

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

**МЕТОДОЛОГІЯ УДОСКОНАЛЕНОГО
КЕРУВАННЯ ТЕХНОЛОГІЧНИМИ ПРОЦЕСАМИ
Комп'ютерний практикум**

*Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для здобувачів ступеня доктора філософії за освітньо-науковою
програмою «Автоматизація та комп'ютерно інтегровані технології» спеціальності 151
«Автоматизація та комп'ютерно-інтегровані технології»*

Київ
КПІ ім. Ігоря Сікорського
2021

Методологія удосконаленого керування технологічними процесами. Комп'ютерний практикум [Електронний ресурс]: навч. посіб. для аспірантів спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» / Укладачі: О. В. Степанець, Ю. І. Маріяш; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 865 кБайт). – Київ : КПІ ім. Ігоря Сікорського, 2021. – 26 с.

*Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол № 2 від 09.12.2021 р.)
за поданням Вченої ради Теплоенергетичного факультету (протокол № 3 від 26.10.2021 р.)*

Електронне мережне навчальне видання

МЕТОДОЛОГІЯ УДОСКОНАЛЕНОГО КЕРУВАННЯ ТЕХНОЛОГІЧНИМИ ПРОЦЕСАМИ

Комп'ютерний практикум

Укладачі: *Степанець Олександр Васильович, канд. техн. наук, доцент
Маріяш Юрій Ігорович, асистент*

Рецензент: *Ковальчук Артем Михайлович, канд. техн. наук, доцент*

Відповідальний редактор: *Баган Тарас Григорович, канд. техн. наук, доцент*

У навчальному посібнику наведено базовий теоретичний матеріал та практичні приклади удосконаленого керування технологічними процесами. Посібник містить завдання для практичного виконання, здійснення яких дозволить студентам оволодіти методами та техніками прогнозуючого керування, а також питання для самоперевірки.

Посібник призначений для аспірантів, які навчаються за спеціальністю 151 "Автоматизація та комп'ютерно інтегровані технології".

© О. В. Степанець, Ю. І. Маріяш
© КПІ імені Ігоря Сікорського, 2021

ЗМІСТ

ОСНОВНІ ТЕОРЕТИЧНІ ПОЛОЖЕННЯ	4
Задача MPC-УПРАВЛІННЯ	4
Базова задача MPC-управління з лінійною моделлю і квадратичним функціоналом	5
Розробка MPC-регулятора за допомогою Matlab MPC Designer	12
Розробка MPC-регулятора за допомогою robust optimal control toolbox	19
ПРАКТИЧНІ ЗАВДАННЯ	24
Завдання 1	24
Завдання 2	24
Завдання 3	24
КОНТРОЛЬНІ ЗАПИТАННЯ	25
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	26

ОСНОВНІ ТЕОРЕТИЧНІ ПОЛОЖЕННЯ

Задача MPC-УПРАВЛІННЯ

Одним із сучасних підходів до аналізу і синтезу систем керування, що базуються на математичних методах оптимізації, є теорія управління динамічними об'єктами з використанням прогнозуючих моделей - Model Predictive Control (MPC). Розвиток ідей управління з прогнозуючими моделями відбувається в напрямку використання нелінійних моделей, забезпечення стійкості керуючого впливу по Ляпунову, надання робастних властивостей замкнутій системі управління і застосування сучасних методів оптимізації.

Ідея оптимізації прогнозованої керуючої дії, яка складає основу MPC-методів, виникла в рамках двох незалежних, проте близьких по суті підходів. Перший, Dynamics Matrix Control (DMC), розвивався зусиллями фахівців компанії Shell Oil в середині 60-х років, а другий - Model Algorithmic Control (MAC) - був розроблений французькими інженерами хімічної промисловості в кінці 60-х. На базі останнього підходу вперше був створений комерційний пакет програм IDCOM (Identification and Command), який певною мірою послужив прообразом сучасних методів управління з прогнозом.

Ідеологія MPC-підходу полягає у наступній схемі управління (рис. 1) динамічними об'єктами за принципом зворотного зв'язку:

1. Розглядається математична модель об'єкта (відносно проста), початковими умовами для якої служить її поточний стан. При заданому управлінні виконується прогноз руху об'єкта на деякому кінцевому відрізку часу (горизонті прогнозу);
2. Виконується оптимізація управління, метою якого служить наближення регулюючих змінних прогнозуючої моделі до відповідного заданого значення (уставки) на горизонті прогнозу;
3. Реалізується знайдене оптимальне управління і здійснюється вимір (або відновлення по виміряним змінним) фактичного стану об'єкта на кінець кроку;
4. Горизонт прогнозу зсувається на крок вперед, і повторюються пункти 1-3 даного алгоритму.

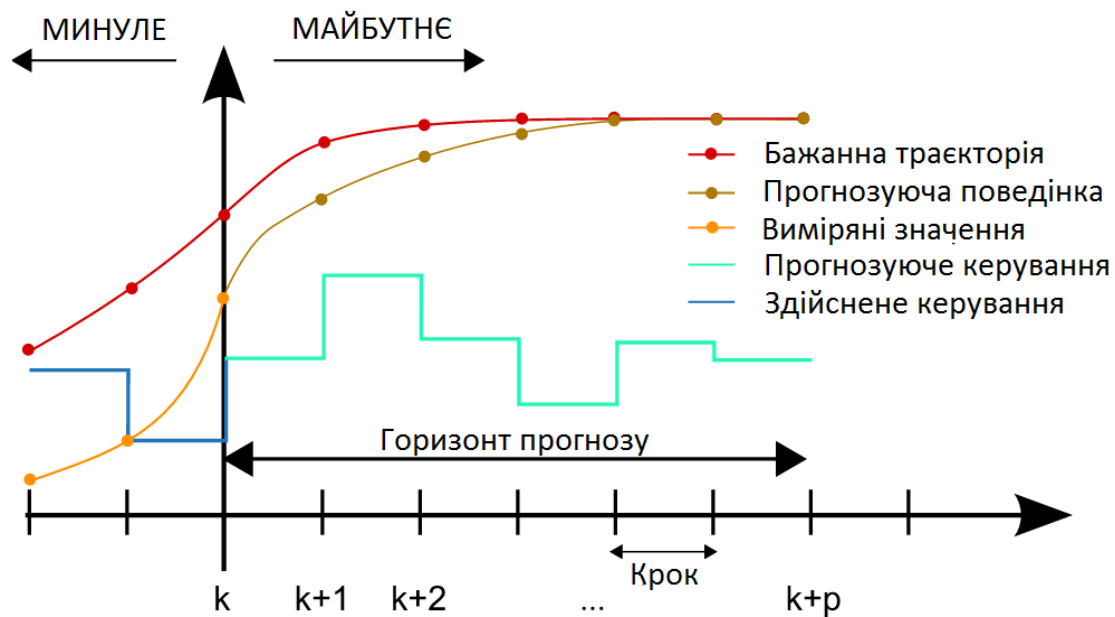


Рис. 1 Схеми управління динамічними об'єктами з використанням MPC-підходу

Основною перевагою MPC-підходу, що визначає його успішне використання в практиці побудови та експлуатації систем управління, служить відносна простота базової схеми формування зворотного зв'язку, що поєднується з високими адаптивними властивостями. Остання обставина дозволяє управляти багатовимірними і багатозв'язними об'єктами зі складною структурою, що включає нелінійність, оптимізувати процеси в режимі реального часу в рамках обмежень на керуючі і керовані змінні, враховувати невизначеності в завданні об'єктів і збурень. Крім того, можливе врахування транспортного запізнення, змін критеріїв якості регулювання і відмов датчиків.

