1. CCC

2024年11月23日 22:51

第一题

DH 12分，1.1 Kinematics

DH 参数8分，1.1 Kinematics

第二题

Hybrid Position Force Controller 13分，1.2 Robot control

有扰动的steady state error 考察， 7-8分，1.2 Robot control

第三题

Rolling Sliding Constraints，10分，1.3 Mobile Robot

运动学逆解或者雅可比， 10分，1.1 Kinematics

1.1 Kinematics

2024年11月23日 22:53

P18 yaw-pitch-roll(YPR),1 2 3, normal, sliding, approach
P30 rotation

$$R_1(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{pmatrix}$$

$$R_2(\phi) = \begin{pmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{pmatrix}$$

$$R_3(\phi) = \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

notice that the sign of R2 changed.

Kindly note that if we want to rotate M according to base coordinate F, we need to pre-multiply R
$P_F = RP_M$, if we rotate M according to M, post-multi
$P_M R$
If we want to rotate back, either use $R^{-1}(\varphi)$ or $R(-\varphi)$

P 40

A *4x4* **homogeneous transformation matrix**

$$T = \begin{pmatrix} \overset{rotation}{R} & \overset{translation}{p} \\ 0 & 1 \end{pmatrix}$$

Inverse Homogeneous Transformation

The inverse of T is:

$$T^{-1} = \begin{pmatrix} & R^T & & -R^T p \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Kinematic Parameters

Arm parameter	Symbol	Revolute joint (R)	Prismatic Joint (P)
Joint angle	θ	**Variable**	Fixed
Joint distance	d	Fixed	**Variable**
Link length	a	Fixed	Fixed
Link twist angle	α	Fixed	Fixed

D-H representation

Algorithm: D-H Representation

Number the joints from 1 to n starting with the base and ending with the tool yaw, pitch, and roll, in that order.

Assign a right-handed orthonormal coordinate frame L_0 to the robot base, making sure that z^0 aligns with the axis of joint 1. Set $k=1$.

1. The z^k axis aligns with the axis of joint $k+1$.
2. The intersection of z^k and z^{k-1} axes is selected as the origin of L_k. If they do not intersect, use the intersection of z^k with a common normal between z^k and z^{k-1}.
3. Select x^k to be orthogonal to both z^k and z^{k-1}. If z^k and z^{k-1} are parallel, point x^k away from z^{k-1}.
4. Select y^k to form a right-handed orthonormal coordinate frame L_k.

Set $k = k+1$. If $k < n$, go to step 1; else, continue.
Set the origin of L_n at the tool tip. Align z^n with the approach vector, y^n with the sliding vector, and x^n with the normal vector of the tool.

Next, define the four kinematic parameters (see page 43 and 44) in the following ways:

- θ_k is defined as the angle between x^{k-1} and x^k axes about the z^{k-1} axis.
- d_k is the distance between x^{k-1} and x^k axes **along** z^{k-1} axis.
- a_k is the distance between z^{k-1} and z^k axes **along** x^k axis.
- α_k is the angle between z^{k-1} and z^k axes **about** x^k axis.

…

Inverse kinematics:

tool configuration vector: as input data, need to be concise

$$w = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \end{pmatrix} = \begin{pmatrix} p \\ exp(q_n/\pi)\ r^3 \end{pmatrix}$$

in R^6 space.

for each specified robot, we can use direct kinematics to obtain the tool-config in such form:

$$w(q) = \begin{pmatrix} C_1(a_2C_2 + a_3C_{23} + a_4C_{234} - d_5S_{234}) \\ S_1(a_2C_2 + a_3C_{23} + a_4C_{234} - d_5S_{234}) \\ d_1 - a_2S_2 - a_3S_{23} - a_4S_{234} - d_5C_{234} \\ -[\exp(q_5/\pi)]C_1S_{234} \\ -[\exp(q_5/\pi)]S_1S_{234} \\ -[\exp(q_5/\pi)]C_{234} \end{pmatrix}$$

now let $w_{target}=w(q)$ and try to solve each unknown q

normally first step is $q_1=\arctan(w_2,w_1)$

the last $q_{4 \text{ or } 5}$ can be always retrieved by

$$q_5 = \pi \ln(w_4^2 + w_5^2 + w_6^2)^{1/2}$$

Trajectory Planning:

path: no time, no speed.
trajectory: need speed specification.

