

# PROGRAMAÇÃO GENÉTICA E REDES NEURAIS PARA CONTROLE DE SISTEMAS DINÂMICOS

---

Artur Henrique G. C. Alves

Orientador: Prof. Dr. Vinícius Veloso de Melo

Programa de Pós-Graduação em Ciência da Computação

Universidade Federal de São Paulo - UNIFESP

# MPC: Model Predictive Control

Abordagem para controle de sistemas dinâmicos

Utilizado em aplicações autônomas

Ajusta a atuação do sistema de acordo com a diferença entre o comportamento desejado e o obtido

Permite o controle de sistemas complexos sem necessidade de conhecimento teórico sobre o funcionamento da aplicação por parte do desenvolvedor

Utiliza uma técnica de modelagem para estimar estados de sistemas lineares e não-lineares

# MPCDesafios

Identificação de sistemas multivariáveis

Monitoramento de desempenho e diagnósticos

Estimativa de estados não-lineares

Controle de sistemas em lote

# MPCDesafios

Identificação de sistemas multivariáveis

Monitoramento de desempenho e diagnósticos

Estimativa de estados não-lineares: atividade mais crítica e custosa

Controle de sistemas em lote

# MPC Formulação

Controle preditivo baseado em modelo linear de tempo discreto

$$x(k + 1) = Ax(k) + Bu(k)$$

Onde  $x(k)$  e  $u(k)$  representam o estado e o controle de entrada, respectivamente

A cada instante  $k$ , o controle efetua a predição de  $p$  instantes futuros, onde  $p$  é o horizonte de predição

A predição é otimizada utilizando uma determinada função de custo  $J$

Somente o controle do próximo instante, ou seja,  $u(k + 1)$ , é implementado

O controle desliza o horizonte de predição para frente, onde  $k = k + 1$

# MPC Questões práticas

Como avaliar a viabilidade do problema descrito anteriormente?

O algoritmo sempre gera uma ação de controle implementável?

O sistema de controle é estável em um circuito fechado?

Circuito aberto: controle sem retorno do estado do sistema (modelo)

Circuito fechado: controle com retorno do estado do sistema (aplicação)

Qual é o desempenho em circuito fechado do controle otimizado para circuito aberto?

# MPC Viabilidade

Por algum distúrbio, o controle otimizado pode gerar um controle inviável em um instante futuro

A otimização em circuito aberto pode levar o circuito fechado a um estado fora da região factível

Na prática, costuma-se utilizar restrições fortes e fracas

- Restrição forte: limites específicos para o controle

- Restrição fraca: penalizações aplicadas à função-objetivo

Problemas importantes

- Magnitude da violação versus seu tempo de duração

  - Violações longas e de baixa magnitude podem ter grande impacto sem serem restringidas

- Restrições fracas podem descartar uma solução viável

# MPC Desempenho

Somente a primeira ação de controle calculada é implementada

A sequência de controle real pode divergir de forma significativa daquela calculada em um determinado instante

O objetivo do horizonte finito da predição pode ter pouca relação com o valor obtido após a implementação das ações

No pior dos casos, as ações calculadas podem levar o sistema a sair da zona factível



# MPC Modelos não-lineares

Problemas não-lineares podem ser controlados seguindo a mesma estrutura

As dificuldades enfrentadas anteriormente crescem significativamente

- Viabilidade

- Discrepância de desempenho entre modelo e realidade

Otimização destes problemas não garante convergência para ótimos globais

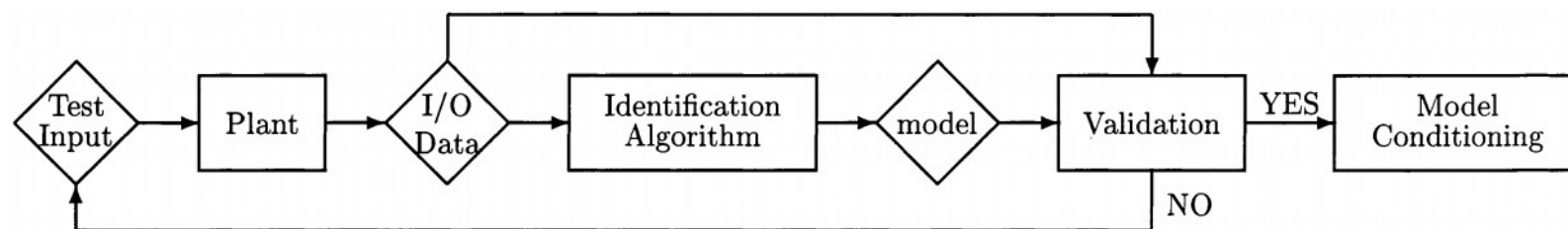
- Aplicações podem não exigir desempenho ótimo

