

# Modelsko prediktivno upravljanje procesom hlađenja zatvorenog prostora

---

Marvin, Mikael

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:168:105509>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-03-29**



*Repository / Repozitorij:*

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repository](#)



SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 83

**MODELSKO PREDIKTIVNO UPRAVLJANJE PROCESOM  
HLAĐENJA ZATVORENOG PROSTORA**

Mikael Marvin

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 83

**MODELSKO PREDIKTIVNO UPRAVLJANJE PROCESOM  
HLAĐENJA ZATVORENOG PROSTORA**

Mikael Marvin

Zagreb, lipanj 2024.

## DIPLOMSKI ZADATAK br. 83

Pristupnik: **Mikael Marvin (0036526193)**  
Studij: Informacijska i komunikacijska tehnologija  
Profil: Automatika i robotika  
Mentor: prof. dr. sc. Stjepan Bogdan

Zadatak: **Modelsko prediktivno upravljanje procesom hlađenja zatvorenog prostora**

### Opis zadatka:

Modelsko prediktivno upravljanje (engl. Model Predictive Control, MPC) je metoda upravljanja procesima koja koristi dinamički model procesa te primjenjuje optimizacijski postupak za određivanje vrijednosti upravljačkih varijabli. Za primjenu MPC-a potrebno je mjeriti ili estimirati sve varijable stanja dinamičkog modela. Razmatrani proces je hlađenje zatvorenog prostora, pri čemu je upravljačka veličina brzina vrtnje ventilatora za hlađenje. Kao poremećaj mogu djelovati otvaranje i zatvaranje vrata ili prisutnost grijaćih elemenata u prostoru. Potrebno je u Matlabu simulirati model hlađenja zatvorenog prostora dostupan u literaturi te za njega implementirati upravljanje primjenom MPC-a. Analizirati ponašanje sustava za promjene referentnih i poremećajnih veličine.

Rok za predaju rada: 28. lipnja 2024.

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. xxxx

**Modelsko prediktivno upravljanje  
procesom hlađenja zatvorenog prostora**

Mikael Marvin

Prazna stranica.

## Sadržaj

Uvod .....	1
1. Matematičko modeliranje .....	2
1.1. Modeliranje temperature.....	2
1.2. Simulink model temperature .....	4
1.3. Odziv temperature .....	5
1.4. Pojednostavljeni model podatkovnog centra .....	7
1.5. Model temperature i pritiska.....	8
1.6. Kombiniranje modela .....	11
1.7. Linearizacija modela.....	11
2. Modelsko prediktivno upravljanje.....	14
2.1. Teorija.....	14
2.1.1. Pregled s više razine .....	14
2.1.2. Blok MPC-a.....	15
2.2. Implementacija .....	17
2.2.1. Definiranje optimizacijskog problema u YALMIP-u.....	17
2.2.2. Simulacija .....	18
2.3. Kriterijska funkcija MPC hlađenja podatkovnog centra .....	20
3. Rezultati implementacije na sobu podatkovnog centra .....	23
3.1. Odziv .....	23
3.2. Utjecaj parametara na odziv simulacije.....	24
3.3. Odziv s modificiranom kriterijskom funkcijom .....	26
3.4. Odziv uz varijabilni poremećaj .....	29
3.5. Usporedba s PID upravljanjem.....	30
Zaključak .....	32
Literatura .....	33

Sažetak.....	34
Summary.....	35



# Uvod

Učinkovito hlađenje zatvorenih prostora, poput podatkovnih centara, postaje sve važnije zbog povećanih zahtjeva za računalnim kapacitetima i visoke gustoće elektroničke opreme, što uzrokuje značajan porast temperature i povećava rizik od pregrijavanja. Ključni izazovi u ovom području uključuju održavanje stabilne temperature i razlike tlaka, uzimajući u obzir tehnička ograničenja sustava.

Modelsko prediktivno upravljanje (MPC) ističe se kao prikladan pristup upravljanju hlađenjem jer omogućava optimizaciju u realnom vremenu, uzimajući u obzir dinamičke promjene i ograničenja sustava. Ovaj rad bavi se razvojem matematičkog modela temperature i pritiska unutar zatvorenog prostora te implementacijom MPC algoritma za optimizaciju upravljanja.

Prvo poglavlje obrađuje matematičko modeliranje temperature i pritiska, dok se u drugom poglavlju detaljno opisuje teorija i primjena MPC-a. Treće poglavlje posvećeno je analizi rezultata implementacije MPC-a, s posebnim naglaskom na utjecaj različitih parametara i usporedbu s klasičnim metodama poput PID regulacije. Rezultati simulacija ukazuju na značajna poboljšanja u preciznosti i učinkovitosti upravljanja hlađenjem prostorije podatkovnog centra.

Zaključak potvrđuje da MPC donosi značajne prednosti u upravljanju hlađenjem, no daljnja istraživanja potrebna su za optimizaciju modela i primjenu u praksi

# 1. Matematičko modeliranje

Općenito, matematičko modeliranje procesa vrlo je važan korak u upravljanju procesom te glavna svrha matematičkog modeliranja procesa je analiza sustava te sinteza sustava upravljanja.

## 1.1. Modeliranje temperature

Za modeliranje temperature [1], u ovom radu koristio se prvi zakon termodinamike u obliku toka energije za otvorene sustave:

$$\frac{d}{dt} E_{sys} = E_{in} - E_{out} \quad (1)$$

$E_{in}$  predstavlja energiju koja ulazi u sustav, dok  $E_{out}$  predstavlja energiju koja izlazi iz sustava.  $E_{sys}$  je energija sustava koji se promatra, može se dalje rastaviti te uzeti u obzir „tromost“ sustava koja će kasnije najviše određivati brzinu odziva sustava:

$$E = mc\Delta T \quad (2)$$

Gdje je  $m$  masa tijela, a  $c$  specifični toplinski kapacitet materijala. Uzmemo li derivaciju energije, jedina varijabla koja nije konstanta je temperatura  $\Delta T$ , koja nam je u ovom slučaju zanimljiva.

Kod reguliranja temperature podatkovne sobe, potrebna nam je informacija o temperaturama. U ovom slučaju dvije su varijable stanja odabrane, temperatura sobe i temperatura servera (računala). Uz ove dvije varijable stanja slijedi sustav diferencijalnih jednažbi [2] koje je potrebno izvesti:

$$k_{server} \frac{d}{dt} T_{server} = W - \dot{Q}_{kond} \quad (3)$$

$$k_{soba} \frac{d}{dt} T_{soba} = \dot{Q}_{kond} + \dot{m}_{in} c_{zr} T_{in} - \dot{m}_{out} c_{zr} T_{soba} \quad (4)$$

Gdje  $W$  predstavlja energiju koju dovode računala svojim radom, u jednažbi (3),  $\dot{Q}_{kond}$  predstavlja gubitak energije iz sustava servera – gubitak energije putem kondukcije,  $T_{server}$  predstavlja temperaturu servera,  $T_{soba}$  predstavlja temperaturu sobe,  $\dot{m}_{in}$  predstavlja protok zraka koji ulazi u sobu,  $\dot{m}_{out}$  predstavlja protok zraka koji izlazi iz sobe, a  $c_{zr}$  je specifična toplinska konstanta zraka, te  $k_{server}$  i  $k_{soba}$  su varijable koje predstavljaju toplinski kapacitet zraka u prostoriji i serveru. Ovi su parametri ključni za modeliranje kako brzo ili

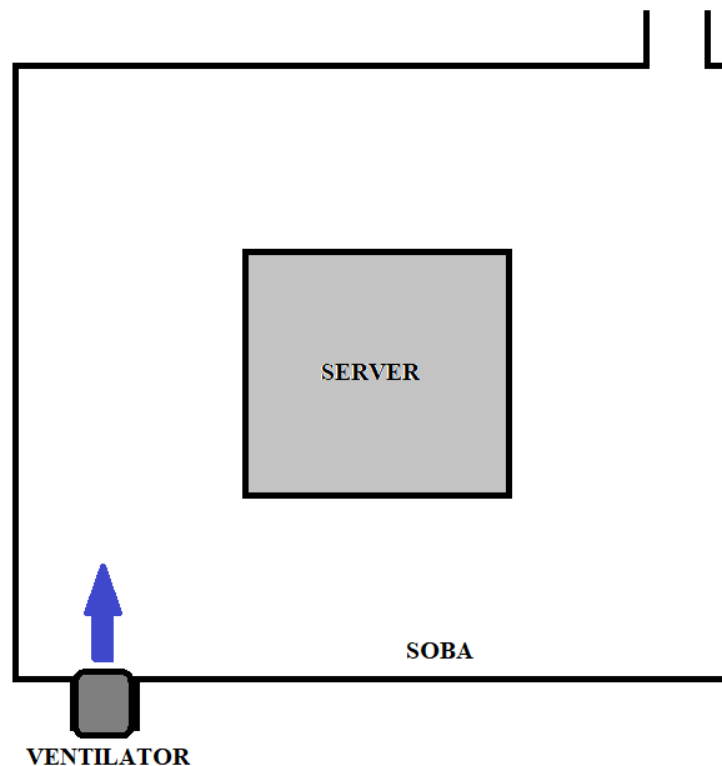
sporo zrak u prostoriji ili u serveru reagira na promjene u temperaturi zbog vanjskih faktora ili hlađenja. Izvedu li se jednadžbe razinu više, dobivamo:

$$k_{server} \frac{d}{dt} T_{server} = W - kA(T_{server} - T_{soba}) \quad (5)$$

$$k_{soba} \frac{d}{dt} T_{soba} = kA(T_{server} - T_{soba}) + \dot{m}_{in} c_{zr} T_{in} - \dot{m}_{out} c_{zr} T_{soba} \quad (6)$$

Gdje je konstanta  $k$  toplinska kondukcija aluminijske kućišta servera (pretpostavka da su kućišta servera napravljena od aluminijske), a konstanta  $A$  ukupna površina servera.

U ovom slučaju energija kondukcije je uzeta u obzir. Osim varijabli stanja, postoje i upravljačke varijable. U ovom procesu upravljamo ventilatorom koji je ugrađen na jednoj strani sobe. Brzina vrtnje ventilatora u okretajima po minuti definira maseni protok zraka kroz sobu. Maseni protok zraka se može rastaviti u empirijski određenu konstantu (koja ovisi o površini i lopaticama ventilatora) koja se množi brojem okretaja.



Slika 1 Prikaz pojednostavljenog otvorenog sustava sobe

Na slici (Slika 1) možemo primijetiti kako varijable stanja zapravo opisuju temperaturu dva odvojena tijela u otvorenom sustavu. Zrak koji ulazi kao posljedica ventilatora (rashlađeni zrak) na neki način mora izaći ista količina s druge strane – posljedično, varijable  $\dot{m}_{in}$  i  $\dot{m}_{out}$  su zapravo jednake te se druga jednadžba (6), može dalje pojednostavniti:

$$k_{server} \frac{d}{dt} T_{server} = W - kA(T_{server} - T_{soba}) \quad (7)$$

$$k_{soba} \frac{d}{dt} T_{soba} = kA(T_{server} - T_{soba}) + \dot{m}_{in} c_{zr} (T_{in} - T_{soba}) \quad (8)$$

Osim pojednostavljenja protoka zraka  $\dot{m}$ , moguće je dalje rastaviti protok  $\dot{m}_{in}$ :

$$\dot{m}_{in} = k_{vent} U \quad (9)$$

Gdje  $U$  predstavlja broj okretaja u minuti (eng. rpm – rotations per minute), dok  $k_{vent}$  predstavlja konstantu koja povezuje brzinu okretaja ventilatora i maseni protok.

