We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



10

Predictive Control of Tethered Satellite Systems

Paul Williams
Delft University of Technology
Australia

1. Introduction

Tethered satellite systems have many potential applications, ranging from upper atmospheric research (Colombo et al., 1975) to momentum transfer (Nordley & Forward, 2001; Williams et al., 2004). The major dynamical features of the system have been studied extensively (Misra & Modi, 1986), but there still remain open questions with regard to control (Blanksby & Trivailo, 2000). Many of the open issues stem from the fact that there have been limited flight tests. The most recent flight of the Young Engineers' Satellite 2 (YES-2) highlighted from its results that tether dynamic modelling is relatively mature, but that there is a need to provide fault tolerant design in the control and sensor subsystems (Kruijff et al., 2009).

In applications such as momentum transfer and payload capture, it is imperative that robust, accurate and efficient controllers can be designed. For example, although it is conceivable to use onboard thrusters to manipulate the motion of the tethered satellite, this negates some of the advantages of using tethers, i.e., little to no fuel expenditure in ideal circumstances. The main source of control, therefore, has to be sought from manipulating the length of deployed tether. This has two main aims: first, the length of tether directly controls the distance of the tether tip from the main spacecraft, and second, changes in tether length induce Coriolis-type forces on the system due to the orbital motion, which allows indirect control over the swing motion of the tether (librations). Typically, control over the tether length is achieved via manipulating the tension at the mother satellite (Rupp, 1975; Lorenzini et al., 1996). This can help to prevent the tether from becoming slack – a situation that can lead to loss of control of the system.

A variety of different control strategies have been proposed in the literature on tethered systems. Much of the earlier work focused on controlling the deployment and retrieval processes (Xu et al., 1981; Misra & Modi, 1982; Fujii & Anazawa, 1994). This was usually achieved by combining an open-loop length control scheme with feedback of the tether states, either appearing linearly or nonlinearly. Other schemes were devoted to ensuring nonlinear asymptotic stability through the use of Lyapunov's second method (Fujii & Ishijima, 1989; Fujii, 1991; Vadali & Kim, 1991). Most of these techniques do not ensure well-behaved dynamics, and can be hard to tune to make the deployment and retrieval fast.

Because deployment and retrieval is an inherent two-point boundary value problem, it makes much more sense to approach the problem from the point-of-view of optimal control. Several examples of the application of optimal control theory to tethered satellite systems can be found (Fujii & Anazawa, 1994; Barkow, 2003; Lakso & Coverstone, 2000). However, the direct application of the necessary conditions for optimality leads to an extremely numerically sensitive two-point boundary value problem. The state-costate equations are well-known to suffer from instability, but the tethered satellite problem is notorious because of the instability of the state equations to small errors in the control tension. More recent work has focused on the application of direct transcription methods to the tethered satellite problem (Lakso & Coverstone, 2000; Williams, 2008; Williams & Blanksby, 2008). This provides advantages with respect to robustness of convergence and is typically orders of magnitude faster than other methods.

In recent work, the effect of the performance index used in solving the optimal control problem for tethered satellites was examined in detail (Williams, 2008). The work in (Williams, 2008) was prompted by the fact that bang-bang tension control trajectories have been proposed (Barkow, 2003), which is extremely undesirable for controlling a flexible tether. The conclusions reached in (Williams, 2008) suggest that an inelastic tether model can be sufficient to design the open-loop trajectory, provided the cost function is suitably selected. Suitable costs include the square of the tether length acceleration, tension rate or tension acceleration. These trajectories lead to very smooth variations in the dynamics, which ultimately improves the tracking capability of feedback controllers, and reduces the probability of instabilities.

Much of the previous work on optimal control of tethered satellites has focused on obtaining solutions, as opposed to obtaining *rapid* solutions. Some of the ideas that will be explored in this paper have been discussed in (Williams, 2004), which presented two approaches for implementing an optimal-based controller for tethered satellites. One of the methods was based on quasilinearization of the necessary conditions for optimality combined with a pseudospectral discretization, whereas the second was a direct discretization of the continuous optimal control problem. In (Williams, 2004), NPSOL was used as the nonlinear programming (NLP) solver, which implements methods based on dense linear algebra, and is significantly slower than the sparse counterpart SNOPT (Gill et al., 2002).

The aforementioned YES-2 mission had the aim of deploying a 32 km long tether in two phases. The first phase had the objective of stabilizing the tether swinging motion (librations) at the local vertical with the tether length at 3.5 km. The second phase had the objective of inducing a sufficient swinging motion at the end of deployment to allow a specially designed payload to re-enter the atmosphere and be recovered in Khazikstan. The deployment controller consisted of using a reference trajectory computed offline via direct transcription (Williams et al., 2008), in combination with a feedback controller to stabilize the deployment dynamics. The feedback controller used a time-varying feedback gain calculated via a receding horizon approach documented in (Williams, 2005). Flight results showed that despite very large perturbations from nominal, the tether was deployed successfully in the first phase. An issue with one of the sensors that measured the deployment rate caused the feedback controller to believe that the tether was being deployed too slowly. As a consequence, the tether was deployed too quickly. It has been

shown that the tether was nonetheless fully deployed, making it the longest tether ever deployed in space.
The aim of this Chapter is to explore the possibility of providing real-time optimal control for a tethered satellite system. A realistic tether model is combined with a nonlinear Kalman filter for estimating the tether state based on available measurements. A nonlinear model predictive controller is implemented to satisfy the mission requirements.

2. System Model

In order to generate rapid optimal trajectories and test closed-loop performance for a real system, it is necessary to introduce mathematical models of varying fidelity. In this chapter, two models are distinguished: 1) a high fidelity truth model, 2) a low fidelity control model. A truth model is required for testing the closed-loop performance of the controller in a representative environment. Typically, the truth model will incorporate effects that are not present in the model used by the controller. In the simplest case, these can be environmental disturbances. Truth models are usually of higher fidelity than the control model, and as such, they become difficult to use for real-time closed-loop control. For this reason, it is necessary to employ a reduced order model in the controller. It should be pointed out that a truth model will typically include a set of parameter perturbations that alter the characteristics of the simulated system compared to the assumptions made in the control model. Such perturbations are used in Monte Carlo simulations of the closed-loop system to gather statistics on the controller performance.
For the particular case of a tethered satellite system, there are a number of important dynamics that exist in the real system: 1) Rigid-body, librations of the tether in- and out-plane, 2) Lateral string oscillations of the tether between the tether attachment points, 3) Longitudinal spring-mass oscillations of the tether, 4) Rigid body motions of the end bodies, 5) Orbital perturbations caused by exchange of angular momentum from the tethered system with orbital angular momentum. All of these dynamic modes are coupled to varying degrees. However, the dominant dynamics are due to (1) and (2) as these directly impact the short-term response of the system.
The following subsections derive the fundamental equations of motion for modeling the tethered system taking into account the dominant dynamics. A simplified model suitable for model predictive control is then developed.

2.1 Truth Model

The most sophisticated models for tethered satellite systems treat the full effects of tether elasticity and flexibility. Examples include models based on discretization by assumed modes (Xu et al., 1986) or discretization by lumped masses (Kim & Vadali, 1995). In a typical lumped mass model, the tether is discretized into a series of point masses connected by elastic springs. The tension in each element can be computed explicitly based on the positions of the adjacent lumped masses. It is well known that the equations of motion for the system are 'stiff', referring to the fact that the dynamics occur over very different timescales, requiring small integration step sizes to capture the very high frequency modes. For a tethered satellite system, the high frequency modes are the longitudinal elastic modes, followed by the string modes of the tether, libration modes, and finally the orbital motion. For short duration missions or analysis, the longitudinal modes are unlikely to have a

significant effect on the overall motion (provided the tether remains taut). Thus, in this model the effects of longitudinal vibrations are ignored, and the tether is divided into a series of point masses connected via inelastic links. The geometric shortening of the distance to the tether tip is accounted for due to the changes in geometry of the system, but stretching of the tether is not. The degree of approximation is controlled by the number of discretized elements that are used.

The tether is modeled as consisting of a series of n point masses connected via inelastic links, as shown in Fig. 1. The (x,y,z) coordinate system rotates at the orbit angular velocity and is assumed to be attached at the center of mass of the orbit (mother satellite). Although not a necessary assumption in the model, it is assumed that the orbit of the mother satellite is prescribed and remains Keplerian. In general, this coordinate system would orbit in a plane defined by the classical orbital elements (argument of perigee, inclination, longitude of ascending node). In the presence of a Newtonian gravitational field, the orientation of the orbital plane does not affect the system dynamics. However, it does affect any aerodynamic or electrodynamic forces due to the nature of the Earth's rotating atmosphere and magnetic field. These effects are not considered here.

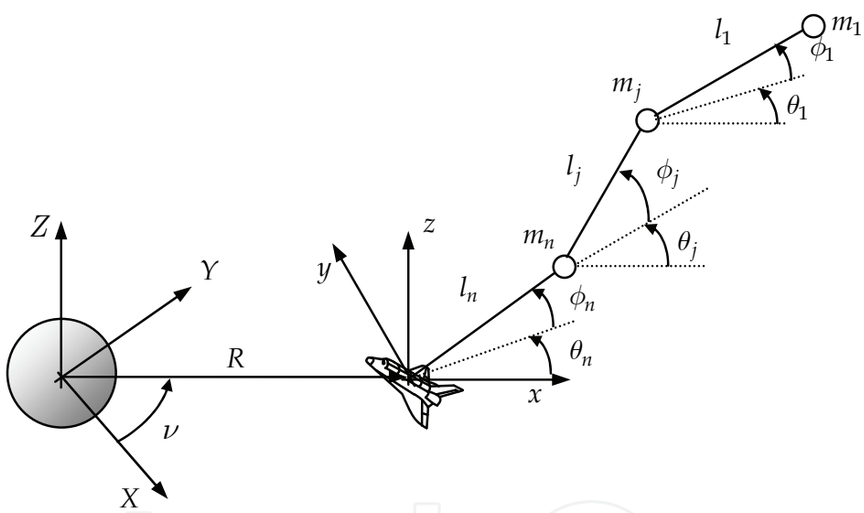


Fig. 1. Discretized multibody tether model.

