机器人算法

Robotics Algorithms

(1) Robot Autonomy

① Robot = sensors + actuators + computing + communication + ability to move

传感器 执行器 计算 交流 能移动

② Real World

Perception Motion Contror

Estimation "Position" global map "Mission Commands" Planning

③ Robot Kinematics

1. Forward / Inverse Kinematics

- Actuators and their constraints

① Wheels : Rolling constraint ; No slippage

② Legs.

- Motion Control

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = f(\dot{\varphi}_1, \cdots, \dot{\varphi}_n, \theta, \text{geometry})$$

2. Robot Feedback Control : 通过传感器来精确控制

- Robot motion is inherently uncertain 机器人的运动内在不稳定的.

- Sensing : Camera ; Leser-range-finder ; IMU

1. Sensing To Pereception : 通过传感器去感知

①识别特征(features) ②推断(Reasoning)不确定或不完整信息

③将初步信息与模型 with 实验数据相结合.

2. Environment Modeling 环境建模

① Representation 对环境的表示 : Continous : (x, y, θ) 连续时间表示位置与相对角度
Discrete : Matrix Grid 离散时间通过矩阵块表示位置信息
Discrete Topological : Topological grid.

② Modeling 建模使用 :
1. Raw Sensor Data (laser scan, image) 高数据容量，低独特 大量信息
2. Low level features (geometry features 几何特征) 中等 保留大多有用信息
3. High level features (doors, road) 低数据容量，高度独特 保留重要信息

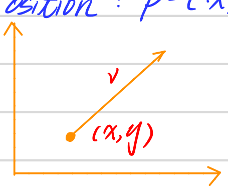
3. Kinematics :

- Forward Kinematics : Compute position of end-effector from joint parameter 从当前位置计算末端执行器位置
- Inverse Kinematics : Compute joint parameters for a given end-effector position 由末端执行器计算当前位置的参数

4. Basics

① Cartesian coordinate frame:

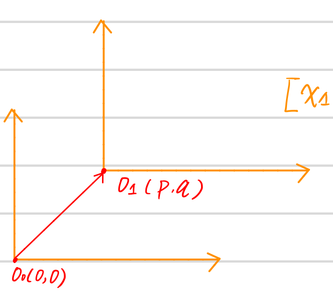
1. Position : $p=(x,y)$ 2. Velocity $v=(\dot{x},\dot{y})$ 3. Acceleration $a=(\ddot{x},\ddot{y})$



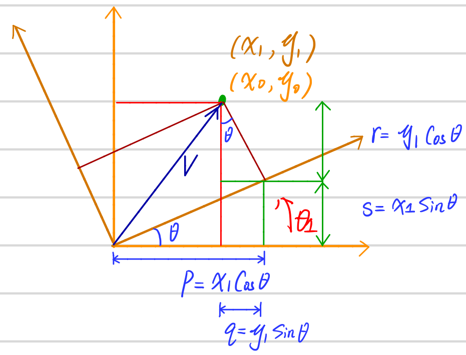
② Transformation 位移与转动

1. Translation : $V_0=(x_0,y_0)$ $V_1=(x_1,y_1)$ ${}^0O_1=(p,q)$ $V_1=(x_0+p, y_0+q)$
⇓ 原点1的座标，基于原点0.

$[x_1\ y_1] = T_0 + [x_0\ y_0]$



2. Rotation

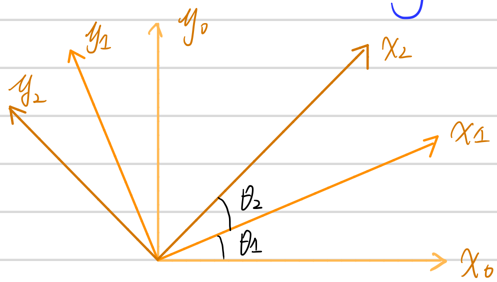


$$\left.\begin{aligned} x_0 &= x_1\cos\theta - y_1\sin\theta \\ y_0 &= y_1\cos\theta + x_1\sin\theta \end{aligned}\right\} \Rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix}$$

