example, for the Cassini Saturn Orbit Insertion ($\Delta v \approx 625$ m/s), an algorithm that measured the energy change of the spacecraft was used. This energy change was monitored autonomously by Cassini during the burn, using real-time measurements from onboard accelerometers [177]. The insertion maneuver was successfully terminated when the desired energy change was reached, and Cassini became the first spacecraft to orbit Saturn. Similarly, the smart impactor of the Deep Impact mission successfully targeted the comet Tempel 1 on July 4, 2005. Precision implementation of trajectory correction maneuvers (TCMs), made possible by the attitude determination and control (ADCS) software and accelerometers, is credited as a primary reason for mission success [178]. The TCM-1 maneuver had a command Δv of 28.568 m/s, and the control system delivered 28.561 m/s – a performance within 0.03% accuracy.

This chapter discusses the impact of Δv implementation error on the performance of spacecraft formation flying, rendezvous and docking. It also explores how accelerometers can be used to improve performance by providing accurate measurements of the applied Δv.

11.1 PLAN IMPLEMENTATION

If knowledge of the initial state of the spacecraft is sufficiently accurate to determine a good plan, the next key step is to accurately implement the plan. A good plan is one that meets performance objectives (formation geometry, fuel management, drift-free). If such a plan is improperly implemented, the performance objectives might not be met. At best, this probably means replanning and trying again, resulting in a loss of time and fuel. At worst, it could lead to the loss of a spacecraft or an entire formation. A spacecraft executes a Δv command as a sequence of one or more thruster firings. The on and off times of the thrusters can be pre-computed beforehand or calculated on-the-fly. Considering the simple case of a single thruster pointing in the direction of motion, one option is to calculate the burn duration based on an idealized thruster model, and then execute the burn. This open-loop strategy is

$$t_b = \frac{m \Delta v_d}{T_d} \tag{11.1}$$

where t_b is the burn duration, m is the mass of the spacecraft (assumed constant throughout the burn), Δv_d is the desired velocity change, and T_d is the expected force that is provided by the thruster. This approach is easy to implement, but it is usually a poor idea. Although data on the expected thrust would typically be available, this data is likely based on laboratory testing under specific or idealized conditions. As the operating conditions of the thruster change, so too will its performance. This makes it difficult to predict the performance of the thruster or the open-loop system. The actual delivered thrust can instead be modeled as

$$T_a = T_d(1+\delta) + w \tag{11.2}$$

where T_a is the actual thrust delivered, δ is a small number representing a bias on the expected thrust, and w is a small random variable that accounts for any additional variations in the thrust. These additional variations are not limited to just unpredictable fluctuations in the engine thrust, but could also account for a spacecraft that is spinning slowly while thrusting (provided the time scale of the spin is small compared to the burn duration). Using the strategy in Eq. (11.1), and imposing the actuator model in Eq. (11.2), the actual Δv implemented using the open-loop strategy can be written as

$$\Delta v = \left(\frac{T_d}{m}\right) t_b + \left(\frac{T_d\delta + w}{m}\right) t_b \tag{11.3}$$

which is just Δv_d plus an error term. It is clear that the resulting error from this implementation will be proportional to δ as well as the length of the burn. For high Δv maneuvers, this error could be quite large, and is therefore unacceptably risky.

11.1.1 Using accelerometers to improve Δv implementation

A straightforward way to improve the performance of the actuator is to design a feedback control loop that uses sensors to measure thruster performance. Rather than relying on GPS or CDGPS to measure the Δv changes *after* the burn, this subsection explores using other sensors as a means to measure and track the Δv *as it is applied*. This approach uses direct feedback and the addition of an *axial accelerometer* along the thrust direction to achieve this objective. To simplify the following analysis, it is assumed that the thruster can deliver a continuous range of thrusts. Figure 11.1 depicts a block diagram for the closed-loop thruster control system. For clarity, the state is chosen as

$$\mathbf{x} = \begin{bmatrix} V^* \\ V_a \\ \hat{V} \\ e \end{bmatrix} \tag{11.4}$$

The commanded velocity is V^*, the actual velocity of the spacecraft is V_a, and the estimated velocity of the spacecraft is $\hat{V}$. The final element of the state vector, e, is the integral of the estimated velocity error. It will be used later by

