

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Diplomski sveučilišni studij elektrotehnike

Programski zadatak iz kolegija *Mehatronički sustavi*

SIMULACIJA TEMPERATURNOG REGULACIJSKOG PROCESA

Rijeka, prosinac 2024.

Leonard Mikša
0069086808

Sadržaj

1. Uvod	2
2. Regulacija temperature zraka u prostoriji	3
2.1. Model sustava	3
2.1.1. Grijач	3
2.1.2. Model procesa	4
2.1.3. Senzor temperature	6
2.1.4. Skaliranje referentne temperature	7
2.2. Programski kod i simulacija u Pythonu	8
2.2.1. Početni parametri	8
2.2.2. Funkcije	9
2.2.3. Glavna funkcija i kreiranje animacije	11
2.2.4. Rezultati	14
3. Zaključak	17
Bibliografija	18

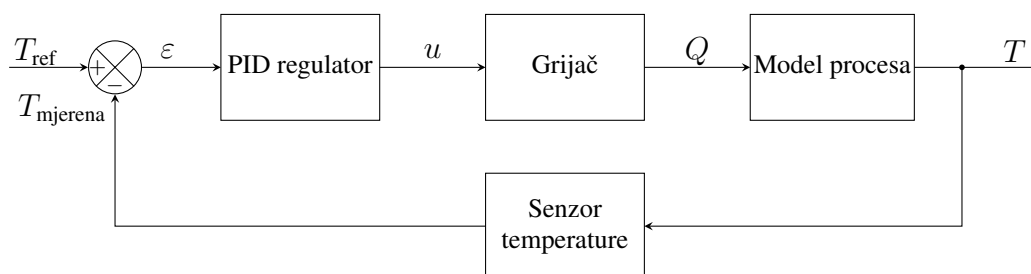
1. Uvod

Zadatak ovog programskog zadatka je simulacija temperaturnog regulacijskog procesa. Ova tema otvara razne mogućnosti što se tiče same primjene - od pasterizacije i fermentacije u prehrambenoj industriji, sterilizacije i kemijskih reakcija u farmaceutskoj industriji, HVAC sustava i toplinskih procesa u motorima i baterijama u energetskom sektoru i automobilske industriji, pa sve do kućnih primjena poput grijanja prostorije. Upravo to će biti i tema ovog rada.

2. Regulacija temperature zraka u prostoriji

2.1. Model sustava

Kako bismo uspješno implementirali algoritam upravljanja, najprije ga je potrebno matematički modelirati. Kao i svaki regulacijski sustav, i ovaj će se sastojati od regulatora, aktuatora - u ovom slučaju grijača, te objekta upravljanja - u ovom slučaju prostorije. S obzirom da se radi o zatvorenom regulacijskom krugu, u povratnoj vezi se nalazi senzor temperature.



Slika 2.1. Shema zatvorenog temperaturnog regulacijskog sustava

Slika 2.1 prikazuje načelnu shemu regulacijskog sustava. Signal razlike ε između željene temperature i stvarne (mjerene) temperature predstavlja ulazni signal u PID regulator, a u predstavlja izlaz iz regulatora, odnosno upravljački signal za grijač. Izlaz iz grijača je Q , odnosno toplina dovedena u sustav.

Stvarna toplina u prostoriji mjeri se senzorom u povratnoj vezi te uspoređuje s referentnom vrijednošću. Ta usporedba najčešće je realizirana diferencijalnim pojačalom.

2.1.1. Grijač

Korišteni grijač je **FIREROD®Cartridge Heater** marke **WATLOW**, prikazan na slici 2.2. Njegove specifikacije dostupne su u *datasheetu* na [1]. Snaga grijača većih promjera ove serije doseže i 2760 W, dok se snaga grijača manjih dimenzija (četvrtine do polovine inča) kreće između 500 W i 1000 W. Maksimalni napon iznosi 480 V, a maksimalna gustoća snage do $62 \frac{\text{W}}{\text{cm}^2}$.



Slika 2.2. WATLOW FIREROD Cartridge Heater

Grijač ćemo modelirati kao sustav nultog reda, odnosno kao jednostavan sustav s pojačanjem

$$Q(t) = k \cdot u(t), \quad (2.1)$$

gdje je $Q(t)$ izlazna toplina u vatima, $u(t)$ normirana vrijednost (od 0 do 1) upravljačkog signala PID regulatora, a k proporcionalno pojačanje grijača. Za k ćemo uzeti vrijednost koja odgovara maksimalnom kapacitetu grijača - tada će za maksimalni upravljački signal grijač davati svoj maksimum u grijanju prostorije.

