

EXPERIMENTO PARA DISCIPLINA DE ROBÓTICA MÓVEL BASEADO EM PROGRAMAÇÃO GENÉTICA

Perci Ayres Antiquiera – perci.ayres@gmail.com

Universidade Tecnológica Federal do Paraná (UTFPR) – Curso de Pós-graduação em Engenharia elétrica e Informática Industrial (CPGEI).

Avenida Sete de Setembro, 3165

CEP 80.230-910 – Curitiba - Paraná

Carlos Raimundo Erig Lima – erig@utfpr.edu.br

Universidade Tecnológica Federal do Paraná (UTFPR) – Curso de Pós-graduação em Engenharia elétrica e Informática Industrial (CPGEI) – Departamento Acadêmico de Eletrônica (DAELN).

Av. Sete de Setembro, 3165

CEP 80230-901 - Curitiba - PR

Resumo: *A proposta de uma disciplina de Robótica Móvel dentro de um curso de Engenharia visa propiciar uma integração de várias competências necessárias à formação de um profissional. Este trabalho descreve um dos vários experimentos a serem explorados pelos alunos dentro da disciplina de Robótica Móvel, enfatizando o aspecto de Inteligência Artificial. Neste aspecto a Programação Evolucionária é uma das disciplinas com grande aplicação na área de engenharia, bem como de outras ciências físicas e biológicas, sendo dividida basicamente em Algoritmos Genéticos e Programação Genética. Este trabalho também demonstra a aplicabilidade da Programação Genética no desenvolvimento automático de software de controle embarcado para um robô móvel, que deve ser capaz de seguir uma faixa pintada no solo. A validação do sistema é feito em um software simulador do ambiente e do robô, podendo posteriormente, ser aplicado a um caso real e ser utilizado como parte de um trabalho prático para cursos de Engenharia Eletrônica ou de Engenharia de Automação.*

Palavras-chave: *Programação Genética, Robótica Móvel*

1 INTRODUÇÃO

Nos últimos anos, muito se tem estudado sobre Programação Genética e suas aplicações práticas. Dentre estas aplicações, destaca-se o desenvolvimento automático de software para controle de robôs. Tais softwares, muitas vezes estão localizados dentro do circuito robótico e são chamados de software embarcados. Além disto, os modernos circuitos de memória permitem que este software seja reinstalado sem a necessidade de desativar o circuito, o que possibilita uma reconfiguração comportamental do robô, para enfrentar e contornar situações inusitadas e imprevistas que poderiam impedir a realização de uma determinada tarefa. Neste caso, a Programação Genética aparece como uma solução, pois é capaz de gerar um software que se adapta às novas condições, de forma automática, independentemente de intervenção humana. Para isto, utiliza várias funções de entrada, previamente codificadas, combinando-as de forma aleatória para a produção de um novo software. Deste modo, vários softwares são gerados e avaliados, sendo os de melhor desempenho, selecionados enquanto os piores são descartados. De forma análoga ao que ocorre na natureza, é realizado um cruzamento entre os

softwares que permanecem, gerando-se descendentes que podem ou não ser melhores que seus pais e que também participam de uma nova seleção. Assim, de geração em geração a tendência é que haja uma evolução no desempenho dos indivíduos. Por fim, atingido algum critério limítrofe, o melhor indivíduo é selecionado.

2 CONCEITOS SOBRE ROBÔS MÓVEIS

Os conceitos apresentados nesta seção visam proporcionar uma visão geral sobre robôs Móveis de usos interno e construção simples, que possam ser implementados como trabalhos práticos em cursos de engenharia. Assim, são discutidos os componentes básicos de um robô Móvel e os tipos mais comuns. Para mais detalhes ver (CORREIA, 2006).

