

Model Predictive Control with Reach-avoid Analysis

基于可达-规避分析模型预测控制

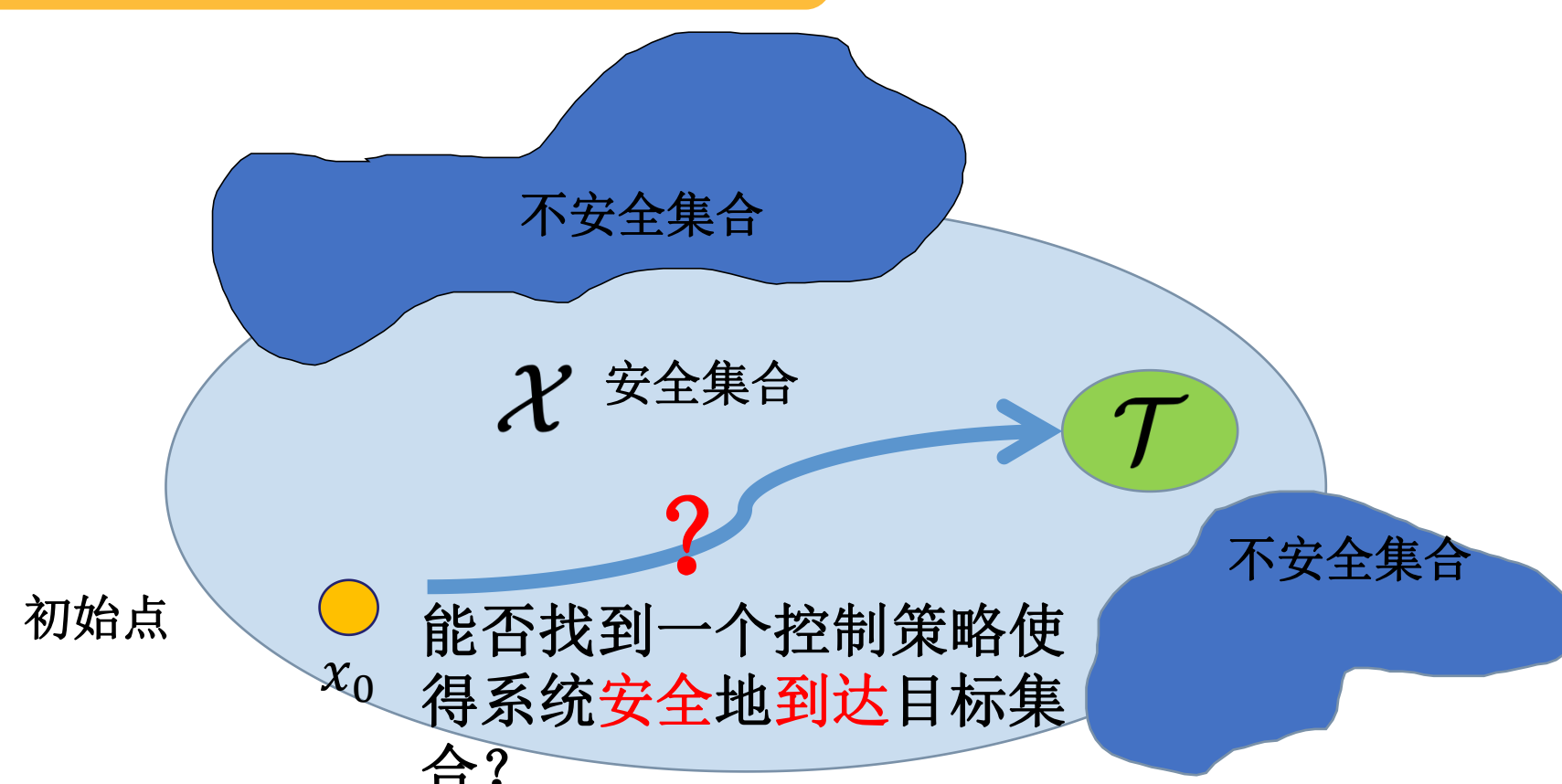
Dejin Ren, Wanli Lu, Jidong Lv, Lijun Zhang and Bai Xue

In Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI 2023)

主要联系人: 任德金 手机号: 18801258257 邮箱: rendj@ios.ac.cn

背景介绍

- 在安全攸关的领域, 实际的应用需要我们解决这样一个控制问题: 给定一个控制系统, 我们如何设计一个控制策略, 使得控制系统满足可达-规避性。所谓的可达-规避性, 就是系统安全到达目标的性质。



- 我们在控制生成的过程中融入了可达-规避验证的方法, 用验证的方法指导满足可达-规避性质控制器的生成。

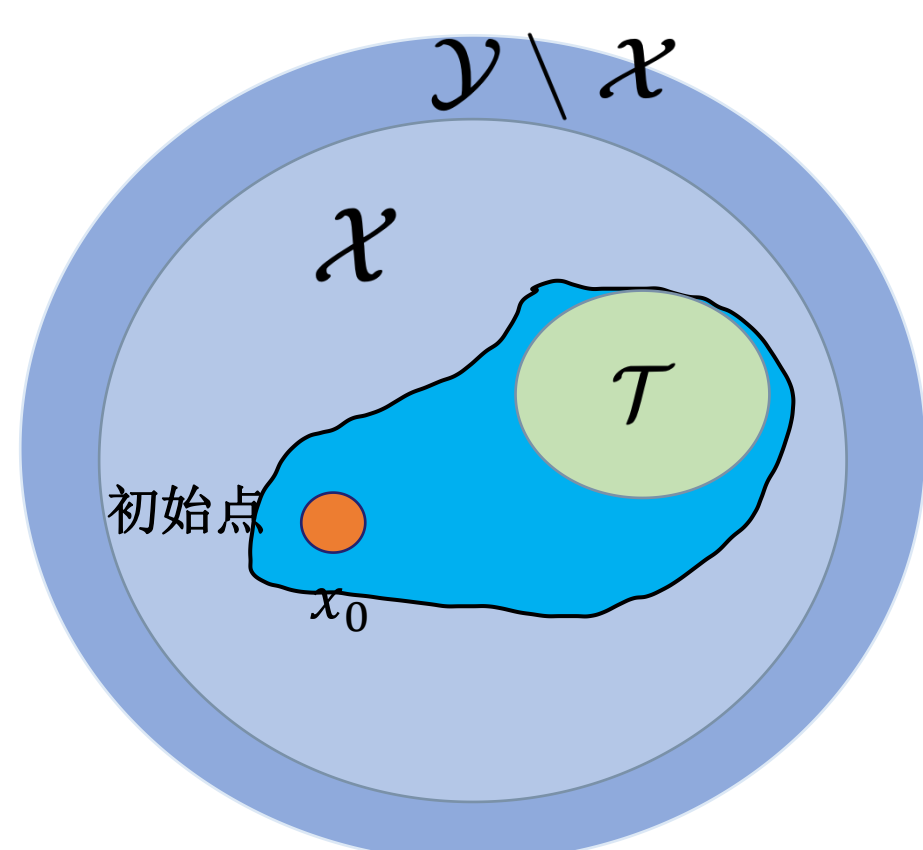
可达-规避分析

针对离散控制系统 $x(t+1) = f(x(t), u(t))$

如果能够找到一个有界函数 $v(x)$ 满足

$$\begin{cases} v(f(x, \hat{u}(x))) \geq \lambda v(x), \forall x \in \mathcal{X} \setminus \mathcal{T}, \lambda > 1 \\ v(x) \leq 0, \forall x \in \mathcal{Y} \setminus \mathcal{X}, \\ v(x) \leq M, \forall x \in \mathcal{T}, \\ v(x_0) > 0, \end{cases}$$

那么该系统满足可达-规避性。



$$\mathcal{Y} = \{y \mid y = f(x, u), u \in \mathcal{U}, x \in \mathcal{X}\} \cup \mathcal{X}$$

