روباتیک ۸

* منابع :

1. Introduction To Robotics, Mechanics, and Control (2nd edition)
By: John J. Craig
Addison Wesley, 1989

2. Introduction To Autonomous Mobile Robots
By: Siegwart & Nourbakhsh
MIT Press, 2004

* ارزیابی :
- امتحان میان ترم ٪ ۲۰
- امتحان پایان ترم ٪ ۲۰
- تمارین
- ارائه
- پروژه

* مدرس :
- کورش مشکی
- ۰۹۱۲-۲۱۱۹۶۰۴
- Kourosh.Meshgi@gmail.com

Subject:

Year Month Date ( )

* اطلاعات کلیدی درباره روبات

- سیستم هوشمند
- دستگاه مکانیکی ← درجات آزادی ، قدرت جابجایی ، روبات های صنعتی / روبات های متحرک
- کار خاص
- قدرت حس کردن
- قدرت عمل کردن
- قدرت پردازش

* مکانیک روبات

- سینماتیک مستقیم ، غیر مستقیم ← مستقیم : زاویه تعیین می شود ← مکان ابزار مورد دار است
  ← معکوس : مکان ابزار تعیین می شود ← زاویه مورد دار است
- دینامیک
- تولید مسیر
- کنترل

* بینایی ← ۹۰٪ اطلاعات ورودی
  ← تشخیص چهره ، احساس ، شناسایی چهره ، مکان یابی ، ژست

* شنوایی ← جهت صدای گوینده ، تشخیص گفتار ، تصدیق هویت
  ← نویز

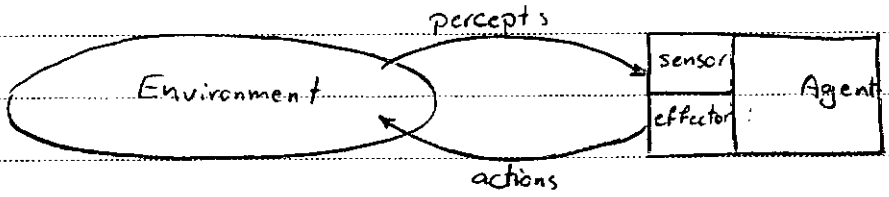
* انواع روبات

۱- متحرک (Mobile)

۲- صنعتی (Manipulator) ← بازوی روبات

* روبات‌ها

• عامل



• تفاوت روبات با عامل: روبات‌ها وجود سخت‌افزاری دارند.

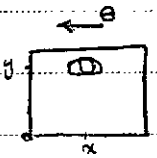
• روبات‌های هوشمند: عمل یک موجود هوشمند را تقلید می‌کنند.

* درجه آزادی

Degree of Freedom (DoF)

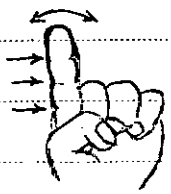
• تعداد پارامترهای لازم برای تعیین موقعیت روبات در محیط

☑ ربات در صفحه



3 درجه (x, y, θ)

☑ انگشت



4 درجه $(\theta_1, \theta_2, \theta_3, \varphi)$

* روبات صنعتی

• Base ← پایه بسیار محکم ← در برابر نیروهای وارد بر روبات مقاومت می‌کند

← محل آن می‌چرخد

• Joint ← محل اتصال بازوها

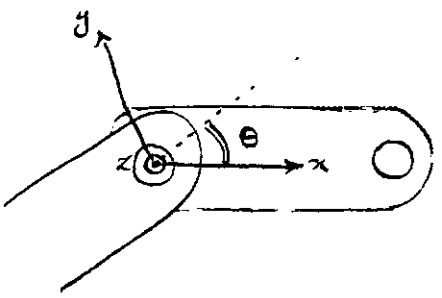
• Link ← بندهای بازو

• برای 6 درجه آزادی ← 3 تا مختصات + 3 تا جهت ابزار

Subject :
Year . . Month . . Date . . .

* سینماتیک ربات‌ها : محورهای مختصات

• مفاصل دورانی



— زاویه = اختلاف زاویه بازوهای پشت و جلو
— مبدأ مختصات = مفصل بازو
— Z همیشه محور مفصل دوران

• آدرس دهی

— محور مختصات جهانی ← آدرس نسبت به یک مرجع ثابت
← مستقل از ربات

• محاسبات

— محور مختصات مرجع OXYZ

— محور مختصات متصل به جسم O'UVW

— آدرس نقطه‌ای مانند P مفروض است ← تبدیل آدرس جسم در دستگاه دوم به دستگاه اول
تبدیل آدرس جسم در دستگاه اول به دستگاه دوم

— مختصات نقطه
$P_{XYZ} = [P_x, P_y, P_z]$
$P_{UVW} = [P_u, P_v, P_w]$

— اگر در دو دستگاه منطبق باشند
$P_{XYZ} = P_{UVW}$
$P_x = P_u$ ، $P_y = P_v$ ، $P_z = P_w$

- حالتی که دستگاه‌ها با دوران دیگری بسته بیاید

$\vec{P}_{xyz} = \vec{R} \cdot \vec{P}_{uvw}$

ضرب داخلی

R : ماتریس دوران

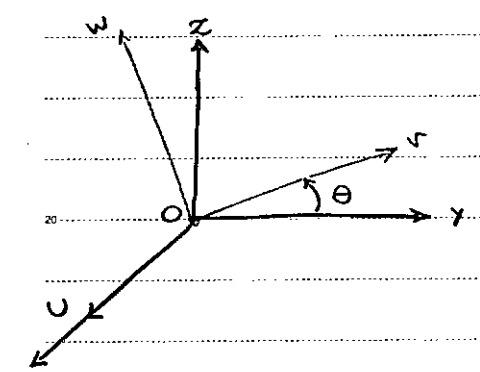
$P = P_u \cdot i_u + P_v \cdot j_v + P_w \cdot k_w$

$P_x = i_x \cdot P = i_x \cdot (\sim) = i_x \cdot i_u \cdot P_u + i_x \cdot j_v \cdot P_v + i_x \cdot k_w \cdot P_w$

$$R = \begin{bmatrix} i_x \cdot i_u & i_x \cdot j_v & i_x \cdot k_w \\ j_y \cdot i_u & j_y \cdot j_v & j_y \cdot k_w \\ k_z \cdot i_u & k_z \cdot j_v & k_z \cdot k_w \end{bmatrix}$$

← دوران حول محور X

* OX و OU منطبق باقی می مانند
* میزان دوران برابر است با θ
* علامت دوران طبق قانون دست راست با شروع از OX

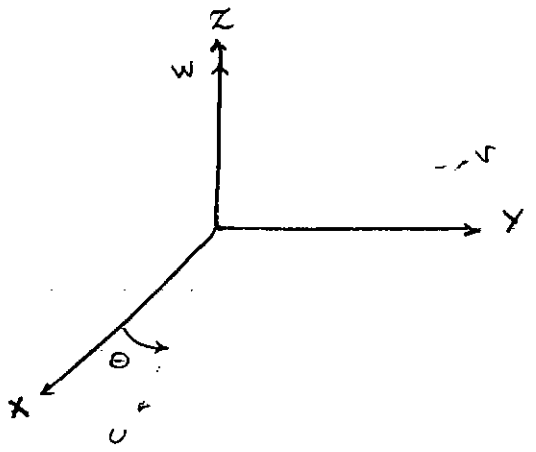


$$Rot(x,\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \rightarrow \cos\left(\frac{\pi}{2}+\theta\right)$$

Subject:
Year ... Month ... Date ...

← دوران حول محور Z

* زاویه دوران θ : OW ~ OZ



$$Rot(Z,\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

← دوران حول محور Y

* زاویه دوران θ : OU ~ OY

$$Rot(Y,\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

← تحقیقات نسبت به مبدا را داریم ، تحقیقات نسبت به مرجع دوم را می خواهیم .

$$\vec{P}_{XYZ} = \vec{R} \cdot \vec{P}_{UVW}$$
$$\vec{P}_{UVW} = \vec{R}^{-1} \cdot \vec{P}_{XYZ}$$

$R^{-1} = R^T$

← دوران ترکیبی

* اول حول X دوران می دهیم و بعد نسبت به Y و بعد به Z

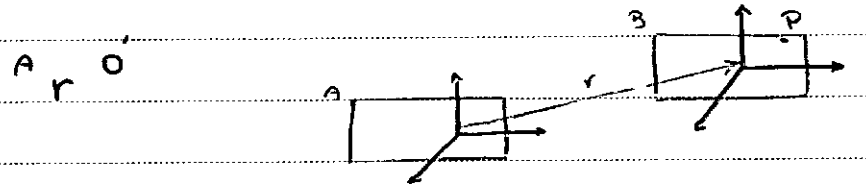
* دوران نسبت به مرجع OXYZ قبل از بقیه ماتریس‌ها ظاهر می‌گیرد و دوران ضرب می‌شود.

* دوران نسبت به مرجع OUVW بعد از بقیه ماتریس‌ها ظاهر می‌گیرد و دوران ضرب می‌شود.

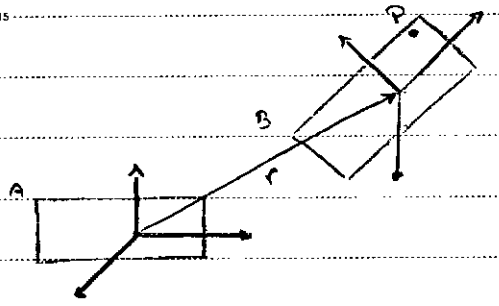
ϕ about OY → θ about OW → α about OU = $Rot(Y,\phi) \times I \times Rot(W,\theta) \times Rot(U,\alpha)$ ☒

← جابجایی

* بردار جابجایی: میزان جابجایی مبدأ O' نسبت به مبدأ O (جسم A)



← حالت کلی



$$ {}^{A}r^{P} = {}^{A}R_{B} \cdot {}^{B}r^{P} + {}^{A}r^{O'} $$

* نقطه انتقال: ${}^{A}R_{B} = 1$

* نقطه دوران: ${}^{A}r^{O'} = 0$

* نمایش پیمان: تبدیل رابطه شامل ضرب برجع به ماتریس تبدیل

$$ \begin{bmatrix} {}^{A}r^{P} \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{A}R_{B} & {}^{A}r^{O'} \\ 0_{1\times3} & 1 \end{bmatrix} \begin{bmatrix} {}^{B}r^{P} \\ 1 \end{bmatrix} $$

Subject:
Year. Month. Date.

$$ {}^{A}r^{P} = {}^{A}T_{B} \times {}^{B}r^{P} $$

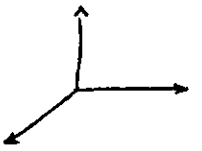
$$ \begin{bmatrix} x_A \\ y_A \\ z_A \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{A}R_{B} & {}^{A}r_{O'} \\ 0_{1\times3} & 1 \end{bmatrix} \cdot \begin{bmatrix} x_B \\ y_B \\ z_B \\ 1 \end{bmatrix} \Rightarrow {}^{A}T_{B} = \begin{bmatrix} R_{3\times3} & P_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix} $$

. قانون کلی در حالت کلی سه عضو است .

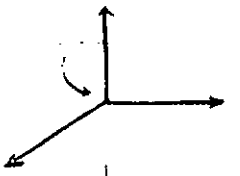
Rot α about OX → Trans a along OX → Tra. d along OZ → Rot θ about OZ

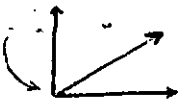
$$ T = T_{z,\theta}\ T_{z,d}\ T_{x,a}\ T_{x,\alpha}\ I_{4\times4} $$

. تا وقتی مبدا جهان حرکتی قبل است ، پیش ضرب می کنیم



Rot x I





Rot x I x T

Subject:

Year. Month. Date. 5

* سینماتیک: مستقیم و معکوس

Link Space ⟵ مستقیم ⟶ Tool Space

متغیر n

$(\theta_1, \dots, \theta_n)$

↙ زاویه‌ی مفصل

معکوس

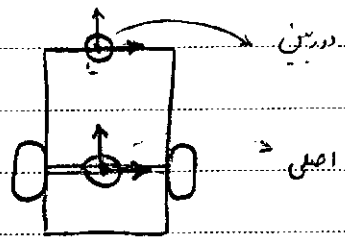
متغیر 6

$(x, y, z, \theta_x, \theta_y, \theta_z)$

• انتخاب مبدأ مختصات

— آسان شدن محاسبات

— یک اصلی و چند فرعی وجود دارد.



• متغیرهای فضای مفصل

— θ : دورانی

— d : حرکت رفت و برگشتی

• سینماتیک معکوس جواب یکتا ندارد. بلکه مجموعه‌ای از جواب‌ها مورد نظر است که بر اساس سایر پارامترها و عوامل نظیر انرژی مصرفی، زمان، ... یکی از آن‌ها استفاده می‌شود.

6 var ⟶ n var

✎ تحقیق:

تحت سینماتیک مستقیم و معکوس برای یکی از ربات‌های زیر را مطالعه و بررسی نمایید:

1. AIBO
2. Humanoid

• درجات آزادی

• راه حل‌های مورد استفاده برای حل معادلات

• مبدأهای مختصات

• فاکتورهای داخل در سینماتیک معکوس

• هدف از استفاده از ربات چیست؟

Subject :
Year Month Date

* سینماتیک بازوی روبات

- زنجیره‌ای از اجسام صلب متصل به هم
- مفصل
  - مفصل چرخشی

Revolute Joint

1 Dof ⟶ θ

  - مفصل رفت و برگشتی

Prismatic Joint

1 Dof ⟶ d

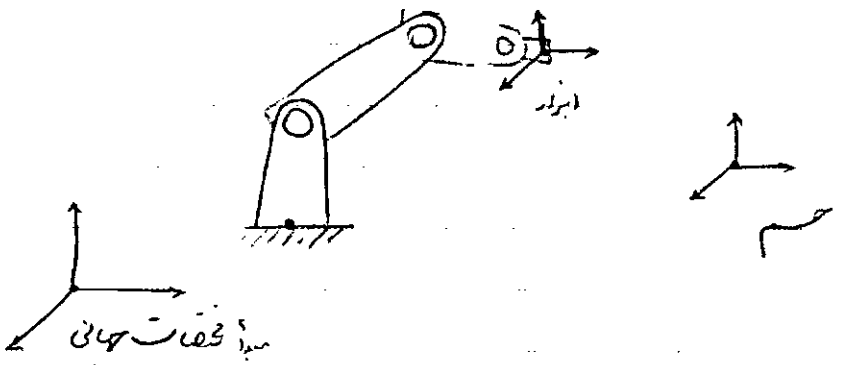
  - مفصل کروی

- Spherical

3 Dof ⟶ $\theta_1, \theta_2, \theta_3$

برای مختصات ، این مفصل به سه مفصل که در یک نقطه هستند تبدیل می‌شود.

- بازوی PUMA



- ویژگیهای Link
  - محور چرخش ← مفصلهای چرخشی
  - اندازه لینک ← اندازه خط عمود بین دو محور چرخشی که در دو انتهای لینک قرار دارد
    - a : برای i ام
  - زاویه بین ← میزانی که باید دو لینک متصل را چرخاند تا پرداشت‌های دو مفصل با هم منطبق شوند
    - α : i-1 ام را حول محور i ام تا به i ام برسد.

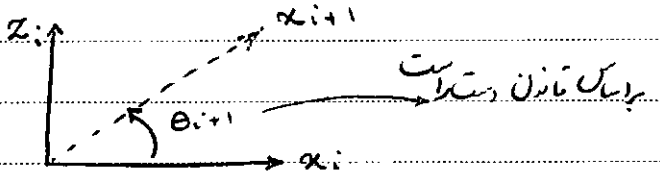
• ژوئیه‌ی Joint :
— جابجایی ← d : ارتفاع مفصل
— زاویه چرخش ← θ : زاویه بین راستای ۱ و لینک مفصل
تنها پارامتر متغیر

• پارامترهای متغیر ← مفصل چرخشی ← θ
مفصل رفت و برگشتی ← d

• در استقرار مختصات بین ۲ مفصل چرخشی ← همواره z را در راستای چرخش و نقاری گیرنده
همواره x را در راستای a قرار می‌گیرد.
y از قانون دست راست بدست می‌آید.

