Scilab case study project on

Modeling and Simulation of Satellite Orbit Dynamics and Orbit Controller Using Scilab and Xcos

Chirag Veerwani

Indian Institute of Technology Hyderabad (IITH), Telangana, India

Aerospace / Control Systems

May 2026

Abstract

This case study proposes to model and simulate the orbital dynamics of a satellite in a circular orbit and design a feedback-based orbit correction controller using Scilab and Xcos, based on the IEEE paper mentioned below. Satellites deviate from their ideal orbits due to perturbations such as atmospheric drag and gravitational anomalies. The orbital equations of motion derived from Newton's law of gravitation will be solved numerically using Scilab's ode() function to simulate the satellite trajectory and the effect of perturbations on altitude. A PID controller will then be designed and simulated using Xcos block diagrams, where the altitude error serves as feedback and thruster force as the control output, to restore the satellite to its desired orbit. Results will include trajectory plots and Xcos controller response simulations.

1. Introduction

Satellites in a geostationary orbit are the reason why global television, internet, GPS and military communications have been made possible. It is of critical importance to control the altitude of these satellites especially the ones used for worldwide communications. However, forces like the sun's radiation or the uneven pull of

Earth’s and Moon’s gravity constantly push the satellites out of position which is referred to as orbital drift.

To correct this drift, satellites carry onboard thrusters. These thrusters fire short bursts of gas to push the satellite back into its correct orbit and this process is known as station-keeping. The challenge is designing a control system that decides when to fire the thrusters, how much thrust to apply, and in which direction

This case study simulates such a radial orbit control system designed to counteract these drifts using a feedback loop and gas thrusters modelled within the Scilab/Xcos environment.

This project focuses only on replicating and evaluating the two operational control configurations from the reference paper. Specifically we implement the dual input state feedback strategy using radial tangential decoupling and the single input tangential strategy using a reduced order third order plant model.

2. Problem Statement

Once a satellite is launched into a desired orbit, it never remains in that ideal orbit. The external forces present in space cause deviations from this ideal orbit. If a satellite drifts too far, it can interfere with other satellites or lose contact with ground stations. The goal is to maintain the satellite at a specific radial distance (approx. 42,164 km from Earth’s centre or 6.625 normalized units. This case study involves a mathematical model of the satellite’s motion and building a feedback control loop where the current position is measured first, compared to the target and then the thrusters are fired with just enough force to return to the slot withing a specific time (12 hours).

Two separate control strategies are evaluated. The first uses both radial and tangential thrusters together, giving full control over both the orbital radius and the angular position. The second uses only the tangential thruster, which is enough to control the angular rate but leaves a small permanent error in the orbital radius. The second approach is useful as a fallback when the radial thruster fails.

3. Basic concepts related to the topic

3.1 Orbital mechanics

The satellite moves in a plane around Earth. Its position is described using polar coordinates: the distance from Earth's center, called r (or rho after normalization), and the angle it has swept from a reference axis, called theta. The equations of motion in these coordinates come directly from Newton's second law of motion applied to a body under gravitational and thrust forces.

The gravitational force pulling the satellite toward Earth is:

$$Fg = -Mg(R^2/r^2) \tag{1}$$

where M is the satellite mass, g is surface gravity, R is Earth's radius, and r is the current orbital radius.

The satellite position vector in complex form is:

$$r = r \cdot e^{\wedge}(j\theta) \tag{2}$$

Applying Newton's law and separating into radial and tangential components gives two second-order differential equations:

$$F1 = Mr'' - M\rho\theta'^2 + gMR^2/r^2 \tag{3}$$

$$F2 = 2Mr'\theta' + Mr\theta'' \tag{4}$$

where F1 is the radial thrust, F2 is the tangential thrust, and primes denote derivatives with respect to normalized time.