$$\Gamma = \{w(\lambda) \in R^6 : 0 \le \lambda \le 1\}$$
$$\lambda = s(t) \qquad 0 \le t \le T$$

t is time, s or λ is distance, ds is speed, Γ is 6-dim curve, that means at each portion of total distance, we need to know the exactly w^6

we want trajectory, so we need to determine the speed, i.e. determine each joint speed. The most intuitive thought is derivative, but it is difficult.

path: given target and intermediate points, we often use polynomial to fit the curve, since high order polynomial can approximate any shape of curve. for 2 points, use 3 order polynomial, be aware that it defines a curve in 6 dimensional space. after we have the curve, we know each point at arbitrary time.

Tool configuration Jacobian Matrix

Since we have direct form of Tool-Config x = w(q), we can get

$$\dot{x} = V(q)\dot{q}$$

x is x(t), means a direct kinematic matrix according to time, dot is the derivative of time.

q is the joint variable, angle or length, dot is the derivative of time.

V(q) is 6*n Jacobian Mat.

[v1, v2,v3,…,vn]

example in page 135

Use inverse/pseudo inverse of Jacobian

$A^{+}=(A^{T}A)^{-1}A^{T}$

thus we have

$$\dot{q} = V(q)^{-1}\dot{x} \qquad q(0) = w^{-1}(x(0))$$

where V(q) is Jacobian

1.2 Robot control

2024年11月23日 22:53

回顾：第一部分Kinematic只考虑了建系，位置变化矩阵，以及位置和时间的关系（也即速度），是为机器人运动学

接下来是机器人动力学Dynamics，会额外考虑力（动力和阻力），由此可以派生出加速度和能量。实际上电机的torque也是通过设置voltage确定的

1. 控制系统

关键见解：对于某个特殊的非线性的动力学方程，通过设计模型，来构造一个线性伺服系统。

$$m\ddot{x} + b\dot{x} + kx^3 = \alpha v + \beta$$

现在就可以去直接控制v或者x''而不是x。

我们把加速度作为受控变量

$$\ddot{\mathrm{x}} = v$$

也就有系数设计如下

$$a = m$$
$$\beta = b\dot{\mathrm{x}} + kx^3$$

根据pd控制理论，误差e应该满足：

$$\ddot{e} + k_v\dot{e} + k_p e = 0$$

标准的二阶线性微分方程特征方程为：

该方程通常来源于二阶微分方程:

$$\ddot{x}(t) + 2\zeta\omega_n\dot{x}(t) + \omega_n^2 x(t) = 0$$

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0$$

- ζ is the damping ratio
- ω_n is the underdamped natural frequency

- ω_n: 系统的**自然频率**，描述系统在没有阻尼时的固有振荡频率。
- ζ: **阻尼比**，用于描述系统阻尼的相对强弱:
  - $\zeta > 1$: 过阻尼（系统没有振荡，返回平衡点较慢）。
  - $\zeta = 1$: 临界阻尼（系统最快回到平衡点，无振荡）。
  - $0 < \zeta < 1$: 欠阻尼（系统会出现衰减振荡）。
  - $\zeta = 0$: 无阻尼（系统以固定频率振荡，无能量损耗）。

