

摘要

在实际应用中，复杂系统具有模型复杂、规模大、变量多、约束多等特点，传统的控制方法无法满足复杂系统控制的要求。近些年来，分布式模型预测控制（Distributed Model Predictive Control，简称DMPC）以其优良的控制性能、有效处理约束的能力和结构的灵活性等优点得到了广泛关注，成为了处理大规模复杂系统的重要工具。目前，该方法尚处在研究初期，依然存在许多研究难点。例如，如何处理子系统之间的耦合，使得各个子系统的优化问题能在分布式结构下求解；如何设计每个子系统各自的终端成分，减小在线计算量；如何满足优化问题的迭代可行性和整个系统的闭环稳定性。

本文以多智能体系统为研究对象，开展了分布式模型预测控制的系列研究，分别研究了多智能体系统的调节问题、一致问题和编队问题，主要研究成果如下：

1、研究了多个异构线性系统的分布式模型预测调节控制问题，智能体系统之间同时存在成本耦合和约束耦合。在每一个采样时刻，所有智能体同步更新。提出了改进的相容性约束，保证每个智能体实际轨迹和假设轨迹的一致性。通过求解李亚普诺夫方程得到终端权重矩阵，从而设计了终端成本；利用耦合约束的信息，设计了正定不变的终端集，并且保证终端集中的状态满足耦合约束。设计了分布式优化算法，保证局部约束和耦合约束的迭代可行性，以及整个系统的闭环稳定性。最后通过仿真实验比较了集中式模型预测控制方法和分布式模型预测控制方法的性能，对比表明在保证系统性能和集中式的性能接近的条件下，分布式方法大大缩短了优化时间，减小了在线计算量。

2、研究了多个二阶非线性系统的分布式模型预测一致控制问题，智能体系统之间存在成本耦合，所有智能体共享一个参考轨迹。利用同步更新策略，提出了改进的相容性约束上界，保证系统的收敛性；在分布式结构下，利用参考轨迹，设计了每个智能体的终端成本、终端控制器和正定不变的终端集。提出了分布式优化算法，保证优化问题的迭代可行性和整个系统的闭环收敛性。最后，以2-自由度机械臂为例，通过仿真实验验证了算法的有效性。

3、研究了多个线性二阶系统的分布式模型预测编队控制问题，每个智能体到达各自的目标位置形成队形，智能体系统之间存在耦合成本，且需要满足避碰约束。利用同步更新策略，设计了改进的相容性约束，包括位置相容性和状态相容性，同时提出了改进的避碰约束。充分利用系统模型的特点和编队目标，设计了分布式结构下的终端成分，

并且在终端集中依然能保证避碰约束的满足。提出了分布式优化算法，保证优化问题的迭代可行性、整个系统的闭环收敛性以及避碰约束的满足。最后通过仿真实验验证了算法的有效性。

最后，归纳总结了本文的主要工作，并对今后的工作进行了展望。

关键词: 模型预测控制；分布式控制；终端成本；终端控制器；终端集。

Abstract

In the practical applications, the complex systems have the features of complex model, large scale, a great amount of variables and constraints. However, the traditional control methods cannot satisfy the requirements of the complex systems. Over the past years, distributed model predictive control (DMPC) is becoming an important tool to handle the large-scale complex systems due to its good control performance, capability of handling constraints and the structural flexibility. In the embryonic stage, the research of DMPC is confronted with many difficulties. For example, how to deal with the couplings between subsystems in order to make the optimization problem of each subsystem solvable; how to design the terminal ingredients (terminal cost, terminal controller and positive invariant terminal set) for each subsystem; how to guarantee the recursive feasibility and the closed-loop stability of the whole system.

Motivated by this, this dissertation focuses on the research of DMPC for multi-agent systems and explores the problems of regulation, consensus and formation, respectively. The main contents and results in this dissertation are summarized as follows:

1. A DMPC algorithm is presented for the regulation of linear multi-agent systems with the coupled cost and coupled constraints. At each sampling time, all the agents are permitted to synchronously optimise. An improved compatibility constraint is constructed to ensure the consistency between the actual state trajectory of each agent and its assumed one. With the parameters of the coupled constraints, a positive invariant terminal set, which can definitely ensure the satisfaction of the coupled constraint, and an associated terminal cost (a local Lyapunov function) are designed in a distributed manner. By applying the proposed distributed optimization algorithm, the recursive feasibility with respect to both local and coupled constraints and the closed-loop stability of the whole system are guaranteed. In final, the numerical results of the comparisons between the DMPC algorithm and the centralised model predictive control (CMPC) are given to show the effectiveness of the proposed algorithm. The results show that the DMPC algorithm not only dramatically decreases the optimizing time but also maintains most of the control

performance of CMPC.

2. A DMPC algorithm is investigated for the consensus of second-order nonlinear multi-agent systems with coupled cost. All the agents share one reference trajectory. With the synchronous update strategy, a time-varying compatibility constraint, which plays an important to ensure the stability, is presented to ensure that the actual state trajectory of each subsystem does not deviate too much from its assumed one. Furthermore, a positively invariant terminal region and a corresponding auxiliary controller are developed for each agent. Given the designed terminal ingredients and compatibility constraints, the recursive feasibility and closed-loop stability of the whole system are guaranteed. A numerical example of 2-Degree-of-Freedom (2-DoF) robotic manipulator illustrates the efficacy of the proposed algorithm.

3. A DMPC algorithm is explored for the formation of second-order linear multi-agent systems with collision avoidance. With the synchronous update strategy, the improved compatibility constraints with respect to both the position compatibility and the state compatibility, are designed for each agent. The collision avoidance constraints are tightened by replacing the actual positions of each agent's neighbors with their assumed ones. By using the second-order model and the formation, the terminal ingredients are designed. Particularly, the states in the positively invariant terminal sets satisfy the collision avoidance constraints. By implementing the proposed distributed optimization algorithm, the recursive feasibility of the optimization problem, the closed-loop stability of the whole system and the collision avoidance between agents are guaranteed. A numerical example illustrates the effectiveness of the proposed algorithm.

At the end of this dissertation, the main results are concluded and the problems to be solved in the future are presented.

Keywords: Model predictive control (MPC); Distributed control; Terminal cost; Terminal controller; Terminal set.

目录

目录	V
第1章 小型化车载导航组合系统方案设计	1
1.1 引言	1
1.2 车载组合导航系统坐标定义及转换	1
1.2.1 导航坐标系定义	1
1.2.2 坐标系变换原理	3
1.3 导航定位原理	5
1.3.1 四元数与姿态更新	6
1.3.2 速度更新	10
1.3.3 位置更新	11
1.4 小型化车载组合定位原理	12
1.5 车载组合定位系统总体方案设计	12
1.6 本章小结	13
第2章 小型化组合定位系统硬件设计	15
2.1 引言	15
2.2 主要器件选型	15
2.2.1 ARM芯片	15
2.2.2 MEMS器件	16
2.2.3 卫星接收模块	16
2.2.4 电源管理模块	17
2.3 硬件电路设计	18
2.3.1 系统原理图设计	18
2.3.2 系统PCB板设计	20
2.4 分布式模型预测控制	21
2.4.1 相容性约束的设计	22
2.4.2 终端成分的设计	23
2.4.3 分布式实现算法	26
2.5 主要结果	27
2.6 仿真结果和分析	31
2.7 本章小结	33

第1章 小型化车载导航组合系统方案设计

1.1 引言

车载组合定位系统是采用车载传感器进行组合定位的实时系统，随着MEMS 惯性器件工程化及其精度的提升，其小型化、低成本、低功耗抗稳定性好的应用特点在航姿测量系统得到广泛的应用。本文基于MEMS传感器，通过合理的布局设计及系统设计，将惯性测量、惯性导航、卫星导航、姿态解算、组合导航集成在一块单板上。

本章首先介绍导航系统研究中几种常见的坐标系及相互之间的转换，分析车载定位系统的定位原理并给出基于四元数和旋转矢量法的MEMS惯导姿态、位置和速度更新方程。最后给出车载组合定位系统的总体方案设计。

1.2 车载组合导航系统坐标定义及转换

宇宙中的物体小到分子原子大到行星恒星都在不停地运动，但运动是相对的，单个没有参考的物体是无运动可言的。一个物体在空间的位置只能相对另一个物体确定，后一个物体就是描述前一个物体运动的参考系，可以在该物体上建立直角坐标系。在导航定位过程中，运动体的姿态、速度、位置等信息的确定都必须建立在相应的参考坐标系中。对于不同的研究对象和任务要求，需要选择不同的坐标系。

1.2.1 导航坐标系定义

在陆地车辆组合导航系统中，常见的坐标系有：惯性坐标系、地球坐标系、地理坐标系、机体坐标系、导航坐标系。

(1)惯性坐标系(i系)

惯性坐标系(以下简称惯性系)是遵循牛顿定律的坐标系，即相对于惯性空间做匀速直线运动或绝对静止的坐标系，惯性坐标系为惯性器件测量的参考基准。完全理想的惯性坐标系本身并不存在，在实际应用中通过使用近似的地心惯性坐标系来代替惯性坐标系进行运算。

(2) 地球坐标系(e系)

与地球固连，以地球质心为坐标原点，轴与地球自转轴方向一致。

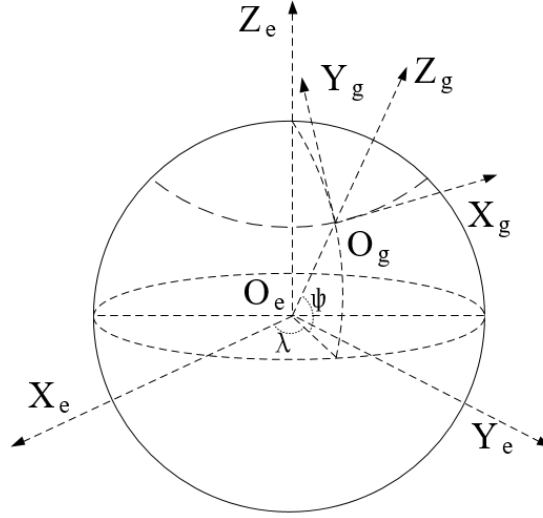


图 1.1 地球坐标系与地理坐标系示意图

(3) 地理坐标系(g系)

如图所示，地理坐标系 $OEN\xi$ 的原点 O 取在载体的重心， E 轴在当地水平面内的指东， N 轴在当地水平面指北， ξ 轴则沿当地地垂线指向天空的方向， $EN\xi$ 三轴构成右手直角坐标系。地理系是随着载体运动的，可以按“东北天”“北西天”或者“北东地”顺序构成，本文中的地理坐标系选择“东北天”坐标系描述。

(4) 载体坐标系(b系)