Базова задача MPC-управління з лінійною моделлю і квадратичним функціоналом

Нехай поведінка об'єкта керування описується залежністю:

$$\dot{x}(t) = f(t, u(t)), \quad x(0) = x_0, \quad (1)$$

де $x \in E^n$ – вектор стану, $u \in E^m$ – вектор керуючої дії, $t \in [0, \infty)$.

Введемо поняття області допустимих значень стану $X \subseteq E^n$ та керуючої дії $U \subseteq E^m$, такі що для $\forall t \in [0, \infty) \rightarrow x(t) \in X, u(t) \in U$. Наприклад, область допустимих значень:

$$\begin{aligned} U &= \{u \in E^m : u_{\min} \leq u_i \leq u_{\max}, i = \overline{1, m}\} \\ X &= \{x \in E^n : x_{\min} \leq x_j \leq x_{\max}, j = \overline{1, n}\} \end{aligned} \quad (2)$$

де $u_{\min}, u_{\max}, x_{\min}, x_{\max}$ – задані дійсні числа

Будемо вважати, що метою управління об'єкта (1) забезпечення виконання рівності:

$$\lim_{t \rightarrow \infty} \|x(t) - r_x(t)\| = 0; \lim_{t \rightarrow \infty} \|u(t) - r_u(t)\| = 0, \quad (3)$$

де $r_x(t)$ та $r_u(t)$ задана векторна функція, що визначає бажаний рух об'єкта (1).

Якість управління охарактеризуємо за допомогою функціоналу виду:

$$J_0 = J_0(x(t), u(t)). \quad (4)$$

Будь-яка задача оптимального управління полягає у пошуку такої керуючої дій з області її допустимих значень U , яка забезпечить досягнення поставленої мети (3) з врахуванням обмежень (2) і досягає мінімуму цільової функції (4).

Розглянемо положення і принципи, що прийняті в задачі МРС-управління з лінійною моделлю. Введемо поняття лінійної прогнозуючої моделі об'єкта управління у виді системи рівнянь:

$$\begin{aligned} x_{i+1} &= Ax_i + Bu_i, \\ y_i &= Cx_i, \\ i &= k + j, j = 0, 1, 2, \dots, \end{aligned} \quad (5)$$

де k - номер такту; $x_i \in E^n$ – стан об'єкта; $y_i \in E^r$ – виміри; $u_i \in E^m$ – керуюча дія;

Будемо вважати, що за динамічними властивостями прогнозуюча модель (5) в якомусь сенсі близька до реального об'єкта (1). Нехай прогнозуюча модель (5) у поточний момент часу ($j=0$) ініціалізується станом $\tilde{x}_k$ реального об'єкта керування та виконується рівність $u_i = \tilde{u}_i$ для $\forall i = k, k+1, \dots, k+P$. Тоді послідовність векторів $\{x_i\} (i = k+1, \dots, k+P)$, яка отримується з моделі (5), представляє собою

прогноз руху $\{\tilde{x}_k\}$ реального об'єкта з горизонтом прогнозу P (рис. 1). Якість управління охарактеризуємо за допомогою лінійно-квадратичного функціоналу:

$$J_k = J_k(\bar{y}, \bar{u}) = \sum_{j=1}^P \left[(y_{k+j} - r_{k+j})^T R_{k+j} (y_{k+j} - r_{k+j}) + u_{k+j-1}^T Q_{k+j} u_{k+j-1} \right], \quad (6)$$

де R_{k+j} та Q_{k+j} – додатно визначені симетричні матриці.

Допоміжні вектори:

$$\begin{aligned} \bar{y} &= (y_{k+1} \ y_{k+2} \ \dots \ y_{k+P})^T \in E^{rP}, \\ \bar{u} &= (u_k \ u_{k+1} \ \dots \ u_{k+P-1})^T \in E^{mP}. \end{aligned} \quad (7)$$

Враховуючи, що рух об'єкта (1.5) на тактах $k+j$, $j=1, 2, \dots, P$ визначається тільки $\bar{u}$, то $J_k = J_k(\bar{y}(\bar{u}), \bar{u}) = J_k(\bar{u})$. При цьому може бути сформульована задача оптимізації по відношенні до функціоналу (6):

$$J_k(\bar{u}) \rightarrow \min_{\bar{u} \in E^{mP}}. \quad (8)$$

Знайдемо рішення задачі (8). Враховуючи (7) представимо функціонал (6) в виді:

$$\begin{aligned} J_k &= J_k(\bar{u}) = (\bar{y} - \bar{r})^T R (\bar{y} - \bar{r}) + \bar{u}^T Q \bar{u} \\ y_{k+1} &= Cx_{k+1} = CAx_k + CBu_k \\ y_{k+2} &= Cx_{k+2} = CAx_{k+1} + CBu_{k+1} = CA^2x_k + CABu_k + CBu_{k+1} \\ \text{і т.д. } y_{k+P} &= Cx_{k+P} = CA^Px_k + CA^{P-1}Bu_k + CBu_{k+1} \\ \Rightarrow \bar{y} &= Lx_k + M\bar{u}, \text{ де} \end{aligned}$$

$$L = \begin{pmatrix} CA \\ CA^2 \\ \dots \\ CA^P \end{pmatrix} \quad M = \begin{pmatrix} CB & 0 & \dots & 0 \\ CAB & CB & \dots & 0 \\ \dots & \dots & \dots & \dots \\ CA^{P-1}B & CA^{P-2}B & \dots & CB \end{pmatrix} \Rightarrow$$

$$\begin{aligned} J_k &= J_k(\bar{u}) = (Lx_k + M\bar{u} - \bar{r})^T R (Lx_k + M\bar{u} - \bar{r}) + \bar{u}^T Q \bar{u} \\ \frac{\partial J_k}{\partial \bar{u}} &= \frac{\partial}{\partial \bar{u}} \left[(Lx_k + M\bar{u} - \bar{r})^T R (Lx_k + M\bar{u} - \bar{r}) + \bar{u}^T Q \bar{u} \right] = 0 \end{aligned}$$

Враховуючи, що $\frac{\partial}{\partial x}(x^T Sx) = 2Sx$, де S – симетрична матриця маємо:

$$\frac{\partial J_k}{\partial u} = M^T R (Lx_k + M\bar{u} - \bar{r}) + Q\bar{u} = 0$$

$$\Rightarrow \bar{u}^* = \bar{K}x_k + \bar{T}\bar{r}$$

$$\bar{K} = -(M^T RM + Q)^{-1} M^T RL, \bar{T} = (M^T RM + Q)^{-1} M^T R$$

Згідно стратегії MPC-підходу застосовується керуюча дія – перший елемент отриманого вектору $\bar{u}^*$:

$$u^* = Kx_k + Tr. \quad (9)$$

де K – перші m строк матриці $\bar{K}$, T – верхній блок розміру $m \times r$ матриці $\bar{T}$.

З (9) випливає, що для лінійно-квадратичної задачі управління MPC-стратегія формування керуючого впливу зводиться до звичайного пропорційного зворотного зв'язку (регулятора) по стану об'єкта.

Алгоритм пошуку матриць K і T :

1. Маючи матриці A , B , C , D і горизонт прогнозу P сформувати матриці L і M ;
2. Використовуючи вхідні дані у вигляді матриць R та Q обрахувати додаткові матриці $\bar{K}$ і $\bar{T}$ по формулам $\bar{K} = -(M^T RM + Q)^{-1} M^T RL, \bar{T} = (M^T RM + Q)^{-1} M^T R$.
3. Виділити верхні блоки розміром $m \times n$ та $m \times r$ відповідно з матриць $\bar{K}$ і $\bar{T}$.

