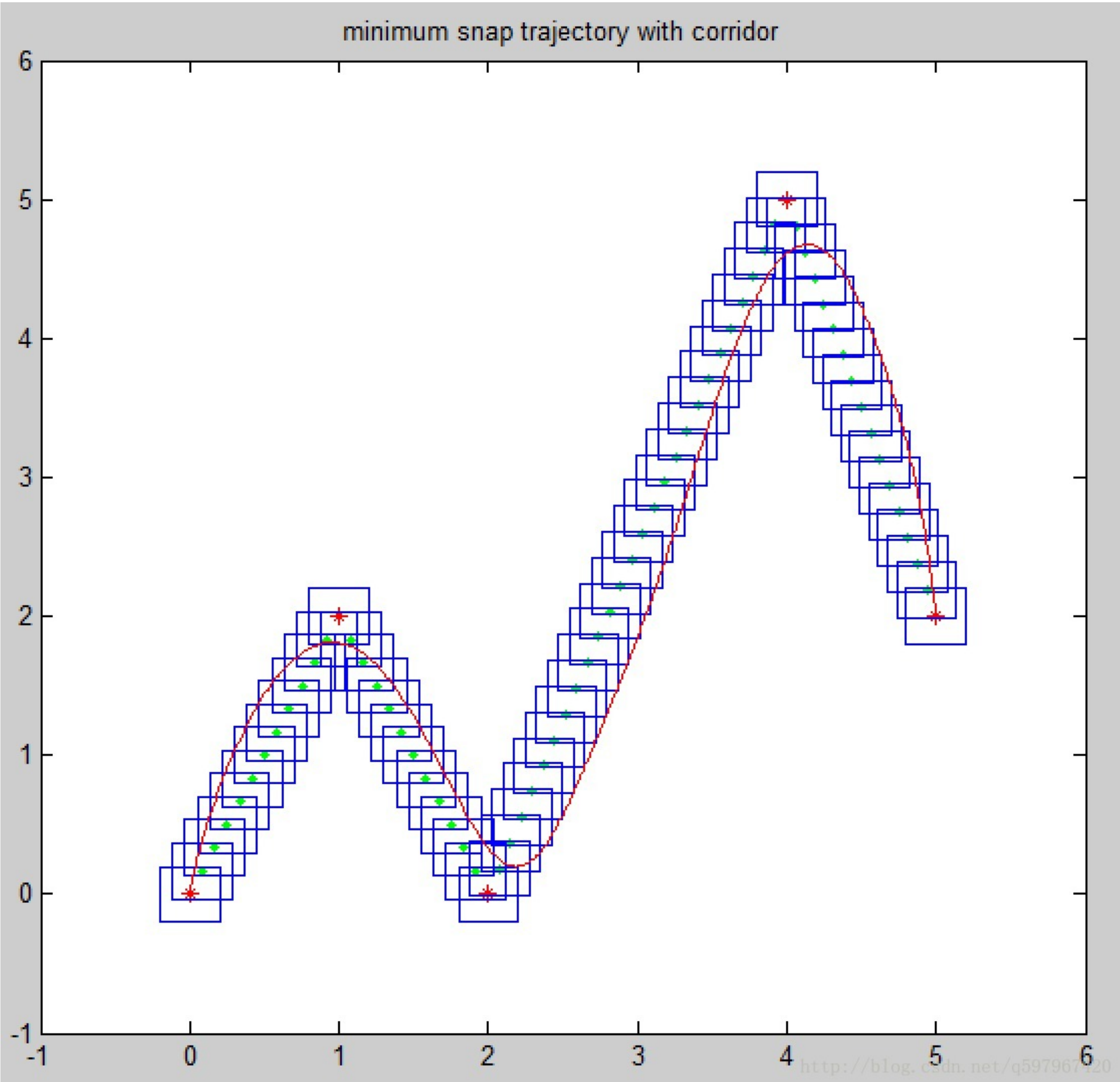
Minimum Snap轨迹规划详解（4）guiding trajectory

轨迹规划

上一节《Minimum Snap轨迹规划详解（3）闭式求解》讲的在只有等式约束时的轨迹闭式求解，现在又回到一般的情况，回顾《Minimum Snap轨迹规划详解（2）corridor与时间分配》中轨迹规划的结果，



虽然通过添加corridor使得轨迹在路径点附近，但我们发现优化出轨迹总倾向于在corridor的边缘，总是不在路径上，我们希望轨迹能够尽可能贴近路径，比如：在路径点之间尽可能走直线，在拐弯处又尽可能平滑。（要求真TM多~[嫌弃脸]）

