# 弹道模型

## 一、前言

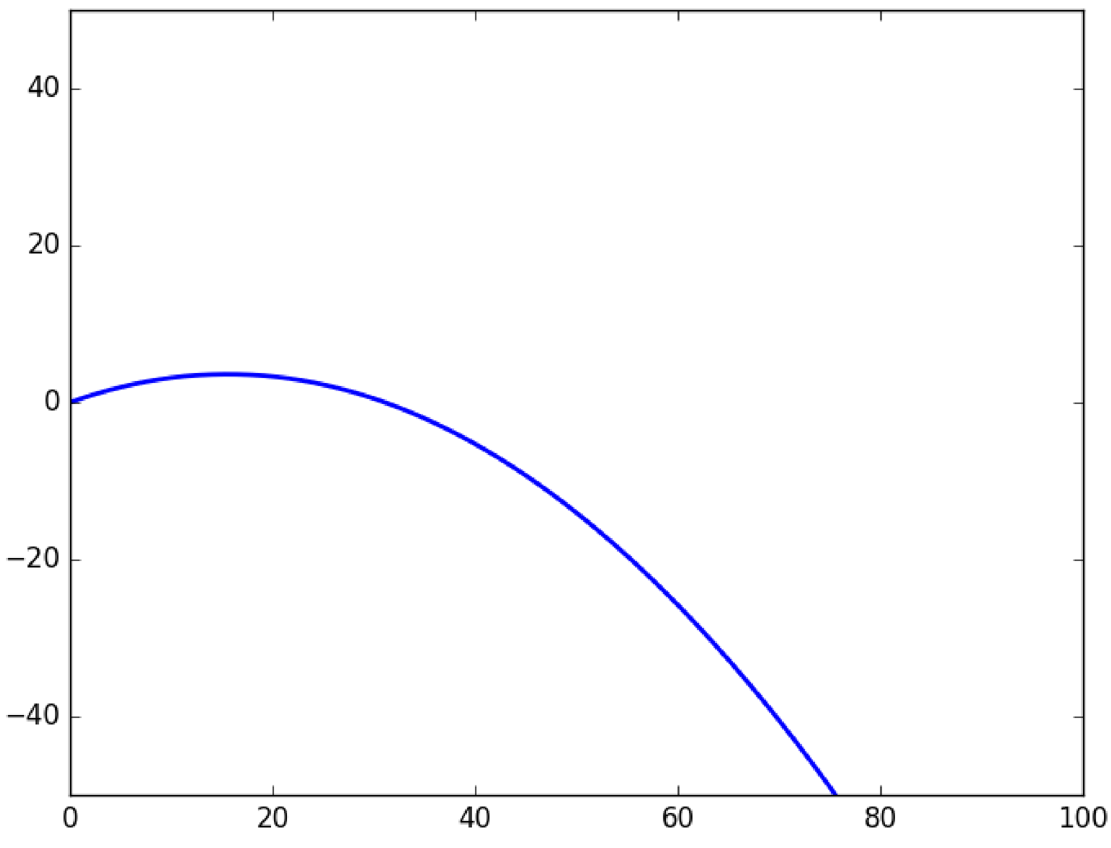
在RoboMaster比赛中，子弹飞行模型对于命中率具有一定影响，对于远距离打击，如果不进行弹道修正，由于存在重力下落，子弹会打到目标偏下的位置。

