

# Kylinsky 3D

基于进化算法的步态稳定优化

# 目录

## CONTENTS

**01.** 进化算法的基本思想  
The basic idea of evolutionary algorithms

**02.** 基于倒立摆模型的建模  
Modeling based on inverted pendulum model

**03.** CMA-ES  
Introduction to the CMA - ES

# PART 01

---

## 进化算法的基本思想

The basic idea of evolutionary algorithms

# 进化算法的基本思想

The basic idea of evolutionary algorithms

进化算法，也被称为演化算法（evolutionary algorithms，简称EAs），它不是一个具体的算法，而是一个“算法簇”。与传统的基于微积分的方法和穷举方法等优化算法相比，进化计算是一种成熟的具有高鲁棒性和广泛适用性的全局优化方法，具有自组织、自适应、自学习的特性，能够不受问题性质的限制，有效地处理传统优化算法难以解决的复杂问题。

进化算法的关键在于进化。从达尔文进化论的角度来讲，进化是在一个系统中，个体在每一代的更替中通过变异和选择来不断的产生下一代群体。所以，这便是进化算法所模拟的进化系统。

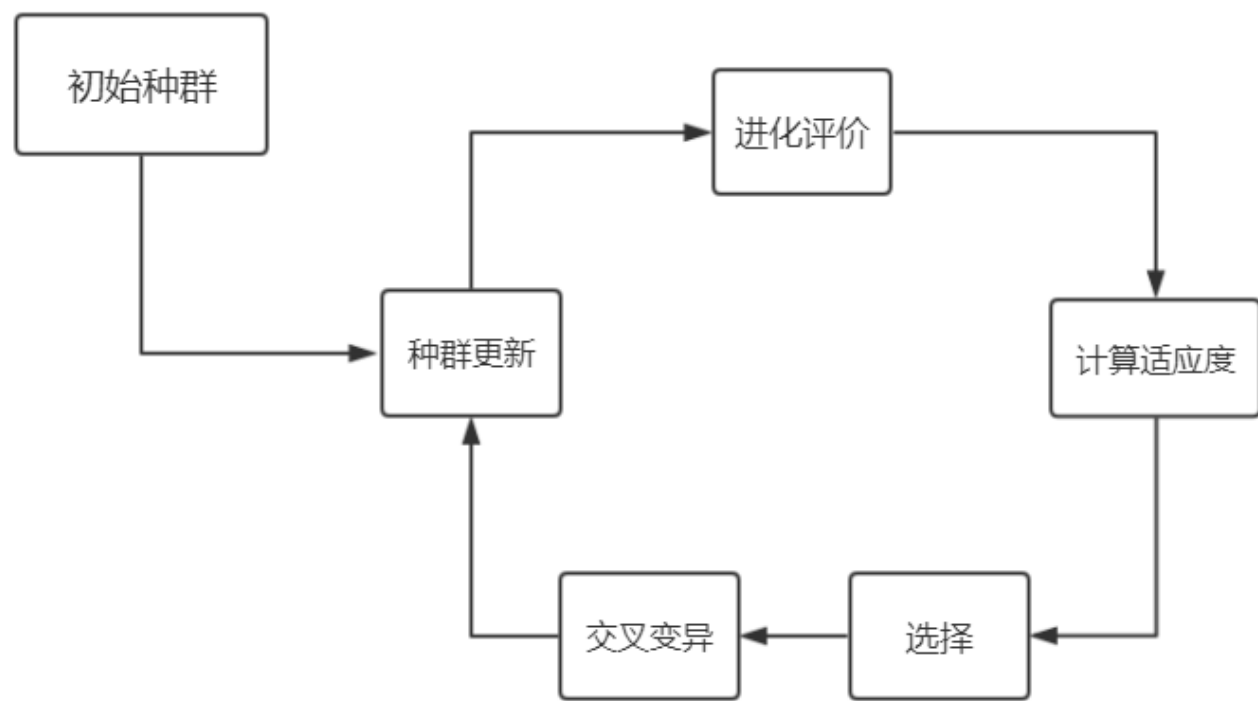
# 进化算法的基本思想

The basic idea of evolutionary algorithms

在进化算法中，从一组随机生成的初始个体出发，主要采用复制、交换、突变这三种一串操作，衍生出下一代的个体。再根据适应度的大小进行个体的优胜劣汰，提高新一代群体的质量，在经过反复多次迭代，逐步逼近最优解。从数学角度讲，进化算法实质上是一种搜索寻优的方法。仿效生物的遗传方式，其应包含以下四点：