$$\begin{aligned} V_0 &= (x_0, y_0) \\ V_1 &= (x_1, y_1) \end{aligned} \Rightarrow [{}^1x_0, {}^1y_0] = \begin{bmatrix} \cos\theta_1 & \sin\theta_1 \\ -\sin\theta_1 & \cos\theta_1 \end{bmatrix} = R(\theta_1)^T$$

$$V_1 = R(\theta_1)^T V_0$$

3. Composition of Rotation



① From coordinate 1 rotate to coordinate 2.

$$[{}^1x_2 \quad {}^1y_2] = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 \\ \sin\theta_2 & \cos\theta_2 \end{bmatrix} = R(\theta_2)$$

② From coordinate 0 rotate to coordinate 2.

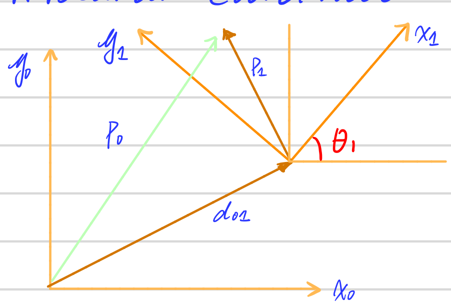
$$[{}^0x_2 \quad {}^0y_2] = [R(\theta_1){}^1x_2 \quad R(\theta_1){}^1y_2] = R(\theta_1)[{}^1x_2 \; {}^1y_2] = R(\theta_1)R(\theta_2) = R(\theta_1 + \theta_2)$$

③ Conclusion

$$[{}^ix_j \quad {}^iy_j] = R(\theta_{i+1} + \cdots + \theta_j) = R(\theta_{ij})$$

Where $\theta_{ij} = \theta_{i+1} + \cdots + \theta_j$

4. Planar Coordinate Transforms. 平面座标系的转换

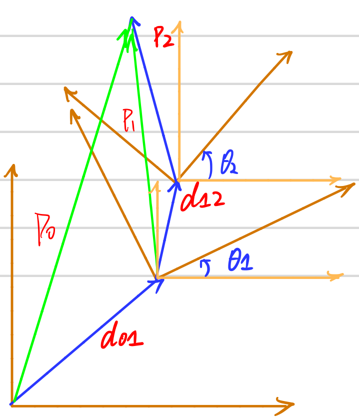


$${}^0P_0 = {}^0d_{01} + {}^0P_1$$

$${}^0P_1 = {}^0R_1 \, {}^1P_1$$

$${}^0P_0 = \underbrace{{}^0d_{01}}_{\text{初始座标系}} + \underbrace{{}^0R_1 \, {}^1P_1}_{\text{由初始座标系中 transfer 到变换座标系。}}$$

5. Planar Transform Composition



$$P_1 = d_{12} + P_2$$

$$P_0 = d_{01} + P_1$$

$${}^1P_1 = {}^1d_{12} + {}^1R_2 \, {}^2P_2$$

$$\begin{aligned} {}^0P_0 &= {}^0d_{01} + {}^0R_1 \, {}^1P_1 \\ &= {}^0d_{01} + {}^0R_1 ({}^1d_{12} + {}^1R_2 \, {}^2P_2) \\ &= {}^0d_{01} + {}^0R_1 \, {}^1d_{12} + {}^0R_2 \, R_2 \\ &= {}^0d_{01} + {}^0d_{12} + {}^0P_2 \end{aligned}$$

6. · Scaling : Alter object size 缩放

$$\begin{bmatrix}x'\\y'\end{bmatrix}=\begin{bmatrix}S_x&0\\0&S_y\end{bmatrix}\begin{bmatrix}x\\y\end{bmatrix}$$ 对x,y的缩放

· Shearing 平移

· Reflection 镜像

③ Robot Action 机器人行为

1. Locomotion 运动：环境是固定的，对环境施力来进行移动

Challenges

① Stability：(a) 接触点的形状与数量 (b) 重心 (c) 静/动态稳定 (d) 地形的倾斜.

