

PID 制御

P: Proportional	比例
I: Integral	積分
D: Differential	微分

 $\Leftrightarrow$ On/off 制御

操作量 = $K_p \times \text{偏差} + K_i \times \text{偏差の累積値} + K_d \times \text{前回偏差との差}$

$$MV_n = MV_{n-1} + \Delta MV_n$$

$$\Delta MV_n = K_p \cdot (e_n - e_{n-1}) + K_i \cdot e_n + K_d ((e_n - e_{n-1}) - (e_{n-1} - e_{n-2}))$$

MV_n : 操作量
 ΔMV_n : 操作量差分
 e_n : 偏差

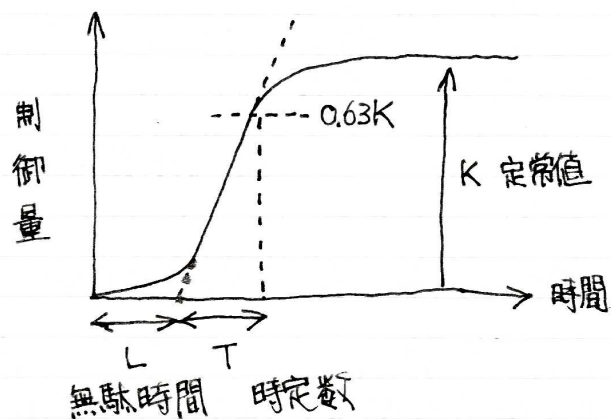
SV: Set Point Variable	目標値
PV: Process Variable	測定値
MV: Manipulated Variable	操作量

ステップ応答法	step response method
限界感度法	ultimate sensitivity method

限界感度法

操作量に飽和があると比例ゲイン K_p を大きくしても
 正確に安定限界を求めることができない。

ステップ応答法 (インディシャル応答法)



制御量の例
・モーター速度
・水温

	K_p	K_i	K_d
P制御	$0.3 \sim 0.7 T/KL$	0	0
PI制御	$0.35 \sim 0.6 T/KL$	$0.3 \sim 0.6 / KL$	0
PID制御	$0.6 \sim 0.95 T/KL$	$0.6 \sim 0.7 / KL$	$0.3 \sim 0.45 T/K$

Chien Hrones Reswick の最適値設定推奨値

	P	I	D
P	$333 K_p \cdot L/T$		
I	$286 K_p \cdot L/T$	$1.2 T$	
D	$77 K_p \cdot L/T$	T	$0.5 L$

Ziegler Nichols (25%減衰) 最適値設定推奨値

	P	I	D
P	$100 K_p \cdot L/T$		
I	$110 K_p \cdot L/T$	$3.3 L$	
D	$83 K_p \cdot L/T$	$2 L$	$0.5 L$

速度形デジタル演算式

$$\Delta MV_n = K_p \left\{ (e_n - e_{n-1}) + \frac{\Delta t}{T_i} e_n + \frac{T_D}{\Delta t} (e_n - 2e_{n-1} + e_{n-2}) \right\}$$

車輪直径 81.6 mm
 " 円周 256.4 mm
 " 間隔 63.0 mm
 ロータリエンコーダ $360^\circ = 360$ カウント

無負荷の最大スピード
 875 ~ 900 カウント/秒
 約 3.5 カウント/4msec

ハードウェア 時間

・ A/D コンバータ 変換時間 3 ミリ秒
 ・ メイン MPU (ARM) ⇄ サブ MPU (AVR)

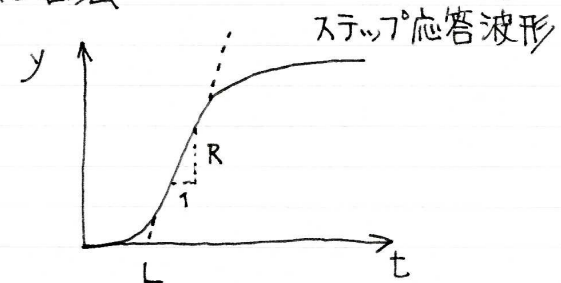
「 」 「 」 「 」 「 」

各々 2 ミリ秒 周期

1 ミリ秒毎に交互に 送信と受信を 繰り返す

モーター		光センサー		位置	バッテリー	前面
TACHO	PWM	PTn	LED	センサー	電圧	ボタンSW
↓	↑	↓	↑	↓	↓	↓
ARM	AVR	AVR	AVR	AVR	AVR	AVR
						(錠 ON/OFF)

過渡応答法



	比例ゲイン	積分時間	微分時間
	K_p	T_I	T_D
P	$1/RL$	-	-
PI	$0.9/RL$	$3.33L$	-
PID	$1.2/RL$	$2L$	$0.5L$

$$\begin{aligned} \text{伝達関数 } G_c(s) &= K_p + K_I \frac{1}{s} + K_D \cdot s \\ &= K_p \left(1 + \frac{1}{T_I \cdot s} + T_D \cdot s \right) \end{aligned}$$

$$K_I \equiv \frac{K_p}{T_I}, \quad K_D \equiv K_p \cdot T_D$$

言語 リンケージ

「C++ランゲージ ワイクリファレンス」 p.27

- ・ "C++" (デフォルト)
- ・ "C"

★ C++プログラム $\xrightarrow{\text{関数コール}}$ Cプログラム
"C" リンケージ

```
(例) extern "C" void cfunc(int);  
  
extern "C" {  
    void cfoo(int);  
    void cbar(char *);  
}
```

★ Cプログラム $\xrightarrow{\text{関数コイル}}$ C++プログラム
"C"リンク-3"

```
(14) extern "C" void cppfunc(int);  
  
extern "C" {  
    void    cppfoo(int);  
    void    cppbar(char *);  
}
```

Q. C++ の全ての関数も "C" リンケージにすれば?

A. オーバーロードが出来ない (にともない)

⇒ 特定の名称の "C" リンケージを持つ関数は 1つ
複数の多重定義のうち1つだけ

Linker (リンカー)

