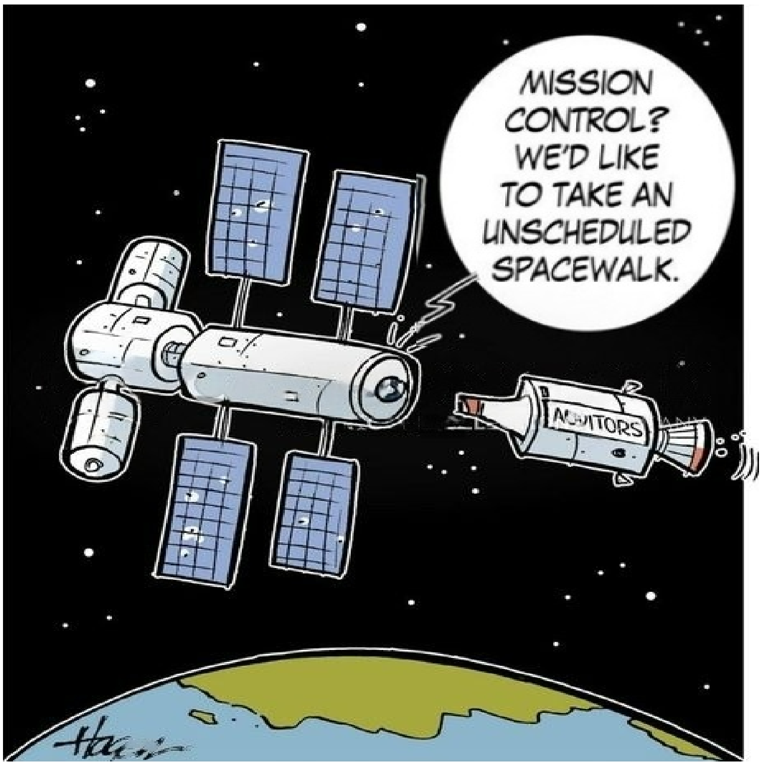
Robotic Control of a Satellite orbiting a Planet

Computational methods and optimization (FM 216)



Project summary

This project studies control strategies for a robotic system managing the orbit of a satellite on an elliptical trajectory around a planet. The first part of the project focuses on numerically solving Kepler's equation to determine the satellite's position, using root-finding techniques— fixed-point iteration, Newton's and Secant methods—and analyzing their convergence. The second part explores optimization techniques to minimize the mass of a rocket, employed for launching the satellite, under a constraint on its final velocity. Further, the robot is also tasked with setting optimal control configurations of the satellite to minimize its fuel consumption.

1 Project part - I

1.1 Introduction : Robotic control of a satellite

In modern space missions, autonomous robotic systems play a crucial role in controlling and managing satellites. The physics of a satellite orbiting a planet follows the principles of planetary motion. In this project, we simulate a scenario where a robot is tasked with maintaining and modifying the orbit of a satellite. The satellite's motion is governed by Kepler's equation, that cannot be solved by analytical methods. Thus the robot must have an on-board computing device that relies on numerical methods to predict the satellite's real-time position and an automatic control apparatus to apply corrective actions that achieves specific orbital characteristics.

Kepler's equation describes the relationship between the mean anomaly (θ_M), the eccentric anomaly (θ_E), and the orbital eccentricity (e) of the satellite. The equation is given by

$$\theta_M = \theta_E - e \sin \theta_E, \tag{1}$$

where $\theta_M \in [0, 2\pi]$, $\theta_E \in [0, 2\pi]$, $e \in [0, 1)$.

1.1.1 Parameters of Kepler's equation

Let's understand some fundamental concepts related to the orbit of a satellite around a planet. We'll skip the detailed derivation of the Kepler's equation here -interested readers can refer Casselman 2018.

1. **Orbit:** It refers to the path of a satellite around a planet. This path is considered to be an ellipse as shown in Fig 1. The planet is located at one of the foci of the ellipse.