Odabrat ćemo grijač promjera $\frac{3}{8}$ inča (približno 1 cm) čija je maksimalna snaga 800 W [1]. Sada je

$$Q(t) = 800u(t) \quad (2.2)$$

2.1.2. Model procesa

S obzirom na relativno nisku snagu i dimenzije grijača, toplinsku dinamiku prostorije modelirat ćemo pod pretpostavkom malih dimenzija prostorije, sa površinom 10m^2 i visinom stropa 2.5 metra. Također, grijač grije prostoriju prijenosom topline **konvekcijom**.

Naravno, u stvarnosti se ne prilagođava prostorija grijaču, nego grijač prostoriji. Međutim, zbog jednostavnosti procesa izrade simulacijskog modela odabran je ovakav pristup. Parametri potrebni za model su volumen prostorije, gustoća zraka, te specifični toplinski kapacitet zraka, iz kojih možemo odrediti **toplinski kapacitet prostorije**.

S obzirom na odabrane dimenzije prostorije, volumen zraka je

$$V_{\text{zrak}} = 10\text{m}^2 \cdot 2.5\text{m} = 25\text{m}^3, \quad (2.3)$$

a gustoća zraka iznosi

$$\rho_{\text{zrak}} = 1.225 \frac{\text{kg}}{\text{m}^3} \quad (2.4)$$

pri sobnoj temperaturi i atmosferskom tlaku; gustoća zraka je inače ovisna o tlaku i temperaturi zraka, ali ovdje ćemo ta odstupanja zanemariti.

Specifični toplinski kapacitet zraka je konstanta koja iznosi

$$c_{\text{zrak}} = 1005 \frac{\text{J}}{\text{kgK}} \quad (2.5)$$

Konačno, toplinski kapacitet prostorije jednak je umnošku mase i specifičnog toplinskog kapaciteta

$$C = m_{\text{zrak}} \cdot c_{\text{zrak}} = \rho_{\text{zrak}} \cdot V_{\text{zrak}} \cdot c_{\text{zrak}} \quad (2.6)$$

$$C = 1.225 \frac{\text{kg}}{\text{m}^3} \cdot 25\text{m}^3 \cdot 1005 \frac{\text{J}}{\text{kgK}} \quad (2.7)$$

$$C \approx 30800 \frac{\text{J}}{\text{K}} \quad (2.8)$$

Idući važan parametar je **toplinski otpor** koji opisuje intenzitet izmjene topline s okolinom. Prema [2], umjereni toplinski otpor se kreće između 0.9 i 1.3 $\frac{\text{m}^2\text{K}}{\text{W}}$. Zbog jednostavnosti proračuna i modela, uzimamo

$$R = 1 \frac{^\circ\text{C}}{\text{W}} \quad (2.9)$$

Posljednji bitan parametar za model procesa je **toplinska vremenska konstanta**, definirana kao umnožak toplinskog kapaciteta i toplinskog otpora. S obzirom da su Celzijevi stupnjevi i Kelvini ekvivalentne mjerne jedinice, one se mogu pokratiti u umnošku te ostaviti samo sekunde kao omjer džula i vata.

$$\tau = R \cdot C = 1 \frac{^\circ\text{C}}{\text{W}} \cdot 30800 \frac{\text{J}}{\text{K}} \quad (2.10)$$

$$\tau = 30800 \text{ s} \quad (2.11)$$

Toplinska vremenska konstanta fizikalno predstavlja vrijeme potrebno da se temperatura u prostoriji promjeni isključivo uslijed utjecaja temperature okoline, odnosno razlike u temperaturi prostorije i okoline (drugim riječima, kada je regulacijski sustav isključen). U ovom slučaju, ona iznosi približno 8.5 sati. U stvarnosti, ovako velika toplinska vremenska konstanta nije realna, ali ovdje u obzir nisu uzeti različiti faktori koji doprinose gubicima topline poput debljine zidova, nedovoljne zabrtvljenosti otvora; vrata i prozora, čak i do namještaja u samoj prostoriji. Međutim, za potrebe ove simulacije ovi faktori su zanemareni.

Sada konačno možemo, **na temelju prvog zakona termodinamike**, opisati toplinsku dinamiku u prostoriji - promjena energije sustava jednaka je razlici dodane i disipirane topline. Stoga slijedi

$$\frac{dT}{dt} = \frac{Q}{C} - \frac{T - T_{\text{okoline}}}{\tau} \quad (2.12)$$

Temperatura okoline djeluje na sustav kao poremećaj i nju ćemo u simulaciji odabrati kao konstantu od 20°C.

2.1.3. Senzor temperature

Korišteni senzor je **LM35** marke *Texas Instruments*, prikazan na slici 2.3. Njegov temperaturni raspon kreće se od -55°C do 150°C , što je svakako u granicama primjene za mjerenje temperature u prostoriji. **Linearni faktor skaliranja mu iznosi $10\text{mV}/^{\circ}\text{C}$** , a preciznost na sobnoj temperaturi mu je 0.5°C . Nelinearnost se kreće u rasponu $\pm 0.25^{\circ}\text{C}$, što je za potrebe ove simulacije zanemarljivo.