• قرار دادن محور مختصات در مفاصل ← بدون پیچیدگی محاسبات است.
در مفصل‌های دارای پیچش محل تلاقی محور z و a
↳ خطی بیتاست

• محاسبات ← a : فاصله در امتداد راستای x مفصل قبل
d : فاصله در امتداد راستای z مفصل قبل
θ : چرخش در دستگاه مختصات حول محور z مفصل قبل

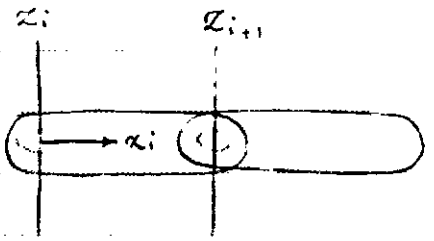


Subject :
Year . Month . Date . :)

* روش D-H برای انجام محاسبات

- a_i

فاصله z_i و z_{i+1} در راستای x_i



- α_i

زاویه بین z_i و z_{i+1} حول محور x_i

- d_i

فاصله بین x_{i-1} و x_i ی محور z_i

- θ_i

زاویه بین x_{i-1} و x_i حول محور z_i

$$T_{i}^{i+1} = T_{z,\theta} \times T_{z,d} \times T_{x,a} \times T_{x,\alpha} \times I_{4\times4}$$

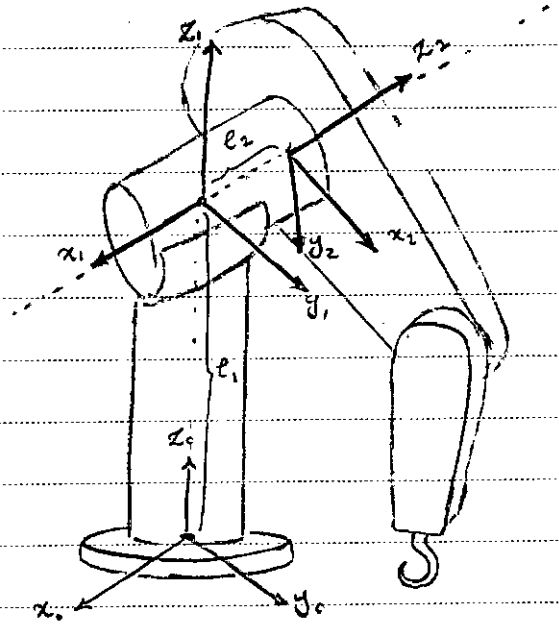
$$= \begin{bmatrix} c\theta_i & -s\theta_i . c\alpha_i & s\theta_i . s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i . c\alpha_i & -c\theta_i . c\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

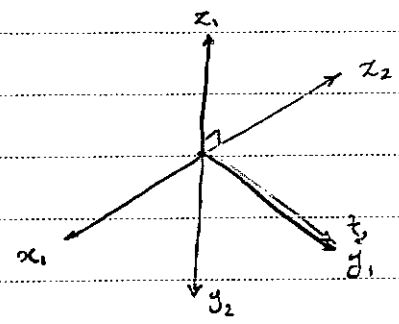
- در لینک‌های لولایی سرآل مفصل مبنا (لینک اول و آخر)، مقادیر a و α را برابر صفر قرار می‌دهیم.
- به ازای هر مفصل، چهارتایی $<\alpha, a, d, \theta>$ را تشکیل داده و طبق D-H، ماتریس انتقال هر مفصل بدست می‌آید.

Subject:

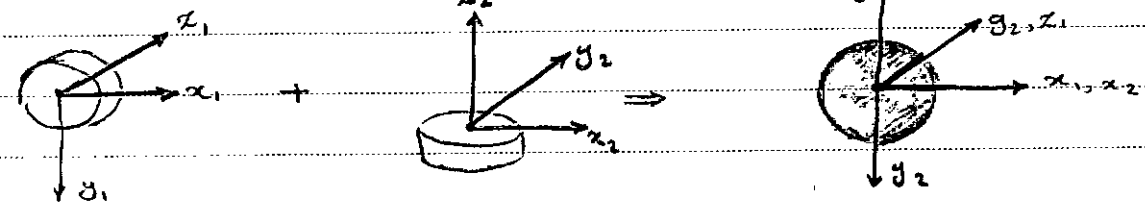
Year. Month. Date. (7)

PUMA 560 بازوی ☑





مفصل کروی ☑



فقط نسبت به هم دوران دارند.

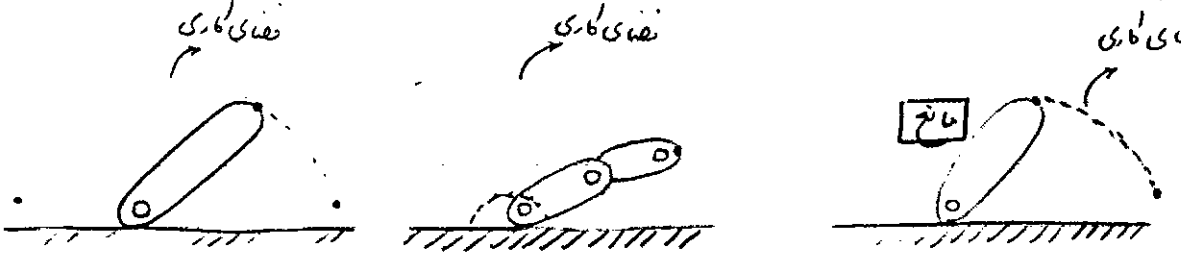
Subject :
Year . Month . Date .

Inverse Kinematic Problem (IKP)

* سینماتیک معکوس

- مشکلات
  - غیر خطی
  - عدم وجود تناظر یک به یک (چند جواب) ← انتخاب براساس هزینه و زمان
  - وجود موانع

* فضای کاری
- مجموعه از نقاط است که توسط End-Effector قابل دسترسی است.



- انواع فضای کاری
  - Reacible — فضای کاری که حداقل از یک جهت می‌توان به آن دست یافت
  - Dextrous — دست از هر جهت بتوان به آن دست یابد

* راه حل
- در روابط معادله closed form مطلوبست. یعنی θ را برحسب سایر پارامترها بدست بیاید:

$$\theta_1 = f(\alpha_i, d_i, a_i, \theta_2, \theta_3, \dots)$$

- اگر closed form وجود نداشته باشد، راه حل عددی وجود دارد.

تکرار ⟹ راه حل عددی
زمان ⟹ تکرار

کاربرد بالا، یک α راه حل عددی

- راه حل از نوع !!!

Subject:

Year. Month. Date. (8)

* روش جبری برای حل سینماتیک معکوس

. حالت دوبعدی برای حالت سه بعدی

نقطه : (x, y, φ) ⟶ زوایا ؟

$${}^{0}T_3 = {}^{0}T_1 \times {}^{1}T_2 \times {}^{2}T_3$$

$${}^{0}T_3 = \begin{bmatrix} c_{123} & -s_{123} & 0 & l_1 c_1 + l_2 c_{12} \\ s_{123} & c_{123} & 0 & l_1 s_1 + l_2 s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

برابرند

$$\text{ابزار} : \begin{bmatrix} c_\varphi & -s_\varphi & 0 & x \\ s_\varphi & c_\varphi & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\Rightarrow \begin{cases} \cos\varphi = \cos(\theta_1+\theta_2+\theta_3) \\ \sin\varphi = \sin(\theta_1+\theta_2+\theta_3) \\ x_1 = l_1\cos\theta_1 + l_2\cos(\theta_1+\theta_2) \\ y_1 = l_1\sin\theta_1 + l_2\sin(\theta_1+\theta_2) \end{cases}$$

① $\Rightarrow x_1^2 + y_1^2 = l_1^2 + l_2^2 + 2l_1l_2\cos\theta_2 \Rightarrow \theta_2 = ?$

② $\Rightarrow x = (l_1 + l_2 c_2)\cos\theta_1 - (l_2 s_2)\sin\theta_1 \Rightarrow \theta_1 = ?$

$y = (l_1 + l_2 c_2)\cos\theta_1 + (l_2 s_2)\sin\theta_1$

③ $\Rightarrow \tan\varphi = \tan(\theta_1+\theta_2+\theta_3) \Rightarrow \varphi = \theta_1+\theta_2+\theta_3 \Rightarrow \theta_3 = ?$

$$\Rightarrow \begin{cases} \theta_2 = \text{Arctan2}(s_2, c_2) \\ \theta_1 = \text{Arctan2}(y, x) - \text{Arctan2}(k_2, k_1) \\ \theta_3 = \varphi - \theta_1 - \theta_2 \end{cases}$$

• جواب ← ۲ جواب برای θ_2

از روی θ_2 ، θ_1 بدست می‌آید

θ_3 وابسته به θ_1 و θ_2 است.

• انتخاب جواب

- براساس تابع انتخاب
- روش حداقل جابجایی ← انتخاب دوم به باید

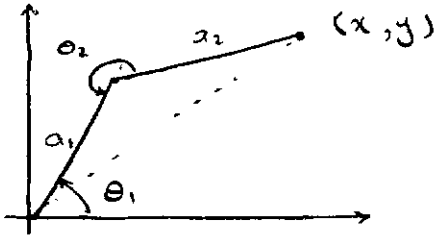
$$f = w_1 |\Delta\theta_1| + w_2 |\Delta\theta_2| + w_3 |\Delta\theta_3|$$

- لحاظ کردن حرکت آسان‌تر ← انتخاب از حرکت کمتر

← هزینه: عمر مفید کم می‌شود.

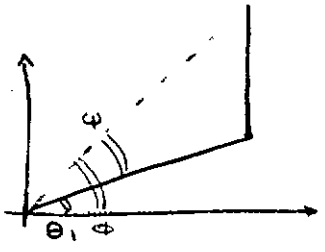
* راه حل هندسی برای حل سینماتیک معکوس

• حرکت سه بازوی دو درجه‌ای دوبعدی



$$x^2+y^2 = a_1^2+a_2^2-2a_1a_2\cos(\pi-\theta_2)$$

$$\theta_2 = \pm 2\times \text{Arctan}\left(\frac{(a_1+a_2)^2-(x^2+y^2)}{(x^2+y^2)-(a_1-a_2)^2}\right)$$



$\theta_1 = \Phi - \Psi$ Elbow Down •

$\Phi = \text{Arctan}\ \frac{y}{x}$

$\Psi = \text{Arctan}\left(\frac{a_2 \sin\theta_2}{a_1+a_2\cos\theta_2}\right)$

Elbow Up •

۶ تمرینات ← پیوست ، صفحات ← ۲۸ تا ۳۰ ← فصل ۱: ۱۹ و ۲۰ / فصل ۳: ۴، ۱۰ و ۱۷، ۲۲

Subject:
Year. Month. Date. (9)

* تعیین مسیر روبات
  - نقطه شروع
  - نقطه هدف
  - چند نقطه میانی
  - شبیه‌سازی محیط ← راه حل تغییر زمانی

* تفاوت Path و Trajectory
  - Path : حرکت از مبدأ به مقصد
  - Trajectory : حرکت روی مسیر در زمانبندی معین (مسیر در زمان مشخص)

☑ نازدی حرکتی

سرعت ← پیوستگی
حرکت نرم و Smooth ← طول عمر قطعات
کاهش استهلاک روبات

* Link Space Scheme

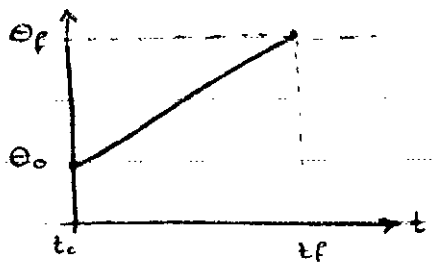
$(x_1, y_1, z_1, \phi_{11}, \phi_{21}, \phi_{31}) \xrightarrow{Ik} (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)_0$

$(x_2, y_2, z_2, \phi_{12}, \phi_{22}, \phi_{32}) \longrightarrow (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)_f$

$\begin{bmatrix}\theta_1 \\ \vdots \\ \theta_6\end{bmatrix}_0 \Longrightarrow$ به طور همزمان $\Longrightarrow \begin{bmatrix}\theta_1 \\ \vdots \\ \theta_6\end{bmatrix}_f$

- مسئله = رساندن θ به مقادیر نهایی به طور همزمان
- هر لینک برای یک موتور

Subject:
Year. Month. Date. ( )



• راه حل خطی

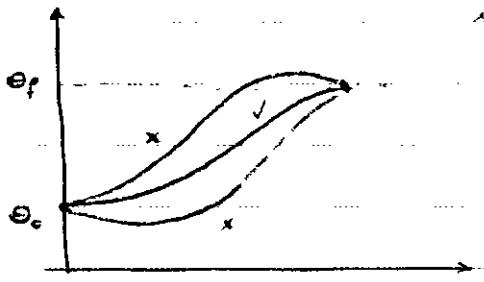
— از لحاظ کنترلر آسان است.

— در لحظه شروع حرکت، trajectory ، مقادیر شتاب بسته می‌شود و مقدار مثبت.

— حرکت smooth مثبت.

— شتاب بی‌نهایت

• راه حل غیر خطی



— بدون over shoot ← از θ_f بیشتر نباشد

— بدون undershoot ← از θ_f کمتر نباشد

— منحنی درجه ۳ مناسب است

— شتاب مسیر، سرعت است، درجه ۲ است.

— شتاب در حقیقت خطی است.

• راه حل درجه سوم

$$\theta(0) = \theta_c \longrightarrow \theta(t_f) = \theta_f$$

$$\theta'(0) = 0 \longrightarrow \theta'(t_f) = 0$$

$$\begin{cases} \theta = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \\ \theta' = a_1 + 2a_2 t + 3a_3 t^2 \\ \theta'' = 2a_2 + 6a_3 t \end{cases} \Rightarrow \begin{cases} a_0 = \theta_c \\ a_1 = 0 \\ a_2 = \\ a_3 = \end{cases}$$

— سرعت تا نقطه میانی زیاد می‌شود و سپس کم می‌شود

(Acceleration → Deceleration) — شتاب در مبدا مسیر مثبت می‌شود.

— سرعت تا مقصد منفی می‌شود

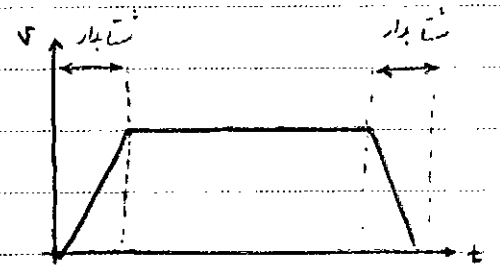
Subject:

Year. Month. Date. 10

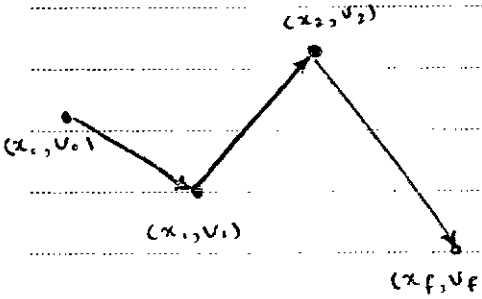
• سرعت ثابت

— تقریب سهمی‌وار

— سرعت ثابت ← شتاب صفر



• رد شدن از نقاط میانی (Via Point)



$\theta'(0) = \theta'_0 \longrightarrow \theta'(t_f) = \theta'_f$

سهمی بین هر دو نقطه

$$\begin{cases} \theta = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \\ \theta' = a_1 + 2a_2 t + 3a_3 t^2 \\ \theta'' = 2a_2 + 6a_3 t \end{cases} \Longrightarrow \begin{cases} a_0 = \theta_0 \\ a_1 = \theta'_0 \\ a_2 = \frac{3}{t_f^2}(\theta_f - \theta_0) - \frac{2}{t_f}\theta'_0 - \frac{1}{t_f}\theta'_f \\ a_3 = -\frac{2}{\quad} \end{cases}$$

• اگر شتاب در نقاط میانی مهم باشد ← معادله درجه پنج استفاده می‌شود

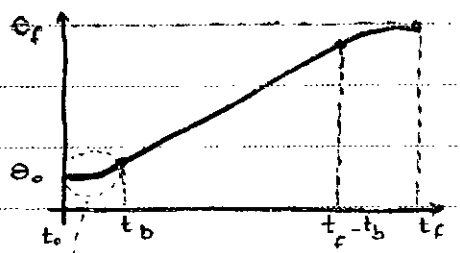
از نقاط نمی‌گذرد و از حدود آن نمی‌گذرد.

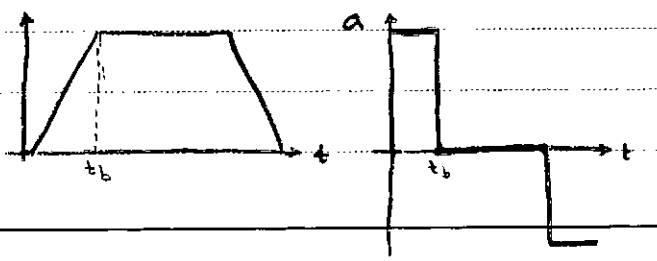
• استفاده از حرکت خطی برای تقریب حرکت سرعت ثابت

— پراکنده زمان t_b حرکت با شتاب ثابت

صورت می‌گیرد.

