LFT modelling for the analysis of relative motion controllers in eccentric orbits

Andrés Marcos, Luis F. Peñín and Emanuele Di Sotto

Abstract— **Currently, circular orbit controllers are being developed for rendezvous & docking (RVD) and have acquired a high level of maturity. Nevertheless, the new exploration missions being envisioned, and for which elliptical orbit RVD presents appealingly advantages, indicate the need to explore the validity of these proven circular-orbit controllers for elliptical scenarios as well as the possibility of designing advanced elliptical RVD systems. In order to do this, it is essential to show that modern control techniques and tools (most based on the linear fractional transformation paradigm) developed for time/parameter varying and uncertain systems are mature and relatively straight-forward to use by the space industry. In this article, it is shown in a tutorial manner how these LFT techniques and tools are well suited to address all the pertinent issues while presenting the validation of a circular-orbit controller for elliptical orbits.**

Notation

μ	Gravitational constant (Mars)
$\omega/\dot{\omega}$	Angular orbital velocity/rate
θ	True anomaly
a	Semi-major axis
$a_{x/y/z}$	X/Y/Z-axis thrust acceleration commands
h	Orbital angular momentum
k	Constant, $k = \mu/h^{1.5}$
n	Mean motion, $n = \sqrt{mu/a^3}$
$x/\dot{x}/\ddot{x}$	Relative position/velocity/acceleration

I. Introduction

Rendezvous & Docking (RVD) technologies have been studied and developed in Europe from the early 80s and have achieved full maturity with the recent successful launch and operation of ESA Automated Transfer Vehicle (ATV). The next challenge for RVD technology is found in the future exploration missions which will require autonomous GNC systems in order to carry out the precise complex operations envisioned.

In the context of ESA-Aurora program DEIMOS Space has developed a preliminary vision-based RVD GNC system using as reference mission the Mars Sample Return (MSR), where a circular orbit around Mars is proposed for the rendezvous of the Orbiter and the Mars Ascent Vehicle (MAV) vehicles. The Orbiter and the MAV will carry out autonomous rendezvous operations in this orbit to transfer the collected sample from one vehicle to the other. The choice of a circular orbit has been driven by existing European technology for autonomous RVD. However, it appears worth investigating the elliptical orbit scenario as it represents a very appealing solution due to the possibility of saving considerably propellant mass during RVD manoeuvres. For example, two of the most critical Orbiter manoeuvres identified as benefiting from this saving are the Target Orbit Acquisition (TOA) and the Mars Escape (ME) manoeuvres. Similarly, the rendezvous approach phase –where the most demanding manoeuvre brings the two vehicles into coplanar orbits– is also expected to result in substantial propellant savings if performed in an elliptical orbit.

The downside of elliptical orbit RVD is the new challenge it presents in terms of relative motion controller design and analysis as it has to deal with time/parameter varying and uncertain systems. Techniques developed specifically for this type of systems exists [2], [12] but are very slowly being introduced in space due to their associated steep learning curve and, until recently, lack of appropriate tools. In this respect, it is essential to show that these advanced techniques are applicable to the validation of circular-orbit controllers in elliptical orbits as well as for the design and analysis of advanced elliptical-orbit controllers. To accomplish this, it is required to use the LFT modelling paradigm which involves both using proper tools for the manipulation as well as using the proper formulation of the system.

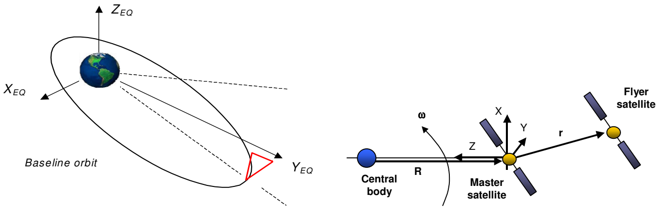
In the aeronautic domain there are many published works dealing with both issues [8], [13], [3]. In the space (RVD) domain, there are very well-known works [11], [15], [7] dealing with the proper formulation of the relative motion of spacecraft in elliptical orbits. But, it is scarce on the appropriate formulation and use (design and analysis) with modern robust techniques, except for the notable exceptions of [5], [14]. The first examined the LFT formulation of the relative dynamics (using Lawden's formulation and including J2 Earth oblateness perturbation), the resulting LFTs were of too high-order to be useful for design but represent a very nice example on the use of LFT for this type of systems. In the second reference, a controller was designed using $D-K$ iteration synthesis and the same relative motion equations but considering eccentricity and semi-major axis as parametric uncertainty and J2 perturbations and actuator/sensor noise as functional uncertainty yielding a more simpler but also more conservative LFT model. Both articles require in-depth understanding on LFT manipulation or LFT tools usage.