Slika 2.3. Texas Instruments LM35 Analog Temperature Sensor [3]

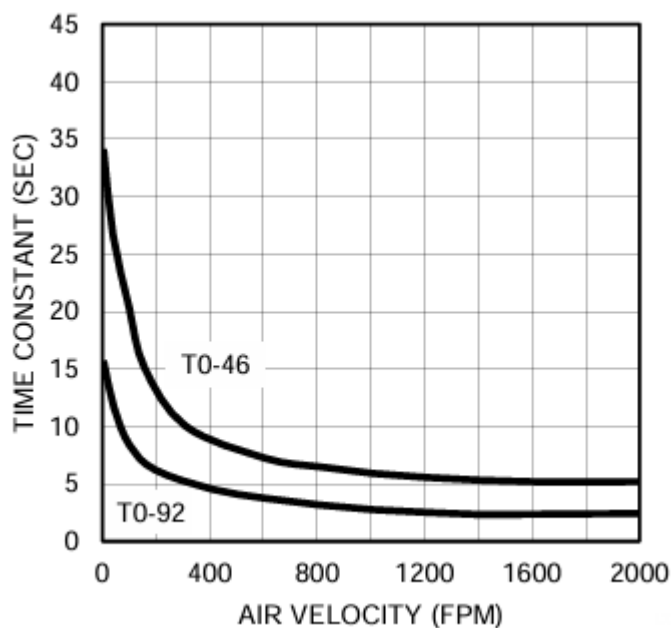
Slika 2.4 prikazuje ovisnost vremenske konstante senzora u ovisnosti o okruženju u kojem se nalazi. Za ovu primjenu promatramo vrijednost vremenske konstante na brzini zraka od 0 s obzirom da smo u zatvorenom prostoru. T0-46 i T0-92 su oznake koje se odnose na različita pakiranja senzora, međutim obje vrijednosti su puno manje od vremenskih konstanti samog procesa. Toplin-ski procesi su izrazito tromi, odnosno imaju velike vremenske konstante, tako da su ove vrijednosti zanemarive.

Zbog toga ćemo i senzor - kao što smo i grijač - umjesto kao sustav prvog reda, modelirati kao jednostavan sustav s pojačanjem, a to pojačanje će zapravo biti linearni faktor skaliranja.

$$T_{\text{mjerena}}(t) = k_T \cdot T(t), \quad (2.13)$$

odnosno izraženo u voltima

$$T_{\text{mjerena}} = 0.01T \quad (2.14)$$



Slika 2.4. Toplinska vremenska konstanta LM35 temperaturnog senzora [3]

2.1.4. Skaliranje referentne temperature

S obzirom da regulacijski sustav obrađuje električne signale, preostaje nam još referentnu vrijednost temperature preskalirati na naponsku vrijednost koja će se uspoređivati sa izlaznim signalom senzora. S obzirom da mjerena temperatura od 0°C odgovara izlaznom naponu senzora od 0 V, a faktor skaliranja je 10mV/°C, na isti način ćemo preskalirati i referentnu temperaturu.

Na primjer, ako je referentna temperatura 20°C, odgovarajući naponski ekvivalent iznosi 200mV. Nadalje, ako je temperatura 35°C, odgovarajući naponski ekvivalent iznosi 350mV, itd.

$$V_{\text{ref}} = 0.01T_{\text{ref}} \quad (2.15)$$


```

15 k = 800          #Maksimalna snaga grijača; pojačanje podsustava grijača (W)
16 k_T = 0.01       #Linearni faktor skaliranja senzora LM35 (V/°C)
17
18 #Parametri PID regulatora:
19 K_p = 10.0        #Proporcionalno pojačanje
20 K_i = 0.1         #Integracijsko pojačanje
21 integral = 0.0
22 K_d = 0.5         #Derivacijsko pojačanje
23 prev_error = 0.0
24
25 #Parametri simulacije:
26 dt = 10.0         #Vremenski korak simulacije (s)
27 simulation_time = 3600 #Ukupno vrijeme simulacije (s) = 1 sat