如图所示，载体坐标系的原点 O_b 位于载体重心位置，与运载体固联， x_b 轴沿载体的横轴方向， y_b 沿载体的纵轴方向， z_b 垂直机体水平面向上， $x_b y_b z_b$ 轴构成右手直角坐标系，载体坐标系相对于地理坐标系的方位称为载体的姿态和航向。

(5) 导航坐标系(n系)

导航坐标系是惯性导航系统求解导航参数时所采用的坐标系用 $Ox_n y_n z_n$ 。针对捷联惯导来说，传感器所采的数据是沿载体坐标系轴向的，必须将该采集的信息分解到便于描述和求解导航参数的坐标系内，然后进行导航解算，这个坐标系即导航坐标系。本文选择的导航坐标系为地理坐标系即东北天坐标系。

当物体在地球上运动时，物体相对地球的位置不断改变；同时，地球上的点的地理坐标系相对于地球系的角位置是不相同的。运载体相对于地球运动引起地理坐标系相对

地球坐标系的转动。此时，地理坐标系相对惯性坐标系的转动角速度有两部分组成：一个是地球坐标系相对于惯性坐标系的转动角速度；另一部分是地理坐标系相对于地球坐标系的转动角速度。两部分具体的计算过程在后续章节详细介绍。确定物体的地理位置，就是确定地理坐标系相对于地球坐标系的方位关系，即经度 λ 纬度 φ 和到地球地心的距离 R ；而物体的姿态则是确定载体坐标系相对于地理坐标系的方位关系。

1.2.2 坐标系变换原理

由上面定义的几种坐标系可知，在分析惯性导航系统运动时，将用到多种坐标系，而这些坐标系并不是相互孤立的。虽然空间中一个向量在不同的坐标系里的坐标是不同的，但是通过坐标变换，可以将一个坐标系下的向量坐标转换到另一个坐标系下。这种变换包括平移和旋转，在惯性导航中坐标系的变换我们更关注坐标之间的角度关系，即旋转坐标变换，具体的转换方法见文献。以下介绍两种坐标之间的转换。

(1) 地球坐标系与地理坐标系之间的转换

根据图可知，地球坐标系和地理坐标系的原点不重合，当用经度 λ 和纬度 φ 来表示旋转角度时，其转动过程可以描述为地球坐标系绕 Z_e 轴转动 $(90^\circ + \lambda)$ ，再绕变换后的 x 轴转动 $(90^\circ - \varphi)$ ，最后绕变换后的 z 轴旋转 90° 而得到地理坐标系。

$$\begin{aligned}
 \begin{bmatrix} E \\ N \\ \xi \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(90^\circ - \varphi) & 0 & -\sin(90^\circ - \varphi) \\ 0 & 1 & 0 \\ \sin(90^\circ - \varphi) & 0 & \cos(90^\circ - \varphi) \end{bmatrix} \begin{bmatrix} \cos \lambda & \sin \lambda & 0 \\ -\sin \lambda & \cos \lambda & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} \\
 &= \begin{bmatrix} -\sin \lambda & \cos \lambda & 0 \\ -\sin \varphi \cos \lambda & -\sin \varphi \sin \lambda & \cos \varphi \\ \cos \varphi \cos \lambda & \cos \varphi \sin \lambda & \sin \varphi \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} \quad (1.1)
 \end{aligned}$$

用 C_e^n 表示变换矩阵，有

$$C_e^n = \begin{bmatrix} -\sin \lambda & \cos \lambda & 0 \\ -\sin \varphi \cos \lambda & -\sin \varphi \sin \lambda & \cos \varphi \\ \cos \varphi \cos \lambda & \cos \varphi \sin \lambda & \sin \varphi \end{bmatrix} \triangleq \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \quad (1.1)$$

式中, \mathbf{C}_e^n 为地球坐标系到地理坐标系的变换矩阵; \mathbf{C}_{ij} 为变换矩阵的元素 ($i = 1, 2, 3; j = 1, 2, 3$)。

(2) 地理坐标系和载体坐标系之间的转换

载体的姿态和航向实际上就是载体坐标系和地理坐标系之间的方位关系。载体的姿态可用如下欧拉角表示:

俯仰角(θ): 载体绕横轴 x_b 转动时, 载体的纵轴与水平面之间会形成一个夹角, 该角即为俯仰角, 一般用 θ 表示, 向上角度为正, 向下为负, 俯仰角取值范围为 $[-90^\circ, 90^\circ]$ 。

翻滚角(γ): 载体绕纵轴 y_b 转动时, 载体的横轴与水平面之间会形成一个夹角, 该角即为翻滚角, 一般用 γ 表示, 以右端向下为正, 翻滚角取值范围为 $[-90^\circ, 90^\circ]$ 。

航向角(ψ): 载体绕方位轴转动时, 载体纵轴 y_b 在地理坐标系下的水平投影与地理子午线 N 的夹角称为航向角, 一般用 ψ 表示, 以正北方向为起点并以北偏东方向逐渐增加, 航向角取值范围为 $[0^\circ, 360^\circ]$ 。

载体坐标系和导航坐标系之间可以绕 $Z - X - Y$ 的顺序转动三次完成转换, 载体姿态矩阵 \mathbf{C}_b^n 实现转换, 变换公式如下:

$$\begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} = \mathbf{C}_b^n \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} \quad (1.2)$$

第一次转动: 坐标系绕 Z 轴负向转动角 ψ :

$$\mathbf{C}_b^{n1} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.3)$$

第二次转动: 坐标系绕 X 轴负向转动角 θ :

$$\mathbf{C}_{n1}^{n2} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix} \quad (1.4)$$

第三次转动：坐标系统Y轴负向转动角 ψ ：

$$\mathbf{C}_{n2}^n = \begin{bmatrix} \cos \gamma & 0 & -\sin \gamma \\ 0 & 1 & 0 \\ \sin \gamma & 0 & \cos \gamma \end{bmatrix} \quad (1.5)$$

$$\begin{aligned} \mathbf{C}_b^n &= \mathbf{C}_{n2}^n \mathbf{C}_{n1}^{n2} \mathbf{C}_b^{n1} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \cos \gamma & 0 & -\sin \gamma \\ 0 & 1 & 0 \\ \sin \gamma & 0 & \cos \gamma \end{bmatrix} \\ &= \begin{bmatrix} \cos \gamma \cos \psi + \sin \gamma \sin \psi \sin \theta & \sin \psi \cos \theta & \sin \gamma \cos \psi - \cos \gamma \sin \psi \sin \theta \\ -\cos \gamma \cos \psi + \sin \gamma \sin \psi \sin \theta & \cos \psi \cos \theta & -\sin \gamma \cos \psi - \cos \gamma \sin \psi \sin \theta \\ -\sin \gamma \cos \theta & \sin \theta & \cos \gamma \cos \theta \end{bmatrix} \end{aligned}$$

定义 \mathbf{C}_b^n 为如下形式：

$$\mathbf{C}_b^n = \begin{bmatrix} \mathbf{T}_{11} & \mathbf{T}_{12} & \mathbf{T}_{13} \\ \mathbf{T}_{21} & \mathbf{T}_{22} & \mathbf{T}_{23} \\ \mathbf{T}_{31} & \mathbf{T}_{32} & \mathbf{T}_{33} \end{bmatrix} \quad (1.6)$$

\mathbf{C}_b^n 就是导航系统中通常所称的捷联矩阵，载体的姿态角可以计算如下：

$$\begin{cases} \psi_{\pm} = \arctan\left(\frac{\mathbf{T}_{13}}{\mathbf{T}_{22}}\right) \\ \theta_{\pm} = \arcsin(\mathbf{T}_{32}) \\ \gamma_{\pm} = \arctan\left(\frac{-\mathbf{T}_{31}}{\mathbf{T}_{33}}\right) \end{cases} \quad (1.7)$$

对于地面车辆导航系统，式中，俯仰角 θ 、翻滚角 γ 定义在区间 $[-90^\circ, 90^\circ]$ 中，在计算反正弦和反正切函数中没有多值问题；航向角 ψ 定义在 $[0^\circ, 360^\circ]$ 区间，需要分情况考虑根据主值区间和方向矩阵的数值来确定角度大小，其真值表如表所示。

1.3 导航定位原理

在捷联惯性导航系统(SINS)中惯性测量器件(陀螺仪和加速度计)直接与运载体固联，通过导航计算机采集惯性器件的输出信息并进行数值积分求解运载体的姿态、速度和位

表 1.1 航向角 ψ 的真值表

T_{22}	T_{12}	ψ
$\rightarrow 0$	+	90°
$\rightarrow 0$	-	270°
+	+	ψ_{\pm}
+	-	$180^\circ + \psi_{\pm}$
-	+	$\psi_{\pm} + 180^\circ$
-	-	$\psi_{\pm} - 180^\circ$

置等导航信息，这三个过程分别是姿态更新、速度更新和位置更新。MEMS惯导一般采用角速率输出采样方式，简单的将输出信号乘以采样间隔 T_s 即可变换为近似的角增量，利用单位时间角增量数据可以实现载体姿态的更新，姿态指的是载体坐标系相对于导航坐标系的角位置，具体定义及坐标变换已在上一章具体介绍。MEMS惯导信息的解算核心是姿态更新，对整个系统的解算精度影响最为突出，具有重要的研究和应用价值。

1.3.1 四元数与姿态更新

根据所选的变量参数不同，常用的姿态更新算法有方向余弦法、欧拉角法和四元数法(Quaternations)。这几种姿态解算方法分别适合不同的姿态变化情况。

欧拉角算法通过求解欧拉角微分方程直接求解姿态角。欧拉角解算简单明了，概念直观，容易理解，但求解过程中需要频繁解算三角函数，运算比较困难，而且当俯仰角接近 90° 时会引起万向节死锁问题，所以该方法适合水平姿态变化不大，实时性要求不高的情况下。

方向余弦和欧拉角并没有本质上的区别，因为是用欧拉角表示的方向余弦阵，避免了退化的问题，可以计算全姿态，但是姿态矩阵微分方程包含九个未知量，因此算法设计计算量大，实时性差，不利于工程上的应用。

四元数表示方法虽然有些抽象，但它的优点有很多，利用起来非常方便，算法实现上简单易操作，是很实用的工程方法。本课题是基于四元数法完成姿态矩阵的解算。四

元数是由威廉卢云哈密尔顿在1843年爱尔兰提出的数学概念，其基础思想为：一个坐标系到另一个坐标系的变换可以通过绕一个定义在参考系中的矢量 μ 的单次转动来实现，四元数提供该转动的数学描述，一个单位四元数可以为表示一个旋转。四元数法实质上是旋转矢量法中的单子样算法，对有限旋转引起的的补偿程度不够，只适用于低动态运载体的姿态解算，而对于高动态运载体的姿态解算中漂移非常严重，对于低动态的车载组合导航系统来说，四元数方法十分适用。