Звернемо увагу на те, коли всі вихідні матриці не змінюються в часі, а горизонт прогнозу залишається постійним, то шукані матриці K і T в стратегії (9) так само не змінюються в часі, тобто залишаються незмінними на кожній ітерації. Це дозволяє істотно спростити реалізацію MPC-стратегії управління за рахунок одноразового розрахунку (наприклад, після виконання чергової ідентифікації моделі) зазначених матриць. Традиційно аналітичний пошук матриці K в законі (9) для стаціонарної задачі найбільш часто здійснюється на базі двох підходів: теорії модального управління і LQ-оптимізації. По відношенню до цих підходів розглянута схема реалізації управління являє собою

альтернативний шлях побудови стаціонарної зворотного зв'язку, який за своєю суттю досить близький до LQ-оптимізації. Більш того, можна отримати умови, при яких MPC-стратегія і LQ-стратегія збігаються.

Задача MPC-управління з квадратичним функціоналом при наявності обмежень

Як було зазначено вище, базова задача MPC-управління зводиться до звичайного пропорційного зворотного зв'язку (регулятору) по стану об'єкта, який принципово не відрізняється від LQ-оптимальних регуляторів. Але ситуація принципово змінюється, якщо враховувати обмеження на керуючий вплив та стан об'єкта, що суттєво обмежує множину регуляторів у задачі LQ-оптимізації. Отримання точного оптимального рішення в реальному часі досить проблематичне, що суттєво підвищує обґрунтованість застосування MPC-стратегії.

Так як і для базової задачі, нехай поведінка об'єкта описується залежністю (1) і використовується рівняння прогнозуючої моделі (5):

$$x_{i+1} = Ax_i + Bu_i,$$

$$y_i = Cx_i,$$

$$i = k + j, j = 0, 1, 2, \dots,$$

де k - номер такту; $x_i \in E^n$ - стан об'єкта; $y_i \in E^r$ - виміри; $u_i \in E^m$ - керуюча дія;

Будемо вважати, що за динамічними властивостями прогнозуюча модель (5) в якомусь сенсі близька до реального об'єкта (1). Нехай прогнозуюча модель (5) у поточний момент часу ($j=0$) ініціалізується станом $\tilde{x}_k$ реального об'єкта керування та виконується рівність $u_i = \tilde{u}_i$ для $\forall i = k, k+1, \dots, k+P$.

Якість управління охарактеризуємо за допомогою лінійно-квадратичного функціоналу:

$$J_k = J_k(\bar{y}, \Delta \bar{u}) = \sum_{j=1}^P \left[(y_{k+j} - r_{k+j})^T R_{k+j} (y_{k+j} - r_{k+j}) + \Delta u_{k+j-1}^T Q_{k+j} \Delta u_{k+j-1} \right], \quad (10)$$

де R_{k+j} та Q_{k+j} - додатно визначені симетричні матриці.

Допоміжні вектори:

$$\begin{aligned}\bar{y} &= (y_{k+1} \ y_{k+2} \ \dots \ y_{k+P})^T \in E^{rP}, \\ \bar{u} &= (\Delta u_k \ \Delta u_{k+1} \ \dots \ \Delta u_{k+P-1})^T \in E^{mP}.\end{aligned}\quad (11)$$

Використання функціоналу (1.10) дозволяє забезпечити астатизм замкнутої системи. Розглянемо задачу оптимізації

$$J_k = J_k(\bar{y}(\Delta \bar{u}), \Delta \bar{u}) = J_k(\Delta \bar{u}) \rightarrow \min_{\Delta \bar{u} \in E^{mP}} \quad (12)$$

про пошук послідовності векторів $\Delta \bar{u}$ яка мінімізує функціонал (10) без врахування обмежень. У відповідності до підходу прийнятого для забезпечення астатизму, сформуємо додаткову прогнозуючу модель

$$\begin{aligned}p_{i+1} &= \bar{A}p_i + \bar{B}v_i, \\ z_i &= \bar{C}p_i, \\ i &= k+j, \ j=0,1,2,\dots,\end{aligned}\quad (13)$$

для якої $p_k = \begin{pmatrix} x_k - x_{k-1} \\ Cx_k \end{pmatrix}$, а матриці $\bar{A}$, $\bar{B}$, $\bar{C}$ формулами:

$$\bar{A} = \begin{pmatrix} A & 0_{n \times r} \\ CA & E_{r \times r} \end{pmatrix}, \bar{B} = \begin{pmatrix} B \\ CB \end{pmatrix}, \bar{C} = (0_{r \times n} \quad E_{r \times r}).$$

Вхідною змінною для моделі (13) служить вектор $v_i = \Delta \bar{u}$, а вихідною $z_i = y_i$.

Враховуючи (13) функціонал (12) можна представити в еквівалентному вигляді:

$$\begin{aligned}J_k &= J_k(\bar{z}(\bar{v}), \bar{v}) = \sum_{j=1}^P \left[(z_{k+j} - r_{k+j})^T R_{k+j} (z_{k+j} - r_{k+j}) + v_{k+j-1}^T Q_{k+j} v_{k+j-1} \right] = \\ &= (\bar{z} - \bar{r})^T R(\bar{z} - \bar{r}) + \bar{v}^T Q \bar{v},\end{aligned}\quad (14)$$

де

$$\begin{aligned}\bar{z} &= (z_{k+1} \ z_{k+2} \ \dots \ z_{k+P})^T \in E^{rP}, \\ \bar{v} &= (v_k \ v_{k+1} \ \dots \ v_{k+P-1})^T \in E^{mP}\end{aligned}\quad (15)$$

Враховуючи залежність $\bar{z} = Lp_k + M\bar{v}$, матриці L , M визначаються формулами:

$$\begin{aligned}
L &= \begin{pmatrix} \bar{C}\bar{A} \\ \bar{C}\bar{A}^2 \\ \dots \\ \bar{C}\bar{A}^P \end{pmatrix} = \begin{pmatrix} CA & E_{r \times r} \\ CA^2 & E_{r \times r} \\ \dots & E_{r \times r} \\ CA^P & E_{r \times r} \end{pmatrix} \\
M &= \begin{pmatrix} \bar{C}\bar{B} & 0 & \dots & 0 \\ \bar{C}\bar{A}\bar{B} & \bar{C}\bar{B} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ \bar{C}\bar{A}^{P-1}\bar{B} & \bar{C}\bar{A}^{P-2}\bar{B} & \dots & \bar{C}\bar{B} \end{pmatrix} = \begin{pmatrix} CB & 0 & \dots & 0 \\ CAB + CB & CB & \dots & 0 \\ \dots & \dots & \dots & \dots \\ \sum_{i=0}^{P-1} CA^i B & \sum_{i=0}^{P-2} CA^i B & \dots & CB \end{pmatrix}
\end{aligned} \tag{16}$$

Функціонал (14) можна представити у виді:

$$J_k = J_k(\bar{v}) = \bar{v}^T H \bar{v} + 2 f^T \bar{v} + g, \tag{17}$$

де $H = M^T R M + Q$, $f = M^T R L p_k - M^T R \bar{r}$, $g = p_k^T L^T R L p_k + r^T R \bar{r} - 2 p_k^T L^T R \bar{r}$.