为了避免共振我们有

$$\omega_n \leq 0.5\omega_{res}$$

把特征方程代回pd控制误差有

$$s^2 + k_v s + k_p = 0$$

$$k_v = 2\zeta w_n$$
$$k_p = w_n^2$$

下一种情况，如果该系统微分方程中又引入了现实中的扰动d：

$$m\ddot{x} + b\dot{x} + kx^3 + d = u$$

$$m(\ddot{e} + k_v \dot{e} + k_p e) = d$$

Taking Laplace transformation gives:

$$\frac{E(s)}{D(s)} = \frac{1}{m(s^2 + k_v s + k_p)}$$

For a constant disturbance such as

$$D(s) = \frac{d}{s}$$

Next, $e_{ss} = \lim_{t \to \infty} e(t) = \lim_{s \to 0} sE(s)$

$$= \lim_{s \to 0} \frac{d}{m(s^2 + k_v s + k_p)}$$

The steady state error is therefore given as

时间趋向于无穷时候，稳态误差有

$$e_{ss} = \frac{d}{mk_p}$$

那么就可以人为增大k_p来减小e_{ss}

但如果想使得e_{ss}完全归零，必须使用PID而不是PD

$$v = \ddot{x}_d + k_v \dot{e} + k_p e + k_i \int e \, dt.$$

2. 动力学建模

在动力学dynamics建模中，我们使用拉格朗日法定义了动能和势能的差值

L = K - P

不使用牛顿力学是为了更简洁。

总应该满足所谓的Lagrange Equation

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} = \tau + \frac{\partial L}{\partial \theta}$$

其中L是拉格朗日标量， θ 是关节变量（角度或者伸缩长度）， τ 是力矩，分别把每个 θ 及其导数，L对其偏导都求出来带入等式。

然后我们总能够整理为如下形式

$$M(\theta)\ddot{\theta} + C\left(\theta,\ \dot{\theta}\right)\dot{\theta} + g(\theta) = \tau$$

其中M是mass matrix或者惯性矩阵
C是科里奥利力和离心力矩阵
g是重力向量
τ 是力矩向量

3. 结合1和2得到motion control
由第二部分证明了现实系统建模后是非线性的，所以我们通过第一部分知识构造一个线性闭环系统

$$M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + g(\theta) = \alpha\ v + \beta$$

Therefore,

$$\alpha = M(\theta) \qquad \beta = C(\theta,\dot{\theta})\dot{\theta} + g(\theta)$$

gives the closed-loop equation:

$$\ddot{\theta} = v$$

The servo law for the

$$v = \ddot{\theta}_d + K_v\dot{E} + K_p E$$

Now, the error equation becomes:

$$\ddot{E} + K_v\dot{E} + K_pE = 0$$

参考P32习题

回想起

$$k_v = 2\zeta w_n$$
$$k_p = w_n^2$$

并且要求

$$\omega_n \leq 0.5\omega_{res}$$

而且一般选择ξ=1 critically damped

立刻就能知道kv kp，带回到

The servo law for the

$$v = \ddot{\theta}_d + K_v\dot{E} + K_pE$$

同样的，如果加了系统外扰动的话又得用PID

4. Force control

stiffness 刚度

为了控制好接触力度，构造出力度控制模型

5. Hybrid Position/ Force control

motion control subspace:运动子空间

force control subspace：正交于运动子空间

The **basic concept** of Hybrid Position/Force control method is to decouple the position and force control problems into two sub-tasks so that motion and force controllers can be designed separately.

Position Control

The servo control law is

$$v_2 = \ddot{y}_d + k_{2v}\dot{e} + k_{2p}e$$

where $e = y_d - y$. Therefore,

$$\ddot{e} + k_{2v}\dot{e} + k_{2p}e = 0$$

Force Control

The normal force exerted on the surface is given by:

$$f = k_e(x - x_e)$$

where k_e is the surface stiffness. Since $\ddot{f} = k_e\ddot{x}$

Let $e_f = f_d - f$, the force servo controller is:

$$v_1 = \frac{1}{k_e}(\ddot{f}_d + k_{1v}\dot{e}_f + k_{1p}e_f)$$

Thus

$$\ddot{e}_f + k_{1v}\dot{e}_f + k_{1p}e_f = 0$$

The force controller requires f and $\dot{f}$, which can be calculated from the position and velocity:

$$f = k_e(x - x_e)$$

1.3 Mobile Robot

2024年11月23日 22:54

前面我们学到的都是机械臂，这一节是移动机器人

1．正向运动学建模

根据各个轮子的速度，能够得到小车整体的速度控制

$$\dot{\xi}_I = [\,\dot{x}, \dot{y}, \dot{\theta}\,]^T$$

$$\dot{\xi}_R = R(\theta)\dot{\xi}_I = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}$$