1、四元数定义

四元数是由四个元构成的数：

$$Q(q_0, q_1, q_2, q_3) = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} = \mathbf{q}_0 + \mathbf{q}_v \quad (1.8)$$

其中， q_0 、 q_1 、 q_2 、 q_3 是实数， q_0 也称为实部， $q_v = q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$ 称为虚部， \mathbf{i} 、 \mathbf{j} 、 \mathbf{k} 是互相正交的单位向量，可以将四元数看作是四维空间中的一个向量或者一个超复数。四元数的虚单位 \mathbf{i} 、 \mathbf{j} 、 \mathbf{k} 之间满足如下乘法运算规则

$$\begin{cases} \mathbf{i} \otimes \mathbf{i} = \mathbf{j} \otimes \mathbf{j} = \mathbf{k} \otimes \mathbf{k} = -1 \\ \mathbf{i} \otimes \mathbf{j} = \mathbf{k}, \mathbf{j} \otimes \mathbf{k} = \mathbf{i}, \mathbf{k} \otimes \mathbf{i} = \mathbf{j}, \mathbf{j} \otimes \mathbf{i} = -\mathbf{k}, \mathbf{k} \otimes \mathbf{j} = -\mathbf{i}, \mathbf{i} \otimes \mathbf{k} = -\mathbf{j} \end{cases} \quad (1.9)$$

其中，运算符“ \otimes ”表示四元数乘法运算，虚部单位矢量的叉乘运算特点，可以将四元数的虚数部分 $\mathbf{q}_v = q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$ 看成是在三维空间中的映像，反之，一个三维矢量可以看做一个零标量四元数。

2、四元数与姿态矩阵的转换

类比于复数的三角表示法，四元数写成三角函数为

$$Q = \|Q\|(\cos \frac{\phi}{2} + \mu \sin \frac{\phi}{2}) \quad (1.10)$$

特别地，当 $\|Q\| = 1$ 时，即对于单位四元数，有， $q_0 = \cos \frac{\phi}{2}$ ， $\mathbf{q}_v = \mu \sin \frac{\phi}{2}$ 且 $q_0^2 + \mathbf{q}_v^T \mathbf{q}_v = 1$ ； μ 为单位长度的三维矢量，即 $\mu^T \mu = 1$ ； ϕ 为某一角度值。在刚体转动理论中，载体系相对于导航系的位置，可以等效成载体系绕某定轴旋转一定角度 θ ，用向量 μ 表示该定轴，则两坐标之间的位置关系可以用角度 θ 和单位向量 μ 来表示如下图。

因此，推导出由四元数表示的载体系至导航系的转换矩阵：

$$C_b^n = \begin{bmatrix} 1 - 2(q_2^2 + q_3^2) & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & 1 - 2(q_1^2 + q_3^2) & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & 1 - 2(q_1^2 + q_2^2) \end{bmatrix} \quad (1.11)$$

3、姿态更新四元数算法

根据四元数理论，坐标系转换过程可以表示为：

$$\begin{aligned} \mathbf{v}^b &= \mathbf{Q}_b^n \otimes \mathbf{v}^n \otimes (\mathbf{Q}_b^n)^* \\ &= \begin{bmatrix} q_0^2 + q_1^2 + q_2^2 + q_3^2 & 0 & 0 & 0 \\ 0 & q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 0 & 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 0 & 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \end{aligned}$$

式中， \mathbf{Q}_b^n 表示导航系n到载体系b的变换四元数； $(\mathbf{Q}_b^n)^*$ 是 \mathbf{Q}_b^n 的共轭表示， \mathbf{v}^n 和 \mathbf{v}^b 表示同意向量分别在导航系n和载体系b下的表示，其向量表示分别为

$$\begin{aligned} \mathbf{v}_n &= \begin{bmatrix} 0 & \mathbf{v}_x^n & \mathbf{v}_y^n & \mathbf{v}_z^n \end{bmatrix}^T \\ \mathbf{v}_b &= \begin{bmatrix} 0 & \mathbf{v}_x^b & \mathbf{v}_y^b & \mathbf{v}_z^b \end{bmatrix}^T \end{aligned}$$

根据方向余弦理论：

$$\begin{aligned} \mathbf{v}^b &= \begin{bmatrix} * & 0 \\ 0 & \mathbf{C}_t^b \end{bmatrix} \mathbf{v}^t \\ &= \begin{bmatrix} * & 0 & 0 & 0 \\ 0 & \cos \gamma \cos \psi + \sin \gamma \sin \psi \sin \theta & -\cos \gamma \cos \psi + \sin \gamma \sin \psi \sin \theta & -\sin \gamma \cos \theta \\ 0 & \sin \psi \cos \theta & \cos \psi \cos \theta & \sin \theta \\ 0 & \sin \gamma \cos \psi - \cos \gamma \sin \psi \sin \theta & -\sin \gamma \cos \psi - \cos \gamma \sin \psi \sin \theta & \cos \gamma \cos \theta \end{bmatrix} \begin{bmatrix} 0 \\ \mathbf{v}_x^n \\ \mathbf{v}_y^n \\ \mathbf{v}_z^n \end{bmatrix} \end{aligned}$$

这样可以得到变换过程中姿态角的四元数计算方法：

$$\begin{cases} \psi_{\pm} = -\arctan\left(\frac{2(q_1q_2 + q_0q_3)}{q_0^2 - q_1^2 + q_2^2 - q_3^2}\right) \\ \theta_{\pm} = \arcsin(2(q_2q_3 - q_0q_1)) \\ \gamma_{\pm} = \arctan\left(\frac{2(q_1q_3 + q_0q_2)}{q_0^2 - q_1^2 - q_2^2 + q_3^2}\right) \end{cases} \quad (1.12)$$

四元数微分方程

已知 Q 为导航系至载体系的四元数表达式，对两端求导得四元数的微分方程：

$$\begin{aligned} d\frac{Q}{t} &= -\sin\frac{\dot{\theta}}{2}\sin\frac{\theta}{2} + \mu^R\frac{\dot{\theta}}{2}\cos\frac{\theta}{2} + \sin\frac{\theta}{2}\frac{d\mu^R}{dt} \\ &= -\sin\frac{\dot{\theta}}{2}\sin\frac{\theta}{2} + \mu^R\frac{\dot{\theta}}{2}\cos\frac{\theta}{2} \\ &= \frac{\dot{\theta}}{2}\mu^R \otimes Q \end{aligned}$$

由刚体动力学知识可知

$$\begin{aligned} \omega_{nb}^n &= \dot{\theta}\mu^R \\ \omega_{nb}^n &= Q \otimes \omega_{nb}^b \otimes Q^* \end{aligned}$$

式？中 ω_{nb}^b 是MEMS陀螺测量载体系相对于导航系下的角速度，结合上述两式式？可以计算记为

$$d\frac{Q}{t} = \frac{1}{2}Q \otimes \omega_{nb}^b \quad (1.13)$$

记

$$\omega_{nb}^b = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T \quad (1.14)$$

矩阵形式：

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (1.15)$$

其中， ω_{nb}^b 可以由下式得

$$\omega_{nb}^b = \omega_{ib}^b - C_n^b(\omega_{ie}^n + \omega_{en}^n) \quad (1.16)$$

上式中， C_n^b 是姿态阵， ω_{nb}^b 不能直接测量获得，MEMS陀螺的输出是载体系相对于惯性系的角速度 ω_{ib}^b ，其中包含了 ω_{ie}^n 是地球自转速率， ω_{en}^n 是位置速率，大小为：

$$\omega_{ie}^n + \omega_{en}^n = \begin{bmatrix} -\frac{V_N}{R_M} \\ \omega_{ie} \cos L + \frac{V_E}{R_N} \\ \omega_{ie} \sin L + \frac{V_E}{R_N} \tan L \end{bmatrix} \quad (1.17)$$

其中， L 是导航系所在地理纬度， V_E 、 V_N 为当时时刻载体的东向和北向速度， ω_{ie} 是地球自转角速度。

1.3.2 速度更新

本文选择“东-北-天”导航坐标系(n 系), 则有惯导比力方程

$$\dot{\mathbf{V}}_{en}^n = \mathbf{C}_b^n \mathbf{f}_{sf}^b - (2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) \times \mathbf{V}_{en}^n + \mathbf{g}^n \quad (1.18)$$

其中 \mathbf{f}_{sf}^b 为加速度计测量的比力, $2\boldsymbol{\omega}_{ie}^n \times \mathbf{v}_{en}^n$ 为由载体运动和地球自转引起的哥氏加速度, $\boldsymbol{\omega}_{en}^n \times \mathbf{v}_{en}^n$ 为由载体运动引起的对地向心加速度, \mathbf{g}^n 为重力加速度, $-(2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) \times \mathbf{v}_{en}^n + \mathbf{g}^n$ 。式表明, 在加速度计输出中去掉有害加速度后, 就可获得载体在导航系下的真实加速度 $\dot{\mathbf{V}}_{en}^n$, 设速度的更新周期为 T , 在每个更新周期内对角增量和速度增量作 N 次采样, 对上式作积分运算, 得到 t_m 时刻载体在导航系内的速度:

$$\mathbf{V}_m^n = \mathbf{V}_{m-1}^n + \mathbf{C}_{b(m-1)}^{n(m-1)} \int_{t_{m-1}}^{t_m} \mathbf{C}_{b(t)}^{b(m-1)} \mathbf{f}_{ib}^b dt + \int_{t_{m-1}}^{t_m} [\mathbf{g}^n - (2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) \times \mathbf{V}_{en}^n] dt \quad (1.19)$$

式中, $\mathbf{V}_m \mathbf{V}_{m-1}$ 分别是 t_m 和 t_{m-1} 时刻载体的速度; $\mathbf{C}_{b(m-1)}^{n(m-1)}$ 是 t_{m-1} 时刻的姿态矩阵。若 $\delta \mathbf{V}_{g/corm}^n$ 表示在时间 $[t_{m-1}, t_m]$ 内有害加速度引起的速度补偿量, $\delta \mathbf{V}_{sfm}^n$ 是由比力引起的速度补偿量, 则式? 可以改写成

$$\mathbf{V}_m^n = \mathbf{V}_{m-1}^n + \mathbf{C}_{b(m-1)}^{n(m-1)} \delta \mathbf{V}_{sfm}^n + \delta \mathbf{V}_{g/corm}^n \quad (1.20)$$

$$\delta \mathbf{V}_{sfm}^n = \delta \mathbf{V}_m^n + \frac{1}{2} \delta \theta_m \times \delta \mathbf{V}_m^n + \frac{1}{2} \int_{t_{m-1}}^{t_m} [\Delta \theta(t) \times \mathbf{f}_{ib}^b + \Delta \mathbf{V}(t) \times \boldsymbol{\omega}(t)] dt \quad (1.21)$$

