

航天器姿态智能控制

本文探讨了航天器姿态控制系统的重要性及其在航天器控制领域的应用。

文章首先介绍了航天器姿态控制的基本概念，包括姿态的稳定性和精确度，以及它们对于航天器长期运行和任务完成的重要性。随后，文章回顾了姿态控制方法的发展历程，从经典的控制理论到现代控制理论，如最优控制理论、变结构控制技术和自适应控制技术。

文章重点介绍了航天器姿态控制的仿真框架，包括姿态动力学模块、姿态运动学模块、控制器模块、轨道动力学模块和空间环境模块。这些模块共同工作，以实现航天器姿态的精确控制。特别地，文章详细讨论了基于 BP 神经网络的 PID 自适应控制方法，这种方法能够通过学习系统性能来实现最佳组合的 PID 控制。

此外，文章还探讨了航天器姿态动力学模型，包括四元数姿态运动学方程和欧拉角姿态运动学方程。通过仿真模型，可以更好地理解和分析姿态控制系统在不同控制律下的行为，优化控制策略。

最后，文章通过实验结果展示了基础 PD 控制、基于 BP 神经网络的 PD 控制器以及基于遗传算法的 PD 控制器的性能对比。结果表明，基于 BP 神经网络和遗传算法的控制方法能够显著提高航天器姿态控制的性能。

本文所有代码均已开源于

https://github.com/LiuZiyue1016/spacecraft_attitude_control

关键词： 航天器姿态控制；仿真框架；BP 神经网络；PID 控制；遗传算法；姿态动力学模型

1、仿真框架

航天器姿态控制系统是航天器控制领域中至关重要的组成部分，它直接关系到航天器在轨后的稳定性和精确度。具体来说，航天器必须能够在预定的轨道或参考系中保持其方位或指向，并且在特定的误差范围内维持稳定。这不仅涉及对惯性系或其他参考系、引力中心体的精确控制，还对航天器的长期运行、任务完成以及对其他天体的观测、数据采集等至关重要。随着空间技术的发展，航天器的规模愈加庞大，姿态控制的性能要求也日益提高。从上世纪六、七十年代经典的航天器姿态控制研究，到八十年代随着最优控制理论、变结构控制技术、自适应控制技术等现代控制理论的提出，姿态控制方法不断演进。到当前，航天器姿态智能控制方法的研究逐步展开，并且在航天器姿态动力学与控制理论的结合上取得了令人瞩目的进展。

在这些研究中，姿态控制的设计方法可以大致分为两类：一类是传统的经典设计方法，另一类则是基于最优控制理论和智能控制的现代方法。

经典设计方法主要包括频域法、根轨迹法和基于系统状态方程的设计方法。这些方法依赖于手动计算系统的频域图（如奈奎斯特图、波德图、尼科尔斯图）以及根轨迹图和状态方程。在系统结构变化时，频域特性、根轨迹特性以及状态方程的变化可能导致较大的调整，因此这一方法的主要问题是系统模型的低重复利用率、设计过程中的继承性差、需要频繁修改系统参数和结构，极大地增加了设计的复杂性和重复劳动。

随着控制理论的发展，现代控制方法逐渐向最优控制理论倾斜。这些方法的目标是通过对控制系统进行优化，使得系统在特定的性能指标下达到最优。最优控制理论包括极大值原理、李雅普诺夫法、梯度法、线性规划和动态规划等。这些方法使用积分泛函作为性能指标，替代了传统的设计指标（如超调量、调整时

间、相位裕量、增益裕量等)。这种方法能够在复杂的控制任务中灵活适应不同的需求, 尤其在高精度的航天器姿态控制中具有优势。

通过采用最优控制方法, 设计者可以在多维性能指标下找到最佳控制律, 并且减少了反复修改的工作量。最优控制方法的引入使得航天器姿态控制的设计不仅更加高效, 而且具有更强的系统鲁棒性和适应性。

为了应对传统控制方法中存在的低重用性和高复杂性问题, 现代航天器姿态控制的设计趋向于建立通用的仿真框架。通过使用 Simulink 等仿真工具, 可以直观且高效地描述待开发系统的各个环节, 并利用模块化设计的思想来构建整个系统的仿真模型。在这一框架下, 系统可以分为五大基本模块: 姿态动力学模块、姿态运动学模块、控制器模块、轨道动力学模块以及空间环境模块。

姿态动力学模块: 该模块负责计算航天器的动力学行为, 结合航天器的质量分布、力矩作用等因素, 提供精确的动力学模型。

姿态运动学模块: 用于描述航天器不同姿态表示之间的转换关系。例如, 将航天器的四元数、欧拉角或旋转矩阵转换为其他形式的表示, 方便控制律的设计和仿真分析。

控制器模块: 这一模块是待设计的核心部分, 负责实现航天器的姿态控制律。控制器的设计需要根据所选用的控制理论(如经典 PID 控制、最优控制或智能控制)来进行优化, 以确保在动态环境下对航天器姿态的精确调节。

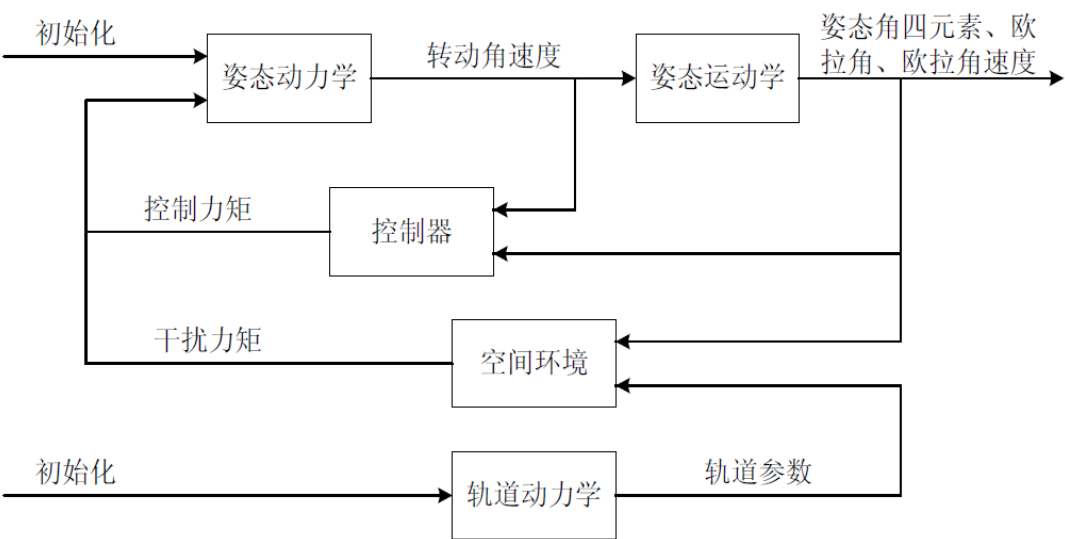
轨道动力学模块: 此模块用于模拟航天器在轨道上的运动, 描述轨道动力学对姿态控制的影响, 帮助分析航天器在不同轨道下的姿态变化。

空间环境模块: 描述外部空间环境对航天器姿态控制的影响。例如, 太阳辐射压力、地球磁场、气动力等因素, 这些都可能对姿态控制系统造成扰动, 需要

在设计中加以考虑。

通过上述模块化的设计方法，航天器姿态控制系统的仿真框架不仅能够提供高效、灵活的系统设计手段，还能显著提高系统的重用性，减少因设计修改和调整带来的重复劳动。此外，该仿真框架有助于快速验证控制律设计的有效性，从而加速航天器姿态控制系统的开发和优化。

