

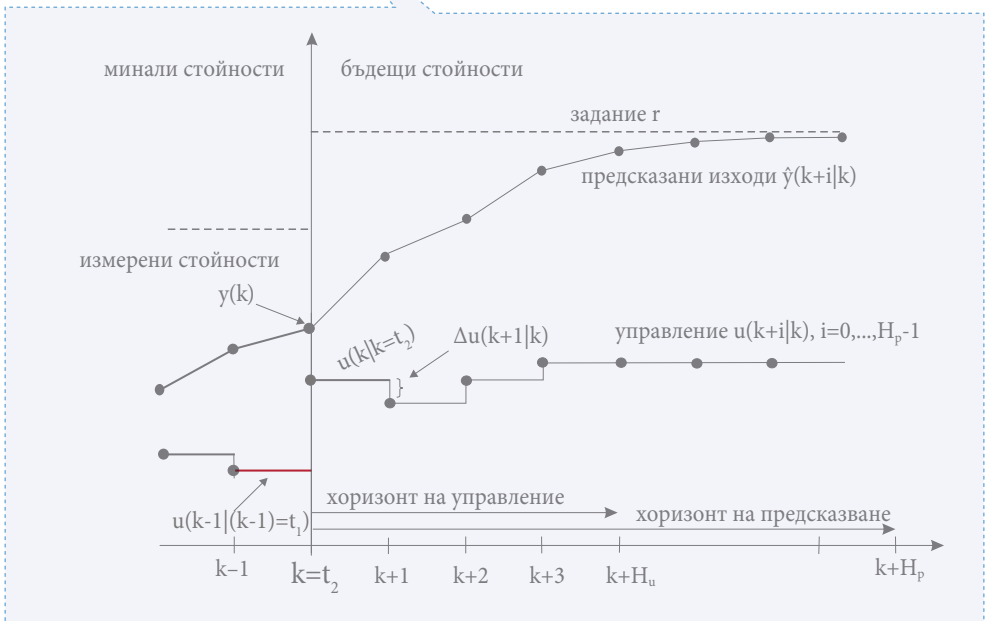
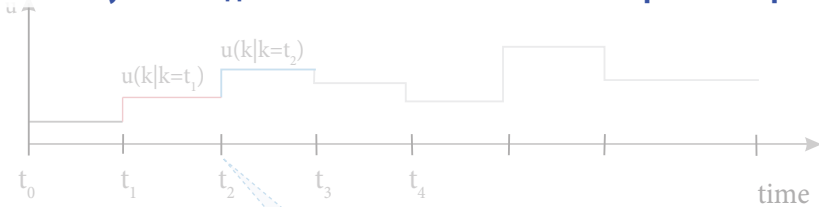


# РЪКОВОДСТВО ЗА ЛАБОРАТОРНИ УПРАЖНЕНИЯ ПО СИСТЕМЕН АНАЛИЗ

(СИСТЕМИ С МОДЕЛНО ПРЕДСКАЗВАЩО УПРАВЛЕНИЕ)

Севил Апула Ахмед

Михаил Георгиев Петров



## **Ръководство за лабораторни упражнения по Системен анализ (Системи с моделно предсказващо управление)**

Ръководството е предназначено за студенти в магистърския курс на обучение в специалност “Автоматика, информационна и управляваща техника” в Технически университет - София, филиал Пловдив. То покрива част от учебния материал по дисциплината “Системен анализ”, а също така може да се използва в курса на обучение и по избираемата дисциплина “Предсказващо управление”.

В ръководството са включени десет теми за лабораторни упражнения, които са представени с теоретични сведения, решени примери, указания за работа и разработени програми и модели за MATLAB® и MATLAB/Simulink®. Разгледани са основните възможности за проектиране и изследване на системи с МПУ с използване на MPC Toolbox™. Поради промени в инструментариума при работа в графичната среда на библиотеката в тема 4 е обърнато специално внимание на използването на библиотеката при версии на MATLAB® преди и след R2015b, съответно работа с MPC Tool и MPC Designer.

ISBN 978-619-04-0375-3 (pdf)

ISBN 978-619-04-0376-0 (ePub)

© Севил Ахмед, автор, 2024 © Михаил Петров, автор, 2024

© Севил Ахмед, издател, 2024

# Съдържание

Основни съкращения.....	4
Основни означения.....	5
Въведение в системите с предсказващо управление .....	6
<b>ТЕМА 1</b>	
Идентификация и построяване на модели от предавателна характеристика на система.....	16
<b>ТЕМА 2</b>	
Получаване на на модели за целите на моделното предсказващо управление.....	24
<b>ТЕМА 3</b>	
Комбиниране на предсказващи модели .....	39
<b>ТЕМА 4</b>	
Графична среда за проектиране и изследване на системи с моделно предсказващо управление.....	47
<b>ТЕМА 5</b>	
Изследване на линейни системи с МПУ в средатана MATLAB/Simulink® .....	68
<b>ТЕМА 6</b>	
Управление на нелинейна система чрез многомоделен предсказващ регулатор .....	77
<b>ТЕМА 7</b>	
Синтез на предсказващ регулатор за управление на агрегат за производство на хартия .....	89
<b>ТЕМА 8</b>	
Система с моделно предсказващо управление на сервомеханизъм .....	101
<b>ТЕМА 9</b>	
Изследване влиянието на параметрите на предсказващия регулатор върху качеството на управление .....	111
<b>ТЕМА 10</b>	
Невронно-размито МПУ с модел в пространството на състоянието.....	118
Използвани литературни източници .....	127

## Основни съкращения

МНМК	-	Метод на най-малките квадрати
МПУ	-	Моделно предсказващо управление
НМПУ	-	Нелинейно моделно предсказващо управление
ARIMAX	AutoRegressive Integrated Moving Average with eXogenous Input	Авторегресия интегрираща пълзяща средна стойност с външна променлива
ARIX	AutoRegressive Integrated with eXogenous inputs,	Авторегресия с интегрираща съставка и външна променлива
ARMAX	AutoRegressive Moving Average eXogenous inputs	Авторегресия пълзяща средна стойност с външна променлива
ARX	AutoRegressive eXogenous inputs	Авторегресия с външна променлива
DMC	Dynamic Matrix Control	Динамично матрично управление
FIR	Finite Impulse Response	Импулсна характеристика
FNMPC	Fuzzy-Neural MPC	Невронно-размито моделно предсказващо управление
FPID	Fuzzy PID	Размито ПИД управление
FSR	Finite Step Response	Преходна характеристика
IAE	Integral of Absolute Error	Интегрална абсолютна грешка
IE	Integral of Error	Интегрална грешка
ISE	Integral of Squared Error	Интегрална квадратична грешка
MAC	Model Algorithmic Control	Моделно алгоритмично управление
MIMO	Multi-Input Multi-Output	Система с много входове и много изходи
MISO	Multi-Input Single-Output	Система с много входове и един изход
MPC	Model Predictive Control	Моделно предсказващо управление
SISO	Single-Input Single-Output	Система с един вход и един изход
SMOC	Shell Multivariable Optimizing Controller	Регулатор с многопараметрична оптимизация на Shell

## Основни означения

$y(k)$	Изход на обекта на управление
$x(k)$	Вектор на състоянията
$u(k)$	Вход на обекта на управление
$\Delta u(k)$	Изменение на входа на обекта на управление
$\hat{x}(k)$	Предсказан/моделиран вектор на състоянията
$\hat{y}(k)$	Предсказан/моделиран изход
$r(k)$	Задание
$n$	Брой променливи на състоянието на обекта на управление
$k$	Текущ момент (такт) на дискретното време
$m$	Брой входове на обекта на управление
$q$	Брой изходи на обекта на управление
$A$	Матрица на състоянията
$B$	Матрица на управлението
$C$	Матрица на изхода
$D$	Тегловна матрица
$J(.)$	Целеви критерий
$f(.)$	Нелинейна функция
$N_u$	Хоризонт на управление
$N_2$	Хоризонт на предсказване
$N_1$	Начален хоризонт на управление
$diag$	Диагонална матрица
$\  \cdot \ $	Норма на функция
$I$	Единична матрица
$\lambda$	Положителен тегловен коефициент (наказващ фактор)
$w(i)$	Коефициенти на тегловната функция на обекта
$h(i)$	Коефициенти на преходната функция на обекта
$v(k)$	Съставка на шума
$A(q-1)$	Полином на изходите на параметричния модел
$B(q-1)$	Полином на входовете на параметричния модел
$C(q-1)$	Полином на външното въздействие (шум) на параметричния модел
$e(k)$	Бял шум
$d$	Чисто закъснение (цяло число)
$Q$	Тегловна матрица приложена върху предсказаната грешка
$R$	Тегловна матрица приложена върху управлението

## Въведение в системите с предсказващо управление

Моделното предсказващо управление (МПУ) има дълга история в областта на управлението, датираща от 70-те години на миналия век. През първите няколко години от развитието си приложението на МПУ е ограничено основно в сферите на нефтената, химическата промишленост и енергетиката [10, 13.19]. Различни алгоритми за моделно предсказващо управление се реализират успешно в редица други сфери като хранителната индустрия, медицина, роботика, транспорт, космонавтика, автомобилостроене и метеорология [15, 15, 17, 19, 21].

Моделното предсказващо управление е една от малкото концепции, които са обект на постоянен интерес както в научните, така и в приложните (инженерни) среди. За тази популярност допринасят няколко важни и основни за МПУ аспекта:

- Интуитивен подход, улесняващ приложението на МПУ за системи с различна сложност и размерност;
- Възможност за налагане и отработване на ограничения от различно естество. Това е изключително предимство на МПУ, тъй като всички обекти и процеси в индустрията трябва да отговарят на различни по тип ограничения свързани с осигуряването на безопасна работа или обезпечаване на икономически показатели;
- Възможност за оптимизация в реално-време, съобразена с изискванията на конкретната система;
- Начинът, по който МПУ съчетава всичко изброено до тук в една сравнително опростена схема за адаптивно управление.

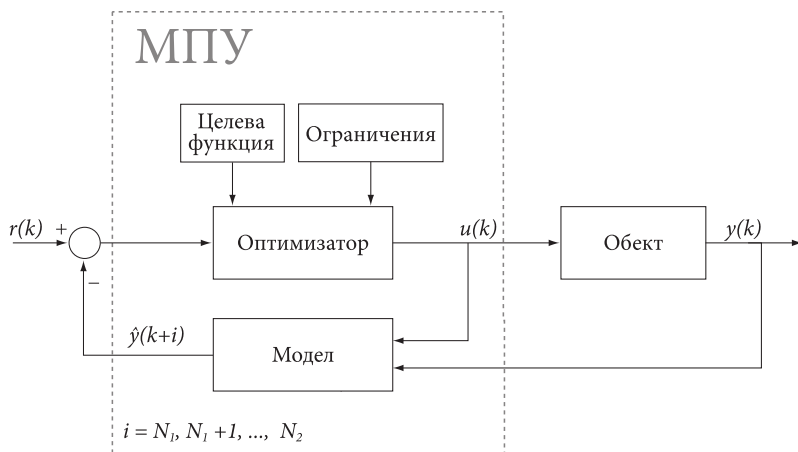
В исторически план развитието на МПУ се свързва с развитието на предсказващото управление. Приема се, че началото е поставено през 70-те години на миналия век, въпреки че идеята за използването на предсказващи модели за целите на управлението не е нова. Пример за това е предикторът на Смит – регулатор, използван основно за системи с голямо транспортно закъснение. В основата на този регулатор е модел без закъснение, който се използва за определяне на бъдещите стойности на изхода на обекта.

## 1. Постановка на моделното предсказващо управление

Моделното предсказващо управление е подход, при който алгоритъмът за предсказващо управление използва модел на обекта на управление за изчисляване на бъдещото поведение на обекта. Алгоритъмът на МПУ решава предварително дефинирана оптимизационна задача във всеки такт на управление, с цел изчисляване последователността от бъдещите оптимални стойности на управляващите въздействия. Само първата стойност от оптималната последователност се подава като вход на системата (принцип на пълзящия хоризонт). Целевият критерий на оптимизационната задача включва предсказаните стойности за изхода на модела и тези на управляващото въздействие.

Основните елементи в класическата постановката на МПУ са:

- Модел на изследваната система;
- Целеви критерий, включващ предсказаните стойности на изхода на системата и управлението;
- Оптимизационен алгоритъм, изчисляващ оптималната последователност на управляващите въздействия, която минимизира целевия критерий съобразно наложените ограничения;
- Принцип на пълзящия хоризонт (RHC, Receding Horizon Control).



Фиг. 1. Основно представяне на системите с МПУ

Моделът в структурата на МПУ (Фиг. 1) описва динамиката на системата. Чрез него се предсказват бъдещите стойности на управляваните величини  $\hat{y}(k+i)$  ( $i = N_1, N_1 + 1, \dots, N_2$ ) в рамките на краен интервал от време  $N_2$ , наречен **хоризонт на предсказване**. Предсказаните стойности са функции на бъдещите управляващи въздействия, изчислени в  $(k+i)$ -ти момент на базата на информация за  $k$ -тия такт ( $u(k+i|k)$ ,  $i = 0, 1, \dots, N_u - 1$ ) и на текущото състояние на системата. **Хоризонтът на управление** е отбелязан като  $N_u$ . На базата на предсказаните стойности се формира целеви критерий, представляващ най-често сума от квадратите на предсказаната системна грешка (разлика между желаната  $r(k+i)$  и предсказана стойност на управляваната величина  $\hat{y}(k+i)$  и изменението на управляващо въздействие  $\Delta u(k+i)$ ). Тази формулировка на целевия критерий (1) е заложена в идеята на обобщеното предсказващо управление [9]:

$$J(k) = \sum_{i=N_1}^{N_2} \|r(k+i) - \hat{y}(k+i)\|^2 + \lambda \sum_{i=0}^{N_u-1} \|\Delta u(k+i)\|^2 \quad (1)$$

където  $\lambda$  е тегловен коефициент, който определя с каква тежест изменението на управляващото въздействие участва в дефинирания критерий. **Минималният хоризонт на предсказване**  $N_1$  задава първата стойност, с която предсказаният изход на системата  $\hat{y}(k+i)$  ще участва в целевия критерий на МПУ.

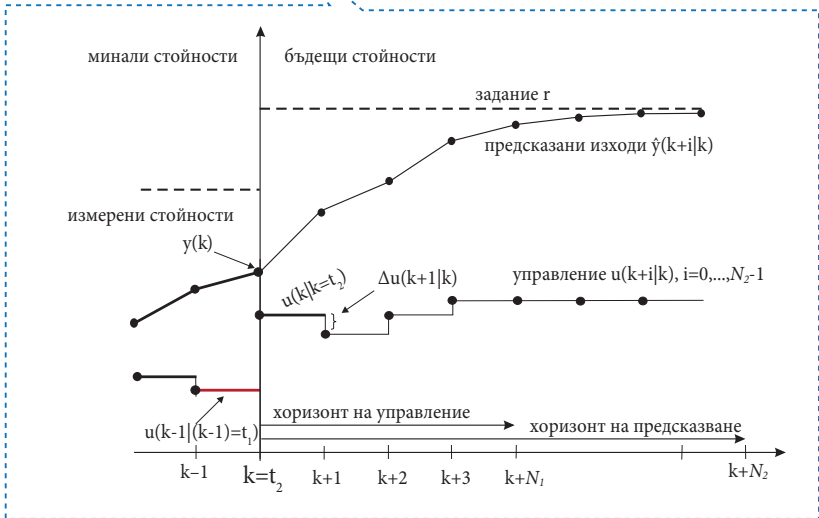
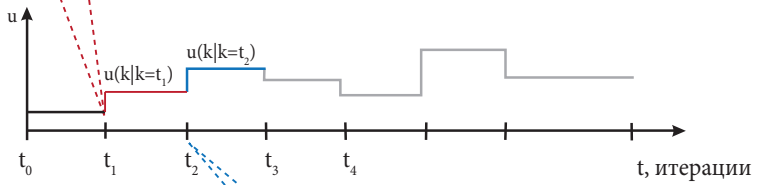
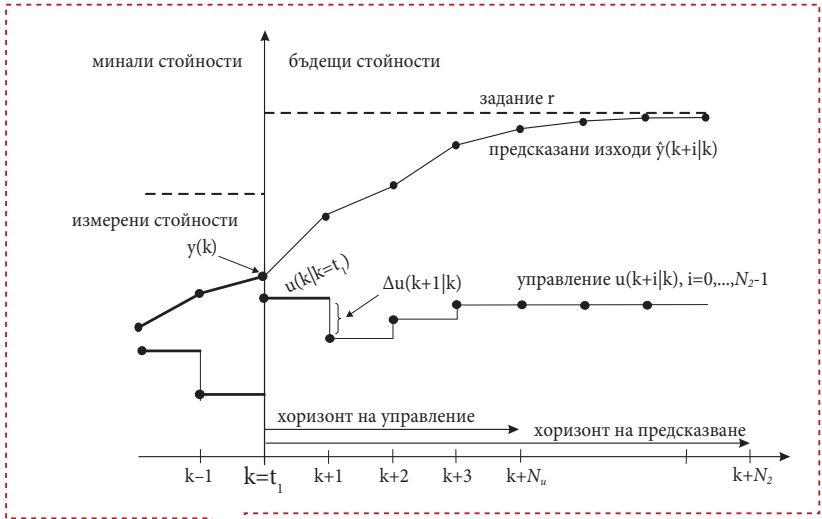
Във всеки такт на управление  $k$  (Фиг. 2) МПУ минимизира целевия критерий от вида (1) с цел получаване на оптимална последователност от бъдещи стойности на управляващото въздействие.

Изчислената оптимална последователност се състои от  $N_u$  на брой бъдещи управляващи въздействия, които осигуряват желаното поведение на системата за целия хоризонт на предсказване  $N_2$ .

$$\min J(\Delta u, k) = \min \left\{ \sum_{i=N_1}^{N_2} \|r(k+i) - \hat{y}(k+i)\|^2 + \lambda \sum_{i=0}^{N_u-1} \|\Delta u(k+i)\|^2 \right\} \quad (2)$$

$$\begin{aligned} \hat{y}(k+i) &\in \hat{Y}, & i &= N_1, N_1 + 1, \dots, N_2 \\ u(k+i) &\in U, & i &= 0, 1, \dots, N_u \\ u(k+i) &= u(k + N_u - 1), & i &= N_u, N_u + 1, \dots, N_2 - 1 \\ \Delta u(k+i) &= u(k+i) - u(k+i-1) \end{aligned}$$





Фиг. 2. Принцип на пълзящия хоризонт

Оптимизационната задача (2) се решава за всяка стойност (всеки такт)  $k$  от хоризонта на предсказване. Хоризонтът на предсказване се измества с една стъпка напред при всяка следваща стъпка на изчисление  $k+i$  ( $i = N_1, N_1 + 1, \dots, N_2$ ). Хоризонтът на управление също се придвижва с една стъпка напред до достигане на крайната си стойност  $N_u$ . Това е причината, поради която този подход на адаптивно управление се отъждествява с термина **пълзящ хоризонт** [11, 12]. При прилагане на принципа на пълзящия хоризонт, хоризонтът на управление е винаги по-малък от хоризонта на предсказване ( $N_u < N_2$ ). Малкият хоризонт на управление придава робастни свойства на управляващата система. От друга страна използването на голям хоризонт  $N_u$  осигурява гъвкавост на управляващия сигнал, а оттам и по-добро качество на управление. Изборът на хоризонтите на предсказване и управление трябва да е съобразен с динамиката на конкретната система за управление. Големината на  $N_2$  определя до каква степен МПУ ще е в състояние да обхване (предскаже) поведението на процеса, а  $N_u$  следва да покрие интервала на дискретизация, за да е адекватно приложеното управление. При работа с големи хоризонти трябва да се има предвид, че се увеличава в значителна степен броя на изчисленията, тъй като те определят броя на променливите в оптимизационната задача на МПУ.

Видът на модела в структурата на МПУ (Фиг. 1) и наложените ограничения определят типа на оптимизационната задача, представена в най-общ вид чрез израза (2). Линейните модели и ограниченията тип неравенства (и липсата на ограничения) гарантират наличието на глобален екстремум, тъй като повърхнината на този тип целеви функции е изпъкнала и непрекъсната. Използването на нелинейни модели и/или ограничения от тип равенства налага решаването на нелинейна оптимизационна задача, за която невинаги съществува глобално решение. Поради това, видът на използваните модели се приема за основен класификатор в системите за МПУ. Фиг. 3 представя опростена класификация на моделно предсказващите алгоритми на базата на вида на използваните в МПУ модели (линейни или нелинейни) [20].

## 2. Видове модели за целите на МПУ

Както вече беше отбелязано, основното предизвикателство пред нелинейното моделно предсказващо управление се състои в намирането на модел, изразяващ възможно най-точно спецификите на управляваната нелинейна система. Именно коректността на използвания модел определя в най-голяма степен ефективността на управлението.



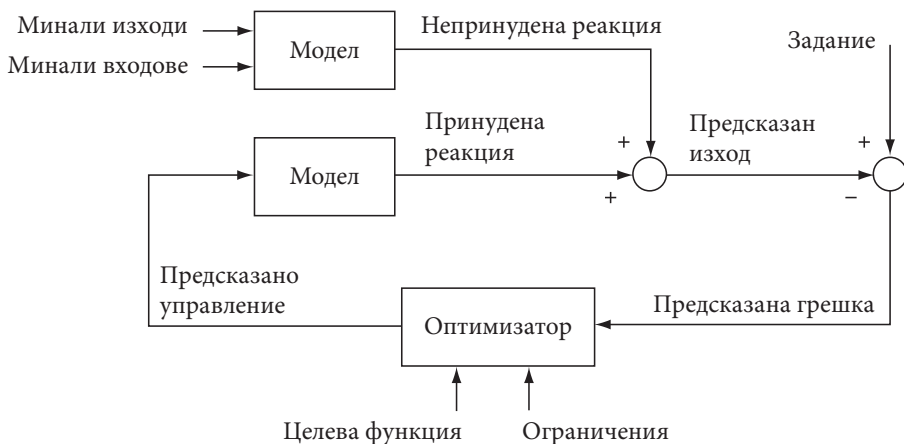
Фиг. 3 Основна класификация на моделно предсказващите алгоритми

От теорията на идентификацията е известно, че основните групи нелинейни динамични модели са: блоково ориентирани модели; обобщени линейни по параметри модели; псевдолинейни модели с параметри, зависещи от външен сигнал; псевдолинейни модели с по части постоянни параметри (мулти-модели). Редица автори обръщат внимание на особеностите при идентификацията на нелинейни системи за целите на предсказващото управление. Основните изисквания при моделирането за целите на нелинейното МПУ, както и обзор на методите за моделиране и идентификация може да бъде намерен в статията на Lee [16]. Трудът на Pearson от 2003 г. [18] представлява своеобразно ръководство за избора на подходяща структура на нелинейния модел. В него авторът описва някои основни класове нелинейни структури, класифицира ги и дава отговор на въпроса какво представлява “добрият” модел. Най-важните критерии, на които трябва да отговарят методите за нелинейно моделиране в предсказващото управление, могат да се обобщят в следните няколко аспекта [6]:

- Моделът трябва да разчита на описание с малък брой параметри (коефициенти), които са лесни за тълкуване;
- Процедура по изграждане на модела трябва да бъде проста, надеждна и да позволява включването на априорна информация в различни форми;

- Моделът трябва да е с подходяща структура, която позволява прилагането на стандартни техники за оптимизация, с цел намаляване на изчислителната тежест на процедурата за намиране на оптималното управление, която в повечето случаи се осъществява в реално време (on-line).

Фиг. 4. представя важноста на модела в схемата на МПУ. Точният модел определя в голяма степен прецизността на алгоритъма за управление като цяло. Предсказанието от модела изход обикновено се разглежда като съставна траектория, която обединява непринудената (свободната) реакция на системата от една страна и траекторията на принуденото движение, зависеща от текущото и бъдещото изменение на управлението от друга страна. Траекторията на свободната реакция участва в оптимизационната задача с еднократно изчислената си стойност, а принудената реакция се преизчислява за всяко предсказано изменение управляващите въздействия.



Фиг. 4. Базова структура на моделно предсказващ регулатор

Понастоящем за целите на МПУ на нелинейни технологични обекти се използват основно: **модели на Волтера; модели на Хамерщайн и Винер; невронни мрежи; модели, базирани върху размита логика**. Моделите на Хамерщайн и на Винер се наричат още блоково ориентирани модели. Тези модели се изграждат на основата на структурни блокови схеми, които пред-

ставяват поотделно безинерционна част, описвана с нелинейна статична характеристика  $y = \varphi(u)$  и линейна инерционна част с предавателна функция  $W(q^{-1})$ . Моделите на Волетера се характеризират с полиномната си структура. Представителите на изкуствения интелект – размита логика и невронни мрежи, не разчитат на механистични модели и са изключително подходящи за моделиране на нелинейни системи. Въпреки приликите на изчислително ниво, двата модела на изкуствения интелект се отличават с характерни особености и свойства. Невронните мрежи са типични представители на апроксиматорите тип “черна кутия”. Докато размитите структури се причисляват към моделите “сива кутия”, позволявайки прилагането на предварително придобити знания в процеса на моделирането. Използването на тези два типа модели в предсказващото управление изисква решаване на оптимизационната задача на МПУ във всеки такт на управление. За целта се прилагат различни техники като итеративни подходи за оптимизация, линеаризация и дискретно търсене [6].

При моделирането със средствата на изкуствения интелект се използват добре познатите модели от линейната идентификация. Те участват в структурата на много от методите за идентификация на нелинейни системи с приложение на размита логика и невронни мрежи. В някои от тях се използват в познатата им форма, в други се модифицират или комбинират с цел подобряване на моделиращите свойства на интелигентните структури. Основните групи конвенционални модели за целите на МПУ са три:

• **Непараметрични модели.** Модели от типа **преходна (FSR, Finite Step Response)** (3) и **импулсна (FIR, Finite Impulse Response)** (4) характеристики на обекта, използвани главно в дискретна форма:

$$y(k) = \sum_{i=0}^n w(i)u(k-i) + v(k) \quad (3)$$

$$y(k) = \sum_{i=0}^n h(i)\Delta u(k-i) + v(k), \quad (4)$$

където  $w(i)$  и  $h(i)$  са съответно коефициентите (стойностите) на тегловната и преходната функции на обекта,  $v(k)$  отчита шума, приведен към изхода на обекта и влиянието на немоделираната динамика. Стойностите на  $n$  зависят

от времето  $t_m$  на затихване на преходния процес и от такта на дискретизация  $T_0$ . Основен недостатък на този тип модели е големият брой коефициенти, които трябва да бъдат експериментално определени [2]. Такива непараметрични модели се използват в алгоритмите с предсказващо управление като динамичното матрично програмиране (DMC, Dynamic Matrix Control) – използва преходна функция (3) и моделното алгоритмично управление (MAC, Model Algorithmic Control), което използва модел от типа FIR (4).

**Параметрични модели.** Най-разпространените и често използвани са ARX и ARMAX моделите. Структурата на **ARMAX** (AutoRegresive Moving Average eXogenous inputs, Авторегресия пълзяща средна стойност с външна променлива) може да се приеме за най-пълен (обобщаващ) от тази група модели [3]. В полиномен вид той изглежда по следния начин:

$$A(q^{-1})y(k) = B(q^{-1})u(k-d-1) + C(q^{-1})e(k)$$

или

(5)

$$y(k) = \frac{B(q^{-1})}{A(q^{-1})}u(k-d-1) + \frac{C(q^{-1})}{A(q^{-1})}e(k)$$

където  $A(q^{-1})$ ,  $B(q^{-1})$  и  $C(q^{-1})$  са полиноми със степени съответно –  $n_y$ ,  $n_u$  ( $n_u < n_y$ ) и  $n_e$  ( $n_y = n_e$ ),  $d$  е чисто закъснение (брой тактове), а  $e(k)$  е бял шум [2]. ARMAX моделът има следното представяне във времевата област:

$$y(k) = \sum_{i=1}^{n_y} a_i y(k-i) + \sum_{i=0}^{n_u} b_i u(k-d-i) + \sum_{i=0}^{n_e} c_i e(k-i)$$

(6)

Моделът **ARX** (AutoRegresive eXogenous inputs, Авторегресия с външна променлива) може да се разглежда като частен случай на ARMAX структурата [3], при който липсва полиномът  $C(q^{-1})$ , описващ влиянието на белия шум  $e(k)$ .

$$y(k) = \frac{B(q^{-1})}{A(q^{-1})}u(k-d-1) + \frac{1}{A(q^{-1})}e(k),$$

(7)

$$y(k) = \sum_{i=1}^{n_y} a_i y(k-i) + \sum_{i=0}^{n_u} b_i u(k-d-i) + e(k) \quad (8)$$

И двата най-предпочитани параметрични модели могат да се използват и в леко модифицирана форма, в която се използва изменението на управляващото въздействие. Това е обусловено от добавянето на интегрална съставка ( $\Delta = 1 - q^{-1}$ ) в описанието на модела с цел по-добро описание на нестационарния характер на смущението. Съответстващият на ARMAX структурата модел е известен като ARIMAX (AutoRegressive Integrated Moving Average with eXogenous Input, Авторегресия интегрираща пълзяща средна стойност с външна променлива) [2]:

$$y(k) = \frac{B(q^{-1})}{A(q^{-1})} u(k-d-1) + \frac{C(q^{-1})}{(1-q^{-1})A(q^{-1})} e(k) = \frac{B(q^{-1})}{\tilde{A}(q^{-1})} \Delta u(k-d-1) + \frac{C(q^{-1})}{\tilde{A}(q^{-1})} e(k) \quad (9)$$

По структура (9) е същият като ARMAX модела, но полиномът  $\tilde{A}$  ( $\tilde{A} = \Delta A$ ) е от степен  $(n_y + 1)$  и вместо управляващия сигнал  $u(k)$  се използва неговото изменение  $\Delta u(k)$ . По аналогичен начин се получава и моделът ARIX (AutoRegressive Integrated with eXogenous inputs, Авторегресия с интегрираща съставка и външна променлива).

• **Модели в пространство на състоянията.** Използват се известните матрични уравнения в съкратена форма:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \end{aligned} \quad (10)$$

където векторите  $x(k)$ ,  $y(k)$ ,  $u(k)$  представляват вектори на състоянието, изхода и входа с размерности  $n$ ,  $q$  и  $m$ . Матриците  $A$ ,  $B$ ,  $C$  са матрици на състоянието, входа, изхода и техните размерности съответстват на тези на векторите. Моделите в пространството на състоянията са предпочитани в последните години. Някои от основните предимствата на тези модели според Camacho и Bordons [8] са: значително намаленият брой на коефициентите, които участват в процеса на моделиране в сравнение с моделите, ползващи предавателни функции; възможност за изследване на устойчивост; значително по-лесно моделиране на многомерни обекти и процеси.

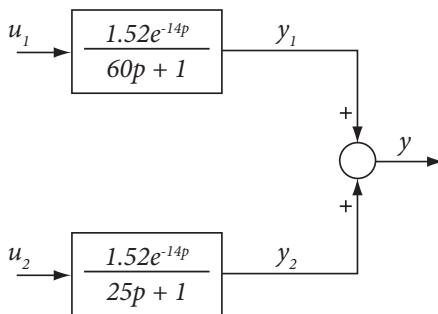
## ТЕМА 1

### Идентификация и построяване на модели от предавателна характеристика на система

Целта на лабораторното упражнение е запознаване с методите за получаване на предавателна характеристика и построяване на моделите на обекта.

