

Smart Power - Dokumentacja

Zespół: Drineczki (W. Łazarski, J. Radzimiński, J. Szumski, K. Kamieniarz)

1. Analiza występujących procesów

W całym cyklu grzewczym wyróżniamy dwa główne procesy: ogrzewanie wody oraz utrzymanie odpowiedniej temperatury. Są to dwa główne źródła, które generują zapotrzebowanie na energię elektryczną. Dodatkowo mamy jeszcze pobór energii na pozostałe urządzenia, jednak przyjmujemy, że jest on stały dla danych godzin operowania.

Ogrzewanie wody:

Zbiornik na wodę pozwala na nagromadzenie 150 litrów ciepłej wody. Średnie zużycie dobowe wody wynosi 180 litrów. W trakcie całego procesu zakładamy, że nasz zbiornik nie traci ciepła wody, nie potrzebujemy więc energii aby utrzymać jej temperaturę. Zakładamy, że w momencie włączenia nagrzewania nasz zbiornik wypełniony jest wodą.

Utrzymanie temperatury w domu:

Konsumpcja energii przez system ogrzewania w głównej mierze zależy od temperatury panującej na zewnątrz. Determinuje ona tempo utraty ciepła z domu oraz pobór mocy wymagany na podniesienie temperatury o 1 stopień Celsjusza.

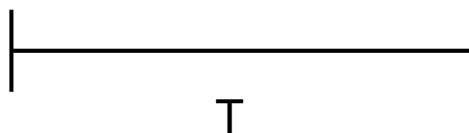
Dodatkowo w domu dostępny jest rekuperator, który pozwala na wyrównywanie średniej temperatury między pokojami. System ogrzewania dąży do utrzymania temperatur zdefiniowanych przez użytkownika.

Mamy dostępne następujące źródła energii elektrycznej:

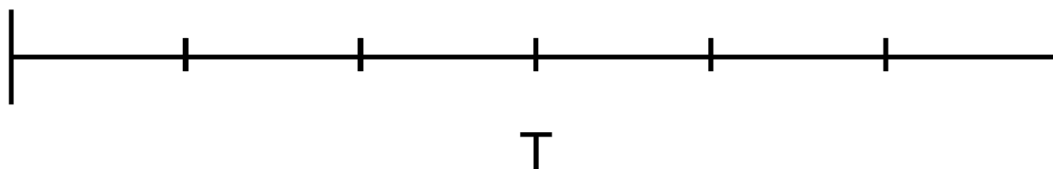
- ogniwa fotowoltaiczne
- akumulator
- sieć elektryczną

2. Wybór i uzasadnienie podejścia

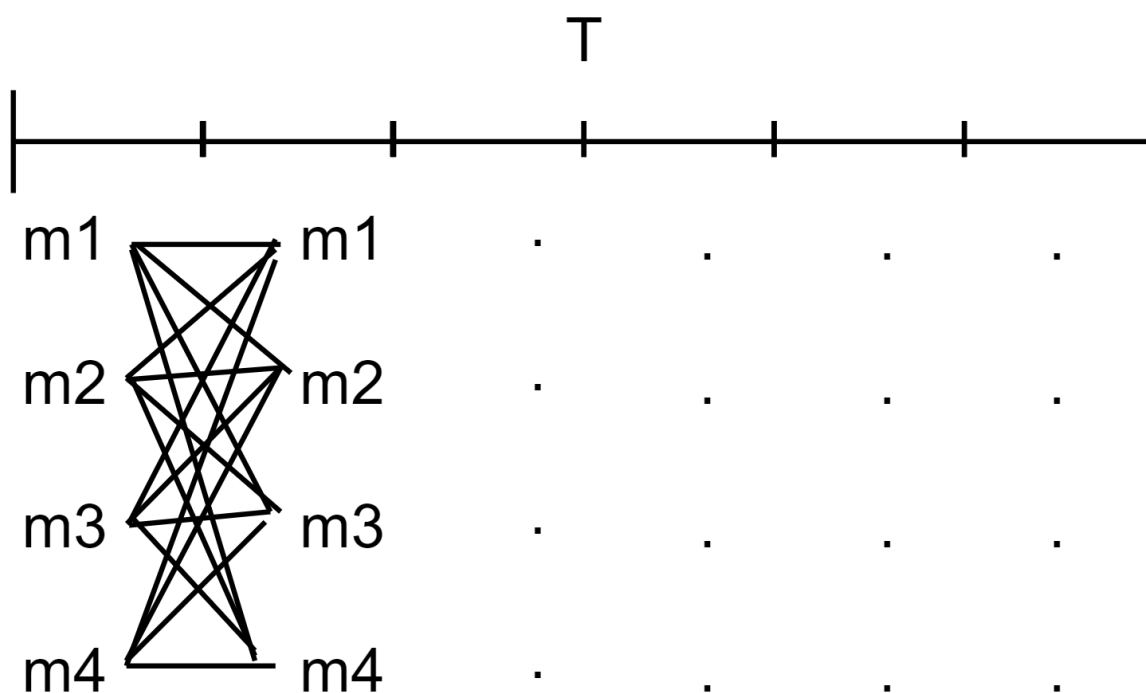
Zacznijmy od zdefiniowania sobie pewnego okresu czasowego T.