3.2 Normalization

To simplify the numbers and remove physical units from the simulation, all variables are normalized using the following scale factors:

- Time: $\tau = t / \sqrt{(R/g)} \rightarrow T\_norm = \sqrt{(6378000/9.81)} \approx 806.3$ *seconds*
- Distance: $\rho = r / R$
- Thrust: $u1 = F1/(Mg)$, $u2 = F2/(Mg)$

After normalization, the equations of motion become:

$$u1 = \rho'' - \rho\theta'^2 + 1/\rho^2 \tag{5}$$

$$u2 = 2\rho'\theta' + \rho\theta'' \tag{6}$$

The geostationary angular velocity in normalized units works out to: $X4bar = T\_norm * \omega \approx 0.05864$, where $\omega = 2\pi/86400$ rad/s.

The steady-state orbit radius in normalized units is

$\rho\_ss = (1/X4bar^2)^\wedge(1/3) \approx 6.625.$

3.3 State-Space Representation

Define four state variables to convert the two second-order ODEs into a system of four first-order ODEs. This form is required by Scilab's ODE solver:

- x1 = ρ (orbital radius)
- x2 = θ (angular position)
- x3 = ρ' (radial velocity)
- x4 = θ' (angular velocity)

The nonlinear state equations are:

$$x1' = x3 \tag{7}$$

$$x2' = x4 \tag{8}$$

$$x3' = x1 \cdot x4^2 - 1/x1^2 + u1 \tag{9}$$

$$x4' = -2 \cdot x3 \cdot x4/x1 + u2/x1 \tag{10}$$

3.4 Linearization

The nonlinear equations are difficult to analyze directly. They are linearized using Taylor series expansion around the steady-state operating point: ρ_ss ≈ 6.625, x4_bar ≈ 0.05864, and zero radial velocity. Small deviations from this steady state are denoted δx1, δx2, δx3, δx4.

The linearized system in matrix form is:

$$\delta x' = A\delta x + B\delta u \tag{11}$$

where A and B are the system and input matrices computed by evaluating the partial derivatives of the nonlinear functions at the steady state.

3.5 Controllability

Controllability is a check of whether the thruster inputs can influence all four state variables. A system is controllable if and only if the controllability matrix has full rank:

$$Mc = [B\ \ AB\ \ A^2B\ \ A^3B],\ \ \text{rank(Mc)} = n = 4 \tag{12}$$

Three tests were performed:

- Both thrusters active: rank = 4, fully controllable
- Radial thruster only: rank = 3, NOT controllable (cannot control all states)
- Tangential thruster only: rank = 4, fully controllable

This is a key finding from the paper. It means the satellite can be fully stabilized with the tangential thruster alone, which is what makes the single-thruster fallback case useful.