式? 在MEMS惯导速度更新中的第二项和第三项分别进行了旋转效应和划桨效应的补偿, 旋转效应指的是运动物体在某个方向进行线运动的同时发生空间旋转变化的引起, 划桨效应指运动物体在纵轴方向做线振动时又在横轴方向发生同向频的角振动。 $\delta \theta_m$ 、 $\Delta \mathbf{V}_m$ 为更新周期内的陀螺角增量和加速度计速度增量。因为载体角速度和加速度变化复杂, 无法通过确定的数学公式加以描述, 工程中常采用曲线拟合的方法来近似求解。根据比力方程, 可以得到离散系统时速度在导航系的增量为:

$$\Delta \mathbf{V}^n = \mathbf{C}_b^n \mathbf{f}_{ib}^b \cdot T - (2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) \times \mathbf{V}^n \cdot T + \mathbf{g}^n \cdot T \quad (1.22)$$

由于低成本MEMS传感器属于低精度, 陀螺精度(零偏稳定性及重复性)为 0.1° 量级, 加速度精度为 $5mg$, 传感器标定在下节介绍具体方法, 此处忽略划桨效应, 可对其作大幅简化。

1.3.3 位置更新

如图，假设地球表面运动的物体在地理坐标系下的坐标记为 $O_g(\lambda, L, h)$ 。如果 O_o 点是 O_g 与地心连线在地球表面的交点，以该点建立直角坐标系，纬圈切线指东为 $O_o x_o$ 轴，经圈切线指北 $O_o y_o$ 轴、椭球面法线方向指天 $O_o z_o$ 轴， O_g 点的地理坐标表示，记为 $O_g(\lambda, L, h)$ 。

在图?，如果 O_o 点对地球坐标系 $O_x y_e z_e$ 的速度在 $O_x o y_o z_o$ 系的投影记为 $v_e^o = [v_{x_o} \ v_{y_o} \ v_{z_o}]^T$ 。注意，由于 $O_o X_o$ 轴与纬圈相切，故经度会随着 v_{x_o} 而变化，根据地理几何有：

$$\dot{\lambda} = \frac{v_{x_o}}{R_N \cos L} \quad (1.23)$$

对于地理高度为 h 的 O_g 点，假设其速度为 $\mathbf{v}_{eg}^g = [v_x \ v_y \ v_z]^T$ ，根据图? 中几何关系，有

$$\frac{v_{x_o}}{R_N} = \frac{v_x}{R_N + h} \quad (1.24)$$

$$\frac{v_{y_o}}{R_M} = \frac{v_y}{R_M + h} \quad (1.25)$$

上述两式分别带入

$$\dot{\lambda} = \frac{v_x}{(R_N + h) \cos L} \quad (1.26)$$

$$\dot{L} = \frac{v_y}{(R_M + h)} \quad (1.27)$$

$$\dot{h} = v_z \quad (1.28)$$

记：

$$\mathbf{p} = \begin{bmatrix} \lambda \\ L \\ h \end{bmatrix}, \mathbf{M}_{pv} = \begin{bmatrix} 0 & 1/[(R_N + h) \cos L] & 0 \\ 1/(R_M + h) & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.29)$$

则矩阵形式，

$$\dot{\mathbf{p}} = \mathbf{M}_{pv} \mathbf{v}^n \quad (1.30)$$

与捷联惯导姿态和速度更新算法相比，位置更新算法引起的误差一般比较小，采用梯形积分法将上式离散化，得

$$\mathbf{p}_t = \mathbf{p}_{t-1} + \mathbf{M}_{pv} \times (\mathbf{v}_{t-1}^n + \mathbf{v}_t^n) \times \frac{T}{2} \quad (1.31)$$

1.4 小型化车载组合定位原理

结合GPS定位的长期稳定性与SINS系统定位的短期精确性，整合两套设备的导航信息可以提高整个系统的定位性能，目前常见的组合方法有：GPS/SINS松组合、紧组合和超紧组合等模式。

(1)GPS/SINS松组合

松组合是采用滤波计算的较为基础的组合方式，从GNSS接收机和惯导系统各自输出的速度、位置之差作为系统量测，通过组合滤波器对惯导系统的速度、位置误差和姿态误差进行估计，然后作为反馈信息完成对MEMS惯导传感器的校正，从而实现SINS系统的累积误差的消除，提高组合系统的导航精度。

松组合的特点是：系统结构简单，易于实现，计算量小等，GNSS可以限制惯导误差的增长，当GNSS失去信号时惯导可短时间内自主导航，惯性传感器误差实时估计并修正。适合车载导航、民用航空等领域。其结构示意图如下？

(2)紧组合

根据

1.5 车载组合定位系统总体方案设计

车载组合定位系统是一个复杂的系统，包括传感器的数据采集与处理，实时数据解算和融合，上位机导航信息显示等设计。本课题根据本章前几节的导航解算理论知识，根据实际工作环境完成芯片选型与硬件设计，并编写导航软件实现导航定位功能，最终可以成功地为地面车辆提供姿态位置和速度信息，完成了一款自动驾驶辅助系统的小型化导航模块的工程实现。此模块以ARM 处理器为核心，主要由MEMS惯性传感器，GNSS卫星接收机等设备组成。构建的SOPC(system-on-a-programmable-chip)嵌入式系统具有体积小，功耗低，数据计算功能强等优点，为小型化、低功耗，低成本导航系统硬件设计提供了解决方案，其总体方案如图？，

从图中可以看出，首先将MEMS惯性器件通过SPI通信接口以100Hz的频率采样，降频至10Hz送入ARM芯片数据缓存区，降频采用简单的均值滤波并做误差补偿，供后续系统做惯导解算使用；另一路GNSS模块通过串口连接到ARM芯片，并以1Hz 速率将数据送入数据缓存区；ARM芯片在解析惯导数据和GNSS数据后，对结果进行时间对准后

进行卡尔曼滤波融合对导航数据进行最优估计，最后通过串行总线把导航解算数据传送到PC端进行显示，同时也可将结果实时记录在SD中，方便对实时环境的工作测试。

根据上述系统总体方案的介绍，GNSS/INS组合导航系统中，不仅要完成导航信息解算等软件问题、首先要完成硬件平台的开发。结合本系统的功能及特点可以将设计的主要任务主要分成以下几点：

1、硬件平台设计及模块调试

导航模块硬件设计是整个任务要求的第一步，再设计最初，要根据平台的设计功能和算法需求特点进行主要器件的选型，然后依据电气连接规则绘制电路原理图和PCB图并加工制作电路板，硬件设计的过程在将在第三章详细介绍。

2、软件设计在硬件平台搭建完成之后，要实现导航定位的功能还需要软件系统的支持。根据系统总体设计思想，软件设计分为相关芯片驱动部分、惯导数据解算、以及组合滤波误差处理几部分。具体过程为：首先对ARM硬件系统进行初始化，定时器、串口、SPI接口等外设初始化，然后通过定时器对IMU和GNSS数据进行采集，对惯导数据进行导航解算，解算结果与GPS观测量通过卡尔曼滤波得到最优的导航信息，软件的设计将在本文第四章进行介绍。

3、导航数据输出显示一个完整的系统设计最终需要一个人机交互的功能，导航解算后的数据要给用户显示出来，最终的设计将导航结果通过串口连接至电脑端，并能实时监测物体的运行状态和轨迹，同时显示当前搜索到的卫星，并将导航数据保存下来以便调试使用。

较之传统定位系统，本系统具有导航定位独立，运行设计可靠，利于在中低端设备上实现等优点。

1.6 本章小结

本章主要介绍了捷联惯导的基本原理，推导了MEMS惯导的位置、姿态、速度更新算法，以及MEMS惯导的误差模型，给出了航位推算的方法，结合卫星导航与SINS导航组合原理，最后整合各传感器的特点，给出了小型化组合导航系统的总体方案，提出了系统设计的任务要求及需要实现的功能，为后文提供理论基础。

第2章 小型化组合定位系统硬件设计

2.1 引言

根据前一章系统总体方案的设计，本章的重点工作是完成系统的硬件平台。车载组合定位系统硬件设计分为：器件选型、嵌入式ARM系统设计、外围设备设计、数据传感器设计，硬件测试。硬件设计需要考虑以下几个问题：抗干扰问题，GPS信号在复杂的场合，容易受到各种干扰，必须保证信号的稳定性。省电节能问题。

本设计硬件是在Altium Designer 10 上进行设计的。这是一款由Altium 公司开发的基于Windows环境下的电路板设计软件，在电子行业运用十分普遍，是电子工程师设计印刷线路板的首选工具，它具有简单易学，操作便捷，资源丰富，人机交互良好等优点。

2.2 主要器件选型

本课题是利用MIMU构建捷联惯性导航系统，然后接收卫星信号对导航结果进行补偿和融合，得到可信度高的姿态、速度和位置等信息。最终可以应用在地面车辆上，需要良好的稳定性和抗干扰能力。所以，再设计时，不仅需要对其进行规划分析，还要根据工作环境和运行条件进行芯片的选型。以下将介绍几个主要器件的选型。

2.2.1 ARM芯片

明确硬件总体需求情况选择的CPU的处理能力、存储容量及速度、IO端口的分配、接口要求、电平要求等等。综合这些因素考虑本系统采用的处理芯片为ST公司的STM32F405VGT6型号的ARM芯片，其属于STM32F4系列，是基于Cortex-M4内核的，是一款升级版的处理器，具有多方面的优势：1) 相比于Cortex M3芯片,STM32F4最大的优势就是内置单精度FPU和DSP指令，可提升控制算法的执行速度，提高代码执行效率，本设计中涉及导航数据的解算而且导航解算对实时性要求比较高，该优势可以很好地展现出来。

2) 芯片主频达到168Mhz, 片上flash可达1MB,SRAM有192KB, 有着丰富的内存资源, 满足本设计程序和数据的处理和存储要求。

3) 在实现高性能和低功耗方面更灵活。本设计的目的是实现小型化低功耗的导航模块,而STM32F40x的功耗为: 在主频168MHz情况下, 在闪存中执行coreMark基准测试程序, 功耗仅为230uA/MHz。在运行模式下, STM32F4具有更高的功耗效率, 特别适合需要一定运算量的应用场合。

4)功能强大、价格低廉, 性价比高。

STM32F405RGT6是F4系列的一款设备, 我们选用LQNF100封装, 内置有1M闪存。其通讯接口丰富, 主要接口如下:

- 4个10.5Mbit/s的USART, 加上两个UART;
- 3个42Mbit/s的SPI;
- 3个 I^2C ;
- 2个CAN;
- 1个SDIO), 满足本设计的通讯接口的需求;
- 1个全速USB OTG和一个高速USB OTG;