② Characteristics of Contact：(a) 大小,形状 (b) 角度 (c) Friction

接触物的特性

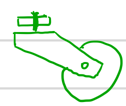
③ Modalities of Locomotion 运动方式 (1) Legs: 1. More power 2. Complex 3. Robust to terrain 鲁棒性:

(2) Wheels: 1. Efficient 2. Simple 3. Sensitive to terrain

④ Wheeled Robots (轮式机器人)

1. Standard wheel
   - 绕着车轴距转动
   - 2个自由度

2. Castor wheel 类似万向轮
   - 围绕轮角,车轴距,接触点转动
   - 2个自由度



3. Swedish wheels
   - 3个自由度

⑤ Wheel Geometry 车轮几何

2. Manipulation 操纵：机械臂是固定的，对机械臂施力来移动物体。

(2) Control

① Robot Autonomy Architecture 机器人自动化的架构

Real World

Perception

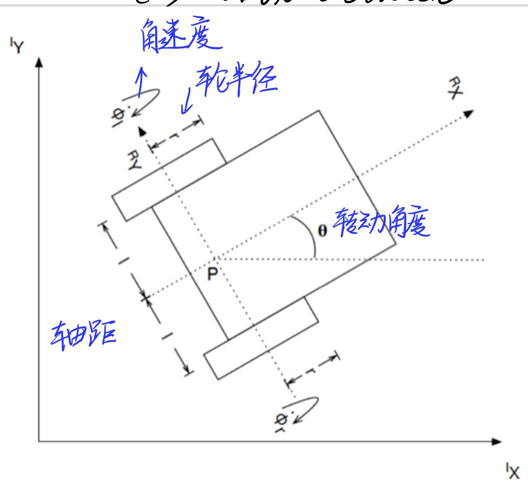
Motion Contror

Estimation → "Position" global map → "Mission Commands" Planning

② Odometry 姿态信息

(1) Robot State : (x, y, θ) 位置座标，朝向角度

(2) Robot Model : From Kinematics



$\{R_x, R_y\} \rightarrow$ Moving Frame

$\{I_x, I_y\} \Rightarrow$ Base Frame

$q = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$ robot posture in base frame.

$$^{I}\dot{\xi} = \underbrace{\begin{bmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}} . \begin{bmatrix} \frac{r.\dot{\phi}_l}{2} + \frac{r.\dot{\phi}_r}{2} \\ 0 \\ \frac{r.\dot{\phi}_r}{2.l} - \frac{r.\dot{\phi}_l}{2.l} \end{bmatrix}$$

Rotation matrix expressing the orientation of the base frame with respect to the moving frame.
旋转矩阵表示基架相对于移动架的方向

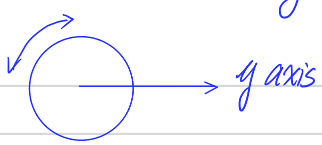
Locomotion is the process of causing an autonomous robot to move
对小车施力

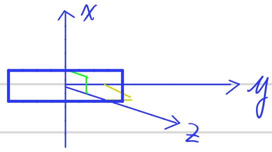
Dynamics (动力) – the study of motion in which these forces are modeled
对力做建模的学

Kinematics (运动) – study of the mathmatics of motion with condering the forces that affect the motion