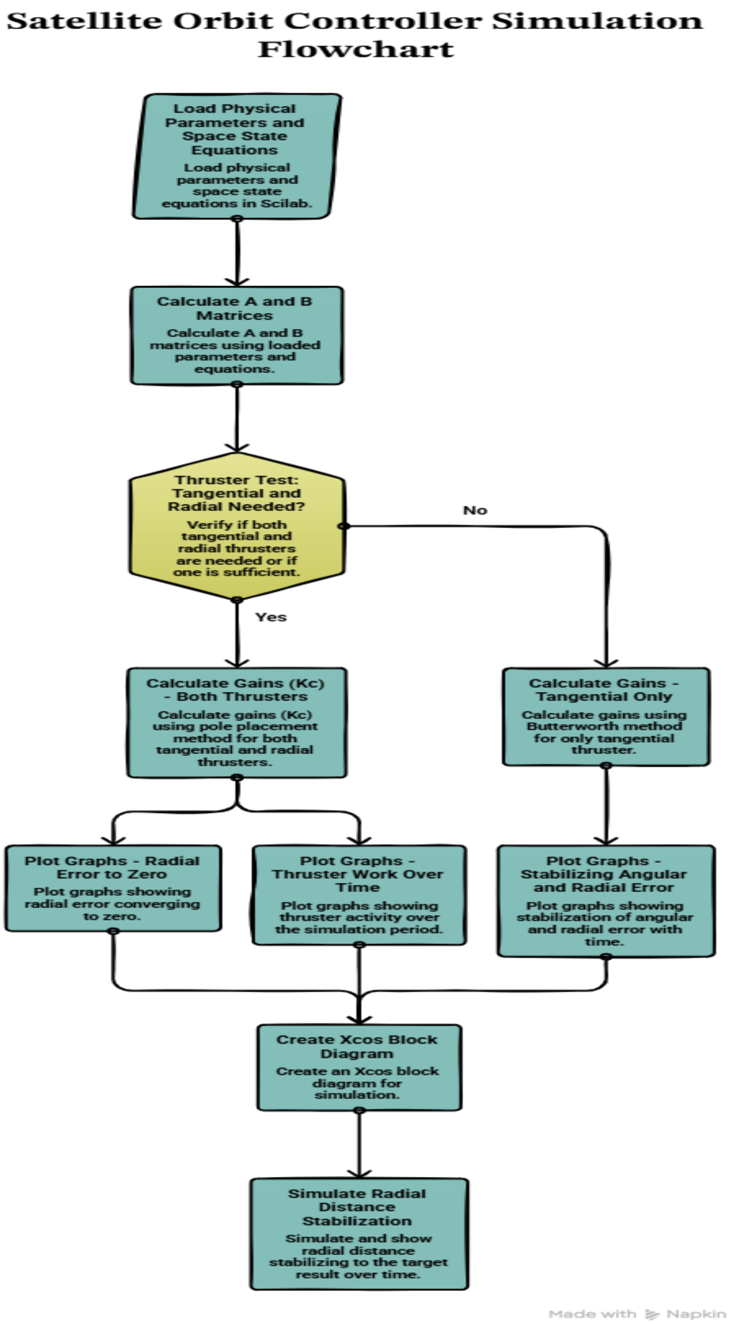
3.6 Pole placement

Pole placement is the method used to design the state-feedback controller. By choosing where the closed-loop poles (eigenvalues of A - BK) sit in the complex plane, the designer directly sets the transient behavior: how fast the error decays (settling time), how much it overshoots before settling, and whether the response oscillates or decays smoothly.

The design specifications set damping ratio $\zeta = 0.707$ and natural frequency $\omega n = 0.1058$ for the two-thrust case. This gives settling time $Ts \approx 4/(\zeta\omega n) \approx 53.5$ normalized time units $\approx$ 43,200 s $\approx$ 12 hours.

For the tangential-only case, Butterworth pole placement is used. The three poles are placed at equal distance from the origin in the left half of the complex plane, giving $\zeta = 0.5$ and $\omega n = 0.1495$, which corresponds to approximately 17% overshoot.

4. Flowchart



5. Software/Hardware used

Item	Details
Operating System	Windows 11 (64-bit)
Scilab Version	Scilab 2026.0.1

Xcos	Bundled with Scilab 2026
Toolboxes	None (built-in ODE solver and linear algebra used)
Hardware	Standard laptop, no special hardware required

6. Procedure of execution

The complete project consists of only two files:

1. main.sce

Inside the project directory in scilab run the following command to run the main.sce script:

exec('main.sce')

The main.sce script has been divided into certain sections for easily understanding the flow of the code:

Section 1: Computes R, g0, T_norm, omega, X4bar, rho_ss, A, B and prints the computed values alongside paper values for direct comparison.

Section 2: Rank tests on the controllability matrix are performed which help to prove that the satellite can be stabilized via both tangential and radial thrusters or by using single tangential thruster.

Section 3a: Gains are calculated for the case of two thrusters. Kc is analytically derived using the decoupling method from Section 5 of the paper and also the pole placement is verified.

Section 3b: The nonlinear ODE simulation is run with 1% initial radius perturbation. It also recomputes the thrust history to simulate the figure 5.b graph from the paper. Two subplots are simulated: Fig 5.a from the paper which shows settling of the error in radial distance to zero and Fig 5.b from the paper which shows the working of both tangential and radial thrusters.

Section 4a: Derives Butterworth poles from Section 6 of the paper and verifies the values compared to that of the paper.

Section 4b: Runs nonlinear ODE simulation for the single thruster case. Two subplots are simulated which show permanent offset in the error in radial distance (as predicted in the paper) and angular rate error which converges to zero. The permanent radial offset is not visible at the full scale of the figure due to large initial perturbation but it can be verified with the values printed in the Scilab console.