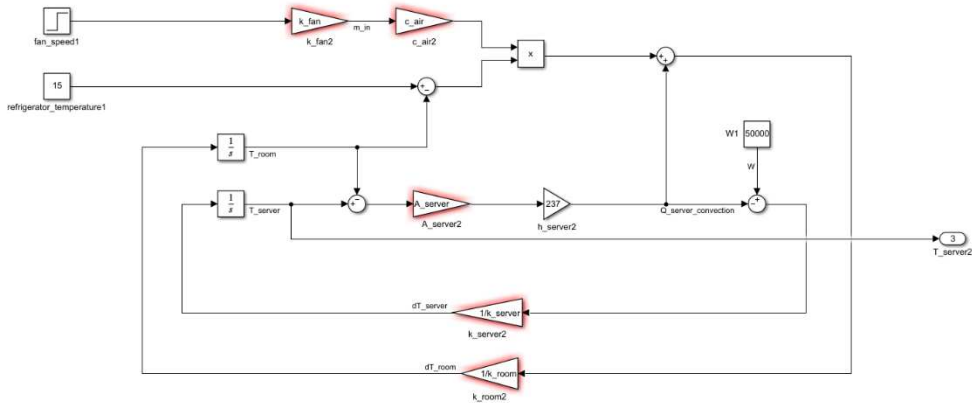
$$k_{server} \frac{d}{dt} T_{server} = W - kA(T_{server} - T_{soba}) \quad (10)$$

$$k_{soba} \frac{d}{dt} T_{soba} = kA(T_{server} - T_{soba}) + k_{vent} c_{zr} (T_{in} - T_{soba}) U \quad (11)$$

Jednadžbama (10) i (11) opisan je pojednostavljeni model temperature.

## 1.2. Simulink model temperature

Za validaciju i podešavanje odziva modela temperature korišten je simulink zajedno s matlabom. U nastavku prikazan je simulink model te zahtjevi na model.



Slika 2 Blokovski dijagram modela sobe (temperatura)

S lijeve strane slike (Slika 2) mogu se primijetiti dva integratora, nakon kojih signal predstavlja temperaturu servera i temperaturu sobe. Blok dijagram je korišten za podešavanje konstanti i provjeru stabilnosti modela. Slijedi model u obliku prostora stanja (A i B matrice):

$$\begin{bmatrix} k_{server} \frac{d}{dt} T_{server} \\ k_{soba} \frac{d}{dt} T_{soba} \end{bmatrix} = \begin{bmatrix} -kA & kA \\ kA & -kA - k_{fan} c_{zr} U \end{bmatrix} \begin{bmatrix} T_{server} \\ T_{soba} \end{bmatrix} + \begin{bmatrix} 0 \\ k_{fan} c_{zr} T_{in} \end{bmatrix} U + \begin{bmatrix} W \\ 0 \end{bmatrix} \quad (12)$$

Može se primijetiti da je sustav nelinearan, A matrica ovisi o kontrolnom signalu U, te čini matricu A sustavom koji se mijenja u vremenu.

### 1.3. Odziv temperature

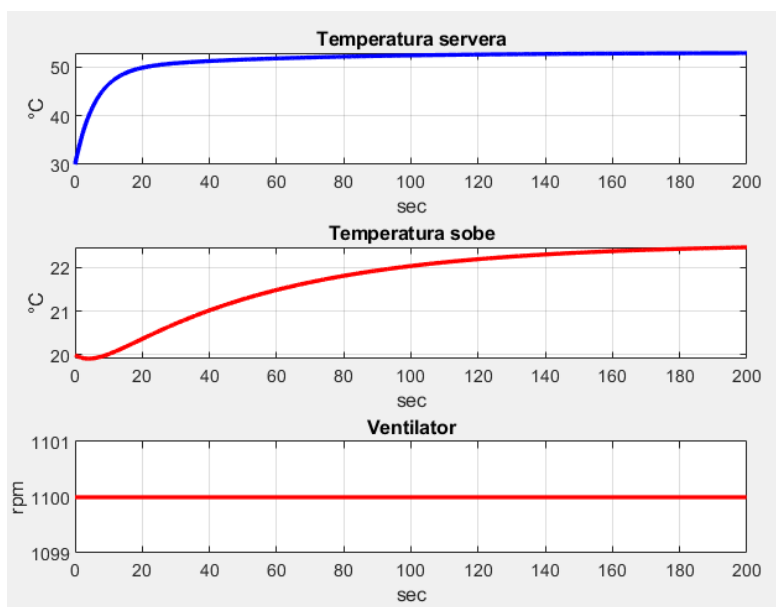
Za podešavanje parametara odziva korišteni su podaci izmjereni na realnom sustavu za zadanu snagu koju proizvode serveri te za zadani broj okretaja ventilatora:

- Vrijeme smirivanja od otprilike 60 sekundi za temperaturu servera
- Razlika između temperature servera i sobe oko 10 stupnjeva u stacionarnom stanju

Zbog nelinearnosti sustava i ovisnosti sustava o upravljačkom signalu nije moguće dobiti egzaktne vrijednosti zbog promjene A matrice kroz vrijeme.

Za simuliranje odziva temperature u praksi se koristio simulink, a nelinearnost A matrice je kompenzirana na način da se svakim korakom ažurira matrica ako se kontrolni signal mijenjao.

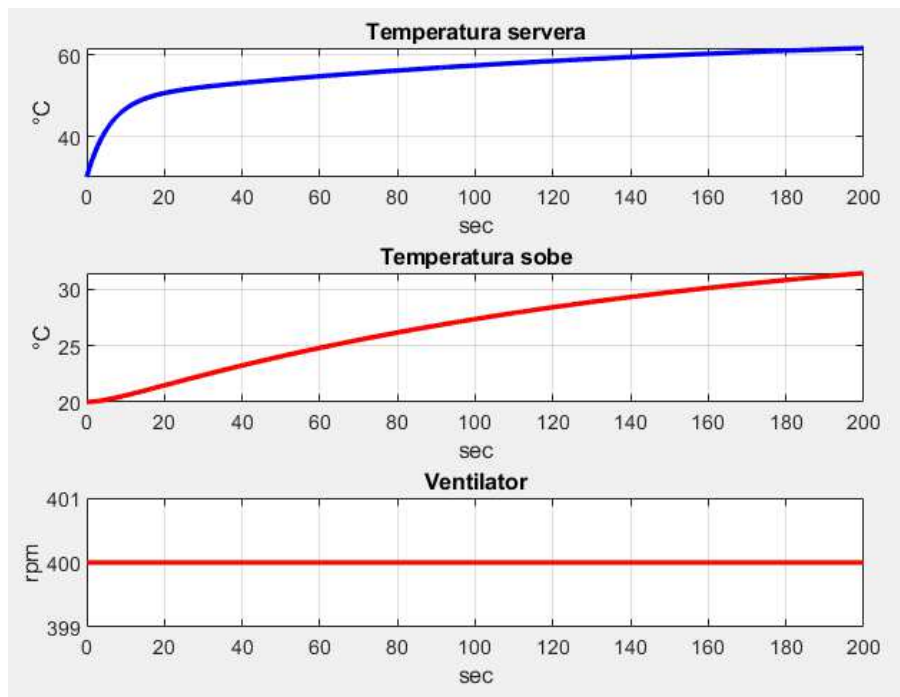
Na slici (Slika 3) je primjer odziva temperatura servera i sobe uz skokoviti kontrolni signal od nule do 1100 okretaja po minuti za snagu servera od 50 kW.



Slika 3 Odziv temperatura na skokovitu pobudu

Također uz sljedeću sliku (Slika 4) odziva može se primijetiti kako vrijeme smirivanja temperature nije fiksno već se mijenja ovisno o upravljačkom signalu. Na prethodnoj slici

(Slika 3) može se primijetiti, na početku simulacije, kako odziv temperature sobe se prvo kreće prema nižim temperaturama, te tek nakon par sekundi promijeni smjer prema višim temperaturama. Ovo se događa jer je sustav višeg reda te postoji kašnjenje između energije koja stiže iz servera, dok energija koja se oduzima preko ventilatora je gotovo instantna.

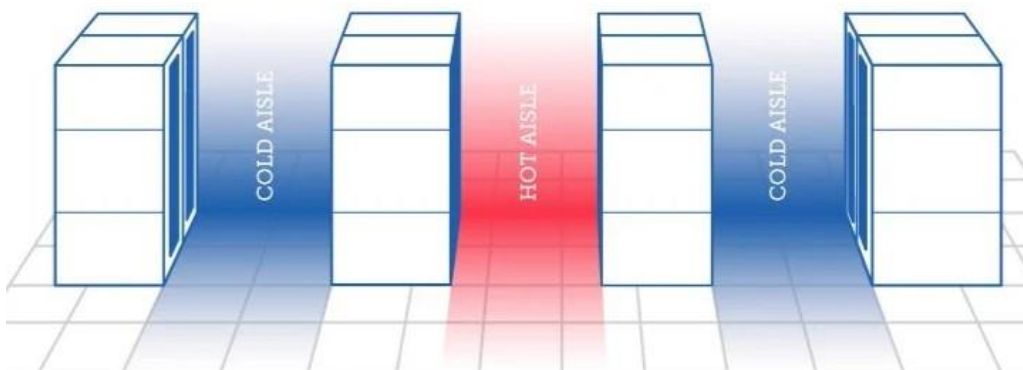


Slika 4 Odziv temperatura

## 1.4. Pojednostavljeni model podatkovnog centra

Općenito kod dizajniranja geometrije sobe u kojoj će se nalaziti server računala, nastoji se organizirati računala u redove visokih ormara. Najveća prednost takve organizacije jest stvaranje toplih i hladnih područja, naizmjenično.

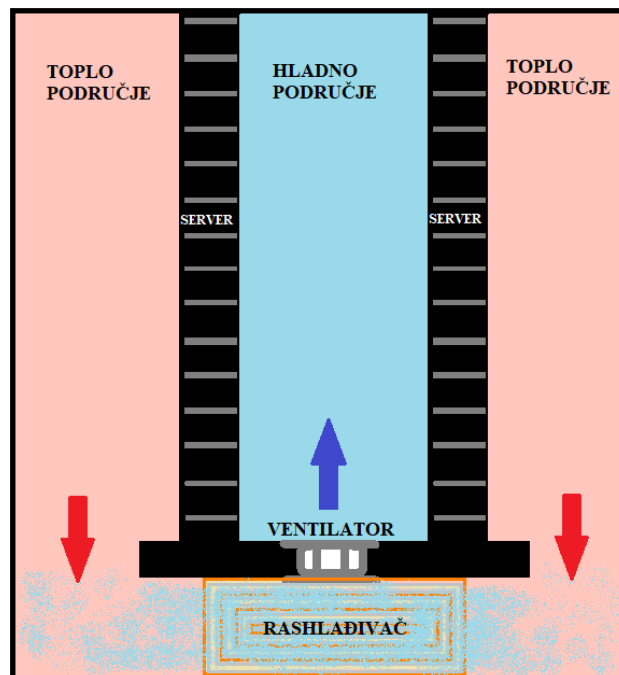
Na slici (Slika 5) nalazi se primjer stvaranja takozvanih eng. hot and cold aisles.



Slika 5 Osnovna organizacija za topla i hladna područja

Svaki red predstavlja skupinu računala koje s jedne strane usisava hladan zrak iz hladnog područja, te nakon što ohladi kritične komponente izbacuje topli zrak u područje toplog zraka. Kao pojednostavljenje zatvorene sobe postavljeno je niz pretpostavki:

- Dva reda servera koja odijeljuju sobu u tri područja: u sredini hladno područje, s lijeve i desne strane, simetrično dva topla područja.
- Na jednom kraju sobe, u sredini, nalazi se ventilator, koji tjera hladan zrak konstantne temperature u središnje, hladno područje.
- Temperatura središnjeg, hladnog područja je homogena, tj. ista je, bezobzira na visinu i lokaciju unutar istog područja.
- Topla područja su simetrična, temperatura je ista i homogena.
- Iza ventilatora, topli zrak ulazi u rashlađivač, topli zrak se hladi na temperaturu hladnog područja te ulazi u hladno područje.
- Geometrija prostora je zanemarena.
- **Temperatura hladnog područja je konstanta od 15°C.**
- **Temperatura toplog područja računa se po modelu prijašnjeg potpoglavlja te je jednaka temperaturi sobe**
- **Temperatura servera se ne mjeri u stvarnosti odnosno nemamo informaciju o njoj, ali potrebna je za upotpunjeni model temperature.**



Slika 6 Izgled zatvorene sobe

Na slici (Slika 6) možemo vidjeti organizaciju zatvorene sobe te kako izgledaju glavne strujnice zraka.