```

2.2.2. Funkcije

Funkcijama su implementirani podsustavi regulacijskog sustava - PID regulator, grijač, model procesa i temperaturni senzor. Također, ovdje su ostvarene i funkcije za definiranje vanjske temperature i referentne temperature. One su definirane kao skokovite promjene u vremenu kako bi se intuitivno prikazalo da regulacijski sustav radi ispravno.

```

1 def outside_temperature(time_elapsed):
2     """
3     Proizvoljno zadavanje temperature okoline.
4     Ulaz: vrijeme
5     Izlaz: vanjska temperatura koja se promatra kao poremećaj na sustav
6     """
7     if time_elapsed < 1000:
8         return 10.0
9     elif 1000 <= time_elapsed < 2000:
10        return 5.0
11    else:
12        return 0.0
13
14
15 def reference_temperature(time_elapsed):
16     """
17     Proizvoljno zadavanje željene temperature.
18     Ulaz: vrijeme
19     Izlaz: referentna temperatura
20     """
21    if time_elapsed < 2500:
22        return 25.0
23    else:
24        return 30.0

```

Funkcija PID regulatora je implementirana na način da se njegov izlazni signal preskalira na vrijednost između 0 i 1. S obzirom da je taj signal ulaz u grijač, ovime smo dobili regulaciju rada grijača u rasponu od 0% do 100%.

```

1  def pid_controller(error , K_p, K_i, K_d, dt):
2      """
3      PID regulator.
4      Ulaz: signal greške; razlika između željene i stvarne temperature
5      Izlaz: upravljački signal za grijač
6      """
7      global integral , prev_error
8      P = K_p * error
9      integral += error * dt
10     I = K_i * integral
11     D = K_d * (error - prev_error) / dt
12
13     u = P + I + D
14     u = max(0.0 , min(1.0 , u))      #Skaliranje upravljačkog signala na [0, 1]
15
16     prev_error = error                #Ažuriranje stanja za iduću iteraciju
17
18     return u
19
20
21 def heater(control_signal , max_capacity):
22     """
23     Grijač.
24     Ulaz: upravljački signal iz PID regulatora
25     Izlaz: snaga grijača; količina topline koju proizvodi
26     """
27     control_signal = max(0.0 , min(1.0 , control_signal))
28     heat_output = max_capacity * control_signal
29     return heat_output

```

Model procesa - toplinska dinamika prostorije - implementirana je funkcijom (2.12), a stanje je ažurirano metodom Eulerove integracije. Eulerova integracija je najjednostavnija numerička metoda rješavanja diferencijalnih jednadžbi, kod koje je iduće stanje jednako zbroju trenutnog stanja i diferencijala pomnoženim s vremenskim korakom.

```

1  def room_temperature(T_current , Q, T_outside , C, tau , dt):
2      """
3      Model procesa.
4      Ulazi: snaga grijača , toplinski kapacitet prostorije , temperature
           prostorije i okoline , vremenska konstanta
5      Izlaz: promjena temperature u prostoriji ovisno o ulazima
6      """
7      dT_dt = (Q / C) - ((T_current - T_outside) / tau)
8      T_new = T_current + dT_dt * dt                                #Eulerova integracija
9      return T_new
10
11
12  def temperature_sensor(T_actual , k_T):
13      """
14      Temperaturni senzor.
15      Ulaz: trenutna temperatura u prostoriji
16      Izlaz: izmjerena temperatura; ekvivalentni naponski signal
17      """
18      T_measured = k_T * T_actual
19      return T_measured

```

2.2.3. Glavna funkcija i kreiranje animacije

Nakon inicijaliziranja varijabli i polja potrebnih za simulaciju, stvoren je jedan *figure* s 3 *subfigurea*. Na prvom je prikazana referentna i stvarna temperatura u prostoriji, na drugom temperatura okoline, a na trećem snaga kojom grijač grije prostoriju, odnosno koliko topline predaje prostoriji. Inicijalizacija varijabli i slika prikazana je u nastavku.

```

1  #Inicijalizacija varijabli
2  T_current = T_measured    #Početna temperatura
3  time_elapsed = 0.0        #Inicijalizacija vremena
4  time_steps = []           #Polje za vrijeme
5  temperatures = []         #Polje za temperature
6  control_signals = []      #Polje za upravljački signal
7  heater_outputs = []       #Polje za snagu grijača
8
9  #Stvaranje figurea
10  fig , (ax1 , ax2 , ax3) = plt.subplots(3 , 1 , figsize=(10 , 10))
11  fig.tight_layout(pad=5)
12
13  #Plot 1: Temperatura u prostoriji
14  ax1.set_xlim(0 , simulation_time)
15  ax1.set_ylim(10 , 40)
16  line1 , = ax1.plot([], [], label='Stvarna temperatura (°C)', color='blue')
17  line_ref , = ax1.plot([], [], label='Željena temperatura (°C)', color='red',
           linestyle='--')