This article intention is to present modern control techniques and tools, such as the structured singular value analysis and LFT modelling, in a tutorial manner from a practical stand-point while at the same time validating a circular-orbit RVD controller for elliptical-orbit missions.

Andrés Marcos and Luis F. Peñín are with Deimos Space S.L., Spain - Emanuele Di Sotto is with Deimos Engenharia, Portugal {andres.marcos,luis-felipe.penin}@deimos-space.com, emanuele.disotto@deimos.com.pt

II. Problem Definition

The selected elliptical orbit for control design evaluation has a semi-major axis of 5,543 Km and an eccentricity of 0.1869 around Mars. The mass of each satellite is 1454 kg with moments of inertia computed assuming a uniform distribution of the mass.



(a) Nominal Orbit.

(b) Formation LVLH.

Fig. 1. Depiction of the nominal orbit and a two-spacecraft formation in LVLH reference frame.

Figure 1 shows a pictorial representation of the orbit and the so-called local-vertical-local-horizon (LVLH) reference frame in the formation. The LVLH frame is the standard coordinate frame used to mathematically model relative spacecraft motion around a central body. It is centred in one of the formation satellites (denoted master or leader), and with respect to which the motion of the rest of the satellites (fliers or followers) is given. The following convention is adopted in defining the LVLH: radial relative motion (Z axis) is in the nadir direction (e.g. from master to central body), out-of-plane motion (Y axis) is in the opposite direction to that given by ω and the in-plane motion (X-axis) completes the right-hand frame.

Equations of relative motion in a central body gravity field are readily available in the literature, especially in the area of guidance and control for rendezvous & docking operations [4], [15]. They typically assume that the distance between satellites is relatively small compared to the distance to the central body. After some manipulation the following general equation, valid for any type of orbit, describing the relative motion of two spacecrafts in the LVLH reference frame is obtained [15]:

$$\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix} = \begin{bmatrix}a_x\\ a_y\\ a_z\end{bmatrix} + \begin{bmatrix}0 & 0 & 2\omega\\ 0 & 0 & 0\\ -2\omega & 0 & 0\end{bmatrix}\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\end{bmatrix} + \begin{bmatrix}\omega^2 - k\omega^{3/2} & 0 & \dot{\omega}\\ 0 & -k\omega^{3/2} & 0\\ -\dot{\omega} & 0 & \omega^2 + 2k\omega^{3/2}\end{bmatrix}\begin{bmatrix}x\\ y\\ z\end{bmatrix} \tag{1}$$

A relative position controller expressed in the LVLH frame is designed for forced approach and station keeping control. It assumes uncoupled position control mode (i.e. the rotational and translational control are completely independent) and thus their designs can be performed independently. Classical SISO controller design techniques, i.e. PD using specifications on desired natural frequency and damping responses, are used for each axis yielding a diagonal 3x6 gain matrix whose input vector is the relative LVLH position/velocity and the output vector the acceleration thrust commands $[a_x\, a_y\, a_z]$. This controller was designed and validated for circular orbits.

III. LFT Atoms

In order to apply modern robust analysis and design techniques [16], the LFT modelling paradigm must be used. This involves using proper tools for the manipulation as well as using the proper formulation of the system. This section is devoted to the first, showing the availability, ease of use and flexibility of such tools for LFT modelling.

In terms of availability, the recent appearance of a set of LFT toolboxes [8], [13], [6], [10] has shown the interest of applied research and industrial groups in these techniques while providing with the fundamental required tools – we specially highlight the publicly available toolbox from Jean-Francois Magni [8] which easily represents the most complete toolbox for LFT manipulations.

Rather than representing the closed-loop system into one highly complex equation and then attempting to transform it into an LFT, it is always better to consider the independent subsystems and then trying to capture as LFTs the essential forming elements [3]. We will refer to these fundamental building blocks as LFT atoms (borrowing the expression from [1]). The LFT atoms can be used quite straight forwardly in an iterative manner to represent the more complex components until the full system is modelled.