The acceleration of a mass in the rotating frame is given by

$$\ddot{\boldsymbol{r}} = \left(\ddot{x} - \dot{\omega}y - 2\omega\dot{y} - \omega^2 x\right)\boldsymbol{i} + \left(\ddot{y} + \dot{\omega}x + 2\omega\dot{x} - \omega^2 y\right)\boldsymbol{j} + \ddot{z}\boldsymbol{k} \tag{1}$$

where $\omega = \kappa^2\sqrt{\mu / p^3}$ is the orbital angular velocity, $p = a(1-e^2)$ is the semilatus rectum, μ is the Earth's gravitational parameter, e is the orbit eccentricity, $\kappa = 1 + e\cos\nu$, and a is the orbit semimajor axis. The contribution of forces due to the gravity-gradient is given by

$$\boldsymbol{F}_j^g = 2m_j\frac{\mu x_j}{R^3}\boldsymbol{i} - m_j\frac{\mu y_j}{R^3}\boldsymbol{j} - m_j\frac{\mu z_j}{R^3}\boldsymbol{k} = 2m_j\frac{\omega^2 x_j}{\kappa}\boldsymbol{i} - m_j\frac{\omega^2 y_j}{\kappa}\boldsymbol{j} - m_j\frac{\omega^2 z_j}{\kappa}\boldsymbol{k} \tag{2}$$

Note that in Equation (1), the contributions due to the center of mass motion R and corresponding true anomaly ν are cancelled with the Newtonian gravity terms for the system center of mass. This is valid if the system is assumed to be in a Keplerian orbit.
Define the tension vector in the j th segment as

$$\boldsymbol{T}_j = T_j\left(\cos\theta_j\cos\phi_j\boldsymbol{i} + \sin\theta_j\cos\phi_j\boldsymbol{j} + \sin\phi_j\boldsymbol{k}\right) \tag{3}$$

Also, define the j th mass in terms of the generalized coordinates,

$$x_j = \sum_{k=j}^{n} l_k\cos\theta_k\cos\phi_k \tag{4}$$

$$y_j = \sum_{k=j}^{n} l_k\sin\theta_k\cos\phi_k \tag{5}$$

$$z_j = \sum_{k=j}^{n} l_k\sin\phi_k \tag{6}$$

The tension forces on the j th mass are given by

$$\boldsymbol{F}_j^{\text{tension}} := \begin{cases} \boldsymbol{T}_{j-1} - \boldsymbol{T}_j, & 1 < j \le n \\ -\boldsymbol{T}_j, & j = 1 \end{cases} \tag{7}$$

The equations of motion can be expressed in component form as

$$\begin{aligned} \ddot{x}_j - \dot{\omega}y_j - 2\omega\dot{y}_j - \omega^2 x_j - 2\frac{\omega^2}{\kappa}x_j &= \frac{F_j^x}{m_j} \\ \ddot{y}_j + \dot{\omega}x_j + 2\omega\dot{x}_j - \omega^2 y_j + \frac{\omega^2}{\kappa}y_j &= \frac{F_j^y}{m_j} \\ \ddot{z}_j + \frac{\omega^2}{\kappa}z_j &= \frac{F_j^z}{m_j} \end{aligned} \tag{8}$$

where m_j is the mass of the jth cable mass, and (F_j^x, F_j^y, F_j^z) is the vector of external forces acting on the jth mass in the orbital frame. Substitution of Equations (4) through (6) into Equation (8) gives the governing equations of motion in spherical coordinates. The equations of motion may be decoupled by employing a matrix transformation and forward substitution of the results. By multiplying the vector of Equation (8) by the matrix

$$\left[C_j\right] = \begin{bmatrix} -\sin\theta_j & \cos\theta_j & 0 \\ -\cos\theta_j\sin\phi_j & -\sin\theta_j\sin\phi_j & \cos\phi_j \\ \cos\theta_j\cos\phi_j & \sin\theta_j\cos\phi_j & \sin\phi_j \end{bmatrix} \tag{9}$$

the general decoupled equations of motion can be expressed as

$$\ddot{\theta}_j = -\dot{\omega} + 2\left(\dot{\theta}_j + \omega\right)\left[\dot{\phi}_j \tan\phi_j - \frac{\dot{l}_j}{l_j}\right] - 3\frac{\omega^2}{\kappa}\sin\theta_j\cos\theta_j - \frac{T_{j-1}\sin\theta_j\cos\theta_{j-1}\cos\phi_{j-1}}{m_j l_j\cos\phi_j}$$
$$-\frac{F_j^x\sin\theta_j}{m_j l_j\cos\phi_j} + \frac{T_{j-1}\cos\theta_j\sin\theta_{j-1}\cos\phi_{j-1}}{m_j l_j\cos\phi_j} + \frac{T_{j+1}\cos\theta_j\sin\theta_{j+1}\cos\phi_{j+1}}{m_{j+1} l_j\cos\phi_j} \quad (10)$$
$$-\frac{T_{j+1}\sin\theta_j\cos\theta_{j+1}\cos\phi_{j+1}}{m_{j+1} l_j\cos\phi_j} + \frac{F_j^y\cos\theta_j}{m_j l_j\cos\phi_j} + \frac{F_{j+1}^x\sin\theta_j}{m_{j+1} l_j\cos\phi_j} - \frac{F_{j+1}^y\cos\theta_j}{m_{j+1} l_j\cos\phi_j}$$

$$\ddot{\phi}_j = -2\frac{\dot{l}_j}{l_j}\dot{\phi}_j - \left[\left(\dot{\theta}_j + \omega\right)^2 + \frac{3\omega^2\cos^2\theta_j}{\kappa}\right]\sin\phi_j\cos\phi_j$$
$$+\frac{T_{j-1}}{m_j l_j}\left(\cos\phi_j\sin\phi_{j-1} - \cos\theta_j\sin\phi_j\cos\theta_{j-1}\cos\phi_{j-1} - \sin\theta_j\sin\phi_j\sin\theta_{j-1}\cos\phi_{j-1}\right)$$
$$+\frac{T_{j+1}}{m_{j+1} l_j}\left(\cos\phi_j\sin\phi_{j+1} - \cos\theta_j\sin\phi_j\cos\theta_{j+1}\cos\phi_{j+1} - \sin\theta_j\sin\phi_j\sin\theta_{j+1}\cos\phi_{j+1}\right) \quad (11)$$
$$+\frac{F_j^z\cos\phi_j}{m_j l_j} - \frac{F_{j+1}^z\cos\phi_j}{m_{j+1} l_j} + \frac{F_{j+1}^y\sin\theta_j\sin\phi_j}{m_{j+1} l_j} - \frac{F_j^y\sin\theta_j\sin\phi_j}{m_j l_j} - \frac{F_j^x\cos\theta_j\sin\phi_j}{m_j l_j}$$
$$+\frac{F_{j+1}^x\cos\theta_j\sin\phi_j}{m_{j+1} l_j}$$

$$\ddot{l}_j = l_j\left[\left(\dot{\theta}_j + \omega\right)^2\cos^2\phi_j + \dot{\phi}_j^2 - \frac{\omega^2}{\kappa}\left(1 - 3\cos^2\theta_j\cos^2\phi_j\right)\right] - \frac{T_j}{m_j} - \frac{T_j}{m_{j+1}}$$
$$+\frac{F_j^z\sin\phi_j}{m_j} - \frac{F_{j+1}^z\sin\phi_j}{m_{j+1}} + \frac{F_j^y\sin\theta_j\cos\phi_j}{m_j} - \frac{F_{j+1}^y\sin\theta_j\cos\phi_j}{m_{j+1}}$$
$$+\frac{F_j^x\cos\theta_j\cos\phi_j}{m_j} - \frac{F_{j+1}^x\cos\theta_j\cos\phi_j}{m_{j+1}} \quad (12)$$
$$\frac{T_{j-1}}{m_j}\left[\cos\theta_j\cos\phi_j\cos\theta_{j-1}\cos\phi_{j-1} + \sin\phi_j\sin\phi_{j-1} + \sin\theta_j\cos\phi_j\sin\theta_{j-1}\cos\phi_{j-1}\right]$$
$$\frac{T_{j+1}}{m_{j+1}}\left[\cos\theta_j\cos\phi_j\cos\theta_{j+1}\cos\phi_{j+1} + \sin\phi_j\sin\phi_{j+1} + \sin\theta_j\cos\phi_j\sin\theta_{j+1}\cos\phi_{j+1}\right]$$

These equations may be nondimensionalized by utilizing the following relationships

$$\frac{\mathrm{d}}{\mathrm{d}t} = \omega\frac{\mathrm{d}}{\mathrm{d}\nu}, \qquad \frac{\mathrm{d}^2}{\mathrm{d}t^2} = \dot{\omega}\frac{\mathrm{d}}{\mathrm{d}\nu} + \omega^2\frac{\mathrm{d}^2}{\mathrm{d}\nu^2} \quad (13)$$

$$l_j = \Lambda_j L,\ u_j = \frac{T_j}{m_j\omega^2 L} \quad (14)$$

$$\dot{\omega} = -2\frac{\omega^2}{\kappa}e\sin\nu \quad (15)$$

Thus, the following nondimensional equations of motion are obtained

$$\theta_j'' = 2\left(\theta_j' + 1\right)\left[\frac{e\sin\nu}{\kappa} + \phi_j'\tan\phi_j - \frac{\Lambda_j'}{\Lambda_j}\right] - \frac{3}{\kappa}\sin\theta_j\cos\theta_j + \frac{u_{j+1}\cos\theta_j\sin\theta_{j+1}\cos\phi_{j+1}}{\Lambda_j\cos\phi_j}$$
$$+\frac{m_{j-1}}{m_j}\frac{u_{j-1}\cos\theta_j\sin\theta_{j-1}\cos\phi_{j-1}}{\Lambda_j\cos\phi_j} - \frac{m_{j-1}}{m_j}\frac{u_{j-1}\sin\theta_j\cos\theta_{j-1}\cos\phi_{j-1}}{\Lambda_j\cos\phi_j}$$
$$-\frac{u_{j+1}\sin\theta_j\cos\theta_{j+1}\cos\phi_{j+1}}{\Lambda_j\cos\phi_j} + \frac{F_j^y\cos\theta_j}{m_j\omega^2L\Lambda_j\cos\phi_j} - \frac{F_j^x\sin\theta_j}{m_j\omega^2L\Lambda_j\cos\phi_j} + \frac{F_{j+1}^x\sin\theta_j}{m_{j+1}\omega^2L\Lambda_j\cos\phi_j} \tag{16}$$
$$-\frac{F_{j+1}^y\cos\theta_j}{m_{j+1}\omega^2L\Lambda_j\cos\phi_j}$$

$$\phi_j'' = 2\frac{e\sin\nu}{\kappa}\phi' - 2\frac{\Lambda_j'}{\Lambda_j}\phi_j' - \left[\left(\theta_j' + 1\right)^2 + \frac{3\cos^2\theta_j}{\kappa}\right]\sin\phi_j\cos\phi_j + \frac{m_{j-1}}{m_j}\frac{u_{j-1}}{\Lambda_j}(\cos\phi_j\sin\phi_{j-1}$$
$$-\cos\theta_j\sin\phi_j\cos\theta_{j-1}\cos\phi_{j-1} - \sin\theta_j\sin\phi_j\sin\theta_{j-1}\cos\phi_{j-1}) + \frac{u_{j+1}}{\Lambda_j}(\cos\phi_j\sin\phi_{j+1}$$
$$-\cos\theta_j\sin\phi_j\cos\theta_{j+1}\cos\phi_{j+1} - \sin\theta_j\sin\phi_j\sin\theta_{j+1}\cos\phi_{j+1}) + \frac{F_j^z\cos\phi_j}{m_j\omega^2L\Lambda_j} \tag{17}$$
$$-\frac{F_{j+1}^z\cos\phi_j}{m_{j+1}\omega^2L\Lambda_j} + \frac{F_{j+1}^y\sin\theta_j\sin\phi_j}{m_{j+1}\omega^2L\Lambda_j} - \frac{F_j^y\sin\theta_j\sin\phi_j}{m_j\omega^2L\Lambda_j} - \frac{F_j^x\cos\theta_j\sin\phi_j}{m_j\omega^2L\Lambda_j} + \frac{F_{j+1}^x\cos\theta_j\sin\phi_j}{m_{j+1}\omega^2L\Lambda_j}$$

$$\Lambda_j'' = 2\frac{e\sin\nu}{\kappa}\Lambda_j' + \Lambda_j\left[\left(\theta_j' + 1\right)^2\cos^2\phi_j + \phi_j'^2 - \frac{1}{\kappa}\left(1 - 3\cos^2\theta_j\cos^2\phi_j\right)\right] - u_j - \frac{m_j}{m_{j+1}}u_j$$
$$+\frac{F_j^z\sin\phi_j}{m_j\omega^2L} - \frac{F_{j+1}^z\sin\phi_j}{m_{j+1}\omega^2L} + \frac{F_j^y\sin\theta_j\cos\phi_j}{m_j\omega^2L} - \frac{F_{j+1}^y\sin\theta_j\cos\phi_j}{m_{j+1}\omega^2L}$$
$$+\frac{F_j^x\cos\theta_j\cos\phi_j}{m_j\omega^2L} - \frac{F_{j+1}^x\cos\theta_j\cos\phi_j}{m_{j+1}\omega^2L} + \frac{m_{j-1}}{m_j}u_{j-1}(\cos\theta_j\cos\phi_j\cos\theta_{j-1}\cos\phi_{j-1} \tag{18}$$
$$+\sin\phi_j\sin\phi_{j-1} + \sin\theta_j\cos\phi_j\sin\theta_{j-1}\cos\phi_{j-1}) + u_{j+1}(\cos\theta_j\cos\phi_j\cos\theta_{j+1}\cos\phi_{j+1}$$
$$+\sin\phi_j\sin\phi_{j+1} + \sin\theta_j\cos\phi_j\sin\theta_{j+1}\cos\phi_{j+1})$$