## 1.5. Model temperature i pritiska

Kod modeliranja pojednostavljenog modela podatkovnog centra koristio se model temperature iz prijašnjeg potpoglavlja uz nekoliko promjena te dodatna varijabla stanja – pritisak.

Za primjenu MPC-a, potrebno je imati informaciju o razlici između toplog i hladnog područja te razlike tlaka između toplog i hladnog područja. Iz tog razloga dodan je približni model pritiska između hladnog i toplog područja, na koji će u sljedećim poglavljima biti postavljeni neki zahtjevi. Osim pritiska model temperature nije promijenjen, ali drugačije se definiraju varijable stanja.

Kako bismo dobili temperaturu središnjeg hladnog područja relativno jednostavnim putem za potrebe ovog rada, temperatura je jednaka temperaturi na kojoj rashlađivač hladi topli zrak koji se natrag ventilatorom vraća u prostoriju. Ta temperatura je konstanta od 15°C.



Temperatura toplog područja se računa isto kao i u prošlom potpoglavlju, odnosno temperatura toplog područja je jednaka temperaturi sobe.

U sljedećim jednadžbama prikazan je model temperature za pojednostavljeni podatkovni centar :

$$\begin{bmatrix} k_{server} \frac{d}{dt} T_{server} \\ k_{soba} \frac{d}{dt} T_{toplo} \end{bmatrix} = \begin{bmatrix} -kA & kA \\ kA & -kA - k_{fan} C_{zr} U \end{bmatrix} \begin{bmatrix} T_{server} \\ T_{toplo} \end{bmatrix} + \begin{bmatrix} 0 \\ k_{fan} C_{zr} T_{hladno} \end{bmatrix} U + \begin{bmatrix} W \\ 0 \end{bmatrix} \quad (13)$$

Model pritiska dobiven je dodavanjem PT1 kašnjenje stacionarnom stanju koje je odgovaralo realnim vrijednostima sobe.

U sljedećim jednadžbama prikazan je proces dobivanja modela razlike pritiska toplog i hladnog područja u obliku modela u prostoru stanja:

$$\Delta P(s) = \frac{k_p U(s)}{s + \frac{1}{T}} U(s) \quad (14)$$

Gdje  $k_p$  predstavlja konstantu koja određuje iznos razlike pritiska u stacionarnom stanju, a konstanta  $T$  je vremenska konstanta modela, te određuje vrijeme pri kojem pritisak dostiže oko 63% iznosa stacionarnog stanja pri stepeničastoj pobudi.

Jednadžba (14) predstavlja model pritiska u obliku prijenosne funkcije u Laplaceovoj domeni. Model treba prebaciti u model prostora stanja:

$$\Delta P(s) = \frac{k_p U(s)}{s + \frac{1}{T}} U(s) \quad (15)$$

$$\Delta P(s) s + \frac{1}{T} \Delta P(s) = k_p U^2(s) \quad (16)$$

$$s \Delta P(s) = -\frac{1}{T} \Delta P(s) + k_p U(s) U(s) \quad (17)$$

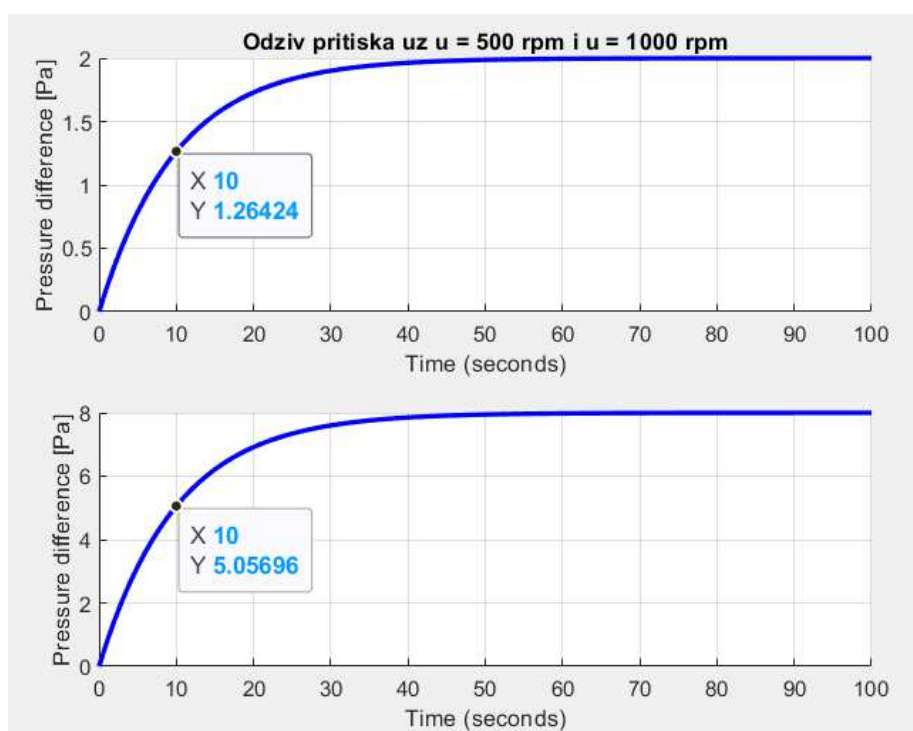
Jednadžba (17) predstavlja oblik koji se može direktno prebaciti u vremensku domenu:

$$\frac{d}{dt} \Delta p(t) = -\frac{1}{T} \Delta p(t) + k_p u(t) u(t) \quad (18)$$

Iz jednadžbe (18) direktno se mogu pročitati A i B matrice stanja. Model je opet nelinearan jer B matrica ovisi o kontrolnom signalu (nepoznata funkcija) u vremenu.

Parametri korišteni za model pritiska odabrani su tako da odziv odgovara mjerenjima realnog sustava:

- Vremenska konstanta od 10 sekundi ( $T = 10$ )
- Ugoditi  $k_p$  parametar tako da se otprilike dobiju sljedeći iznosi u stacionarnom stanju:
  - $\Delta p_{stac}(u = 500) = 1 \text{ Pa}$
  - $\Delta p_{stac}(u = 1000) = 10 \text{ Pa}$



Slika 7 Odzivi razlike pritiska

Na slici (Slika 7) prikazana su dva odziva modela razlike pritiska, jasno je označena točka u desetoj sekundi, vremenska konstanta modela, dobiva se 63.2% stacionarne vrijednosti u desetoj sekundi. Osim provjere vremenske konstante, otprilike je podešen parametar za dobivanje stacionarnog stanja iznosa 1 Pa i 10 Pa.

## 1.6. Kombiniranje modela

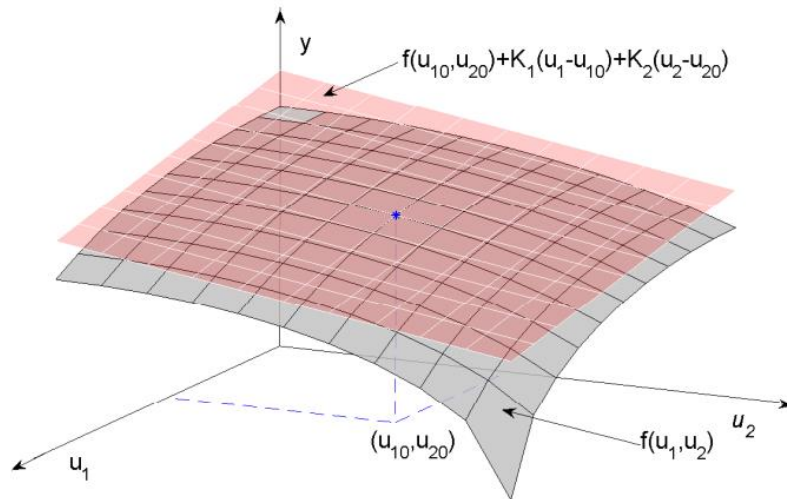
Za korištenje modela pojednostavljenog podatkovnog centra u upravljanju bilo kojom varijablom stanja potrebno je kombinirati modele u jednu matricu sustava (A matrica).

$$\begin{bmatrix} k_{server} \frac{d}{dt} T_{server} \\ k_{soba} \frac{d}{dt} T_{toplo} \\ \frac{d}{dt} \Delta p(t) \end{bmatrix} = \begin{bmatrix} -kA & kA & 0 \\ kA & -kA - k_{fan} C_{zr} U & 0 \\ 0 & 0 & -\frac{1}{T} \end{bmatrix} \begin{bmatrix} T_{server} \\ T_{toplo} \\ \Delta p \end{bmatrix} + \begin{bmatrix} 0 \\ k_{fan} C_{zr} T_{hladno} \\ k_p U \end{bmatrix} + \begin{bmatrix} W \\ 0 \\ 0 \end{bmatrix} \quad (19)$$

## 1.7. Linearizacija modela

Trenutni model je nelinearan, a krajnji cilj rada je upravljanje modelom uz pomoć modelsko prediktivnog algoritma. Da bi se to postiglo, potrebno je linearizirati nelinearni model sustava hlađenja sobe. Linearizacija omogućuje primjenu MPC (eng. Model Predictive Control) tehnika koje su računalno učinkovitije i jednostavnije za implementaciju. Postupak linearizacije temelji se na odabiru radne točke oko koje se model linearizira korištenjem Taylorovog reda prvog stupnja, čime se dobivaju linearne jednadžbe koje opisuju dinamiku sustava u blizini te točke.

Grafička interpretacija linearizacije:



Slika 8 Linearizacija

Na slici (Slika 8) se može vidjeti kako u u blizini točke  $(u_{10}, u_{20})$  linearizirani model je vrlo blizu nelinearnom modelu, te udaljavanjem je greška sve veća.

Da bismo postigli linearizaciju, potrebno je odabrati radnu točku  $x^* = (T^*_{server}, T^*_{toplo}, U^*)$  oko koje ćemo linearnizirati. Koristeći Taylorovu ekspanziju prvog reda, dobivamo linearizirane jednadžbe koje opisuju dinamiku sustava u blizini ove radne točke. Ekspanzija funkcije u Taylorov red prvog stupnja oko radne točke može se prikazati kao:

$$f(x) \approx f(x^*) + \left. \frac{\partial f}{\partial x} \right|_{x=x^*} (x - x^*) \quad (20)$$

Sljedeće jednadžbe opisuju linearizaciju oko proizvoljne točke:

$$\frac{d}{dt} T_{server} = \frac{W(t)}{k_{server}} - \frac{kA}{k_{server}} (T_{server} - T_{toplo}) = f_1 \quad (21)$$

$$\frac{d}{dt} T_{toplo} = \frac{kA}{k_{toplo}} (T_{server} - T_{toplo}) + \frac{k_{fan} C_{Zr}}{k_{toplo}} (T_{hladno} - T_{toplo}) U = f_2 \quad (22)$$

$$\frac{d}{dt} \Delta p(t) = -\frac{1}{T} \Delta p(t) + k_p U(t) U(t) = f_3 \quad (23)$$

Primijenimo li jednadžbu (20) na (21), (22) i (23):

$$\frac{d}{dt} \Delta T_{server} = f_1(x^*) + \left. \frac{\partial f_1}{\partial T_{server}} \right|_{x=x^*} (T_{server} - T^*_{server}) + \left. \frac{\partial f_1}{\partial T_{toplo}} \right|_{x=x^*} (T_{toplo} - T^*_{toplo}) \quad (24)$$

$$\begin{aligned} \frac{d}{dt} \Delta T_{toplo} = f_2(x^*) + \left. \frac{\partial f_2}{\partial T_{server}} \right|_{x=x^*} (T_{server} - T^*_{server}) + \left. \frac{\partial f_2}{\partial T_{toplo}} \right|_{x=x^*} (T_{toplo} - T^*_{toplo}) \\ + \left. \frac{\partial f_2}{\partial U} \right|_{x=x^*} (U - U^*) \quad (25) \end{aligned}$$

$$\frac{d}{dt} \Delta p = f_3(x^*) + \left. \frac{\partial f_3}{\partial p} \right|_{x=x^*} (p - p^*) + \left. \frac{\partial f_3}{\partial U} \right|_{x=x^*} (U - U^*) \quad (26)$$