1. Guiding trajectory原理

通过最小化Jerk/Snap已经保证了轨迹的平滑性，所以我们现在要做的就是使得轨迹尽可能贴近路径。因此，在优化项中，除了优化Jerk/Snap，再添加一个轨迹与期望轨迹的误差项，即：

$$\min \underbrace{\int_0^T \|p^{(3)}(t)\|^2 dt}_{jerk} + \lambda \underbrace{\int_0^T \|p(t) - p_s(t)\|^2 dt}_{guiding\ error}$$

其中，$p_s(t)$为期望的轨迹，称之为guiding trajectory，λ为guiding项的权重，调节平滑性(jerk项)和目标曲线跟踪性(guiding误差项)的比重，当$\lambda = 0$则变成了minimum jerk，λ越大，则轨迹越接近guiding trajectory。

2. QP Formulation

引入guiding项以后，目标函数仍然能够formulate成QP的形式，即$\min p^T Qp + 2b^T p$。
设guiding trajectory为：

$$p_s(t) = [1, t, t^2, \ldots, t^n]c$$

其中c为 **给定的** guiding trajectory的轨迹参数（guiding trajectory必须和待求的轨迹一样为n阶多项式），进而对每一段polynomial，guiding error项可以写为：

$$\begin{aligned}
&\int_{t_{i-1}}^{t_i} \|[1, t, \ldots, t^n]p - [1, t, \ldots, t^n]c\|^2 dt \\
=&\int_{t_{i-1}}^{t_i} (p - c)^T [1, t, \ldots, t^n]^T [1, t, \ldots, t^n](p - c)dt \\
=&(p - c)^T \underbrace{\int_{t_{i-1}}^{t_i} [1, t, \ldots, t^n]^T [1, t, \ldots, t^n]dt}_{H_i}(p - c) \\
=&p^T H_i p - 2c^T H_i p + c^T H_i c
\end{aligned} \tag{1}$$

令$b_i = -H_i^T c$，去掉常数项$c^T H_i c$，从而化成$p^T H_i p + b^T p$形式。
进而，目标函数可以写为：

$$\begin{aligned}
&\min \int_{t_{i-1}}^{t_i} \|p^{(3)}(t)\|^2 dt + \lambda \int_{t_{i-1}}^{t_i} \|p(t) - p_s(t)\|^2 dt \\
=&\min p^T Q_i p + \lambda p^T H_i p + \lambda b_i^T p \\
=&\min p^T \underbrace{(Q_i + \lambda H_i)}_{Q_i^{'}} p + \underbrace{\lambda b_i^T}_{b_i^{'}} p
\end{aligned} \tag{2}$$