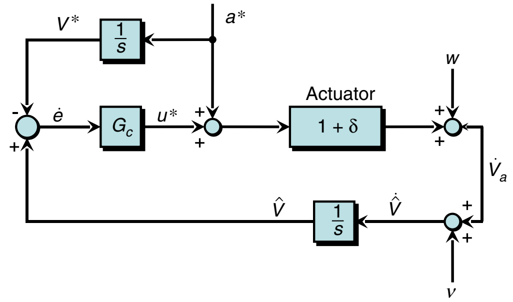


FIGURE 11.1 Closed-loop control system for a single thruster.

the controller. Although each of these velocities is actually a velocity *change*, the Δ is omitted for notational simplicity. The velocity command enters the system through the control input a^*, such that

$$\dot{V}^* = a^* \tag{11.5}$$

The actual acceleration experienced by the spacecraft is influenced by a number of factors. First, the acceleration command a^* is sent directly to the *thruster*, but is distorted by the *actuator* ("actuator" and "thruster" are used interchangeably) dynamics. Additionally, a corrective acceleration u^* is applied by the controller, but once more, the actuator dynamics distort u^* so that $u = (1+\delta)u^*$. This corrective acceleration should be thought of as a throttling of the thruster. For example, if $\delta > 0$, then u^* will be negative, and the thrust command will be reduced to compensate for the unexpected amplification of the signal that happens in the actuator. Finally, random thruster process noise w is added.

$$\dot{V}_a = (a^* + u^*)(1+\delta) + w \tag{11.6}$$

The accelerometer is mounted on the spacecraft and measures the actual acceleration directly with noise ν

$$\dot{\hat{V}} = \dot{V}_a + \nu \tag{11.7}$$

It is assumed that both w and ν are uncorrelated Gaussian, white-noise processes

$$w \sim \mathcal{N}(0, \sigma_w) \text{ and } \nu \sim \mathcal{N}(0, \sigma_\nu) \tag{11.8}$$

As noted above, e is the integral of the estimated velocity error.

$$e = \int_0^t (\hat{V} - V^*)dt \tag{11.9}$$

A proportional-integral (PI) controller $G_c(s)$ with gains k_p and k_i can be used to drive the estimated velocity error to zero, and is therefore a reasonable choice. The state of the integrator is e, as shown in Eq. (11.9). The control law for the PI controller may be written as

$$u^* = -k_p(\hat{V} - V^*) - k_i e \tag{11.10}$$

Substituting Eq. (11.10) into Eq. (11.6) and Eq. (11.7) yields the following state relationships:

$$\dot{V}^* = a^* \tag{11.11a}$$
$$\dot{V}_a = [-k_p(\hat{V} - V^*) - k_i e](1 + \delta) + a^*(1 + \delta) + w \tag{11.11b}$$
$$\dot{\hat{V}} = [-k_p(\hat{V} - V^*) - k_i e](1 + \delta) + a^*(1 + \delta) + w + v \tag{11.11c}$$
$$\dot{e} = \hat{V} - V^* \tag{11.11d}$$

Or, in matrix form

$$\begin{bmatrix} \dot{V}^* \\ \dot{V}_a \\ \dot{\hat{V}} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ (1+\delta)k_p & 0 & -(1+\delta)k_p & -(1+\delta)k_i \\ (1+\delta)k_p & 0 & -(1+\delta)k_p & -(1+\delta)k_i \\ -1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} V^* \\ V_a \\ \hat{V} \\ e \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ (1+\delta) & 1 & 0 \\ (1+\delta) & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a^* \\ w \\ v \end{bmatrix} \tag{11.12}$$

Equations (11.11) and (11.12) provide a convenient form for simulating the system response to a range of acceleration commands. To gain more insight into the steady state behavior of this controller, begin with the generic system

$$\begin{aligned} \dot{\mathbf{x}} &= A\mathbf{x} + B_u\mathbf{u} + B_w\mathbf{w} \\ \mathbf{y} &= C\mathbf{x} + \boldsymbol{v} \end{aligned} \tag{11.13}$$

where $\mathbf{x}$ is as in Eq. (11.4). In this formulation, the control inputs $\mathbf{u}$ and the random perturbations $\mathbf{w}$ are separated. If the control law is chosen as linear state feedback, such that

$$\mathbf{u} = -K\mathbf{x} \tag{11.14}$$

then Eq. (11.13) may be written as

$$\begin{aligned} \dot{\mathbf{x}} &= (A - B_uK)\mathbf{x} + B_w\mathbf{w} \Rightarrow \\ \dot{\mathbf{x}} &= A_{cl}\mathbf{x} + B_w\mathbf{w} \end{aligned} \tag{11.15}$$

