Adaptivni sustav upravljanja temeljen na podacima

Matijević, Petra

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:168:162384

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2025-03-23**

Repository / Repozitorij:

FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repozitory

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 87

ADAPTIVNI SUSTAV UPRAVLJANJA TEMELJEN NA PODACIMA

Petra Matijević

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 87

ADAPTIVNI SUSTAV UPRAVLJANJA TEMELJEN NA PODACIMA

Petra Matijević

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

DIPLOMSKI ZADATAK br. 87

Pristupnica:	**Petra Matijević (0036524045)**
Studij:	Informacijska i komunikacijska tehnologija
Profil:	Automatika i robotika
Mentorica:	izv. prof. dr. sc. Tamara Petrović

Zadatak: **Adaptivni sustav upravljanja temeljen na podacima**

Opis zadatka:

Za upravljanje nelinearnim procesima sve se više koriste metode koje se zasnivaju na podacima, a ne isključivo na temelju matematičkog modela procesa. Ovo je posebno izraženo ako je proces promjenjiv te je potrebno prilagođavati parametre, a sam model sustava je nepoznat. Razmatrani proces je hlađenje prostora, pri čemu je upravljačka veličina brzina vrtnje ventilatora za hlađenje. Kao poremećaj može djelovati otvaranje i zatvaranje vrata ili prisutnost grijaćih elemenata u prostoru. Prvi je korak implementacija i upoznavanje s modelom koji opisuje proces i dostupan je u literaturi, a koji će poslužiti kao izvor podataka za sintezu adaptivnog regulatora. Potom je potrebno proučiti dva algoritma upravljanja i odabrati onaj primjereniji zadanom procesu. Prva metoda je DeepPC koja se temelji na tzv. ponašajnoj teoriji sustava. Druga metoda uključuje primjenu neuronske mreže za modeliranje sustava te primjenu optimalnog upravljanja. Odabranu metodu upravljanja na temelju podataka implementirati u Matlabu te validirati na zadanom procesu. Usporediti upravljanje s klasičnim upravljanjem primjenom PID regulatora.

Rok za predaju rada: 28. lipnja 2024.

Sadržaj

1. Uvod 1
2. Matematički model procesa 3
2.1. Diferencijalne jednadžbe 3
2.2. Shema matematičkog modela u Simulinku 5
3. Neuronska mreža procesa 9
3.1 Osnove neuronskih mreža 9
3.2 Matlab aplikacija za identifikaciju procesa 13
4. Prediktivno upravljanje 18
4.1 Modelsko prediktivno upravljanje 18
4.2 Prediktivno upravljanje temeljeno na podacima 20
5. Rezultati 23
5.1 Dinamika sustava bez upravljanja 23
5.2 Performanse NARX modela 25
5.3 Upravljanje temperaturom primjenom MPC regulatora 26
5.4 Usporedba MPC i PID regulatora 30
5.5 Uključivanje šuma mjerenja i primjena PT1 filtra 35
6. Zaključak 37
Literatura 38
Sažetak 39
Summary 40

1. Uvod

Kako sustavi postaju sve složeniji, tehnike koje se temelje na matematičkom modelu procesa postaju manje praktične. Alternativa tome su tehnike temeljene na podacima. Industrije temeljene na podacima, poput umjetne inteligencije zahtijevaju uređaje za pohranu velikog kapaciteta i računalne sustave visokih performansi.

Podatkovni centri su objekti sastavljeni od umreženih računala, storage sustava i računalne infrastrukture koje kompanije upotrebljavaju za čuvanje, obradu i dijeljenje velikih količina podataka. Kapaciteti podatkovnih centara stalno se povećavaju kako bi zadovoljili rastuće potrebe korisnika. Velike tehnološke kompanije poput Googlea, Facebooka, Amazona i Microsofta imaju globalne mreže podatkovnih centara raspoređenih diljem svijeta radi bolje dostupnosti i otpornosti na kvarove.

Kako podatkovni centri moraju raditi bez prekida danas su među najvećim svjetskim korisnicima električne energije. Oko 50-60% energije u je potrošeno na rad poslužitelja, uključujući njihovu obradu, pohranu podataka i mrežnu komunikaciju. Hlađenje poslužitelja i održavanje optimalne temperature u podatkovnom centru troši 30-40% energije. Ostatak energije troši se na neprekidno napajanje, osvjetljenje itd. Smanjenje potrošnje energije u podatkovnim centrima postaje sve važnije jer velika potrošnja ima značajan ekološki utjecaj, uključujući emisije stakleničkih plinova i potrošnju prirodnih resursa. Također, energija predstavlja jedan od najvećih troškova, pa povećanje energetske učinkovitosti može značajno doprinijeti smanjenju tih troškova i poboljšanju profitabilnosti poslovanja. Osim toga, manja potrošnja energije znači manje topline koju treba odvoditi iz centra, što može smanjiti rizik od pregrijavanja opreme. Međutim, povećanje energetske učinkovitosti postojećih podatkovnih centara je teško budući da mnogi centri koriste stariju opremu te je zamjena te opreme s novom i učinkovitijom skupa i duga. Također, podatkovni centri su složeni sustavi s raznim komponentama i poboljšanje energetske učinkovitosti može biti zahtjevno za implementaciju. Iz tih razloga potrebna je metoda koja je lako primjenjiva i zahtjeva minimalne izmjene hardvera.

Cilj ovoga diplomskog rada je implementirati sustav prediktivnog upravljanja temeljenog na podacima za upravljanje temperature. U posljednje vrijeme zabilježen je značajan napredak u integraciji strojnog učenja s modelskim prediktivnim upravljanjem (MPC) pod različitim uvjetima neizvjesnosti [4]. Ovi pristupi obuhvaćaju učenje modela iz podataka sustava te razvoj učinkovitih aproksimacija za online optimizaciju, što omogućuje poboljšanu izvedbu upravljačkih sustava u složenim okruženjima. Nadalje, metodologija upravljanja temeljena na podacima, poput DeePC algoritma, pokazala je učinkovito praćenje optimalnih trajektorija nepoznatih sustava uz osiguravanje zadovoljavanja zadanih ograničenja [1]. Oslanjajući se na ove koncepte, slično kao u radu [2], gdje je razvijen adaptivni model umjetne neuronske mreže za predikciju temperature i potrošnje energije sustava hlađenja, ovaj rad koristi neuronsku mrežu u kombinaciji s prediktivnim upravljanjem (MPC) za precizno upravljanje temperature. Kao i u spomenutom radu, matematičko modeliranje i prikupljanje podataka iz tog modela temelj su za treniranje i optimizaciju upravljačkog algoritma. Slični pristupi su već dokazali svoju učinkovitost u različitim inženjerskim poljima, gdje primjena MPC-a omogućuje kontrolu sustava koje je teško upravljati klasičnim regulatorima [3].

U sljedećem poglavlju ovog rada predstavljene su diferencijalne jednadžbe matematičkog modela koje opisuju termodinamičko ponašanje prostora. Ovaj model služi kao temelj za daljnje analize i simulacije. Treće poglavlje opisuje korištenje umjetnih neuronskih mreža za modeliranje nelinearnih odnosa unutar sustava. Ovdje se detaljno razmatra proces treniranja mreže na temelju podataka dobivenih iz matematičkog modela. U četvrtom poglavlju opisana je ideja metode prediktivnog upravljanja i razmatra se njena primjena na sustav. Također je opisan način korištenja prediktivnog regulatora zasnovanog na podacima. U petom poglavlju prikazani su rezultati simulacija u različitim scenarijima. Analizirani su utjecaji promjena ulaznih parametara na izlaznu temperaturu i uspoređuju se performanse prediktivnog regulatora i klasičnog PID regulatora. Konačno, u šestom poglavlju sažeti su ključni zaključci rada, istaknute su prednosti korištenih metoda, te su predloženi smjerovi za daljnje istraživanje i unapređenje sustava upravljanja.

2. Matematički model procesa

Matematički model procesa je opis fizičkog sustava pomoću varijabli, parametara, početnih ili graničnih uvjeta i jednadžbi ili nejednadžbi koji omogućuje bolje razumijevanje, optimiziranje ili predviđanje njegovih stanja ili svojstava. Ovaj model omogućuje simulaciju i analizu ponašanja sustava pod različitim uvjetima bez potrebe za stvarnim eksperimentima.

2.1. Diferencijalne jednadžbe

Matematički model prostora, opisan diferencijalnim jednadžbama, pruža temelj za razumijevanje i predviđanje dinamike temperature unutar prostora. Točan model omogućuje generiranje velikog broja simuliranih podataka koji realno odražavaju ponašanje sustava u različitim uvjetima.

Model sustava opisan je s pomoću sljedećih diferencijalnih jednadžbi:

$$(T_{out})' = \left[m' * c_p * \left(T_{in} - T_{out}\right) + k * \frac{A}{\Delta x} * (T_s - T_{out})\right] * \frac{1}{k_1} \qquad (2.1)$$

$$(T_s)' = [P - k * A * (T_s - T_{out})] * \frac{1}{k_2} \qquad (2.2)$$

Pri čemu je:

T_{in} ulazna temperatura (temperatura zraka koji ulazi u prostor) [°C],

T_{out} izlazna temperatura [°C],

T_s temperatura servera [°C],

m maseni protok [kg/s],

c_p specifični toplinski kapacitet zraka [J/kg°C],

k koeficijent toplinske provodljivosti aluminija [W/m°C],

A površina kroz koju se odvija prijenos topline (površina servera) [m^2],

Δx udaljenost na kojoj se prenosi toplina [m],

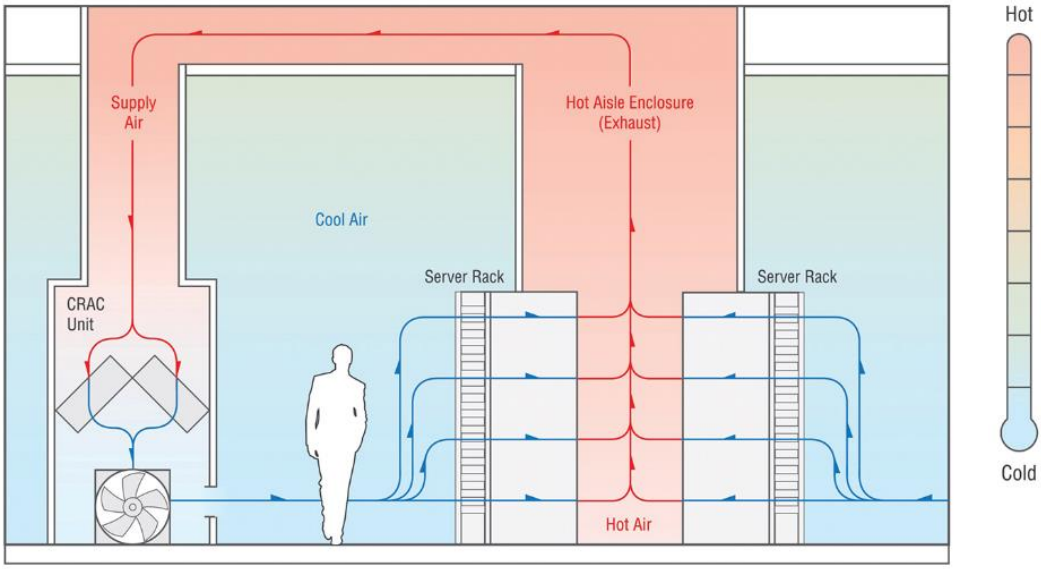
k_1 koeficijent brzine promjene izlazne temperature,

P snaga [W],

k_2 koeficijent brzine promjene temperature servera

Prva jednadžba opisuje promjenu izlazne temperature u vremenu pri čemu k_1 određuje brzinu te promjene. Što je k_1 manji temperatura će se brže mijenjati. Na desnoj strani jednakosti m' odnosno maseni protok i c_p odnosno specifični toplinski kapacitet množimo s razlikom u temperaturi između ulaznog i izlaznog zraka. Sve dok je ulazna temperatura manja od temperature prostora ovaj dio jednakosti imat će negativan predznak te tako smanjivati izlaznu temperaturu. Drugi član unutar zagrade predstavlja gubitak topline servera kroz materijal s koeficijentom toplinske provodljivosti k, površinom A na udaljenosti Δx na temperaturnoj razlici $T_s - T_{out}$. Budući da je to toplina koju server predaje okolini ovaj član će povećavati izlaznu temperaturu.

Druga diferencijalna jednadžba opisuje promjenu temperature servera T_s s vremenom. Od snage P koju generira server oduzima se toplina koju server predaje okolini te se zatim množi s recipročnom vrijednosti od k_2 kako bi se regulirala brzina promjene te temperature. U prikazanim diferencijalnim jednadžbama (2.1) i (2.2) možemo identificirati različite oblike prijenosa topline, odnosno konvekciju i kondukciju. Konvekcija predstavlja prijenos topline pomoću strujanja fluida, u ovom slučaju zraka, te se pojavljuje u jednadžbi (2.1) kao dio koji opisuje prijenos topline između ulazne temperature zraka T_{in} i izlazne temperature zraka T_{out}. Taj dio je izražen kao: $m' * c_p * (T_{in} - T_{out})$. Dakle, ovaj dio jednadžbe opisuje prijenos topline putem strujanja zraka, što je karakteristika konvekcije. Kondukcija, s druge strane, predstavlja prijenos topline kroz materijal bez strujanja fluida. U jednadžbi (2.1) i (2.2) kondukcija je prisutna u dijelu koji opisuje prijenos topline između površine servera temperature T_s i zraka T_{out}. Taj dio jednadžbi je izražen kao: $k * \frac{A}{\Delta x} * (T_s - T_{out})$. On opisuje prijenos topline kroz materijal (površinu servera) zbog temperaturnog gradijenta, što je karakteristično za kondukciju.