Тепер накладемо умову виконання обмежень

$$\begin{cases} u_i^- \leq u_i \leq u_i^+, \\ \Delta u_i^- \leq \Delta u_i \leq \Delta u_i^+, \\ y_i^- \leq y_i \leq y_i^+, \end{cases} \tag{18}$$

де u_i^- , u_i^+ , Δu_i^- , Δu_i^+ , y_i^- , y_i^+ – задані вектори. Враховуючи (18) введемо наступні позначення:

$$\begin{aligned}
\bar{y}_{\min} &= (y_{k+1}^- \ y_{k+2}^- \ \dots \ y_{k+P}^-), & \bar{y}_{\max} &= (y_{k+1}^+ \ y_{k+2}^+ \ \dots \ y_{k+P}^+), \\
\bar{u}_{\min} &= (u_k^- \ u_{k+1}^- \ \dots \ u_{k+P-1}^-), & \bar{u}_{\max} &= (u_k^+ \ u_{k+1}^+ \ \dots \ u_{k+P-1}^+), \\
\Delta \bar{u}_{\min} &= (\Delta u_k^- \ \Delta u_{k+1}^- \ \dots \ \Delta u_{k+P-1}^-), & \Delta \bar{u}_{\max} &= (\Delta u_k^+ \ \Delta u_{k+1}^+ \ \dots \ \Delta u_{k+P-1}^+).
\end{aligned}$$

Будемо вважати, що керуюча дія u_{k-1} відомо. Тоді для кроку k справедливо

$$u_k = u_{k-1} + \Delta u_k. \text{ Звідси}$$

$$\bar{u} = M_0 u_{k-1} + M_u \Delta \bar{u}, \tag{19}$$

$$M_0 = \begin{pmatrix} E_{m \times m} \\ E_{m \times m} \\ \dots \\ E_{m \times m} \end{pmatrix}, M_u = \begin{pmatrix} E_{m \times m} & 0 & \dots & 0 \\ E_{m \times m} & E_{m \times m} & \dots & 0 \\ \dots & \dots & \dots & 0 \\ E_{m \times m} & E_{m \times m} & \dots & E_{m \times m} \end{pmatrix}$$

Враховуючи $\bar{z} = Lp_k + M\bar{v}$ та (19), обмеження (18) можуть бути представленні в виді системи лінійних нерівностей:

$$A_v \bar{v} \leq A_{im} + A_p p_k + A_u p_{k-1}. \quad (20)$$

Тепер замість задачі (12) можна підставити задачу з обмеженнями

$$J_k = J_k(\bar{v}) = \bar{v}^T H \bar{v} + 2 f^T \bar{v} + g \rightarrow \min_{v \in V \subset E^{mp}} \quad (21)$$

на допустимій множині

$$V = \{ \bar{v} \in E^{mp} : A_v \bar{v} \leq A_{im} + A_p p_k + A_u p_{k-1} \}. \quad (22)$$

Результат вирішення даної задачі визначить оптимально послідовність векторів $v_i = \Delta u_i$, $i = k, k+1, \dots, k+P-1$. По формулі $u_k = u_{k-1} + \Delta u_k$ визначається керуюча дія на поточному кроці та згідно стратегії MPC процес визначення оптимальної траєкторії повторюється.

Варто зазначити, що якщо матриця H в функціоналі (21) визначена додатно, то мінімізація функціоналу при наявності обмежень зводиться до стандартної задачі чисельного аналізу – випуклого квадратичного програмування.

Розробка MPC-регулятора за допомогою Matlab MPC Designer

Виконаємо дизайн MPC-регулятора з квадратичним функціоналом при наявності обмежень за допомогою засобів пакету Matlab MPC Designer. Дизайн MPC-регулятора буде виконано використовуючи математичну модель киснево-конвертерного процесу.

Перехідний процес зміни швидкості зневуглецювання v_c від зміни відстані фурми до рівня спокійної ванни H описується аперіодичною ланкою першого порядку (22) виду:

$$W_{v_c}(s) = \frac{k_{v_c}}{T_{v_c}s + 1}, \quad (22)$$

де $k_{v_c} \frac{m}{xв \cdot m}$ – коефіцієнт передачі по каналу відстань фурми до рівня спокійної ванни – швидкість зневуглецювання; $T_{v_c} c$ – стала часу.

В загальному вигляді залежність v_c від H можна представити як:

$$v_c = \frac{\alpha_0 - 20,42 \cdot H}{0,93 \cdot [1,031 + 0,102 \cdot (H - 1,5)^2] \cdot 10^3}, \quad (23)$$

Коефіцієнт α_0 змінюється протягом продувки (24): для першої частини продувки, коли вигоряють домішки чавуна, він збільшується відповідно зменшення швидкості вигорання домішок, для другої, коли швидкості окиснення силіцію, мангану і фосфору й відповідно витрати кисню на ці реакції наближені до нуля, визначається тільки інтенсивністю і хімічним складом дуття, в кінцевому періоді продувки знову зменшується, відповідно із зростанням у цей період кількості кисню, що йде на реакції окиснення заліза і допалення CO до CO₂. Тривалість першого періоду становить 0,25 тривалості продувки, третій починається з 0,8 тривалості продувки. При інтенсивності подачі дуття $400 \frac{m^3}{xв}$, масі чавуну 125 т, вмісту в чавуні силіцію 0,8 %, мангану 0,6 %, фосфору 0,08 %, зміну коефіцієнта α_0 в першому періоді, що для 160 тонного конвертера можна представити рівнянням:

$$\alpha_0 \left[\frac{m^3}{xв} \right] = \begin{cases} 173,23 + 72,933 \cdot \tau - 7,934 \cdot \tau^2 + 0,3797 \cdot \tau^3, & 1 \text{ період} \\ 386,98, & 2 \text{ період} \\ 386,98 - 109,038 \cdot (\tau - 16) + 11,87 \cdot (\tau - 16)^2 - 0,5677 \cdot (\tau - 16)^3, & 3 \text{ період} \end{cases} \quad (24)$$

Якщо значення k_{v_c} легко знайти, то при знаходженні T_{v_c} (25) виникають труднощі, що пов'язані з перехідними процесами у вимірювачі швидкості зневуглецювання. Тому для визначення постійної часу використовували

вимірювання тиску газів у перехідному газоході конвертера. Величина постійної часу нестационарна і також залежала від періоду плавки. Вона може бути описана функціями:

$$T_{v_c} [c] = \begin{cases} 1,143 + 4,446 \cdot \tau - 0,484 \cdot \tau^2, & 1 \text{ період} \\ 11,267, & 2 \text{ період} \\ 11,267 - 4,446 \cdot (\tau - 16) + 0,484 \cdot (\tau - 16)^2, & 3 \text{ період} \end{cases} \quad (25)$$

де $\tau [xв]$ – час продувки

Зміна швидкості зневуглецювання приводить до зміни ступеня окиснення вуглецю до CO_2 у порожнині конвертера. Цей процес також описується аперіодичною ланкою першого (26) порядку виду:

$$W_{\gamma_{CO_2}}(s) = \frac{k_{\gamma_{CO_2}}}{T_{\gamma_{CO_2}} s + 1}, \quad (26)$$

де $k_{\gamma_{CO_2}} \frac{\%CO_2 \cdot xв}{m}$ – коефіцієнт передачі по каналу швидкості зневуглецювання – ступінь окиснення вуглецю до CO_2 ; $T_{\gamma_{CO_2}} c$ – стала часу. За результатами експериментальних досліджень $T_{\gamma_{CO_2}} = 2,146 c$, коефіцієнт передачі по каналу швидкості зневуглецювання – ступінь окиснення вуглецю до CO_2 визначається з балансового рівняння витрати дугтя на продувку – $k_{\gamma_{CO_2}} = 3,33 \frac{\%CO_2 \cdot xв}{m}$.

Обидві ланки з'єднані послідовно (27) й передаточна функція системи в якій вхідна величина відстань фурми до рівня спокійної ванни, а вихідна – ступінь окиснення вуглецю до CO_2 буде мати вид:

$$W_{v_c}(s) = \frac{k_{v_c} \cdot k_{\gamma_{CO_2}}}{(T_{v_c} s + 1)(T_{\gamma_{CO_2}} s + 1)} = \frac{k_{\gamma_{CO_2}}^H}{(T_{v_c} s + 1)(T_{\gamma_{CO_2}} s + 1)}, \quad (27)$$

де $k_{v_c} \frac{\%CO_2}{m}$ – коефіцієнт передачі по каналу відстань фурми до рівня спокійної ванни – швидкість зневуглецювання;

ТОУ у АСР витрати кисню кисневого конвертера утворений фізичним з'єднанням пневмопривода, регулюючого клапана (РК) та ділянкою

трубопроводу з витратоміром (Д), що представляє ємкість кисню, яка створює опір потоку речовини.