- ・ リンケージ エディター
- ・ 関係編集 プログラム

★ TOPPERS/JSP の言語リンケージ

kernel.h 不要 コード内に #ifndef __cplusplus が
埋め込まれているから

kernel_id.h 不要 ID番号の定義のみだから

タスクのハンドラ 必要

性能評価用システム時刻取得関数

```
SYSCALL ER vxget_tim (SYSUTIM *p_sysutim)
```

toppers.jsp ¥ config ¥ armv4 ¥ at91sam7s ¥ sysconfig.h
67行目に おいて
#define SUPPORT_YXGET_TIM
が 無効化されている。

toppers_ jsp%config%army4%at91sam7s%hw-timer.h
の 中 の

hw_timer_get_current() と hw_timer_fetch_interrupt() は ELW、MMRE 772227118。 (P55<E5<E5<

航空用語

航法 navigation

1. 地文航法 pilotage navigation
パイロットが地上の海岸線や鉄道線路と見ながら飛行する方法
2. 推測航法 dead reckoning navigation
既知の地点からの方位と距離を計算し、
ここに風向と風速を加味して、機首の
向きと到達時間を推測する方法
3. 慣性航法 inertial navigation
加速度を積分することで速度を求め
速度を積分することで移動距離を求める
加速度(慣性)を利用した方法

機首方位 heading

北を基準とした機首の方向
北を 0° として時計回りに 360° までの角度で表す

相対方位 relative bearing

航空機の機軸(前後軸)を基準とした
外界対象物との方位
機首方向を 0° とし左右に 180° まで
あるいは時計回りに 360° までの角度で表す

ITRON の優先度

1 から連続した正の値
値が小さいほど優先して処理される
= 優先度が高い

スタンダード・プロファイル

	1	~	16
QoS 優先度	TMIN-TPRI	~	TMAX-TPRI
メッセージ優先度	TMIN-TPRI	~	TMAX-HPRI

相対時間 (RELTIM 型)

一般的な実装においては、相対時間に 1 が
指定されると、サービスコールが呼び出されてから
2 回目のタイムティックでイベント処理を行う。

また、相対時間に 0 が指定されると
サービスコールが呼び出された後の最初のタイムティックで
イベント処理を行う。

TOPPEPS / JSP の C++ 対応

...toppers_cxxrt - reset - specific (wid)

JSP 1.4 との互換性のために空の関数が残っているだけ

ITRON 初期化ルーチンの追加 (静的 API)
ATT_INI (ATR iniatr, 属性
VP_INI exinf, 拡張情報
FP inirtn); 起動番地

システム時刻の参照

get_tim()

非タスクコンテキストでの使用不可

スケッチ プログラミング言語 Processing

LOGO ライクなサンプルコード

```
size (120, 120) // 描画画面サイズ
translate (50, 50) // 座標移動
for (int i = 0; i < 16; i++) {
  int x = 20;
  int y = 0;
  line (0, 0, x, y); // (0, 0) → (20, 0) 直線
  translate (x, y); // 終点へ座標移動
  rotate (2 * PI / 16.0); // 座標回転
}
```

systick_wait_ms() 関数 <systick.h>

使用注意!

実装は シンプルな whileループ

- ・ サンプルコード
- ・ テストコード
- ・ LCD 初期化コード
- ・ Bluetooth 初期化コード (含. コネクション確立)

以上のコードを除くと 2ms 以内の待ち処理にしか
使用していない。

⇒ systick_wait_ms() の修正は 不要と判断

ウォッチ・ドッグ・タイマ (WDT)

AT91 ARM Thumb Microcontrollers

Application Note (doc6293.pdf)

Page6 3.2.1.7 Low-Level Initialization: Watchdog

AT91C_BASE_WDTC → WDTC_WDMR

= AT91C_WDTC_WDDIS;

Mindstorms NXT 基本ファームウェア

AT91SAM7S256 ¥ SAM7S256 ¥ Include ¥

Cstartup-SAM7.c

AT91F_LowLevelInit 関数内

line: 43

// watchdog Disable

AT91C_BASE_WDTC → WDTC_WDMR = AT91C_WDTC_WDDIS;

東京地区 技術教育 1

大会ルール Q&A

Q. ゴール / 中間ポスト の 判定基準

A. 走行体の 一部でも 越えれば OK

Q. ゴール 停止 の 判断 (NXT の場合)

A. 倒れた 場合は NG

ゴールラインから オーバーラインの手前でぶつくるのはOK

但し、並走車にぶつかってはダメ

NXT スタート時の注意

もし、少しでも 動く 場合は 手で押えて止めて下さい

2009.5.29 参加者ML 本部技術委員 西川さん

モーターユニット

入力 → AVR PWM
出力 ← ARM7 INT (割込み)

NXT コネクタ 端子

1 入力 → AVR (A/D) 4 電源 ← NXT
2 GND 5 出力 0 ← ARM
3 GND 6 出力 1 ← ARM (未使用)

I/O ポート 結線

ポート名	ピン名	ピン番号	結線名
PORT A	PA23	15	DIGIA0
PORT B	PA28	38	DIGIB0
PORT C	PA29	41	DIGIC0
PORT D	PA30	42	DIGID0