A very important consideration for the LFT atoms, and in general for any LFT modelling, is to not perform parameter normalization until the complete LFT is obtained. In this manner, since a diagonal structure of the LFT is always obtained at system level the dimension of the LFT is not increased by normalization (the dimension is critical due to current design and analysis algorithms' limitations). An aeronautical example of the dimension increase from normalizing the parameters before LFT modelling can be seen in [9].

The closed-loop is assumed to be represented by the relative spacecraft motion from equation 1 ($RelMot$) closed in feedback with the relative position controller ($Position_K$) and further assuming loss of efficiency and misalignment in all sensor and actuator channels. Furthermore, a parametric mass parameter is used to capture the uncertainty associated with the satellite actual mass (due to ergol consumption for example). Thus, three main LFT atoms can be considered: loss of efficiency (LOE), misalignment factor (MIS) and mass uncertainty (MASS).

A. Misalignment LFT atom

Misalignment is a very relevant problem in spacecraft. For example, thrusters can slightly jitter during deployment and their effective position be 'misaligned' with respect to the designed mounting matrix. Equation (2) shows the mathematical representation of this effect, ε_{ij} indicates misalignment on the i-axis by the j-axis thrust force:

$$\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}_{ideal}\begin{bmatrix}T_x\\ T_y\\ T_z\end{bmatrix} = \begin{bmatrix}1 & \varepsilon_{xy} & \varepsilon_{xz}\\ \varepsilon_{yx} & 1 & \varepsilon_{yz}\\ \varepsilon_{zx} & \varepsilon_{zy} & 1\end{bmatrix}_{MIS}\begin{bmatrix}T_x\\ T_y\\ T_z\end{bmatrix} \tag{2}$$

Similarly, measuring suites such as the four-to-three gyro sensing units typically found in satellites can suffer of misalignment arising from sensor defects yielding defective measures. Two very simple possibilities to model the MIS LFT are available. The first, uses symbolic parameters and the command *sym2lfr* from [8] (a detail description of the functions can be found in the corresponding toolbox manual):

$$>> \ MIS = sym('MIS','real')$$
$$>> \ MIS_{lft} = sym2lfr(MIS, MIS, 0)$$

The second possibility just uses the command *lfrs* from the toolbox: $>> lfrs \ MIS_{lft}$

B. Loss of efficiency LFT atom

The LOE LFT atom is modelled as an additive model, $y = u + \delta * u = (1+\delta) * u$, which allows capturing any type of efficiency loss in the selected channel (and thus, allowing also modelling of faults whose effects result in a loss of actuation/sensing accuracy). By virtue of the additive modelling, the δ parameter can be considered normalized by default to $[-1, \ 1]$ with $\delta = 0$ indicating nominal behaviour, $\delta = -1$ complete loss of the channel and $\delta = 1$ a 100% (measurement or actuation) increased effect. Nevertheless, a more precise LOE modelling can be performed assuming δ changes within a component-specific set $[a, \ b]$ (see subsequent section on normalization). A possible command chain to obtain this atom is:

$$>> \ LOE = sym('LOE','real')$$
$$>> \ LOE_{lft} = sym2lfr(1 + LOE, LOE, 0)$$

C. Inverse (mass) LFT atom

Representing uncertain inverse parameters in LFT format is difficult if attempted directly. Rather, the corresponding δ^{-1} should be considered as an individual uncertain parameter (e.g. $\hat{\delta} = \delta^{-1}$) which is referred to as a Laurent variable [6], [10]. After this 'conceptual' transformation of the parameter it is fairly straight forward to obtain an LFT through its normalization. This is one of the few instances where normalization can be, and should be, performed initially (a code example for such a process is given in the next section).

For the specific case of satellites, the more relevant parameter to be inverted is the mass since the forces commanded by the controller are required to be transformed into accelerations before entering the relative motion equation. This mass has to be considered uncertain due to (mostly) fuel consumption.

D. Normalization

Since this is one of the most critical LFT operations when dealing with physical systems, this subsection is included for completeness –it is recommended to read also the corresponding section in [8].