随着航天器技术和控制理论的不断进步，姿态控制系统的设计方法日益复杂且多样化。传统的经典控制方法仍然有其应用价值，但最优控制和智能控制方法在提升设计效率和系统性能方面展现了巨大的潜力。通过引入基于仿真框架的设计方法，可以有效提升系统的开发效率，并确保航天器姿态控制在各种复杂环境下的稳定性和精度。未来，随着更多先进控制技术的出现，航天器姿态控制系统将变得更加智能、高效，并能够应对更为复杂和苛刻的任务需求。



2、航天器姿态动力学模型

2.1、航天器姿态动力学方程

航天器的姿态运动是指其绕自身质心的转动。当航天器被看作为刚体时，其姿态动力学方程就可以直接从刚体相对于质心的动量矩定理导出。

航天器的姿态运动是指其绕自身质心的转动。当航天器被看作为刚体时，其姿态动力学方程就可以直接从刚体相对于质心的动量矩定理导出。

设航天器在空间以角速度 ω 旋转，其动量矩为 H 。为了方便起见，基准点选航天器本体坐标系 O_{xyz} 的原点，也即航天器质心 O ， M 是作用在航天器相对于质心 O 的合外力矩，因此航天器的动量矩为

$$H = \int_m r \times \frac{dr}{dt} dm$$

式中，矢量 r 是刚体内质量元 dm 相对于质心的矢径； $\frac{dr}{dt}$ 是质量元在空间相对于质心的速度矢量； m 为航天器的总质量。于是在本体坐标系中，刚体的 ω ， H ， r 和 M 可以分别表示成

$$\begin{aligned}\omega &= \omega_x \mathbf{i} + \omega_y \mathbf{j} + \omega_z \mathbf{k} \\ H &= h_x \mathbf{i} + h_y \mathbf{j} + h_z \mathbf{k} \\ r &= x \mathbf{i} + y \mathbf{j} + z \mathbf{k} \\ M &= M_x \mathbf{i} + M_y \mathbf{j} + M_z \mathbf{k}\end{aligned}$$

式中， i, j, k 是航天器本体坐标系各轴的单位矢量，上式右端的系数则是相应矢量沿各坐标轴的分量。将动量矩对时间 t 求取导数，得动量矩 H 在空间的变化率，即

$$\frac{dH}{dt} = \dot{h}_x \mathbf{i} + \dot{h}_y \mathbf{j} + \dot{h}_z \mathbf{k} + h_x \frac{d\mathbf{i}}{dt} + h_y \frac{d\mathbf{j}}{dt} + h_z \frac{d\mathbf{k}}{dt}$$

由于刚体在空间中以 ω 的角速度旋转，所以与其固连的本体坐标系各轴方向也在相应变化。由理论力学中的泊桑公式，坐标轴单位矢量的导数为

$$\frac{d\mathbf{i}}{dt} = \omega \times \mathbf{i}, \quad \frac{d\mathbf{j}}{dt} = \omega \times \mathbf{j}, \quad \frac{d\mathbf{k}}{dt} = \omega \times \mathbf{k}$$

代入并根据动量矩定理得

$$M = \frac{dH}{dt} = \dot{H} + \omega \times H$$

因为

$$\omega \times H = (\omega_y h_z - \omega_z h_y) \mathbf{i} + (\omega_z h_x - \omega_x h_z) \mathbf{j} + (\omega_x h_y - \omega_y h_x) \mathbf{k}$$

所以在航天器本体坐标系中可以展开为

$$\mathbf{M} = M_x \mathbf{i} + M_y \mathbf{j} + M_z \mathbf{k} = (\dot{h}_x + \omega_y h_z - \omega_z h_y) \mathbf{i} + (\dot{h}_y + \omega_z h_x - \omega_x h_z) \mathbf{j} + (\dot{h}_z + \omega_x h_y - \omega_y h_x) \mathbf{k}$$

其在各轴的分量为

$$M_x = \dot{h}_x + \omega_y h_z - \omega_z h_y$$

$$M_y = \dot{h}_y + \omega_z h_x - \omega_x h_z$$

$$M_z = \dot{h}_z + \omega_x h_y - \omega_y h_x$$

或表示成矩阵矢量形式

$$\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} \dot{h}_x \\ \dot{h}_y \\ \dot{h}_z \end{bmatrix} + \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \begin{bmatrix} h_x \\ h_y \\ h_z \end{bmatrix}$$

对距离矢量求导可得

$$\frac{d\mathbf{r}}{dt} = \dot{\mathbf{r}} + \boldsymbol{\omega} \times \mathbf{r}$$

若刚体内各质点相对于质心的位置不变，所描述的动量矩即为

$$\mathbf{H} = \int_m \mathbf{r} \times (\boldsymbol{\omega} \times \mathbf{r}) dm$$

利用矢量叉乘可得

$$\begin{aligned} \mathbf{r} \times (\boldsymbol{\omega} \times \mathbf{r}) &= [\omega_x(y^2 + z^2) - \omega_y(xy) - \omega_z(xz)] \mathbf{i} + \\ &[-\omega_x(xy) + \omega_y(x^2 + z^2) - \omega_z(yz)] \mathbf{j} + \\ &[-\omega_x(xz) - \omega_y(yz) + \omega_z(x^2 + y^2)] \mathbf{k} \end{aligned}$$

上式联立可得

$$\begin{aligned} h_x &= I_x \omega_x - I_{xy} \omega_y - I_{xz} \omega_z \\ h_y &= -I_{xy} \omega_x + I_y \omega_y - I_{yz} \omega_z \\ h_z &= -I_{xz} \omega_x - I_{yz} \omega_y + I_z \omega_z \end{aligned}$$

即

$$\begin{bmatrix} h_x \\ h_y \\ h_z \end{bmatrix} = \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{xy} & I_y & -I_{yz} \\ -I_{xz} & -I_{yz} & I_z \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \stackrel{\text{def}}{=} \mathbf{I} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

式中, \mathbf{I} 为惯性矩阵; I_x, I_y, I_z 分别为刚体绕坐标轴 Ox, Oy, Oz 的转动惯量;

I_{xy}, I_{yz}, I_{xz} 称为惯量积。它们分别为

$$I_x = \int_m (y^2 + z^2)dm, I_y = \int_m (x^2 + z^2)dm, I_z = \int_m (y^2 + x^2)dm$$

$$I_{xy} = \int_m (xy)dm, I_{yz} = \int_m (yz)dm, I_{xz} = \int_m (xz)dm$$

惯量积的数值可正可负，它们与坐标系的选取密切相关。如果在某一坐标系中， $I_{xy} = I_{yz} = I_{xz} = 0$ ，则该坐标系称为主轴坐标系， Ox, Oy, Oz 轴就是刚体的主惯量轴。