Sl. 2.1 Dijagram hlađenja podatkovnog centra

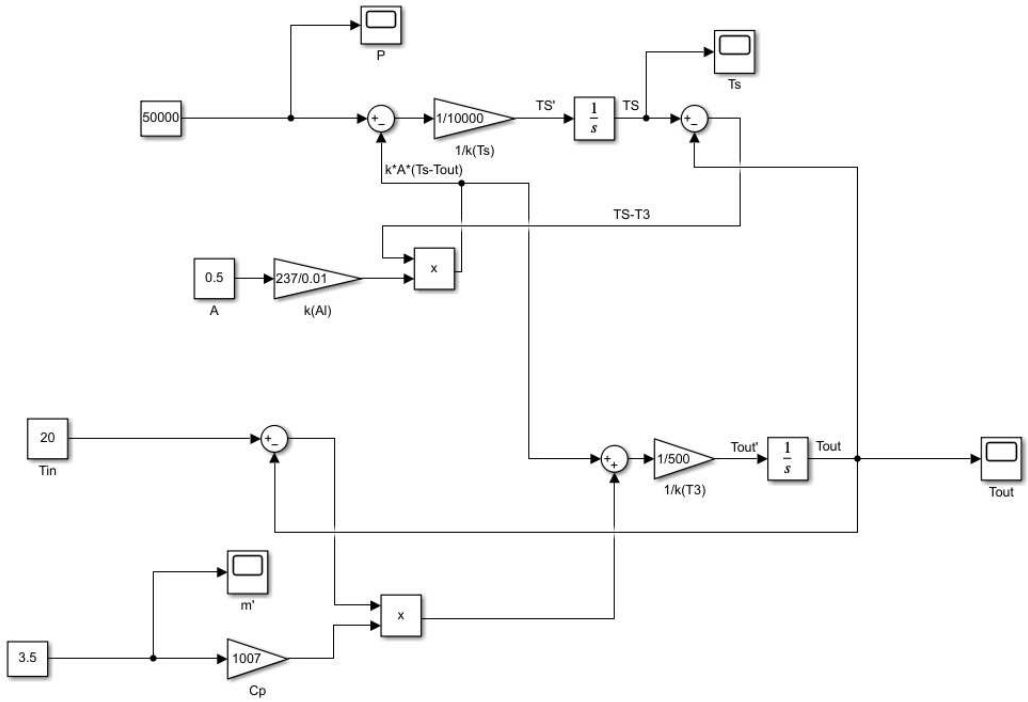
Na slici Sl. 2.1 prikazan je dijagram hlađenja podatkovnog centra. Hlađenje u podatkovnim centrima radi na principu izmjene topline , gdje se višak topline iz zraka uklanja i zamjenjuje hladnijim zrakom. Pojednostavljena varijanta ovog procesa je matematički opisani diferencijalnim jednadžbama (2.1) i (2.2). Računala i mrežni uređaji u podatkovnim centrima proizvode značajnu količinu topline tijekom rada. Vrući zrak se izbacuje iz prostora, a zatim se vanjski, hladniji zrak uvodi u unutrašnjost objekta. Uređaji za hlađenje ulaznog zraka i njegova cirkulacija unutar prostora osiguravaju učinkovito upravljanje temperaturom.

2.2. Shema matematičkog modela u Simulinku

Jednostavni model HVAC sustava, odnosno sustav grijanja, ventilacije i klimatizacije *(eng. Heating, Ventilation, and Air Conditioning)* napravljen je u Simulinku. Simulink je grafičko programsko okruženje u MATLAB-U namijenjeno za modeliranje, simulaciju i analizu dinamičkih sustava.

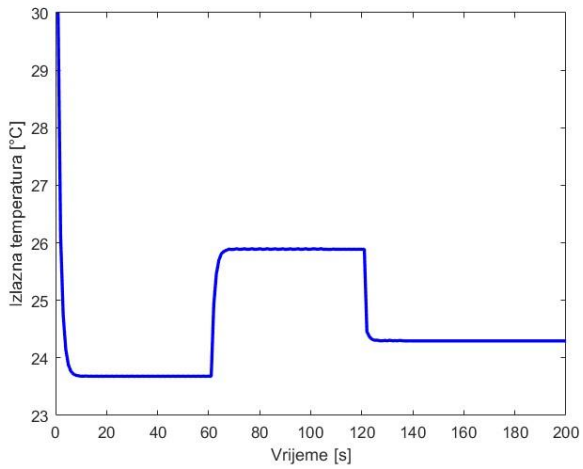
Gore navedene diferencijalne jednadžbe mogu se prikazati u Simulinku tako da se za pojedine varijable koriste konstantni blokovi, a zatim se povezuju matematičkim blokovima koji obavljaju matematičke operacije između njih. Integracijom desnih strana jednakosti dobiju se iznosi temperatura T_s i T_{out} koje se u povratnoj petlji vraćaju u sustav. Za integraciju u Simulinku se

koristi blok Integrator ($\frac{1}{s}$) u kojemu možemo postaviti početni uvjet odnosno inicijalnu temperaturu servera i prostora. Ulazna temperatura T_{in} postavljena je na 20 °C, početna temperatura servera T_s iznosi 50 °C, a početna temperatura prostora 30 °C. Sustav definiran blokovskom shemom u Simulink modelu prikazan je na slici Sl. 2.2.

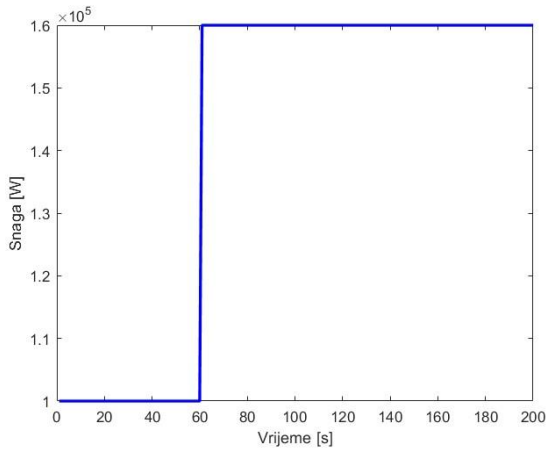


Sl. 2.2 Simulink shema matematičkog modela procesa

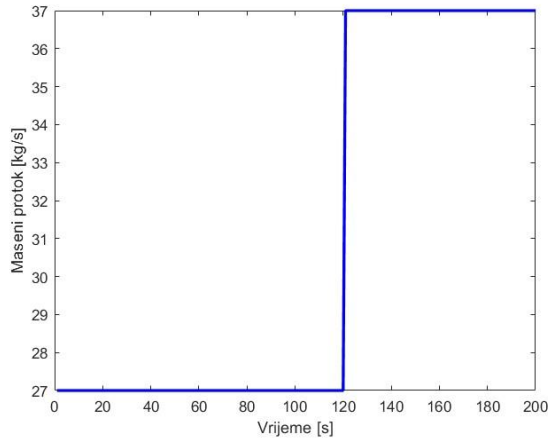
Spomenuti sustav se može analizirati tako da se mijenjaju iznosi snage i masenog protoka zraka te uočiti kako te promjene utječu na izlaznu temperaturu i temperaturu servera pomoću bloka *scope*. Unaprijed se može zaključiti da će veći protok zraka pridonositi snižavanju izlazne temperature, a veća snaga povećavanju.



Sl. 2.3 Izlazna temperatura (T_{out})

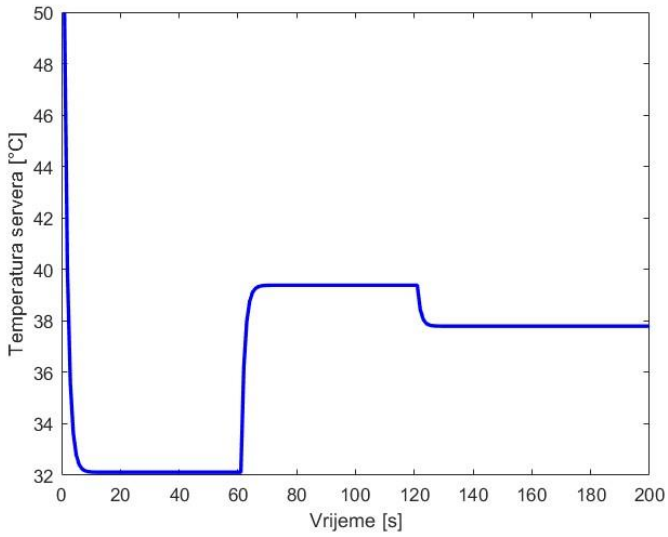


Sl. 2.4 Snaga (P)



Sl. 2.5 Maseni protok (m)

Slika Sl. 2.3 je rezultat simulacije u kojoj je početna vrijednost varijable P = 100 000 W, a varijable m = 27 kg/s. S tako podešenim vrijednostima izlazna temperatura T_{out} s početne vrijednosti 30 °C dolazi u stacionarno stanje s vrijednosti 23.7 °C. U 60. sekundi vrijednost P se poveća sa 100 000 W na 160 000 W pomoću bloka *Step* što se vidi na slici Sl. 2.4, a vrijednost m se ostavi istom. To rezultira povećanjem izlazne temperature koja sada u stacionarnom stanju iznosi 25.9 °C. Konačno, u 120. sekundi poveća se m s 27 na 37 kg/s što se vidi na slici Sl. 2.5, a varijabla P i dalje iznosi 160 000 W što rezultira smanjenjem izlazne temperature na 24.3 °C.



Sl. 2.6 Temperatura servera (T_s)

Temperatura servera prikazana slikom Sl. 2.6 ponaša se slično kao temperatura prostora, no ima različite iznose početne temperature i temperature u stacionarnim stanjima.

Ovaj model će služiti za dobivanje podataka iz simulacija koji će biti korišteni za treniranje neuronske mreže. Također, podaci iz različitih simulacija bit će korišteni za testiranje. Budući da će se koristiti metoda temeljena na podacima, smatrat će se da matematički model nije dostupan.

3. Neuronska mreža procesa

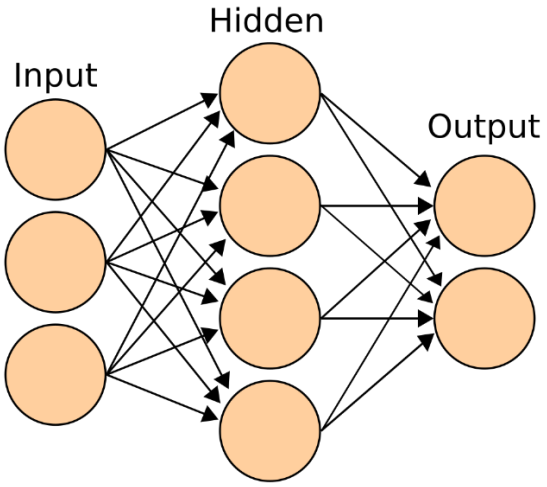
Neuronska mreža procesa koristi se za modeliranje, predikciju i kontrolu složenih sustava poput industrijskih procesa. Ona ima sposobnost učenja nelinearnih odnosa iz podataka, što je ključno za precizno predviđanje i optimizaciju u dinamičkim sustavima. Neuronske mreže uče iz podataka, prepoznajući obrasce i odnose između ulaznih i izlaznih varijabli u svrhu predviđanja budućih stanja sustava ili optimizaciju parametara u stvarnom vremenu.

3.1 Osnove neuronskih mreža

ANN *(engl. Artificial Neural Network)* je računalni model koji oponaša način na koji ljudski mozak radi kako bi obradio podatke i donio odluke. Umjetne neuronske mreže sastoje se od umjetnih neurona (čvorova) organiziranih u slojeve.

Glavne komponente ANN-a, također prikazane na slici Sl. 3.1, su:

1. Ulazni sloj: Sloj neurona koji prima ulazne podatke i prosljeđuje ih dalje u neuronsku mrežu.
2. Skriveni slojevi: Jedan ili više slojeva neurona koji obavljaju obradu ulaznih podataka, omogućuju neuronskoj mreži da nauči složenije obrasce u podacima.
3. Izlazni sloj: Sloj neurona koji generira izlazne rezultate ili predikcije na temelju obrade ulaznih podataka.