NXT way-ET ポート 配線

PORT	My Robot	Sample
1	(NC)	TOUCH SW
2	TOUCH SW	(NC)
3	LIGHT SEN	←
4	GYRO SEN	←
A	(NC)	←
B	MOTOR RIGHT	←
C	MOTOR LEFT	←

光センサー 外乱光影響測定

2009.5.31
battery = 82

	屋内 蛍光灯のみ	屋内 外光あり	屋外 曇天
WHITE	541	523	395
GRAY	611	631	415
BLACK	728	700	461

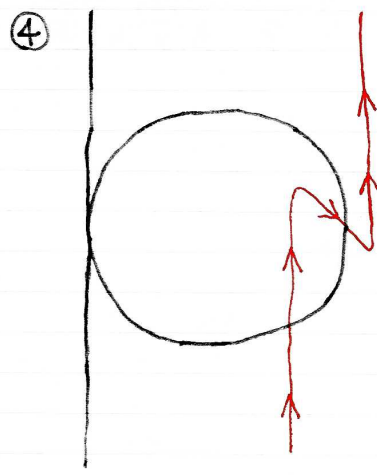
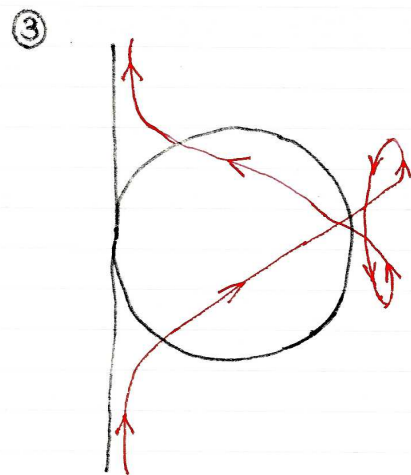
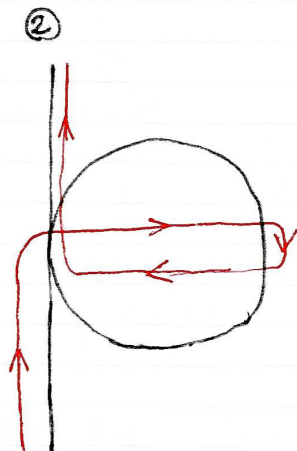
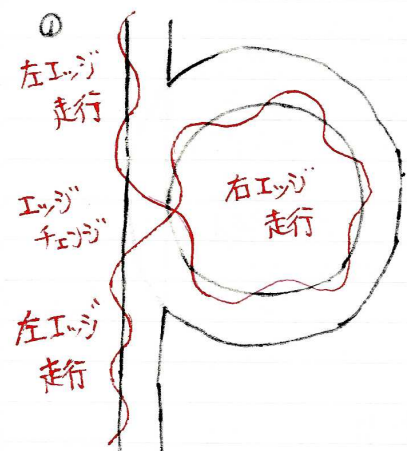
光センサー 直射日光影響測定

2009.6.8 晴天

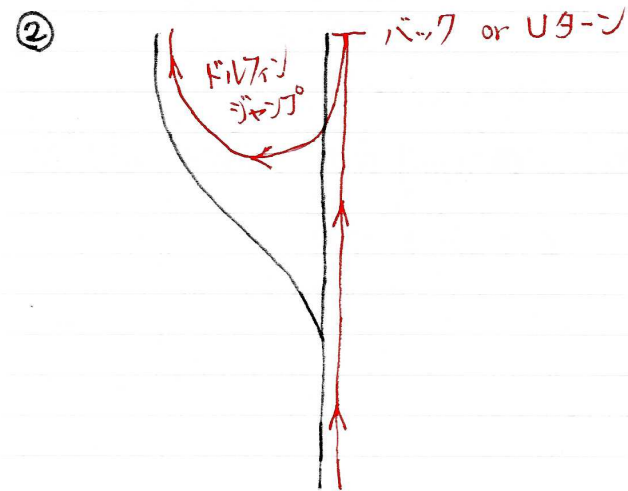
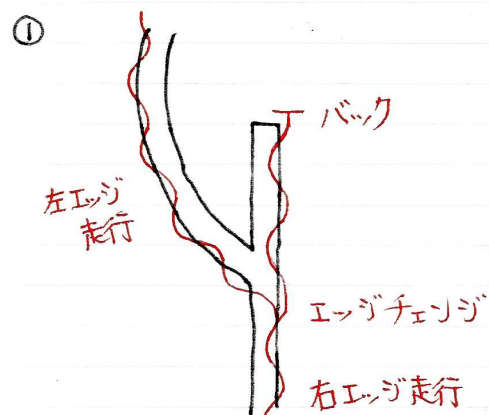
	センサーケース あり	なし
影響小	880	826
“大	838	565

* LED 消灯 状態
受光素子は黒ラシヤ紙で完全にふさぐ

ツインループ 攻略コース



トレジャーハント 攻略コース



光センサー 非線形補正式

$$f(x) = 4 \times 10^{-10} \times x^{4.0876}$$

x : LED消灯時のセンサー値

整数演算による高速化

$$P = \frac{2^5}{L_D}$$

L_D : ダーク値 (LED消灯時のセンサー値)

P が 8ビット長 (256) の範囲に入るためには

$$L_D > 128$$

でなければならぬ

$$\text{非線形補正值 } P = \underbrace{P \times P \times P \times P}_{\text{Max} = 2^{36}} / 2^{16}$$

$$(想定) \quad 43 \leq P \leq 2153 \quad \downarrow \quad 65536$$

$$\text{フロア値 } F = (L_D - L_B) \times P / 2^{16}$$

L_B : ブライト値 (LED点灯時のセンサー値)

$$(想定) \quad 0 \leq F \leq 160$$

光センサー 明滅変化測定

time tick	BLACK		time tick	WHITE	
	LightON	LightOFF		LightON	LightOFF
3872	683	715	4800	531	661
13472	499	501	14944	409	435
24320	438	436	23008	363	377
33088	414	414	31968	308	316
40448	608	618	41504	497	565

2009. 10. 18

浮動小数点演算を活用した 2次多項式補正 (近似)

$$\text{lightDiff} \simeq a \times (\text{lightDowned})^2 + b \times \text{lightDowned} \times c$$

lightDiff: LED消灯と点灯の差分量

lightDowned: LED消灯時のセンサー値

a : 2次補正係数

b : 1次補正係数

c : 切り片

窓からの距離で外乱光を変えた実験データより

$$a = 1.0382 \times 10^{-3}$$

$$b = -6.3295 \times 10^{-1}$$

$$c = 1.1024 \times 10^2$$

第8回 ミーティング

出席: 吉田

場所: A会議室

日時: 7/7 (火) 18:00 ~ 20:00

試走会1に向けて取り組む課題

- ・ センサのロックインアンプ及び走行への影響
- ・ シンプルなライントレース時のタコタのログ収集
- ・ 走行距離測定
 - ・ コース上の距離と(左右にふらつきながらの)測定距離の差が、どの程度か?
- ・ エッジチェンジの実現 For ツインループ攻略
- ・ 走行制御クラス(モジュール)へ渡すパラメータ
 - ・ エッジ
 - ・ 速度

その他

試走会1前のミーティング

7/20 酒の日

試走会1の申込日

7/25 (土)

有力

7/26 (日)

バッテリー電圧の取得(更新)

check - NXT - buttons ()

↓

ecrobot - poll - nextstate ()

↓

battery - voltage ()

