Összefoglaló(1)

Abstract(1)

Bevezetés

* Feladat(0,5)
* Szakdolgozat felépítése(1)

Szabályzó feladata(1)

Szabályzó basic működése(0,5)

Megvalósított teszt szabályzó(0,5)

Egyes egységek bemutatása

* Inverter(Napelem Park)(1,5)
* Loadbank(Célja felhasználása)(0,5)
* Engine(0,5)
* Battery(0,5)

Szimulációk alapvető koncepció(3)

Digital Twin koncepció bemutatása(2)

Szimulációs eredmények(3)

Technológiák bemutatása

* Kotlin(2)
* DSL(1)
* Gradle(1)
* IntelliJ(0,5)
* Kalasim(4)
  + Open Source
  + Koin probléma

Tervezési megfontoltságok

* Hiearchia(ok-okozat)

Összefoglalás

A mai világban az elektromos áram mindenhol megjelenik. Az autóiparban egyre nagyobb teret nyer, bármilyen munkát akarunk végezni ahoz kell, valamint otthonaikat már el sem tudnánk képzelni lámpa, mosógép, wifi, TV és egyéb háztartási eszközök nélkül. Azért nagyon fontos az, hogy hogyan termeljük az áramot. Erre már nagyon sok megoldást találtunk.

A két fő kategória, a megújuló és a nem megújuló között, azon kívül, hogy csak a megújuló terem újra emberi időléptékkel, az is nagy különbség, hogy lehet az egyes termeléseket, milyen pontossággal lehet megjósolni. A kőolajból és földgázból való energia előállítás során, pontosan meglehet mondani azt, hogy adott mennyiségű fűtőanyagból mennyi elektromosáramot tudunk generálni. Ezzel ellentétben, viszont a folyó folyásának erősége, a szél sebességét és a napsütés mennyiségét nem lehet pontosan megjósolni.

Az energiatermelés fejlettségé mellett viszont, a tárolásra való megoldások le vannak maradva. Ez azért probléma, mivel a háztartásokban sem egy fix mennyiségű fogyasztás van. Ezen okok, miatt az energiatermelésnek szüksége van egy változó komponensre, amely követni tudja a változó fogyasztást is, ezek lesznek a megújuló energiát felhasználó termelők.

Ezen termelők érdeke az, hogy pontosan annyit termeljenek amennyit felvesznek az egyes fogyasztók, mivel ők annyit tudnak eladni. Emiatt van szükség a szabályzásra, amely a felsőbb szervezettől kapott termelési mennyiséget betudja állítani a telepeken és ezzel minimalizálva a feleslegesen termelt energiát.

A szabályzók tesztelése viszont nem egy egyszerű feladat, mivel legegyszerűbben éles környezetben lehetne tesztelni, viszont ez egyértelmű okokból nem a legszerencsésebb. Ezért szakdolgozatom témajaként egy olyan szimulációt valósítottam meg, mely különböző termelő vagy fogyasztó egységeket működését másolja le, ez a Digitális iker koncepciója, és ezeken lehessen tesztelni a szabályzó kódját.

A szakdolgozatban befogom mutatni a szabályzók működését, a szimulált egységeket ismertetem, valamint bemutatom a megvalósításhoz használt technológiákat, koncepciókat, majd a megoldásom fogom részletezni.

# Bevezetés

Napjainkban a fogyasztott áram mennyiségét nem lehet pontosan meghatározni, mivel például a háztartásokban nem tudjuk pontosan megmondani, hogy adott napon mennyit fogun fogyasztani. Emiatt van szükségünk szabályzókra, melyek dinamikusan változtatják a telepek energia teremlését.

A szakdolgozatom célja, hogy egy ilyen szabályzót lehessen kontrolált környezetben tesztelni, egy szimuláción keresztül. Ennek parémeteri, hogy mely termelő vagy fogyasztó egységek szerepeljenek a szimulációba, hogy mi legyen a milyen random értékkel induljon ezzel befolyásolva a végeredményeket, hogy valós idejű legyen-e a szimuláció, valamint egy objektum, amiben a konstans értékeket lehet beállítani.

A második fejezetben a szabályzó feladatát és működését fogom bemutatni és azt, hogy az általam megvalósított szabályzó milyen logika alapján működik. Ezek mellett bemutatom az egyes szimulált egységek feladatát és működését, mint például az invertert, a terhelőt (loadbank), a gázmotort és az akkumulátort.

Utána a szimulációkról fogok általánosságban írni, valamint bemutaton részletesen a digitális iker koncepcióját.

A negyedik fejezetben a használt technológiákkal fogom folytatni. A Kotlin nyelven írtam, így ezt fogom majd részletezni, belemerülve a nyelv által nyújtott DSL (Domain Specific Language)-be is. Utána a keretrendszert fogom bemutatni melyet használtam, valamint beszélek róla, milyen élményeim voltak vele. Ez egy nyitott forráskodú projekt, amely a Kalasim névre hallgat. Technológiák közé fog még tartozni a Gradle, IntelliJ IDEA, Koin , Kotlin Coroutines.

Ezek után kezdem el kifejteni a szimulációt és a benne lévő megoldásaimat, valamint futási eredményeket is mutatok majd diagrammok formájában.

A végén pedig kitérek, hogy milyen fejlődési lehetőségei vannak még a szimulációnak.