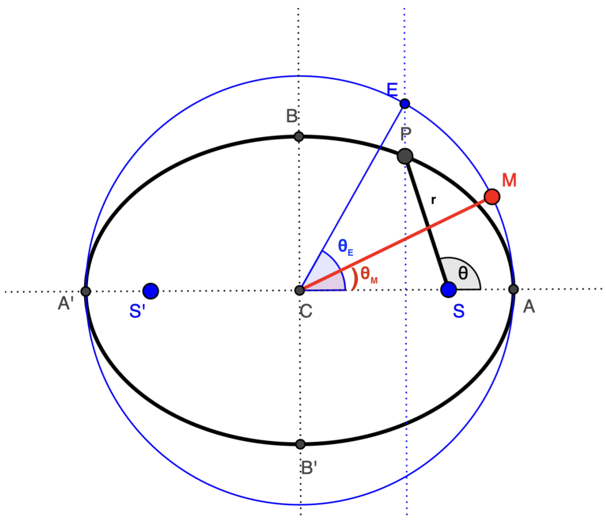


Figure 1: Mean Anomaly, Eccentric Anomaly, True Anomaly

2. **True Anomaly (θ):** The angle between the direction of perihelion (A) with respect to the centre (C) and the current position of the planet (P) as seen from the focus (S) of the ellipse.

3. **Mean Anomaly** (θ_M): It represents the fraction of the orbit's period that has elapsed since the last perihelion. It increases linearly with time from 0 to 2π as the planet completes its orbit. It is given by:
$$\theta_M = \frac{2\pi}{T}(t),$$
where T is the orbital period and t is the time elapsed since passage through the last perihelion.

4. **Eccentric Anomaly** (θ_E): The eccentric anomaly is an angular parameter that helps to describe the position of a satellite along its orbit. It is related to the mean anomaly through the Kepler's equation.

5. **Eccentricity (e)**: A measure of how much the orbit deviates from a circular shape. For a perfect circle, $e = 0$. For highly elongated ellipses, e approaches 1.

You may also refer to this animation for an immersive understanding of the parameters involved in the Kepler's equation.

Equation 1 cannot be solved analytically. Hence, we rely on numerical techniques to compute θ_E for a given θ_M and e.

The position of the satellite is obtained in terms of polar coordinates (r, θ), where r represents the radial coordinate of the satellite and θ represents the true anomaly. Once θ_E is computed, equation 2 can be used to calculate θ and equation 3 to compute the radial position r.

$$\theta = 2\arctan\left(\sqrt{\frac{1+e}{1-e}}\tan\frac{\theta_E}{2}\right), \tag{2}$$
$$r = a(1 - e\cos\theta_E), \tag{3}$$

where a is the semi-major axis of the orbit. Based on the calculated values of r and θ, the robot can adjust the satellite's thrust and velocity vectors to correct any deviations from the desired orbit. This involves continuously solving Kepler's equation as the satellite progresses in its orbit.

1.2 Brief overview of numerical techniques

For this project, we will utilize three numerical methods for solving equations. Each method has its own approach and convergence properties, making them suitable for different types of problems.

1.2.1 Fixed-point iteration

The fixed-point iteration method solves equations of the form $x = g(x)$ by repeatedly applying the update rule
$$x_{n+1} = g(x_n).$$
This process converges to a unique fixed point if the continuous function $g(x)$ in $[a, b]$ is differentiable for all $x \in [a, b]$ with $|g'(x)| \leq k < 1$, for some positive constant k. For pseudocode, refer to Appendix A.1.

1.2.2 Newton's method

Newton's method is one of the most widely used techniques for finding roots of a function $f(x)$. The method uses the update rule

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

This method typically converges rapidly when the initial guess is close to the true root. However, it requires the calculation of the derivative of $f(x)$ which can be computationally expensive. For pseudocode, refer to Appendix A.2.

1.2.3 Secant Method

The secant method is a derivative-free alternative to Newton's method. It approximates the derivative using the finite difference between two successive iterates as follows.

$$x_{n+1} = x_n - f(x_n)\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}.$$

This method is useful when the derivative is difficult to compute. For pseudocode, refer to Appendix A.3. For further theoretical details refer Burden et al. 2011.