The normalization process will transform the selected uncertainty parameter δ bounded in $[a, \ b]$ into an uncertainty parameter $\hat{\delta}$ ranging between $[-1, \ 1]$ by a simple transformation:

$$\delta = \frac{b+a}{2} + \frac{b-a}{2}\hat{\delta} \tag{3}$$

The command *normlfr* from [8] automates this process. The following code exemplifies the normalization of the inverse of the mass. This special case (e.g. normalization of an inverse parameter) shows the flexibility, and sometimes required ingenuity, afforded by LFT operations. Note, that rather than attempting to obtain an LFT of $1/mass$ with *mass* as the independent parameter, it is easier to define directly *mass* as an LFT, normalize it (in the present case around ± 5% of the nominal mass) and then define $1/mass$ as the inverse of the normalized *mass* LFT (which is directly an LFT by virtue of the property that "LFTs operations are LFTs"):

$$>> lfrs \ mass$$
$$>> mass_{norm} = normlfr(mass, 0.95 * mass_{sat}, 1.05 * mass_{sat})$$
$$>> mass_{lft} = 1/mass_{norm}$$
$$>> act_{mass_{lft}} = [mass_{lft} \ \ 0 \ 0; \ \ 0 \ \ mass_{lft} \ \ 0; \ \ 0 \ 0 \ \ mass_{lft}]$$

The last line of the code above shows how the mass LFT atom is used to form the transformation matrix from thrust forces to accelerations required by equation 1, which is again an LFT by virtue of the aforementioned LFT property.

IV. Relative Motion Closed-Loop Modelling

Armed with the previous LFT atoms it is now quite straight forward to form the more complex functions required to model as an LFT the relative motion closed-loop system outlined before. This principally entails appropriately posing the system components in a formulation amenable for LFT modelling (i.e. the relative motion from equation 1 as a multivariate polynomial matrix [10]).

A. Relative Motion modelling

Equation 1 could be directly transformed into an LFT by assuming the k constant and the angular orbital velocity/rate are independent symbolic parameters but this will yield an excessively conservative model of relatively high order (due to the non-integer powers found in the equation). Thus, the first step is to transform the equation of relative motion into a more LFT-prone formulation. From [15], [7], [5] a set of relations between the orbital elements can be obtained:

$$\omega = n\frac{(1+e\,cos\theta)^2}{(1-e^2)^{1.5}} \qquad n = \sqrt{\frac{\mu}{a^3}} \tag{4}$$

$$\dot{\omega} = -2n^2 e\,sin\theta\frac{(1+e\,cos\theta)^3}{(1-e^2)^3} \qquad |R| = \frac{a(1-e^2)}{1+e\,cos\theta} \tag{5}$$

The $k\omega^{1.5}$ element in equation 1 is troublesome but using the above right-column relations, the following is obtained:

$$k\omega^{1.5} = n^2\frac{(1+e\,cos\theta)^3}{(1-e^2)^3} \tag{6}$$

Substituting into equation 1 and assigning the following symbolic parameters ($\rho_1 = n$, $\rho_2 = \frac{1}{(1-e^2)^{1.5}}$, $\rho_3 = (1+e\,cos\theta)$ and $\rho_4 = e\,sin\theta$) the following multivariate polynomial matrix equation is obtained (where $\hat{x} = [x\, y\, z]^\top$ and $\hat{a} = [a_x\, a_y\, a_z]^\top$):

$$\begin{aligned}\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix} =& -2\rho_1\rho_2\rho_3^2 \begin{bmatrix}0 & 0 & -1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ 1 & 0 & 0 & 0 & 0 & 0\end{bmatrix} \begin{bmatrix}\dot{\hat{x}}\\ \hat{x}\end{bmatrix}\\ & -\rho_1^2\rho_2^2\rho_3^4 \begin{bmatrix}0 & 0 & 0 & -1 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & -1\end{bmatrix} \begin{bmatrix}\dot{\hat{x}}\\ \hat{x}\end{bmatrix}\\ & -\rho_1^2\rho_2^2\rho_3^3 \begin{bmatrix}0 & 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 0 & -2\end{bmatrix} \begin{bmatrix}\dot{\hat{x}}\\ \hat{x}\end{bmatrix}\\ & -2\rho_1^2\rho_2^2\rho_3^3\rho_4 \begin{bmatrix}0 & 0 & 0 & 0 & 0 & -1\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & 0\end{bmatrix} \begin{bmatrix}\dot{\hat{x}}\\ \hat{x}\end{bmatrix} + \hat{a}\end{aligned} \tag{7}$$

Note, that once an orbit is selected (i.e. the eccentricity and mean motion are fixed), the symbolic parameters ρ_1 and ρ_2 become constant while the other two parameters are related through trigonometric relations of $\theta(t)$ (reference [5] shows a very smart approach to remove this dependency without overtly increasing the LFT dimension). It is noted that this equation is dependent on time (as highlighted by the explicit dependence of $\theta(t)$).

