自1974年Mandani[1]提出第一种模糊逻辑控制器（FLC）以来，人们对FLC的许多应用进行了研究，如[2]和[3]。FLC使用“IF[condition]THEN[action]形式的规则从语言上描述输入/输出关系。隶属函数将语言项转换为精确的数值。采用人工语言建模的控制方法具有计算简单、鲁棒性强、不需要寻找系统的传递函数、适用于非线性系统等优点，人性化控制得到了广泛的应用。特别是在系统传递函数非线性、时变、不确定的情况下，模糊控制相对于经典控制或现代控制具有更好的控制效果。

大多数FLC是根据专家的经验或知识设计的。然而，通常情况下没有专家。因此，通常采用试错法寻找模糊控制规则和隶属函数。为了提高效率，需要对控制规则和隶属函数进行优化设计。第一个遗传算法（GA）是由荷兰在1975年开发的[4]。许多研究扩展了遗传算法在搜索、优化和机器学习中的应用[5]、[6]。

遗传算法具有全局性和鲁棒性。搜索程序依赖于自然遗传学的机制。所有自然物种都能通过适应生存是遗传算法的潜在力量。遗传算法结合达尔文的适者生存策略，消除不合适的成分，利用随机信息交换，利用旧解中包含的知识，以惊人的力量和速度实现搜索机制。遗传算法使用多个并行的搜索点，称为“染色体”，它通过三种遗传操作（繁殖、交叉和突变）来生成新的搜索点，称为“后代”，以供下一次迭代使用。这些操作确保以适当的方式找到问题的最佳解决方案。最近，有一些研究使用遗传算法来设计隶属度函数[7]，[8]，而其他研究则使用遗传算法来设计FLC的控制规则[9]。然而，这些FLC的设计仍然需要利用专家的经验，例如，为前者设计控制规则或为后者设计隶属函数。为了更有效地设计模糊控制器，提出了一种基于遗传算法的模糊控制器隶属度函数和控制规则同时优化选择策略，使模糊控制器的设计更为简单有效。

遗传算法是一种模仿自然遗传学机制的搜索算法。它们是解决需要有效和高效搜索的问题的有用方法，在商业、科学和工程领域的应用非常广泛。在优化设计的应用中，遗传算法可以用来获得单变量或多变量优化问题的近似解。在应用遗传算法之前，应将优化问题转化为一个适当描述的函数。相应的函数称为“适应度函数”，它表示问题的性能。适应值越高，系统的性能越好。遗传算法的目标是模拟遗传操作过程，例如繁殖、交叉或变异，以获得与适应值相对应的解。最近，许多遗传算法被提出。遗传算法的基本结构可以简单描述如下

1. 定义染色体串：首先定义优化问题的搜索参数串。这些参数是染色体中的基因，可以是二进制编码或实数编码，称为“染色体”。不同的染色体代表不同的可能解决方案。
2. 定义适应度函数：适应度函数是遗传算法求解每个染色体生存能力的性能指标。适应度函数的设计是根据问题的性能要求，如收敛值、误差、上升时间等进行的。
3. 生成初始种群：在使用遗传操作之前，应随机生成染色体组。这些染色体被称为初始群体。人口规模，是根据优化问题的复杂程度来选择的。一般来说，较大的值需要较少的世代才能得到收敛解。但是，总的计算效果取决于生成次数的倍数。
4. 生成下一代或停止：遗传算法利用繁殖、交叉、变异等操作生成下一代。代代相传，每一代的适应值都达到最大值。

a）再生产：再生产是根据适应值将旧的字符串带入新的种群的操作。具有高适应度值的字符串在下一代中获得更多副本。这种操作的示例如表一。

b） 交叉：交叉是一种结合了繁殖的重组算子。这是一种有效的信息交换和重组高适应度个体片段的方法。交叉过程是从配对池中随机选择一对字符串，然后随机确定交叉位置。操作示例如表二所示。

c） 变异：变异算子用于避免将局部最优值误认为全局最优值的可能性。这是基于变异概率的字符串位置的偶然随机变化。操作示例如表三所示。

本文简要介绍了用遗传算法设计模糊控制器的方法。在一个模糊控制器的设计，重点是模糊化过程中隶属函数的设计和模糊控制规则中的后续变量的设计。利用语言值NB、ZO和PB三个模糊变量进行模糊逻辑控制器的优化设计。如果FLC的性能这样设计出来的模糊变量不能满足用户的需求，模糊变量的数量会自动增加一个，直到满足需求为止。所使用的隶属函数是三角形的，如图1所示。三角形的隶属函数可以由基部的两个顶点ap和an参数化。使用的解模糊算法是简单的最大对应法，如图2所示。遗传算法的染色体包括模糊控制表上的后续变量和隶属函数的参数两部分。为了减少染色体上的基因数目，使用了离散的实数编码基因。下面是染色体上基因搭配的一个例子