1.3 Problem statements

1. **Well-posedness of the problem:** First, we must ensure that the problem we are trying to solve is well-posed. Using appropriate plots and reasoning, demonstrate that for any allowed value of θ_M, there is a *unique* value of θ_E that satisfies equation 1 for all elliptical orbits, i.e., for $0 \le e < 1$.

2. **Fixed-point iteration:** Convert the root-finding problem in equation 1 into an appropriate fixed-point problem, $g(\theta_E) = \theta_E$.

   (a) For nearly circular orbits (i.e., e close to 0) and $\theta_M = 1$, use the *fixed-point iteration* method, with the initial guess $\theta_{E_0} = \theta_M$, to determine the fixed points. (set tolerance $= 10^{-9}$).

   (b) For e close to 1, show that convergence to the fixed point is slower than in the case where e is close to 0. Explain why this occurs and suggest modifications that could accelerate convergence to the fixed point.

3. **Newton-Raphson and secant methods:** Apply the *Newton-Raphson* and *secant* methods as root-finding techniques to solve equation 1 for $e = 0.9$ and $\theta_M = 1$. (set tolerance $= 10^{-9}$).

   (a) Perform a comparative convergence analysis of the three techniques (fixed-point iteration, Newton-Raphson, and secant methods) you used to solve equation 1. Identify the most efficient method, particularly for varying values of the eccentricity e.

4. **Orbital positions and robotic control of the satellite:** A robot-guided satellite moves in an elliptical orbit with an eccentricity $e = 0.8$ and a semi-major axis $a = 2\,\text{km}$. The time period of the orbital motion is $T = 100$ seconds.

(a) Discretize the time interval from 0 to T into 20 equally sized intervals and calculate the position of the satellite (r, θ) at each of these time intervals.

(b) Plot the elliptical trajectory of the satellite.

(c) The control station expects the robot to place the satellite at specific positions at certain times. The table below shows the expected positions at key times $T/4$, $T/2$, $3T/4$, and T.

Time (s)	**Expected r (km)**	**Expected θ (radians)**
$T/4$	2.95696778	2.81033528
$T/2$	3.6	3.14159265
$3T/4$	2.95696778	3.47285002
T	0.4	0

Table 1: Expected satellite positions at key times.

Compare the positions (r, θ) obtained from the robotic control with the expected positions provided in the table. Calculate the error in both r and θ at times $T/4$, $T/2$, $3T/4$, and T.

5. **Generalized Kepler's equation:** The original Kepler's equation can be perturbed due to gravitational forces and the satellite's inclination relative to the orbital plane (López et al. 2018). This leads to the generalized form of the equation,

$$\theta_M = \theta_E - e \sin \theta_E + \frac{\epsilon^*}{(1-e^2)^3} \left[2\left(e^2+2\right)\theta_E - 8e \sin \theta_E + e^2 \sin 2\theta_E \right], \qquad (4)$$

where ϵ^* is a dimensionless parameter influenced by gravity and inclination. Assume $\epsilon^* = 0.0003$.

(a) Using appropriate plots and reasoning, demonstrate that for some specific *allowed values* of θ_M, there are *multiple* values of θ_E that satisfy equation 4 for all elliptical orbits, i.e., for $0 \leq e < 1$.
(*Hint: Whenever θ_M for a given θ_E exceeds 2π, map it to a value within $[0, 2\pi]$. For example, $\theta_M = 3\pi$ should be mapped to π, $\theta_M = 8\pi$ should be mapped to 2π, and so on*)

(b) Numerically compute θ_E for orbital parameters $T = 6$s and $e = 0.8$. You can solve it for discretized time as in question 4(a). Your task is to obtain the solution (within tolerance 10^{-9}) in the least number of iterations possible. Is the root θ_E unique for the given orbital parameters?

(c) How does the computational time for determining θ_E varies with different values of ϵ^* in the range $[0.01, 0.00001]$? Why do you think this happens?

(d) Compare the solution for (r, θ) in the presence of a small perturbation ϵ^* with the unperturbed case ($\epsilon^* = 0$) for different times (t) in tabular form. Use the parameters of question 5(b) and take $a = 2$ km. Animate both the trajectories.