得到目标函数以后，再按之前的方法构造等式约束和不等式约束，进而求解出轨迹。

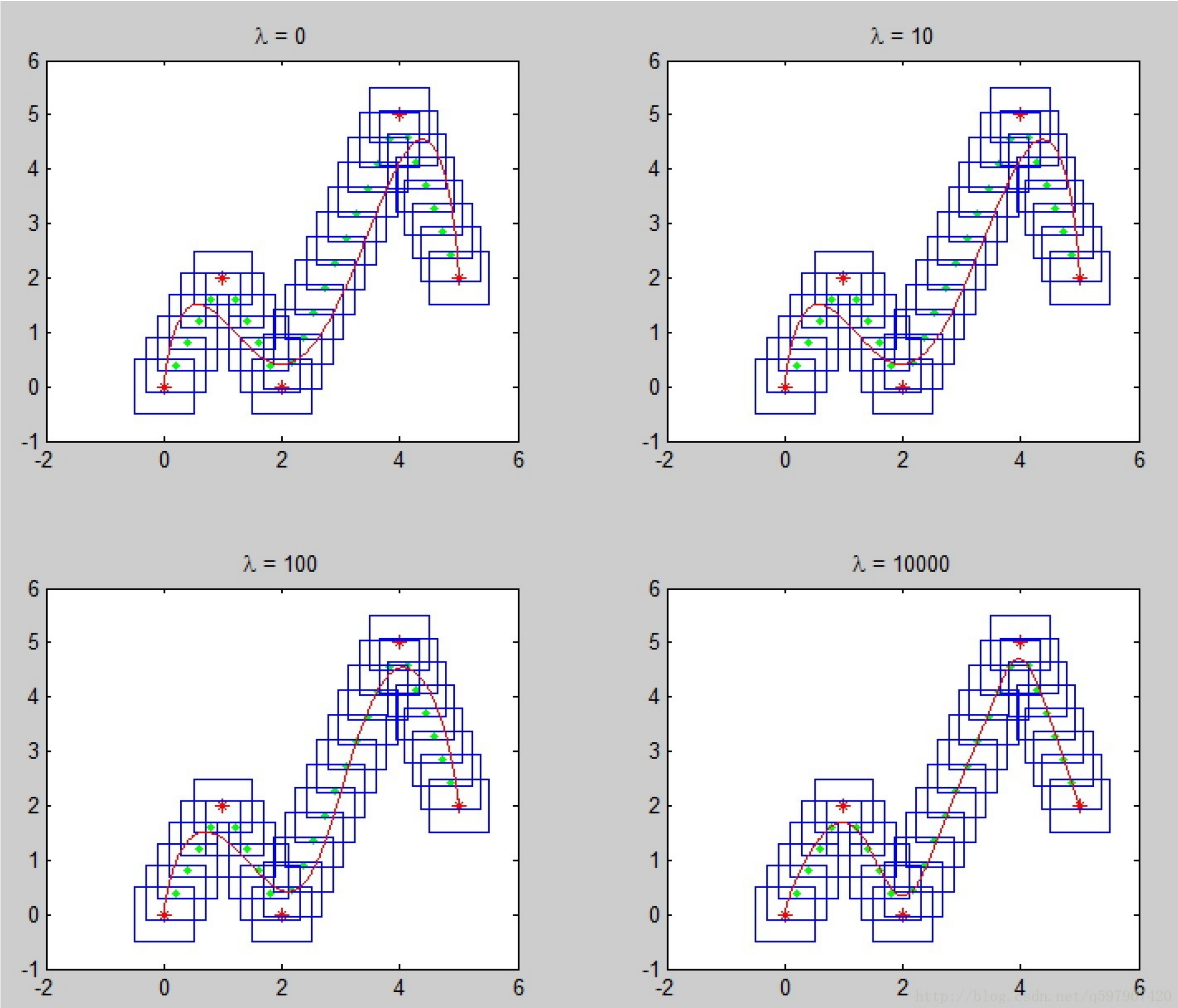
3. 实验结果

这里的实验和《Minimum Snap轨迹规划详解（2）corridor与时间分配》中的基本一样，**唯一的区别在于目标函数中加入了guiding项**。其中，每一段的guiding trajectory设置为两个端点之间的直线，

- 例如，第i段轨迹的两个断点位置分别为$p_{t_{i-1}}, p_{t_i}$，则guiding trajectory参数c_i为：

$$c_i = [p_{t_{i-1}} - kt_{i-1}, k, 0, 0, \ldots, 0]^T,\ k = (p_{t_i} - p_{t_{i-1}})/(t_i - t_{i-1})$$

代码在这里，实验结果如下：



可以看到，λ越大，轨迹与路径越接近，但速度加速度峰值也同样会变得越大，也就是说明接近guiding项与平滑性本质上是矛盾的，鱼和熊掌不可兼得，需要用λ来trade off。这也说明了λ的取值十分关键，实际中λ也很难调，一个固定的λ很难满足所有情况。

4.广义guiding trajectory

上面介绍的是position的guiding项，但同样也可以加入guiding 速度或者guiding加速度，甚至多个guiding项的加权，formulate的方法和guiding position一样，只是会稍微复杂一点，比如给定guiding速度曲线$v(t) = [1, t, t^2, \ldots, t^{(n-1)}]c_v$，需要转换成$p(t) = [1, t, t^2, \ldots, t^n]c$使得$p^{'}(t) = v(t)$，可以推导出

$$c = \begin{bmatrix} 0 & \dots & & & & \\ 1 & 0 & \dots & & & \\ 0 & \frac{1}{2} & 0 & \dots & & \\ 0 & 0 & \frac{1}{3} & 0 & \dots & \\ \vdots & & & & \ddots & \\ 0 & & \dots & 0 & \frac{1}{n-1} & 0 \end{bmatrix} c_v$$

可以看到，n阶轨迹的guiding速度曲线最多为n-1阶，以此类推guiding加速度曲线最多为n-2阶。

参考文献

1. Chen J, Liu T, Shen S. Tracking a moving target in cluttered environments using a quadrotor[C]//Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on. IEEE, 2016: 446-453.
2. Chen J, Liu T, Shen S. Online generation of collision-free trajectories for quadrotor flight in unknown cluttered environments[C]//Robotics and Automation (ICRA), 2016 IEEE International Conference on. IEEE, 2016: 1476-1483.