Equations (16) through (18) utilize the orbit true anomaly ν as independent variable, and L is a scaling length representing the length of each tether element when fully deployed. The applicable boundary conditions are

$$m_0 = 0, \qquad u_0 = 0, \qquad m_{n+1} = \infty, \qquad u_{n+1} = 0 \tag{19}$$

The equations (16) through (18) define the dynamics of the tethered satellite system using spherical coordinates. These are not as general as Cartesian coordinates due to the singularity introduced when $\phi_j = -\frac{\pi}{2}, \frac{\pi}{2}$. This represents very large out of plane librational motion or very large out of plane lateral motion. Although this is a limitation of the model, such situations need to be avoided for most practical missions.

2.1.1 Variable Length Case

The tether is modeled as a collection of lumped masses connected by inelastic links, which makes dealing with the case of a variable length tether more difficult than if the tether was modeled as a single link. In particular, it is necessary to have a state vector of variable dimension and to add and subtract elements from the model at appropriate times. When the tether is treated as elastic, great care needs to be exercised to ensure that the introduction of new elements does not create unnecessary cable oscillations. This can happen if the position of the new mass results in the incorrect tension in the new element. However, for an inelastic tether, the introduction of a new mass occurs such that it is placed along the same line as the existing element. Thus, the new initial conditions for the incoming element are that it has the same angles and angle rates as the existing element (closest to the deployer). Alternative formulations based on the variation principle of Hamilton-Ostrogradksi and which transform the deployed length to a fixed interval by means of a new spatial coordinate have also been used (Wiedermann et al., 1999). However, this was not considered in this work.

If the critical length for introduction of a new element is defined as $\Lambda^* \triangleq 1 + k^*$, then the new element is initialized with a length of k^* in nondimensional units, and the same length rate as the previous nth element. During retrieval, elements must be removed. Here, the nth element to be removed and the (n-1)th element need to be used to update the initial conditions for the new n^* th element. In this work, the position and velocity of the (n-1)th mass is used to initialize the n^* th element. Thus, let

$$\begin{aligned} x_{n-1} &= \Lambda_n \cos\theta_n \cos\phi_n + \Lambda_{n-1} \cos\theta_{n-1} \cos\phi_{n-1} \\ y_{n-1} &= \Lambda_n \sin\theta_n \cos\phi_n + \Lambda_{n-1} \sin\theta_{n-1} \cos\phi_{n-1} \\ z_{n-1} &= \Lambda_n \sin\phi_n + \Lambda_{n-1} \sin\phi_{n-1} \end{aligned} \tag{20}$$

From which

$$\Lambda_{n^*} = \sqrt{x_{n-1}^2 + y_{n-1}^2 + z_{n-1}^2},\ \theta_{n^*} = \text{atan2}(y_{n-1}, x_{n-1}),\ \phi_{n^*} = \sin^{-1}(z_{n-1} / \Lambda_{n^*}) \tag{21}$$

where atan2 represents the four quadrant inverse tangent where the usual arctangent is defined by $\tan^{-1}(\frac{y_{n-1}}{x_{n-1}})$. Similarly, the relative velocity of the (n-1)th mass in the rotating frame is given by

$$\begin{aligned} x'_{n-1} &= \Lambda'_n \cos\theta_n \cos\phi_n - \Lambda_n \theta'_n \sin\theta_n \cos\phi_n - \Lambda_n \phi'_n \cos\theta_n \sin\phi_n \\ &\quad + \Lambda'_{n-1} \cos\theta_{n-1} \cos\phi_{n-1} - \Lambda_{n-1} \theta'_{n-1} \sin\theta_{n-1} \cos\phi_{n-1} - \Lambda_{n-1} \phi'_{n-1} \cos\theta_{n-1} \sin\phi_{n-1} \\ y'_{n-1} &= \Lambda'_n \sin\theta_n \cos\phi_n + \Lambda_n \theta'_n \cos\theta_n \cos\phi_n - \Lambda_n \phi'_n \sin\theta_n \sin\phi_n \\ &\quad + \Lambda'_{n-1} \sin\theta_{n-1} \cos\phi_{n-1} + \Lambda_{n-1} \theta'_{n-1} \cos\theta_{n-1} \cos\phi_{n-1} - \Lambda_{n-1} \phi'_{n-1} \sin\theta_{n-1} \sin\phi_{n-1} \\ z'_{n-1} &= \Lambda'_n \sin\phi_n + \Lambda_n \phi'_n \cos\phi_n + \Lambda'_{n-1} \sin\phi_{n-1} + \Lambda_{n-1} \phi'_{n-1} \cos\phi_{n-1} \end{aligned} \tag{22}$$

From which

$$\begin{aligned} \Lambda'_{n^*} &= x'_{n-1} \cos\phi_{n^*} \cos\theta_{n^*} + y'_{n-1} \cos\phi_{n^*} \sin\theta_{n^*} + z'_{n-1} \sin\phi_{n^*} \\ \theta'_{n^*} &= \left(y'_{n-1} \cos\theta_{n^*} - x'_{n-1} \sin\theta_{n^*} \right) / \left(\Lambda_{n^*} \cos\phi_{n^*} \right) \\ \phi'_{n^*} &= \left(z'_{n-1} \cos\phi_{n^*} - x'_{n-1} \sin\phi_{n^*} \cos\theta_{n^*} - y'_{n-1} \sin\phi_{n^*} \sin\theta_{n^*} \right) / \Lambda_{n^*} \end{aligned} \tag{23}$$

It should be noted that these updates keep the position and velocity of the (n-1)th mass the same across the update. The reason for this is that the positions and velocities of all subsequent masses depend on the position/velocity of the nth mass. Hence, if this is changed, then the position and velocity of all masses representing the tether change instantaneously. The accuracy of the updates depend on the transition parameter k^{**}, which is used to monitor the length of the nth segment. An element is removed when $\Lambda_n < k^{**}$. Because the tether is inelastic, altering the length of the new nth element does not keep the total tether length or mass constant unless the nth and (n-1)th elements are tangential. Therefore, by choosing k^{**} small enough, the errors in the approximation can be made small.

For control purposes, it is assumed that the rate of change of reel-rate is controlled. Thus, Λ_n'' is specified or determined through a control law. This means that the nth element is allowed to vary in length, but all other segments remained fixed in length. The problem is to then solve for the unknown tension constraints that enforce constant total length of the remaining segments, as well as the acceleration of the nth segment. Once these are known, they are back-substituted into Equations (16) and (17), as well as Equation (18) for the nth element. The equations formed by the set (18) are linear in the tensions u_j, and can thus be solved using standard techniques. This assumes that the segment lengths, length rates, and length accelerations are specified. In this work, LAPACK is utilized in solving the simultaneous equations.

2.1.2 Fixed Length Case

To simulate the case of a fixed length tether, Equations (18) are set to zero for $j = 1,...,n$, allowing the unknown tensions $u_j, j = 1,...,n$ to be determined. The resulting tensions are substituted back into the librational dynamics to determine the evolution of the system dynamics.

2.2 Control Model

The predominant modeling assumption that is used in the literature insofar as control of tethered satellite systems is concerned is that the system can be modeled with three degrees of freedom (Williams, 2008). In other words, when dealing with the librational motion of the system, it is sufficient to model it using spherical coordinates representing the dynamics of the subsatellite. This effectively treats the tether as a straight body, which can either be modeled as an inelastic or elastic rod. Early work has neglected the tether mass since its contribution to the librational motion can be considered relatively small (Fujii & Anazawa, 1994). This is due to the fact that the tether is axisymmetric. When large changes in length are considered, the effect of tether mass becomes more important. Moreover, it is essential to include the effects of tether mass when designing tension control laws because there is a nonlinear relationship between tension and tether mass. However, when performing preliminary analyses, it is sufficient to ignore such effects and compensate for these later in the design.

Although the assumption of treating the tether as a straight rod is often a good one, it can create some problems in practice. For example, all tether string vibrations are neglected,

which play a very important role in electrodynamic systems or systems subjected to long-term perturbations. Furthermore, large changes in deployment velocity can induce significant distortions to the tether shape, which ultimately affects the accuracy of the deployment control laws. Earlier work focused much attention on the dynamics of tethers during length changes, particularly retrieval (Misra & Modi, 1986). In the earlier work, assumed modes was typically the method of choice (Misra & Modi, 1982). However, where optimal control methods are employed, high frequency dynamics can be difficult to handle even with modern methods. For this reason, most optimal deployment/retrieval schemes consider the tether as inelastic.

2.1 Straight, Inelastic Tether Model

In this model, the tether is assumed to be straight and inextensible, uniform in mass, the end masses are assumed to be point masses, and the tether is deployed from one end mass only. The generalized coordinates are selected as the tether in-plane libration angle, θ, the out-of-plane tether libration angle, ϕ, and the tether length, l.

The radius vector to the center of mass may be written in inertial coordinates as

$$\boldsymbol{R} = R\cos\nu\boldsymbol{i} + R\sin\nu\boldsymbol{j} \tag{24}$$

From which the kinetic energy due to translation of the center of mass is derived as

$$T_{\mathrm{t}} = \tfrac{1}{2}m\left(\dot{R}^2 + R^2\dot{\nu}^2\right) \tag{25}$$

where $m = m_1 + m_t + m_2$ is the total system mass, $m_1 = m_1^0 - m_t$ is the mass of the mother satellite, m_t is the tether mass, m_2 is the subsatellite mass, and m_1^0 is the mass of the mother satellite prior to deployment of the tether.