2 Project part - II

2.1 Introduction: Optimal resource consumption for satellite launch

Satellites are launched into orbit using multi-stage rockets. Most modern rockets utilize three stages during their ascent into space. The first stage initially propels the rocket until its fuel is depleted, at which point the stage is jettisoned to reduce the mass of the rocket. The second and third stages then take over to place the rocket's payload (satellite) into orbit around the planet. The first goal in the second part of the project is to determine the individual masses of the three stages so that the total mass of the rocket is minimized while ensuring that it reaches the desired velocity.

For a single-stage rocket consuming fuel at a constant rate, the change in velocity resulting from the acceleration of the rocket is modeled by

$$\Delta V = -c_r \log\left(1 - \frac{(1-S)M_r}{P + M_r}\right), \tag{5}$$

where M_r is the mass of the rocket engine including initial fuel, P is the mass of the payload, S is a structural factor determined by the design of the rocket, and c_r is the (constant) speed of exhaust relative to the rocket.

Once the satellite is in a desired orbit, orbital adjustments and station-keeping maneuvers of a satellite requires fuel consumption. Let's *assume* that the fuel consumption $F(\omega, \zeta; t)$ of a satellite is a function of the angular velocity ω, thrust ζ and time t, described by the following equation

$$F(\omega, \zeta; t) = a(\omega - \omega_0)^2 + b(\zeta - \zeta_0)^2 + c\sin(\omega t), \tag{6}$$

where

1. **Quadratic terms** $a(\omega - \omega_0)^2$ **and** $b(\zeta - \zeta_0)^2$: These terms ensure that the fuel consumption increases as the angular velocity ω and thrust ζ deviate from their optimal values ω_0 and ζ_0, respectively. The coefficients a and b control how sensitive the fuel consumption is to changes in ω and ζ.

2. **Sinusoidal term** $c\sin(\omega t)$: This term introduces non-linearities to the fuel consumption model due to heterogeneities in space.

We wish to find the optimal angular velocity and thrust that minimizes the fuel consumption at every instant, t, of the satellite's orbital trajectory. Once these parameters are known, the robot controlling the satellite can adjust any deviations accordingly so that the fuel consumption can be minimized while maintaining the satellite's trajectory in the desired orbit. It may be noted that the scientific objectives of the satellite does not depend on the fuel minimization exercise by the robot.

2.2 Brief overview of optimization techniques

2.2.1 Lagrange multipliers for constrained optimization

Lagrange multipliers are a powerful technique used in optimization problems where the objective function needs to be minimized (or maximized) subject to one or more constraints.
Consider a function $f(x, y)$ that we want to minimize subject to a constraint given by $g(x, y) = c$, where c is a constant. The function $f(x, y)$ is called the objective function and $g(x, y) = c$ is the constraint.

To solve this problem using Lagrange multipliers, we introduce a new variable λ (called the Lagrange multiplier) and construct the Lagrangian function $\mathcal{L}(x, y, \lambda)$.

$$\mathcal{L}(x, y, \lambda) = f(x, y) + \lambda\,(g(x, y) - c)$$

Here, the term $\lambda(g(x, y) - c)$ adds the constraint to the objective function.

To find the critical points, we take the partial derivatives of the Lagrangian function with respect to x, y, and λ, and set them equal to zero.

$$\frac{\partial \mathcal{L}}{\partial x} = \frac{\partial f(x, y)}{\partial x} + \lambda \frac{\partial g(x, y)}{\partial x} = 0,$$

$$\frac{\partial \mathcal{L}}{\partial y} = \frac{\partial f(x, y)}{\partial y} + \lambda \frac{\partial g(x, y)}{\partial y} = 0,$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = g(x, y) - c = 0,$$

These equations form a system of equations that can be solved simultaneously to find the values of x, y, and λ.

