

Diplomová práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra počítačů

Simulátor hybridního zdroje elektrické energie

Bc. Petr Stejskal

Vedoucí: doc. Ing. Přemysl Šůcha, Ph.D.
Studijní program: Otevřená informatika
Specializace: Softwarové inženýrství
Květen 2023

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Stejskal** Jméno: **Petr** Osobní číslo: **466225**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra počítačů**
Studijní program: **Otevřená informatika**
Specializace: **Softwarové inženýrství**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Simulátor hybridního zdroje elektrické energie

Název diplomové práce anglicky:

Simulator of a hybrid source of electric energy

Pokyny pro vypracování:

Simulátor hybridního zdroje elektrické energie

Cena energií a způsob její výroby je v dnešní době prioritní otázka rezonující celou Evropou. Důležitou složkou pro poskytování elektrické energie jsou takzvané podpůrné služby, sloužící k zajištění spolehlivosti energetické soustavy. Cílem této práce je návrh simulátoru - digitálního dvojčete, které bude simulovat chování skutečného hybridního zdroje elektrické energie vzhledem k danému řídicímu algoritmu a požadavkům na plnění podpůrných služeb. Cílem je vyhodnotit chování algoritmů, řídicí hybridní zdroj elektrické energie, a posoudit schopnost hybridního zdroje plnit požadavky na poskytované podpůrných služeb. Úkoly práce jsou:

- 1) Seznamte se vlastnostmi hybridního zdroje elektrické energie a parametry jednotlivých podpůrných služeb.
- 2) Specifikujte požadavky na simulátor hybridního zdroje a navrhnete jej.
- 3) Implementujte simulátor podle návrhu.
- 4) Vyhodnoťte chování simulátoru.

Seznam doporučené literatury:

- [1] Juan I. Pérez-Díaz, Ignacio Guisández, Manuel Chazarra, Arild Helseth, Medium-term scheduling of a hydropower plant participating as a price-maker in the automatic frequency restoration reserve market, Electric Power Systems Research, Volume 185, 2020.
- [2] H. Firoozi, H. Khajeh and H. Laaksonen, Optimized Operation of Local Energy Community Providing Frequency Restoration Reserve, IEEE Access, Volume 8, page 180558-180575, 2020.
- [3] D. Sojma, Simulation and optimization of laboratory processes, Master's thesis, Czech Technical University in Prague, 2022.

Jméno a pracoviště vedoucí(ho) diplomové práce:

doc. Ing. Přemysl Šůcha, Ph.D. katedra řídicí techniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **11.07.2022**

Termín odevzdání diplomové práce: _____

Platnost zadání diplomové práce: **19.02.2024**

doc. Ing. Přemysl Šůcha, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

Datum převzetí zadání

Podpis studenta

Poděkování

Děkuji doc. Ing. Přemyslu Šůchovi Ph.D. za spolupráci během projektu s cílem vytvoření digitálního dvojče a vedení mé diplomové práce. Jsem velice rád, že jsem mohl pracovat na práci, která nezapadla do šuplíku a má reálný dopad. Rovněž poděkování patří mým rodičům za podporu po celou dobu studia. V neposlední řadě děkuji své přítelkyni, která vnesla světlo do mého života.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 25. května 2023

Abstrakt

Diplomová práce se zabývá návrhem simulátoru hybridního zdroje skládajícího se ze spalovacích turbín a bateriového systému. Účelem zařízení je působit v oblasti podpůrných služeb, které zajišťují spolehlivost energetické soustavy. Cílem práce je vytvořit digitální dvojče, které umožní provést simulaci chování hybridního zdroje vzhledem k danému řídicímu algoritmu a požadavkům na plnění podpůrných služeb. Systém je koncipován, aby bylo možné integrovat různé řídicí algoritmy, které komunikují se simulací pomocí univerzální rozhraní. Pro ilustraci funkčnosti celého konceptu je implementováno jednoduché tzv. manuální řízení vydávající řídicí povel podle specifikované sekvence dané souborem. První část práce se zabývá popisem podpůrných služeb, způsobem jejich uvažované realizace v rámci pravidel a bližším popisem konceptu hybridního zdroje. Dále následuje parametrický popis technologií a nastavení simulace, rozbor myšlenky rozhraní mezi řídicím systémem a simulací. V druhé části je popsán návrh řešení digitálního dvojčete pomocí požadavků, architektury a zvolených technologií. Výsledek práce je demonstrován na ukázce jednoduchého příkladu použití s popsánými vstupy a výstupy. Součástí je také složitější scénář pro ilustraci, jak některé výstupy vypadají na reálnějším použití. Implementace je provedena v jazyce Python, pro simulaci využita knihovna SimPy, pro grafy Plotly.

Klíčová slova: Digitální dvojče, hybridní zdroj, podpůrné služby, simulace

Vedoucí: doc. Ing. Přemysl Šůcha, Ph.D.

Abstract

The thesis is focused on the design of a simulator for a hybrid power source composed of combustion turbines and a battery system. The purpose of the facility is to provide ancillary services that ensure the reliability of the energy system. The goal of the thesis is to create a digital twin that allows simulating the behavior of the hybrid power source based on a given control algorithm and requirements for providing ancillary services. The system is designed to integrate various control algorithms that communicate with the simulation through a universal interface. To illustrate the functionality of the entire concept, a simple manual control, issuing control commands according to a specified sequence defined in a file, is implemented. The first part of the thesis describes the ancillary services, the proposed realization within the rules, and provides a detailed description of the hybrid power source concept. It also includes a parametric description of the technologies and simulation settings, as well as an analysis of the interface between the control system and the simulation. The second part presents the design of the digital twin, including requirements, architecture, and selected technologies. The outcome of the work is demonstrated through a sample of a simple usage example with described inputs and outputs. Additionally, a more complex scenario is included to illustrate how some outputs appear in a more realistic application. The implementation is carried out in the Python language, utilizing the SimPy library for simulation and Plotly for graphs.

Keywords: Digital twin, hybrid power source, ancillary services, simulation

Title translation: Simulator of a hybrid source of electric energy

Obsah

1 Úvod	1
2 Motivace	3
3 Existující řešení	5
3.1 Koncepty pro plnění podpůrných služeb	5
3.2 Využití digitálních dvojčat	7
4 Přínos práce	11
5 Uvažovaný hybridní zdroj	13
5.1 Podpůrné služby	13
5.1.1 Interpretace a realizace povelu	14
5.1.2 FCR	15
5.1.3 aFRR	16
5.1.4 mFRR	16
5.1.5 RR	16
5.1.6 Shrnutí podpůrných služeb ..	17
5.2 Koncepce zařízení	17
6 Popis problému	21
6.1 Turbíny	21
6.2 Bateriový systém	26
6.3 Podpůrné služby - formát vstupu	26
6.4 Nastavení simulace	27
6.5 Rozhraní řízení	28
7 Návrh řešení	33
7.1 Požadavky	33
7.1.1 Požadavky turbíny	37
7.1.2 Požadavky baterie	46
7.1.3 Požadavky simulace	50
7.1.4 UI aplikace	55
7.2 Architektura	56
7.3 Zvolené technologie	59
8 Experimenty - ilustrace řešení	61
8.1 Ilustrace jednoduchého příkladu	61
8.2 Ilustrace rozsáhlejší simulace ...	74
8.3 Výkonnost digitálního dvojče ...	77
8.4 Zhodnocení výsledku	78
9 Závěr	79
A Literatura	81
B Seznam použitých zkratk	85
C Elektronické přílohy	87

Obrázky

5.1 Ukázka realizace povelu na službu výkonem v čase	15
5.2 Ukázka průběhu výkonu na základě povelu $mFRR_{12,5}$ v čase t	17
5.3 Koncept hybridního zdroje pro poskytování SVR[1]	18
6.1 Základní stavový diagram turbíny	22
6.2 Úplný stavový diagram turbíny .	23
7.1 Architektura digitálního dvojče systému	57
8.1 Ukázka vstupu nominálních výkonů v souboru ve formátu .xlsx	64
8.2 Ukázka vstupu nominálních spotřeb plynu v souboru ve formátu .xlsx	64
8.3 Ukázka výstupu – graf služeb . .	68
8.4 Ukázka výstupu – graf výkonů turbín	69
8.5 Ukázka výstupu – graf bilance . .	70
8.6 Ukázka výstupu – graf spotřeb elektrické energie	71
8.7 Ukázka výstupu – graf spotřeb plynu	72
8.8 Ukázka výstupu – graf bilance, rozsáhlejší scénář	75
8.9 Ukázka výstupu – graf výkonů turbín, rozsáhlejší scénář	76

Tabulky

5.1 Přehled podpůrných služeb	17
6.1 Datová reprezentace stavu	24
6.2 Tabulka nominálních příkonů plynu v závislosti na výkonu a teplotě. Uvažovanou jednotkou jsou kW. Hodnoty jsou smyšlené. Jaké hodnotě odpovídá např. 10% výkonu uvádí následující tabulka 6.4.	24
6.3 Datová reprezentace přechodu mezi stavy	25
6.4 Tabulka nominálních hodnot výkonu pro vyjádření procentem v závislosti na teplotě. Uvažovanou jednotkou jsou kW. Hodnoty jsou smyšlené.	25
6.5 Uvažované parametry pro popis turbíny	26
6.6 Uvažované parametry pro popis baterie	26
6.7 Formát vstupního souboru pro požadavky služeb	27
6.8 Uvažované parametry pro nastavení simulace	28
6.9 Uvažované parametry pro nastavení simulace	31
7.1 Uvedení proměnné času v simulaci	34
7.2 Datová reprezentace turbíny pomocí proměnných	35
7.3 Datová reprezentace baterie pomocí proměnných	36
8.1 Tabulka rychlostí výpočtu simulace dle délky simulovaného období a počtu turbín v simulaci	77

Kapitola 1

Úvod

Elektrická energie má na naši společnost každodenní dopad, někdy díky různým okolnostem si uvědomíme, že není samozřejmostí. Po dlouhá léta jsme zvyklí na tradiční výrobu pomocí uhelných, jaderných, vodních elektráren. Zájmy ochrany klimatu kvalifikují určité zdroje jako emisní a v budoucí energetické koncepci se s nimi přestává počítat. Jejich pozici by měli převzít bezemisní či obnovitelné zdroje. Nespornou výhodou těchto fosilních zdrojů (zejména uhelné elektrárny) je jejich stálost dostupnosti a tím i schopnost neustále působit v oblasti podpůrných služeb, které přispívají k zajištění stability energetické soustavy. Paradoxně výpadkem stálých zdrojů a jejich nahrazením většinou méně stabilními obnovitelnými zdroji se poptávka po podpůrných službách ještě prohlubuje. Otázkou zůstává, jak nahradit postupně odstávající zdroje novými.

V České republice existuje výzkumný projekt, který se zabývá výzkumem nového zdroje, který kombinuje konvenční spalovací turbíny s bateriovým systémem. Myšlenka zdroje je poskytování pouze podpůrných služeb, kdy emise vznikají jen v případě, že zařízení je aktivní v jejich poskytování. Což je výhodou například oproti uhelné elektrárně, kdy zdroj musí běžet neustále, ať se podílí na stabilizaci či nikoliv. Vytvořením nového zdroje schopného poskytovat podpůrné služby získáme možnost nahradit tyto původní fosilní a přispět k dosažení vytyčených cílů dekarbonizace. Spojení dvou technologií si však žádá vyšší nároky na řízení, které musí být spolehlivé a schopné plnit předem neznámý průběh výkonu, který je potřeba do sítě uvolnit. Pro potřeby výzkumu tohoto konceptu z hlediska řízení bylo rozhodnuto o vytvoření digitálního dvojče. Myšlenka digitálního dvojče byla představena v roce 2002[2] a můžeme ho definovat jako virtuální reprezentaci reálného systému, která je aktualizována na základě výměny informací mezi reálným a virtuálním systémem[3].

Cílem této práce je vytvoření digitálního dvojče zmíněného konceptu hybridního zdroje. Jelikož žádný reálný systém nemáme k dispozici, digitální dvojče budeme koncipovat jako simulátor, do kterého vložíme vstupní informace a na výstupu budeme moci vidět, jak zdroj pracoval. Výsledkem by mělo být, že zadavatel (řešitelé výzkumného projektu) bude díky němu lépe schopen studovat chování zdroje a testovat nad ním různé parametrizace technologií a implementace řízení vznikající během vývoje řídicího algoritmu.

Tato práce čtenáře nejdříve provede kratičkým povídáním o energetice a s ní souvisejícími pojmy v rámci motivace. Díky rámcové představě o základních pojmech navážeme řešerší zabývající se koncepty pro plnění podpůrných služeb v první části, abychom získali představu, jak lze podpůrné služby realizovat mimo tradiční zdroje. V druhé části se budeme zajímat o využití konceptu digitálního dvojče. Na závěr uvedeme přínosy této práce.

