

---

## DESENVOLVIMENTO DE UM MANIPULADOR ROBÓTICO INTELIGENTE

**GABRIEL, A. T.<sup>1</sup>; NOGUEIRA, R. G.<sup>2</sup>**

### 1. INTRODUÇÃO (justificativa e objetivos)

O que motivou a elaboração deste projeto foi o desafio de construir um sistema robótico que aprenda e execute tarefas. Nos dias atuais, tarefas humanas são automatizadas mas pouco inteligentes. Os robôs são capazes de executar, mas nem sempre podem lidar com variações nas tarefas, e muito menos tomar decisões sobre o que fazer na ocorrência de uma.

A proposta é construir um agente inteligente composto de sensores e atuadores reais. O agente percebe modificações no seu ambiente através de um tabuleiro (sensor), decide a melhor jogada (programa de agente) e executa as jogadas por meio de um manipulador robótico (atuador).

Os jogos ocuparam as faculdades intelectuais dos seres humanos desde que surgiu a civilização. Para pesquisadores de Inteligência Artificial (IA), a natureza abstrata de jogos os torna um assunto atraente para estudo. É fácil representar o estado de um jogo e, em geral, os agentes restringem-se a um pequeno número de ações cujos resultados são definidos por regras precisas [1].

Na vida real, existem muitos eventos externos imprevisíveis que levam a situações inesperadas. Muitos jogos refletem essa imprevisibilidade incluindo um elemento aleatório, como o lançamento de dados. Desse modo, conduzem a uma etapa mais perto da realidade e vale a pena observar como isso afeta o processo de tomada de decisões [1]. Este estudo de caso pode ser uma contribuição para que possa ser alcançado o Santo Graal dos Sistemas Inteligentes, retratado nos filmes de ficção em que robôs realizam todos os tipos de tarefas e têm até mesmo emoções.

### 2. METODOLOGIA

O projeto será composto de três partes: o tabuleiro do jogo como sensor do agente, o *software* que toma decisões e decide qual a melhor jogada (programa de agente) e o manipulador robótico que expressará no mundo real as jogadas (atuador).

Microcontroladores, como os chips PIC, serão testados e os recursos do hardware como porta paralela e porta de jogos serão explorados para se obter a melhor configuração entre o software embarcado e o controle computadorizado.

Para o manipulador robótico, as configurações mais comuns serão estudadas e a melhor configuração será escolhida para a aplicação e o programa de agente será expresso por algum algoritmo de aprendizado ou de busca.

#### 2.1. O TABULEIRO

Esta etapa do projeto será composta por um tabuleiro de 64 casas, como no jogo de damas, e em cada casa, um *light dependent resistor* (LDR) polarizando um transistor como chave. O bloqueio da incidência de luz no LDR eleva sua resistência de 15K ohms (resistência com luminosidade natural) para 200K ohms (resistência que o LDR atinge com ausência de luz), fazendo com que o transistor opere em região de corte e configurando o bit 0 para o barramento de leitura.

Para leitura de todas as casas do tabuleiro, temos duas opções de utilização da porta paralela. Na primeira, serão utilizados multiplexadores de sinal para selecionar a linha a ser lida. As saídas 0, 1 e 2 da porta paralela deverão ser conectadas respectivamente aos pinos 11, 10 e 9 dos multiplexadores (neste caso, os multiplexadores comerciais 74151). Esses pinos têm a função de selecionar uma das entradas dos MUXs. Por exemplo, quando a porta paralela enviar o valor binário "000" aos pinos de seleção, todas as entradas do pino 4 dos MUXs serão selecionadas.

As entradas da porta paralela farão a leitura de todos os MUXs simultaneamente, informando o estado atual do tabuleiro. Para a leitura de um estado, serão necessárias oito varreduras na porta paralela. Como uma alternativa para a leitura com multiplexadores, a leitura direta pela porta paralela poderá ser implementada. Para isso, serão necessários os oito bits de saída para polarizar

transistores e fazer a seleção de qual linha será lida naquele instante e os oito bits de entrada para a leitura da linha selecionada.

A porta paralela EPP (Enhanced Parallel Port) comum possui velocidade de 3MB/s o que permite a leitura de aproximadamente trezentos mil estados por segundo. Essa velocidade é suficientemente grande para detectar movimentações muito rápidas de uma peça do tabuleiro de uma casa para a outra.

A utilização de multiplexadores possui a vantagem de economizar 5 bits de saída da porta paralela que poderão ser utilizados para envio de comandos para os motores do manipulador robótico. Para proteger a porta paralela de altas correntes, será utilizado um buffer. Neste caso, será testado o circuito integrado (CI) comercial 74541.