Następnie T dzielimy na równe okresy z których każdy wynosi np. godzinę



W kolejnym kroku obliczamy wszystkie możliwe kombinacje pracy naszego systemu zarządzania energią.



Gdzie m_i to tryb działania systemu

Następnie dla każdego odcinka czasowego w T obliczamy koszt wydanej energii z sieci przy uwzględnionych parametrach wyjściowych z poprzedniego odcinka czasowego (ilość mocy

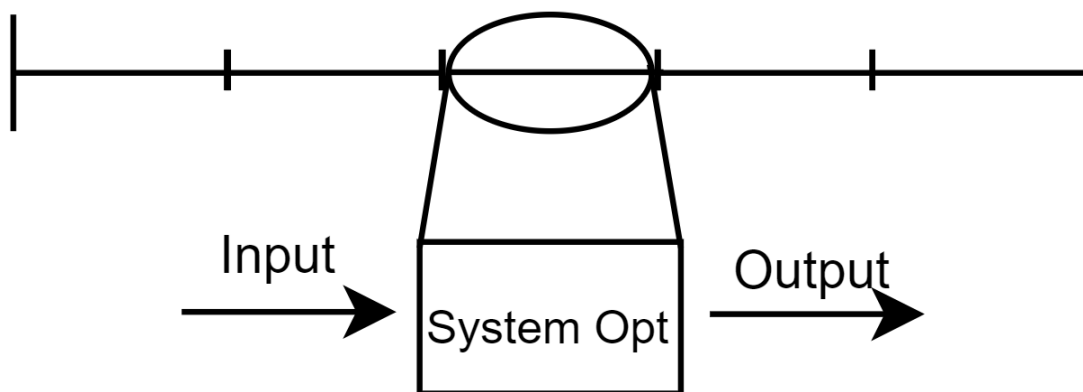
akumulatora, temperatura pomieszczenia) Dodatkowo wykorzystujemy OpenWeather API aby prognozować pogodę i zachmurzenie na następny odcinek czasowy.

Input:

- temperatura pomieszczenia osiągnięta w poprzednim odcinku czasowym
- wartość naładowania akumulatora osiągnięta w poprzednim odcinku czasowym
- przewidywana temperatura dla nowo rozpatrywanego odcinka czasowego (OpenWeather API)
- przewidywane zachmurzenie dla nowo rozpatrywanego odcinka czasowego (OpenWeather API)

Output:

- nowa temperatura pomieszczenia po odcinku czasowym przy założeniu trybu m_i
- nowa wartość mocy akumulatora przy założeniu trybu m_i
- estymowana wartość kosztów energii dla danej iteracji



Ostatecznie, sumujemy całkowity koszt energii elektrycznej z sieci dla całego ciągu trybów działania

$$\{m_1, m_4, m_2, m_1\} \rightarrow 5$$

$$\{m_2, m_1, m_2, m_4\} \rightarrow 7$$

$$\{m_4, m_1, m_4, m_2\} \rightarrow 6$$

$$\{m_1, m_1, m_3, m_3\} \rightarrow 13$$

Jako tryb pracy systemu w następnym odcinku czasowym wybieramy pierwszy tryb z ciągu, który zwrócił najmniejszy przewidywany koszt.