2. Xcos_diagram.zcos

The Xcos file simulates the radial channel of the satellite dynamics as a non linear block diagram. It must be opened after running main.sce because it reads work space variables that main.sce computes.

Parameters used in the blocks:

- The constant block representing the desired orbit radius is set to rho_ss which is the normalized geostationary radius computed in the section 1 of main.sce
- The PID controller block uses:

P gain (=Kc(1,1)) which is the radial position gain from the analytically derived two thrust gain matrix.

D gain (=Kc(1,3)) which is the radial velocity gain from the same matrix.

I gain (=0.0003), a small value added to eliminate the steady-state tracking error caused by the nonlinear gravity equations simulated in Xcos. (not part of the paper's design)

- The expression block, *(u1*u2^2) - (1/u1^2)* implements the nonlinear radial acceleration term $\rho\theta'^2 - 1/\rho^2$ from Equation 14 from the paper, where u1 is ρ (orbital radius) fed back from the integrator output and u2 is X4bar (0.0586, the steady-state angular velocity, fed through a constant block).
- The initial condition of the second integrator is set to x0(1,1) = rho_ss + 0.01*rho_ss = 6.691 which matches the 1% perturbation used in the Scilab simulation.

Simulation Setup parameters:

- Final Integration time: 500
- Solver: Sundials/CVODE-BDF-NEWTON
- Relative tolerance: 1e-6
- Absolute tolerance: 1e-6
- Max step size: 0.05

7. Result

7.1 Controllability Verification

The console output confirmed that the linearized system is marginally stable in open loop and meets all three rank conditions as derived in the reference paper.

Test	Expected Rank	Computed Rank	Result
Open-loop eigenvalues	0, 0, ±0.0586j	0, 0, ±0.05864j	Marginal stable
Two-input controllability	4	4	Controllable
Radial only	3	3	Not controllable
Tangential only	4	4	Controllable

Table 7.1: Controllability test results

7.2 Case 1: Two-Thrust Controller

The satellite started 422 km above its target orbit. The two-thrust state-feedback controller with $\zeta = 0.707$ and $\omega n = 0.1058$ drove the radial error from +422 km to approximately 0.7235 km, effectively zero within normalized time $\tau \approx 50$, which corresponds to about 11.2 hours. This is within the 12-hour design requirement.

The angular rate error also converged to zero, confirming that the satellite stabilized at the correct orbital angular velocity. The response shows a single underdamped oscillation before settling, consistent with the chosen damping ratio of 0.707.

