在本文的动力学分析中，机器人下肢的关节力矩由PD轨迹跟踪的控制方法实现: 即以机器人期望轨迹与实际轨迹的差值为PD控制的比例项，使用差分近似导数的方法构成控制中的微分项。具体地，基于PD控制算法的机器人关节力矩可以表示为

$$\tau_j = K_{j,\mathrm{P}} e_j(t) + K_{j,\mathrm{D}} \dot{e}_j(t) \quad (j = 1,\ldots,7)$$

其中，τ_j 为关节 j 的力矩，$K_{j,\mathrm{P}}$、$K_{j,\mathrm{D}}$ 为比例和微分增益，$e_j(t)$ 和 $\dot{e}_j(t)$ 分别表示了当前时刻下关节 j 的位置误差和速度误差。

为体现机器人对环境的自适应性，PD控制的参数 $K_{j,\mathrm{P}}$ 和 $K_{j,\mathrm{D}}$ 不采用固定值，而是通过具有出色搜索能力的启发式算法自适应地调节。具体地，本研究选取粒子群优化(PSO)算法来确定PD参数以达到自适应生成机器人关节力矩的目的。这是因为其能够在没有得知太多问题信息的情况下，有效地搜索庞大的解空间。

粒子群优化（Particle Swarm Optimization, PSO）是一种基于群体智能的搜索算法，通过模拟粒子在多维空间内的运动，来寻找全局最优解。算法的基本思想是，假设有一群粒子在多维空间内飞行，每个粒子代表一种可能的解，它们通过向着全局最优解和个体最优解的方向运动来改善自己的位置。在每一步迭代中，粒子根据全局最优解和个体最优解的位置来更新自己的速度和位置，并不断尝试寻找更优解。粒子群优化算法具有许多优秀的特性，如全局搜索能力强、对初始解不敏感、适用于高维空间等。然而，由于粒子群算法存在种群收敛性差的问题，因此需要通过一些技巧来提高其全局搜索能力和收敛性。

基于粒子群优化算法和PD轨迹跟踪控制生成机器人关节力矩的流程如下图所示，粒子群优化算法的参数列于下表。先说一下7维分别是哪7维。其中，D 为粒子维度，即待优化PD控制参数的维度，前7维为 K_{P} 参数，后7维为 K_{D} 参数；m 为种群规模，n 为总迭代次数，根据仿真所需时间给定；ω 为惯性权重，体现了粒子保持前一运动状态的能力；k 为当前迭代次数，c_1 和 c_2 为个体学习因子和群体学习因子，用于调节学习最大步长。

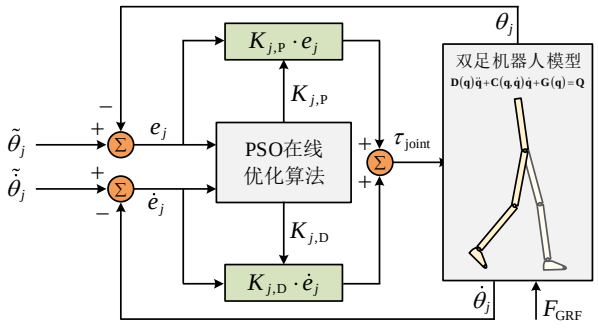

图 基于粒子群优化的机器人关节力矩PD控制流程

表 粒子群优化算法参数值

Parameters	Values	Parameters	Values
D	14	ω	1
m	100	c_1	2
n	50	c_2	2

本研究中，动力学数值计算通过欧拉法实现，其计算步长被设为0.1 ms，每个粒子的作用时间为2 ms。PSO算法中速度更新和位置更新公式如下:

$$\boldsymbol{V}_i^{k+1} = \omega \boldsymbol{V}_i^k + c_1 r_1 \left(\boldsymbol{P}_i - \boldsymbol{X}_i^k \right) + c_2 r_2 \left(\boldsymbol{P}_g - \boldsymbol{X}_i^k \right) \tag{29}$$

$$\boldsymbol{X}_i^{k+1} = \boldsymbol{X}_i^k + \boldsymbol{V}_i^k \tag{30}$$

其中，$\boldsymbol{V}_i^k$ 为第 k 次迭代中粒子 i 的速度矢量，$\boldsymbol{X}_i^k$ 为第 k 次迭代中粒子 i 的位置矢量，r_1 和 r_2 为区间[0,1] 内的随机数。$\boldsymbol{P}_i$ 为粒子 i 的当前个体最优位置矢量，$\boldsymbol{P}_{\mathrm{g}}$ 为整个粒子群的当前全局最优位置矢量。r_1，r_2 的引入增加了种群搜索的随机性，提升了粒子群算法避免陷入局部最优的能力。

粒子群优化的损失函数定义为:

$$f(t_s) = \frac{1}{t_s} \sum_{t=0}^{t_s} \left(\alpha \sqrt{\sum_{j=1}^{7} e_j^{\,2}(t)} + \beta \sqrt{\sum_{j=1}^{7} \dot{e}_j^{\,2}(t)} \right) \tag{31}$$

其中，α 和 β 为位置误差和速度误差的权重，损失函数表征了在 t_s 时刻系统位置和速度的平均累计误差。

基于图 23 的具体流程如下: 根据每个粒子当前的位置生成机器人各个关节的力矩并作用于动力学模型(25)，每个粒子的作用时间为 2 ms，计算当前的损失函数 $f(t_i)$ 并与之前的损失函数进行比较，确定并更新该粒子的个体最优位置 $\boldsymbol{P}_i$ 和粒子群的全局最优位置 $\boldsymbol{P}_{\mathrm{g}}$。当 $m=100$ 个粒子依次执行完 2 ms(共计 0.2 s)后，基于当前的个体最优位置 $\boldsymbol{P}_i$ 和当前全局最优位置 $\boldsymbol{P}_{\mathrm{g}}$，利用式(29)和(30)更新每个粒子的位置和速度，并开始下一轮迭代。上述算法具有较好的收敛性和在线性，随着迭代次数的增加，损失函数将很快收敛到最小值。