The dynamics in Eq. (11.14) are that of a system driven by random process noise. For a linear time-invariant (LTI) system, if the process noise $\mathbf{w}$ is stationary, then the mean square value of the state, as $t \to \infty$, satisfies the Lyapunov equation

$$0 = A_{cl}X_{ss} + X_{ss}A_{cl}^T + B_w R_{ww} B_w^T \tag{11.16}$$

Given positive definite R_{ww}, Eq. (11.16) has a positive definite solution for the state covariance matrix X_{ss} if A_{cl} is stable [179]. Although Eq. (11.12) is convenient for simulating all of the parameters of interest, the A_{cl} matrix has two eigenvalues of 0, and therefore solution of Eq. (11.16) is not possible. By introducing $\epsilon = \hat{V} - V^*$ and $n = (w + \nu)$, a lower order system is obtained:

$$\begin{bmatrix} \dot{\epsilon} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} -(1+\delta)k_p & -(1+\delta)k_i \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \epsilon \\ e \end{bmatrix} + \begin{bmatrix} \delta & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} a^* \\ n \end{bmatrix} \tag{11.17}$$

Although a^* is not actually a random process noise, this formulation is still useful; provided $k_p > 0$ and $k_i > 0$, then A_{cl} for this system will be stable and there is a unique solution to the Lyapunov equation. While there are numerical algorithms for solving the equation, explicitly solving for the 2 × 2 case is feasible. The spectral intensity matrix R_{ww} for this problem is

$$R_{ww} = \begin{bmatrix} \sigma_{a^*}^2 & 0 \\ 0 & \sigma_n^2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & \sigma_n^2 \end{bmatrix} \tag{11.18}$$

Since $\sigma_{a^*}^2$ is the reference input, it is known exactly and therefore the upper left entry of R_{ww} is 0 in Eq. (11.18). The steady-state covariance matrix for the state is

$$X_{ss} = \begin{bmatrix} \sigma_\epsilon^2 & \rho_{\epsilon e}\sigma_\epsilon\sigma_e \\ \rho_{e\epsilon}\sigma_e\sigma_\epsilon & \sigma_e^2 \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix} \tag{11.19}$$

Here, σ and ρ represent standard deviations and correlation coefficients, respectively. If the entries of A_{cl} are numbered in the same manner as Eq. (11.19), then the solution of the Lyapunov equation is straightforward. Performing the matrix multiplications in Eq. (11.16) yields a system of four equations for the four unknowns in X_{ss}:

$$2A_{11}x_{11} + A_{12}x_{12} + x_{21} = \sigma_n^2 \tag{11.20}$$

$$A_{21}x_{11} + (A_{11} + A_{22})x_{12} + A_{12}x_{22} = 0 \tag{11.21}$$

$$A_{21}x_{11} + (A_{11} + A_{22})x_{21} + A_{12}x_{22} = 0 \tag{11.22}$$

$$A_{21}x_{12} + A_{21}x_{21} + 2A_{22}x_{22} = 0 \tag{11.23}$$

Simultaneous solution of these equations gives the steady state performance of the closed-loop control system driven by the random input **w**:

$$X_{ss} = \begin{bmatrix} \dfrac{(\sigma_w + \sigma_\nu)^2}{2(1+\delta)k_p} & 0 \\ 0 & \dfrac{(\sigma_w + \sigma_\nu)^2}{2(1+\delta)k_p k_i} \end{bmatrix} \tag{11.24}$$

The diagonal elements x_{11} and x_{22} in Eq. (11.24) are the expected variances in the state variables ϵ and e, respectively. As might be expected, if either the process noise or sensor noise increases, the steady state behavior of the estimate degrades.

However, an important distinction is that Eq. (11.24) describes how well the *estimated velocity* will converge on the command velocity. Of perhaps more importance to overall performance is how well this estimated velocity tracks the actual imparted velocity, V_a. From Fig. 11.1,

$$\dot{\hat{V}} - \dot{V}_a = \nu \Rightarrow \hat{V} - V_a = \int_0^t \nu dt \tag{11.25}$$

As seen in Eq. (11.25), the estimation error during the burn grows as the integral of the random sensor noise ν. While the expected value of the error is zero for any burn duration, the variance of the estimate will increase for longer burns. This means that for long burns, although the expected implementation error remains zero, the uncertainty grows. Fortunately, the sensor noise for a good accelerometer will be quite small, and the benefit of the added information far outweighs any potential pitfalls. For example, the Mars Reconnaissance Orbiter (MRO) accelerometers had a measurement noise level of just 0.005 mm/s^2 [180]. Furthermore, for long burns, it might be possible to periodically obtain velocity measurements from other sensors to reduce the estimate error.