— دوره میانی حرکت خطی است.





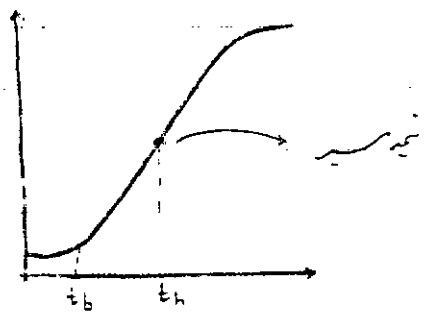
Subject:

Year. Month. Date. ( )

$$\ddot{\theta}(t_b) = \frac{\theta_h - \theta_b}{t_h - t_b}$$

شتاب حاصل از blend — سرعت در ناحیه خطی

$$\theta_b = \theta_c + \frac{1}{2}\dot{\theta} \cdots$$

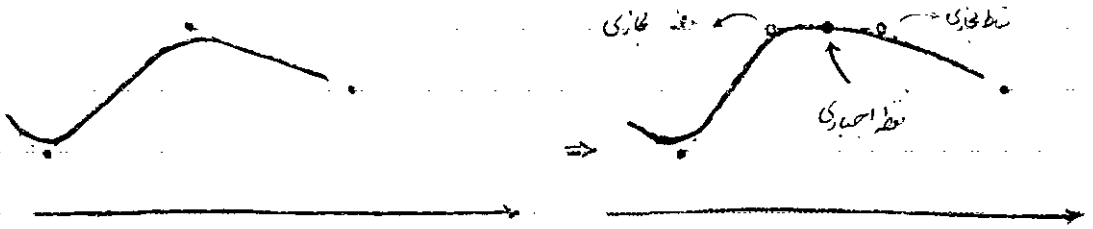


$$t_b = \frac{t}{2} - \sqrt{\frac{\ddot{\theta} t^2 - 4\dot{\theta}(\theta_f - \theta_c)}{2\dot{\theta}}} \xrightarrow{\Delta > 0} \boxed{\ddot{\theta} \geq \frac{4(\theta_f - \theta_c)}{t^2}}$$

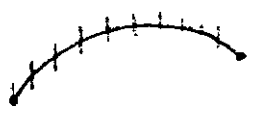
← آستانه پایین شتاب

- $\ddot{\theta} \uparrow \longrightarrow$ ناحیه خطی ↓
- $\ddot{\theta} = \frac{4(\theta_f - \theta_c)}{t^2} \longrightarrow$ ناحیه خطی = 0

- روش‌های اجباری از میانه‌ی نقاط
  - باید از نقاط با ناحیه خطی حرکت کرد.
  - باید via point اضافی در دو طرف آن تعریف کرد



- حرکت روی مسیر
  - گسستن مسیر بر حسب زمانی ← محاسبه مختصات نقاط روی مسیر $r = f(t)$
  - تعیین چند via point روی مسیر
  - مسطح بودن



Subject:

Year. Month. Date. 11

چه چیز داریم:

کنترل ← Control ← نیروی موتورها ← Dynamic ← شتاب

سرعت ← Jacobian ← سرعت زاویه‌ای موتورها

$$\text{Jacobian} = \frac{\text{مشتق بردار سرعت}}{\text{مشتق بردار زمان}}$$

controller باید این‌ها را در نظر داشته باشد:

- Workspace Violation ← خروج از WS *
- singularity ← شتاب یا سرعت بی‌نهایت *
- Joint Limits ← نقض غیرقابل *

* ویژگی‌های ایده‌آل

- علت استفاده از آن راحتی محاسبات برای ربات است.
- هرچرخش نسبت به واقعیت خطی محاسبه می‌شود.

Subject:

Year. Month. Date. ( )

Subject:

Year. Month. Date. 12

◎ روباتهای متحرک:

* صورت مسئله:

«روباتی طراحی کنید که با پخش جمله Come Here به سمت شما بیاید»

۱ Locomotion ← مکانیک ← چگونه روبات حرکت کند

* روشهای مختلف حرکت

• چرخ ← طراحی ساده

تک چرخ ← تعادل ×

دو چرخ ← چرخ با اندازه کافی بزرگ باشد.
قدرت مانور کم ×

سه چرخ ← سیستم پایدار با قدرت مانور بالا

چهار چرخ ← مشکل تعلیق ×

کمک فنر ← سیستم تعلیق ← هزینه

مناسب برای محیطهای صنعتی ولی نه خانگی

• پا ← تک پا

دو پا ← تعادل ×

قدرت مانور ✓

چهار پا ← تعادل ✓

تعداد زیاد درجات آزادی ← هر پا دارای ۳ درجه آزادی

شش پا ← سطح نسبتاً هموار ✓

توالی حرکت (Gait Control) ← تعداد پا: K

حالات مختلف: $(2K-1)!$

Bio-Mimic

Subject:
Year. Month. Date. ( )

- شنی دار ← عبور از جای ناهموار ✓
  - کنترل دشوار ← لغزش و سرخوردن
  - زنش feedback دشوار

- ریل ← نیاز به دستگاهی در محیط دارد و محدود در حرکت ماشین.
  - omni direction ← در هر لحظه می تواند در هر جهت بچرخد

* انواع چرخ

- ثابت
- فرمانپذیر (Steerable)
- هرزگرد (Castor) ← جهت وابسته به نیرو
  - کنترل نمی شود
- تمام جهت (Omni wheel) ← لوزی
  - سوزنی ← جبران است
  - در جهت مماس، تعدادی حلقه دارد

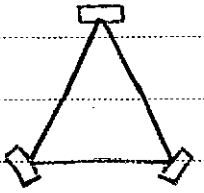
* انواع روبات از لحاظ چرخ

- دوچرخ ← دو چرخه
  - دو چرخ موازی
- سه چرخ ← دو چرخ ثابت + یک هرزگرد
  - دو چرخ ثابت + یک فرمانپذیر
  - دو چرخ ثابت مستقل + یک فرمانپذیر
  - سه چرخ تمام جهت
- چهارچرخ

Subject:

Year. Month. Date. (3)

* رویات تمام جهت



• تمام جهت خطای
بالا با صله جهت را مدض می کند

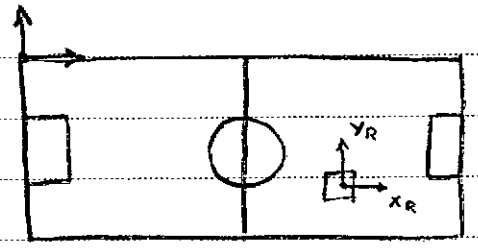
• تمام جهت با تاخیر
برای نترل جهت زمان می برد.

* سینماتیک رویاتهای متحرک

• تعریف: توصیف عکلرد مکانیکی رویات به منظور طراحی و نترل

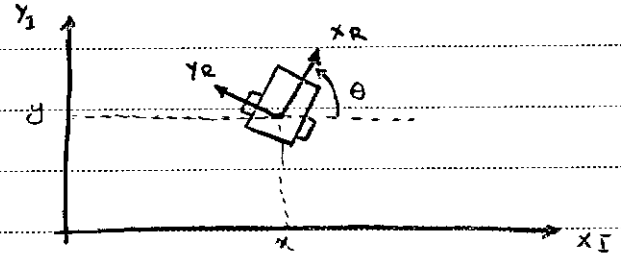
• محدودیت مرجع

- کار سختی است.
- برای رباتی که کاربایی دارد راحت است.



- برای رباتی که چند منظوره است (@ Home) محدودیت مرجع تعریف نمی شود.
- برای رویات نظامی می توان توسط GPS این نقطه را تعریف کرد.

• نشان دادن موقعیت رویات



- در روبجه

ابزار انتخاب دستگاه مختصات برای رویات
سادگی محاسبات است.

$$\xi = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

Subject:
Year. Month. Date. ( )

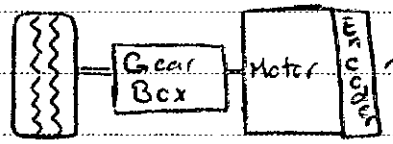
• صورت مسئله سینماتیک

— وسیله متحرک : چرخ

— وسیله تحت کنترل : چرخ

— جعبه دنده : تبدیل سرعت به گشتاور

— Encoder : سرعت چرخش موتور را اندازه می‌گیرد.



φ = موقعیت زاویه‌ای چرخ

$\dot{\varphi}$ = سرعت چرخش

— سنسور : سرعت چرخها ← سینماتیک ← موقعیت و سرعت در محیط

دشوار ←

سرعت چرخها ← سرعت و موقعیت در محیط

— چرخ ← سرعت چرخ : n تا $\dot{\varphi}$ چرخ

زاویه چرخ : m تا φ چرخ متحرک و فرمانپذیر

سرعت فرمان : m تا سرعت فرمان دادن به چرخ

φ : سرعت چرخ حول محور افقی

β : زاویه حول محور عمودی

$\dot{\beta}$: سرعت چرخش حول محور عمودی

$$\xi^{\cdot} = \begin{bmatrix} x^{\cdot} \\ y^{\cdot} \\ \theta \end{bmatrix} = f\left(\underline{\varphi}_{1..n}, \underline{\beta}_{1..m}, \underline{\dot{\beta}}_{1..m}\right)$$

سرعت ← $\int_{\Delta t}$ ← مقدار حرکت

Subject:

Year Month Date 14

Problem : Wall Following ☑

$y^{\cdot} = 0$ ← ربات باید در خط راست حرکت کند.

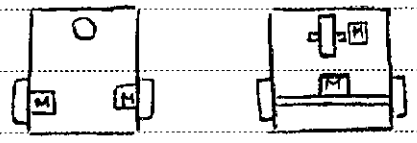
در دست انداز ها، ربات دچار خطا می‌شود. ← سرعت لحظه‌ای روی راستای y

خطا در راستای y وارد می‌شود و به طور جمع شونده ظاهر می‌شود

• مسائل سینماتیک ربات ها بستگی به ساختار ربات است.

ربات تفاضلی (Differential Robot) —

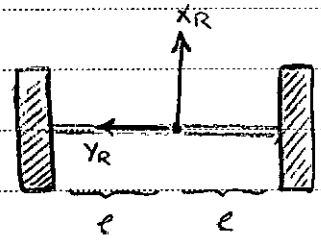
* ۳ چرخ
* دو تا موتور



✓ X

* چرخ ستون می‌چرخد.

مبدا مختصات، وسط فاصله دو چرخ



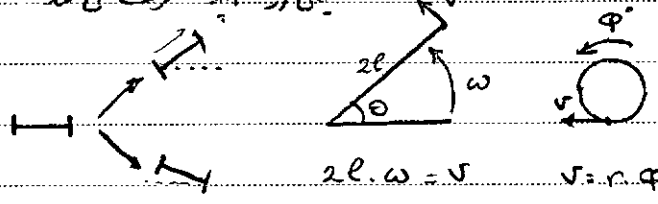
* چرخش: حرکت چرخ با سرعت متفاوت

$$\dot{x}_R = r \cdot \frac{\dot{\varphi}_1 + \dot{\varphi}_2}{2}$$

$\dot{y}_R = 0$ ← چون ربات نسبت به محور y خود جابجا نمی‌شود

یعنی از پهلو حرکت نمی‌کند.

$$\dot{\theta} = \frac{r}{2l}(\dot{\varphi}_1 - \dot{\varphi}_2)$$



$2l \cdot \omega = v$ $\quad v = r\dot{\varphi}$

Subject:

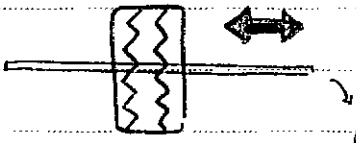
Year. Month. Date. ( )

$$\dot{\xi}_I = R^{-1}(\theta) \cdot \begin{bmatrix} -\frac{r\dot{\phi}_1}{2} + \frac{r\dot{\phi}_2}{2} \\ 0 \\ \frac{r\dot{\phi}_1}{2l} - \frac{r\dot{\phi}_2}{2l} \end{bmatrix}$$

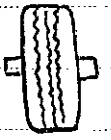
پس تنها داشتن $\dot{\phi}$ کافی است که توسط Encoder بدست می‌آید.

• محدودیتهای سینماتیکی

- محدودیت سرخوردن : چرخ ثابت در راستای محور دوران اینرسی حرکتی ندارد.



محور دوران اینرسی

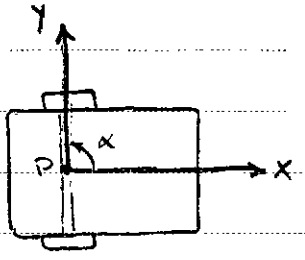


جهت غلتیدن ↑

جهت سرخوردن →

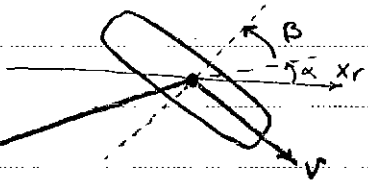
- محدودیت غلتیدن : پس وارد نشد!

سرعت خطی در نقطه تماس با زمین صفر است.



$\Rightarrow \beta = 0$

$\alpha = 90^\circ \rightarrow$ زاویه چرخ ثابت با محور x ها



عامل حرکت ← $r\dot{\phi}$

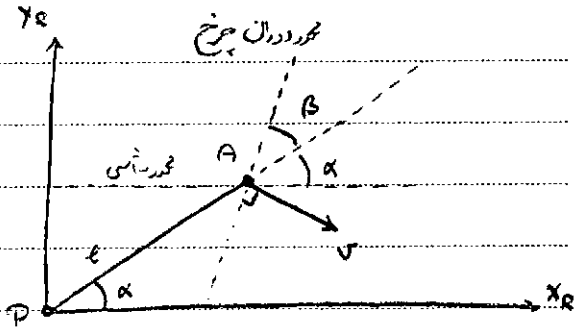
مجموع حرکت در راستای عمود بر چرخ = Φ

Subject:

Year. Month. Date. 05

محدودیت غلتیدن : $r\dot{\Phi} = v$

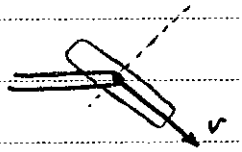
محدودیت سرخوردن : $\beta = 0$



P : مرکز مختصات

Y : محور محور چرخ

β : زاویه خط محور چرخ با خطی که از P بر مرکز چرخ می‌گذرد



محدودیت غلتیدن ← OMNI wheel

محدودیت غلتیدن ← Castor

ثابت ← محدودیت سرخوردن ، غلتیدن

• محدودیت غلتیدن چرخ ثابت

$$\left[\sin(\alpha+\beta) \quad -\cos(\alpha+\beta) \quad (-\ell)\cos\beta\right] R_\theta \dot{\xi} = r\dot{\Phi}$$

• محدودیت سرخوردن چرخ ثابت

$$\left[\cos(\alpha+\beta) \quad \sin(\alpha+\beta) \quad \ell \sin\beta\right] R_\theta \dot{\xi} = 0$$

• پس چرخ ثابت در مقابل حرکت محدودیت ایجاد می‌کند.

→ 1 limit

→ 2 limit

→ 3 limit → Locked

→ 2 limit

①، ② → محدودیت‌های آن دو کاملاً است

Subject:
Year. Month. Date. ( )

* چرخ فرمانپذیر

$\beta = f(t)$

• محدودیت:

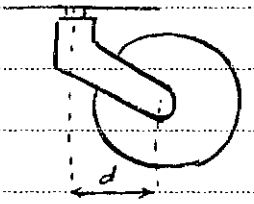
$$[\sin(\alpha+\beta) \quad -\cos(\alpha+\beta), (-\ell)\cos\beta]\, R(\theta)\, \xi^{\cdot} = r\varphi$$
$$[\cos(\alpha+\beta) \quad \sin(\alpha+\beta), \ell \sin\beta]\, R(\theta).\, \xi^{\cdot} = 0$$

• یک محدودیت اعمال می‌کند.

* چرخ هرزگرد

$\beta = f(t)$

$d \neq 0$



• محدودیت غلتیدن ثابت باقی‌مانده است.

• محدودیت سرخوردن

$$[\cos(\alpha+\beta), \sin(\alpha+\beta)\, d + \ell \sin\beta]\, R(\theta)\, \xi^{\cdot} + d\beta^{\cdot} = 0$$

$-d\beta^{\cdot}$ ← سرعتی که خود چرخ به خودی خود تا ابد آزادانه بچرخد

• چرخ هرزگرد محدودیت سرخوردن ندارد.
• چرخ کروی نیز محدودیتی ایجاد نمی‌کند.