↓

io - from - avr (AVRからのメッセージ受信バッファ)

ecrobot - get - voltage 関数に上記のプロセッサ
(定期的に)更新される ecrobot - inputs.battery - state
を参照している

モーター回転速度 計算方法

32 ミリ秒 (8 tick) 間隔で回転角の差分を計算する

間隔: 小 のとき、回転角の変化が小さい (0 or 1)

間隔: 大 のとき、回転速度計算に過剰が生じる

走行中のバッテリー電圧変動

全カ前後進 (20090714_185203.csv)

MIN 7643 mV

MAX 8225 mV

ΔV 582 mV

7.1%

ライントレース (20090714_183431.csv)

MIN 7768 mV

MAX 8211 mV

ΔV 443 mV

5.4%

実験ログ

20090714_181445.csv

ライトレース

20090714_183431.csv

ライトレース → 6秒後 後進

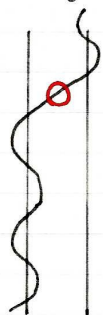
20090714_185203.csv

2秒おきに 全カ(±100)で 前進と後進

20090714_193417.csv

5カウト(約3.6cm)範囲内へ 前進

Edge Change の条件



	ライン 進入中	ライン 脱出中
ライン内	○	×
ライン外	×	×

ログ 20090714_181445.csv 解析結果

首振り周期 332ミリ秒 ~ 744ミリ秒

試走会1 直前特訓

7月20日(月) 海の日

102会議室

吉田

10:20 ~ 17:45

1. 7月26日(日) 試走会1

試走会 西新宿 TDI 12:00 ~ 14:00

行き あずさ8号 塩尻 9:00 → 新宿 11:36

⇒ 指定席手配 谷口さん

帰り 新宿 15:00 → 松本 17:48

“ “ 16:00 “ “ 18:34

《持ち物》

公式電池、ロボット、PC

Eclipse コードネーム

バージョン	コードネーム	リリース日
3.2	Callisto	2006/6/30 *
3.3	Europa	2007/6/29
3.4	Ganymede	2008/6/25
3.5	Galileo	2009/6/24