2.1 Sensores

Os sensores fornecem ao robô informações sobre o mundo externo permitindo uma realimentação para controle do seu comportamento. Os sensores mais simples são:

- Interruptor com mola, normalmente posicionado na parte dianteira do robô e chamado de bigode de gato, indica quando ocorre uma colisão com um obstáculo.
- Fotosensor ou célula-fotoelétrica é um sensor passivo que responde com uma variação de resistência de acordo com a intensidade de luz detectada.
- Sensor infravermelho ativo é composto por um emissor e um receptor de luz infravermelha, a qual é refletida em um objeto externo e captada pelo receptor. de acordo com a amplitude do sinal recebido pode-se determinar a presença ou mesmo sua distância deste objeto ao sensor.
- Sensor de interrupção de feixe é um caso particular de sensor infravermelho ativo onde o emissor e o receptor, são colocados um de frente para o outro possibilitando que o feixe de luz seja interrompido por um obstáculo entre ambos. Caso este obstáculo seja um disco com raias transparentes acoplado a um eixo, pode-se através da contagem de pulsos em um determinado período, chegar-se a valores de ângulos ou velocidade de rotação.
- Sensor de Ultra-som é um sensor ativo onde um sinal sonoro de frequência acima da capacidade de percepção do ouvido humano é emitido e ao encontrar um obstáculo é refletido na forma de um eco. De acordo com o tempo transcorrido entre a emissão e a recepção do sinal, pode-se calcular a distância do referido obstáculo.
- Flexo-sensor é uma tira metálica que varia sua resistência quando flexionada, permitindo a medição de flexão em braços robóticos.
- Outros sensores mais complexos também podem ser utilizados tais como lasers, câmeras de imagem, acelerômetros, gps, etc.

2.2 Atuadores

Os atuadores são os componentes que produzem a movimentação do robô, permitindo sua atuação no mundo externo. Estes atuadores podem ser de diversos tipos tais como eletromecânicos, eletropneumáticos, hidráulicos, etc. Para robôs simples utilizados em trabalhos práticos em cursos de engenharia, o tipo de atuador mais utilizado é o motor elétrico. Os dois tipos de motores mais utilizados são:

- Motor DC, que é um dispositivo movido à corrente contínua, que causa a rotação de seu eixo em alta velocidade. A rotação produzida depende da tensão aplicada ao

motor. O consumo de corrente elétrica se eleva à medida que a velocidade do motor diminui devido a um maior esforço e será máximo com o motor travado. Este fato pode ser utilizado para sensoriamento das condições de esforço do motor. Os motores DC apresentam geralmente alta velocidade e baixo torque. Assim, frequentemente os mesmos são associados a conjuntos redutores com engrenagem ou polias, que proporcionam velocidade mais baixa e alto torque.

- Servomotor, que é um dispositivo constituído por um motor DC associado a uma redução com engrenagens e um codificador que realimenta a posição do eixo de saída, a um sistema de controle em malha fechada. Desta forma pode-se através de um sinal de comando posicionar o eixo em um determinado ângulo de rotação, com grande precisão. Pode-se também, adaptar um servomotor para girar continuamente. Estas facilidades fazem com que os servomotores sejam muito utilizados em robótica móvel.

2.3 Controle do Robô

Para que o robô realize sua ação no mundo externo de forma conveniente é necessário um circuito de controle que, baseado nas informações percebidas pelos sensores, comande os atuadores de forma adequada.

O controle dos robôs quanto ao posicionamento, velocidade e aceleração de seus motores é normalmente realizado por circuitos tradicionais de controle em malha fechada, podendo caso seja necessário, ser um controle do tipo proporcional integral derivativo ou PID. Este tipo de controle resulta em movimentos rápidos e precisos do robô.

Outro tipo é o chamado controle reativo do robô, que decide as ações realizadas pelo mesmo a cada momento em função de seus objetivos e das informações recebidas pelos sensores.

Um dos controles reativos mais simples é encontrado nos chamados veículos Braitenberg, onde dois sensores fotoelétricos posicionados na parte frontal do robô controlam diretamente dois motores independentes responsáveis pela sua movimentação. Se cada sensor controla o motor do mesmo lado, direito ou esquerdo, o veículo tende a fugir de uma fonte de luz. Caso as posições sejam invertidas, ou seja, o sensor do lado direito controla o motor do lado esquerdo e vice versa, o veículo tende a seguir a fonte de luz.

Em controles reativos mais elaborados, as ações são realizadas por um conjunto de funções independentes chamadas comportamentos. Os comportamentos podem ser relativamente simples e sem variação sendo chamados então de padrões de ação fixos (PAF). Neste caso a verificação dos sensores para tomada de decisão é feita apenas antes do início da ação. Assim, estes comportamentos não apresentam grande precisão. Em um comportamento mais complexo a avaliação dos sensores pode ser realizada antes e durante a ação, como por exemplo, rotacionar o robô até que um sensor fotoelétrico detecte luz.