#### 1. Задачи за изпълнение.

**Задача 1.** Да се извърши идентификация на обекта от Фиг. 1.1 с два входа и един изход (MISO).



Фиг. 1.1 Структурна схема на обекта на идентификация

**Задача 2.** Да се изведе предавателната характеристика на обекта за двата канала.

**Задача 3.** Да се разпечатаат отбелязаните с „theta“ в програмата променливи.

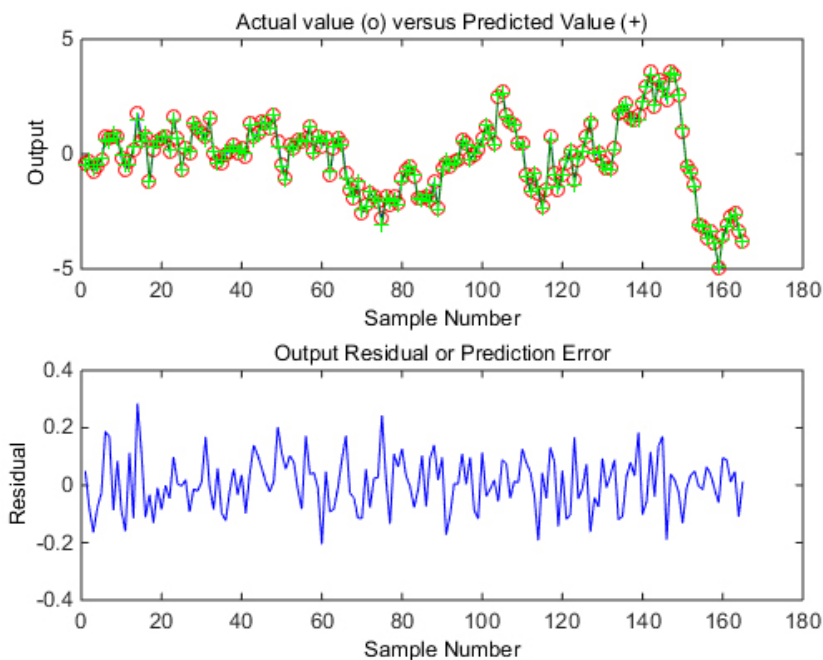
**Задача 4.** Да се преобразува модела във формат преходна характеристика - step (imp2step).



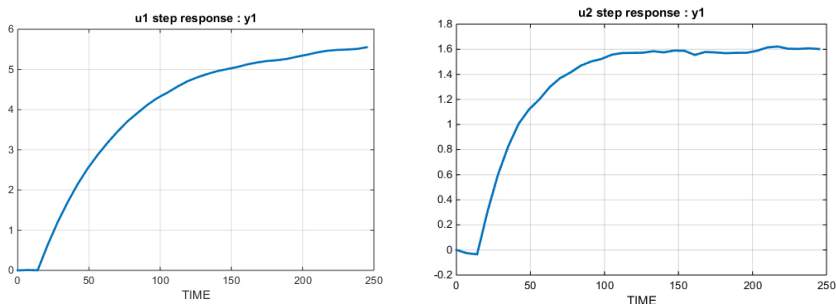
## 2. Опитни данни и резултати.

С цел изпълнение на поставените задачи, са създадени двете програми `mlrex.m` `plrex.m` в MATLAB®, представени съответно в Табл. 1.1 и Табл. 1.2. Програмата `mlrex.m` използва обобщеният метод на най-малките квадрати (ОМНМК), докато `plrex.m` извършва идентификация по частния метод на най-малките квадрати (ЧМНМК). В Табл. 1.3 са представени получените оценки на двата на обекта, получени по двата метода за идентификация.

На Фиг. 1.2 са показани изменението на истинската стойност на изхода на обекта и тази на предсказаната, получена по обобщения метод на най-малките квадрати. Освен това е изобразена и грешката от предсказване, както се вижда тя не превишава границата от 0.3 относителни единици. Преходите характеристики на двата канала, получена по обобщения МНМК са представени на Фиг. 1.3.



Фиг. 1.2. Истинска стойност и предсказана стойност на изхода на обекта. Грешка от предсказване



Фиг. 1.3. Преходна характеристика на обекта по двата канала получена по обобщения метод на най-малките квадрати (Multiple Linear Regression)

Табл. 1.1 Обобщеният метод на най-малките квадрати *Mlrex.m*

```

=====
% Copyright 1994-2002 The MathWorks, Inc. $Revision: 1.10 $
echo on
% Example routine for mlr
% Purpose: Demonstrate the use of identification routines.
% The system considered here has two inputs and one output.
%
% Load the input & output data. The input and output data are generated
% from the following transfer functions and random zero-mean noises.
% TF from input 1 to output 1:  $g_{11} = 5.72\exp(-14s)/(60s+1)$ 
% TF from input 2 to output 2:  $g_{21} = 1.52\exp(-15s)/(25s+1)$ 
% Sampling time of 7 minutes was used.
%
load mlrdat.mat
%
% Determine the standard deviations for input
% data using the function autosc.
%
[ax,mx,stdx] = autosc(x);
%
% Let us scale the input data by their standard deviations only.
%
mx = [0 0];
sx = scal(x,mx,stdx);

```

```

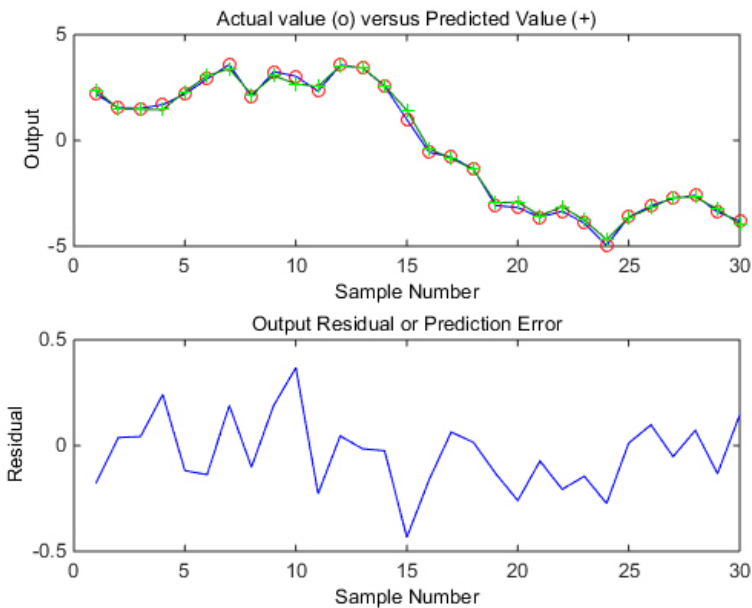
pause
%
% Put the input & output data in a form such that they can be used to
% determine the impulse response coefficients. 35 impulse response
% coefficients (n) are used.
%
n = 35;
[xreg,yreg] = wrtreg(sx,y,n);
%
% Determine the impulse response coefficients via mlr. No penalties on
% theta and delt theta are used in mlr. By specifying plotopt of 2,
% two plots - plot of predicted output and actual output, and plot of
% the residue output (or predicted error) - are produced.
%
ninput = 2;
plotopt = 2;
[theta,yres] = mlr(xreg,yreg,ninput,plotopt);
%
% Scale theta based on the standard deviations used in scaling the input.
%
theta = scal(theta,mx,stdx)
pause
%
% Convert the impulse model to a step model to be used in MPC design.
% Sampling time of 7 minutes was used in determining the impulse model.
% Number of outputs (1 in this case) must be specified.
%
nout = 1;
delt = 7;
model = imp2step(delt,nout,theta);
%
% Plot the step response coefficients.
%
plotstep(model)
echo off
=====

```

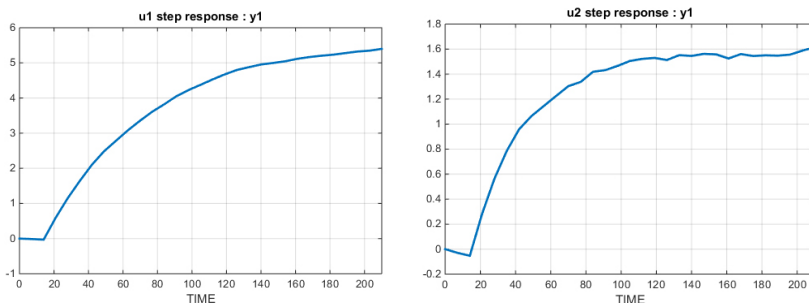
Табл. 1.2 Частен метод на най-малките квадрати Plrex.m

```
=====  
% Example routine for plsr  
%  
% Purpose: Demonstrates the use of identification routines.  
% Load the input and output data.  
%  
load plsrdat;  
  
% Put the input and output  
% in a form such that they can be used to determine the impulse  
% response coefficients. 35 impulse response coefficients are used.  
%  
n = 30;  
[xreg,yreg] = wrtreg(x,y,n);  
%  
% Determine the impulse response coefficients via plsr using 10 latent  
% variables. By specifying plotopt=2, two plots - plot of predicted output  
% and actual output, and plot of the residue output (or predicted error) -  
% are produced.  
%  
pause  
ninput = 2;  
lv = 10;  
plotopt = 2;  
theta = plsr(xreg,yreg,ninput,lv,plotopt)  
%  
% Use a new set of data to validate the impulse model.  
%  
[newxreg,newyreg] = wrtreg(newx,newy,n);  
yres = validmod(newxreg,newyreg,theta,plotopt);  
%  
% Convert the impulse model to a step model to be used in MPC design.  
% Sampling time of 7 minutes was used in determining the impulse model.  
% Number of outputs (1 in this case) must be specified.  
  
nout = 1;  
delt = 7;  
model = imp2step(delt,nout,theta);  
  
% Plot the step response.  
plotstep(model)  
echo off  
=====
```

Изменението на истинската стойност на изхода на обекта е сравнено с полученият модел на изхода от направената идентификация по частния ММК, като резултатът е изобразен на Фиг. 1.4. Грешката от предсказване в този случай, има максимална стойност от 0.4 относителни единици. Следователно при частния метод на МНМК отклонението от реалната стойност на изхода или грешката от предсказване е малко по-голяма от тази, получена при ОММК. Преходите характеристики на двата канала, получена по частния метод на най-малките квадрати са представени на Фиг. 1.5.



Фиг. 1.4 Истинска стойност и предсказана стойност на изхода на обекта. Грешка от предсказване



Фиг. 1.5 Преходна характеристика на обекта по двата канала получена по частния метод на най-малките квадрати (Partial least-squares regression)

Табл. 1.3 Разпечатване на получените оценки на изходите на двата канала

„theta” при обобщен МНМК	„theta” при частен МНМК
0.0083 -0.0254	-0.0132 -0.0297
-0.0061 -0.0091	-0.0172 -0.0229
0.6262 0.3408	0.6253 0.3368
0.5670 0.2919	0.5575 0.2797
0.4976 0.2277	0.4913 0.2224
0.4456 0.1792	0.4586 0.1726
0.3917 0.1145	0.3805 0.1051
0.3374 0.0806	0.3039 0.0808
0.3053 0.0982	0.3002 0.0802
0.2783 0.0711	0.2716 0.0791
0.2554 0.0453	0.2530 0.0334
0.2073 0.0541	0.2122 0.0813
0.2047 0.0339	0.2288 0.0131
0.1698 0.0200	0.1745 0.0340
0.1333 0.0342	0.1495 0.0393
0.1481 0.0129	0.1538 0.0167
0.1337 0.0010	0.1421 0.0077
0.0991 0.0017	0.1255 -0.0167
0.0824 0.0117	0.0796 0.0388
0.0737 -0.0088	0.0766 -0.0061
0.0500 0.0139	0.0426 0.0163

<b>„theta” при обобщен МНМК</b>	<b>„theta” при частен МНМК</b>
0.0510 -0.0016	0.0480 -0.0041
0.0649 -0.0329	0.0697 -0.0328
0.0480 0.0234	0.0509 0.0352
0.0377 -0.0034	0.0376 -0.0152
0.0192 -0.0063	0.0307 0.0055
0.0333 0.0029	0.0448 -0.0028
0.0545 0.0004	0.0442 0.0082
0.0487 0.0159	0.0253 0.0323
0.0562 0.0266	0.0505 0.0266
0.0421 0.0075	
0.0211 -0.0176	
0.0094 -0.0017	
0.0155 0.0045	
0.0416 -0.0050	

## ТЕМА 2

### Получаване на модели за целите на моделното предсказващо управление

#### 1. Теоретична част.

Целта на лабораторното упражнение е запознаване на студентите с:

- основните функции на MPC Toolbox™ за проектиране на системи с моделно предсказващо управление;
- формите за представяне на модела на обекта и функциите за преобразуване;
- командите за проектиране на предсказващ регулатор и симулиране на система с МПУ в програмната среда на MATLAB®.

#### 1. 1. Основни формати за представяне на модели и функции за преобразуването им в MPC Toolbox™ на MATLAB®.

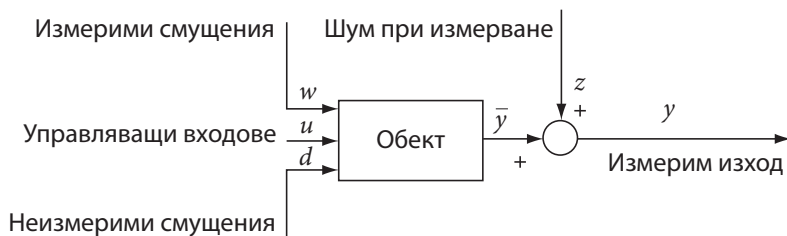
**Модели в пространство на състоянията.** Дискретното описание на обекта от Фиг. 2.1 в пространство на състоянието в MPC Toolbox™ има вида [7]:

$$\begin{aligned}x(k+1) &= \Phi x(k) + \Gamma_u u(k) + \Phi_w w(k) \\y(k) &= \bar{y}(k) + x(k) = \\&= \underbrace{Cx(k) + D_u u(k) + D_d d(k) + D_w w(k)}_{\bar{y}(k)} + z(k)\end{aligned}\tag{2.1}$$

където  $x$  е вектор на променливите на състоянието с размерност  $n \times 1$ ,  $u$  представлява векторът на управляващите въздействия с размерност  $1 \times n_u$ ,  $d$  представлява измеримите входни смущения ( $1 \times n_d$ ),  $w$  представлява неизмеримите смущение ( $1 \times n_w$ ), а  $y$  е вектор с  $n_y$ -те изхода на обекта,  $z$  е измерим шум;  $\Phi$ ,  $\Gamma_u$  и др. са постоянни матрици с определена големина. Променливата  $\bar{y}(k)$  представлява изхода на обекта преди добавянето на измеримия шум. В MATLAB® матриците се инициализират като:

$$\Gamma = \begin{bmatrix} \Gamma_u & \Gamma_d & \Gamma_w \end{bmatrix} \quad D = \begin{bmatrix} D_u & D_d & D_w \end{bmatrix}$$





Фиг. 2.1 Обект на управление в система с МПУ

**MPC mod формат.** MPC Toolbox™ работи със специален формат модели в пространството на състоянието, наречени **mod формат**. Този формат представлява матрица, която съдържа матриците от описанието в пространство на състоянието  $\Phi$ ,  $\Gamma$ ,  $C$  и  $D$ , и още допълнителна информация под формата на групирани елементи. Нека матрицата  $M$  представя горната система в общен **mod формат**. Тогава нейният размер е:

- брой редове =  $n + n_y + 1$ , където  $n_y = n_{yn} + n_{yu}$
- брой колони =  $\max(7, 1 + n + n_u + n_d + n_w)$

Векторът **minfo** представлява първите седем елемента от първия ред на матрицата  $M$ . Елементите на **minfo** са:

- minfo(1)  $T$ , стъпката на дискретизация, използвана за създаване на модела.
- minfo(2)  $n$ , брой състояния.
- minfo(3)  $n_u$ , брой управляващи въздействия.
- minfo(4)  $n_d$ , брой измерими смущения.
- minfo(5)  $n_w$ , брой неизмерими смущения.
- minfo(6)  $n_{yn}$ , брой измервани изходи.
- minfo(7)  $n_{yu}$ , брой неизмервани изходи.

Останалата част от матрицата  $M$  съдържа дискретните матрици от пространството на състоянието:

$\Phi$  в редове 2 до  $n + 1$  в колони 2 до  $n + 1$

$\Gamma$  в редове 2 до  $n + 1$  в колони  $n + 2$  до  $n + n_u + n_d + n_w + 1$

$C$  в редове  $n + 2$  до  $n + n_y + 1$  в колони 2 до  $n + 1$

$D$  в редове  $n + 2$  до  $n + n_y + 1$  в колони  $n + 2$  до  $n + n_u + n_d + n_w + 1$

Тъй като векторът **minfo** изисква седем колони, това е минималният възможен брой на колоните в **mod** формат, независимо от размерите на матриците от описанието в пространството на състоянието (2.1).

**Пример:** Матрица представяща системата от Фиг. 2.1, за която:

$$n = 2; n_u = 1; n_{yn} = 1; n_d = 0; n_w = 0; n_{yu} = 0.$$

$$M = \begin{array}{c|cccccc} & T & n & n_u & n_d & n_w & n_{yn} & n_{yu} \\ \hline NaN & & & & & 0 & 0 & 0 \\ \hline 0 & & \Phi & & \Gamma & 0 & 0 & 0 \\ \hline 0 & & C & & D & 0 & 0 & 0 \end{array}$$

Освен това, първата колона е резервирана за други цели от MPC Toolbox™. Ето защо матриците от описанието в пространството на състоянието започват от втората колона, както беше описано по-горе. **За да може да разпознава MPC Toolbox™ матриците в MPC mod формат, (2.1) елементът е винаги NaN (Not-a-Number).**

Табл. 2.1. Преобразуване на модели

<p>poly2tfd</p>	<p>Преобразува предавателна функция (непрекъсната или дискретна) от стандартния формат на MATLAB® във <b>MPC tf</b> формат</p> <p><b><math>g = \text{poly2tfd}(\text{num}, \text{den}, \text{delta}, \text{delay})</math></b></p> $G(s) = \frac{b_0 s^n + b_1 s^{n-1} + \dots + b_n}{a_0 s^n + a_1 s^{n-1} + \dots + a_n} \quad G(z^{-1}) = \frac{b_0 + b_1 z^{-1} + \dots + b_n z^{-n}}{a_0 + a_1 z^{-1} + \dots + a_n z^{-n}}$ <ul style="list-style-type: none"> <li>• <b>delta</b> – стъпка на дискретизация</li> <li>• <b>delay</b> – закъснение в системата</li> </ul>
<p>tfd2step</p>	<p>Преобразува модел от формат <b>MPC tf</b> във формат <b>MPC step</b></p> <p><b><math>\text{plant} = \text{tfd2step}(\text{tfinal}, \text{delta}, \text{nout}, \text{g1}, \dots, \text{g25})</math></b></p> <ul style="list-style-type: none"> <li>• <b>tfinal</b> – времето за получаване на преходната характеристика за <b>MPC step</b> модела</li> <li>• <b>delta</b> – стъпка на дискретизация</li> <li>• <b>nout</b> – брой на изходите в системата</li> <li>• <b>g1,g2,g3,...,g25</b> – модели в <b>MPC tf</b> формат</li> </ul>
<p>mod2ss</p>	<p>Преобразува модел от формат <b>MPC mod</b> в модел в пространството на състоянията (<b>ss</b>)</p> <p><b><math>[\text{phi}, \text{gam}, \text{c}, \text{d}] = \text{mod2ss}(\text{mod})</math></b></p> $x(k+1) = \Phi x(k) + \Gamma_u u(k) + \Gamma_d d(k) + \Gamma_w w(k)$ $y(k) = \bar{y}(k) + z(k) = Cx(k) + D_u u(k) + D_d d(k) + D_w w(k) + z(k)$ <p>където <math>d = D = [D_u \ D_d \ D_w]</math>,  <math>\text{gam} = \Gamma = [\Gamma_u \ \Gamma_d \ \Gamma_w]</math></p>

<p>tfd2mod</p>	<p>Преобразува модел от формат <b>MPC tf</b> във формат <b>MPC mod</b></p> <p><b><i>model = tfd2mod(delta,ny,g1,g2,g3,...,g25)</i></b></p> <ul style="list-style-type: none"> <li>• <b>delta</b> – стъпка на дискретизация</li> <li>• <b>ny</b> – брой на изходите в системата</li> <li>• <b>g1,g2,g3,...,g25</b> – модели в <b>MPC tf</b> формат</li> </ul>
<p>ss2mod</p>	<p>Преобразува модел в пространството на състоянието <b>ss</b> във формат <b>MPC mod</b></p> <p><b><i>pmod = ss2mod(phi,gam,c,d)</i></b></p> <ul style="list-style-type: none"> <li>• <b>phi, gam, c, d</b> – матрици от описанието в пространството на състоянията</li> </ul>
<p>mod2mod</p>	<p>Променя стъпката на дискретизация на модел във формат <b>MPC mod</b></p> <p><b><i>newmod = mod2mod(oldmod, delta)</i></b></p> <ul style="list-style-type: none"> <li>• <b>oldmod</b> – модел в MPC mod формат,</li> <li>• <b>delta</b> – новата стъпка на дискретизация</li> </ul>
<p>ss2step</p>	<p>Преобразува модел в пространството на състоянията в <b>MPC step</b> формат</p> <p><b><i>plant = ss2step(phi,gam,c,d,tfinal)</i></b></p> <ul style="list-style-type: none"> <li>• <b>phi, gam, c, d</b> – матрици от описанието в пространството на състоянието</li> <li>• <b>tfinal</b> – времето за получаване на преходната характеристика за <b>MPC step</b> модела</li> </ul>

mod2step	<p>Преобразува модел представен с <b>MPC mod</b> формат в <b>MPC step</b> формат</p> <p><b><i>[plant,dplant] = mod2step(mod,tfinal,delta,nout)</i></b></p> <ul style="list-style-type: none"> <li>• <b>mod</b> – модел в <b>MPC mod</b> формат</li> <li>• <b>tfinal</b> – времето за получаване на преходната характеристика за <b>MPC step</b> модела</li> <li>• <b>delta</b> – стъпка на дискретизация</li> <li>• <b>nout</b> – брой на изходите в системата</li> </ul>
----------	---

Табл. 2.2. Изчертаване и информация

mpcinfo	<p>Дава информация за система за управление, представена в <b>MPC mod</b> или <b>MPC step</b> формат</p> <p><b><i>mpcinfo(model)</i></b></p> <ul style="list-style-type: none"> <li>• <b>model</b> – модел на обекта в <b>MPC mod</b> или <b>MPC step</b> формат</li> </ul>
plotall	<p>Изчертава входовете и изходите, получени по време на симулация</p> <p><b><i>plotall(y, u, dt)</i></b></p> <ul style="list-style-type: none"> <li>• <b>y</b> – изход на системата</li> <li>• <b>u</b> – вход на системата</li> <li>• <b>dt</b> – такт на дискретизация; определя времето за симулация като крайната му стойност е <b><math>t=(\text{size}(y,1)-1)*dt</math></b></li> </ul>
plotstep	<p>Изчертава реакцията на <b>MPC step</b> модела</p> <p><b><i>plotstep(model)</i></b></p>

**1.2. Функции за проектиране на предсказващ регулатор и симулиране на система с МПУ в програмната среда на MATLAB® с използване на функции от MPC Toolbox™.**

Табл. 2.3. Проектиране на МПУ - MPC step формат

<p>mpc</p>	<p>Решава задача от квадратичното програмиране с цел симулиране на поведението на затворената система при наличие на ограничения за входа и изхода на системата.</p> <p><b><math>[y, u] = \text{mpc}(\text{plant}, \text{model}, \text{ywt}, \text{uwt}, M, P, \dots, \text{tend}, r, \text{ylim}, \text{ylim})</math></b></p> <ul style="list-style-type: none"> <li>• <b>plant</b> – модел на обекта в MPC step формат</li> <li>• <b>model</b> – модел в MPC step формат, отразяващ наблюдател на състоянието (<b>model = plant</b>, ако не е предвиден наблюдател)</li> <li>• <b>ywt, uwt</b> – тегловни матрици</li> <li>• <b>M</b> – хоризонт на управление</li> <li>• <b>P</b> – хоризонт на предсказване</li> <li>• <b>tend</b> – време на симулацията</li> <li>• <b>r</b> – задание</li> <li>• <b>ylim, ylim</b> – ограничения наложени на изходния и входния сигнал</li> </ul>
<p>mpccon</p>	<p>Изчислява матрицата на управлението за целите на МПУ (при отсъствие на ограничения)</p> <p><b><math>K_{\text{mpc}} = \text{mpccon}(\text{model}, \text{ywt}, \text{uwt}, M, P)</math></b></p> <ul style="list-style-type: none"> <li>• <b>model</b> – модел в MPC step формат</li> <li>• <b>ywt, uwt</b> – тегловни коефициенти/матрици</li> <li>• <b>M</b> – хоризонт на управление</li> <li>• <b>P</b> – хоризонт на предсказване</li> </ul>

<p>mpcsim</p>	<p>Симулира затворена система за МПУ с опционално ограничение от тип насищане за входния сигнал (manipulated variables)</p> <p><b><math>[y, u] = mpcsim(plant, model, Kmpc, tend, r, usat)</math></b></p> <ul style="list-style-type: none"> <li>• <b>plant</b> – модел на обекта в MPC <b>step</b> формат</li> <li>• <b>model</b> – модел в MPC <b>step</b> формат, отразяващ наблюдател на състоянието (<b>model = plant</b>, ако не е предвиден наблюдател)</li> <li>• <b>Kmpc</b> – управляващи въздействия</li> <li>• <b>tend</b> – време на симулацията</li> <li>• <b>r</b> – задание</li> <li>• <b>usat</b> – ограничение от тип насищане</li> </ul>
---------------	---

Табл. 2.4. Проектиране на МПУ - MPC mod формат

<p>scmpc</p>	<p>Решава задача от квадратичното програмиране с цел симулиране на поведението на затворената система при наличие на ограничения за входа и изхода на системата.</p> <p><b><math>[y, u] = scmpc(pmod, imod, ywt, uwt, M, P, ... tend, r, ulim, ylim)</math></b></p> <ul style="list-style-type: none"> <li>• <b>pmod</b> – модел на обекта в MPC <b>mod</b> формат</li> <li>• <b>imod</b> – модел в MPC <b>mod</b> формат, отразяващ наблюдател на състоянието (<b>pmod = imod</b>, ако не е предвиден наблюдател)</li> <li>• <b>ywt, uwt</b> – тегловни матрици</li> <li>• <b>M</b> – хоризонт на управление</li> <li>• <b>P</b> – хоризонт на предсказване</li> <li>• <b>tend</b> – време на симулацията</li> <li>• <b>r</b> – задание</li> <li>• <b>ylim, ulim</b> – ограничения наложени на изходния и входния сигнал</li> </ul>
--------------	--

<p>smpcccon</p>	<p>Изчислява матрицата на управлението за целите на МПУ (при отсъствие на ограничения)</p> <p><b><math>K_{mpc} = smpcccon(pmod, ywt, uwt, M, P)</math></b></p> <ul style="list-style-type: none"> <li>• <b>pmod</b> – модел в MPC <b>mod</b> формат</li> <li>• <b>ywt, uwt</b> – тегловни коефициенти/матрици</li> <li>• <b>M</b> – хоризонт на управление</li> <li>• <b>P</b> – хоризонт на предсказване</li> </ul>
<p>smpcsim</p>	<p>Симулира затворена система за МПУ с опционално ограничение от тип насищане за входния сигнал (manipulated variables)</p> <p><b><math>[y, u] = smpcsim(pmod, imod, K_{mpc}, tend, r, usat)</math></b></p> <ul style="list-style-type: none"> <li>• <b>pmod</b> – модел на обекта в MPC <b>mod</b> формат</li> <li>• <b>imod</b> – модел в MPC <b>mod</b> формат, отразяващ наблюдател на състоянието (<b>pmod=imod</b>, ако не е предвиден наблюдател)</li> <li>• <b>K<sub>mpc</sub></b> – управляващи въздействия</li> <li>• <b>tend</b> – време на симулацията</li> <li>• <b>r</b> – задание</li> <li>• <b>usat</b> – ограничение от тип насищане</li> </ul>

## 2. Задачи за изпълнение.

**Задача 1.** Запознаване с формите за представяне на модела на обекта в системи с МПУ и функциите на преобразуването му, предлагани от MPC Toolbox™.

**Задача 2.** Запознаване с функциите за проектиране на МПР и симулиране на система с моделно предсказващо управление.

**Задача 3.** Решаване на симулационни примери.



3. 1. Да се моделира обект от тип SISO, зададен чрез (2.2)

$$y(k) = 0.5y(k-1) + u(k-3) \quad (2.2)$$

и да се представи в двата формата - MPC mod и MPC step, чрез командите за преобразуване (Табл. 2.1).

3. 2. Да се проектира МПУ без ограничения със следните параметри: хоризонт на управление  $N_u = 2$ , хоризонт на предсказване  $N_2 = 6$ , задание  $r = 3$ ; да се използва MPC step формат на модела от задача 3.1.

### 3. Опитни данни и резултати.

Изпълнението на задачите свързани с проектирането на моделен предсказващ регулатор в средата на MATLAB® с помощта на MPC Toolbox™ е осъществена с програмата представена в Табл. 2.5. Като първа стъпка при решаване на Задача 3 е необходимо моделът на обекта, зададен в (2.2), да бъде представен чрез еквивалентната му дискретна предавателна функция (2.3)

$$g(z) = \frac{z^{-3}}{1 + 0.5z^{-1}} \quad (2.3)$$

Именно от запис (2.3) могат да бъдат определени параметрите **num** и **den**, които се използват при създаването на изходния модел във вид на дискретна предавателна функция. Веднъж създаден, моделът в **tfd** формат, може да бъде използван за създаването на модели в **MPC mod** и **MPC step** формати (виж Табл. 2.1).

Важен момент при задаването на параметрите **num** и **den** (коефициентите в числителя и знаменателя на предавателната функция) е коректното определяне на стойността на чистото закъснение в системата. За конкретния случай закъснението е 2 итерации ( $\text{delay} = 2$ ). Това е видно и в двата записа, които описват модела на обекта ((2.2) и (2.3)), като се има в предвид, че общата форма на модела (7).

Табл. 2.5 Проектиране на моделно предсказващ регулатор с  
команди от MPC Toolbox™

```
=====
% Example 2.1
%Model Predictive Control - Model Conversion

clc
clear all

% plant - defining the discrete transfer function

num=[1];
den=[1 0.5];
delt=0.1;
delay=2;
g=poly2tfd(num, den, delt, delay);

tfinal=2;
delt2=0.1;
nout=1; % number of outputs

% Step response

plant=tf2step(tfinal, delt2, nout, g);
figure(1); clf; plotstep(plant);

% Model in MPC mod format

mod1=tf2mod(delt2, nout, g);

%MPC mod to MPC step

plant_step=mod2step(mod1, tfinal, delt2, nout); % in order to plot the plant
figure(2); clf;
plotstep(plant_step);

% Controller

Nu=2; N2=6; ywt=1; uwt=0.1;
KmpcMod=smpccon(mod1, ywt, uwt, Nu, N2); %state space
KmpcStep=mpccon(plant, ywt, uwt, Nu, N2); %step response

% Simulation of the MPC with model presented by MPC step format
```

```
[yp, u, ym]=mpcsim(plant, plant, KmpcStep, tfinal, 3);
figure(3); clf;
plotall(yp, u, delt2);
```

% Simulation of the MPC with model presented by MPC mod format

```
[yp1, u1, ym1]=smpcsim(mod1, mod1, KmpcMod, tfinal, 3);
figure(4); clf; plotall(yp1, u1, delt2);
```

Обектът се описва в MPC mod формат със следната матрица:

mod1 =	0.1	4	1	0	0	1	0
	NaN	-0.5	0	1	1	0	0
	0	1	0	0	0	0	0
	0	0	1	0	0	0	0
	0	0	0	1	0	0	0



Определете и коментирайте параметрите на системата, като използвате информационния вектор **minfo**. Проверете коректността на матриците от описанието в пространството на състоянието от матрицата *mod1*, чрез използване на подходяща трансформация на модела.