- Melhoria gradual da solução ou uma solução simplesmente factível podem ser suficientes

Sistemas podem ser linearizados antes da otimização

# MPC Modelagem

Os passos tradicionais para modelagem de uma aplicação são, geralmente:



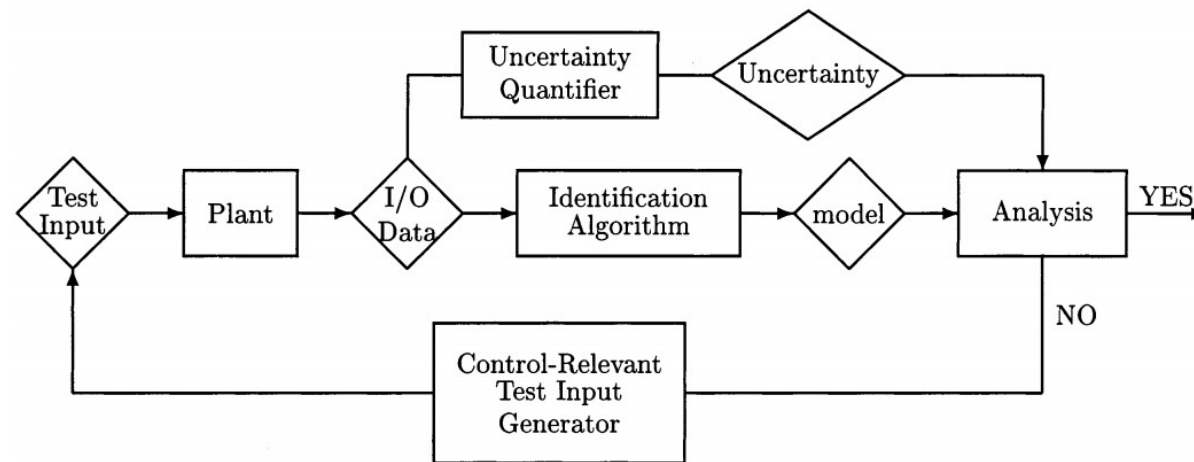
*M. Mojari, J. H. Lee. Model predictive control: past, present and future.*

# MPC Modelagem

- Os estímulos de entrada podem ser identificados de formas diferentes:
- SISO: cada entrada é perturbada separadamente e as saídas são medidas separadamente
  - O desempenho dos modelos SISO pode ser muito bom individualmente, mas isto não se repete no modelo combinado multivariável
- MISO: caso as perturbações sejam desenvolvidas independentemente, a análise de várias entradas simultaneamente também pode apresentar baixo desempenho
  - Em processos altamente interativos, correlações podem aparecer nas saídas
- MIMO: pode levar em considerações correlações e ainda identificar características estocásticas do processo

# MPC Modelagem

Uma abordagem integrada de identificação e controle pode auxiliar nestes pontos:



*M. Mojari, J. H. Lee. Model predictive control: past, present and future.*

# MPCModelagem

A abordagem integrada apresenta:

Geração de entradas de teste ótimas de acordo com informações da planta, objetivos do circuito fechado e restrições existentes

Quantificação da incerteza do modelo

Análise de estabilidade e desempenho de acordo com o modelo e sua incerteza

# MPC Modelagem

Na prática, o sucesso do MPC se deve a sua eficiência

No entanto, em problemas não-lineares, não há um método estabelecido para a modelagem

A pesquisa por modelagem empírica é de grande importância

Determinação da estrutura do modelo

Os modelos podem ser desenvolvidos em cima de estruturas estabelecidas, como NARMAX, Wiener, Hammerstein, etc.

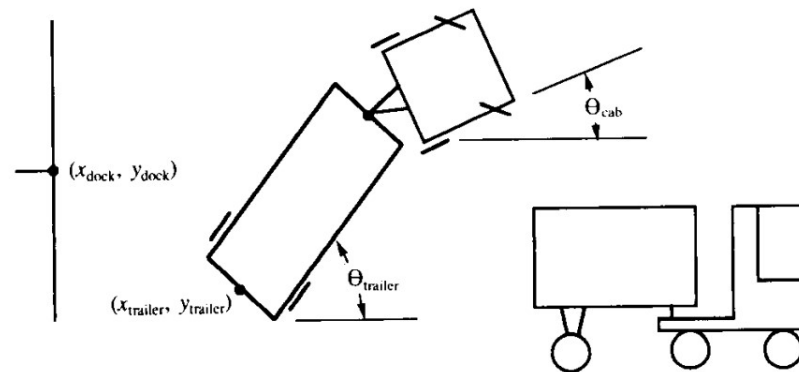
Também podem ser desenvolvidos sem estruturas específicas, como por redes neurais ou abordagens de computação evolutiva