2．Constraints

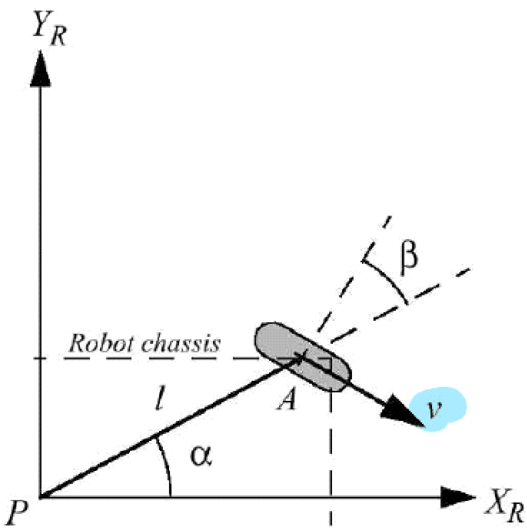
对于固定轮fixed standard

The **rolling constraint** for this wheel is,

$$v = \dot{\varphi}\; r = \dot{x}_r \sin(\alpha+\beta) - \dot{y}_r \cos(\alpha+\beta) - \dot{\theta}\; l\cos\beta$$

wheel

$$= [\sin(\alpha+\beta) \quad -\cos(\alpha+\beta) \quad -l\cos\beta] \begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta} \end{bmatrix}$$



The above *rolling constraint* can be written as:

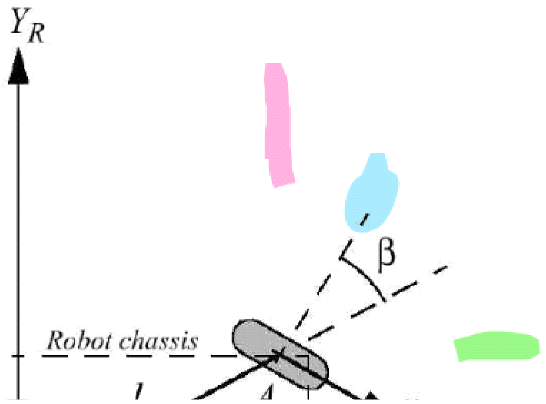
$$[\sin(\alpha+\beta) \quad -\cos(\alpha+\beta) \quad -l\cos\beta] R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0$$

or

$$\boxed{\begin{array}{l} j\; R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0 \\ \text{where } j = [\sin(\alpha+\beta) \quad -\cos(\alpha+\beta) \quad -l\cos\beta]. \end{array}}$$

The **sliding constraint** for this wheel is,

$$\dot{x}_r \cos(\alpha+\beta) + \dot{y}_r \sin(\alpha+\beta) + \dot{\theta}\; l\sin\beta = 0$$



The above *sliding constraint* can be written as:

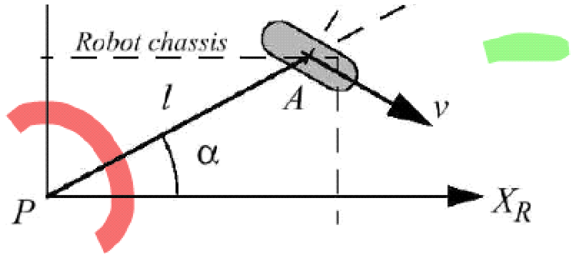
$$[\cos(\alpha+\beta) \quad \sin(\alpha+\beta) \quad l\sin\beta] R(\theta)\dot{\xi}_I = 0$$

or

$$\boxed{c\; R(\theta)\dot{\xi}_I = 0}$$

or

$$\boxed{\begin{array}{l} c\ R(\theta)\dot{\xi}_I = 0 \\ \text{where } c = [\cos(\alpha+\beta) \quad \sin(\alpha+\beta) \quad l\sin\beta]. \end{array}}$$