Po úvodní části navážeme širším popisem problematiky podpůrných služeb, aby čtenář mohl pod tímto pojmem získat konkrétnější představu. Rovněž důkladněji popíšeme koncept uvažovaného hybridního zdroje. Poté se budeme zabývat problematikou turbín a bateriového systému. Zaměříme se na popis jejich vlastností a vstupů týkajících se těchto technologií. Se vstupy popíšeme rovněž parametry simulace a v jakém formátu budou uváděny požadavky na podpůrné služby. Uvedeme základní myšlenky o propojení řízení se zbytkem systému. Na základě těchto informací provedeme formulaci požadavků na digitální dvojče, které budou popisovat i dílčí koncepční fungování komponent. Specifikaci systému doplníme popisem architektury, který nám koncepci zachytí do komponent a ukáže vazby mezi nimi. Celý rozbor uzavřeme pojednáním o využitých technologiích při implementaci.

Po rozboru řešení ukážeme výsledky podoby digitálního dvojče. Na jednoduchém příkladu popíšeme veškeré vstupy ve finální podobě v rámci vstupních souborů. Rovněž ukážeme výstupy v podobě logu událostí a grafů, které vizualizují nasbíraná data o chování v čase. Po seznámení se základním příkladem ukážeme i podobu výsledku na delším časovém horizontu. Na závěr se budeme zabývat výkonností simulace vzhledem k délce simulovaného horizontu a počtu turbín.

Na závěr provedeme zhodnocení dosaženého výsledku a nastíníme další možné využití vytvořeného digitálního dvojče.

Kapitola 2

Motivace

V České republice se energetika v poslední době stala opět aktuálním tématem a je jí věnována větší pozornost než dříve. Skutečnost, že není radno robustnost energetiky podceňovat nám připomněla hlavní událost roku 2022 a tou bylo rozpoutání války na Ukrajině a v jejím důsledku tzv. energetická krize. Dědictvím minulé éry a režimu bylo velké množství zbudovaných zdrojů energie, které dostatečně pokrývali naši spotřebu. Tento fakt potvrzuje, že jsme dlouhou dobu byli a ještě stále jsme exportní zemí v oblasti elektřiny. Tato skutečnost se však v řádu několika let změní a staneme se naopak zemí závislou na dovozu této komodity.

Jednou z motivací budování velkého množství zdrojů byla myšlenka, pokud existuje více zdrojů než potřebujeme na pokrytí aktuální spotřeby, můžeme si vybírat, které zdroje aktuálně využít především s ohledem na cenu. Z dlouhodobého pohledu spotřeba elektřiny nadále rostla a nedocházelo k budování nových zdrojů. Toto vyústilo v situaci, kdy kapacity aktuálních zdrojů téměř pokrývají spotřebu, čili nemůžeme si už vybírat nejlevnější zdroje pro aktivaci, ale cena nám je určena tím co máme. Tento trend opomenutí budování nových zdrojů není jen u nás, ale objevuje se napříč Evropou.

Druhým tlakem je tzv. Green deal, který se zaměřuje na omezení emisí CO_2 . V oblasti energetiky se zaměřuje na postupné odstavování emisních zdrojů jako jsou především uhelné elektrárny a dle klasifikace některých zemí i jaderné elektrárny. Rušení těchto zdrojů bývá nahrazeno zdroji klasifikovanými jako obnovitelné. Především myslíme fotovoltaické a větrné elektrárny. Tyto zdroje sebou přináší nejen pozitiva, ale i negativa. Hlavním negativem, které nás bude zajímat je nestálost zdroje. Slunce nesvítlí po celou dobu dne, vítr rovněž vždy nefouká. Intenzita výroby kolísá. V kontrastu s obnovitelnými zdroji vystupují emisní zdroje, které naopak vynikají vlastností stálosti výroby. Při zavádění obnovitelných zdrojů se otázka vyrovnání výroby a spotřeby stane důležitější než jsme doposud zvyklí. Při této transformaci se dočasně akceptuje využití plynových elektráren, které mají rovněž vlastnost stálosti, ale stále je to fosilní zdroj, který má být tolerován jen po přechodné období.

Vyvážení výroby a spotřeby můžeme zjednodušit na daný okamžik, který je vyjádřen stabilitou přenosové sítě. Hlavním ukazatelem vyváženosti je frekvence v síti, kde rovnovážný stav v síti pro Českou republiku je 50 Hz. Výroba probíhá na točivém zdroji. Jakmile rychlost otáčení klesá, nejsme

schopni plně uspokojit spotřebu vzhledem k výrobě a dochází ke zpomalování točivého stroje. Tento projev jsme schopni odhalit právě pomocí sledování frekvence. Naopak pokud frekvence stoupá od referenční hodnoty, převažuje nám výroba. O stabilitu sítě v České republice se stará organizace ČEPS. Zjednodušeně řešeno její úlohou v oblasti stabilizace je sledování stavu odchylky sítě a dle toho dávají pokyn k jednotlivým výrobním zdrojům, jaký výkon je třeba dodávat. Pro tyto účely jsou vyhrazeny výrobní rezervy, k jejich použití dochází v rámci podpůrných služeb. Více o podpůrných službách si řekneme později v kapitole 5.1.

Otázkou současné doby není mít pouze výrobní zdroje na pokrytí výroby silové elektřiny, ale rovněž dostatek vhodných zdrojů pro podpůrné služby. Dle současného pojetí se na stabilizaci v síti významnou měrou podílí právě uhelné elektrárny. Postupným uzavíráním těchto zdrojů budeme přicházet o jejich kapacity, úkolem pro budoucnost bude hledání nových zdrojů, které je v tomto úkolu nahradí. Jednou z vizí je oddělení výroby silové elektřiny od té regulační. V tomto konceptu pro potřeby dodávek regulační energie v rámci podpůrných služeb by mohl posloužit hybridní zdroj, jehož modelováním se tato práce bude zabývat. Pojmem hybridní zdroj je myšleno výrobní zařízení, které využívá více vektorů výroby energie. Typicky se setkáme s jedním zdrojem výroby energie a bateriovým úložištěm. Ve zmíněném konceptu jako zdroj výroby elektrické energie je využito spalovacích turbín, bateriový systém slouží pro vylepšení vlastností celkového zdroje. V souvislosti s tímto pojmem okrajem zmíníme i pojem virtuální elektrárna. Toto řešení představuje agregaci více nezávislých kapacit zdrojů do jednoho na vnějšek homogenního zdroje. Cílem našeho zájmu bude hybridní zdroj, který může v některých aspektech přinést výhody, ale rovněž přináší nevýhody. Tou hlavní je, že více různých vektorů výroby si bude žádat koordinaci při jejich spolupráci – každý vektor může mít jinou dostupnost, limity atp., což zvyšuje nároky na řízení takového zdroje. Vytvoření nástrojů pro studium a rozvoj těchto zdrojů (zejména jejich řízení) bude výhodou.

V následující kapitole se podíváme na existující koncepty hybridních zdrojů v zahraničí, kde řešení této otázky má určitý náskok. Rovněž provedeme šetření o využití digitálních dvojčat.

Kapitola 3

Existující řešení

V předchozí kapitole jsme se seznámili se současnými výzvami a některými pojmy v oblasti energetiky, nyní se zaměříme na vybraná související řešení. V první části se zaměříme na koncepty a přístupy k podpůrným službám, provedeme ukázkou některých hybridních zdrojů. V druhé části se podíváme na pojem digitální dvojče, zmíníme možné výhody zavedení této technologie a pokusíme se ukázat příklady některých prací v naší oblasti zájmu.

3.1 Koncepty pro plnění podpůrných služeb

V této části se zaměříme na existující vědecké práce související s poskytováním podpůrných služeb. Na problematiku se podíváme z několika úhlu pohledu. Provedeme ilustraci možných konceptů pro realizaci podpůrných služeb. Dotkneme se možných výzev v řízení některých zdrojů a na závěr zkusíme shrnout některé vyplývající obecné principy.

Již jsme zmínili, že v tradičním pojetí energetiky, jsou podpůrné služby realizovány především na točivých zdrojích. V České republice se především využívají uhelné či paroplynové elektrárny a vodní. Z těchto vyjmenovaných největším problémem z hlediska klimatických závazků jsou uhelné elektrárny, které bude nutné v příštích letech uzavřít. V jiné situaci je například Švédsko, kde hlavní skladbu výroby elektřiny tvoří vodní a jaderné elektrárny. Otázka podpůrných služeb je převážně řešena pomocí plynových turbín. V rámci magisterské práce[4] zkoumali úvahu, zda lze nahradit plynovou turbínou pomocí bateriového systému a napomocť tak k dosažení nulových emisí dle národních cílů (do roku 2045 - jak zmiňuje práce). Dle jejich výsledků lze z 97% případů turbínu nahradit. Na trhu bateriových systémů v současné době dominují Li-on technologie, nicméně autoři se zabývají i zkoumáním možnosti použití vanadových baterií (Vanad redox flow), které jsou levnější, odolnější a mají nižší míru degradace. V obou případech dochází k závěru, že bateriový systém je schopen zastat funkci podpůrných služeb a uvádí tvrzení, že baterie budou velkým hráčem proměny v této oblasti. S podobným závěrem přichází práce[5] zabývající se situací ve Spojeném království, kde uvádí fakt, že v uplynulých letech rostl podíl bateriových systémů na vykrývání nedostatků energie a očekává se, že tento trend bude pokračovat (6.04% - 2016, 13.5% - 2020 a odhad 29.1% - 2035).

Téma využití bateriových systémů je populární a zabývá se jím řada prací z různých úhlů pohledu. Jedním z nich je životnost a návratnost investice. Následující článek[6] pohlíží na baterii jako na vyčerpatelný zdroj. Její pořízení je investicí, která by se měla postupným využitím splatit a na to baterie nabízí omezený počet cyklů. Při použití baterie na podpůrné služby, kde dodaná MWh se obchoduje za vyšší ceny než silová elektřina stejně nemusí dojít k pokrytí investice. V jejich studovaném případě vyšlo, že pořizovací hodnota se z provozu nezaplátí. Ilustrační příklad vycházel z obchodních dat z roku 2012 na americkém trhu. Autoři uvádí, že zařízení bylo málo využíváno. V potaz můžeme vzít fakt, že od té doby se mohl změnit poměr ceny z prodeje jedné MWh a nákladů na zbudování zdroje. Pro každý konkrétní případ je nutné si udělat rozvahu. Obecným přínosem této práce je myšlenka, že se musíme rozmyslet, jak se zdrojem zacházet, aby byl výdělečný. To může například znamenat, co nejvíce využít jednoho cyklu baterie. V některých případech pořizovací cena baterie nemusíme být tak vysoká. V článku[7] se zabývají využitím již použitých baterií z elektromobilů. Uvádí, že baterie, která je u elektromobilu prohlášena za neprovozuschopnou má hodnotu životnosti někde okolo 80%. Z těchto baterií vyzkoušeli zbudovat bateriové úložiště propojené s plynovou elektrárnou do hybridního zdroje. Prokázali, že baterie napomáhají s efektivnějším využitím plynových kogeneračních elektráren. Zařízení může pracovat více "schodovitě", a drobnější výkyvy jsou poté pokrývány baterií. Můžeme tak například využít efektivnějšího pásma výrobního zdroje z hlediska výkonu a spotřeby, nebo stálosti výrobního výkonu, kde klesá spotřeba. Využití bateriového zdroje a zisk z něho plynoucí se dá vylepšit, pokud bychom znali budoucnost a věděli jaká situace bude na aukčním trhu s regulační energií. V článku[8] představují přístup pomocí strojového učení, které poskytne pravděpodobností ohodnocení, zda zdroj bude přijat v obchodní nabídce na poskytování podpůrné služby s určitou hodnotou příjmu po danou dobu.

Bateriový systém dokáže vylepšit vlastnosti obnovitelných zdrojů, které většinou trpí na nestálost. Pokud jsou přímo napojeny do sítě bez nějakého řízení, paradoxně zvyšují nároky na stabilizaci v síti. Jedním z příkladů jsou větrné elektrárny. Základní scénářem je výroba silové elektřiny, článek[9] se koncepčně zajímá, zda větrná elektrárna může působit i na poli podpůrných služeb. Bez použití baterie lze snižovat výkon oproti svému možnému maximu pouze v době, kdy fouká vítr. Při použití bateriového systému se zařízení zvýší schopnost regulovat svůj výkon oběma směry a může tak lépe zapůsobit na trhu s regulační energií a svým vlastníkům tak zvýšit výnos. Návratnost investice však záleží dle vývoje cen na trhu. S podobnou konfigurací zdrojů pracuje jiný článek[10], zabývá se přímo řízením větrných elektráren v kombinaci s bateriovým systémem. Je zde řešena problematika unit commitmentu (nasazení strojů). Podařilo se jim vyvinout řídicí strategii, která umožňuje poskytovat služby. Při vývoji se snažili navrhnout řízení tak, aby nebylo třeba příliš velké baterie a řízení bylo šetrné na její životnost. Řízení bylo testováno na simulační platformě MATLAB Simulink. Jiným konceptem řízení se zabývá článek[11], kde máme skupinu baterií, každá o jiné kapacitě a možném výkonu.