(3) Wheeled Mobil Robots

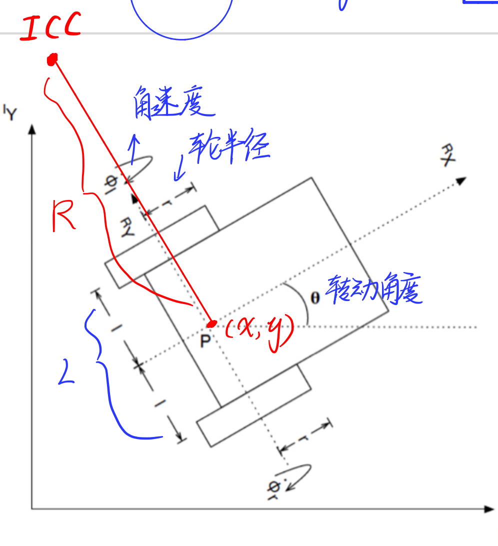
① Idealized Rolling Wheel





Wheel Parameters:

- r = wheel radians
- γ = wheel linear velocity
- w = wheel angular velocity



$\{R_x, R_y\} \rightarrow$ Moving Frame

$\{I_x, I_y\} \Rightarrow$ Base Frame

$q = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$ robot posture in base frame.

$$
{}^{I}\dot{\xi} = \underbrace{\begin{bmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}} \cdot \begin{bmatrix} \frac{r.\dot{\phi}_l}{2} + \frac{r.\dot{\phi}_r}{2} \\ 0 \\ \frac{r.\dot{\phi}_r}{2.l} - \frac{r.\dot{\phi}_l}{2.l} \end{bmatrix}
$$

Rotation matrix expressing the orientation of the base frame with respect to the moving frame.

旋转矩阵表示基架相对移动架的方向

② Two drive rolling wheels

$ICC = (x - R\sin\theta,\ y + R\cos\theta)$ Instaneous Center of Curvature 瞬时曲率中心

$V_{right}(t)$ —— linear velocity of right wheel ← Control variables

$V_{left}(t)$ —— linear velocity of left wheel

r —————— nominal radius of each wheel 车轮半径

R —————— instantaneous curvature radius of the robot trajectory 弹道, relative to the mid point 机器人瞬时曲率半径，相对于中点轴

$\Downarrow$

$R - \frac{L}{2} / R + \frac{L}{2}$ — Curvature radius of trajectory described by Left Wheel (Right Wheel)

$$
\left.\begin{aligned} W(t) &= \frac{V_r(t)}{R + \frac{L}{2}} \\ W(t) &= \frac{V_r(t)}{R + \frac{L}{2}} \end{aligned}\right\} \Rightarrow \left.\begin{aligned} W(t) &= \frac{V_r(t) - V_l(t)}{L} \\ R &= \frac{L(V_l(t) + V_r(t))}{2(V_l(t) - V_r(t))} \end{aligned}\right\} \Longrightarrow \gamma(t) = W(t)R = \frac{1}{2}(V_r(t) + V_l(t))
$$

③ Kinematic model in the robot frame

$$
\begin{bmatrix} \gamma_x(t) \\ \gamma_y(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} r/2 & r/2 \\ 0 & 0 \\ -r/L & r/L \end{bmatrix} \begin{bmatrix} W_l(t) \\ W_r(t) \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{r.\phi_l}{2} + \frac{r\phi_r}{2} \\ 0 \\ \frac{r\phi_r}{2.l} - \frac{r\phi_l}{2l} \end{bmatrix} \Leftrightarrow \begin{bmatrix} \frac{V_l}{2} + \frac{V_l}{2} \\ 0 \\ \frac{V_r}{2l} - \frac{V_l}{2l} \end{bmatrix}
$$