## 2.2. O PROGRAMA DE AGENTE

Um agente de IA que pode jogar jogos de tabuleiro data de antes da IA e da computação moderna. Jogos de tabuleiro são parte integrante da civilização há milênios. Jogar jogos de tabuleiro é um sinal de inteligência e evidência de aprendizado. Gerações inteiras de pessoas assumiram, como desafio, jogos de tabuleiros através de todas as culturas, credos e tempos. O jogo de xadrez data de mil e quatrocentos anos, gamão, dois mil anos e o jogo da velha, três mil e quinhentos anos. Assim, é muito natural perguntar-se se computadores podem, como nós, aprender a jogar e se podem aprender a jogar, podem aprender a montar, a operar, a tomar decisões e outras situações do mundo real mais facilmente mapeáveis do que jogos de tabuleiro.

A etapa de inteligência do agente investigará duas hipóteses:

A primeira hipótese a ser investigada é a aplicação de aprendizagem por reforço e redes neurais para produzir um agente que pode jogar e aprender com seus próprios erros e derrotas. A maneira mais simples de se entender a aprendizagem por reforço é comparando-a à aprendizagem supervisionada. Na aprendizagem supervisionada, é dito ao agente como responder a uma determinada situação. Na aprendizagem por reforço, o agente é penalizado ou premiado de acordo com suas livres escolhas certas ou erradas. Em caso de penalidades, o agente sabe que aquele estado é um mau estado e tenta evitá-lo.

A segunda hipótese a ser explorada e avaliada é o algoritmo MINIMAX com corte Alfa-Beta. A idéia desse algoritmo é iniciar na posição corrente, usar o gerador de movimentos para gerar o conjunto de possíveis posições sucessoras e aplicar a função de avaliação a estas posições, escolhendo-se a melhor. Como meta do MINIMAX, têm-se minimizar os efeitos da jogada do adversário e maximizar o efeito da própria jogada. A árvore de busca é gerada até um limite (número de passos ou tempo estipulado). A partir daí uma função de avaliação é aplicada nos nós folhas. Como o objetivo do oponente é minimizar suas chances de vitória (minimizar a função de avaliação), no nível que representa a escolha dele, o valor mínimo é obtido e retornado. No nível que representa sua escolha, o valor máximo é retornado.

## 2.3. O MANIPULADOR

Para a idealização do robô, foram estudadas todas as configurações tradicionais de robô. São elas:

### 2.3.1. Configuração polar

A configuração polar, apresentada na figura 1, usa um braço telescópico, que pode ser levantado ou abaixado ao redor de um pino horizontal. O pino está montado numa base rotativa. De acordo com essa configuração, o robô pode movimentar seu braço em um envoltório esférico e, por isso, é também chamado de “robô de coordenadas esféricas”

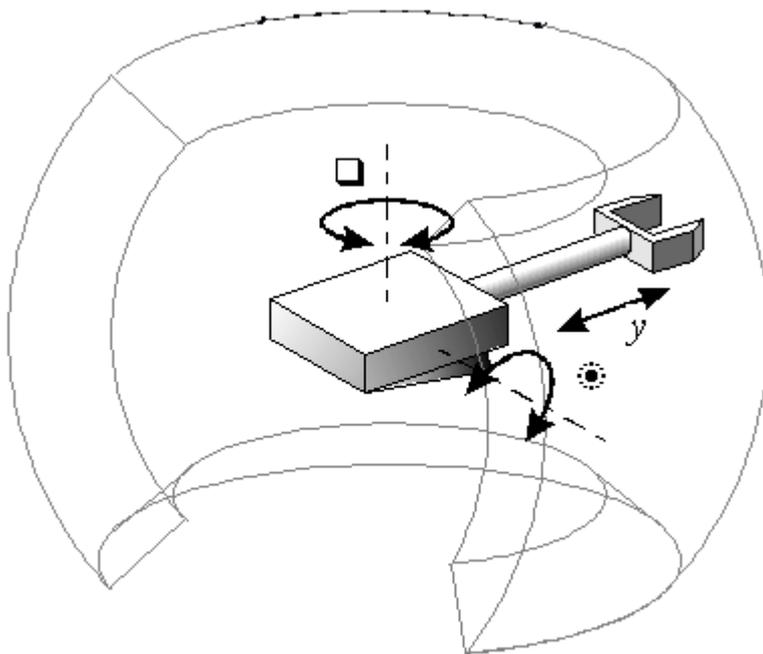


Figura 1: Robô de coordenada polar

### 2.3.2. Configuração cilíndrica

A configuração cilíndrica, representada na figura 2, usa uma coluna vertical e um suporte que pode ser movido para cima ou para baixo ao longo da coluna. O braço do robô está ligado ao suporte para que possa ser movido radialmente em relação à coluna. Girando-se a coluna, o robô é capaz de alcançar um envoltório de trabalho que se aproxima de um cilindro.