Subject:

Year Month Date 16

* چرخ سوئدی

- زاویه چرخهای لبه چپ سوئدی : γ
- محدودیت سرخوردن

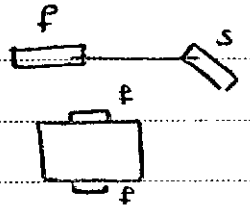
با طراحی مناسب، این محدودیت از بین می‌رود.

$$[\cos(\alpha+\beta+\gamma), \sin(\alpha+\beta+\gamma), \ell \sin(\beta+\gamma)]\, R(\theta)\, \dot{\xi} = r\dot{\varphi}\sin\gamma + r_{sw}\dot{\varphi}_{sw}$$

* محدودیت غیر holonomic رباتیک

- تنها وابسته به چرخهای ثابت و متحرک است.
- چرخهای ثابت: N_f , β_f , $\varphi_f(t)$
- چرخهای متحرک: N_s , $\beta_s(t)$, $\varphi_s(t)$

☑ دو چرخ



- $$\varphi(t) = \begin{bmatrix} \varphi_f(t) \\ \varphi_s(t) \end{bmatrix}$$

- محدودیت غلتش

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

J_2 ⟵ ماتریس شعاعی

$$J_1(\beta_s) = \begin{bmatrix} J_{1f} \\ J_{1s}(\beta_s) \end{bmatrix}$$

- محدودیت سرخوردن

$$C_1(\beta_s)\, R(\theta)\, \dot{\xi} = 0$$

Subject:
Year. Month. Date. ( )

• تعداد سطرهای ماتریس = تعداد چرخ

هر کدام از سطرها ـ ازای ثابت یا ماندن بودن چرخ به صورت آن ماتریس سه مشتقه مربوط خواهد بود.

$$\begin{bmatrix} J_1(\beta_3) \\ C_1(\beta_3) \end{bmatrix} R(\theta) \cdot \xi_I^{\cdot} = \begin{bmatrix} J_2\dot{\phi} \\ 0 \end{bmatrix}$$

☑ ربات دیفرانسیلی

$J_1(\beta_3)$ ⟶ دو سطر

$C_1(\beta_3)$ ⟶ دو سطر

چون محدودیت سرخوردن دو چرخ روبرو را نداریم، آنرا به صورت تابعی در نظر می‌گیریم.

$$\begin{bmatrix} \begin{bmatrix} 1 & 0 & \ell \\ 1 & 0 & -\ell \end{bmatrix} \\ \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \end{bmatrix} R(\theta)\ \xi_I^{\cdot} = \begin{bmatrix} J_2\dot{\phi} \\ 0 \end{bmatrix}$$

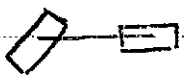
$$\xi_I^{\cdot} = R^{-1}(\theta) \begin{bmatrix} 1 & 0 & \ell \\ 1 & 0 & -\ell \\ 0 & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} J_2\dot{\phi} \\ 0 \end{bmatrix} = R^{-1}(\theta) \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \\ \frac{1}{2\ell} & \frac{-1}{2\ell} & 0 \end{bmatrix} \begin{bmatrix} J_2\dot{\phi} \\ 0 \end{bmatrix}$$

این رابطه کلی است. در ربات بدون محدودیت این رابطه برحسب $\dot{\phi}$ به دست آمد.

* قابلیت مانور ربات

• عبارتست از مجموعه درجه آزادی که ربات قادر به انجام آنهاست.

☑ دو چرخه



⟶ تغییر جهت چرخها = ماندن ⟶ 1 DoF

تغییر جهت چرخها = فرمان بدن ⟶ 1 DoF

Subject:
Year Month Date 17

• رو با یک، تغییر سرعت چرخها و تغییر جهت چرخها، قدرت مانور دارند.

$\delta_M = \delta_m + \delta_s$

δ_s : degree of steerability ⟵ تغییر جهت

δ_m : degree of mobility ⟵ تغییر سرعت

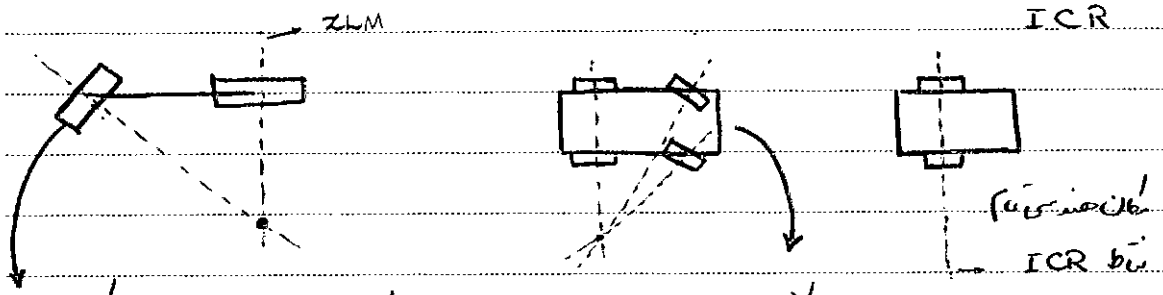
☑ دو چرخه و ماشین قدرت مانور بالایی دارند. علت آن عدم استقلال چرخهای از هم است.

δ_m •

• محدودیت سرخوردن دارد است.

روی محور عمود بر چرخ حرکت نداریم ⟵ Zero Line Motion

تقاطع خطوط ZLM ⟵ دوران حول دایره‌ای به مرکز آن می‌چرخد ⟵ Instantaneous Center of Rotation



☑

• هر خط ICR می‌تواند هر کدام از این نقاط باشد، و حالی آن توسط سرعت چرخها کنترل می‌شود.

برای پیدا کردن برای C_4 (β_s) یک خط رسم می‌کنیم

☑ دو چرخه

$\delta_M = 1 + 1 = 2 \longrightarrow \delta_m = 1 \;,\; \delta_s = 1$

☑ دوران دیفرانسیلی

$\delta_M = 2 + 0 = 2 \longrightarrow \delta_m = 2 \;,\; \delta_s = 0$

Subject:
Year. Month. Date. ( )

• حداکثر درجه ماتریس $= 3 \longleftarrow x, y, \theta$

$C_i =$ [معادلات محدودیت سرخودران چرخ]

$rank(C_i) = C_i$ تعداد سطرهای مستقل $\longrightarrow 0 \leq rank(C_i) \leq 3$

$\delta_m = 3 - rank[C_1(\beta_s)]$

$C_{is} =$ [معادلات محدودیت سرخودران چرخهای فرمانپذیر]

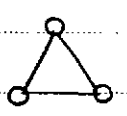
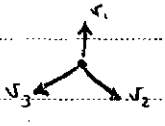
$rank(C_{is}) = C_{is}$ تعداد سطرهای مستقل $\longrightarrow 0 \leq rank(C_{is}) \leq 2$

چرخ فرمانپذیر ← $\delta_m --$, $\delta_s ++$ | $\delta_M = cte$ ←

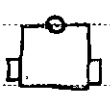
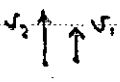
$\delta_s \longrightarrow$ برای ماندن نیاز به زمان چرخیدن چرخ است.

$\delta_m \longrightarrow$ تغییر سرعت چرخها نسبتا سریع است.

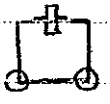
☑ سه چرخ

$\delta_m = 3$, $\delta_s = 0$ | $\Rightarrow \delta_M = 3$

$\delta_m = 2$, $\delta_s = 0$ | $\Rightarrow \delta_M = 2$



$\delta_m = 2$, $\delta_s = 1$ | $\Rightarrow \delta_M = 3$

Subject:

Year Month Date 18

$S_m = 1$

$S_s = 2 \Rightarrow S_M = 3$

Differential Degree of Freedom (DDoF) = S_m = Mobility

DoF = DDoF + Steering

DoF ← دینامیکی نقطه ☑ x, y, θ

DDoF ← دینامیکی مسیر ☑ S_M

DoF = 3 → در هر نقطه از محیط قرار گیرد

DDoF = 3 → در هر مسیری از محیط قرار گیرد

Holonomic Robot

رباتی که محدودیت سینماتیکی غیر هولونومیک نداشته باشد.

محدودیت سینماتیکی هولونومیک فقط تابعی از موقعیت رباتها باشد.

☑ محدودیت سرخوردن ← $\dot{\xi}$ ← دارای مشتق ← غیر هولونومیک

☑ اگر رباتها محدودیت سینماتیکی نداشته باشند، رباتها را هولونومیک می نامند.

☑ دوچرخه ای با دوچرخ ثابت

دوچرخ ثابت ← محدودیت سینماتیکی

حرکت در خط راست ← محدودیت سرخوردن ← $y = 0, \theta = 0$

محدودیت غلتش ← $\varphi = (x, r) + \varphi_0$

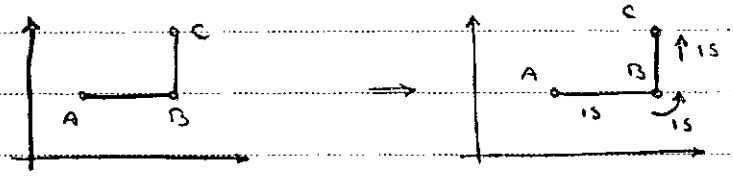
☑ رباتها Omni-Directional

محدودیت سینماتیکی ندارد ← هولونومیک

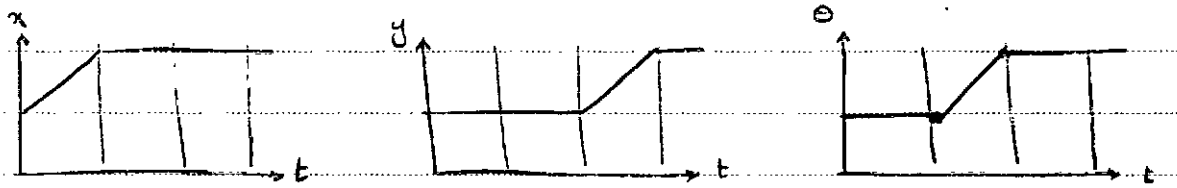
Subject:

Year. Month. Date. ( )

Trajectory = Path + Time



☑ روبات تمام جهت



☑ روبات دیفرنسیال

در مدتی که چرخ فرمانپذیری می‌چرخد، β_s تغییری نمی‌کند ولی (x, y, θ) بدون تغییری ماند.
به همین دلیل یعنی تواند این مسیر را در همان زمان برود که روبات تمام جهت بود، پس
Trajectory آن متفاوت است. (DDoF = 1)

$$DDoF \leq \delta_M \leq DoF$$

δ_M Trajectory — δ_M Path

Motion Control

نقاط آغاز و پایان

← مسیر

← تراز گیری

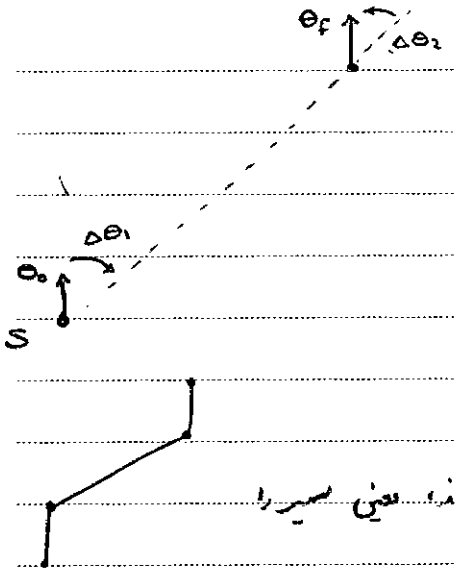
← سینماتیک

← سرعت چرخ‌ها

روش حلقه باز
روش حلقه بسته

Subject:

Year. Month. Date. 19



• حلقه باز ← غیر Smooth

۱- تغییر جهت به اندازه $\Delta\theta_1$

۲- حرکت روی خط راست

۳- تغییر جهت به اندازه $\Delta\theta_2$

۴- ربات برخورد با موانع مسیر را مجدداً تغییر می‌دهد، یعنی مسیر را تا لحظه مسیر ۴ طی نمی‌کند.

✓ تغییر goal ← اجرای دوباره ← غیر Smooth ← حلقه باز

• حلقه بسته ← نیاز به feedback

۱- اختلاف بین حالت S و G را به صورت یک متغیر خطای e نشان می‌دهد.

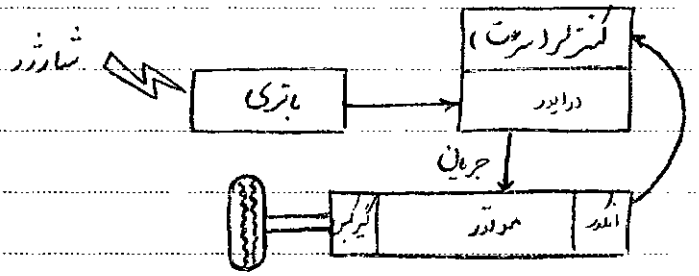
Subject:
Year. Month. Date. ( )

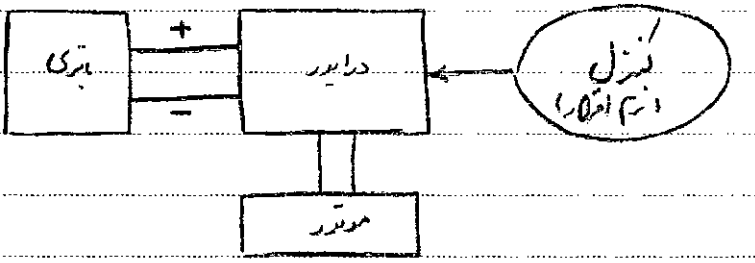
* ادراک (Perception)

Perception
↳ Cognition
↳ Action

- اطلاعات از Sensor بدست می‌آید.

☑ مثال: برای ساخت رباتی که به فرمان Come Here طبق العمل نشان دهد.





* انواع سنسور

- Bumper ← تشخیص برخورد با اشیاء (سوئیچ)
- Laser Range Scanner ← اندازه‌گیری فاصله (لیزر)
  - تهیه نقشه محیط ← پیدا کردن فضای باز
- Sonar ← اندازه‌گیری فاصله (فراصوت)
  - پرهیز از اشیاء موانع
  - تهیه نقشه محیط
- Pan-Tilt Camera ← یافتن اشیاء و تشخیص (دوربین با درجه آزادی)

Subject:

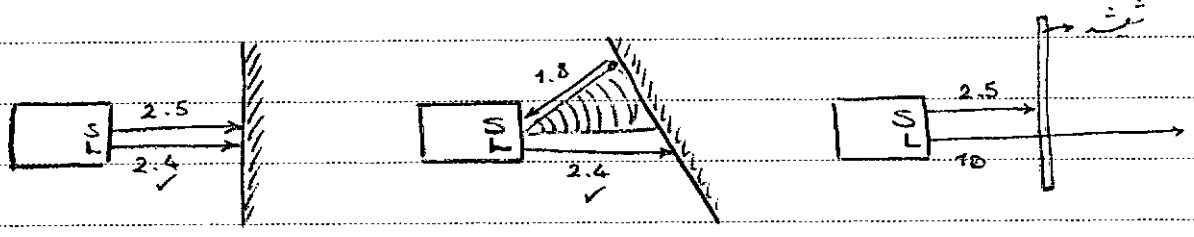
Year. Month. Date. 20

- Omni-Directional Cam. ← یک تصویر کلی از محیط
  - نسبت پایین و اعوجاج
  - میدان دید محدود
- IMU ← اندازه‌گیری مسافت طی شده و جهت ربات
  Inertia Measurment Unit
- Wheel Encoder ← کنترل سرعت ربات
  اندازه‌گیری فاصله و جهت
- Microphone ← استفاده از چند میکروفون برای تشخیص جهت

Sensor Fusion ✱

- دلایل ← نویز سنسور
  - محدودیت دقت
  - خرابی و غیرقابل اعتماد بودن
  - پوشش محدود محیط
  - عدم توصیف دقیق
  - ران بودن سنسور
- ترکیب داده‌های سنسورهایی که روی یک مسئله بحث می‌کنند.

Sonar , Laser ☑



- روش‌ها ← چند سنسور مختلف
  - چند موقعیت مختلف
  - در زمان‌های مختلف

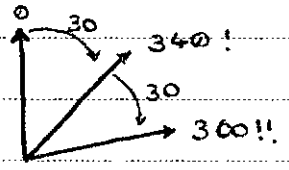
Subject:
Year. Month. Date. _( )

۱۰۱

• تقسیم :
- روش کالن
- روش بیزین
- روش شبکه عصبی

☑ قطب نما

مثال غیرخطی بودن



$30^\circ \rightarrow 20^\circ$!

$30^\circ \rightarrow 40^\circ$!

خطی ← جبران کردن ← کالیبره کردن ← نرم افزار ← ورودی غیرخطی

Subject:
Year. Month. Date. 21

* مکان یابی روبات (Localization)

- روبات میداند جای خود را در محیط تشخیص دهد.