Sl. 3.1 Pojednostavljeni prikaz neuronske mreže

Neuron prima ulazne podatke, a svaki ulaz se množi s odgovarajućom težinom (w), dok se rezultati zbrajaju s pristranosti (*bias* b):

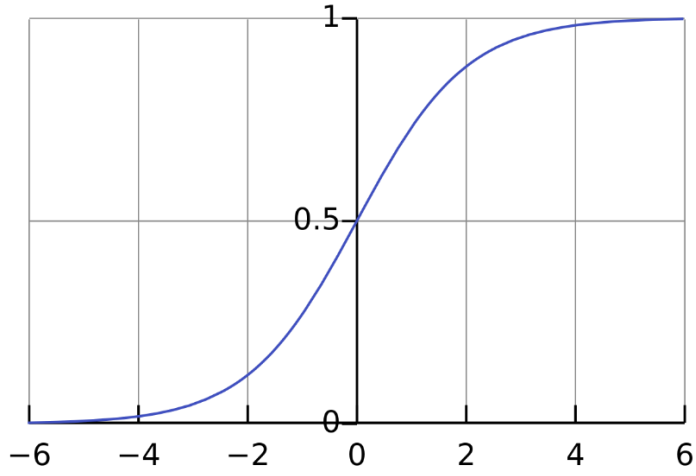
$$Z = w_1x_1 + w_2x_2 + \cdots + w_nx_n + b \qquad (3.1)$$

Zatim se na izračunatu sumu Z primjenjuje aktivacijska funkcija koja odlučuje hoće li se neuron "aktivirati" i kakva će biti njegova izlazna vrijednost: a=σ(Z), gdje je a izlaz neurona, a sigma je aktivacijska funkcija. Aktivacijske funkcije su ključne jer uvode nelinearnost u mrežu omogućavajući joj da nauči složene obrasce. Neke od najčešće korištenih aktivacijskih funkcija su:

1. Sigmoidalna funkcija:

   $$\sigma(z) = \frac{1}{1+e^{-z}} \qquad (3.2)$$

   Prikazana slikom Sl. 3.2 sigmoidalna funkcija je slična step funkciji, ali je glatka te je njezin izlaz uvijek između 0 i 1. Obično se koristi u binarnoj klasifikaciji, u slojevima gdje je potrebno ograničiti izlaz između 0 i 1, što može biti interpretirano kao vjerojatnost pripadnosti jednoj od dvije klase. Graf sigmoidalne funkcije pokazuje kako se vrijednost funkcije brzo mijenja oko z = 0 i asimptotski se približava 0 za negativne ulaze i 1 za pozitivne ulaze.

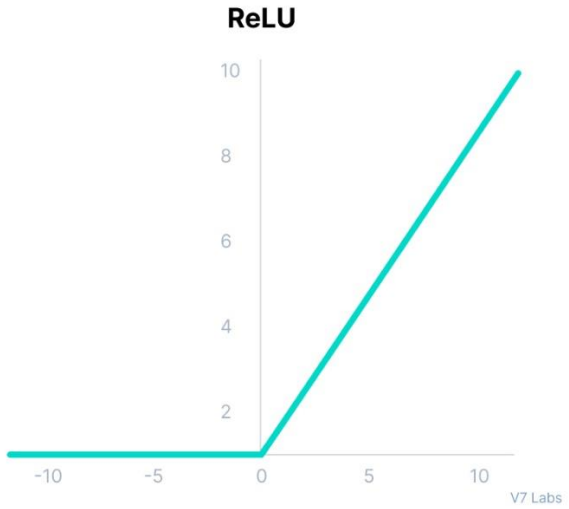


Sl. 3.2 Sigmoidalna funkcija

2. ReLU (Rectified Linear Unit):

   $f(z) = 0, ako\ je\ z < 0; = z, ako\ je\ z \geq 0$ (3.3)

Prikazana slikom Sl. 3.3 ReLu funkcija vraća 0 za negativne ulaze i samu vrijednost za pozitivne ulaze. Često se koristi u skrivenim slojevima zbog svoje jednostavnosti i učinkovitosti.

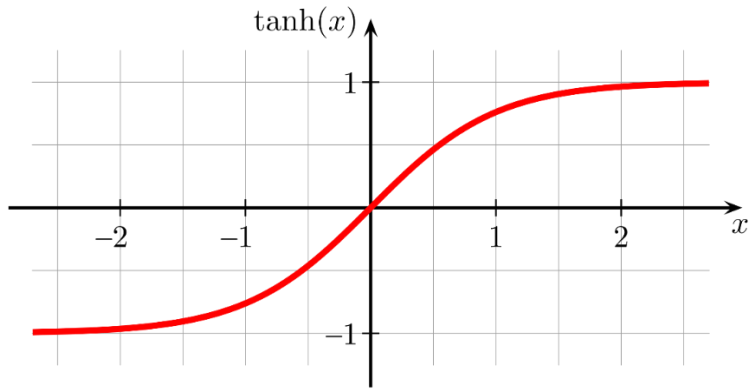


Sl. 3.3 ReLu funkcija

3. Tanh (tangens hiperbolički):

   $\tanh(z) = \frac{2}{1+e^{-2z}} - 1$ (3.4)

   Funkcija tanh prikazana slikom Sl. 3.4 daje vrijednosti u rasponu od -1 do +1. Što je ulaz veći (pozitivniji), to će izlazna vrijednost biti bliža 1, dok će što je ulaz manji (negativniji), izlazna vrijednost biti bliža -1. Tanh je u središtu nule, što znači da je njen izlaz simetričan oko ishodišta koordinatnog sustava što se često smatra prednošću jer može pomoći da algoritam učenja brže konvergira.



Sl. 3.4 Tanh funkcija

4. Softmax:

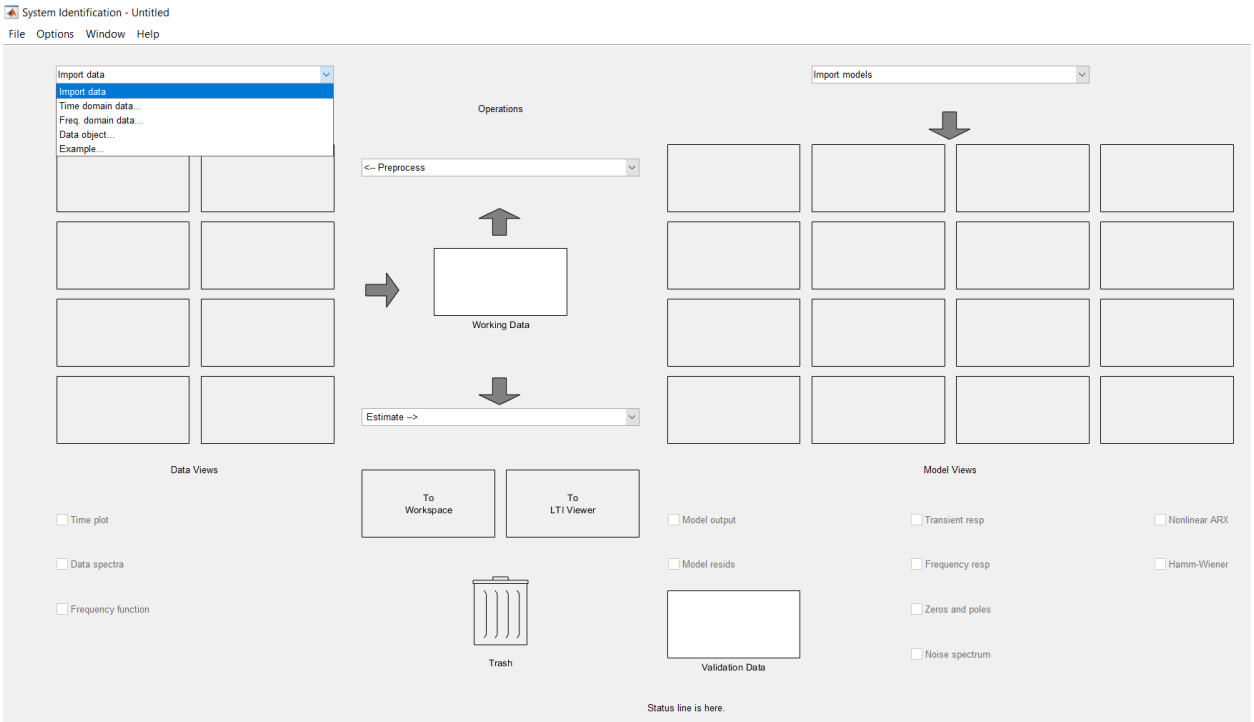
$$softmax(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \qquad (3.5)$$

Funkcija softmax transformira ulazne vrijednosti u vjerojatnosti koje se zbrajaju. Zbroj svih vjerojatnosti nakon primjene softmax funkcije jednak je 1. Najčešće se koristi u višeklasnoj klasifikaciji u izlaznom sloju neuronske mreže kako bi se dobila vjerojatnost za svaku klasu. Softmax funkcija nema jednostavan 2D graf jer operira na vektorima. Međutim, rezultirajući vektor vjerojatnosti pokazuje kako se svaka komponenta transformira u vrijednost između 0 i 1, s ukupnim zbrojem vjerojatnosti jednakim 1. Koristi se u izlaznom sloju.

Nadalje, postoji nekoliko tipova neuronskih mreža, a u ovom radu se koristi *Feedforward Neural Network* (FFNN). FFNN je najosnovniji tip neuronske mreže. Podaci teku samo u jednom smjeru, od ulaznog sloja preko skrivenih slojeva do izlaznog sloja. Naredbom *„neuralNet = feedforwardnet(10)“* u MATLAB-U se stvara neuronska mreža koja ima 10 neurona u skrivenom sloju. Podaci se automatski dijele na tri skupa: train (trening), test i validation (validacija) u omjeru 70:15:15. Tijekom treniranja modela mreža prilagođava težine i pristranosti kako bi minimizirala grešku na trening skupu, a povremeno se testira na validacijskom skupu kako bi se provjerilo kako model generalizira. Nakon što je trening završen test skup služi kao potpuno neovisan skup podataka za procjenu kako bi se dobila objektivna mjera performansi modela.

3.2 Matlab aplikacija za identifikaciju procesa

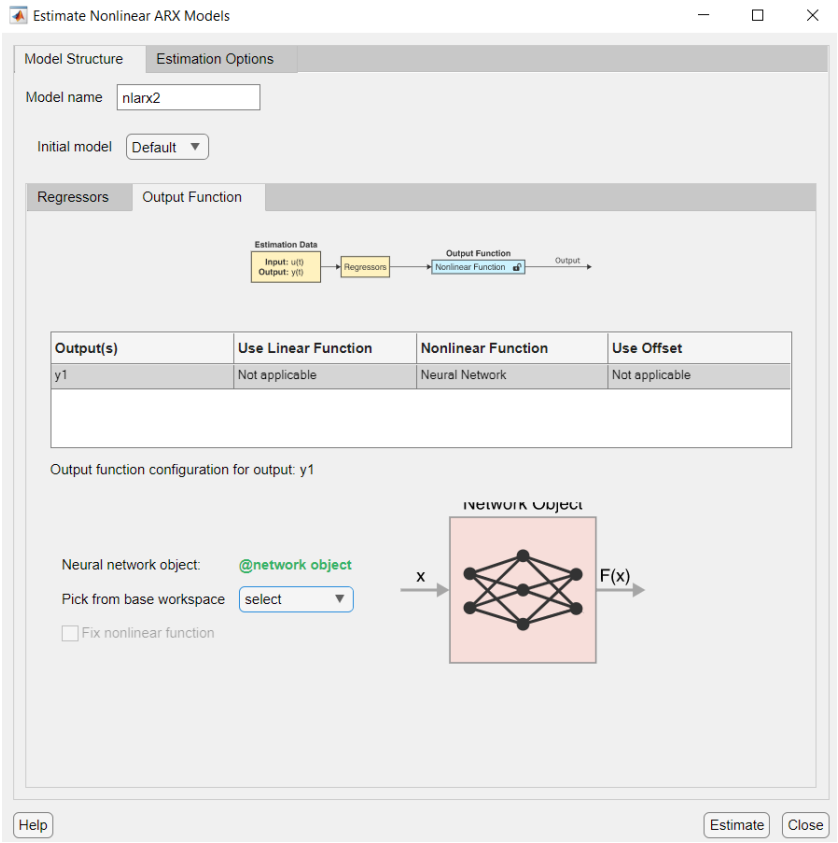
Metode temeljene na podacima omogućuju relativno jednostavno modeliranje jer u ovom pristupu nije potrebno specijalno znanje jer su korelacije između ulaza i izlaza utvrđene s pomoću simulacija i podataka iz njih. *System Identification App* u MATLAB-u je intuitivno grafičko sučelje za identificiranje modela sustava na temelju ulazno-izlaznih podataka, a prozor tog sučelja je prikazan slikom Sl. 3.5. *System Identification App* nudi Matlab funkcije, Simulink blokove i aplikaciju za dinamičko modeliranje sustava, analizu vremenskih serija i predviđanje. Mogu se naučiti dinamički odnosi među izmjerenim varijablama za stvaranje funkcija prijenosa, modela procesa i modela prostora stanja u kontinuiranom ili diskretnom vremenu dok se koriste podaci u vremenskoj ili frekvencijskoj domeni.



Sl. 3.5 Prozor alata *System Identification*

Alat omogućuje procjenu dinamike nelinearnog sustava korištenjem nelinearnih ARX modela s tehnikama strojnog učenja kao što su Gaussovi procesi (GP), strojevi potpornih vektora (SVM) itd. Identificirani modeli se lako mogu integrirati u Simulink. Nelinearni ARX *(AutoRegressive with eXogenous inputs)* model sastoji se od regresora modela i izlazne funkcije. Regresori su ključni podaci koji se koriste kao ulazi u izlaznu funkciju modela. Oni uključuju: prošle vrijednosti

izlaza koje model koristi da bi predvidio trenutni izlaz (y(t−1), y(t−2), ...)) i prošle vrijednosti ulaza u sustav (u(t−1), u(t−2), ...). Izlazna funkcija koristi regresore za stvaranje predikcije izlaza. Koristi i nelinearnu funkciju koja omogućava modelu da obuhvati složenija, nelinearna ponašanja u sustavu. U svrhu predviđanja temperature prostora za ulaz je korišten maseni protok (u1) te snaga (u2). Definirani su linearni regresori kao prethodne dvije vrijednosti ulaza i izlaza (y(t-1), y(t-2), u1(t-1), u1(t-2), u2(t-1), u2(t-2)). Budući da se koristi 6 regresora prije svake simulacije moraju se definirati početni uvjeti u modelu kao 1x6 vektor gdje su prve dvije vrijednosti temperatura, zatim protok te zadnje dvije početna snaga. Za nelinearnu funkciju izlaza se koristi neuronska mreža. Neuronska mreža je prethodno stvorena u MATLAB-u, a nakon toga ju treba istrenirati.