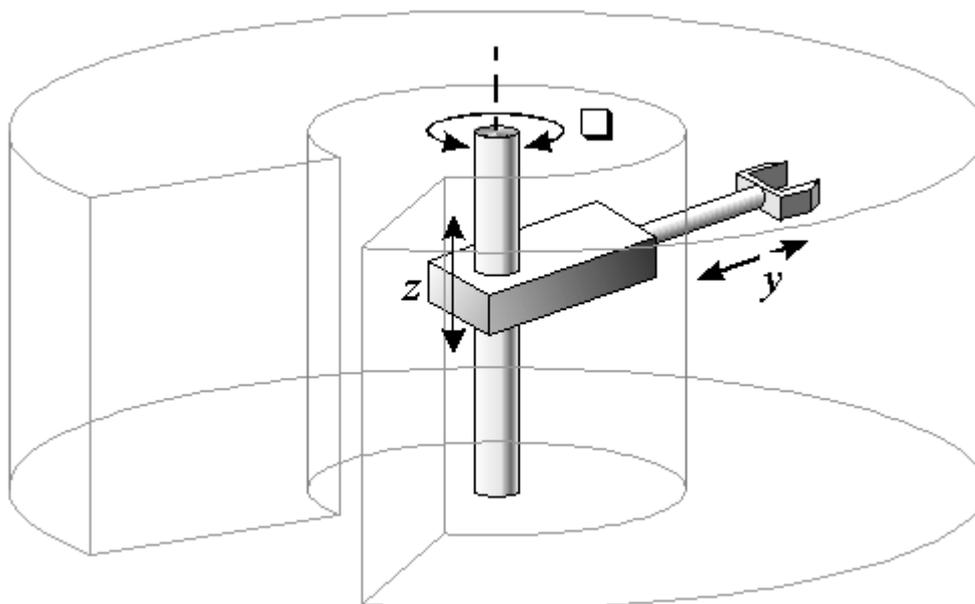


Figura 2: Robô cilíndrico

### 2.3.3. Configuração cartesiana

O robô de coordenadas cartesianas, ilustrado na figura 3, usa três réguas perpendiculares para

construir os eixos  $x$ ,  $y$  e  $z$ . Este robô pode, também, ser chamado de robô  $xyz$ . Deslocando-se as três réguas, uma em relação à outra, o robô pode operar dentro de um envoltório de trabalho cúbico.

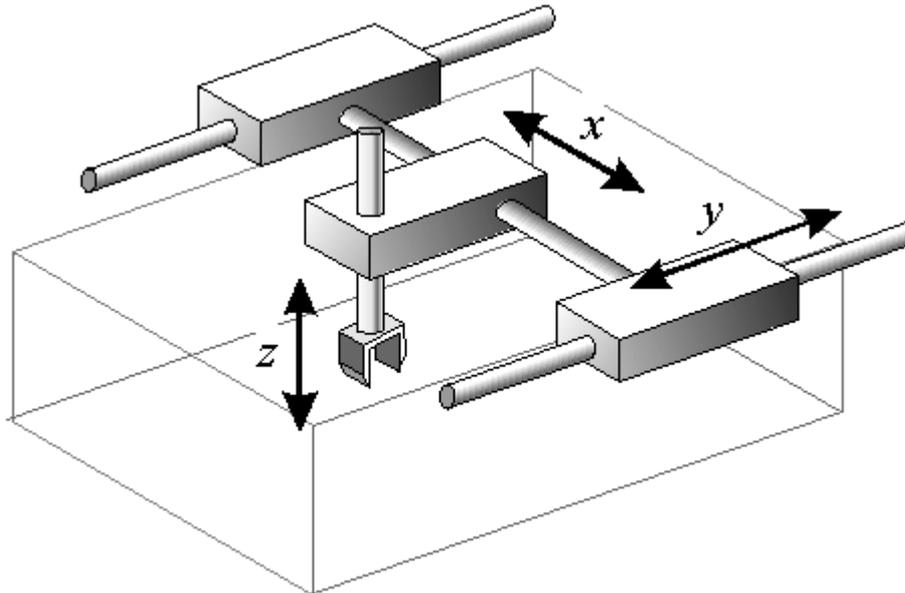


Figura 3: Robô de coordenadas cartesianas

#### 2.3.4. Configuração SCARA

O robô chamado SCARA (*Selective Compliant Articulated Robot for Assembly*), ilustrado na figura 4, é uma configuração bastante utilizada, como seu próprio nome diz, para montagens. Embora tenha a mesma configuração que o robô de coordenadas polares, o SCARA é bastante diferente no que tange aplicações e aparência.