因此，如果取航天器的本体坐标系为主轴坐标系，则有

$$h_x = I_x \omega_x$$

$$h_y = I_y \omega_y$$

$$h_z = I_z \omega_z$$

把它们代入欧拉力矩方程式，并忽略质量变化，就可以得到

$$I_x \frac{d\omega_x}{dt} + \omega_y \omega_z (I_z - I_y) = M_x$$

$$I_y \frac{d\omega_y}{dt} + \omega_x \omega_z (I_x - I_z) = M_y$$

$$I_z \frac{d\omega_z}{dt} + \omega_x \omega_y (I_y - I_x) = M_z$$

这就是基于本体坐标系的航天器姿态动力学方程组，也可以表示为

$$M_c + T_d = I\dot{\omega} + \omega \times I\omega$$

其中 M_c 为操控力矩， T_d 为干扰力矩。

2.2、四元数姿态运动学方程

四元数姿态运动学方程为

$$\dot{\mathbf{q}} = \frac{1}{2} \mathbf{q} \otimes \mathbf{A}(\omega_r)$$

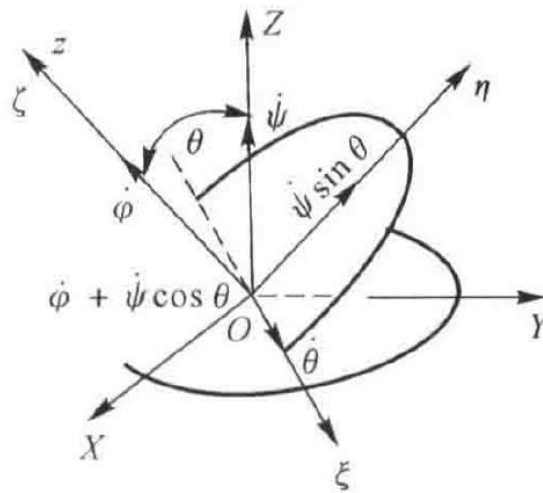
式中 $\mathbf{A}(\omega_r) = [0 \quad \omega_r^T]^T$ 或表示为矩阵形式

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \begin{bmatrix} 0 \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

四元数运动学方程无奇异点和三角函数计算，其计算量仅是欧拉角运动学方程的 35%左右，且计算更为精确。因此，四元数描述法更具优越性，在航天器姿态控制研究中得到了广泛应用。

2.3、欧拉角姿态运动学方程

欧拉角姿态运动学方程与欧拉角的旋转顺序有关，当以“3-1-3”为例，将航天器相对参考坐标系的转动角速度 ω ，在机体坐标系中的分量 $\omega_x, \omega_y, \omega_z$ 用欧拉角表示，从而推导出航天器的姿态运动学方程。



如图所示，将角速度 $\dot{\psi}$ ， $\dot{\theta}$ 和 $\dot{\phi}$ 投影到机体坐标系，其结果表示如下

$$\begin{aligned} \dot{\phi} &= \omega_z - (\omega_x \sin \phi + \omega_y \cos \phi) \cot \theta \\ \dot{\theta} &= \omega_x \cos \phi - \omega_y \sin \phi \\ \dot{\psi} &= (\omega_x \sin \phi + \omega_y \cos \phi) \csc \theta \end{aligned}$$

即为航天器的一组姿态运动学方程

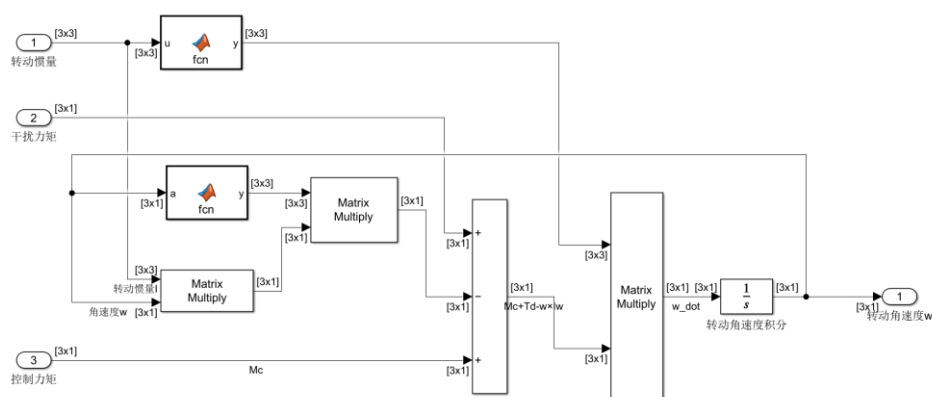
3、Simulink 程序设计

3.1、动力学与运动学模块设计

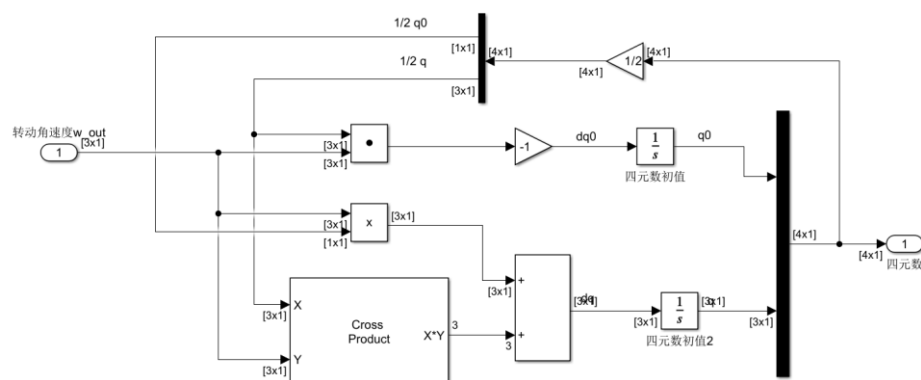
实现完成欧拉动力学方程计算的模块，将转动惯量、控制力矩、干扰力矩作为输入计算得到绝对角速度导数 $\dot{\omega}$ ，因此进行变形，得到如下表达式

$$\dot{\omega} = I^{-1}(M_c + T_d - \omega \times I \omega)$$

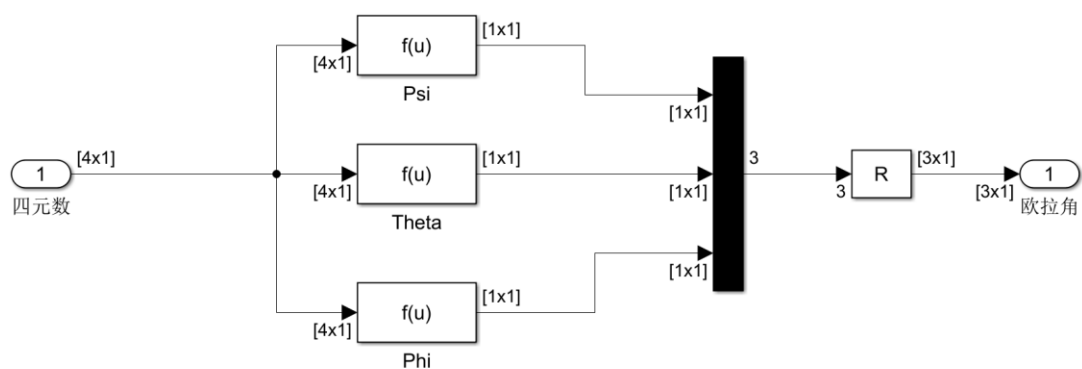
然后对 $\dot{\omega}$ 积分得到绝对角速度 ω ，同时作为下一次计算的输入，具体模型如图所示。