对于转向轮steered standard

$$[\sin(\alpha+\beta) \quad -\cos(\alpha+\beta) \quad -l\cos\beta]\, R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0$$

$$[\cos(\alpha+\beta) \quad \sin(\alpha+\beta) \quad l\sin\beta]\, R(\theta)\dot{\xi}_I = 0$$

or

$$\boxed{\begin{array}{l} j(\beta)\, R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0 \\ c(\beta)\, R(\theta)\dot{\xi}_I = 0 \end{array}}$$

对于caster wheel

For the castor wheel, the rolling constraint is identical to the standard wheel because the offset axis plays no role during motion that is *aligned with the wheel plane*:

$$[\sin(\alpha+\beta) \quad -\cos(\alpha+\beta) \quad -l\cos\beta]R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0$$

or

$$j(\beta)R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0$$

The castor geometry does, however, have significant impact on the sliding constraint.

$$[\cos(\alpha+\beta) \quad \sin(\alpha+\beta) \quad l\sin\beta]R(\theta)\dot{\xi}_I + d\dot{\beta} = 0$$

or

$$c(\beta)R(\theta)\dot{\xi}_I + d\dot{\beta} = 0$$

It can be surmised from the above equations that:

Given any robot chassis motion $\dot{\xi}_I$, there exists some value for spin speed $\dot{\varphi}$ and steering speed $\dot{\beta}$ such that the constraints are met.

现在开始考虑所有轮子对小车贡献的整体限制

3.1.4 Robot kinematic constraints

We now consider a general mobile robot with N wheels. We use the following subscripts to identify quantities relative to these 4 classes of wheels:

- f for fixed standard wheel,
- s for steerable standard wheel,
- c for castor wheels, and
- sw for Swedish wheels.

For example, the numbers of wheels of each type are denoted N_f, N_s,, N_c,, N_{sw}, φ_f, φ_s, φ_c, φ_{sw} denote the rotation angles of the wheels, and β_s, β_c denote the steering angles of the wheels.

Combining the wheel constraints imposes the overall constraints for the vehicle.

The *rolling constraints* of all wheels can now be collected in the following general expressions in matrix form:

$$J_1(\beta_s,\beta_c)R(\theta)\dot{\xi}_I + J_2\dot{\varphi} = 0$$

The sliding constraints of all standard wheels can be expressed into a single expression

$$C(\beta_s,\beta_c)R(\theta)\dot{\xi}_I + D\dot{\beta} = 0$$

For a vehicle with only standard wheels (fixed or steered), the above equation reduces to:

$$C(\beta_s)R(\theta)\dot{\xi}_I = 0$$

$\dot{\beta} = 0$

2. Hu Guoqiang

2024年11月23日 22:53

第四题

两小问，几乎都是H矩阵计算和其性质 20分

2.3 Pose Estimation

- "Unwrap" $H_n \in \mathbb{R}^{3x3}$ to a vector $h \in \mathbb{R}^8$

$$H_n = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & 1 \end{bmatrix} \Longrightarrow h = [h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8]^T$$

- Define a 2x8 matrix M_j for the j^{th} feature point

$$M_j = \begin{bmatrix} m_x^*, m_y^*, 1, 0, 0, 0, -m_x^* m_x, -m_y^* m_x \\ 0, 0, 0, m_x^*, m_y^*, 1, -m_x^* m_y, -m_y^* m_y \end{bmatrix} \text{ such that } M_j h = m_j$$

- Concatenate all N vectors m_j and matrices M_j to form matrix $M \in \mathbb{R}^{2Nx8}$