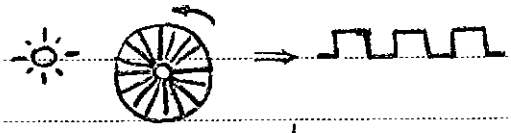
✓ سنسورهای مرتبط

Camera ، Range Scaner ، Sonar ، Encoder ، IMU

- مکان یابی فاز اول تقسیم کاری است.

* انکودر

- صفحه مدرج
- چرخش را به پالس ریاضی تبدیل می‌کند



- با نصب سنسورهای نوری متناوب با اختلاف فاز، می‌توان دقت انکودر را بالا برد و جهت چرخش را مشخص کند.

- Incremental Encoder ← کدگذاری ← موقعیت نسبی
- Absolute Encoder ← کدگذاری ← موقعیت مطلق

* Resolver

- دو سیم پیچ عمود بر هم
- یک سیم پیچ به محور چرخش متصل است.

* تکنیکی که از روی اندازه‌گیری حرکت چرخ مکان یابی می‌کند مربوط به Dead Reckoning است
در آن تغییرات موقعیت و جهت نسبت به یک فریم مرجع مشخص می‌شود.

Subject:
Year. Month. Date. ( )

* مشکلات Odometry
  - خطای جمع شونده (cumulative)
  - عدم قطعیت
  - عدم خطا
  * اختلاف چرخش موتورها
  - موقع چرخیدن ← لیز خوردن ، تفاوت چرخ ، تفاوت سرعت موتورها
* کاربرد Odometry
  - ساده و ارزان
  - قابل اعمال در فواصل کوتاه
  - استفاده از Sensor Fusion برای ترکیب اعداد متری و بینایی ماشین

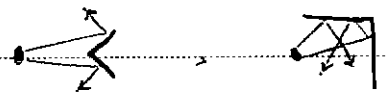
* ژیروسکوپ
  - How Stuff works.com //
  - اندازه گیری نسبی میزان چرخش به کمک روابط نیرویی ژیروسکوپ
  - خیلی دقیق برخلاف قطب نما
  - فاقد Cross Sensivity برخلاف قطب نما
  - گران قیمت
  - حساس بدن به ضربه

* Laser & Sonar
  - موج صوتی به محیط می فرستد / بسته زدن به محیط می فرستد
  - با گرفتن آن فاصله را مشخص می کند ← خاصیت Time OF Flight
  - کاربرد در تشخیص موانع
  - تهیه نقشه از محیط اطراف
  - Sonar معمولاً بهنای 30° را جهت پوشش فراهم می کند برای همین معمولاً به طور آرایه‌ای در ربات قرار می گیرد در این حالت Cross talk را ایجاد می کند. این مشکل با

اصطلاح نازل ، استراتژی آتش کردن حل می‌شود. می‌توان در سنسورهای فراصوتی از فرکانس‌های متفاوت (40 ~ 180 KHz) استفاده کرد.

☑ سنسور مبدأ راجعی باریک برای جمع‌آوری از موانع کاربرد دارد. دقیقاً مشخص نیست که در چه محیطی کار می‌کند.

• مشکلات سنسور



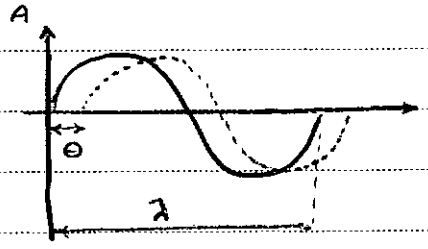
محدودیت در جهت ، تغییرهای محیط ،

LaSer *

• باریکه نور را روی سطح می‌تابند و بازگشت آنرا دنبال می‌کنند.

• می‌توان از رابطه $d = c \cdot t/2$ استفاده کرد، چون سرعت بالایی دارد، زمان time_of_Flight بسیار کوچک است و قابل اندازه‌گیری نیست.

• اندازه‌گیری فاصله با Phase Shift



$D' = h + 2D = h + \frac{\theta}{2\pi}\lambda \rightarrow D = \frac{\lambda}{4\pi}\theta$

$\theta > 2\pi \Leftrightarrow D > \lambda \rightsquigarrow$ ambiguity

• استفاده از آینه چرخان برای 2D

Resolution ~ feedback داشته باشد ~ * موتور چرخاننده آینه باید

* تعداد اسکن در ثانیه

* عمق مورد اسکن

• مشکلات

* سطح شفاف

* سطح صیقلی و آینه

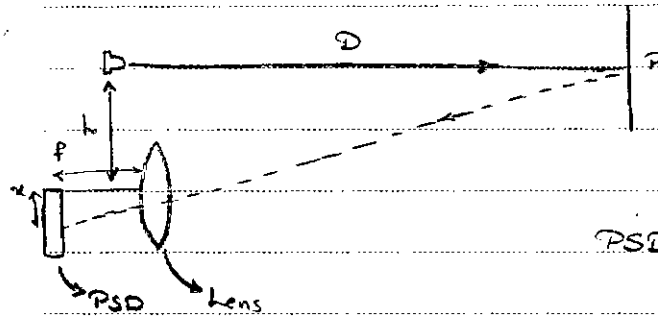
Subject:
Year. Month. Date. ( )

• روش Triangulation

* استفاده از خواص هندسی

* روش active

*



$D = f \frac{h}{x}$

* f : فاصله کانونی

* h : فاصله لیزر تا فرستنده

* x : فاصله پرتو برگشت روی PSD

* دقت این روش به خصوص دستگاه است (متغیر بین h و f)

* بازه اندازه‌گیری 8 ~ 80 cm

$D = f \frac{h}{x}$ $D \uparrow \Rightarrow x \downarrow \Rightarrow$ از دقت PSD خارج می‌شود

* تبدیل به 2D

استفاده از CCD به جای PSD

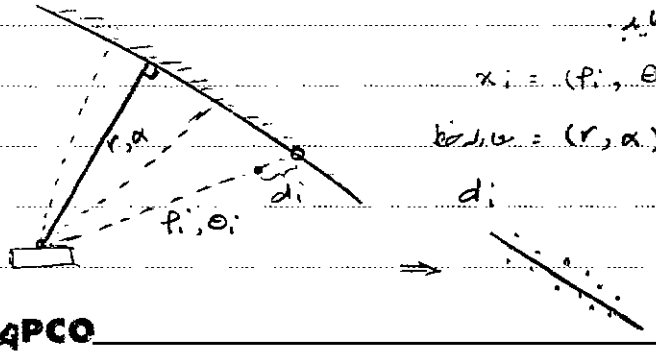
استفاده از آینه‌ای خاص برای تابش

• استفاده از لیزر با دور مرتعش

* نوع حرارتی : منابع گرما، لامپ فلورسنت

* لیزر فرکانس خاص را می‌فرستد و همان را تفکیک می‌کند.

• خطا در متدهای اندازه‌گیری را باید بررسی کرد.



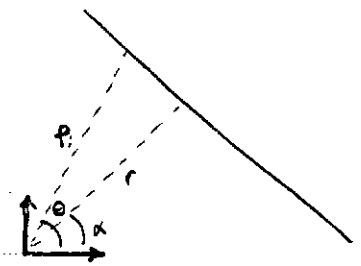
$x_i = (P_i, \theta_i)$ داده‌ی اندازه‌گیری

(r, α) = بردار خطا مطلوب

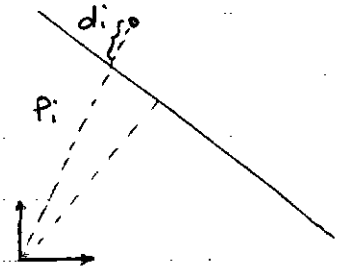
d_i خطا در دقت اندازه‌گیری

PAPCO

$P_i \sim N(\rho_i, \sigma_{\rho_i}^2)$ → ناشی از لیزر
$Q_i \sim N(\theta_i, \sigma_{\theta_i}^2)$ → ناشی از انکدر
} مسئله

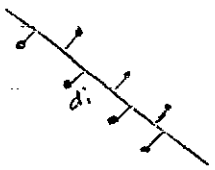


$\Rightarrow$



$\cos(\theta-\alpha) = \frac{r}{\rho_i}$ $\cos(\theta-\alpha) = \frac{r+d_i}{\rho_i}$

نقطه ⇐ خط :
regression
feature extraction



$\varepsilon : \sum d_i^2 , \sum (\rho_i \cos(\theta_i - \alpha) - r)^2$
بهترین خط ، میزان ε را min می کند.

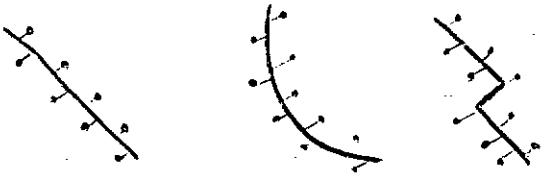
$\frac{\partial S}{\partial r} = 0 , \frac{\partial S}{\partial \alpha} = 0$

- در سنسورهایی مانند لیزر ، فاصله روی خطا تأثیرزیادی دارد. یعنی اگر فاصله بیشتر باشد، خطا بیشتر خواهد بود. پس به ازاء لیزرها وزن اعمال می کنیم. به عنوان مثال :

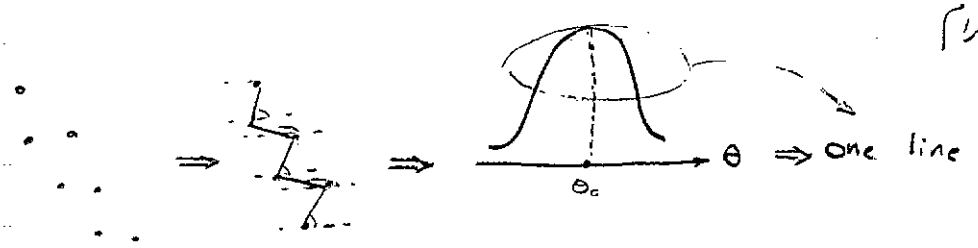
$\omega_i = 1/\sigma_i^2 \Rightarrow S = \sum \omega_i \cdot d_i$

* نیاز به سنسور σ_i^2 به صورت offline

- حال اگر دیوار روبرو به جای دیوار مستقیم ، حالت زیری باشد.



- الگوریتم هیستوگرام



Subject :

Year . Month . Date :

* سنسورهای بینایی

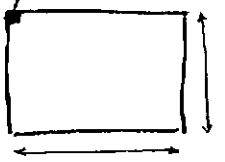
CCD CMOS

تکنولوژی	براساس تکنولوژی CMOS	براساس حافظه
نسبت	پایین تر	بالاتر
قیمت	ارزان تر	گرانتر
اندازه	کوچکتر	بزرگتر

• تابل تقدیر

* مدل رنگی

* تجزیه ای از bw (Picture element . pixel



* Binary

یک بیت کافی است

* Grey Level

هر نقطه یک Byte در نظر گرفته می شود

0000 0000 : Black

1111 1111 : White

* Color Image

هر نقطه سه Byte در نظر گرفته می شود

RGB

R G B

0 0 0 : Black

255 255 255 : White

* کاربردهای بینایی

Target Detection —

Target Tracking —

Stereo Vision —

Subject:

Year. Month. Date. 24

Laser + Stereo Vision ← Obstacle Avoidance —

Navigation —

Mapping —

Navigation ← دو دسته کلی عبارتند از

Object Recognition ←

• عملیات دی تصویر جهام

Pre-Processing * (Conditioning)

حذف نویز

حذف اطلاعات اضافه تصویر مانند پس زمینه

روشها ← Filtering

Edge Detection

Thresholding

Labeling / Grouping *

روشی ← Connected Component Lbl.

Correlation

Hough - Transformation

Extraction *

روشی ← Disparity

Matching *

متالیه بدل ←

☑ مرتب کردن در زمین فوتبال

Pre-Process *

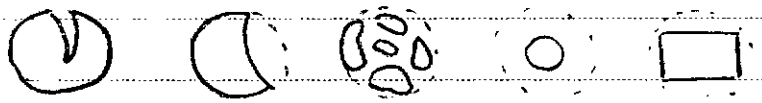
حذف زمین سبز رنگ

تشخیص دنبال شدت های قرمز

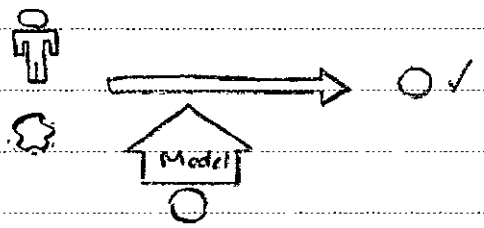
Subject:

Year. Month. Date. ( )

Labeling *



Model Matching *



• استفاده از بینایی برای تشخیص فاصله

Visual Range Sensor *

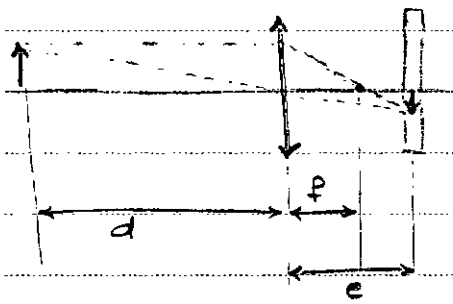
1 بردین ← Depth From Focus

2 بردین ← Stereo Vision

روشنایی *

Depth from Focus •

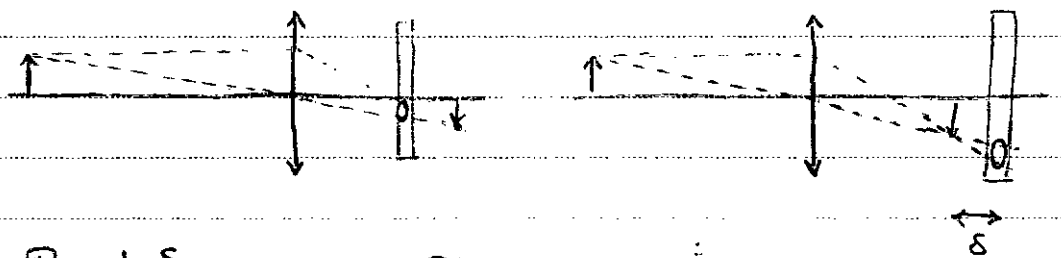
تقریب زدن فاصله کانونی *



$$\frac{1}{f} = \frac{1}{d} + \frac{1}{e}$$

Blur دیدن *

نقاطی که دوی سنسور تبدیل نشدند، یک دایره می دهند تار هستند.



$R = \frac{L\delta}{2e}$ ~ Blur شعاع دایره

با داشتن R، L و δ می توان e را بدست آورد.

P4PCO

↳ باید تحلیل شود.

Subject:

Year Month Date 25

* وضوح تصویر (Sharpness)

اختلاف نقاط کنار هم

$Sh_1 = \sum |I(x,y) - I(x-1,y)|$

$Sh_2 = \sum |I(x,y) - I(x-2,y-2)|$

باید مقدار این Max گردد.

این Auto Focus در دوربین‌ها کاربرد دارد.

* ایده اصلی کار

جسم 3D ⟵ تصویر 2D + عدم وضوح

عدم وضوح ⟵ Blur ⟵ فاصله تا دوربین

Stereo Vision ○

* با این روش می‌توان فاصله را بدست آورد.

از دو دوربین و Sensor Fusion استفاده می‌کنند.

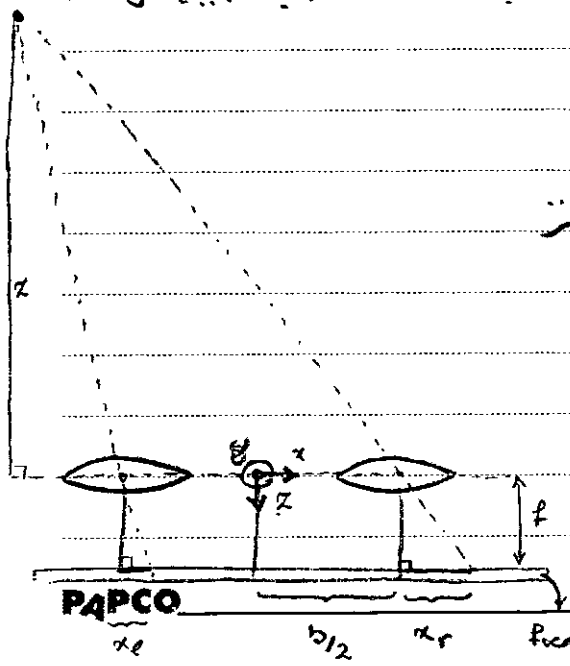
اگر دوربین‌ها خطی از هم جدا باشند تقریباً منظره یکسانی را می‌بینند

اگر دو دوربین موازی باشند تنها جای اشیاء در تصویر فرق می‌کند. به این جابجایی disparity می‌گویند.