The rotational kinetic energy is determined via

$$T_{\mathrm{r}} = \tfrac{1}{2}\boldsymbol{\omega}^T[I]\boldsymbol{\omega} \tag{26}$$

where $\boldsymbol{\omega}$ is the inertial angular velocity of the tether in the tether body frame

$$\boldsymbol{\omega} = \left(\dot{\nu}\sin\phi + \dot{\theta}\sin\phi\right)\boldsymbol{i} - \left(\dot{\phi}\right)\boldsymbol{j} + \left(\dot{\nu}\cos\phi + \dot{\theta}\cos\phi\right)\boldsymbol{k} \tag{27}$$

Thus we have that

$$T_{\mathrm{r}} = \tfrac{1}{2}m^*l^2[\dot{\phi}^2 + \left(\dot{\nu} + \dot{\theta}\right)^2\cos^2\phi] \tag{28}$$

and $m^* = \left(m_1 + \frac{m_t}{2}\right)\left(m_2 + \frac{m_t}{2}\right) / m - m_t / 6$ is the system reduced mass. The kinetic energy due to deployment is obtained as

$$T_{\mathrm{e}} = \tfrac{1}{2}\frac{m_1(m_2+m_t)}{m}\dot{l}^2 \tag{29}$$

which accounts for the fact that the tether is modeled as stationary inside the deployer and is accelerated to the deployment velocity after exiting the deployer. This introduces a thrust-like term into the equations of motion, which affects the value of the tether tension.
The system gravitational potential energy is (assuming a second order gravity-gradient expansion)

$$V = -\frac{\mu m}{R} + \frac{\mu m^* l^2}{2R^3}\left(1 - 3\cos^2\theta\cos^2\phi\right) \tag{30}$$

The Lagrangian may be formed as

$$\begin{aligned} L = {} & \tfrac{1}{2}m\left(\dot{R}^2 + R^2\dot{\nu}^2\right) + \tfrac{1}{2}m^* l^2[\dot{\phi}^2 + \left(\dot{\nu}+\dot{\theta}\right)^2\cos^2\phi] \\ & + \tfrac{1}{2}\frac{m_1(m_2+m_t)}{m}\dot{l}^2 + \frac{\mu m}{R} - \frac{\mu m^* l^2}{2R^3}\left(1 - 3\cos^2\theta\cos^2\phi\right) \end{aligned} \tag{31}$$

Under the assumption of a Keplerian reference orbit for the center of mass, the nondimensional equations of motion can be written as

$$\theta'' = 2\left(\theta'+1\right)\left[\frac{e\sin\nu}{\kappa} + \phi'\tan\phi - \frac{m_1\left(m_2+\frac{m_t}{2}\right)}{mm^*}\frac{\Lambda'}{\Lambda}\right] - \frac{3}{\kappa}\sin\theta\cos\theta + \frac{Q_\theta}{m^*\Lambda^2 L_r^2\dot{\nu}^2\cos^2\phi} \tag{32}$$

$$\phi'' = \frac{2e\sin\nu}{\kappa}\phi' - 2\frac{m_1\left(m_2+\frac{m_t}{2}\right)}{mm^*}\frac{\Lambda'}{\Lambda}\phi' - \left[\left(\theta'+1\right)^2 + \frac{3}{\kappa}\cos^2\theta\right]\sin\phi\cos\phi + \frac{Q_\phi}{m^*\Lambda^2 L_r^2\dot{\nu}^2} \tag{33}$$

$$\begin{aligned} \Lambda'' = {} & \frac{2e\sin\nu}{\kappa}\Lambda' - \frac{(2m_1-m)\frac{m_t}{2}}{m_1(m_2+m_t)}\frac{\Lambda'^2}{\Lambda} + \left(\frac{m_2+\frac{m_t}{2}}{m_2+m_t}\right)\Lambda[\phi'^2 + \left(\theta'+1\right)^2\cos^2\phi \\ & + \frac{1}{\kappa}\left(3\cos^2\theta\cos^2\phi - 1\right)] - \frac{T}{m_1\dot{\nu}^2 L_r(m_2+m_t)/m} \end{aligned} \tag{34}$$

where $\Lambda = l/L_r$ is the nondimensional tether length, L_r is a reference tether length, T is the tether tension, and $()' = \mathrm{d}()/\mathrm{d}\nu$. The generalized forces Q_θ and Q_ϕ are due to distributed forces along the tether, which are typically assumed to be negligible.

3. Sensor models

The full dynamic state of the tether is not directly measurable. Furthermore, the presence of measurement noise means that some kind of filtering is usually necessary before directly using measurements from the sensors in the feedback controller. The following measurements are assumed to be available: 1) Tension force at the deployer, 2) Deployment rate, 3) GPS position of the subsatellite. Models of each of these are developed in the subsections below.

3.1 Tension Model

The tension force measured at the deployer differs from the force predicted by the control model due to the presence of tether oscillations and sensor noise. The magnitude and direction of the force in the tether is obtained from the multibody tether model. The tension force in the orbital frame is given by

$$\begin{aligned} T_x &= u_n m_n \omega^2 L \cos\theta_n \cos\phi_n + w_{T_x} \\ T_y &= u_n m_n \omega^2 L \sin\theta_n \cos\phi_n + w_{T_y} \\ T_z &= u_n m_n \omega^2 L \sin\phi_n + w_{T_z} \end{aligned} \tag{35}$$

where the w terms are zero mean, Gaussian measurement noise with covariance R_T.

3.2 Reel-Rate Model

In general, the length of the deployed tether can be measured quite accurately. In this chapter, the reel-rate is measured at the deployer according to

$$\dot{L} = \Lambda_n' \omega L_n + w_{\dot{L}} \tag{36}$$

where $w_{\dot{L}}$ is a zero mean, Gaussian measurement noise with covariance $R_{\dot{L}}$.

3.3 GPS Model

GPS measurements of the two end bodies significantly improve the estimation performance of the system. The position of the mother satellite is required to form the origin of the orbital coordinate system (in case of non-Keplerian motion), and the position of the subsatellite allows observations of the subsatellite range and relative position (libration state). Only position information is used in the estimator. The processed relative position is modeled in the sensor model, as opposed to modeling the satellite constellation and pseudoranges. The processed position error is modeled as a random walk process

$$\delta\dot{x} = \frac{w_x}{\tau_{GPS}},\ \delta\dot{y} = \frac{w_y}{\tau_{GPS}},\ \delta\dot{z} = \frac{w_z}{\tau_{GPS}} \tag{37}$$

where $w_{x,y,z}$ are zero mean white noise processes with covariance R_{GPS}, and τ_{GPS} is a time constant. This model takes into account that the GPS measurement errors are in fact time-correlated.

4. State Estimation

In order to estimate the full tether state, it is necessary to combine all of the measurements obtained from the sensors described in Section 3. The most optimal way to combine the measurements is by applying a Kalman filter. Various forms of the Kalman filter are available for nonlinear state estimation problems. The two most commonly used filter implementations are the Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF). The UKF is more robust to filter divergence because it captures the propagation of

uncertainty in the filter states to a higher order than the EKF, which only captures the propagation to first order. The biggest drawback of the UKF is that it is significantly more expensive than the EKF. Consider a state vector of dimension n_x. The EKF only requires the propagation of the mean state estimate through the nonlinear model, and three matrix multiplications of the size of the state vector ($n_x \times n_x$). The UKF requires the propagation of $2n_x + 1$ state vectors through the nonlinear model, and the sum of vector outer products to obtain the state covariance matrix. The added expense can be prohibitive for embedded real-time systems with small sampling times (i.e., on the order of milliseconds). For the tethered satellite problem, the timescales of the dynamics are long compared to the available execution time. Hence, higher-order nonlinear filters can be used to increase performance of the estimation without loss of real-time capability.
Recently, an alternative to the UKF was introduced that employs a spherical-radial-cubature rule for numerically integrating the moment integrals needed for nonlinear estimation. The filter has been called the Cubature Kalman Filter (CKF). This filter is used in this chapter to perform the nonlinear state estimation.

4.1 Cubature Kalman Filter

In this section, the CKF main steps are summarized. The justification for the methodology is omitted and may be found in (Guess & Haykin, 2009).
The CKF assumes a discrete time process model of the form

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{v}_k, t_k) \tag{38}$$

$$\boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{w}_k, t_k) \tag{39}$$

where $\boldsymbol{x}_k \in \mathbb{R}^{n_x}$ is the system state vector, $\boldsymbol{u}_k \in \mathbb{R}^{n_u}$ is the system control input, $\boldsymbol{y}_k \in \mathbb{R}^{n_y}$ is the system measurement vector, $\boldsymbol{v}_k \in \mathbb{R}^{n_v}$ is the vector of process noise, assumed to be white Gaussian with zero mean and covariance $\boldsymbol{Q}_k \in \mathbb{R}^{n_v \times n_v}$, $\boldsymbol{w}_k \in \mathbb{R}^{n_w}$ is a vector of measurement noise, assumed to be white Gaussian with zero mean and covariance $\boldsymbol{R}_k \in \mathbb{R}^{n_w \times n_w}$. For the results in this paper, the continuous system is converted to a discrete system by means of a fourth-order Runge-Kutta method.
In the following, the process and measurement noise is implicitly augmented with the state vector as follows

$$\boldsymbol{x}_k^a = \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{v}_k \\ \boldsymbol{w}_k \end{bmatrix} \tag{40}$$

The first step in the filtering process is to compute the set of cubature points as follows

$$\mathcal{X}_{k-1} = \left[\hat{\boldsymbol{x}}_{k-1}^a + \boldsymbol{I}_{n_a \times n_a} \sqrt{\boldsymbol{P}_k}, \hat{\boldsymbol{x}}_{k-1}^a - \boldsymbol{I}_{n_a \times n_a} \sqrt{\boldsymbol{P}_k} \right] \tag{41}$$

where $\hat{\boldsymbol{x}}^a$ is the mean estimate of the augmented state vector, and $\boldsymbol{P}_k$ is the covariance matrix. The cubature points are then propagated through the nonlinear dynamics as follows

$$\mathcal{X}^*_{k|k-1} = \boldsymbol{f}(\mathcal{X}_{k-1}, \boldsymbol{u}_k, t_k) \tag{42}$$

The predicted mean for the state estimate is calculated from

$$\hat{\boldsymbol{x}}_k^- = \frac{1}{2n_a}\sum_{i=0}^{2n_a}\mathcal{X}^*_{i,k|k-1} \tag{43}$$

The covariance matrix is predicted by

$$\boldsymbol{P}_k^- = \frac{1}{2n_a}\sum_{i=0}^{2n_a}\mathcal{X}^*_{i,k|k-1}\mathcal{X}^{*T}_{i,k|k-1} - \hat{\boldsymbol{x}}_k^-\hat{\boldsymbol{x}}_k^{-T} \tag{44}$$

When a measurement is available, the augmented sigma points are propagated through the measurement equations

$$\mathcal{Y}_{k|k-1} = \boldsymbol{h}(\mathcal{X}_{k|k-1}, \boldsymbol{u}_k, t_k) \tag{45}$$

The mean predicted observation is obtained by

$$\hat{\boldsymbol{y}}_k^- = \frac{1}{2n_a}\sum_{i=0}^{2n_a}\mathcal{Y}_{i,k|k-1} \tag{46}$$

The innovation covariance is calculated using

$$\boldsymbol{P}_k^{yy} = \frac{1}{2n_a}\sum_{i=0}^{2n_a}\mathcal{Y}_{i,k|k-1}\mathcal{Y}^T_{i,k|k-1} - \hat{\boldsymbol{y}}_k^-\hat{\boldsymbol{y}}_k^{-T} \tag{47}$$

The cross-correlation matrix is determined from

$$\boldsymbol{P}_k^{xy} = \frac{1}{2n_a}\sum_{i=0}^{2n_a}\mathcal{X}_{i,k|k-1}\mathcal{Y}^T_{i,k|k-1} - \hat{\boldsymbol{x}}_k^-\hat{\boldsymbol{y}}_k^{-T} \tag{48}$$

The gain for the Kalman update equations is computed from

$$\mathcal{K}_k = \boldsymbol{P}_k^{xy}(\boldsymbol{P}_k^{yy})^{-1} \tag{49}$$

The state estimate is updated with a measurement of the system $\boldsymbol{y}_k$ using

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + \mathcal{K}_k\left(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k^-\right) \tag{50}$$

and the covariance is updated using

$$\boldsymbol{P}_k^+ = \boldsymbol{P}_k^- - \mathcal{K}_k \boldsymbol{P}_k^{yy} \mathcal{K}_k^T \tag{51}$$

It is often necessary to provide numerical remedies for covariance matrices that do not maintain positive definiteness. Such measures are not discussed here.