Pretpostavljamo da se linearizacija obavlja u stacionarnoj točki ( $\frac{df}{dt} = 0$ ) pa se konstantni član Taylorovog reda izjednači s nulom:

$$\frac{d}{dt} \Delta T_{server} = -\frac{kA}{k_{server}} (T_{server} - T^*_{server}) + \frac{kA}{k_{server}} (T_{toplo} - T^*_{toplo}) \quad (27)$$

$$\begin{aligned} \frac{d}{dt} \Delta T_{toplo} = \frac{kA}{k_{toplo}} (T_{server} - T^*_{server}) + \frac{-kA - k_{fan} C_{Zr} U^*}{k_{toplo}} (T_{toplo} - T^*_{toplo}) + \\ \frac{k_{fan} C_{Zr}}{k_{toplo}} (T_{hladno} - T^*_{toplo}) (U - U^*) \quad (28) \end{aligned}$$

$$\frac{d}{dt} \Delta p = -\frac{1}{T} (p - p^*) + 2k_p U^* (U - U^*) \quad (29)$$

Kako bismo prikazali linearizirani model u obliku koji je pogodan za MPC algoritam, koristimo matični oblik stanja:

$$\frac{d}{dt} \Delta X = A \Delta X + B \Delta U \quad (30)$$

Gdje:

$$A = \begin{bmatrix} -\frac{kA}{k_{server}} & \frac{kA}{k_{server}} & 0 \\ \frac{kA}{k_{toplo}} & \frac{-kA - k_{fan}C_{zr}U^*}{k_{toplo}} & 0 \\ 0 & 0 & -\frac{1}{T} \end{bmatrix} \quad (31)$$

$$B = \begin{bmatrix} 0 \\ \frac{k_{fan}C_{zr}}{k_{toplo}} (T_{hladno} - T^*_{toplo}) \\ 2k_p U^* \end{bmatrix} \quad (32)$$

## 2. Modelsko prediktivno upravljanje

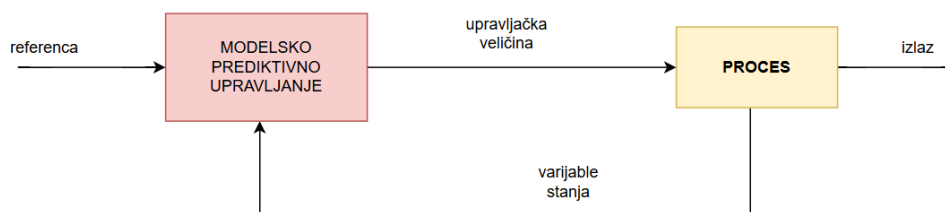
U ovom poglavlju bit će obrađena teorijska osnova modelskog prediktivnog upravljanja, prikazan pregled njegovih glavnih komponenti te analiziran postupak implementacije u sustav hlađenja zatvorenog prostora.

### 2.1. Teorija

Modelsko prediktivno upravljanje (MPC) predstavlja naprednu strategiju upravljanja koja koristi matematički model sustava za predviđanje njegovog budućeg ponašanja i optimizaciju upravljačkih signala u realnom vremenu. MPC je posebno pogodan za složene dinamičke sustave s ograničenjima na ulaze i izlaze, kao što je sustav hlađenja sobe, jer omogućava učinkovito balansiranje između konkurentnih ciljeva, poput održavanja željene temperature i minimiziranja potrošnje energije [4].

#### 2.1.1. Pregled s više razine

Kako se radi o algoritimu koji generira upravljačke signale, integriira se u sustav prije procesa na način kako je prikazano na sljedećoj slici:

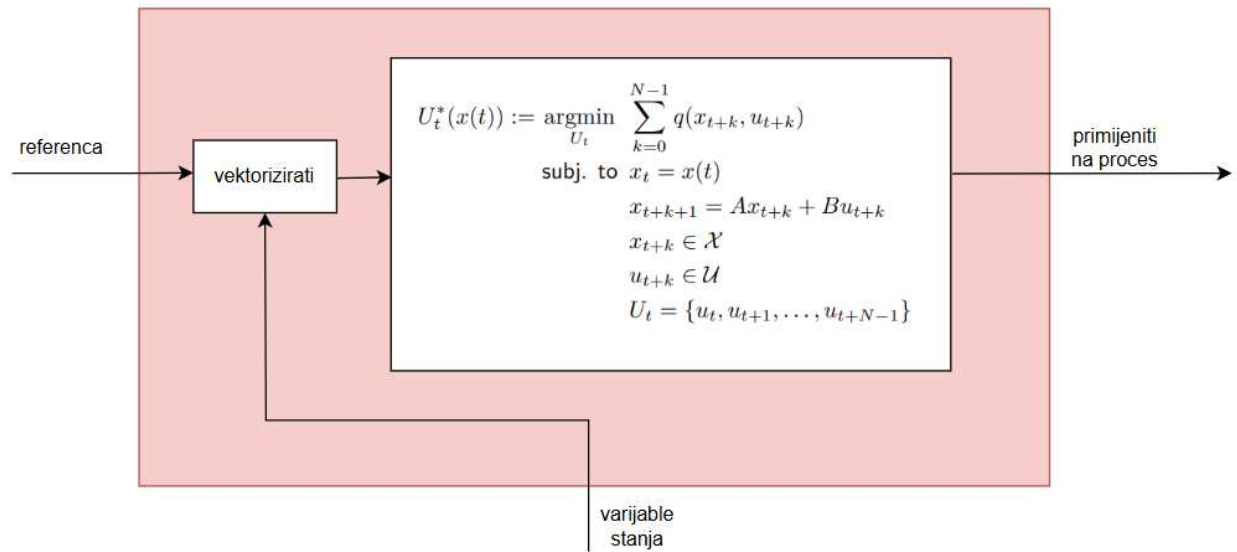


Slika 9 Prikaz integracije MPC-a u sustav

Implementacija bloka MPC sastoji se od više koraka:

- Dobivanje ažurnih varijabli stanja
- Određena referenca
- Ograničenja na sustav i upravljačku veličinu
- Model procesa kojeg se pokušava upravljati
- Definirana kriterijska funkcija

## 2.1.2. Blok MPC-a



Slika 10 MPC algoritam

Na slici (Slika 10), prvi je korak skupljanje trenutnih varijabli stanja te reference. Ako je za referencu moguće dobiti više koraka u budućnosti, moguće je predati optimizacijskom problemu  $N$  koraka reference u budućnosti.

$N$  je takozvani predikcijski horizont. Može ga se usporediti sa vidom vozača koji vozi automobil na cesti, za vozača ovisno o situaciji na cesti može predvidjeti skretanje na cesti u sljedećih 100 metara te pripremiti se za skretanje (usporavanje, ...).

U modelsko prediktivnom upravljanju predikcijski horizont predstavlja za koliko simulacijskih koraka će algoritam predviđati budućnost razvitka modela s obzirom na trenutna stanja koja su poslana u optimizacijski problem.

Optimizacijski problem koji je prikazan na slici (Slika 10) unutar blok dijagrama sastoji se od kriterijske funkcije:

$$J = \sum_{k=0}^{N-1} q(x_{t+k}, u_{t+k}) \quad (33)$$

Kriterijska funkcija  $J$  je obično funkcija koju je pogodno minimizirati s obzirom na upravljački signal, minimizacijom funkcije dobiva se optimalni upravljački signal.

U ovom radu se koristila uglavnom koveksna kriterijska funkcija, kvadratna funkcija. Takva funkcija uvijek ima globalni minimum, na dnu parabole.

Primjer kriterijske funkcije uz težinske matrice:

$$J = \sum_{k=0}^{N-1} (x(k) - r(k))' Q (x(k) - r(k)) + u(k)' R u(k) \quad (34)$$

Gdje:

- $x(k) - r(k)$  predstavlja razliku između varijable stanja kojom želimo upravljati
- $u(k)$  predstavlja amplitudu upravljačke veličine

Kao što se može primijetiti u jednadžbi (34) kriterijska funkcija  $J$  uzima u obzir kako će se ponašati kriterijska u sljedećih  $N$  koraka (predikcijski horizont) tako da zbraja razliku varijabli stanja i reference te amplitudu upravljačke veličine.

$Q$  i  $R$  su u općenitom slučaju dijagonalne matrice koje određuju koliko će optimizator favorirati praćenje reference (dovesti razliku između varijable stanja i reference na nulu) ili minimizirati upravljačku veličinu (štednja energije).

Na slici (Slika 10), osim kriterijske funkcije prikazana su i neka ograničenja (eng. subject to)[5]. Ograničenja se dijele na ograničenja jednakosti i ograničenja nejednakosti. Najvažnije ograničenje jednakosti je model kojim se upravlja:

$$x_{t+k+1} = Ax_{t+k} + Bu_{t+k} \quad (35)$$

Ovo ograničenje osigurava da pri minimiziranju kriterijske funkcije, optimizator ne uzima u obzir varijable stanja koje nisu realne za sustav, na primjer, bicikl koji vozi prema naprijed ne može proklizati lateralno da ispravi svoju putanju.

Sljedeća dva ograničenja odnose se na dopušteni prostor varijabli stanja i prostor upravljačke veličine, ovo su općenito ograničenja nejednakosti u kojima se definiraju intervali za dopuštene iznose varijabli stanja i dopuštene iznose upravljačke veličine.

Primjer:

$$Mx \leq m \quad (36)$$

$$Vu \leq v \quad (37)$$

Primjer sustava s dvije varijable stanja koji ima zahtjeve za dopuštena stanja te primjer sustava s dvije upravljačke veličine:

$$\begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x1 \\ x2 \end{bmatrix} \leq \begin{bmatrix} gornja\_granica\_za\_x1 \\ donja\_granica\_za\_x1 \\ gornja\_granica\_za\_x2 \\ donja\_granica\_za\_x2 \end{bmatrix} \quad (38)$$



$$\begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} u1 \\ u2 \end{bmatrix} \leq \begin{bmatrix} gornja\_granica\_za\_u1 \\ donja\_granica\_za\_u1 \\ gornja\_granica\_za\_u2 \\ donja\_granica\_za\_u2 \end{bmatrix} \quad (39)$$

## 2.2. Implementacija

U radu se prikazuje implementacija MPC-a korištenjem YALMIP-a i Gurobi optimizatora. YALMIP pruža korisnicima mogućnost da jednostavno definiraju varijable, ograničenja i kriterijske funkcije u MATLAB okruženju, dok Gurobi kao vodeći optimizacijski solver nudi vrhunske performanse u pronalaženju optimalnih rješenja.

### 2.2.1. Definiranje optimizacijskog problema u YALMIP-u

Implementacija MPC u ovom radu tekla je postepeno, odnosno, prvo od najjednostavnijeg modela. Prva implementacija bila je na modelu temperature te reguliranje temperature servera. Na sljedećoj stranici objašnjeno je kako se definira problem u matlabu.

Prvo se definiraju sve varijable koje će biti korištene u formulaciji problema:

```
dx = sdpvar(3, N, 'full');
dr = sdpvar(N, 1, 'full');
u = sdpvar(1, N, 'full');
u0 = sdpvar(1, 1, 'full');
p0 = sdpvar(1, 1, 'full');

A_vect= sdpvar(9, 1, 'full');
B_lin_d_var = sdpvar(3, 1, 'full');
```

Gdje:

- ***dx*** je varijabla koja prati kako se sustav razvija u N koraka tj. unutar horizonta, dimenzije 3xN, tri varijable stanja (temperatura servera, sobe i pritiska) te prostor za spremanje za N koraka (horizont).
- ***dr*** je varijabla koja sadrži sljedećih N koraka reference.
- ***u*** je varijabla po kojoj optimizator optimira te sadrži N vrijednosti za svaki korak u horizontu.
- ***u0*** i ***p0*** su samo trenutne vrijednosti upravljačke veličine i pritiska koje se predaju optimizatoru prije optimiranja.
- ***A\_vect*** i ***B\_lin\_d\_var*** su matrice sustava koje algoritam koristi za dobiti ***dx*** unutar horizonta.