Increasing the gains k_p and k_i in the controller will cause the estimated velocity to converge faster to the command velocity, but it does not affect the long-term tracking of the actual velocity; the Δv accuracy is purely limited by the sensor noise. As mentioned previously, provided the accelerometer is of reasonably good quality, then $\nu \ll \left(\frac{T_d}{m}\right)\delta$ and the error for the closed-loop system will grow much more slowly than for the open-loop system.

11.1.2 Discrete example

The analysis in Section 11.1.1 is for a continuous system, but in reality, a control system would be implemented digitally. This section discusses the extension of the previous results to the digital domain. The steady state performance of the system can be directly obtained from Eq. (11.24) by scaling the process and sensor noise. The relationship between *continuous noise* and *discrete noise* models is discussed in [181]. For this case, they are

$$\sigma_w = \sigma_{Dw}\sqrt{T} \tag{11.26a}$$

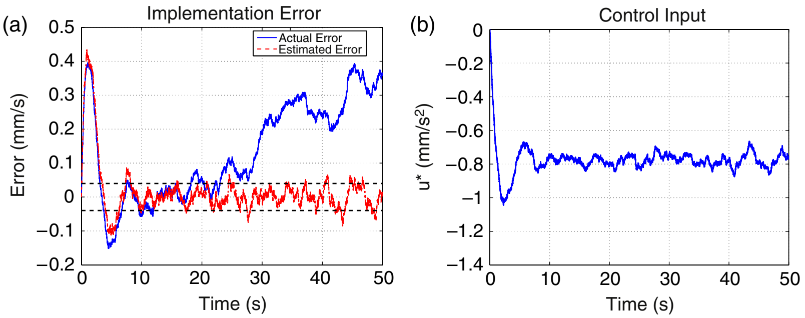


FIGURE 11.2 Results of a discrete simulation of the closed-loop algorithm.

$$\sigma_\nu = \sigma_{D\nu}\sqrt{T} \tag{11.26b}$$

$$\sigma_n = \sigma_{D\nu}\sqrt{T} \tag{11.26c}$$

where $\sigma_{D(*)}$ represents the standard deviation of a digital measurement or process, and T is the sample period. Substituting Eq. (11.26) into Eq. (11.24) gives the mean square performance of the digital estimate:

$$\sigma_\epsilon{}^2 = \frac{(\sigma_{Dw} + \sigma_{D\nu})^2 T}{2(1+\delta)k_p} \tag{11.27}$$

Now, the steady state convergence can be improved by increasing the sampling frequency (and reducing T). Figure 11.2 shows the results of a simulation of the closed-loop control system. The parameters for the simulation were set as follows: $a^* = 4$ cm/s^2, $V^* = 200$ cm/s, $\delta = 0.02$, $\sigma_{Dw} = 0.08$ cm/s, $\sigma_{D\nu} = 0.1$ cm/s^2, $T = 0.001$ s, $k_d = 1$, and $k_i = 1$.

For a shorter burn, the gains could be set higher to obtain a faster convergence of the estimate, but they were left low in this case so that the corrective action of the controller is clearly visible. Since $\delta > 0$, the actual thrust is initially higher than the commanded thrust. Therefore, near the beginning of the burn, the accelerometers detect the variation and the estimated error starts to increase. The controller then applies a differential thrust in the negative direction to counteract δ. Because a^* is 4 cm/s^2 and $\delta = 0.02$, this differential thrust should be about -0.8 mm/s^2. The control history shown in Fig. 11.2(b) confirms this prediction.

For the values used in this simulation, Eq. (11.27) predicts that $\sigma_\epsilon \approx 0.04$ mm/s. This region is marked by the dashed line in Fig. 11.2(a) and closely matches the actual behavior. The data also shows how the actual error tends to drift from the estimated error as a consequence of Eq. (11.25). The overall performance of the closed-loop system results in an error of 0.36 mm/s, for a burn of 2000 mm/s, or less than 0.02% error. For the open-loop controller, the error would have been δ, or 2%. The closed-loop system delivers 100 times better performance.