Using symbolic order-reducing LFT methods, such as the *symtreed* command in [8] or the Logic Horner Tree (LHT) algorithm from [10], it is direct to obtain an LFT of order 25 (6 repetitions for each ρ_1 and ρ_2, 11 for ρ_3 and 2 for ρ_4). Once the first two parameters are iteratively fixed at some value the LFT dimension reduces to 13 and two parameters which is enough to perform modern robust analyses.

B. Sensor component modelling

Once the equation of motion is in LFT form, the sensor and actuator components are modelled next. In order to simplify the modelling complexity, the sensors and actuators are initially modelled both as a full-block misalignment matrix concatenated in series with a loss of efficiency matrix (α_i indicates the percentage loss and ε_{ij} is as before the misalignment value, both ranging between $[a,\, b]$):

$$\begin{bmatrix}1+\alpha_1 & 0 & 0\\ 0 & 1+\alpha_2 & 0\\ 0 & 0 & 1+\alpha_3\end{bmatrix}_{LOE} \begin{bmatrix}1 & \varepsilon_{xy} & \varepsilon_{xz}\\ \varepsilon_{yx} & 1 & \varepsilon_{yz}\\ \varepsilon_{zx} & \varepsilon_{zy} & 1\end{bmatrix}_{MIS} \tag{8}$$

The case of the misalignment matrix needs to be carefully approached since using directly the MIS LFT atoms will activate LFT reduction techniques by assuming the ε_{ij} parameters are all the same. It is desirable to assess which misalignment can be potentially worse (this is of interest for example in analyzing fault tolerance properties where determination of the largest deviation of a component can help to assess the need for aborting the mission), thus the following code is used (functions *lfthorzcat*, *lftvertcat* and *lftmult* have been specially developed to horizontally/vertically concatenate and multiply LFTs respecting the order of the symbolic parameters):

$$\begin{aligned}&>> sen_{mis1} = lfthorzcat([1 \;\; MIS_{lft}], MIS_{lft})\\ &>> sen_{mis2} = lfthorzcat([MIS_{lft} \;\; 1], MIS_{lft})\\ &>> sen_{mis3} = lfthorzcat(MIS_{lft}, [MIS_{lft} \;\; 1])\\ &>> sen_{MIS} = lfthorzcat(sen_{mis1}, sen_{mis2}, sen_{mis3})\end{aligned}$$

For the loss of efficiency matrix, the following command gives directly the LFT model based on the LOE LFT atom:

$$>> sen_{LOE} = lftvertcat([LOE_{lft}\, 0\, 0], [0\, LOE_{lft}\, 0], [0\, 0\, LOE_{lft}])$$

Finally, combining the above matrices by LFT multiplication the full sensor component is obtained (note that two different sensing units are considered, one for the relative velocity vector and the other for the position vector):

$$\begin{aligned}&>> sen_{lft_{vel}} = lftmult(sen_{LOE}, sen_{MIS})\\ &>> sen_{lft_{pos}} = sen_{lft_{vel}}\\ &>> Sen_{lft} = lftvertcat([sen_{lft_{vel}}\; zeros(3)], [zeros(3)\; sen_{lft_{pos}}])\end{aligned}$$

The actuator component is similarly obtained (note that using clusters with 4, 8 or more thrusters is efficiently accomplished by just properly augmenting the matrices). The final closed-loop system *CLPsys* is obtained by multiplying all the LFTs and closing the resulting augmented open-loop system *OPLsys* in negative feedback with the relative position gain controller $Position_K$:

$$\begin{aligned}&>> OPLsys = lftmult(Sen_{lft}, RelMot_{lft}, Act_{lft})\\ &>> CLPsys = feedback(OPLsys, Position_K, -1)\end{aligned}$$

V. Analysis Results

After the general LFT model of the closed-loop is obtained, and thanks to the direct manner of changing the nominal values and normalization limits (i.e. the $[a\, b]$ range for the LFT parameters *mass*, ε_{ij} and α_i), it is quite straight forward to not only assess the (circular-orbit) position controller robustness and performance for an elliptical orbit but also to provide feedback for instrumentation sizing (e.g. the analyses presented here can serve a satellite systemist to choose between different sets of thrusters, sensors and payload in a fast and automated manner).