Dobivanje kriterijske funkcije i dodavanje ograničenja u optimizacijskom problemu relativno je jednostavno. Kriterijska funkcija je suma po horizontu, posljedično koristimo *for* petlju u matlabu za računanje:

```
for i = 1:N-1
    con = [con, dx(:,i+1) == A_proper*(dx(:,i)) + B_lin_d_var*(u(:,i)-u0)];
    con = [con, M*u(:,i) <= m];
    obj = obj + (dx(2, i)-dr(i, 1))'*Q*(dx(2, i)-dr(i, 1)) + u(:,i)'*R*u(:,i);
end
obj = obj + (dx(2, N)-dr(N, 1))'*Q*(dx(2, N)-dr(N, 1)) + u(:, N)'*R*u(:, N);
```

U ovom primjeru, kriterijska funkcija izgleda jednako kao i u jednadžbi (31), te upravlja se upravljačkom veličinom (brzina okretaja ventilatora) kako bi što preciznije temperatura sobe pratila zadanu referencu.

Optimizacijski problem se definira **prije** same simulacije, dok se tokom simulacije predaju odgovarajuće varijable.

## 2.2.2. Simulacija

Tri se glavna procesa događaju u svakom koraku simulacije:

- Priprema reference
- Linerizacija sustava (računanje jakobijana matrica)
- Optimizacija
- Primjena optimalne upravljačke veličine na sustav

U nastavku je prikazan pojednostavljeni programski kod:

```
for i = 1:Nsteps-N
    % Populate horizon
    reference = populate_reference(i,N)

    % Linearize system and get jacobians
    A_lin, B_lin = getJacobians(sol.u(:,i), sol.x(2,i));

    % Discretize system and pass to optimizer
    A_lin_d, B_lin_d = discretizeSystem();

    % Prepare variables to pass to optimizer
    system_states = [0; 0; 0];
    variables_to_pass = [system_states;
                        reference-sol.x(2, );
                        A_lin_d;
                        B_lin_d;
                        sol.u(:,i);
                        sol.x(3,i)]

    % Optimize
```

```

uopt = optimizer{variables_to_pass};

% Get only the first element of the optimized control signal
sol.u(:,i+1) = uopt;

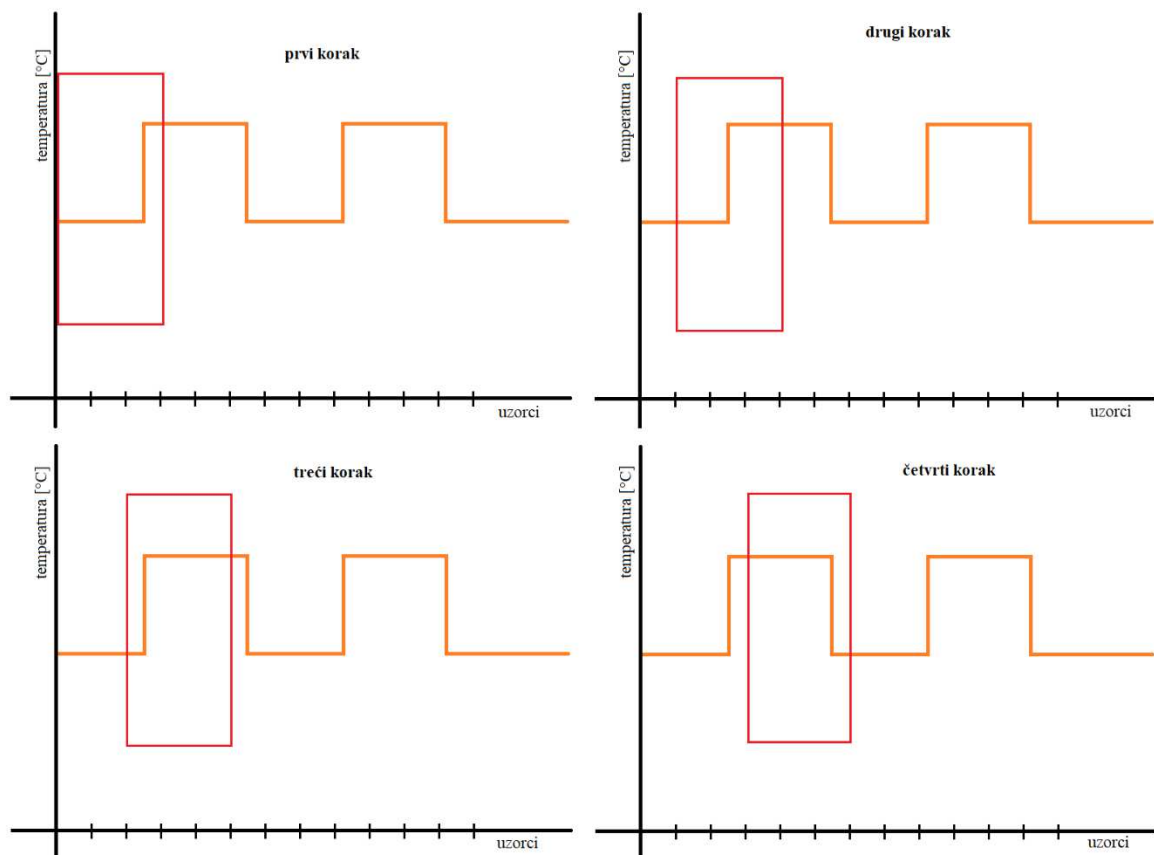
% Update the system matrices
A, B = updateSystemMatrices(sol.u(:,i+1))
A_d, B_d = discretizeSystem();

% Apply optimal control action to system
sol.x(:,i+1) = A_d*sol.x(:,i) + B_d*sol.u(:,i+1) + W(i);
end

```

Prva linija nakon početka petlje, zove funkciju *populate\_reference(i, N)*. Prije simulacije već je određena referenca koju temperatura sobe mora pratiti, a funkcija *populate\_reference(i, N)* samo uzima sljedećih N koraka u referenci počevši od *i*-tog elementa reference

Na sljedećoj slici (slika broj 11) vizualno prikazuje kako se varijabla *reference* ažurira svakim korakom:



Slika 11 Prozor reference kroz simulaciju

Nakon pripreme reference, potrebno je linearizirati sustav. U funkciji *getJacobians(sol.u(:,i), sol.x(2,i))*, računaju se matrice A i B prema jednadžbi broj (31) i (32). Nakon linearizacije, matrice se diskretiziraju te predaju optimizatoru.

U optimizator se predaju varijable sustava u obliku trodimenzionalnog nul vektora. Razlog nul vektora je jednostavan, linearizacija se događa oko trenutnog koraka u simulaciji te pogledamo li jednadžbe (27), (28) i (29) jasno je, optimizator računa računa s razlikama između trenutnog stanja i točke linearizacije, a trenutno stanje je točka linearizacije.

Optimizacijom se dobiva rješenje upravljačke veličine koja minimizira kriterijsku funkciju, upravljačka veličina se u zadnjem koraku primijenjuje na sustav te se dodaje i poremećaj.

## 2.3. Kriterijska funkcija MPC hlađenja podatkovnog centra

Kriterijska funkcija korištena u ovom radu podrazumijeva:

- Upravljanje razlikom temperature između toplog i hladnog područja
- Penaliziranje promjene upravljačke veličine
- Održavanje razlike pritiska između toplog i hladnog područja iznad 2 Pa

Prvi zahtjev je jednostavan za primijeniti, jedina promjena se nalazi u slaganju kriterijske funkcije, nova kriterijska funkcija bi trebala uzeti u obzir trenutnu temperaturu toplog područja i temperaturu hladnog područja, izračunati razliku te usporediti s referencom, odnosno ponovo izračunati razliku.

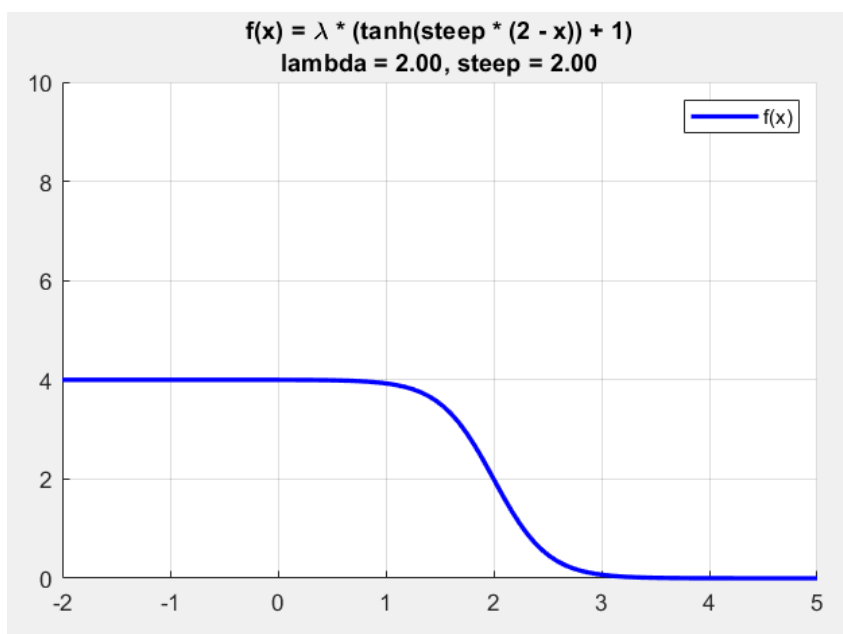
Dok drugi zahtjev podrazumijeva računanje razlike između sljedećeg rješenja MPC (rješenje upravljačke veličine trenutnog koraka) algoritma te upravljačke veličine prethodnog koraka. U sljedećoj jednadžbi prikazana je verzija kriterijske funkcije koja uzima u obzir prvi i drugi zahtjev:

$$J_i = \sum_{k=0}^{N-1} (x_i(k) - T_{hladno} - r(k))' Q (x_i(k) - T_{hladno} - r(k)) + (u_i(k) - u_{i-1})' R (u_i(k) - u_{i-1}) \quad (40)$$

Gdje je  $x_i(k)$  trenutno (index  $i$  trenutni korak) stanje temperature toplog područja,  $T_{hladno}$  je temperatura hladnog područja,  $r(k)$  je  $k$ -ti korak reference,  $u_i$  je upravljačka veličina koju MPC algoritam računa (nepoznanica), a  $u_{i-1}$  je upravljačka veličina prošlog koraka.

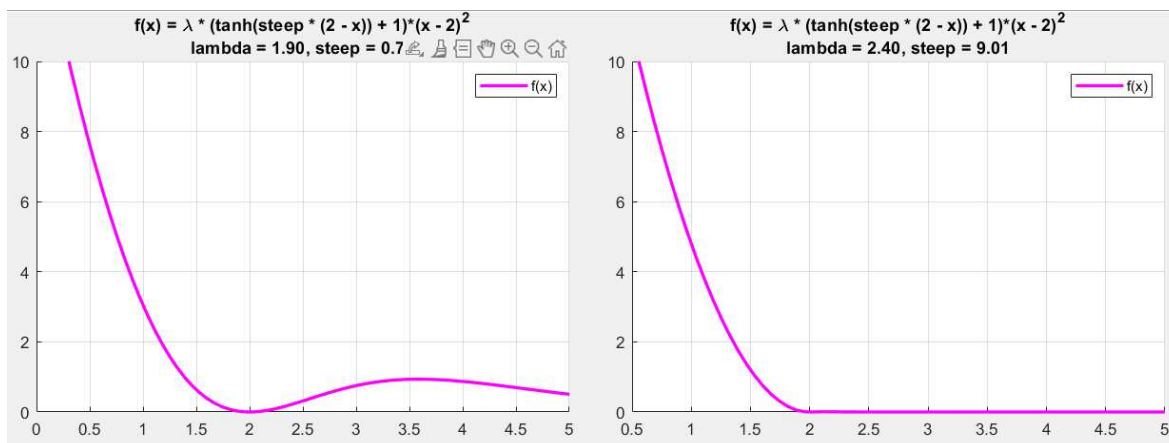
Implementacija trećeg zahtjeva podrazumijeva dodavanje funkcije penalizacije tlaka ispod 2 Pa, koja je jednaka nuli kada je tlak iznad 2 Pa, a raste čim tlak padne ispod te vrijednosti.