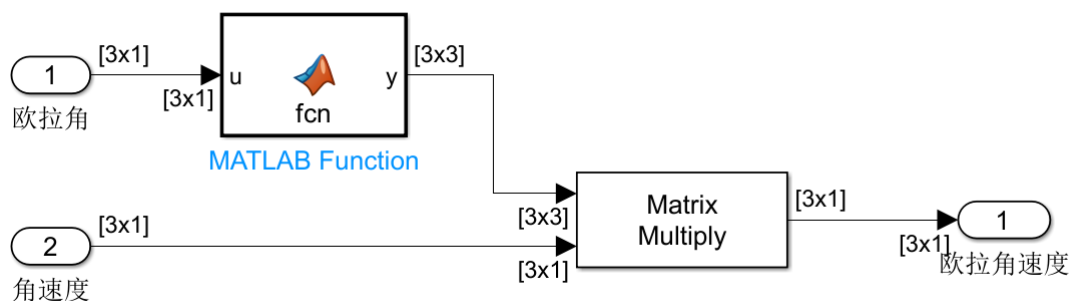
接着利用转动角速度求解四元数



利用四元数求解欧拉角



利用欧拉角和转动角速度求解欧拉角速度



在航天器姿态控制系统的仿真框架中，动力学模块和运动学模块输出的姿态信息被作为输入，通过设计的控制律计算得到所需的控制力矩。这些力矩作为控制模块的输出信号，再传递回动力学模块，形成一个闭环反馈控制系统。通过这种方式，姿态动力学与控制的仿真流程得以完成，最终实现了整个系统模型框架的开发。这一过程的核心在于控制模块与特定控制律的密切结合，当需要研究不同的控制律时，可以相应地开发和替换控制模块。为了说明这一过程，本文将飞轮 PD 控制为例，详细展示如何设计和实现控制律，并给出整个系统控制律设计的 Simulink 模型。

重点是刚体航天器相对于惯性空间的稳定控制问题。给出了基本的仿真流程，并实现了几个关键的仿真模块设计，这不仅用于验证航天器姿态动力学与控制仿真框架在姿态控制律设计中的有效性，也为后续的姿态控制算法的开发

和研究提供了必要的实验基础。通过这个仿真框架，能够深入理解和分析姿态控制系统在不同控制律下的行为，优化控制策略，从而提升航天器的姿态控制精度和鲁棒性。

在这一仿真框架下，每个模块的功能都清晰而独立，模块间的相互作用通过输入输出的方式进行通信。具体而言，姿态动力学模块提供了航天器的动态行为模型，包括航天器的转动惯量矩阵、外部力矩等，运动学模块则提供不同姿态描述方式（如四元数、欧拉角等）之间的转换关系。控制模块根据这些输入信息，采用预定的控制律计算所需的控制力矩，并将其传递给动力学模块，以调整航天器的姿态。通过这种模块化设计，能够灵活地调整不同控制策略的实现方式，并能够在设计阶段快速评估各种控制律的性能。

在控制律设计中，飞轮 PD 控制策略作为一种经典的姿态控制方法，能够有效地调整航天器的姿态误差和角速度，确保航天器姿态的稳定性和精度。飞轮的作用是通过控制力矩的施加，调节航天器的转动状态，而 PD 控制器则通过比例项（P）和微分项（D）来实时调整控制力矩，以最小化姿态误差和角速度误差。在 Simulink 模型中，控制器的设计可根据这些原理进行实现，并与其他模块（如动力学模块和运动学模块）连接，形成一个完整的仿真系统。

虽然当前的模型框架已经能够完成基本的姿态控制仿真，但为了增强模型的工程实用性和准确性，还需要进一步完善轨道动力学模块和空间环境模块。轨道动力学模块将模拟航天器在不同轨道上的运动，并分析轨道扰动对姿态控制系统的影响。空间环境模块则考虑外部空间环境因素，如太阳辐射、地球磁场、气动力扰动等，这些因素会对航天器的姿态产生一定影响，因此在设计控制系统时需要考虑这些环境因素的扰动对姿态控制系统的影响。通过进一步细

化这些模块，可以使得仿真模型更加接近实际的工程应用场景，并为航天器姿态控制系统的优化提供更加全面的参考。

通过对当前仿真框架的不断完善，未来可以扩展该模型的适用范围，研究更加复杂的姿态控制算法，例如自适应控制、鲁棒控制和智能控制等，以应对复杂的动态环境和任务需求。同时，还可以通过引入多传感器数据融合技术，提高姿态测量的精度和稳定性，进一步优化航天器姿态控制的效果。最终，该框架将为航天器姿态控制系统的设计和验证提供一个强有力的工具，推动未来航天任务中姿态控制技术的发展。

3.2、控制模块设计

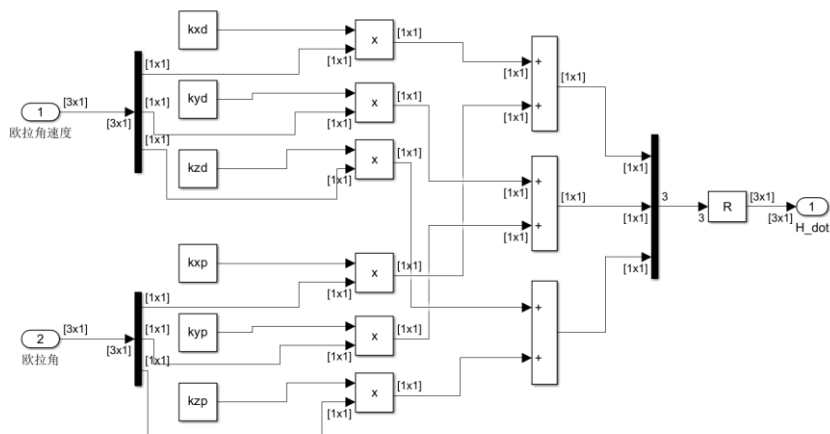
考虑一类带动量轮系统的整舱零动量航天器，这样的航天器利用飞轮机构的转动惯量变化产生的反作用力矩来对航天器进行姿态稳定控制，认为飞轮角动量为 \mathbf{H} ，则控制力矩为

$$\mathbf{M}_c = -\dot{\mathbf{H}} + \mathbf{H} \times \boldsymbol{\omega} \quad (6)$$

针对该类航天器姿态运动特点，基于其动力学模型对飞轮的反作用力矩进行设计，不考虑姿态三通道间的耦合，采用 PID 控制理论设计 PD 控制器，得到

$$\dot{\mathbf{H}} = \begin{bmatrix} \dot{H}_x \\ \dot{H}_y \\ \dot{H}_z \end{bmatrix} = \begin{bmatrix} k_{xp}\psi + k_{xd}\dot{\psi} \\ k_{yp}\theta + k_{yd}\dot{\theta} \\ k_{zp}\varphi + k_{zd}\dot{\varphi} \end{bmatrix}$$

