传统运控： 1、2、3、4、7、9、10、11
具身智能机器人学习总线
1. 数学与物理基础（Foundations）
1.1 线性代数（必修 • 控制 & RL 基石）
- **向量/矩阵运算**
    - 点乘、叉乘、Hadamard product
    - 范数（L1/L2/Frobenius）
    - 向量/矩阵的几何意义（投影、旋转、缩放）
    - 条件数（用于 Jacobian 稳定性与数值稳定分析）
- **矩阵分解**
  - 特征值分解（EVD）
    - 对称矩阵的对角化
    - 离散线性系统稳定性分析（谱半径 < 1）
  - 奇异值分解（SVD）
    - 降维、最小二乘稳定性
    - 多体动力学矩阵、控制器数值稳定性
  - QR / Cholesky 分解
    - QP 求解、正定矩阵分解
    - 正交矩阵的数值优势
- **正定矩阵 / 半正定矩阵**
  - SPD/PSD 的判别方法（特征值、Cholesky）
  - Mass matrix 一定是 SPD（刚体动力学基础）
  - QP cost matrix 必须 SPD 才能稳定、唯一解
- **伪逆（Moore-Penrose）**
  - 逆雅可比 IK 求解（欠驱/冗余系统）
  - 过约束系统力分配（least norm solution）
  - Damped least squares（Tikhonov 正则）
- **最小二乘（Least Squares）**
  - 正规方程（Normal Equation）
  - 加权最小二乘（WLS）
  - 在线最小二乘（递推最小二乘）
  - 应用：力分配、IK、传感器标定、状态估计
- **Jacobian（速度/力映射核心）**
  - 关节速度 → 足端线速度、角速度
  - 足端力 → 关节力矩映射（$\tau = J^T F$）
  - 雅可比奇异性检测（$\det(JJ^T)$ 接近 0）
  - Jacobian 数值计算 vs 解析计算
- **掌握标准**
  - 能在纸上算简单矩阵运算、特征值、SVD 思路；
  - 会用 Numpy / Eigen 调用 EVD / SVD / QR / Cholesky；
  - 能用伪逆 + Jacobian 实现简单机械臂/单腿 IK（Python 或 C++）；
  - 能识别 Jacobian/矩阵病态并解释对控制数值稳定性的影响。

1.2 微积分与优化（梯度 / Hessian / 优化方法）
- **微分基础**

  - 一元函数导数、极值
  - 多元偏导数、方向导数
  - 全微分与链式法则（深度网络反向传播本质）
- **梯度与 Hessian**
  - 梯度的几何意义（最速上升方向）
  - Hessian 矩阵（曲率、凸性判定）
  - 标量场的二阶近似（Taylor 二阶展开）
  - 在控制与优化中的角色：
    - 二次近似（Newton 法、LQR）
    - QP 中 H 就是 Hessian 或其近似
- **优化方法（无约束）**
  - 梯度下降（GD）
  - 随机梯度下降（SGD）
  - 动量、Adam（深度 RL 标配）
  - Newton 法 & 拟牛顿（BFGS、L-BFGS）
- **约束优化与 KKT**
  - 等式约束、拉格朗日乘子
  - 不等式约束、KKT 条件
  - 应用：
    - 有约束最优控制
    - QP / MPC / WBC 的理论基础
- **凸优化**
  - 凸集、凸函数定义
  - 强凸、Lipschitz 梯度
  - 一阶/二阶最优性条件
  - 在机器人学中的应用：QP-based 控制、MPC
- **数值积分（动力学仿真必备）**
  - Euler 前向 / 后向
  - 中点法
  - Runge-Kutta（RK2/RK4）
  - 稳定性 vs 计算开销（仿真步长的选择）
- **掌握标准**
  - 能对常见多元函数求偏导，并用链式法则手算一次简单 BP；
  - 理解梯度和 Hessian 的几何意义（最陡下降 + 曲率）；
  - 知道 GD / Adam / Newton 的更新公式、优缺点；
  - 看得懂论文里写的优化问题：min 1/2 $x^T Hx + f^T x$ s.t. $Ax \leqslant b$，知道 H 是 Hessian 或其近似；
  - 能用 Euler / RK4 在 Python 中写一个简单双积分/倒立摆仿真。

1.3 概率与统计（状态估计 & RL 噪声建模基础）
- **概率分布**
  - 一维 / 多维高斯分布
  - 指数分布、Beta、Dirichlet（可选）
  - 多元高斯协方差矩阵的意义
- **条件概率与贝叶斯**