Nakon ispitivanja nekoliko iteracija različitih funkcija, najzadovoljavajući rezultati postignuti su kombinacijom hiperbolnog tangensa i kvadratne funkcije. Hiperbolni tangens s određenim parametrima učinkovito aproksimira stepeničastu funkciju (eng. step function), a budući da je kontinuirana funkcija, značajno ubrzava simulaciju. Kvadratna funkcija, s druge strane, omogućuje progresivno povećanje kazne kada tlak padne ispod 2 Pa. Na sljedećoj slici je vizualno prikazana funkcija hiperbolni tangens korištena u radu:



Slika 12 Funkcija tangens hiperbolni

U naslovu slike (Slika 12) napisana je funkcija uz dva parametara,  $\lambda$  i  $\text{steep}$ . Povećanjem  $\lambda$ , dio funkcije koji je s lijeve strane broja dva (2 Pa) se skalira na više vrijednosti. S druge strane, povećanjem parametra  $\text{steep}$  rezultira oštrijim prijelazom s visokih na niske vrijednosti, na taj se način može aproksimirati stepeničastu funkciju. Množenjem pomaknute kvadratne funkcije s funkcijom hiperbolnog tangensa dobiva se samo lijeva strana parabole kao što je prikazano na slici:



Slika 13 Funkcija penalizacije: niska vrijednost konstante *steep* (lijevo), visoka vrijednost konstante *steep* (desno)

Na slici (Slika 13) prikazan je problem koji se javlja kod kombinirane funkcije, gdje zbog postepenog prijelaza dolazi do nepoželjnog blagog povišenja s desne strane broja 2 na x-osi. Tijekom regulacije penalizacije (skaliranje varijablom *lambda*), dostižu se relativno visoke vrijednosti, što uzrokuje dodatno povećanje kombinirane funkcije hiperbolnog tangensa i kvadratne funkcije. Kao rješenje, potrebno je proporcionalno povećati konstantu *steep*, čime se problem rješava unutar određenog (dovoljno širokog) intervala vrijednosti varijable *lambda*.

Ideja odabira ove funkcije je da, pri razlikama u tlaku manjim od 2 Pa, optimizacija pronalazi rješenja koja osiguravaju udaljavanje sustava od vrijednosti od 2 Pa, dok pri razlikama većim od 2 Pa optimizacija ne pridaje veliku važnost ovom dijelu kriterijske funkcije, već približavanju referenci (parametar Q) ili minimiziranju promjena u upravljačkom signalu (paramtar R).

Konačna penalizacijska funkcija:

$$f(x) = \textit{lambda} * \tanh((\textit{steep} * (2 - x) + 1) * (x - 2)^2) \quad (41)$$

Konačna kriterijska funkcija:

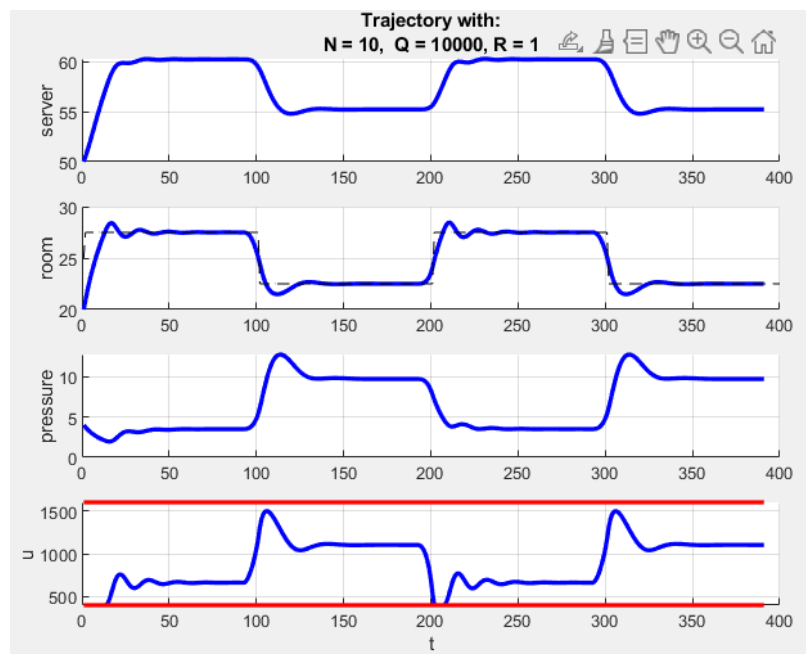
$$J_i = \sum_{k=0}^{N-1} (x_i(k) - T_{hladno} - r(k))' Q (x_i(k) - T_{hladno} - r(k)) + (u_i(k) - u_{i-1})' R (u_i(k) - u_{i-1}) + \textit{lambda} * \tanh((\textit{steep} * (2 - x_i(k)) + 1) * (x_i(k) - 2)^2) \quad (42)$$

### 3. Rezultati implementacije na sobu podatkovnog centra

Nakon osnovnog objašnjenja implementacije MPC algoritma u radu, slijedi primjena istog algoritma regulacije na prostoriju podatkovnog centra.

#### 3.1. Odziv

Uz jednostavan konstantni poremećaj od 50kW kao snaga koju serveri dovode u sustav dobiva se odziv kao na sljedećoj slici (Slika 14):



Slika 14 Odziv testne simulacije

Na slici (Slika 14) jasno se vidi kako temperatura sobe pokušava pratiti referencu, u naslovu odziva mogu se pročitati i vrijednosti za predikcijski horizont, i težinske matrice Q i R.

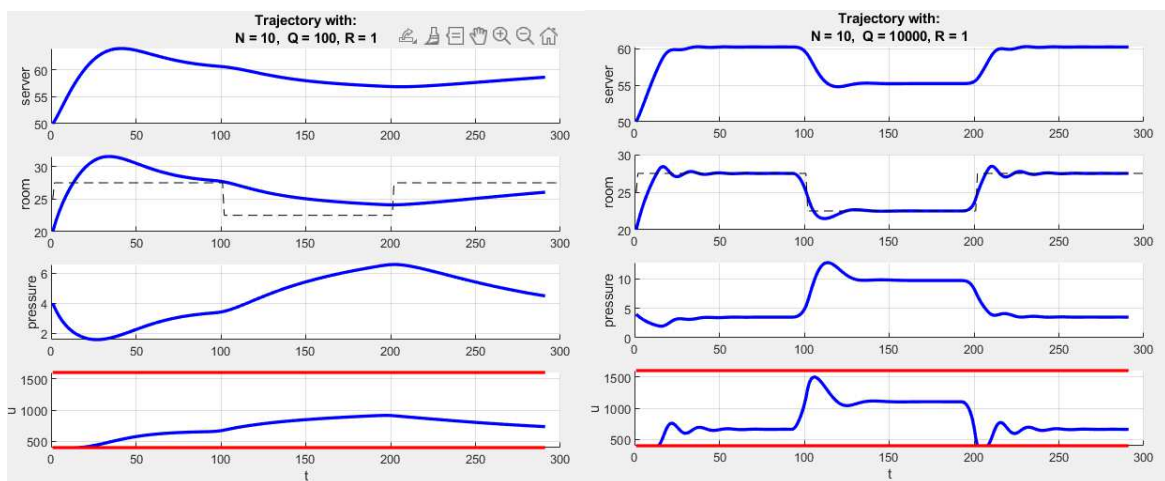
Predikcijski horizont je pomalo teško za zamisliti koliko daleko u budućnosti računa, no pogleda li se zadnjih nekoliko koraka simulacije vidljivo je da graf nije iscrtan do kraja, što nam dočarava koliki je otprilike prozor predikcije, u ovom je slučaju 10 koraka, svaki korak je jednak jednoj sekundi.

Težinske matrice postavljene su u omjeru  $Q = 10000:1$ , čime se značajno veći naglasak stavlja na praćenje referentne vrijednosti u usporedbi sa smanjenjem promjene upravljačke veličine. Ovaj omjer uvelike ovisi o mjerama kojima se penalizira određena veličina; primjerice, temperature se kreću do  $100^{\circ}\text{C}$ , dok upravljačka veličina doseže do 1600 okretaja u minuti. Oscilacije, brzina odziva i nadvišenje mogu se, naravno, smanjiti eksperimentiranjem s različitim parametrima za matrice  $Q$  i  $R$ . Eksperimentiranje s različitim parametrima uglavnom je vođeno načelom da povećanje vrijednosti matrice  $Q$  ubrzava sustav i smanjuje pogrešku između trenutnog stanja sustava i referentne vrijednosti, pri čemu je, s negativne strane, moguće uvesti oscilacije i nadvišenja u simulaciju. S druge strane, povećanjem vrijednosti matrice  $R$  dolazi do smanjenja oscilacija, sporijeg odziva i manjih nadvišenja. Potrebno je postići ravnotežu između ovih dvaju parametara kako bi se osigurao optimalan kompromis između brzine odziva sustava i stabilnosti.

## 3.2. Utjecaj parametara na odziv simulacije

U ovom potpoglavlju biti će prikazane i analizirane simulacije uz različite parametre i ograničenja.

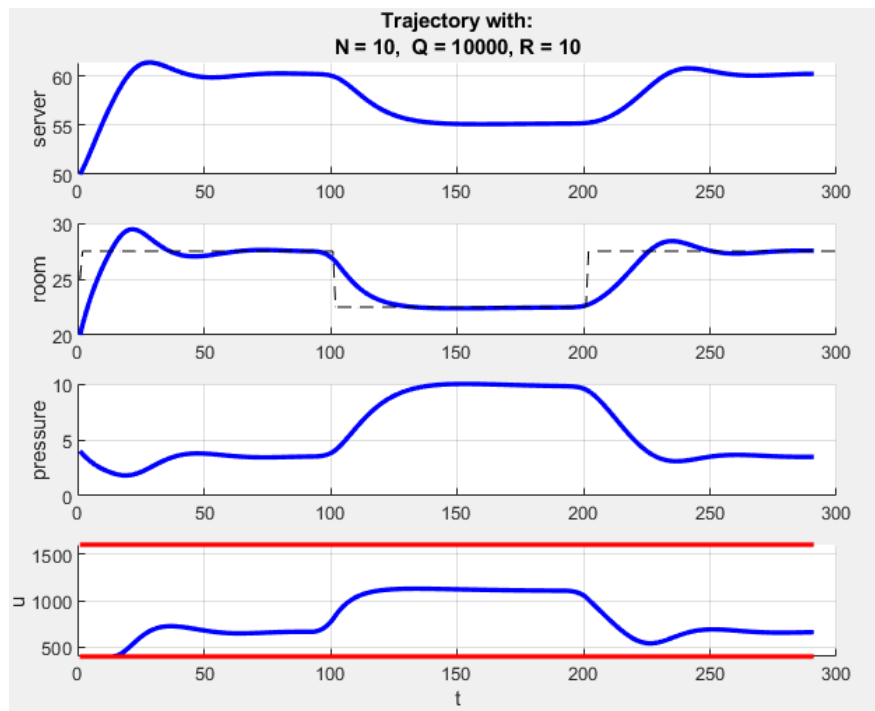
Pogledajmo što se dešava povećanjem omjera između  $Q$  i  $R$ :



Slika 15 Utjecaj  $Q$  parametra

Na slici (Slika 15), na lijevoj strani, gdje je  $Q:R$  omjer  $100:1$ , jasno je vidljivo kako sustav vrlo sporo i tromo reagira na promjenu reference. Povećanjem omjera vidljiva je pojava oscilacija u temperaturi sobe te u upravljačkoj veličini. Način na koji se mogu ublažiti oscilacije jest povećanjem  $R$  parametra:



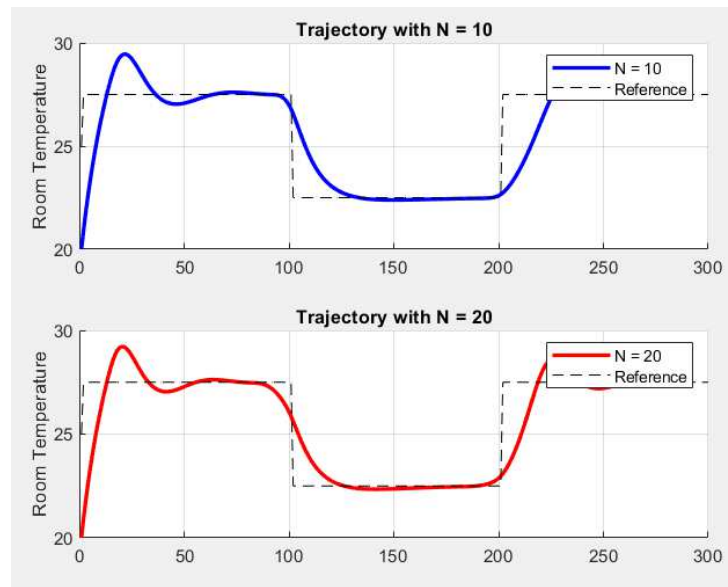


Slika 16 Odziv uz povećanje R parametra

Kao što je vidljivo na slici (Slika 16), povećanjem parametra R smanjile su se oscilacije, ali usporio se odziv na promjenu reference. Kao što je spomenuto u prethodnom potpoglavlju.