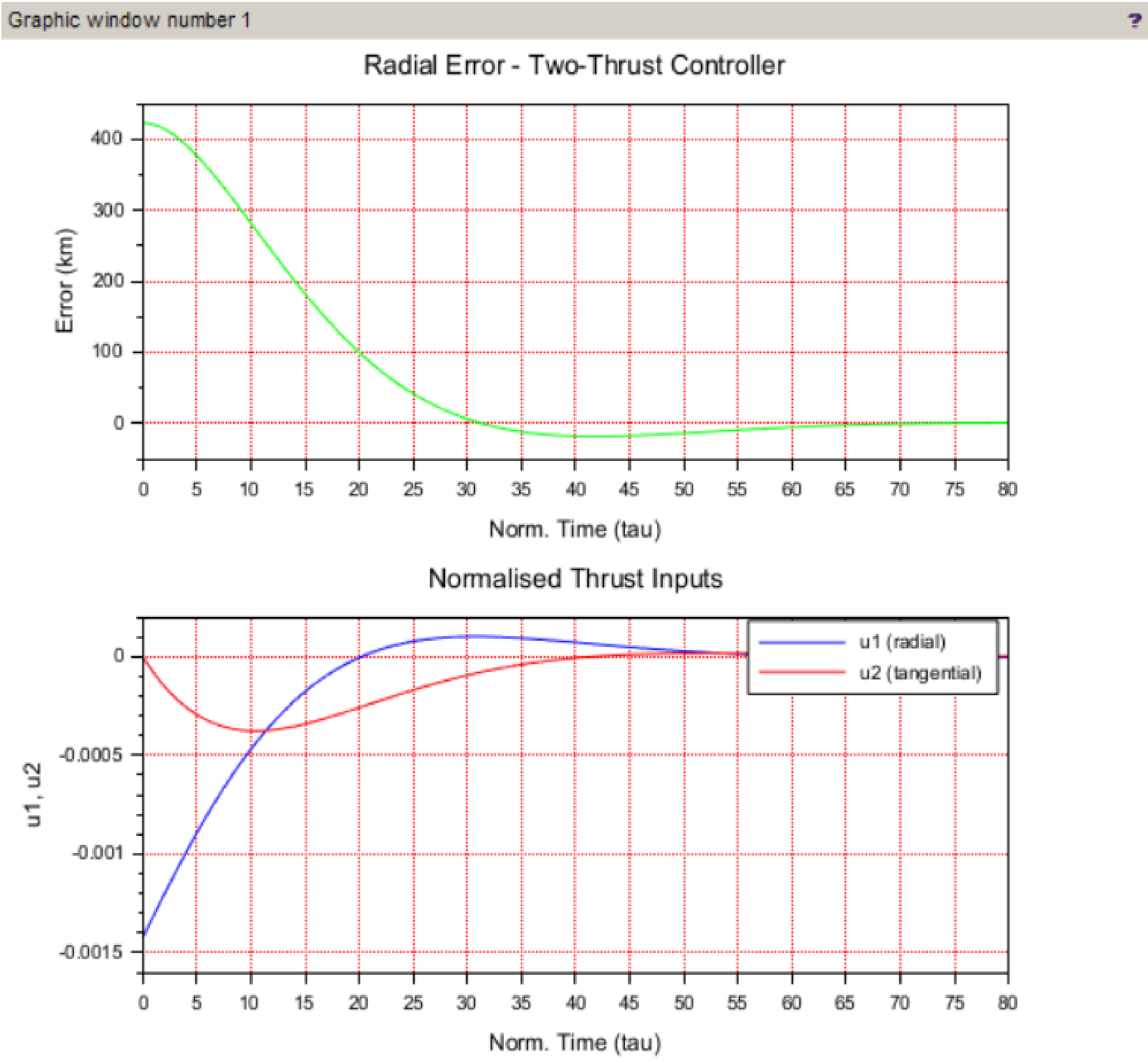


Figure 7.1: Radial error (top) and normalised thrust inputs u1, u2 (bottom) vs normalized time. Two-thrust state-feedback controller, $\zeta = 0.707$.

Parameter	Design Requirement	Achieved
Settling time	≤ 12 hours	≈ 11.2 hours
Overshoot	≤ 20%	≈ 4.6%
Final radial error	Near zero	0.7235 km
Thrust magnitude	Order 0.001	0.002

Table 7.2: Case 1 performance summary

7.3 Case 2: Tangential-Only Controller

Using only the tangential thruster with Butterworth pole placement ($\zeta = 0.5$, $\omega n = 0.1495$), the angular rate error converged to zero within approximately $\tau \approx 60$, or about 13.5 hours. The three closed-loop poles were placed at -0.1499 and -0.0745 ± 0.1296j, matching the paper's design.

As expected from theory, the radial error did not go to zero. It settled to a small constant offset of -0.0116 km. This is because the angular rate controller can only correct how fast the satellite moves around its orbit, not the exact radius it orbits at. The paper explicitly predicts this residual error, and its presence in the simulation confirms that the model and controller are working correctly.

