

RobLib

Uma Biblioteca de Modelos de Robôs Industriais

1. Introdução

O uso de "software packages" é uma boa alternativa à utilização de equipamento laboratorial dispendioso. Por esta razão o desenvolvimento de aplicações nas áreas da indústria, da pesquisa e da educação conduziram ao aparecimento de vários programas na área da robótica [10-18]. Dentro desta linha de ideias surgiu o *RobLib* cuja implementação, dada a popularidade do ambiente *Windows*, foi desenvolvida em *Microsoft Visual Basic*.

O *RobLib* corresponde a uma "biblioteca" de modelos e algoritmos de robôs industriais. O programa disponibiliza ao utilizador um conjunto de diversos robôs com diferentes estruturas mecânicas e possuindo até seis juntas (rotacionais e prismáticas). Para cada robô o utilizador pode calcular as cinemáticas directa e inversa para as posições, velocidades e acelerações utilizando, alternativamente, Matrizes de Transformação Homogénea (MTH), Parâmetros de Euler ou Ângulos de Euler. Além disso, foram desenvolvidos algoritmos de planeamento de trajectórias que podem ser executados utilizando procedimentos de interpolação linear ou "splines" cúbicas.

Este artigo de apresentação do *RobLib* está estruturado em duas secções principais. Na secção dois descreve-se o funcionamento do software, ilustrando-se os cálculos referentes às cinemáticas directa e inversa e ao planeamento de trajectórias. Na última secção apresentam-se as conclusões do trabalho desenvolvido e indicam-se desenvolvimentos futuros para o programa.

2. Descrição do Software

O *RobLib* foi desenvolvido de modo a tirar total do ambiente *Windows*. Assim, todos os comandos e dados são inseridos através de "pull down" menus, botões e caixas de diálogo. O programa possui uma estrutura modular, pelo que qualquer melhoria ou alteração pode ser efectuada sem afectar outros módulos. Para além disso, novos módulos podem ser facilmente adicionados para expandir o programa.

A janela principal (fig. 1) apresenta uma barra de menu incluindo as seguintes opções: File, Robot e Help.

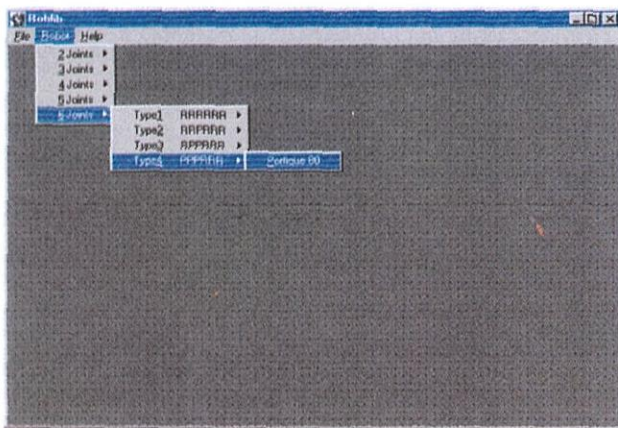


Fig. 1 - Janela principal do *RobLib*

por Carla S. M. Cordeiro * e J. A. Tenreiro Machado**

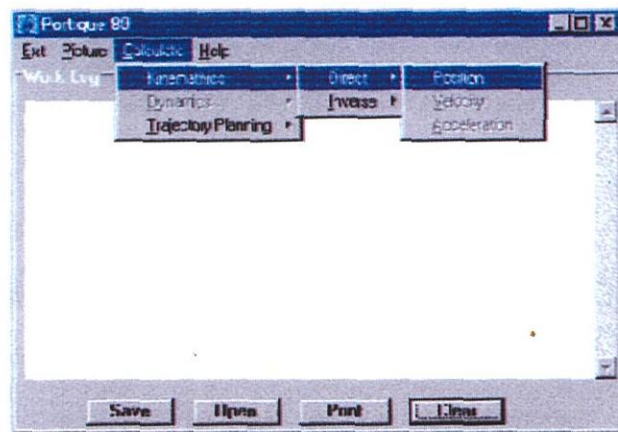


Fig. 2 - Janela de trabalho do robô Portique 80

O menu Robot permite ao utilizador escolher um robô para trabalhar. O robô está classificado de acordo com a série de juntas que possui, da base até ao punho (e.g. Portique 80 - PPPRRR). Depois de efectuada a selecção, o *RobLib* apresenta uma nova janela correspondente ao robô escolhido (fig. 2), cujo menu Calculate tem duas escolhas principais: Cinemática e Planeamento de Trajectórias.