```

```

18 ax1.set_title("Temperatura u prostoriji")
19 ax1.set_xlabel("Vrijeme (s)")
20 ax1.set_ylabel("Temperatura (°C)")
21 ax1.legend()
22 ax1.grid(True)
23
24 #Plot 2: Temperatura okoline
25 ax2.set_xlim(0, simulation_time)
26 ax2.set_ylim(-5, 15)
27 line2, = ax2.plot([], [], label='Vanjska temperatura (°C)', color='green')
28 ax2.set_title("Vanjska temperatura")
29 ax2.set_xlabel("Vrijeme (s)")
30 ax2.set_ylabel("Temperatura (°C)")
31 ax2.legend()
32 ax2.grid(True)
33
34 #Plot 3: Snaga grijača
35 ax3.set_xlim(0, simulation_time)
36 ax3.set_ylim(0, k + 100)
37 line3, = ax3.plot([], [], label='Snaga grijača (W)', color='orange')
38 ax3.set_title("Izlazna snaga grijača")
39 ax3.set_xlabel("Vrijeme (s)")
40 ax3.set_ylabel("Snaga (W)")
41 ax3.legend()
42 ax3.grid(True)

```

U glavnoj funkciji pozivaju se sve prethodno definirane funkcije podsustava. Ova funkcija je ujedno i ažuriranje varijabli/polja za prikaz na animaciji. Kao dodatna provjera i kao pomoć pri parametriranju regulatora, korištena je mogućnost ispisa varijabli u konzoli (Anaconda Prompt) - vremena, temperature u prostoriji, upravljačkog signala i snage grijača.

```

1 def update(frame):
2     """
3     Iteracijska petlja za simulaciju i animaciju.
4     Pozivanje ranije definiranih funkcija u svakom ciklusu izvršavanja i
5     spremanje rezultata u polja.
6     """
7     global T_current, time_elapsed, integral, prev_error
8
9     #Dohvaćanje temperature okoline:
10    T_outside = outside_temperature(time_elapsed)
11    #Dohvaćanje željene temperature u prostoriji:
12    T_ref = reference_temperature(time_elapsed)
13    #Skaliranje željene temperature u napon:
14    T_ref_scaled = k_T * T_ref
15

```

```

16     #Dohvaćanje trenutne temperature senzorom:
17     T_measured = temperature_sensor(T_current , k_T)
18     #Računanje signala greške:
19     error = T_ref_scaled - T_measured
20
21     #Pozivanje PID regulatora:
22     control_signal = pid_controller(error , K_p, K_i, K_d, dt)
23
24     #Pozivanje grijača:
25     Q = heater(control_signal , k)
26
27     #Pozivanje modela procesa:
28     T_current = room_temperature(T_current , Q, T_outside , C, tau , dt)
29
30     time_elapsed += dt                #Ažuriranje vremena
31
32     #Spremanje rezultata za prikaz u animaciji
33     time_steps.append(time_elapsed)
34     temperatures.append(T_current)
35     control_signals.append(control_signal)
36     heater_outputs.append(Q)
37
38     #Ispis vrijednosti u konzoli (korišteno za otklanjanje grešaka i
39     #parametriranje regulatora)
40     print(f"Time: {time_elapsed:.1f}s, Temp: {T_current:.2f}°C, u: {
41     control_signal:.2f}, Q: {Q:.2f}W")
42
43     #Ažuriranje grafova
44     line1.set_data(time_steps , temperatures)
45     line_ref.set_data(time_steps , [reference_temperature(t) for t in
46     time_steps])
47     line2.set_data(time_steps , [outside_temperature(t) for t in time_steps])
48     line3.set_data(time_steps , heater_outputs)
49
50     return line1 , line2 , line3