* 2009/1 現在 NXTOsek の動作確認
Eclipse 3.2 + CDT 3.1.1

GNU ARM Eclipse Plug-in

インストールの注意

Group items by Category のチェックを外す

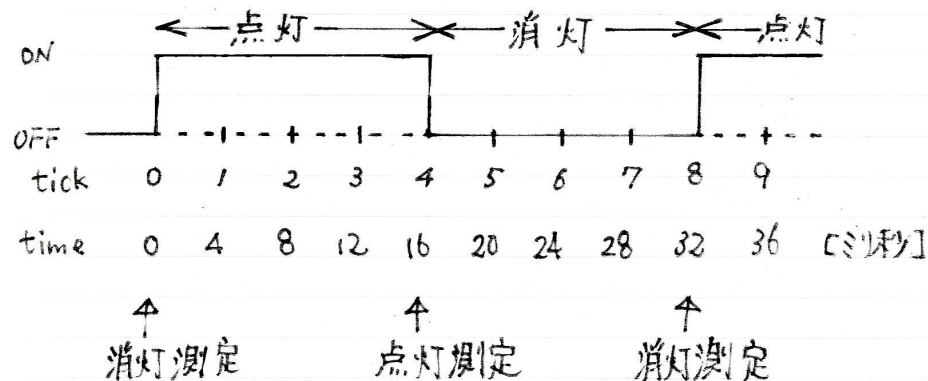
Eclipse のアンインストール方法

ディレクトリ (例 %Program Files\Eclipse) をと
削除で OK

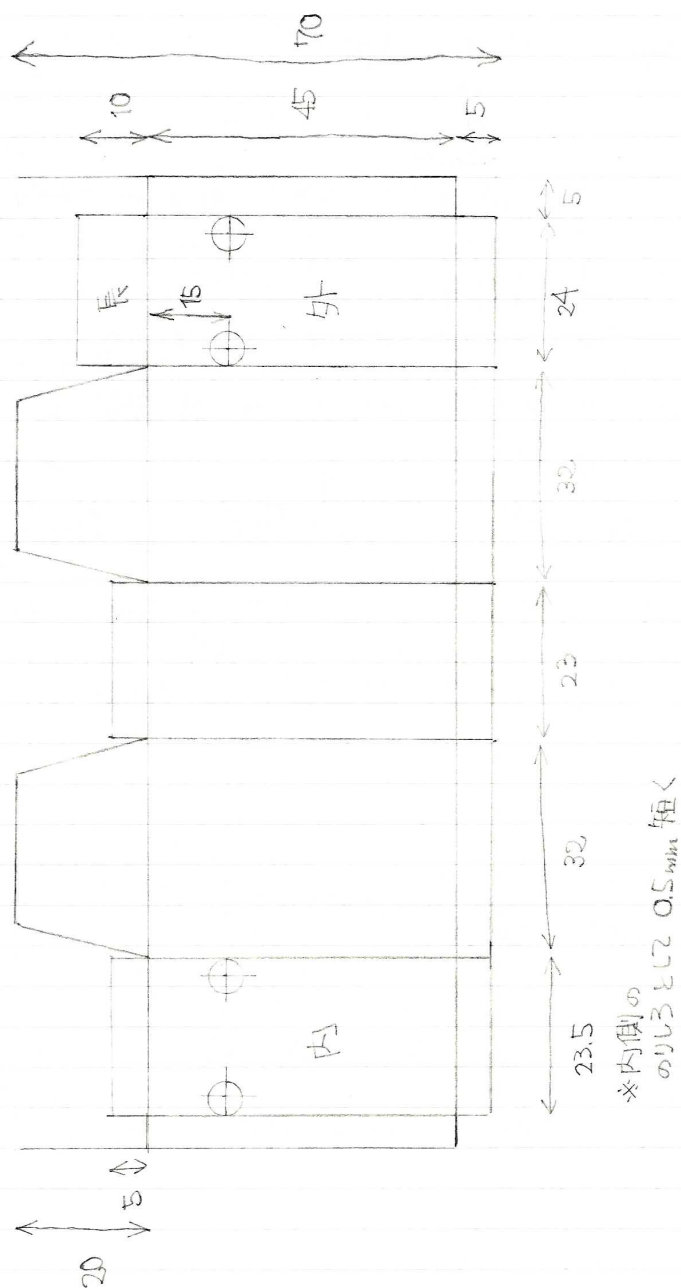
%cygwin%bin のパスの通し方

Properties → C/C++ Build → Environment
Variable PATH を追加して編集

ロックインアソプ タイミング



遮光ケース



モータ-PWM制御量

balance-control 周数 最大制御量(±100)
のとき, pwm量は±90の範囲内

20090724-212802.csv

倒立制御中のライン測定

	白	黒	白	グレー
通常方式	556	746	543	638
ロックインアンプ方式 ×128	70	26	70	52

電池交換

新品エボルタ

20090726-120853.csv

S

20090726-134103.csv

リチウムイオン充電電池

20090726-134140.csv

S

20090726-134954.csv

試走会1 7/26

ライン測定

	黒	グレー	白
通常方式	730~734	647~650	642~652
ロックインアンプ方式 ×128	21 2688	42~44 5504	69~91 10240

ライトレース

スレッシュホールド	64.7 (7000)
トレース振幅	28
	85

EVOLTA (アルカリ電池) 電圧変化測定

直線コース首振り周期 約 700 ミリ秒
※ 試走会データ 20090726-134954.csv

★ 2009年7月26日(日) 西新宿 TDI 試走会ログ ★

20090726-120853.csv 黒ライン測定のための直進前進&後退

20090726-121606.csv 灰(2-カ)測定のための

20090726-122507.csv 9秒間ラインを見失い放浪. ジャイロと電圧の両方ない

20090726-123503.csv 6秒間ライントレス後コースアウト. ジャイロと電圧両方ない

※ ここまで Data1=move, Data2=turn / ここから Data1=ロックインアンプ

20090726-124530.csv 10秒間ライントレス後コースアウト. ジャイロと電圧両方ない

20090726-125046.csv 7秒間ライントレス後コースアウト.

20090726-132918.csv 13秒間ライントレス後坂道頂上でコースアウト

20090726-133117.csv 70秒間ライントレス後 突然転倒

20090726-133529.csv 30秒間ライントレス

20090726-133638.csv 0.4秒後に放浪

20090726-134103.csv 2秒後に放浪

電池交換 新品 EVOLTA ⇒ Li-ion

20090726-134440.csv インコース 115秒で一周回成巧

20090726-134955.csv エッジ設定ミスで 中断

20090726-134954.csv アウトコース ほぼ一周回. 坂道入口でコースアウト

第10回ミ-ティング

★ 吉田

7月28日(火) 19:00~21:00

カスト 店

《 試走会の結果 》

・ 走行距離の計算精度は高い

1周 20m で ±5cm 以内

・ 方位の計算精度は低い

1周で 90°位ずれる?

・ ライントレスの首振り周期

500ミリ秒, 吉田 700ミリ秒

・ 後退しながらのライントレスは困難

・ 一旦、コースラインを外れ2から 直角(90°)で再侵入は困難

1. ライントレス時の首振り角度が大きすぎる
2. ツインループ(曲線区間)でのエッジング成功率が低い
3. 坂道の入口, 頂上, 出口でラインを見失う
4. コースのつながり目でラインを見失う
5. 曲線の内側エッジは曲率が大きいため曲り切れないうれ
6. 新品電池で不安定になる
7. ロボットの方位計算誤差大

《 提出モデル目次 》

1. ユースケース図, 分析クラス, オブジェクト図
2. 状態遷移図, シーケンス図 (RhapsodyにUMLモデル・テスト)
3. 要素技術 (ロックインアンプ, 電池)
4. 難所攻略法 (ツインループ, トレジャーハント)
5. Rhapsodyによる自動コード生成

7月30日(木) ~ 8月4日(火) 休暇

8月7日(土) or 8日(日) に 合宿

Dead Reckoning 法 誤差

本番コース走行ログドリ

- ・一周周回で Y 方向 400mm 程度
- ・積算 (演算) 周期

4msec と 32msec で ほぼ差なし (2mm 程度)

課題

- ・ ARM での 三角関数 (sin, cos), 浮動小数点演算
- ・ 試走会で 首振り が 小さかった 訳

制御量 上下制限 (飽和) 演算

(上限) 制御量 $\leftarrow$ 最小値 (制御量 or 上限値)
 $Mv = std :: \min (Mv, upper_limit)$

(下限) 制御量 $\leftarrow$ 最大値 (制御量 or 下限値)
 $Mv = std :: \max (Mv, lower_limit)$

【キーワード】

Odometry: オドメトリ

Dead-reckoning: デッドレコニング

自己位置推定

自己位置姿勢 $P(t) = (x, y, \theta)$

accumulated: 蓄積された

移動ロボット: Mobile Robot

カルマンフィルター: Kalman Filter

estimation: 推定

Travel Distance

Yaw Angle ϕ : 車体向き

actual: 現実の, 実際の

モーター反転時 バッテリー電圧変動測定

(ログ)

20090801_082849.csv	両輪反転	EVOLTA
20090801_094709.csv	" "	Li-ion
20090801_102141.csv	片輪反転	Li-ion
20090801_111158.csv	" "	EVOLTA

(測定条件) ・ pwm 常に 70
 ・ 500ms 正転 $\Rightarrow$ 500ms 逆転

(測定結果)

	最低電圧	最高電圧 [mV]
Li-ion	7500	8300
EVOLTA	7450	8600

(考察)

Li-ion と EVOLTA で 最高電圧は異なるが
 モーター回転速度の変化(立ち上り)は同じである

(ログ)

20090801_185015.csv	両輪反転	EVOLTA
20090802_014636.csv	" "	Li-ion
20090802_014343.csv	" "	(gyro の異常振動)

(測定条件) ・ pwm 常に 100
 ・ 500ms 正転 $\Rightarrow$ 500ms 逆転

(測定結果)

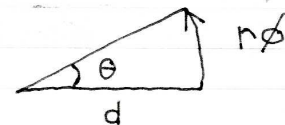
	最低電圧	最高電圧
Li-ion	7400	8240
EVOLTA	7000	8500

(考察)