2.2.2 MEMS器件

本设计采用的MEMS器件是德国博世公司研发的六轴惯性传感器芯片BMI160, 该模块集成了三轴陀螺仪、三轴加速度计和温度传感器, 陀螺仪和加速度计的测量噪声分别是体积小 ($2.5mm \times 3.0 \times 0.83mm$)

BMI160的性能指标如表? 所示

2.2.3 卫星接收模块

卫星接收模块采用的是U-blox公司的NEO-M8N系列芯片。NEO-M8N是一款高性能,高灵敏度的卫星信号接收模块, 虽然其外形尺寸只有 $16 \times 12.2 \times 2.4mm$,却可以提供丰

富的卫星数据信息，模块的高灵敏度、低功耗及轻巧的体积，非常适用于车载、手机、无人机及其他移动定位系统的应用。其实物如图？所示：

U-blox8拥有22个跟踪通道，66个捕获通道，热启动下捕获卫星信号的速度不足一秒，模块速度极快，其捕获和跟踪灵敏度甚至达到-160dBm。同时，这款芯片采用创新设计和技术，能有效抑制干扰源和多路径效应，在恶劣环境下也能提供有效的导航信息。NEO-M8N的性能指标如表？所示：

NEO-M8N有多种接口方式：UART、SPI以及USB接口三种模式；数据输出格式有：NMEA、UBX和RTCM。本文将NEO-M8N与STM32芯片相连，采用最为方便利于开发的USART接口连接方式。据格式采用工程中常用的NMEA-0183协议，该协议是美国国家海洋电子协会(national marine electronics association)为海用电子设备指定的一套标准格式，目前绝大部分卫星接收机都采用NMEA-0183协议输出位置、速度、星历等导航信息

2.2.4 电源管理模块

电源是一个系统中及其重要的一部分，电源的设计直接影响到系统能否正常工作。根据设计要求，模块通过USB供电或者电池供电，为了使整个系统稳定的工作，设计通过稳压芯片输出稳定的电压。系统中需要的电压有5V 和3.3V。因此本系统对电源做了两级处理，前级电源输出5V的电压，末端设计系统需要的3.3V电压，保证了系统供电的稳定性。本系统选择德州仪器的TPS73133具有反向电流保护的超低功耗电压稳压器件，系列线性稳压器是设计用于功耗敏感类应用的超低静态电流器件。一个精密带隙和误差放大器在温度范围内的精度为2。只有1 μ A的静态电流使这些器件成为要求极小闲置状态功率耗散的电池供电常开系统的理想解决方案。为了增加安全性，这些器件还具有热关断、电流限制和反向电流保护功能。通过将使能(EN)引脚下拉至低电平可将这些稳压器置于关断模式。这个模式的关断电流低至150nA（典型值）。

为使传感器能够稳定工作，设计过程中采用隔离的供电方式，电源设计采用了两个稳压芯片，一个单独供GPS传感器供电，通过程序控制芯片使能端管脚的电平，另外一路稳压为主控芯片和其他外设供电。

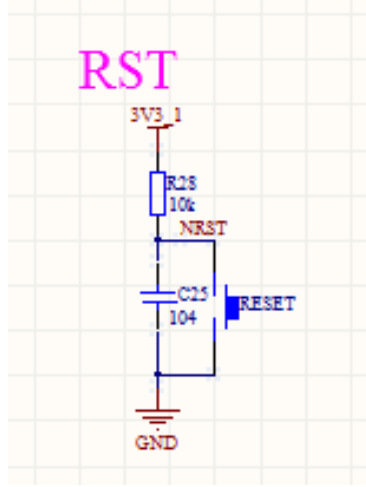


图 2.2 复位电路

表 2.1 启动模式设置接口

<i>BOOT0</i>	<i>BOOT1</i>	启动模式	说明
0	<i>X</i>	用户闪存存储器	用户闪存存储器，即从Flash启动
1	0	系统存储器	系统存储器启动，用于串口下载
1	1	SRAM启动	SRAM启动，可在SRAM中调试代码

电源时钟

2.3.2 系统PCB板设计

考虑 N_a 个动态解耦二阶非线性系统：

$$\begin{cases} \dot{x}_i = v_i \\ \dot{v}_i = f(t, x_i, v_i) + u_i \end{cases} \quad i = 1, 2, \dots, N_a, \quad (2.1)$$

其中 $x_i, v_i \in \mathbb{R}^n$ 为智能体 i 的状态， $u_i \in \mathbb{R}^n$ 为其控制输入。在特定情况下， x_i 可以表示智能体 i 的位置， v_i 表示速度。

每个智能体 i 被标记为图 $\mathcal{G} = \{\mathbb{N}_a, \mathcal{E}, \mathcal{A}\}$ 的一个顶点，其中 $\mathbb{N}_a = \{1, 2, \dots, N_a\}$ 为顶点的集合，边的集合 $\mathcal{E} \subseteq \{(i, j) \in \mathbb{N}_a \times \mathbb{N}_a \mid i \neq j\}$ 表示智能体之间的连接拓扑，相邻矩阵 \mathcal{A} 的元素 $a_{ij} \geq 0$ 被定义为 $a_{ij} > 0 \Leftrightarrow (i, j) \in \mathcal{E}$ 。假设图 \mathcal{G} 为无向图，即 $a_{ij} = a_{ji}$ 。定义智能体 i 的邻居的集合为 $\mathcal{N}_i = \{j \in \mathbb{N}_a \mid (i, j) \in \mathcal{E}\}$ ，智能体 i 的邻居的数量为 $|\mathcal{N}_i|$ 。不失一般性，本章假设智能体 i 与其邻居通过约束或成本耦合，且智能体 i 可以与其邻居通信。

假设参考轨迹由一个虚拟智能体产生：

$$\begin{cases} \dot{x}_r = v_r \\ \dot{v}_r = f(t, x_r, v_r), \end{cases} \quad (2.2)$$

其中 $x_r, v_r \in \mathbb{R}^n$ 为参考状态。在本章中，我们假设所有智能体都能提前获得参考轨迹。

定义 $z_i = [x_i^T, v_i^T]^T \in \mathbb{R}^{2n}$ 和 $z_r = [x_r^T, v_r^T]^T \in \mathbb{R}^{2n}$ 。进一步，令 $z_{-i} = [z_{j_1}^T, \dots, z_{j_{|\mathcal{N}_i|}}^T]^T$ 表示智能体 i 所用的邻居的状态构成的向量，即 $j_k \in \mathcal{N}_i$ ，且 $k = 1, 2, \dots, |\mathcal{N}_i|$ 。整个系统的控制目标为：当 $t \rightarrow \infty$ ，对于所有的 $i, j = 1, \dots, N_a$ ， $\|z_i - z_j\| \rightarrow 0$ 且 $\|z_i - z_r\| \rightarrow 0$ 。

给定权重参数 α_i 、 β_i 和 γ_i ，智能体 i 的理想成本函数定义为：

$$J_i(t_k, z_i, z_{-i}, z_r, u_i) = \int_{t_k}^{t_k+T} L_i(\tau|t_k, z_i, z_{-i}, z_r, u_i) d\tau + L_{if}(z_i(t_k + T|t_k), z_r(t_k + T)), \quad (2.3)$$

其中阶段成本

$$\begin{aligned} L_i(\tau|t_k, z_i, z_{-i}, z_r, u_i) = & \alpha_i \sum_{j \in \mathcal{N}_i} a_{ij} \|z_i(\tau|t_k) - z_j(\tau|t_k)\|^2 \\ & + \beta_i \|z_i(\tau|t_k) - z_r(\tau)\|^2 + \gamma_i \|u_i(\tau|t_k)\|^2, \end{aligned} \quad (2.4)$$

终端成本 $L_{if}(z_i(\tau|t_k), z_r(\tau))$ 为连续可微的函数。

2.4 分布式模型预测控制

和上章类似，本章假设所有智能体同步求解各自的优化问题，然而在存在耦合成本的情况下，更新智能体不可能得到其邻居的实际预测状态，因此优化问题不可解。为了解决这个问题，我们引入假设轨迹的概念。代替传递实际的预测轨迹，每个智能体计算传递各自的假设轨迹给自己的邻居。为了区别假设输入和最优输入，我们首先给出一些符号说明，如表2.2所示。

相似地，分别定义最优预测轨迹和假设预测轨迹为 $z_i^*(\tau|t_k)$ 和 $\hat{z}_i(\tau|t_k)$ 。在MPC框架下，最优控制输入的第一部分一般被作为实际控制输入，即实施 $u_i^*(\tau|t_k)$ ， $\tau \in [t_k, t_{k+1})$ ，我们可得到实际的状态 $z_i(\tau|t_k) = z_i^*(\tau|t_k)$ ， $\tau \in [t_k, t_{k+1}]$ 。

和[??]类似，假设的参考轨迹定义为：

$$\hat{u}_i(\tau|t_k) = \begin{cases} u_i^*(\tau|t_{k-1}) & \tau \in [t_k, t_{k-1} + T) \\ \kappa_i(z_i^k(\tau|t_{k-1}), z_r(\tau)) & \tau \in [t_{k-1} + T, t_k + T], \end{cases} \quad (2.5)$$

表 2.2 智能体 i 的变量

$u_i^*(\tau t_k)$	最优控制输入
$z_i^*(\tau t_k)$	最优状态
$\tilde{u}_i(\tau t_k)$	可行控制输入
$\tilde{z}_i(\tau t_k)$	可行状态
$\hat{u}_i(\tau t_k)$	假设控制输入
$\hat{z}_i(\tau t_k)$	假设状态

其中 $\kappa_i(z_i^\kappa(\tau|t_{k-1}), z_r(\tau))$ 是与终端状态 $z_i^\kappa(\tau|t_{k-1})$ 和参考状态 $z_r(\tau)$ 相关的终端控制器。而终端状态 $z_i^\kappa(\tau|t_{k-1})$ 是根据动力学方程 (2.1), 以 $z_i^\kappa(t_{k-1} + T|t_{k-1}) = z_i^*(t_{k-1} + T|t_{k-1})$ 为起始状态, 输入为 $u_i(\tau|t_k) = \kappa_i(z_i^\kappa(\tau|t_{k-1}), z_r(\tau))$ 计算得到的。相应的, 假设预测轨迹定义为:

$$\hat{z}_i(\tau|t_k) = \begin{cases} z_i^*(\tau|t_{k-1}) & \tau \in [t_k, t_{k-1} + T), \\ z_i^\kappa(\tau|t_{k-1}) & \tau \in [t_{k-1} + T, t_k + T]. \end{cases} \quad (2.6)$$