③ Kinematic Model in the World Frame

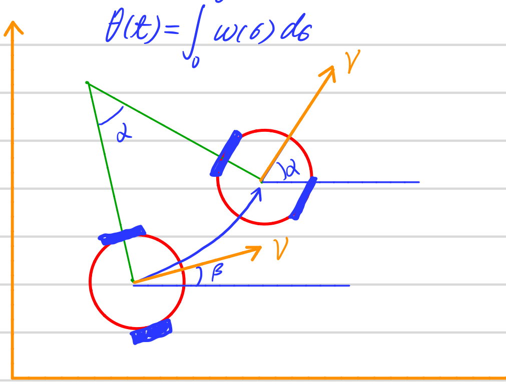
机器人向前速度向量

$v(t) = w(t) \cdot R = \frac{1}{2}(v_r(t) + v_l(t))$

$w(t) = \dfrac{v_r(t) - v_l(t)}{L}$

$$\Rightarrow \quad \dot{x}(t) = v(t)\cos\theta(t), \quad \dot{y}(t) = v(t)\sin\theta(t), \quad \dot{\theta}(t) = w(t)$$

$$\Longleftrightarrow \quad x(t) = \int_0^t v(\sigma)\cos(\theta(\sigma))\,d\sigma, \quad y(t) = \int_0^t v(\sigma)\sin(\theta(\sigma))\,d\sigma, \quad \theta(t) = \int_0^t w(\sigma)\,d\sigma$$



Conclusion:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} \cos\theta(t) & 0 \\ \sin\theta(t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ w(t) \end{bmatrix} \Longleftrightarrow \begin{bmatrix} \frac{r\cdot\phi_l}{2} + \frac{r\phi_r}{2} \\ \frac{r\phi_r}{2\cdot l} - \frac{r\phi_l}{2l} \end{bmatrix}$$

(4) Building Blocks of Control

① State: Representation of what system is doing

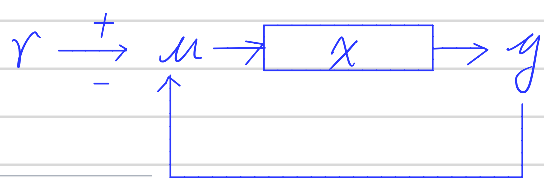
② Dynamics: Description of how the state change with time

③ Reference: What we want the system to do

④ Output: Measurement of (some aspects of) the system

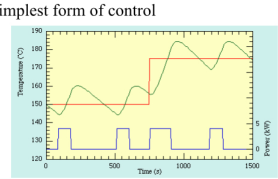
⑤ Input: Control signal

⑥ Feedback: Mapping from outputs to inputs



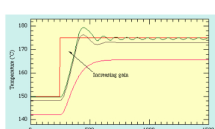
Bang-Bang Control

- Input signal either on or off
- Simplest form of control



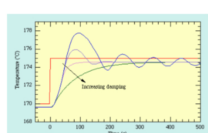
Proportional Control

- React to the difference between output and reference



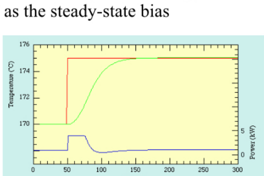
PD-Control

- React to difference plus the rate of change of input

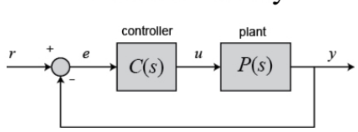


PID-Control

- React to the error, rate of change of error as well as the steady-state bias



Control Theory



- Feedback based on error (e)

$$u(t) = K_p e(t) + K_i \int e(t)dt + K_p \frac{de}{dt}$$

- Proportional: Correct for transient error
- Integral: Correct for steady-state error
- Derivative: Improve stability

(3) ROS TF

① Goal of tf

1. Encode positional association in data to facilitate logging and playback and reprocessing 编码数据中位置的关联，便于记录

② Introduction

1. Allow ROS nodes to keep track of multiple frames and relationships between the frames. 允许ROS去跟踪多个框架之间的关系.
2. Build a tree structure:
   (a) One node (frame) can have only one parent
   (b) Each node can have multiple children.

(4) Planning —— Local Planning for Obstacle Avoidance.