Sl. 3.6 Prozor estimacije NLARX modela

Klikom na gumb *„Estimate“* u prozoru prikazanog slikom Sl. 3.6 pokreće se proces treniranja neuronske mreže. Epohe su ključni koncept u treniranju neuronskih mreža i označavaju koliko puta je cijeli skup podataka prošao kroz mrežu tijekom procesa treniranja. Nakon što se cijeli skup podataka prođe kroz mrežu, epoha je završena, a treniranje nastavlja s još jednom epohom,

ponovno prolazeći kroz cijeli skup podataka. Prevelik broj epoha može dovesti do pretreniranosti, gdje model ima jako malu grešku na trening podacima, ali loše generalizira na nove, neviđene podatke. Premali broj epoha može dovesti do nedovoljnog treniranja, gdje model ne nauči dovoljno o podacima, što rezultira slabim performansama. Redak *„Performance“* prikazuje vrijednost funkcije gubitka *(engl. loss function)* ili metrike pogreške koja se koristi za ocjenu koliko dobro mreža predviđa ciljne vrijednosti u odnosu na stvarne vrijednosti. Ona se obično smanjuje tijekom treniranja, što ukazuje na to da se mreža prilagođava podacima i smanjuje pogrešku. *„Performance“* je obično srednja kvadratna pogreška (MSE, *Mean Squared Error*). MSE mjeri prosječnu kvadratnu razliku između predviđenih i stvarnih vrijednosti.

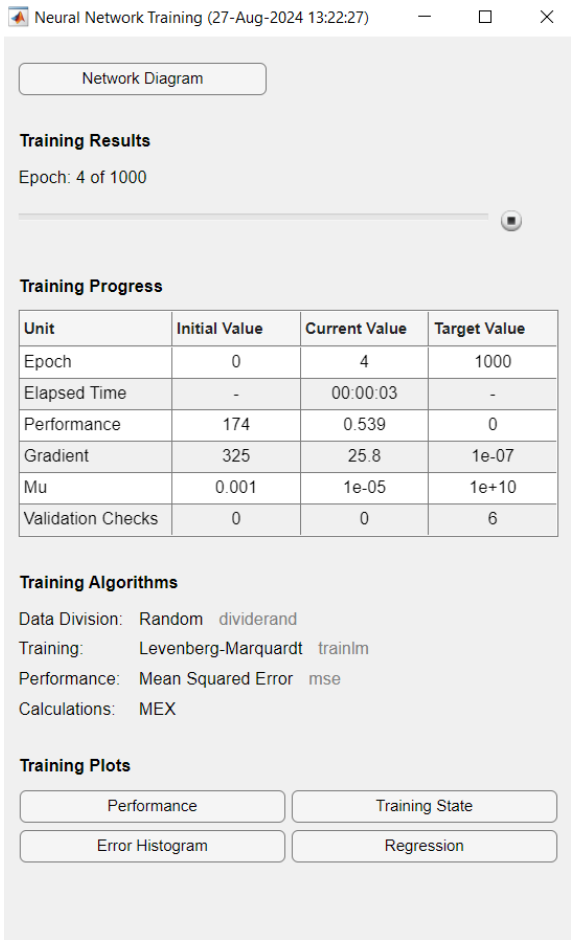
$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y^i - \widehat{y^i})^2 \qquad (3.6)$$

gdje su y^i stvarne vrijednosti, a $\widehat{y^i}$ predviđene vrijednosti. Gradijent je vektor koji sadrži parcijalne derivacije funkcije gubitka s obzirom na svaki parametar mreže. Koristi se za ažuriranje parametara mreže tijekom procesa treniranja. Gradijent pokazuje smjer u kojem treba promijeniti parametre kako bi se smanjila vrijednost funkcije gubitka. Ako je gradijent veliki, to znači da su potrebne veće promjene u težinama kako bi se smanjila pogreška. Parametri se ažuriraju tako da se kreću u smjeru negativnog gradijenta, jer to smanjuje funkciju gubitka. Formula za ažuriranje parametara je:

$$\theta_{t+1} = \theta_t - \eta * \nabla L(\theta_t) \qquad (3.7)$$

gdje je θ_t trenutni parametar (npr. težina), η brzina učenja (*learning rate*), a $\nabla L(\theta_t)$ gradijent funkcije gubitka s obzirom na taj parametar. Parametar Mu koristi se u Levenberg-Marquardt algoritmu za optimizaciju. Levenberg-Marquardt algoritam je hibrid između Gauss-Newton metode i gradijentnog spusta te se koristi za rješavanje nelinearnih problema najmanjih kvadrata, a parametar Mu kontrolira prelazak između Gauss-Newton metode (koja je brža kad je blizu optimalnog rješenja) i gradijentnog spusta (koji je stabilniji kad je daleko od optimalnog rješenja). Drugim riječima, to je faktor koji prilagođava koliko algoritam daje prednost jednoj metodi u odnosu na drugu. Kada je Mu veliki, algoritam se ponaša više kao gradijentni spust, što je korisno kada je daleko od rješenja, jer osigurava stabilnost. Kada je Mu mali, algoritam se ponaša više kao Gauss-Newton metoda, što je korisno kada je blizu rješenja, jer omogućava bržu konvergenciju. *„Validation Checks“* prati koliko epoha uzastopno mreža nije pokazala poboljšanje performansi

na validacijskom skupu. Ako se performanse na validacijskom skupu ne poboljšaju nakon određenog broja epoha, proces treniranja se automatski zaustavlja. Takva tehnika koja koristi validacijski skup za zaustavljanje treniranja prije nego što dođe do pretreniranosti se naziva rano zaustavljanje (*engl early stopping*). Nakon što se treniranje završi prikazuje se *„Training finished: Met validation criterion“* , to znači da je proces treniranja neuronske mreže zaustavljen jer je ispunjen kriterij za validaciju, a ne zato što je dostignut maksimalan broj epoha. Tijekom treniranja, mreža se periodično procjenjuje na validacijskom skupu. Ako mreža počne previše dobro performirati na trening skupu, ali se performanse na validacijskom skupu počnu pogoršavati, to je znak pretreniranosti. Ako se performanse na validacijskom skupu ne poboljšavaju (ili se pogoršavaju) nakon određenog broja epoha, treniranje se zaustavlja. U ovom slučaju broj epoha prija zaustavljanja treniranja mreže iznosi 6. *"Data Division: Random"* znači da su podaci podijeljeni nasumično u trening, validacijski, i testni skup, što pomaže u stvaranju reprezentativnih podskupova za efikasnije treniranje i evaluaciju neuronske mreže. Na slici Sl. 3.7 prikazan je opisani proces treniranja neuronske mreže.



Sl. 3.7 Prozor treniranja neuronske mreže

Sa završetkom treniranja neuronske mreže se stvori nelinearni ARX model te je moguće procijeniti njegovu točnost usporedbom predikcija modela s pravim izlaznim podacima na skupu podataka za validaciju. Posljednji korak je eksportiranje modela u radni prostor MATLAB-a i korištenje istog prilikom simulacija.

4. Prediktivno upravljanje

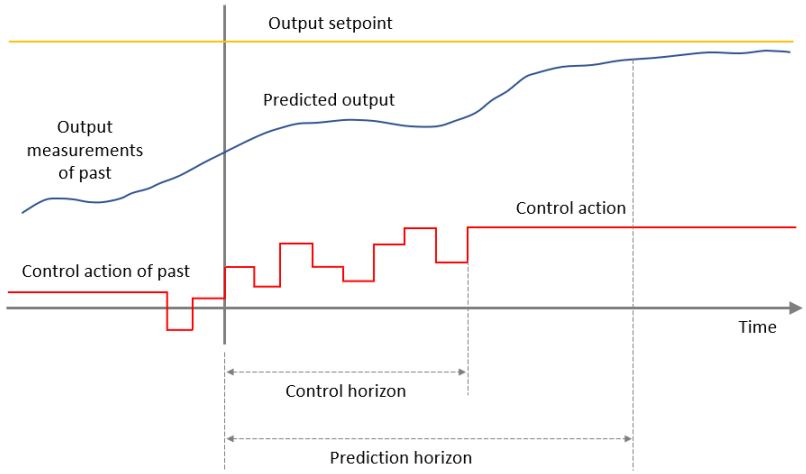
Prediktivno upravljanje je zasnovano na ideji da se za proračun upravljačkih signala iz regulatora koristi predikcija ponašanja procesa kojim se upravlja. Poznavanje budućeg ponašanja omogućava da se odabere upravljački signal koji će na optimalan način dovesti izlaz do referentne vrijednosti. Prednost ovog pristupa je to što se upravljanje započne prije promjene reference npr. što osigurava malo nadvišenje odziva i brzu stabilizaciju izlaza.

4.1 Modelsko prediktivno upravljanje

Model prediktivne kontrole (MPC) je napredna metoda regulacije koja koristi matematički model sustava za predviđanje budućih ponašanja procesa i optimizaciju upravljačkih akcija u konačnom vremenskom okviru, horizontu. MPC koristi model da bi predvidio buduće vrijednosti izlaza sustava na temelju trenutnog stanja i budućih upravljačkih signala. To znači da regulator može gledati u budućnost i prilagoditi svoje akcije kako bi postigao željeni ishod. MPC rješava optimizacijski problem u stvarnom vremenu, minimizirajući razliku između željenog i predviđenog izlaza. Moguće je postaviti ograničenja na ulaze, izlaze i stanja sustava pa je koristan kada postoje fizička ili operativna ograničenja. Metoda modelskog prediktivnog upravljanja funkcionira tako da se u svakom diskretnom vremenskom koraku k izračunava predikcija izlaza procesa za definirani predikcijski horizont N, a proračun upravljačke sekvence se dobiva minimizacijom kriterijske funkcije. Nakon što se izračuna predikcijski vektor upravljačke veličine:

$$\vec{u}(k) = [u(k)u(k+1) \dots u(k+N-1)], \quad (4.1)$$

njegov prvi element $u(k)$ šalje se na izlaz regulatora, a ostali elementi se zanemaruju. Postupak proračuna se ponavlja u svakom sljedećem vremenskom koraku. Ovaj princip opisuje pomični horizont (*engl. Receding Horizon*) odnosno broj budućih vremenskih koraka za koje MPC regulator predviđa ponašanje sustava i rješava optimizacijski problem. Princip rada MPC-a prikazan je i slikom Sl. 4.1.



Sl. 4.1 Princip rada MPC-a

Kada je horizont predikcije kratak, MPC predviđa buduće ponašanje sustava na samo nekoliko vremenskih koraka unaprijed. To omogućava brže izračune, ali daje ograničen uvid u dugoročno ponašanje sustava. Kratki horizont često dovodi do brze reakcije sustava na promjene, što može biti korisno u situacijama kada su brze prilagodbe potrebne, ali to može dovesti i do prekomjerne regulacije ili nestabilnosti.

Kada je horizont predikcije dug, MPC može bolje predvidjeti dugoročne efekte upravljačkih akcija i optimizirati kontrolu sustava na način koji uzima u obzir ne samo trenutne, već i buduće promjene. On obično vodi do boljih performansi sustava, jer regulator može unaprijed vidjeti i spriječiti buduća odstupanja od referentne vrijednosti. Međutim, s dugim horizontom dolazi i veća složenost računanja. Svaki korak MPC regulatora zahtijeva rješavanje optimizacijskog problema, pa dulji horizont znači da se mora riješiti više budućih upravljačkih koraka, što povećava računski trošak. Također, može sporije reagirati na brze promjene jer optimizacija daje veći prioritet dugoročnim efektima. Kako bi dobili što bolju funkcionalnost bitno je napraviti dobar kompromis između dugog i kratkog horizonta. Ako MPC koristi jednadžbe za opis matematičkog modela procesa on je obično opisan sljedećim jednadžbama:

$$x(t+1) = Ax(t) + Bu(t) \qquad (4.2)$$

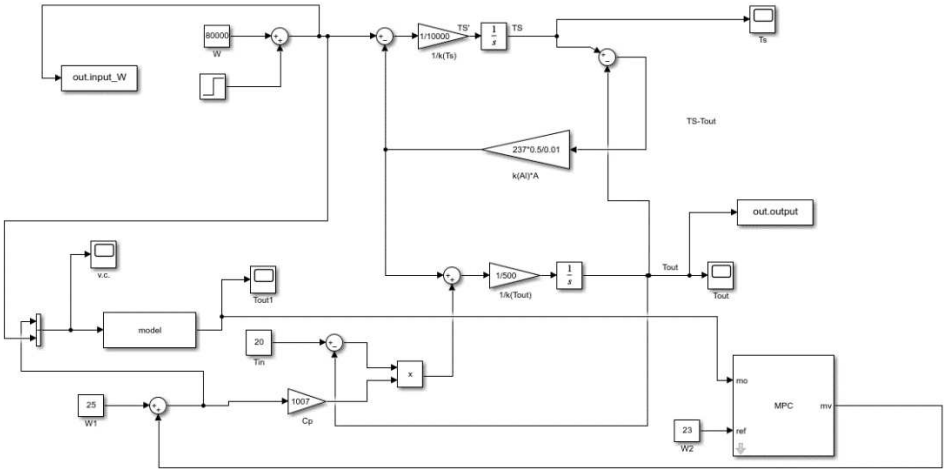
$$y(t) = Cx(t) + Du(t) \qquad (4.3)$$

gdje su x(t) stanja sustava koja se mijenjaju tijekom vremena, u(t) ulazni signali odnosno upravljačke varijable koje MPC optimizira i y(t) izlaz sustava odnosno varijabla koju MPC kontrolira. A, B, C i D su matrice koje definiraju dinamiku sustava.