- ・ EVOLTA は 電圧の変動幅(降下量)が大きい
- ・ モーターを急反転すると 実際のモーター回転が時間的に遅れる (ソフト制御による電圧降下は確認)
- ・ gyro の異常な振動は ケーブルとタイヤ溝のこすり振動

車輪回転計によるロボット旋回角推定 (dead reckoning)

(理論値) 車輪半径 $r = 81.6 / 2 = 40.8$ [mm]
 左右車輪間隔 $d = 63.0$ [mm]
 車輪回転角(差) ϕ
 ロボット旋回角 θ



$$\theta d = r \phi$$

$$\theta = \frac{r}{d} \phi = \frac{40.8}{63.0} \times \phi \approx 0.65 \phi$$

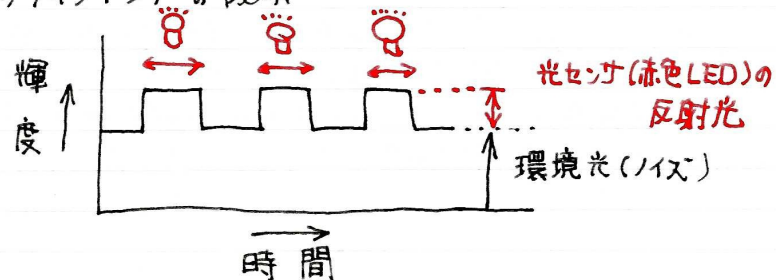
(実測値) 7/26 試走会1 走行ログ

コース1周 ($\Rightarrow$ ロボット旋回角 $\theta = 360$ deg) のとき
 左右車輪回転差 $\phi = 600$ deg

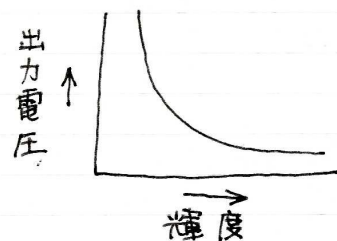
$$\theta : \phi = 360 : 600$$

$$\theta \approx 0.6 \phi$$

ロックインアンプの説明



光センサ(フォトトランジスタ)の非線形特性



センサ出力が輝度に対し
非線形に変化する特性を
(実験値から)プログラムで
補正している

試走会 2 予約 23日(日) 15:20
現地集合

郵便提出物 17日(月) 中

16日(日) 集合

203教室 9:00~17:00
9:00 自宅前

提出物 ファイル名

201-maimai-concept.pdf
201-maimai-model.pdf

印刷物 各5部 (指示用1, 審査用4)

郵送先

〒141-0031

東京都品川区 品川 1-1-13

株式会社ビッツ

ETロボコン担当

半径 200mm の円 (ツインループ) を 1 周するには?

$$\begin{aligned}\text{移動距離} &= \text{円周の長さ} \\ &= \text{半径} \times 2 \times \pi \\ &= 200 \times 2 \times 3.14 \\ &\doteq 1260 \text{ [mm]}\end{aligned}$$

$$\begin{aligned}\text{左右車輪カウンタ } \phi_{\text{add}} &= \phi_{\text{LEFT}} + \phi_{\text{RIGHT}} \\ &= 1260 \text{ [mm]} \times \frac{360 \text{ [deg]}}{256.4 \text{ [mm]}} \times 2 \text{ [個]} \\ &\doteq 3540 \text{ [deg]}\end{aligned}$$

$$\begin{aligned}\text{左右車輪カウンタ } \phi_{\text{diff}} &= \phi_{\text{LEFT}} - \phi_{\text{RIGHT}} \\ &= 600 \text{ [deg]} \\ &\quad \text{※ 試走会の実測値}\end{aligned}$$

$$\frac{\phi_{\text{diff}}}{\phi_{\text{add}}} = \frac{600}{3540} \doteq 0.17$$

NXTway-ET による実測

20090810-190725.csv move = 50
走行軌跡 直径 約 35mm (半径 17.5mm)
走行収束 move $\rightarrow$ 50 $\pm$ 0, turn $\rightarrow$ 20 $\pm$ 5

20090810-192117.csv move = 100
走行軌跡 直径 約 40mm
走行収束 move $\rightarrow$ 70 $\pm$ 10, turn $\rightarrow$ 30 $\pm$ 10

方位角のPID制御

$$\begin{aligned}\text{turn} &= -K_p \times \theta_{\text{err}} \\ &= -K_p \times (\theta_{\text{desired}} - \theta_{\text{actual}})\end{aligned}$$

※ 速度制御なし

Kp = 20 20090811-234448.csv
オーバーシュートあり
操作量 turn が Max で飽和する Bang-Bang 制御
目標到達時間 336 msec

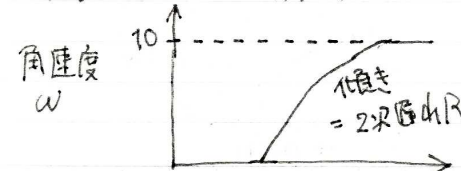
Kp = 10 20090811-235432.csv
オーバーシュートあり
キックは操作量 Max の Bang-Bang
オーバーシュートからの戻りは P 制御
目標到達時間 312 msec

Kp = 5 20090812-000658.csv
オーバーシュートあり
目標到達時間 350 msec

ステップ応答法

$$\begin{aligned}\text{1次遅れ } L &= 20 \text{ [msec]} \\ \text{2次遅れ } R &= \frac{40 \text{ [deg]}}{180 \text{ [msec]}} \doteq 0.22 \text{ [deg/msec]}\end{aligned}$$

turn = 100 (MAX) のとき



角速度 ω の定義
32.77 秒間の
wheelL - wheelR

新走行ログ

位置	型	名前	範囲
0	U32	systime	
4	S8	move	-100~100
5	S8	turn	-100~100
6	S8	v	-64~64
7	S8	omega	-12~12
8	S16	x	
10	S16	y	
12	S16	theta	0~360
14	U16	bright	0~10,000以上
16	S16	ItOn	
18	S16	ItOff	
20	U16	battery	
22	S16	gyro	
24	S32	motorL	
28	S32	motorR	