  - 条件概率、全概率公式
  - 贝叶斯定理
  - 先验、似然、后验
- **统计量**
  - 期望、方差、协方差、相关系数
  - 样本估计、置信区间（可选）
- **参数估计**
  - 最大似然估计（MLE）
  - 最大后验估计（MAP）
  - 在传感器标定、噪声估计中的应用
- **随机过程**
  - 马尔可夫链（Markov Chain）
  - 白噪声、彩色噪声
  - 在 RL 中的 MDP 假设与状态转移
- **掌握标准**
  - 能写出 1D / 多维高斯的公式；
  - 会用 Python 生成高斯噪声、绘制分布；
  - 能用贝叶斯公式做简单推断；
  - 理解 EKF 中协方差 P 的意义（不必推完整公式）；
  - 知道 RL 中 MDP 的“马尔可夫性”假设。

1.4 数值方法（数值稳定 & 求解器）
- **线性方程组求解**
  - 直接法：LU/QR/Cholesky
  - 迭代法：CG、GMRES（大规模问题）
  - 预处理技术：雅可比、iLU、多重网格
  - __用途__：
    - 解 MPC / QP / LQR 里的线性系统
    - WBC 里求解大规模方程（可能用迭代）
  - __建议__：
    - 先把 LU/QR/Cholesky + CG 基本思路吃透；
    - GMRES、多重网格可以先知道名词，之后用到再细看
- **特征值与 SVD 数值求解**
  - 数值稳定性
  - 大规模稀疏矩阵求解
  - 随机算法，随机 SVD 用于近似计算
  - **用途：**
    - 模型降阶、特征模式分析
    - 做大规模仿真/日志降维时可能用随机 SVD
  - **建议：**
    - 小规模 SVD 用在最小二乘 / 伪逆【必修】
    - 大规模/随机 SVD 那块先当“知道有这东西”，以后需要时再查。
- **误差传播与稳定性**
  - 条件数与病态问题
  - 浮点误差对控制和动力学仿真的影响

  - 前向误差 vs 后向误差分析
  - **用途：**
    - 动力学仿真时间步太大 → 数值不稳定
    - 雅可比病态 → 足端力/位置很抖
    - RL 里面 reward/obs 量纲差异大 → 梯度爆炸/消失
- **ODE 数值求解**
  - 刚性/非刚性系统求解
  - 步长控制、误差控制
  - 微分代数方程求解（用于带约束的动力学系统）
  - **用途：**
    - 机器人动力学本质就是一堆 ODE/DAE
    - Isaac Gym / Mujoco 内部就干这个
    - 在 Python 里自己写倒立摆/摆杆仿真也会用到
  - **建议：**
    - 至少理解 显式 Euler / RK4 / 隐式方法 的区别
    - “刚性”系统 + 小步长对数值稳定性的影响，多结合你实际仿真经验写几条。
- **非线性问题数值求解**
  - 非线性方程组：牛顿-拉夫森法
  - 非线性优化：拟牛顿法（BFGS）
  - 非线性最小二乘：高斯-牛顿法、LM 算法
  - **用途：**
    - IK 数值解 → 牛顿法/LM
    - 标定问题 → 非线性最小二乘
    - MPC / 参数估计 → 非线性优化
  - **建议：**
    - 至少手推一遍 牛顿-拉夫森 的公式
    - 知道 LM 在“拟合 + 噪声”场景下更鲁棒（以后你搞传感器标定会用到）。
- **实现与性能考量**
  - 并行计算（CPU/GPU 并行化）
  - 自动微分
  - 数值精度（FP32, FP64, FP16）的选择与混合
  - **用途：**
    - Isaac Gym / PPO 训练 → GPU 并行
    - PyTorch → 自动微分
    - 以后做 ONNX / TensorRT 推理时要考虑 FP16/FP32
  - **建议：**
    - “换成 FP16 之后训练发散”
    - “自动微分比数值求导稳定多了”

1.5 力学 / 物理（Embodied 的“物理身体”基础）
- **刚体动力学**
  - 质心、转动惯量
  - Euler 方程（刚体旋转）
  - 动量、角动量守恒
- **接触动力学**

  - 接触约束（位置/速度/加速度层面的约束）
  - 冲量、碰撞模型（弹性/非弹性碰撞）
  - 摩擦模型（库仑摩擦、粘滞摩擦）
- **多体系统动力学**
  - 链式刚体系统
  - Featherstone 算法（ABA / RNEA）
  - 在仿真引擎（MuJoCo、Isaac Gym）中的实现思想