2.2.2 Gradient descent for minimizing fuel consumption

Gradient descent is an optimization algorithm used to minimize a cost function by iteratively moving along the steepest descent path which is the negative gradient of the function. For the fuel consumption model $F(\omega, \zeta; t)$ given by Equation 6, the gradient with respect to the control variables ω and ζ can be computed as

$$\nabla F = \left(\frac{\partial F}{\partial \omega}, \frac{\partial F}{\partial \zeta} \right).$$

The update rules for ω and ζ at each iteration are given by

$$\omega^{(n+1)} = \omega^{(n)} - \eta_\omega \frac{\partial F}{\partial \omega},$$

$$\zeta^{(n+1)} = \zeta^{(n)} - \eta_\zeta \frac{\partial F}{\partial \zeta},$$

where η_ω and η_ζ are the learning rates that control the step size.

The objective is to find the values of ω and ζ that minimize $F(\omega, \zeta; t)$ ensuring optimal fuel consumption during orbital adjustments. For pseudocode refer A.4 and for further details refer Hass et al. 2022.

2.3 Problem statements

Consider a multi-stage rocket intended for launching a satellite of mass M_s into orbit. The rocket consists of three stages, each with its own mass M_i (for $i = 1, 2, 3$), where M_1 is the mass of the first stage, M_2 is the mass of the second stage, and M_3 is the mass of the third stage. Assume that external forces, such as gravity and atmospheric drag, are negligible during the rocket's ascent, and that c_r and S are constant for each stage.

Initially, the rocket's first stage has a mass of M_1 (excluding the payload), and its payload consists of an additional combined mass of the second and third stages and the satellite of mass M_s. The rocket's second stage has a mass of M_2 (excluding the payload), and its payload consists of an additional combined mass of the third stage, and the satellite of mass M_s. Thus, the subsequent stages operate similarly after the previous stage is jettisoned.

1. **Derive the final velocity:** Using the rocket equation 5, derive the expression for the final velocity v_f achieved after all three stages have been expended. Show that this velocity is given by

$$v_f = c_r \left[\log\left(\frac{M_1 + M_2 + M_3 + M_s}{SM_1 + M_2 + M_3 + M_s}\right) + \log\left(\frac{M_2 + M_3 + M_s}{SM_2 + M_3 + M_s}\right) + \log\left(\frac{M_3 + M_s}{SM_3 + M_s}\right)\right]. \tag{7}$$

2. **Minimization problem:** We aim to minimize the total mass of the rocket stages, $M = M_1 + M_2 + M_3$, subject to the constraint that the final velocity v_f as given by equation 7 is attained. To simplify the minimization problem, define new variables N_i so that the constraint equation can be expressed as

$$v_f = c_r \left(\log N_1 + \log N_2 + \log N_3\right).$$

Given this, show that

$$\frac{M_1 + M_2 + M_3 + M_s}{M_2 + M_3 + M_s} = \frac{(1-S)N_1}{1 - SN_1},$$
$$\frac{M_2 + M_3 + M_s}{M_3 + M_s} = \frac{(1-S)N_2}{1 - SN_2},$$
$$\frac{M_3 + M_s}{M_s} = \frac{(1-S)N_3}{1 - SN_3},$$

and conclude that

$$\frac{M + M_s}{M_s} = \frac{(1-S)^3 N_1 N_2 N_3}{(1 - SN_1)(1 - SN_2)(1 - SN_3)}.$$

3. **Minimization with Lagrange multipliers:** Verify that $\log\left(\frac{M+M_s}{M_s}\right)$ is minimized at the same values of M as the original function. Use the method of Lagrange multipliers to find the expression for N_i where this minimum occurs subject to the constraint $v_f = c_r \left(\log N_1 + \log N_2 + \log N_3\right)$.

4. **Expression for minimum mass:** Derive an expression for the minimum total mass M of the rocket stages as a function of the desired final velocity v_f.

5. **Application to orbital insertion:** To place a satellite into a low Earth orbit, approximately 160 kilometers (100 miles) above the Earth's surface, a final velocity of 28,000 kilometers per hour (17,500 miles per hour) is required. Consider the structural coefficient $S = 0.2$ and the exhaust speed $c_r = 9,600$ kilometers per hour (6,000 miles per hour).

   (a) Find the minimum total mass M of the rocket stages as a function of the satellite mass M_s.

   (b) Determine the individual masses M_1, M_2, and M_3 of each rocket stage as functions of M_s (noting that the stages are not of equal size).