Vytvořený model umožňuje vzhledem ke známým požadavkům rozložit nabíjení/vybíjení baterií tak, aby byl poskytnut plný potenciál všech zapojených baterií. Rozvrhování probíhá po jednotlivých hodinách, algoritmus se snaží zvyšovat spolehlivost a dostupný využitelný výkon. Nyní se přesuneme od globálních systémů a přejdeme k lokálnímu ostrovnímu systému / mikro síti[12] (systém, který veškerou svou spotřebu musí pokrýt sám, neuvažuje se připojení na síť). V této již starší práci autoři uvažují fotovoltaiku, bateriový systém, 3 plynové mikroturbíny. Předpokládají 24 hodinový výhled výroby fotovoltaiky a spotřeby v síti. Cílem optimalizace je minimalizace produkce CO_2 , čili minimalizovat běh plynových turbín a maximalizovat využití fotovoltaiky. Představený algoritmus vytváří plán unit commitmentu s horizontem na 24 hodin. Řešení probíhá pomocí dynamického programování.

Bateriové systémy nejsou jedinou technologií schopnou ukládat přebytky energie. Dalším možným směrem je ukládání energie do vodíku. Vodík lze vyrobit z vody pomocí elektrolyzérů poháněném elektrickou energií, ukládání probíhá v tlakových nádobách, v případě potřeby lze pak vodík spálit v tomu přizpůsobené turbíně. Konkrétnější popis nalezneme v článku[13]. Dobrou zprávou je, že i tento koncept lze využít k dodávce elektřiny v rámci podpůrných služeb. Vyrobený vodík se může stát palivem pro auta na vodík se spalovacím motorem[14]. Vodík lze rovněž přimíchávat do zemního plynu v plynové soustavě (Power-to-gas). V Německu řešili otázku, zda je možné dlouhodobě skladovat přebytky elektrické energie z obnovitelných zdrojů pomocí této technologie[15]. Jako modelový scénář si vybrali data z rekordního dne, kdy podíl z celodenní produkce elektrické energie na území Německa z 85% zastávali obnovitelné zdroje. Autoři došli k závěru, že tato technologie a plynová soustava je schopna pojmout významné přebytky energie. Muselo by se však vhodně navrhnout umístění infrastruktury vzhledem ke kapacitám plynovodů. Dle jejich analýzy přebytků a nedostatků energie v rámci spolkových republik by se plnicí stanice museli nacházet zejména v severní části země.

Na závěr této části shrneme dvě myšlenky, energetická úložiště (baterie, vodík, Power-to-gas) dokážou pomoci v oblasti podpůrných služeb, napomáhají rozkládat aktuální nedostatek / přebytek energie v čase pomocí jiného vektoru zdroje. Pokud k jednomu zdroji elektrické energie přidáme úložný systém, vznikne nám hybridní zdroj, který vylepšuje vlastnosti samotného původního zdroje, nicméně se zvyšují nároky na řízení.

3.2 Využití digitálních dvojčat

V úvodní části zmíníme článek[16], který sice pojednává o digitálních dvojčatech (digital twin - DT) z oblasti ropného a plynárenského průmyslu, nicméně některé myšlenky a informace lze převzít v obecné rovině. V rámci práce provedli analýzu odborných prací dotýkající se tématu digitálních dvojčat v jejich sledovaném odvětví. Z jejich průzkumu je vidět, že počet prací se ve dvou předcházejících letech (2018, 2019) rapidně zvýšil před publikací výsledků v roce 2020. Autoři uvádí, že ropný a plynárenský průmysl zaostává

za jinými oblastmi jako například letectví, automobilový průmysl, zdravotnictví a maloobchod. V těchto odvětvích se již prokázaly výhody zavádění technologie DT. Většina publikací související s DT pochází již z průmyslového prostředí než z akademického prostředí. Je tedy vidět, že v ropném a plynárenském průmyslu započal intenzivní zájem o tuto technologii. Digitální dvojčce je v současnosti jednou ze standardních metod pro zlepšení návrhu, testování a údržby systému pomocí modelování a simulace[17]. První simulačně založená řešení byly vyvíjeny především pro potřeby optimalizace a predikce chyb. Finální podoba technologie DT pak vznikla postupnou integrací se senzorovou sítí, digitalizovanými výrobními prostředky a systémy produkce ve zpracovatelském průmyslu. Hlavní rozdílem mezi simulací a digitálním dvojčtem je opravdové propojení s prostředky zařízení a senzorovou sítí. DT mají svůj přínos v monitorování, analýze dat, pochopení a optimalizaci procesů fyzických prostředků, možnost vyzkoušení scénářů "co kdyby", ať z důvodů zkoumání možných chyb nebo nových procesů před zavedením do fyzické infrastruktury. Autoři poznamenávají, že technologie DT v roce 2018 figuruje v žebříčku 10 strategických technologických trendů a předpovídají enormní nárůst zájmu v příštích letech. Při budování nových projektů je dobré mít na paměti možnost zapojení DT hned na začátku plánování nového zařízení[16]. Následně je výhodné udržet kontinuální propojení fyzické infrastruktury s tou virtuální digitální dvojčce po celou dobu životního cyklu zařízení. Možnou nevýhodou tohoto spojení je především vyšší možnost kybernetické zranitelnosti, které je vystavena fyzická infrastruktura. Díky spojení mohou vzniknout vrátka do jinak odděleného systému fyzické infrastruktury skrze zranitelnosti přímo plynoucí z propojení nebo virtuální části digitálního dvojčce. Kdyby spojení neexistovalo, útočník by vůbec nemusel mít takovou možnost se k jinak izolovanému fyzickému systému dostat. Záleží však na konkrétním případě – využití technologie, nastavená úroveň zabezpečení. Nicméně není radno tento aspekt podceňovat. Stačí manipulovat data ze senzoru, na jehož měření závisí úspěch určitého procesu. Při reakci na data, která nejsou pravdivá, nebo naopak nereakci, kdy k ní mělo správně nastat, může dojít lehce k nehodě s nepříznivými následky.

Nyní se zaměříme přímo na oblast energetiky, kde představíme několik prací týkajících se simulací a digitálních dvojčat. V následující práci ilustrují model vytvářen jako digitální dvojčce[18]. Předmětem modelování je výrobní blok tepelné elektrárny o výkonu 320 MW. Řešení je vytvořeno v nástroji ThermoflowTM. Na vzniklém modelu mohli ověřit optimalizační řešení zaměřující se na snížení spotřeby uhlí. Testování provedli na datech z letního i zimního provozu a jejich optimalizační řešení by mohlo zlepšit provozní účinnost bloku. Další článek[19] se zabývá modelováním dieselového generátoru kombinovaného s bateriovým systémem na plovoucí přímořské elektrárně. Využívají hybridní model, kde matematickou aproximací napodobují chování reálného systému. Jejich předmětem zájmu byly především produkované emise, které se ve virtuálním modelu přibližují reálným číslům. Toto není přímý příklad digitálního dvojčete spíše simulace, nicméně je to jeden z dílčích kroků - modelování nějaké fyzické technologie. Hybridním modelováním se zabývá další

publikace[20]. Uvádí, že chybí vysoce přesné modelování tepelných systémů elektráren. Zaměřili se na tuto problematiku, z fyzikálních mechanismů a provozních dat se snažili vytvořit různé komponenty jako framework. Vyvinuté komponenty poslouží jako nezbytná podpora pro vývoj digitálního dvojčete tepelných elektráren. Dále postoupili v článku[21], řešili model hybridního zdroje, kde figuroval diesellový generátor a bateriový systém. Vytvořili model spalovacího motoru, baterie v nástroji MATLAB Simulink, rovněž vyvinuli řídicí systém zdroje. Virtuální systém zkoušeli provozovat v reálném čase, problémem byla odezva komponenty spalovacího motoru díky výpočetně náročnému modelu. Ten museli zjednodušit, aby stíhal v přijatelném čase reagovat.

Příbuznou oblastí, kde se pojem digitální dvojče objevuje je téma mikro sítí - decentralizovaných sítí. Lokální energetika, distribuované zdroje, jsou nyní rovněž aktuální témata. Z těchto oblastí uvedeme dva články, které se problematikou zabývají, jen jako referenci, protože už nejsou přímým předmětem našeho zájmu a necháme čtenáři pro případné vlastní prostudování[22, 23]. Příklady digitálních dvojčat v pravém slova smyslu se příliš nalézt nepodařilo, z uvedených prací se spíše jednalo o otázku modelování komponent, nebo jen simulací systému. Digitální dvojče je pojem aktuální, objevující se v posledních letech a využití této technologie bude růst na významu, jak vyplývá z některých publikací.

Kapitola 4

Přínos práce

Z prezentovaných článků vyplynulo, že myšlenka hybridního zdroje není chybná. Bateriový systém v kombinaci s výrobním zdrojem může být řešením pro realizaci podpůrných služeb. Z koncepčního úhlu pohledu má smysl se zabývat nastíněným řešením a pokračovat v jeho dalším zkoumání. Tématem digitálního dvojče se příliš článků nezabývalo, jak bychom rádi očekávali. Řešili spíše dílčí úlohy, jak přesně modelovat technologie, ze kterých se zdroj skládá. Postrádali jsme do detailu více popsanou ukázkou aplikace konceptu digitálního dvojče. Nepodařilo se nalézt, že by se někdo zabýval nám podobným zadáním. Z obou dvou zkoumaných pohledů je legitimní se nadále věnovat návrhu digitálního dvojče pro náš problém.

Výhody plynoucí ze zavedení tohoto konceptu na analýzu chování zdroje v souvislosti s vývojem řízení jsou jasné.

- Šetření nákladů díky realizaci testů nad virtuální reprezentací
- Možnost realizace testů, které by na reálném zařízení nebyly možné jednoduše provést – např. nelze produkovat výkon mimo domluvené aktivace
- Možnost testovat chování různých druhů turbín, které se liší parametry – vhodné při porovnání různých nabídek turbín na trhu před zakoupením
- Možnost testovat potřebnou dimenzi baterie – analýza před nákupem technologie
- Vývoj a ladění řídicího algoritmu se známým efektem ve virtuální reprezentaci
- Odhad možných příjmů a nákladů při modelových scénářích

Některé body potvrzují, že je výhodné zavést při počátku projektu, pokud možno.

Tato práce se zabývá modelováním konkrétního typu hybridního zdroje, avšak komponenty technologií (turbína, baterie) jsou popsány co nejvíce parametricky, aby bylo možné jejich případné další využití na budoucí příležitosti, které nyní neznáme. Klíčové v této práci je návrh způsobu napojení různých druhů řízení od různých autorů spřízněných s výzkumem.

Kapitola 5

Uvažovaný hybridní zdroj

V rámci této kapitoly čtenáře seznámíme se souvislostmi pojící se hybridnímu zdroji. V první části si povíme něco podrobnějšího o podpurných službách, aby čtenář mimo tento obor získal základní představu. Rovněž uvedeme, jakým způsobem služby budeme realizovat. Druhá část se bude zabývat konceptem hybridního zdroje, jehož činnost se simulátor v rámci této práce bude snažit simulovat.

5.1 Podpurné služby

Jak již víme, podpurné služby jsou prostředkem pro stabilizaci elektrické sítě. Aby byla síť v rovnováze, v každém okamžiku bychom potřebovali vyvážit produkci a spotřebu elektrické energie. Faktem je, že síť je permanentně rozkolísaná, jednoduchou ilustrací jsou například situace - rozjíždějící elektrický vlak či tramvaj, v domácnosti zapínáme rychlovarnou konvici, či jsme si přivolali výtah v kancelářské budově. Kolísání nemusí být vytvářeno jen na straně spotřeby. Vzpomeňme případ, kdy sluníčko se dostane zpoza mraku a plný svit dopadne na fotovoltaickou elektrárnu - náhle se zvýší produkce. Pro sledování stavu sítě byla zvolena fyzikální veličina frekvence. Elektrická energie se vyrábí na točivých strojích (pomineme fotovoltaické zdroje), z této technologie vzešel střídavý proud. Pokud spotřeba energie je vyšší než její dodávka, snižuje se rychlost cyklů vlny tj. frekvence, dochází ke snížení napětí. To je rizikem pro spotřebiče, které vlivem špatné kvality napájení mohou špatně fungovat či dokonce může dojít k rozbití zařízení. Jednoduše můžeme stejnou situaci aplikovat obráceně. Pokud převažuje výroba nad spotřebou, frekvence v síti se bude zvyšovat.