Os processos mais habituais para se determinar a sequência de comportamentos executados são a arbitragem onde cada comportamento recebe uma prioridade e é executado sequencialmente e a combinação de ações onde se tenta combinar vários comportamentos simultaneamente.

3 PROGRAMAÇÃO GENÉTICA

3.1 Definição

Programação Genética é um método de geração automática de soluções para um determinado problema computacional. Este problema ou objetivo é especificado em alto nível e a solução é buscada através da combinação aleatória de fragmentos de código, gerando-se

uma população de programas que podem ser mais ou menos adequados. Os programas são avaliados, através de uma função chamada função de *fitness*, que mede sua adequabilidade, recebendo uma pontuação, de acordo com os critérios de cumprimento do objetivo. Por um processo similar ao da seleção natural, os programas mais adequados são selecionados e participam de operações genéticas como cruzamentos, mutação, reprodução, duplicação e exclusão de genes, gerando uma nova população de programas. Este processo é repetido várias vezes e a cada nova geração há uma tendência de se obter programas melhores, ou seja, há uma evolução. Após um determinado número de gerações pode-se atingir um predeterminado grau de satisfação quanto à solução do problema. O processo termina com a escolha do melhor indivíduo desta população (KOZA & POLI, 2005).

3.2 Estrutura dos Programas Gerados

Os programas gerados por Programação Genética são representados em uma estrutura de árvore. Por exemplo, para representar um trecho de programa que move um robô sobre uma faixa, poderíamos ter a árvore da Figura 1.

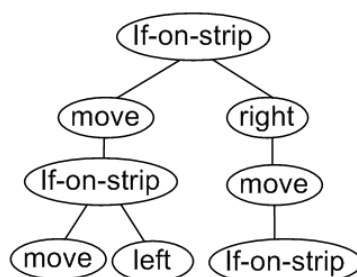


Figura 1 – Árvore para movimento do robô.

Os nós da árvore representam as variáveis e constantes ou funções, os quais se interligam pelos ramos, que indicam a sequência de execução e a passagem de possíveis parâmetros. As folhas da árvore são chamadas de terminais.

4 TRABALHOS SIMILARES

4.1 Robô que Localiza e Move uma Caixa

No trabalho apresentado em (KOZA & RICE, 1992), a programação genética é utilizada para evoluir um software que controla um robô, que deve localizar uma caixa no meio de uma sala irregular e move-la para a borda, dentro de um determinado tempo. O robô possui 12 sensores ultrassônicos, cobrindo cada um deles, um ângulo de 30° que são utilizados como terminais de entrada. Possui também um sensor de colisão e um sensor de travamento e realiza movimentos para frente em uma distância de um pé ou de rotação de 30° para esquerda ou para a direita. Os 3 tipos de movimentos são os terminais de saída. Além destas primitivas, são utilizadas 4 funções (IFBMP, IFSTK, IFLTE, PROGN2). As funções IFBMP e IFSTK, utilizam dois argumentos e executam o primeiro em caso de colisão (*Bump*) ou travamento (*Stuck*), respectivamente. Caso contrário, executam o segundo argumento. A função IFLTE (*If-Less-Than-or-Equal*) utiliza 4 argumentos e realiza o terceiro se o primeiro for menor ou igual ao segundo. Caso contrário, executa o quarto argumento. A função PROGN2 executa seus dois argumentos em sequência. A função de *fitness* mede a distância da caixa à parede mais próxima, após o término do tempo estipulado..

4.2 Robô que Evita Obstáculos

No trabalho apresentado em (NORDIN & BANZHAF, 1995), temos um exemplo de aplicação da programação genética utilizada para controlar um robô miniatura microcontrolado. O objetivo do robô é manter-se em movimento contornando possíveis obstáculos, num contexto de sentir-pensar-agir, onde, no processo de evolução, cada indivíduo utiliza seis terminais de entrada referentes a sensores de proximidade infravermelhos e gera dois valores de saída para acionamento dos motores. Neste caso não é realizada nenhuma simulação. Assim, o software evolui em tempo real através de um processo de sensoriamento do ambiente e seleção por torneio entre os indivíduos gerados. Para isso, é utilizada uma função de *fitness* que tem uma parte negativa que é a soma das proximidades dos sensores em relação aos obstáculos e, para evitar que o mesmo fique parado ou em rotação, gera pontos positivos enquanto o robô estiver se movendo rapidamente em linha reta por um determinado período de tempo.