- (1). 一个或多个种群中的所有个体竞争有限资源
- (2). 种群受到个体的出生或死亡率等因素的影响而不断地改变
- (3). 个体生存繁衍的能力由其自身的环境适应能力而定
- (4). 新增个体的特征从父代继承，但是由于交叉或变异还会产生新的特征

因而，进化是一个不断迭代的过程，在每一次迭代中，通过交叉、变异和选择等基本算子产生新的个体。



图一 进化算法基本执行流程

# PART 02

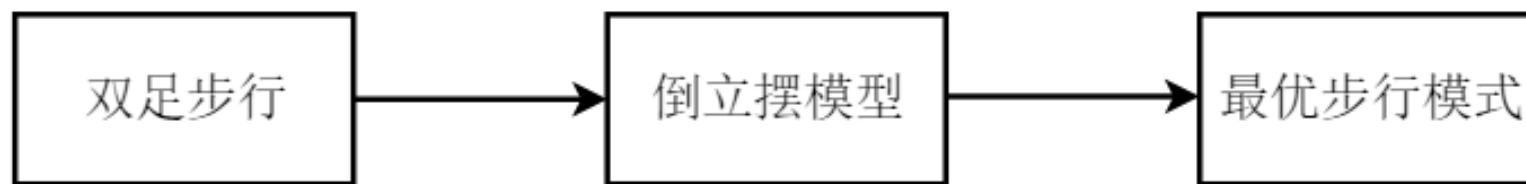
---

## 基于倒立摆模型的建模

Modeling based on inverted pendulum model

# 基于倒立摆模型的建模

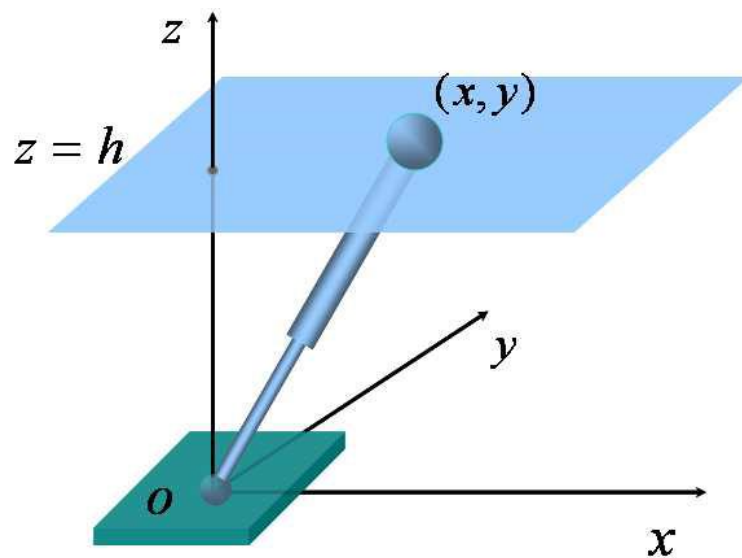
Modeling based on inverted pendulum model