* دو دوربین با محور اپتیکی موازی

مبدا مختصات روی مرکز خط واصل دو لنز

فاصله دو لنز (b) : Base



تشابه مثلثی

$$\frac{z}{f} = \frac{x - b_{12}}{x_l}$$

$$\frac{x_l}{f} = \frac{x - b_{12}}{z}$$

$$\Rightarrow z = b\frac{f}{x_l - x_r}$$

Subject:
Year. Month. Date. ( )

f و b ثابت اند.

$z = b \frac{f}{x_e - x_r}$ عمق ← $x_e - x_r$ میزان جابجایی است

* مشکلات

- f با کارخانه سازنده با دقت بالا اندازه‌گیری میشود.
- اگر دوربین را خود ما ببندیم ← b دست خود ماست.
  - هر دوربین ۲ باید کاملاً موازی باشد.
- عمق مستقیماً به b و دقت آن وابسته است.
- پیدا کردن نقطه در دو تصویر برای محاسبه عمق کار دشواری است.
  Correspondance Problem ←

* روشهای تشخیص Correspondance

- Edge Detection
  - اگر دوربین ۲ نسبت به هم بالا و پایین نباشند (خاصیت Epi-Polar)
  - آنگاه می‌توان با شروع از یک ارتفاع خاص ($z = c_k$) با سیگنال اولین لبه، آنرا با اولین لبه دوربین دیگر تطبیق دهیم.
  - اعمال لاپلاسین (مشتق گیری از شدت نور تصویر)

* خواص

- ناصله نسبت عکسی با disparity دارد.
- مقدار disparity با b رابطه مستقیم دارد.

$b \uparrow \Rightarrow$ دقت $\uparrow$

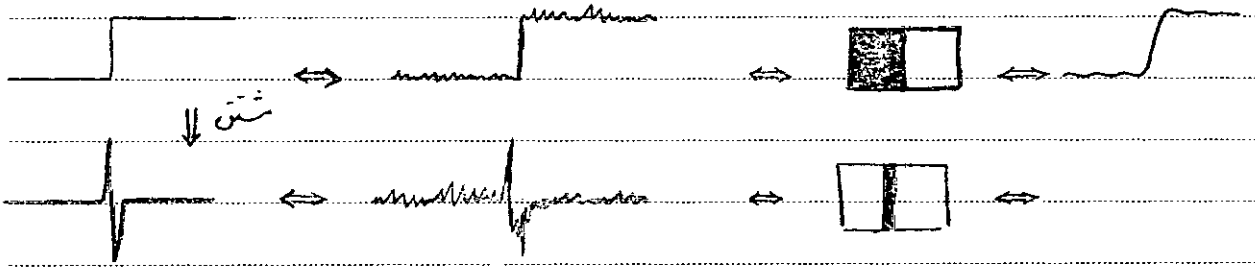
- اگر b خیلی زیاد باشد ممکن است دسته‌ای از تصویر خارج شود.

* تعریف لبه: تغییرات قابل ملاحظه Brightness

Subject:

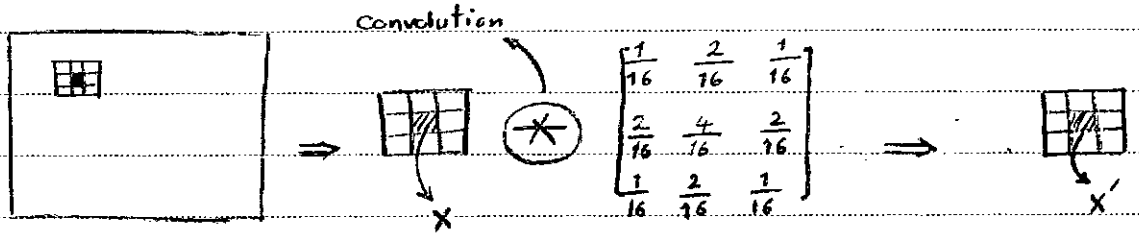
Year Month Date 26

* تشخیص لبه با لاپلاس



Step / Edge with Noise Real Image with Filter

* حذف نویز با فیلتر گوسی (Smoothing)



$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 0 \end{bmatrix} \rightarrow$$ مقدار 10 نویز است $\rightarrow$ ⊡

$$\begin{bmatrix} 0 & 10 & 0 \\ 0 & 10 & 0 \\ 0 & 10 & 0 \end{bmatrix} \rightarrow$$ لبه $\rightarrow$ ▯▯

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 0 \end{bmatrix} \rightarrow X' : 0 \times \frac{1}{16} + 0 \times \frac{2}{16} + \cdots = 10 \times \frac{4}{16} = 2.5$$

* محاسبه لاپلاس تصاویر

— Laplacian

$$L = \frac{\partial I^2}{\partial^2 x} + \frac{\partial I^2}{\partial^2 y}$$

— Kernel

$$L = P \circledast I = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \circledast I$$

Subject:
Year. Month. Date. ( )

* خلاصه سازی Correspodance

Correspondance
↳ Find Two Conjucated Point
↳ Find Point where Intensity Changes
↳ Noise Filitering (e.g. Smoothing)
↳ Differentiation (e.g. Laplac)
↳ kernel

* برای اطمینان از صحت correspodance آنرا دوباره فیلتر می‌کنند و به تصویر Confidence می‌دهند و آن نقاطی که به نظر آنها بیشتر ثابت می‌شود سطح می‌شود (بر اساس texture مناسب)

✎ موضوع ارائه:
یک پایان نامه دکتری مربوط به سالهای 2000 به بعد
درمورد مباحث مربوط به رباتهای متحرک (فصل ۴، ۵ کتاب)
مثل: Localization, Mapping, Navigation, Sensor Fusion, ...
انتخاب و ارائه شود. (۲۰ دقیقه)
موضوع را تقریباً مباحث کنید.

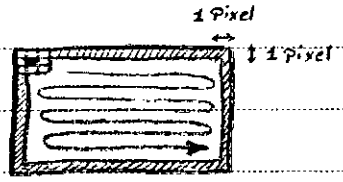
✎ تمرینهای سری چهارم
تحویل ۸۷, ۹, ۱۹ ← سید سعید قریشی

P4PCO

Subject:

Year. Month. Date. 27

* روشهای اعمال Kernel



- نقطه وسط با پیکسل Pixel های مختلف قرار گیرد.
- گاهی سطر ستون اول و آخر را در نظر نمی گیرند.
- برای هر نقطه 9 ضرب، 9 جمع نیاز است.

* تشخیص لبه به روش Canny

- مراحل آن: تصویر I ابتدا هموار شود و سپس مشتق گرفته شود، در مرحله پایانی انجام می دهد.

$$(G \circledast I)' = G' \circledast I = R(x)$$

$G_\sigma(x) = N(0, \sigma) \longrightarrow G'_\sigma(x) = ?$

- برای تشخیص لبه از تابع $G' \circledast I$ بدست آمده، آستانه گیری می شود.
- تصویر دوبعدی است. برای لبه یابی بعدی نیاز به فیلتر جداپذیر است.

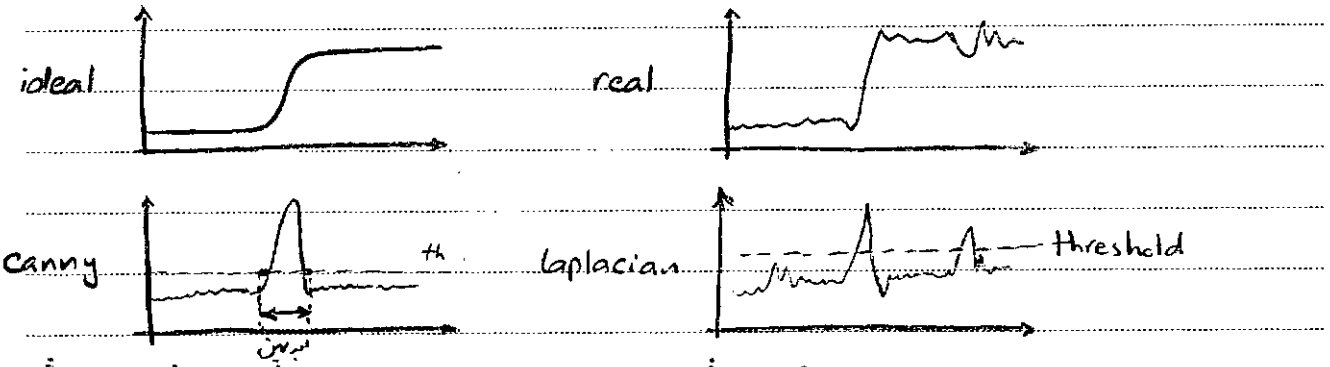
$G_\sigma(x,y) = G(x) \cdot G(y)$

$F_V(x,y) = G'(x) \cdot G(y) \longrightarrow R_V = F_V \circledast I$

$F_H(x,y) = G(x) \cdot G'(y) \longrightarrow R_H = F_H \circledast I$

$\longrightarrow R^2 = R_V^2 + R_H^2 \longrightarrow$ آستانه گیری

- لبه های ضخیمی بدست می دهد. لبه ها از لازم های تصویر مشتق گرفته شده است.
- لبه ها ممکن است نسبت به threshold پهن شود.



- لبه های پهن را می توان توسط روش Non Maxima Suppression لاغر شود. این روش از نقطه یابی استفاده می کند ← ولی خط های پهن به نقطه تبدیل می شود.
- زمانبر است ← برای کاربرد Real-Time مناسب نیست

Subject:
Year. Month. Date. ( )

* تقریب های اولیه Canny
- برای یافتن زوال بهتر
- مشتق را با تفاضل تقریب می‌زند. ← روش گرادیان
- عمل ضرب را حذف کرده است.
- Roberts ← برای لبه‌های کج مناسب است
- Prewitt ← لبه‌ی افقی و عمودی + آستانه‌گیری
- Sobel ← لبه‌ی افقی و عمودی با تأکید بر مرکز کرنل + آستانه‌گیری
- ابعاد kernel بزرگتر ← سرعت ↓
دقت ↑

* تشخیص اشیاء بر اساس رنگ
Color Tracking Sensor
• ویژگی های بدنه
☑ تب تاریخی در زمین فوتبال
- شکل ← فاصله نزدیک ← گرد
فاصله دور ← تخت
Model ←
- حرکت
- اندازه
• ساده سازی مسئله
☑ رنگ تب تاریخی ← هر چیزی که تاریخی بود، تب است؟
رنگ روبات آبی
☑ پروژه Come Here
انسان ← دمای بدن ← خیلی بیشتر بوده است.
شکل ← حتماً انسان باید سرپا باشد.
صورت ← معمولاً صورت کسی که با روبات کار دارد به سمت اوست.
Face Detection ← معمولاً پوشیده نیست (تغییر می‌کند)

Subject:

Year. Month. Date. 28

صورت ← شکل ← ابعاد ظاهری صورت ← نسبتها ← مثلاً طول سر برعرض سر نسبت است

↓ تداخل

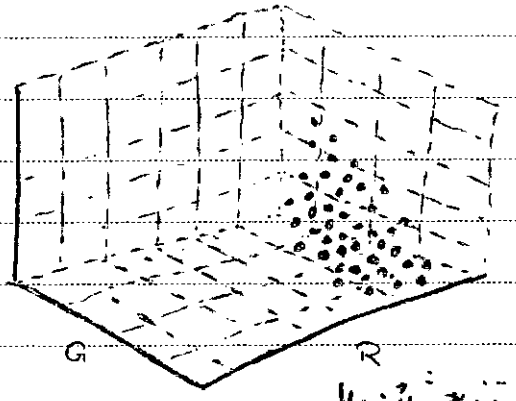
نوزاد ← نادیده گرفتن خط تداخل چشم و لب

↓

بچه ← بیشتر به جزئیات صورت توجه می کند.

↓

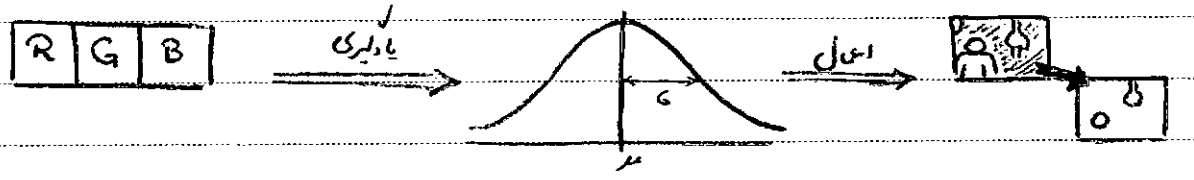
یک صورت چیست ← بررسی آماری روی یک صورت



بردار میانگین

توزیع نرمال

یک صورت برداشت شده از تمام نقطه های دیگر آن به عنوان مقدار استفاده می شود.



سپس با استفاده از Data Fusion ، لایه های دیگر را روی شکل جداسازی می کند.

• تبدیلات رنگ

* شکل RGB ← تغییرات شدت روشنایی

* مدل H.I.S

Hue + Intensity + Saturation

↓

هر رنگ یک S خاص دارد.

RGB 1 : 0 100 0

RGB 2 : 100 255 0 ⟹ این S یک رنگ خاص

Subject:
Year . . Month . . Date . ( )

پروفایل دایره‌ای : H
شدت روشنایی : I

مثلاً H صورت یک سیاه پوست دیگ سفید پوست دیگران است و I آنها فرق می‌کند.

YUV *

اطلاعات تصویر را از بین نمی‌برد.
V معادل I است.

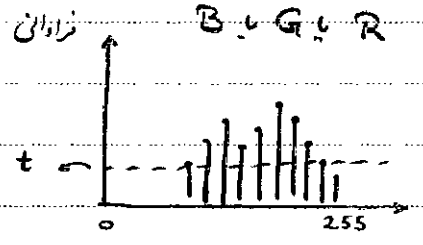
• تشخیص کف زمین — Floor Plane Extraction

* روش پر دردسر

عکس ورودی → مدل از رنگ پوست → تشخیص رنگ → تشخیص زمین

* روش هیستوگرام

روش هیستوگرام → تشخیص رنگ → تشخیص زمین



اگر تعداد از t بیشتر است. آنها ⟹ t به عنوان آستانه ⟹ شمارش پایانی انجام شود

Subject:
Year Month Date 29

* انتشار خطا

- هر اندازه گیری دارای خطاست.
- خطا منتشر شونده است.

نقطه ⟹ خط ⟹ دیوار

(خطا) (خطا) (خطا)

انتشار خطا

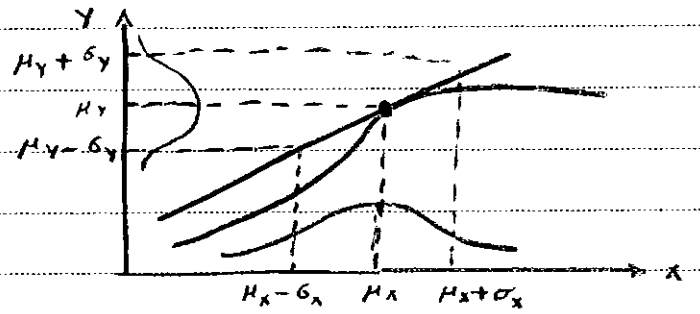
- برای مسایلی که انتشار خطا غیر خطی است.

C_x : کوواریانس خطای ورودی

C_y : کوواریانس خطای خروجی

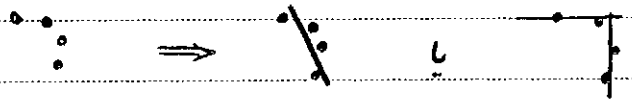
F_x : مشتق بردار نسبت به بردار ← ژاکوبین

$$C_y = F_x C_x F_x^T$$



- Hough Transform

* پیدا کردن خطی در مجموعه‌ای که نمی‌دانیم چند خط وجود دارد.



* ① معادله خط

$y = ax + b$ ⟶ معلوم: x, y ⟶ $b = (-x)a + y$

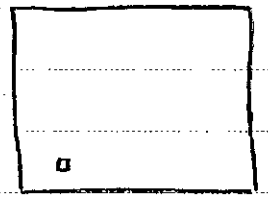
② به ازای نقاط مختلف معادلات مختلفی برای b و a بدست می‌آید.

③ یک بردار b برحسب a را رسم می‌کند و به ازای هر معادله روی تمام نقاط آن یکی اضافه می‌کند.

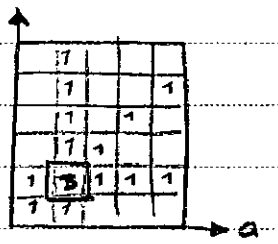
Subject:
Year. Month. Date. ( )

④ در نقاط متناظر نقاط بالا خواهند بود.