4.3 Robô que Contorna Paredes

Um interessante trabalho utilizando programação genética para controle de robô móvel é encontrado em (LAZARUS & HU, 2001). Neste trabalho o robô deve ser capaz de se mover ao longo de uma parede em uma dada direção e sem se afastar dela. O software de controle do robô é gerado a partir de terminais de entrada representando os sensores de obstáculos em oito direções (N, NE, NW, S, SE, SW, L e O), terminais de saída ou ações produzindo movimentos em quatro direções (north, east, south, west) e quatro conectores lógicos (if, and, or, not). A função de *fitness* é calculada com base em cada movimento realizado pelo robô. O software de controle ganha um ponto se o movimento ocorre rente à parede e em uma célula que ainda não foi utilizada.

5 MODELAGEM DO EXPERIMENTO

5.1 Modelo do Robô

Para modelagem do nosso robô utilizou-se um modelo de três rodas, sendo duas ligadas a motores independentes que podem girar nos dois sentidos e uma terceira roda de giro livre. Isto permite ao robô movimentos com bastante liberdade, podendo inclusive rotacionar sobre seu centro geométrico. O robô é microcontrolado e possui um sensor óptico à sua frente que percebe se está ou não sobre uma faixa de tonalidade clara pintada no solo. Esta informação deve ser processada pelo microcontrolador de acordo com um software de forma que o robô possa caminhar seguindo a faixa, até atingir seu objetivo. A Figura 2 mostra um desenho ilustrativo do robô.

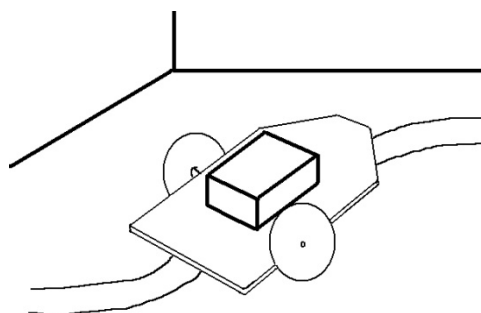


Figura 2 - Robô seguindo uma faixa.

As coordenadas $R(x,y)$ do robô são referenciadas segundo um sistema de eixos cartesianos, considerando-se o seu centro de rotação. O sensor, por sua vez, é referenciado, calculando-se suas coordenadas através da distância d entre o centro do robô e o sensor e do ângulo α , que indica a direção do robô a cada instante, como mostrado na Figura 3.

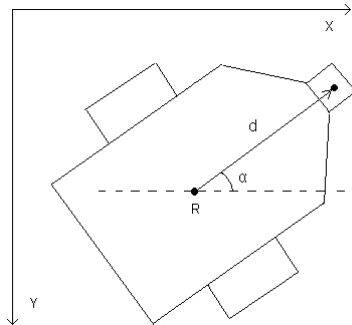


Figura 3 – Coordenadas do sensor S.

No modelo utilizou-se $d=4\text{cm}$. Assim estando robô em um coordenada $R(x,y)$ e apontado para uma direção α , as coordenadas do sensor serão $S(x_s, y_s)$, onde:

$$x_s = x + d \cdot \cos \alpha \quad (1)$$

$$y_s = y + d \cdot \sin \alpha \quad (2)$$

5.2 Modelo dos Movimentos do Robô

Os movimentos do robô são modelados de acordo com as funções de movimento, que serão utilizadas para gerar o software de controle. Nesta abordagem, foram utilizadas três funções de movimento. A função "move", que faz o robô andar para frente por um espaço fixo $s=1\text{cm}$ e as funções "right" e "left", que fazem o robô girar para a direita ou para a esquerda respectivamente, em um ângulo de 5° a partir de sua direção atual. Assim, estando o robô em uma coordenada $R(x,y)$ e apontado para uma direção α , as novas coordenadas após a execução da função "move" seriam $R(x',y')$, onde:

$$x' = x + s \cdot \cos \alpha \quad (3)$$

$$y' = y + s \cdot \sin \alpha \quad (4)$$

Já os ângulos α' e α'' , após a execução das funções "right" e "left" respectivamente, seriam:

$$\alpha' = \alpha + 5 \quad (5)$$

$$\alpha'' = \alpha - 5 \quad (6)$$

5.3 Modelo do Percurso

A modelagem do percurso foi feita com base no problema da formiga, contido no livro Genetic Programming (KOZA, 1992), onde é utilizada uma matriz 90x90, representando o espaço físico onde o robô deve atuar. A escala utilizada foi de 1 caractere por 1cm. O caminho a ser seguido pelo robô é indicado por uma faixa pintada no solo e que é representada por uma sequência de caracteres sustenido, preenchidos na matriz.