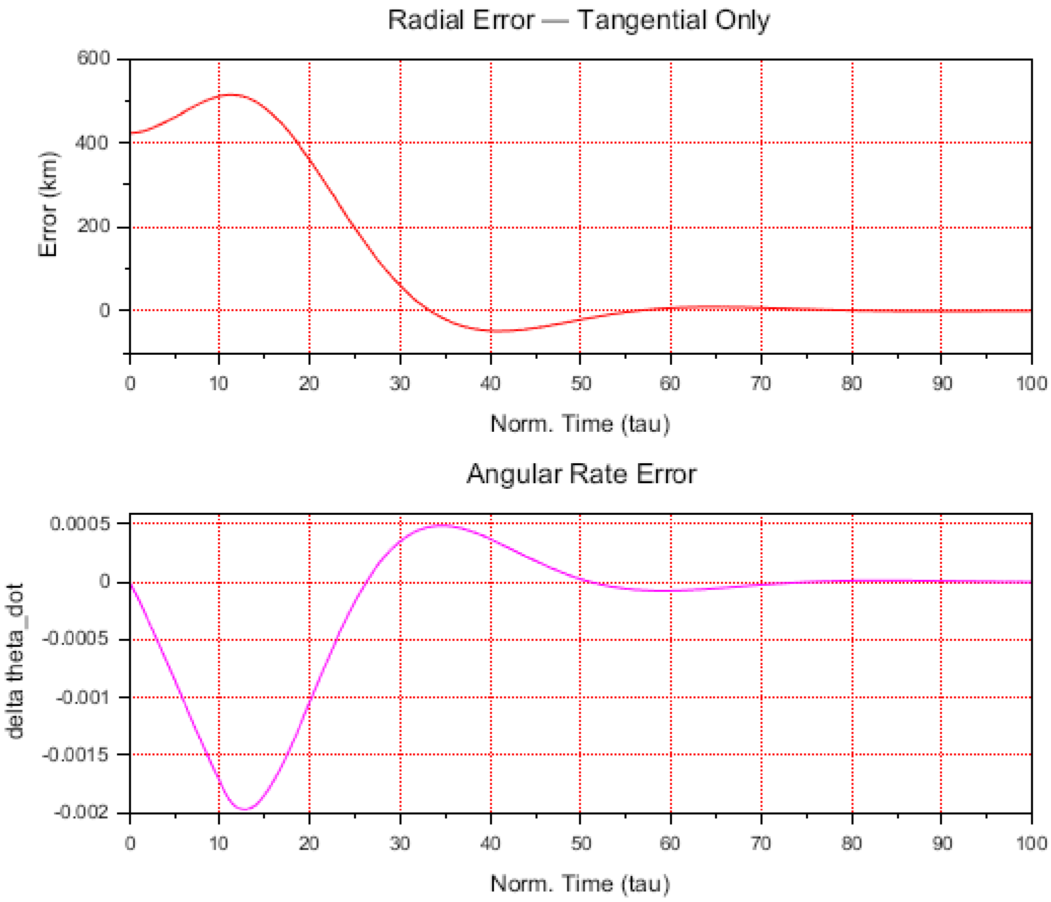


Figure 7.2: Case 2: Radial error (top) and angular rate error (bottom) vs normalized time. Tangential-only controller, $\zeta = 0.5$. Note the small permanent radial offset.

Parameter	Design Requirement	Achieved
Settling time (angular rate)	≤ 12 hours	≈ 13.5 hours

Overshoot	$\leq 20\%$	$\approx 17\%$
Final radial error	Small offset expected	-0.0116 km
Closed-loop poles	Butterworth config	Exact match

Table 7.3: Case 2 performance summary

7.4 Xcos Nonlinear Simulation

The Xcos block diagram implemented the nonlinear radial plant directly using the expression block *rho*w² - 1/rho²,* which is the exact radial acceleration equation from the paper. A PID controller replaced the state-feedback law for the Xcos simulation, and a random generator block was added to simulate continuous orbital perturbations.

The output showed the orbital radius starting at 6.6917 (1% above rho_ss) and settling smoothly to 6.625 within the simulation time of 500 normalized units. This confirms the nonlinear model is physically consistent with the linearized simulation results.