V České republice nad rovnováhou přenosové soustavy dohlíží společnost ČEPS, a.s.[24], která vystupuje v roli operátora a regulátora přenosové soustavy. Pokud dochází k nerovnováze v síti, především zasahuje na výrobní straně. Může však zasáhnout i na straně spotřeby – pomocí flexibilních prostředků spotřeby. Snaha o udržení rovnováhy začíná dopředu již odhadem spotřeby a dle toho se připravuje diagramový výkon, podle kterého jednotlivé elektrárny (výrobní bloky) mají do sítě dodávat výkon. Samozřejmě nám bude vznikat odchylka mezi skutečnou spotřebou a tímto plánem výroby. Neexistují prediktory, které by přesně určili jaký skutečný výkon bude potřeba dodávat

v budoucnu. Proto musí být připraven a bude využit zálohový výkon, který je poskytován k aktivaci operátoru sítě (ČEPS) od jednotlivých poskytovatelů. Tato záloha je poskytována v několika formách - nazýváme je jako podpůrné služby, kde každá je popsána danými charakteristickými vlastnostmi. Přesná pravidla definuje Kodex přenosové soustavy - podpůrné služby[25]. Pokud je poskytovatel certifikován k nabídce a poskytování těchto služeb, při aktivaci se zavazuje dodat požadovaný výkon, pokud ho není schopen kvalitativně poskytnout, za nesplnění je pokutován. Spolu s využitím těchto rezerv je udržována rovnováha v síti.

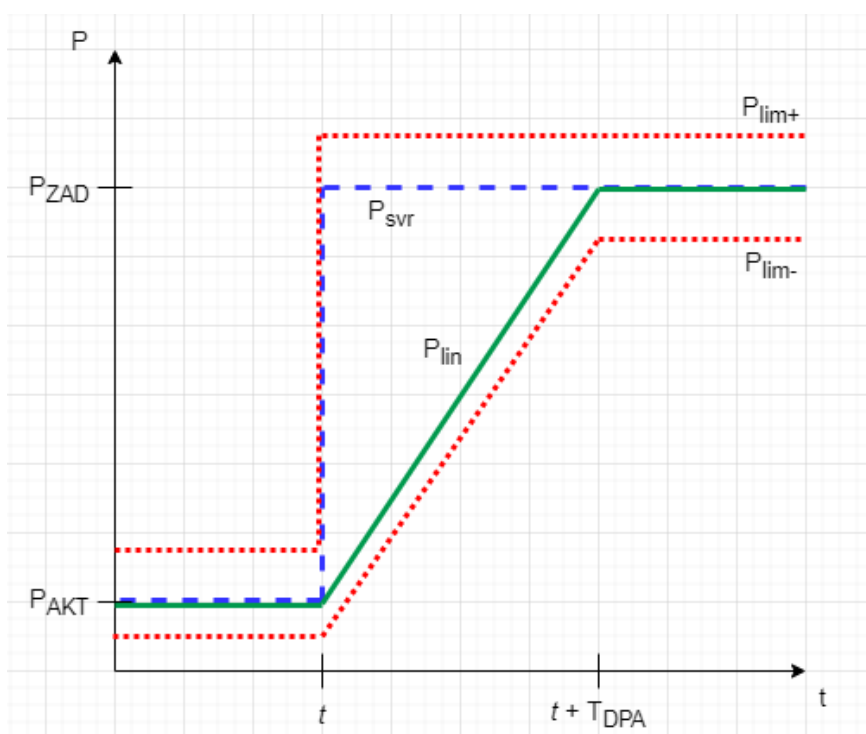
Podpůrné služby se dělí na dvě základní kategorie - služby výkonové rovnováhy (SVR) a ostatní podpůrné služby. Naším předmětem zájmu bude pouze první kategorie (druhá se zabývá obnovením dodávek, např. blackoutu atp.). V kategorii SVR nalezneme 4 druhy služeb - Zálohy pro automatickou regulaci frekvence (FCR), zálohy pro regulaci výkonové rovnováhy s automatickou regulací (aFRR), zálohy pro regulaci výkonové rovnováhy s manuální aktivací (mFRR), zálohy pro náhradu (RR). Z pohledu výrobního bloku můžeme vyjádřit zjednodušenou rovnicí 5.1, která vyjadřuje, jak je tvořena aktuální hodnota skutečného výkonu P_t^{SKUT} dodávaného do sítě v čase t .

$$P_t^{SKUT} = P_t^{DG} + P_t^{FCR} + P_t^{aFRR} + P_t^{mFRR} + P_t^{RR} \quad (5.1)$$

P_t^{DG} odpovídá plánovanému diagramovému výkonu v čase t , P_t^{FCR} vyjadřuje dodávaný výkon službou FCR v čase t , P_t^{aFRR} vyjadřuje dodávaný výkon službou aFRR v čase t , P_t^{mFRR} vyjadřuje dodávaný výkon službou mFRR v čase t , P_t^{RR} vyjadřuje dodávaný výkon službou RR v čase t . Výstupní výkon výrobního bloku je tedy dán součtem všech zmíněných složek. Ohledně podpůrných služeb si ještě zavedeme klasifikaci na kladnou a zápornou službu. Jako kladnou službu (regulační energii) označíme případ, kdy dodáváme do sítě výkon nad rámec plánovaného výkonu (vzniká nám kladné saldo oproti plánu). Jako zápornou službu (regulační energii) označíme případ, kdy odebíráme ze sítě více než je plánovaný výkon (vzniká nám záporné saldo oproti plánu). Jednotlivé služby si nadále konkrétněji představíme v rozsahu nezbytných informací pro tuto práci, pro další podrobnosti čtenáře případně odkážeme na kapitulu o podpůrných službách z diplomové práce[26] či přímo na kodex[25]. Napřed ještě uvedeme, jak povel na službu chápat a jak v této práci budeme průběh výkonu realizovat.

5.1.1 Interpretace a realizace povelu

Již jsme uvedli, že společnost ČEPS má nasmlouvané poskytovatele připravené k aktivaci služeb. Tato aktivace probíhá pomocí povelu. Povel především obsahuje informaci o žádaném výkonu. Zároveň dle typu služby je určena doba do plné aktivace, po kterou má dojít k dosažení cílového žádaného výkonu od vydání povelu. Na obrázku 5.1 si ukážeme, jak povel na žádaný výkon může být realizován. Výchozím stavem naší obecně znázorněné služby je, že dodáváme výkon v rámci služby o hodnotě P_{AKT} . V čase t obdržíme povel na žádanou hodnotu P_{ZAD} s dobou plné aktivace T_{DPA} . Tento povel



Obrázek 5.1: Ukázka realizace povelu na službu výkonem v čase

můžeme pohledem diagramového plánování v čase vyjádřit jako okamžitý nárůst výkonu na žádanou hodnotu, jak je znázorněno průběhem P_{svr} (modrá čárkovaná křivka). Skokový nárůst z technologických důvodů není možné realizovat na většině výrobních blocích např. uhelné elektrárny, proto je dovoleno skokový nárůst rozložit v čase a výkonem se pohybovat po křivce určenou lineární funkcí mezi aktuálním a požadovaným výkonem po dobu do plné aktivace, znázorněno průběhem P_{lin} (zelená křivka). Jelikož pro realizaci je dovolena určitá odchylka, je zavedeno pásmo, ve kterém se můžeme výkonem v rámci podpůrné služby pohybovat. Toto pásmo je shora ohraničeno průběhem P_{lim+} a zdola omezeno průběhem P_{lim-} (červené tečkované křivky). Pokud je výkon, zjednodušeně řečeno, realizován mimo toto pásmo, jedná se o porušení kvalitativních požadavků a poskytovatel je pokutován za nesplnění. V rámci této práce pro realizaci povelu na podpůrnou službu, uvažujeme plnění pomocí lineárního průběhu P_{lin} .

5.1.2 FCR

Tato služba je prvotní, která reaguje na nerovnováhu v síti. FCR (zálohy pro automatickou regulaci frekvence) je lokální automatickou funkcí, která je spojena s aktuální odchylkou frekvence. Výrobní blok se sám proaktivně snaží eliminovat aktuální odchylku. Míra výkonu, který se v rámci této služby má uvolňovat/zadržovat se určuje proporcionálním regulátorem ve zpětné vazbě. Maximální hodnota výkonu určená pro FCR je dosažena v případě, že

odchylka frekvence dosáhne hodnoty 200 mHz, což je maximální povolená odchylka v síti. Vyšší hodnota odchylky není uvažována, protože hrozí přetížení sítě. Určení aktuální požadované hodnoty služby FCR nemusí poskytovatel určovat sám, může se dohodnout s regulátorem a nechat si zasílat povely se žádaným výkonem pro tuto službu, stejně jako probíhá u následujících služeb. Doba do plné aktivace pro FCR by měla být okamžitá, z hlediska tolerancí můžeme říci až do 30 sekund.

■ 5.1.3 aFRR

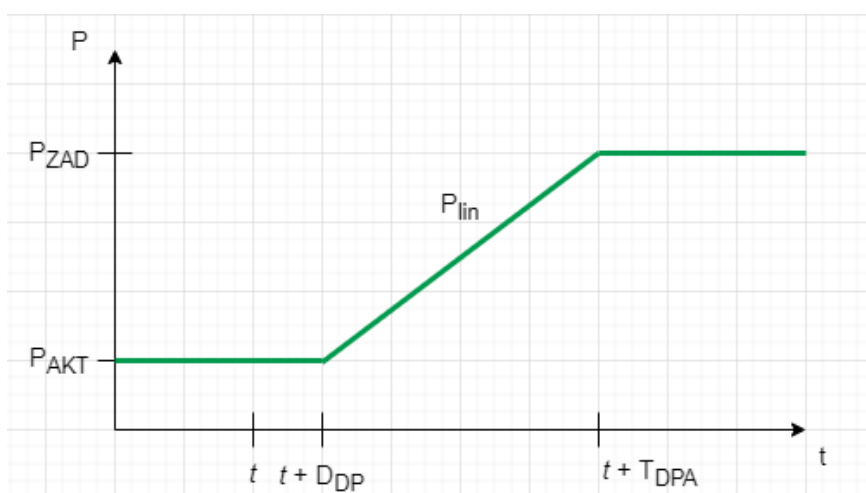
Předcházející služba je navržena pro pokrytí menších odchylek, jako další regulační prvek se zapojuje služba aFRR (zálohy pro regulaci výkonové rovnováhy s automatickou regulací). Tato služba je aktivována na základě požadavku regulátora sítě, který poskytovateli zašle hodnotu žadaného výkonu této služby. Povely jsou vytvářeny automaticky na základě algoritmu regulátoru dispečinku ČEPS. Co se týče doby do plné aktivace, v současné době je stanovena na 7,5 minuty. Do konce roku 2024 je plánováno zkrácení této doby na 5 minut. Pokud bude potřeba nadále v této práci pracovat s touto hodnotou, budeme uvažovat již nově plánovanou hodnotu.

■ 5.1.4 mFRR

Další službou v řetězci je mFRR (zálohy pro regulaci výkonové rovnováhy s manuální aktivací). Ta je prvním neautomatizovaným prvkem regulace, o její aktivaci rozhoduje dispečer na dispečinku. Existuje ve dvou podobách $mFRR_{12,5}$ a $mFRR_5$. U $mFRR_{12,5}$ doba plné aktivace je stanovena na 12,5 minuty s odlišností oproti předchozím službám, doba se dělí na dobu přípravy 2,5 minuty a dobu rampování 10 minut. Ilustrace dob je znázorněna na obrázku 5.2, kde doba příchodu povelu je v čase t . Do uplynutí doby na přípravu $t + T_{DP}$ se žadaný výkon nemění (výchozí hodnota P_{AKT}), po této době probíhá rampování na žadaný výkon P_{ZAD} do okamžiku $t + T_{DPA}$. U této služby jsou toleranční pásma definována jinak než jsme ukázali na obrázku 5.1 popisující obecný průběh. "Trojúhelník" tolerančního pásma se nachází pod průběhem lineární rampy, s dodávkou se můžeme spíše zpozdít. Nicméně vyhodnocení realizovaného výkonu v rámci tolerance není předmětem této práce, pro podrobnější informace čtenáře odkážeme na kodex[25]. Služba $mFRR_5$ má doba plné aktivace kratší a to 5 minut. Neobsahuje žádný čas na přípravu, rampování začíná okamžitě. Tento produkt je koncipován na vodní elektrárny, které nemají problém uvolnit výkon téměř okamžitě.

■ 5.1.5 RR

RR (zálohy pro náhradu) jsou realizované poskytnutím sjednané regulační energie RE. Je aktivována příkazem z dispečinku provozovatele sítě s dobou do plné aktivace 30 minut. Průběh služby se podobá mFRR, na rozdíl od mFRR jsou zde jiná pravidla povinnosti tuto službu poskytnout. Tuto služba



Obrázek 5.2: Ukázka průběhu výkonu na základě povelu $mFRR_{12,5}$ v čase t

zde uvádíme pro ucelený přehled, RR nebyla v této práci podrobně studována a následující text s ní nepracuje.