Частина об'єкта РК–Д (вхідною величиною є положення РК, вихідною– витрата кисню) описується аперіодичною ланкою першого порядку зі

запізненням: $W_{об}(s) = \frac{K_{об}}{T_{об}s + 1} e^{-\tau_{об}s}$. Коефіцієнт передачі: $K_{об} = \frac{\Delta y}{\Delta u} = \frac{600 \text{ м}^3/\text{хв}}{100\% \text{ РК}} = 6 \frac{\text{м}^3/\text{хв}}{\% \text{ РК}}$.

З довідника “Автоматический контроль и регулирование в черной металлургии/ М.Д.Климовицкий; А.П.Копелович; «Металлургия»” стала часу та запізнення (28) об'єкту рівні: $T_{об} = 0,62 \text{ с}; \tau_{об} = 0,26 \text{ с}$ Передаточна функція об'єкту на ділянці РК–Д:

$$W_{РК-Д}(s) = \frac{K_{РК-Д}}{T_{РК-Д}s + 1} e^{-\tau_{РК-Д}s} = \frac{6}{0,62s + 1} e^{-0,26s} \quad (28)$$

Описана вище модель киснево-конвертерного процесу була розроблена у середовищі Matlab Simulink (рис. 2).

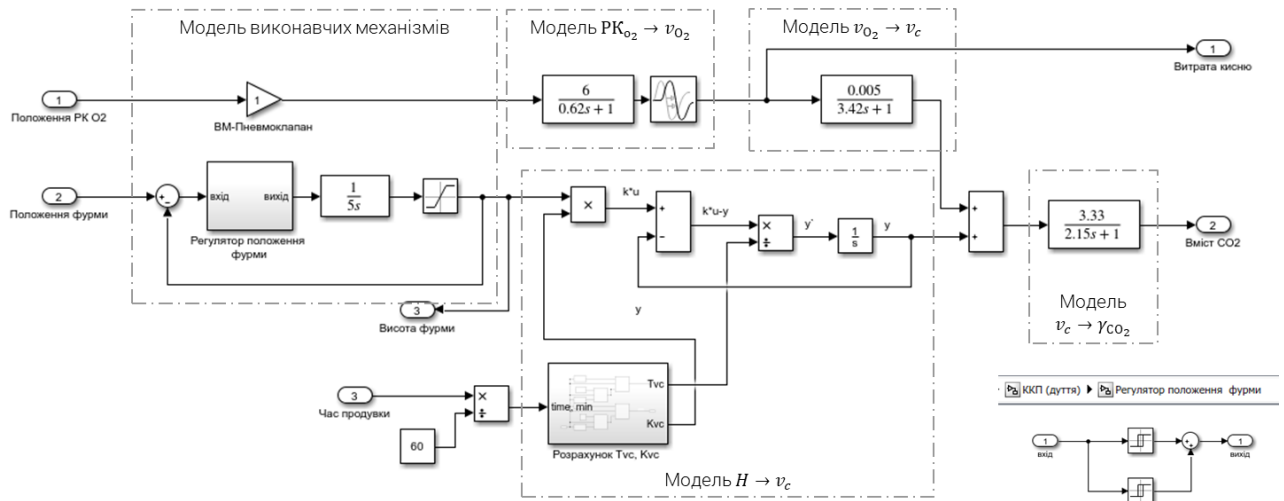


Рис. 2 Модель киснево-конвертерного процесу

У вікні MPC Designer обираємо розділ MPC Structure (рис. 3), де буде описано структуру MPC-регулятора та виконано лінеаризацію моделі киснево-конвертерного процесу.

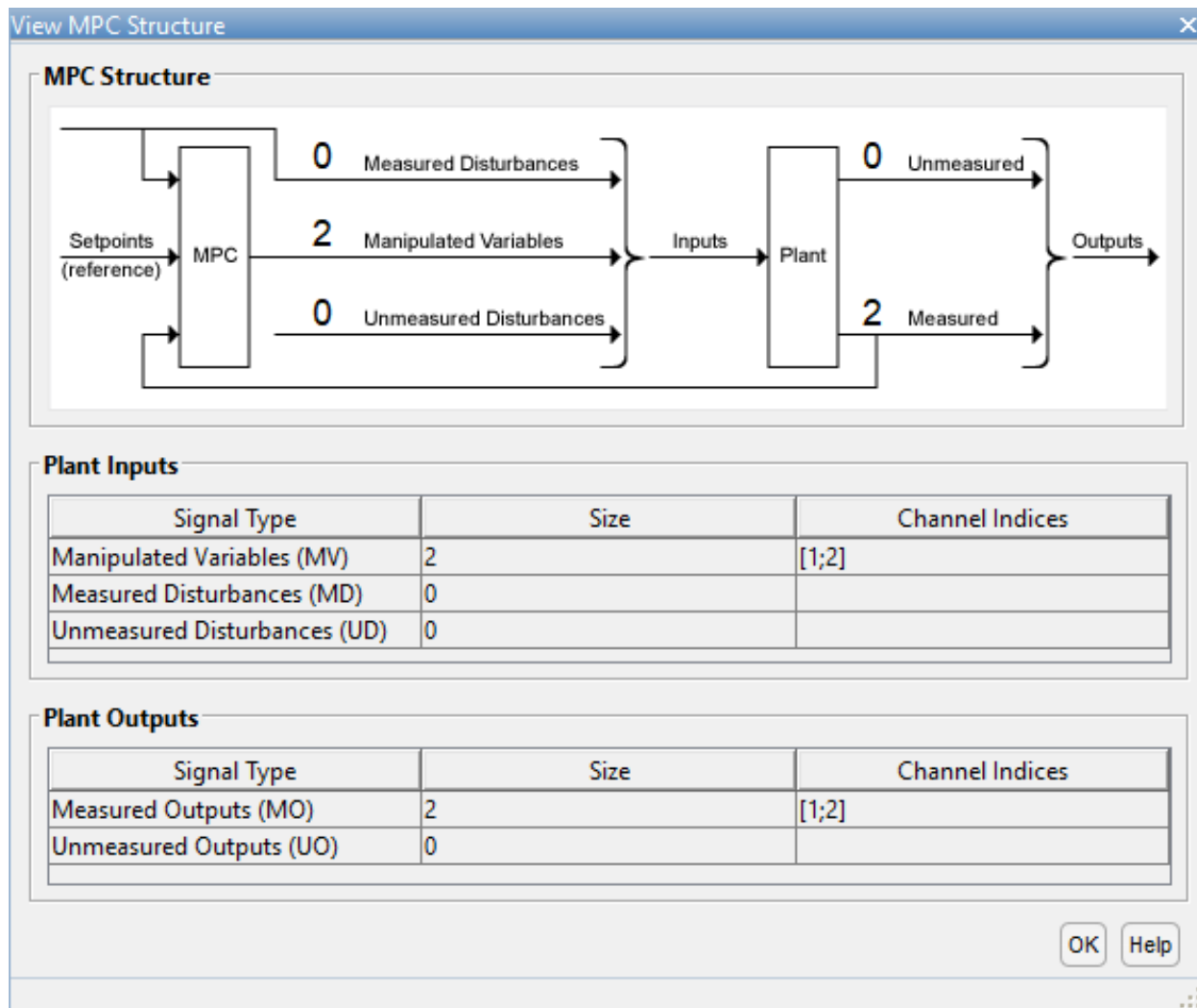


Рис. 3 Опис структури MPC-регулятора

Визначимо атрибути вхідних і вихідних каналів. На вкладці Designer MPC виберіть I/O Attributes (рис. 4). У діалоговому вікні Input and Output Channel Specifications вмажемо ім'я для кожного вхідного та вихідного каналів (рис. 5).