パケット長 32 bytes

8/19 (水) 試走会 前 の 結合

👤 ちと 吉田 の 結合 中 明日までに
吉田確認

- エッジチェンジ は 上位シヤで
黒白判定あり) 2バージョン用意
なし
- ポート番号は 2人で一致しているはず。
- エッジチェンジ の 成功率 100% を下回りで工夫する。
- ① 近くスタート ツインループ°
② スタート位置から ツインループ°
③ スタート → ツインループ° → ゴールストップ°
- NXT 内部状態 表示用の LED → カラーセンサで解決
- speed命令を反映させる (ゴールで停止させるため)

8月23日(日) 試走会2 以降の課題

- ・ 走行パス定義と その距離データ計算定義
- ・ 安定したエッジ切替のための方法確率と実装
- ・ 走行制御パッケージのインターフェース調整
(色取得, 制限速度設定)

試走会2 走行ログ 解析

20090823_174918.csv イコノスモ右エッジ2回周回

左右車輪カウンタ差とロボット旋回角度の関係

ロボット旋回角度 [deg]	左右カウンタ差 [count]
360	: 550
1	: 1.53
0.655	: 1

回転半径 $r = 340 \text{ mm}$ における ログ

	systime [msec]	旋回角 θ [count] [deg]	移動距離 L [count] [mm]
開始位置	64896	550 約360	38636 約13522
終了位置	70496	281 約184	42009 約14703
差	5600	269 約176	3373 約1181

⇒ 32msec あたり 約1.53 カウント (= 約1 deg) 旋回
1 sec あたり 約48 カウント (= 約31 deg) 旋回

ロボットの旋回能力 (停止旋回 90° の実験結果)

最大 (turn=100) 10count / 32msec
急停止可能範囲 3count / 32msec

AWK の 使い方

区切文字 (Field Separator) の 変更方法

BEGIN { FS = " ," }

試走会2 走行ログ 20090823_174918.csv

有効なデータは Time < 188000 ミリ秒

∴ 188400 ミリ秒に gyro が 大きな乱れあり

移動距離 と TRIP カウンタ の 変換

移動距離	TRIP カウンタ
0.35mm	1 count
1 mm	2.86 count
10 cm	286 count

「試走会2 ログ」

20090823_164907.csv	OUT	1周半	×	コースアウト
" 165159.csv	OUT	1周	◎	
" 170442.csv	IN	1周	×	
" 172244.csv	OUT	2周	◎	
" 173134.csv	OUT	2周	◎	
" 174918.csv	IN	2周	×	

2009.8.27
位置(機首角)フィードバック制御実験

比例フィードバック係数 K_p

$K_p = 0.5$ なめらかな直進

$K_p = 1.0$ 小さな左右振動あり

$K_p = 10.0$ あきらかな左右振動(発振)あり

20090827_201548.csv $K_p = 0.5$ turn = ± 20 位で適切

20090827_201911.csv $K_p = 1.0$ turn = $\pm 45 \sim 60$ で大き過ぎる

8月30日(日)
吉田

102会議室
9:45 AM ~

IN 担当
OUT 担当 吉田

坂道の頂上が見付けられない

→ 電池電圧の変動が大きすぎるため無理

→ マーカー検出不可能

・ 首振りがゆくり過ぎる

・ コースアウトの時間が長い(最大7~1~2秒)

東京地区大会 予定 9月6日(日)

ゼッケン67 試走会グループC

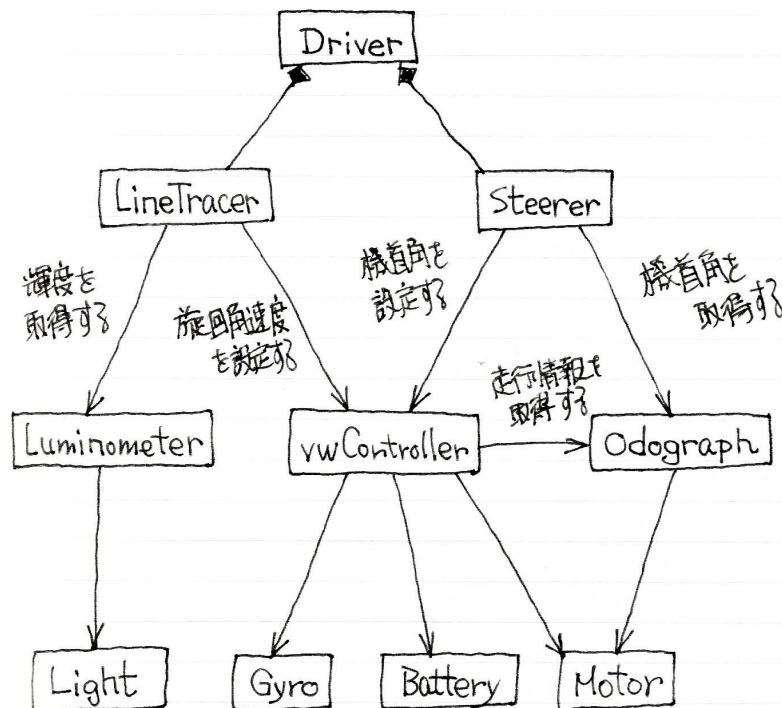
9:00	受付開始
10:00 ~ 10:15	C-Group 試走
11:00 ~ 11:15	" "
11:15 ~ 11:30	C-Group 車検
12:00 ~ 12:20	開会式
12:20 ~ 13:50	競技会 1 休憩
14:10 ~ 15:40	競技会 2 移動
16:00 ~ 18:30	モデルワークショップ・表彰式 移動
19:00 ~ 21:00	懇親会

モデルワークショップ 参加人数 2人
懇親会 参加人数 2人

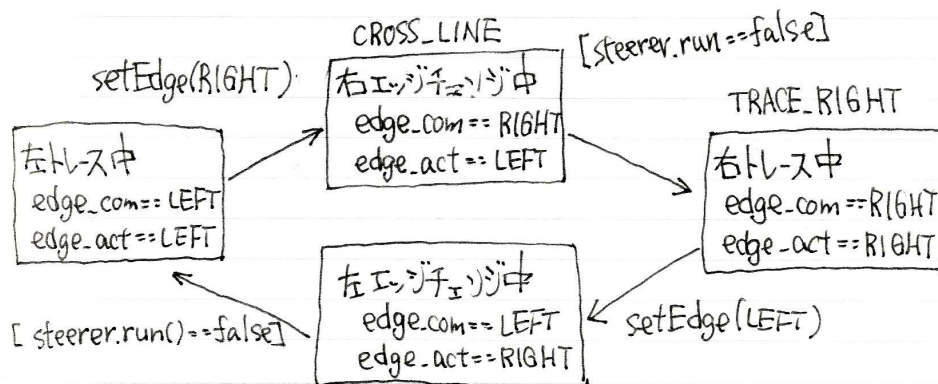
当日入り 吉田 前泊

荷物 ロボット 1台づつ 新品電池を詰めなおす
遮光 スケート 吉田
1-トPC, Bluetooth, USB (開発環境)