---
2. 编程与系统
2.1 Python（ML/RL 主力语言）
- Numpy / Scipy（数值运算、线性代数）
- Matplotlib / Seaborn（结果可视化）
- PyTorch（Tensor、Autograd、Optimizer、nn.Module）
- 项目结构设计（package、module、config）
- 多进程 / 多线程（数据采集 + 训练并行）

2.2 C++（机器人底层 / 实时控制）
- 现代 C++（C++17/20 特性）
- 内存管理（RAII、智能指针）
- 模板与泛型编程（Eigen、控制库常用）
- Eigen（矩阵运算、几何变换）
- CMake（构建系统）
- 实时循环（固定控制周期、计时器、中断感）

2.3 CUDA（仿真加速 / 并行计算）
- CUDA 编程模型（grid / block / warp）
- 设备内存管理（global / shared / constant）
- Kernel 设计与优化（coalesced memory）
- 基本并行模式（map / reduce / scan）

2.4 工程系统（DevOps）
- Linux 系统管理（进程、权限、脚本）
- Git（分支、合并、冲突解决）
- Docker / 容器化（环境隔离、部署）
- CI / CD 基础意识（可选）

---
3. 机器人学
3.1 运动学（Kinematics）
- DH 参数与刚体位姿表示（齐次变换）
- 前向运动学 FK（base → 足端）
- 逆运动学 IK（解析法 / 数值法）
- Jacobian（速度映射、力映射）

- 奇异性分析（行列式 / 秩 / 条件数）
- 冗余自由度分解（null space 投影）

3.2 动力学（Dynamics）
- Lagrangian 形式：L = T - V
- Newton-Euler 方程
- Mass matrix M(q) 推导与性质（SPD）
- Coriolis / Centrifugal C(q, qdot)
- Gravity g(q)
- 逆动力学（给定 q, qdot, qddot → τ）

3.3 多体系统（Multi-body）
- 关节树结构（URDF 中的 Link/Joint）
- Featherstone RNEA（O(n) 逆动力学）
- Featherstone ABA（O(n) 正动力学）
- 复合刚体惯量（CRBA）

3.4 接触动力学
- 接触约束的描述（位置/速度）
- 摩擦锥（friction cone）与线性近似（pyramid）
- 约束求解：
  - LCP（Linear Complementarity Problem）
  - QP（力优化）
- 正压力、切向力的物理意义
- 多足接触、切换顺序

3.5 机器人建模
- URDF / Xacro（结构、惯量、关节类型）
- SDF（更复杂的仿真模型）
- 碰撞模型 / 可视化模型分离
- 质量、惯量标定与估计

3.6 运动控制（Control）
- PID 控制（位置 / 速度 / 力矩）
- LQR / TVLQR（线性系统最优控制）
- MPC（Model Predictive Control）
  - 预测时域、代价函数设计
  - 约束（力、姿态、接触模式）
- Whole-Body Control（WBC）
  - 任务空间控制（Task-Space）
  - QP-based 力/加速度分配
- Impedance Control（阻抗控制）

  - 足端顺应性
  - 位置 + 力的混合控制

---
4. 传感器与状态估计
- **IMU（姿态 / 角速度 / 加速度）**
  - 陀螺/加速度计偏置
  - 姿态解算（Mahony/Madgwick/EKF）
- **Encoder（关节角 / 角速度）**
  - 绝对编码器 / 增量编码器
  - 速度估计与滤波
- **力传感器（Force/Torque）**
  - 足端力估计
  - 接触检测门限 / 滤波
- **状态估计滤波**
  - EKF / UKF（非线性状态估计）
  - ESKF（误差状态卡尔曼滤波）
  - 融合 IMU + 足端 + 编码器 → 机体速度
- **足端接触估计**
  - 基于力阈值
  - 基于速度/位置不连续
  - 基于概率模型
- **VIO / SLAM（可选）**
  - VIO（视觉惯性里程计）
  - LiDAR SLAM
  - 地图构建与姿态修正

---
5. 深度学习（DL）
- CNN/ViT（图像特征提取）
- MLP（RL 中 policy & value baseline）
- 自监督学习（contrastive, masked modeling）
- 对比学习（SimCLR, BYOL）
- 表征学习（AE/VAE）
- 模型压缩（剪枝、蒸馏、量化）
- 多模态模型（CLIP、VLM、视觉-语言模型）