Simulink 程序设计如图所示



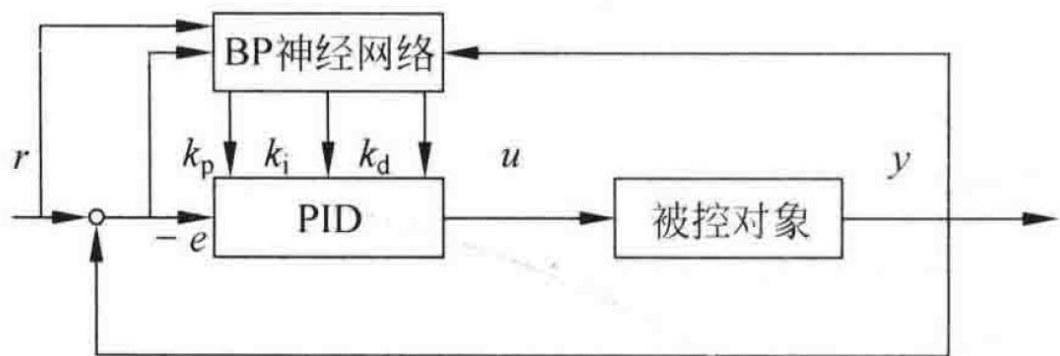
4、基于 BP 神经网络的 PID 自适应控制

PID 控制要想取得好的控制效果，就必须调整好比例、积分和微分三种控制的作用，形成控制量中相互配合又相互制约的关系。神经网络具有逼近任意非线性函数的能力，而且结构和学习算法简单明确，可以通过对系统性能的学习来实现最佳组合的 PID 控制。采用基于 BP 神经网络的 PID 自适应控制，可以建立参数 k_p 、 k_i 、 k_d 自学习的神经 PID 控制，从而达到参数自行调整的目的。

实例控制器由以下两部分组成：

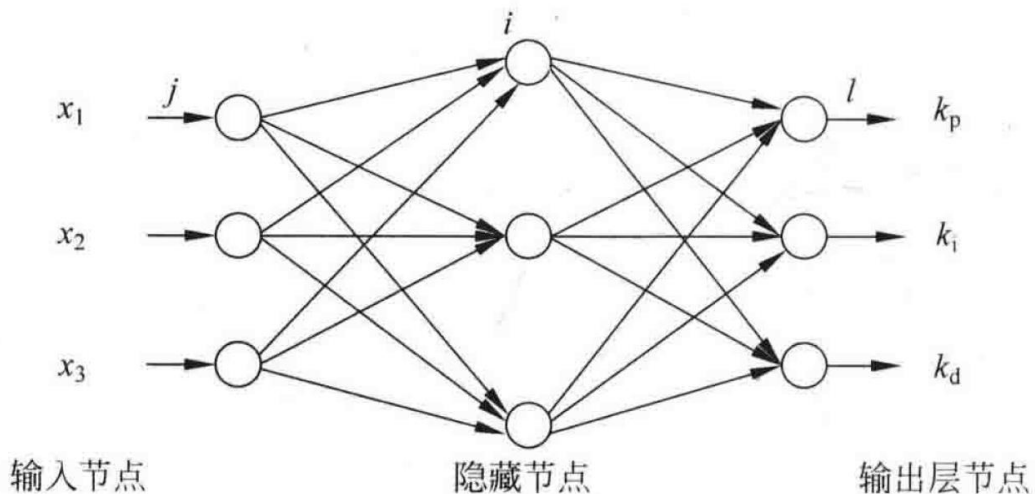
(1) 经典的 PID 控制器：直接对被控对象进行闭环控制，仍然是靠改变三个参数 k_p 、 k_i 、 k_d 来获得满意的控制效果。

(2) 神经网络：根据系统的运行状态，调节 PID 控制器的参数，以期达到某种性能指标的最优化。采用如图 12-3 所示的系统结构，即使得输出层神经元的输出状态对应于 PID 控制器的三个可调参数 k_p 、 k_i 、 k_d ，通过神经网络的自身学习和加权系数调整，从而使其稳定状态对应于某种最优控制规律下的 PID 的控制器的各个参数。



4.1、BP 神经网络

BP 神经网络一般为如图形式，采用三层结构：一个输入层、一个隐含层、一个输出层。 j 表示输入层节点， i 表示隐层节点， l 表示输出层节点。输入层有 m 个输入节点，隐含层有 q 个隐含节点，输出层有 3 个输出节点。输入节点对应所选的系统运行状态量，如系统不同时刻的输入量和输出量、偏差量等。输出层神经元活化函数取非负的 Sigmoid 函数。



由图可见，此处 BP 神经网络的输入层输出为

$$O_j^{(1)} = x(j), \quad j = 1, 2, 3, \dots, m$$

隐含层输入为

$$\text{net}_i^{(2)}(k) = \sum_{j=0}^m w_{ij}^{(2)} O_j^{(1)}$$

隐含层输出为

$$O_i^{(2)}(k) = g\left(\text{net}_i^{(2)}(k)\right), \quad i = 1, 2, 3, \dots, q$$

其中, $w_{ij}^{(2)}$ 为输入层到隐含层的加权系数, 上标 (1)、(2)、(3) 分别代表输入层、隐含层、输出层, $g(x)$ 为正负对称的 Sigmoid 函数, 即

$$g(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

最后网络输出层三个节点的输入为

$$\text{net}_l^{(3)}(k) = \sum_{i=0}^q w_{il}^{(3)} O_i^{(2)}(k)$$

最后输出层的输出为

$$o_l^{(3)}(k) = f\left(\text{net}_l^{(3)}(k)\right), \quad l = 1, 2, 3$$

其中, $w_{il}^{(3)}$ 为隐层到输出层的加权系数, 输出层神经元活化函数为

$$f(x) = \frac{1}{1 + e^{-x}}$$

取性能指标函数

$$E(k) = \frac{1}{2} (r(k) - y(k))^2$$

用梯度下降法修正网络的权系数, 并附加一个使搜索快速收敛到全局极小的惯性项, 则有

$$\Delta w_{li}^{(3)}(k) = -\eta \frac{\partial E(k)}{\partial w_{li}^{(3)}} + \alpha \Delta w_{li}^{(3)}(k-1)$$

其中, η 为学习率, α 为惯性系数, 且

$$\frac{\partial E(k)}{\partial w_{li}^{(3)}} = \frac{\partial E(k)}{\partial y(k)} \cdot \frac{\partial y(k)}{\partial u(k)} \cdot \frac{\partial u(k)}{\partial O_l^{(3)}(k)} \cdot \frac{\partial O_l^{(3)}(k)}{\partial \text{net}_l^{(3)}(k)} \cdot \frac{\partial \text{net}_l^{(3)}(k)}{\partial w_{li}^{(3)}}$$

这里需要用到变量 $\frac{\partial y(k)}{\partial u(k)}$, 由于模型可以未知, 所以 $\frac{\partial y(k)}{\partial u(k)}$ 未知, 但是可以测

出 $u(k)$ 、 $y(k)$ 的相对变化量, 即

$$\frac{\partial y}{\partial u} = \frac{y(k) - y(k-1)}{u(k) - u(k-1)}$$

由此带来的计算上的误差可以通过调整学习速率 η 来补偿。这样做一方面可以简化运算，另一方面避免了当 $u(k)$ 和 $u(k-1)$ 很接近时导致式(12-1)趋于无穷的情况。这种替代在算法上是可以的，因为 $\frac{\partial y(k)}{\partial u(k)}$ 是式中的一个乘积因子，它的符号的正负决定着权值变化的方向，而数值变化的大小只影响权值变化的速度，但是权值变化的速度可以通过学习步长加以调节。

由式

$$u(k) = u(k-1) + O_1^{(3)}(e(k) - e(k-1)) + O_2^{(3)}e(k) + O_3^{(3)}(e(k) - 2e(k-1) + e(k-2))$$

可得

$$\begin{aligned}\frac{\partial u(k)}{\partial O_1^{(3)}(k)} &= e(k) - e(k-1) \\ \frac{\partial u(k)}{\partial O_2^{(3)}(k)} &= e(k) \\ \frac{\partial u(k)}{\partial O_3^{(3)}(k)} &= e(k) - 2e(k-1) + e(k-2)\end{aligned}$$

