# 自动控制理论——稳定性分析

## Python 综合应用与仿真考察

2025 年 12 月 7 日

---

**说明：** 本测试旨在考察将 PPT 中的稳定性理论（Routh 判据、相平面、Lyapunov 方法）转化为 Python 计算代码和仿真分析的能力。请结合 PPT 内容及 Python 编程完成以下题目。

## 题目 1：参数变化对系统稳定性的影响（基于 Routh 判据的验证）

**题目背景：** 参考 PPT 中的例子，已知闭环传递函数的特征方程与增益 $K$ 有关。PPT 通过手算 Routh 表得出 $0 < K < 128$ 时系统稳定。在实际工程中，参数往往不是固定的，我们需要直观地看到 $K$ 值变化如何导致系统从稳定走向不稳定。

1. **理论计算：** 选取一个具体的三阶系统特征方程，例如 $s^3 + 4s^2 + 5s + K = 0$。利用 PPT 中的 Routh 判据规则，手算系统稳定的 $K$ 值范围。
2. **Python 仿真：**
   - 使用 scipy.signal 定义该传递函数。
   - 编写一个循环，让 $K$ 从 0 增加到临界值的 1.5 倍。
   - 在每一步中，计算系统的极点（使用 numpy.roots 或 scipy.signal.tf2zpk），并在复平面上动态绘制“根轨迹”图，标记出极点穿过虚轴的时刻。
   - 选取三个 $K$ 值（小于临界值、等于临界值、大于临界值），分别使用 scipy.signal.step 绘制系统的阶跃响应曲线。
3. **分析：** 结合仿真结果，解释 PPT 中提到的“实部为零的特征根”在时域响应图中对应什么现象？

## 题目 2：二阶系统相平面分析与平衡点分类

**考察点：** 状态空间法、特征值分析、相轨迹绘制、平衡点类型（节点、鞍点、焦点）。

**题目背景：** PPT 详细对二阶系统的特征值分布与平衡点类型（如鞍点、稳定焦点、中心点等）进行了分类。为了深入理解这些抽象概念，需要通过相平面图来直观展示。

1. **模型构建：**考虑通用的二阶线性系统 $\dot{\mathbf{z}} = \mathbf{A}\mathbf{z}$。构造三个不同的矩阵 $\mathbf{A}$，分别对应 PPT 中提到的三种情况：
   - **情况 A（稳定节点）：**两个负实数特征值。
   - **情况 B（鞍点）：**一正一负实数特征值。
   - **情况 C（稳定焦点）：**实部为负的共轭复数特征值。
2. **Python 仿真：**
   - 使用 numpy.linalg.eig 计算你自己设计的矩阵特征值，验证符合上述分类。
   - 使用 matplotlib.pyplot.streamplot 或 quiver 函数，在相平面（$z_1$ vs $z_2$）上绘制这三种情况的向量场（Vector Field）和典型的相轨迹。
3. **分析：**观察情况 C 的相轨迹，解释 PPT 中"渐近稳定"的几何含义在图中是如何体现的？

## 题目 3：非线性系统的线性化与稳定性判断（Lyapunov 第一法）

**考察点：**非线性系统线性化、雅可比矩阵、Lyapunov 第一方法。

**题目背景：** PPT 提到了单摆系统存在两个平衡点（直上 A 和直下 B），其中 A 是不稳定的，B 是稳定的。PPT 介绍了通过泰勒展开和雅可比矩阵将非线性系统近似为线性系统来判断稳定性的方法。

1. **数学推导：**写出无阻尼单摆的非线性状态方程 $\dot{x}_1 = x_2, \dot{x}_2 = -\frac{g}{L}\sin(x_1)$。求解其两个平衡点 $(0, 0)$ 和 $(\pi, 0)$ 处的雅可比矩阵 $\mathbf{J}$。
2. **Python 仿真：**
   - 定义非线性系统的导数函数。
   - 使用 scipy.integrate.odeint 模拟单摆在两个平衡点附近的运动。
     - 对于 $(0, 0)$ 点，给一个微小扰动，观察系统是否在附近振荡（Lyapunov 意义下的稳定）。
     - 对于 $(\pi, 0)$ 点，给一个微小扰动，观察系统是否远离平衡点。
   - 计算两个平衡点处线性化矩阵的特征值，利用 Python 判断其稳定性。
3. **分析：**对照 PPT 解释为什么算出 $(\pi, 0)$ 处的特征值有一个正实部，就能断定原非线性系统在该点不稳定？

## 题目 4：Van der Pol 振荡器与极限环（综合案例分析）

**考察点：**非线性系统、Lyapunov 第一法局限性、大范围渐近稳定概念。
**题目背景：** PPT 提到了 Van der Pol 方程 $\ddot{x}-(1-x^2)\dot{x}+x=0$。这是一个经典的非线性系统，它既不是简单的渐近稳定，也不是完全的发散，而是存在极限环。

1. **Python 仿真：**
   - 将该二阶微分方程转换为状态空间形式。
   - 使用 scipy.integrate.odeint 从相平面上多个不同的初始位置(如 $(0.1,0),(2,2),(-3,-1)$)开始仿真系统响应。
   - 在同一张相平面图上绘制所有轨迹。
2. **可视化分析：**
   - 观察轨迹最终是否汇聚到一条闭合曲线上?
   - 计算原点 $(0,0)$ 处线性化系统的特征值。
3. **开放性讨论：**结合 PPT 渐近稳定和 Lyapunov 意义下的稳定的定义，讨论 Van der Pol 振荡器的原点稳定性。如果原点是不稳定的，为什么轨迹没有发散到无穷远？这与线性系统有何本质区别?

## 题目 5：Lyapunov 函数的可视化验证（Lyapunov 第二法）

**考察点：** Lyapunov 第二方法（直接法)、正定函数构造、能量观点。
**题目背景：** PPT 介绍了通过构造标量函数 $V(x)$ 来判别稳定性的方法。PPT 强调 $\dot{V}(x)$ 必须是半负定或负定的。这在几何上可以理解为状态轨迹总是“穿过”等高线向“谷底”运动。

1. **函数构造：**选取一个简单的二阶线性稳定系统（例如 $\dot{x}=-x$ 的二维扩展)，并构造一个标准的二次型 Lyapunov 函数 $V(\mathbf{z})=z_1^2+z_2^2$（即 PPT 提到的最简单形式)。
2. **Python 3D 可视化：**
   - 使用 matplotlib 的 plot_surface 绘制 $V(\mathbf{z})$ 的 3D 碗状曲面。
   - 在曲面上投影出系统的状态轨迹（Trajectory)。
   - 计算并绘制 $\dot{V}(\mathbf{z})$ 随时间变化的曲线。
3. **验证与分析：**
   - 验证在仿真过程中 $\dot{V}(t)$ 是否始终小于等于 0?
   - 如果将系统改为不稳定系统（例如将矩阵特征值改为正)，重复上述步骤，观察轨迹与 $V(\mathbf{z})$ 曲面的关系。这如何解释 PPT 中提到的“系统将不断地从外界吸收能量”?