6 EXECUÇÃO E TESTES

O experimento foi executado no software LilGP, tomando-se como base o problema da formiga e fazendo-se as devidas modificações nas funções de movimento.

6.1 Funções Utilizadas

Foram utilizadas as seguintes funções:

- A função f-if-on-strip verifica se o sensor óptico está sobre a faixa pintada no solo, executando seu primeiro argumento em caso positivo e o segundo argumento em caso negativo.
- A função f-move move o robô em linha reta por uma distância fixa s .
- A função f-right rotaciona o robô para a direita, em um ângulo de 5° a partir de sua direção atual.
- A função f-left rotaciona o robô para a esquerda, em um ângulo de 5° a partir de sua direção atual.
- A função f-progn2 executa sequencialmente seus dois argumentos.
- A função f-progn3 executa sequencialmente seus três argumentos.
- A função f-progn4 executa sequencialmente seus quatro argumentos.

6.2 Parâmetros do Programa

Para rodar o programa LilGP, foram utilizados os seguintes parâmetros:

- Tempo máximo = 800
- Número máximo de gerações = 100
- Tamanho da população = 500
- Semente randômica = 123
- Método = half and half
- Profundidade inicial = 2-3
- Máxima profundidade = 10
- Fases de procriação = 2
- Fase1 crossover = 0,9
- Fase2 reproduction = 0,1

6.3 Resultados Obtidos

Para uma rodada do programa, obteve-se o melhor *fitness* na geração 58:

=== BEST-OF-RUN ===

- Generation: 58; Nodes: 65 ; Depth: 10; Hits: 155

- Raw Fitness: 155.0000; Standardized Fitness: 162.0000
- Adjusted Fitness: 0.0061
- BEST OF RUN INDIVIDUAL: Time Taken: 800/800; On Strip: 155/317

A árvore obtida foi a seguinte:

```
TREE: (if-on-strip (progn2 (if-on-strip (if-on-strip (if-on-strip (progn3 move(progn3 move
(if-on-strip (if-on-strip move move) right) left) left)(progn2 move move)) (if-on-strip (progn2
(if-on-strip (progn2 (if-on-strip right move) (if-on-strip (if-on-strip left move)move)) (if-on-
strip (if-on-strip move right) (progn2 move left))) left) right)) (if-on-strip (progn3 move right
move) (if-on-strip (if-on-strip (if-on-strip (if-on-strip move (progn3 move move move)) right)
(if-on-strip (if-on-strip move move) (if-on-strip move right))) move)))
```

A Figura 4 mostra o resultado obtido pelo software de melhor *fitness*, onde os pontos marcados com um caractere "o" representam as coincidências entre as coordenadas do robô e a faixa pintada no solo.



Figura 4 – Percurso seguido pelo robô.

7 COMPETÊNCIAS E HABILIDADES DESENVOLVIDAS

A utilização do projeto de robô com programação genética possibilita que o aluno desenvolva habilidades através de uma aplicação prática dos conhecimentos adquiridos na disciplina de Inteligência Artificial associados a conhecimentos de eletrônica e robótica, permitindo que vários conceitos do aprendizado teórico sejam reforçados e vivenciados, possibilitando o desenvolvimento de uma visão crítica interdisciplinar. Considerando-se que este tipo de atividade pode e deve ser realizado por um grupo de alunos, até mesmo como um trabalho final de alguma disciplina de um curso de engenharia, pode-se também esperar um crescimento nas habilidades em relações interpessoais e de planejamento, envolvendo trabalho em equipe, divisão de tarefas, estabelecimento de prazos e metas. Uma possibilidade interessante é a participação de alunos de vários cursos de Engenharia, aplicando

conhecimentos específicos de suas respectivas áreas. Espera-se assim que o aluno de Engenharia desenvolva através deste trabalho, competências para atuação em áreas de desenvolvimento e aperfeiçoamento de produtos bem como em áreas de pesquisa, conseguindo aplicar de forma integrada os conhecimentos mecânica, eletrônica digital e analógica, automação e robótica associados ainda à aplicação de conceitos avançados de programação e Inteligência Artificial.