11.2 IMPACT ON AUTONOMOUS RENDEZVOUS AND DOCKING

Autonomous rendezvous and docking is an area of ongoing research and promises to enable both space exploration and on-orbit assembly and servicing. The effect of process noise on an autonomous rendezvous scenario was simulated for two satellites initially separated by about 150 meters. One satellite, the chaser, must maneuver to the location of the second satellite, the target, over the course of one orbit. Only the chaser satellite fires its thrusters, and the orbit is circular at an altitude of 335 km (the approximate altitude of the International Space Station).

For a circular orbit, the time-invariant relative dynamics of two point-mass spacecraft separated by a short distance are described by the CW equations, discussed in Chapter 5; however, as we showed in Chapter 9, the CW equations do not provide accurate modeling of relative dynamics when the target and chaser spacecraft are arbitrarily shaped due to rotation-translation coupling. If the latter effect is neglected, the resulting perturbation must be absorbed by control accelerations.

In the presence of a control acceleration input, $\mathbf{u} = [u_x, u_y, u_z]^T$, the CW equations in the absence of rotation-translation coupling are given by

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3n^2 & 0 & 0 & 0 & 2n & 0 \\ 0 & 0 & 0 & -2n & 0 & 0 \\ 0 & 0 & -n^2 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} \tag{11.28}$$

where, as usual, $n = \sqrt{\mu/a^3}$ is the mean motion on the reference orbit and a is the semimajor axis of the reference orbit.

The linear nature of Eq. (11.28) enables the use of *convex optimization* techniques to calculate a fuel-optimized plan [40]. The chaser's objective is to fire its thrusters and move to the origin (target satellite) over one orbit period. The orbit period is discretized into 1000 segments, with control inputs allowed during every segment. The only objectives for the rendezvous problem are to minimize fuel use and reach the target, and these objectives yield the simple optimization

$$\min_{\mathbf{u}} \|\mathbf{u}\|_1 \qquad \text{subject to} \quad \mathbf{x}_f = \mathbf{x}_d \tag{11.29}$$

$\mathbf{x}_f$ is the actual state of the chaser at the end of the orbit, $\mathbf{x}_d$ is its desired state, and $\mathbf{u}$ is the sequence of control inputs applied at each step of the plan. The orbital dynamics enter through the constraint $\mathbf{x}_f = \mathbf{x}_d$, and the initial condition and control inputs are chosen to satisfy it.

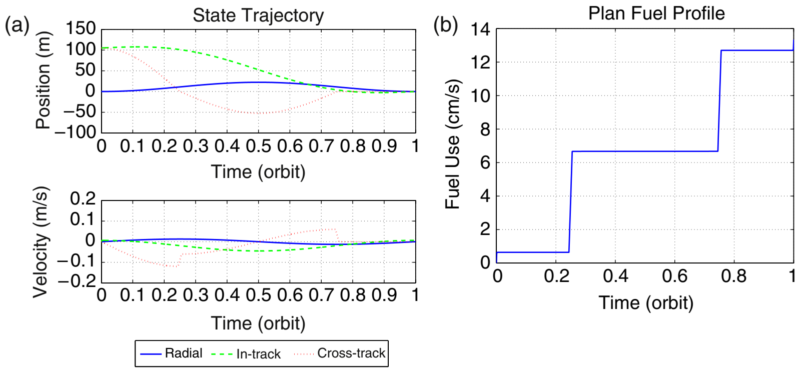


FIGURE 11.3 The ideal rendezvous trajectory and plan.

The minimization problem in Eq. (11.29) is a *linear programming* (LP) problem. It can be solved efficiently using one of many available commercial or free solvers (e. g., `linprog` in `MATLAB`® or `COIN-LP`[2]).

Ideally, the control system would add no process noise, and the plan would be executed perfectly. Figure 11.3 shows the ideal trajectory. At the end of the orbit, the chaser has reached the target (Fig. 11.3(a)). The total fuel consumed is 13.34 cm/s, with control applied at four different points in the orbit; two burns at the beginning and end of about 6.4 mm/s each, and two more burns of 6 cm/s around a quarter and three quarters of the way through the orbit. As this is the ideal performance case, all real control systems will perform worse, either by using more fuel or by failing to satisfy the terminal constraint.