6. **Fuel consumption extrema:**

   (a) Find the critical points for the fuel function $F(\omega, \zeta; t)$ in equation 6. Do critical points exist for any values of the constants involved in equation 6? Provide appropriate justification for your answer.

   (b) Using the second derivative test, analyze the nature of the critical points found.

7. **Gradient descent implementation:**

   (a) Implement the gradient descent algorithm for $t \in [0, 2\pi]$ (discretized into 10 equally spaced points) to minimize the fuel consumption function $F(\omega, \zeta; t)$. Take $a = b = \omega_0 = \zeta_0 = 1.0$ and $c = d = 0.5$ and consider the allowed values of ω and ζ in the range $[0, 3]$.

   (b) Analyze the effect of varying the learning rates η_ω and η_ζ on the convergence of the algorithm and the accuracy of the minima found. What learning rates did you find best to solve this problem?

   (c) How does changing the coefficients a, b, c affect the efficiency of the satellite's maneuvers? (you need to show this through appropriate plots and justification)

A Appendix

A.1 Fixed-point method

Input: Function $g(x)$, initial guess x_0, tolerance ϵ, maximum iterations $N_{\max}$
Output: Approximate solution x
Set $x \leftarrow x_0$
for $n = 1, 2, \ldots, N_{\max}$:
    $x_{\text{new}} \leftarrow g(x)$
    if $|x_{\text{new}} - x| < \epsilon$:
        break
    $x \leftarrow x_{\text{new}}$
return x

A.2 Newton's method

Input: Function $f(x)$, derivative $f'(x)$, initial guess x_0, tolerance ϵ, maximum iterations $N_{\max}$
Output: Approximate solution x
Set $x \leftarrow x_0$
for $n = 1, 2, \ldots, N_{\max}$:
    $x_{\text{new}} \leftarrow x - \frac{f(x)}{f'(x)}$
    if $|x_{\text{new}} - x| < \epsilon$:
        break
    $x \leftarrow x_{\text{new}}$
return x

A.3 Secant method

Input: Function $f(x)$, initial guesses x_0 and x_1, tolerance ϵ, maximum iterations $N_{\max}$
Output: Approximate solution x_1
for $n = 1, 2, \ldots, N_{\max}$:
    $x_{\text{new}} \leftarrow x_1 - f(x_1) \times \frac{x_1 - x_0}{f(x_1) - f(x_0)}$
    if $|x_{\text{new}} - x_1| < \epsilon$:
        break
    $x_0 \leftarrow x_1$
    $x_1 \leftarrow x_{\text{new}}$
return x_1

A.4 Gradient descent method

Input: Function $f(\mathbf{w})$, gradient $\nabla f(\mathbf{w})$, initial guess $\mathbf{w}_0$, learning rate α, tolerance ϵ, maximum iterations N_{max}
Output: Optimal $\mathbf{w}$
Set $\mathbf{w} \leftarrow \mathbf{w}_0$
for $n = 1, 2, \ldots, N_{\text{max}}$:
  Compute the gradient at $\mathbf{w}$: $\mathbf{g} \leftarrow \nabla f(\mathbf{w})$
  Update $\mathbf{w}$: $\mathbf{w}_{\text{new}} \leftarrow \mathbf{w} - \alpha \cdot \mathbf{g}$
  if $\|\mathbf{w}_{\text{new}} - \mathbf{w}\| < \epsilon$:
    break
  Set $\mathbf{w} \leftarrow \mathbf{w}_{\text{new}}$
return $\mathbf{w}$

References

1. Burden, Richard L. and J. Douglas Faires (2011). *Numerical Analysis*. 9th. Cengage Learning.
2. Casselman, Bill (2018). "Planetary Motion and Kepler's Equation". In: *Essays on Mathematical Astronomy* 12.
3. Hass, Joel R. et al. (2022). *Thomas' Calculus*. 15th. Pearson.
4. López, Rosario, Denis Hautesserres, and Juan Félix San-Juan (2018). "The solution of the generalized Kepler's equation". In: *Monthly Notices of the Royal Astronomical Society* 473.2, pp. 2583–2589.