Матрицата на управлението на моделно-предсказващ регулатор, работещ с:

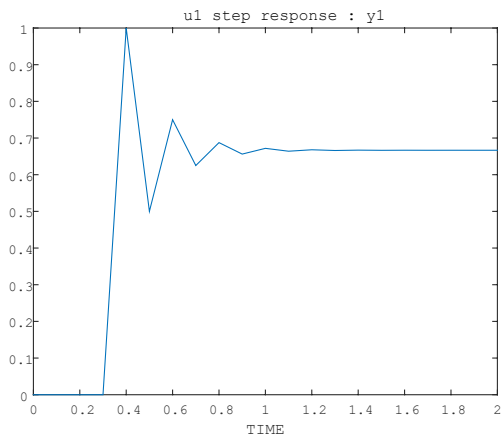
- **модел в mod формат има вида:**

```
KmpcMod = [0.9946 -0.4987 -0.9946 -0.9946 0 -0.9946]
```

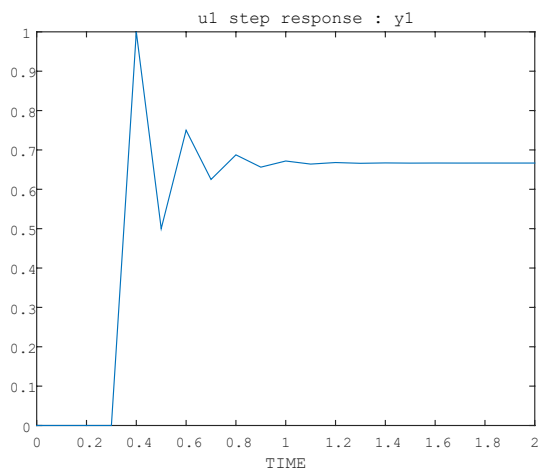
- **модел в MPC step формат (пребходна характеристика):**

```
KmpcStep = [2.9e-17 0 0 0.8231 -0.1601 0.3315]
```

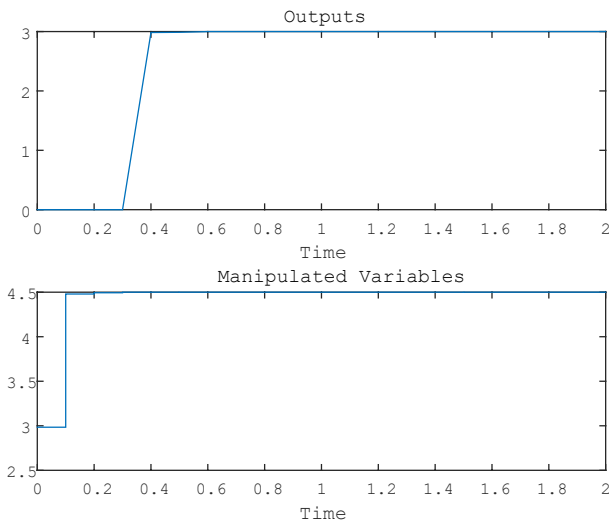
На Фиг. 2.1 и Фиг. 2.2 са показани преходните процеси на модела на обекта съответно получени от формат на представяне на обекта преходна характеристика и в mod формат. Както се вижда от графиките, те са еднакви



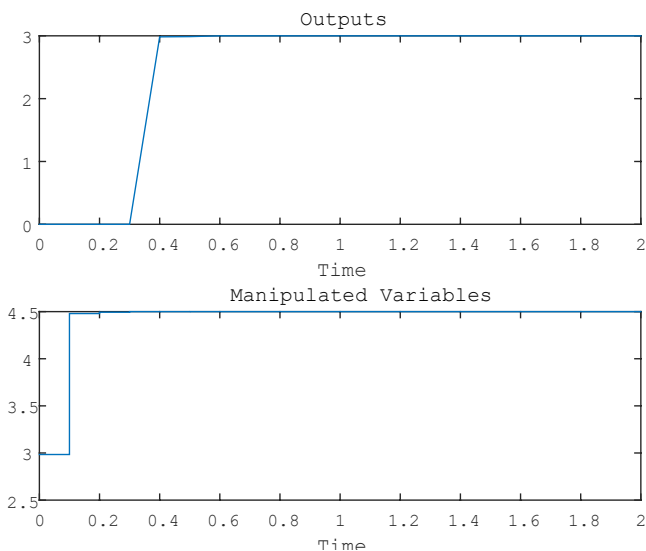
Фиг. 2.1 Преходна характеристика на обекта получена във формат MPC step



Фиг. 2.2 Преходна характеристика на обекта получена във формат MPC mod



*Фиг. 2.3 Реакция на изхода на системата и изменение на управлящото въздействие при работа на проектирания предсказващ регулатор с обекта получена във формат MPC step*



*Фиг. 2.4 Реакция на изхода на системата и изменение на управлящото въздействие при работа на проектирания предсказващ регулатор с обекта получена във формат MPC mod*



#### Допълнителни задачи:

**Задача 4.** Да се създадат модели в **MPC mod**, съответстващи на обекти зададени чрез следващите две описания:

$$x(k+1) = \begin{bmatrix} -0.7 & 0 \\ 0 & 0.8 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(k) + \begin{bmatrix} -1 \\ 0 \end{bmatrix} d(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w(k)$$
$$y(k) = \begin{bmatrix} 0.2 & -0.4 \end{bmatrix} x(k)$$

$$x(k+1) = -0.2x(k) + 1u(k) - 0.5d(k) + 0.2w(k)$$
$$y(k) = 3x(k)$$

## ТЕМА 3

### Комбиниране на предсказващи модели

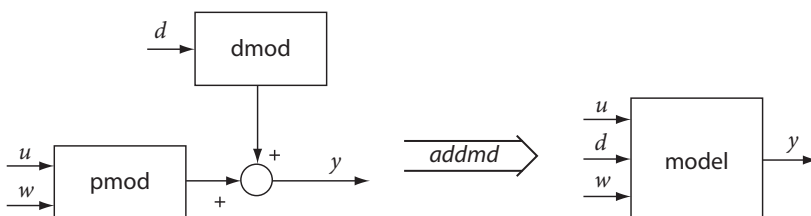
Целта на лабораторното упражнение е запознаване на студентите с възможностите за създаване на по-сложни модели на обекти за целите на МПУ, които MPC Toolbox™ предлага. Разглеждат се основни функции за комбиниране и обединение на примитиви на по-прости модели.

#### 1. Теоретична част.

**Комбиниране на модели в MPC mod формат.** Функциите **addmod**, **addmd**, **addumd** и **appmod**, позволяват комбинирането на прости модели в **mod** формат, с цел създаване на по-сложни структури, представляващи модела на обекта на управление в системи с МПУ. Всички междинни модели (примитиви) трябва да бъдат в **mod** формат, т.е. създадени, използвайки една от функциите **tfd2mod**, **ss2mod** или **th2mod**, описани в Тема 2.

##### 1.1. Функция **addmd**

Функцията **addmd** добавя ефекта от едно или повече измерими смущения  $d$  в съществуващ модел, както е показано на Фиг. 3.1.



Фиг. 3.1. Добавяне на едно измеримо смущение към съществуващ модел с функцията **addmd**

- **pmod** съдържа едно или повече управляващи въздействия -  $u(k)$ , а в някои случаи едно или повече неизмерими смущения -  $w(k)$  приложени към изхода/изходите  $y(k)$ .
- **dmod** съдържа влиянието на измеримото смущение  $d(k)$  върху

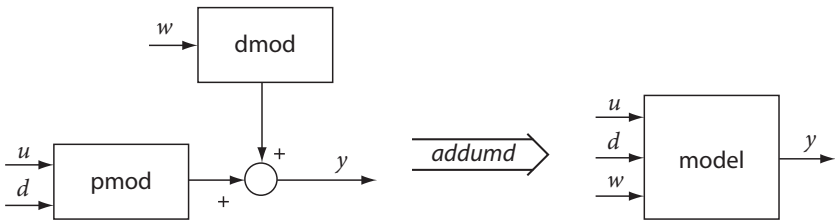
същите изходи.

Функцията има следния синтаксис:

**model = addmd (pmod, dmod)**

### 1. 2. Функция addumd

Функцията **addumd** добавя ефекта от едно или повече неизмерими смущения  $w$  в съществуващ модел, както е показано на Фиг. 3.2.



Фиг. 3.2. Добавяне на едно неизмеримо смущение към съществуващ модел с функцията *addumd*

- **pmod** съдържа едно или повече управляващи въздействия -  $u(k)$ , и в някои случаи едно или повече измерими смущения -  $d(k)$  приложени към изхода/изходите  $y(k)$ .
- **dmod** съдържа влиянието на неизмеримото смущение  $w(k)$  върху същите изходи.

Функцията има следния синтаксис:

**model = addumd (pmod, dmod)**

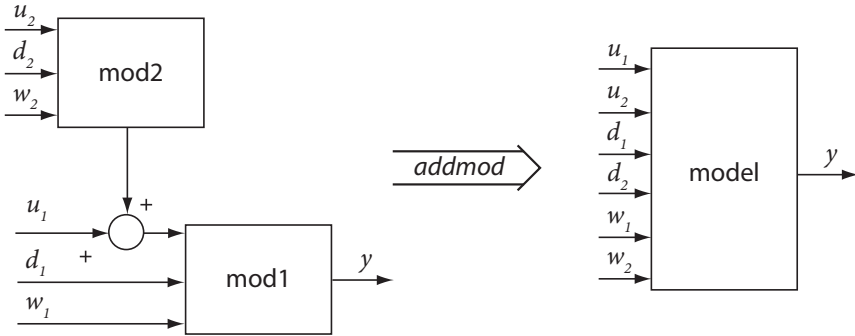
### 1. 3. Функция addmod

Тази функция обединява два модела от **MPC mod** формат. Използва се изключително рядко, като нейната главна цел е да състави модел със сложна структура от вида показан на Фиг. 3.3. Изходът/изходите на модела **mod2** се добавят към управляващото въздействие на модела **mod1**, за да се създаде обобщеният модел **model**.



Функцията има следния синтаксис:

**model = addmod (mod1, mod2)**

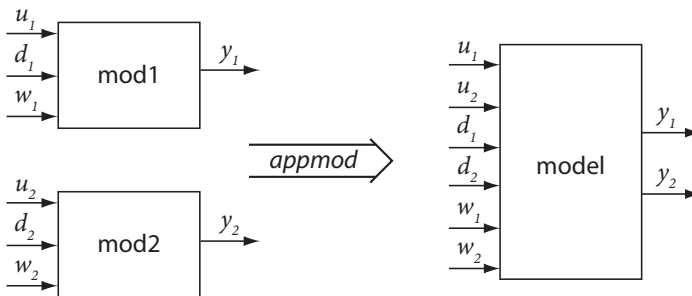


Фиг. 3.3. Комбиниране на два модела в MPC mod формат с функцията *addmod*

#### 1.4. Функция **appmod**

Функцията **appmod** обединява два модела в един общ. Двата модела се сливат както е показано на Фиг. 3.4. Функцията има следния синтаксис:

**model = appmod (mod1, mod2)**



Фиг. 3.4. Комбиниране на два модела *mod1* и *mod2* в модел *model* с функцията *appmod*

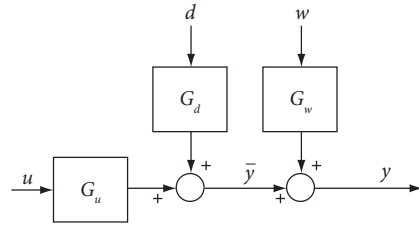
## 2. Задачи за изпълнение.

**Задача 1.** Да се построи модел по зададена структурна схема (Фиг. 3.5), като управляващото въздействие, измеримото и неизмеримото смущение са представени с предавателните функции  $G_u(z^{-1})$ ,  $G_d(z^{-1})$ ,  $G_w(z^{-1})$ :

$$G_u(z^{-1}) = \frac{0.7}{1 - 0.9z^{-1}}$$

$$G_d(z^{-1}) = \frac{-1.5}{1 - 0.85z^{-1}}$$

$$G_w(z^{-1}) = \frac{1}{1 + 0.6z^{-1}}$$



Фиг. 3.5. Структурна схема на модела

**Задача 2.** Да се комбинират следните модели зададени чрез:

$$x(k+1) = \begin{bmatrix} -0.7 & 0 \\ 0 & 0.8 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(k) + \begin{bmatrix} -1 \\ 0 \end{bmatrix} d(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w(k)$$

$$y(k) = \begin{bmatrix} 0.2 & -0.4 \end{bmatrix} x(k)$$

$$x(k+1) = -0.2x(k) + 1u(k) - 0.5d(k) + 0.2w(k)$$

$$y(k) = 3x(k)$$

## 3. Опитни данни и резултати.

**Изпълнение на Задача 1.** Един от начините за построяване на структурата от Фиг. 3.5 е предавателните функции на смущенията и управляващото въздействие да се представят в пространство на състоянието с командата **tf2ss** и след това да се премине в **mod** формат с функцията за преобразуване

**ss2mod.** Друг подход е да се получи дискретна предавателна функция посредством трансформацията **poly2tfd**, която в последствие да се представи в **mod** формат с командата **tfd2mod**. След получаване на трите примитива се прилагат последователно командите за обединение **addmd** и **addmod**.

Вече комбинираният модел на обекта се описва със следната матрица в **mod** формат:

```

modelUDW =      2   3   1   1   1   1   0
                NaN 0.9   0   0   1   0   0
                0   0   0.85 0   0   1   0
                0   0   0   -0.6 0   0   1
                0   0.7 -1.5  1   0   0   0
    
```

Табл. 3.1 Добавяне на смущения към модел с функциите *addmd* и *addmod*

---



---

```

%-----Model Predictive Control - Model Conversion-----
    
```

```

clc; clear all
    
```

```

%Plant Gd
numGd=[-1.5];
denGd=[1 -0.85];
    
```

```

%Plant Gu
numGu=[0.7];
denGu=[1 -0.9];
    
```

```

%Plant Gw
numGw=[1];
denGw=[1 0.6];
delt=2;
    
```

```

gd=poly2tfd(numGd,denGd,delt)
gu=poly2tfd(numGu,denGu,delt)
gw=poly2tfd(numGw,denGw,delt)
    
```

```

tfinal=150;
    
```

```

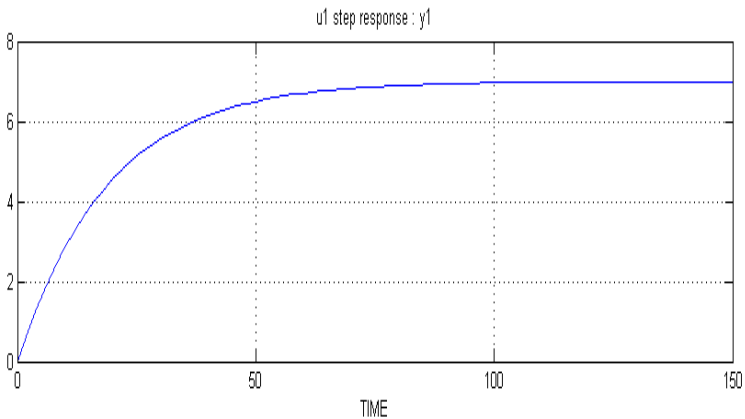
delt2=2;
nout=1;

modD=tf2mod(delt2,nout,gd)
modU=tf2mod(delt2,nout,gu)
modW=tf2mod(delt2,nout,gw)

modelUD=addmd(modU,modD)
modelUDW=addumd(modelUD,modW)
plant=mod2step(modelUDW,tfinal,delt2,nout)
figure(1); clf;
plotstep(plant);grid on;

```

---



Фиг. 3.6. Реакция на изхода на модела на обекта на управление, създаден с функциите *addmd* и *addumd*

**Изпълнение на Задача 2.** За да се комбинират двата модела на обекти представени в пространството на състоянието се създава програмата, представена в Табл. 3.2. Двата модела се описват в пространство на състоянието съгласно описанието (2.1), след това с функцията **ss2mod** се преобразуват в изисквания **mod** формат. Вече преобразувани в подходящ формат, двата модела се комбинират с функцията **addmod**. Матрицата в **mod** формат, получена след обединението на двата модела има следния вид:

modFF =	2	3	2	2	2	1	0	0	0	0
	NaN	-0.7	0	3	1	0.3	-1	0	0	0
	0	0	0.8	0	0	0	0	0	1	0
	0	0	0	-0.2	0	1	0	-0.5	0	0.3
	0	0.2	-0.4	0	0	0	0	0	0	0

Табл. 3.2 Комбиниране на два модела с функцията *addmod*

```

=====
clc; clear all
tfinal = 150;
delt=2;

%Plant 1
F1=[-0.7 0; 0 0.8];
Gu1=[1;0]; Gd1=[-1;0]; Gw1=[0;1];
GAM1=[Gu1 Gd1 Gw1];
C1=[0.2 -0.4]; Du1=[0 0 0];

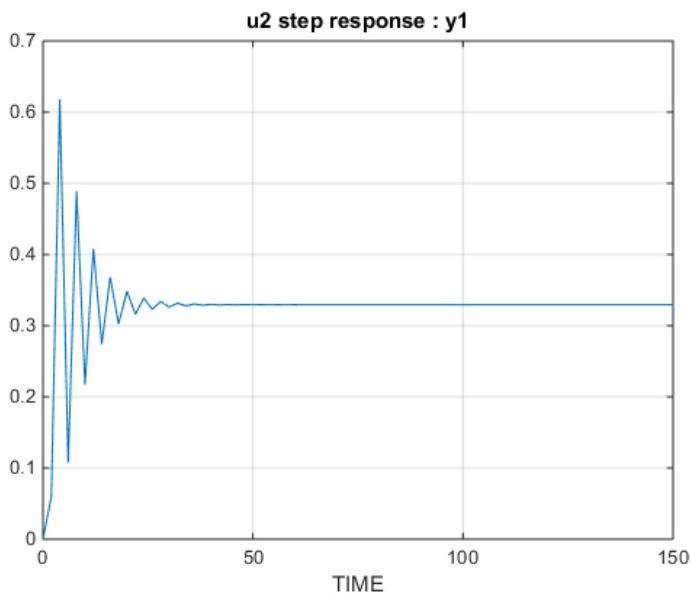
%Plant 2
F2=-0.2;
Gu2=1; Gd2=-0.5; Gw2=0.3;
GAM2=[Gu2 Gd2 Gw2];
C2=3; Du2=[0.3 0 0];

% State space to mpcmod
dt=2;n=2;nu=1;nd=1;nw=1;nym=1;nyu=0;
minfo1=[dt,n,nu,nd,nw,nym,nyu];
modF1=ss2mod(F1,GAM1,C1,Du1,minfo1);

dt2=2;n2=1;nu2=1;nd2=1;nw2=1;nym2=1;nyu2=0;
minfo2=[dt2,n2,nu2,nd2,nw2,nym2,nyu2];
modF2=ss2mod(F2,GAM2,C2,Du2,minfo2);

modFF=addmod(modF1,modF2)
plant=mod2step(modFF,tfinal,delt,nym)
figure(1); clf;
plotstep(plant); grid on;
=====

```



Фиг. 3. 7. Реакция на изхода на модела на обекта на управление, създаден с функцията *addmod*



Определете и коментирайте размерността (брой състояния, брой входове, брой изходи, измерими и неизмерими смущаващи въздействия) и параметрите на получените модели.

## ТЕМА 4

### Графична среда за проектиране и изследване на системи с моделно предсказващо управление

Целта на лабораторното упражнение е запознаване на студентите с графичната среда на MPC Toolbox™ за проектиране и изследване на системи с МПУ в средата на MATLAB®. Лабораторното упражнение е основа за всички следващи теми. То дава основни насоки за работа със специализирания графичен интерфейс на библиотека MPC Toolbox™.

#### 1. Теоретична постановка

Библиотека MPC Toolbox™ предлага голям набор от конфигурационни модули, които са представени в систематизирана графична среда. Тя улеснява както самото проектиране на системата с МПУ, така и нейното изследване чрез създаване на различни симулационни сценарии.

##### 1. 1. Конфигуриране на система с МПУ

Графичният интерфейс се отваря чрез въвеждане на следната команда в командния прозорец на MATLAB®:

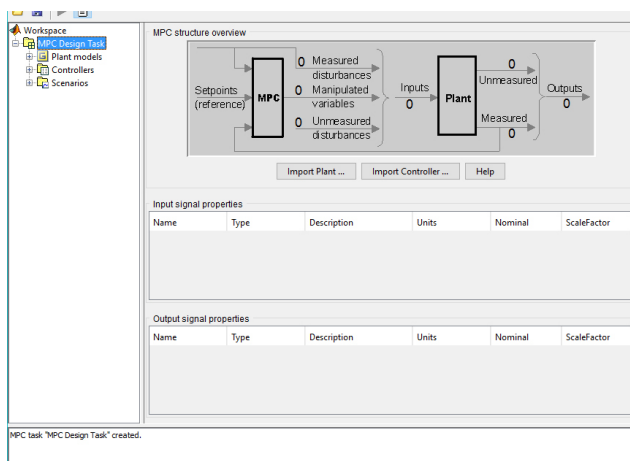
```
>> mpctool (за версии на MATLAB® преди R2015b)
```

```
>> mpcDesigner
```

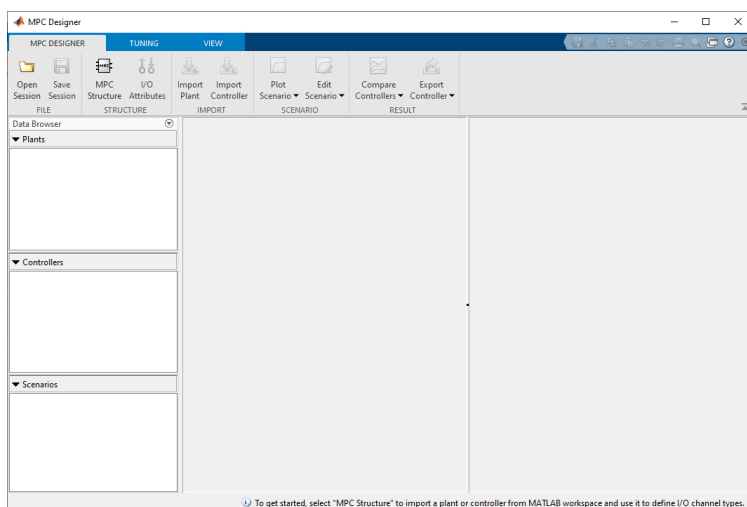
На Фиг. 4.1 е представен основния изглед на графичната среда. Той съдържа информация за размерността на модела, представен под формата на своеобразна структурна схема.

Проектирането на система с МПУ в графичната среда на библиотеката се състои от конфигуриране на модела, структурата и параметрите на предсказващия регулатор. Препоръчително е да се започне с конфигуриране на модела на обекта на управление. Той трябва да бъде предварително създаден по един от начините описани в Тема 2. Въвеждането му в графичната среда се

осъществява чрез бутона **Import Plant**<sup>1</sup> (или с избор на MPC-->Import Plant от менюто на основния прозорец), който отваря диалогов прозорец с две възможности за избор на местонахождението на вече създадения модел: определена директория, в която е съхранен MAT-файлът на модела или работното пространство на MATLAB® (MATLAB® Workspace) - Фиг. 4.2.



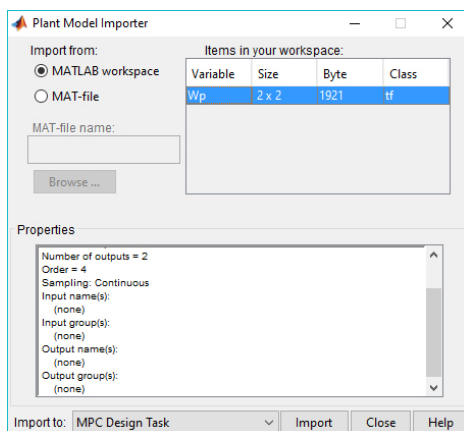
(а) при инструмента MPC Tool



(б) при инструмента MPC Designer

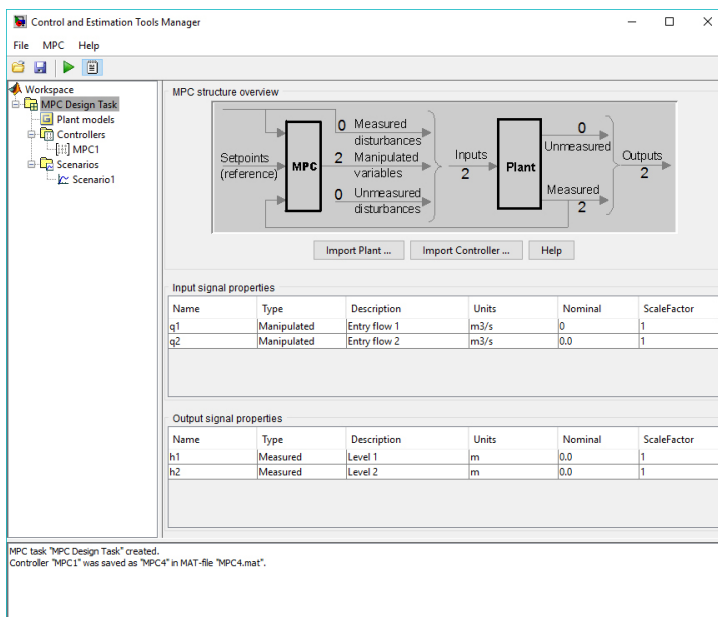
Фиг. 4.1 Графичен прозорец с въвеждане модел на обекта и описание на входно – изходните сигнали





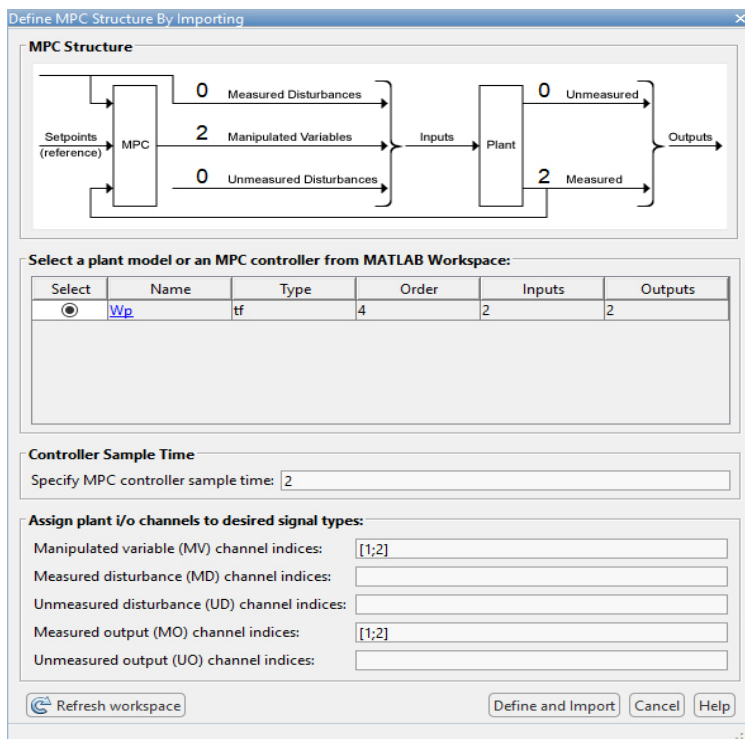
Фиг. 4.2 Графичен прозорец за зареждане на предварително създаден модел на обекта на управление на система с МПУ

Избраният модел се добавя с бутона **Import**. Диалоговият прозорец трябва да се затвори след въвеждане на желаните модели.



Фиг. 4.3 Графичен прозорец с конкретна информация за структурата на проектираната система с МПУ (пример с обект скачени резервоари)

При работа с версия на MATLAB® R2015b или по-висока въвеждането на модел на обекта се осъществява чрез бутона **MPC Structure** от **основния изглед (Фиг. 4.1а)**, който отваря диалогов прозорец (Фиг. 4.4) аналогичен на този от Фиг. 4.3.



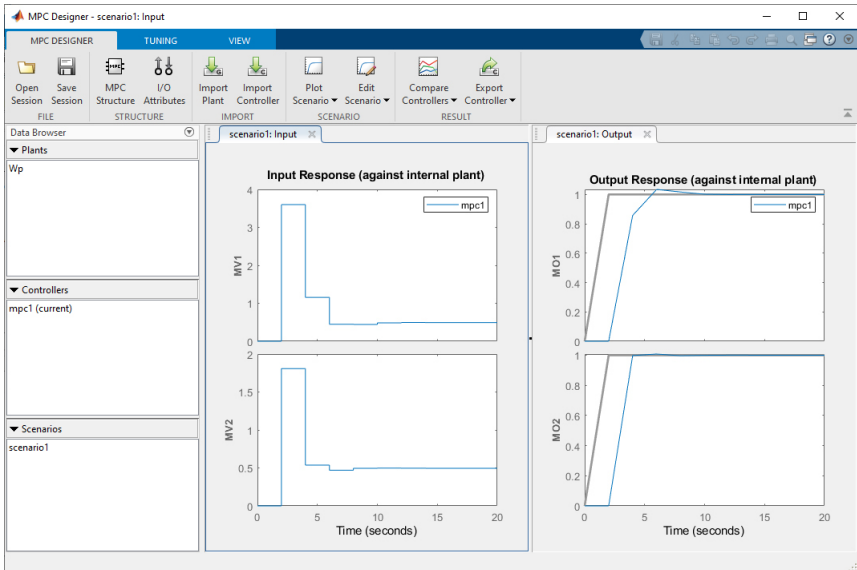
Фиг. 4.4 Графичен прозорец с възможност за избор на модел и структура на проектираната система с МПУ

Избраният модел се добавя с бутона **Define and Import**, а диалоговият прозорец трябва да се затвори след въвеждане на желаните модели. Резултатът се визуализира както е показано на следващата фигура (Фиг. 4.5).

Добавянето на модел променя автоматично изгледа на основния прозорец (Фиг. 4.3 и Фиг. 4.4). Появява се информация за размерността на проектираната система с МПУ: *брой входове (управляващи въздействия, измерими и неизмерими смущения), брой изходи (измерими и неизмерими).*

- **При работа с MPC Tool:** В секциите **Input signal properties** и **Output signal properties** се задават имената (полето **Description**), мерните единици (полето **Units**) и номиналните стойности (полето **Nominal**) за всички променливи в системата. Това улеснява значително по-нататъшната конфигурация на системата с МПУ. При добавяне на модели, които не са дефинирани в пространството на състоянието, средата автоматично ги преобразува в такива.
- **При работа с MPC Designer:** Добавянето на модел променя автоматично изгледа на основния прозорец (Фиг. 4.5). Появява се информация за размерността на проектираната система с МПУ: брой входове (управляващи въздействия, измерими и неизмерими смущения), брой изходи (измерими и неизмерими). От бутон **I/O Attributes** се отваря диалогов прозорец (Фиг. 4.6) съдържащ две полета: **Plant Inputs** и **Plant Outputs**, всяко от което съдържа различни полета. В тях можем да видим вида на сигнала (поле **Channel**), типа на променливата (поле **Type**), имената (полето **Name**), мерните единици (полето **Units**), номиналните стойности (полето **Nominal Value**) и скалата (поле **Scale Factor**) за всички променливи в системата. Това улеснява значително по-нататъшната конфигурация на системата с МПУ. При добавяне на модели, които не са дефинирани в пространството на състоянието, средата автоматично ги преобразува в такива.

От съществено значение е да се уточни вида на сигналите, преди да се продължи с проектирането на системата с МПУ. По подразбиране, ако видът на входните и изходните сигнали не е дефиниран в заредения модел, всички входове се възприемат като управляващи, а всички изходи се приемат за измерими. Видът на съответната променлива (вход или изход) може да бъде сменен от падащото меню в полето **Type**. Така за изходите (Outputs) може да се избере: **Measured** (измерим - MO), **Unmeasured** (неизмерим - UO), **Neglected** (не се взима под внимание - NO). За входовете опциите са: **Manipulated** (управляващо въздействие - MV), **Meas. disturb.** (измеримо смущение - MD), **Unmeas. disturb.** (неизмеримо смущение - UD), **Neglected** (не се взима под внимание).