为了和 z_i 一致, 令 \hat{z}_{-i} 为智能体 i 所有邻居假设状态构成的向量。

问题2 在 t_k 时刻, 对于每个智能体 $i = 1, 2, \dots, N_a$, 给定 $z_i(t_k)$ 、 $z_r(\tau)$ 和 $\hat{z}_{-i}(\tau|t_k)$, $\tau \in [t_k, t_k + T]$,

$$J_i^*(t_k, z_i^*, \hat{z}_{-i}, z_r, u_i^*) = \min_{u_i(\tau|t_k)} J_i(t_k, z_i, \hat{z}_{-i}, z_r, u_i)$$

使得对于所有的 $\tau \in [t_k, t_k + T]$

$$z_i(t_k|t_k) = z_i(t_k), \quad (2.7)$$

$$\begin{cases} \dot{x}_i(\tau|t_k) = v_i(\tau|t_k) \\ \dot{v}_i(\tau|t_k) = f_i(\tau, x_i(\tau|t_k), v_i(\tau|t_k)) + u_i(\tau|t_k), \end{cases} \quad (2.8)$$

$$z_i(\tau|t_k) \in \mathbb{Z}_i, \quad (2.9)$$

$$u_i(\tau|t_k) \in \mathbb{U}_i, \quad (2.10)$$

$$\|z_i(\tau|t_k) - \hat{z}_i(\tau|t_k)\|^2 \leq \xi_i^2(t_k), \quad (2.11)$$

$$z_i(t_k + T|t_k) \in \mathbb{Z}_{if}(t_k), \quad (2.12)$$

其中 \mathbb{Z}_i 是凸集, \mathbb{U}_i 为包含原点的凸集, (2.11) 表示相容性约束, $\mathbb{Z}_{if}(t_k)$ 表示终端集。

2.4.1 相容性约束的设计

正如上章提到的, 相容性约束主要保证智能体的假设轨迹不能偏离实际轨迹过远。除此之外, 它在确保系统收敛性中也起到了重要作用。

令 $\sigma_i(t_k) = \max_{j \in \mathcal{N}_i} \max_{\tau \in [t_k, t_k+T]} \|\hat{z}_i(\tau|t_k) - \hat{z}_j(\tau|t_k)\|$ 和 $\iota_i(t_k) = \min_{j \in \mathcal{N}_i} \min_{\tau \in [t_k, t_{k+1}]} \|\hat{z}_i(\tau|t_k) - \hat{z}_j(\tau|t_k)\|$ 。 $\sigma_i(t_k)$ 表示时间区间 $[t_k, t_k + T]$ 内智能体 i 的假设轨迹和它的邻居的假设轨迹之间的最大偏差。相似地, $\iota_i(t_k)$ 表示时间区间 $[t_k, t_{k+1}]$ 内智能体 i 的假设轨迹和它的邻居的假设轨迹之间的最小偏差。

令 $\Delta_i(t_k) = \frac{T-\delta}{\delta \alpha_i \sum_{j \in \mathcal{N}_i} \alpha_{ij}} \sum_{j \in \mathcal{N}_i} \alpha_{ij} a_{ij} (2\sigma_j(t_k) + \sigma_i(t_k))$ 。为了保证收敛性, 相容性约束的上界需要满足下面不等式:

$$0 < \xi_i(t_k) < \frac{2\iota_i(t_k) + \Delta_i(t_k) - \sqrt{(2\iota_i(t_k) + \Delta_i(t_k))^2 - 4\iota_i^2(t_k)}}{2}. \quad (2.13)$$

注意, (2.13) 暗含 $\xi_i(t_k) < \iota_i(t_k) < \sigma_i(t_k)$ 。

注 2.1 当 $\iota_i(t_k) = 0$, $\xi_i(t_k) = 0$ 。在这种情况下, 由迭代可行性可得假设输入是优化问题的最优解。

2.4.2 终端成分的设计

文献[?]的假设1提出了针对单个系统的终端元素需要满足的三个条件。下面, 针对本章的控制目标, 我们提出了相似的条件。

定义 2.1 终端集 $\mathbb{Z}_{if}(t_k) \subseteq \mathbb{Z}_i$ 和终端控制器 $\kappa_i(\cdot)$ 应满足: 对于任意的 $z_i(t|t_k) \in \mathbb{Z}_{if}(t_k)$ 和 $\tau > t$, 应用 $u_i(\tau) = \kappa_i(z_i(\tau|t_k), z_r(\tau))$,

$$z_i(\tau|t_k) \in \mathbb{Z}_{if}(t_k), \quad (2.14)$$

$$\kappa_i(z_i(\tau|t_k), z_r(\tau)) \in \mathbb{U}_i, \quad (2.15)$$

$$\sum_{i=1}^{N_a} (\dot{L}_{if}(z_i(\tau|t_k), z_r(\tau)) + L_i(\tau|t_k, z_i, z_{-i}, z_r, u_i)) \leq 0. \quad (2.16)$$

条件 (2.14) 确保不变性; 条件 (2.15) 表示输入约束满足; 条件 (2.16) 确保局部收敛性。下面设计本章的终端集和相应的终端控制器。

- 终端成本为

$$L_{if}(z_i(\tau|t_k), z_r(\tau)) = \rho_i \|z_i(\tau|t_k) - z_r(\tau)\|^2, \quad (2.17)$$

其中 $\rho_i > 0$ 为权重系数;

- 终端集为

$$\begin{aligned} \mathbb{Z}_{if}(t_k) = \{z_i(\tau|t_k) \mid \|z_i(\tau|t_k) - z_r(\tau)\| \leq \|z_i(t_k) - z_r(t_k)\| \\ \text{且 } x_i(\tau|t_k) - x_r(\tau) = -\mu_i(v_i(\tau|t_k) - v_r(\tau))\}, \end{aligned} \quad (2.18)$$

其中 μ_i 为正常数, $\tau \geq t_k$;

- 终端控制器为

$$\begin{aligned} \kappa_i(z_i(\tau|t_k), z_r(\tau)) = -(f(x_i(\tau|t_k), v_i(\tau|t_k)) \\ - f(x_r(\tau), v_r(\tau))) - k_i(v_i(\tau|t_k) - v_r(\tau)), \end{aligned} \quad (2.19)$$

其中 k_i 为正常数。

考虑以上的终端元素, 给出以下假设来确保定义2.1中的条件满足。

假设 2.1 方程 (2.1) 和 (2.2) 中的非线性函数 $f(\cdot)$ 满足局部 *Lipschitz* 条件, 即

$$\|f(t, x, v) - f(t, y, z)\| \leq L_1 \|x - y\| + L_2 \|v - z\|. \quad (2.20)$$

假设 2.2 对于任意的 $z_i(\tau) \in \mathbb{Z}_{if}(t_k)$, 终端控制器 (2.19) 满足 $\kappa_i(z_i(\tau), z_r(\tau)) \in \mathbb{U}_i$ 。

注 2.2 本章引入这个假设是因为我们不清楚 \mathbb{U}_i 的确切形式。类似的假设可以参见文献 [?]。如果集合 $\mathbb{U}_i = \{u \mid u_{min} \leq u \leq u_{max}\}$, 那么可以离线选择合适的反馈增益 k_i 使得假设2.2满足。

定理 2.1 给定参数 a_{ij} 、 α_i 、 β_i 和 λ_i , 在假设2.1和假设2.2下, 如果 $\Lambda_{i1} + \Lambda_{i2} \leq 0$ 且 $\mu_i k_i = 1$, 其中

$$\begin{aligned} \Lambda_{i1} = \rho_i - 2\rho_i k_i + \gamma_i k_i^2 + \gamma_i k_i L_1 + 2\gamma_i k_i L_2 + \gamma_i L_2^2 \\ + \gamma_i L_1 L_2 + 2\alpha_i \sum_{j \in \mathcal{N}_i} a_{ij} + 2 \sum_{j \in \mathcal{N}_i} \alpha_j a_{ij} + \beta_i, \end{aligned} \quad (2.21)$$

$$\Lambda_{i2} = \mu_i^2 (\rho_i + \gamma_i k_i L_1 + \gamma_i L_1^2 + \gamma_i L_1 L_2 + 2\alpha_i \sum_{j \in \mathcal{N}_i} a_{ij} + 2 \sum_{j \in \mathcal{N}_i} \alpha_j a_{ij} + \beta_i), \quad (2.22)$$

那么定义2.1中的所有条件都能满足。

证明: 对于任意 $z_i(t) \in \mathbb{Z}_{if}(t_k)$, $t \geq t_k$ 且 $\tau > t$, 利用 $\kappa_i(\cdot)$ 可得

$$\begin{aligned}
 & \dot{x}_i(\tau) - \dot{x}_r(\tau) + \mu_i(\dot{v}_i(\tau) - \dot{v}_r(\tau)) \\
 &= (v_i(\tau) - v_r(\tau)) + \mu_i(f(\tau, x_i, v_i) + u_i(t) - f(\tau, x_r, v_r)) \\
 &= (v_i(\tau) - v_r(\tau)) - \mu_i k_i(v_i(\tau) - v_r(\tau)) \\
 &= 0.
 \end{aligned} \tag{2.23}$$

因此, 对于 $\tau = t^+ = \lim_{\Delta t \rightarrow 0}(t + \Delta t)$, 条件 $x_i(t) - x_r(t) = -\mu_i(v_i(\tau|t_k) - v_r(\tau))$ 满足。另外, 从假设 2.2 可得 (2.15) 满足。

下面, 我们将证明条件 (2.16) 成立。利用不等式 $\|z_i - z_j\|^2 \leq 2\|z_i - z_r\|^2 + 2\|z_j - z_r\|^2$, 如果

$$\sum_{i=1}^{N_a} (\dot{L}_{if} + 2\alpha_i \sum_{j \in \mathcal{N}_i} a_{ij} (\|z_i - z_r\|^2 + \|z_j - z_r\|^2) + \beta_i \|z_i - z_r\|^2 + \gamma_i \|u_i\|^2) \leq 0, \tag{2.24}$$