⑤ با آستانه گرفتن از اعداد بدست آمده، شکل مشخص می‌شود.



P:	x	y	$b=(-x)a+y$
P_1	0	1	$b=1$
P_2	1	2	$b=-a+2$
P_3	2	3	$b=-2a+3$



* نمودار a و b را با جدول نشان می‌دهد. این جدول تقریب خوبی از خطوط است. هرچقدر ابعاد آن بالا رود تقریب بهتری می‌شود ولی ابعاد بزرگ مشکل ساز است.

Subject:

Year. Month. Date. 30

Localization ✱

• ربات که می‌خواهد در محیط حرکت کند باید بداند کجاست.

• Navigation
  - Perception
  - Localization
  - Path Planning
  - Motion Control

(Cognition حالت خاصی از)

• آیا همیشه Mapping لازم است؟ در دنیای واقعی وجود دارد: خیر و بله (!!!!!!)

Mapping ← ساخت نقشه ← بله

خیر ← ربات ← قابل شناسایی

Planning دشوار می‌شود.

• نقشه

— از پیش ساخته باشد.

— یاد گرفته شود.

برای ساخت آن نیاز به حرکت ربات در محیط است تا نقشه را بسازد.

نقشه پیش نیاز Navigation است.

اول مرغ بود یا تخم مرغ ؟!

SLAM: Simutaneous Localization And Mapping

Behavior Based (رفتار تعریف می‌شود): دنبال کردن یک زمین، مشخص کردن در بسته‌ی ...

Model Based (نقشه می‌سازد): محل ربات در نقشه مشخص می‌شود. عمل ربات نیز بدانی می‌شود. یک سیر از ربات به در بسته‌ی تولید می‌شود.

P4PCO

Localization → Target Detection → Path Planning → Motion Control

Subject :

Year . Month . Date . ( )

Behavior - Based *

- واحد تعیین Goal
- واحد آشکار محیط ← واحد تصمیم گیری ← عمل ران
- واحد های دیگر
- در عمل بیشتر جاهای محیط در یک به بن بست می خورد. مثلاً مدتهای مسیریاب در ماریچ

نمایش نقشه *

- چه چیزهایی از محیط باید نمایش داده شود
  - ☑ خط ← چون سنسور می تواند خط را تشخیص بدهد
  - آرایه دوبعدی ← محیط آزاد + موانع
  - گراف

- قبل از اینکه نحوه نمایش نشان داده شود ، باید دید که سنسور چه چیزی به دست می دهد
  - فاصله
  - خط
  - ...

- ابعاد نقشه به دقت سنسور بستگی دارد.
  - ☑ زمین ۳x۳ با تقریب دقت ۵cm ← آرایه دوبعدی ۶۰ x ۸۰

- Geometric Map
  - شامل موقعیت تقریبی اشیا ← خط، زاویه، بافت و ...
  - همه اشیا با چند ضلعی های Convex نشان داده می شود.
  - مناسب برای نقشه های دوبعدی
  - تمام اشیاء محیط (یا جداقل آنها که توسط سنسور گرفته می شود) در نقشه می آید.
  - دقتی نقشه دقیقاً مشخص باشد مناسب است

Subject:
Year. Month. Date. 31

☑ داشتن نقشه CAD ساختمان

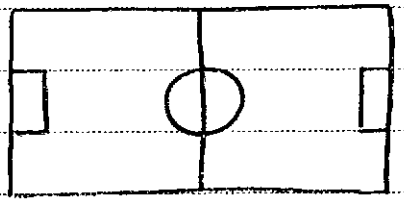
☑ ربات small size

ابعاد زمین کاملاً مشخص است.

ساده است.

سنسور مورد استفاده بینایی است.

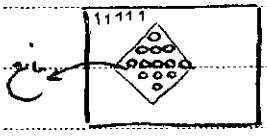
دوربین با ثابت شدن ابعاد رابطه ثابت را دهد.



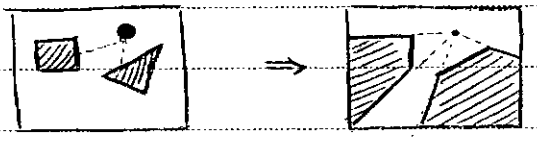
- موانع از پیش تعیین نشده و جابجا شونده در نقشه اولیه قرار گرفته نمی‌شود. در محیط تحت کنترل مانند کارخانه که موانع ثابت است مناسب است.
- ساختمان داده می‌تواند خطی باشد. ← ضربت: دقت بالا

• Cell Decomposition

- خود ربات تخاهد موانع را تشخیص دهد.
- اندازه سلول‌ها ← ثابت یا متغیر
- ساختمان داده می‌تواند کرایه درختی از ۰ و ۱ باشد.



☑ ساختن نقشه با بیش از ۳۶۰ درجه



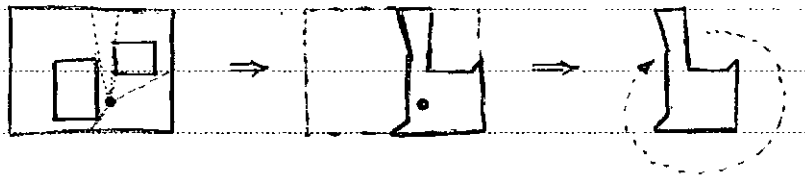
• پیدا کردن جای ربات در نقشه

Pattern Matching

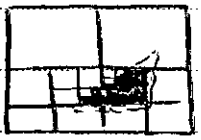
← بهترین نقشه‌ای که درآورده است، با کجای محیط مطابق است.

← چون orientation ربات معلوم نیست، باید اندازه نیز چرخاند

Subject:
Year. Month. Date. ( )



— می‌توان اندازه سلول‌ها را به طور تطبیقی مدوّن کرد.



← جایی که مانع نباشد سلول بزرگ
← جایی که مانع باشد سلول کوچک
← نقشه : Grid Map

• Topologic Map

— Landmark ← علامت مشخصه یک مکان

— بازنمایی با گراف ← یال گراف ← امکان رسیدن روبات به یک نود
نودهای گراف ← Landmark ٤

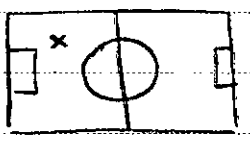
— اطلاعات کم می‌شود ← حجم کم ✓
شناسایی نودها سخت می‌شود ✓

* حال روبات را یک جای نقشه روشن می‌شود و یک چیزی را احساس می‌کند، حال روبات کجای نقشه قرار داریم؟
به این احساس روبات از موقعیت، باور یا Belief گفته می‌شود.

* Belief

• برد دینامیک، قطعی، احتمالاتی
• دقیقاً می‌داند که کجای نقشه قرار دارد. ← قطعی

☑ روبات Small size



جای روبات دقیقاً مشخص است.

- روبات احتمال می دهد که کجای نقشه قرار دارد. ← تک فرضیه ای / چند فرضیه ای
- روشهای احتمالاتی ← روش مارکوف / روش کالمن / روش مونت کارلو
- در خیلی از مسائل تخمین خطی کنترل نشده، اعلام موقعیت به صورت نقطه ای ممکن نیست. دلیل این امر عدم قطعیت سنسور است.
- در مکان یابی بر پایه احتمالات، موقعیت به صورت احتمالات بیان می شود و درصدی برای خطا نگاه داشته می شود. هدف این روش به حداقل رساندن عدم قطعیت سنسور است.

☑ یک روبات توسط سنسور مافوق را با sonar و فاصله ۵۰cm تشخیص می دهد. اگر بازه سنسور ۴ ~ 15 cm باشد این عدد تقریباً درست است ولی در مورد فاصله ۵۰۰ cm چه می توان گفت؟

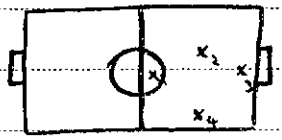
- هدف

Prob (obstacle | sense)

- قانون بیز

$$P(obs \mid s) = \frac{P(s \mid obs) \times P(obs)}{P(s)}$$

☑



→ P(داخل دروازه | s = بودن در وسط زمین)

=

$$\frac{P(s \mid \text{بودن در وسط زمین}) \times P(\text{بودن در وسط زمین})}{P(s)}$$

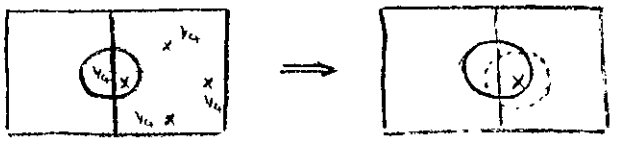
Subject :

Year . Month . Date . ( )

P (احتمال بودن در وسط زمین) = $1/4$ ⟶ x_i , $i = 1,2,3,4$

P (در وسط زمین | سنسور دایره پیدا کند) = 95 % ⟶ آزمایش

$P(S)$ ⟶ می‌دان از آن صرفنظر کرد و معمولاً سیگنال‌های نرمال گوسی به کار می‌رود.

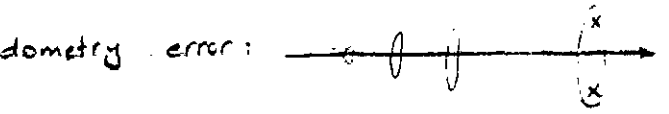


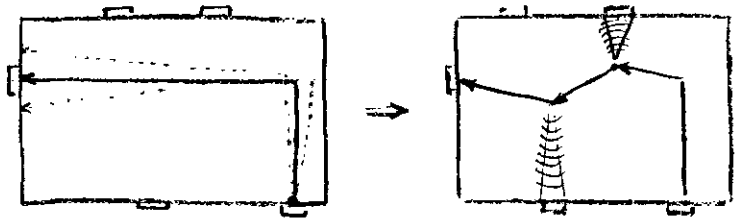
حال فرض می‌کنیم ربات در وسط زمین است، حالی که دوربین دایره احتمال بالایی دارد.
اگر ده نمونه حال، سنسور یکبار هم دایره ندید، احتمال که بالا رفته است، عدم قطعیت را جبرانی می‌کند

$P(S)$ ⟶ $P(S|obs) \times P(obs) + P(S|!obs) \times P(!obs)$

☑ روباتی مجهز به انکودر دوربین است. روبات حرکت می‌کند و پس از مدتی خطای جمع شده آن باعث می‌شود که متوجه باشد که روبات کجاست. فرض می‌شود که بینایی می‌دان درب آنها را تشخیص دهد.

odometry error :





* مکان یابی بر حسب نشانه احتمالاتی

Action update •

$S'_t = Act(O_t, S_{t-1})$

از موقعیت قبلی با استفاده از قرائت انکودر، موقعیت جدید رباط بدست می آوریم.

از سینماتیک رباط حاصل می شود

Perception update •

$S_t = Percept(i_t, S'_t)$

موقعیت دقیق فعلی را با استفاده از موقعیت غیرقطعی و حس محیط بدست می آوریم.

از Sensor Fusion حاصل می شود.

• روش مارکوف

— محیط را به موقعیتهای قابل تشخیص تقسیم می کند.

Grid ←

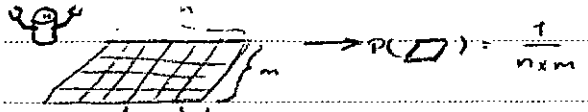
☑

فضای آزاد → 15 cm x 15 cm x 1 degree

رباط → 8 موقعیت در زمین

— برای هرکدام از موقعیتها یک احتمال در نظر گرفته می شود (با در چند فرضیه ای)

☑



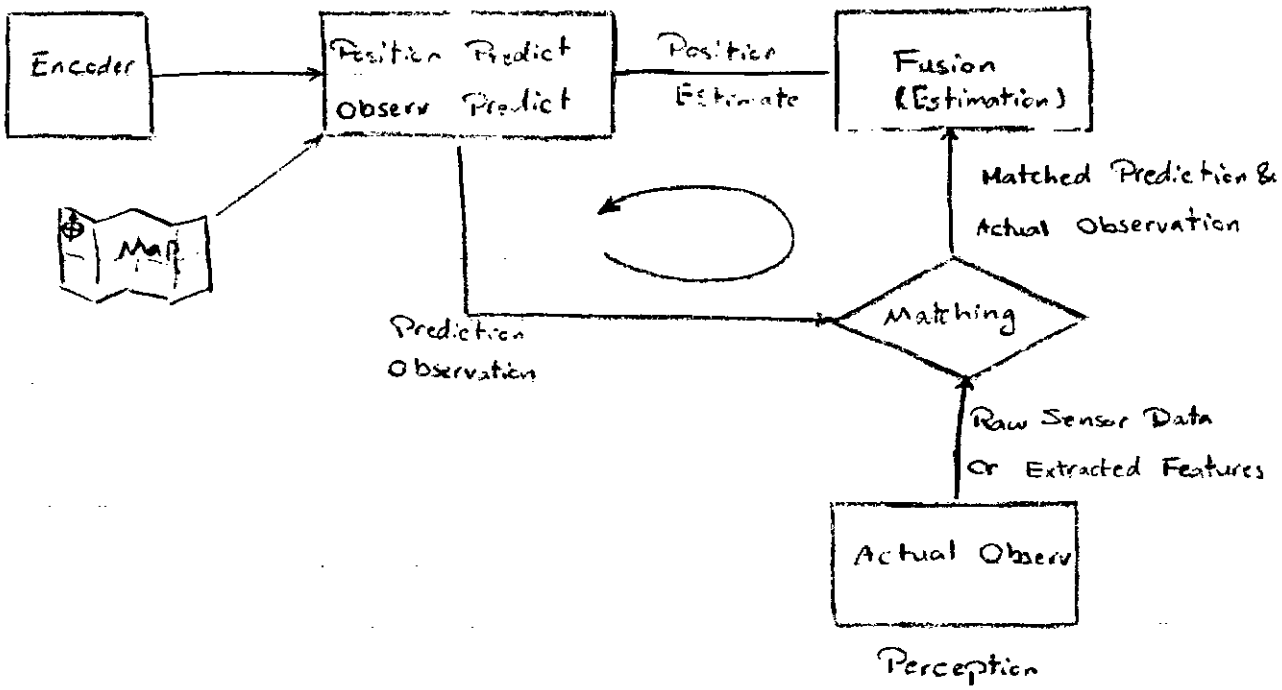
— رباط آنقدر حرکت می کند تا در جایی Perception بتواند تشخیص دهد.

— برای به روز کردن تابع Percept ← از Bayes

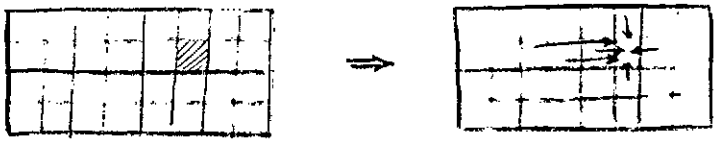
— برای به روز کردن تابع Act ← احتمال رسیدن به وضعیت فعلی از تمام وضعیتهای قبلی جمع می شود.

Subject :

Year . Month . Date . ( )



$$P(l_t \mid O_t) = \int P(l_t \mid l'_{t-1}, o_t) \cdot P(l'_{t-1})\, d\, l'_{t-1}$$



- روش مارکف

$P(n)$

۱ـ یک احتمال اولیه برای حضور در نقطه n

۲ـ شاهدهٔ i صورت می‌گیرد.

۳ـ از فرمول Bayes استفاده می‌شود (محاسبه likelihood سمبل تخمین)

$$P(n \mid i) = P(i \mid n) \times P(n)$$

۴ـ $P(i \mid n)$ آزمایش بدست می‌آید و یک جدول از قبل ذخیره شده است.