$$Mh = m$$

第五题

卡尔曼迭代计算以及传感器融合 20分

2.4 Kalman

2.1 Sensor

2024年11月23日　　22:54

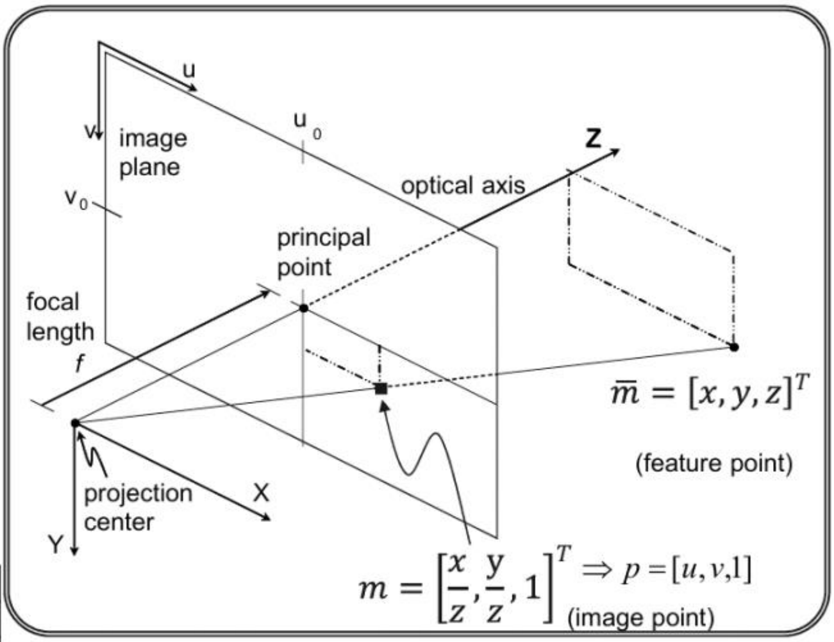
多种传感器，重点关注相机

2.2 Vision

2024年11月23日 22:55

Pinhole Camera Model

- When dealing with a digital camera, we must account for the number of pixels, shape of pixels, and size of pixel elements
- Image point m is discretized to pixel coordinates $p=[u,v,1]^T$



$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f\sigma_x & -f\sigma_x \tan\alpha & u_0 \\ 0 & f\sigma_y \sec\alpha & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{x}{z} \\ \frac{y}{z} \\ 1 \end{bmatrix}$$

$p = Am$

2.3 Pose Estimation

2024年11月23日 22:55

Structure from motion:通过多张相片进行三维重建

Pose estimation: 两个视角之间的位姿估计

从两张相片对应特征点我们构造出如下关系

$$\bar{m}_j = R\bar{m}_j^* + x,$$

用8点法（3+3+2） Essential Matrix

$$z_j^* m_j^T [x]_\times R m_j^* = 0$$

Loss of scale information

$$m_j^T E m_j^* = 0$$

holds for all points in the images, and is known as the Essential Constraint or Epipolar Constraint

带入8行9列的特征点矩阵M

$$E = \begin{bmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{bmatrix}^T \Rightarrow e = [e_1, \; e_2, \; e_3, \; e_4, \; e_5, \; e_6, \; e_7, \; e_8, \; e_9]^T$$

- Define a row vector M_j for the jth feature point

$$M_j = [m_x^* m_x, m_x^* m_y, m_x^*, m_y^* m_x, m_y^* m_y, m_y^*, m_x, m_y, 1]$$

- The Epipolar constraint can be rewritten as

$$M_j e = 0$$

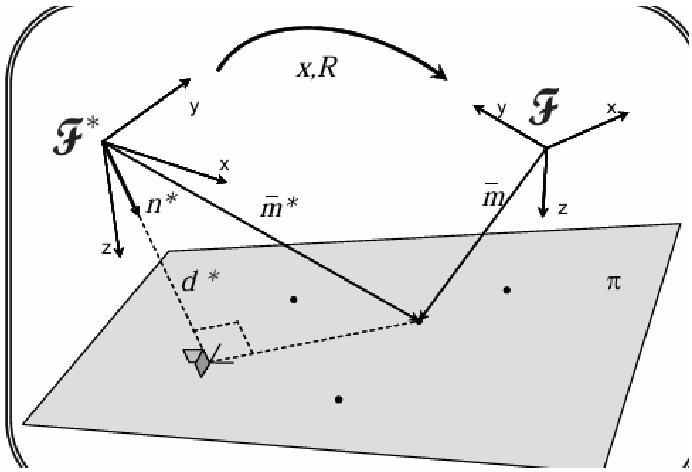
- Concatenate all N vectors M_j to form matrix $M \in \mathbb{R}^{N\mathrm{x}9}$