5. Optimal Trajectory Generation

Most of the model predictive control strategies that have been suggested in the literature are based on low-order discretizations of the system dynamics, such as Euler integration. Dunbar et al. (2002) applied receding horizon control to the Caltech Ducted Fan based on a B-spline parameterization of the trajectories. In recent years, pseudospectral methods, and in particular the Legendre pseudospectral (PS) method (Elnagar, 1995; Ross & Fahroo, 2003), have been used for real-time generation of optimal trajectories for many systems. The traditional PS approach discretizes the dynamics via differentiation operators applied to expansions of the states in terms of Lagrange polynomial bases. Another approach is to discretize the dynamics via Gauss-Lobatto quadratures. The approach has been more fully described by Williams (2006). The latter approach is used here.

5.1 Discretization approach

Instead of presenting a general approach to solving optimal control problems, the Gauss-Lobatto approach presented in this section is restricted to the form of the problem solved here. The goal is to find the state and control history $\{\boldsymbol{x}(t), \boldsymbol{u}(t)\}$ to minimize the cost function

$$\mathcal{J} = \mathcal{M}\left[\boldsymbol{x}(t_f)\right] + \int_{t_0}^{t_f} \mathcal{L}\left[\boldsymbol{x}(t^*), \boldsymbol{u}(t^*), t^*\right] \mathrm{d}t^* \tag{52}$$

subject to the nonlinear state equations

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\left[\boldsymbol{x}(t), \boldsymbol{u}(t), t\right] \tag{53}$$

the initial and terminal constraints

$$\boldsymbol{\psi}_0\left[\boldsymbol{x}(t_0)\right] = \boldsymbol{0} \tag{54}$$

$$\boldsymbol{\psi}_f\left[\boldsymbol{x}(t_f)\right] = \boldsymbol{0} \tag{55}$$

the mixed state-control path constraints

$$\boldsymbol{g}_L \leq \boldsymbol{g}\left[\boldsymbol{x}(t), \boldsymbol{u}(t), t\right] \leq \boldsymbol{g}_U \tag{56}$$

and the box constraints

$$\boldsymbol{x}_L \leq \boldsymbol{x}(t) \leq \boldsymbol{x}_U, \qquad \boldsymbol{u}_L \leq \boldsymbol{u}(t) \leq \boldsymbol{u}_U \tag{57}$$

where $\boldsymbol{x} \in \mathbb{R}^{n_x}$ are the state variables, $\boldsymbol{u} \in \mathbb{R}^{n_u}$ are the control inputs, $t \in \mathbb{R}$ is the time, $\mathcal{M} : \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}$ is the Mayer component of cost function, i.e., the terminal, non-integral cost in Eq. (52), $\mathcal{L} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R} \to \mathbb{R}$ is the Bolza component of the cost function, i.e., the integral cost in Eq. (52), $\boldsymbol{\psi}_0 \in \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_0}$ are the initial point conditions, $\boldsymbol{\psi}_f \in \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_f}$ are the final point conditions, and $\boldsymbol{g}_L \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R} \to \mathbb{R}^{n_g}$ and $\boldsymbol{g}_U \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R} \to \mathbb{R}^{n_g}$ are the lower and upper bounds on the path constraints.
The basic idea behind the Gauss-Lobatto quadrature discretization is to approximate the vector field by an N th degree Lagrange interpolating polynomial

$$\boldsymbol{f}(t) \approx \boldsymbol{f}_N(t) \tag{58}$$

expanded using values of the vector field at the set of Legendre-Gauss-Lobatto (LGL) points. The LGL points are defined on the interval $\tau \in [-1,1]$ and correspond to the zeros of the derivative of the N th degree Legendre polynomial, $L_N(\tau)$, as well as the end points –1 and 1. The computation time is related to the time domain by the transformation

$$t = \frac{(t_f - t_0)}{2}\tau + \frac{(t_f + t_0)}{2} \tag{59}$$

The Lagrange interpolating polynomials are written as

$$\boldsymbol{f}_N(t) = \sum_{k=0}^{N} \boldsymbol{f}_k \phi_k(\tau) \tag{60}$$

where $t = t(\tau)$ because of the shift in the computational domain. The Lagrange polynomials may be expressed in terms of the Legendre polynomials as

$$\phi_k(\tau) = \frac{\left(\tau^2 - 1\right) L_N'(\tau)}{(\tau - \tau_k) N(N+1) L_N(\tau_k)}, \qquad k = 0,...,N \tag{61}$$

Approximations to the state equations are obtained by integrating Eq. (60),

$$\boldsymbol{x}_k = \boldsymbol{x}_0 + \frac{(t_f - t_0)}{2} \int_{-1}^{1} \sum_{j=0}^{N} \phi_j(\tau) \boldsymbol{f}(t_j)\, \mathrm{d}\tau, k = 1,...,N \tag{62}$$

Eq. (62) can be re-written in the form of Gauss-Lobatto quadrature approximations as

$$\boldsymbol{x}_k = \boldsymbol{x}_0 + \frac{(t_f - t_0)}{2} \sum_{j=0}^{N} \mathcal{I}_{k-1,j} \boldsymbol{f}(t_j), \qquad k = 1,...,N \tag{63}$$

where the entries of the $N \times (N+1)$ integration matrix $\mathcal{I}$ are derived by Williams (2006). The cost function is approximated via a full Gauss-Lobatto quadrature as

$$\mathcal{J}_N = \mathcal{M}[\boldsymbol{x}_N] + \frac{(t_f - t_0)}{2} \sum_{j=0}^{N} \mathcal{L}[\boldsymbol{x}_j, \boldsymbol{u}_j, t_j]\, w_j \tag{64}$$

Thus the discrete states and controls at the LGL points $(\boldsymbol{x}_0,...,\boldsymbol{x}_N,\boldsymbol{u}_0,...,\boldsymbol{u}_N)$ are the optimization parameters, which means that the path constraints and box constraints are easily enforced. The continuous problem has been converted into a large-scale parameter optimization problem. The resulting nonlinear programming problem is solved using SNOPT in this work. In all cases analytic Jacobians of the cost and discretized equations of motion are provided to SNOPT.

Alternatives to utilization of nonlinear optimization strategies have also been suggested. An example of an alternative is the use of iterative linear approximations, where the solution is linearized around the best guess of the optimal trajectory. This approach is discussed in more detail for the pseudospectral method in (Williams, 2004).

5.2 Optimal Control Strategy

Using the notation presented above, the basic notion of the real-time optimal control strategy is summarized in Fig. 2. For a given mission objective, a suitable cost function and final conditions would usually be known a priori. This is input into the two-point boundary value problem (TPBVP) solver, which generates the open-loop optimal trajectories $\boldsymbol{x}^*(t), \boldsymbol{u}^*(t)$. The optimal control input is then used in the real-system, denoted by the "Control Actuators" block, producing the observation vector $\boldsymbol{y}(t_k)$. This is fed into the CKF to produce a state estimate, which is then fed back to update the optimal trajectory by letting $t_0 = t$, and using $t_f - t$ as the time to go.

Imposing hard terminal boundary conditions can make the optimization problem infeasible as $t_f - t \to 0$. In many applications of nonlinear optimal control, a receding horizon strategy is used, whereby the constraints are always imposed at the end of a finite horizon $T = t_f - t$, where T is a constant, rather than at a fixed time. This can provide advantages with respect to robustness of the controller. This strategy, as well as some additional strategies, are discussed below.

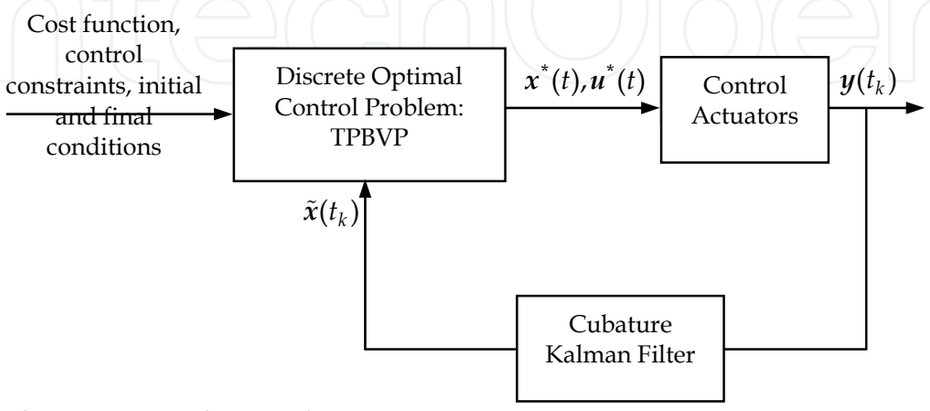


Fig. 2. Real-Time Optimal Control Strategy.

5.3 Issues in Real-Time Optimal Control

Although the architecture for solving the optimal control problem presented in the previous section is capable of rapidly generating optimal trajectories, there are several important issues that need to be taken into consideration before implementing the method. Some of these have already been discussed briefly, but because of their importance they will be reiterated in the following subsections.

5.3.1 Initial Guess

One issue that governs the success of the NLP finding a solution rapidly is the initial guess that is provided. Although convergence of SNOPT can be achieved from random guesses (Ross & Gong, 2008), the ability to converge from a bad guess is not really of significant benefit. The main issue is the speed with which a feasible solution is generated as a function of the initial guess. It is conceivable for many scenarios that good initial guesses are available. For example, for tethered satellite systems, deployment and retrieval would probably occur from fixed initial and terminal points. Therefore, one would expect that this solution would be readily available. In fact, in this work, it is assumed that these "reference" trajectories have already been determined. Hence, each re-optimization would take place with the initial guess provided from the previous solution, and the first optimization would take place using the stored reference solution. In most circumstances then, the largest disturbance or perturbation would occur at the initial time, where the initial state may be some "distance" from the stored solution. Nevertheless, the stored solution is still a "good" guess for optimizing the trajectory. This essentially means that the study of the computational performance should be focused on the initial sample, which would conceivably take much longer than the remaining samples.

5.3.2 Issues in Updating the Control

For many systems, the delay in computing the new control sequences is not negligible. Therefore, it is preferable to develop methods that adequately deal with the computational delay for the general case. The simplest way of updating the control input is illustrated in Fig. 3. The method uses only the latest information and does not explicitly account for the time delay. At the time $t = t_i$, a sample of the system states is taken $x(t_i)$. This information is used to generate a new optimal trajectory $x_i(t), u_i(t)$. However, the computation time required to calculate the trajectory is given by $\Delta t_i = t_{i+1} - t_i$. During the delay, the previous optimal control input $u_{i-1}(t)$ is applied. As soon as the new optimal control is available it is applied (at $t = t_{i+1}$). However, the new control contains a portion of time that has already expired. This means that there is likely to be a discontinuity in the control at the new sample time $t = t_{i+1}$. The new control is applied until the new optimal trajectory, corresponding to the states sampled at $x(t_{i+1})$, is computed. At this point, the process repeats until $t = t_f$. Note that although the updates occur in discrete time, the actual control input is applied at the actuator by interpolation of the reference controls.