在目前的弹道研究，多针对于实际的炮弹，或枪弹模型，考虑因素非常多，模型相对精确，也相对复杂，基本上射击微分方程的数值求解，对于实时性影响较大。

在本文中，先提出一个简单的理想抛物线模型，然后加入空气阻力的考虑，并对模型进行简化，建立一个单方向空气阻力模型。

## 二、理想抛物线模型

- 只考虑重力对弹道的影响，理想弹道如下，是一条抛物线。



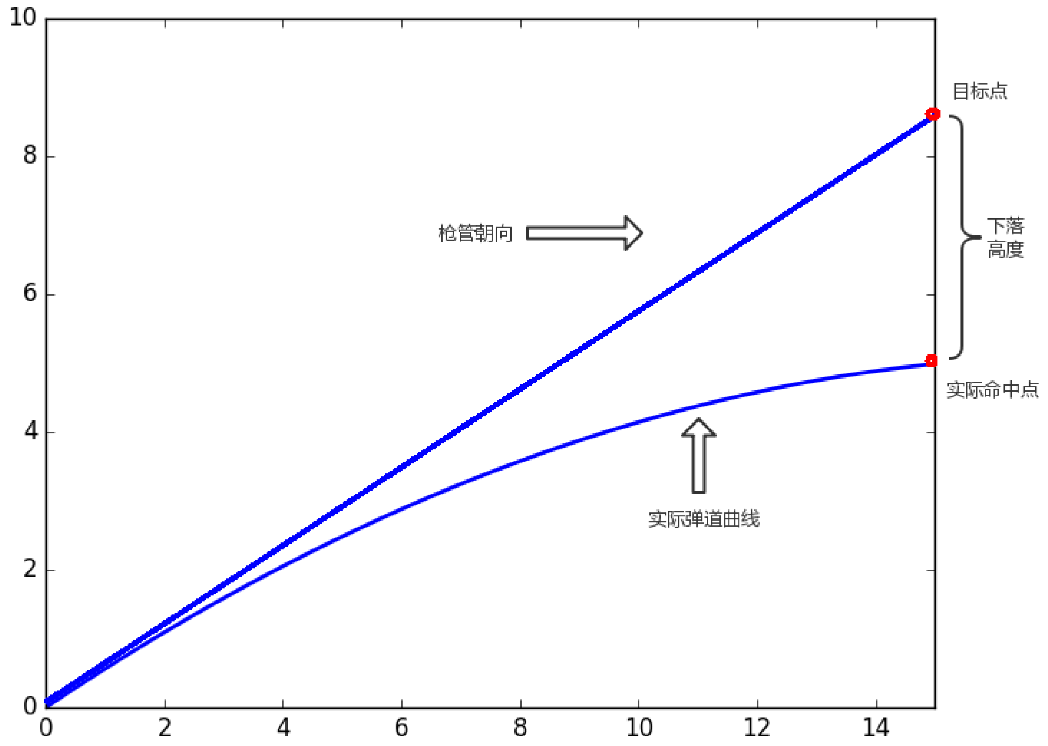

**现在给定目标位置坐标(x,y)，求解出射角（炮台仰角）。**

- 抛物线模型是非常简单的，但是求解出射角度是子弹运动的逆运算，需要反推模型，是可以推算出的，其优点是求解精确，但公式复杂，且存在多解，无解的情况，编程也不太方便。

基于以上考虑，这里给出了一个**基于模型前向迭代的数值解法**：

首先引入几个概念：

-




显然，当我们直接瞄准目标点（枪管朝向目标点）时，会有一个下落高度。

- 我们将利用这个**下落高度**进行迭代**补偿。**

---

算法过程如下：

- 设置最终目标点targetPoint
- 设临时目标点tempPoint=targetPoint
- 循环迭代10次：
  - 计算仰角angle=枪管指向tempPoint的角度
  - 利用抛物线模型，计算实际命中点realPoint.
  - 得到误差，即下落高度deltaH=targetPoint-realPoint
  - 更新tempPoint=tempPoint+deltaH
- 输出仰角angle，与误差deltaH

---

根据算法过程，编写代码进行测试，结果如下：

```
枪口速度25.000000,目标坐标（10.000000,1.000000)
----------------------------
第1次迭代: 仰角: 5.710593,临时目标点y值:1.791840,高度误差:0.791840
第2次迭代: 仰角: 10.158681,临时目标点y值:1.809172,高度误差:0.017332
第3次迭代: 仰角: 10.254868,临时目标点y值:1.809661,高度误差:0.000489
第4次迭代: 仰角: 10.257582,临时目标点y值:1.809675,高度误差:0.000014
第5次迭代: 仰角: 10.257658,临时目标点y值:1.809675,高度误差:0.000000
第6次迭代: 仰角: 10.257661,临时目标点y值:1.809675,高度误差:0.000000
第7次迭代: 仰角: 10.257661,临时目标点y值:1.809675,高度误差:0.000000
第8次迭代: 仰角: 10.257661,临时目标点y值:1.809675,高度误差:0.000000
第9次迭代: 仰角: 10.257661,临时目标点y值:1.809675,高度误差:0.000000
第10次迭代: 仰角: 10.257661,临时目标点y值:1.809675,高度误差:0.000000
----------------------------
最终结果:10.257661,最终误差:0.000000
```

测试结果:

- 最终误差小于1mm（目标在30m以内），满足精度需求.
- 在6代i7cpu上测试，时间小于0.1ms，满足实时性需求
- 目标距离越近，收敛速度越快，5m目标，3次迭代，就能达到误差小于1mm。

以上均为理想模型，未考虑空气阻力，并不能在实际中应用。（未实际测试）

# 二、单方向空气阻力模型

由于抛物线模型过于简单，所以实际应用场景并不多，但是抛物线模型是其他所有弹道模型的基础。这里我们考虑空气阻力，进一步修正模型，使得模型可以实际应用。

目前，已经存在许多先进的弹道模型，但是模型过于复杂，不易理解，而且涉及微分方程数值求解，比较耗时，所以，引入空气阻力模型，但本着简化模型的原则，对其进行简化。

- 由于在比赛中，子弹飞行不会过于斜抛（吊射基地除外），所以子弹飞行过程中，受到的阻力更多来源与水平方向（x方向）。
- 基于以上分析，**对子弹飞行做运动分解，只考虑水平方向的空气阻力**，不考虑垂直方向的空气阻力。