吉田
参加費 ¥21,000-
懇親会 ¥6,000-(2名分)
おみやげ ¥1,950-
やまびこ ¥1,900-
差額 ¥11,575-
合計 ¥30,850-



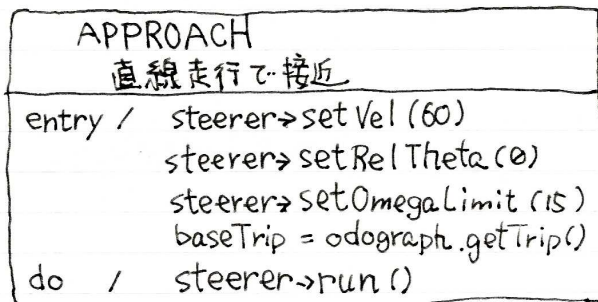
エンジェリジのステートマシン図



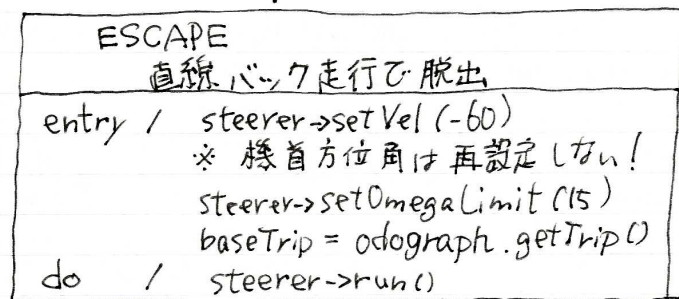
トラジャ-ハント攻略

STEP0 右エッジトレース
 STEP1 10cm 直線前進
 STEP2 10cm 直線後退
 STEP3 180° 回転
 STEP4 左エッジトレース
 STEP5 180° 回転
 STEP6 左エッジトレース

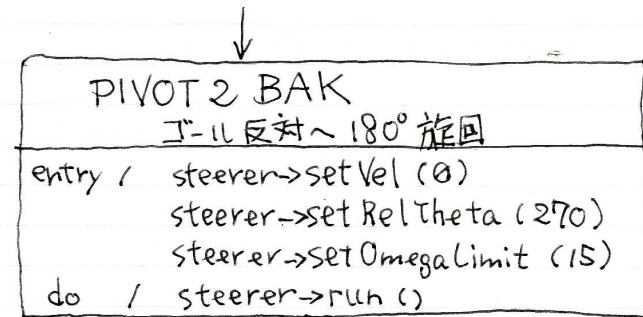
状態遷移



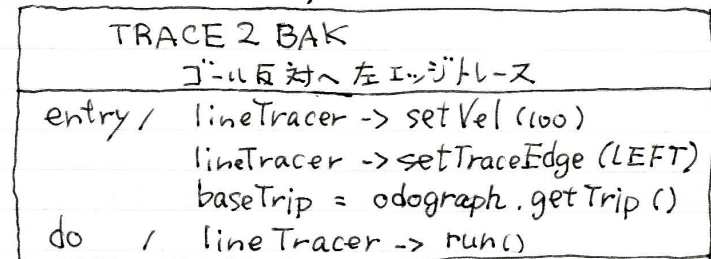
↓ 15cm 前進



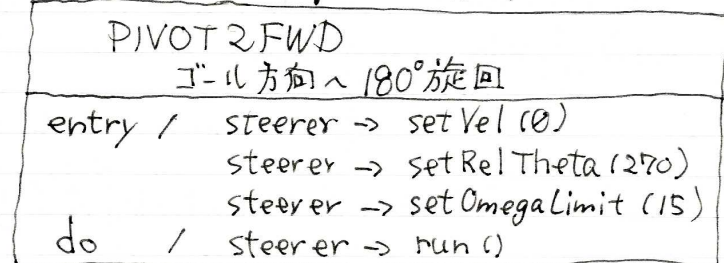
↓ 15cm 後退



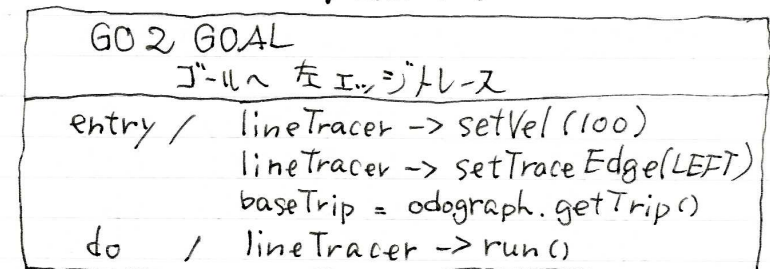
↓ 旋回完了



↓ 80cm 前進



↓ 旋回完了



トリガーハント内の長行距離計算

		entryTrip (スロ距離)	
		○	○以外
exitTrip	○	通過前	⇒ 通過中
(出口距離)	○以外	(未定義)	⇓ 通過後

試走会2 データのTRIPカウンタ解析

	1 回目	2 回目
20090823_165159.csv	53353 (-283)	なし
172244.csv	53941 (+305)	53027
173134.csv	53615 (-21)	53481
1 回目データ平均	53636	

∴ ± 100mm (300 count) の誤差あり

ゴールストップを 54000 カウントから本番調整

試走会2 トリガーハント位置解析

ロケ	距離 (trip)	時刻 (time)	Y 軸
165159	37077 (+63)	60224	2505
172244	36912 (-102)	60128	2927
173134	37053 (+39)	55552	2621
平均	37014	← ここから調整	

トリガーハント パラメータ調整