* فیلتر کالمن

- هدف: Sensor Fusion
- فرض می‌کنیم دو سنسور داریم که اندازه‌گیری‌های آن‌ها $\hat{q}_1$ و $\hat{q}_2$ می‌نامیم. رابطه بین دو اندازه‌گیری ثابت است. (استاتیک)
- * مبنای فیلتر کالمن: اندازه‌گیری‌ها دارای توزیع نرمال باشند.
- * پس اندازه‌گیری‌ها:

$$\hat{q}_1 = N(q_1, \sigma_1^2)$$
$$\hat{q}_2 = N(q_2, \sigma_2^2)$$

$S = \sum \omega_i (\hat{q} - q_i)^2$ ~ weighted least square

$$\frac{\partial S}{\partial \hat{q}} = 2 \sum \omega_i (\hat{q} - q_i) = 0$$

$$\hat{q} = \frac{\sum \omega_i q}{\sum \omega_i}$$ ~ weighted average

* انتخاب وزن

$$\omega_i = \frac{1}{\sigma_i^2}$$

$\uparrow \omega \leftarrow \downarrow \sigma \leftarrow \uparrow$ دقت

سنسور دقیق‌تر وزن بیشتری دارد.

$$\hat{q} = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} q_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} q_2 = q_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} (q_2 - q_1)$$

$$\hat{q} = N(q, \sigma_f^2) \longrightarrow \sigma_f^2 = \frac{\sigma_1^2 \cdot \sigma_2^2}{\sigma_1^2 + \sigma_2^2}$$

* رابطه کالمن

$\hat{q} = q_1 + K(q_2 - q_1)$ ⟶ K: ضریب کالمن

$$\hat{x}_{k+1} = \hat{x}_k + K_{k+1} (z_{k+1} - \hat{x}_k)$$

Subject:
Year. Month. Date. ( )

☑ لیدار سونار
☑ اندرد vision

x = encoder

z = vision

$\hat{x}_{k+1} = \hat{x}_k + K_{k+1} (z_{k+1} - \hat{x}_k)$

مرحله قبل → موقعیت نهایی اندرد ها
نهایی → موقعیت بدست آمده از بینایی
ضریب کالمن

موقعیت نهایی اندرد

روبات آندرد حرکت می‌کند تا vision بتواند landmark ای را بشناسد و مثلاً به یک بصد، آنگاه موقعیت اندرد بهبود و ساخته می‌شود

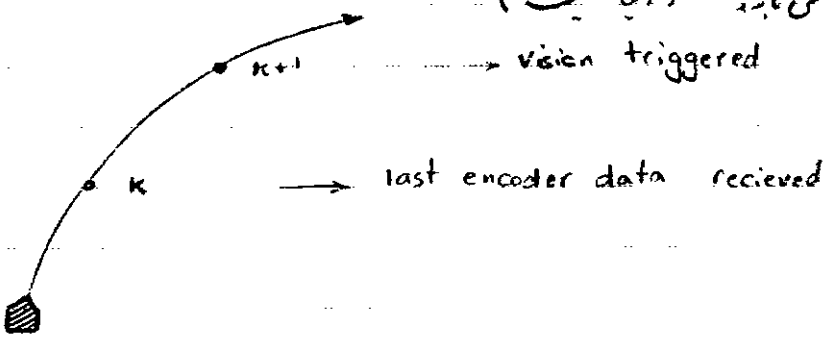
$$K_{k+1} = \frac{\sigma_k^2}{\sigma_k^2 + \sigma_z^2}$$

$$\begin{cases} \sigma_k^2 = \sigma_1^2 \\ \sigma_z^2 = \sigma_2^2 \end{cases}$$

$$\sigma_{k+1}^2 = \sigma_k^2 - K_{k+1}\sigma_k^2$$

ابهام معیار اندرد به تدریج کم می‌شود.
ابهام معیار بینایی می‌تواند متغیر باشد.

• اگر روبات در حال حرکت با سرعت v باشد و نیز ω را دارد. بین دو اندازه‌گیری روبات در حال حرکت می‌باشد (دینامیک)



$$\hat{x}_{k'} = \hat{x}_k + u\,(t_{k+1} - t_k)$$

$$\hat{x}_{k+1} = \hat{x}_{k'} + K_{k+1}\,(z_{k+1} - \hat{x}_{k'})$$

$$\sigma_{k'}^2 = \sigma_k^2 + \sigma_\omega^2\,(t_{k+1} - t_k) \Longrightarrow K_{k+1} = \frac{\sigma_{k'}^2}{\sigma_{k'}^2 + \sigma_z^2}$$

vision + laser scanner ☑

دوربین ← موقعیت
لیزر ← خطا
تشخیص

position belief ~ Normally distributed

R : Robot Frame

S : Sensor Frame

W : World Frame

$u(t)$ ~ known

اطلاعات انکودر (موقعیت ربات) update می‌شود.

$$p = [x \;\; y \;\; \theta]^T$$

$$p' = p + [\Delta x \;\; \Delta y \;\; \Delta\theta]^T$$

تصحیح شده توسط چرخ

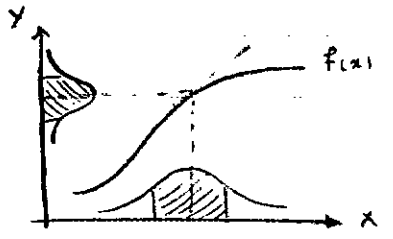
$$\Delta s = \frac{\Delta s_r + \Delta s_l}{2}$$

$$\Delta x = \Delta s \cdot \cos\left(\theta + \frac{\Delta\theta}{2}\right)$$

$$\Delta y = \Delta s \cdot \sin\left(\theta + \frac{\Delta\theta}{2}\right)$$

$$\Delta\theta = \frac{\Delta s_r - \Delta s_l}{b}$$ ← تصحیح چرخ

Subject:
Year. Month. Date. ( )



نحوه تاثیر انتشار خطا

$$C_Y = F_X \cdot C_X \cdot F_X^T$$

$$F_X = \nabla f = \left[\nabla_x \cdot f_{(x)}^T \right]^T$$

$$= \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots \\ \vdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

odometry

$$\hat{P}_{k+1} = f(\hat{P}_k, u_k)$$

$$\Sigma_{P(k+1)} = \nabla_P f \cdot \Sigma_{P(k)} \cdot \nabla_P f^T + \nabla_u f \cdot \Sigma_{u(k)} \cdot \nabla_u f^T$$

مدل خطا برای odometry

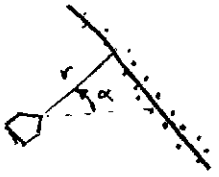
$$\Sigma_\Delta = \begin{bmatrix} k_r |\Delta s_r| & 0 \\ 0 & k_\theta |\Delta s_\theta| \end{bmatrix}$$

هرچه حرکت بیشتر باشد، خطای آن چرخ بیشتر است.

فرق فیلتر کالمن با مدل مارکوف این است که فیلتر کالمن نیاز به موقعیت اولیه دارد.

اطلاعات بدست آمده از لیزر:

$$z_{k+1} = \begin{bmatrix} \alpha_{k+1} \\ r_{k+1} \end{bmatrix}$$



$\Sigma_k \sim$ آزمایش

اطلاعات بدست آمده از نقشه:

از موقعیت قبلی ربات استفاده می‌کند: $\bar{P}(k+1)$

با استفاده از این موقعیت تعدادی از خطوط حدس می‌شوند چون ربات نمی‌داند آنها را ببیند

آنها را با h نشان می‌دهد. بدین صورت ربات تخمینی از درودهای احتمالی لیزر را

بدست می‌آورد که آنها را با z_k نشان می‌دهد.

پس:

$x \longrightarrow$ ذرات $\longrightarrow$ موقعیت (P)

$z_t \longrightarrow$ خطاطی که لیزری تواند اندازه بگیرد (در لحظات مرجع) $\longrightarrow$ ذرات + تشت $\xrightarrow{h(x)}$ z_i

$z_i \longrightarrow$ خطاطی که لیزر اندازه می‌گیرد (در لحظات ربات) $\longrightarrow$ لیزر

وجود موانع که در تشت لحاظ نشده است، خود را در z_k نشان می‌دهد.

برای همین برای ایمنی ربات بهتر است که قرار دارد باید بسیار چیزی با تشت تفاوت دارد. این مقدار innovation نشان می‌شود.

$$v = z_i - z_j$$

حال با آستانه‌گیری می‌توان تصمیم را انجام داد.

$$P(k+1) = P(k) + K(k+1) \cdot V(k+1)$$

Subject:

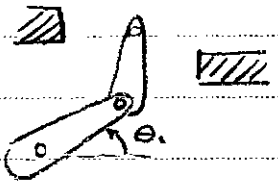
Year. Month. Date. ( )

Planning

Sensor → Perception → Localization → Planning

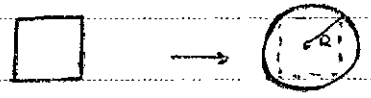
• Configuration Space

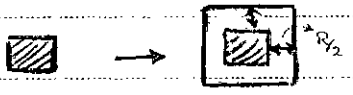
به دلیل وجود موانع ، برای هر متغیر یک بعد ، متغیر ایجاد و یا محدودی می شود .



برای θ_1 ، متغیر θ_2 محدود است .

• حرکت ربات متحرک ساده سازی به این صورت می گیرد .

Robot : 

Obstacles : 

به عنوان مثال ، جمع Minkovski

Direction : Non-Holonomic → Omni-Directional

• روش گراف پیداری (Visibility Graph)

* گره ها ← نقطه شروع
نقطه هدف
رئوس چند ضلعی ها

* یالها ← خطوط مستقیم انتقال دهنده دو نقطه
یالها نباید موانع را قطع کنند .

* جستجوی کوتاهترین مسیر در گراف

* ادراکات ← حرکت خاص با موانع ← برخورد و دارش Error
  ← برای جلوگیری از برخورد ← شعاع بزرگتر ← نقاط غیرقابل دسترس
  ← مسیرهای غیرقابل عبور
  ← مربع محاطی ← بدبادن فضا
  ← وابستگی شدید به اندازه گیری های موانع
* ادراک ← Reduced visibility G ← دوتن متر حذف می شود
  ← بازهای non-tangant حذف می شود.

• روش دیاگرام ورونوی (Voronoi Diagram)
* دقیقاً مخالف روش قبل، سعی می کند ربات را از جایی عبور دهد که بیشترین فاصله را از موانع داشته باشد.

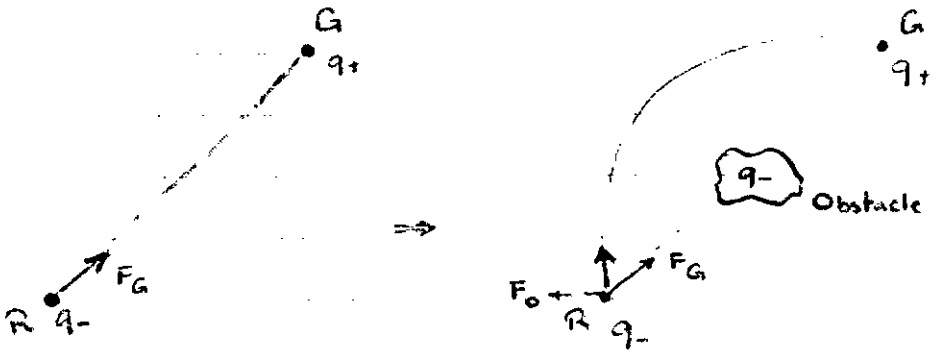
* عملی تر است ← می تواند با اندازه گیری online صورت بگیرد
  ← ربات احتمال برخورد کمتری دارد.

* رسم دیاگرام برای چند نقطه ← عمودمنصف نقطه ها به طور دوبه دو
  ← در محل تقاطع، استاد عمودمنصف ها دور ریخته می شود.

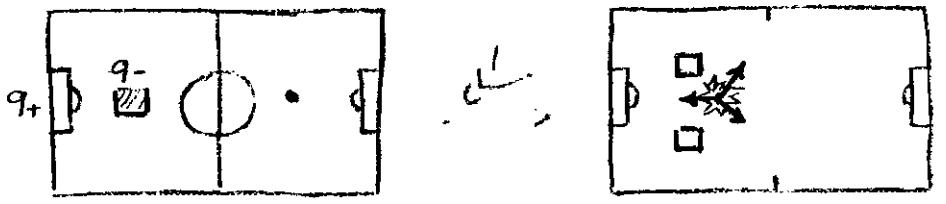
* تعداد voronoi ← رئوس: یک خط
  ← خط: یک خط
  ← یک نقطه و یک خط: یک سهمی

• روش میدان پتانسیل (Potential Field)
* در اطراف هدف میدان تولید می شود که به ربات نیروی جاذبه وارد می شود. اگر مانعی در جایی وجود داشته باشد، با ربات تماس یابد، آن را دفع می کند، نیروی وارد شده برآیند این نیروها است.

Subject:
Year. Month. Date.



☑ طراحی مسیر



برای ربات در میدان محلی

* مزایا ← ساده Smooth
نقشه از قبل آماده شده نیاز خواهد.
* انتخاب میدان ← هدف: میدان جذبی

P_g: فاصله تا هدف

$U_{att} = \frac{1}{2} k . P_g^2$

$P_g \uparrow \Rightarrow U \uparrow$

$\vec{F}_{att} = -\nabla U_{att}$

مانع: میدان دافعی

P_q: فاصله تا مانع

$$U_{rep} = \begin{cases} 0 & P_q \geq P_o \\ \frac{1}{2} k' \left(\frac{1}{P_q} - \frac{1}{P_o} \right)^2 & P_q < P_o \end{cases}$$

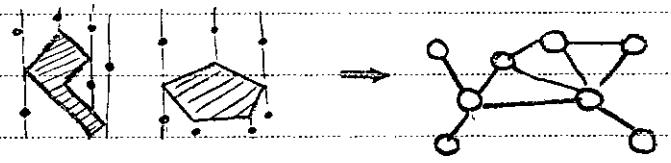
P_o: شعاع اثر دافعی

Subject:
Year. Month. Date. 98

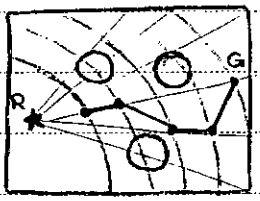
* معایب ← گرانتادن در سیستم محلی
نزدیک دیوار می تواند حرکت کند ← Rotational Retential Field (اصلاح)

• روش تجزیه سلولی (Cell Decomposition)
* سلول بندی فضای کاری ← فضای آزاد
فضای اشغال

* سپس تولید گراف بین سلول ها ← گره ها : سلول ها
یال : نقاط متصل کننده وسط مرز سلول ها



small size برای روش سلولی



* می توان با سنسور، محیط را نرم کرد.

Approximate ← سلول نوچه ← محاسبات بالا
دقت بالا

Adaptive ← سلول تطبیقی

* برای پیدا کردن کوتاهترین مسیر از Wave front Planner استفاده می شود.

PqPCO ← سلول استاندارد

هدف برنامه نوع مرج است حتما

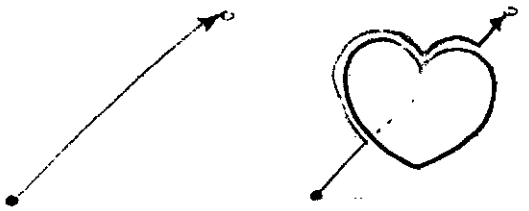
Subject :

Year . Month . Date . ( )

Obstacle Avoidance ★

- Bug 2

ربات محیط مانع را دور می‌زند و به محض اینکه راهی به سوی هدف پیدا کرد از آن جدا می‌شود.



- Bug 1

در این روش، ربات یک دور کامل دور شیء زده و سپس از نزدیک‌ترین نقطه به هدف از مانع جدا می‌شود

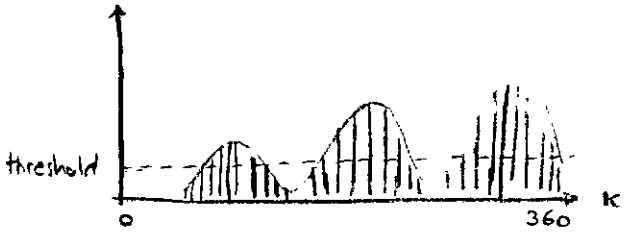
- (VFH) Vector Field Histogram

★ بلا درنگ

★ محیط را به سلول‌های تجزیه می‌کند.

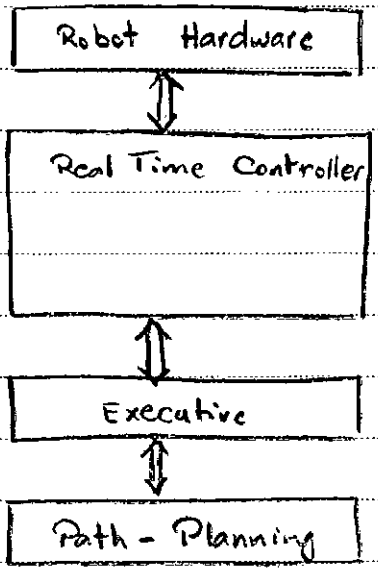
★ با استفاده از حسگر حوزه، به هر سلول مانع میزانی را تخصیص می‌دهد.

★ نمودار هیستوگرام قطبی استعاری را تشکیل می‌دهد.



★ تعداد دره ٤ = تعداد مسیرهای آزاد

* معماری نرم افزار روبات



o انواع معماری

1. Deliberative

Look → Think → Plan → Act

نمونه ساده

2. Reactive

No Look → React

نمونه بدون برنامه ریزی

3. Hybrid

{ Deliberative → Think Slowly
{ Reactive → React Quickly

4. Behavior Based

Finite State Machine → states
→ events

Subject:

Year Month Date ( )