UFMG
UNIVERSIDADE FEDERAL DE MINAS GERAIS

Introdução à Robótica
Controle (2/2)

Prof. Douglas G. Macharet
douglas.macharet@dcc.ufmg.br

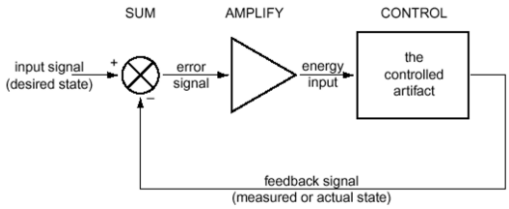
VERLab Visão e Robótica DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Introdução

- Abordagens de controle
  - Coordenação entre percepção e atuação
- Classificação
  - Feedback
  - Algorithmic
  - Reactive

Feedback Control

- Objetivo
  - Fazer o sinal de erro ir para zero
  - Termostato (feedback é a temperatura)



Feedback Control

Exemplo

- Tarefa: *Wall following*
- Sensor IR (óptico-reflexivo)
- Apenas um *threshold*
  - Muito perto/muito longe (variável objetivo)

Feedback Control

Exemplo

```
/* wallfol1.c: threshold-based wall follower with data collection */
int LEFT_MOTOR = 0;               /* motor and sensor ports */
int RIGHT_MOTOR = 3;
int LEFT_WALL = 0;

persistent int goal;              /* wall conditions */
persistent int data[1000];        /* data capture */
persistent int ix;
```


Feedback Control

Exemplo

```
void calibrate() {
   while (1) {
      int wall = analog(LEFT_WALL);
      printf("goal is %d; wall is %d\n", goal, wall);
      if (start_button()) {
         goal = wall; beep();
      }
      if (stop_button()) {
         printf("Set goal to %d\n", goal);
         beep(); sleep(0.5); break;
      }
      msleep(50L); /* give a pause for the display */
   }
}
```

Feedback Control

Exemplo

```
void main() {
   calibrate();
   ix = 0;
   while (1) {
      int wall= analog(LEFT_WALL);
      printf("goal is %d; wall is %d\n", goal, wall);
      if (wall < goal) left(); /* too far from wall -- turn in */
      else right();            /* turn away from wall */
      data[ix++] = wall;       /* take data sample */
      msleep(100L);            /* 10 iterations per second */
   }
}
```

```
void left() {
   motor(RIGHT_MOTOR, 100);
   motor(LEFT_MOTOR, 0);
}

void right() {
   motor(LEFT_MOTOR, 100);
   motor(RIGHT_MOTOR, 0);
}
```


Feedback Control

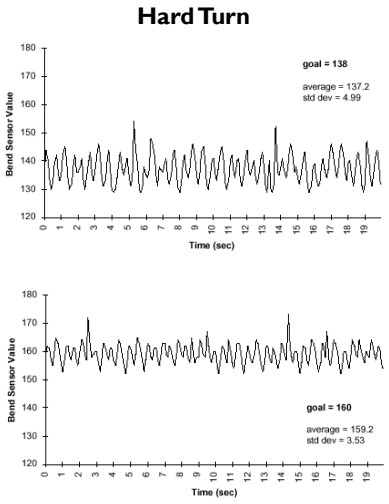
Exemplo

```
void dump_data() {
   int i;
   disable_pcode_serial();
   printf("Press START to send data...\n");
   start_press();
   for (i=0; i< ix; i++) {
      printdec(data[i]);
      serial_putchar(10);
      serial_putchar(13);
   }
   printf("Done sending data.\n");
   beep();
}
```

serialio.c, printdec.c

Feedback Control
Exemplo

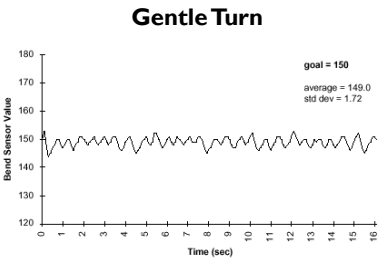
- Robô oscila muito
  - Não anda em linha reta
  - Varia em torno do goal
- É possível resolver?
  - Como?



Feedback Control
Exemplo

```
void left() {
   motor(RIGHT_MOTOR, 100);
   motor(LEFT_MOTOR, 50);
}

void right() {
   motor(LEFT_MOTOR, 100);
   motor(RIGHT_MOTOR, 50);
}
```



- É possível melhorar mais?
  - *Hysteresis*
  - 3-state algorithm (esquerda, direita, reto)