4.2 Prediktivno upravljanje temeljeno na podacima

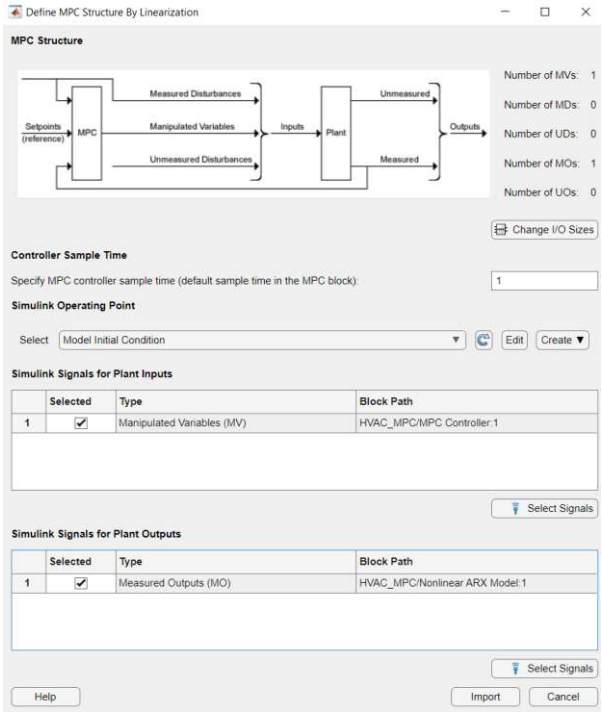
Problem nastaje jer za mnoge složene sustave nije lako dobiti analitičke modele (jednadžbe) zbog složenih interakcija među komponentama ili nepoznate dinamike. Modeli temeljeni na podacima omogućuju modeliranje sustava i kad analitički modeli nisu dostupni ili su prekomplicirani za izvođenje te je dovoljno prikupiti kvalitetne podatke iz simulacije bez dubokog razumijevanja unutarnjih mehanizama sustava. Osim toga matematički modeli su često pojednostavljeni i ignoriraju ili određene nelinearnosti, dok modeli temeljeni na podacima automatski uključuju sve efekte prisutne u podacima, što može dovesti do preciznijeg modeliranja i boljih performansi kontrolnog sustava prikazanog slikom Sl.4.2.

U ovom radu koristila se identifikacija sustava kako bi se iz eksperimentalnih podataka stvorio model koji će se koristiti prilikom dizajniranja MPC regulatora. MPC regulator dodaje se u simulaciju odabirom *MPC Controller* bloka iz Simulink biblioteke. Blok ima ulaze mo, ref te izlaz mv. MO (*Measured Outputs*) predstavlja izmjerene izlaze sustava koji se uspoređuju s referentnim signalima kako bi se izračunale pogreške koje MPC pokušava minimizirati. Na mo doveden je izlaz iz NARX modela koji predviđa izlaznu temperaturu. REF (*Reference*) predstavlja referentni signal koji želimo da izlazne varijable sustava prate. Ovo su ciljne vrijednosti koje MPC pokušava dosegnuti minimiziranjem razlike između referentnih signala i izlaza (MO). MV (*Manipulated Variables*) predstavlja upravljačke varijable koje MPC regulator izračunava i šalje natrag u sustav kako bi se postiglo željeno ponašanje izlaza. Optimalne mv se izračunavaju na temelju razlike između mo i ref, kao i predikcije budućeg ponašanja sustava. Izlaz iz regulatora odnosno mv dovodi se na ulaz u sustav te predstavlja maseni protok zraka.



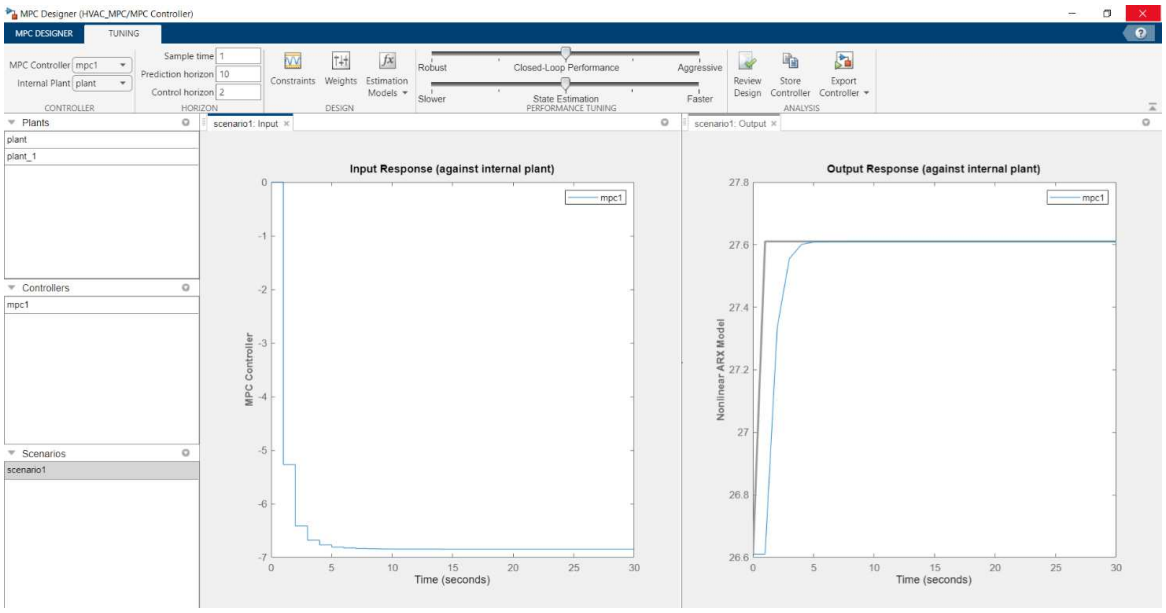
Sl. 4.2 Simulink shema s MPC regulatorom

Nakon definiranja ulaza i izlaza otvara se blok modelskog prediktivnog regulatora prikazanog unutar Simulink sheme na slici Sl. 4.2 te pokreće dizajniranje MPC-a. Za ovo je potreban *Model Predictive Control Toolbox* te dodatni alat *Simulink Control Design*. Klikom na *„MPC Structure" MPC Designer* obavlja linearizaciju sustava koji se nalazi između dovedenih signala mv i mo, a prozor koji to prikazuje nalazi se na slici Sl. 4.3.



Sl. 4.3 Linearizacija sustava unutar *MPC Designera*

Nakon što se definirao proces (*engl. plant*) te formirao objekt MPC moguće je odrediti parametre regulatora odabirom prozora „TUNNING“. Pomoću klizača se određuje ponašanje zatvorenog kruga i brzina estimacije. Upisom u određena polja određujemo vrijeme uzorkovanja, predikcijski i kontrolni horizont. Predikcijski horizont je broj vremenskih koraka unaprijed za koje MPC regulator predviđa ponašanje sustava, a kontrolni horizont je broj vremenskih koraka unutar predikcijskog horizonta tijekom kojih MPC aktivno optimizira upravljačke varijable. Kontrolni horizont obično je kraći od predikcijskog horizonta jer optimiziranje upravljačkih varijabli preko cijelog predikcijskog horizonta može biti pretjerano računanje. Ograničenja ulaznih i izlaznih signala se definiraju unutar polja *„Constraints”*, a težinski parametri se mogu unijeti u polju *„Weights”*. Nakon promjene određenog parametra, iscrtavaju se vremenski dijagrami ulaza i izlaza zatvorenog kruga upravljanja što omogućava što bolji odabir parametara za željeno ponašanje sustava. Na slici Sl. 4.4 prikazan je prozor unutar kojega se odvija odabir parametara.

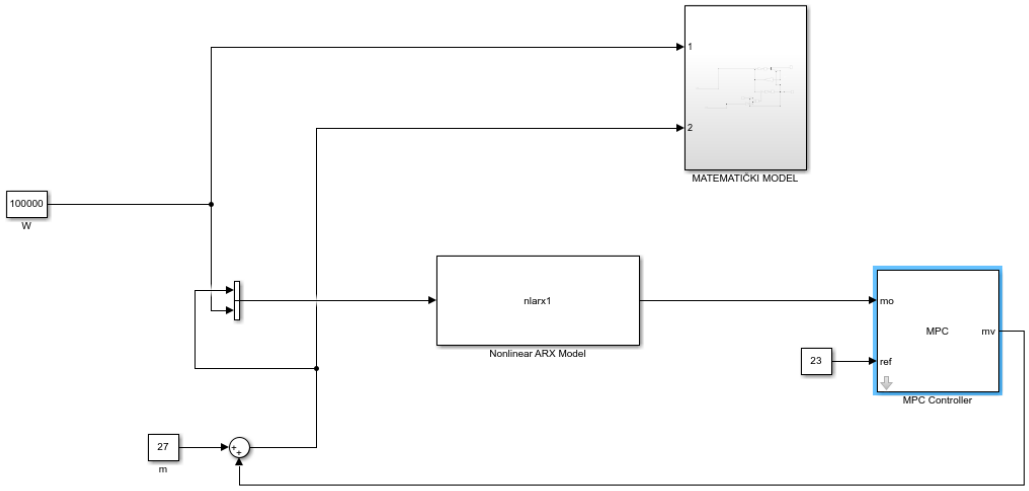


Sl. 4.4 Prozor *MPC Designera* s parametrima regulatora

5. Rezultati

Ovo poglavlje predstavlja rezultate simulacija izvedenih na modelu HVAC prostora razvijenom u Simulinku. Model se sastoji od matematičkog modela prostora, koji simulira stvarnu dinamiku temperature unutar prostora, te nelinearnog ARX (NLARX) modela, koji predviđa temperaturu na temelju ulaznih varijabli: maseni protok zraka i snaga grijanja. Za upravljanje temperaturom u prostoru, primijenjen je prediktivni regulator temeljen na podacima (MPC), koji koristi predviđenu temperaturu iz NLARX modela kao ulaznu varijablu. Cilj regulatora je održavanje temperature prostora što bliže zadanoj referenci.

Slika Sl. 5.1 prikazuje cjelokupni Simulink model koji obuhvaća HVAC sustav s NLARX modelom i MPC regulatorom. Matematički model prostora simulira stvarnu dinamiku temperature, dok NLARX model, treniran na temelju stvarnih podataka, predviđa buduću temperaturu prostora. MPC regulator koristi ovu predviđenu temperaturu kako bi prilagodio maseni protok zraka, te tako osigurao da temperatura prati zadanu referentnu vrijednost.

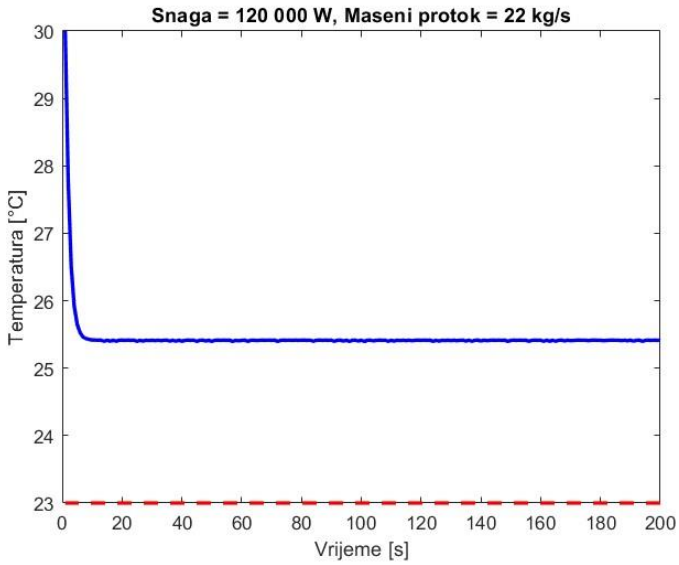


Sl. 5.1 Simulink shema sustava s NLARX modelom i MPC regulatorom

5.1 Dinamika sustava bez upravljanja

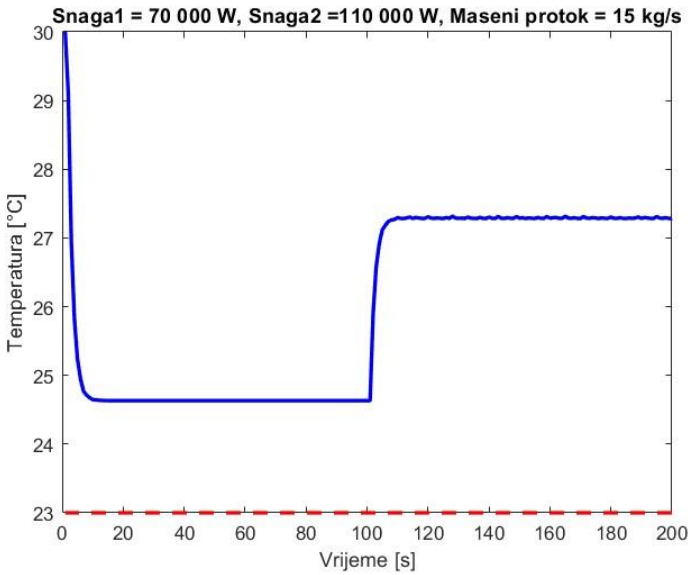
U početku je simulacija izvedena bez aktivacije MPC regulatora kako bi se analizirala osnovna dinamika sustava. Slika Sl. 5.2 prikazuje promjenu temperature prostora kroz vrijeme kada maseni protok i snaga grijanja ostaju konstantni. Kao što se može vidjeti, temperatura se stabilizira na

određenoj razini, ali ne prati referentnu vrijednost. Ovaj rezultat ističe potrebu za primjenom prediktivnog regulatora kako bi se postigla željena temperatura.