① Motion Planning Problem

1. Problem: Find the path in the workspace from an initial location to a goal location avoid all collisions with obstacles.
   问题：在工作空间中找从初始状态到目标，并避免障碍的路径。
2. Assumption: There exist a good map of the environment for navigation

② Motion Planning

1. Generally, we distinguish between
   - Path planning (Global) 全局的角度：路径规划
   - Obstacle avoidance (Local) 局部的角度：避障
2. Algorithm
   - Transform map into a representation useful for planning (Planner dependent)
     将图转化成可计算的形式
   - Plan a path on the transformed map
     通过算法计算路径
   - Send motion commands to controllers (Planner dependent)
     将动作控制信号送至控制器.

③ Motion Planning in Robotics

(1) State space and obstacle representation
- Workspace 工作环境
- Configuration space 默认空间

(2.) Global motion planning
- Optimal control 全局的计划是最优的解
- Deterministic graph search 确定性的图搜索
- Potential field 潜在区域
- Probabistic / randomized approaches 概率方法 与随机法

(3) Local collision avoidance 局部避免碰撞.
- BUG Algorithm
- VFH Algorithm

④ Local Path Planning

(1) Goal : Avoid collisions with local obstracles.

(5) Sensors

(1) Classification Based on robot function (What?)

基于机器人的功能

(1) Proprioceptive：本体感受：测量机器人内部的值，
电动机的速度，关节角度，电池电压

(2) Exteroceptive：外部感受：测量有关机器人环境的内容。
范围，光强度，声音幅度，使用相机检测特征.

(2) Classification based on sensor function (how)

基于传感器的功能

(1) Passive 被动：测量进入传感器的周围能量
CMOS摄像头，麦克风

(2) Active 主动：将能量释放入环境，并看环境的反应
声波测距，Encoder 编码器（驳）

General Classification (typical use)	Sensor Sensor System	Proprioceptive or Exteroceptive	Active or Passive
Tactile sensors (detection of physical contact with closeness; security switches)	Contact switches, bumpers, Optical barriers, Noncontact Non-contact proximity sensors 接触开关，保险杠， 光学屏障， 非接触式非接触式接近传感器	EC EC EC	Passive Active Active
Wheel/motor sensors	Brush encoders Optical encoders Potentiometers 刷编码器 光学编码器 电位器	PC PC PC	Passive Active Passive
Heading sensors	Compass 罗盘 Gyroscopes陀螺仪	EC PC	Passive Passive

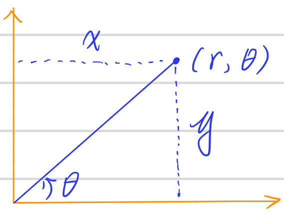
General Classification (typical use)	Sensor Sensor System	Proprioceptive or Exteroceptive	Active or Passive
Ground-based beacons	GPS Optical/RF beacons Ultrasonic beacons 全球定位系统 光学/射频信标 超声波信标	PC EC EC	Passive Active/Passive Active
Ranging	Reflectivity sensors Ultrasonic sensors Laser rangefinder 反射率传感器 超声波传感器 激光测距仪	EC EC EC	Active Active Active
Motion/Speed sensors	Doppler radar Doppler sound 多普勒雷达 多普勒声音	EC EC	Active Active
Vision-based sensors	CMOS camera Visual ranging CMOS相机 视觉范围	EC EC	Passive Passive

(6) Range Sensing

① Range Sensors

1. Laser range finders give depth information
2. Data in polar coordinate 极座标
3. Angular resolution 0.25 ~ 1 degree
4. Depth range from 5cm to 200m with resolution 10~15 mm