模型推导:

$$
\begin{aligned}
\text{空气阻力模型:}\ & f = CpSv_x^2/2 \\
\text{简化公式得:}\ & f_x = k_0 v_x^2 \ldots\ldots (k_0 = CpS/2) \\
\text{牛顿定律:}\ & -f_x/m = a = \frac{dv_x}{dt} \\
& \Longrightarrow -\frac{k_0}{m}v_x^2 = \frac{dv_x}{dt} \Rightarrow k_1 dt = -\frac{dv_x}{v_x^2} \ldots\ldots (k_1 = \frac{k_0}{m}) \\
& \Longrightarrow k_1 t + C = \frac{1}{v_x} \text{且} (v_x(t=0) = v_{x0}) \\
\text{即得到水平方向速度模型}\ & v_x = \frac{v_{x0}}{k_1 v_{x0} t + 1} \\
\text{即得到水平方向位移模型}\ & x = \int_0^t v_x dt = \frac{1}{k_1}\ln(k_1 v_{x0} t + 1)
\end{aligned}
$$

其中C为球体在空气中的摩擦系数，RoboMaster 2019 ICRA人工智能挑战赛所使用的弹丸取值为0.47, p为空气密度，在温度为**0摄氏度**、**标准大气压**下取值为1.293kg/m^3, **25摄氏度**、**标准大气压**取值为1.169kg/m^3, S接触面积。

不难发现，当**系数k1趋近于0**时，该模型**退化**为**理想抛物线模型**

$$\lim_{k_1 \to 0} x = \lim_{k_1 \to 0} \frac{\ln(k_1 v_{x0} t + 1)}{k_1} = \lim_{k_1 \to 0} \frac{k_1 v_{x0} t}{k_1} = \lim_{k_1 \to 0} v_{x0} t = v_{x0} t$$

得到**运动模型**后，利用**基于模型前向迭代的数值解法**，对模型进行迭代求解，步骤同上。

由于空气密度受空气湿度、大气压强、温度影响，所以系数k1在不同环境下取值有所区别，以下是系数k1=0.1时的迭代结果

```
枪口速度25.000000,目标坐标（10.000000,1.000000)
----------------------------
第1次迭代: 仰角: 5.710593,临时目标点y值:2.619620,高度误差:1.619620
第2次迭代: 仰角: 14.679461,临时目标点y值:1.591977,高度误差:-1.027643
第3次迭代: 仰角: 9.045448,临时目标点y值:2.229931,高度误差:0.637954
第4次迭代: 仰角: 12.570891,临时目标点y值:1.828138,高度误差:-0.401793
第5次迭代: 仰角: 10.360058,临时目标点y值:2.078997,高度误差:0.250858
第6次迭代: 仰角: 11.744473,临时目标点y值:1.921498,高度误差:-0.157499
第7次迭代: 仰角: 10.876800,临时目标点y值:2.020041,高度误差:0.098544
第8次迭代: 仰角: 11.420302,临时目标点y值:1.958250,高度误差:-0.061791
第9次迭代: 仰角: 11.079740,临时目标点y值:1.996943,高度误差:0.038693
第10次迭代: 仰角: 11.293093,临时目标点y值:1.972693,高度误差:-0.024250
第11次迭代: 仰角: 11.159414,临时目标点y值:1.987884,高度误差:0.015191
第12次迭代: 仰角: 11.243165,临时目标点y值:1.978365,高度误差:-0.009518
第13次迭代: 仰角: 11.190692,临时目标点y值:1.984328,高度误差:0.005963
第14次迭代: 仰角: 11.223567,临时目标点y值:1.980592,高度误差:-0.003736
第15次迭代: 仰角: 11.202970,临时目标点y值:1.982933,高度误差:0.002341
第16次迭代: 仰角: 11.215874,临时目标点y值:1.981466,高度误差:-0.001466
第17次迭代: 仰角: 11.207790,临时目标点y值:1.982385,高度误差:0.000918
第18次迭代: 仰角: 11.212854,临时目标点y值:1.981809,高度误差:-0.000576
第19次迭代: 仰角: 11.209682,临时目标点y值:1.982170,高度误差:0.000361
第20次迭代: 仰角: 11.211669,临时目标点y值:1.981944,高度误差:-0.000226
----------------------------
最终结果:11.211669,最终误差:-0.000226
```

- 相对无空气阻力模型，有一个1度的角度修正。
- 该模型相对复杂，在收敛过程中会引入振荡，需要更多的迭代次数，保证收敛精度，即误差小于1mm（这里选择20次迭代）。

**本文注意:**

- 该弹道模型并未经过实际测试，效果未知。
- 以上误差分析，都是数值计算与模型之间的误差，并非实际测试误差，可评估数值解法的精度，不能评估模型的好坏。