5.1.6 Shrnutí podpůrných služeb

V této kapitole jsme si uvedli nezbytné informace týkající se podpůrných služeb. Víme, že existují různé druhy služeb, které se využívají za určitých okolností, v procesu regulace mají svou návaznost, z technického pohledu průběhu výkonu se liší především dobou plné aktivity, případným odsunutím rampování dobou přípravy. Způsob vyhodnocení plnění se pro každou službu může lišit. Realizaci výkonu v této práci budeme uvažovat po lineárním průběhu mezi aktuálním a cílovým výkonem po dobu rampování. Na závěr uvádíme tabulku 5.1 pro shrnutí služeb a jejich dobu aktivity.

Zkratka	Název	Doba plné aktivity
FCR	zálohy pro automatickou regulaci frekvence	Okamžitá
aFRR	zálohy pro regulaci výkonové rovnováhy s automatickou regulací	5 minut (7,5 minuty do roku 2024)
mFRR	zálohy pro regulaci výkonové rovnováhy s manuální aktivací	12,5 minuty
RR	zálohy pro náhradu	30 minut

Tabulka 5.1: Přehled podpůrných služeb

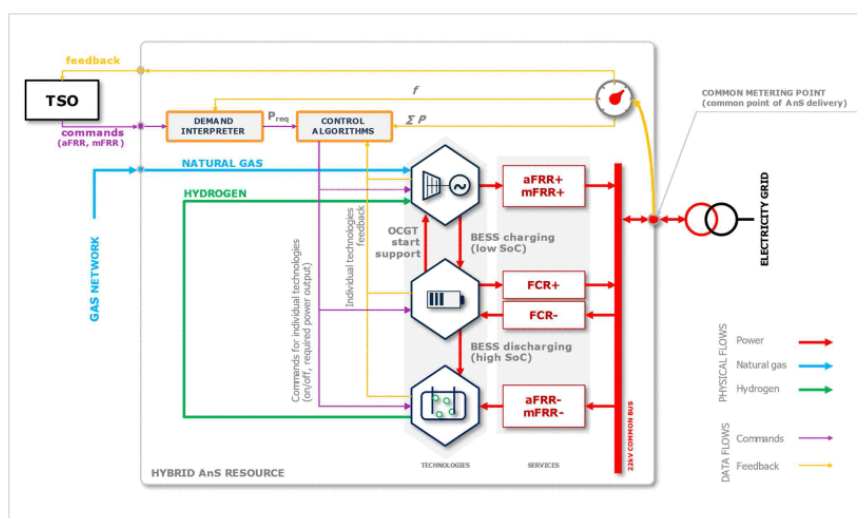
5.2 Koncepce zařízení

Cílem této práce je vytvoření simulátoru pro hybridní zdroj. Jako modelový příklad pro vznik tohoto digitálního dvojče byl využit koncept hybridního

zdroje z výzkumného projektu. Tento projekt se snaží pomoci v úsilí o posílení energetické bezpečnosti a dekarbonizace ČR. Bude nutné vytvořit nové zdroje energie, které nahradí v budoucnu odstavované fosilní zdroje, zejména u nás uhelné elektrárny, které mají kromě výroby silové elektřiny rovněž důležitý podíl na poskytování SVR[1].

Řešení hybridního zdroje předpokládá složení z velkokapacitního bateriového úložiště a několika spalovacích turbín s relativně rychlým startem - řádově desítky až stovky sekund, vypínací sekvence trvá o poznání déle - několikanásobně delší než startovací sekvence. Bateriový systém je připraven k poskytování energie okamžitě avšak s omezeným výkonem a kapacitou, spalovací stroje mají čas dostupnosti k dodávce tedy o poznání horší, nicméně jsou pak připraveny na větší a déletrvající požadavky. Dochází tak ke kooperaci dvou zdrojů s odlišnými vlastnostmi. Při malých požadavcích může být využito pouze zdroje bateriového systému. Při větších požadavcích dochází k součinnosti bateriového systému, který zpočátku přebírá plnění služby do doby než je zátěž převzata nastartovaným spalovacím strojem. Zároveň spalovací turbíny slouží k dobíjení bateriového systému v případě potřeby.

Zařízení tohoto typu je zaměřeno pouze na poskytování SVR a zapadá do konceptu snahy oddělení výroby silové elektřiny a poskytování podpurných služeb. Výroba silové energie může probíhat na ekologicky přijatelnějších obnovitelných zdrojích. Tyto zdroje mají slabší potenciál v oblasti podpurných služeb, tyto nedostatky může kompenzovat právě tento hybridní zdroj. Bylo by možné namítnout, že energie z tohoto zařízení není stále nízkoemisní, ale na rozdíl od konvenčních (uhelných) zdrojů, které jsou v provozu nepřetržitě, toto zařízení je udržováno v pohotovostním režimu s minimální emisní stopou. Za cenu vyšších emisí je provozováno jen v případě potřeby služeb SVR. Z celkového podílu výroby regulační energie tvoří velice malý podíl, je tedy možné dosáhnout celkově nižších emisí oproti současné koncepci uhelných zdrojů.



Obrázek 5.3: Koncept hybridního zdroje pro poskytování SVR[1]

Zmíněný zdroj je zamýšlen k poskytování podpůrných služeb - kladné FCR+, záporné FCR- a kladných aFRR+, mFRR+. Strukturu a schopnosti zdroje si můžeme graficky představit dle obrázku 5.3. Zde máme několik komponent - TSO (Transmission system operator) - operátora přenosové soustavy, který vydává pokyny na služby. Další komponenta, která je již součástí zařízení je demand interpreter, který provádí interpretaci požadavků pro řídicí algoritmus. Z technologické části zde máme vyobrazen zdroj představující spalovací turbíny spotřebovávající plyn, bateriový zdroj a navíc hydrolyzér, který umí přeměňovat přebytečnou energii na vodík, který může být spalován pomocí turbín. Tímto rozšířením je zařízení schopné poskytovat i záporné služby aFRR- a mFRR-. Zapojení technologie vodíku do zmíněného zdroje nebylo prozatím více rozvíjeno, kvůli dle propočtů diskutabilní nízké účinnosti při uložení a zpětném získání energie. Dle obrázku můžeme vidět, že pro účely poskytování FCR se uvažuje především použití baterie, pro služby aFRR+, mFRR+ se uvažuje využití spalovacích turbín, nicméně mezi zdroji je zamýšlena kooperace.

Kapitola 6

Popis problému

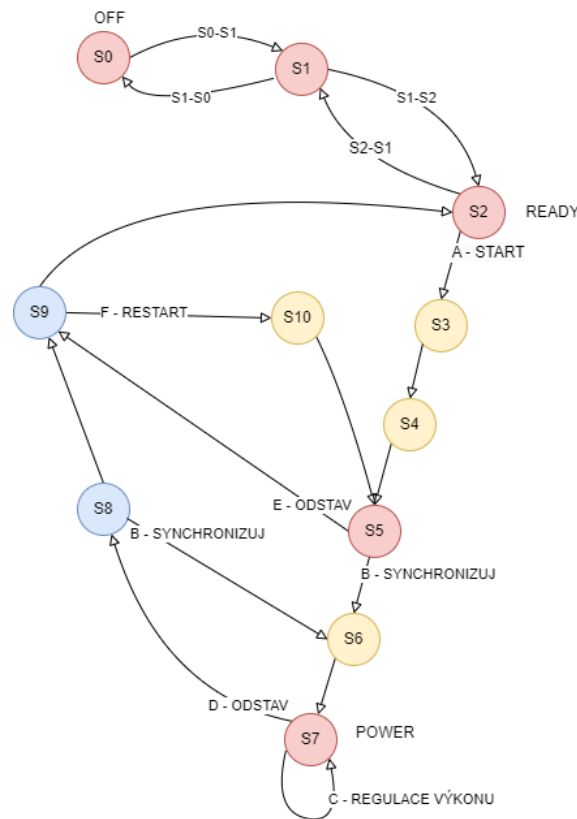
V předchozí kapitole jsme si představili nutné znalosti týkající se podpůrných služeb pro potřeby této práce a koncept hybridního zdroje, který simulátor v této práci bude schopen modelovat. Pro úplnost shrneme, že simulátor by měl být schopen modelovat technologie hybridního zdroje skládajícího se z turbín a bateriového systému. Úkolem zařízení bude realizace příchozích požadavků na dodávky regulační energie v rámci podpůrných služeb pomocí zmíněných dvou technologií. Je zřejmé, že tento úkol bude muset dělat nějaký řídicí prvek. Součástí digitálního dvojče bude tedy i rozhraní pro řídicí algoritmus, které bude poskytovat data o stavech technologií a bude možné skrze něj předávat povely pro ovládání těchto technologií. Nyní se zaměříme na detailnější popis jednotlivých komponent včetně jejich vlastností. Cílem je flexibilní návrh, proto konkrétní hodnoty budou uvedeny minimálně a pokusíme se vše co nejvíce zavést obecně pomocí parametrů.

6.1 Turbíny

Uvažované zařízení může obsahovat $0 \dots N$ identických spalovacích turbín. Turbína zadána pro tuto práci pro výrobu elektrické energie spaluje plyn a rovněž pro svůj provoz spotřebovává elektrickou energii. V rámci svého životního cyklu provozu se nachází v několika stavech. Základně můžeme určit tři významné stavy:

- zcela vypnutá
- zapnutá v režimu připraveném pro start
- výkonový stav schopný dodávat elektrický výkon

Ve vypnutém stavu turbína nespotebovává žádnou elektrickou energii, neposkytuje data o svém stavu, není možné vydávat řízením žádné pokyny. Jedná se o stav, který umožňuje turbínu zcela ze systému vyčlenit například pro potřeby údržby. Nadále se budeme setkávat s označením *OFF* stav. Manuální aktivací je turbína schopna přejít do stavu zapnuto, kdy se nachází v pohotovostním režimu připravena pro aktivaci do výkonového stavu. Alternativně lze říci, že turbínu uvedeme do provozu. V tomto stavu je turbína



Obrázek 6.2: Úplný stavový diagram turbíny

- Stav s timeoutem s možností povelu** - Stejně jako předchozí uvedený stav může mít více vstupních hran a pouze jednu výstupní označenou jako timeoutovanou. Navíc má $1 \dots N$ dalších výstupních hran určené povelu. Tyto hrany slouží pro opuštění stavu pomocí povelu před uplynutím doby timeoutu. Na obrázku 6.2 znázorněn modrou barvou.

Ve stavovém diagramu se rovněž vyskytují stavy zastupující zmíněné tři významné stavy - *OFF*, *READY* a *POWER*. U příkladu na obrázku 6.2 konkrétně stav *S0* v roli *OFF*, *S2* v roli *READY* a *S7* v roli *POWER* stavu. Na závěr k diagramu je potřeba zmínit, že je potřeba si zaznamenat ještě jeden významný povel a to umožňující regulaci výkonu. Na obrázku 6.2 se jedná o povel *C*, nad stavem *S7*. V obrázku je sice hrana explicitně uvedena, nicméně v cílovém řešení nebudeme chtít tuto hranu uvažovat jako nějaký přechod, spíše ji budeme chtít modelovat jako vnitřní chování stavu. Pro potřeby možnosti vydání tohoto povelu řízením, potřebujeme mít uloženo, jaký název pro regulační povel bude systém využívat.

Z hlediska datové reprezentace každý stav je popsán názvem stavu *NAZEV_STAVU*, spotřebou elektrické energie *EE_STAVU* a plynu *GAS_STAVU*. Popis datové struktury shrnuje tabulka 6.1.

Proměnná	Typ	Komentář k typu	Popis
ZE_STAVU	řetězec	Název stavu čerpající z proměnné $NAZEV_STAVU$.	Přechod ze stavu.
DO_STAVU	řetězec	Název stavu čerpající z proměnné $NAZEV_STAVU$.	Přechod do stavu.
$POVEL_PRECHODU$	řetězec / "TIMEOUT"	V případě zadání řetězce mimo "TIMEOUT", reaguje přechod na povel s poskytnutým řetězcem. Pokud byl zadán "TIMEOUT", přechod je vykonán po uplynutí doby specifikované doby v následujícím parametru. Zakázán je řetězec určen pro regulační povel, kterému je přiřazen speciální význam regulace výkonu v rámci jednoho stavu.	Aktivován uvedeným povellem nebo timeoutem.
$TIMEOUT_STAVU$	číslo	Využije se pokud $POVEL_PRECHODU$ má hodnotu TIMEOUT.	Případná hodnota timeoutu.