No caso de se optar pela cinemática existem as possibilidades da cinemática directa e da cinemática inversa.

2.1. Cinemática Directa

A cinemática directa consiste em determinar a posição e orientação do órgão terminal do robô no espaço cartesiano a partir dos valores das variáveis das juntas. Os fundamentos matemáticos da cinemática [1-7] baseiam-se no conceito de MTH e no formalismo de Denavit-Hartenberg (DH), o que implica uma atribuição de eixos de coordenadas a cada junta. O *RobLib* fornece informação acerca dos parâmetros DH para cada robô (fig. 3), assim como uma representação esquemática da sua estrutura mecânica.

Parameters	1	2	3	4	5	6
a_1	1	1	1	0	0	0
α	$\pi/2$	$\pi/2$	0	$\pi/2$	$\pi/2$	$\pi/2$
d_1	0	0	0	0	0	0
θ_1	0	0	0	q_4	q_5	q_6
r_1	q_2	q_3	q_4	0	0	0

Fig. 3 - Parâmetros DH do robô Portique 80

Os resultados da cinemática directa podem ser obtidos através da MTH (representando as coordenadas do órgão terminal expressas em função do sistema de coordenadas da base) ou, alternativamente, utilizando coordenadas cartesianas para a posição e Parâmetros ou Ângulos de Euler para a orientação [8]. A fig. 4 mostra, para o robô Portique 80, o resultado da cinemática directa de posição, expressa em Parâmetros de Euler correspondente a um determinado

conjunto de valores numéricos. Os limites (Min, Max) impostos às juntas definem a gama de valores válidos para cada junta. Para as juntas rotacionais os limites são especificados através de ângulos mínimos e máximos, enquanto que para as juntas prismáticas a especificação é feita por comprimentos mínimos e máximos. Estes limites não podem ser desrespeitados e aparecerá uma mensagem de aviso se o utilizador tentar inserir valores inadequados.

Estão disponíveis procedimentos idênticos para as velocidades e as acelerações. O cálculo da velocidade das juntas implica um cálculo prévio das posições das juntas. Do mesmo modo, o cálculo das acelerações implica o cálculo prévio das posições e velocidades das juntas. Consequentemente, estas opções são activadas sucessivamente ao longo de uma sessão de trabalho.

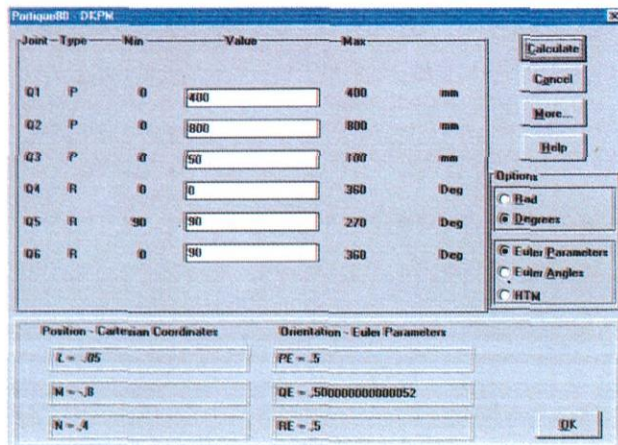


Fig. 4 - Cinemática directa de posição do Portique 80 expressa em Parâmetros de Euler

2.2. Cinemática Inversa

Para uma dada configuração do robô, a cinemática inversa implica a determinação dos valores das variáveis das juntas, o que pode levar a mais do que uma solução [1,2,3]. De facto, muitos robôs possuem soluções múltiplas para a mesma configuração do "gripper" (fig. 5). A solução analítica da cinemática inversa de posição é obtida, ou por desacoplamento das equações de cinemática [8] ou através de métodos trigonométricos [3]. Todos os cálculos de cinemática são realizados com base em equações optimizadas de forma a minimizar o número de operações necessárias à sua resolução [9].

Todos os resultados obtidos durante uma sessão de trabalho são registados num "work log" e podem ser guardados num ficheiro de texto. O utilizador poderá ainda fazer anotações e alterações directamente no "work log".