Sl. 5.2 Izlazna temperatura uz konstantnu snagu i protok, bez upravljanja

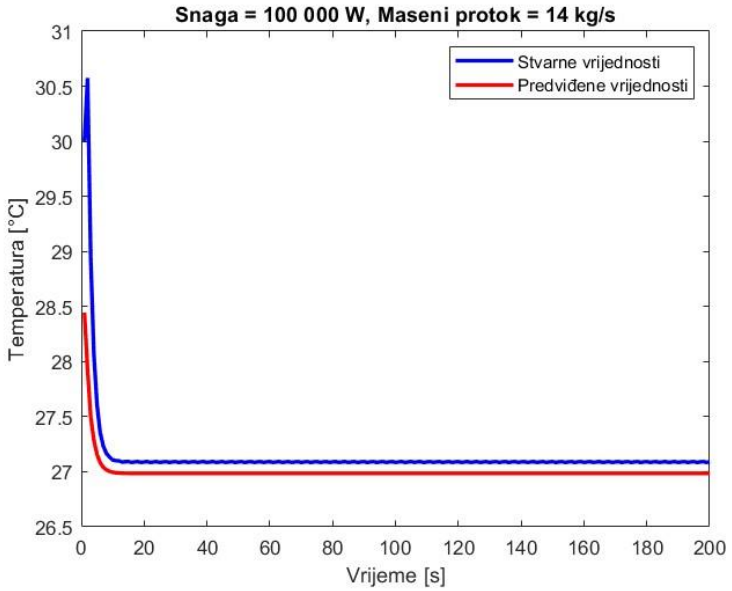
Na slici Sl. 5.3 je prikazan odziv sustava bez upravljanja kada se snaga poveća za 40 000 W u 100. sekundi, a protok je konstantan. To je rezultiralo povećanjem temperature koja se prvo stabilizirala na 24.63, a zatim na 27.29 °C.



Sl. 5.3 Izlazna temperatura uz povećanje snage i konstantan protok, bez upravljanja

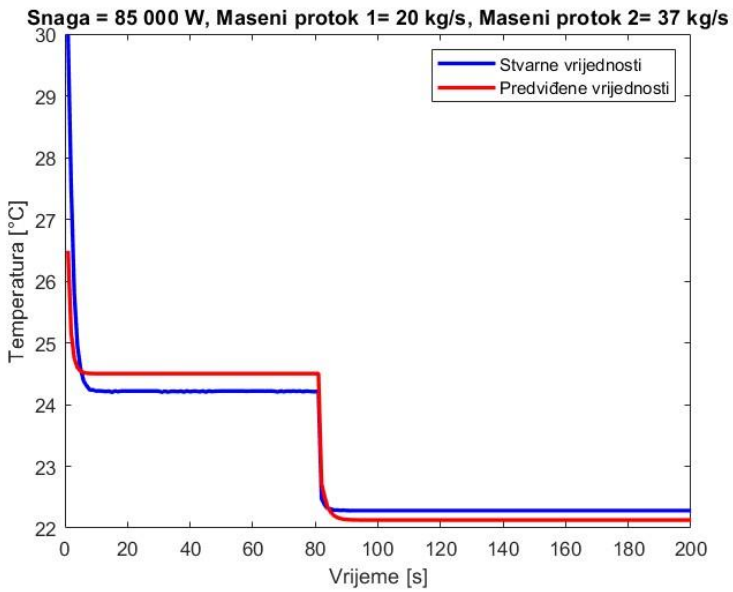
5.2 Performanse NARX modela

Prije primjene modelskog prediktivnog upravljanja, ocijenjena je točnost NLARX modela. Slika Sl. 5.4 prikazuje usporedbu između predviđene temperature NLARX modela i stvarne izlazne temperature iz matematičkog modela. Analizom je utvrđeno da NLARX model točno predviđa promjene temperature s malim odstupanjima, što ga čini prikladnim za upotrebu unutar MPC regulatora. U ovom slučaju greška u stacionarnom stanju iznosi 0.08 °C.



Sl. 5.4 Predviđena i stvarna izlazna temperatura uz konstantnu snagu i protok

Slika Sl. 5.5 prikazuje usporedbu izlaza NLARX i matematičkog modela u slučaju kada je snaga konstantna, a protok se poveća u 80. sekundi. Ovdje greška u stacionarnom stanju iznosi 0.3 odnosno 0.1 °C.

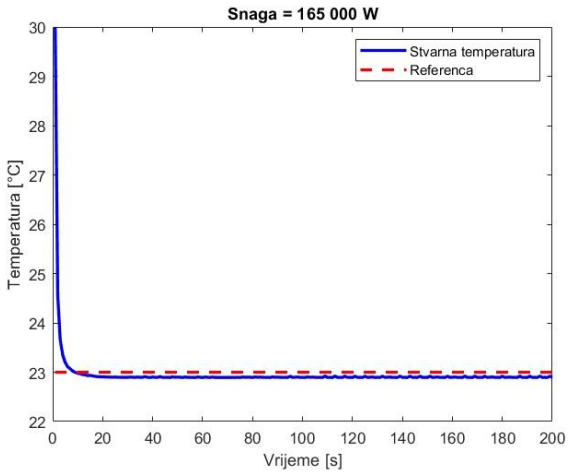


Sl. 5.5 Predviđena i stvarna temperatura uz konstantnu snagu i povećani protok

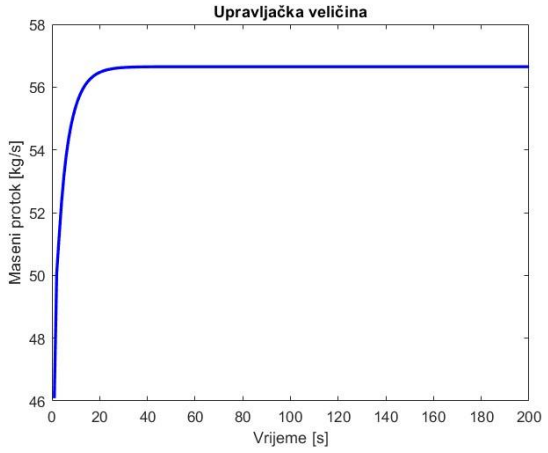
5.3 Upravljanje temperaturom primjenom MPC regulatora

Kada je MPC regulator aktiviran, sustav prilagođava maseni protok zraka u skladu s predviđenom temperaturom i zadanom referencom. U ovom dijelu bit će prikazani rezultati simulacija u kojoj MPC regulator uspješno održava temperaturu prostora blizu referentne vrijednosti, čak i uz promjene u snazi grijanja.

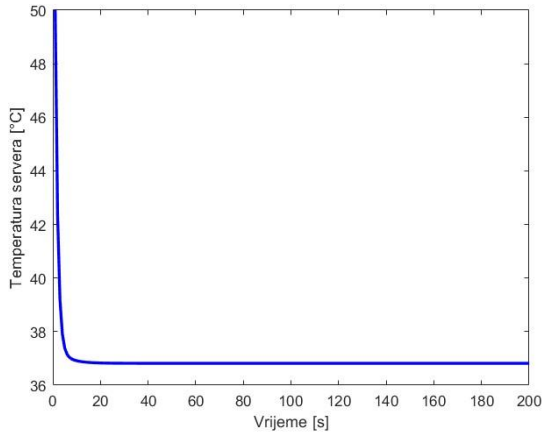
Slika Sl. 5.6 prikazuje odziv simulacije s aktiviranim MPC regulatorom uz konstantnu snagu. Referentna temperatura je postavljena na 23 °C, a stvarna temperatura se stabilizirala na 22.9 °C što je zadovoljavajuć rezultat upravljanja. Promjena upravljačke veličine kroz vrijeme prikazana je na slici Sl. 5.7, a temperature servera na slici Sl. 5.8.



Sl. 5.6 Izlazna temperatura uz konstantnu snagu, s upravljanjem

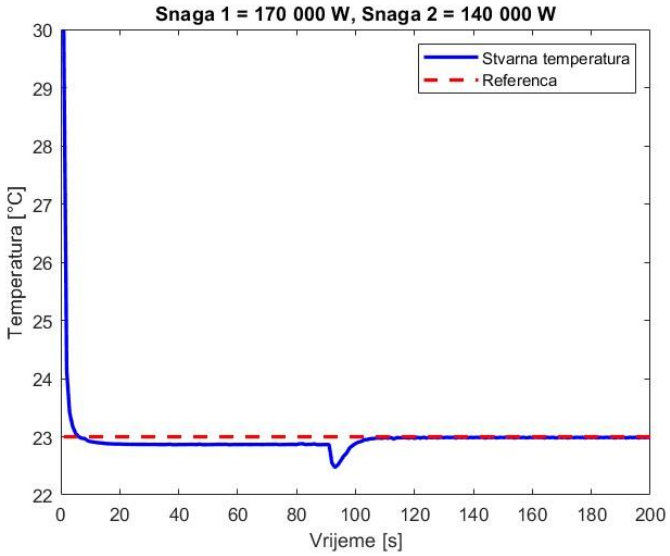


Sl. 5.7 Upravljačka veličina uz konstantnu snagu

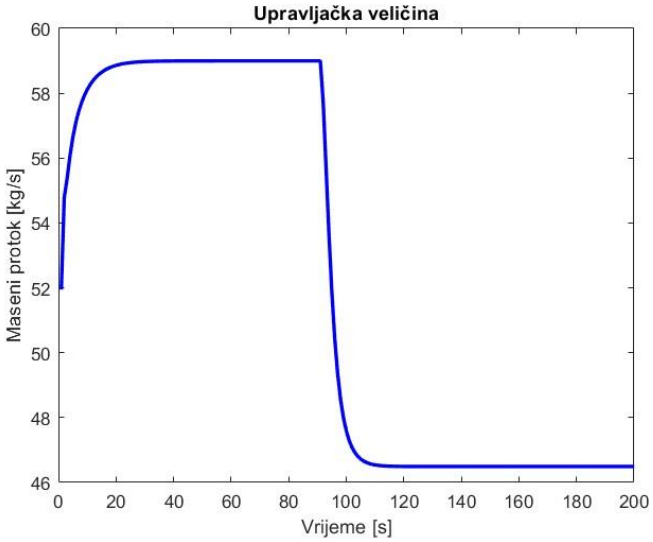


Sl. 5.8 Temperatura servera uz konstantnu snagu

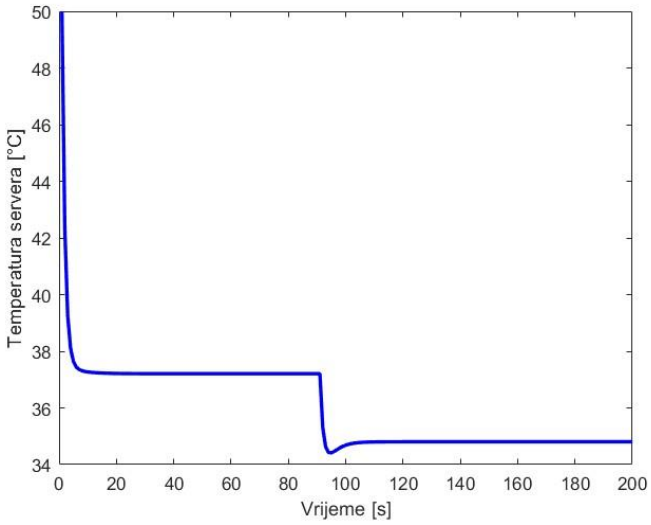
U sljedećoj simulaciji prikazanoj na slici Sl. 5.9 također se testirala uspješnost upravljanja, ali umjesto konstantne vrijednosti snage ona se smanjila za 30 000 W u 90. sekundi. Može se primijetiti kako temperatura dođe blizu vrijednosti reference koja iznosi 23 °C, međutim zbog smanjenja snage u 90. sekundi temperatura na kratko padne prije nego što se ponovo stabilizira blizu referentne vrijednosti. Slika Sl. 5.10 prikazuje upravljačku veličinu za istu simulaciju, pa vidimo kako se u trenutku smanjenja snage smanji i maseni protok. Temperatura servera se također smanji u 90. sekundi što se vidi na slici Sl. 5. 11.