$$Me = 0$$

会得到一系列解 λE

(后续是通过SVD进而把x和R分离出来)
依旧是4解，根据m符号和相对深度找到唯一解

4点法 Homography的思想：
Homography 的核心是通过单个矩阵 H 建立两幅图像中平面点的几何关系，简化图像变换和对齐问题。



在F*视角中，对平面做法线n*可得距离表达式

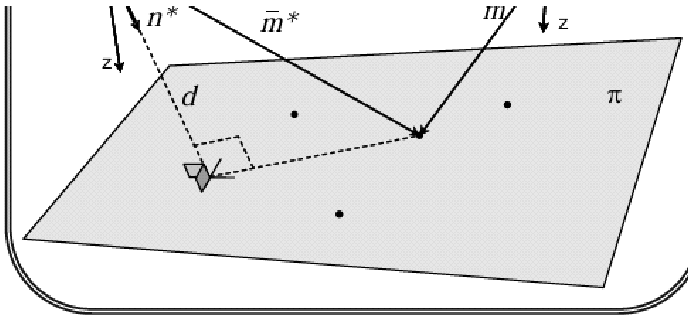
$$d^* = n^{*T}\overline{m}_j^*$$

因为处于平面上的任何m*点，与F*的向量点乘n*就是在其上的投影，都是d*

to get

Loss of scale information

$$m_j = \frac{z_j^*}{z_j}\left(R + \frac{x}{d^*}n^{*T}\right)m_j^*$$
$$= \alpha_j H m_j^*$$



- $H = R + \frac{x}{d^*}n^{*T}$ is the **Euclidean Homography Matrix** that contain relative pose information and some structure information
- $\alpha_j = \frac{z_j^*}{z_j}$ is a ratio of depths (called depth ratio)

（同样通过SVD求解）
验证距离大于0

$$n^{*T}\overline{m}_j^* = d^* > 0 \quad \Longrightarrow \quad n^{*T}m_j > 0 \ \ \forall j$$

此时有两个待选解，通过第三张照片，或
者其他传感器来辅助确定

确定好了R和x'
特征点的相对深度就得知了

$$\lambda_j = \frac{1}{m_j^T n^*}$$

H相对于E囊括了更多信息，并且E需要有一
定大小的x，而H只需要多拍一张照片

2.4 Kalman

2024年11月23日　　22:55

最基本的传感器融合思想是加权平均，但这是一阶统计量，我们想引入二阶统计Kalman

原始的系统状态建模和传感器建模

$$x(t+1) = \Phi(t)x(t) + B(t)u(t) + G(t)w(t) \qquad (2)$$

and the measurement model

$$z(t) = H(t)x(t) + v(t) \qquad (3)$$

Zero-mean Gaussian noise

这门课里w和v特指0均值高斯白噪声

所以他俩的autocorrelation是

variance

$$E\{w(t_i)w^T(t_j)\} = Q(t_i)\delta_{ij} \qquad (4)$$

$$E\{v(t_i)v^T(t_j)) = R(t_i)\delta ij \qquad (5)$$

此处看到方差和时间有关，非平稳

以下不考：

设计修正项和预测项

kalman sensor provide mesurments

correction

$$\hat{X}(t|t) = \hat{x}(t|t-1) + K(t)[z(t) - H(t)\hat{x}(t|t-1)] \quad (6)$$

$$\hat{x}(t+1|t) = \Phi(t)\hat{x}(t|t) + B(t)u(t) \quad (7)$$

prediction

那么卡尔曼系数K就能通过误差的方差P求出

$$K(t) = P(t|t-1)H^T(t)[H(t)P(t|t-1)H^T(t) + R(t)]^{-1} \quad (8)$$

where $P(t|t-1) = E\{(x(t) - \hat{x}(t|t-1))(x(t) - \hat{x}(t|t-1))^T\}$

发现P又通过K可求

$$P(t+1|t) = \Phi(t)P(t|t)\Phi^T(t) + G(t)Q(t)G^T(t) \quad (9)$$

where $P(t|t) = P(t|t-1) - K(t)H(t)P(t|t-1)$

The initial conditions for the recursion are given by $\hat{x}(0|0) = \hat{x}_0$ and $P(0|0) = P_0$.