```

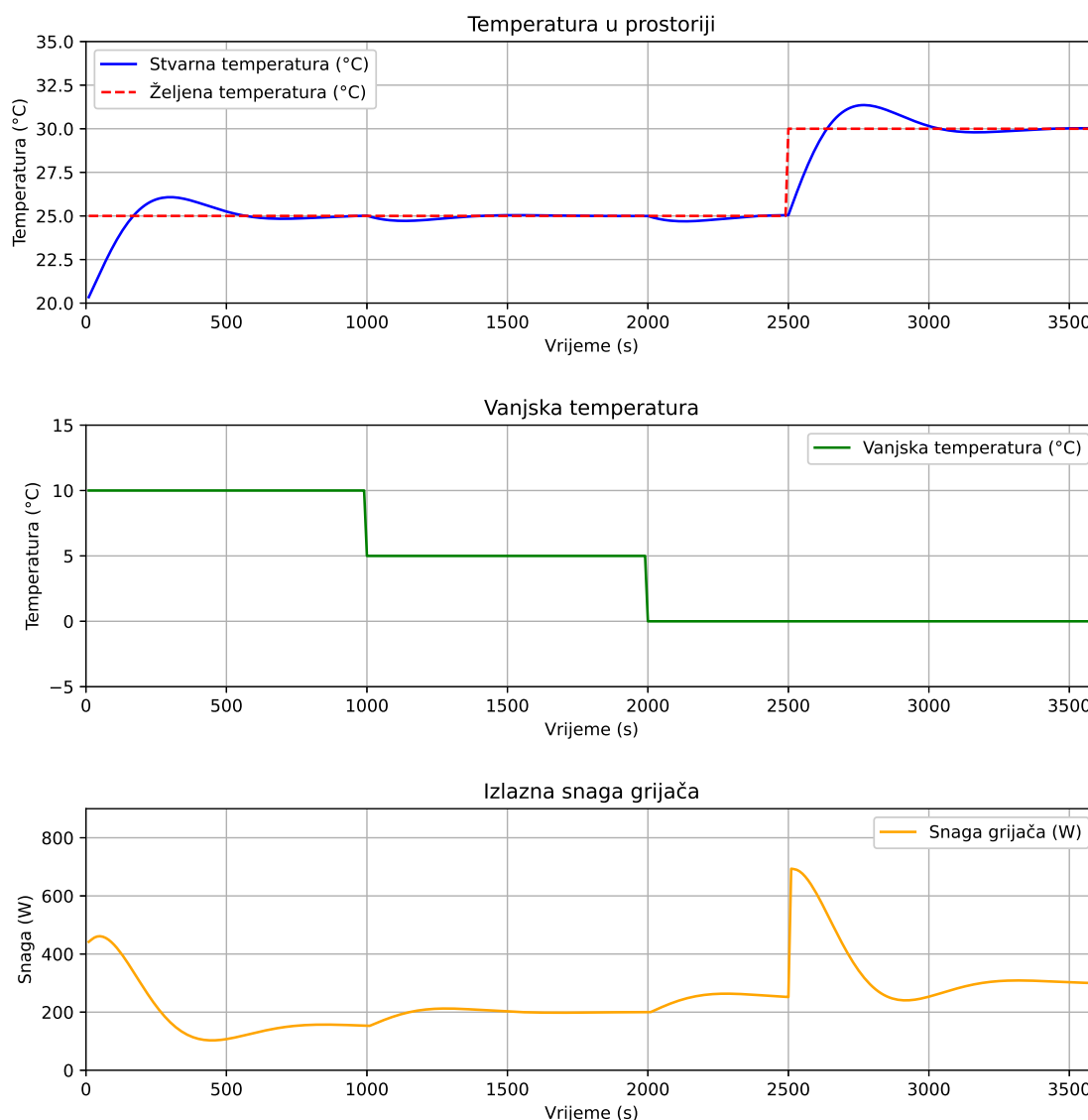
Na samom kraju je pozvana funkcija iz matplotlib.animation paketa - FuncAnimation - koja kao argumente prima parametre potrebne za izvršavanje simulacije poput *figurea*, podataka, varijabli i polja, intervala, itd.

```

1     #Pozivanje animacije
2     ani = FuncAnimation(fig , update , frames=int(simulation_time/10) , blit=False ,
3     interval=0.1)
4     plt.show()