Utjecaj povećanja ili smanjivanja predikcijskog horizonta je sljedeći: iako je uvijek poželjno povećati predikcijski horizont radi poboljšanja performansi sustava, to je uvijek ograničeno zbog značajnog povećanja trajanja vremena potrebno da MPC izračuna jedan korak. Znači, povećavanjem predikcijskog horizonta, sustav prije reagira na promjene reference (ne ubrzava) ili približavanje ograničenja te bolje ustabiljuje sustav, ali uvelike povećava korištenje računalnih resursa.

U sljedećoj slici prikazana je usporedba:



Slika 17 Prikaz odziva uz različiti predikcijski horizont

S predikcijskim horizontom jednakim  $N = 10$ , simulacija se izvršila u otprilike 32 sekunde, dok za  $N = 20$ , potrebno je oko 46 sekundi.

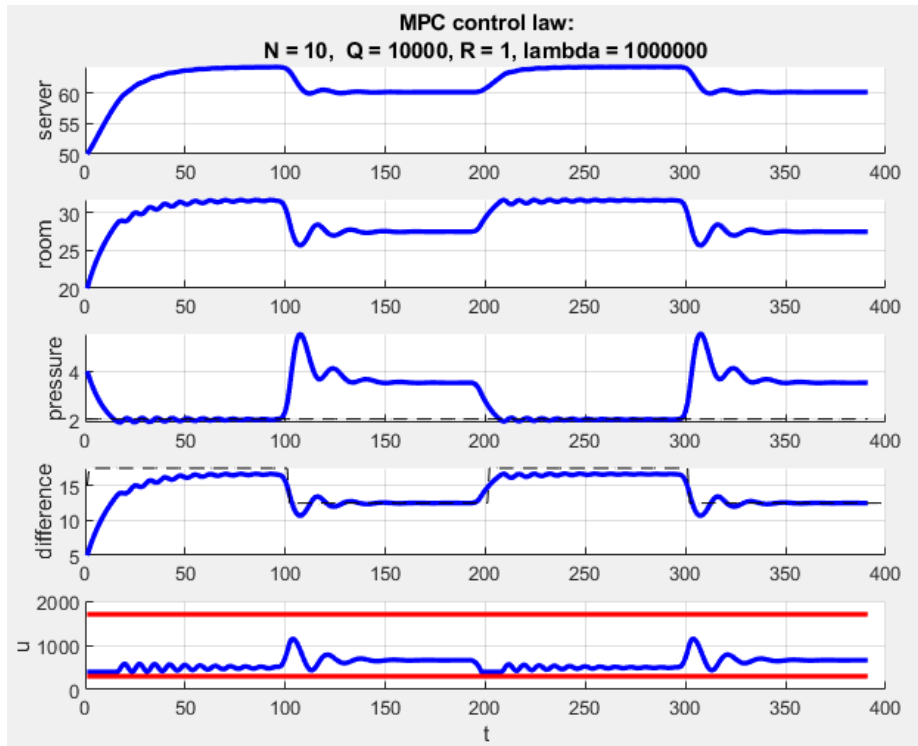
Uz parametre  $Q$ ,  $R$  i  $N$ , postoje i ograničenja nejednakosti. U ovom slučaju, jedino primijenjeno ograničenje nejednakosti odnosi se na upravljačku veličinu. Ovo ograničenje proizlazi iz realnih uvjeta, gdje je očekivano da ventilator konstantno održava određenu brzinu vrtnje. Razlog tome može biti potreba za održavanjem minimalne razlike tlaka, o čemu će biti više riječi u sljedećem poglavlju.

### 3.3. Odziv s modificiranom kriterijskom funkcijom

Nakon konstruiranja penalizacijske funkcije za održavanje razlike pritiska iznad 2 Pa, nova funkcija se jednostavno sumira s postojećom kriterijskom funkcijom. Sumiranjem se nadodaje već postojećim parametrima za MPC algoritam  $Q$ ,  $R$  i  $N$ , dodatni parametar  $\Lambda$ .

Testiranje kriterijske funkcije je postignuto na sljedeći način, zadati alternirajući signal reference koji predstavlja razliku između toplog i hladnog područja podatkovnog centra, prosječna vrijednost signala bi trebala biti dovoljno visoka da sustav iziskiva dovoljno niske vrijednosti upravljačke veličine (brzina okretaja ventilatora) da bi u stacionarnom stanju

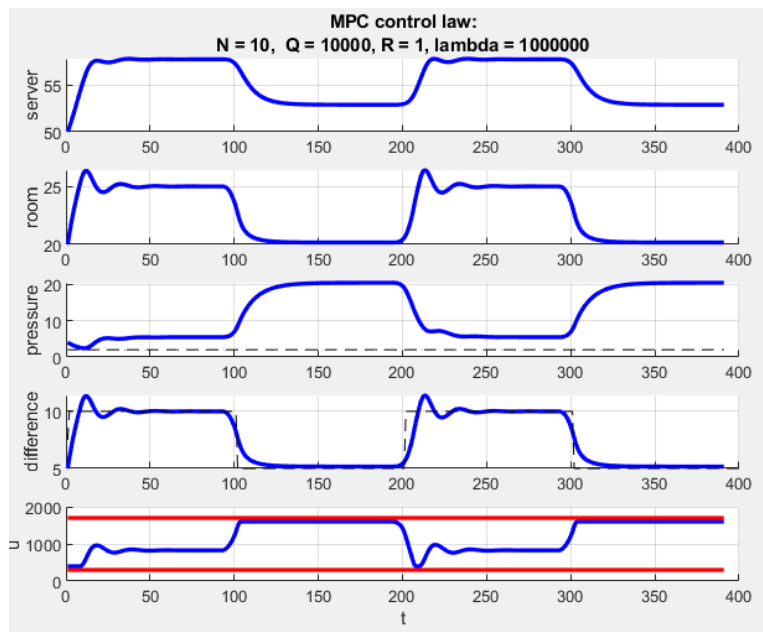
razlike pritiska, vrijednost razlike pritiska bila ispod 2 Pa. Na taj način, očekuje se da će algoritam MPC odabirati vrijednosti upravljačke veličine koje održavaju razliku pritiska iznad 2 Pa kada je razlika između toplog i hladnog područja ispod zadane reference.



Slika 18 Odziv sustava

Na slici (Slika 18) se upravo to i dešava, referentni signal alternira između  $17.5^{\circ}\text{C}$  i  $12.5^{\circ}\text{C}$  te kada je razlika toplog područja i hladnog područja ispod referentnog signala, kao što je slučaj kada je referentni signal  $17.5^{\circ}\text{C}$  upravljačka veličina održava razliku pritiska iznad 2 Pa. Kada referentni signal iznosi  $12.5^{\circ}\text{C}$  razlika toplog i hladnog područja prati referentni signal.

Sljedeća slika prikazuje odziv sustava uz referentni signal koji alternira između  $10^{\circ}\text{C}$  i  $5^{\circ}\text{C}$  (malo manja razlika temperature toplog i hladnog područja) uz iste parametre  $Q, R, N$  i  $\lambda$ :



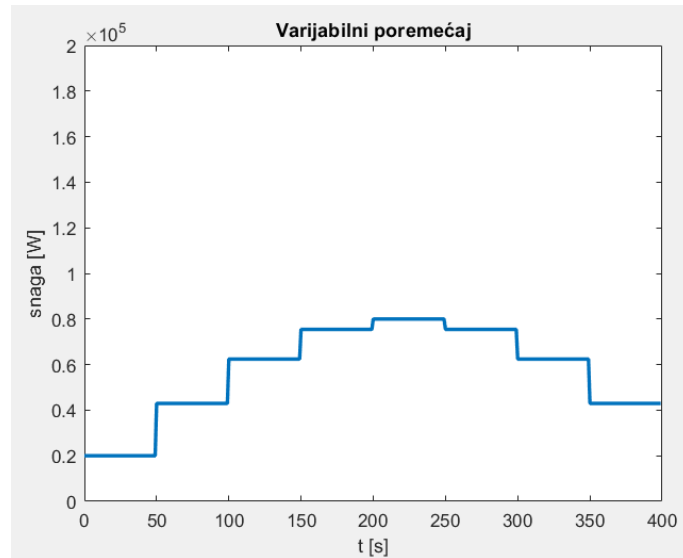
Slika 19 Odziv sustava na alternirajuću referencu razlike temperature

U slučaju na slici (Slika 19), postizanje zadane reference ne izaziva problem s pritiskom jer je razliku toplog i hladnog područja puno manja pa temperatura toplog područja ne mora postizati iznose za koje MPC algoritam mora odabirati upravljačke veličine koje izazivaju pad razlike tlaka na ispod 2 Pa.

Za postizanje željenog odziva, potrebno je snažnije penalizirati tlak ispod 2 Pa nego odstupanje od zadane reference. Iako bi se moglo pretpostaviti da će MPC algoritam u takvom slučaju zanemariti praćenje reference zbog znatno veće vrijednosti parametra lambda u usporedbi s parametrom Q, važno je napomenuti da funkcija penalizacije razlike tlaka ostaje približno jednaka nuli kada je razlika tlaka iznad 2 Pa. Stoga, kada je razlika tlaka dostatno velika, praćenje reference ponovno postaje prioritet MPC-a.

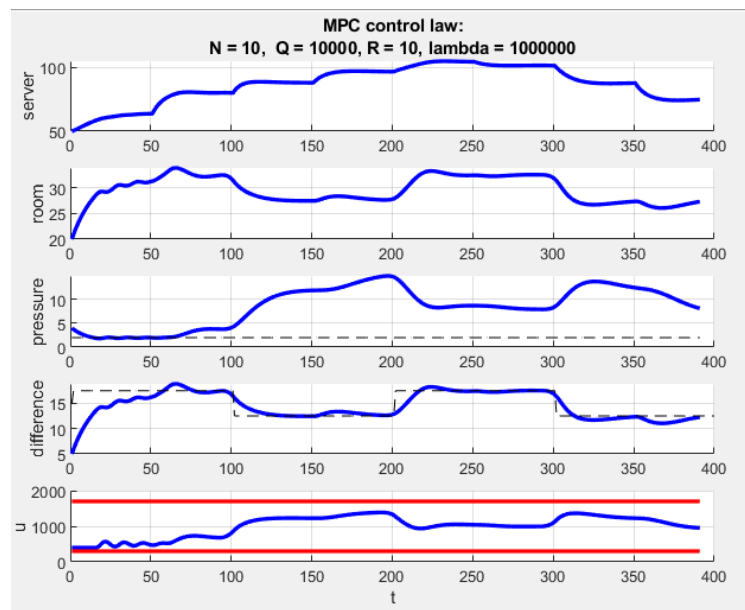
### 3.4. Odziv uz varijabilni poremećaj

Odzivi su do sad pretpostavljali konstantnu vrijednost poremećaja uzrokovanog energijom koju serveri unose u sustav (50 kW). Međutim, u stvarnom sustavu taj poremećaj varira ovisno o opterećenju servera, zbog čega je potrebno ispitati kako se MPC algoritam prilagođava promjenjivom poremećaju. U sljedećem odzivu sustava bit će simuliran varijabilni poremećaj koji se postupno povećava i smanjuje.