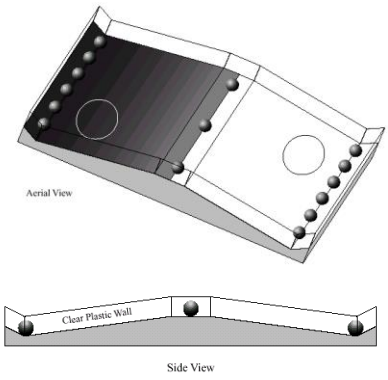
Algorithmic Control

- Feeback control apropriado para uma tarefa
  - Por exemplo, seguir uma parede
- Como combinar comportamentos simples?
- Algorithmic control
  - Sequência de etapas ou fases que um programa deve executar para conseguir realizar uma tarefa

Algorithmic Control

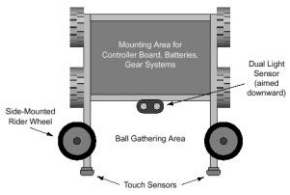
Exemplo

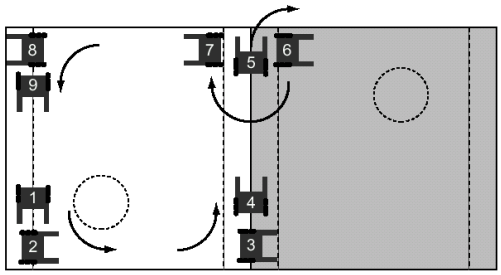
- Robo-Pong Contest
  - Transportar as bolas
  - Diferentes estratégias
- Como resolver?
  - Sequência de passos
  - Algoritmo



Algorithmic Control

Exemplo – Groucho





Algorithmic Control

Exemplo

```
void groucho() {
   while (1) { /* loop indefinitely */
      /* for simplicity, assume robot starts at position 1 */
      forward();
      waituntil_hit_wall();
      rotate_left_ninety();        /* now at position 2 */
      forward();
      waituntil_see_black();       /* position 3 */
      rotate_left_ninety();        /* position 4 */
      forward();
      waituntil_hit_wall();        /* position 5 */
      rotate_left_ninety();        /* position 6 */
      rotate_onehundred_eighty(); /* position 7 */
      forward();
      waituntil_hit_wall();        /* position 8 */
      rotate_left_ninety();        /* position 9 */
   }
}
```

Algorithmic Control

- Vantagens
  - Simplicidade, objetividade e previsibilidade
    - Quando as coisas vão de acordo com o plano!
- Desvantagens
  - Incapacidade de detectar ou corrigir problemas ou circunstâncias inesperadas na execução
  - Se uma etapa falhar, toda a solução vai falhar

Algorithmic Control

- Melhorias
  - Incorporar loops de controle por feedback em cada uma das etapas do controle algoritmico
  - Aumenta a confiabilidade/robustez
- Condições de saída
  - Detectar e corrigir falhas no plano
  - Ex: acertou um oponente ao invés da parede

Algorithmic Control

- Timeouts
  - Se um certo período de tempo tiver decorrido, a sub-rotina sai mesmo se o plano falhou
  - Acompanhar o tempo gasto na execução de uma sub-rotina e tomar alguma decisão
- Rotinas IC
  - **`seconds()`**
  - **`mseconds()`**

Algorithmic Control

```
/* define exit codes */
int NORMAL = 0;
int TIMEOUT = 1;

int follow_edge_to_wall() {
   long timeout = mseconds() + 4000L;
   while (1) {
      if (left_eye() == 0) veer_left();
      else if (right_eye() == 1) veer_right();
      else forward();
         if (left_touch() || right_touch()) return NORMAL;
         if (mseconds() > timeout) return TIMEOUT;
   }
}
```

Algorithmic Control

```
/* define exit codes */
int NORMAL= 0;
int TIMEOUT= 1;
int EARLY= 2;

/* sample timing parameters */
long TOO_LONG= 4000L;
long TOO_SHORT= 1500L;

int follow_edge_to_wall() {
   long start= mseconds();
   long timeout= start + TOO_LONG;
   while (1) {
      if (left_eye() == 0) veer_left();
      else if (right_eye() == 1) veer_right();
      else forward();
         if (left_touch() || right_touch())
            if (mseconds() < (start + TOO_SHORT)) return EARLY;
            else return NORMAL;
         if (mseconds() > timeout) return TIMEOUT;
   }
}
```


Reactive Control

- Considera uma coleção de comportamentos estímulo-resposta dinamicamente disparados e interrompidos na navegação do robô
- Obtidos pela interação com o mundo físico
  - Percepção sensorial

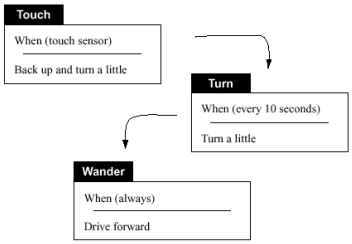
Reactive vs. Algorithmic Control

- Algorithmic
  - Conjunto de passos ou ações ordenados
  - Ambiente e interações bem estruturados
  - Dificuldade em lidar com situações inesperadas
- Reactive
  - Coleção de vários "mini-programas"
  - Todos rodando ao mesmo tempo