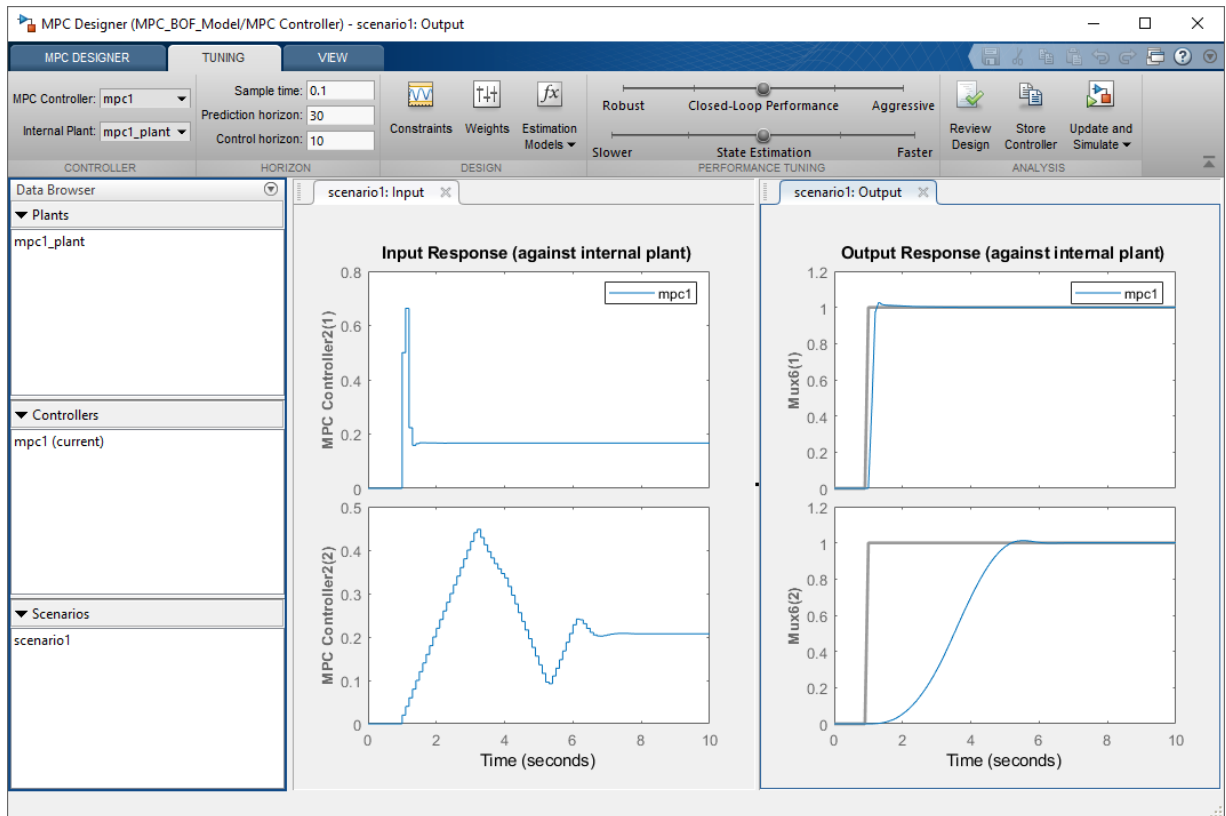


Рис. 4 Designer MPC

У стовпці Unit ви можете вказати одиниці для кожного каналу. Оскільки модель в просторі станів визначається за допомогою відхилень від номінальної робочої точки, встановіть Nominal Value для кожного вхідного та вихідного каналів до 0. Коефіцієнт масштабу для кожного каналу за замовчуванням складає 1.

Input and Output Channel Specifications					
Plant Inputs					
Channel	Type	Name	Unit	Nominal Val...	Scale Factor
u(1)	MV	Положення клапан кисню	%	0	1
u(2)	MV	Положення фурми	м	0	1
Plant Outputs					
Channel	Type	Name	Unit	Nominal Value	Scale Factor
y(1)	MO	Витрата кисню	м ³ /хв	0	1
y(2)	MO	Вміст CO2	%	0	1
OK Apply Cancel Help					

Рис. 1.5 Атрибути вхідних і вихідних каналів

Введемо у MPC-регулятор обмеження на вхідні величини виходячи з динамічних властивостей виконавчих механізмів (рис. 5).

Constraints (mpc1)					
Input Constraints					
Channel	Type	Min	Max	RateMin	RateMax
u(1)	MV	0	100	-0.5	0.5
u(2)	MV	0	2	-0.02	0.02

Рис. 5 Введені у MPC-регулятор обмеження на вхідні величини

Згідно MPC-стратегії здійснюється прогноз поведінки об'єкта, виконується оптимізації отриманої структури, в результаті якої буде знайдено оптимальне управління киснево-конвертерним процесом, описаним раніше. Отримане оптимальне управління застосовується на поточному кроці, після чого відбувається здви́г горизонту прогнозу і описана послідовність дій повторюється.

Процедура моделювання виконувалась у середовищі Matlab Simulink. Було обрано алгоритм вирішення рівнянь ode23s (stiff/mod. Rosenbrock) зі зміною величиною кроку (variable-step). Абсолютна і відносна точність розрахунків – 0,0001. Модель системи автоматичного регулювання киснево-конвертерним процесом з використанням MPC-стратегії наведена на рис. 6.

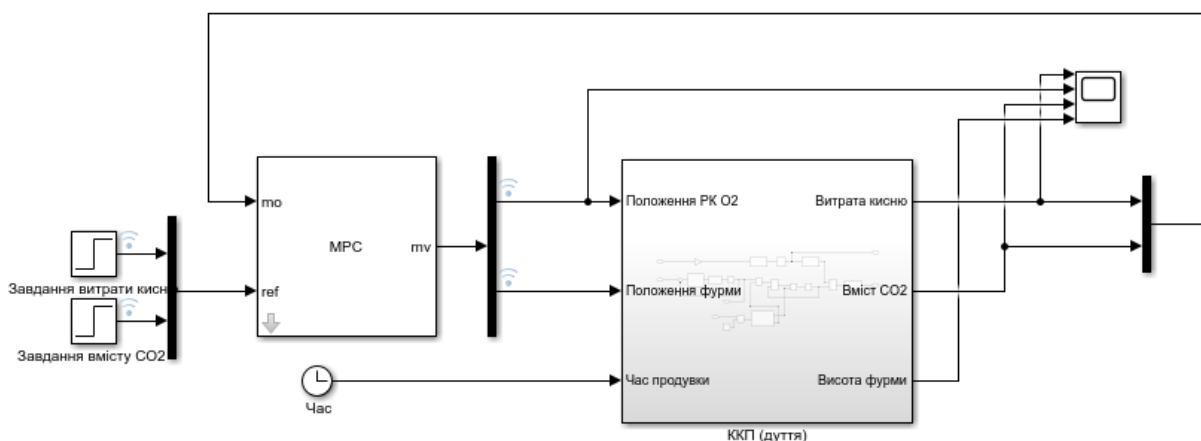


Рис. 6 Модель системи регулювання з MPC-регулятором

На рисунку 7 наведено перехідні процеси системи керування, в моделюванні завдання на витрату кисню встановлено 300 м³/хв, початкове

завдання на вміст CO₂ – 10%, на 50 сек продувки завдання встановлено на рівні 15%.