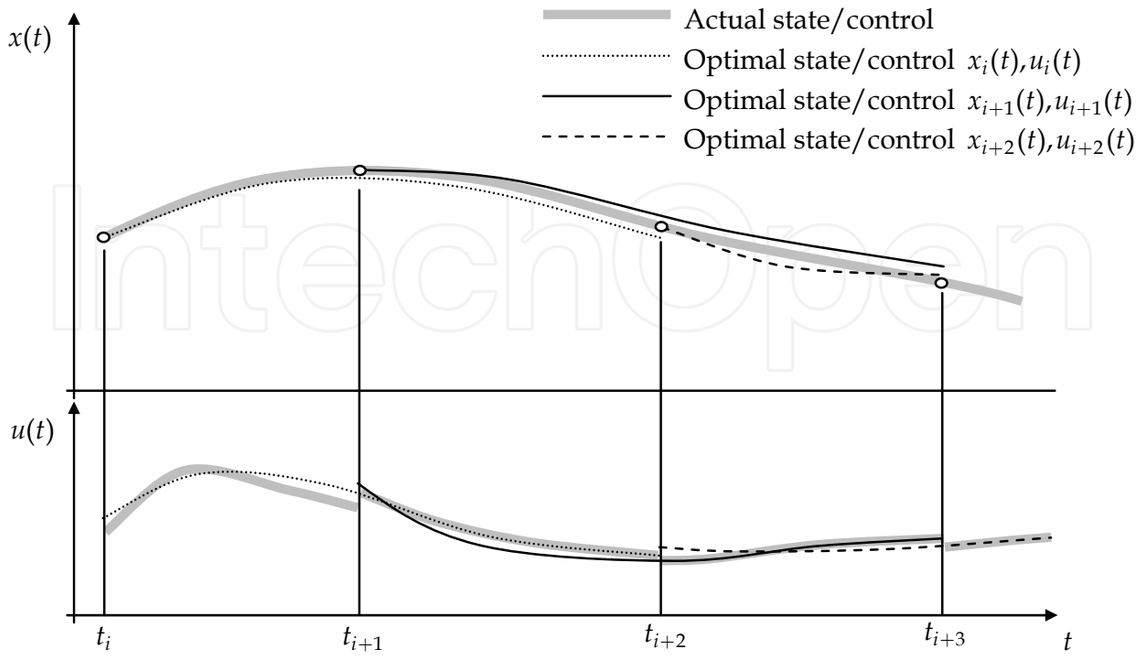


Fig. 3. Updating the Optimal Control using Only Latest Information.

Due to sensor noise and measurement errors, the state sampled at the new sample time $x(t_{i+1})$ is unlikely to correspond to the optimal trajectory that is computed from $x_i(t_{i+1})$. Therefore, in this approach, it is possible that the time delay could cause instability in the algorithm because the states are never matching exactly at the time the new control is implemented. To reduce the effect of this problem, it is possible to employ model prediction to estimate the states. In this second approach, the sample time is not determined by the time required to compute the trajectory, but is some prescribed value. The sampling time must be sufficient to allow the prediction of the states and to solve the resulting optimal control problem, t_{sol}. Hence, $\Delta t_i > t_{\text{sol}}$. The basic concept is illustrated in Fig. 4. At time $t = t_i$, a system state measurement is made $x(t_i)$. This measurement, together with the previously determined optimal control and the system model, allows the system state to be predicted at the new sample time $t = t_{i+1}$,

$$\hat{x}(t_{i+1}) \approx x(t_i) + \int_{t_i}^{t_{i+1}} \dot{x}(u_i(t))\, \mathrm{d}t \tag{65}$$

The new optimal control is then computed from the state $\hat{x}(t_{i+1})$. When the system reaches $t = t_{i+1}$, the new control signal is applied, $u_{i+1}(t)$. At the same time, a new measurement is taken and the process is repeated. This process is designed to reduce instabilities in the system and to make the computations more accurate. However, it still does not prevent discontinuities in the control, which for a tethered satellite system could cause elastic vibrations of the tether. One way to produce a smooth control signal is to constrain the initial value of the control in the new computation so that

$$u_{i+1}(t_{i+1}) = u_i(t_{i+1}) \tag{66}$$

That is, the initial value of the new control is equal to the previously computed control at time $t = t_{i+1}$. It should be noted that the use of prediction assumes coarse measurement updates from sensors. Higher update rates would allow the Kalman filter to be run up until the control sampling time, achieving the same effect as the state prediction (except that the prediction has been corrected for errors). Hence, Fig. 4 shows the procedure with the predicted state replaced by the estimated state.

5.3.3 Implementing Terminal Constraints

In standard model predictive control, the future horizon over which the optimal control problem is solved is usually fixed in length. Thus, the implementation of terminal constraints does not pose a theoretical problem because the aim is usually for stability, rather than hitting a target. However, there are many situations where the final time may be fixed by mission requirements, and hence as $t_f - t \to 0$ the optimal control problem becomes more and more ill-posed. This is particularly true if there is a large disturbance near the final time, or if there is some uncertainty in the model. Therefore, it may be preferable to switch from hard constraints to soft constraints at some prespecified time $t = t_{\text{crit}}$, or if the optimization problem does not converge after n_{crit} successive attempts. It is important to note that if the optimization fails, the previously converged control is used until a new control becomes available. Therefore, after n_{crit} failures, soft terminal constraints are used under the assumption that the fixed terminal conditions can not be achieved within the control limits. The soft terminal constraints are defined by

$$\mathcal{M} = \tfrac{1}{2}\left[\, \boldsymbol{x}(t_f) - \boldsymbol{x}_f \,\right]^{\top} \boldsymbol{S}_f \left[\, \boldsymbol{x}(t_f) - \boldsymbol{x}_f \,\right] \tag{67}$$

The worst case scenario is for fixed time missions. However, where stability is the main issue, receding horizon strategies with fixed horizon length can be used. Alternatively, the time to go can be used up until $t = t_{\text{crit}}$, at which point the controller is switched from a fixed terminal time to one with a fixed horizon length defined by $T = t_f - t_{\text{crit}}$. In this framework, the parameters t_{crit} and n_{crit} are design parameters for the system.

It should also be noted that system requirements would typically necessitate an inner-loop controller be used to track the commands generated by the outer loop (optimal trajectory generator). An inner-loop is required for systems that have associated uncertainty in modeling, control actuation, or time delays. In this chapter, the control is applied completely open-loop between control updates using a time-based lookup table. The loop is closed only at coarse sampling times.

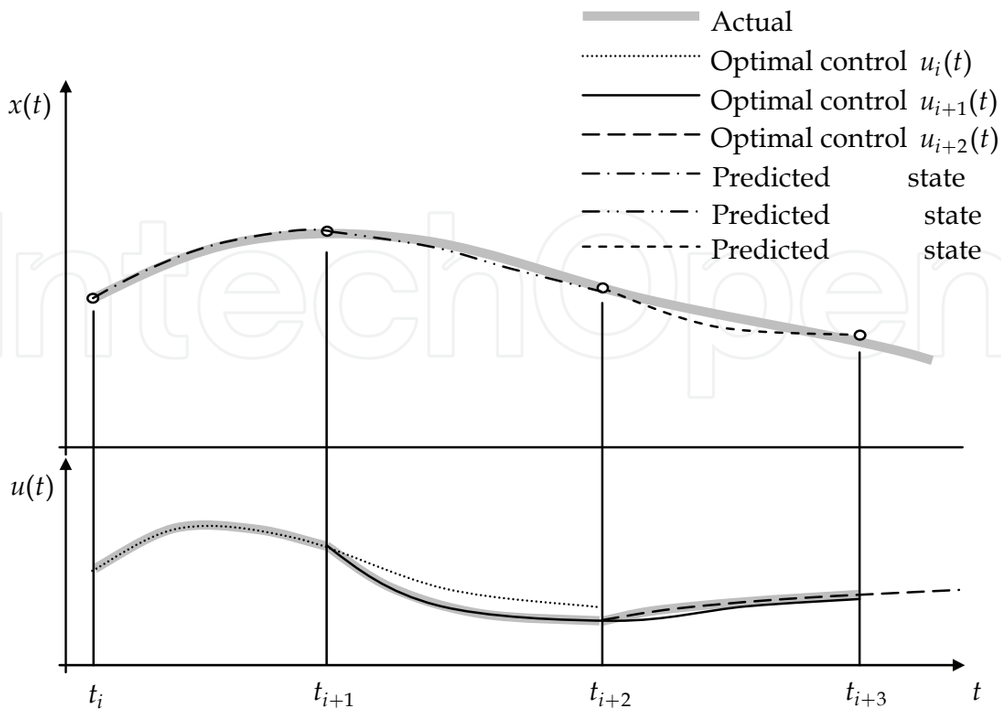


Fig. 4. Updating the Optimal Control with Prediction and Initial Control Constraint.

5.4 Rigid Model In-Loop Tests

To explore the possibilities of real-time control for tethered satellite systems, a simple, but representative test problem is utilized. Deployment and retrieval are two benchmark problems that provide good insight into the capability of a real-time controller. Williams (2008) demonstrated that deployment and retrieval to and from a set of common boundary conditions leads to an exact symmetry in the processes. That is, for every optimal deployment trajectory to and from a set of boundary conditions, there exists a retrieval trajectory that is mirrored about the local vertical. However, it is also known that retrieval is unstable, in that small perturbations near the beginning of retrieval are amplified, whereas small perturbations near the beginning of deployment tend to remain bounded. Therefore, to test the effectiveness of a real-time optimal controller, the retrieval phase is an ideal test case.

The benchmark problem is defined in terms of the nondimensional parameters as: Minimize the cost

$$J = \int_{t_0}^{t_f} (\Lambda'')^2 \, \mathrm{d}t \tag{68}$$

subject to the boundary conditions

$$[\theta, \theta', \Lambda, \Lambda']_{t=t_0} = [0,0,1,0], \qquad [\theta, \theta', \Lambda, \Lambda']_{t=t_f} = [0,0,0.1,0] \tag{69}$$

and the tension control inequality

$$0.01 \leq u \leq 4 \tag{70}$$

which is designed to prevent the tether from becoming slack, and to prevent the tether from severing. The control input for this test case is defined as $u = T / [m_1 \dot{\nu}^2 L_r (m_2 + m_t) / m]$.

5.4.1 Preliminary Study on Computation Time

To gauge the effectiveness of performing computations of the optimal control in real-time, the problem of tether retrieval was solved using cold-starts with random perturbations to the initial conditions. Since the computation of the control is most critical at the initial time (because the initial state may be very far from the reference state), a numerical study of the performance of the solution algorithm was run for 1000 computations. In terms of actual implementation, if the sampling time is short enough, subsequent convergence is almost always quicker than the initial computation.

The retrieval problem is posed in nondimensional units, with a nondimensional time of 6 rad. For a tether system in low Earth orbit at an altitude of 500 km, the total maneuver time is roughly 5450 sec. The update time with a good guess of the trajectory averages 0.09 sec in MATLAB 2009a on a Core 2 Processor running Windows XP. Clearly, this easily allows real-time computation of the trajectory with over 50000 samples. However, as noted, the critical time is the first update when the trajectory may be far from the reference or when a good initial guess may not be available. A study of 1000 computations with different initial conditions, but with the same infeasible guess for the trajectory was performed. The initial conditions were distributed randomly in the ranges $|\delta\theta(0)| \leq 0.2$ rad, $|\delta\theta'(0)| \leq 0.1$, and $|\delta\Lambda(0)| \leq 0.02$. Fig. 5 shows a summary of the results from these computations. The level of discretization was set to be N = 30 for this study. The mean computation time was determined to be 0.164 sec.