3. Sposób wykorzystania poszczególnych informacji i danych

Wszystkie dostępne informacje są wykorzystane w celu estymacji energii elektrycznej którą użytkownik będzie zmuszony pobrać z sieci aby następnie wyliczyć koszty tego poboru. Podane informacje tworzą nietrywialną funkcję, która następnie minimalizujemy algorytmem opisanym wcześniej.

4. Wybór technologii, wymagania systemowe

Jako główną architekturą systemu zastosowaliśmy standardową architekturę webową - klient-serwer. Klientem jest aplikacja webowa napisana w TypeScript przy użyciu frameworku React.js. Serwer został napisany w Pythonie jako formie REST API, przy użyciu frameworka Flask. Algorytmy użyte w funkcjach optymalizacyjnych również zostały napisane w Pythonie jako oddzielne moduły wykorzystywane przez serwer. Całość komunikacji odbywa się przy użyciu zapytań HTTP.

Taka architektura pozwala na uruchomienie aplikacji na każdym urządzeniu obsługującym przeglądarki internetowe. Dodatkowo, większość kalkulacji odbywa się na oddzielnym serwerze (docelowo umieszczonym w chmurze) przez co moc obliczeniowa danego urządzenia nie jest istotna. Serwer w chmurze pozwoliłby również na łatwe przyłączanie kolejnych modułów / urządzeń do systemu w przyszłości.

5. Testowalność rozwiązania

Testy naszego systemu polegałyby na stworzeniu scenariuszy symulacji systemu i monitorowania rozwiązania pod względem estymaty kosztów względem ludzkiego ustalania trybów pracy. Oczywiście szereg scenariuszy testowych pozwoliłby nam na monitorowanie systemu i jego zachowań w różnych sytuacjach.

Scenariusz to nic innego jak asynchroniczny szereg wywołań różnych zmian dziejących się zarówno w domu i jego otoczeniu. Dzięki licznym „mockowym” modułom symulującym urządzenia domowe (które stworzyliśmy jako niezależne moduły Pythonowe), system można przetestować niezależnie od brakujących w danym momencie (fizycznych) urządzeń czy elementów.

6. Zrealizowanie sterowania zarządzaniem energią

Sterowanie trybem działania sterownika zarządzającego energią zaimplementowane jest po stronie serwera przy użyciu algorytmu opisanego powyżej.

7. Biblioteka obsługująca sterowalne komponenty

W celu testowania odpowiednich komponentów systemu stworzyliśmy moduł pythonowy działający po stronie serwera, który imituje działanie urządzeń w różnych porach roku - aby dostosować temperatur - oraz dnia - aby dostosować nasłonecznienie, potrzebne w celu wyliczenia efektywności paneli fotowoltaicznych.

8. Konfigurowalność

System jest w pełni konfigurowalny oraz skalowalny, wymaga to tylko i wyłącznie zaimplementowania i wdrożenia kolejnych implementacji urządzeń. Sama konfiguracja wartości działania urządzeń dla danych odcinków czasowych można zostać ustawiona na konfigurację domyślną - sparsowane z napisanego api lub ustawić sztywno na wartość domyślną. Żadna z tych opcji nie wyklucza stworzenia modułu do manualnego ustawienia tych parametrów przez użytkownika.

9. Niezawodność systemu

System jest niezawodny w tym sensie, że automatycznie dobiera możliwie najlepszy tryb pracy biorąc pod uwagę przewidywania pogodowe dot. przyszłości. Jednocześnie jeśli system niekoniecznie dokona prawidłowej estymaty np. poprzez złą prognozę pogody to ciągle odpowiedni tryb pracy dostarczy odpowiednie zapotrzebowanie energetyczne. Z punktu widzenia użytkownika nic się nie zmieni - dom sam w inteligentny sposób będzie się starał dobrać tryb pracy systemu aby dostosować odpowiednią temperaturę powietrza i wody, minimalizując przy tym koszty energii elektrycznej. .