染色体上的第一到第九个基因是控制规则表的元素，如表4所示。模糊控制规则表上的数字1、2和3分别表示语言值PB、ZO和NB。第十到第十五个基因是成员ap、an、bp、bn、cp和cn的参数，如图3所示。从0到1和从0到-1划分了16个区段，分别是从这些离散点中搜索每个最优隶属函数的参数。在左顶点（ap、bp和cp），PL被指定为语言值，在右顶点（an、bn和cn），ZR被指定。

本文提出的适应度函数由系统阶跃响应的三个性能指标组成：最大超调量、上升时间和累积误差。最大过冲是最大过冲值与最终值的百分比。上升时间是阶跃响应从0到其最终值的时间。累积误差是采样瞬间阶跃响应绝对误差的总和。将每个性能指标转换为从0到100的刻度。平均值用作适应值。可根据系统要求对各指标赋予不同权重。例如，对于希望缩短上升时间的系统，上升时间的权重可以更改为1.3，其他权重可以更改为0.9和0.8。人口按适应度函数排序。第一条染色体具有最高的适应度函数。如果FLC的性能满足设计要求，则GA的运行将停止；否则，它将继续生成下一代或增加模糊变量的数目。

遗传算法的进化过程如图4所示。初始群体的N条染色体在初始片段中随机产生。然后，对每个染色体的适应度函数进行评价。第一条染色体具有最高的适应值，即该染色体具有本代系统的最佳阶跃响应。如果达不到这一要求，当前一代的染色体将经历三次遗传操作，即繁殖、杂交和变异，以产生下一代。GA操作将重复该程序，直到达到要求。如果适应值在一定的世代数（如世代数）内保持不变，遗传算法将自动增加一个模糊变量，并重新初始化过程。

为了测试所提出的方法，我们使用模糊比例积分微分（PID）控制器结构[10]，如图5所示。PID控制是主控制器，模糊控制是从控制器，以增强主控制器的性能一个。那个模糊控制规则的先行变量是系统阶跃响应的误差（）和误差率。和的定义如下：

其中是系统的参考输出，是采样周期。结果变量是FLC系统中的错误变量。FLC利用这种变化来调整系统的误差，从而改善系统的阶跃响应性能。GA调谐模糊PID控制的结构如图6所示。

模拟装置的传递函数为

PID控制器参数最初是根据Ziegler–Nichols规则[11]选择的。Kp、Ki和Kd的结果值分别为9.257、8.6和1.45。利用MATLAB对模拟PID控制对象系统进行离散化。人口数定为20人。最大超调、累积误差和上升时间的权重分别为1.2、1.2和0.6。停车条件为：最大超调量小于1.5%，累积误差小于12，上升时间小于0.25s，即适应值必须大于87.27。重新初始化的生成数设置为50。具有最高适应值的染色体被复制，第二个到第十一个被交叉，第十二个到第二十个被突变。采用两种方法加快搜索速度：1）改变变异率；2）建立适应值数据库。当染色体的最高适应值在三代内保持不变时，遗传操作将提高变异率，从而提高搜索速度。数据库存储每个已经生成的染色体的放弃值。在计算适应度函数之前，系统将搜索同一染色体的适应度值。图7示出了每一代的最高适应值。图8显示了第一、第五、第十、第二十和第二十四代适应度最高的染色体的阶跃响应。结果表明，代际适应值较高。GA调谐模糊PID控制系统和PID控制系统之间的阶跃响应的比较如图9所示。FLC的最佳染色体出现在第24代。最优FLC的控制规则表和隶属度函数如图10所示。请注意，在FLC系统中实际只使用第三、第五和第六个控制规则。仿真结果表明，遗传算法是获得最优模糊控制的有效方法。

本文利用遗传算法设计了一种最优模糊控制器。仿真结果表明，该方法是有效的。与传统的试错设计程序相比，该技术可以节省时间。通过系统搜索得到的最优模糊控制器只需要几个模糊变量。工厂的模型不需要额外的专业知识或数学分析。

在未来的研究中，将遗传算法应用于不同的应用领域是值得的。