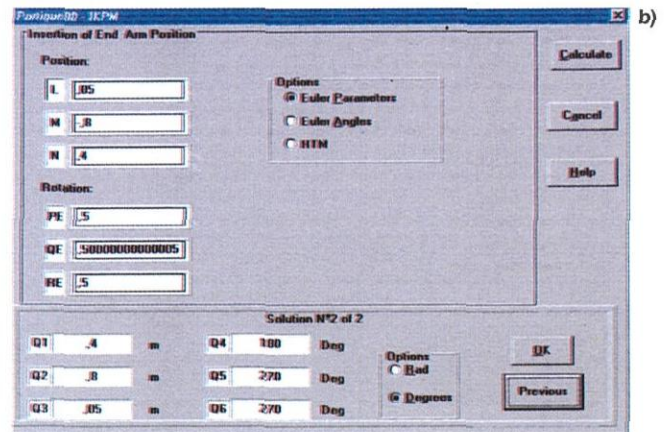
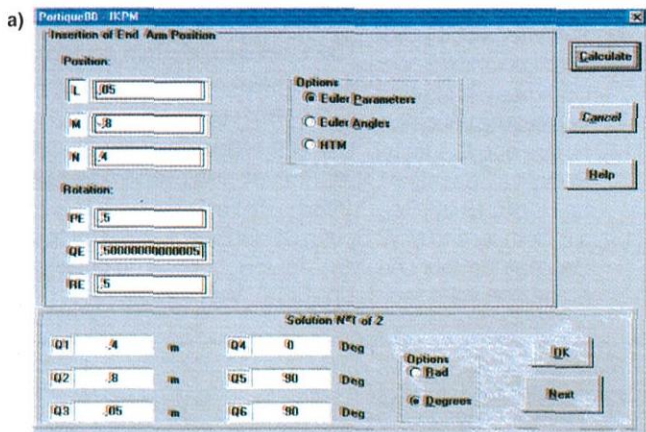


Fig. 5 - Cinemática inversa de posição:

- a) primeira solução
- b) segunda solução

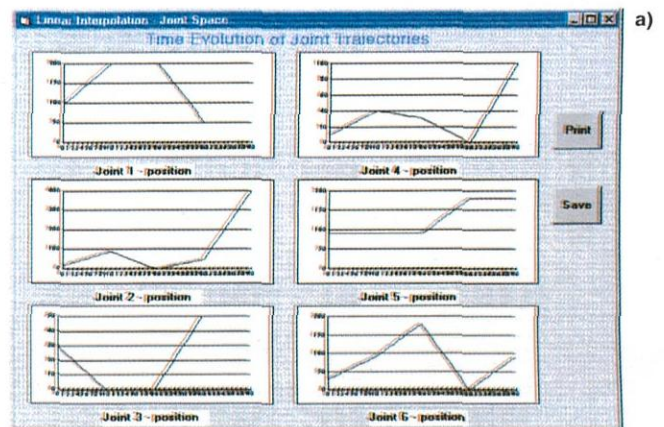
2.3. Planeamento de Trajectórias

O planeamento de trajectórias trata da evolução no espaço e no tempo das posições, das velocidades e das acelerações que um robô deve descrever. Dado um conjunto de pontos através dos quais o órgão terminal do robô deve passar, o procedimento de interpolação aproxima o percurso desejado por uma função polinomial e define uma sequência temporal de pontos que o robô deve seguir desde a posição inicial até à final [1].

O programa RobLib permite que a definição dos pontos possa ser feita quer no espaço cartesiano quer no espaço das juntas, enquanto que a interpolação pode ser efectuada utilizando linhas rectas ou "splines" cúbicas. Se o utilizador escolher linhas rectas, pode definir o tempo de aceleração permitido nos pontos de união das rectas, ou seja, pode definir a suavidade de movimento nos pontos de transição. O algoritmo calcula a evolução temporal das posições, velocidades e acelerações para cada junta.

Por exemplo, a fig. 6 ilustra a evolução temporal das posições, dados seis pontos definidos no espaço das juntas, usando uma interpolação linear sem suavização (Fig. 6a) e com um tempo de aceleração de 2 segundos (Fig. 6b) para o robô Portique 80. Por seu lado, a fig. 7 mostra evolução temporal das posições, velocidades e acelerações para os mesmos pontos utilizando uma interpolação com "splines" cúbicas.