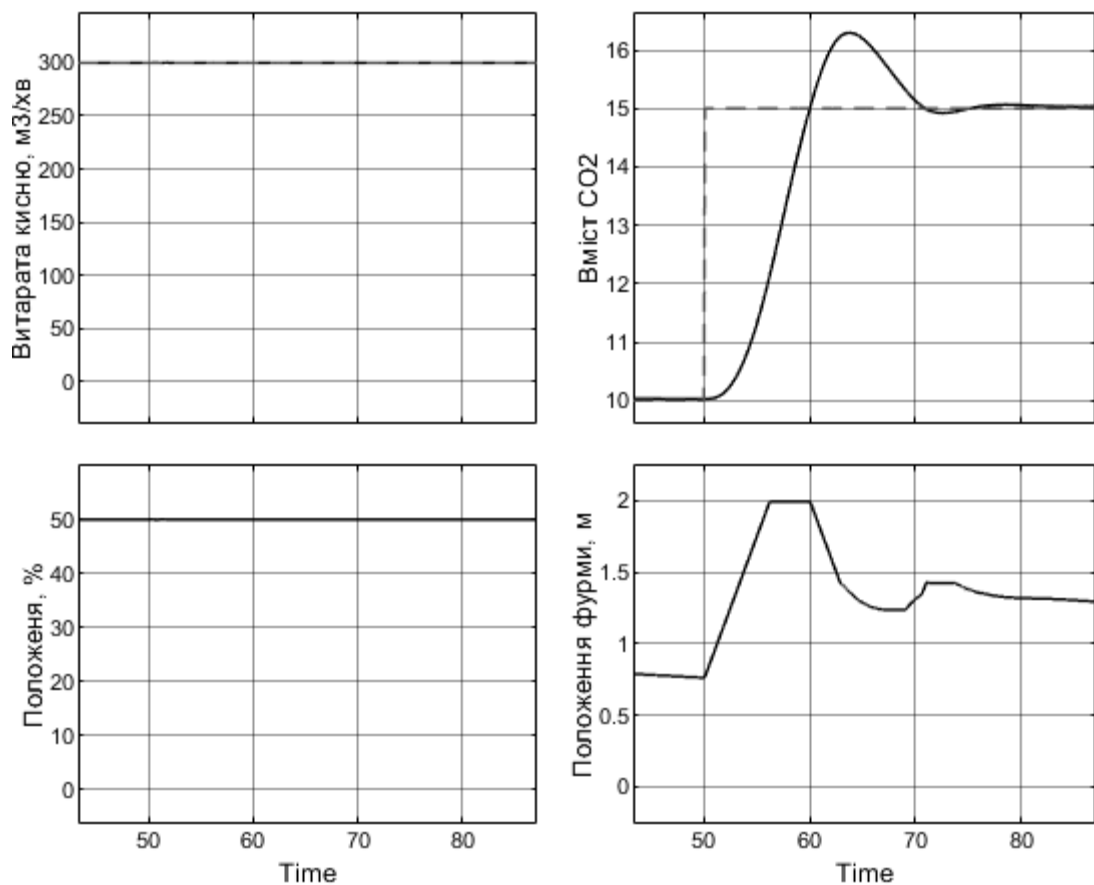


Рис. 7 Передні процеси системи автоматичного регулювання параметрами продувки киснево-конвертерної плавки

Отримані перехідні процеси системи автоматичного регулювання киснево-конверторним процесом з використанням MPC-стратегії забезпечують вимоги поставленні до якості роботи системи.

Розробка MPC-регулятора за допомогою **robust optimal control toolbox**

Для роботи з пакетом **robust optimal control toolbox** потрібно мати встановлене середовище розробки python (IDE). Спочатку потрібно імпортувати базові модулі та пакет **do-mpc**:

```
import numpy as np
import sys
from casadi import *
# Add do_mpc to path. This is not necessary if it was installed via pip
sys.path.append('../..')
# Import do_mpc package:
import do_mpc
```

Дизайн МРС-регулятора буде виконано використовуючи математичну модель біореактора, що має 4 технологічні параметри та 1 керуючу дію. Виконаємо ініціалізацію моделі:

```
model_type = 'continuous' # either 'discrete' or 'continuous'
model = do_mpc.model.Model(model_type)
```

Чотири технологічні параметри - це концентрація біомаси X_s , концентрація субстрату S_s , концентрація продукту P_s та об'єм V_s . Керуючий вхід - це витрата подачі u_{inp} S_s :

```
# States struct (optimization variables):
X_s = model.set_variable('_x', 'X_s')
S_s = model.set_variable('_x', 'S_s')
P_s = model.set_variable('_x', 'P_s')
V_s = model.set_variable('_x', 'V_s')

# Input struct (optimization variables):
inp = model.set_variable('_u', 'inp')
```

Модель системи описується звичайним диференціальним рівнянням:

$$\begin{aligned}\dot{X}_s &= \mu(S_s)X_s - \frac{u_{inp}}{V_s}X_s, \\ \dot{S}_s &= -\frac{\mu(S_s)X_s}{Y_x} - \frac{vX_s}{Y_p} + \frac{u_{inp}}{V_s}(S_{in} - S_s), \\ \dot{P}_s &= vX_s - \frac{u_{inp}}{V_s}P_s, \\ \dot{V}_s &= u_{inp},\end{aligned}$$

$$\mu(S_s) = \frac{\mu_m S_s}{K_m + S_s + (S_s^2/K_i)},$$

S_{in} - концентрація субстрату на вході, μ_m , K_m , K_i та v - кінетичні параметри Y_x та Y_p - коефіцієнти виходу. Концентрація підкладки на вході S_{in} та Y_x невизначена, тоді як інші параметри вважаються визначеними:

```
# Certain parameters
mu_m = 0.02
K_m = 0.05
K_i = 5.0
v_par = 0.004
Y_p = 1.2
```

```
# Uncertain parameters:
Y_x = model.set_variable('_p', 'Y_x')
S_in = model.set_variable('_p', 'S_in')
```

На наступному кроці встановлюється ODE для кожного стану:

```
# Auxiliary term
mu_S = mu_m*S_s/(K_m+S_s+(S_s**2/K_i))

# Differential equations
model.set_rhs('X_s', mu_S*X_s - inp/V_s*X_s)
model.set_rhs('S_s', -mu_S*X_s/Y_x - v_par*X_s/Y_p + inp/V_s*(S_in-S_s))
model.set_rhs('P_s', v_par*X_s - inp/V_s*P_s)
model.set_rhs('V_s', inp)
```

Нарешті, ініціалізація моделі завершена:

```
# Build the model
model.setup()
```

Далі налаштовується контролер. Спочатку об'єкт класу `mpc` генерується з моделлю прогнозування, визначеною вище:

```
mpc = do_mpc.controller.MPC(model)
```

Виконаємо налаштування регулятора:

```
setup_mpc = {
    'n_horizon': 20,
    'n_robust': 1,
    'open_loop': 0,
    't_step': 1.0,
    'state_discretization': 'collocation',
    'collocation_type': 'radau',
    'collocation_deg': 2,
    'collocation_ni': 2,
    'store_full_solution': True,
    # Use MA27 linear solver in ipopt for faster calculations:
    # 'nlpsol_opts': {'ipopt.linear_solver': 'MA27'}
}

mpc.set_param(**setup_mpc)
```

Біореактор використовується для виробництва пеніциліну. Тому метою контролера є максимізація концентрації продукту P_s .

```

mterm = -model.x['P_s'] # stage cost
lterm = -model.x['P_s'] # terminal cost

mpc.set_objective(mterm=mterm, lterm=lterm)
mpc.set_rterm(inp=1.0) # penalty on input changes

```

На наступному кроці встановлюються обмеження. У цьому випадку для кожного параметру та входу є лише верхня та нижня межі:

```

# lower bounds of the states
mpc.bounds['lower', '_x', 'X_s'] = 0.0
mpc.bounds['lower', '_x', 'S_s'] = -0.01
mpc.bounds['lower', '_x', 'P_s'] = 0.0
mpc.bounds['lower', '_x', 'V_s'] = 0.0

# upper bounds of the states
mpc.bounds['upper', '_x', 'X_s'] = 3.7
mpc.bounds['upper', '_x', 'P_s'] = 3.0

# upper and lower bounds of the control input
mpc.bounds['lower', '_u', 'inp'] = 0.0
mpc.bounds['upper', '_u', 'inp'] = 0.2