那么条件 (2.16) 一定满足。

对于任意 $i \in \mathbb{N}_a$,

$$\begin{aligned}
 & \dot{L}_{if} + 2\alpha_i \sum_{j \in \mathcal{N}_i} a_{ij}(\|z_i - z_r\|^2) + 2 \sum_{j \in \mathcal{N}_i} \alpha_j a_{ji}(\|z_i - z_r\|^2) + \beta_i \|z_i - z_r\|^2 + \gamma_i \|u_i\|^2 \\
 &= \dot{L}_{if} + 2\alpha_i \sum_{j \in \mathcal{N}_i} a_{ij}(\|z_i - z_r\|^2) + 2 \sum_{j \in \mathcal{N}_i} \alpha_j a_{ji}(\|z_i - z_r\|^2) + \beta_i \|z_i - z_r\|^2 + \gamma_i \|u_i\|^2 \\
 &= 2\rho_i(x_i - x_r)^T(v_i - v_r) + 2\rho_i(v_r - v_i)^T(f(x_i, v_i) - f(x_r, v_r) + u_i) \\
 &\quad + 2\alpha_i \sum_{j \in \mathcal{N}_i} a_{ij}(\|z_i - z_r\|^2) + 2 \sum_{j \in \mathcal{N}_i} \alpha_j a_{ji}(\|z_i - z_r\|^2) + \beta_i \|z_i - z_r\|^2 + \gamma_i \|u_i\|^2 \\
 &\leq \rho_i \|x_i - x_r\|^2 + \rho_i \|v_i - v_r\|^2 - 2\rho_i k_i \|v_i - v_r\|^2 + \gamma_i \|k_i(v_i - v_r) + f(x_i, v_i) \\
 &\quad - f(x_r, v_r)\|^2 + 2\alpha_i \sum_{j \in \mathcal{N}_i} a_{ij}(\|z_i - z_r\|^2) + 2 \sum_{j \in \mathcal{N}_i} \alpha_j a_{ji}(\|z_i - z_r\|^2) + \beta_i \|z_i - z_r\|^2 \\
 &\leq \rho_i \|x_i - x_r\|^2 + \rho_i \|v_i - v_r\|^2 - 2\rho_i k_i \|v_i - v_r\|^2 + \gamma_i k_i^2 \|v_i - v_r\|^2 \\
 &\quad + 2\gamma_i k_i L_1 \|x_i - x_r\| \|v_i - v_r\| + 2\gamma_i k_i L_2 \|v_i - v_r\|^2 + \gamma_i L_1^2 \|x_i - x_r\|^2 \\
 &\quad + \gamma_i L_2^2 \|v_i - v_r\|^2 + 2\gamma_i L_1 L_2 \|x_i - x_r\| \|v_i - v_r\| + 2\alpha_i \sum_{j \in \mathcal{N}_i} a_{ij}(\|z_i - z_r\|^2) \\
 &\quad + 2 \sum_{j \in \mathcal{N}_i} \alpha_j a_{ji}(\|z_i - z_r\|^2) + \beta_i \|z_i - z_r\|^2 \\
 &\leq (\rho_i - 2\rho_i k_i + \gamma_i k_i^2 + \gamma_i k_i L_1 + 2\gamma_i k_i L_2 + \gamma_i L_2^2 + \gamma_i L_1 L_2 \\
 &\quad + 2\alpha_i \sum_{j \in \mathcal{N}_i} a_{ij} + 2 \sum_{j \in \mathcal{N}_i} \alpha_j a_{ji} + \beta_i) \|v_i - v_r\|^2 + (\rho_i + \gamma_i k_i L_1 \\
 &\quad + \gamma_i L_1^2 + \gamma_i L_1 L_2 + 2\alpha_i \sum_{j \in \mathcal{N}_i} a_{ij} + 2 \sum_{j \in \mathcal{N}_i} \alpha_j a_{ji} + \beta_i) \|x_i - x_r\|^2 \\
 &= (\Lambda_{i1} + \Lambda_{i2}) \|v_i - v_r\|^2 \leq 0. \tag{2.25}
 \end{aligned}$$

因此, 不等式

$$\dot{L}_{if} \leq -\beta_i \|z_i - z_r\|^2 \tag{2.26}$$

成立。它意味着在终端集内, 应用终端控制器 $\kappa_i(\cdot)$, $\|z_i - z_r\|^2$ 将指数递减为 0, 即对于任意 $z_i(t) \in \mathbb{Z}_{if}(t_k)$, $t \geq t_k$ 且 $\tau > t$, $\|z_i(\tau) - z_r(\tau)\|^2 < \|z_i(t) - z_r(t)\|^2 \leq \|z_i(t_k) - z_r(t_k)\|^2$ 。根据 (2.23) 和 (2.26) 可得, 条件 (2.14) 满足。□

注 2.3 本章的终端集和终端控制器相对严格。然而, 和 [??] 中的结果相比, 本章设计的终端集可以保证不变性, 从而有助于确保迭代可行性。除此之外, 给定上述的终端元素, 本章的优化问题是一个凸优化问题, 能够有效求解。

2.4.3 分布式实现算法

在上一节，我们提出了**问题2**与给出了智能体的假设控制输入（2.5）和假设轨迹（2.6）。下面，我们将详细地描述同步分布式实现算法。

算法2

离线阶段：对于每个智能体 i ，给定权重参数 α_i 、 β_i 、 γ_i 和 φ_i ；选择合适的参数 ρ_i 、 k_i 和 μ_i 。

在线阶段：在初始时刻 t_0 ，给定初始最优解 $u_i^*(\tau|t_0)$ 和初始最优轨迹 $z_i^*(\tau|t_0)$ ， $\tau \in [t_0, t_0 + T]$ 。

(i) 在时间区间 $[t_k, t_{k+1})$ ， $k \geq 0$ ，每个智能体 i

- 实施实际轨迹 $z_i(t) = z_i^*(t|t_k)$ ， $t \in [t_k, t_{k+1})$ ；
- 计算假设控制输入 $\hat{u}_i(\tau|t_{k+1})$ 和假设轨迹 $\hat{z}_i(\tau|t_{k+1})$ ， $\tau \in [t_{k+1}, t_{k+1} + T]$ ；
- 将 $\hat{z}_i(\cdot)$ 传递给它的邻居，接收来自邻居 $j \in \mathcal{N}_i$ 的 $\hat{z}_j(\cdot)$ ；
- 计算 $\sigma_i(t_k)$ ， $\iota_i(t_k)$ ；
- 将 $\sigma_i(t_k)$ 传递给它的邻居，接收来自邻居 $j \in \mathcal{N}_i$ 的 $\sigma_j(t_k)$ ；
- 根据（2.13）计算 $\xi_i(t_k)$ 。

(ii) 在每个采样时刻 t_k ， $k > 0$ ，每个智能体 i

- 采样当前状态 $z_i(t_k)$ ；
- 求解**问题2**，得到最优控制输入 $u_i^*(\tau|t_k)$ 和最优轨迹 $z_i^*(\tau|t_k)$ ， $\tau \in [t_k, t_k + T]$ 。

(iii) 返回(i)。

文献[?]给出了一种得到初始最优解的方法。为了避免集中式MPC复杂的计算，这些文献设定初始假设轨迹 $\hat{z}_i(\tau|t_0)$ 为 $z_i(t_0)$ ， $\tau \in [t_0, t_0 + T]$ ，然后求解优化问题可得初始最优解。另外，本章假设求解最优化问题的时间可以忽略不计。

2.5 主要结果

利用上节的分布式算法，闭环系统具有下面的迭代可行性和闭环稳定性。

定理 2.2 在每个采样时刻 t_k ，每个动力学如 (2.1) 的智能体求解问题2。对于每个智能体，如果在初始时刻 t_0 存在一个初始可行解，那么问题2在任意时刻 t_k ， $k > 0$ 均是可行的。进一步地，整个多智能体系统是渐进稳定的，即当 $t \rightarrow \infty$ ，对于所有的 $i, j = 1, 2, \dots, N_a$ ， $\|z_i - z_j\| \rightarrow 0$ 且 $\|z_i - z_r\| \rightarrow 0$ 。

证明: 首先分析可行性。假设在 t_k 时刻，对智能体 i ，问题2是可行的。它的最优控制输入和最优轨迹分别为 $u_i^*(\tau|t_k)$ 和 $z_i^*(\tau|t_k)$ ， $\tau \in [t_k, t_k + T]$ 。定义 $\tilde{u}_i(\tau|t_{k+1})$ 为可行控制输入， $\tilde{z}_i(\tau|t_{k+1})$ 为相应的轨迹， $\tau \in [t_{k+1}, t_{k+1} + T]$ 。

令 $\tilde{u}_i(\tau|t_{k+1}) = \hat{u}_i(\tau|t_{k+1})$ ， $\tau \in [t_{k+1}, t_{k+1} + T]$ 。从终端集和终端控制器的定义可得，带入 $\tilde{u}_i(\cdot)$ ，问题2所有的约束都满足，即 $\hat{u}_i(\tau|t_{k+1})$ ， $\tau \in [t_{k+1}, t_{k+1} + T]$ 是 t_{k+1} 时刻的一个可行解。因此，初始可行意味着后续的所有优化问题均是可行的。

下面，我们证明闭环系统的稳定性。对于多智能体系统，定义李亚普诺夫函数为单个系统最优成本的和，即

$$J_{\Sigma}^*(t_k) = \sum_{i=1}^{N_a} J_i^*(t_k, z_i^*, \hat{z}_{-i}, u_i^*). \quad (2.27)$$

令 $\tilde{u}_i(\tau|t_{k+1}) = \hat{u}_i(\tau|t_{k+1})$ 为 t_{k+1} 时刻的可行解， $\tau \in [t_{k+1}, t_{k+1} + T]$ 。因此，

$$\begin{aligned} & J_{\Sigma}^*(t_{k+1}) - J_{\Sigma}^*(t_k) \\ & \leq \sum_{i=1}^{N_a} (J_i(t_{k+1}, \tilde{z}_i, \tilde{z}_{-i}, \tilde{u}_i) - J_i^*(t_k, z_i^*, \hat{z}_{-i}, u_i^*)) \\ & \leq \sum_{i=1}^{N_a} (\alpha_i \sum_{j \in N_i} a_{ij} \int_{t_k}^{t_{k+1}+T} (\|z_i^*(\tau|t_k) - z_j^*(\tau|t_k)\|^2 - \|z_i^*(\tau|t_k) - \hat{z}_j(\tau|t_k)\|^2) d\tau \\ & \quad - \int_{t_k}^{t_{k+1}} L_i(\tau|t_k, z_i^*, \hat{z}_j, z_r, u_i^*) d\tau + \int_{t_k+T}^{t_{k+1}+T} L_i(\tau|t_k, z_i^{\kappa}, z_j^{\kappa}, z_r, \kappa_i) d\tau \\ & \quad + L_{if}(z_i^{\kappa}(t_{k+1} + T|t_k), z_r(t_{k+1} + T)) - L_{if}(z_i^*(t_k + T|t_k), z_r(t_k + T))). \end{aligned} \quad (2.28)$$