Reactive Control

- Os diferentes comportamentos podem assumir o controle a qualquer momento
- Produz resultados melhores em situações complexas e com interações imprevisíveis
- Exemplo
  - Processo para o sensor de toque
  - Processo periódico de giro
  - Processo de movimento aleatório

Reactive Control



- A seta indica prioridade de execução

Reactive Control

Multi-tasking

- Execução em paralelo de vários programas

```
void back_and_forth () {
   while (1) {
      fd(0); msleep(500L);
      bk(0); msleep(500L);
   }
}

void sensor_beep () {
   while (1) {
      if (analog(0) < 100) beep();
   }
}
```

```
void main () {
   start_process(back_and_forth());
   start_process(sensor_beep());
}
```

Reactive Control
Priorização

- Forma de evitar conflitos entre os programas
- Solução
  - Atribuir PID e Prioridade
- PID
  - Identifica a tarefa enviando comandos de motor
- Prioridade
  - Permite determinar a precedência de execução

Reactive Control
Priorização

```
/* set process_enable entry to process's priority level */
void enable (int pid) {
   process_enable[pid] = process_priority[pid];
}

/* set process_enable entry to 0 */
void disable (int pid) {
   process_enable[pid] = 0;
}
```

Reactive Control
Priorização

```
/* touch sensor process */

void touch (int pid) {
   while (1) {
      if (left_touch()) {
         enable(pid);
         backward(pid); msleep(500L);
         right(pid); msleep(500L);
         disable(pid);
      } else if (right_touch()) {
         enable(pid);
         backward(pid); msleep(500L);
         left(pid); msleep(500L);
         disable(pid);
      }
   }
}
```


Reactive Control
Priorização

```
/* periodic turn: every 10 secs, turn a bit */

void periodic_turn (int pid) {
   while (1) {
      if (((int)seconds() % 10) == 9) {
         enable(pid);
         right(pid); msleep(500L);
         disable(pid);
         msleep(500L);
      }
   }
}
```

Reactive Control
Priorização

```
/* wander process: just go forward */

void wander (int pid) {
   enable(pid);
   forward(pid);
}
```


Reactive Control
Priorização

```
void main () {
   int pid= 0;

   /* touch sensor */
   process_priority[pid] = 3;
   process_name[pid] = "Touch";
   start_process(touch(pid++));

   /* periodic turn */
   process_priority[pid] = 2;
   process_name[pid] = "Turn";
   start_process(periodic_turn(pid++));

   /* wander */
   process_priority[pid] = 1;
   process_name[pid] = "Wander";
   start_process(wander(pid++));

   /* motor arbitration process */
   num_processes = pid;
   start_process(prioritize());
}
```

Reactive Control

Priorização

```
void prioritize () {
   int max, i;

   while (1)  {
      /* find process with maximum priority */
      max= 0;
      for (i=0; i< num_processes; i++)
         if (process_enable[i] > max) max= process_enable[i];

      /* if no processes enabled, turn off motors */
      if (max == 0)  {
         motor(LEFT_MOTOR_PORT, 0); motor(RIGHT_MOTOR_PORT, 0);
         printf("No tasks enabled\n");
      } else {
         /* get pid of active process */
         /* if more than one at highest level, get the first one */
         for (i=0; i< num_processes; i++)
            if (process_enable[i] == max) break;
            active_process= i;

        /* set the motors based on the commands of this process */
        motor(LEFT_MOTOR_PORT, left_motor[active_process]);
        motor(RIGHT_MOTOR_PORT, right_motor[active_process]);

        /* display name of active process */
        printf("Running %s\n", process_name[active_process]);
      }
   }
}
```


Reactive Control

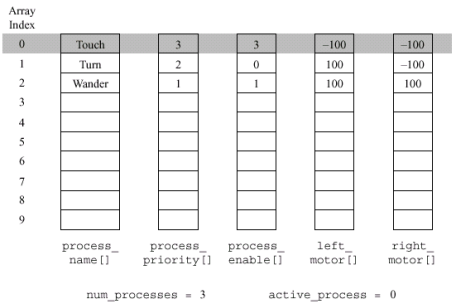
Priorização

```
/* priority.c: arbitration program for
multi-behavior robot control */

/* define process tables */
int process_priority[10];
int process_enable[10];
char process_name[0][10];

/* define motor output tables */
int left_motor[10];
int right_motor[10];

/* globals */
int num_processes = 0;
int active_process = 0;
```



Reactive Control
Exemplo

- Robo-Pong Contest – Groucho
- 8 comportamentos e 5 níveis de prioridade
  - Search
  - Scoop
  - Deliver
  - Dump
  - Return

Reactive Control
Exemplo

- Tarefas de supervisão
  - Utilizados para garantir que o robô não fique preso na execução de alguma das sub-rotinas
- Processos
  - Panic
  - Touch
  - B-Brain

Reactive Control

Exemplo – Groucho