Os algoritmos de interpolação garantem a continuidade da velocidade e da aceleração em todos os pontos, incluindo a posição inicial e a final, através da introdução de dois pontos adicionais na trajectória [1].



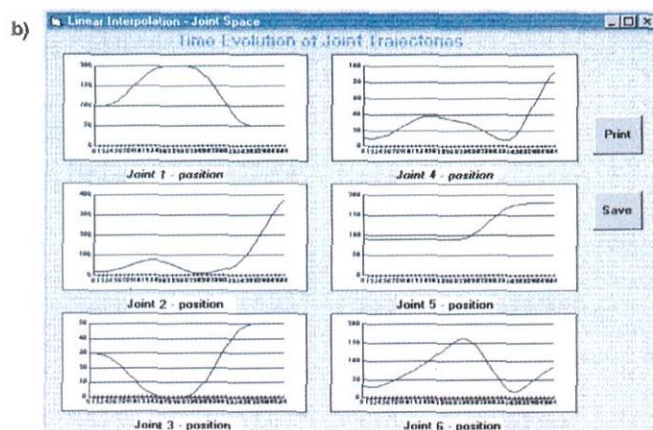


Fig. 6 - Evolução temporal da posição usando interpolação linear a) sem suavização; b) com um tempo de aceleração de 2 segundos

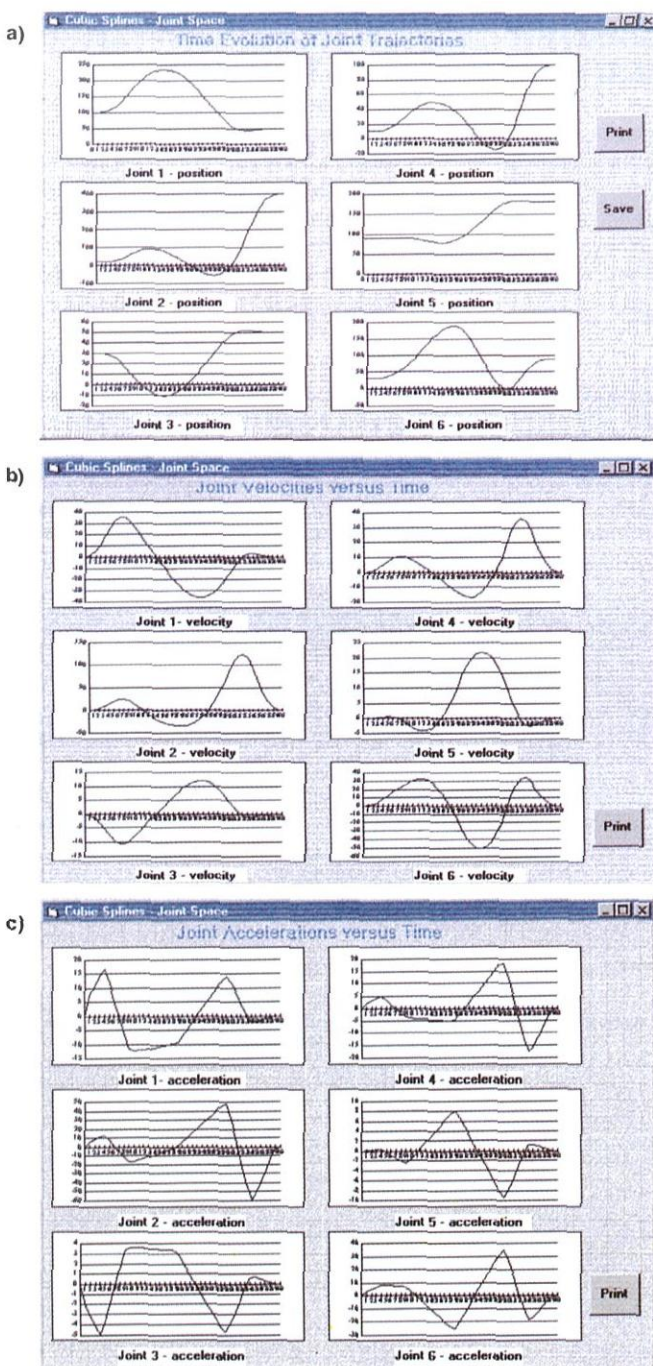


Fig. 7 - Evolução temporal das variáveis das juntas usando uma interpolação com "splines" cúbicas a) posições, b) velocidades, c) acelerações