Tabulka 6.3: Datová reprezentace přechodu mezi stavy

Stavový diagram není jediný, který nám popisuje chování turbíny, zavedeme ještě parametry pro omezení charakteristiky chování ve výkonovém stavu. Turbína v tomto stavu dodává výkon, ten se musí pohybovat v určených mezích. Musíme určit minimální možný výkon pojmenovaný jako minimální technický provozní výkon P_{min} , který turbína poskytuje při vstupu do tohoto stavu. Rovněž, aby bylo možné stav opustit, výkon musí být na této hodnotě. Maximální výkon je proměnlivý dle teploty stejně jako byla spotřeba, bude muset být dopočítán v závislosti na teplotě uvažované při spuštění simulace. Hodnotu opět určíme dle nominálních výkonů na základě teploty. Příklad možné podoby zobrazuje tabulka 6.4. Z hlediska označení pro maximální výkon turbíny zavedeme P_{max} .

x	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
15°C	0	600	1200	1770	2310	2810	3250	3610	3930	4230	4430
20°C	0	570	1140	1682	2195	2670	3088	3430	3734	4019	4209
25°C	0	542	1083	1597	2085	2536	2933	3258	3547	3818	3998

Tabulka 6.4: Tabulka nominálních hodnot výkonu pro vyjádření procentem v závislosti na teplotě. Uvažovanou jednotkou jsou kW. Hodnoty jsou smyšlené.

Dodávaný výkon v čase bude možné měnit. Změna výkonu bude mít rovněž svá omezení. Budeme chtít omezení na možný maximální gradient, tj. maximální změna výkonu z vteřiny na vteřinu. Hodnota maximálního gradientu může být odlišná pro zvyšování výkonu P_{max-up}^{grad} a snižování výkonu $P_{max-down}^{grad}$. Na závěr budeme chtít parametr, který nám omezuje maximální změnu gradientu z vteřiny na vteřinu $P_{max-change}^{grad}$. Toto omezení plyne z technologických rizik, kdy při velkých náhlých změnách výkonu mohou být některé komponenty turbíny nadměrně namáhány. Zmíněné parametry shrnuje ta-

má být konkrétní požadavek obdrženo, aby digitální dvojice zařízení nevidělo do budoucna. Pro tento vstup bude využit následující formát, který byl vyžadován poskytovatelem testovacích dat. Uvažujeme strukturu vstupního souboru, kde první řádek obsahuje výchozí čas, od kterého se požadavky určují. Další časové údaje jsou zadány jen jako inkrementy k tomuto výchozímu času. Od druhého řádku dále pak každý řádek popisuje jeden požadavek. Struktura řádku obsahuje typ služby (na výběr ze zástupných jmen FCR, aFRR, mFRR) a časovou řadu popisující průběh žádaného výkonu. Průběh je popsán dvojicemi (čas - jako inkrement, výkon), kde na počátku je zadán počet bodů, kterým je průběh popsán. Po tomto čísle následuje daný počet dvojic hodnot. Podobu struktury shrnuje tabulka 6.7

Formát časové řady: <výchozí čas> <jméno> <počet> <čas1> <výkon1> <čas2> <výkon2> <čas3> <výkon3> ...		
Proměnná	Jednotka	Popis
<výchozí čas>	datetime	Výchozí čas pro časové řady
<jméno>	enum	Typ časové řady z definované množiny služeb
<počet>	int	Počet dvojic (čas, výkon)
<čas>	s	Čas požadovaného výkonu, počet sekund od <výchozí čas>
<výkon>	MW	Výkon v daném čase

Tabulka 6.7: Formát vstupního souboru pro požadavky služeb

6.4 Nastavení simulace

Další částí se bude zabývat nastavením simulačního prostředí. Nyní uvedeme parametry týkající se konfigurace modelovaného zařízení a prostředí, ve kterém se nachází. Budeme chtít, aby simulace odrážela reálný čas, proto musíme určit počátek simulace T_{start}^{sim} a její konec T_{end}^{sim} . V rámci tohoto časového okna akceptujeme požadavky na podpůrné popsání v sekci 6.3 o formátu podpůrných služeb. Již jsme uvedli, že maximální výkon turbíny a její spotřeba plynu je závislá na teplotě, proto pro simulaci zavedeme parametr $Temp_{sim}$, který nám hodnotu teploty určí. Pro účely reportování výsledků či možné rozhodování řídicích algoritmů je vhodné zavést uvažovanou cenu elektrické energie $Cost_{EE}$ za 1 MWh, rovněž cenu plynu $Cost_{GAS}$ za 1 MWh.

Pro popis konfigurace zařízení budeme potřebovat určit počet existujících turbín v simulaci, zavedeme parametr B_{count} . Zároveň pro každou turbínu budeme chtít určit její výchozí stav na počátku simulace. Zavedeme pole $B_{power}[]$, které bude obsahovat počet hodnot shodné s B_{count} . Každá hodnota udává výkon turbíny na počátku simulace. Na základě hodnoty této proměnné určíme, ve kterém stavu z významných (*OFF*, *READY*, *POWER*) bude turbína inicializována. Pokud hodnota je menší než 0, budeme chtít nastavit stav *OFF*. Pokud hodnota bude v intervalu $(0, P_{min})$, budeme chtít nastavit stav *READY*. Pro hodnoty větší nebo rovno P_{min} , turbína se bude nacházet

ve stavu *POWER* s daným výkonem. V případě, že hodnota překročí spočítanou hodnotu P_{max} , na základě počáteční teploty $Temp_{sim}$, bude nastaven výkon pouze na možné maximum. Pro bateriový systém budeme chtít určit výchozí stav nabití, k tomu nám poslouží parametr E_{init} , který vyjádří stav nabití v MWh. Hodnota se musí pohybovat v rozmezí $\langle E_{min}, E_{max} \rangle$. Zmíněné parametry shrnuje tabulka 6.8.

Zkratka	Jednotka	Popis
T_{start}^{sim}	datetime	Počáteční datum a čas simulace
T_{end}^{sim}	datetime	Datum a čas konce simulace
$Temp_{sim}$	°C	Uvažovaná teplota během simulace
$Cost_{EE}$	Kč	Cena elektrické energie za 1 MWh pro potřeby vypočtení nákladů
$Cost_{GAS}$	Kč	Cena plynu za 1 MWh pro potřeby vypočtení nákladů
B_{count}	int	Počet identických turbín uvažovaných v hybridním zdroji
$B_{power}[]$	[kW]	Pole počátečních výkonů jednotlivých turbín, které určují výchozí stav
E_{init}	MWh	Počáteční stav energie v baterii

Tabulka 6.8: Uvažované parametry pro nastavení simulace

6.5 Rozhraní řízení

Simulace je jen prostředek pro poznání, jak se virtuální dvojče zdroje chová. Logiku, jak se zdrojem nakládat do něj vnáší právě řízení. Součástí této práce není přímý vývoj řídicího algoritmu. Avšak úkolem je navrhnout vhodné rozhraní, které umožní řízení na simulátor zdroje napojit. Pro ilustraci řízení a schopnosti ovládat chování zdroje bude v této práci implementováno jednoduché řízení, které můžeme nazvat jako manuální řízení. Do dávkového souboru bude možné zapsat sekvenci povelů s časovou značkou, kdy má být povel v simulaci vydán. Struktura povelu je velice snadná, pro vysvětlení se budeme odkazovat na uvažovaný stavový diagram zobrazený na obrázku 6.2. Povel obsahuje čas jeho vydání, číslo turbíny (uvažujeme $1 \dots B_{count}$), textovou reprezentaci jakéhokoliv povelu uvedeného ve stavovém diagramu (např. A, B, D, \dots), nadále to může být i povel pro regulaci výkonu (C), který jsme si označili jako významný a musíme si jeho textové označení pamatovat. Jedním z důvodů je právě tento účel, protože není přímo uveden ve stavovém modelu. V případě vydání regulačního povelu také musíme uvést cílový výkon v rozsahu $\langle P_{min}, P_{max} \rangle$, kterého chceme dosáhnout. Rovněž uvedeme gradient, pomocí kterého chceme výkon dosáhnout v rozsahu $\langle 0, P_{max-up}^{grad} \rangle$ nebo $\langle 0, -P_{max-down}^{grad} \rangle$ v závislosti zda chceme výkon zvýšit či snížit.

Ukázku můžeme vidět ve výpisu 1, kde první povel je na turbínu 1, kde se má vydat povel na přechod A ze stavového diagramu, druhý povel obdobně na turbínu 2 s názvem na povel D . Následující dva povely jsou oba regulační. Pro turbínu 3 chceme dosáhnout výkonu 1000 kW s gradientem 60 kW/s. Pro turbínu 4 chceme výkon 2000 kW s gradientem -80 kW/s. Uvedení znaménka

```

1  {
2      "commands": [
3          ["2022-06-30 15:30:22", 1, "A"],
4          ["2022-06-30 15:30:25", 2, "D"],
5          ["2022-06-30 15:39:48", 3, "C", 1000, 60],
6          ["2022-06-30 15:39:50", 4, "C", 2000, -80],
7      ]
8  }

```

Výpis 1: Ukázka nízkourovňových povelů pro ovládání turbín ve formátu JSON[27]

při snížení výkonu v zásadě není podstatné, simulátor by měl vědět, jakým směrem se výkon bude měnit a je pouze potřeba uvést míru změny mezi aktuálním výkonem v době uvolnění povelu a cílového výkonu.

Všechny tyto povelů můžeme označit jako nízkourovňové, protože vychází přímo z definice stavového diagramu. Avšak v praxi pro vytváření komplexnějšího řízení, kdy nás dílčí kroky nezajímají, bychom pro zjednodušení ovládání uvítali i vysokoúrovňové povelů. Z obrázku 6.2 je zřejmé, že startovací sekvence se skládá z více povelů, které musí být v určitém časovém odstupu vydány po sobě (povel *A*, povel *B*), pro nejrychlejší přechod bychom toto rádi zapouzdřili do jednoho povel, který provede celou tuto sekvenci ze stavu *READY* do stavu *POWER*. Požadavek na funkci ještě zobecníme a na povel *ON* budeme chtít, aby turbína v co nejkratším čase přešla do výkonového stavu *POWER* z kteréhokoliv stavu, ve kterém se původně nachází. Podobně pro uvedení turbíny do stavu *READY* budeme chtít povel *OFF*, který provede vypínací sekvenci co nejdříve. Pro doplnění povelů z kosmetických důvodů zavedeme název *POWER* pro regulaci výkonu, který je pouze zapouzdřením pro významný povel pro regulaci, dle našeho příkladu konkrétně povel *C*.

Ukázku vysokoúrovňových povelů můžeme vidět ve výpisu 2. Jako první povel je prosté volání povelu *ON* na turbínu 1. Následuje volání povelu *OFF* na turbínu 2. Po turbíně 3 budeme chtít dosáhnout výkon 1000 kW s gradientem 60 kW/s. Další dva povelů jsou rozšířenou verzí *ON* a *OFF*. Pro ještě větší zjednodušení budeme chtít uvést turbínu do stavu *POWER* a ihned po jejím přechodu do výkonového stavu provést regulační povel. Toto nám zabezpečí povel volaný na turbínu 4, kde následně cílovým výkonem bude 1000 kW s gradientem 60 kW/s. Pro opačný případ, kde budeme chtít odstavit turbínu, která se nachází ve výkonu, můžeme provést příkaz jako pro turbínu 5. Z jejího aktuální výkonu dojde nejdříve ke snížení výkonu na minimální technický gradientem 80 kW/s a ihned po jeho dosažení dojde k odstavení pomocí povelu *OFF*.

```
1 {  
2   "commands": [  
3     ["2022-06-30 16:30:22", 1, "ON"],  
4     ["2022-06-30 16:30:25", 2, "OFF"],  
5     ["2022-06-30 16:39:48", 3, "POWER", 1000, 60],  
6     ["2022-06-30 16:39:50", 4, "ON", 1000, 60],  
7     ["2022-06-30 16:39:52", 5, "OFF", -80],  
8   ]  
9 }
```

Výpis 2: Ukázka vysokoúrovňových povelů pro ovládání turbín ve formátu JSON

Uvedli jsme si formát povelů, které poslouží pro zápis manuálního řízení a zároveň v podobném datové logice budeme chtít povely předávat mezi řízením a technologiemi turbín simulace. Nadále uvedeme požadavky na datové rozhraní, které bude poskytovat informace o stavu technologií řídicímu algoritmu. Zmíněné manuální řízení tyto informace nebudeme potřebovat, nicméně pro vývoj komplexnějších řídicích algoritmů založených na aktuálním stavu technologií, například to může být podmínkové, heuristické, pravděpodobnostní řízení, tyto informace budou nezbytné. Konkrétní architekturu budeme řešit až v následující kapitole. V základní představě budeme chtít, aby řízení každou vteřinu obdrželo aktuální data o zařízení, mohlo provést svou implementaci a zpět do simulace předat povely na řízení turbín, které budou ihned vykonány. Řízení bude vydávat i povel na baterku, kde uvažujeme aktuálně požadovaný výkon a gradient výkonu do příští vteřiny. Předpokládáme, že povely na turbíny budou vydávány dle potřeby, povel na baterku bude vydán vždy. Součástí poskytovaných dat by měli být informace o turbínách, baterii, požadavcích na služby.