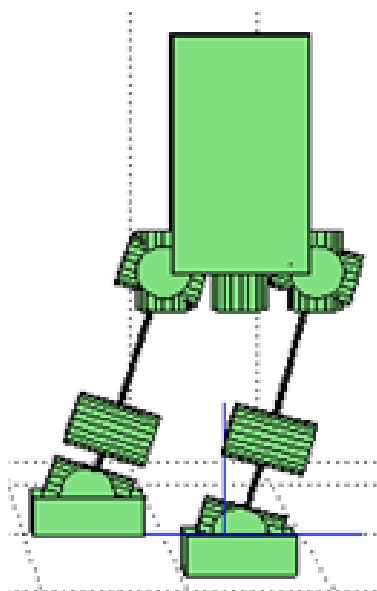
图二 足步行问题的模型优化



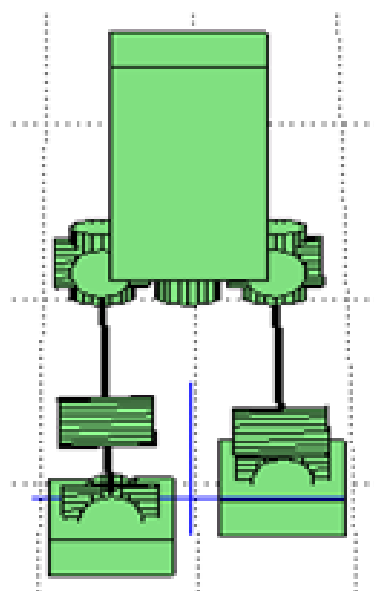
- 假定机器人的所有质量集中于其质心位置
- 假设机器人的腿无质量，它与地面的接触是通过一个可以转动的支点实现



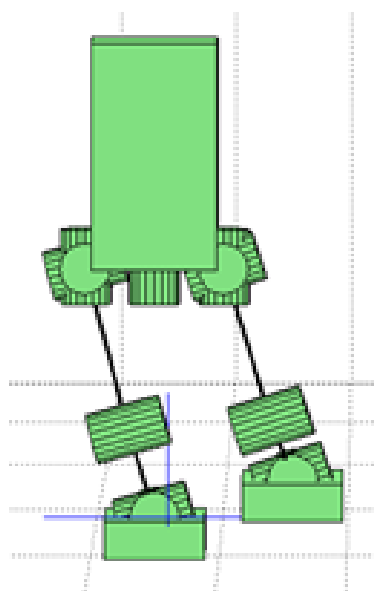
## 倒立摆模型的三个阶段：



(a) 单足支撑落脚阶段



(b) 双足支撑阶段



(c) 单足支撑抬脚阶段

# PART 03

---

## CMA-ES

Introduction to the CMA - ES

# CMA-ES

Introduction to the CMA - ES



协方差矩阵自适应进化策略，英文名称为Covariance Matrix Adaptation Evolutionary Strategies，简称CMA-ES。

其主要用于解决连续优化问题，尤其在病态条件下的连续优化问题。进化策略算法主要作为求解参数优化问题的方法，模仿生物进化原理，假设不论基因发生何种变化，产生的结果（性状）总遵循这零均值，某一方差的高斯分布。

在CMA-ES算法中，新个体在多维正态分布中采样产生。假设当前是第g代，则第g+1代中的每个个体服从一下分布：

$$x_k^{(g+1)} \sim m^{(g)} + \sigma^{(g)} \mathcal{N}(0, C^{(g)}), k=1, \dots, \lambda$$

其中  $m^{(g)}$  是第g代的样本均值， $\sigma^{(g)}$  是第g代的样本标准差。 $\mathcal{N}(0, C^{(g)})$  是均值为0的正态分布，其协方差矩阵  $C^{(g)}$  是第g代的样本协方差矩阵。 $\lambda$  是种群中的个体数目。



遗传算法(Genetic Algorithm,GA)

差分进化算法(Differential Evolution, DE)

粒子群优化算法 ( Particle Swarm Optimization , PSO )

我们选择了：

**CMA-ES**



```
#define CMA_ES
```

```
... ..
```

```
extern ParseMessage messageParser;
```

```
extern WorldModel wm;
```

```
... ..
```

```
Action SoccerbotBehavior::GetMyBeamCoordinateForKick()
```

```
{
```

```
    stringstream ss("");
```

```
    Vector3 beamPos;
```

```
    switch( wm.GetMyNumber() )
```

```
{
```

```
... ..
```

```
}
```

```
    wm.SetMyBeamCoordinate(beamPos);
```

```
    ss<<"(beam "<<beamPos.x()<<" "<<beamPos.y()<<" "<<beamPos.z()<<)"
```

```
    return ss.str();
```

```
}
```

# 我们为什么会选择使用这个算法？

非线性非凸包性问题

很多的恒定性质

应用领域



# 谢谢观赏

Thank You

## 不足之处请批评指正

Please point out the shortcomings