Фиг. 4.5 Графичен прозорец с конкретна информация за структурата на проектираната система с МПУ (пример с обект скачени резервоари)

Input and Output Channel Specifications

Plant Inputs					
Channel	Type	Name	Unit	Nominal Value	Scale Factor
u(1)	MV	MV1		0	1
u(2)	MV	MV2		0	1

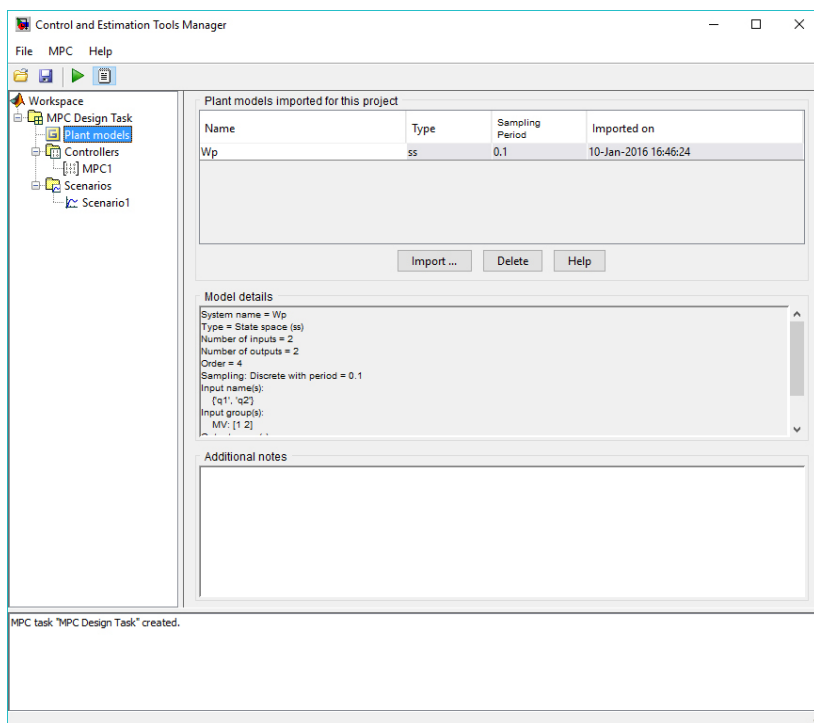
Plant Outputs					
Channel	Type	Name	Unit	Nominal Value	Scale Factor
y(1)	MO	MO1		0	1
y(2)	MO	MO2		0	1

OK Apply Cancel Help

Фиг. 4.6 Диалогов прозорец за I/O Attributes

### 1.1.1. Конфигуриране на параметрите на предсказващия регулатор при използване на инструмента MPC Tool

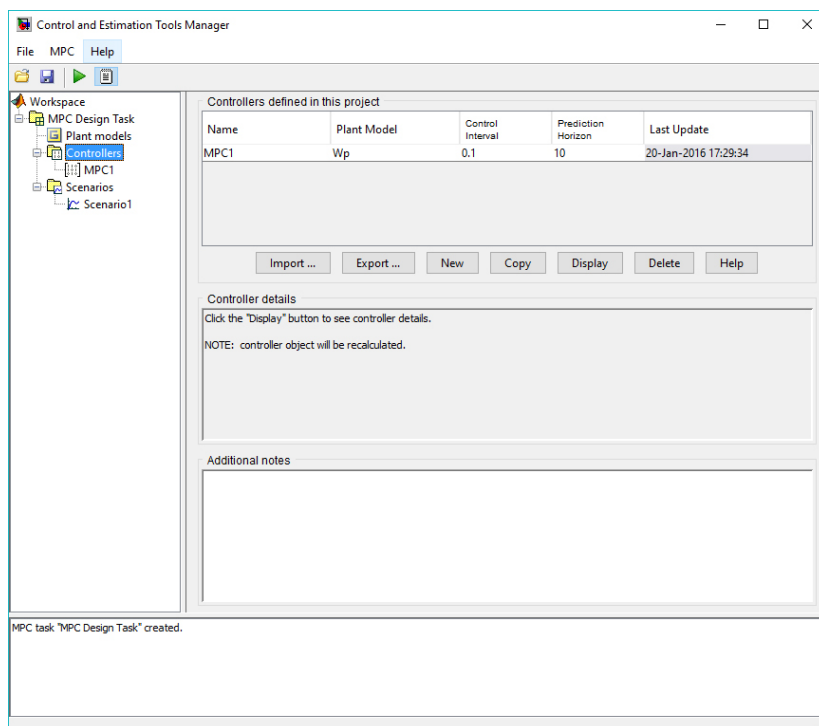
В левия навигационен панел на основния прозорец се намира дървото с директории/папки, принадлежащи на един или повече проекти, които са заредени в графичния интерфейс. Избирането (кликването) на дадена папка предизвиква появата на съответно меню в десния панел на прозореца. Когато проектният прозорец се стартира, той активира основна папка с наименование **MPC Design Task**, както е показано на Фиг. 4.3.



Фиг. 4.7 Графичен прозорец с конкретна информация за моделите в даден проект

Папката **Plant models** (модели на обекта на управление) съдържа всички заредени модели в проекта **MPC Design Task**. При маркиране (избор) на **Plant models**, в десния панел на проектния прозорец се появяват детайли за неговия тип, размерност, време на създаване и др. (Фиг. 4.7). Средата предоставя и възможност за добавяне на бележки. Бутоните **Import** и **Delete** позволяват да се зареди нов модел или изтрие някой, който вече не е необходим.

Следващата папка от дървото е **Controllers** (регулатори). Символът “+” вляво показва, че се съдържат подпапки (в случая това са регулатори) Фиг. 4.8. Кликването върху “+” разширява дървото (както е показано на Фиг. 4.8). Така всички регулатори в проекта стават видими за потребителя. По подразбиране при създаване на проекта регулаторът е един - MPC1. Списъкът на всички регулатори в проекта е видим при маркиране на папката **Controllers**, подобно на **Plant models**. В рамките на един проект могат да се създават и изследват неограничен брой предсказващи регулатори.



Фиг. 4.8 Графичен прозорец с конкретна информация за регулаторите в даден проект

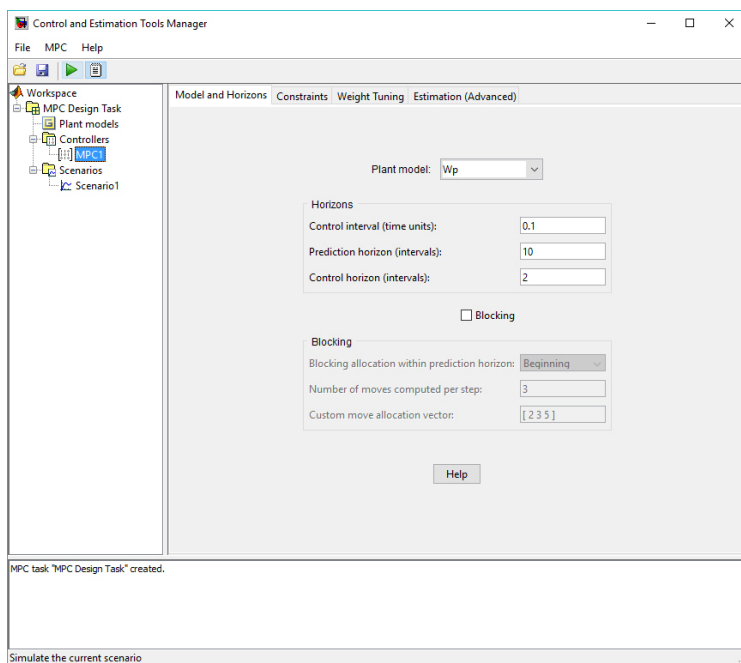
Бутоните в графичния прозорец дават възможност за:

- Импортиране (**Import**) на регулатор проектиран преди това и съхранен в работното пространство или в MAT-файл;
- Експортиране (**Export**) на избрания регулатор в работното прос-

транство на MATLAB® (съхранение под формата на MAT-файл в желана директория);

- Създаване на нов (**New**) регулатор, който ще се инициализира в MPC Toolbox™ по подразбиране;
- Копиране (**Copy**) на избрания регулатор, за да създадете дубликат, който след това да модифицира;
- Изтриване (**Delete**) на избрания регулатор.

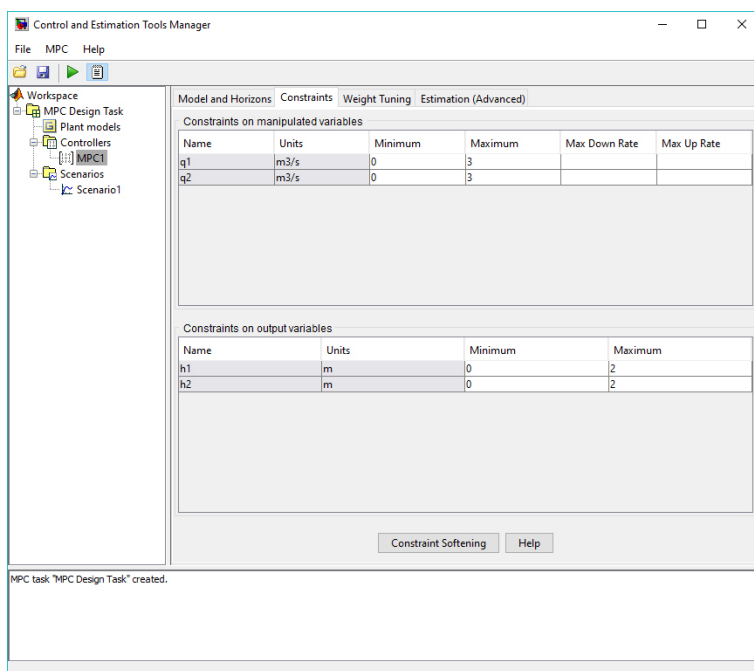
В отделни прозорци се показват настройките за конкретния предсказващ регулатор: модел на обекта, който се използва; периода на дискретизация на регулатора; хоризонта на предсказване и хоризонта на управление. Всички параметри имат възможност да бъдат променяни.



Фиг. 4.9 Прозорец с параметрите на регулатора – модел и хоризонти

След избиране на регулатор MPC1, главният панел трябва да се промени към изгледа, показан на Фиг. 4.9. Ако избраният **Prediction model** е непрекъснат Control Interval (период на дискретизация) по подразбиране е равен на 1 времева единица.

Разделът (табът) **Constraints** позволява задаване на ограниченията в системата с МПУ (Фиг. 4.10). Тук се въвеждат съответните гранични стойности за всяка променлива в проектираната система с МПУ. Добре е да се впишат всички известни ограничения на управляващите въздействия. Пропускането на ограниченията за управляваните променливи е допустимо. Полето **Max down rate** приема неположителна стойност. Тази настройка на предсказващия регулатор определя максималната стъпка на изменение на съответното управляващо въздействие в отрицателна посока. По аналогия, **Max up rate** определя максималната стъпка на нарастване на съответното управляващо въздействие. Тя трябва да е зададена като неотрицателна стойност. *При липса на ограничения полетата се оставят празни.*

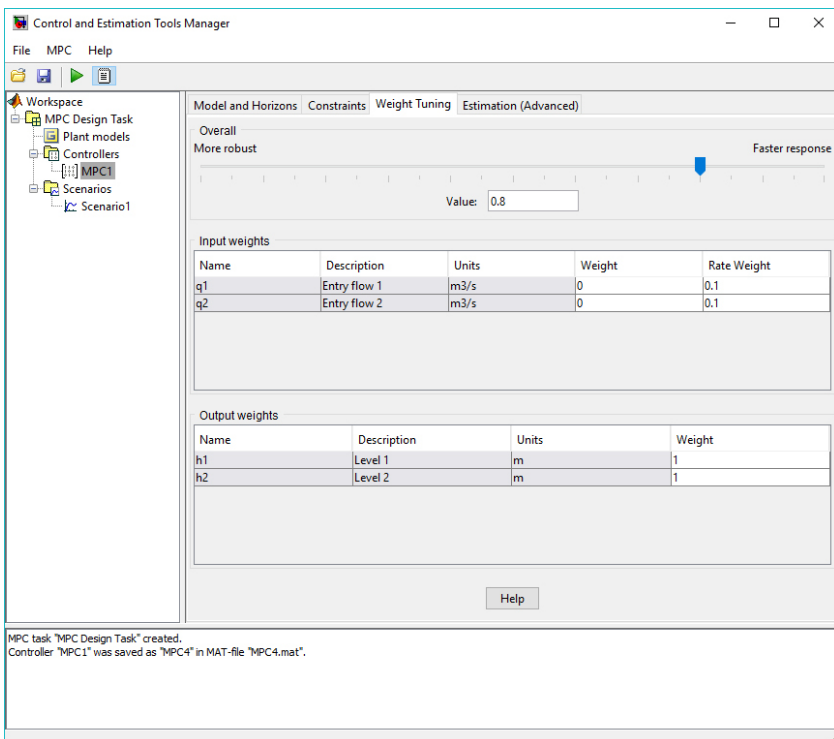


Фиг. 4.10 Прозорец за конфигуриране на ограниченията в системата с МПУ

Следващият раздел е **Weight Tuning** (настройка на тегловните фактори/коэффициенти) - Фиг. 4.11. Тегловните коефициенти определят поведението на предсказващия регулатор. Секцията **Output Weights** позволява задаване на тегловни коефициенти на изходите на системата с МПУ. Регу-



латорът ще се опита да сведе до минимум отклонението на всеки изход от неговото задание. За всеки такт на дискретизация в хоризонта на предсказване, регулаторът мултиплицира предсказаните отклонения за всеки изход с тегловния им коефициент, повдига на квадрат резултата, и сумира всички моментни стойности и всички изходи. Една от целите на регулатора е да се сведе до минимум тази сума, т.е., да се осигури добро следене на заданието (виж въвеждащата тема). По аналогия, секцията **Inputs Weights** позволява контрол над отклонението на управляващите въздействия от номиналните им стойности. Тук потребителят има възможност да влияе и върху скоростта на изменение на управляващите сигнали чрез полето **Rate Weight**. Този тегловен коефициент потиска резки промени в управлението. Стойността по подразбиране е 0.1.



Фиг. 4.11 Прозорец за настройка на тегловни коефициенти

Плъзгачът най-отгоре настройва цялостното съотношение между въздействието на регулатора и следенето на заданието. Преместването на плъзгача наляво оказва по-голямо общо въздействие на управляващите въздействия, което ги прави по-малки. Това обикновено увеличава робастността на регулатора, но отработването на заданието става по-бавно. Секцията **Estimation** позволява настройка на реакцията на регулатора при неизмерими смущения.

При определяне на тегловните коефициенти, относителните стойности са по-важни от абсолютните стойности. Трябва да бъдат отчетени разлики в измервания диапазон за всяка променлива. Например, ако отклонение от 0.1 единици за променливата А е също толкова важно, както отклонение от 100 единици за променливата В, то теглото на променливата А трябва да е 1000 пъти по-голямо от това за променливата В.

### **1. 1. 2. Конфигуриране на параметрите на предсказващия регулатор при използване на инструмента MPC Designer**

В левия навигационен панел на основния прозорец са разположени три полета, описващи дървото с директории. Полето **Plant** (модел на обекта на управление) съдържа всички заредени модели в проекта. При маркиране (избор) на **Plant**, в десния панел на проектния прозорец се отваря информационен прозорец с детайли за неговия тип, размерност, време на създаване и др. (Табл. 4.1).

Следващото поле е **Controllers** (регулатори). По подразбиране при създаване на проекта регулаторът е един - MPC1. Символът “▼” вляво показва, че се съдържат подпапки (в случая това са регулатори) Фиг. 4.5. Кликването върху “▼” разширява дървото. Така всички регулатори в проекта стават видими за потребителя. В рамките на един проект могат да се създават и изследват неограничен брой регулатори.

Табл. 4.1 Информационен прозорец за конкретен модел в полето Plant

Wp 2x2 tf

val =

From input 1 to output...

$$3 s + 1$$

1: -----

$$24 s^2 + 11 s + 0.75$$

$$0.5$$

2: -----

$$24 s^2 + 11 s + 0.75$$

From input 2 to output...

$$0.5$$

1: -----

$$24 s^2 + 11 s + 0.75$$

$$8 s + 1$$

2: -----

$$24 s^2 + 11 s + 0.75$$

Input groups:

Name	Channels
Manipulated	1,2

Output groups:

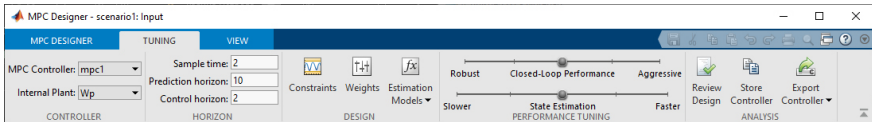
Name	Channels
Measured	1,2

Continuous-time transfer function.

Допълнителни настройки за предсказващия регулатор могат да бъдат използвани от главното поле **Tuning** (Фиг. 4.12) намиращо се в горният ляв ъгъл на прозореца.

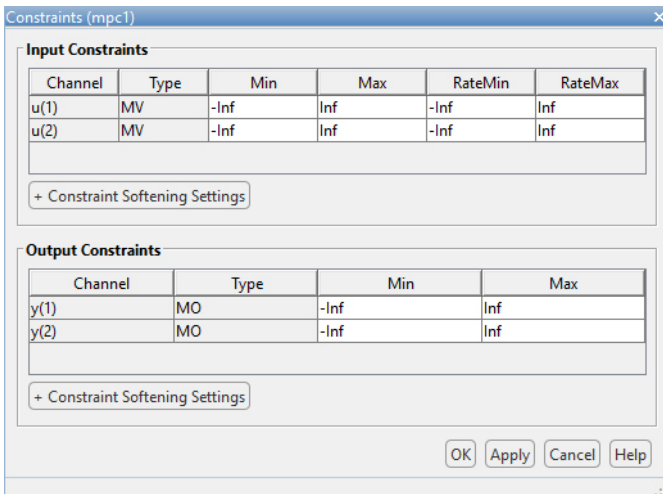
Полето предлага избор на:

- зареден модел;
- период на дискретизация на регулатора (Sample time);
- хоризонт на предсказване (Prediction horizon);
- хоризонт на управление (Control horizon).



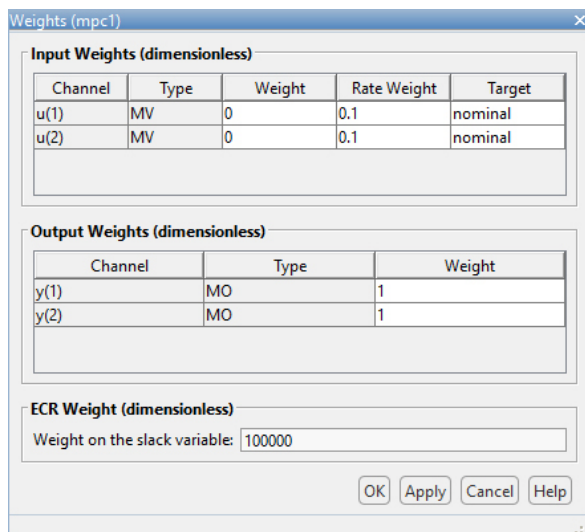
Фиг. 4.12 Параметри за настройка в полето Tuning

Бутонът **Constraints** отваря диалогов прозорец за конфигуриране на граничните стойности към всяка променлива в проектираната система МПУ (Фиг. 4.13), подобен на този представен на Фиг. 4.10. Разликата е в наименованията на полетата, като групите променливи, на които могат да бъдат налагани ограничения, са запазени. Така например, полето **RateMin** приема неположителна стойност, която определя максималната стъпка на изменение на съответното управляващо въздействие в отрицателна посока. По аналогия, **RateMax** определя максималната стъпка на нарастване на съответното управляващо въздействие. Тя трябва да е зададена като неотрицателна стойност. Отново, при липса на ограничения полетата се оставят празни.



Фиг. 4.13 Параметри за настройка в полето Constraints

Бутонът **Weights** отваря диалогов прозорец, който дава възможност за задаване на стойности за тегловните коефициенти на предсказващия регулатор (Фиг. 4.14). Полетата в него са аналогични на описаните по-горе за версия на MATLAB® R2015a или по-ранна (Фиг. 4. 11).



Фиг. 4.14 Параметри за настройка в полето *Weights*

В следващото поле - **Performance tuning** (Фиг. 4.12), е разположен плъзгачът **Closed-Loop Performance**, който настройва цялостното съотношение между бързодействието на регулатора и точността при следене на задание-то както вече е описано в точка 1.1.1 .

В графичния интерфейс има няколко бутони даващи възможност за:

- **Import Controller** - Стартиране на регулатор проектиран преди това и съхранен в работното пространство или в MAT-файл;
- **Export Controller** - Експортиране на избрания регулатор в работното пространство на MATLAB® (съхранение под формата на MAT-файл в желана директория);
- **Import Plant** - Зареждане на готов модел;
- **Compare Controllers** – Сравнява няколко регулатори.

## 1. 2. Създаване на симулационен сценарий

Сценарият е набор от симулационни условия. Като първа стъпка при дефинирането на сценарий трябва да бъде избрана двойката модел-регулатор (Фиг. 4.15). Регулаторът и модела могат да бъдат избрани от падащите менюта **Controller** и **Plant**. В тях присъстват наименованията на всички регулатори и модели от текущия проект.

Следват задаването на продължителността на симулацията (**Duration<sup>1</sup>/Simulation Duration<sup>2</sup>**), конфигурирането на заданията (**Setpoints<sup>1</sup>/Reference Signals (setpoints for all outputs)<sup>2</sup>**) и смущенията (**Unmeasured disturbances<sup>1</sup>/Output Disturbances (added at MO channels)<sup>2</sup>**- смущения на входа на обекта и **Load Disturbances (added at MV channels)<sup>2</sup>**- смущения на изхода на обекта) в системата с МПУ.

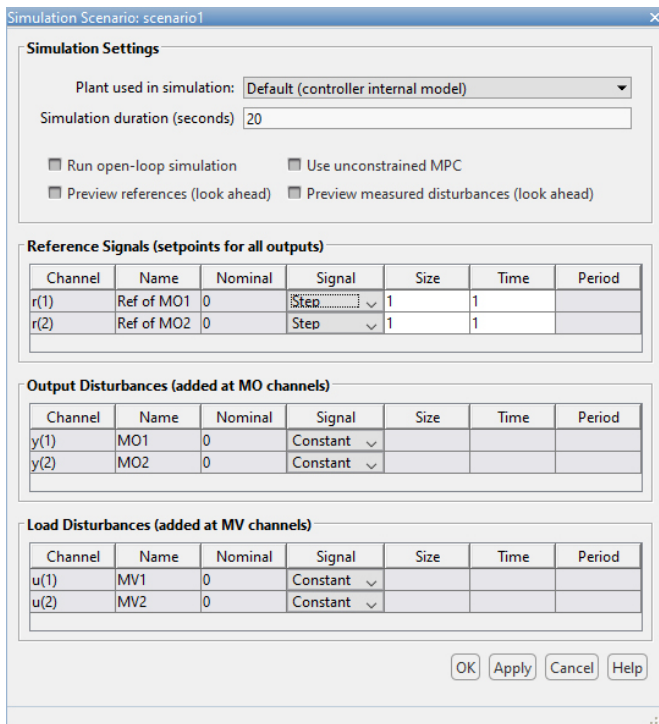
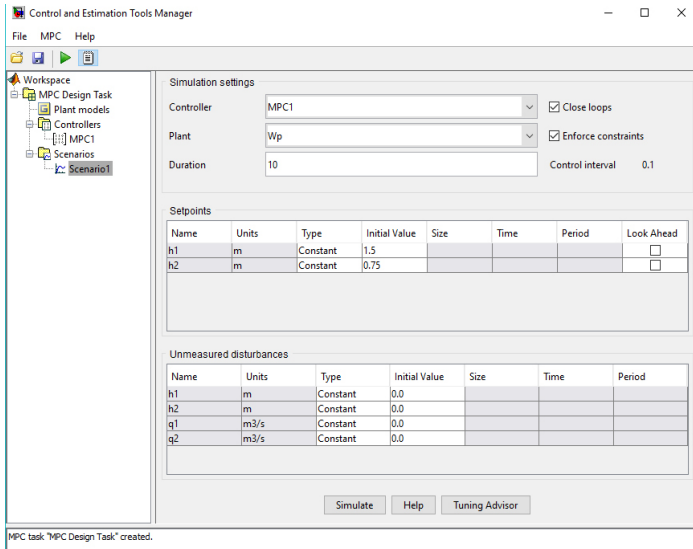
Видът на заданията може да бъде избран от падащото меню в полето **Type** или **Signal**. Потребителят може да избира между постоянно, стъпаловидно, синусоидално, линейно нарастващо във времето и др. задания. Опцията **Look ahead** означава, че всички бъдещи изменения на заданията са известни (програмно управление). В този случай, регулаторът може да генерира управляващи въздействия към подобряване на преходния процес.

След задаване на всички параметри на симулацията, тя може да се изпълни с бутона **Simulate** (зелената стрелка в лентата с инструменти)/**Plot Scenario**. Резултатът от изпълнението са две фигури, които изобразяват входовете и изходите на проектираната система с МПУ. На всяка от графиките осите са автоматично означени с имената на променливите, зададени при конфигурирането на модела.

---

1 за версии на MATLAB® преди R2015b

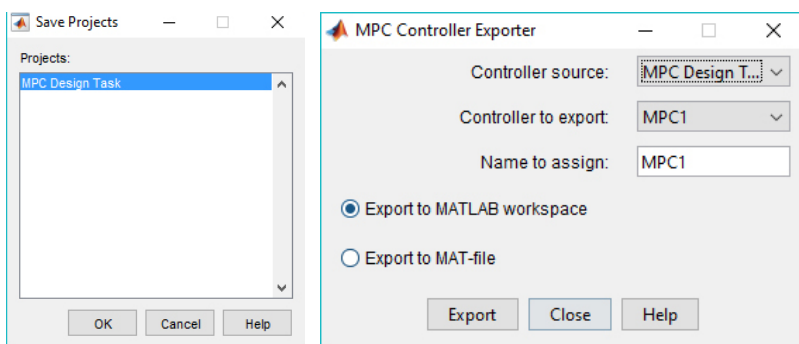
2 за версии на MATLAB® R2015b и по-висока



Фиг. 4.15 Настройка на симуляционния сценарий

### 1. 3. Запазване на проекта

При използване на инструмента MPC Tool всеки проект може да бъде запазен с цел по-нататъшно използване чрез избор File-->Save от менюто на основния прозорец. Отварянето/зареждането на вече създаден проект се прави с избор на File-->Load. Възможно е също запазване само на избран предсказващ регулатор чрез бутона **Export** (Фиг. 4.8), който отваря диалоговия прозорец от Фиг. 4.16. **Controller source** е проектът, към който принадлежи регулатора. **Controller to export** е изборът по подразбиране, сочещ към последно избрания регулатор. Полето **Name to assign** позволява преименуване на регулатора.

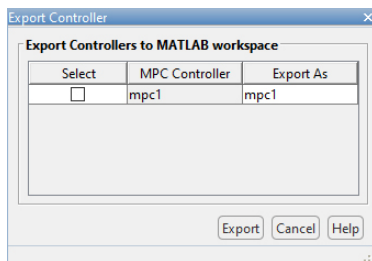


а)

б)

Фиг. 4.16 Експортиране/съхранение на а) проект и б) регулатор

По аналогия, при използване на инструмента MPC Designer, всеки проект може да бъде запазен чрез бутона **Save Session** под полето MPC Designer. Репективно, отварянето/зареждането на вече създаден проект се прави чрез бутона **Open Session**. Възможно е също запазване само на избран предсказващ регулатор чрез бутона **Export Controller** (Фиг. 4.17).



Фиг. 4.17 Диалогов прозорец Export Controller при използване на MPC Designer



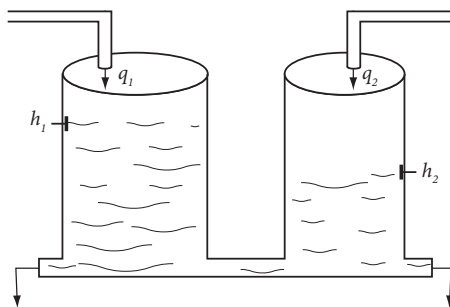
Експортирането/съхранението на регулатори позволява последващото им използване както в графичната среда на MPC Toolbox™, така и в средата на MATLAB/Simulink® (виж Тема 5).

## 2. Задачи за изпълнение.

**Задача 1.** Да се построи модел на обект по зададена матрична предавателна функция (4.1), съответстващ на многомерен обект с два входа и два изхода (Фиг. 4.19) [4].

$$W(p) = \begin{bmatrix} \frac{3p+1}{24p^2+11p+0.75} & \frac{0.5}{24p^2+11p+0.75} \\ \frac{0.5}{24p^2+11p+0.75} & \frac{8p+1}{24p^2+11p+0.75} \end{bmatrix} \quad (4.1)$$

Обектът представлява система от два резервоара свързани помежду си чрез тръба с постоянно сечение. Входовете на обекта са дебитите  $q_1$  и  $q_2$  на течност, вливаща се във всеки от резервоарите, а изходи са нивата в двата резервоара  $h_1$  и  $h_2$ . Всеки от резервоарите има постоянно напречно сечение:  $s_1 = 8 \text{ m}^2$ ,  $s_2 = 3 \text{ m}^2$  съответно за първия и втория резервоар. Ограниченията на входните и изходните сигнали са  $q_{1,2} = [0; 3] \text{ m}^3/\text{s}$ ,  $\Delta u = [-0.5; 0.5] \text{ m}^3/\text{s}$  и  $h_{1,2} = [0; 2] \text{ m}$ . Еквивалентните съпротивления на двете изходящи и свързващата тръби са със стойност  $0.5 \text{ m}^2/\text{s}$ .



Фиг. 4.19 Обект на управление – скачени резервоари

**Задача 2.** В графичната среда на MPC Toolbox™ да се проектира и симулира система с МПУ за обекта, описан в задача 1.

**Задача 3.** Да се снимат преходни процеси при постоянно и променливо задание.

### 3. Опитни данни и резултати.

**Изпълнение на задача 1.** За построяването на модел на обект по зададена матрична предавателна функция, съответстващ на MIMO обект – скачени резервоари се използва функцията **tf** в MATLAB®, както е показано в Табл. 4.2. Коефициентите в числителите (**num**) и знаменателите (**den**) на всяка от предавателните функции, съставлящи описание (4.1), се записват в променливи от тип cell.

*Табл. 4.2 Създаване на модела на многомерния обект на управление*

```
=====
clc;
clear
num={{[3 1] 0.5; 0.5 [8 1]};
den={{[24 11 0.75] [24 11 0.75]; [24 11 0.75] [24 11 0.75]};
Wp=tf(num,den);
=====
```

Съответствието между резултата от изпълнението на програмата и зададеното описание (4.1) може да се провери с отпечатване на променливата *Wp* в командния прозорец на MATLAB® (Табл. 4.3).

---

---

Wp =

From input 1 to output...

$$3s + 1$$

1: -----

$$24s^2 + 11s + 0.75$$

$$0.5$$

2: -----

$$24s^2 + 11s + 0.75$$

From input 2 to output...

$$0.5$$

1: -----

$$24s^2 + 11s + 0.75$$

$$8s + 1$$

2: -----

$$24s^2 + 11s + 0.75$$

Continuous-time transfer function.

---

---



*Използвайте създадения модел при решаване на задачи 2 и 3.  
Хоризонтите на предсказване и управление са съответно 10 и 2.*

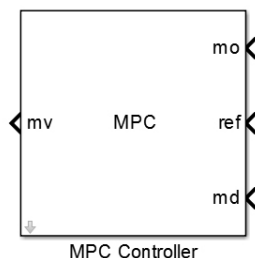
## ТЕМА 5

### Изследване на линейни системи с МПУ в средата на MATLAB/Simulink®

Целта на лабораторното упражнение е запознаване с начина и особеностите на работа при създаване на системи с МПУ в средата на Simulink®. В упражнението се акцентира върху използването на блока MPC Controller от Simulink® библиотеката MPC Toolbox™.

#### 1. Теоретична част.

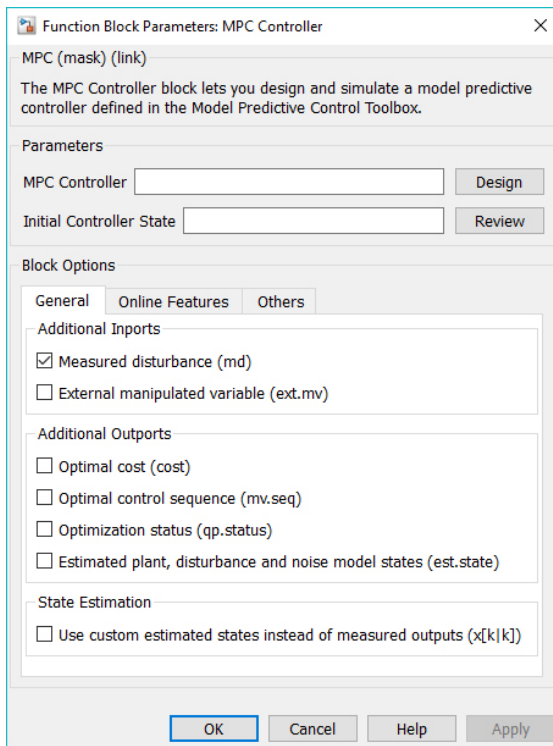
При проектирането на системи с МПУ в средата на MATLAB/Simulink® се използва блокът MPC Controller (Фиг. 5.1). Той позволява създаването както на едномерни така и многомерни системи с МПУ.