由条件 (2.16) 可得

$$\begin{aligned}
 & J_{\Sigma}^*(t_{k+1}) - J_{\Sigma}^*(t_k) \\
 & \leq \sum_{i=1}^{N_a} (J_i(t_{k+1}, \tilde{z}_i, \tilde{z}_{-i}, \tilde{u}_i) - J_i^*(t_k, z_i^*, \hat{z}_{-i}, u_i^*)) \\
 & \leq \sum_{i=1}^{N_a} (\alpha_i \sum_{j \in \mathcal{N}_i} a_{ij} \int_{t_k+1}^{t_k+T} (\|z_i^*(\tau|t_k) - z_j^*(\tau|t_k)\|^2 - \|z_i^*(\tau|t_k) - \hat{z}_j(\tau|t_k)\|^2) d\tau \\
 & \quad - \int_{t_k}^{t_k+1} L_i(\tau|t_k, z_i^*, \hat{z}_j, z_r, u_i^*) d\tau. \tag{2.29}
 \end{aligned}$$

根据 $\|z_i(\tau|t_k) - \hat{z}_i(\tau|t_k)\| \leq \xi_i(t_k) < \sigma_i(t_k)$ 可得

$$\begin{aligned}
 & \sum_{i=1}^{N_a} (\alpha_i \sum_{j \in \mathcal{N}_i} a_{ij} \int_{t_k+1}^{t_k+T} (\|z_i^*(\tau|t_k) - z_j^*(\tau|t_k)\|^2 - \|z_i^*(\tau|t_k) - \hat{z}_j(\tau|t_k)\|^2) d\tau) \\
 & \leq \sum_{i=1}^{N_a} (\alpha_i \sum_{j \in \mathcal{N}_i} a_{ij} \int_{t_k+1}^{t_k+T} (2\|z_i^*(\tau|t_k) - \hat{z}_j(\tau|t_k)\| \|z_j^*(\tau|t_k) - \hat{z}_j(\tau|t_k)\| \\
 & \quad + \|z_j^*(\tau|t_k) - \hat{z}_j(\tau|t_k)\|^2) d\tau) \\
 & \leq \sum_{i=1}^{N_a} (\alpha_i \sum_{j \in \mathcal{N}_i} a_{ij} \int_{t_k+1}^{t_k+T} (2(\|z_i^*(\tau|t_k) - \hat{z}_i(\tau|t_k)\| + \|\hat{z}_i(\tau|t_k) - \hat{z}_j(\tau|t_k)\|) \\
 & \quad \times \|z_j^*(\tau|t_k) - \hat{z}_j(\tau|t_k)\| + \|z_j^*(\tau|t_k) - \hat{z}_j(\tau|t_k)\|^2) d\tau) \\
 & \leq \sum_{i=1}^{N_a} (\alpha_i \sum_{j \in \mathcal{N}_i} a_{ij} \int_{t_k+1}^{t_k+T} (2(\xi_i(t_k) + \sigma_i(t_k)) \times \xi_j(t_k) + \xi_j^2(t_k)) d\tau) \\
 & \leq \sum_{i=1}^{N_a} (\alpha_i \sum_{j \in \mathcal{N}_i} a_{ij} \int_{t_k+1}^{t_k+T} (2\sigma_i(t_k) \times \xi_j(t_k) + \sigma_j(t_k) \times \xi_j(t_k)) d\tau) \\
 & = \sum_{i=1}^{N_a} (\sum_{j \in \mathcal{N}_i} \alpha_j a_{ij} (T - \delta) (2\sigma_j(t_k) \times \xi_i(t_k) + \sigma_i(t_k) \times \xi_i(t_k))). \tag{2.30}
 \end{aligned}$$

另外, 由 $\xi_i(t_k) < \iota_i(t_k)$ 可得

$$\begin{aligned}
 & \int_{t_k}^{t_k+1} L_i(\tau|t_k, z_i^*, \hat{z}_j, z_r, u_i^*) d\tau \\
 & = \int_{t_k}^{t_k+1} \alpha_i \sum_{j \in \mathcal{N}_i} a_{ij} \|z_i^*(\tau|t_k) - \hat{z}_j(\tau|t_k)\|^2 + \beta_i \|z_i^*(\tau|t_k) - z_r(\tau)\|^2 + \gamma_i \|u_i^*(\tau|t_k)\|^2 d\tau \\
 & \geq \int_{t_k}^{t_k+1} \alpha_i \sum_{j \in \mathcal{N}_i} a_{ij} \|z_i^*(\tau|t_k) - \hat{z}_i(\tau|t_k) + \hat{z}_i(\tau|t_k) - \hat{z}_j(\tau|t_k)\|^2 d\tau
 \end{aligned}$$

$$\begin{aligned}
 &\geq \int_{t_k}^{t_{k+1}} \alpha_i \sum_{j \in \mathcal{N}_i} a_{ij} (\|\hat{z}_i(\tau|t_k) - \hat{z}_j(\tau|t_k)\| - \|z_i^*(\tau|t_k) - \hat{z}_i(\tau|t_k)\|)^2 d\tau \\
 &\geq \int_{t_k}^{t_{k+1}} \alpha_i \sum_{j \in \mathcal{N}_i} a_{ij} (\iota_i(t_k) - \xi_i(t_k))^2 d\tau \\
 &= \delta \alpha_i \sum_{j \in \mathcal{N}_i} a_{ij} (\iota_i(t_k) - \xi_i(t_k))^2.
 \end{aligned} \tag{2.31}$$

根据相容性约束的上界 (2.13)，我们可以得出对于所有的 $z_i \neq z_j$ ，

$$J_{\Sigma}^*(t_{k+1}) - J_{\Sigma}^*(t_k) < 0. \tag{2.32}$$

因此，整个系统是渐进稳定的。 \square

2.6 仿真结果和分析

本节考虑文献[?]中2-自由度的机械臂模型作为仿真模型。为了简化，此仿真忽略摩擦力的影响。模型的形式如 (2.1) 所示，其中 $f = [-4.646 \sin x_{i1} \ 0]^T$ 。仿真利用工具箱 ICLOCS^[?] 和 IPOPT^[?] 在 Matlab 平台上进行。

关联矩阵 \mathcal{A} 取值为

$$\mathcal{A} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

另外， $L_1 = 4.646$ ， $L_2 = 0$ 。根据定理2.1，其他参数选取如下：

$$T = 3, \quad \delta = 0.5, \quad \alpha_i = \beta_i = 0.1, \quad \gamma_i = 0.01, \quad \rho_i = 5, \quad \mu_i = 0.6, \quad k_i = 1/0.6 = 1.67$$

$$\mathbb{U}_i = \{u_i \mid -2.5 \leq u_{i1}, u_{i2} \leq 2.5\}$$

这些参数能保证 $\Lambda_{i1} + \Lambda_{i2} < 0$ 。

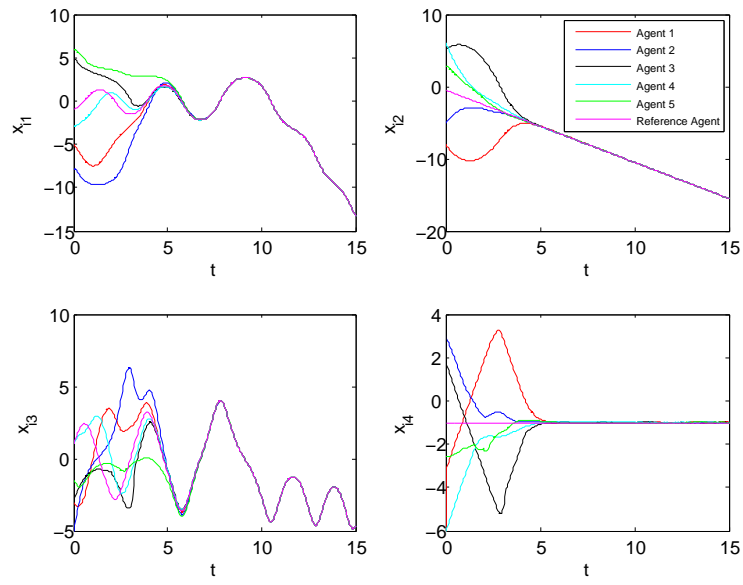


图 2.3 所有智能体的状态

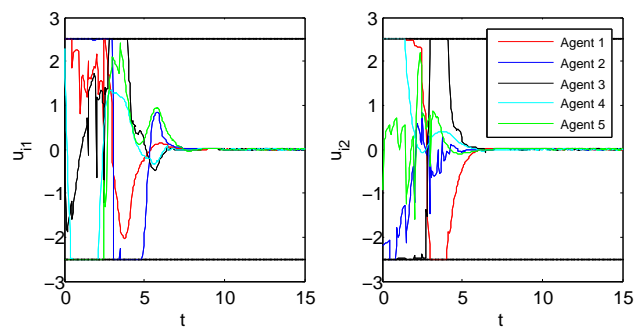


图 2.4 所有智能体的控制输入

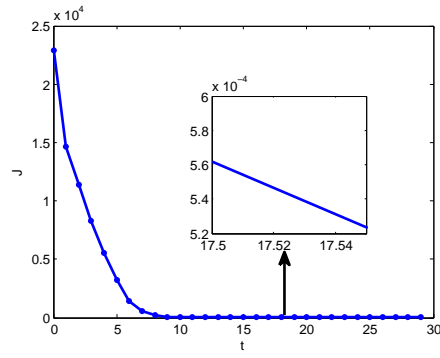


图 2.5 整个系统的成本

每个智能体的初始状态为：

$$\begin{aligned}
 z_1(0) &= \begin{bmatrix} -5 & -8 & -3 & -6 \end{bmatrix}^T, \\
 z_2(0) &= \begin{bmatrix} -8 & -5 & -5 & 3 \end{bmatrix}^T, \\
 z_3(0) &= \begin{bmatrix} 5 & 5 & -3 & 4 \end{bmatrix}^T, \\
 z_4(0) &= \begin{bmatrix} -3 & 6 & 1 & -6 \end{bmatrix}^T, \\
 z_5(0) &= \begin{bmatrix} 6 & 3 & -1.5 & -2.5 \end{bmatrix}^T,
 \end{aligned}$$

参考智能体的初始状态为 $z_r(0) = \begin{bmatrix} -1 & -0.5 & 1 & -1 \end{bmatrix}^T$ 。

仿真结果如图2.3-图2.5。图2.3表示每个智能体的状态均收敛到参考轨迹，从而实现了多智能体系统的一致。图2.4给出了每个智能体的控制输入轨迹，由图可知，控制输入满足输入约束。在图2.5中，我们可以看出整个系统的成本渐进收敛到0，因此整个系统是渐进稳定的。总的来说，仿真结果证明了所提出算法的有效性。

2.7 本章小结

本章主要研究了多个二阶非线性系统的分布式模型预测一致控制问题，智能体系统之间存在成本耦合，所有智能体共享一个参考轨迹。利用同步更新策略，在上一章的基础上，提出了新的相容性约束上界；利用参考轨迹，设计了每个智能体的终端成分，并给出了其需要满足的条件；给出了分布式优化算法；分析了优化问题的迭代可行性和整个系统的闭环收敛性；最后通过仿真实验验证了算法的有效性。