8 CONCLUSÃO

Dentro do ensino de Engenharia, particularmente nas áreas de eletrônica, automação e robótica, considera-se que projetos práticos semelhantes ao aqui descrito são de grande valia, principalmente por possibilitarem a aplicação prática do aprendizado obtido em diversas disciplinas, motivando o aluno a buscar um aprofundamento nos temas tratados.

Neste contexto, a disciplina de Robótica Móvel apresenta-se como uma excelente oportunidade de agregar diferentes competências adquiridas ao longo do curso de Engenharia. Dentro desta disciplina existe a preocupação de permitir ao aluno a consolidação de conteúdos de disciplinas como Controle Contínuo, Controle Discreto, Interfaces de Computadores, Sistemas Microprocessados, Eletrônica de Potência, Sensores e Atuadores, Protocolos de Comunicação e Engenharia de Software, entre outras. Disciplinas avançadas, normalmente apresentadas como disciplinas optativas podem ser também exploradas. Entre estas, a disciplina de Inteligência Artificial engloba entre as possíveis estratégias o uso de Algoritmos Genéticos e de Programação Genética.

Usando esta abordagem, verificou-se que a utilização de Programação Genética para software embarcado em robô móvel produz resultados bastante interessantes, pois o dispositivo a ser controlado pode se adaptar a situações inusitadas, alterando o seu comportamento sem que haja intervenção humana e atingindo o objetivo esperado. Este tipo de abordagem apresenta ao aluno as vantagens e desvantagens do uso de técnicas de Inteligência Artificial para resolver problemas práticos de engenharia. Por outro lado, basear esta abordagem em robôs móveis, permite ao aluno a fácil visualização dos resultados, quer em simulação, quer usando um robô real. Além disto, a abordagem usada permite facilmente a adoção de outras topologias de robôs móveis, inclusive bípedes ou hexápodes (robôs de patas), bem como a operação com outras tecnologias de sensores.

Assim, uma proposta para trabalhos futuros seria a aplicação desta metodologia em outras topologias de robôs móveis e com diferentes configurações de sensores. Outra proposta é aplicação e comparação de outras técnicas de inteligência artificial, a exemplo de redes neurais e de lógica *fuzzy*.

REFERÊNCIAS BIBLIOGRÁFICAS

CORREIA, L. Tópicos introdutórios sobre robôs móveis. Faculdade de Ciências Universidade de Lisboa. Portugal, 2006, p. 1 – 9.

KOZA, J. R. & POLI, R. Genetic Programming. In: BURKE E. K. & KENDALL G. Search Methodologies - Introductory Tutorials in Optimization and Decision Support Techniques. Springer US, 2005, p. 127-159

KOZA, J. R. & RICE, J. P. Automatic Programming of Robots using Genetic Programming Proceedings of the Tenth National Conference on Artificial Intelligence, The MIT Press, 1992, p. 194-201

NORDIN, P. & BANZHAF, W. Genetic Programming Controlling a Miniature Robot. In: WORKING NOTES FOR THE AAAI SYMPOSIUM ON GENETIC PROGRAMMING, AAAI, 1995, p. 61-67

LAZARUS, C. & HU, H. Using Genetic Programming to Evolve Robot Behaviors. In: Proceedings of the 3 rd British Conference on Autonomous Mobile Robotics & Autonomous Systems, 2001

KOZA, J. R. Genetic Programming. The MIT Press, 1992

GENETIC PROGRAMMING FOR MOBILE ROBOT

Abstract: *The proposal of a Mobile Robotics discipline in an Engineering Course has a goal of propitiate an integration of several competences necessary for a professional formation. This work describes one among several experiments to be explored by students in a Mobile Robotics discipline, emphasizing the Artificial Intelligence aspect. The Evolutionary Programming is a discipline that has large applicability in areas of Engineering as well as in physics and biological sciences, being divided basically in Genetic Algorithms and Genetic Programming. This document also demonstrates the applicability of Genetic Programming in automatic development of embedded control software for a microcontrolled mobile robot, which must be able to follow a stripe painted on the floor. The system validation shall be done in an environment and robot simulator software and further applied in a real case and be utilized as part of a practical work for Electronic or Automation Engineering courses.*

Key-words: *genetic programming, mobile robotic.*