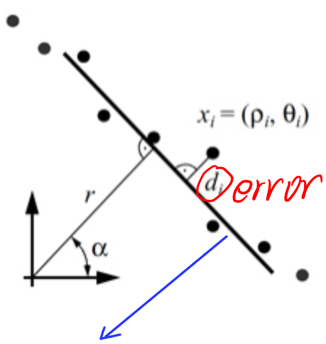
② Polar To Cartesian Coordinates



$r=\sqrt{x^2+y^2}$ $\quad x = r\cos\theta$

$\theta=\tan^{-1}\left(\frac{y}{x}\right)$ $\quad y = r\sin\theta$

③ Estimate Line Model Parameters



拟合直线

Line parameters are estimated based on an optimization criterion

eg. minimize error (least squares) $\Longrightarrow$

(1) r 为法线

(2) 任意 range point $x_i(\rho_i, \theta_i)$

$\rho\cdot\cos(\theta-\alpha)=r \Rightarrow \rho\cos(\theta-\alpha)-r=0$

(3) 对于任意点存在误差 error

$\rho_i\cos(\theta-\alpha)-r=d_i$

$\Rightarrow S=\sum_i d_i^2=\sum_i(\rho_i\cos(\theta_i-\alpha)-r)^2$

$\frac{\partial S}{\partial \alpha}=0 \quad \frac{\partial S}{\partial r}=0$ Unweighted least squares

未加权的最小二乘

④ Line Extraction Algorithm 根据 range data (Scatter) 提取线

(1) Divided measurement set into segments which potentially belong to one line.

(2) Merge lines back together if need to be.

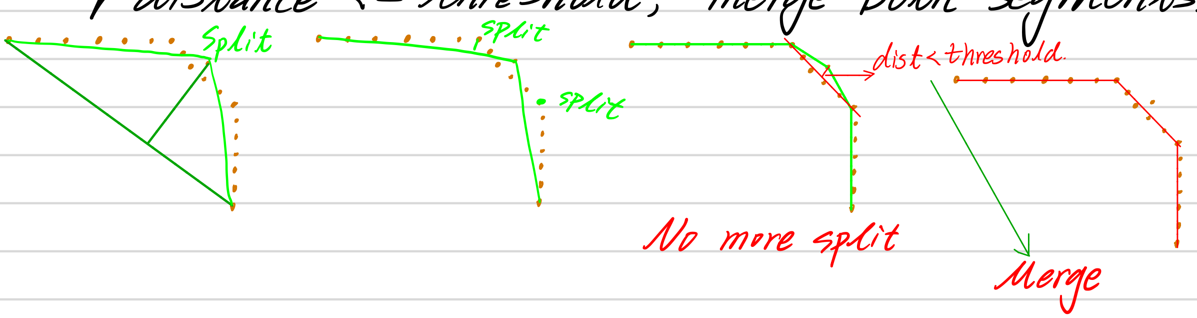
Algorithm 1: Initialise set S to contain all points

Split

- Fit a line to points in current set
- Find the most distant point to the line
- If distant > threshold ⇒ split & repeat with left and right point set

Merge

- If two consecutive segments are close / collinear enough, obtain the common line and find the most distant point
- If distance <= threshold, merge both segments.



Algorithm 2: Linear Regression

1. Initialize sliding window size Nf
2. Fit a line to every Nf consecutive points (a window)
3. Compute a line fidelity array, each is the sum of Mahalanobis distances between every 3 adjacent windows
4. Construct line segments by scanning the fidelity array for consecutive elements having values less than a threshold using an AHC algorithm
5. Merge overlapped line segments and recompute line parameters for each segments.

算法2：线性回归
I.初始化滑动窗口大小Nf
2.在每个Ny连续点Ca窗口上拟合一条线）
3.计算线保真度阵列，每个是每3个相邻窗口之间的马哈拉诺比斯距离的总和
4.通过使用AHC算法扫描具有小于阈值的连续元素的保真度阵列来构造线段
5.合并重叠的线段并重新计算每个线段的线参数。

Algorithm 3 RANSAC ⇒ Random Sample Consensus

① Applicable to any problem where goal is to identify inliers which satisfy the pre-defined model

② Typical applications: 2D data ⇒ Line ; 3D data ⇒ Plane.; feature matching

1. Sample 2 points at random
2. Calculate model parameters
3. Calculate error function for each point ⇒ $\rho\cos(\theta-\alpha)-r=d_i$
4. Select inliers and outliers
5. Repeat.