这样，可得 BP 神经网络输出层权值计算公式为

$$\Delta w_{li}^{(3)}(k) = \eta e(k) \frac{\partial y(k)}{\partial u(k)} \frac{\partial u(k)}{\partial O_l^{(3)}(k)} f'(\text{net}_l^{(3)}(k)) O_i^{(2)}(k) + \alpha \Delta w_{li}^{(3)}(k-1)$$

则有

$$\begin{aligned}\Delta w_{li}^{(3)}(k) &= e(k) \text{sgn} \left(\frac{y(k) - y(k-1)}{u(k) - u(k-1)} \right) \eta \frac{\partial u(k)}{\partial O_l^{(3)}(k)} f'(\text{net}_l^{(3)}(k)) O_i^{(2)}(k) + \alpha \Delta w_{li}^{(3)}(k-1), \\ i &= 1, 2, \dots, q\end{aligned}$$

可令

$$\delta_l^{(3)} = e(k) \text{sgn} \left(\frac{y(k) - y(k-1)}{u(k) - u(k-1)} \right) \frac{\partial u(k)}{\partial O_l^{(3)}(k)} f'(\text{net}_l^{(3)}(k))$$

则上式可得

$$\Delta w_{li}^{(3)}(k) = \eta \delta_l^{(3)} O_i^{(2)}(k) + \alpha \Delta w_{li}^{(3)}(k-1)$$

同理可得隐含层权计算公式为

$$\Delta w_{ij}^{(2)}(k) = \eta g'(\text{net}_i^{(2)}(k)) \sum_{l=1}^3 \delta_l^{(3)} w_{li}^{(3)}(k) O_j^{(1)}(k) + \alpha \Delta w_{li}^{(2)}(k-1), i = 1, 2, \dots, q$$

令 $\delta_i^{(2)} = g f'(\text{net}_i^{(2)}(k)) \sum_{l=1}^3 \delta_l^{(3)} w_{li}^{(3)}(k)$, 则

$$\Delta w_{ij}^{(2)}(k) = \eta \delta_i^{(2)} O_j^{(1)}(k) + \alpha \Delta w_{li}^{(2)}(k-1), i = 1, 2, \dots, q$$

该控制器的算法如下

(1) 确定 BP 神经网络的结构, 即确定输入节点数 m 和隐含层节点数 q , 并给定各层加权系数的初值 $w_{ij}^{(0)}$ 和 $w_{ij}^{(0)}$, 选定学习速率 η 和惯性系数 α , 此时 $k = 1$ 。

(2) 采样得到 $r_{in}(k)$ 和 $y_{out}(k)$, 计算该时刻误差 $error(k) = r_{in}(k) - y_{out}(k)$ 。

(3) 计算神经网络 NN 各层神经元的输入、输出, NN 输出层的输出即为 PID 控制器的三个可调参数 k_p 、 k_i 、 k_d 。

(4) 根据经典增量数字 PID 的控制算法计算 PID 控制器的输出

$$u(k) = u(k-1) + K_p(error(k) - error(k-1)) + K_i error(k) + K_d(error(k) - 2error(k-1) + error(k-2))$$

(5) 进行神经网络学习, 在线调整加权系数 $w_{ij}^{(k)}$ 和 $w_{ij}^{(k)}$ 实现 PID 控制参数的自适应调整。

(6) 置 $k = k + 1$, 返回到步骤 (1)。

5、基于遗传算法的 PID 控制器

遗传算法的核心是优化目标函数。对于 PID 控制器, 通常优化的目标是控制系统的性能指标, 比如积分绝对误差 (IAE)、积分平方误差 (ISE)、积分时间绝对误差 (ITAE) 等, 目标函数可以表示为

$$J = w_1 \cdot \text{IAE} + w_2 \cdot \text{ISE} + w_3 \cdot \text{ITAE}$$

其中, w_1, w_2, w_3 是权重。

将 PID 参数 k_p, k_i, k_d 作为遗传算法中的染色体进行编码。直接将 k_p, k_i, k_d 表示为染色体的连续变量

$$[k_p, k_i, k_d]$$

初始化种群时生成一个包含多个染色体（个体）的种群，每个染色体表示一组随机的 PID 参数

$$[k_p^1, k_i^1, k_d^1], [k_p^2, k_i^2, k_d^2], \dots, [k_p^n, k_i^n, k_d^n]$$

其中, n 是种群大小。

对每个个体的染色体，使用适应度函数评估其控制系统性能，计算适应度值 $f(x)$ 。适应度函数通常基于目标函数值的倒数

$$f(x) = \frac{1}{J + \epsilon}$$

其中, ϵ 是一个很小的值，用于防止分母为零。

通过仿真模型或实际控制系统，评估不同参数组合的性能。

遗传算法通过选择、交叉和变异来优化种群中的个体，选择是根据适应度值选择优秀个体（如轮盘赌选择、锦标赛选择）。交叉是对两个父代染色体进行交叉，生成子代染色体（如单点交叉、两点交叉）。变异是随机改变子代染色体中的某些基因，以增加种群的多样性。

遗传算法通常在达到最大迭代次数，或目标函数值低于某个阈值时停止迭代，迭代结束以后，选择适应度值最高的个体，其对应的 k_p, k_i, k_d 即为 PID 控制器的最优参数。

6、实验结果

6.1、基础 PD 控制

航天器惯量参数为

$$\mathbf{I} = \begin{bmatrix} 1759 & 0 & 0 \\ 0 & 2758 & 0 \\ 0 & 0 & 2171 \end{bmatrix} (\text{单位: } kg \cdot m^2)$$

航天器的飞轮角动量在航天器体坐标轴上投影的值为: $H_x = H_y = H_z =$

$30N \cdot m \cdot s$ 且飞轮固定安装。

设航天器在控制力矩作用下, 将姿态从初始状态

$$\begin{cases} \varphi(t_0) = 4^\circ & \omega_x(t_0) = 0 \\ \theta(t_0) = 3^\circ & \omega_y(t_0) = 0 \\ \psi(t_0) = 2^\circ & \omega_z(t_0) = 0 \end{cases}$$

控制到精度为

$$\begin{cases} \varphi(t_0) = \pm 0.1^\circ \\ \theta(t_0) = \pm 0.1^\circ \\ \psi(t_0) = \pm 0.1^\circ \end{cases}$$

