Estudo Comparativo do Projeto de Controladores PI Multivariáveis baseados em Normas de Sinais e Sistemas

Vitor Vidal de Negreiros* Eduardo Nunes Gonçalves**

* Programa de Pós-graduação em Engenharia Elétrica UFSJ - CEFET-MG, Av. Amazonas 7675, Belo Horizonte MG (e-mail: vitor.negr@gmail.com).

** Departamento de Engenharia Elétrica - CEFET-MG, Av. Amazonas 7675, Belo Horizonte, MG (e-mail: eduardong@cefetmg.br)

Abstract: Multivariable PI controllers for systems with time delays can be tuned with applied optimization using different formulations for the objective function. Strategies based on H_∞ and H_2 system norms impose the need of approximation of the time delays by Padé functions. On the other hand, optimization of a performance index based on signal norms rather than system norms are more direct and straightforward to formulate, although more time-consuming. In this paper, it is analysed how the different ways to formulate the optimization problem in terms of system and signal norms can impact on the closed loop performance and on computational effort. A centralized PI controller is tuned with Differential Evolution method using a suite of different control cost functions. Simulations are performed on a 2×2 distillation column and the results compared.

Resumo: Controladores PI multivariáveis para sistemas com atrasos de transporte podem ser sintonizados por meio de otimização usando-se diferentes formulações da função objetivo. Estratégias baseadas em normas H_∞ e H_2 colocam a necessidade de aproximar os atrasos de transporte com funções de Padé. Por outro lado, a otimização de um índice de desempenho baseado em normas de sinais ao invés de normas de sistemas é mais direta e simples de ser formulada, embora o custo computacional seja maior. Neste trabalho é analisado como as diferentes maneiras de formular o problema de otimização em termos de normas de sistemas e de sinais impacta no desempenho do sistema em malha fechada e no esforço computacional. Um controlador PI centralizado é sintonizado usando o método Evolução Diferencial e uma suíte de diferentes funções de custo. Simulações são executadas num modelo de coluna de destilação 2×2 e os resultados são comparados.

Keywords: Optimization on Control and Automation systems; MIMO control; Centralized PI; Differential Evolution Method; Control Theory.

Palavras-chaves: Otimização em Sistemas de Controle e Automação; Controle MIMO; Controle PI centralizado; Método Evolução Diferencial; Teoria de Controle.

1. INTRODUÇÃO

A maior parte dos sistemas industriais são do tipo MIMO (do inglês *multiple-input multiple-output*). Em contraste com o caso SISO (do inglês *single-input single-output*), sistemas MIMO são de difícil controle e estabilização devido à existência de interações entre as diferentes malhas de controle do sistema. Cada entrada causa efeitos em todas as saídas, fazendo com que a sintonia do controlador seja mais complexa.

Dentre as estruturas que podem ser aplicadas, os controladores MPC (do inglês *Model Predictive Control*) e ANN-Fuzzy (baseados em redes neurais artificiais e lógica Fuzzy) têm recebido bastante atenção na literatura (Shah and Engell, 2011; Kramer and Morgado-Dias, 2018). Não

obstante, esses esquemas são de difícil implementação prática na indústria e possuem entendimento não trivial. Os tradicionais controladores PI/PID, por sua vez, são de fácil compreensão e implementação computacional. Devido a isso, as estruturas PI/PID multivariáveis continuam a ser essenciais para o controle de sistemas MIMO. As principais estruturas PI/PID MIMO são: controlador centralizado, descentralizado com e sem desacoplador e esparso.

O controlador descentralizado é uma estrutura simples e funciona bem para sistemas com acoplamento entre malhas limitado. Algumas das principais técnicas de projeto dessa classe de controladores podem ser encontradas em L. Luyben (1986), Hovd and Skogestad (1994), Campestrini et al. (2009) e Xiong et al. (2007). Apesar de simples e práticos, controladores descentralizados possuem desempenho insatisfatório para sistemas com alto acoplamento. Para

* Os autores agradecem os apoios das agências CAPES, CNPq, e FAPEMIG, [APQ-02943-15] - Edital 01/2015 - Demanda Universal.

essas situações, pode-se implementar um desacoplador ou utilizar uma estrutura centralizada ou esparsa.

A introdução do desacoplador em complemento ao controlador descentralizado torna o sistema de malha fechada mais complexo, além de que o desempenho do desacoplador é muito dependente do quão próximo o modelo usado é do sistema real. Em razão disso, os controladores centralizados e esparsos são ideais para sistemas com alto grau de acoplamento.

O controlador centralizado é uma estrutura que consiste numa matriz cheia de ordem N , em que N é o número de entradas ou de saídas do sistema (considerando um sistema quadrado). Cada elemento da matriz é um controlador PI/PID.

Os controladores esparsos são um caso intermediário entre o descentralizado e centralizado, onde alguns elementos fora da diagonal são nulos, diminuindo o número total de controladores PI individuais. Controladores centralizados são sintonizados tanto por técnicas analíticas, como as apresentadas em Shen et al. (2014) e Lieslehto (1996), e também por estratégias de otimização evolucionária.

Exemplos de aplicações de métodos evolucionários no projeto de controladores MIMO podem ser encontrados em Coelho and Wicthoff Pessoa (2011), Coelho and Mariani (2012), Iruthayarajan and Baskar (2009), Iruthayarajan and Baskar (2010), Turco Neto et al. (2017), Zeng et al. (2015) e Devikumari and Velappan (2015).

Com algumas exceções, a maioria dos trabalhos que lançam mão de estratégias de otimização utilizam os tradicionais índices ISE (Integral do erro ao quadrado), IAE (Integral do erro absoluto), ITSE (Integral do erro ao quadrado multiplicado pelo tempo) e ITAE (Integral do erro absoluto multiplicado pelo tempo) e as normas de sistemas H_∞ e H_2 como função objetivo. Alguns trabalhos se dedicaram à comparação de alguns desses índices, mas para plantas SISO e sem atrasos de transporte (Balestrino et al., 2006; Meenakshi Kishnani and Gupta, 2014).

A flexibilidade em definir a função objetivo é uma vantagem que a otimização dá ao engenheiro de controle, mas ao mesmo tempo, ela coloca questões importantes. Qual o impacto da escolha da função objetivo no comportamento do sistema em malha fechada? Qual a diferença entre usar normas de sistemas e de sinais?

A principal contribuição deste trabalho é identificar as diferenças entre a utilização de diferentes funções objetivo aplicadas ao projeto de controladores PI MIMO para plantas com atraso. Em especial, busca-se identificar quais as vantagens e desvantagens, no contexto do controle baseado em modelo de referência, em se utilizar normas de sistemas (onde a planta é comparada a um modelo de referência) e normas de sinais (em que a saída simulada do sistema é comparada a um sinal de referência).

Para endereçar as questões colocadas, um conjunto de funções objetivo é estabelecido e usado com algoritmo de otimização Evolução Diferencial (DE, do inglês, *Differential Evolution*) para sintonizar uma série de controladores centralizados PI para o tradicional modelo de coluna de destilação desenvolvido em Wood and Berry (1973). Os resultados obtidos são comparados em termos das caracte-

terísticas temporais (tempo de subida, sobressinal e tempo de acomodação), esforço computacional e o índice ISE agregado (índice ISE para todas as malhas do sistema somadas).

As próximas seções estão organizadas da seguinte maneira. Na seção 2 é estabelecido o conjunto de funções objetivo que serão comparadas. Na seção 3 os princípios fundamentais do algoritmo DE são apresentados. A seção 4 apresenta os resultados e simulações. Por fim, na seção 5 são apresentadas as conclusões e as linhas gerais para pesquisa futura.

2. FUNÇÕES OBJETIVO APLICADAS À OTIMIZAÇÃO DE CONTROLADORES MIMO

A escolha da função objetivo a ser utilizada no projeto de controle depende do problema e dos objetivos do projetista. A seguir é apresentado uma série de normas de sinais e de sistemas que são comumente usadas para o projeto de controladores MIMO tendo em vista o problema de rastreamento, conforme revisado em Reynoso-Meza and Martinez (2014).

São consideradas duas famílias de funções objetivo, a primeira são índices que podem ser otimizados diretamente e expressam um objetivo de controle em si mesmos. A segunda são funções objetivo que se aplicam ao controle baseado em modelo de referência, isto é, são funções custo que fazem o sistema em malha fechada se aproximar de um sistema ou sinal definido previamente como referência.

2.1 Funções objetivo sem modelo de referência

Alguns critérios de desempenho comumente usados são:

- Integral do erro absoluto

$$IAE = \int_{t_0}^{T_f} |e(t)| dt, \quad (1)$$

- Integral do erro absoluto multiplicado pelo tempo

$$ITAE = \int_{t_0}^{T_f} t |e(t)| dt \quad (2)$$

- Integral do erro ao quadrado

$$ISE = \int_{t_0}^{T_f} e(t)^2 dt \quad (3)$$

- Integral do erro ao quadrado multiplicado pelo tempo

$$ITSE = \int_{t_0}^{T_f} t e(t)^2 dt \quad (4)$$

Seja $C(s)$ o controlador PI centralizado e $G(s)$ um sistema quadrado com atraso de transporte a ser controlado. A função sensibilidade do sistema em malha fechada é definida como:

$$S(s) = (I + C(s)G(s))^{-1} \quad (5)$$

Seja a função sensibilidade complementar associada:

$$T(s) = I - S(s) \quad (6)$$

Assim, pode-se definir as seguintes funções objetivo:

- Norma H_∞ da função sensibilidade

$$J_S = \|S(s)\|_\infty \quad (7)$$

- Norma H_∞ da função sensibilidade complementar

$$J_T = \|T(s)\|_\infty \quad (8)$$

2.2 Índices de controle baseados em modelo de referência

Seja $T_{ri}(s)$ um modelo de referência que possua a dinâmica desejada pelo projetista. Assim, define-se as seguintes funções objetivo:

- Norma H_∞ da diferença entre o sistema em malha fechada e o modelo de referência

$$J_{MF\infty} = \|T_{zw}(s) - T_{ri}(s)\|_\infty \quad (9)$$

- Norma H_2 da diferença entre o sistema em malha fechada e o modelo de referência

$$J_{MF2} = \|T_{zw}(s) - T_{ri}(s)\|_2 \quad (10)$$

- Integral do sinal E_1 ao quadrado, em que $E_1(t) = y_{r1}(t) - y(t)$ é a diferença de um sinal de saída de referência em relação ao sinal de saída simulado do sistema em malha fechada, tendo como *setpoint* uma função degrau unitário.

$$J_{MFISE} = \int_{t=t_0}^{T_f} E_1(t)^2 dt \quad (11)$$

As funções de custo (9), (10) e (11) foram simuladas para dois modelos de referência distintos, um com dinâmica temporal lenta (modelo 1), e outro com dinâmica temporal mais rápida (modelo 2).

O modelo de referência 1 é dado por:

$$T_{r1}(s) = \begin{bmatrix} \frac{1e^{-s}}{20s^2+15s+1} & 0 \\ 0 & \frac{1e^{-3s}}{20s^2+15s+1} \end{bmatrix} \quad (12)$$

O modelo de referência 2 é dado por:

$$T_{r2}(s) = \begin{bmatrix} \frac{4.87e^{-s}}{12.23s^2+9.49s+4.88} & 0 \\ 0 & \frac{2.47e^{-3s}}{18.75s^2+11.16s+1.47} \end{bmatrix} \quad (13)$$

Conforme apresentado em R. Gonçalves da Silva et al. (2018), a escolha do modelo de referência deve obedecer a certas restrições sob o risco do controlador resultante não ter o desempenho desejado. Assim, ao usar um sinal ao invés de um sistema de referência, o projetista tem mais liberdade em definir as características da saída do sistema em malha fechada.

3. FUNDAMENTOS DO MÉTODO EVOLUÇÃO DIFERENCIAL

O método Evolução Diferencial foi introduzido em Storn and Price (1997). Trata-se de uma técnica evolucionária

que ganhou notoriedade devido a sua formulação simples, implementação prática e excelentes resultados com funções multimodais e não-diferenciáveis.

O método utiliza N_p vetores de dimensão D como população base que é evoluída a partir do cálculo sucessivo das operações de mutação, cruzamento e seleção. Essas três operações são definidas a seguir em forma sequencial.

Seja a população inicial P dada por:

$$P = x_i \text{ com } i = 1, 2, \dots, N_p \quad (14)$$

- Inicialização da população inicial:

$$x_i = \underline{\lambda_j} + (\overline{\lambda_j} - \underline{\lambda_j}) \times rand(1, D)$$

Em que $rand(1, D)$ gera vetores de dimensão $1 \times D$ com números aleatórios de 0 a 1 e $\underline{\lambda_j} \leq x_i \leq \overline{\lambda_j} \forall i, j$.

Começa o processo de iteração em $G = 1$ até $G = G_{max}$, em que G_{max} é o número máximo de iterações. As operações do método DE são:

- (1) Mutação: para cada i -ésimo vetor pai, um vetor mutante é gerado de acordo com a seguinte regra:

$$v_{i,G+1} = x_{r_1,G} + F(x_{r_2,G} - x_{r_3,G}) \quad (15)$$

Em que r_1, r_2 e r_3 são todos números aleatórios de 1 até N_p e sempre diferentes uns dos outros. Além disso, F varia aleatoriamente entre 0.5 e 1.

- (2) Cruzamento: um conjunto de vetores candidatos é gerado a partir da mistura de vetores mutantes com vetores da população pai de acordo com a seguinte regra:

if $rand() \leq CR$ or $j = rnbr(i)$,

$$u_{ij,G+1} = v_{ij,G+1}$$

else

$$u_{ij,G+1} = x_{ij,G}$$

Em que $rand()$ é um gerador de números aleatórios entre 0 e 1. CR é a constante de Cruzamento, também entre 0 e 1, tendo sido adotado nesse trabalho $CR = 0,9$. A função $rnbr(i)$ gera um índice aleatório na faixa de 1 até D .

- (3) Seleção: avalia o valor da função objetivo para cada i -ésimo vetor candidato e para os vetores da população pai aplicando a seguinte regra:

if $f(u_{i,G+1}) \leq f(x_{i,G})$

$$x_{i,G+1} = u_{i,G+1}$$

else

$$x_{i,G+1} = x_{i,G}$$

A variante do método DE implementada para esse trabalho é a DE/rand/1/bin. Mais detalhes sobre o método e avanços recentes podem ser encontrados em Das and Suganthan (2016).

4. SIMULAÇÕES E RESULTADOS

Para comparação das diferentes funções objetivo, o tradicional modelo de Woody & Berry é utilizado. O modelo é um

sistema quadrado 2×2 com alto grau de acoplamento entre as malhas e atrasos de transporte. Todas as simulações foram feitas com processador Intel Core i5 de dois núcleos e 6GB de memória RAM.

O modelo é dado por:

$$G(s) = \begin{bmatrix} \frac{12.8e^{-s}}{16.7s+1} & \frac{18.9e^{-3s}}{21.0s+1} \\ \frac{6.6e^{-7s}}{10.9s+1} & \frac{19.4e^{-3s}}{14.4s+1} \end{bmatrix} \quad (16)$$

O controlador PI multivariável possui a seguinte estrutura:

$$K(s) = \begin{bmatrix} k_{11} & \dots & k_{1j} \\ \vdots & \ddots & \vdots \\ k_{i1} & \dots & k_{ij} \end{bmatrix} \quad (17)$$

Em que $k_{ij}(s)$ é dado por:

$$k_{ij}(s) = kp_{ij} + \frac{ki_{ij}}{s} \quad (18)$$

Assim, o problema de encontrar o controlador $K(s)$ consiste num problema com 8 variáveis de otimização, em que os 4 controladores individuais que compõem $K(s)$ serão otimizados. Para cada função de custo, o algoritmo DE foi executado 40 vezes, e os valores médio e desvio padrão foram compilados.

A tabela 1 apresenta os dados comparativos para as funções de custo (1)-(4), (7) e (8). As tabelas 2 e 3 apresentam os resultados para as funções (9)-(11), sendo a tabela 2 para o modelo de referência 1 (dinâmica lenta) e a tabela 3 para o modelo de referência 2 (dinâmica rápida).

A figura 1 apresenta a saída obtida com as funções de custo (1)-(4). A figura 2 apresenta os resultados para as funções de custo (7) e (8). Optou-se por gráficos separados para não perder a clareza visual dos mesmos.

Vê-se pelos resultados da figura 1 e tabela 1 que as funções objetivo ISE e IAE são as que possuem a resposta mais equilibrada. O índice ISE é o que apresenta menor sobressinal médio, seguido pelo IAE. O índice ITSE apresenta o menor tempo de subida, porém, também apresenta alto acoplamento entre malhas e resposta inversa muito alta. O índice ITAE apresentou acoplamento elevado e sobressinal maior do que os demais índices.

Pela figura 2 vê-se que a função objetivo (7) resultou em uma resposta transitória com alto grau de oscilação e sobressinal de 70% em média. A função de custo (8) tem comportamento praticamente igual à função (7), mas com sobressinal de 45% em média. O desvio padrão para as duas funções é bastante elevado. Isso significa que diferentes execuções resultaram em controladores também muito diferentes, de forma que o grau de repetibilidade da formulação com essas funções objetivo é muito menor do que com as funções objetivo (1)-(4). O tempo de simulação para as funções (1)-(4) é cerca de três vezes maior do que para as funções (7) e (8). A diferença se deve à necessidade de simular a resposta do sistema em malha fechada antes de calcular os índices de desempenho (1)-(4). No caso das funções de custo (7) e (8), como são normas de sistema, as

(a)

(b)

0

Figura 1: (a) Saídas para a malha 1. (b) Saídas para a malha 2

(a)

(b)

Figura 2: (a) Saídas para a malha 1. (b) Saídas para a malha 2

mesmas podem ser calculadas diretamente pelas equações do sistema em malha fechada.

Em algumas execuções usando as equações (2), (4), (7) e (8) o tempo de acomodação foi maior que o tempo de simulação, o que resultou em valores não numéricos (NaN).

Tabela 1: Funções de custo sem modelo de referência

Funções	ISE		Tempo de subida(min)		Tempo de acomodação(min)		Sobressinal(%)		Tempo de execução(min)	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
ISE - (3)	7,18	0,01	5,53	0,04	66,26	0,73	30,27	0,98	72,62	7,85
IAE - (1)	7,73	0,01	6,86	0,03	34,61	0,06	39,05	0,45	75,94	15,85
ITSE - (4)	21,09	4,92	3,60	0,53	199,72	184,36	58,57	47,01	72,62	7,85
ITAE - (2)	9,32	0,49	6,67	0,19	52,28	0,22	64,02	7,61	67,16	1,41
J_S - (7)	36,01	37,24	6,26	5,31	296,79	210,59	67,23	50,64	18,69	1,92
J_T - (8)	22,22	19,13	11,67	27,86	349,72	155,31	45,29	27,38	18,66	0,45

Tabela 2: Funções de custo baseadas no modelo de referência 1

Funções	ISE		Tempo de subida(min)		Tempo de acomodação(min)		Sobressinal(%)		Tempo de execução(min)	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
J_{MF∞} - (9)	19,96	0,33	56,25	2,56	145,88	15,43	3,69	0,57	11,45	0,57
J_{MF2} - (10)	20,34	0,03	50,05	0,06	171,16	0,46	5,38	0,06	11,51	0,52
J_{MFISE} - (11)	19,92	0,21	62,48	0,78	99,20	5,57	1,67	0,29	68,38	4,73

Tabela 3: Funções de custo baseadas no modelo de referência 2

Funções	ISE		Tempo de subida(min)		Tempo de acomodação(min)		Sobressinal(%)		Tempo de execução(min)	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
J_{MF∞} - (9)	11,03	1,20	10,55	0,09	286,15	95,10	13,89	0,57	14,89	1,97
J_{MF2} - (10)	9,50	0,01	10,69	0,01	105,53	0,74	14,42	0,01	14,04	0,83
J_{MFISE} - (11)	8,85	0,01	10,74	0,02	57,95	0,36	21,55	0,07	64,89	1,98

Nesses casos, esses valores foram excluídos do cômputo do resultado médio do tempo de acomodação.

As figuras 3 e 4 apresentam os resultados para as funções objetivo das tabelas 2 e 3.

Figura 3: Resultados com modelo de referência 1 - Vermelho (11), Azul (10) e Preto (9)

Observa-se que para o modelo de referência 2 (dinâmica rápida), o ISE final é menor quando se utiliza norma de sinal. Para esse mesmo modelo, o tempo de subida é semelhante para as três funções objetivo, mas o tempo de acomodação, sobressinal e tempo de execução são muito diferentes. A estratégia com norma de sinal apresenta, além do menor ISE, o menor tempo de acomodação. Em contrapartida, é a que tem maior sobressinal e também tempo de execução maior.

Figura 4: Resultados com modelo de referência 2 - Vermelho (11), Azul (10) e Preto (9)

Para o modelo de referência 1 (dinâmica lenta), a função de custo (9) resultou em um ISE idêntico à função (11). Interessantemente, para esse caso o tempo de subida para a estratégia com norma de sinal é maior do que utilizando-se norma de sistema, e o sobressinal é menor. O tempo de execução também é muito maior na abordagem com norma de sinal.

5. CONCLUSÃO

Na estratégia de controle por modelo de referência, a utilização de normas de sinais para sintonia de controladores MIMO para sistemas com atraso resulta no melhor resultado agregado em termos do índice ISE, principalmente

para sinais de referência com dinâmica rápida. Para sinais ou modelos de referência com dinâmica lenta, a estratégia com norma de sinal e com norma de sistema resultaram em índices ISE praticamente iguais, tendo a estratégia com norma de sinal resultados melhores em termos de sobressinal e tempo de acomodação.

Assim, no contexto da sintonia de controladores baseados em modelos de referência, o benefício da utilização de funções de custo baseadas em normas de sinal é o melhor desempenho agregado medido pelo índice ISE e a maior simplicidade para definir o sinal de referência. A desvantagem, por outro lado, é o maior tempo de simulação, que é mais de quatro vezes maior.

Em relação aos índices de controle diretos, que são as funções objetivo (1)-(4), (7) e (8), observou-se que as funções (2) - ITAE e (4) - ITSE resultam em sistemas em malha fechada com alto grau de acoplamento e elevado sobressinal. As funções (7) e (8), baseadas no cálculo da função sensibilidade e sensibilidade complementar, apresentaram baixo grau de repetibilidade para sintonia do controlador PI com algoritmo DE. Isso significa na prática que para obter um controlador adequado, é necessário fazer várias execuções. As funções objetivo (1) - IAE e (3) - ISE são as que apresentam os melhores resultados e respostas mais equilibradas.

Em prosseguimento à pesquisa feita neste trabalho, os autores visam implementar uma versão modificada do algoritmo DE para diminuir o número de vezes que a função objetivo é calculada. Esse fator é a principal razão para os projetos baseados em norma de sinal serem tão demorados, uma vez que a função objetivo nesses casos envolve a simulação do sistema para obtenção da saída simulada necessária ao cômputo da norma de sinal.

REFERÊNCIAS

- Balestrino, A., Landi, A., Medaglia, M.V., and Satler, M.F. (2006). Performance indices and tuning in process control. *2006 14th Mediterranean Conference on Control and Automation*, 1–6.
- Campestrini, L., Filho, L.C.S., and Bazanella, A.S. (2009). Tuning of multivariable decentralized controllers through the ultimate-point method. *IEEE Transactions on Control Systems Technology*, 17(6), 1270–1281. doi:10.1109/TCST.2008.2006495.
- Coelho, L. and Mariani, V. (2012). Firefly algorithm approach based on chaotic tinkerbelle map applied to multivariable PID controller tuning. *Computers & Mathematics with Applications*, 64, 2371–2382. doi:10.1016/j.camwa.2012.05.007.
- Coelho, L. and Wicthoff Pessoa, M. (2011). A tuning strategy for multivariable PI and PID controllers using differential evolution combined with chaotic zaslavskii map. *Expert Syst. Appl.*, 38, 13694–13701. doi:10.1016/j.eswa.2011.04.156.
- Das, S. and Suganthan, P.N. (2016). Recent advances in differential evolution - an updated survey. *Swarm and Evolutionary Computation*, 27, 1–30.
- Devikumari, A. and Velappan, V. (2015). Decentralized PID controller design for 3x3 multivariable system using heuristic algorithms. *Indian Journal of Science and Technology*, 8. doi:10.17485/ijst/2015/v8i15/70394.
- Hovd, M. and Skogestad, S. (1994). Sequential design of decentralized controllers. *Automatica*, 30, 1601–1607.
- Iruthayarajan, M.W. and Baskar, S. (2009). Evolutionary algorithms based design of multivariable PID controller. *Expert Syst. Appl.*, 36, 9159–9167.
- Iruthayarajan, M.W. and Baskar, S. (2010). Covariance matrix adaptation evolution strategy based design of centralized PID controller. *Expert Syst. Appl.*, 37, 5775–5781.
- Kramer, A. and Morgado-Dias, F. (2018). Applications of artificial neural networks in process control applications: A review. 1–6. doi:10.1109/ICBEA.2018.8471743.
- L. Luyben, W. (1986). Simple method for tuning SISO controllers in multivariable systems. *Industrial & Engineering Chemistry Process Design and Development*, 25. doi:10.1021/i200034a010.
- Lieslehto, J. (1996). MIMO controller design using SISO controller design methods. *Proceeding of the 13th IFAC World Congress*, 29, 169–173. doi:10.1016/S1474-6670(17)57820-9.
- Meenakshi Kishnani, S.P. and Gupta, D.R. (2014). Comparison of different performance index factor for ABC-PID controller. *International Journal of Electronic and Electrical Engineering*, 7, 177–182.
- R. Goncalves da Silva, G., Bazanella, A., and Campestrini, L. (2018). On the choice of an appropriate reference model for control of multivariable plants. *IEEE Transactions on Control Systems Technology*, PP, 1–13. doi:10.1109/TCST.2018.2842178.
- Reynoso-Meza, Gilberto e Blasco, X.e.S.J. and Martinez, M. (2014). Controller tuning using evolutionary multi-objective optimisation: Current trends and applications. *Control Engineering Practice*, 28, 58–73. doi:10.1016/j.conengprac.2014.03.003.
- Shah, G. and Engell, S. (2011). Tuning MPC for desired closed-loop performance for MIMO systems. In *Proceedings of the 2011 American Control Conference*, 4404–4409. doi:10.1109/ACC.2011.5991581.
- Shen, Y., Sun, Y., and Xu, W. (2014). Centralized PI/PID controller design for multivariable processes. *Industrial & Engineering Chemistry Research*, 53(25), 10439–10447. doi:10.1021/ie501541s.
- Storn, R. and Price, K.V. (1997). Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optimization*, 11, 341–359.
- Turco Neto, E., Imtiaz, S., and Ahmed, S. (2017). A sequential optimization procedure to tune MIMO PID controllers. In *Conference: 2017 Indian Control Conference (ICC)*, 415–420. doi:10.1109/INDIANCC.2017.7846510.
- Wood, R. and Berry, M. (1973). Terminal composition control of a binary distillation column. *Chemical Engineering Science*, 28(9), 1707–1717. doi:https://doi.org/10.1016/0009-2509(73)80025-9.
- Xiong, Q., Cai, W.J., and He, M.J. (2007). Equivalent transfer function method for PI/PID controller design of MIMO processes. *Journal of Process Control*, 17, 665–673. doi:10.1016/j.jprocont.2007.01.004.
- Zeng, G., Chen, J., Chen, M.R., Dai, Y.X., Li, L.M., Lu, K.D., and Zheng, C.W. (2015). Design of multivariable PID controllers using real-coded population-based extremal optimization. *Neurocomputing*, 151, 1343–1353.