Two parameters were independently varied for the simulations: Process noise and replan frequency. Process noise is modeled in the same manner as in Eq. (11.2), with a random δ for each burn which acts as a percentage error on the magnitude of each thruster firing. This way, longer burns with an inaccurate actuator lead to larger implementation errors. The random δ has a mean value $\bar{\delta}$ and a standard deviation σ_δ. If $\bar{\delta} \neq 0$ then the actuator is biased. As the process noise varies from one simulation to the next, σ_δ is the parameter that changes. Reducing $\bar{\delta}$ or σ_δ is equivalent to improving the actuator. Tested values for σ_δ ranged anywhere from 0.01% to 10% error. The replan frequency f_p is how many times the chaser satellite re-solves the optimization during a single orbit rendezvous mission. When the optimization is re-solved, the initial condition of the chaser satellite is adjusted to its current position. Errors in the initial conditions due to sensor limitations are not considered here; it is assumed that the relative positions and velocities are exactly known.

For the ideal case (with no process noise), replanning is not needed because the control inputs are applied precisely, but when process noise is introduced, errors in the implementation of the plan will cause the chaser satellite to deviate from its expected position. If left unchecked, this error will propagate all the

[2] See http://www.coin-or.org/index.html, accessed March 26, 2009.

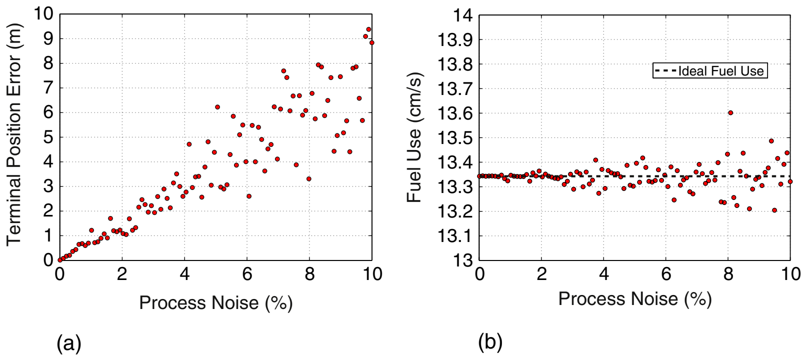


FIGURE 11.4 The effect of varying process noise on rendezvous performance. (a) Effect on terminal position. (b) Effect on fuel use.

way to the end of the trajectory. For a rendezvous mission, if this error is large enough, both the chaser and target satellites could be put at risk. Replanning provides a way to compensate for process error by detecting deviations and modifying the rendezvous trajectory accordingly.

In general, reducing process noise results in improved performance, as does increasing the replan frequency. Figure 11.4(a) shows how varying the process noise influences the performance of the rendezvous. The position error is measured as the rectilinear distance from the chaser to the target satellite at the end of the maneuver. For this set of simulations, the actuator was unbiased ($\bar{\delta} = 0$) and only the magnitude of σ_δ was varied. The control plan was created at the start of the maneuver and executed from start to finish, without replanning. Since no replanning was done, the process noise should have no effect on average fuel use because the deviations caused by incorrect Δv implementation are ignored; the planner does not expend fuel to try to correct them later. Figure 11.4(b) confirms this. The increased dispersion for larger σ_δ reflects the increased randomness of the thrusting, but there is no trend in the average value. For smaller σ_δ, the fuel use converges to the ideal case.

If instead, process noise is held constant while varying the replanning frequency, the performance changes as shown in Fig. 11.5. The process noise σ_δ was held fixed at 5% while replanning frequencies from 1 to 20 times per orbit were investigated. Additionally, in these tests, a bias of $\bar{\delta} = 10\%$ was introduced. As expected, the terminal position error is improved by increasing the replanning rate because replanning allows the effects of inaccurate thrusting to be caught and corrected. This enables the chaser to still reach the target satellite at the desired time. Still, as a result of the positive bias in the actuator, the fuel use remains above the ideal value. The true cause of the problem is not in the plan, but in the actuator that poorly implements it. Continuing to use the open-loop strategy while simply increasing the replanning frequency does nothing to address this root cause. The key distinction is that increasing the planning rate is a *reactive* solution; the errors must happen before they can be observed and corrected by the planner. For maneuvers

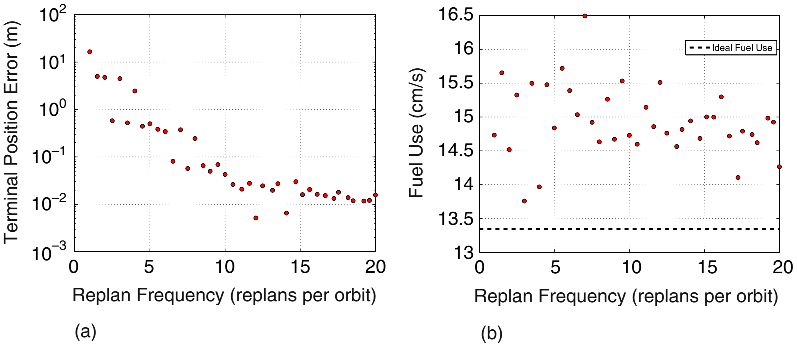


FIGURE 11.5 The effect of replanning frequency on rendezvous performance, with $\bar{\delta} = 10\%$ and $\sigma_{\delta} = 5\%$. (a) Effect on terminal position. (b) Effect on fuel use.