```

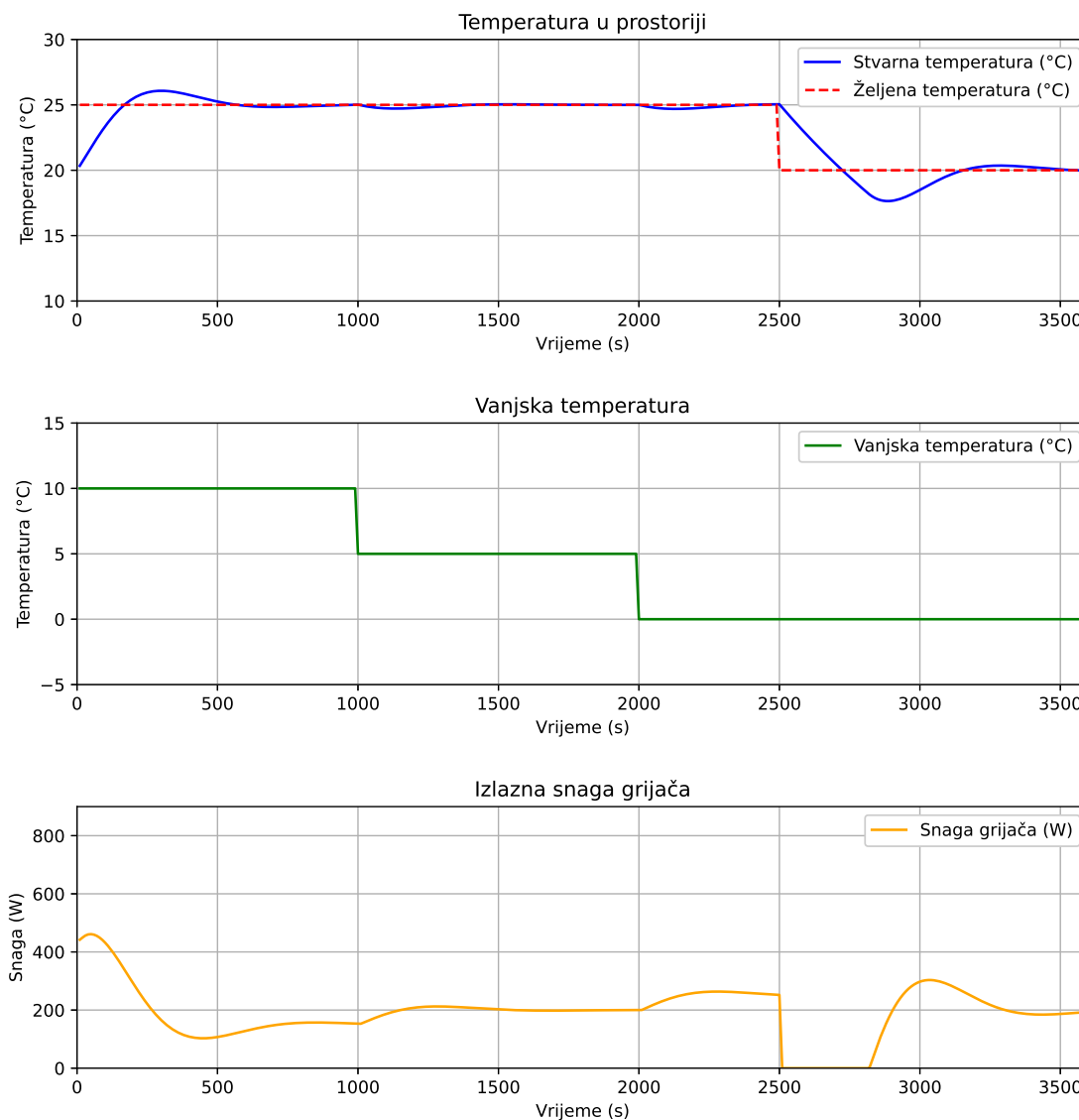
2.2.4. Rezultati



Slika 2.5. Rezultati simulacije s povećanjem referentne temperature (grijanje)

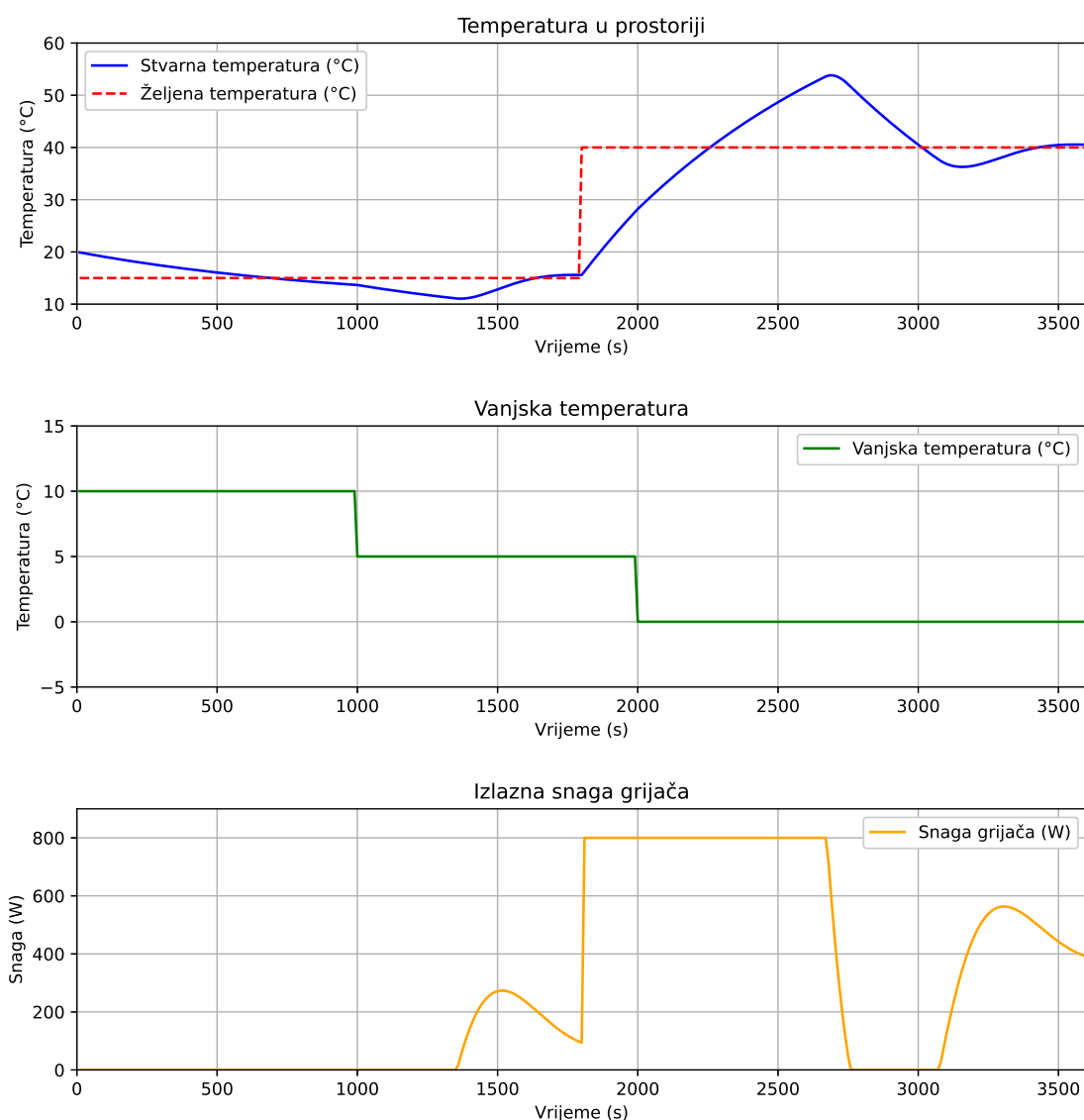
Slika 2.5 prikazuje rezultate dobivene implementacijom ove simulacije. Vanjska temperatura skokovitim promjenama pada s 10°C na 5°C, te zatim na 0°C. Naravno da ovakve skokovite promjene u vanjskoj temperaturi nisu realne, ali su implementirane na ovaj način kako bi se vidjele performanse regulacijskog sustava - vidljivo je da temperatura u prostoriji malo propadne prije nego je sustav izregulira natrag na željenu vrijednost. Također, vidljivo je da propadom vanjske temperature, unatoč tome što je referentna temperatura ista, snaga grijača poraste. To je i logično s obzirom da se toplina u prostoriji pasivno disipira ovisno o toplinskom otporu i toplinskom kapacitetu prostorije, te o razlici temperatura u prostoriji i vani.