安全集合一步到达区域

有界函数 $v(x)$ 被称为 **引导-栅栏函数 (guidance-barrier function)**
其中 $\mathcal{R} = \{x \in \mathcal{X} \mid v(x) > 0\}$ 被称为 **可达-规避集**。
从可达-规避集出发的系统轨迹都能安全地到达目标区域。

通过如下设计, 我们得到了能够生成满足系统可达-规避性控制策略的 MPC 优化问题, 并且设计了迭代算法, 保证每次迭代都有可行解以及每次迭代都会提高控制策略的表现。

Reach-Avoid Model Predictive Control (RAMPC)

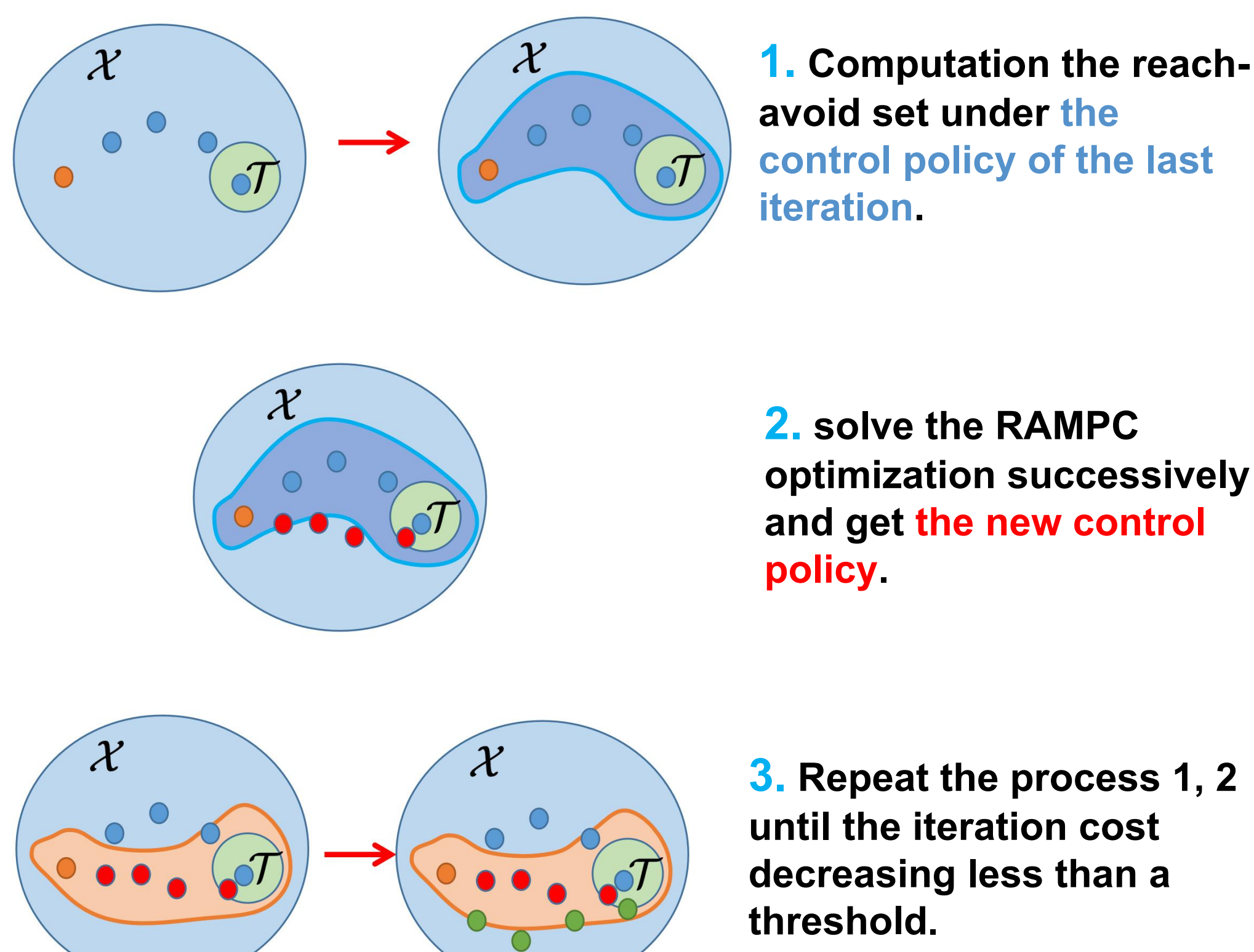
设计了损失函数, 其中 $Q^j(x_{N|t}^j, \pi_{\hat{u}^{j-1}})$ 表示从终端状态出发, 在上一次迭代的控制下到达目标集合系统轨迹的 cost。每次迭代都会提高控制策略的表现。