where a specific trajectory must be tracked closely or constraints avoided safely, replanning alone may not be enough to achieve the necessary performance. Additionally, the ability of a satellite to replan its trajectory is limited by several considerations. Accurate position and velocity estimation is required for both the chaser and the target satellites; repeatedly planning trajectories based on bad estimates is not only inefficient, but it could even result in a completely incorrect result. Moreover, the planning strategy or available computing power might place an upper bound on how often a trajectory can be replanned. Finally, thruster impingement constraints or science goals might limit the number and timing of possible firings.

On the other hand, using accelerometers to monitor the burn performance and more accurately implement Δv commands is a *proactive* solution; errors can be prevented from even happening. Provided the accelerometers are properly calibrated, the control system in Fig. 11.1 eliminates bias and significantly reduces the equivalent process noise.

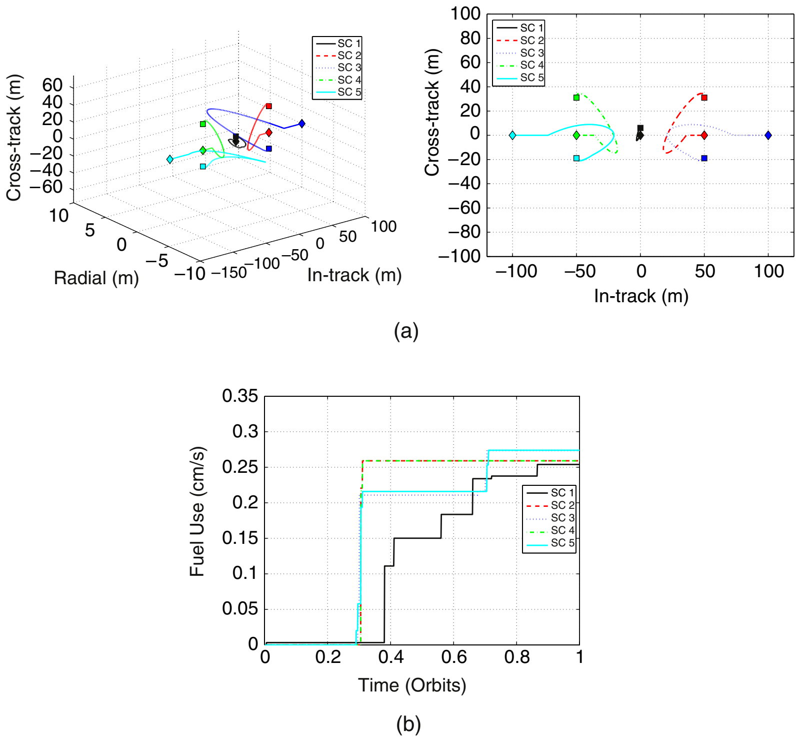
11.2.1 Impact on formation reconfiguration

This section investigates the performance degradation of a formation reconfiguration maneuver when process noise is added to the system. It utilizes an optimization-based planner. A formation of five spacecraft are in an elliptical orbit defined by

$$\boldsymbol{\oe}_{ref}(t_0) = [\,4.69549 \quad 0.471 \quad 1.10497 \quad 4.24115 \quad 3.7350\pi\,]^T$$

where the first element of $\boldsymbol{\oe}_{ref}$, the semimajor axis, is in units of Earth radii, the eccentricity is unitless, and the inclination, ascending node, argument of perigee, and mean anomaly are in radians.