Řízení obdrží reálný čas v simulaci a rovněž simulační čas, který je ve formátu počtu sekund od zahájení simulace. O každé turbíně obdrží informace o aktuálním výkonu a gradientu, aktuální stav ze stavového diagramu a dobu setrvání turbíny v tomto stavu, nakonec je poskytnuto číslo turbíny, které poslouží pro rozlišení a bude dobré pro vydání povelu k turbíně. O baterii budou poskytnuty informace o výkonu a gradientu výkonu, stav nabití. Jako další část budou informace o službách, které budou pro všechny tři služby identické. Řízení obdrží aktuálně žádaný výkon služby a příslušný gradient výkonu do příští vteřiny. Zároveň bude poskytnuta informace o posledním obdrženém požadavku uvedeném pomocí průběhu. Průběh je popsán body ve formátu čas a příslušný výkon. Shrnutí rozhraní přináší tabulka 6.9. Pro úplnost dodáme, že řízení bude muset mít k dispozici parametry o technologiích jako například u baterie maximální kapacita, které jsme zavedli a popsali dříve.

Atribut	Jednotka	Popis
Čas simulace	s	Počet sekund od zadaného počátečního data simulace
Reálný čas	datetime	Reálné datum v simulaci
Pro každou turbínu		
Index	int	Číslo turbíny v rozsahu $\langle 1, B_{count} \rangle$, který je shodný s uváděným při vydání povelu
Výkon	W	Aktuální výkon dodávaný turbínou
Gradient	W/s	Aktuální gradient výkonu turbíny
Stav	ref	Reference na aktuální stav ze stavového diagramu popisující stavy turbíny
Doba setrvání ve stavu	s	Doba, jak dlouho se turbína již nachází v aktuálně uvedeném stavu.
Baterie		
Výkon	W	Aktuální výkon dodávaný baterií
Gradient	W/s	Aktuální gradient výkonu baterie
Stav nabití	W	Aktuální úroveň nabití baterie
Služby		
FCR výkon	W	Aktuální žádaný výkon na FCR dle interpretace povelu
FCR gradient	W/s	Aktuální gradient výkonu FCR dle interpretace povelu
FCR požadavek	[(čas, výkon)]	Žádaný průběh posledního požadavku na FCR
aFRR výkon	W	Aktuální žádaný výkon na aFRR dle interpretace povelu
aFRR gradient	W/s	Aktuální gradient výkonu aFRR dle interpretace povelu
aFRR požadavek	[(čas, výkon)]	Žádaný průběh posledního požadavku na aFRR
mFRR výkon	W	Aktuální žádaný výkon na mFRR dle interpretace povelu
mFRR gradient	W/s	Aktuální gradient výkonu mFRR dle interpretace povelu
mFRR požadavek	[(čas, výkon)]	Žádaný průběh posledního požadavku na mFRR

Tabulka 6.9: Uvažované parametry pro nastavení simulace

Kapitola 7

Návrh řešení

V předchozí kapitole jsme provedli úvod do problematiky, kde jsme poskytli základní popis některých účastníků se komponent včetně zavedení parametrů, které mají na jejich chování vliv. Mimo to jsme uvedli informace o některých vstupech. V této kapitole na tyto informace navážeme a popíšeme vzniklé řešení. V první části uvedeme požadavky, které budou popisovat nároky na části systému, ale zároveň bude v nich detailněji zachycena koncepční myšlenka jejich funkčnosti. Navážeme částí, kde načrtnuté celky propojíme skrze architekturu a uvedeme vazby a funkce účastníků se komponent. V poslední části zmíníme informace k implementaci z hlediska použitých technologií.

Často zde zazněli pojmy jako simulátor a řízení. Obě komponenty dohromady tvoří jeden celek a to digitální dvojče. Existuje ve formě programu založen na jednorázovém běhu. Na počátku načte ze souborů vstupní data, dle toho inicializuje prostředí simulace a komponenty tvořící celé digitální dvojče. Obsluha simulace není vytvořena od základu, ale je využito frameworku SimPy[28], který je založen na událostech a procesech. Je třeba to zmínit takto explicitně předem, protože v požadavcích se s tímto zvoleným konceptem již počítalo. V průběhu simulace se na základě povelů ze řízení vykonávají aktualizace stavů technologií a o těchto změnách se průběžně zaznamenávají data. Po ukončení simulace jsou data vizualizována pomocí vhodných grafů a jsou rovněž poskytnuta jako výstupy uložením do souborů.

7.1 Požadavky

V následující části se zaměříme na popis systému skrze požadavky. *Na úvod čtenáře musíme upozornit, že se nebude jednat o standardní formát požadavků, který se obvykle využívá v oblasti softwarového inženýrství. Většina následujících požadavků bude popsána více do hloubky a uvedený popis chování bude často na úrovni pseudokódu – popis chování na zavedených proměnných. Tento formát byl při vývoji zvolen záměrně z důvodu, aby se v rané fázi vývoje ještě před zahájením implementace odhalili základní koncepční chyby. Seznam požadavků sloužil k ujasnění rozsahu systému spolu se zadavatelem před zahájením implementace a urychlil samotný vývoj, protože nebylo již potřeba v plném rozsahu rozmýšlet návrh funkčnosti komponent.* Požadavky

rozčleníme do několika kategorií podle toho, jaké části systému se týkají. Zaměříme se na požadavky týkající se turbín, baterie, simulace, uživatelského rozhraní.

Na úvod zavedeme některé proměnné, které využijeme k popisu požadavků týkajících se turbíny a baterie. V našem systému bude figurovat čas t , který se pohybuje přes celý průběh simulace v rozsahu od počátku simulace vyjádřeném 0 až do konce určeném nějakým N . Implicitně se předpokládá, že v požadavcích pokud bude zmíněno t je tím zamýšlen aktuální čas. Tento parametr se v požadavcích objevuje často, proto nebude výslovně uveden jako vstup. Pro zdůraznění tento parametr uvádíme tabulkou 7.1

Zkratka	Jednotka	Výchozí hodnota	Popis
t	s	0	Čas v simulaci. Předpokládá se jako vstup do všech požadavků, kde je potřeba.

Tabulka 7.1: Uvedení proměnné času v simulaci

Turbínu si popíšeme pomocí následujících proměnných. TUR_STAV_i , která představuje aktuální stav turbíny z výčtů stavů uvedených v definici stavového diagramu. Časová značka $TUR_STAV_TIMESTAMP_i$, která ukládá informaci, kdy došlo k poslední změně stavu. $TUR_VYSTUPNI_VYKON_{ti}$ ukládá hodnotu výstupního výkonu. Při povelu změny výkonu nám poslouží $TUR_VYSTUPNI_VYKON_POCATECNI_i$ k zaznamenání počátečního výkonu při zahájení změny výkonu a $TUR_VYSTUPNI_VYKON_CILOVY_i$ nám říká, jakého výkonu chceme dosáhnout. Zároveň musíme evidovat časy, kdy jsme zahájili poslední změnu výkonu $TUR_ZMENA_VYKONU_CAS_i$ a čas kdy dosáhneme, nebo bylo dosaženo cílového výkonu $TUR_ZMENA_VYKONU_CAS_CILOVY_i$. Pro uložení jakým gradientem je výkon měněn poslouží $TUR_ZMENA_VYKONU_GRADIENT_i$. Pro potřeby zachycení spotřeby elektrické energie se může hodit proměnná $TUR_SPOTREBA_EL_{ti}$ a u plynu $TUR_SPOTREBA_GAS_{ti}$. Jelikož je využito událostmi řízeného frameworku, tak u některých událostí, které mají nastat (například přechod do jiného stavu pomocí timeoutu, ukončení změny výkonu turbíny) je vhodné si uložit referenci na proces, který má provést zamýšlenou návaznou změnu. Tuto referenci využijeme pro případ přerušení procesu, pokud se změnil záměr a původně zamýšlený efekt již nemá proběhnout. Na tuto referenci poslouží proměnná $TUR_PLANOVA_UDALOST_i$. Pro zmíněný přechod do jiného stavu po uplynutí timeoutu se nám bude hodit ještě $TUR_DOBA_TIMEOUT_i$. Uvedené proměnné přehledně shrnuje tabulka 7.2.

Zkratka	Jednotka	Výchozí hodnota	Popis
TUR_STAV_i	-	<i>OFF</i> stav	Stav turbíny z výčtu stavů, ve kterém se turbína aktuálně nachází.
$TUR_STAV_ -$ $TIMESTAMP_i$	s	0	Časová značka, která říká, jaký byl čas simulace při poslední změně stavu turbíny.
$TUR_VYSTUPNI_ -$ $VYKON_{t_i}$	W	0	Hodnota výkonu na výstupu turbíny i v čase t .
$TUR_VYSTUPNI_ -$ $VYKON_ -$ $POCATECNI_i$	W	0	Při povelu změny výkonu (poslední povel) říká, jaký byl výkon na počátku změny.
$TUR_VYSTUPNI_ -$ $VYKON_CILOVY_i$	W	0	Při povelu změny výkonu říká, jakého výkonu chceme dosáhnout.
$TUR_ZMENA_ -$ $VYKONU_CAS_i$	s	0	Čas posledního povelu na změnu výkonu.
$TUR_ZMENA_ -$ $VYKONU_CAS_ -$ $CILOVY_i$	s	0	Čas dosažení cílového výkonu posledním povellem.
$TUR_ZMENA_ -$ $VYKONU_ -$ $GRADIENT_i$	W/s	0	Gradient, kterým má být výkon měněn.
$TUR_SPOTREBA_ -$ EL_{t_i}	W	0	Aktuální spotřeba elektrické energie turbíny i v čase t .
$TUR_SPOTREBA_ -$ GAS_{t_i}	W	0	Aktuální spotřeba plynu turbíny i v čase t .
$TUR_PLANOVANA_ -$ $UDALOST_i$	ref	null	Reference na případnou naplánovanou událost pro přechod turbíny do jiného stavu za nějakou dobu, nebo ukončení změny výkonu.
$TUR_DOBA_ -$ $TIMEOUT_i$	s	0	Pomocná hodnota pro uložení doby timeoutu pro aktuální stav.

Tabulka 7.2: Datová reprezentace turbíny pomocí proměnných

Podobně popíšeme baterii, poskytovaný výkon zavedeme jako $BAT_ -$
 $VYKON$. V souvislosti s výkonem u baterie zavedeme konvenci. Pokud je hodnota kladná, výkon je baterií dodáván (dochází k vybíjení). Naopak pokud je hodnota záporná, dochází k absorbování výkonu (dochází k nabíjení). Jako další musíme udržovat informaci o stavu nabití BAT_STAV . Baterie umožňuje měnit výkon, který poskytuje nebo absorbuje, pro účely zachycení tohoto chování budeme potřebovat $BAT_ZMENA_VYKONU_GRADIENT$, který zachycuje gradient od poslední změny výkonu a $BAT_ZMENA_ -$
 $VYKONU$, který ukládá o informaci o časové značce, kdy došlo k poslední změně výkonu. Zmíněné proměnné přehledně shrnuje tabulka 7.3.

Zkratka	Jednotka	Výchozí hodnota	Popis
BAT_VYKON_t	W	0	Výkon baterie, který je poskytován v čase t . V případě kladné hodnoty baterie dodává energii (vybízí se), v případě záporné hodnoty je energie akumulována (nabízí se).
BAT_STAV_t	W	E_{init}	Stav uložené energie v baterii v čase t .
$BAT_ZMENA_VYKONU_GRADIENT$	W/s	0	Gradient, kterým je výkon postupně měněn.
BAT_ZMENA_VYKONU	W	0	Časová značka, která říká, jaký byl čas simulace při poslední změně výkonu.

Tabulka 7.3: Datová reprezentace baterie pomocí proměnných

V předchozí části jsme uvedli seznam proměnných potřebných pro detailnější popis, nyní můžeme uvést samotné požadavky rozdělené do tematických celků.

Na závěr ještě poznamenejme, že podoba požadavků není zcela v produkční formě, zachycují koncepční návrh komponent a částí systému. Cílové řešení ilustrované v následující kapitole obsahuje další aktualizace této verze (opravy nepřesností a ladění, další požadavky vzešlé od zadavatele). Nicméně základní myšlenka ze které se vycházelo je v těchto požadavcích uvedena.

7.1.1 Požadavky turbíny

Následující uvedená skupina požadavků se zabývá modelováním turbíny. Popis je proveden na základě definice turbíny pomocí proměnných dle tabulky 7.2. Jsou zde popsány schopnosti turbíny – inicializace do výchozího stavu, určení spotřeby plynu a maximálního výkonu, změna stavu na základě povelů nebo vynucených hran *timetoutem* dle stavového diagramu, změna výkonu a hlídání technických omezení. Podrobný popis vnitřního fungování následuje v požadavcích uvedených pomocí tabulek.