Slika 20 Prikaz poremećaja primijenjenog na sustav

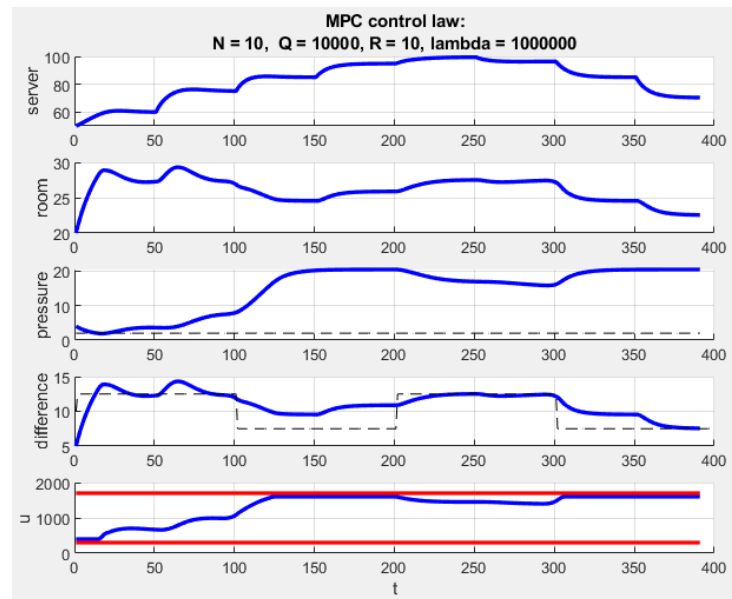
Odziv sustava na poremećaj sa prethodne slike (Slika 20) :



Slika 21 Odziv na varijabilni poremećaj

Na prikazanoj slici (Slika 21) može se vidjeti da MPC algoritam na početku nastoji održati razliku tlaka iznad 2 Pa. U pedesetom koraku dolazi do naglog povećanja poremećaja na približno 45 kW, što uzrokuje naglo povećanje razlike između toplog i hladnog područja te MPC daje prioritet praćenju zadane reference (10°C).

Spuštanjem prosječne vrijednosti razlike temperature toplog i hladnog područja, postaje izazovnije održavati referencu:



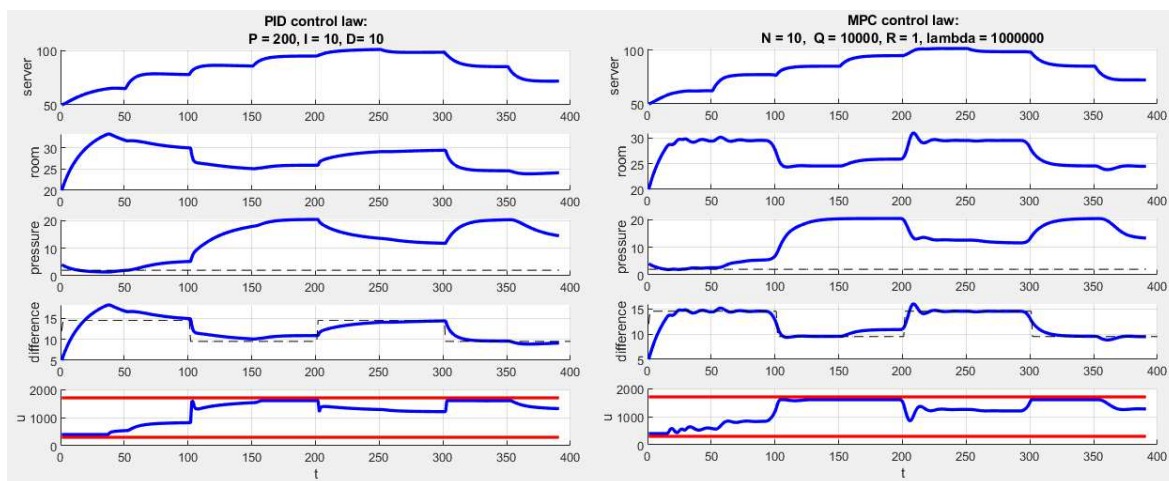
Slika 22 Odziv sustava na praćenje niže reference

Na slici (Slika 22) u intervalima od 100. do 200. koraka te od 300. do 400. koraka, MPC algoritam ispravno primjenjuje najveći mogući iznos upravljačke veličine kako bi smanjio temperaturu u toplom području. Istovremeno je vidljivo da se na početku simulacije tlak ne spušta ispod 2 Pa.

### 3.5. Usporedba s PID upravljanjem

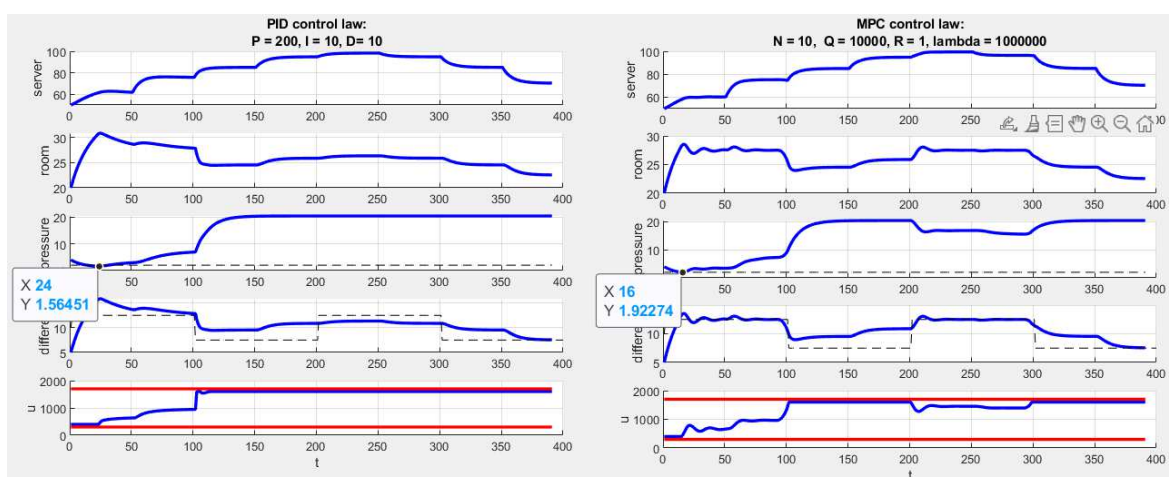
U ovom poglavlju uspoređuju se MPC i PID upravljanje, s ciljem analiziranja njihovih prednosti i nedostataka u kontekstu regulacije sustava hlađenja sobe podatkovnog centra. Usporedba će obuhvatiti ključne karakteristike oba pristupa, njihove performanse u različitim uvjetima rada te prikazati situacije u kojima svaki od njih ima optimalnu primjenu.

Ručno podešen PID regulator [3] uspoređen je sa MPC upravljanjem na sljedećoj slici:



Slika 23 Usporedba PID i MPC upravljanja

Na slici (Slika 23) prikazana je usporedba PID i MPC regulatora, vidljivo je da je praćenje reference uz MPC upravljanje puno bolje, također, na početku simulacije vidljivo je kako razlika tlaka pada ispod 2 Pa za PID regulator. Na sljedećoj slici takav slučaj je bolje prikazan:



Slika 24 Usporedba PID i MPC regulatora na nižu referencu

Na slici (Slika 24) odabrana je najniža točka razlike tlaka kod oba odziva, osim što kod se PID regulatora razlika tlaka ne održava iznad 2 Pa, praćenje reference kod nižih razlika temperatura toplog i hladnog područja, puno je lošije nego kod MPC regulatora.

# Zaključak

U ovom diplomskom radu prikazano je modeliranje i implementacija modelskog prediktivnog upravljanja (MPC) na primjeru sustava hlađenja zatvorenog prostora, konkretno sobe podatkovnog centra. Kroz izradu rada, prikazan je proces matematičkog modeliranja temperature i pritiska te njihovo korištenje u okviru MPC algoritma za optimizaciju upravljanja sustavom.

Rezultati provedenih simulacija i implementacije pokazuju da modelsko prediktivno upravljanje može pružiti značajne prednosti u odnosu na klasične metode poput PID regulacije. MPC omogućava učinkovito upravljanje temperaturom unutar podatkovnih centara, uzimajući u obzir fizička ograničenja sustava i promjenjive uvjete.

Iako je MPC metoda pokazala zadovoljavajuću učinkovitost u simulacijama, njezina primjena u stvarnim uvjetima donosi određene izazove, kao što su zahtjevi za računalnim resursima i potreba za preciznim modeliranjem sustava. Nadalje, ključan je odabir odgovarajućih parametara MPC algoritma, poput predikcijskog horizonta, težinskih matrica kriterijske funkcije koji značajno utječu na odziv i stabilnost sustava.

Općenito, ovaj rad potvrđuje da je modelsko prediktivno upravljanje prikladna i napredna metoda za regulaciju složenih dinamičkih sustava kao što je hlađenje podatkovnog centra. Međutim, daljnja istraživanja su potrebna kako bi se dodatno optimizirali modeli i algoritmi, te prilagodili specifičnim zahtjevima i uvjetima rada u stvarnim aplikacijama.



## Literatura

- [1] G.F.C. Rogers, Yon Mayhew, *Engineering Thermodynamics, work & heat transfer*, 4. izdanje, 1992.
- [2] William B. Baratuci, *Learn Thermodynamics Workbook*, 4. izdanje, 2018.
- [3] Katsuhiko Ogata, *Modern Control Engineering*, 5. izdanje, 2009.
- [4] Wikipedia, *Model predictive control*, Poveznica: [Model predictive control - Wikipedia](#); pristupljeno 30. kolovoza 2024.
- [5] Steve Brunton. *Model Predictive Control*, Youtube, 11. lipnja, 2018., [Model Predictive Control](#)

## Sažetak

Ovaj diplomski rad istražuje primjenu modelskog prediktivnog upravljanja (MPC) u sustavima hlađenja zatvorenih prostora, s posebnim naglaskom na sobe podatkovnih centara. Rad obuhvaća proces matematičkog modeliranja temperature i pritiska te razvoj algoritma MPC za optimizaciju rada sustava u realnom vremenu. Provedene simulacije pokazuju da MPC upravljanje nudi značajna poboljšanja u odnosu na tradicionalne metode poput PID regulacije, posebno u pogledu učinkovitosti i preciznosti održavanja temperature. Unatoč prednostima, primjena MPC-a zahtijeva optimizaciju parametara i računalnih resursa. Zaključeno je da je MPC metoda prikladna za složene sustave hlađenja, no potrebna su daljnja istraživanja kako bi se poboljšala njezina primjena u stvarnim uvjetima.

## Summary

This thesis explores the application of Model Predictive Control (MPC) in cooling systems for enclosed spaces, with a particular focus on data center rooms. The work covers the mathematical modeling of temperature and pressure and the development of an MPC algorithm for real-time system optimization. The simulations demonstrate that MPC offers significant improvements over traditional methods such as PID control, especially in terms of efficiency and precision in maintaining. Despite its advantages, MPC application requires parameter optimization and computational resources. The conclusion is that MPC is a suitable method for complex cooling systems, but further research is needed to improve its implementation in real-world conditions.