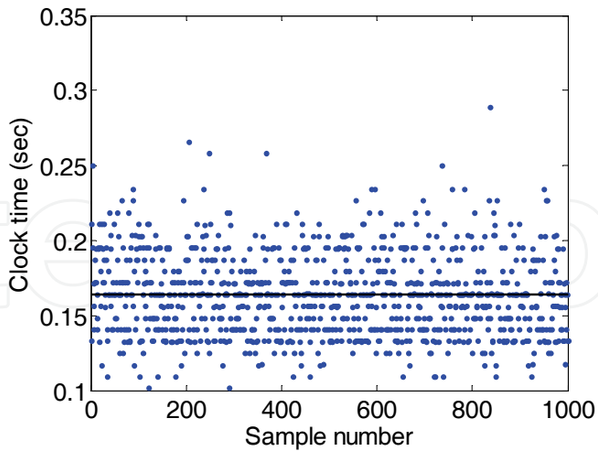


Fig. 5. Summary of Results from Study of Computation of Optimal Trajectories.

The minimum time was 0.102 sec and the maximum time was 0.290 sec. Even in the worst possible case, it would still be possible to implement a sampled-data feedback controller (using MATLAB) with roughly 18000 samples. It should also be noted that convergence was achieved in every case. The CPU time as calculated in Windows represents the worst

case that could be achieved using a dedicated embedded system. The Windows scheduler can schedule the control process in- and out- at different times. The resolution of the scheduler can be seen in the discrete banding of the mean CPU time in Fig. 5, rather than completely random times.

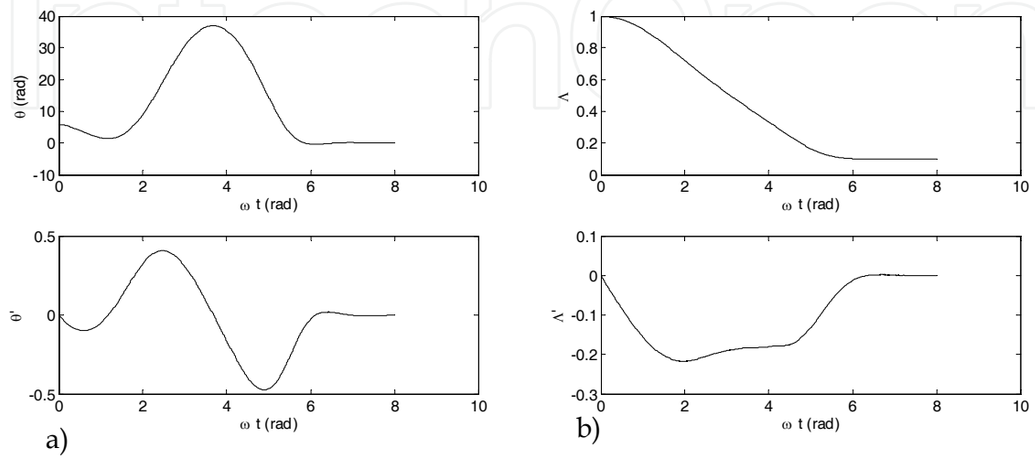
5.4.2 Closed-Loop Control

To examine the actual performance of the controller for dealing with disturbances, the control model is used with external perturbations included via the Q_θ and Q_Λ terms in the equations of motion. For simplicity, the perturbations are generated randomly such that $|Q_\theta| \leq 0.05$ and $|Q_\Lambda| \leq 0.05$. This corresponds to disturbances on the subsatellite on the order of several Newtons, whose actual values depend on the system geometry. The number of major iterations was limited to 50.

The terminal weighted matrix is selected as $\boldsymbol{S}_f = \text{diag}[100,100,100,100]$, and the controller is switched at 4 rad from hard terminal constraints to soft constraints. Numerical results are shown in Fig. 6. Fig. 6a and 6b shows that the terminal constraints are met reasonably accurately, despite not being enforced with hard constraints. The mean CPU time for the whole trajectory is 0.159 sec, the standard deviation is 0.0744 sec, the minimum time is 0.04 sec, and the maximum time is 1.442 sec. Prior to the change in controller, the mean CPU time is 0.1265 sec, whereas after the change the mean CPU time increases to 0.223 sec. Therefore, the smooth control input in the terminal phases of the trajectory comes at the expense of a 76% increase in mean computation time. This is still well within the sampling time of the controller.

6. Closed-Loop Control in Simulation Environment

The results presented in the previous section utilized tension as the control input. Tension has been widely used as the control input in the literature, but it has several drawbacks. It introduces long-term errors in the trajectories because of inaccuracies in the system properties, errors in the gravity model, and tether oscillations. A better choice is to control the reel speed or rate of change of reel speed. In the high fidelity simulation environment, the control is implemented as the rate of change of nondimensional reel rate.



a) b)

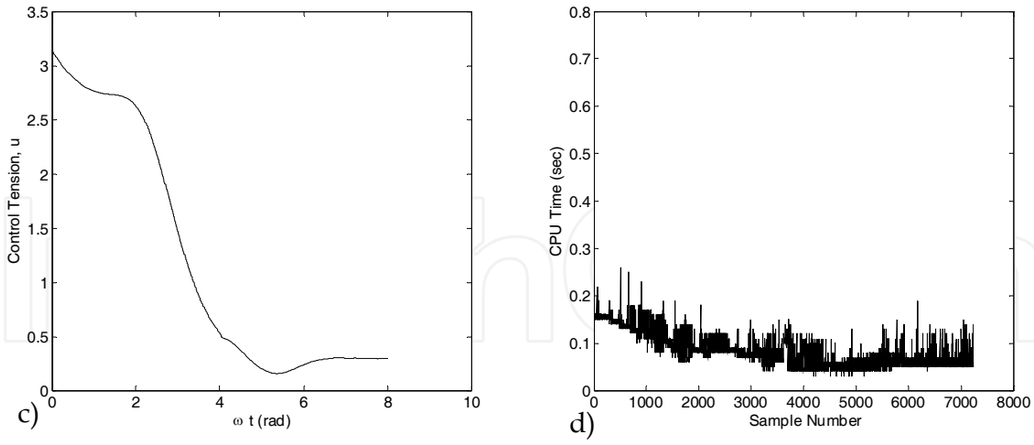


Fig. 6. Real-Time Computation of Retrieval Trajectory with 1 sec Sampling Time, Receding Horizon after $\omega t = 4$ rad and Model Prediction of States with Continuous Control Enforced, a) Libration Dynamics, b) Length Dynamics, c) Control Tension, d) Computation Time.

6.1 Simulation Environment

The simulation environment used for testing the closed-loop control behavior is built in Simulink™, which is itself based on the MATLAB environment. Simulink provides a graphical approach for modeling and control of complex systems. It has the distinct advantage of being able to provide generated C-code targeting real-time operation directly from the underlying model. This feature requires additional supporting tools available from Mathworks. In the context of the current chapter, a Simulink model is used to simulate four distinct elements of the system. Fig. 7 illustrates the interconnections of the four system elements. These are: 1) Variable-Step, Multibody Propagation (bead tether model), 2) Sensor models, 3) Tether state estimation, and 4) Pseudospectral predictive control. One of the complicating factors in simulating the predictive control system is that a high-fidelity, variable step integration algorithm is needed to propagate the multibody dynamic equations.

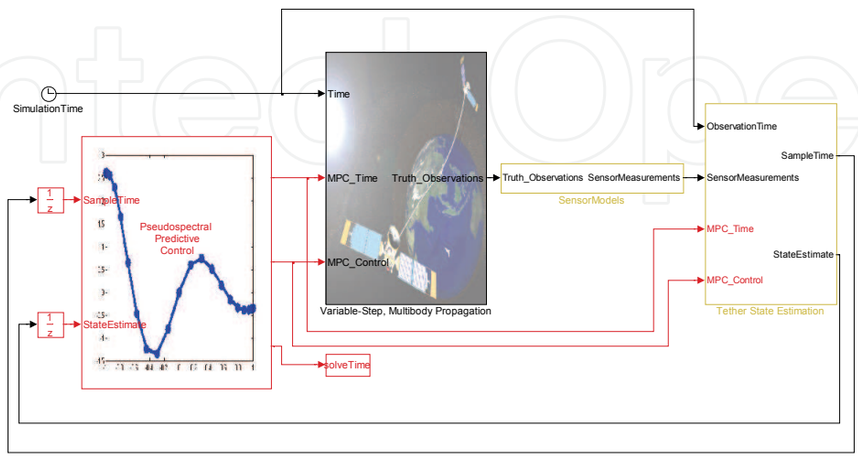
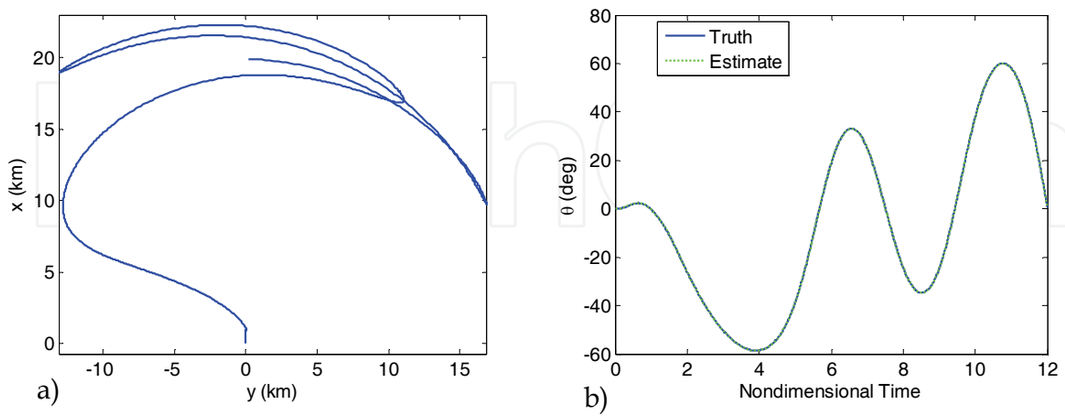


Fig. 7. Simulink simulation model for closed-loop model predictive control.

Although Simulink supports variable-step integration algorithms, it does not easily allow for the combination of variable-step integration and discrete sampling updates of the system being propagated. For example, the multibody model requires regular checks on the length of the deploying segment for the introduction or removal of an element from the model. To overcome this, a custom S-function block is used which employs the LSODA variable-step integration library. The LSODA library is coded in Fortran, but was ported to C via f2c.
The sensor models block implements the tension and GPS models for the system. The tether state estimation block implements the Kalman filter for estimating the tether state in a discrete-time manner. Finally, the pseudospectral predictive control block implements the predictive controller.

6.2 Example: Closed-Loop Control with State Estimator

One of the future applications of tethered satellite systems is for capture and rendezvous of a satellite in a coplanar orbit. In such an application, timing is critical for mission success. A similar application where timing is not as critical is the deorbit of a payload, similar to the idea of the YES-2 mission. In these examples, the control objective is similar in that it requires the generation of a large in-plane swinging motion. As an example, the control objective of rendezvous with a target satellite is used. The rendezvous conditions have been derived in detail by Williams (2006) for the general case of circular and elliptical orbits as a function of tether length.
The objective in this section is to deploy the tether from a length of 1 km to a length of 20 km to achieve a nondimensional in-plane libration rate of -1.5. For a target satellite in a circular orbit, the reel-rate at capture must be zero. The cost function that aids in minimizing tether oscillations is given in Eq. (68). The tether mass density is 1 kg/km, the subsatellite mass is 200 kg, and the orbit radius is 500 km. The tension measurement noise is 0.5 N, the reel-rate noise is 0.05 m/s, and the GPS error terms noise are R_{GPS} = 25 m², τ_{GPS} = 0.01 (nondimensional). Solutions are obtained using N = 30, with a fixed sampling time of 0.01 rad ≈ 9 sec. The final time is set at 12 rad in nondimensional units.