```

Значення двох невизначених параметрів Y_x і S_{in} , які розглядаються в дереві сценаріїв, задаються:

```

Y_x_values = np.array([0.5, 0.4, 0.3])
S_in_values = np.array([200.0, 220.0, 180.0])

mpc.set_uncertainty_values(Y_x = Y_x_values, S_in = S_in_values)

```

Налаштування MPC регулятора закінчено:

```

mpc.setup()

```

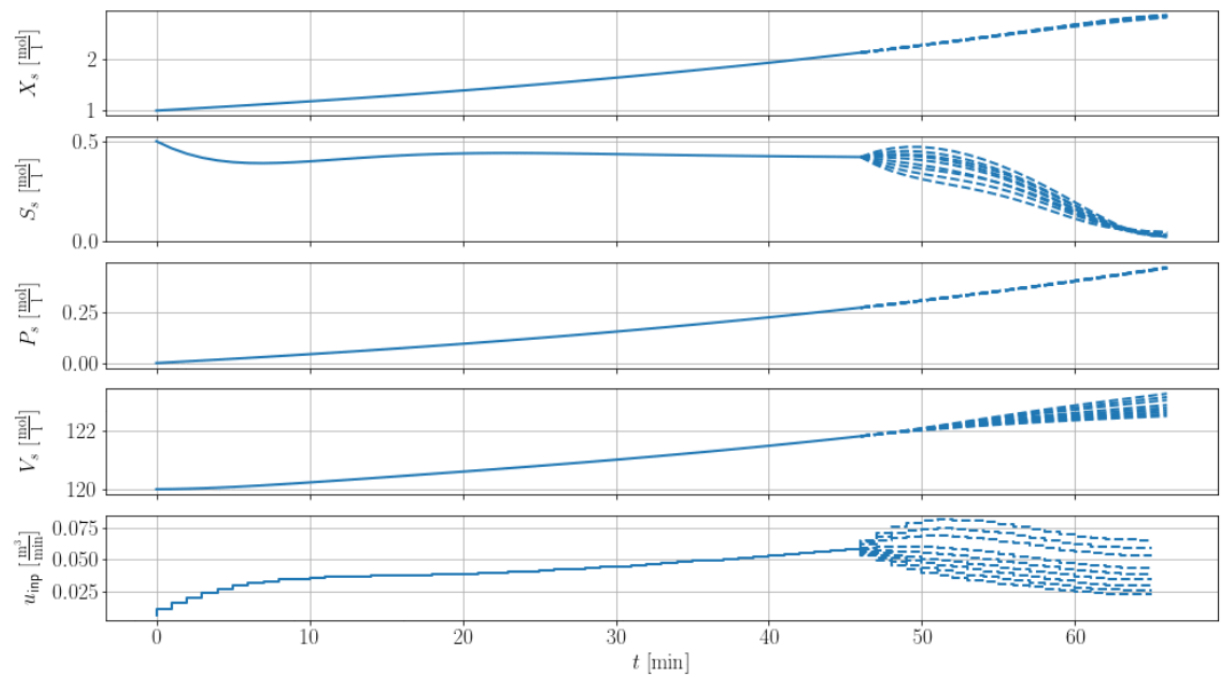


Рис. 8 Передні процеси системи автоматичного регулювання параметрами біореактора

Результат моделювання наведений вище (рис. 8), де суцільні лінії - це записані траєкторії, а пунктирні – прогнози можливої поведінки.

ПРАКТИЧНІ ЗАВДАННЯ

Завдання 1

Зібрати дані про динаміку процесу з тематики дисертаційної роботи. Провести аналіз і підготовку даних. На основі зібраних даних побудувати моделі процесу та провести його моделювання. Побудувати прогнозуючу модель обраного процесу для подальшого використання в синтезі модельно-прогнозуючого регулятора.

Завдання 2

Виконати розробку MPC-регулятора за допомогою пакету Matlab MPC Designer та виконати моделювання отриманої системи регулювання. Під час розробки регулятора врахувати технологічні обмеження процесу та вимоги до якості регулювання. Навести перехідні процеси системи та зробити висновки про якість регулювання.

Завдання 3

Виконати розробку MPC-регулятора за допомогою пакету robust optimal control toolbox та виконати моделювання отриманої системи регулювання. Під час розробки регулятора врахувати технологічні обмеження процесу та вимоги до якості регулювання. Навести перехідні процеси системи та зробити висновки про якість регулювання.

КОНТРОЛЬНІ ЗАПИТАННЯ

- 1) У чому полягає підхід Model Predictive Control?
- 2) Опишіть базову задачу Model Predictive Control.
- 3) Наведіть алгоритм пошуку матриць K і T базової задачі MPC.
- 4) За яким принципом згідно стратегії MPC-підходу застосовується знайдена оптимальна керуюча дія?
- 5) Чим характеризується якість управління в MPC-підході?
- 6) Які функціонали використовуються при синтезі регулятора, яка особливість лінійно-квадратичного функціоналу?
- 7) За яким принципом обираються елементи матриць R_{k+j} та Q_{k+j} в лінійно-квадратичному функціоналі?
- 8) В чому відмінність MPC та LQ регуляторів?
- 9) Опишіть задачу MPC-управління при наявності обмежень.
- 10) На які змінні можуть бути накладені обмеження в MPC-регуляторі?
- 11) З яких міркувань обирається горизонт прогнозу?
- 12) В чому переваги та недоліки MPC-підходу у порівнянні з іншими підходами?

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Mulholland M. Applied Process Control: Essential methods / Wiley-VCH. Weinheim, 2016. P. 460.
2. Wade H. Basic and Advanced Regulatory Control: System Design and Application, Third Edition / International Society of Automation. 2017. P. 567.
3. Camacho E. F., Bordons A. C. Model predictive control / Springer. 2007. P. 405.
4. Model predictive control toolbox design for nonstationary process/ O. Stepanets, Y. Mariiash // KPI Science News - 2021. №1 - pp. 42-49
<https://doi.org/10.20535/kpissn.2021.1.217992>
5. Abbas, H. S., Männel, G., né Hoffmann, C. H., & Rostalski, P. (2019). Tube-based model predictive control for linear parameter-varying systems with bounded rate of parameter variation. Automatica, 107, 21–28.
6. Korda, M., & Mezić, I. (2018). Linear predictors for nonlinear dynamical systems: Koopman operator meet model predictive control. Automatica, 93, 149–160.
7. Hilliard, T.; Kavcic, M.; Swan, L. Model predictive control for commercial buildings: Trends and opportunities. Adv. Build. Energy Res. 2016, 10, 172–190.
8. Stepanets O. Model Predictive control application in the energy saving technology of basic oxygen furnace / O. Stepanets, Y. Mariiash // Informatyka, Automatyka, Pomiar w Gospodarce i Ochronie Środowiska. - 2020. -№10/2. – pp. 70-74. (Index Copernicus) <https://doi.org/10.35784/iapgos.931>
9. Mayne, D. (2016). Robust and stochastic model predictive control: Are we going in the right direction? Annual Reviews in Control, 41, 184–192 .
10. Saltik, M. B., Özkan, L. , Ludlage, J. H. , Weiland, S. , & Van den Hof, P. M. (2018). An outlook on robust model predictive control algorithms: Reflections on performance and computational aspects. Journal of Process Control, 61, 77–102.

Посилання на інформаційні ресурси в Інтернеті

11. <https://www.mathworks.com/products/model-predictive-control.html>
12. https://www.do-mpc.com/en/latest/getting_started.html#Creating-the-model