Slika 2.6 prikazuje slučaj smanjenja referentne temperature s 25°C na 20°C . U tom slučaju, snaga grijača padne na 0W , odnosno grijač se isključi. Prostorija se tada hladi pasivno, tj. disipacijom topline u okolinu.



Slika 2.6. Rezultati simulacije sa smanjenjem referentne temperature (hlađenje)

Karakteristike odziva temperature su u slučaju hlađenja lošije - ima malo veći prebačaj i vrijeme ustaljivanja zbog toga što je grijač ušao u stanje zasićenja. Naime, da snaga može biti negativna, ona bi u ovom slučaju otišla ispod 0W , što je naravno nemoguće. Karakteristike odziva bi se mogle poboljšati preciznijim parametriranjem regulatora.



Slika 2.7. Rezultati simulacije s ekstremnim promjenama referentne temperature i zasićenjem

Slika 2.7 prikazuje ekstreman slučaj prethodno opisanog problema. Sada i u slučaju grijanja prostorije grijač ulazi u zasićenje (maksimalna snaga mu je 800W), ali regulator i dalje nastoji forsirati temperaturu. Ova pojava je poznata kao *integrator windup* i neizbježna je u sustavima u kojima aktuatori imaju neke limite, što je u praksi slučaj gotovo uvijek. Ovaj problem je rješiv, međutim u ovoj simulaciji nije implementiran zbog kompleksnosti, ali i realističnosti - ovakve nagle promjene u vanjskoj ili referentnoj temperaturi se u stvarnosti ne očekuju.

3. Zaključak

Temperaturni regulacijski proces je uspješno implementiran. Za kućne potrebe, kao naprimjer grijanje kuhinje ili spavaće sobe, ovaj sustav zadovoljava kriterije. Međutim, za neke industrijske potrebe gdje regulacija temperature ima stroge zahtjeve na performanse, bilo bi nužno implementirati i neke zaštitne algoritme kako bi se riješio spomenuti *integrator windup*.

Završna napomena je da je korišteni grijač **FIREROD Cartridge Heater** pogodniji za grijanje fluida/tekućina, međutim princip i algoritam upravljanja su potpuno identični - jedina razlika je u parametrima grijača i njegovim performansama, te potrošnji energije.

Bibliografija

- [1] **FIREROD®Cartridge Heater** Datasheet
- [2] M. Posani, R. Veiga, V. Freitas: "Post-Insulating traditional massive walls in Southern Europe: A moderate thermal resistance can be more effective than you think", Science Direct, Energy and Buildings Volume 295, 2023.
- [3] **Texas Instruments LM35 Analog Temperature Sensor** Datasheet