Sl. 5.9 Izlazna temperatura uz smanjenje snage, s upravljanjem

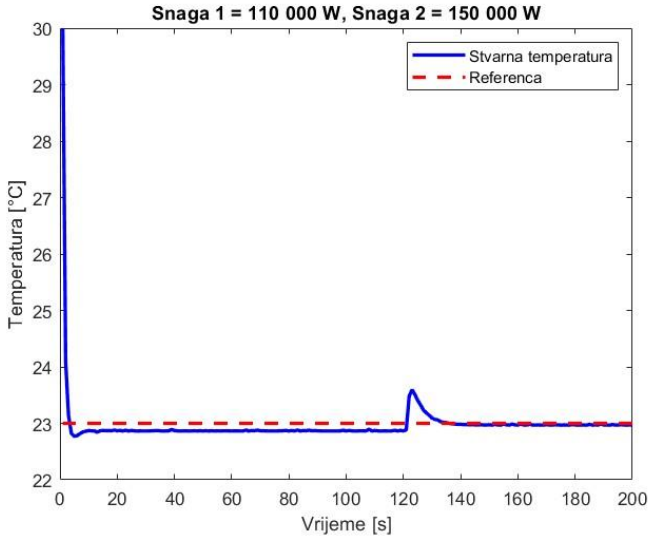


Sl. 5.10 Upravljačka veličina uz smanjenje snage

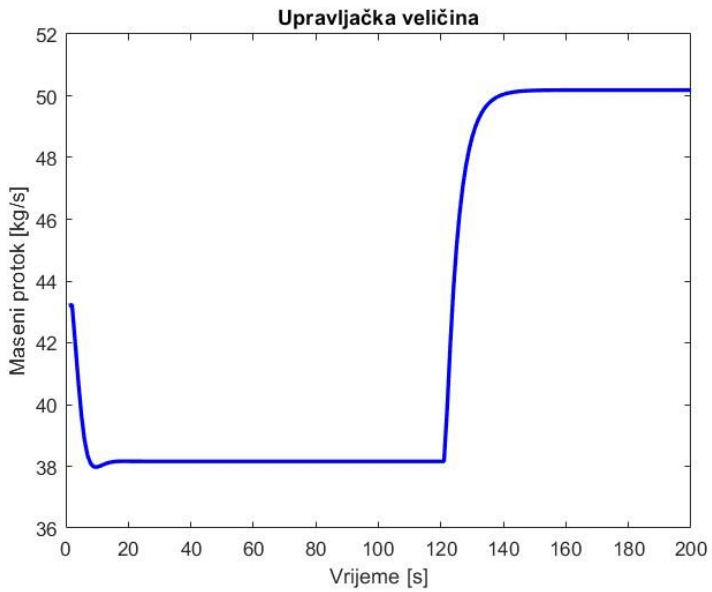


Sl. 5.11 Temperatura servera uz smanjenje snage

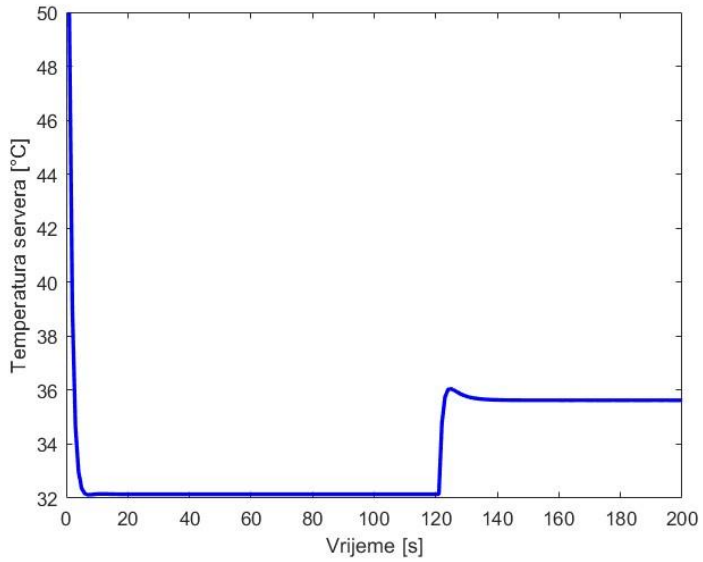
Posljednja simulacija kojom će se testirati uspješnost upravljanja MPC regulatorom bit će sa povećanjem snage u nekom trenutku. Na slici Sl. 5.12. prikazana je izlazna temperatura u slučaju kada je snaga narasla za 40 000 W u 120. sekundi. Može se primijetiti kako je u tom trenutku izlazna temperatura kratko skočila, ali se svejedno ubrzo stabilizirala na referentnu vrijednost. Slika Sl. 5.13 prikazuje maseni protok koji se u 120. sekundi morao povećati kako bi se temperatura stabilizirala na 23 °C unatoč povećanju snage. Temperatura servera je narasla s povećanjem snage što je prikazano slikom Sl. 5.14.



Sl. 5.12 Izlazna temperatura uz povećanje snage, s upravljanjem



Sl. 5.13 Upravljačka veličina uz povećanje snage



Sl. 5.14 Temperatura servera uz povećanje snage

5.4 Usporedba MPC i PID regulatora

U ovom dijelu će se analizirati i uspoređivati performanse prediktivnog regulatora (MPC) i klasičnog PID regulatora u upravljanju temperaturom u prostoru. Oba regulatora koriste izlaz iz nelinearnog ARX modela za upravljanje temperaturom, pri čemu se PID regulator koristi kao klasični pristup, a MPC kao napredniji, prediktivni pristup.

Proporcionalno-integralno-derivacijski regulator (PID odnosno tročlani regulator) je mehanizam upravljačke petlje koji koristi povratnu petlju. PID regulator je daleko najzastupljeniji regulator u industriji (pokriva 95% industrijskih primjena), a moguće su P, I, PI, PD, PID inačice djelovanja. PID regulator kontinuirano izračunava vrijednost pogreške kao razliku između željene zadane vrijednosti i izmjerene procesne varijable i primjenjuje korekciju temeljenu na proporcionalnom, integralnom i derivativnom članu. Formula regulatora u vremenskoj domeni je:

$$u(t) = K_p * e(t) + K_i * \int_0^t \mathrm{e}(\tau)\mathrm{d}\tau + K_d * \frac{de(t)}{dt} \qquad (5.1)$$

Gdje je:

u(t) izlaz PID regulatora u vremenskom trenutku t

e(t) = r(t) − y(t) pogreška (razlika između referentne vrijednosti r(t) i stvarnog izlaza y(t))

K_p proporcionalni koeficijent

K_i integralni koeficijent

K_d derivativni koeficijent

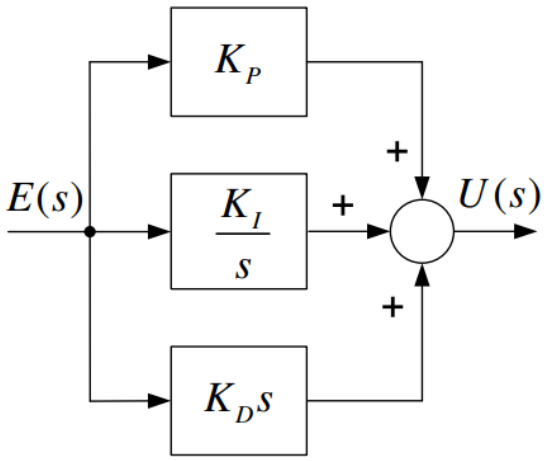
Na slici Sl. 5.15 je prikazan standardni PID regulator te se prema njoj može izvesti prijenosna funkcija idealnog PID regulatora:

$$G_r(s) = \frac{\mathrm{U(s)}}{E(s)} = K_p + \frac{K_i}{s} + K_d s = K_p(1 + \frac{1}{T_i s} + T_d s) \qquad (5.2)$$

Gdje je:

$T_i = \frac{\mathrm{Kp}}{Ki}$, integracijska vremenska konstanta

$T_d = \frac{\mathrm{Kd}}{Kp}$, derivativna vremenska konstanta

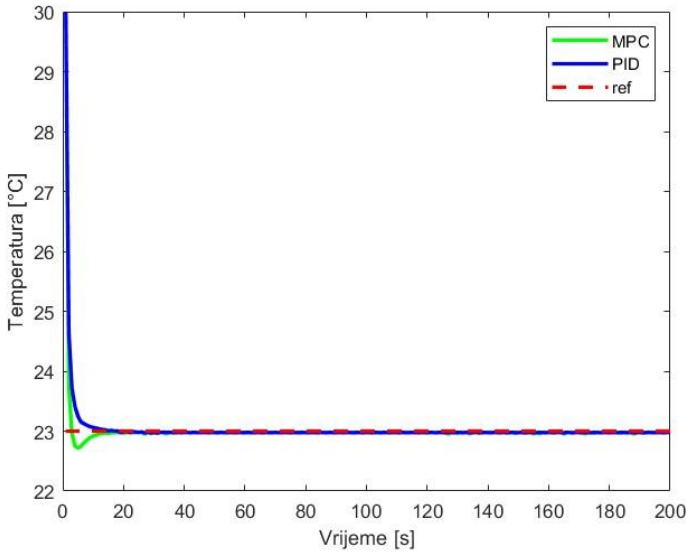


Sl. 5.15 Shema PID regulatora

Parametri regulatora K_p, T_i i T_d uobičajeno se mogu podešavati u određenom području vrijednosti. Izborom podesivih parametara regulatora, regulator se prilagođava vladanju procesa tako da se postigne najpovoljnije regulacijsko vladanje sustava. Proporcionalni dio reagira na trenutnu pogrešku. Veći K_p znači snažniji odgovor na pogrešku, što može ubrzati sustav, ali također može uzrokovati oscilacije ako je prevelik. Integrativni dio sumira pogrešku kroz vrijeme, a njegov cilj je eliminacija pogreške u ustaljenom stanju. Derivativni dio reagira na brzinu promjene pogreške. Pomaže u smanjenju oscilacija i prekomjernog podešavanja smanjenjem odziva na brze promjene pogreške.

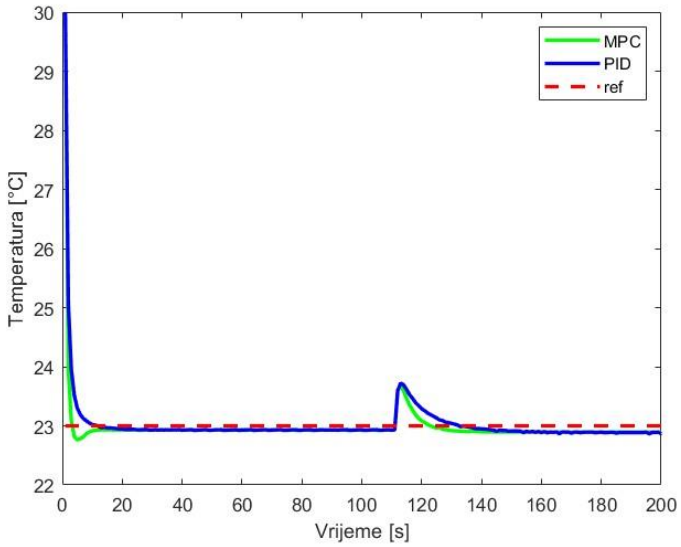
Modelski prediktivni regulator (MPC) za razliku od PID-a ne reagira samo na trenutnu pogrešku već uzima u obzir buduće stanje sustava preko predikcija i optimizacija. Parametri PID regulatora u ovom radu su određeni eksperimentalno, a princip rada MPC regulatora detaljnije je objašnjen u prethodnom poglavlju.

Slika Sl. 5.16 prikazuje rezultate simulacije u kojoj je snaga konstantna. Simulacija je prvo provedena s MPC, a zatim i PID upravljanjem. Odzivi obje simulacije prikazani su na jednoj slici s ciljem lakše usporedbe upravljanja. Prilikom ove simulacije snaga je postavljena na 130 000 W, a početni protok iznosi 30 kg/s.



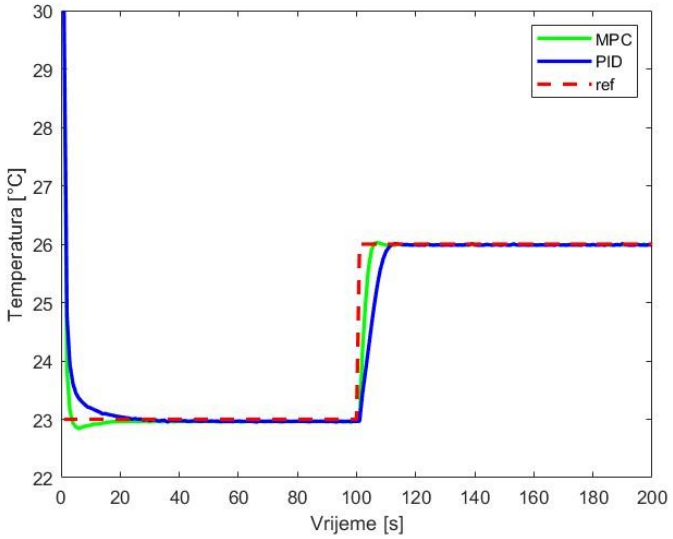
Sl. 5.16 Izlazna temperatura uz konstantnu snagu i referencu te MPC i PID upravljanje

Na slici Sl. 5.17 provedena je simulacija u kojoj je snaga 120 000 W, a u 110. sekundi se poveća na 165 000 W, a početni protok je 25 kg/s.



Sl. 5.17 Izlazna temperatura uz povećanje snage i konstantnu referencu te MPC i PID upravljanje

Slika Sl. 5.18 prikazuje rezultate simulacije kada snaga iznosi 150 000 W, a početni protok 31 kg/s, no referenca više nije konstanta nego step.