Фиг. 5.1 Блокът MPC Controller

Основният изглед на блока MPC Controller съдържа 3 входа и 1 изход. Към първия от входовете (**mo**, manipulated outputs) се подават управляваните изходи на обекта. При многомерни обекти изходите се обединяват чрез блока мултиплексор (mux). Вторият вход (**ref**, references) очаква заданията за всеки от изходите, а третият се използва при наличие на измерими смущения (**md**, measured disturbances). Изходът на блока, извежда изчислените управляващи въздействия (**mv**), които се прилагат към обекта на управление. Моделът в структурата на МПУ и параметрите на предсказващия регулатор (хоризонти на предсказване и управление, тегловни коефициенти, стъпка на дискретизация) се задават през графичния интерфейс на MPC Toolbox™,

който се отваря автоматично при натискане на бутона **Design** от диалоговия прозорец на блока (Фиг. 5.2). При наличие на предварително проектирана структура - модел и предсказващ регулатор, тя може да бъде заредена в блока посредством съответстващото ѝ име (името се записва в полето **MPC controller**).



Фиг. 5.2 Диалогов прозорец на блока MPC Controller

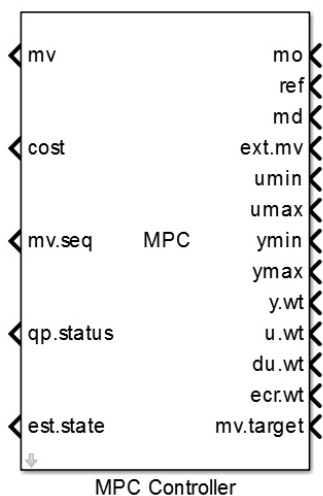
Диалоговият прозорец позволява допълнителни настройки на проектираната система с МПУ като:

- задаване на начални стойности на управляващите променливи (**Initial controller state**);
- добавяне на допълнителни входове към блока (секция **Optional inputs**), които дават възможност за добавяне на гранични интервали за входовете и изходите на системата с МПУ (**Input and**

**output limits**), външен управляващ сигнал (**Externally supplied MV signal**), контролирано превключване/разрешаване на оптимизационната процедура (**Optimization enabling switch**);

- добавяне на допълнителни изходи към блока (секция **Optional outports**), които извеждат стойността на целевата функция (**Optimal coast**), оптималната последователност от управляващи въздействия (**Optimal sequence**) и текущото състояние на оптимизационната процедура на МПУ (**Optimization status**);
- добавяне на допълнителни входове (секция **Online tuning inports**) за промяна на някои от настройките на предсказващия регулатор по време на изпълнение на симулационния модел: настройка на тегловните коефициенти на входовете (**Weights on control inputs**) и изходите (**Weights on control outputs**), смекчаване на ограниченията в МПУ (**Weights on overall constraints softening**).

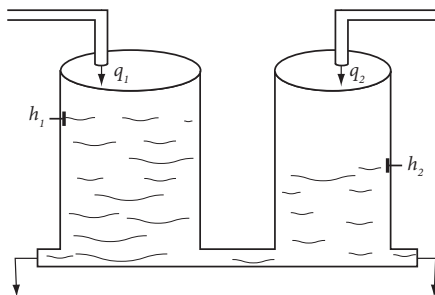
Всяко добавяне на вход или изход към блока MPC Controller автоматично изменя изгледа му (Фиг. 5.3). Добавянето на вход не задължава потребителя с неговото използване.



Фиг. 5.3 Блокът MPC Controller с всички входове и изходи

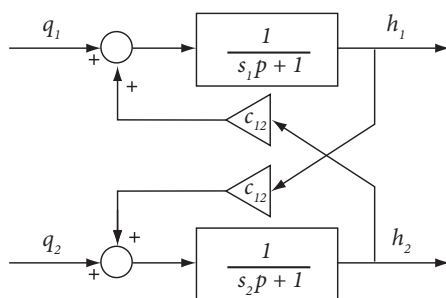
## 2. Задачи за изпълнение.

**Задача 1.** Да се моделира система с МПУ за многомерен обект в средата на MATLAB/Simulink®. Обектът представлява модел на скачени резервоари, въведен в Тема 4.



Фиг. 5.4 Обект на управление – скачени резервоари

1.1. За моделирането на обекта в Simulink® да се използва структурната схема от Фиг. 5.5, където напречните сечения на резервоарите са  $s_1 = 8 \text{ m}^2$ ,  $s_2 = 3 \text{ m}^2$ , а еквивалентните съпротивления на двете изходящи и свързващата тръби са зададени като  $c_1 = c_2 = c_{12} = 0.5 \text{ m}^2/\text{s}$ .



Фиг. 5.5 Структурна схема на обекта на управление

1.2. Да се конфигурира блока за регулатора. Ограниченията на входните и изходните сигнали са  $q_{1,2} = [0; 3] \text{ m}^3/\text{s}$ ,  $\Delta u = [-0.5; 0.5] \text{ m}^3/\text{s}$  и  $h_{1,2} = [0; 2] \text{ m}$ , а хоризонтите на предсказване и управление са съответно 10 и 2.

1. 3. Да се симулира затворената система в средата на MATLAB/Simulink®.

**Задача 2.** Да се моделира система с МПУ за обект зададен чрез описанието в пространство на състоянията (5.1):

$$A = \begin{bmatrix} -0.5 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -0.714 & -1.428 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad C = [0 \ 1 \ 1 \ 0 \ 1.428]$$

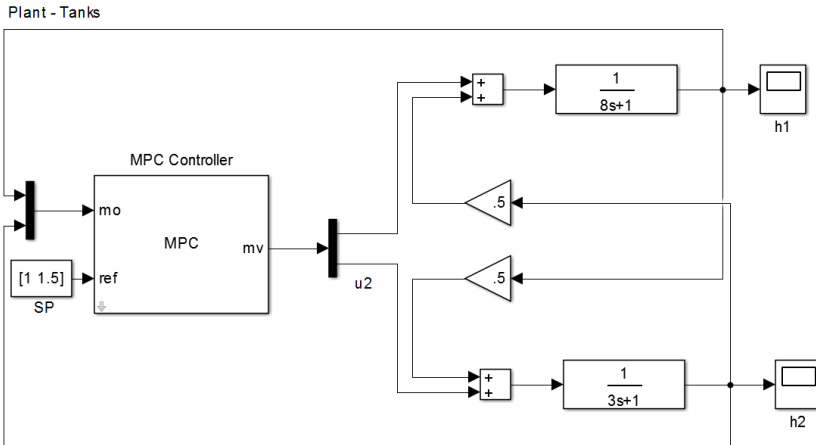
$$D = [0 \ 0 \ 0]$$
(5.1)

Ограниченията на входните и изходните сигнали са  $u = [0; 1]$ ,  $\Delta u = [-0.1; 0.1]$  и  $y = [0; 1]$ , а хоризонтите на предсказване и управление са съответно 10 и 2.

### 3. Опитни данни и резултати.

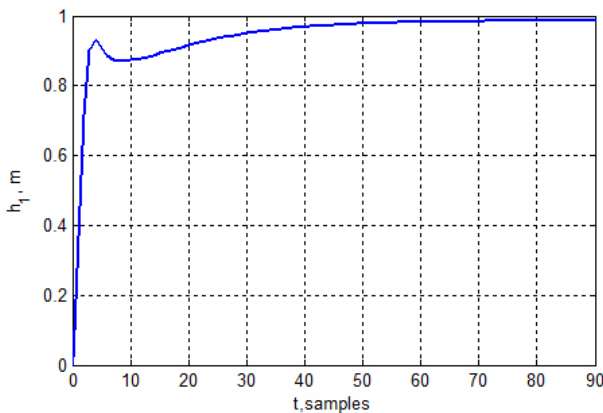
**Изпълнение на задача 1:** Симулационният модел на обекта на управление е създаден съобразно структурната схема на Фиг. 5.5. За моделиране на системата с МПУ към него е добавен предсказващия регулатор - блокът MPC Controller и блока Constant, с помощта на който са зададени желаните нива на точността за всеки от резервоарите (Фиг. 5.6).



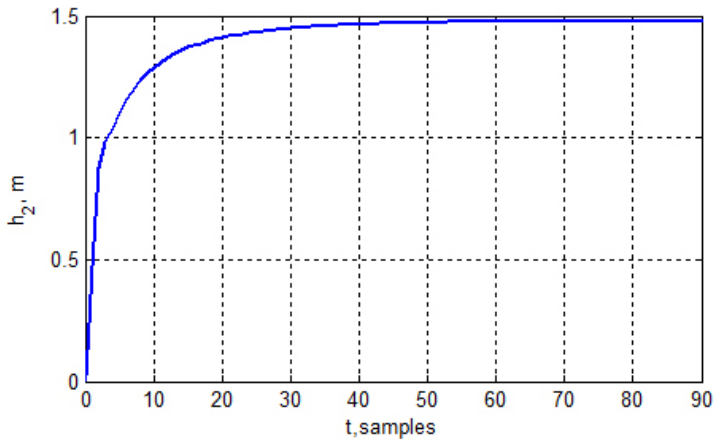


Фиг. 5.6 Структурна схема система с МПУ на обект скачени резервоари

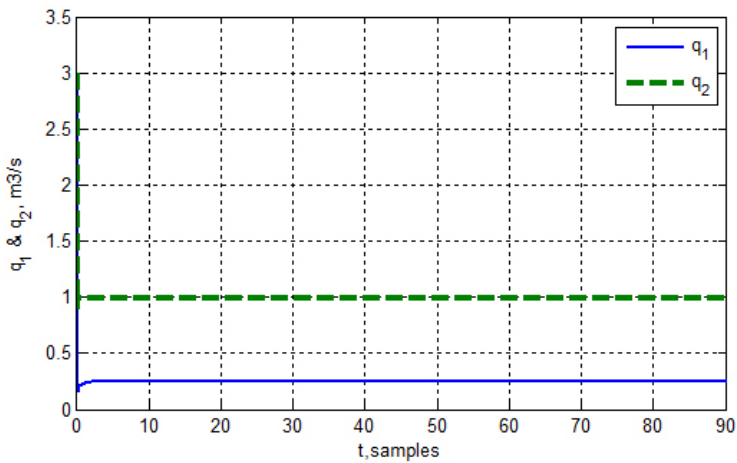
Моделът и параметрите на системата с МПУ са зададени и настроени с помощта на графичната среда за проектиране на МПУ в средата на MATLAB® (виж Тема 4). Преходните процеси на изхода на така проектираната затворена система. Изменението на нивото във всеки от резервоарите е представено на Фиг. 5.7 и Фиг. 5.8, а изменението на дебитите  $q_{i,2}$  е показано на Фиг. 5.9.



Фиг. 5.7 Преходен процес на нивото  $h_1$  (задание = 1 m)



Фиг. 5.8 Преходен процес на нивото  $h_2$  (задание = 1.5 m)



Фиг. 5.9 Управляващи сигнали  $q_1$  и  $q_2$



Симулирайте системата с променливи задания за нивата. Сравнете получените резултати с тези от Тема 4. Ако е необходимо, повторете експериментите с цел уеднаквяване на условията.

**Изпълнение на задача 2:** С цел, създаването на система за автоматично управление на обект зададен в пространство на състоянието е създадена програмата от Табл. 5.1. Тя създава модела на обекта, използвайки матриците от описанието в пространството на състоянието. За целите на МПУ моделът се трансформира в дискретен вид (функция **c2d**), дефинират се броя на измеримите и неизмеримите смущения (функция **setmpcsignals**(име на модела, 'тип на входа', 'номер на входа')).

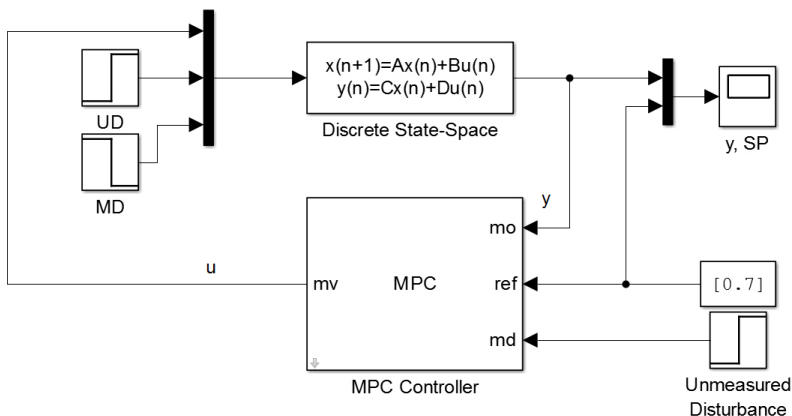
Табл. 5.1 Създаване и инициализиране на модела на обекта

```

=====
A1 = [-0.5 -1 0 0 0; 1 0 0 0 0; 0 0 -1 0 0; 0 0 0 -0.714 -1.428; 0 0 0 1 0];
B1 = [1 0 0; 0 1 0; 0 0 1; 0 0 0; 0 0 0];
C1 = [0 1 1 0 1.428];
D1 = [0 0 0];
delta = 0.2;
model_ss = ss(A1,B1,C1,D1);
model_dss = c2d(model_ss, delta);
model = setmpcsignals(model_dss,'MV',1;MD;2;UD;3);
=====

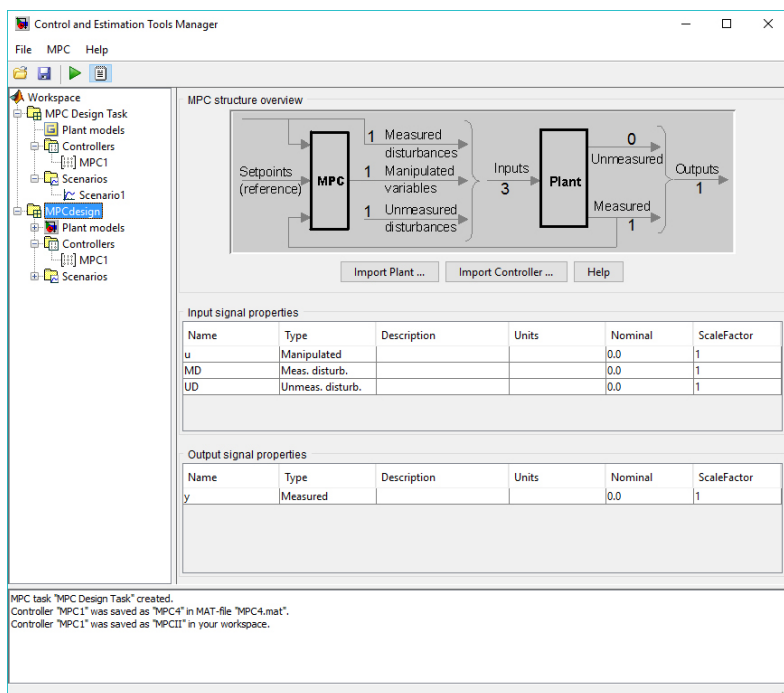
```

Структурата на системата с МПУ за този обект в Simulink® е представена на Фиг. 5.10.



Фиг. 5.10 Структурна схема в Simulink® на система с МПУ с обект, зададен с описание в пространство на състоянието

Блокът на регулатора се настройва в графичната среда на MPC Toolbox™ за настройка и проектиране на МПУ. Въвеждат се ограниченията на входните и изходните сигнали  $u = [0; 1]$ ,  $y = [0; 1]$  и  $\Delta u = [-0.1; 0.1]$ , както и хоризонтите на предсказване и управление зададени в условието на задачата.



Фиг. 5.11 Настройка на МПУ при изпълнение на Задача 2

Както се вижда на Фиг. 5.11 входните сигнали в този случай са три: управляващото въздействие, измеримото смущение и неизмеримо смущение, които бяха дефинирани при създаване на модела на обекта (Табл. 5.1).



**Симулирайте системата от Фиг. 5.10 при отсъствие и наличие на смущаващи въздействия. Представете графично получените резултати.**

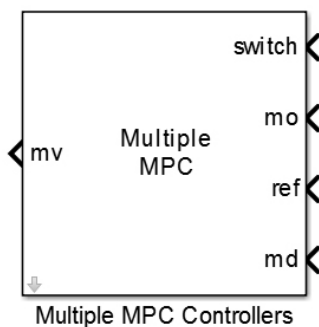
## ТЕМА 6

### Управление на нелинейна система чрез многомоделен предсказващ регулатор

Целта на лабораторното упражнение е запознаване с начините за реализиране на многомоделен предсказващ регулатор с използване на MPC Toolbox™ в средата на MATLAB/Simulink®.

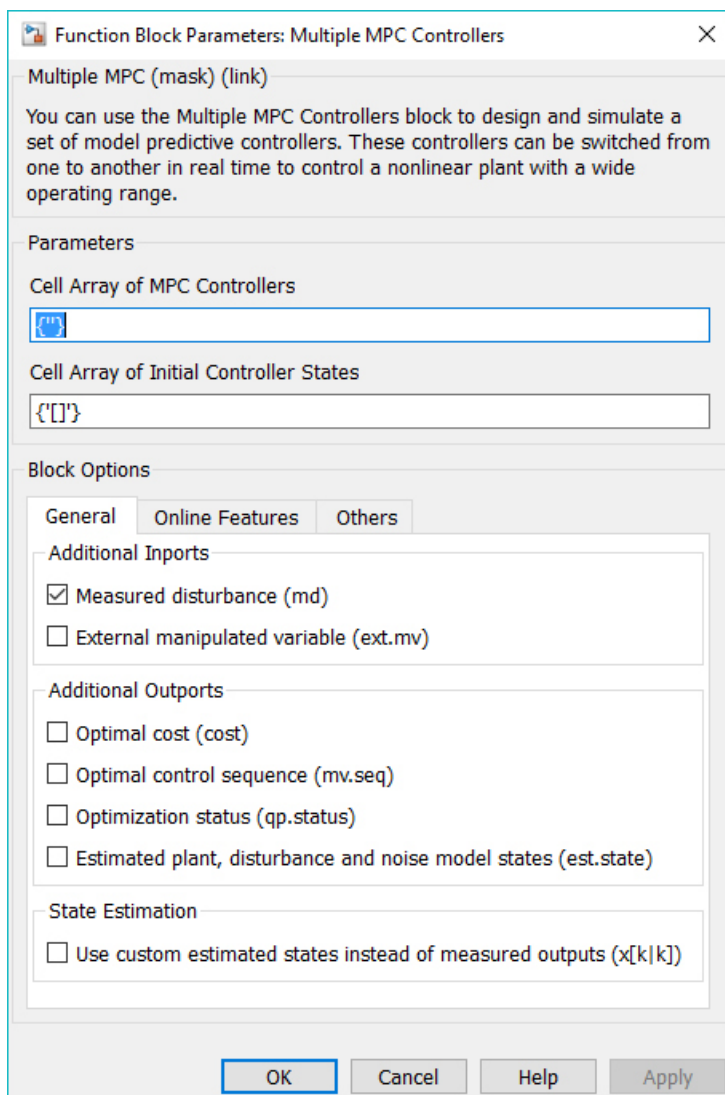
#### 1. Теоретична постановка

За моделиране на многомоделния предсказващ регулатор в Simulink® се използва блока **Multiple MPC Controllers** от библиотеката на MPC Toolbox™ (Фиг. 6.1).



Фиг. 6.1 Блокът *Multiple MPC Controllers*

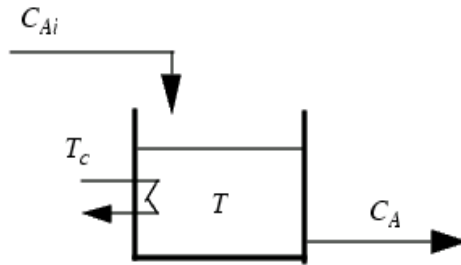
Той позволява използването на неограничен брой модели на обекта на управление, за всеки от които могат да се конфигурират параметрите на съответен регулатор. Така системата с МПУ разполага с множество предсказващи регулатори, превключващи при дефинирани от потребителя условия. Диалоговият прозорец (Фиг. 6.2) наподобява този на блока **MPC Controller** (виж Тема 5). Липсват секциите за добавяне на допълнителни изходи и допълнителни входове за промяна на някои от настройките на предсказващия регулатор.



Фиг. 6.2 Диалогов прозорец на блока *Multiple MPC Controllers*

На Фиг. 6.3 е представена принципна схема на химичен реактор. Системата за управление има три входа:  $C_{Ai}$  - концентрация на постъпващия материал,  $T_i$  - температура на постъпващия материал и  $T_C$  - температура на охлаждане. Изходите са два:  $T$  - температура в реактора и  $C_A$  - концентрация на основния реактант. Процесът в реактора е екзотермичен (отделя се топлина),

именно това обуславя и постановката на задачата за управление. Целта е предотвратяване на температурни загуби чрез поддържане на определена температура в реактора. Възможно решение е управление на  $C_A$  посредством управляващия вход  $T_c$ . Така чрез концентрацията на основния реактант може да се контролира, протичащата в реактора, екзотермична реакция [7].



Фиг. 6.3 Принцилна схема на химичен реактор

## 2. Задачи за изпълнение.

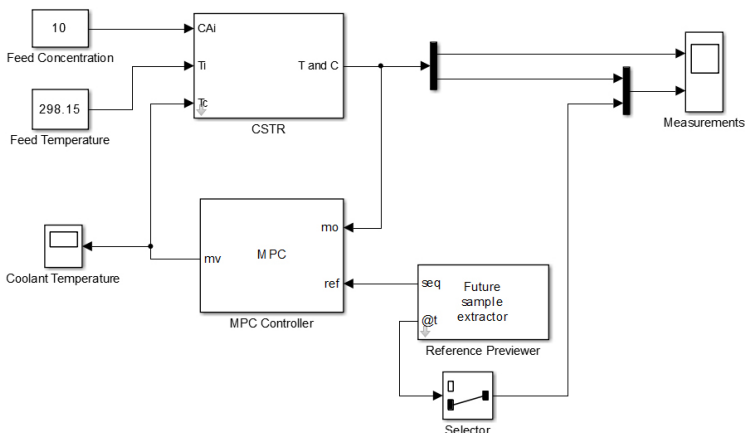
**Задача 1.** Моделиране на система за управление на нелинеен обект – химичен реактор.

**Задача 2.** Проектиране на многомоделен предсказващ регулатор.

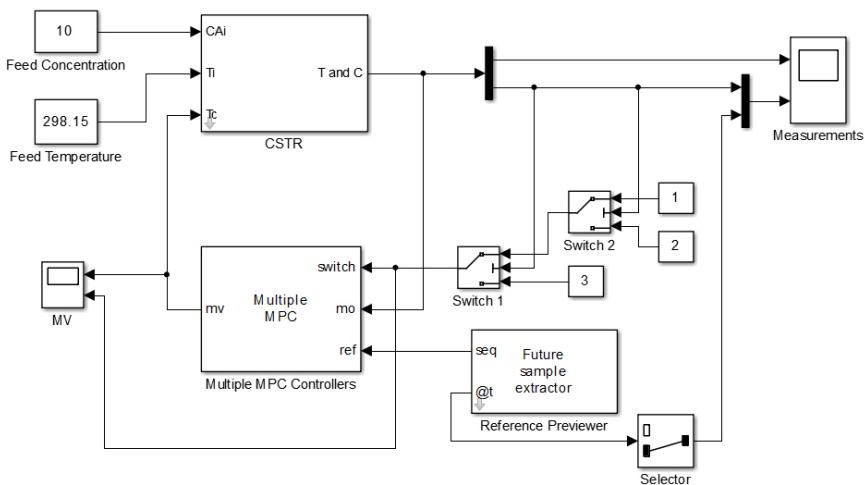
**Задача 3.** Изследване поведението на системата за управление.

## 3. Опитни данни и резултати.

На Фиг. 6.4 е представена структурна схема за управление на нелинейния обект с модел, получен чрез линеаризация в една работна точка, а на Фиг. 6.5 е представена система за управление чрез многомоделния предсказващ регулатор, който използва три модела на нелинейния химичен реактор, получени в три различни точки на линеаризация. В програмата **moccstr.m** са направени три линеаризации на нелинейния обект на управление, като за всяка зона на линеаризация е проектиран отделен предсказващ регулатор без ограничения. Превключването между трите регулатора се осъществява с блока **Switch** в Simulink® (Фиг. 6.5).



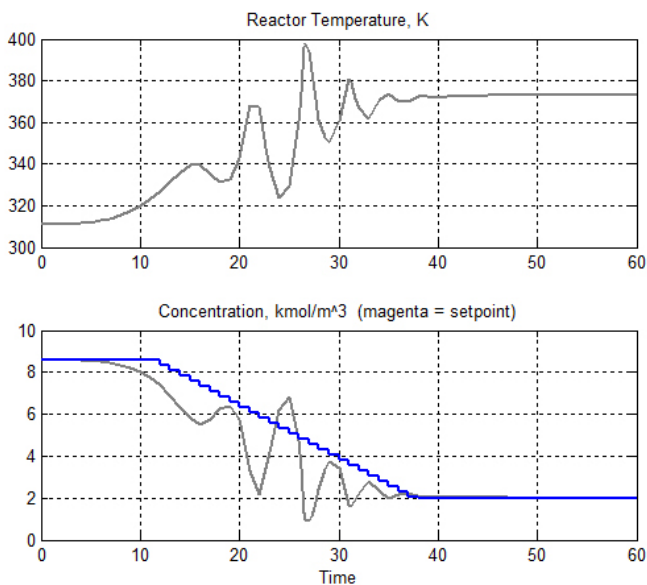
Фиг. 6.4 Структурна схема в Simulink® за управление на химичен реактор с един предсказващ регулатор



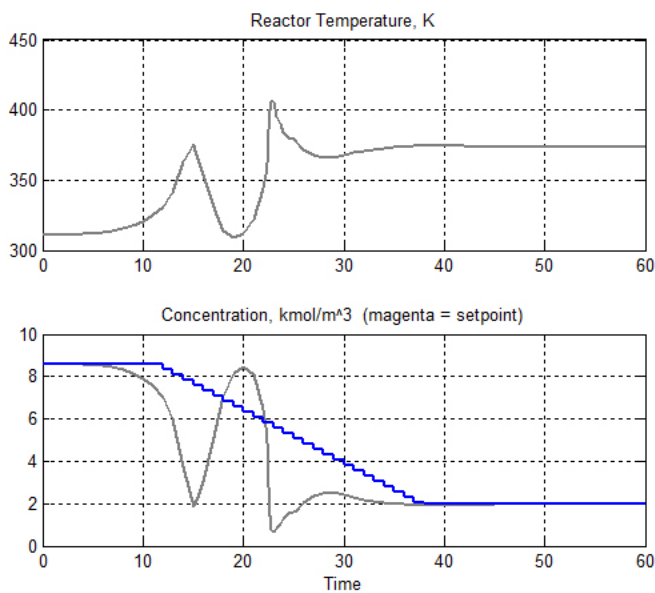
Фиг. 6.5 Структурна схема в Simulink® за управление на химичен реактор с многомоделния предсказващ регулатор

Резултатите от симулациите на системата за управление с един предсказващ регулатор MPC, MPC2 и MPC5 са представени съответно на Фиг. 6.6, Фиг. 6.7 и Фиг. 6.8. Както се вижда зависимостта на концентрацията от температурата в реактора е обратнопропорционална.

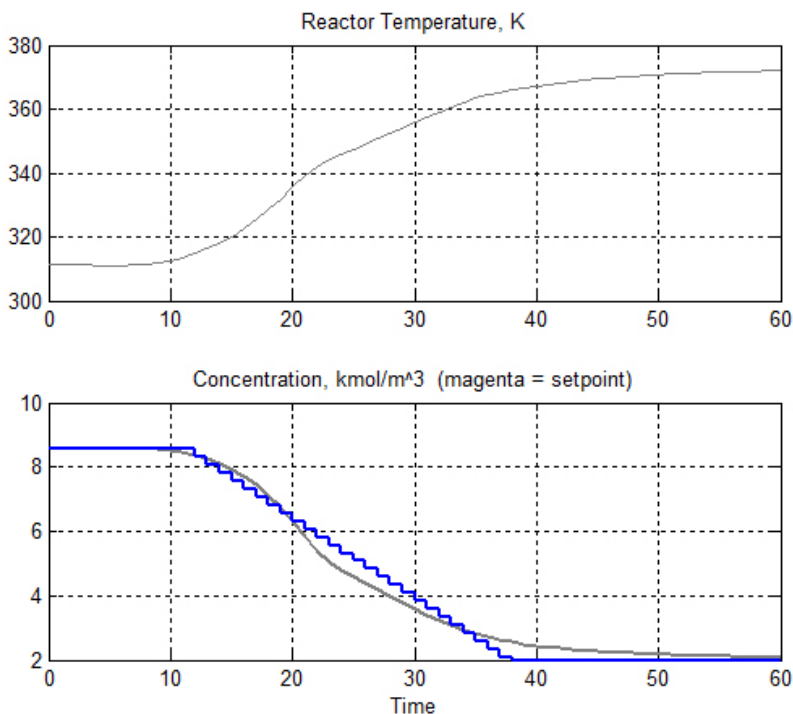




Фиг. 6.8 Преходни процеси при работа на системата с предсказващ регулатор MPC



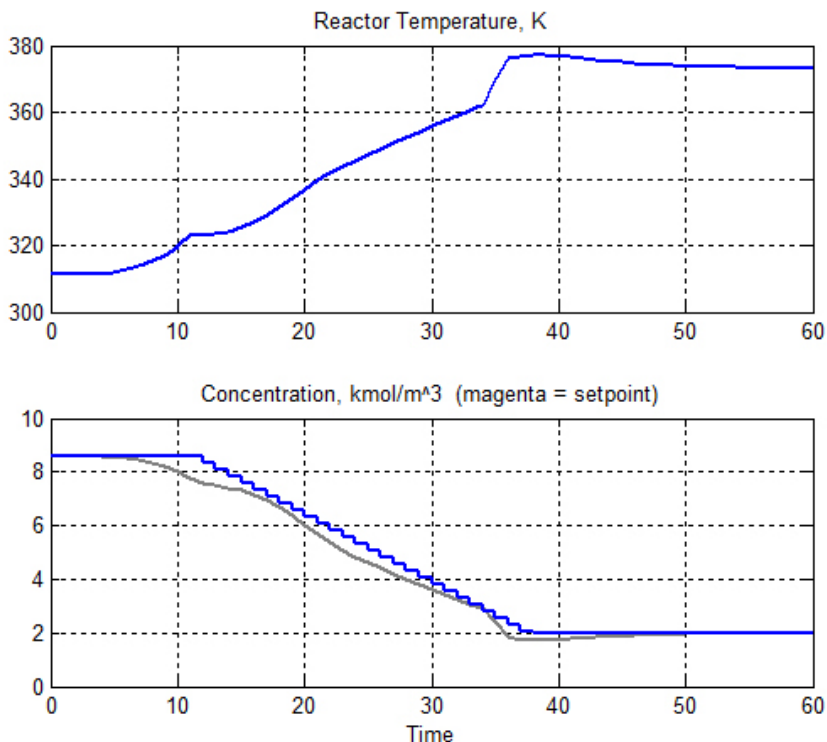
Фиг. 6.7 Преходни процеси при работа на системата с предсказващ регулатор MPC2



Фиг. 6.8 Преходни процеси при работа на системата с предсказващ регулатор MPC5

За сравнение на Фиг. 6.9 са представени преходните процеси на двата изхода на системата при работа с многомерен предсказващ регулатор. Превключващ между трите регулатора при следните условия:

- при  $C_A \geq 8 \text{ kmol/m}^3$ , използва се регулатор MPC1;
- при  $3 \leq C_A < 8 \text{ kmol/m}^3$ , използва се регулатор MPC5;
- при  $C_A < 3 \text{ kmol/m}^3$ , използва се регулатор MPC2.