的稳定状态。在整个飞行过程中, 航天器始终受到

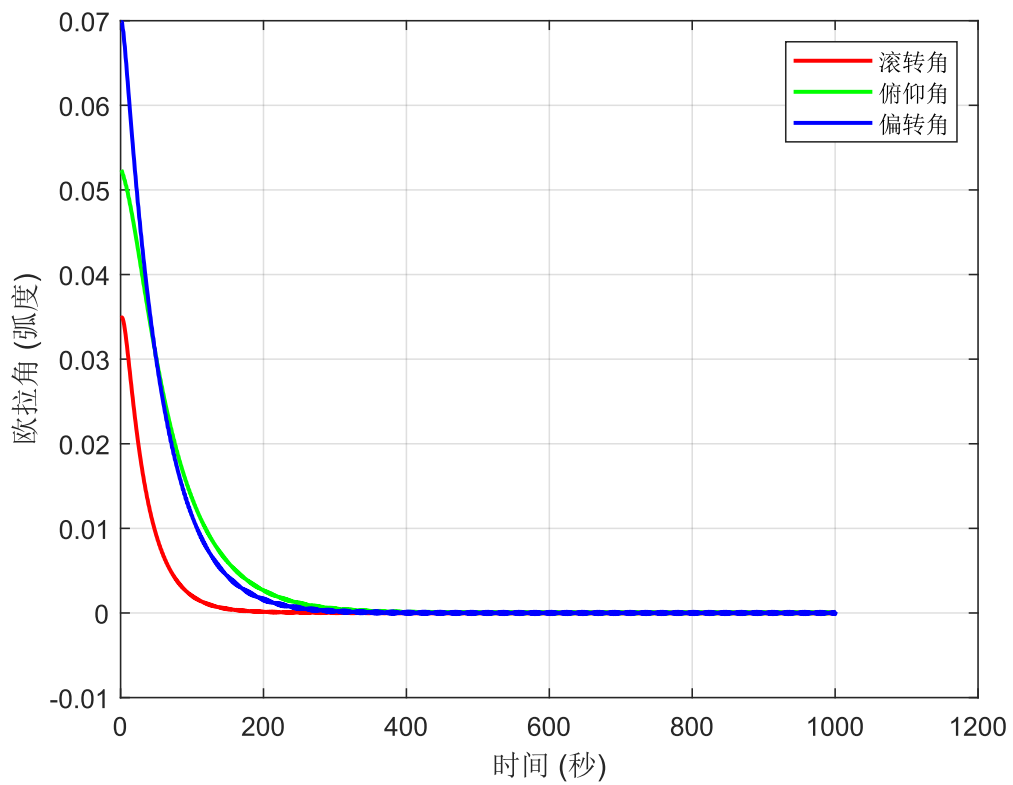
$$\mathbf{T}_d = \begin{bmatrix} \sin(3t) \\ 2 * \cos(3t + \pi/3) \\ 3 * \cos(3t) \end{bmatrix}$$

的干扰力矩

设置控制参数为

$$\begin{cases} k_{xp} = 10.2425 & k_{xd} = 394.7569 \\ k_{yp} = 2.3318 & k_{yd} = 165.8386 \\ k_{zp} = 11.1371 & k_{zd} = 636.8378 \end{cases}$$

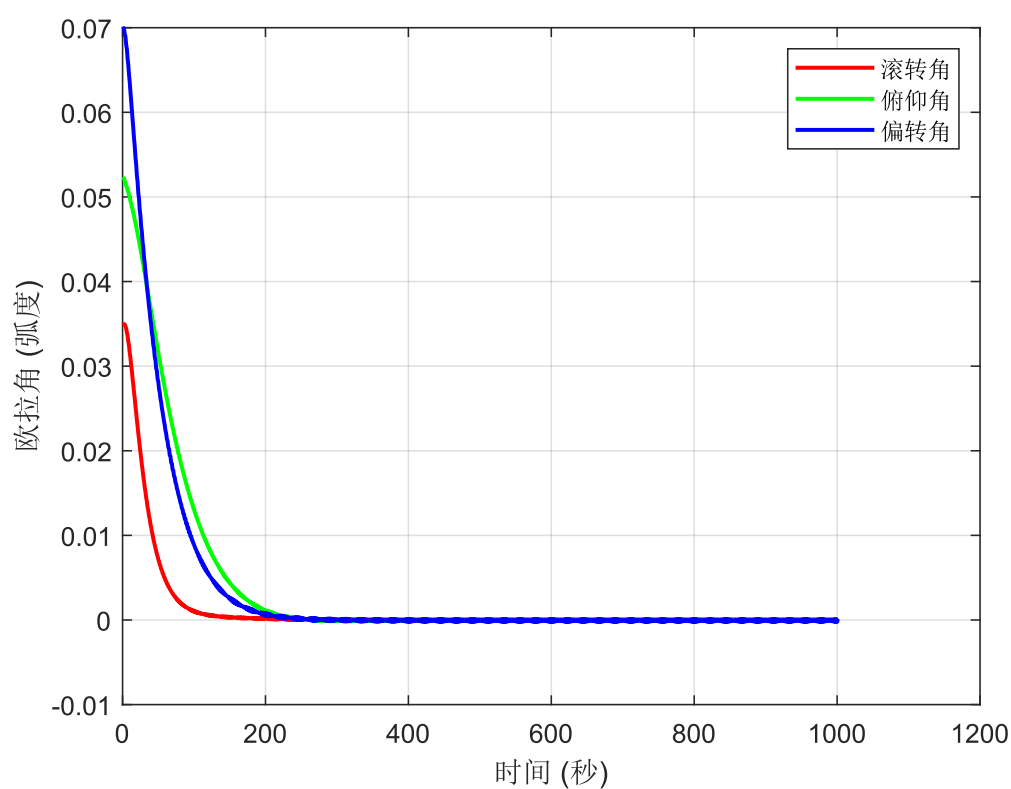
得到最终的欧拉角曲线



如图所示，在干扰力矩作用下，欧拉角在 300 秒左右达到稳定状态，稳定在 0 左右，达到预期状态。

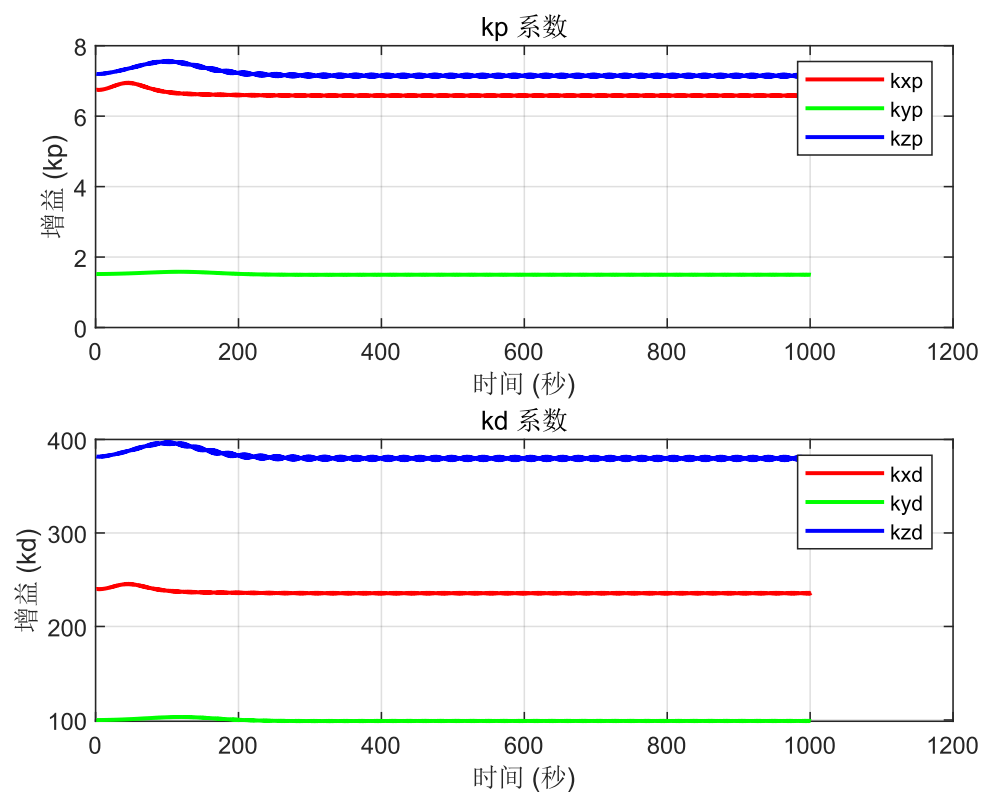
6.2、基于 BP 神经网络的 PD 控制器

在 6.1 的基础上，我们将 PD 控制器进行改进，设置 BP 神经网络输入层节点数为 3，隐含层节点数为 4，输出层节点数为 2，将 6.1 设置的控制参数进行优化，得到欧拉角的曲线为