$$J_{t \rightarrow t+N}^{\text{RAMPC}, j}(x_t^j) = \min_{u_{k|t}} \left[\sum_{k=0}^{N-1} h(x_{k|t}^j, u_{k|t}^j) + Q^j(x_{N|t}^j, \pi_{\hat{u}^{j-1}}) \right]$$

$$\text{s.t.} \begin{cases} x_{k+1|t}^j = f(x_{k|t}^j, u_{k|t}^j), \\ u_{k|t}^j \in \mathcal{U}, x_{k|t}^j \in \mathcal{X}, \\ k = 0, 1, \dots, N-1, \\ v^{j-1}(x_{N|t}^j) \geq \begin{cases} \lambda^N v^{j-1}(x_0), & \text{if } t = 0, \\ \lambda v^{j-1}(x_{N|t-1}^j), & \text{otherwise.} \end{cases} \\ x_{0|t}^j = x_t^j \end{cases}$$

加入了带有 **guidance-barrier function** 的终端限制, 保证了生成控制的可达-规避性。并且保证了每次迭代都有可行解。

RAMPC 迭代计算框架



控制生成

我们采用模型预测控制 (MPC) 作为底层的控制技术, MPC 的控制生成可以形式化地定义为如下的目标优化问题,

$$\min_{u_{k|t}, k=0,1,\dots,N-1} L(x_{0|t}, x_{1|t}, \dots, x_{N|t}, u_{0|t}, u_{1|t}, \dots, u_{N-1|t})$$

$$\text{s.t.} \begin{cases} x_{k+1|t} = f(x_{k|t}, u_{k|t}), \\ u_{k|t} \in \mathcal{U}, k = 0, 1, \dots, N-1, \\ x_{k|t} \in \mathcal{X}, k = 0, 1, \dots, N \\ g(x_{0|t}, x_{1|t}, \dots, x_{N|t}, u_{0|t}, u_{1|t}, \dots, u_{N-1|t}) \leq 0 \end{cases}$$

- 满足一定约束条件下预测未来 N 步的控制状态
- 实际执行第一步或前几步的控制

与现有方法比较

混合整数非线性规划

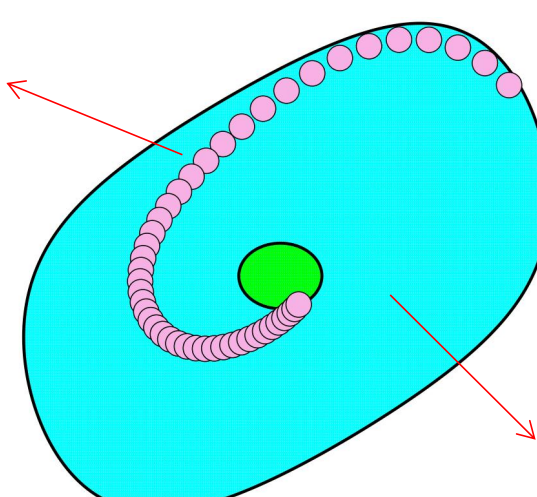
LMPC

$$J(x(t)) = \min_{u_0, \dots, u_{N-1}} \sum_{k=0}^{N-1} (x_k^T Q x_k + u_k^T R u_k) + V^{j-1}(x_N)$$

$$\text{s.t.} \begin{cases} x_{k+1} = f(x_k, u_k), \\ x_0 = x(t), \\ x_k \in \mathcal{X}, u_k \in \mathcal{U}, \\ x_N \in \mathcal{SS}^{j-1}, \\ \forall k \in [0, \dots, N-1] \end{cases}$$