Фиг. 6.9 Преходен процес на концентрацията в реактора при работа на системата с многомоделен предсказващ регулатор

Симулационните изследвания както с едномоделния така и с многомоделния предсказващ регулатор могат да бъдат изпълнени със стартиране на програмата **mpccstr.m**. Тя може да се отвори през командния прозорец на MATLAB® с командата **open('mpccstr')**.

---

```
%% MPC Control of a Chemical Reactor with Multiple Operating Conditions
```

```
%%
```

```
% The objective is to control a chemical reactor (CSTR) as it transitions from an initial condition at low (~15%) conversion to a desired steady state at 80% conversion. The reaction is exothermic (liberates heat) and the CSTR temperature must be controlled to prevent a thermal runaway. Multiple MPC Controllers block is used to illustrate a scheduling based strategy to solve a nonlinear control problem with multiple operating conditions.
```

```
% The reactor is modeled by 'CSTR_INOUT.mdl'. For background, see Seborg,  
% D.E. et al., "Process Dynamics and Control", 2nd Ed., 2004, Wiley, pp.34-36.
```

```
%% Open-Loop Model: Linearize the Nonlinear CSTR Model
```

```
if ~mpcchecktoolboxinstalled('Simulink®')
```

```
    disp('Simulink®(R) is required to run this demo.')
```

```
    return
```

```
end
```

```
%%
```

```
% We need a linear model to use in MPC. The controller will adjust a  
% coolant temperature, which is one of three plant inputs. The others are  
% the concentration of the limiting reactant in the CSTR feed stream and  
% the temperature of this stream. The CSTR states are the temperature and  
% concentration of the limiting reactant in the product stream. We will  
% assume that both can be measured and used for feedback control. As a  
% first attempt, linearize the CSTR model at the initial condition.
```

```
%
```

```
% Set initial states and inputs
```

```
yp = [300; 8.57];
```

```
up = [10; 298.2; 298.2];
```

```
%%
```

```
% Compute initial condition
```

```
[x0, u0, y0, dx] = trim('CSTR_INOUT', [], up, yp, [], [1 2], 2);
```

```
%%
```

```
% Obtain linear model
```

```
[a, b, c, d] = linmod('CSTR_INOUT', x0, u0);
```

```
% Verify that the linear model is open-loop stable at this condition
```

```
eig(a)
```

```
%%
```

```
% All eigenvalues are negative so the plant is open-loop stable.
```

```
%
```

```
% Configure a state-space model for use in MPC
```

```

CSTR = ss(a, b, c, d);
CSTR.InputName = {'C_f','T_f','T_c'};
CSTR.OutputName = {'T_r','C_r'};
%%
% Designate the coolant temperature as the manipulated variable
CSTR.InputGroup.MV = 3;
%%
% The other two inputs are unmeasured disturbances
CSTR.InputGroup.UD = [1, 2];
%%
% Create the MPC model format and set the nominal conditions.
Model.Plant = CSTR;
Model.Nominal.X = x0;
Model.Nominal.Y = y0;
Model.Nominal.U = [0; 0; u0(3)]; % NOTE: UD nominal values must be zero.

%% Define a MPC Controller
Ts = 1; % Sampling period, minutes.
MPC = mpc(Model, Ts);
%%
% The goal will be to track a specified transition in the reactor
% concentration. The reactor temperature will be measured and used in
% state estimation but the controller will not attempt to regulate it
% directly. It will vary as needed to regulate the concentration. Thus,
% set its MPC weight to zero.
MPC.Weights.OV = [0 1];
%%
% All other MPC parameters are at their default values.

%% Test the Controller With a Step Disturbance in Feed Concentration
% 'CSTR_MMPC_1.mdl' contains a Simulink®(R) model with CSTR and MPC
% Controller blocks in a feedback configuration.
open_system('CSTR_MMPC_1')
%%
% Define a constant setpoint for the output.
CSTR_Setpoints.time = [0; 60];
CSTR_Setpoints.signals.values = [y0 y0]';
% Test the response to a 5% increase in feed concentration.
set_param('CSTR_MMPC_1/Feed Concentration','Value','10.5');
%%
% Set plot scales and simulate the response
set_param('CSTR_MMPC_1/Measurements','Open','on')
set_param('CSTR_MMPC_1/Coolant Temperature','Open','on')
set_param('CSTR_MMPC_1/Measurements','Ymin','305~8','Ymax','320~9')

```

```

set_param('CSTR_MMPC_1/Coolant Temperature','Ymin','295','Ymax','305')
sim('CSTR_MMPC_1', 60)

%% Modify Controller Design -- Increase Estimator Signal-to-Noise
% The controller is stable but its disturbance rejection is sluggish. Try
% increasing the state estimator signal-to-noise by a factor of 10 and test
% the design again.
D = ss(getindist(MPC));
D.b = eye(2)*10;
set(D,'InputName',[],'OutputName',[],'InputGroup',[],'OutputGroup',[]);
setindist(MPC, 'model', D);
%%
% Simulate
sim('CSTR_MMPC_1', 60)
% NOTE: Plant inputs 1 and 2 are unmeasured disturbances. By default,
% the controller assumes integrated white noise with unit magnitude at
% these inputs when configuring the state estimator. The above statements
% increase the white noise magnitudes to 10.
%% Test the Modified Controller for the Full Transition
% Define the desired setpoint transition. After a 10-minute warm-up
% period, ramp the concentration setpoint downward at a rate of 0.25 per
% minute until it reaches 2.0 kmol/m3.
CSTR_Setpoints.time = [0 10 11:39]';
CSTR_Setpoints.signals.values = [y0(1)*ones(31.1),[y0(2);y0(2);(y0(2):-0.25:2)';2;2]];
%%
% The MPC Controller block is configured to use these setpoints in a
% look-ahead mode, i.e., anticipating the setpoint transition. This
% generally improves setpoint tracking.
open_system('CSTR_MMPC_1/MPC Controller')
%%
% Remove the 5% increase in feed concentration
set_param('CSTR_MMPC_1/Feed Concentration','Value','10')
%%
% Set plot scales and simulate the response
set_param('CSTR_MMPC_1/Measurements','Ymin','300~0','Ymax','400~10')
set_param('CSTR_MMPC_1/Coolant Temperature','Ymin','240','Ymax','360')
% Simulate
sim('CSTR_MMPC_1', 60)

%% Design Another Controller for the Final Operating Condition
% Performance along the full transition could be improved if another model
% were used. An obvious candidate would be one representing the desired
% final state. Obtain this by linearization as before and design a
% controller identical to the previous except for the prediction model.

```

```

% Determine steady state conditions when CSTR concentration is 2 kmol/m^3.
yp = [300; 2];
[x0, u0, y0, dx] = trim('CSTR_INOUT', [], up, yp, [], [1 2], 2);
%%
% Linearize at this condition.
[a, b, c, d] = linmod('CSTR_INOUT', x0, u0);
%%
% Check stability
eig(a)

% The model is again open-loop stable.
% Form the new controller.
CSTR2 = CSTR;
set(CSTR2, 'a', a, 'b', b, 'c', c, 'd', d);
MPC2 = MPC;
MPC2.Model.Plant = CSTR2;

% Change the controller and re-run the simulation
set_param('CSTR_MMPC_1/MPC Controller', 'mpcobj', 'MPC2')
set_param('CSTR_MMPC_1/Measurements', 'Ymin', '300~0', 'Ymax', '420~10');
sim('CSTR_MMPC_1', 60)
%%
% Close this model
% bdclose('CSTR_MMPC_1')

%% Obtain a Third Controller for the Intermediate Operating Condition
% Performance did not improve significantly. Now define a model
% representing an intermediate CSTR concentration (5.5 kmol/m^3).
yp = [300; 5.5];
[x0, u0, y0, dx] = trim('CSTR_INOUT', [], up, yp, [], [1 2], 2);
[a, b, c, d] = linmod('CSTR_INOUT', x0, u0);
eig(a)
%%
% Note that this model is open-loop unstable. Thus, operation transitions
% from stable to unstable and back to stable. This suggests the use of
% multiple controllers.
%
% Design a new controller using the new model.
CSTR5 = CSTR;
set(CSTR5, 'a', a, 'b', b, 'c', c, 'd', d);
MPC5 = MPC;
MPC5.Model.Plant = CSTR5;

```

```
% NOTE: As the plant is unstable it's important that the disturbance model
% include an unmeasured disturbance input. This allows the Kalman estimator
% to be stable.
```

```
%% Control the Plant With the Multiple MPC Controllers Block
% The following model uses the Multiple MPC Controllers block instead of
% the MPC Controller block to implement three MPC controller across the
% operating range.
```

```
open_system('CSTR_MMPC_3')
```

```
%%
```

```
% Note that it has been configured to use the three controllers in a
% sequence: MPC/MPC5/MPC2
```

```
open_system('CSTR_MMPC_3/Multiple MPC Controllers')
```

```
%%
```

```
% Note also that the two switches specify when to switch from one
% controller to another. The rules are:
```

```
% If CSTR concentration  $\geq 8$ , use MPC
```

```
% If  $3 \leq$  CSTR concentration  $< 8$ , use MPC5
```

```
% If CSTR concentration  $< 3$ , use MPC2
```

```
%%
```

```
% Simulate with the Multiple MPC Controllers block
```

```
set_param('CSTR_MMPC_3/Measurements','Open','on');
```

```
set_param('CSTR_MMPC_3/MV','Open','on');
```

```
sim('CSTR_MMPC_3', 60)
```

```
%%
```

```
% The transition is now well controlled. The major improvement is in the
% transition through the OL-unstable region. The plot of the switching
% signal shows when controller transitions occur. The MV character changes
% at these times because of the change in dynamic characteristics
% introduced by the new prediction model.
```

```
%
```

```
% Close this model
```

```
% bdclose('CSTR_MMPC_3')
```

```
displayEndOfDemoMessage(mfilename)
```

```
=====
```



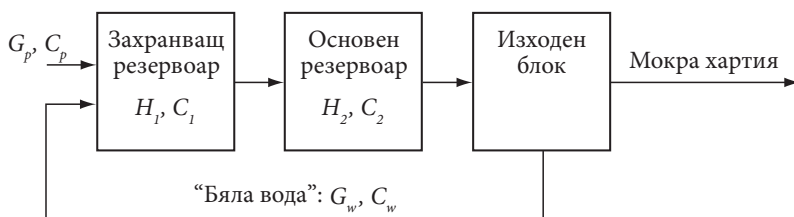
## ТЕМА 7

### Синтез на предсказващ регулатор за управление на агрегат за производство на хартия

Целта на лабораторното упражнение е проектиране на система с МПУ за конкретен обекта на управление - агрегат за производство на хартия.

#### 1. Задачи за изпълнение.

**Задача 1.** Да се създаде модел в пространството на състоянията на обекта на управление, показан на Фиг. 7.1. За създаване на модела да се използват матриците зададени в (7.1).



Фиг. 7.1 Принципна схема на агрегата за производство на хартия

$$A = \begin{bmatrix} -1.93 & 0 & 0 & 0 \\ 0.394 & -0.42 & 0 & 0 \\ 0 & 0 & -0.63 & 0 \\ 0.82 & -0.784 & 0.412 & -0.425 \end{bmatrix} \quad C = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} 1.274 & 1.274 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1.34 & -0.65 & 0.203 & 0.406 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (7.1)$$

Векторът на състоянието е дефиниран като:

$$x = [H_1 \ H_2 \ C_1 \ C_2]^T \quad (7.2)$$

където  $H_1$  е нивото на течността в хранящия резервоар (Feed Tank),  $H_2$  е нивото на сместа в главния резервоар (Headbox),  $C_1$  и  $C_2$  са съответно консис-тенция (плътност) на материала в хранящия и главния резервоар резервоар. Измеримите изходи са три и се дефинират с вектора (7.3):

$$y = [H_2 \ C_1 \ C_2]^T \quad (7.3)$$

Целта на управлението е да се поддържат постоянни стойности на параметрите в главния резервоар  $C_2$  и  $H_2$ . Управляващите въздействия в системата са две, а именно:

$$u = [G_p \ G_w]^T \quad (7.4)$$

където  $G_p$  е поток на пулпа (Stock), влизаща в хранящия резервоар, а  $G_w$  поток на инфилтрант (рециклирана бяла вода). Плътността на пулпа на входа на хранящия резервоар  $C_p$  представлява измеримо смущение :

$$w = C_p \quad (7.5)$$

Неизмеримото смущение в системата е представено от плътността на инфилтранта  $C_w$ :

$$d = C_w \quad (7.6)$$

**Задача 2.** Да се проектира на предсказващ регулатор.

**Задача 3.** Да се снемат на преходни процеси по задание и смущение чрез графичния интерфейс на MPC Toolbox™.

**Задача 4.** Системата да се изследва в средата на MATLAB/Simulink®.

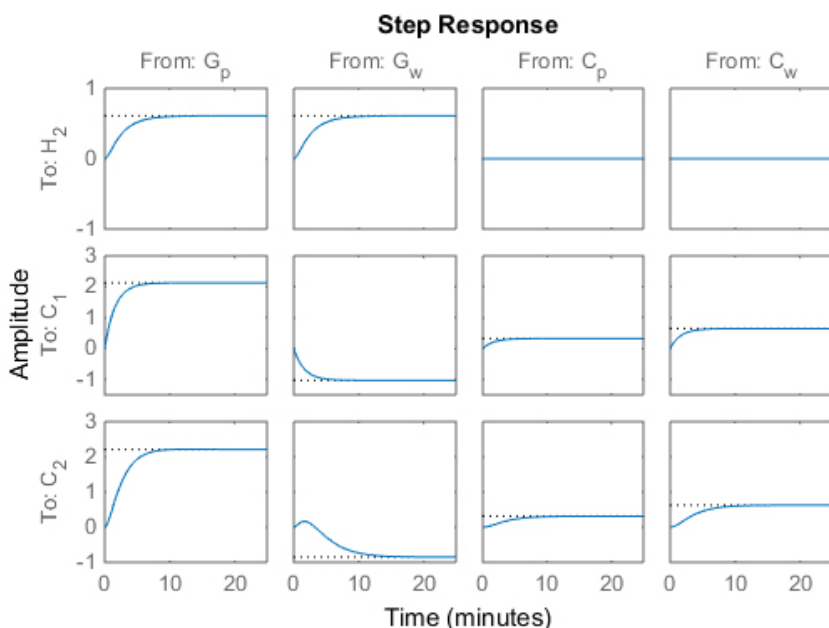
## 2. Опитни данни и резултати.

**Изпълнение на задачи 1, 2 и 4:** За реализиране на обекта на управление, описван с линеаризирания модел в пространство на състоянията е съставена програмата от Табл. 7.1.

Предавателните характеристики на променливите на агрегата за производство на хартия, представени на Фиг. 7.2, показват, че:

- двете управляващи въздействия оказват влияние върху трите изхода;
- ефектът на двете управляващи въздействия върху нивото в главния резервоар е почти еднакво.

Като тези особености не позволяват постигането на точно и независимо управление на  $C_2$  и  $H_2$ .



Фиг. 7.2 Предавателни характеристики на променливите на агрегата за производство на хартия

След като моделът на обекта е създаден (Табл. 7.1), той се зарежда в графичния интерфейс на MPC Toolbox. Входните и изходните сигнали се описват, както е показано на Фиг. 7.3. Регулаторът се настройва със следните параметри [7]:

- **Модели и хоризонти:** интервал на управление = 2 минути;
- **Ограничения:** за  $G_p$  и  $G_w$ , минимум = -10, максимум = 10, минимална скорост = -2, максимална скорост = 2;
- **Тегловни коефициенти:** за  $G_p$  и  $G_w$  тегло = 0, скорост на теглото = 0.4. За  $C_1$ , тегловен коефициент = 0. (Останалите изходи имат коефициенти = 1).

Табл. 7.1 Построяване на модела на обекта представен в пространството на състоянието

```
=====
clear all
clc
A = [ -1.9300  0      0      0
      0.3940 -0.4260  0      0
      0      0      -0.6300  0
      0.8200 -0.7840  0.4130 -0.4260];

B = [ 1.2740  1.2740  0      0
      0      0      0      0
      1.3400 -0.6500  0.2030  0.4060
      0      0      0      0];

C = [ 0  1  0  0
      0  0  1  0
      0  0  0  1];

D = zeros(3,4);

PaperMach = ss(A, B, C, D);
PaperMach.InputName = {'G_p','G_w','N_p','N_w'};
PaperMach.OutputName = {'H_2','C_1','C_2'};
PaperMach.TimeUnit = 'minutes';
step(PaperMach);
=====
```

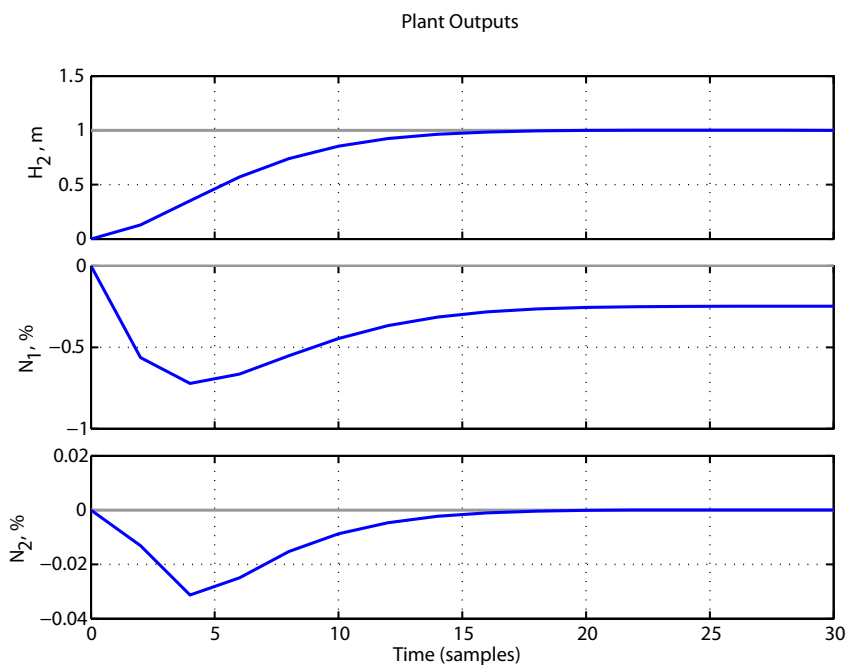
Input signal properties					
Name	Type	Description	Units	Nominal	ScaleFactor
G_p	Manipulated			0.0	1.0
G_w	Manipulated			0.0	1.0
C_p	Manipulated			0.0	1.0
C_w	Manipulated			0.0	1.0

Output signal properties					
Name	Type	Description	Units	Nominal	ScaleFactor
H_2	Measured			0.0	1.0
C_1	Measured			0.0	1.0
C_2	Measured			0.0	1.0

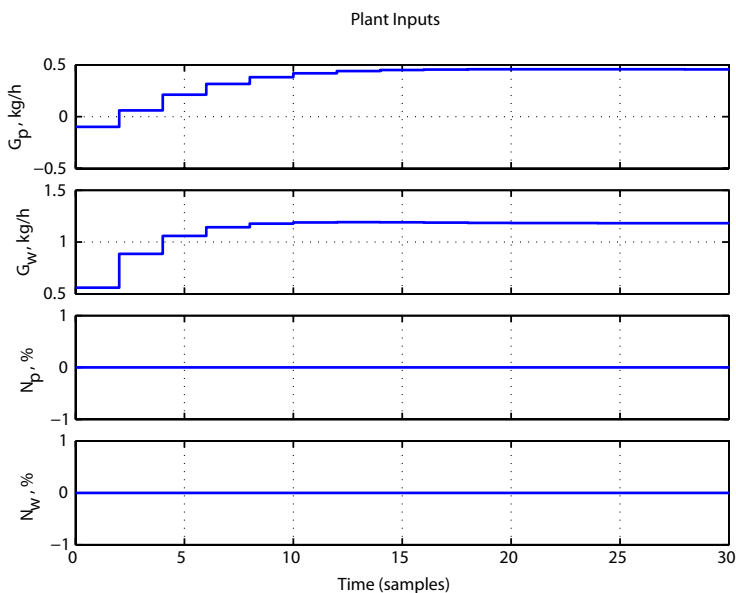
Фиг. 7.3 Описание на входно-изходните величини

Резултатите от симулацията с така настроеният предсказващ регулатор са представени на Фиг. 7.4. Както се вижда времето за достигане на желаната стойност на нивото в главния резервоар е около тринайсет минути.

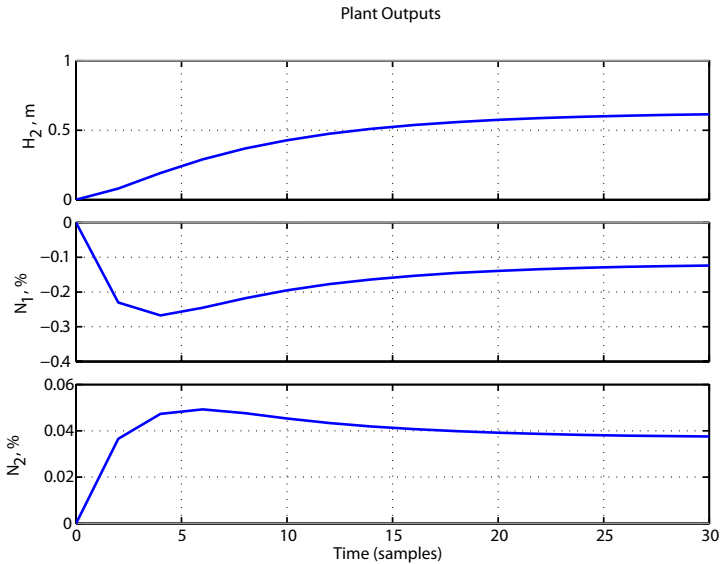


Фиг. 7.4 Преходни процеси на изходите на обекта при управление на променливите на главния резервоар -  $H_2$  (задание = 1 m) и  $C_2$  (задание = 0 %)

**Настройка на тегловните коефициенти.** Бързодействието може да се увеличи по няколко начина: ако се намали интервалът на управление, при намаляване на тегловните коефициенти на управляващите въздействия  $G_p$  и  $G_w$  и/или като се премахнат ограниченията на нарастване и намаляване. Както се вижда от Фиг. 7.4 плътността на сместа в захранващия резервоар  $C_1$  има статичната грешка, като отклонението е около 0.25 единици. Тъй като има само две управляващи въздействия, то е невъзможно да се поддържа желана стойност на трите изхода на системата.

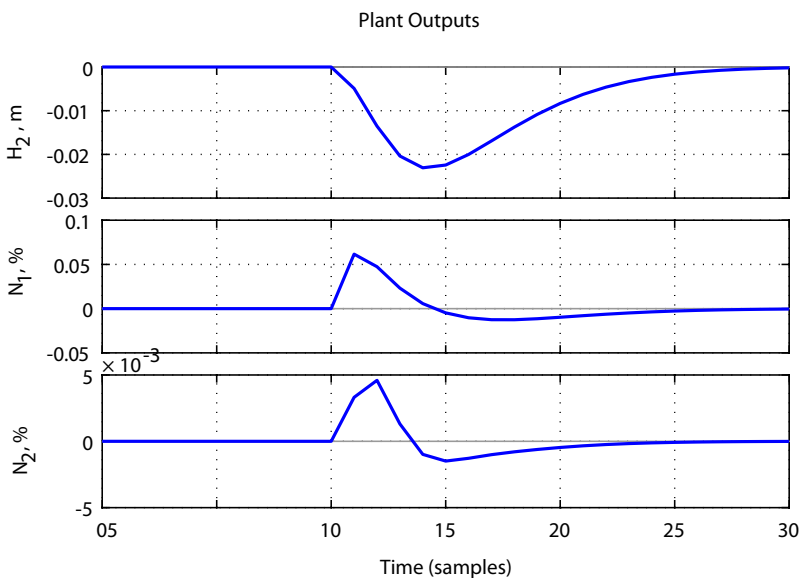


Фиг. 7.5 Преходни процеси на входа на обекта при управление променливите на главния резервоар -  $H_2$  (задание = 1 m) и  $C_2$  (задание = 0 %)



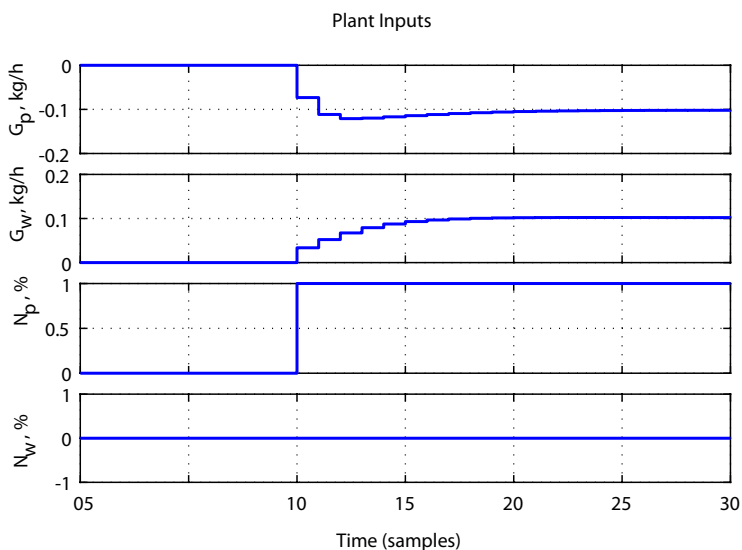
Фиг. 7.6 Преходни процеси на изходите на обекта при управление на величините в главния резервоар и при тегловен коефициент на  $C_1 = 0.5$

**Управление с права връзка (Feedforward control).** В този случай, всички задания за изходите на системата са равни на нула, и в момента  $t=10$  се появява измеримо смущение  $C_p$  (плътност на пулпа) с размер 1. Както се вижда от Фиг. 7.7, изходите  $C_2$  и  $H_2$  се отклоняват много малко от зададената стойност. На следващата фигура са представени преходните процеси по задание при същият режим на работа, но с активирана настройка **Look ahead**, при която се използват известни бъдещи стойности на смущението, при изчислението на управляващото въздействие. В този случай (Фиг. 7.9 и Фиг. 7.10) управляващите променливи започват да се променят още преди появата на смущението.



Фиг. 7.7 Преходни процеси на изходите на обекта по смущение -

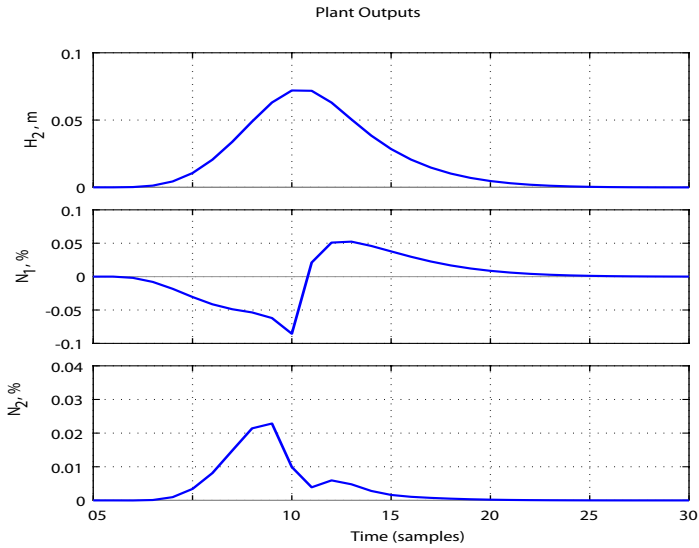
$$C_p = 1 \text{ в } t=10$$



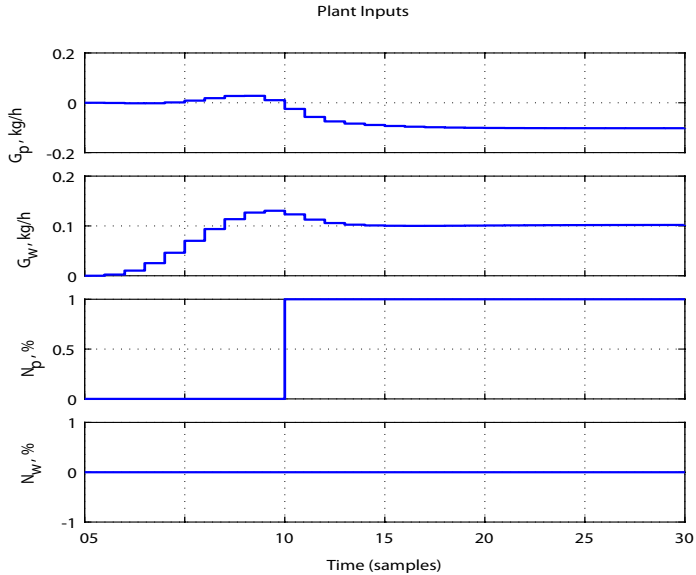
Фиг. 7.8 Преходни процеси на входовете на обекта по смущение -

$$C_p = 1 \text{ в } t=10$$





Фиг. 7.9 Преходни процеси по смущение  $C_p = 1$  в  $t = 10$  при предсказване на бъдещите стойности на смущението



Фиг. 7.10 Преходни процеси на входовете на обекта при предсказване на бъдещите стойности на смущението  $C_p = 1$  в  $t = 10$

По подразбиране смущенията на входа и изхода на обекта са от стъпаловиден вид, това позволява на регулатора да премахва отместването във всичките три изхода на системата. Значението на всяко смущение също е въведено по подразбиране, увеличавайки големината на едно смущение спрямо останалите, би увеличило значимостта на смущението в тази точка. Като трети параметър за настройка се явява измерването на шума. Големината на шума, определя влиянието на всяко измерване върху решението на предсказващия регулатор. Например, ако дадено измерване е относително зашумено, регулаторът ще му даде по-малко тегло, и вместо на него ще разчита на моделното предсказване на изхода.

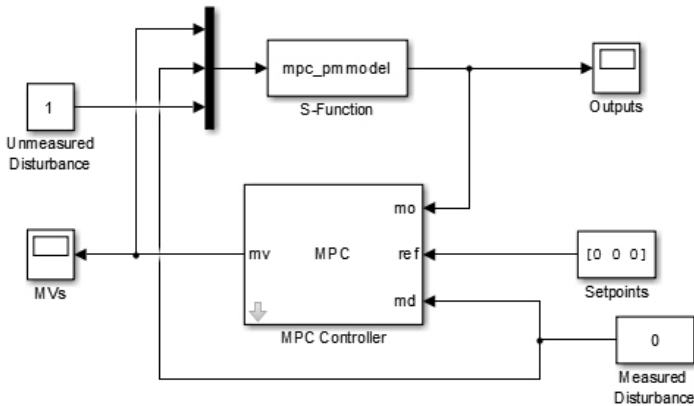


*Довършете изпълнението на Задача 3. Изследвайте влиянието на неизмеримото смущение, приложено на входа на обекта. Нека  $C_p = 0$  и  $C_w = 1$ .*

**Изпълнение за дача 4:** На Фиг. 7.11 е представена Simulink® диаграма на система с МПУ за управление на променливите на главния резервоар на агрегата за производство на хартия. Тя е достъпна чрез **open\_system('mpc\_papermachine')**. Блокът S-functions реализира и симулационния модел на обекта, който е описан в **mpc\_pmmode.m** (Табл .7.2).