---
6. 强化学习（RL）
- **理论基础**
  - MDP（S, A, P, R, γ）
  - Value function / Q-function
  - Policy gradient / Actor-Critic
- **主流算法**
  - DQN / Double DQN（离散动作）
  - PPO / A2C / A3C
  - SAC / TD3 / DDPG（连续控制）

- **Model-Based RL**
  - Learned dynamics model
  - Dreamer 系列
  - MPC + RL 混合架构
- **Imitation Learning**
  - Behavior Cloning（BC）
  - DAGGER
  - DAPG（RL + Demonstration）
- **工程强化技巧**
  - Reward shaping（足端、姿态、能耗）
  - Domain Randomization（质量、摩擦、感知噪声）
  - Curriculum Learning（速度/地形难度渐进）
  - Safety RL（约束条件、惩罚机制）

---
7. 规划（Planning）
- 栅格路径规划：Dijkstra / A* / D*
- 采样规划：RRT / RRT* / PRM
- 轨迹规划：
  - 多项式轨迹（jerk 最小）
  - spline/Bézier 曲线
  - MPC-based trajectory（带约束）
- 足端步态规划（Footstep Planning）
  - ZMP / CoM 稳定性约束
  - 四足/双足步态模式切换

---
8. 具身智能核心（Embodied Cognition）
- 世界模型（World Model）
  - 状态预测模型：p(s_{t+1} | s_t, a_t)
  - 场景表示：occupancy、SDF、NeRF（可选）
- 动作模型（Policy）
  - 从图像 / 语言 / 状态到动作
- 多模态理解（Vision/Touch）
  - 视觉 + 力觉融合
- 因果推理（Causal）
  - 因果图、干预、反事实（科研向）
- Vision-Language-Action（RT-2 / RT-X）
  - 由语言指令驱动机器人行为
  - 高层规划 + 低层控制解耦

---
9. 仿真系统
- Isaac Gym（GPU 并行仿真）
  - Actor / Env 管理
  - 状态缓冲区（root_state_tensor, dof_state）
  - 大规模并行 rollouts

- Mujoco（精确接触仿真）
  - 接触模型
  - XML 模型定义
- PyBullet（教学 / 快速验证）
- Sim2Real gap 建模
  - 感知噪声（IMU/Joints）
  - 摩擦系数偏差
  - 力矩常数、关节死区、延迟

---
10. 实际部署（Sim2Real）
- ROS1/ROS2
  - 节点、Topic、Service、Action
  - TF（坐标系管理）
- 控制周期（200~1000Hz）
  - 实时线程，调度优先级
- 力矩 / 速度控制接口
  - 驱动器通信协议（CAN/EtherCAT）
- 电机/驱动器建模
  - 转矩常数、反电动势
  - 饱和、温度限制
- 延迟补偿
  - 预测控制 / 延迟估计
- 滤波 / 参数估计
  - LPF / HPF
  - 传感器偏置估计
- 安全策略
  - 跌倒检测（roll/pitch）
  - 力矩限制、接触过载保护
  - 急停策略

---
11. 腿足式机器人（Legged）
- Gait（步态类型）
  - Trot / Pace / Bound / Walk
  - Gait pattern 与稳定性
- Phase / Duty Factor
  - 相位定义（0 - 1 周期）
  - duty factor（支撑百分比）
  - 不同速度下的 duty factor 调整
- Swing / Stance 时序
  - Swing 时间、抬脚高度、轨迹形状
  - Stance 相的力分配与稳定控制
  - 过渡边界（lift-off、touch-down）
- Raibert Foot Placement
  - 基于速度和误差的落脚点公式
  - CoM 稳定性近似

  - 结合 RL 的 foot placement reward
- Foot clearance 曲线
  - 抬脚高度参数化（多项式/Bézier）
  - 防撞 / 防拖脚
  - 随速度 / 地形自适应高度
- Turning compensation（转向补偿）
  - 外展偏移（膝外展、髋外展）
  - 角速度补偿（yaw rate）
  - 内外侧腿步幅/相位差调节
- 多接触管理
  - 接触序列设计（哪些腿在支撑）
  - 接触稳定区域（support polygon）
  - 突然失去接触时的恢复策略
- Payload-aware locomotion（质量补偿）
  - 机身质量估计
  - 载荷质量对速度/步态的限制
  - 基于查表（lookup table）的速度上限
  - 质量变化时的控制增益调整