Técnicas de inteligência computacional também podem aparecer na otimização de controles não-lineares

# MPCRedes Neurais

Em 'Neural Networks for Self-Learning Control Systems', é proposto um controle baseado em redes neurais para estacionar uma carreta em uma baia de carga e descarga

Redes Adaline são utilizadas tanto para modelar o sistema como para otimizar os valores de ação de controle

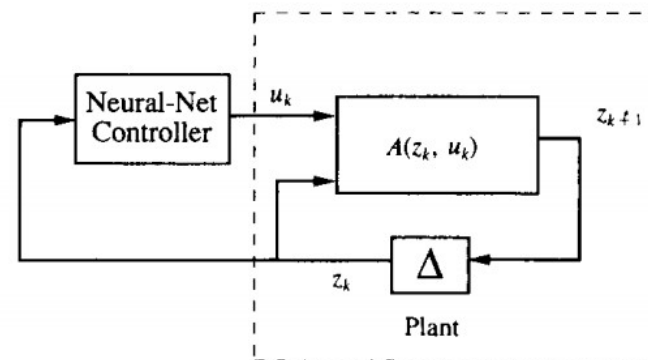
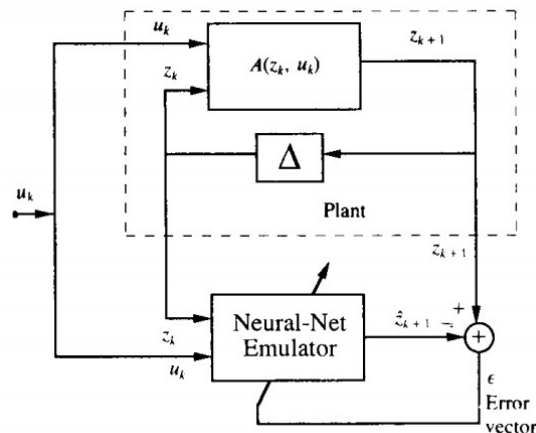


*Derrick H. Nguyen, Bernard Widrow. Neural Networks for Self-Learning Control Systems*

# MPCRedes Neurais

Nesta abordagem, uma rede neural é treinada para emular o comportamento da planta

Esta rede é utilizada posteriormente para treinar a rede que irá efetuar o controle



*Derrick H. Nguyen, Bernard Widrow. Neural Networks for Self-Learning Control Systems*



# MPCRedes Neurais

No problema, o veículo só pode dar ré

A cada instante de tempo, ele sofre um deslocamento fixo para trás

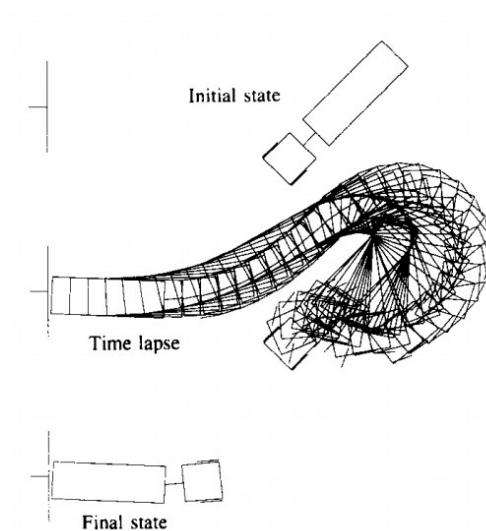
O controle deve determinar o valor do esterçamento da carreta

A complexidade do problema está na possibilidade da “dobra” da carreta

# MPCRedes Neurais

Durante o treinamento, o controle recebeu tarefas progressivamente mais difíceis

Nos testes, a carreta inicia em posições e ângulos aleatórios

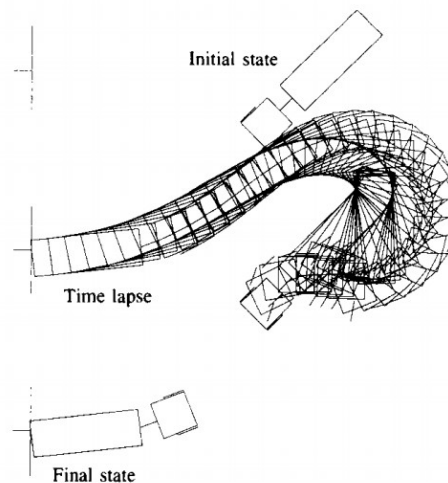


*Derrick H. Nguyen, Bernard Widrow. Neural Networks for Self-Learning Control Systems*

# MPCRedes Neurais

Penalizações podem ser adicionadas à função-objetivo do controle

Neste exemplo, o controle deve minimizar o uso do volante



*Derrick H. Nguyen, Bernard Widrow. Neural Networks for Self-Learning Control Systems*

# MPCRedes Neurais

Uma das grandes desvantagens de se utilizar redes neurais para controle é que não há como analisar o modelo resultante, pela natureza de caixa-preta da técnica

Esta impossibilidade de análise exige que a rede seja muito bem elaborada para evitar especialização e, portanto, queda de robustez

Uma alternativa é a Programação Genética

# MPCProgramação Genética

A programação genética é um método evolutivo capaz de gerar modelos MISO

Os modelos são estruturados em forma de árvore e são gerados em populações

Como em outras técnicas evolutivas, os indivíduos com melhor desempenho têm maior chance de sobrevivência

O gerador de indivíduos tem a sua disposição entradas, funções básicas e constantes

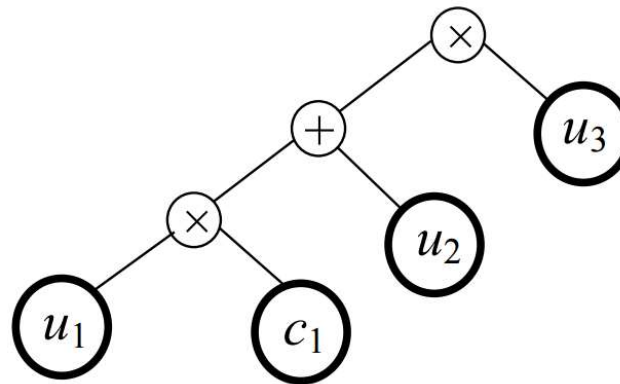
As combinações utilizam subárvores dos indivíduos pais

As mutações substituem elementos aleatoriamente

Um operador de permutação troca subárvores dentro de um único indivíduo

# MPC Programação Genética

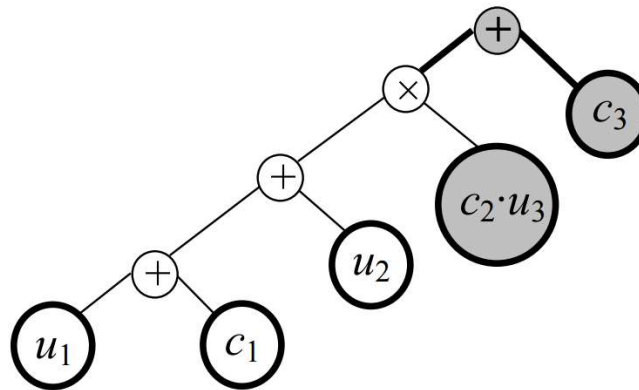
Como exemplo, a equação-modelo  $y = (c_1 * u_1 + u_2) * u_3$  é representada pela seguinte árvore:



*Benjamin Grosman, Daniel R. Lewin. Automated Nonlinear Model Predictive Control using Genetic Programming*

# MPC Programação Genética

Modificações podem ser efetuadas nos indivíduos, como o aumento de constantes para condizer com o grau de liberdade do modelo:



*Benjamin Grosman, Daniel R. Lewin. Automated Nonlinear Model Predictive Control using Genetic Programming*

# MPC Programação Genética

Após a modificação de parâmetros, estes devem ser estimados considerando-se o erro quadrático em relação à saída do processo modelado

Com os parâmetros estimados, cada indivíduo tem sua *fitness* calculada para os próximos passos evolutivos

O cálculo da *fitness* pode acrescentar características de grande importância ao algoritmo, como penalizações para indivíduos altamente complexos

Se a referência de complexidade for atualizada de acordo com os melhores indivíduos, diminui-se a chance de descarte de melhores alternativas complexas



# MPC Programação Genética

O modelo gerado por PG deve seguir a seguinte forma:

$$y(k) = f\{u(k), u(k-1), u(k-2), \dots, y(k-1), y(k-2), \dots\}$$

A otimização paramétrica parte do último instante com ações e saídas conhecidas e estima recursivamente os parâmetros até o fim do horizonte definido

Otimizações com um passo de distância levam a modelos da forma  $y(k) = \kappa y(k-1)$ , que são fracos para predição a longo prazo

# MPC Programação Genética

Um modelo não-linear apresenta...

Um MISO para cada saída da planta da forma:

$$\hat{y}(k) = f(\hat{y}(k-1|k), u(k-1|k))$$

Onde  $\hat{y}(k)$  é a saída predita,  $\hat{y}(k-1|k)$  são as  $P$  saídas anteriores já conhecidas e  $u(k-1|k)$  são as  $P$  entradas anteriores já efetuadas

# MPC Programação Genética

Um modelo não-linear apresenta...

Valores de  $P$  e  $M$  representando o horizonte de predição e o de controle, respectivamente, onde  $M \leq P$ . A resposta do sistema é mais agressiva conforme  $M$  se aproxima de  $P$

Uma matriz de dados históricos para a predição da influência de valores anteriores nas saídas futuras, com  $P$  tuplas

Uma determinada função-objetivo de custo  $J$  a ser minimizada pelo otimizador

# MPC Programação Genética

O primeiro problema apresentado em 'Automated nonlinear model predictive control using genetic programming' é o controle de um tanque de mistura de água doce e salgada

Os *set points* do sistema são a concentração de sal efluente e o nível de água no tanque, enquanto as entradas são os fluxos de água salgada e doce

Os dois melhores modelos encontrados foram:

$$C(k) = 0.818C(k-1) + 0.134Q_S(k-1) - 0.078Q_W(k-2) + 0.68$$

$$H(k) = 0.890H(k-1) + 0.170Q_S(k) + 0.167Q_W(k) - 0.061$$

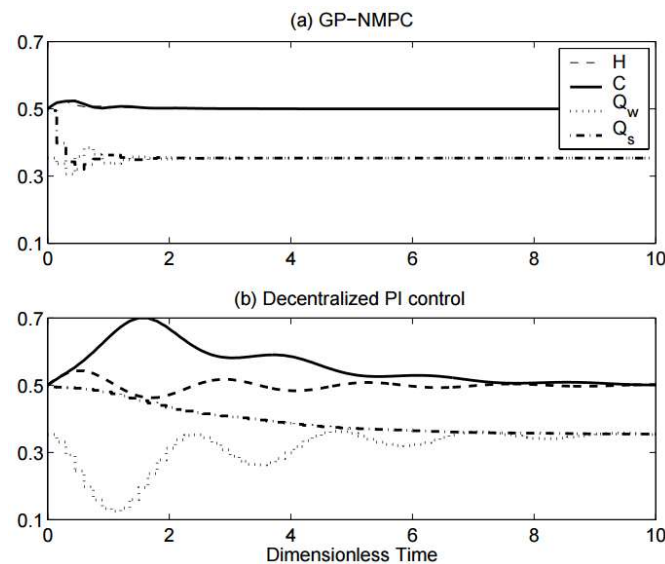
Ambos são lineares, resultado condizente com o problema

Estratégias lineares de MPC já foram aprovadas em simuladores para esta aplicação

# MPC Programação Genética

Os modelos foram comparados a um controle PI em duas situações:

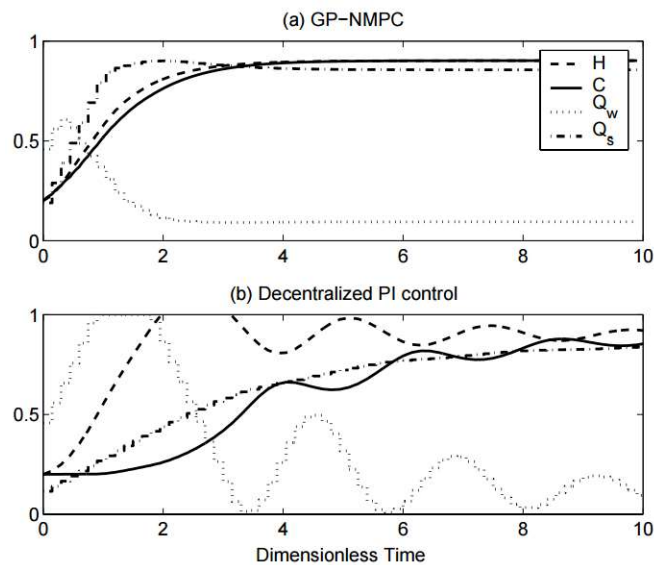
Um pulso de 40% na entrada de água salgada quando o sistema está estável



# MPC Programação Genética

Os modelos foram comparados a um controle PI em duas situações:

Uma alteração de *set points* de  $[0.2, 0.2]$  para  $[0.9, 0.9]$



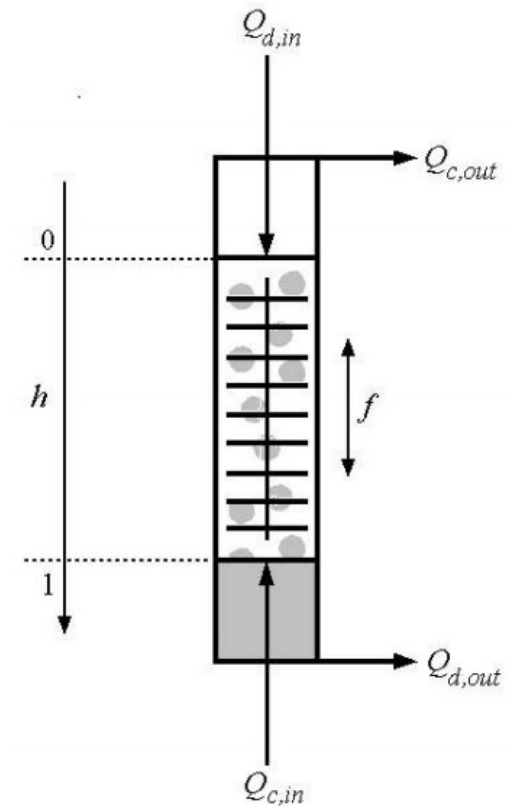
# MPC Programação Genética

O segundo problema apresentado em 'Automated nonlinear model predictive control using genetic programming' é o controle de uma coluna de extração líquido-líquido Karr

Este processo consiste em utilizar uma coluna com um eixo oscilatório e discos perfurados para a separação de solvente de uma solução

As saídas são a concentração do efluente da fase dispersa e o *holdup* da coluna

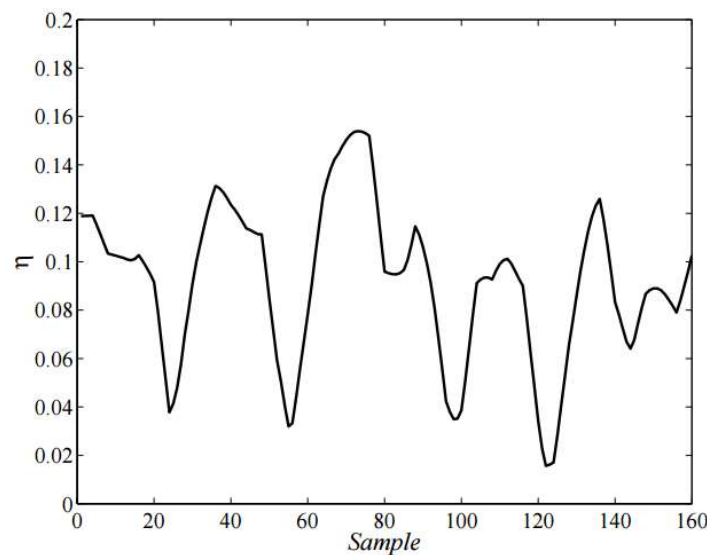
As entradas são a taxa de fluxo da fase contínua, a taxa de fluxo da fase dispersa e a frequência oscilatória da coluna



Benjamin Grosman, Daniel R. Lewin. Automated Nonlinear Model Predictive Control using Genetic Programming

# MPC Programação Genética

Após a amostragem de comportamento da coluna, dados para modelagem e predição foram separados e pode-se conferir o comportamento transiente do sistema





# MPC Programação Genética

Os dois melhores modelos encontrados foram:

$$\begin{aligned} X_{out} &= 0.712X_{out}(k-1)\sqrt{f(k)} + 0.0347 \\ \eta(k) &= 1.088\eta(k-1) - 0.147\eta(k-3) + 0.209Q_{c,in}(k) \\ &\quad - 0.342 \exp\left(0.930\eta(k-1)^{2.195}Q_{d,in}(k) - 0.0147f(k)\right) + 0.844 \end{aligned}$$

# MPC Programação Genética

Este controle foi comparado a um controle linear multivariável criado especificamente para este problema

O controle linear apresenta um resultado perfeito em aproximadamente 600s, enquanto o controle baseado em PG estabiliza o sistema em um terço deste tempo, apesar de apresentar pequena dificuldade antes de estabilizar

Os ajustes necessários para acelerar o controle linear diminuiriam sua robustez

O trabalho conclui que a PG pode ser utilizada para a geração rápida e eficiente de modelos para controles preditivos não-lineares

# MPC Programação Genética

Outro trabalho, 'Generating Robust Control Equations with Genetic Programming for Control of a Rolling Inverted Pendulum', utiliza PG para a construção de um MPC

O problema apresentado consiste em duas etapas:

1. Balançar um bastão apoiado em cima de um veículo de uma posição pendurada para que se estabilize na vertical
2. Mover o veículo com o bastão equilibrado para um determinado ponto no espaço

A PG foi escolhida por gerar modelos robustos, compreensíveis e enxutos

# MPC Programação Genética

O estado do sistema é representado pelo vetor  $y = (\dot{\theta}, \dot{x}, \theta, x)$ , onde cada variável corresponde respectivamente à velocidade angular do bastão, à velocidade do veículo, ao ângulo do bastão e à posição do veículo

Para a geração dos indivíduos, os seguintes conjuntos não-terminais e terminais foram utilizados:

$$\mathcal{F} = \{+, -, *, \%, \sin, \cos\}$$
$$\mathcal{T} = \{\dot{\theta}, \dot{x}, \theta, d, 1.0, 10.0, -1.0, 2.0, 3.0, 4.0, \pi\}$$

A *fitness* dos indivíduos é composta por duas simulações, uma em cada etapa, pois uma única simulação seria incapaz de gerar um controle robusto

# MPC Programação Genética

Trinta simulações foram executadas

Destas, doze equações foram capazes de estabilizar o bastão

Após uma análise teórica do equilíbrio do bastão com o veículo na posição-alvo, seis equações restaram

Destas, cinco são assintoticamente estáveis (a sexta contém uma divisão por zero e não é analisável)

# MPC Programação Genética

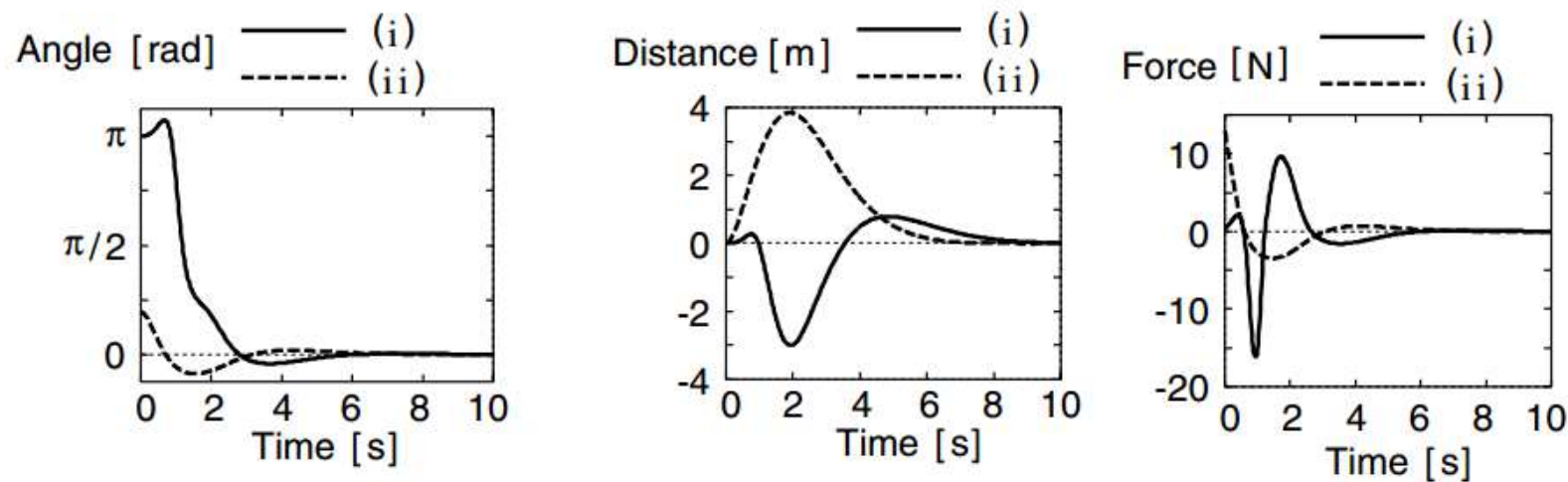
As duas melhores equações foram:

$$\begin{aligned} force_1(\dot{\theta}, \dot{x}, \theta, d) \\ = 5\dot{\theta} + \sin(\dot{\theta}) + \dot{x} + \theta(\theta - 3.03) + 25.15 \sin(\theta) - 0.25 \sin(\theta d) + 0.33d \end{aligned}$$

$$\begin{aligned} force_2(\dot{\theta}, \dot{x}, \theta, d) \\ = \dot{x} + 9.09\theta + (6.0\dot{\theta} + 19.1\theta + d)[\cos(\theta) - \sin\left(\cos\left(\frac{d(\cos(\theta) - 0.02)}{\cos(2\cos(\theta))}\right)\right)] \end{aligned}$$

# MPC Programação Genética

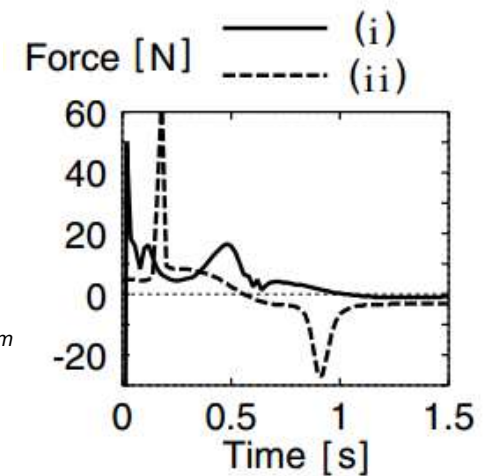
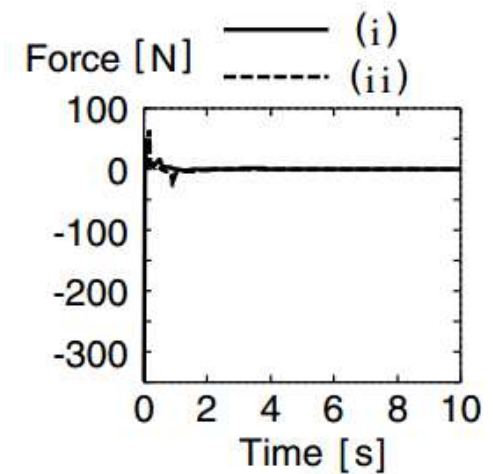
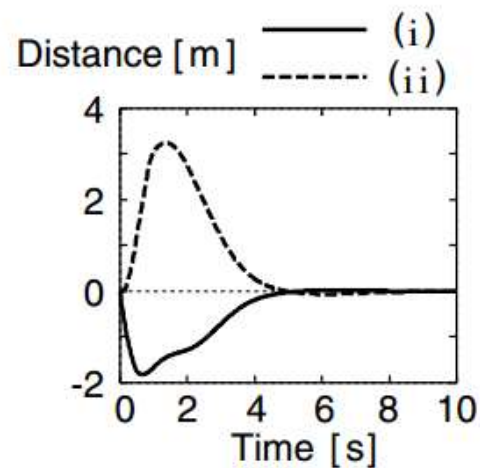
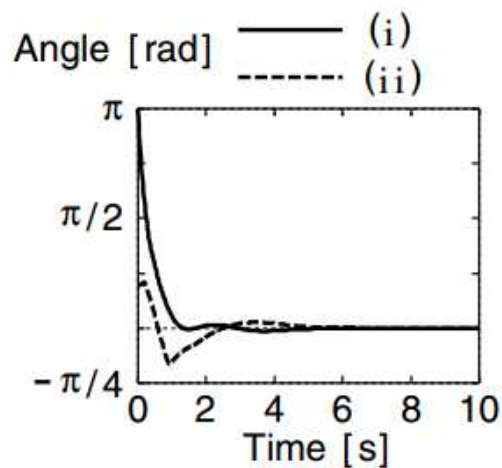
A primeira equação apresenta os seguintes comportamentos em teste:



*Hiroaki Shimooka, Yoshiji Fujimoto. Generating Robust Control Equations with Genetic Programming for Control of a Rolling Inverted Pendulum*

# MPC Programação Genética

Enquanto a segunda se comporta desta forma:



Hiroaki Shimooka, Yoshiji Fujimoto. Generating Robust Control Equations with Genetic Programming for Control of a Rolling Inverted Pendulum



# MPC Programação Genética

Por fim, as duas equações são analisadas quanto à robustez, ao alterar-se os valores de três parâmetros:

	Control Eq. (10)			Control Eq. (11)		
Parameters	$m_c$	$m_p$	$l$	$m_c$	$m_p$	$l$
Max (%)	101	125	103	120	150	140
Min (%)	94	50	90	50	50	92

*Hiroaki Shimooka, Yoshiji Fujimoto. Generating Robust Control Equations with Genetic Programming for Control of a Rolling Inverted Pendulum*