*Довършете изпълнението на Задача 4, като симулирате системата с МПУ при наличие и отсъствие на смущаващи въздействия (измерими и неизмерими).*



Фиг. 7.11 Структурна схема в Simulink® на МПУ на агрегат за производство на хартия

Табл. 7.2 Симулационен модел на обекта - *mpc\_pmmodel.m*

---

```
function [sys,x0,xstr,TS]=mpc_pmmodel(t,x,u,flag,xp0)
```

```
% [sys,x0,xstr,TS]=mpc_pmmodel(t,x,u,flag,xp0)
```

```
% Simulink® representation of the paper machine process described % by Ying, Rao, and Sun, Chem. Eng. Communications, 1992. (See % also, Proceedings of American Control Conference, San Diego, % pp 1917, 1990). The model is bilinear. Using nomenclature in % the paper, process variables are:
```

```
% Manipulated variables: Gp, Gw
```

```
% Measured disturbance: Np
```

```
% Unmeasured disturbance: Nw
```

```
% Measured outputs: H2, N1, N2
```

```
% Unmeasured outputs: H1
```

```
% States: H1, H2, N1, N2
```

```
% Accepts standard Simulink® inputs for a system model.
```

```
%The model expects the input vector (u) to contain [Gp, Gw, Np, Nw] (in that order).
```

```
% The outputs will be H2, N1, N2 (in that order).
```

```
% Use optional parameter xp0 to initialize the state.
```

```
% The default initial condition is zero.
```

```
% Initialization
```

```
if nargin == 0
```

```

    flag=0;
end
if nargin < 5
    xp0=[];
end

if flag == 0

    if nargin == 4
        x0=zeros(4,1);
    elseif isempty(xp0)
        x0=zeros(4,1);
    else
        x0=xp0(:);
        if length(x0) ~= 4
            ctrlMsgUtils.error('MPC:utility:DemoPMMoDelInvalidXp0');
        end
    end
    sys=[4 0 3 4 0 0 1];
    xstr=['H1';'H2';'N1';'N2'];
    TS=[0 0];

    % state update if ABS(FLAG) == 1

elseif abs(flag) == 1
    A0=[-1.93 0 0 0; .394 -.426 0 0; 0 0 -.63 0; .82 -.784 .413 -.426];
    B0=[1.274 1.274;0 0;1.34 -.65;0 0];
    U=u(1:2,1); % Manipulated variables
    W=u(3:4,1); % Measured and unmeasured disturbance inputs.

    sys=A0*x+B0*U;
    sys(3)=sys(3)-.327*x(3)*sum(U)+[.203 .406]*W;

    % Output update if FLAG == 3.
elseif flag == 3
    iy=[2,3,4]; % Picks out correct states to use as output variables.
    sys=x(iy,1);

else
    sys=[];
end

```

---

## ТЕМА 8

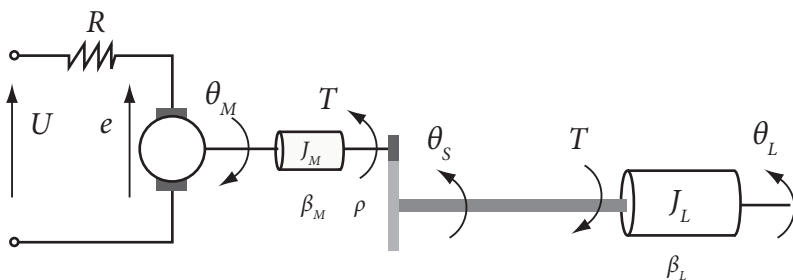
### Система с моделно предсказващо управление на сервомеханизъм

Целта на лабораторното упражнение е да се проектира система с моделно предсказващо управление за сервомеханизъм. Проектираната система следва да бъде изследвана в графичната среда на MPC Toolbox™ и средата MATLAB/Simulink®.

#### 1. Теоретична постановка

##### 1. 1. Описание на сервомеханизъма

Разглежданият позиционен сервомеханизъм се състои от постоянно-ков двигател, предавателна трансмисия с еластичен вал и товар (Фиг. 8.1).



Фиг. 8.1 Структурна схема на позиционен сервомеханизъм

Системата се описва със следната система диференциални уравнения:

$$\begin{aligned}\dot{\omega}_L &= -\frac{k_{\Theta}}{J_L} \left( \theta_L - \frac{\theta_M}{\rho} \right) - \frac{\beta_L}{J_L} \omega_L \\ \dot{\omega}_M &= \frac{k_T}{J_M} \left( \frac{U - k_T \omega_M}{R} \right) - \frac{\beta_M}{J_M} \omega_M + \frac{k_{\Theta}}{\rho J_M} \left( \theta_L - \frac{\theta_M}{\rho} \right)\end{aligned}\quad (8.1)$$

където  $U$  е захранващото напрежение,  $T$  е моментът на усукване върху товара,  $\omega_L$  е ъгловата скорост на товара,  $\omega_M$  е ъгловата скорост на вала, а останалите символи (Табл. 8.1) са коефициенти в уравнения(8.1).

Табл. 8.1 Параметри използвани в модела на сервомеханизъма

Параметър	Стойност (в SI единици)	Наименование
$k_\theta$	1280.2	Торзионна константа
$k_T$	10	Константа на двигателя
$J_M$	0.5	Инерционен момент на мотора
$J_L$	$50 J_M$	Инерционен момент на товара
$\rho$	20	Предавателно отношение
$\beta_M$	0.1	Коефициент на триене на мотора
$\beta_L$	25	Коефициент на триене на товара
$R$	20	Съпротивление на котвата

Ако се дефинират променливите на състоянието чрез вектора  $x_p = [\theta_L \ \omega_L \ \theta_M \ \omega_M]$ , то горният модел (8.1) може да се представи в пространството на състоянията:

$$\dot{x}_p = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k_\theta}{J_L} & -\frac{\beta_L}{J_L} & \frac{k_\theta}{\rho J_L} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k_\theta}{\rho J_M} & 0 & -\frac{k_\theta}{\rho^2 J_M} & -\frac{\beta_M + k_T^2/R}{J_M} \end{bmatrix} x_p + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{k_T}{R J_M} \end{bmatrix} V \quad (8.2)$$

$$y = \begin{bmatrix} \theta_L \\ T \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ k_\theta & 0 & -\frac{k_\theta}{\rho} & 0 \end{bmatrix} x_p \quad (8.3)$$

където  $y$  е изходната променлива.

## 1. 2. Цели на управлението и ограничения

Регулаторът трябва да поддържа ъгловата позиция на товара  $\theta_L$  на желаната стойност чрез изменение на захранващото напрежение  $U$ . В обратната връзка се подава измереното положение. Еластичният вал има ограничена напречна сила, така че моментът на усукване  $T$  трябва да бъде в определени граници:

$$|T| \leq 78.5 Nm \quad (8.4)$$

Освен това захранващото напрежение трябва да бъде в диапазона

$$|U| \leq 220 V \quad (8.5)$$

Обектът на управление се разглежда с един вход, който е управляващото въздействие на регулатора - напрежението  $U$ . Изходите са два (8.3) – първият се измерва и се подава в обратната връзка към регулатора ъгловото положение на товара  $\theta_L$ , а вторият не се измерва и това е моментът на усукване  $T$ .

## 2. Задачи за изпълнение.

**Задача 1.** Да се моделира системата с МПУ за позиционен сервомеханизъм. За създаване на модела на обекта на управление да се използват уравненията в пространството на състоянието (8.2)-(8.3).

**Задача 2.** Да се проектира предсказващ регулатор за системата с МПУ в графичната среда на MPC Toolbox™.

**Задача 3.** Да се изследва работата на системата с МПУ.

## 3. Оптична данни и резултати

**Изпълнение на задача 1:** Първата стъпка при проектиране на системата за управление е програмната реализация на модела на обекта на управление. За целта е подготвен файла **mpcmotormodel.m** (Табл. 8.2).

Табл. 8.2 *mpcmotormodel.m*:

```

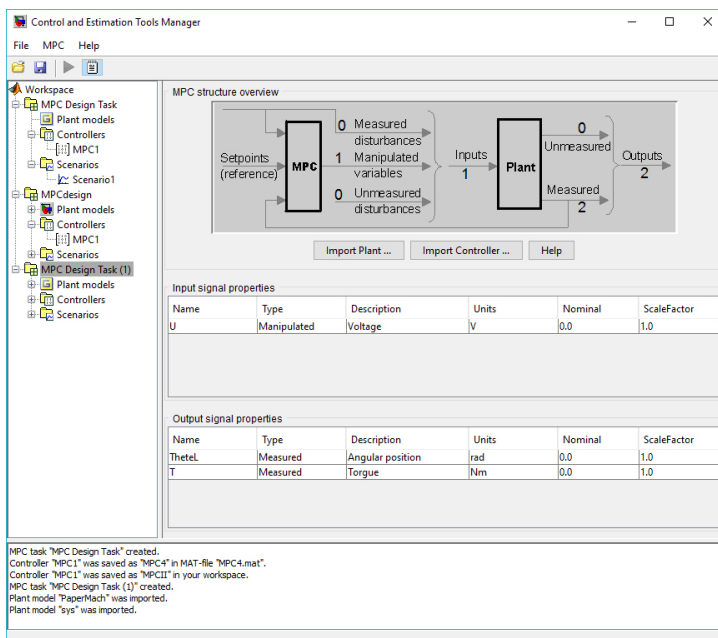
=====
% DC-motor with elastic shaft
%
%Parameters (MKS)
%-----
Lshaft=1.0; %Shaft length
dshaft=0.02; %Shaft diameter
shaftrho=7850; %Shaft specific weight (Carbon steel)
G=81500*1e6; %Modulus of rigidity
tauam=50*1e6; %Shear strength
Mmotor=100; %Rotor mass
Rmotor=.1; %Rotor radius
Jmotor=.5*Mmotor*Rmotor^2; %Rotor axial moment of inertia
Bmotor=0.1; %Rotor viscous friction coefficient (A CASO)
R=20; %Resistance of armature
Kt=10; %Motor constant
gear=20; %Gear ratio
Jload=50*Jmotor; %Load inertia
Bload=25; %Load viscous friction coefficient
Ip=pi/32*dshaft^4; %Polar momentum of shaft (circular) section
Kth=G*Ip/Lshaft; %Torsional rigidity (Torque/angle)
Vshaft=pi*(dshaft^2)/4*Lshaft; %Shaft volume
Mshaft=shaftrho*Vshaft; %Shaft mass
Jshaft=Mshaft*.5*(dshaft^2/4); %Shaft moment of inertia
JM=Jmotor;
JL=Jload+Jshaft;
Vmax=tauam*pi*dshaft^3/16; %Maximum admissible torque
Vmin=-Vmax;
%Input/State/Output continuous time form
%-----
A=[0 1 0 0;
  -Kth/JL -Bload/JL Kth/(gear*JL) 0;
  0 0 0 1;
  Kth/(JM*gear) 0 -Kth/(JM*gear^2) -(Bmotor+Kt^2/R)/JM];
B=[0;0;0;Kt/(R*JM)];
Cyd=[1 0 0 0];
Cvd=[Kth 0 -Kth/gear 0];
Dyd=0;
Dvd=0;
% Define the LTI state-space model
sys=ss(A,B,[Cyd;Cvd],[Dyd;Dvd]);
=====

```



Създаденият чрез програмата в Табл. 8.2 модел се използва при **изпълнение на задача 2**, а именно проектирането на предсказващ регулатор с графичната среда на MPC Toolbox™.

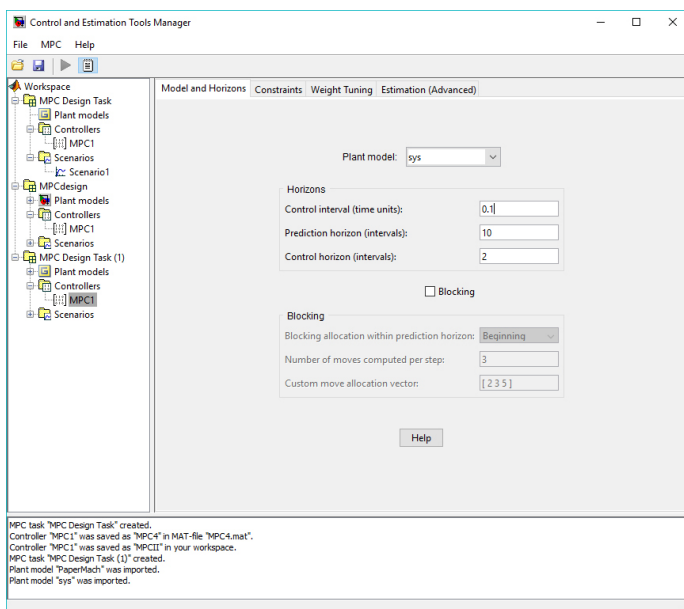
След импортиране на модела на сервомеханизъма, съхранен в променливата **sys**, се правят необходимите конфигурации: задаване на имена на променливи, уточняване на вида на сигналите (входни изходни) - Фиг. 8.2.



Фиг. 8.2 Графичен прозорец с въведения модел на обекта и описание на входно – изходните сигнали

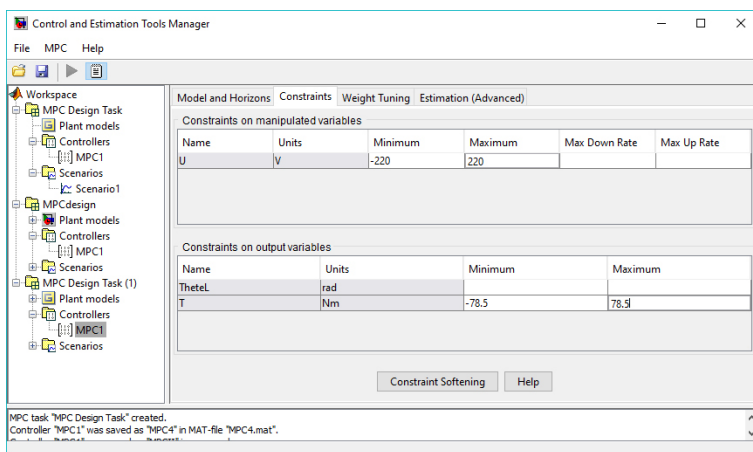
Наименованията на сигналите са избрани да бъдат **U** за управляващото въздействие; **ThetaL** и **T** за двата изхода. Вторият изход на сервомеханизма не се измерва. Ето защо той отбелязан като **Unmeasured** в съответното поле.

Следващата стъпка е задаване на параметрите на предсказващия регулатор. Периодът на дискретизация за конкретното изследване 0.1 времеви единици (Фиг. 8.3). Хоризонтите и продължителността на симулацията за сега остават непроменени.



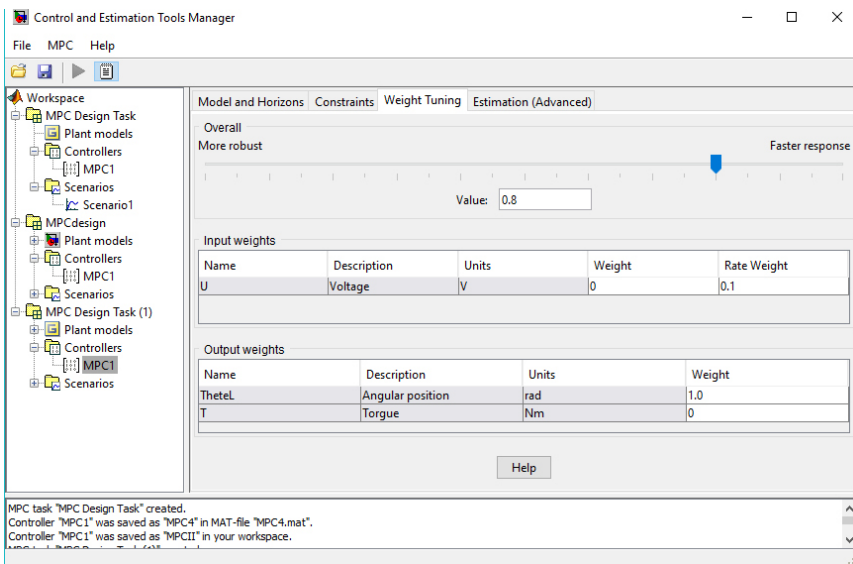
Фиг. 8.3 Прозорец с параметрите на регулатора – модел и хоризонти

Дефинираните чрез изрази (8.4) и (8.5) ограничения се въвеждат в раздела **Constraints** (Фиг. 8.4).



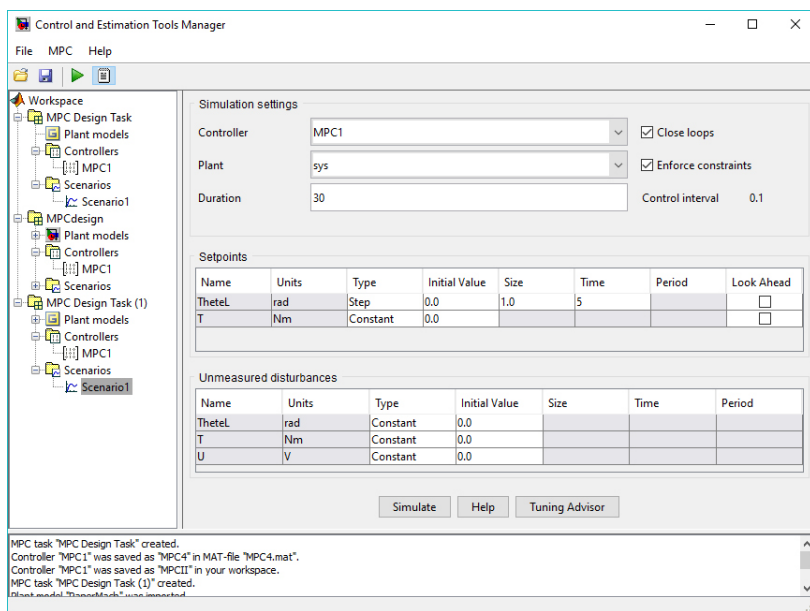
Фиг. 8.4 Прозорец за задаване на ограничения

При избора на тегловните коефициенти за входовете и изходите се взимат предвид спецификите на управлявания обект. Ъгловото положение трябва да следва заданието, а въртящият момент може да варира, при условие, че остава в рамките на зададените ограничения. Ето защо, тегловният коефициент на въртящия момент е избран да бъде равен на нула. Това го изключва от групата на управляваните изходи. По същия начин, приемливо е напрежението да се отклонява от номиналната си стойност (това е необходимо, за да се променя ъгловото положение). Тегловният му коефициент трябва да бъде равен на нула (по подразбиране за управляващи въздействия). От друга страна, не е желателно регулаторът да прави драстични промени в захранващото напрежение. Този ефект може да се контролира чрез тегловния коефициент на изменението на управляващото въздействие - **Rate weight** (въвежда се стойност 0.1) (Фиг. 8.5).



Фиг. 8.5 Прозорец за настройка на тегловни коефициенти

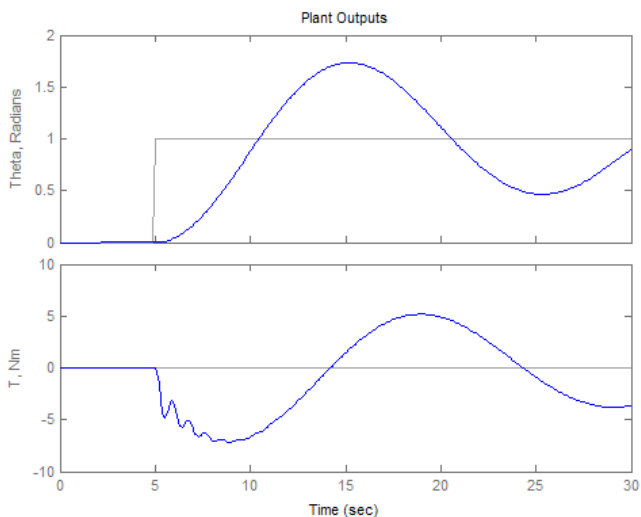
**Изпълнение на задача 3:** Параметрите на симулационния сценарий са показани на Фиг. 8.6. Изследва се реакцията на регулатора при стъплено изменение на заданието на ъгъла. Всички останали входове са оставени константни и с номинални стойности. По подразбиране всички смущаващи въздействия са изключени, т.е. равни на нула.



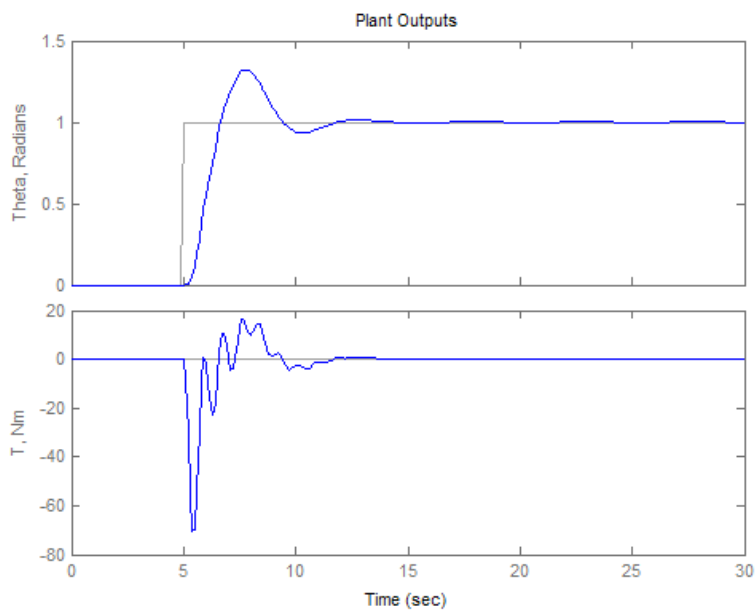
Фиг. 8. 6 Настройка на симулации чрез прозореца "Scenario1"

Получените резултати са представени на Фиг. 8.7. Вижда се, че реакцията е много бавна, и не се установява в рамките на 30 итерации (продължителността на симулацията). За сметка на това, въртящият момент остава в границите, при изменения на захранващото напрежение.

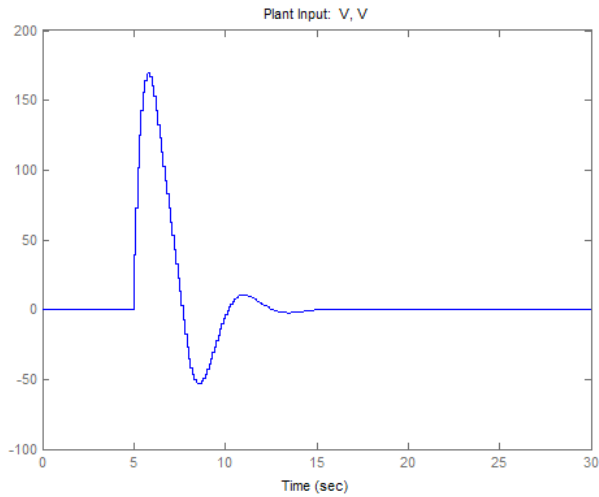
За получаване на по-бърз преходен процес се променя съотношението робастност-бързодействие в секцията **Overall** в раздела **Weight Tuning** (Фиг. 8.5). Следващите графики (Фиг. 8.8 и Фиг. 8.9). показват резултатите от изследванията при напълно преместен в дясно плъзгач, т.е. в посока бързодействие.



Фиг. 8.7 Преходен процес при стъпално изменение на заданието на ъгъла



Фиг. 8. 8 Преходен процес с по-бърза реакция



*Фиг. 8.9 Преходен процес на управляващото въздействие, отговарящ на Фиг. 8.8.*



*Симулирайте поведението на системата при изменение на заданието на ъгъла от 0 на  $\pi$  rad. Представете в графичен вид и коментирайте получените резултати.*

## ТЕМА 9

### Изследване влиянието на параметрите на предсказващия регулатор върху качеството на управление

Целта на лабораторното упражнение е изследване влиянието на параметрите на предсказващия регулатор върху качеството на управление. Изследванията включват поредица от симулационни изследвания, които се провеждат при различни стойности на параметрите на предсказващия регулатор. Това са хоризонтите на предсказване и управление; тегловните коефициенти на входните и изходните сигнали.

#### 1. Задачи за изпълнение.

**Задача 1.** Изследване на влиянието на параметрите на предсказващия регулатор за система с МПУ за агрегат за производство на хартия (Тема 7).

**Задача 2.** Изследване на влиянието на параметрите на предсказващия регулатор за система с МПУ за сервомеханизъм (Тема 8).

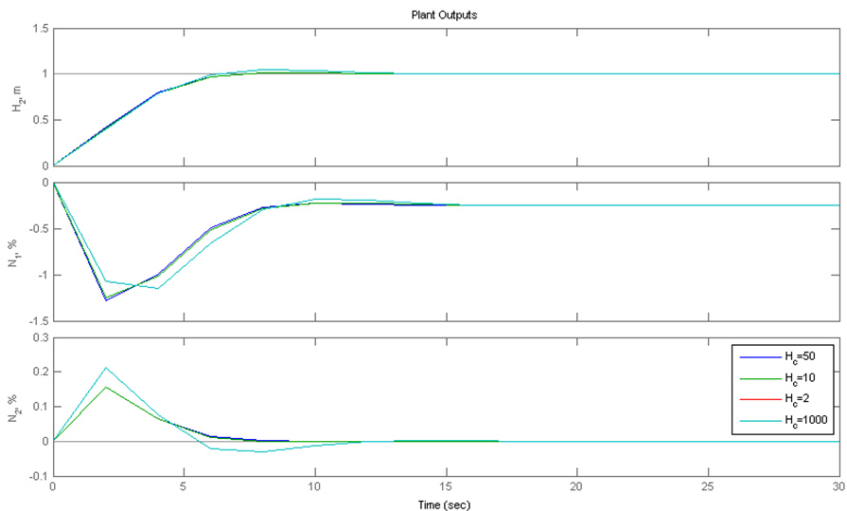
#### 2. Опитни данни и резултати.

**Изпълнение на задача 1:** С проектираната в Тема 7 система с МПУ за агрегат за производство на хартия са проведени 4 групи експерименти. Всеки от които изследва качеството на управление при промяна на един параметър, при фиксирани стойности за останалите параметри.

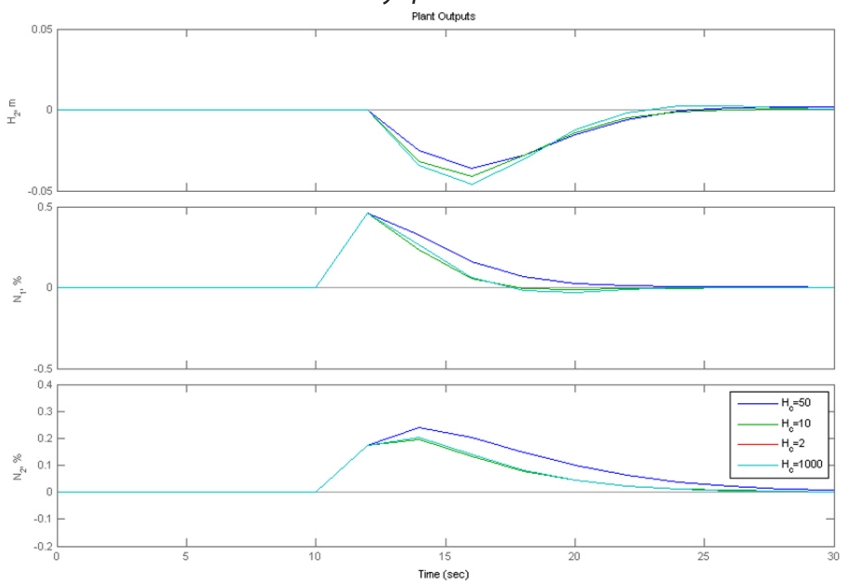
**Група 1:** Изследване на качеството на управление при промяна на хоризонта на управление. Останалите параметри са дефинирани като:

- **Модели и хоризонти:** хоризонт на предсказване = 10, интервал на управление = 2 минути;
- **Ограничения:** за  $G_p$  и  $G_w$ , минимум = -10, максимум = 10, минимална скорост = -2, максимална скорост = 2;
- **Тегловни коефициенти:** за  $G_p$  и  $G_w$  тегло = 0, скорост на теглото

= 0.4. За  $C_I$ , тегловен коефициент = 0. (Останалите изходи имат коефициенти = 1).



Фиг. 7.1 Преходни процеси по задание при промяна на хоризонта на управление



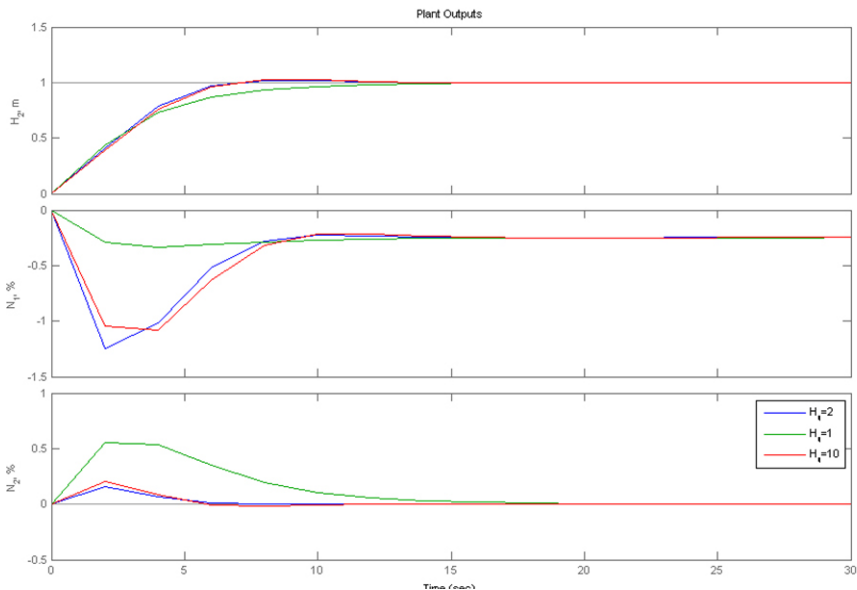
Фиг. 7.2 Преходни процеси по смущение при промяна на хоризонта на управление



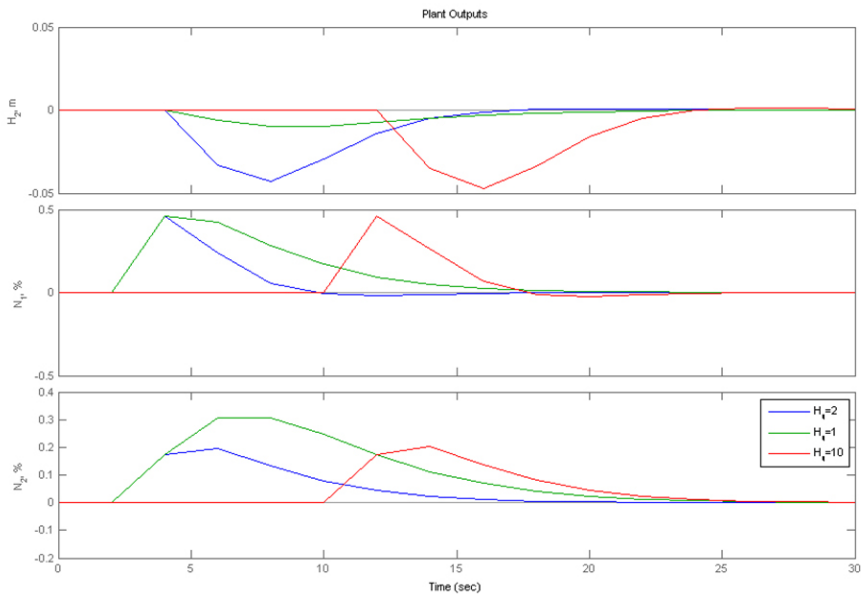
**Група 2:** С цел изследване влиянието на хоризонта на предсказване върху качеството на управление, регулатора е настроен със следните параметри:

- **Модели и хоризонти:** хоризонт на управление = 2, интервал на управление = 2 минути;
- **Ограничения:** за  $G_p$  и  $G_w$ , минимум = -10, максимум = 10, минимална скорост = -2, максимална скорост = 2;
- **Тегловни коефициенти:** за  $G_p$  и  $G_w$  тегло = 0, скорост на теглото = 0.4. За  $C_i$  тегловен коефициент = 0. (Останалите изходи имат коефициенти = 1).

При така настроеният предсказващ регулатор са получени преходните процеси на изходните по задание и по смущение, като заданието на нивото в главния резервоар  $H_2$  е постоянна величина със стойност 1. Резултатите от симулацията са представени на Фиг. 9.3 и Фиг. 9.4.