They begin in an along-track formation, separated by 50 m (Spacecraft 1 at 100 m along-track, Spacecraft 2 at 50 m and so on, so that Spacecraft 5 is at -100 m), with the chief at the center of the formation. The desired configuration is a box in a frame relative to the reference orbit, with the chief



(a)

(b)

FIGURE 11.6 Nominal five spacecraft along-track to passive aperture reconfiguration maneuver. (a) Absolute frame (reference orbit fixed at origin). (b) Fuel use for the fleet.

at the origin and the following relative LVLH states: $[x, y, z, \dot{x}, \dot{y}, \dot{z}]^T$, with position components in meters and velocity components in meters/second):

$$
\begin{aligned}
\mathbf{x}_{d_2} &= \begin{bmatrix} 0 & 50 & 25 & 0 & 0 & 0 \end{bmatrix}^T \\
\mathbf{x}_{d_3} &= \begin{bmatrix} 0 & 50 & -25 & 0 & 0 & 0 \end{bmatrix}^T \\
\mathbf{x}_{d_4} &= \begin{bmatrix} 0 & -50 & 25 & 0 & 0 & 0 \end{bmatrix}^T \\
\mathbf{x}_{d_5} &= \begin{bmatrix} 0 & -50 & -25 & 0 & 0 & 0 \end{bmatrix}^T
\end{aligned}
\tag{11.30}
$$

This formation emulates an interferometer with four telescopes (the deputies) and a combiner located at the center (the chief), which attains a data-gathering geometry once per orbital period. An optimization-based formation reconfiguration plan for a full-orbit planning horizon was generated using the method in Ref. [182] and is shown in Fig. 11.6. Fuel consumption for each spacecraft (Fig. 11.6(b)) is 2.54 mm/s, 2.59 mm/s, 2.74 mm/s, 2.59 mm/s, and 2.74 mm/s, exhibiting good balance. The total fuel use is 13.20 mm/s. Spacecraft 3 and spacecraft 5 have farther to move to reach their desired positions, and so the formation center shifts 6.06 m in the positive out-of-plane direction in Fig. 11.6(a) to help equalize the fuel use.

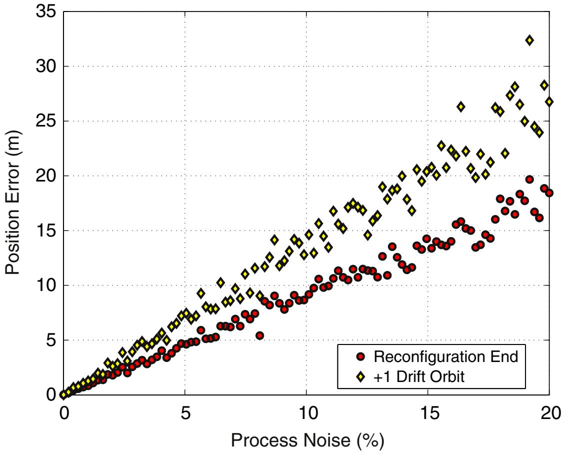


FIGURE 11.7 The effect of varying process noise on a formation reconfiguration.

A series of simulations was run with the formation implementing the plan with varying levels of process noise (0–20%). At the end of the maneuver, the orbit was allowed to continue to propagate for an extra orbit during which no control inputs were allowed (the drift orbit). The position errors for each spacecraft, measured as the rectilinear distance from the actual state to the desired state, were summed at both times. Each process noise level was simulated 20 times and the averaged results are plotted in Fig. 11.7. The circular points mark the position errors at the immediate conclusion of the plan, and the diamonds mark the position errors after the drift orbit. As expected, as the process noise increases, the formation deteriorates. However, the performance of the drift orbit also worsens; if errors are not quickly corrected, they compound. In each case, the position error grew during the drift orbit. The nominal plan does not visibly drift because the planner favors drift free orbits. Drift-free orbits are very sensitive to the initial conditions and this helps explain the rapid deterioration of formations with high implementation error.

SUMMARY

As sensing technology and the ability to design optimal trajectories for formations of spacecraft continue to improve, the need for control systems that accurately implement Δv is becoming more pronounced. This chapter developed a model of Δv implementation error and used it to investigate the dangers of poor thruster performance. For a rendezvous and docking scenario similar to what might be attempted on the International Space Station, small implementation errors of a few percent were found to result in terminal errors on the order of meters. For formation reconfigurations, the dangers are twofold: The desired relative geometry is not initially met, and the formation deteriorates more rapidly if it is allowed to drift. A feedback control system, using accelerometers to directly monitor thruster performance, was developed and analytical expressions describing its expected performance were obtained. For a thruster with 2% deviation from the expected thrust and realistic performance parameters, the closed-loop system improved the accuracy of the implemented Δv by two orders of magnitude. Although simply replanning to correct errors is a possibility, allowable thrust windows are often governed by science goals or other constraints; plans must be implemented correctly without the assumption that mistakes can just be corrected later.