如图所示，可以看到，通过 BP 神经网络的优化，欧拉角在更快的时间达到并稳定在预期状态状态，显著提高了控制性能。

得到控制参数优化为下图



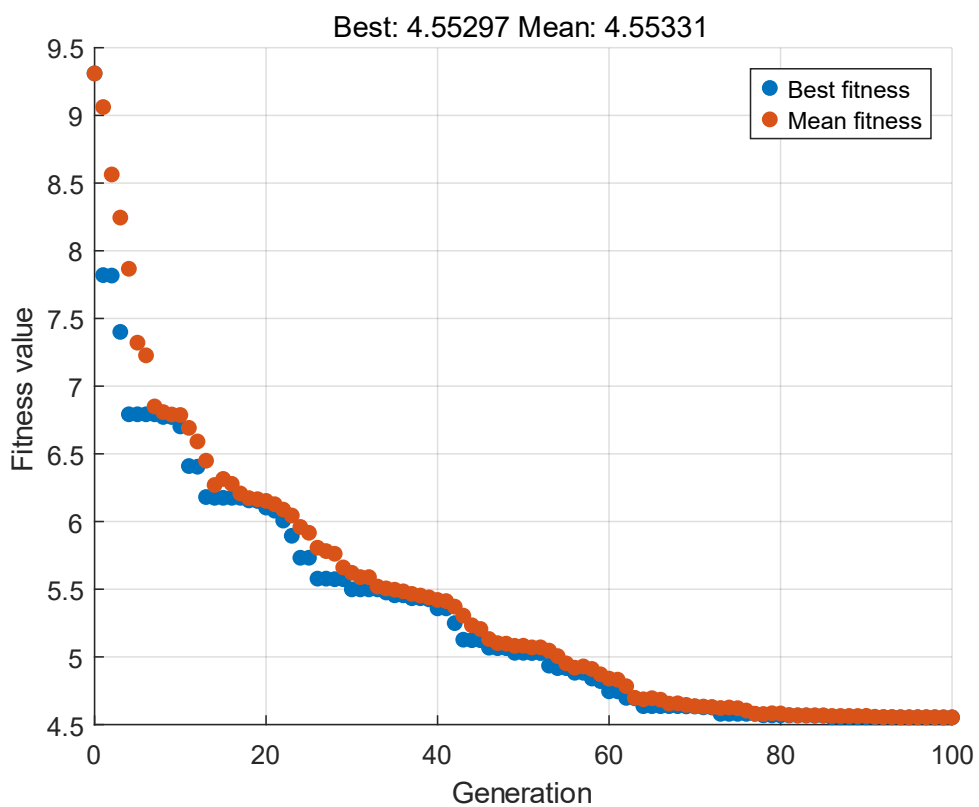
如图所示可以观察得到控制参数通过 BP 神经网络在原有基础上经过轻微调整后趋于稳定。

6.3、基于遗传算法的 PD 控制器

在 6.1 的基础上，我们进行控制参数的优化，设置种群数量为 6，迭代数为 100，收敛容差为 $1e-6$ ，适应度函数为

$$IAE = \int error_x dt + \int error_y dt + \int error_z dt$$

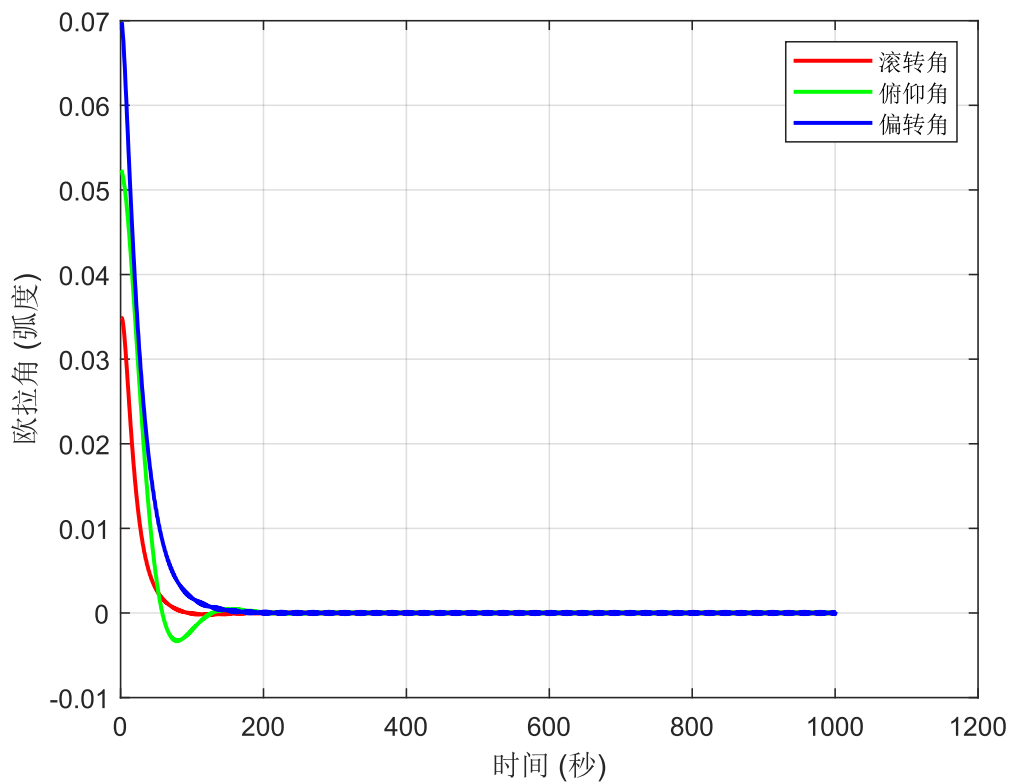
优化信息如图所示



最终得到优化的控制参数为

$$\begin{cases} k_{xp} = 18.1115 & k_{xd} = 395.4913 \\ k_{yp} = 7.3074 & k_{yd} = 172.6452 \\ k_{zp} = 21.1317 & k_{zd} = 636.1855 \end{cases}$$

将该优化得到的参数重新代入 6.1 的控制系统中得到欧拉角的响应为



如图所示，在不到 200 秒的时间欧拉角就已经达到并稳定在了预期状态，显著提高了控制效果。

7、总结

在基础控制中，我们为 PD 控制器设置了不同轴上的比例增益（KP）和微分增益（KD），并得到的欧拉角曲线显示，在 300 秒左右，系统稳定到期望的姿态。

将传统的 PD 控制器改进为基于 BP 神经网络的控制器，通过神经网络优化控制参数后，欧拉角的响应更加迅速，系统在更短的时间内达到了稳定状态，并且显著提高了控制性能。

通过遗传算法优化后的参数，系统在不到 200 秒的时间内，欧拉角迅速达到并稳定在预期状态，控制效果显著提高。

我们还可以再将遗传算法得到的控制参数代入 BP 神经网络中再进行优化，

以达到更好的控制效果。