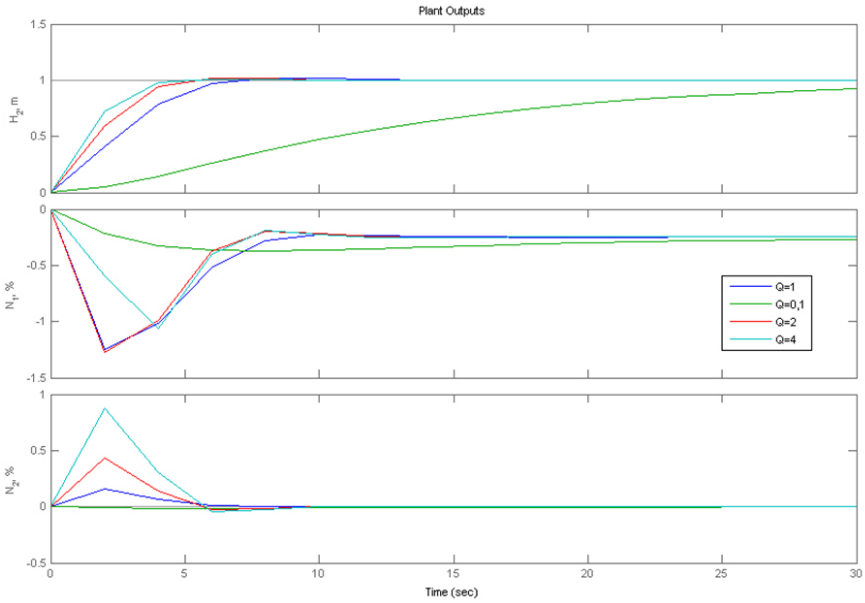
Фиг. 9.3 Преходни процеси по задание при промяна на хоризонта на предсказване  $N_{ii}$



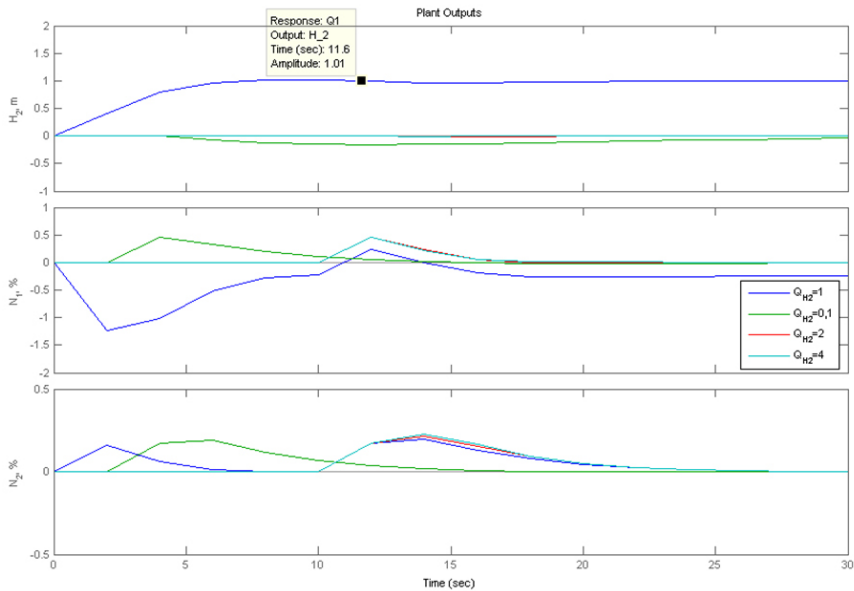
Фиг. 9.4 Преходни процеси по смущение при промяна на хоризонта на предсказване  $N_u$

**Група 3:** Влиянието на тегловният коефициент  $Q$  за изхода  $H_2$  е показано с преходните процеси по задание и по смущение, представени на Фиг. 9.5 и Фиг. 9.6. Регулаторът е настроен с:

- **Модели и хоризонти:** хоризонт на предсказване = 10, хоризонт на управление = 2, интервал на управление = 2 минути;
- **Ограничения:** за  $G_p$  и  $G_w$  минимум = -10, максимум = 10, минимална скорост = -2, максимална скорост = 2;
- **Тегловни коефициенти:** за  $G_p$  и  $G_w$  тегло = 0, скорост на теглото = 0.4. За  $C_I$  тегловен коефициент = 0.



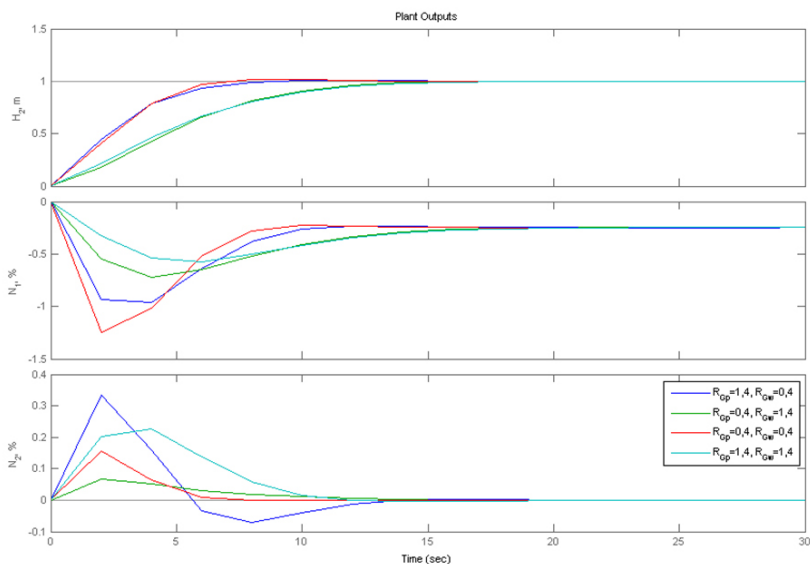
Фиг. 9.5 Преходни процеси по задание при промяна на тегловния коефициент на изхода на обекта  $Q_{H_2}$



Фиг. 9.6 Преходни процеси по смущение при промяна на тегловния коефициент на изхода на обекта  $Q_{H_2}$

**Група 4:** Влиянието на тегловните коефициенти на управляващите въздействия  $G_p$  и  $G_w$  е върху преходните процеси по задание и по смущение на изследваният обект е представено на Фиг. 9.7 и Фиг. 9.8. Регулаторът е настроен със следните стойности:

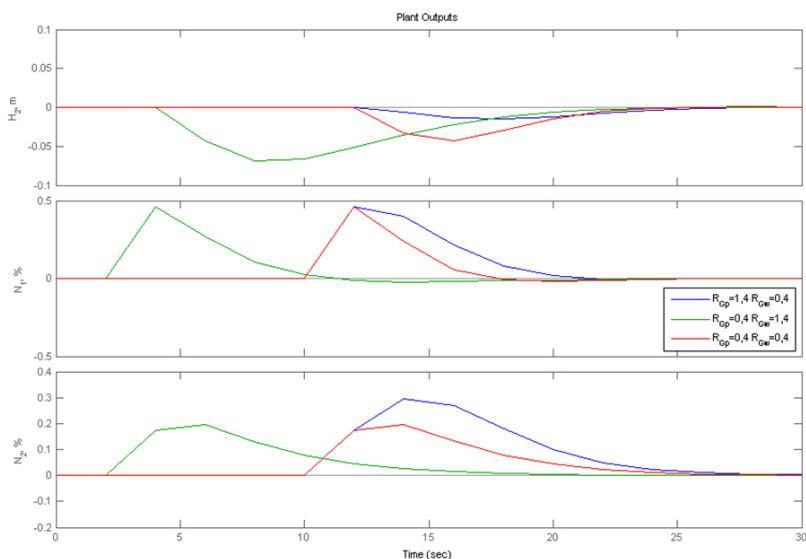
- **Модели и хоризонти:** хоризонт на предсказване = 10, хоризонт на управление = 2, интервал на управление = 2 минути;
- **Ограничения:** за  $G_p$  и  $G_w$ , минимум = -10, максимум = 10, минимална скорост = -2, максимална скорост = 2;
- **Тегловни коефициенти:** за  $G_p$  и  $G_w$  тегло = 0, скорост на теглото = 0.4. За  $N_1$  и  $H_1$ , тегловен коефициенти = 0; за  $N_2$  тегловен коефициент = 1.



Фиг. 9.7 Преходни процеси по задание при промяна на тегловните коефициенти на управляващите въздействия  $G_p$  и  $G_w$

В обобщение на получените резултати, може да се каже че чрез хоризонта на предсказване (началната му стойност) се компенсират чистите закъснения, но от друга страна големите стойности на хоризонта на предсказване утежняват задачата за изчисление на управлението. Затова се търси опти-

мална стойност на хоризонта на предсказване. Друг фактор, който оказва влияние върху работата на регулатора е интервалът на управление (равен на 2 времеви единици). Така настроеният предсказващ регулатор не може да отработи смущението, докато то не окаже влияние върху изходите на системата, т.е. в момента  $t=2$ . Бързодействието може да се увеличи по няколко начина: ако се намали хоризонтът на управление, при намаляване на тегловните коефициенти на управляващите въздействия  $G_p$  и  $G_w$ , и/или като се премахнат ограниченията на нарастване и намаляване.



Фиг. 9.8 Преходни процеси по смущение при промяна на тегловните коефициенти на управляващите въздействия  $G_p$  и  $G_w$



Изпълнете задача 2, като проведете 4 групи симулационни изследвания подобно на задача 1. Представете графично и коментирайте получените резултати.

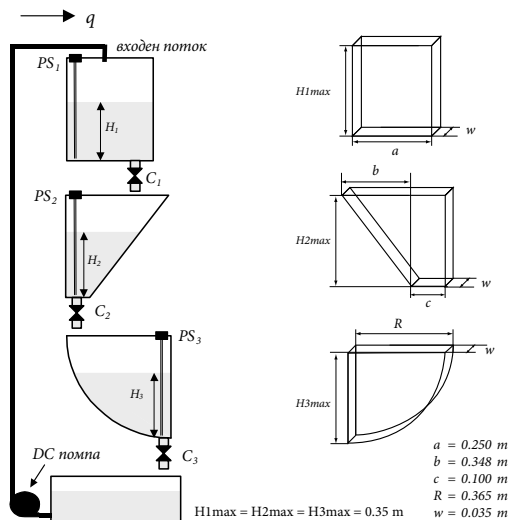
## ТЕМА 10

### Невронно-размито МПУ с модел в пространството на състоянието

Целта на лабораторното упражнение е да се изследва система с моделно предсказващо управление, използваща адаптивен модел на обекта на управление, реализиран чрез невронно-размита структура с механизъм за извеждане на Такаги-Сугено. Представен е симулационен пример на система с невронно-размито МПУ в средата на MATLAB®.

#### 1. Теоретична постановка

Предложените алгоритми за невронно-размито МПУ са изследвани и върху симулационен модел на лабораторен стенд – резервоар за вода (Intesco® MultiTank System) (Фиг. 10.1). Обектът се състои от три отделни резервоара за вода, свързани помежду си с регулируеми вентили (Intesco, 2009). Допълнителният резервоар най-отдолу събира оттичащата се вода. Само най-горният резервоар има константно напречно сечение, останалите имат променящо се сферично и конусовидно сечение.

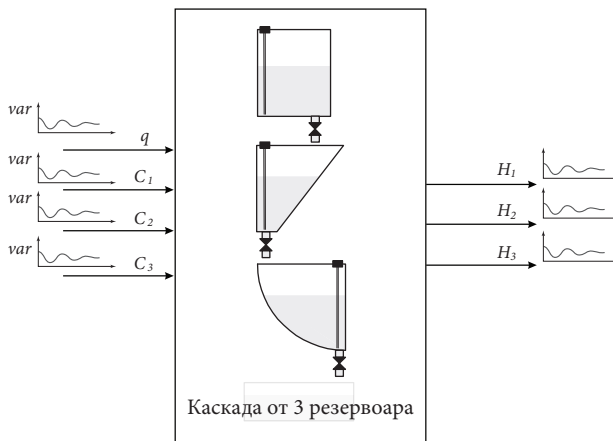


Фиг. 10.1. Схема на изследваната каскада от резервоари

Това създава допълнителна нелинейност в системата. Водата в системата постъпва в най-горния резервоар посредством помпа с регулируем дебит. Течността преминава към по-долните резервоари през свързващите клапи благодарение на силата на гравитацията. Сечението на вентилите  $C_1$ ,  $C_2$  и  $C_3$  може да бъде променяно, което води до различна скорост на оттичане на водата. Всеки от трите резервоара разполага със сензор за ниво ( $PS_1$ ,  $PS_2$ ,  $PS_3$ ).

## 2. Симулационни параметри

Изследваната система от резервоари има три изхода представени от нивата  $H_1$ ,  $H_2$ ,  $H_3$  на течността (водата) във всеки от трите резервоара (Фиг. 10.2). Управляващите въздействия са четири, а именно: входен дебит  $q$  и съпротивленията на сеченията на клапите  $C_1$ ,  $C_2$ ,  $C_3$  между отделните резервоари (Inteco, 2009).



Фиг. 10.2. Постановка на задачата за управление

Освен характерните за този клас обекти, изследваната система от три резервоара се отличава с допълнителни нелинейности групирани като (Inteco, 2009):

- Нелинейности породени от геометрията на резервоарите;
- Нелинейности свързани със системните ограничения (максимал-

но и минимално допустимо ниво);

- Нелинейности, дължащи се на динамиката на изтичане и геометрията на клапите;
- Нелинейности представени от характеристиките на помпата и на входно-изходните вентили (клапи).

**Ограничаващи условия за променливите.** Ограниченията спрямо изходите на обекта на управление са съобразени с геометричните размери на резервоарите. Така максималното ниво на течността е ограничено до 0.35 m, т.е. в сила е следното неравенство  $0 \leq H_i \leq 0.35 \text{ m}$ ,  $i=1,2,3$ . Това условие следва да бъде спазено при действащи ограничения за управляващите входове на обекта, а именно:

- Ограничения върху коефициента на съпротивление за всяка входно-изходна клапа  $0 \leq C_i \leq 2e-04$ ,  $i=1,2,3$
- Ограничения за входния поток  $0 \leq q \leq 1e-04 \text{ m}^3/\text{s}$

**Избор на входове за невронно-размития модел.** С цел конструиране на невронно-размития модел се дефинира регресионния вектор  $\mathbf{z}$  [1]. За конкретния случай той се състои от миналите стойности на входно-изходните променливи на системата (Фиг. 10.2). Всеки от елементите на така дефинирания вектор се размива от терм-множество от три Гаусови функции на принадлежност [1]. Благодарение на адаптивните свойства на модела на всяка стъпка от управлението МПУ разполага с актуален модел на системата от резервоари, осигурен от невронно-размитата структура.

**Параметри на симулационните изследвания.** Проведени са сравнителни експерименти, които невронно-размитото МПУ при наличие и отсъствие на ограничения с калсическия подход при, който моделът се създава еднократно преди стартиране на управлението. Simulink® диаграмите на експеримента са представени в [1]. Параметрите на проведените изследвания са представени в Табл. 10.1.



Табл. 10.1. Симуляционни параметри за обекта

Параметри на МПУ	Параметри на системата	Симуляционни параметри
<p><b>Хоризонти:</b></p> $H_p = 10$ $H_u = 3$ $H_w = 1$	<p><b>Коефициенти на оттичане за всеки резервоар:</b></p> <p>при инициализация:</p> $\alpha_1 = 0.5 \quad \alpha_2 = 0.5$ $\alpha_3 = 0.5$	<p><b>Време за симулация:</b></p> $t = 600 \text{ s}$
<p><b>Тегловни матрици:</b></p> $Q = 1e-02 \text{diag}(1.1, 1)$ $R = 10e04 \text{diag}(1.1, 1.1)$	<p>в симуляционния модел: <math>\alpha_1 = 0.29</math>  <math>\alpha_2 = 0.2256</math>  <math>\alpha_3 = 0.2487</math></p> <p><b>Начални условия:</b></p> <p>1) <math>H_0 = [0, 0, 0]</math>                  2) <math>H_0 = [0.05, 0.25, 0.15]</math></p>	<p><b>Такт на дискретизация:</b></p> $T_s = 1 \text{ s}$

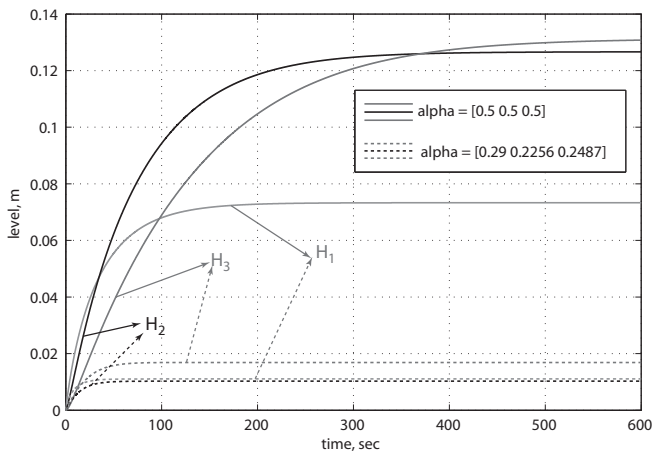
Линеаризираният модел (10.1) се използва за начална инициализация на невронно-размития модел, който участва като предиктор в предложения подход за предсказващо управление. Линеаризирания динамичен модел на каскадната система от резервоари е представен чрез описание в пространството на състоянията, където матриците  $A$ ,  $B$ ,  $C$  и  $D$  имат следния вид (Petrov et al., 2009):

$$A = \begin{bmatrix} \frac{-C_1\alpha_1}{awH_1^{1-\alpha_1}} & 0 & 0 \\ \frac{C_1\alpha_1}{w(c+b\frac{H_2}{H_{2\max}})H_1^{1-\alpha_1}} & \frac{-C_2\alpha_2}{w(c+b\frac{H_2}{H_{2\max}})H_2^{1-\alpha_2}} & 0 \\ 0 & \frac{C_2\alpha_2}{w\sqrt{R^2-(H_{3\max}-H_3)^2}H_2^{1-\alpha_2}} & \frac{-C_3\alpha_3}{w\sqrt{R^2-(H_{3\max}-H_3)^2}H_3^{1-\alpha_3}} \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{1}{aw} & \frac{-1}{awH_1^{1-\alpha_1}} & 0 & 0 \\ 0 & 0 & \frac{-1}{w(c+b\frac{H_2}{H_{2\max}})H_2^{1-\alpha_2}} & 0 \\ 0 & 0 & 0 & \frac{-1}{w\sqrt{R^2-(H_{3\max}-H_3)^2}H_3^{1-\alpha_3}} \end{bmatrix} \quad (5.16)$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad D = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (10.1)$$

Влиянието на коефициентите на оттичане  $\alpha_i, i=1,2,3$  за всеки резервоар върху динамиката на каскадата от резервоари е значително. Това може да се отчете в матриците на линеаризирания модел (10.1). Най-ясна представа обаче дава Фиг. 10.3, на която са показани преходните функции при различни коефициенти на оттичане.

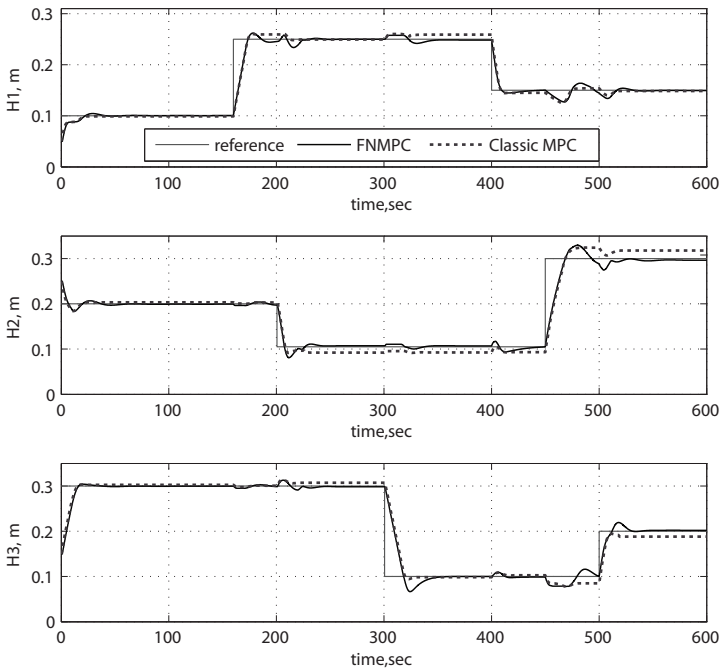


Фиг. 10.3. Преходна функция на обекта на управление при различни коефициенти на оттичане

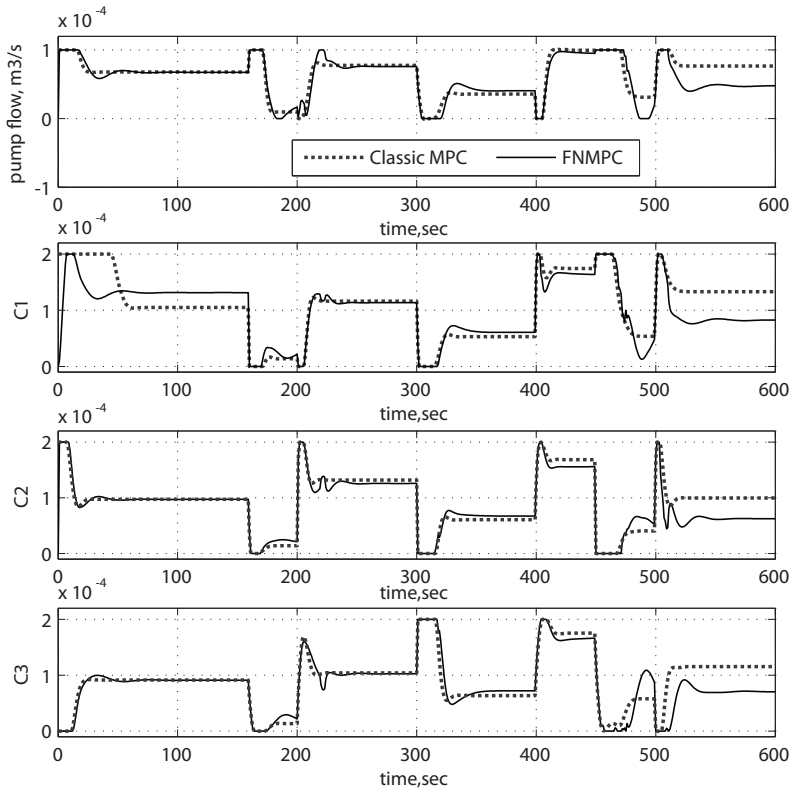
В повечето случаи коефициентите на оттичане при реалните системи не могат да се определят с точност. Това в повечето случаи поражда неточности и в управлението, най-вече когато се разчита на подходи с линеаризация. За да се покаже предимството на адаптивните алгоритми в подобни ситуации, са проведени няколко експеримента с различни  $\alpha$ , при инициализация и симулация.

### 3. Симулационни резултати

Изследването е проведено с цел да се покаже необходимостта от адаптивен модел в постановката на МПУ. Сравнени са класически алгоритъм за МПУ (Classic MPC) и предложеният в [1] подход за невронно-размито МПУ (FNMPС). Класическият предсказващ регулатор използва линеаризиран модел (10.1) на обекта на управление, който не се променя в хода на предсказващото управление. Недостатъчно точният модел в този случай би довел до некоректни решения на задачата за МПУ.

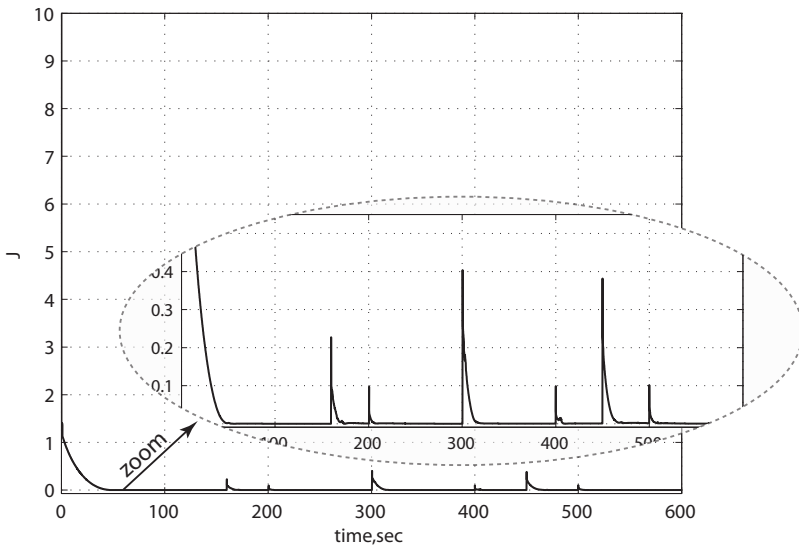


Фиг. 10.4. Управление на нивото в трите резервоара чрез Classic MPC и FNMPС



Фиг. 10.5. Управляващи сигнали получени при Classic MPC и FNMPC

С цел симулиране на подобна ситуация е проведен експеримент, при който началната линеаризация (и инициализация) е направен при  $\alpha_1 = 0.5$ ,  $\alpha_2 = 0.5$ ,  $\alpha_3 = 0.5$ , а заложените в симулационния модел коефициенти са  $\alpha_1 = 0.29$ ,  $\alpha_2 = 0.2256$ ,  $\alpha_3 = 0.2487$ . Фиг. 10.4 – Фиг. 10.7 показват резултатите от този експеримент за МПУ с условна оптимизация.



Фиг. 10.6. Стойности на целевия критерий в задачата за условна оптимизация

Преимуществото на предложения подход за МПУ използващ невронно-размит модел на обекта на управление е видимо (Фиг. 10.4). Актуализацията на модела във всеки такт на управление спомага за компенсиране на неопределеностите възникнали в следствие на неточността при определяне на коефициентите  $\alpha_i$ ,  $i=1,2,3$ . Грешката при регулиране е оценена както чрез интегралната ѝ стойност (IE, **Integral of Error** (10.2)), така и от интеграла на квадратичната (ISE, **Integral of Squared Error** (10.3)) и абсолютната (IAE, **Integral of Absolute Error** (10.4)) ѝ стойности (Табл. 10.2):

$$e_{ISE} = \int_0^T e(t) dt \quad (10.2)$$

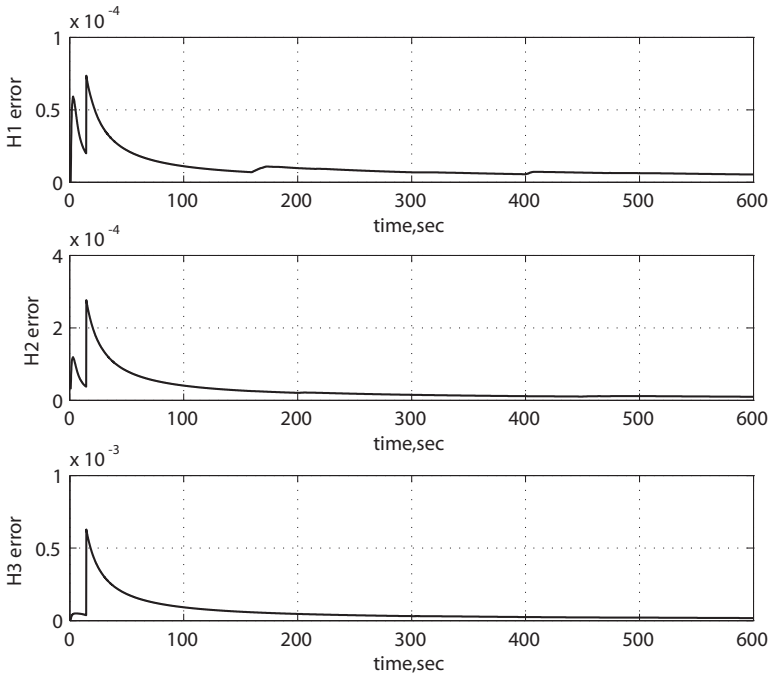
$$e_{ISE} = \int_0^T \{e(t)\}^2 dt \quad (10.3)$$

$$e_{IAE} = \int_0^T |e(t)| dt \quad (10.4)$$

Табл. 10.2. Стойности на IE, ISE и IAE

оценка	$e_{IE}$		$e_{ISE}$		$e_{IAE}$	
	Classic MPC	FNMPCC	Classic MPC	FNMPCC	Classic MPC	FNMPCC
$H_1$	-25.9632	14.7101	2.6020	1.4980	54.6668	34.5650
$H_2$	-60.9176	13.5566	9.1921	2.5583	111.1213	45.6645
$H_3$	-61.4731	3.3568	17.4127	4.2064	125.5518	55.4842

Точността на алгоритъма за обучение на параметрите на невронно-размитата мрежа, използвана за моделиране на обекта на управление, е изследвана чрез стойностите средноквадратичната грешка (MSE) от идентификацията (Фиг. 10.7).



Фиг. 10.7. Изменение на стойностите на MSE на грешката при моделиране

## Използвани литературни източници

1. **Ахмед, С.** (2013) Системи с невронно-размито предсказващо управление, (дисертационен труд)
2. **Велев, К.** (1995) Адаптивни системи за управление, София
3. **Гарипов, Е.** (2007) Идентификация на системи. Част II. Идентификация чрез дискретни стохастични регресионни модели, Издателство на ТУ-София, ISBN 978-594-438-710-5
4. **Ищев, К.** (2000) Теория на автоматичното управление, КИНГ, ISBN 954-9518-07-8
5. **Ahmed, S.,** Petrov, M., Ichtev, A. (2010) Fuzzy Model-Based Predictive Control Applied to Multivariable Level Control of Multi Tank System. Proceedings of 2010 IEEE International Conference on Intelligent Systems (IS 2010), London, UK. pp. 456–461
6. **Babuška, R.,** Sousa, J., Verbruggen, H. (1999) Predictive control of nonlinear systems based on fuzzy and neural models, Proc. ECC'99, pp. CA-pp. 12–4
7. **Bemporad A.,** Morari, M.N., Ricker, R. (2014) Model Predictive Control Toolbox™ User's Guide, Mathworks
8. **Camacho, E.F.,** Bordons, C. (2004) Model Predictive Control, Springer-Verlag, London
9. **Clarke, D.W.,** Mohtadi, C., Tus, S. (1987) Generalized predictive control. The basic algorithm. Automatica 23, pp. 137–148
10. **Daraoui, N.,** Dufour, P., Hammouri, H., & Hottot, A. (2010) Model predictive control during drying stage of lyophilisation. Control Engineering Practice, 18(2010), pp. 483–494
11. **Findeisen, W.** (1997) Control Structures for Complex Processes. Warsaw: Warsaw University of Technology Press, (in Polish)
12. **Findeisen, W.,** Bailey, F.N., Brdys, M., Malinowski, K., Tatjewski, P. and Wozniak,

- A. (1980) *Control and Coordination in Hierarchical Systems*. Chichester: Wiley
13. **Gruber, J.K.**, Ramirez, D.R., Alamo, T., Bordons, C., Camacho, E.F.(2009) Control of a pilot plant using QP based min-max predictive control. *Control Engineering Practice*. 2009; 17(11), pp. 1358–66
  14. **Hassan, M.**, Dubay, R.,&Wang, C.L.R. (2007) Active vibration control of a flexible one-link manipulator using a multivariable predictive controller. *Control Engineering Practice*, 17, pp. 311–323
  15. **Kale, M.M.**, Chipfield, A.J. (2005) Stabilized MPC Formulations for Robust Reconfigurable Flight Control. *Control Engineering Practice*, 13, pp. 771–788
  16. **Lee, J.H.**, (2000) Modeling and Identification for Nonlinear Model Predictive Control: Requirements, Current Status and Future Research Needs, *Nonlinear Model Predictive Control Progress in Systems and Control Theory*, Volume 26, 2000, pp. 269–293
  17. **Pannocchia, G.**, Laurino, M., Landi, A.A. (2010) Model predictive control strategy toward optimal structured treatment optimal structured treatment interruptions in anti-HIV therapy. *IEEE Transactions on Biomedical Engineering*, 57(5), pp. 1040–1050.
  18. **Pearson, R.K.**, (2003) Selecting Nonlinear Model Structures for Computer Control: Review, *Journal of Process Control*, vol. 13, pp. 1–26(26), DOI: [http://dx.doi.org/10.1016/S0959-1524\(02\)00022-7](http://dx.doi.org/10.1016/S0959-1524(02)00022-7)
  19. **Qin, S.J.**, and Badgwell, T.A. (2003) A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7), pp. 733–764
  20. **Tatjewski, P.**, Ławryńczuk, M. (2006) Soft computing in model-based predictive control, *Int. J. Appl. Math. Comput. Sci.*, Vol. 16, No. 1, pp. 7–26
  21. **Yoon, Y.**, Shin, J., Kim, H.J., Park, Y., Sastry, S. (2009) Model-predictive active steering and obstacle avoidance for autonomous ground vehicles, *Control Engineering Practice*, Volume 17, Issue 7, July 2009, Pages 741-750, ISSN 0967-0661, 10.1016/j.conengprac.2008.12.001