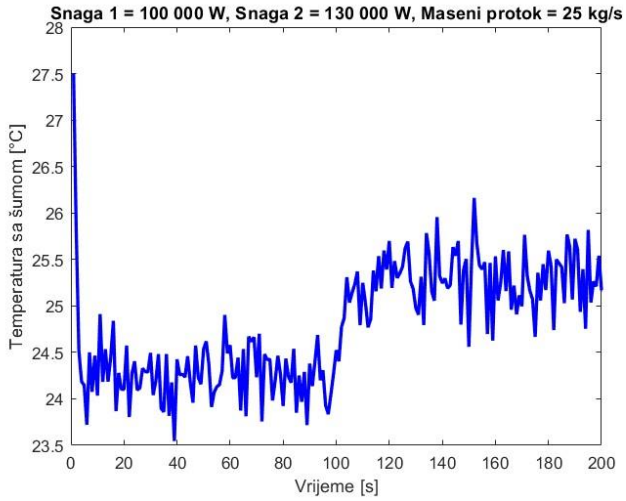


Sl. 5.18 Izlazna temperatura uz konstantnu snagu i step referencu te MPC i PID upravljanje

Usporedbom ovih grafova može se primijetiti kako u slučaju konstantne referentne vrijednosti i snage, oba regulatora pokazuju dobar odziv te se temperatura stabilizira na 23 °C . U slučaju porasta snage MPC se pokazao boljim za korištenje jer se brže vratio na referentnu vrijednost. Zahvaljujući prediktivnoj prirodi brzo reagira na promjene i sprječava oscilacije. U scenariju kada je ulazna snaga konstantna, ali referentna temperatura se mijenja u obliku step funkcije, MPC regulator je minimalno prešao iznad nove referentne vrijednosti prije nego što se stabilizirao, no stabilnost je postignuta brže nego s PID regulatorom. Također, treba uzeti u obzir kako bi drugačije vrijednosti parametara rezultirale drugačijim odzivima regulatora. MPC i PID regulator različito reguliraju temperaturu jer PID regulator reagira isključivo na trenutnu pogrešku između referentne i stvarne temperature. Njegov proporcionalni dio reagira na trenutnu pogrešku, integralni dio akumulira pogreške kroz vrijeme, a derivativni dio predviđa buduće pogreške na temelju brzine promjene. Međutim, PID ne uzima u obzir buduće ulaze ili složene dinamike sustava, što može dovesti do sporijeg odziva ili oscilacija. S druge strane, MPC regulator unaprijed računa optimalne akcije koje će minimizirati odstupanja u budućnosti. To omogućuje MPC-u da uzme u obzir buduće promjene u ulazima i dinamici sustava, čime osigurava preciznije i stabilnije upravljanje s manje oscilacija i bržim s vremenom stabilizacije.

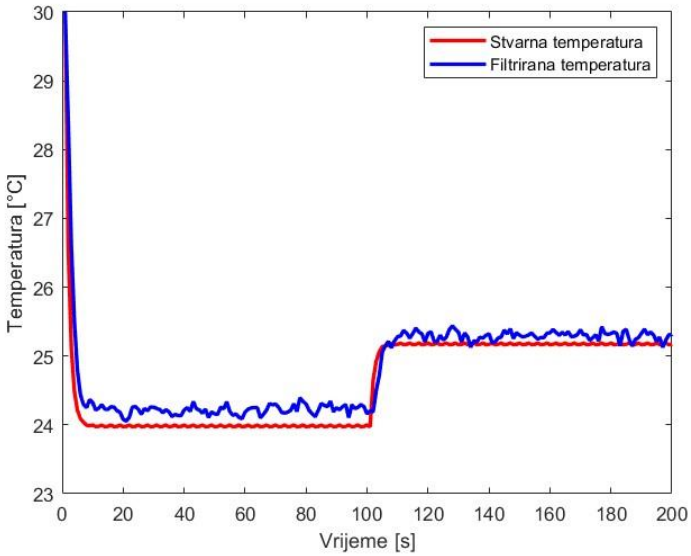
5.5 Uključivanje šuma mjerenja i primjena PT1 filtra

U svakom stvarnom sustavu mjerenja, prisutnost šuma je neizbježna. Šumovi u mjerenjima mogu nastati iz različitih izvora, poput senzorskih nesavršenosti, vanjskih utjecaja ili netočnosti u signalima. Kako bi sustav upravljanja u ovakvim uvjetima zadržao stabilnost i postigao točne performanse, potrebno je uključiti mehanizme za filtriranje šuma. U ovoj fazi simulacije uveden je šum na mjerenje temperature kako bi se ispitalo ponašanje sustava u uvjetima kad senzori ne pružaju idealno čiste podatke. Šum se dodaje izravno na izlaznu temperaturu predviđenu modelom sustava (NLARX model). Taj šum se generira korištenjem bloka *Band-Limited White Noise* u Simulinku, koji omogućuje prilagodbu karakteristika šuma kao što su snaga šuma (veća vrijednost rezultira jačim šumom), vrijeme uzorkovanja (određuje učestalost uzimanja uzoraka signala, treba biti dovoljno malo kako bi šum izgledao kontinuirano) i početna vrijednost (*engl. Seed*, početna vrijednost koja se koristi za generiranje pseudo-slučajnog šuma). U prvoj simulaciji prikazanoj na slici Sl. 5.19, analiziramo sustav bez primjene filtra. Nakon dodavanja šuma, sustav bilježi oscilacije u izmjerenoj temperaturi zbog utjecaja šuma. Ovakvi podaci mogu smanjiti učinkovitost MPC-a i dovesti do lošeg upravljanja.



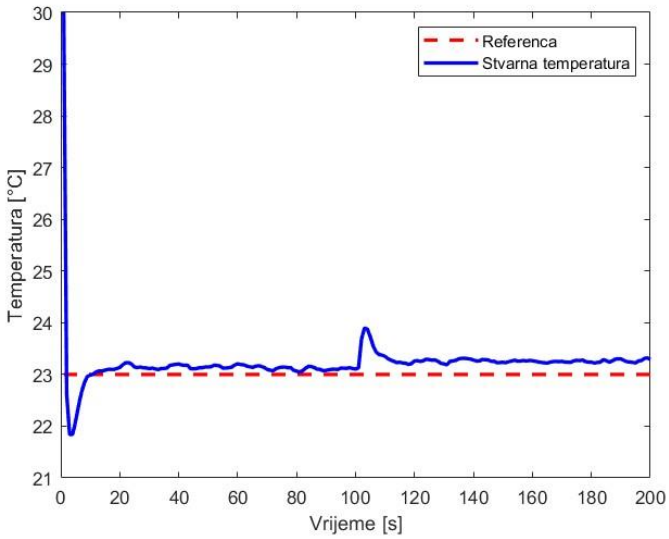
Sl. 5.19 Temperatura na izlazu NLARX modela sa šumom

U drugoj simulaciji prikazanoj slikom Sl. 5.20, šum na mjerenjima je filtriran pomoću PT1 člana (prvi red niskopropusnog filtra). PT1 član u Simulinku je definiran pomoću bloka *Transfer Fcn* formulom $\frac{1}{s+1}$. Nakon primjene filtra izlazna temperatura je stabilnija, a fluktuacije uzrokovane šumom su smanjene.



Sl. 5.20 Stvarna i filtrirana temperatura

Slika Sl. 5.21 prikazuje izlaznu temperaturu nakon primjene MPC regulatora. Cilj filtra je bio eliminirati visoke frekvencije i neželjene smetnje uzrokovane šumom u mjerenju temperature, kako bi se regulator mogao učinkovitije prilagoditi stvarnim uvjetima. Nakon primjene MPC-a, sustav pokazuje zadovoljavajuću stabilnost i brz odgovor. Na početku simulacije i prilikom promjene snage dolazi do pada odnosno skoka temperature, ali se se ona brzo stabilizira blizu referentne vrijednosti.



Sl. 5.21 Izlazna temperatura nakon upravljanja

6. Zaključak

U ovom diplomskom radu razvijen je i analiziran adaptivni sustav upravljanja temeljen na podacima, primijenjen na upravljanje temperaturom unutar HVAC prostora. Fokus istraživanja bio je na korištenju neuronskih mreža i prediktivnog upravljanja (MPC) za modeliranje i optimizaciju složenih nelinearnih procesa, čime se omogućuje precizna i efikasna kontrola u promjenjivim uvjetima. Kroz rad je pokazano da kombinacija matematičkog modeliranja i neuronskih mreža omogućuje uspješno hvatanje složenih odnosa unutar sustava, što je ključno za točno predviđanje izlaznih varijabli poput temperature. Neuronska mreža, trenirana na temelju podataka dobivenih iz simulacija matematičkog modela, uspješno je zamijenila klasične modele te je dokazano kako uz dovoljno kvalitetnih podataka nije potrebno znati matematički model jer neuronska mreža istrenirana na tim podacima predviđa izlaz koji je dovoljno blizu stvarnog izlaza modela. Primjena prediktivnog upravljanja (MPC) pokazala se učinkovitom u ovom slučaju. MPC predviđa buduća stanja te prilagođava akcije upravljanja kako bi postigao optimalne performanse. Rezultati su pokazali da MPC regulator, u svim simuliranim scenarijima (s konstantnom snagom, promjenom snage, i promjenom referentne vrijednosti) uspješno regulira temperaturu blizu referentne vrijednosti. Ovaj rad potvrđuje da adaptivni sustavi upravljanja temeljeni na podacima, uz upotrebu prediktivnog upravljanja, predstavljaju značajan korak naprijed u upravljanju složenih industrijskih procesa. Prediktivno upravljanje omogućuje fleksibilniji pristup koji se može prilagoditi promjenama u sustavu, dok neuronske mreže pružaju moćan alat za modeliranje i predviđanje u nelinearnim uvjetima. Kao budući koraci, predlaže se daljnje istraživanje u smjeru unapređenja neuronskih mreža kako bi se poboljšala točnost predikcija. Na primjer korištenje drugačije arhitekture neuronske mreže ili upotreba većih skupova podataka. Također, istraživanje se može nastaviti s fokusom na poboljšanje prediktivnog upravljanja koje će biti robusnije na vanjske smetnje i tako unaprijediti stabilnost i pouzdanost upravljanja. Ovaj rad ukazuje na potencijal adaptivnih sustava upravljanja temeljenih na podacima, što otvara vrata za njihovu primjenu na širok spektar složenih industrijskih procesa, gdje bi mogli donijeti značajna poboljšanja u efikasnosti, preciznosti i prilagodljivosti upravljanja.

Literatura

[1] Coulson, J., Lygeros, J., Dörfler, F. *Data-Enabled Predictive Control: In the Shallows of the DeePC*, arXiv:1811.05890v2 [math.OC], 2019

[2] Choi, Y., Park, B., Hyun, J., Moon, J. *Development of an adaptive artificial neural network model and optimal control algorithm for a data center cyber–physical system,* Building and Environment, Volume 210, 108704, 2022

[3] Schwenzer, M., Ay, M., Bergs, T., Abel, D. *Review on model predictive control: an engineering perspective*, The Int. J. of Advanced Manufacturing Technology, vol. 117, pp. 1327-1349, 2021

[4] Mesbah, A., Wabersich, K., Schoellig, A., Zeilinger, M., Lucia, S., Badgwell, T., Paulson, J. *Fusion of Machine Learning and MPC under Uncertainty: What Advances Are on the Horizon?,* 2022 American Control Conference (ACC) Atlanta, 2022

[5] MathWorks, *Model Predictive Control Toolbox*, (1994-2023), Poveznica: https://www.mathworks.com/help/mpc/

[6] MathWorks, *System Identification Toolbox,* (1994-2023), Poveznica:

https://www.mathworks.com/help/ident/index.html?s_tid=CRUX_lftnav

[7] MathWorks, *Design Controller Using MPC Designer,* (1994-2023), Poveznica:

https://www.mathworks.com/help/mpc/gs/introduction.html

[8] MathWorks, *idnlarx*, (1994-2023), Poveznica:

https://www.mathworks.com/help/ident/ref/idnlarx.html

[9] Wikipedia, *Model predictive control*, Poveznica: https://en.wikipedia.org/w/index.php?title=Model_predictive_control&oldid=1148678166

[10] Wikipedia, *Neural network (machine learning)*, Poveznica:

https://en.wikipedia.org/wiki/Neural_network_(machine_learning)

Sažetak

Ovaj diplomski rad bavi se primjenom prediktivnog upravljanja (Model Predictive Control, MPC) u upravljanju temperaturom unutar podatkovnih centara. Na početku je proces modeliran pomoću diferencijalnih jednadžbi koje opisuju toplinsku dinamiku prostora, uzimajući u obzir prijenos topline kroz kondukciju i konvekciju. Nakon toga provedene su simulacije u MATLAB-u kako bi se ispitao odziv sustava u različitim uvjetima, uključujući prisutnost šuma mjerenja. Za identifikaciju sustava korištena je neuronska mreža, čime je omogućen pristup temeljen na podacima za predviđanje temperature na temelju masenog protoka i snage. Dodatno, dizajniran je i testiran MPC regulator, pri čemu je uspješno upravljanje temperaturom prikazano kroz simulacije. Ponašanje sustava analizirano je i u prisutnosti šuma, gdje su mjerene vrijednosti filtrirane kako bi se poboljšala točnost upravljanja. Rezultati su pokazali da predloženi pristup uspješno održava temperaturu prostora blizu zadane referentne vrijednosti, čak i u prisutnosti smetnji.

Ključne riječi: upravljanje zasnovano na podacima, prediktivno upravljanje, matematički model procesa, neuronska mreža

Summary

This thesis deals with the application of Model Predictive Control (MPC) in temperature control within data centers. Initially, the process was modeled using differential equations that describe the thermal dynamics of the space, taking into account heat transfer through conduction and convection. Subsequently, simulations were conducted in MATLAB to examine the system's response under various conditions, including the presence of measurement noise. A neural network was used for system identification, enabling a data-driven approach for predicting temperature based on mass flow and power. Additionally, an MPC controller was designed and tested, demonstrating successful temperature control through simulations. The system's behavior was analyzed in the presence of noise, where the measured values were filtered to improve control accuracy. The results showed that the proposed approach successfully maintains the room temperature close to the desired setpoint, even in the presence of disturbances.

Keywords: data-driven control, predictive control, mathematical model of the process, neural network