3. Conclusões e Desenvolvimentos Futuros

Neste artigo descreveu-se o programa *RobLib* que constitui uma biblioteca de robôs industriais. Assim, o "package" permite calcular a cinemática e o planeamento de trajetórias de acordo com diferentes algoritmos. Nesta perspectiva, o programa constitui uma ajuda preciosa para o estudo de vários fenómenos que ocorrem no âmbito da utilização de manipuladores robóticos.

Versões futuras do programa irão ampliar as capacidades do *RobLib* de modo a incluir novos algoritmos de interpolação para o planeamento de trajetórias, aumentar a biblioteca de robôs disponíveis e incluir algoritmos de dinâmica e de controlo.

4. Bibliografia

- [1] K. S. Fu; R. C. Gonzalez; C. S. G. Lee, "Robotics: Control, Sensing, Vision and Intelligence", McGraw-Hill International Editions, 1987.
- [2] J. Robert Schilling, "Fundamentals of Robotics: Analysis and Control", Prentice-Hall Internacional, Inc, 1990.
- [3] Richard P. Paul, "Robot Manipulators: Mathematics, Programming and Control", The MIT Press, 1981.
- [4] Mark W. Spong; M. Vidyasagar, "Robot Dynamics and Control", John Wiley & Sons, 1989.
- [5] Haruhiko Asada, Jean-Jacques E. Slotine, "Robot Analysis and Control", John Wiley & Sons, 1986.
- [6] John J. Craig, "Introduction to Robotics: Mechanics and Control", Addison-Wesley Publishing Company, Inc, 1989.
- [7] William A. Wolovich, "Robotics: Basic Analysis and Design", Holt, Rinehart and Winston, 1987.
- [8] Saïd M. Megahead, "Principles of Robot Modelling and Simulation", John Wiley & Sons, 1993.
- [9] J.A. Tenreiro Machado and Alexandra M. Galhano, "Benchmarking Computer Systems for Robot Control", IEEE Trans. on Education, vol. 38, no. 3, pp. 205-210, Aug. 1995.
- [10] J.A. Tenreiro Machado and Alexandra M. Galhano, "WinRob: An Educational Program for Robotics", Int. Journal Elect. Engineering. Education, vol. 34, pp. 37-47, 1997.
- [11] S. Derby, "In Position: Simulating Robotic Workcells on a Micro", Computers in Mechanical Engineering, pp. 34-37, Sept. 1986.
- [12] R.B. White, R.K. Read, M.C. Koch and R.J. Schilling, "A Graphics Simulator for a Robotic Arm", IEEE Trans. on Education, vol. 32, no. 4, pp. 417-429, Nov. 1989.
- [13] T. Raz, "Graphics Robot Simulator for Teaching Introductory Robotics", IEEE Trans. on Education, vol. 32, no. 2, pp. 153-159, May 1989.
- [14] R.E. Parkin, "An Interactive Robotic Simulation Package", Simulation, vol. 56, no. 5, pp. 337-345, May 1991.
- [15] A.M. Eydgahi and J.J. Sheehan, "A Computer Animation of Robotic Manipulators and Workcells", IEEE Control Systems, vol. 11, n. 4, pp. 56-59, June 1991.
- [16] M.C. Leu and Y.S. Wang, "Studying Robot Kinematics and Dynamics with the Aid of MATHEMATICA", Int. J. of Mechanical Engineering Education, vol. 19, n. 3, pp. 213-228, 1990.
- [17] N. Vira and E. Tunstel, "Use of Symbolic Computation in Robotics Education", IEEE Trans. on Education, vol. 35, no. 1, pp. 18-30, Feb. 1992.
- [18] J.F. Nethery and M.W. Spong, "Robotica: A Mathematica Package for Robot Analysis", IEEE Robotics & Automation Magazine, vol. 1, no. 1, pp. 13-20, March 1994.

* Universidade Moderna do Porto
Departamento de Automação e Controlo

** Instituto Superior de Engenharia do Porto
Departamento de Engenharia Electrotécnica