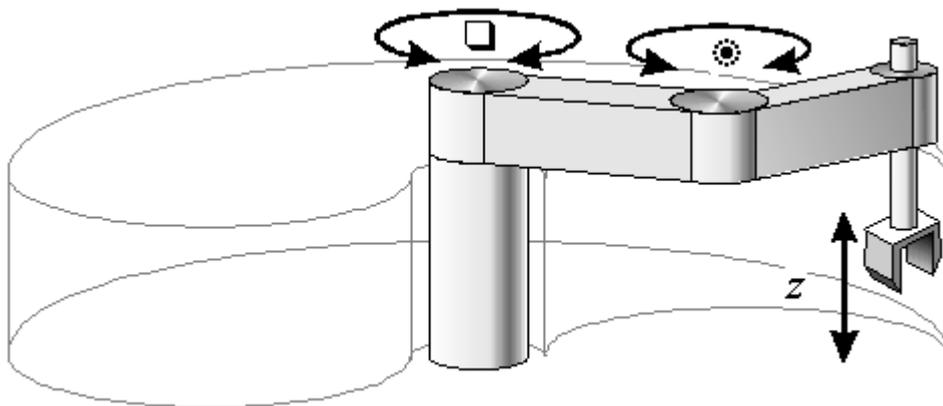


Figura 4: Robô SCARA (Selective Compliant Articulated Robot for Assembly)

O modelo proposto será uma composição entre a configuração cartesiana, a configuração SCARA e, para complementar a movimentação no eixo  $Z$ , a configuração cilíndrica. O eixo  $Y$  se movimentará através de uma esteira que poderá ter qualquer tamanho (poderá percorrer toda uma linha de montagem, por exemplo). O eixo  $X$  será articulado e seu tamanho será determinado pela potência do motor que o aciona. O modelo ainda não foi implementado nem simulado e, em caso de não funcionamento, será utilizada a configuração cilíndrica para a conclusão do projeto.

O motor que movimenta o braço em suas duas coordenadas será contínuo e por esse motivo, o controle de posicionamento será feito através de uma fita milimetrada ou disco perfurado e um transceptor óptico infravermelho. A esse dispositivo damos o nome de encoder. O encoder é um

---

transdutor que converte um movimento angular ou linear em uma série de pulsos digitais elétricos. Esses pulsos gerados podem ser usados para determinar velocidade, taxa de aceleração, distância, rotação, posição ou direção. É necessário que o programa de agente armazene variáveis de posição e que as mesmas sejam zeradas na inicialização do sistema em conjunto com uma verificação de posição zero dos eixos. A partir desse momento, toda posição será relativa ao posicionamento atual.

#### 4. CONCLUSÃO

O Manipulador Robótico Inteligente é uma ferramenta útil para o estudo de agentes inteligentes com atuadores reais robotizados e poderá auxiliar no desenvolvimento de novas tecnologias similares. Na indústria, o modelo de robô proposto poderá ser muito útil em linhas de montagem, pois permite a movimentação Y de tamanho limitado apenas pelo número de bytes que as portas do controlador podem armazenar.

#### 5 – REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Norvig, P.; Russell, S.; Inteligência Artificial; Editora Campus; 2004;
- [2] Spong, M. W.; Vidyasagar, M.; Robot Dynamics and Control; John Wiley & Sons; 1989
- [3] Beer, F. P.; Johnston Jr., E.R.; Mecânica Vetorial para Engenheiros; Makron Books; 1994
- [4] Groover, M. P.; Weiss, M.; Nagel, R. N.; Odrey, N. G.; Robótica Tecnologia e Programação; McGraw-Hill; 1989
- [5] Marques, A.E.B; Cruz, E. C. <sup>a</sup>; Choueri Júnior, S.; Dispositivos Semicondutores: Diodos e Transistores – Estude e Use; Editora Érica; 1996
- [6] Malvino, A.P.; Eletrônica Vol.1; McGraw-Hill;2001
- [7] Thomazini, D; Albuquerque, U.B.; Sensores Industriais – Fundamentos e Aplicações; Editora Érica; 2005
- [8] Watson, M.; Pratical Artificial Intelligence Programming in Java; 2004

---

<sup>1</sup> Bolsista do CNPq - Brasil. Escola de Engenharia Elétrica e de Computação, [andretapxure@hotmail.com](mailto:andretapxure@hotmail.com)

<sup>2</sup> Professor Dr. - Escola de Engenharia Elétrica e de Computação, [reinaldo@eee.ufg.br](mailto:reinaldo@eee.ufg.br)