终端限制在 **先前迭代得到的可行状态的集合中**

SS



终端限制在 reach-avoid set (RA) 中

RA

非线性规划

our approach: RAMPC

$$J_{t \rightarrow t+N}^{\text{RAMPC}, j}(x_t^j) = \min_{u_{k|t}} \left[\sum_{k=0}^{N-1} h(x_{k|t}^j, u_{k|t}^j) + Q^j(x_{N|t}^j, \pi_{\hat{u}^{j-1}}) \right]$$

$$\text{s.t.} \begin{cases} x_{k+1|t}^j = f(x_{k|t}^j, u_{k|t}^j), \\ u_{k|t}^j \in \mathcal{U}, x_{k|t}^j \in \mathcal{X}, \\ k = 0, 1, \dots, N-1, \\ v^{j-1}(x_{N|t}^j) \geq \begin{cases} \lambda^N v^{j-1}(x_0), & \text{if } t = 0, \\ \lambda v^{j-1}(x_{N|t-1}^j), & \text{otherwise.} \end{cases} \\ x_{0|t}^j = x_t^j \end{cases}$$

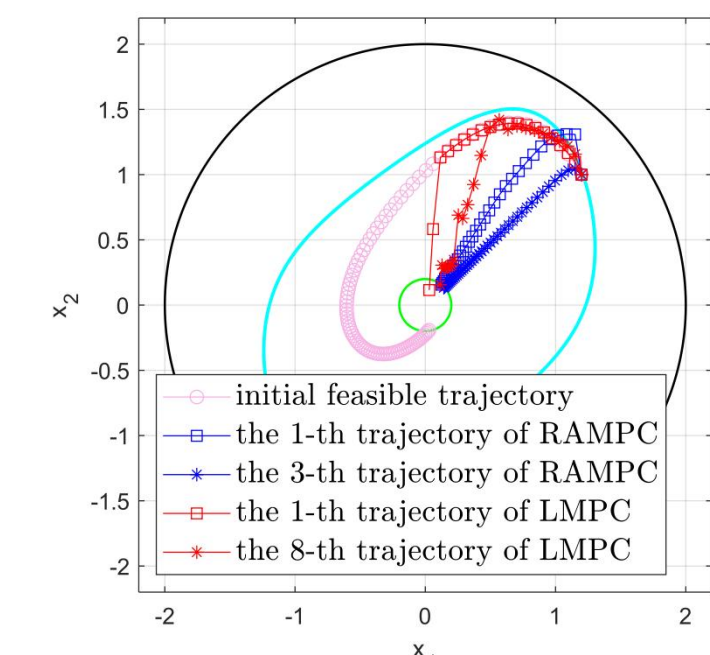
优势:

1. 增大了终端条件的可行域, 求解收敛更快
2. 只是非线性规划, 求解速度更快

实验评估

带控制的 vanderpool 吸引子

$$\begin{cases} x_1(t+1) = x_1(t) - \Delta t x_2(t) \\ x_2(t+1) = x_2(t) - \Delta t((1-x_1(t)^2)x_2(t) - x_1(t)) + u(t) \end{cases}$$



Iteration	Iteration Cost		Iteration	Time Cost(seconds)	
	RAMPC	LMPC		RAMPC	LMPC
0	64.3087	64.3087	1	10.6583	45.0
1	30.9693	40.0714	2	10.2248	59.4
2	29.2919	39.3270	3	10.4010	558.5
3	29.2824	38.5239	4	-	1298
4	-	37.6402	5	-	1653.3
5	-	36.7088	6	-	2033.6
6	-	35.7561	7	-	2115.2
7	-	35.2567	8	-	1706.1
8	-	34.9022	total	31.2841	9469.1

RAMPC 收敛速度更快

RAMPC 花费的时间更少