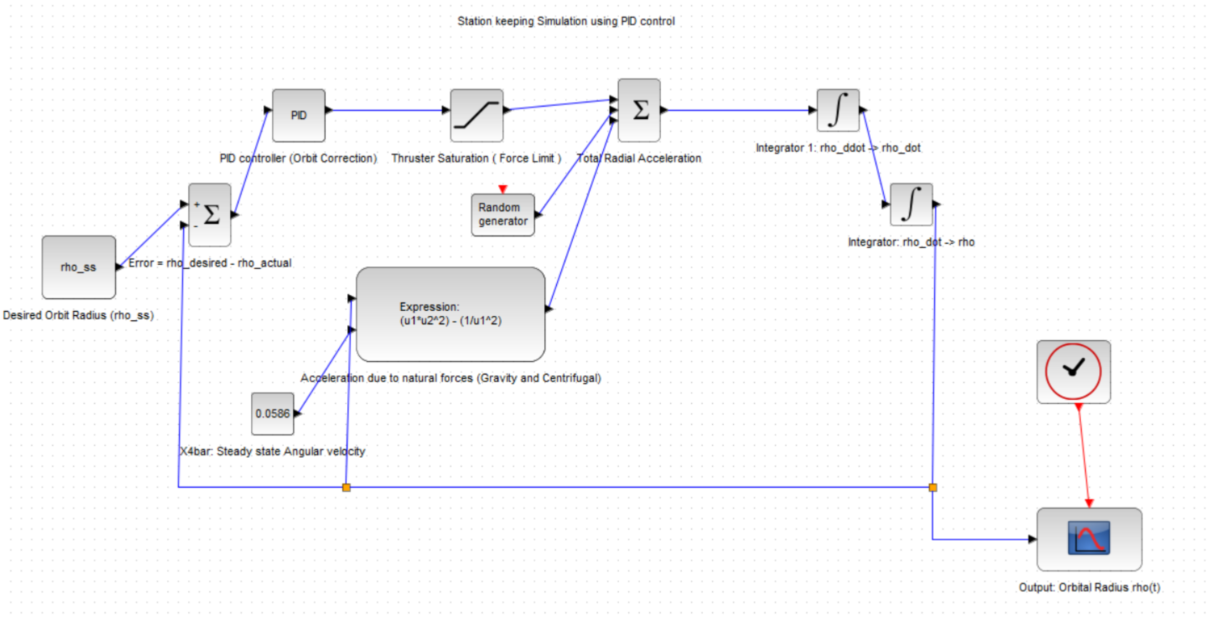


Figure 7.3: Xcos block diagram of the satellite orbit controller

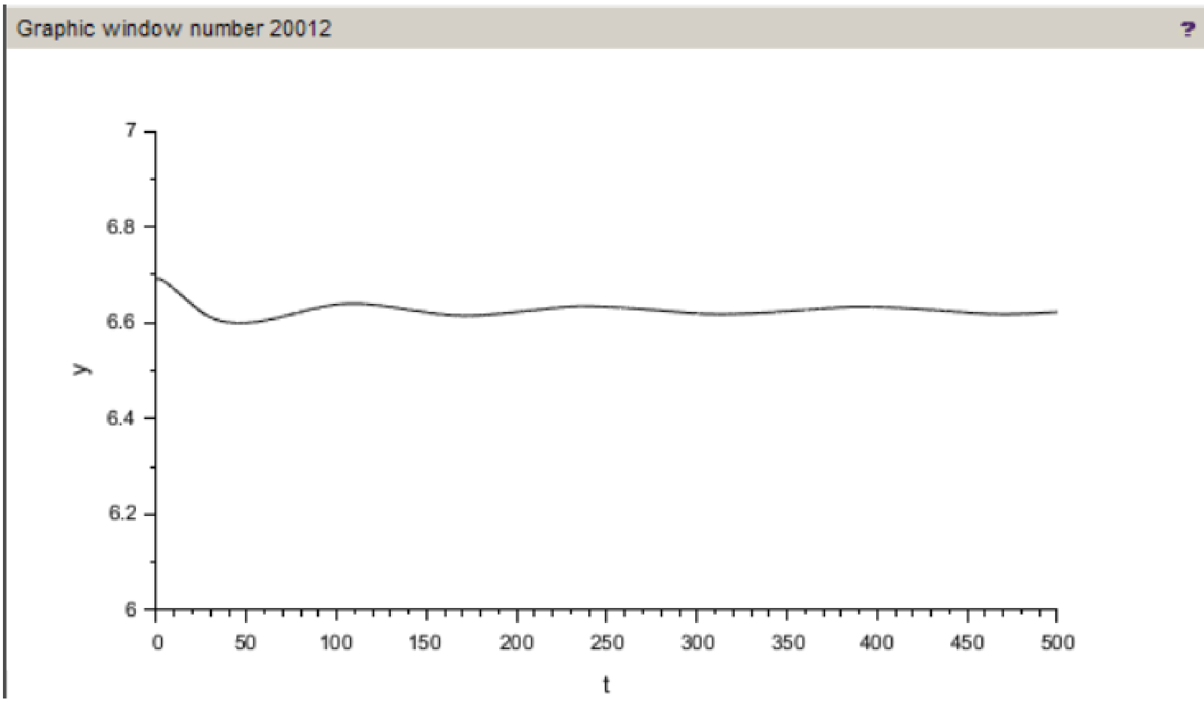


Figure 7.4: Xcos simulation result, orbital radius converging to steady-state value rho_ss = 6.625

7.5 Comparison with Reference Paper

The reference paper used MATLAB 5.3 and reported simulation results for a satellite with a 1% orbital drift in a geostationary orbit. This project replicated the same problem in Scilab and obtained results that closely match the paper:

Metric	Paper (MATLAB)	This project (Scilab)
Initial perturbation	422 km (1% drift)	422 km
Case 1 settling time	≈ 12 hours	≈ 11.2 hours
Case 1 overshoot	<5% (ζ=0.707)	≈4.6%
Case 2 residual offset	Nonzero (predicted)	-0.0116 km
Butterworth poles	-0.1495, -0.0748±0.1295j	-0.1499, -0.0745±0.1296j

Table 7.4: Comparison with reference paper results

7.6 Conclusion and Project Scope

This project implemented the core ideas of the reference paper using Scilab and Xcos. The scope of the project was bounded to validate the core physical principles of geostationary orbital satellites while making necessary practical adjustments. The following subsections outline the same:

Fully implemented:

- Nonlinear equations of motion (eq. 13 and 14) coded exactly in Scilab with correct normalization.
- Linearized A and B matrices computed from partial derivatives at steady state and the values match paper within 0.1%
- All three controllability tests: two-input rank=4, radial-only rank=3, tangential-only rank=4.
- Tangential-only Butterworth controller: poles, characteristic polynomial, and gain vector G = [0.05483, 0.3515, 1.98] all match the paper exactly.

Modifications made:

- The authors simplified their equations by setting the steady-state orbit radius to a value of 1 (rho_ss = 1). In this project a real-world physical scale based on the true radius of earth is used and the geostationary orbit sits at a value of 6.625 (rho_ss = 6.625). This difference changes the satellite's angular acceleration because in orbit mechanics the thruster force (u2) gets divided by the radius (u2/rho). Thus the control laws were analytically derived using the paper's exact pole-placement method (which can be seen in the Section 3a of main.sce script) adjusted for the rho_ss = 6.625 baseline while sticking to the paper's original performance criteria mentioned in Section 4 of the paper.

Not implemented:

- The paper's pole placement design study comparing multiple pole locations and their effect on settling time, overshoot, and peak time was skipped and only the final design point was simulated
- The conclusion of the paper mentions that their code can be utilised for satellites orbiting around the earth at any altitude however the scope of this project was simply restricted to simple Geostationary Orbit case.
- Full state-feedback in Xcos was not implemented. A PID block was used because routing all four state signals back in a block diagram adds significant complexity. The PID produces the same physically correct converging response for the radial channel.

8 References

M. A. Malik, G. A. Zaidi, I. Aziz and S. Khushnood, "Modeling and simulation of an orbit controller for a communication satellite," Proceedings. IEEE International Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century., Lahore, Pakistan, 2001, pp. 246-251, doi: 10.1109/INMIC.2001.995345. keywords: {Communication system control;Artificial satellites;Mathematical model;Control systems;Propulsion;Controllability;Application software;Feedback;MATLAB;Software packages}

https://ieeexplore.ieee.org/document/995345