Two uncertainty sets are used to evaluate the controller, each set depends on a different normalization of the MIS and LOE atoms (the MASS atom is always set to $mass \in [0.95,\, 1.05]$ representing ±5% variations):

$$\begin{aligned}\Delta_1 =& \{LOE \in \pm 15\%,\;\; MIS \in \pm 1\%\}\\ \Delta_2 =& \{LOE \in \pm 25\%,\;\; MIS \in \pm 10\%\}\end{aligned} \tag{9}$$

A. Eigenvalue analysis

A simple iterative process is used to perform an eigenvalue analysis of the controller. First, a closed-loop analysis is performed using minimum, nominal and maximum values (i.e. -1, 0 and 1) and compared for completeness to the open-loop analysis, for each of the two uncertainty sets in equation 9. In order to do this, a sweep of the true anomaly $\theta \in [1:5:360*3]$ degrees is used to analyze the behaviour of the controller along several orbits (i.e. every 5 degrees for three orbits).

Figure 2 shows the result for the Δ_2 uncertainty set. The bars in the plots represent the range of the poles magnitudes. Observe in the top plot the classical marginally instability of open-loop elliptic orbits [5] –note that since we are not using J2, solar pressure or other space environment effects the open-loop system oscillates periodically (with the largest deviations occurring at perigee passages). The controller is shown to make the closed-loop stable for the min/nom/max variations in set Δ_2 (similar result for Δ_1). Observe as well that when the uncertainty is deflected towards its minimum value, the stability margin (distance to the unstable positive values) is reduced.

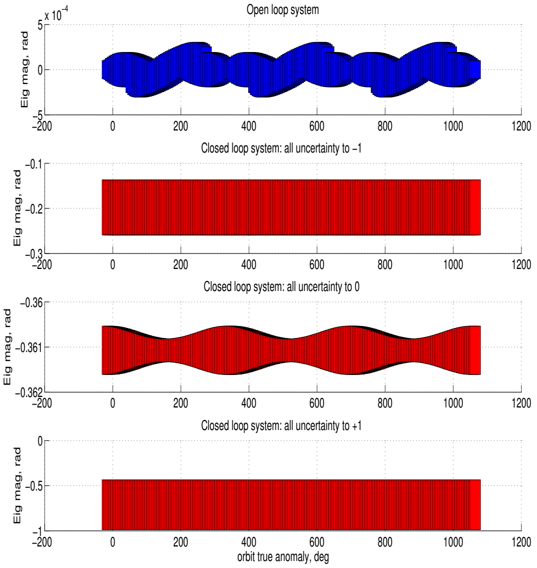


Fig. 2. Eigenvalue analysis: uncertainty set 2.

A snipet of the code used to perform the above analysis is given next:

$$>> RelMot_{fix} = flup(RelMot_{lft}, [NaN, \rho_1, \rho_2, \rho_3, \rho_4])$$
$$>> OPLsys_{tmp} = lftmult(Sen_{lft}, RelMot_{fix}, Act_{lft})$$
$$>> OPLsys_{fix} = flup(OPLsys_{tmp}, \Delta_{1nom})$$
$$>> CLPsys_{fix} = feedback(OPLsys_{fix}, Position_K, -1)$$
$$>> eig_{opl} = eig(RelMot_{fix}.a)$$
$$>> eig_{clp} = eig(CLPsys_{fix}.a)$$

This code is contained within a loop so that at each passage the ρ_3 and ρ_4 parameters are evaluated for a new true anomaly value (ρ_1 and ρ_2 always fixed using the semi-major axis and eccentricity values from Section II). The process then calculates the value of the relative motion LFT $RelMot_{fix}$, uses the result to obtained the augmented open-loop system $OPLsys_{tmp}$ which is then 'fixed' at the selected sensor/actuator uncertainty values (nominal values of Δ_1 set in the code), and then closed in negative feedback with the controller $CLPsys_{fix}$ and analyzed. Note that just by changing the, for example, orbit parameters (ρ_1 and ρ_2) or the sensor/actuator LFT models the same code can be used with minor modifications (i.e. the names used) to perform new analyses.

B. Robust performance analysis

Next, the use of the structured singular value for robust performance analysis of the controller is exemplified. The focus is directly on the robust performance (RP) analysis which requires introducing a fictitious uncertainty block to 'close' the performance channels [16].

Again, the use of an LFT, and associate tools (with their connection to the robust analysis tools of [1]), considerably simplifies performing the analysis while allowing for more extensive testing if desired. For example, the following code shows how to perform a robust performance test for the case of actuator uncertainties set to their nominal value and the sensor Δ allowed to range freely within its $[-1,\ 1]$ bounds (specified using the 'NaN' option):

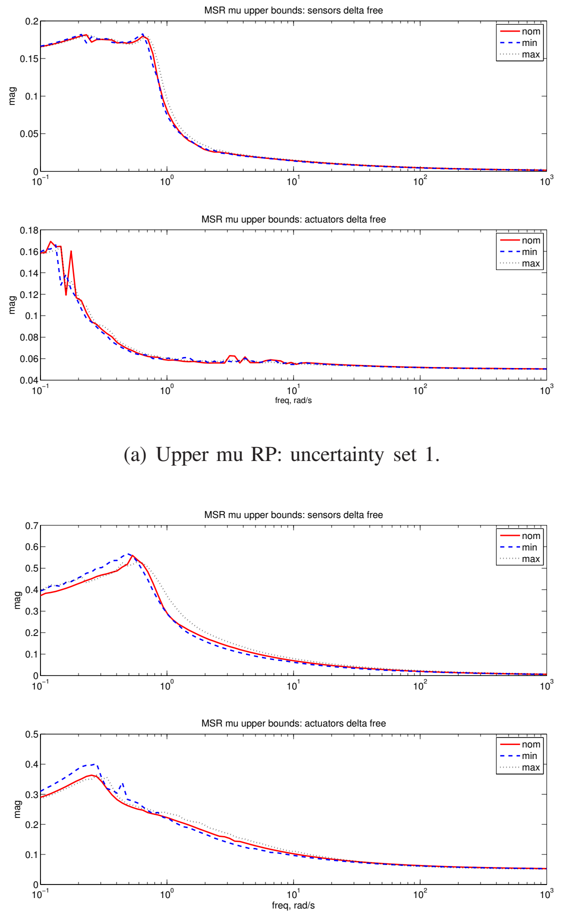
$$>> deltavalues = [NaN, NaN * sen_{vec}, 0 * act_{vec}]$$
$$>> OPLsys_{fix} = flup(OPLsys, deltavalues)$$
$$>> CLPsys_{fix} = fldown(OPLsys_{fix}, -Position_K)$$

The above lines 'fix' the LFT of the open-loop system *OPLsys* at the values specified in *deltavalues*, and then forms the closed-loop system (which is now 'fixed' at the given values since the controller is just a gain matrix). Note that by iteratively changing *deltavalues* (for example, setting one component to minimum/nominal/maximum Δ values) insight on the more critical components can be obtained.

The following code shows how to calculate the bounds on the structured singular value *mu* (spelled out to avoid confusing with the gravitational constant μ) given a grid of frequency points:

$$>> frequ = logspace(-1, 3, 100)$$
$$>> [M, blk] = lfr2mua(CLPsys_{fix}, frequ)$$
$$>> bnds = mu(M, blk, 'wlu')$$

Figure 3 shows two subplots, each presenting the resulting upper bound structured singular value for each of two uncertainty sets Δ_1 and Δ_2. Each subplot has two graphs, the top one showing the case for the sensor Δ left free to vary (with that for the actuator set at min/nom/max value) while the bottom graph corresponds to the actuators left free (and the sensor fixed at min/nom/max condition). By comparing the *mu* upper bounds from the min/nom/max uncertainty value sets for the 'fixed' component, an assessment of the effects its variability has on the robust performance is thus obtained. Observe that this variation on the 'fixed' values does not affect much the RP results as the bounds are very similar indicating that higher levels of LOE/MIS can be used. Note also, that as expected (since it allows for larger levels of uncertainty), the RP results for set Δ_2 are more critical, i.e. closer to the 1 mu bound.



(a) Upper mu RP: uncertainty set 1.

(b) Upper mu RP: uncertainty set 2.

Fig. 3. Robust performance (RP) upper mu bound on the two uncertainty sets with sensors or actuators uncertainty fixed.

Of course, mu is typically not possible to be calculated exactly, thus upper *and* lower bounds need to be obtained to assess the conservativeness of the answer (the closer, the better). Figure 4 shows the mu bounds for the two uncertainty sets allowed to vary free (this is the more dramatic case as observed from the larger values of RP obtained). Note that as before, the Δ_2 set yields the smallest (more critical) robust performance margin.

VI. Conclusion

In this article the use of modern tools for modelling and analysis has been exemplified while analyzing the applicability of a position circular-orbit RVD controller in the more challenging case of elliptic orbits. The analyses show that the robustness characteristics of the controller are valid in the chosen elliptic orbit and for the selected set of actuator/sensor uncertainties. The article is prepared as a tutorial on the use of LFT tools found in recent toolboxes. It has shown that changes in the resulting LFTs can be quickly introduced without major revisions to the code which should help introducing these techniques and tools outside the academic and research communities.

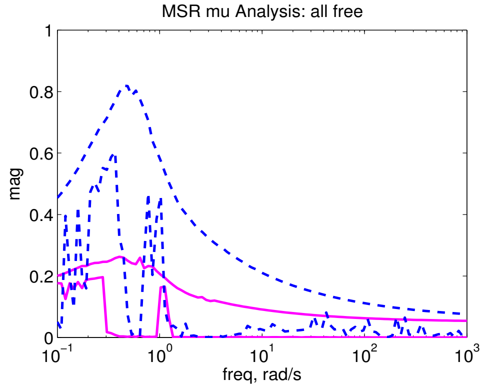


Fig. 4. Robust performance (RP) upper mu bound on the two uncertainty sets with full actuator and sensor uncertainty: solid (set 1), dashed (set 2).

References

[1] Balas, G.J., Chiang, R., Packard, A., and Safonov, M. Robust control toolbox v3.0, 2006.

[2] Becker, G. and Packard, A. Robust Performance of Linear Parametrically Varying Systems Using Parametrically Dependent Linear Dynamic Feedback. *Systems and Control Letters*, 23(3):205–215, 1994.

[3] Biannic, J.M., Marcos, A., Bates, D.G., and Postlethwaite, I. Nonlinear LFT modelling for on-ground transport aircraft. In *Nonlinear Analysis and Synthesis Techniques in Aircraft Control*, chapter 6. Springer-Verlag, 2007.

[4] Fehse, W. *Automated Rendezvous & Docking*. Cambridge University Press, 2003.

[5] Gaulocher, S., Chrétien, J.P., Pittet, C., Delpech, M., and Alazard, D. Closed-Loop Control of Formation Flying Satellites: Time and Parameter Varying Framework. In *2nd International Symposium on Formation Flying Missions & technologies*, Washington D.C., USA, September 2004.

[6] Hecker, S. and Varga, A. Generalized lft-based representation of parametric uncertain models. *European Journal of Control*, 10(4), 2004.

[7] Inalhan, G., Tillerson, M., and How, J.P. Relative Dynamics and Contorl of Spacecraft Formations in Eccentric Orbits. *Journal of Guidance, Control, and Dynamics*, 25(1):48–59, January 2002.

[8] Magni, J.F. Linear Fractional Representation Toolbox for Use with MATLAB. Technical Report TR 240/01 DCSD, ONERA, department of Systems Control and Flight Dynamics, Toulouse, France, December 2001.

[9] Marcos, A., Bates, D.G., and Postlethwaite, I. Flight Dynamics Application of a New Symbolic Matrix Order-Reduction Algorithm. In *International Conference on Polynomial Symbolic Systems*, Paris, FR, November 2004.

[10] Marcos, A., Bates, D.G., and Postlethwaite, I. A Symbolic Matrix Decomposition Algorithm for Reduced Order Linear Fractional Transformation Modelling. *Automatica*, 43(7):1211–1218, July 2007.

[11] Melton, R.G. Time-Explicit Representation of Relative Motion Between Elliptical Orbits. *Journal of Guidance, Control, and Dynamics*, 23(4):604–610, July 2000.

[12] Scherer, C. and Weiland, S. *Linear Matrix Inequalities in Control*. Dutch Institute of Systems and Control (DISC), October 2000.

[13] Varga, A. and Looye, G. Symbolic and Numerical Software tools for LFT-based Low Order Uncertainty Modeling. In *IEEE Symposium on Computed Aided Control System Design*, Hawai'i, USA, August 1999.

[14] Xu, Y., Fitz-Coy, N., Lind, R., and Tatsch, A. mu Control for Satellites Formation Flying. *Journal of Aerospace Engineering*, 20(1):10–20, January 2007.

[15] Yamanaka, K. and Ankersen, F. New State Transition Matrix for Relative Motion on an Arbitrary Elliptical Orbit. *Journal of Guidance, Control, and Dynamics*, 25(1):60–66, January 2002.

[16] Zhou, K., Doyle, J.C., and Glover, K. *Robust and Optimal Control*. Prenticel-Hall, Englewood Cliffs, NJ, 1996.