以下考：

简化过后的向量形式

Given : $x_{k+1} = Ax_k + Bu_k + w_k$ --- (a)

and the measurement model

$z_k = H x_k + v_k$ --- (b)

我们通过kalman修正后的X有

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + K_k[z_k - H\hat{x}_k] \quad \text{---(c)}$$

指导思想就是最小化和真实值的误差
通过最小化方差P_{k+1}来达到目的

$$P_{k+1} = [A - K_k H]^2 E(\tilde{x}_k^2) + E(w_k^2) + K_k^2 E(v_k^2)$$

$$= [A - K_k H]^2 P_k + Q + K_k^2 R \qquad ---(f)$$

其中K是我们可调节的，那就让偏导等于0

And finally, $$K_k = \frac{HAP_k}{H^2 P_k + R}$$

公式带回到f

$$P_{k+1} = -\frac{(AHP_k)^2}{H^2 P_k + R} + Q + A^2 P_k$$

两个传感器融合

$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} x + \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = Hx + v$$

修正项可以用两个卡尔曼系数

$$\hat{x}_{k+1} = \hat{x}_k + K_k [z_k - H\hat{x}_k] \qquad \text{correction}$$

$$= \hat{x}_k + K_1 (z_1 - \hat{x}_k) + K_2 (z_2 - \hat{x}_k)$$

也可以用一个K，但是一个K会导致两个传

感器人为的加权平均

用两个的话能得出这样的解

$$K_1 = \frac{\sigma_2^2 P_k}{\sigma_1^2 P_k + \sigma_2^2 P_k + \sigma_1^2 \sigma_2^2}$$

$$K_2 = \frac{\sigma_1^2 P_k}{\sigma_1^2 P_k + \sigma_2^2 P_k + \sigma_1^2 \sigma_2^2}$$

2.5 Visual Servo

2024年11月23日 22:55

目的是为了控制相机到达指定位置

PBVS：
现实坐标的位姿误差

$$e_p(t) = [x(t)^T, u(t)^T\theta(t)]^T$$

速度误差

$$\dot{e}_p = \underbrace{L_p}_{\text{Jacobian}}\xi \qquad \xi(t) = [v(t)^T, \omega^T(t)]^T$$

设计一个速度反馈

$$\xi = -k_p L_p^{-1} e_p$$

得到闭环误差

$$\begin{aligned} \dot{e}_p &= L_p\xi \\ &= L_p(-k_p L_p^{-1} ep) \\ &= -k_p e_p \end{aligned}$$

是全局渐近稳定的

To prove stability, define Lyapunov function

$$V_p(e_p) = \frac{1}{2} e_p^T e_p, \\ = \frac{1}{2} \| e_p(t) \|^2$$

V_p is pos def

With time derivative

$$\dot{V}_p = e_p^T \dot{e}_p \\ = e_p^T(-k_p e_p) \\ = -k_p \| e_p \|^2$$

$\dot{V}_p$ is neg def

Negative definite Lyapunov function means the cor
Globally Asymptotically Stable and $e_p \to 0$ as $t \to \infty$

这里因为我们要通过相机估计当前位姿再求误差，所以先用到第四节传感器融合得到较为准确的特征点，然后第三节的E矩阵H矩阵得到位姿，也用到了第二节的相机投影。

PBVS缺点就是特征点可能跑到视野外

IBVS：
类似方法有针对于图像点的误差反馈设计

$$\dot{e}_i = L_i \xi$$

$$\xi(t) = [v(t)^T, \omega^T(t)]^T$$

使得

$$\xi = -k_i L_i^{-1} e_i$$

$$\begin{aligned} \dot{e}_i &= -k_i L_i \xi \\ &= -k_i L_i L_i^{-1} e_i \\ &= -k_i e_i \end{aligned}$$

局部渐近稳定，可能会在求矩阵逆时候得到奇异点。