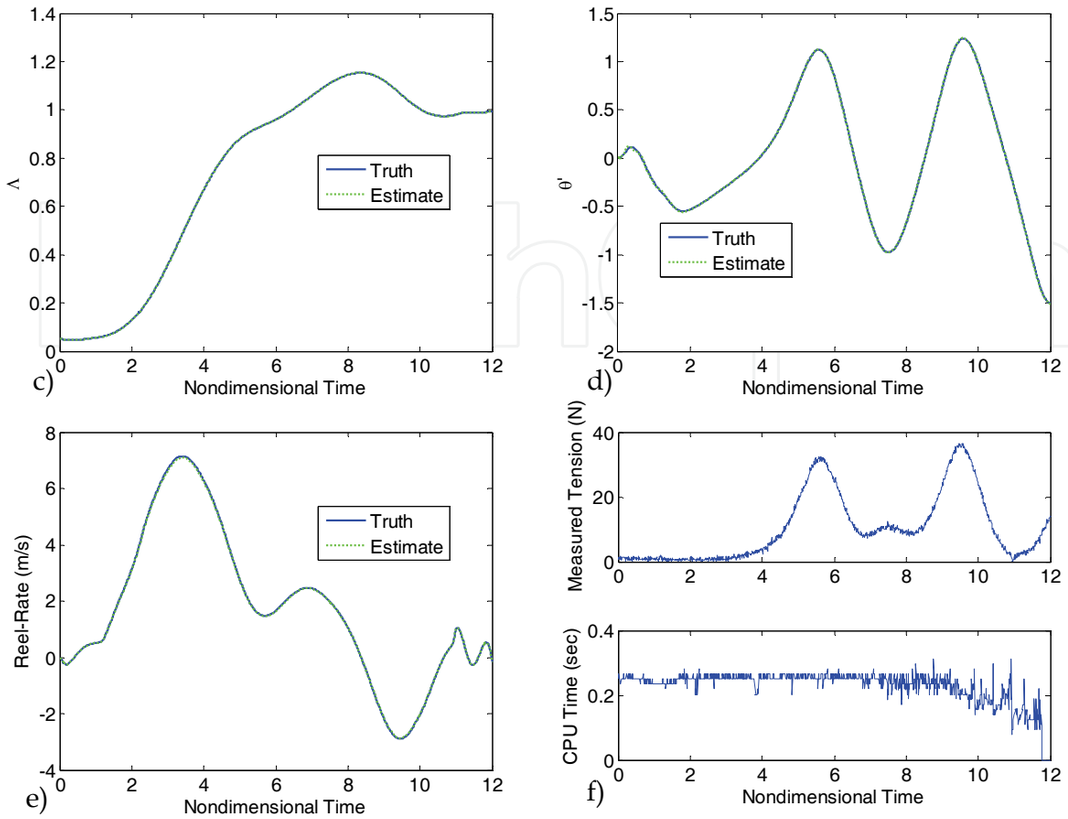


Fig. 8. Closed-loop optimal control of tethered satellite system, a) Tether tip trajectory, b) In-plane libration angle, c) Nondimensional tether length, d) Nondimensional libration rate, e) Reel-rate, f) Measured tension and computation time.

Fig. 8 shows the results of a closed-loop simulation in Simulink using the multibody tether model in combination with the CKF. The results show that the tether is initially over-deployed by about 20%, then reeled back-in to generate the swing velocity required for capture. The final conditions are met to within a fraction of a percent in all state variables despite the measurement errors and uncertainties. The peak reel-rate is approximately 7 m/s, and the variation in reel-rate is smooth throughout the entire maneuver. The average CPU time is 0.23 sec, peaking to 0.31 sec.

7. Conclusion

Modern computing technology allows the real-time generation of optimal trajectories for tethered satellites. An architecture that implements a closed-loop controller with a nonlinear state estimator using a subset of available measurements has been demonstrated for accurately deploying a tether for a rendezvous application. This strategy allows the controller to adapt to large disturbances by recalculating the entire trajectory to satisfy the mission requirements, rather than trying to force the system back to a reference trajectory computer offline.

8. References

Barkow, B.; Steindl, A.; Troger, H. & Wiedermann, G. (2003). Various methods of controlling the deployment of a tethered satellite. *Journal of Vibration and Control*, Vol. 9, 187-208.

Blanksby, C. & Trivailo, P. (2000). Assessment of actuation methods for manipulating tip position of long tethers. *Space Technology*, Vol. 20, No. 1, 31-39.

Colombo, G.; Gaposchkin, E. M.; Grossi, M. D. & Weiffenbach G. C. (1975). The 'skyhook': a shuttle-borne tool for low-orbital-altitude research. *Meccanica*, March, 3-20.

Dunbar, W. B.; Milam, M. B.; Franz, R. & Murray, R. M. (2002). Model predictive control of a thrust-vectored flight control experiment. 15th IFAC World Congress on Automatic Control, Barcelona, Spain.

Elnagar, G.; Kazemi, M. A. & Razzaghi, M. (1995). The pseudospectral legendre method for discretizing optimal control problems. *IEEE Transactions on Automatic Control*, Vol. 40, No. 10, 1793-1796.

Fujii, H. & Ishijima, S. (1989). Mission function control for deployment and retrieval of a subsatellite. *Journal of Guidance, Control, and Dynamics*, Vol. 12, No. 2, 243-247.

Fujii, H. A. & Anazawa, S. (1994). Deployment/retrieval control of tethered subsatellite through an optimal path. *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 6, 1292-1298.

Fujii, H.; Uchiyama, K. & Kokubun, K. (1991). Mission function control of tethered subsatellite deployment/retrieval: In-plane and out-of-plane motion. *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 2, 471-473.

Gill, P. E.; Murray, W. & Saunders, M. A. (2002). SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Journal on Optimization*, Vol. 12, No. 4, 979-1006.

Arasaratnam, I. & Haykin, S. (2009). Cubature kalman filters. *IEEE Transactions on Automatic Control*, Vol. 54, 1254-1269.

Kim, E. & Vadali, S. R. (1995). Modeling issues related to retrieval of flexible tethered satellite systems. *Journal of Guidance, Control, and Dynamics*, Vol. 18, 1169-1176.

Kruijff, M.; van der Heide, E. & Ockels, W. (2009). Data analysis of a tethered spacemail experiment. *Journal of Spacecraft and Rockets*, Vol. 46, No. 6, 1272-1287.

Lakso, J. & Coverstone, V. L. (2000). Optimal tether deployment/retrieval trajectories using direct collocation. *AIAA/AAS Astrodynamics Specialist Conference*, 14-17 Aug. 2000, AIAA Paper 2000-4349.

Lorenzini, E. C.; Bortolami, S. B.; Rupp, C. C. & Angrilli, F. (1996). Control and flight performance of tethered satellite small expendable deployment system-II. *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 5, 1148-1156.

Misra, A. K. & Modi, V. J. (1982). Deployment and retrieval of shuttle supported tethered satellites. *Journal of Guidance, Control, and Dynamics*, Vol. 5, No. 3, 278-285.

Misra, A. K. & Modi, V. J. (1986). A Survey on the dynamics and control of tethered satellite systems. *Advances in the Astronautical Sciences*, Vol. 62, 667-719.

Nordley, G. D. & Forward, R. L. (2001). Mars-earth rapid interplanetary tether transport system. I – Initial feasibility analysis. *Journal of Propulsion and Power*, Vol. 17, No. 3, 499-507.

Ross, I. M. & Fahroo, F. (2003). Legendre pseudospectral approximations of optimal control problems. *Lecture Notes in Control and Information Sciences*, Vol. 295, 327-342.

Ross, I. M. & Gong, Q. (2008). Guess-free trajectory optimization. *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, August, Honolulu.

Rupp, C. C. (1975). A tether tension control law for tethered subsatellites deployed along the local vertical. *NASA TM X-64963*, Marshall Space Flight Center, Alabama.

Vadali, S. R. & Kim, E. S. (1991). Feedback control of tethered satellites using lyapunov stability theory. *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 4, 729-735.

Wiedermann, G.; Schagerl, M.; Steindl, A. & Troger, H. (1999). Simulation of deployment and retrieval processes in a tethered satellite system mission. Paper presented at the International Astronautical Congress, Amsterdam, The Netherlands, Oct.

Williams, P. & Blanksby, C. (2008). Optimal prolonged spacecraft rendezvous using tethers. *International Review of Aerospace Engineering*, Vol. 1, No. 1, 93-103.

Williams, P. (2004). Application of pseudospectral methods for receding horizon control. *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 2, 310-314.

Williams, P. (2004). Guidance and control of tethered satellite systems using pseudospectral methods. *AAS/AIAA Spaceflight Mechanics Meeting*, Wailea, Hawaii, Feb., Paper AAS 04-169.

Williams, P. (2005). Receding horizon control using gauss-lobatto quadrature approximations. *AAS/AIAA Astrodynamics Specialist Conference*, Aug. 7-11, Embassy Suites Hotel, Lake Tahoe Resort, Paper AAS 05-349.

Williams, P. (2006). Dynamics and control of spinning tethers for rendezvous in elliptic orbits. *Journal of Vibration and Control*, Vol. 12, No. 7, 737-771.

Williams, P. (2006). A gauss-lobatto quadrature approach for solving optimal control problems. *ANZIAM Journal* (E), Vol. 47, July, C101-C115.

Williams, P. (2008). Optimal deployment/retrieval of tethered satellites. *Journal of Spacecraft and Rockets*, Vol. 45, No. 2, 324-343.

Williams, P.; Blanksby, C. & Trivailo, P. (2004). Tethered planetary capture maneuvers. *Journal of Spacecraft and Rockets*, Vol. 41, No. 4, 603-613.

Williams, P.; Hyslop, A.; Stelzer, M. & Kruijff, M. (2008) YES2 optimal trajectories in presence of eccentricity and aerodynamic drag. *Acta Astronautica*, to be published.

Xu, D. M.; Misra, A. K. & Modi, V. J. (1981). Three dimensional control of the shuttle supported tethered satellite systems during deployment and retrieval. *Proceedings of the 3rd VPISU/AIAA Symposium on Dynamics and Control of Large Flexible Spacecraft*, Blacksburg, VA, 453-469.

Xu, D. M.; Misra, A. K. & Modi, V. J. (1986). Thruster-augmented active control of a tethered subsatellite system during its retrieval. *Journal of Guidance, Control, and Dynamics*, Vol. 9, 663-672.



Model Predictive Control

Edited by Tao Zheng

ISBN 978-953-307-102-2

Hard cover, 304 pages

Publisher Sciyo

Published online 18, August, 2010

Published in print edition August, 2010

Frontiers of Model Predictive Control Robust Model Predictive Control Nonlinear Model Predictive Control Excellent Applications Guide for Researchers and Engineers Recent Achievements of Authors over the World Theory with Practical Examples Kinds of Algorithms for Choice

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Paul Williams (2010). Predictive Control of Tethered Satellite Systems, Model Predictive Control, Tao Zheng (Ed.), ISBN: 978-953-307-102-2, InTech, Available from: http://www.intechopen.com/books/model-predictive-control/predictive-control-of-tethered-satellite-systems

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License, which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen