



Multi-Objective Particle Swarm Optimization for Predictive Controller Design of the Wood–Berry Distillation Column

Boumegoura Rami^{1,2*}, Bendib Riad^{1,2}, Menighed Kamel¹

¹ Department of Petrochemicals, University of 20th August 1955, Skikda – ALGERIA

² Automatics Laboratory of Skikda (LAS) University of 20th August 1955, Skikda - ALGERIA

Corresponding Author Email: r.boumegoura@univ-skikda.dz

ABSTRACT

Received: June, 02nd, 2026

Accepted: July, 19th, 2026

Published: July, 23rd, 2026

Keywords:

Model Predictive Control (MPC), Multi-Objective Optimization, Particle Swarm Optimization (PSO), MIMO Process Control, Wood–Berry Distillation Column.

Model Predictive Control (MPC) is a powerful advanced control strategy whose performance is heavily dependent on the proper tuning of its parameters, such as prediction and control horizons and weighting matrices. This tuning process often involves a trade-off between competing objectives, primarily reference tracking accuracy and control effort smoothness. This paper presents a systematic, multi-objective framework for the automatic tuning of an offset-free MPC controller. The tuning problem is formulated as a bi-objective optimization, minimizing both output tracking error and input movement. A Multi-Objective Particle Swarm Optimization (MOPSO) algorithm is employed to efficiently explore the tuning parameter space and generate a Pareto front, representing the optimal trade-off between the two objectives. A knee-point selection method is then used to extract a single, balanced controller design from the Pareto-optimal set. The proposed methodology is demonstrated and validated on the classic Wood-Berry binary distillation column benchmark. Simulation results clearly illustrate the trade-off, showing that the selected knee-point controller achieves a favorable compromise, offering a significant reduction in control effort with only a minor sacrifice in tracking performance compared to the aggressive tuning extreme.

1. INTRODUCTION

Modern chemical plants operate near quality, safety, and economic limits, requiring advanced control. Model Predictive Control (MPC) is the industry standard because it predicts future behavior, optimizes moves, and handles constraints directly [1], [2]. via receding-horizon optimization, making it ideal for constrained multivariable processes [1]–[4].

However, MPC performance is not automatic. It depends critically on prediction horizon N_p , control horizon N_c , output weights Q , and input move weights R . Aggressive settings (larger Q , shorter horizons, smaller R) give faster tracking but higher input activity; conservative settings do the opposite. Thus MPC tuning is inherently multi-objective [5].

Formulating tuning as multi-objective optimization (MOO) makes the trade-off explicit, returning a Pareto set from which an engineer can select a balanced point (e.g., a "knee"). Metaheuristics like Genetic Algorithms and Particle Swarm Optimization (PSO) efficiently explore the design space [6]–[10], [22].

However, moving from a Pareto set to one implementable controller is often overlooked [11]. A widely accepted remedy is to prefer knee points — solutions where improving one objective severely worsens the other — representing a balanced compromise [12], [13]. This paper addresses that gap by coupling automatic knee-point selection to MPC tuning: we auto-tune offset-free MPC for a MIMO process via bi-objective formulation (tracking vs. actuation) and extract a balanced "knee" design. We adopt the Wood–Berry binary distillation column, a classic 2×2 benchmark with strong interactions and delays [14], [15].

The remainder of the paper is organized as follows. Section 2 presents the Wood–Berry process model. Section 3 details the bi-objective formulation, offset-free MPC structure, MOPSO procedure, and knee-point selection. Section 4 reports simulation results. Section 5 concludes.

2. PROCESS MODEL: WOOD–BERRY DISTILLATION COLUMN

This section presents the process considered in our study and the dynamic model used for controller design. We first

summarize the Wood–Berry pilot-scale binary distillation column, then give the continuous-time transfer-function model (first order plus time delay per path) with the canonical parameters commonly adopted for this benchmark.

2.1 Process Description

We consider the pilot-scale binary distillation column of Wood and Berry, whose terminal compositions—overhead x_D and bottoms x_B —are regulated by manipulating reflux flow and steam (reboiler) flow, respectively. The plant was interfaced to an IBM 1800 digital computer; control calculations were performed at 2.5-min intervals, in line with the gas-chromatograph analysis cycle.

The column is 9 in in diameter with 8 trays, tray spacing 12 in, each tray fitted with four $2\frac{3}{8} \times 1\frac{1}{8}$ in bubble caps; it employs a total condenser and a basket reboiler. A schematic and instrumentation overview are provided in Appendix A of the original paper, which also lists typical steady-state operating conditions (Table A1).

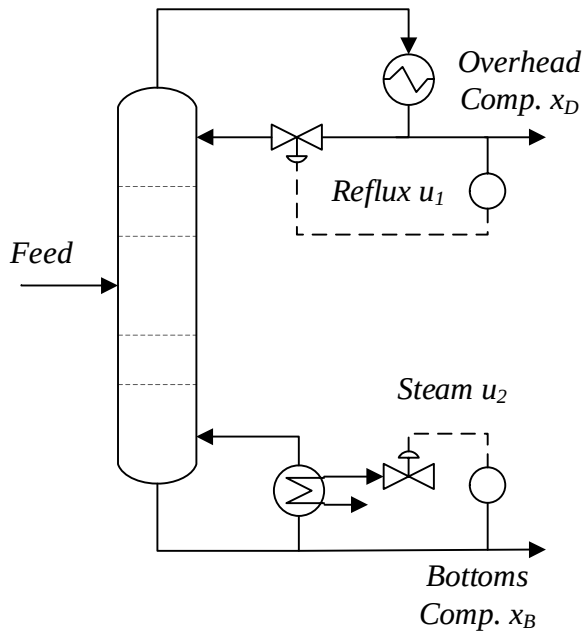


Figure 1. Wood–Berry distillation column schematic. Manipulated variables: reflux u_1 (top valve) and steam u_2 (bottom valve); controlled outputs: overhead composition x_D and bottoms composition x_B . Solid lines indicate process flows; dashed lines indicate supervisory-control links.

2.2 Dynamic Model (Continuous Time)

Wood and Berry established the column dynamics by on-line pulse testing, fitting first-order-plus-time-delay (FOPDT) transfer functions to each input-output path [14]. Time delays were estimated visually; gains and time constants were determined via least-squares using Rosenbrock's direct search. Average parameter values over multiple tests were then used to specify a simplified model for control design.

With manipulated inputs $u = [u_1 \ u_2]^T$ (reflux, steam) and outputs $y = [y_1 \ y_2]^T$ (x_D, x_B), the 2×2 transfer-

function matrix (units in minutes) is taken in the widely adopted canonical Wood–Berry parameterization:

$$G(s) = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} = \begin{bmatrix} \frac{12.8e^{-1.0s}}{16.7s + 1} & \frac{-18.9e^{-3.0s}}{21.0s + 1} \\ \frac{6.6e^{-7.0s}}{10.9s + 1} & \frac{-19.4e^{-3.0s}}{14.4s + 1} \end{bmatrix} \quad (1)$$

These path-wise FOPDT parameters (K, τ, θ) are consistent with Wood–Berry's pulse-test identification methodology and the simplified model they employed for controller synthesis, and they are the standard values used in subsequent benchmark studies and implementations.

Table 1. Canonical Wood–Berry FOPDT parameters

Path	Input → Output	K	τ	θ
$G_{11}(s)$	Reflux → Overhead x_D	12.8	16.7	1.0
$G_{12}(s)$	Steam → Overhead x_D	-18.9	21.0	3.0
$G_{21}(s)$	Reflux → Bottoms x_B	6.6	10.9	7.0
$G_{22}(s)$	Steam → Bottoms x_B	-19.4	14.4	3.0

Notes: (i) K is the steady-state gain, τ is the time constant, and θ is the time delay. (ii) Negative gains indicate counter-directional effects of manipulations. (iii) Values are given in minutes and correspond to the simplified dynamic model identified by Wood & Berry. Notation. Inputs u_1 (reflux), u_2 (steam); outputs $y_1 = x_D$ (overhead composition), $y_2 = x_B$ (bottoms composition).

Units conform to the original study's reporting: flows in lb/min and compositions in wt.% methanol (see Table A1 in Wood–Berry) [14].

Table 2. Variables and units (as used in the original study)

Symbol	Description	Unit
u_1	Reflux flow (manipulated)	lb/min
u_2	Steam flow (manipulated)	lb/min
$y_1 = x_D$	Overhead composition	wt.% MeOH
$y_2 = x_B$	Bottoms composition	wt.% MeOH
t	Time	min

(Units and nomenclature consistent with the experimental appendix and operating table in Wood–Berry.) [14]

2.3 Discrete-Time State-Space Realization (Zero-Order Hold with Input Delays)

For digital control design we adopt a zero-order-hold (ZOH) discretization of each first-order lag and realize the input delays as integer sample lags via per-input shift registers. Let T_s denote the sampling period; for definiteness we take $T_s = 1$ min so that the canonical delays map to integers (1, 3, 7, 3). A single FOPDT path

$$G_{ij}(s) = \frac{K_{ij} e^{-\theta_{ij}s}}{\tau_{ij}s + 1} \quad (2)$$

$$\begin{aligned} x_{ij}[k+1] &= a_{ij} x_{ij}[k] + b_{ij} v_{ij}[k], \\ v_{ij}[k] &= u_i[k - d_{ij}], \end{aligned} \quad (3)$$

with

$$\begin{aligned} a_{ij} &= e^{-\frac{T_s}{\tau_{ij}}}, \\ b_{ij} &= K_{ij} \left(1 - e^{-\frac{T_s}{\tau_{ij}}} \right), \\ d_{ij} &= \frac{\theta_{ij}}{T_s} \in \mathbb{Z}_{\geq 0}. \end{aligned} \quad (4)$$

With $T_s = 1$ min and the parameters of Table 1, the numerical coefficients are

$$\begin{aligned} a_{11} &= 0.9418773, & b_{11} &= 0.7439702, & d_{11} &= 1, \\ a_{12} &= 0.9534970, & b_{12} &= -0.8789076, & d_{12} &= 3, \\ a_{21} &= 0.9123395, & b_{21} &= 0.5785594, & d_{21} &= 7, \\ a_{22} &= 0.9329120, & b_{22} &= -1.3015080, & d_{22} &= 3. \end{aligned}$$

Stacking the four first-order path states and the delay-line states yields a discrete-time, linear time-invariant model

$$\begin{aligned} x_{k+1} &= A_d x_k + B_d u_k, \\ y_k &= C_d x_k + D_d u_k, \\ u_k &= \begin{bmatrix} u_1[k] \\ u_2[k] \end{bmatrix}, \quad y_k = \begin{bmatrix} y_1[k] \\ y_2[k] \end{bmatrix}. \end{aligned} \quad (5)$$

We define the state vector

$$x = [x_{11}, x_{12}, x_{21}, x_{22}, w_1^{(1)}, \dots, w_1^{(7)}, w_2^{(1)}, w_2^{(2)}, w_2^{(3)}]^\top, \quad (6)$$

where $w_\ell^{(q)}$ is the q -th element of the input- ℓ delay line, updated by

$$\begin{aligned} w_\ell^{(1)}[k+1] &= u_\ell[k], \\ w_\ell^{(q+1)}[k+1] &= w_\ell^{(q)}[k] \quad (q \geq 1). \end{aligned} \quad (7)$$

With this ordering, the output and feedthrough matrices are

$$C_d = \begin{bmatrix} 1 & 1 & 0 & 0 & \mathbf{0}_{1 \times 10} \\ 0 & 0 & 1 & 1 & \mathbf{0}_{1 \times 10} \end{bmatrix}, \quad D_d = \mathbf{0}_{2 \times 2}, \quad (8)$$

since $y_1 = x_{11} + x_{12}$ and $y_2 = x_{21} + x_{22}$. Let S_7 and S_3 denote the (7×7) and (3×3) shift matrices with ones on the subdiagonal and zeros elsewhere. The state-transition matrix has the block form

$$A_d = \begin{bmatrix} \text{diag}(a_{11}, a_{12}, a_{21}, a_{22}) & A_{(4 \times 7)} & A_{(4 \times 3)} \\ \mathbf{0}_{7 \times 4} & S_7 & \mathbf{0}_{7 \times 3} \\ \mathbf{0}_{3 \times 4} & \mathbf{0}_{3 \times 7} & S_3 \end{bmatrix}, \quad (9)$$

where the only nonzero entries of the coupling blocks select the appropriate taps from the delay lines,

$$\begin{aligned} A_{(4 \times 7)}(1,1) &= b_{11}, & A_{(4 \times 7)}(3,7) &= b_{21}; \\ A_{(4 \times 3)}(2,3) &= b_{12}, & A_{(4 \times 3)}(4,3) &= b_{22}, \end{aligned} \quad (10)$$

numerically 0.7439702, 0.5785594, -0.8789076, and -1.3015080, respectively. The input matrix injects the current inputs at the heads of the delay lines,

$$B_d = \begin{bmatrix} \mathbf{0}_{4 \times 2} \\ 1 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \text{so that} \quad \begin{aligned} w_1^{(1)}[k+1] &= u_1[k], \\ w_2^{(1)}[k+1] &= u_2[k]. \end{aligned} \quad (11)$$

and the shift submatrices S_7, S_3 propagate older samples. This ZOH-plus-delay realization is algebraically equivalent to the continuous-time FOPDT model with input delays and follows the conventions used in Wood-Berry-based implementations.

3. MULTI-OBJECTIVE TUNING OF MPC VIA PSO

We formulate MPC tuning as a bi-objective optimization problem, then solve it using Multi-Objective Particle Swarm Optimization (MOPSO) and extract a single balanced controller via knee-point selection.

3.1 Problem Formulation

Consider the discrete-time augmented plant (offset-free predictor) [3]

$$\begin{aligned} z_{k+1} &= A_a z_k + B_a u_k, \\ y_k &= C_a z_k + D_a u_k, \end{aligned} \quad (12)$$

with augmented state z_k . The MPC policy is parameterized by $\theta = (N_p, N_c, Q, R)$. For a given θ , closed-loop simulation over horizon T yields outputs $y_k(\theta)$ and inputs $u_k(\theta)$. We define two conflicting objectives:

- **Tracking error (ISE):**

$$\begin{aligned} J_1(\theta) &= \sum_{k=0}^{T-1} \|y_k(\theta) - r_k\|_{W_y}^2 \\ &= \sum_{k=0}^{T-1} (y_k(\theta) - r_k)^\top W_y (y_k(\theta) - r_k), \end{aligned} \quad (13)$$

- **Actuation effort (ISU):**

$$\begin{aligned} J_2(\theta) &= \sum_{k=0}^{T-1} \|\Delta u_k(\theta) - r_k\|_{W_{\Delta u}}^2 \\ &= \sum_{k=0}^{T-1} (u_k(\theta) - u_{k-1}(\theta))^\top W_{\Delta u} (u_k(\theta) - u_{k-1}(\theta)), \end{aligned} \quad (14)$$

The bi-objective tuning problem is

$$\begin{aligned} & \min_{\theta \in \Theta} (J_1(\theta), J_2(\theta)) \\ & \text{subject to} \quad (12)-(13) \end{aligned} \quad (15)$$

where minimization is in the Pareto sense. θ^a dominates θ^b if $J_i(\theta^a) \leq J_i(\theta^b)$ for $i = 1, 2$ and at least one strict [20]. The Pareto set $\mathcal{P}^*$ contains all non-dominated designs.

3.2 Offset-Free MPC Structure

We adopt an offset-free formulation with output disturbance augmentation [18], [19]. The nominal plant is

$$x_{k+1} = A_d x_k + B_d u_k, \quad y_k = C_d x_k + D_d u_k, \quad (16)$$

Appending a disturbance state η_k gives the augmented predictor

$$\begin{aligned} A_a &= \begin{bmatrix} A_d & 0 \\ 0 & I_p \end{bmatrix}, \quad B_a = \begin{bmatrix} B_d \\ 0 \end{bmatrix}, \\ C_a &= [C_d \quad I_p], \quad D_a = D_d \end{aligned} \quad (17)$$

and augmented state $z_k = [x_k^\top \quad \eta_k^\top]^\top$

$$Y = \mathcal{P}_a z_k + \mathcal{U}_a u_{k-1} + \mathcal{S}_a \Delta U, \quad (18)$$

with $\Delta U = [\Delta u_k^\top \cdots \Delta u_{k+N_c-1}^\top]^\top$. The MPC cost is

$$J_{mpc}(\Delta U; \theta) = (Y - R)^\top \bar{Q}(\theta)(Y - R) + \Delta U^\top \bar{R}(\theta) \Delta U \quad (19)$$

leading to the condensed QP

$$\frac{1}{2} \Delta U^\top H(\theta) \Delta U + f(\theta, z_k, u_{k-1}, R)^\top \Delta U + \text{const}, \quad (20)$$

$$H(\theta) = 2(\mathcal{S}_a^\top \bar{Q}(\theta) \mathcal{S}_a + \bar{R}(\theta)), \quad (21)$$

$$f(\theta, \cdot) = 2 \mathcal{S}_a^\top \bar{Q}(\theta) (\mathcal{P}_a z_k + \mathcal{U}_a u_{k-1} - R).$$

The first move $u_k = u_{k-1} + [I_m \ 0 \cdots 0] \Delta U^*$ is applied.

Using input increments $\Delta u_k = u_k - u_{k-1}$, the lifted prediction over horizon N_p is

3.3 Multi-Objective Particle Swarm Optimization

We use MOPSO with P particles [17]. Each particle p has position $\theta_p^{(t)}$ (design vector) and velocity $v_p^{(t)}$. The algorithm maintains an external archive $\mathcal{A}^{(t)}$ of non-dominated designs [21].

Velocity update (with constriction factor [22]):

$$v_p^{(t+1)} = \chi [\omega v_p^{(t)} + c_1 r_{1,p}^{(t)} \circ (\hat{\theta}_p^{(t)} - \theta_p^{(t)}) + c_2 r_{2,p}^{(t)} \circ (\tilde{\theta}_p^{(t)} - \theta_p^{(t)})], \quad (22)$$

Position update with projection onto Θ :

$$\theta_p^{(t+1)} = \Pi_\Theta(\theta_p^{(t)} + v_p^{(t+1)}), \quad (23)$$

Here $\hat{\theta}_p^{(t)}$ is the personal best, $\tilde{\theta}_p^{(t)}$ is a leader selected from $\mathcal{A}^{(t)}$ (e.g., by crowding distance), and $r_{1,p}, r_{2,p}$ are random vectors. Personal bests are updated using Pareto dominance: if the new position dominates the old, replace it; if neither dominates, keep the one with larger crowding distance [17].

The archive is updated by adding all current positions and filtering for non-dominance. If size exceeds capacity, pruning removes points with smallest crowding distance to preserve front spread [21].

3.4 Knee-Point Selection

After MOPSO terminates, we extract a single balanced controller from the final archive $\mathcal{A}$ [12], [13]. Let $J_m^{\min}$ and $J_m^{\max}$ be the extremes of objective m over $\mathcal{A}$. Normalize:

$$\tilde{J}_m(\theta) = \frac{J_m(\theta) - J_m^{\min}}{J_m^{\max} - J_m^{\min} + \varepsilon}, \quad m \in \{1, 2\}, \quad (24)$$

Let $\theta^{(1)}$ and $\theta^{(2)}$ be the designs minimizing J_1 and J_2 , respectively. Define normalized extreme points $z_1 = \tilde{\mathbf{J}}(\theta^{(1)})$, $z_2 = \tilde{\mathbf{J}}(\theta^{(2)})$, chord direction $\hat{c} = (z_2 - z_1) / \|z_2 - z_1\|_2$. The perpendicular distance from any $\theta \in \mathcal{A}$ to the chord is

$$d(\theta) = \|(I_2 - \hat{c}\hat{c}^\top)(\tilde{\mathbf{J}}(\theta) - z_1)\|_2, \quad (25)$$

The knee point is the design with maximum $d(\theta)$:

$$\theta^{\text{knee}} \in \underset{\theta \in \mathcal{A}}{\operatorname{argmax}} d(\theta). \quad (26)$$

If multiple points tie, choose the one with smallest $\tilde{J}_1 + \tilde{J}_2$ (most balanced). This θ^{knee} defines the final controller used in Section 4.

4. SIMULATION AND RESULTS

We validate the proposed tuning framework on the Wood-Berry distillation column benchmark. The plant is the standard 2×2 MIMO system with manipulated flows (u_1, u_2) and controlled compositions (y_1, y_2) , discretized at $T_s = 0.5$ min. The evaluation scenario uses a simultaneous reference step of +0.30 on y_1 and +0.20 on y_2 at $t = 5$ min, simulated for 50 min (100 samples). The offset-free MPC uses the augmented predictor of Section 3.2, with no hard constraints in the QP.

MOPSO parameters: swarm size $P = 40$, 40 iterations, inertia $\omega \in [0.9, 0.4]$, $c_1 = c_2 = 1.9$, archive capacity 300. The decision vector $\theta = \{N_p, N_c, Q_{y_1}, Q_{y_2}, R_{\Delta u_1}, R_{\Delta u_2}\}$ with bounds: $N_p \in [5, 50]$, $N_c \in [5, 30]$, $Q \in [0.1, 50]$, $R \in [10^{-2}, 1]$.

The final non-dominated archive contains 299 designs. The Pareto front (Fig. 2) is smooth and convex,

with J_1 (ISE) ranging $[0.5500, 0.8359]$ and J_2 (ISU) ranging $[0.01018, 2.39135]$. Three representative controllers are selected: min- J_1 (fast tracking), knee (balanced), and min- J_2 (ultra-smooth). Their parameters and costs are given in Table 3.

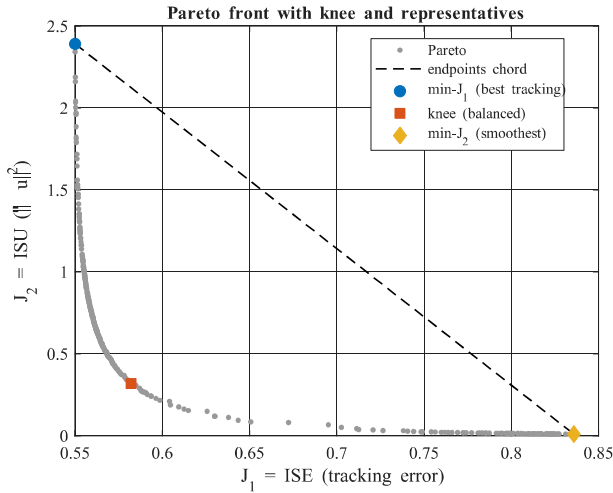


Figure 2. Pareto front (J_1, J_2) with markers for min- J_1 , knee, and min- J_2 .

The knee reduces actuation effort by 86.7% ($J_2: 2.3914 \rightarrow 0.3187$) with only a 5.8% increase in tracking error ($J_1: 0.5500 \rightarrow 0.5821$) compared to the fast extreme. Moving to min- J_2 further reduces J_2 by 96.8% but at a 43.6% penalty in J_1 , confirming the knee as the best compromise.

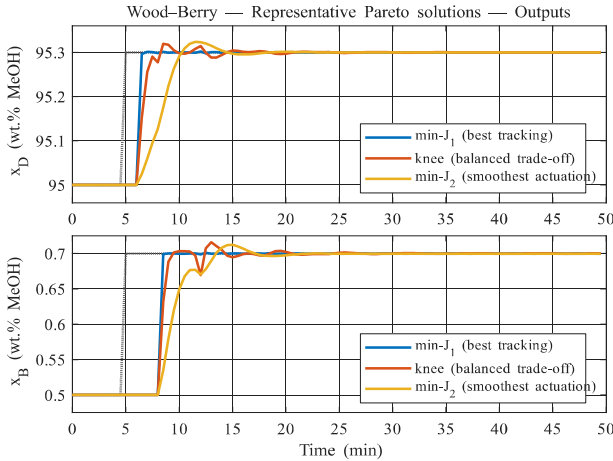


Figure 3. Closed-loop outputs y_1, y_2 (solid) vs. references (dashed) for the three controllers.

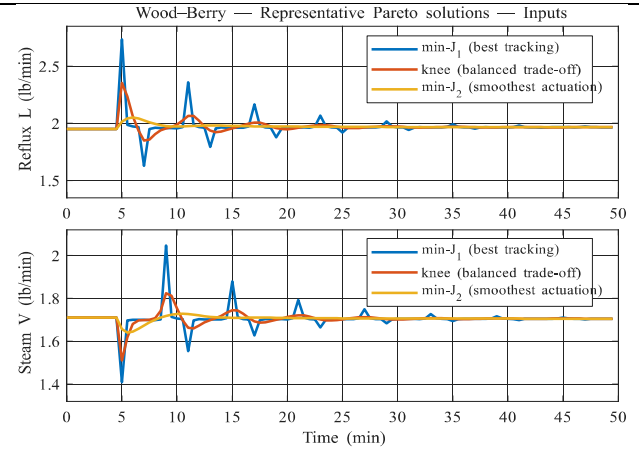


Figure 4. Manipulated inputs u_1, u_2 (top) and increments $\Delta u_1, \Delta u_2$ (bottom) for the three controllers.

The fast controller (min- J_1) achieves the fastest tracking, with settling times of only 1.5 – 3.5 min and negligible overshoot, but at the cost of large input moves (TVU = 7.15). In contrast, the smooth controller (min- J_2) drastically reduces actuation (TVU = 0.38), yet its responses are markedly slower, with rise times of 5–8 min and higher integrated error (IAE = 4.24). The knee controller strikes a practical balance: it cuts control effort by 86.7% relative to the fast extreme (TVU = 2.44) while keeping settling times within 9 – 10.5 min and IAE at 2.88 — a modest 23% increase over the fast design. These trends are clearly visible in the output and input trajectories of Figs. 3–4.

The MOPSO optimization converged steadily, with best-so-far objectives improving rapidly in early iterations and then leveling off. The final Pareto set is well distributed across the achievable trade-off, as shown in Fig. 2. Overall, the proposed framework successfully generates a clear Pareto front, and the automatic knee-point selection returns a single balanced controller without requiring any user-defined weights — making the tuning process transparent and reproducible.

5. CONCLUSION

This paper presented a systematic multi-objective framework for automatic tuning of offset-free MPC. The tuning problem was formulated as a bi-objective optimization minimizing tracking error and actuation effort. MOPSO efficiently explored the design space and generated a Pareto front, and an automatic knee-point selection method extracted a single balanced controller without ad-hoc weights.

Table 3. Pareto representatives and costs

Representative	Candidate index	N_p	N_c	Q_1	Q_2	R_1	R_2	J_1 (ISE)	J_2 (ISU)
min- J_1 (fast)	1	24	20	21.927	44.157	0.010	0.010	0.5500	2.3914
knee (balanced)	199	34	23	6.154	6.183	0.703	0.644	0.5821	0.3187
min- J_2 (smooth)	300	40	8	0.100	0.100	1.000	1.000	0.8359	0.01018

Reading guide. Lower J_1 indicates better tracking; lower J_2 indicates smoother (less variable) actuation. The knee is identified by the maximum perpendicular distance to the chord joining the two extremes in normalized (J_1, J_2) space.

Validation on the Wood–Berry distillation column benchmark showed that the knee controller reduces

control effort by 86.7% compared to an aggressive tuning, with only a 5.8% loss in tracking performance. Future work includes adding hard constraints, testing robustness under disturbances, and extending the approach to larger-scale industrial benchmarks.

REFERENCES

- [1] M. L. Darby and M. Nikolaou, "MPC: Current practice and challenges," *Control Engineering Practice*, vol. 20, no. 4, pp. 328–342, Apr. 2012, doi: 10.1016/j.conengprac.2011.12.004.
- [2] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, Jul. 2003, doi: 10.1016/S0967-0661(02)00186-7.
- [3] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, 2nd edition. Santa Barbara: Nob Hill Publishing, LLC, 2024.
- [4] E. F. Camacho and C. Bordons, *Model Predictive control*. in Advanced Textbooks in Control and Signal Processing. London: Springer London, 2007. doi: 10.1007/978-0-85729-398-5.
- [5] M. Alhajeri and M. Soroush, "Tuning Guidelines for Model-Predictive Control," *Ind. Eng. Chem. Res.*, vol. 59, no. 10, pp. 4177–4191, Mar. 2020, doi: 10.1021/acs.iecr.9b05931.
- [6] M. Vallerio, J. Van Impe, and F. Logist, "Tuning of NMPC controllers via multi-objective optimisation," *Computers & Chemical Engineering*, vol. 61, pp. 38–50, Feb. 2014, doi: 10.1016/j.compchemeng.2013.10.003.
- [7] V. Bachtar, C. Manzie, W. H. Moase, and E. C. Kerrigan, "Analytical results for the multi-objective design of model-predictive control," *Control Engineering Practice*, vol. 56, pp. 1–12, Nov. 2016, doi: 10.1016/j.conengprac.2016.07.009.
- [8] J. H. Van Der Lee, W. Y. Svrcek, and B. R. Young, "A tuning algorithm for model predictive controllers based on genetic algorithms and fuzzy decision making," *ISA Transactions*, vol. 47, no. 1, pp. 53–59, Jan. 2008, doi: 10.1016/j.isatra.2007.06.003.
- [9] V. Ramasamy, R. K. Sidharthan, R. Kannan, and G. Muralidharan, "Optimal Tuning of Model Predictive Controller Weights Using Genetic Algorithm with Interactive Decision Tree for Industrial Cement Kiln Process," *Processes*, vol. 7, no. 12, p. 938, Dec. 2019, doi: 10.3390/pr7120938.
- [10] G. A. Nery Júnior, M. A. F. Martins, and R. Kalid, "A PSO-based optimal tuning strategy for constrained multivariable predictive controllers with model uncertainty," *ISA Transactions*, vol. 53, no. 2, pp. 560–567, Mar. 2014, doi: 10.1016/j.isatra.2013.12.019.
- [11] S. Petchrompo, D. W. Coit, A. Brintrup, A. Wannakrairo, and A. K. Parlikad, "A review of Pareto pruning methods for multi-objective optimization," *Computers & Industrial Engineering*, vol. 167, p. 108022, May 2022, doi: 10.1016/j.cie.2022.108022.
- [12] J. Branke, K. Deb, H. Dierolf, and M. Osswald, "Finding Knees in Multi-objective Optimization," in *Parallel Problem Solving from Nature - PPSN VIII*, vol. 3242, X.
- [13] Yao, E. K. Burke, J. A. Lozano, J. Smith, J. J. Merelo-Guervós, J. A. Bullinaria, J. E. Rowe, P. Tiño, A. Kabán, and H.-P. Schwefel, Eds., in *Lecture Notes in Computer Science*, vol. 3242. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 722–731. doi: 10.1007/978-3-540-30217-9_73.
- [14] S. Bechikh, L. Ben Said, and K. Ghédira, "Searching for knee regions in multi-objective optimization using mobile reference points," in *Proceedings of the 2010 ACM Symposium on Applied Computing*, Sierre Switzerland: ACM, Mar. 2010, pp. 1118–1125. doi: 10.1145/1774088.1774325.
- [15] R. K. Wood and M. W. Berry, "Terminal composition control of a binary distillation column," *Chemical Engineering Science*, vol. 28, no. 9, pp. 1707–1717, Sep. 1973, doi: 10.1016/0009-2509(73)80025-9.
- [16] M. Hamdy, A. Ramadan, and B. Abozalam, "Comparative study of different decoupling schemes for TITO binary distillation column via PI controller," *IEEE/CAA J. Autom. Sinica*, vol. 5, no. 4, pp. 869–877, Jul. 2018, doi: 10.1109/JAS.2016.7510040.
- [17] M. Reyes-Sierra and C. A. Coello Coello, "Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art," *International Journal of Computational Intelligence Research*, vol. 2, no. 3, pp. 287–308, 2006.
- [18] G. Pannocchia, "Offset-free tracking MPC: A tutorial review and comparison of different formulations," in *2015 European Control Conference (ECC)*, Linz, Austria: IEEE, Jul. 2015, pp. 527–532. doi: 10.1109/ECC.2015.7330597.
- [19] L. Wang, *Model Predictive Control System Design and Implementation Using MATLAB®*. in Advances in Industrial Control. London: Springer London, 2009. doi: 10.1007/978-1-84882-331-0.
- [20] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, Jun. 2000, doi: 10.1016/S0005-1098(99)00214-9.
- [21] K. Deb, "Multi-Objective Optimization," in *Search Methodologies*, E. K. Burke and G. Kendall, Eds., Boston, MA: Springer US, 2005, pp. 273–316. doi: 10.1007/0-387-28356-0_10.
- [22] A. Amrane and Y. Zennir, "Performance Analysis of Nature-Inspired Algorithms for PID Control of Electric Wheelchairs," *International Journal of Automation and Safety*, vol. 2, no. 2, pp. 14–19, Dec. 2