#TUR1	Inicializuj turbínu i do stavu <i>OFF/READY</i>
Účel	Výchozí inicializace turbíny i do stavu <i>OFF/READY</i> .
Akce	<p>Výchozím stavem pro turbínu i je stav <i>OFF/READY</i>, nastaví se:</p> <ul style="list-style-type: none"> ■ $TUR_STAV_i = ZADANY_STAV$ (dle určené reference pro <i>OFF/READY</i> stav z konfigurace stavového diagramu) ■ $TUR_STAV_TIMESTAMP_i = 0$ ■ $TUR_VYSTUPNI_VYKON_{ti} = 0$ ■ $TUR_SPOTREBA_EL_{ti} = ZADANY_STAV.EE_STAVU$ ■ $TUR_SPOTREBA_GAS_{ti} = 0$
Vstup	<ul style="list-style-type: none"> ■ $ZADANY_STAV = OFF$ nebo $READY$ (reference na stav popsaný stavovým diagramem)
Výstup	<ul style="list-style-type: none"> ■ TUR_STAV_i ■ $TUR_VYSTUPNI_VYKON_{ti}$ ■ $TUR_SPOTREBA_EL_{ti}$ ■ $TUR_SPOTREBA_GAS_{ti}$ ■ $TUR_ZMENA_VYKONU_GRADIENT_i = 0$

#TUR2	Vypočítej spotřebu plynu
Účel	Vypočítá spotřebu plynu na základě vstupních hodnot, uvažujeme jako funkci $VYPOCITEJ_SPOTREBU_PLYNU(TEPLOTA, VYKON)$.
Akce	<ul style="list-style-type: none"> ■ Zjistíme stav turbíny $STAV = TUR_STAV_i$ ■ Pokud $STAV.GAS_STAVU = VAR$: <ul style="list-style-type: none"> ■ Dle $TEPLOTA$ zjistíme v nominální tabulce výkonů jakému procentuálnímu výkonu odpovídá $VYKON$. Dle potřeby přeskálujeme hodnoty sloupců lineární aproximací na zadanou teplotu dle dvou nejbližších hodnot teplot, mezi kterými zadaná teplota leží. Pak počítáme procentuální poměr mezi dvěma sloupečky, mezi kterými leží $VYKON$. Pokud někde hledaná hodnota přesahuje definované rozsahy pomocí referenčních hodnot, bereme krajní hodnotu. ■ V tabulce nominálních spotřeb hledáme dle $TEPLOTA$ a výkonu vyjádřeného v procentech. Dle potřeby přeskálujeme lineární aproximací opět dle dvou nejbližších definovaných hodnot, mezi kterými námi hledaná hodnota leží. Nalezenou hodnotu označíme jako $NALEZENA_HODNOTA$ ■ $TUR_SPOTREBA_GAS_{ti} = NALEZENA_HODNOTA$ ■ Jinak: <ul style="list-style-type: none"> ■ $TUR_SPOTREBA_GAS_{ti} = STAV.GAS_STAVU$
Vstup	<ul style="list-style-type: none"> ■ $TEPLOTA$ ■ $VYKON$ ■ nominální tabulka výkonů a nominální tabulka spotřeb (ukázku tabulky ilustrují tabulky 6.4 a 6.2)
Výstup	<ul style="list-style-type: none"> ■ $TUR_SPOTREBA_GAS_{ti}$
#TUR3	Maximální výkon
Účel	Vrátí aktuálně možný maximální výkon turbíny i vzhledem k teplotě. V požadavcích uvažujeme jako funkci $MAX_VYKON(TEPLOTA)$.
Akce	<ul style="list-style-type: none"> ■ V nominální tabulce výkonů, hledáme hodnotu výkonu odpovídající 100% při teplotě $TEPLOTA$. Dle potřeby hledanou hodnotu vypočteme lineární aproximací mezi dvěma nejbližšími referenčními hodnotami, mezi kterými se hledaná hodnota nachází. Nalezenou hodnotu označíme jako $NALEZENA_HODNOTA$ ■ $MAX_VYKON = NALEZENA_HODNOTA$
Vstup	<ul style="list-style-type: none"> ■ $TEPLOTA$ ■ nominální tabulka výkonů (ukázku tabulky ilustruje 6.4)
Výstup	<ul style="list-style-type: none"> ■ MAX_VYKON

#TUR4	Inicializuj turbínu i do stavu <i>POWER</i>
Účel	Výchozí inicializace turbíny i do stavu <i>POWER</i> .
Akce	<p>Pro turbínu i se nastaví:</p> <ul style="list-style-type: none"> ■ $TUR_STAV_i = ZADANY_STAV$ ■ $TUR_STAV_TIMESTAMP_i = 0$ ■ $TUR_VYSTUPNI_VYKON_{ti} = \max(P_{min}, \min(POCATECNI_VYKON_i, MAX_VYKON(Temp_{sim})[#TUR3]))$ ■ $TUR_SPOTREBA_EL_{ti} = ZADANY_STAV.EE_STAVU$ ■ $TUR_SPOTREBA_GAS_{ti} = VYPOCITEJ_SPOTREBU_PLYNU(Temp_{sim}, TUR_VYSTUPNI_VYKON_{ti})[#TUR2]$ ■ Ostatní proměnné charakterizující stav turbíny jsou nastaveny dle výchozích hodnot
Vstup	<ul style="list-style-type: none"> ■ $ZADANY_STAV = POWER$ (reference na stav popsaný stavovým diagramem) ■ $POCATECNI_VYKON_i$ ■ $Temp_{sim}$
Výstup	<ul style="list-style-type: none"> ■ TUR_STAV_i ■ $TUR_VYSTUPNI_VYKON_{ti}$ ■ $TUR_SPOTREBA_EL_{ti}$ ■ $TUR_SPOTREBA_GAS_{ti}$ ■ $TUR_ZMENA_VYKONU_GRADIENT_i = 0$

#TUR5	Získej aktuální výkon turbíny
Účel	Vrátí aktuální výkon turbíny i . V požadavcích uvažujeme jako funkci $ZJISTI_AKTUALNI_VYKON(i)$
Akce	<p>Pokud se turbína i nenachází v <i>POWER</i> stavu tak:</p> <ul style="list-style-type: none"> ■ $TUR_VYSTUPNI_VYKON_{ti} = 0$ <p>V případě, že turbína poskytuje výkon, tj. nachází se v <i>POWER</i> stavu tak:</p> <ul style="list-style-type: none"> ■ Pokud $t \geq TUR_ZMENA_VYKONU_CAS_CILOVY_i$: <ul style="list-style-type: none"> ■ vrací $TUR_VYSTUPNI_VYKON_{ti}$ ■ Jinak: <ul style="list-style-type: none"> ■ vrací $TUR_VYSTUPNI_VYKON_{ti} = TUR_VYSTUPNI_VYKON_POCATECNI_i + [(t - TUR_ZMENA_VYKONU_CAS_i) * TUR_ZMENA_VYKONU_GRADIENT_i]$
Vstup	<ul style="list-style-type: none"> ■ TUR_STAV_i ■ $TUR_VYSTUPNI_VYKON_POCATECNI_i$ ■ $TUR_ZMENA_VYKONU_CAS_i$ ■ $TUR_ZMENA_VYKONU_CAS_CILOVY_i$ ■ $TUR_ZMENA_VYKONU_GRADIENT_i$
Výstup	<ul style="list-style-type: none"> ■ $TUR_VYSTUPNI_VYKON_{ti}$

#TUR6	Technické omezení povelu na regulaci výkonu
Účel	Turbína má své technické limity na řízení výkonu, parametry povelu musí tyto omezení respektovat. Omezení na změnu výkonu vzhledem k dvou sousedícím sekundám, minimální a maximální rozsah na gradient změny výkonu, minimální technický výkon. Pokud dojde k porušení, dojde k chybovému stavu.
Akce	<p>Zavedeme si:</p> <ul style="list-style-type: none"> ■ $POZADAVEK_ZMENA_GRADIENT_i = (POZADAVEK_CILOVY_VYKON_i - ZJISTI_AKTUALNI_VYKON(i)[\#TUR5]) / (POZADAVEK_CILOVY_CAS_i - t)$ ■ $TUR_AKTUALNI_GRADIENT_i = TUR_VYSTUPNI_VYKON_{ti} - TUR_VYSTUPNI_VYKON_{t-1i}$ <p>Musí splňovat podmínky:</p> <ul style="list-style-type: none"> ■ $POZADAVEK_CILOVY_VYKON_i \geq P_{min}$ ■ $POZADAVEK_CILOVY_VYKON_i \leq MAX_VYKON(Temp_{sim})[\#TUR3]$ ■ Pokud $POZADAVEK_ZMENA_GRADIENT_i > 0$, tak $POZADAVEK_ZMENA_GRADIENT_i \leq P_{max-up}^{grad}$, pokud $POZADAVEK_ZMENA_GRADIENT_i < 0$, tak $POZADAVEK_ZMENA_GRADIENT_i \leq P_{max-down}^{grad}$ ■ $TUR_AKTUALNI_GRADIENT_i - POZADAVEK_ZMENA_GRADIENT_i \leq P_{max-change}^{grad}$ <p>Pokud došlo k porušení některé z podmínek, vznikne chybový stav a provede se jeho zpracování.</p>
Vstup	<ul style="list-style-type: none"> ■ $POZADAVEK_CILOVY_VYKON_i$ ■ $POZADAVEK_CILOVY_CAS_i$ ■ $TUR_VYSTUPNI_VYKON_{ti}$ ■ $TUR_VYSTUPNI_VYKON_{t-1i}$ ■ $Temp_{sim}$
Výstup	

#TUR7	Povel změna výkonu
Účel	Provede změnu výkonu turbíny i ve stavu schopném dodávat výkon - <i>POWER</i> stavu
Akce	<p>Turbína i musí být v <i>POWER</i> stavu, pokud přijde požadavek na změnu výkonu (regulační povel), zkontroluje se, zda požadavek odpovídá technickým limitům [#TUR6]. Pokud $t < TUR_ZMENA_VYKONU_CAS_CILOVY_i$ (tj. turbína se nachází v procesu změny výkonu), pak zruší naplánovanou událost „ukončení změny výkonu“ [#TUR8] v $TUR_PLANOVAN_UDALOST_i$ a odstraní její referenci. Nastaví se:</p> <ul style="list-style-type: none"> ■ $TUR_TIMESTAMP_STAV_i = t$ ■ $TUR_VYSTUPNI_VYKON_{ti} = ZJISTI_AKTUALNI_VYKON(i)[\#TUR5]$ ■ $TUR_SPOTREBA_GAS_{ti} = VYPOCITEJ_SPOTREBU_PLYNU(Temp_{sim}, TUR_VYSTUPNI_VYKON_{ti})[\#TUR2]$ ■ $TUR_VYSTUPNI_VYKON_POCATECNI_i = ZJISTI_AKTUALNI_VYKON(i)[\#TUR5]$ ■ $TUR_VYSTUPNI_VYKON_CILOVY_i = POZADAVEK_CILOVY_VYKON_i$ ■ $TUR_ZMENA_VYKONU_CAS_i = t$ ■ $TUR_ZMENA_VYKONU_CAS_CILOVY_i = POZADAVEK_CILOVY_CAS_i$ ■ $TUR_ZMENA_VYKONU_GRADIENT_i = (POZADAVEK_CILOVY_VYKON_i - ZJISTI_AKTUALNI_VYKON(i)[\#TUR5]) / (POZADAVEK_CILOVY_CAS_i - t)$ <p>Naplňuje se událost „ukončení změny výkonu“ [#TUR8] v čase $POZADAVEK_CILOVY_CAS_i$. Referenci na událost uloží do $TUR_PLANOVAN_UDALOST_i$.</p>
Vstup	<ul style="list-style-type: none"> ■ TUR_STAV_i ■ $Temp_{sim}$ ■ $POZADAVEK_CILOVY_VYKON_i$ ■ $POZADAVEK_CILOVY_CAS_i$
Výstup	<ul style="list-style-type: none"> ■ $TUR_VYSTUPNI_VYKON_{ti}$ ■ $TUR_SPOTREBA_EL_{ti}$ ■ $TUR_SPOTREBA_GAS_{ti}$ ■ $TUR_ZMENA_VYKONU_GRADIENT_i$ ■ Log: Regulační povel

#TUR8	Ukončí změnu výkonu a aktualizuj stav turbíny
Účel	Představuje moment konce změny výkonu na turbíně i , provede aktualizaci informací turbíny v tomto bodě konce změny výkonu.
Akce	<p>Turbína i musí být v <i>POWER</i> stavu, pokud $t = TUR_ZMENA_VYKONU_CAS_CILOVY_i$, nastaví:</p> <ul style="list-style-type: none"> ■ $TUR_TIMESTAMP_STAV_i = t$ ■ $TUR_VYSTUPNI_VYKON_{ti} = TUR_VYSTUPNI_VYKON_CILOVY_i$ ■ $TUR_ZMENA_VYKONU_GRADIENT_i = 0$ ■ $TUR_SPOTREBA_GAS_{ti} = VYPOCITEJ_SPOTREBU_PLYNU(Temp_{sim}, TUR_VYSTUPNI_VYKON_{ti})[\#TUR2]$ <p>Odstraní reference na tuto událost v $TUR_PLANOVA_UDALOST_i$.</p>
Vstup	<ul style="list-style-type: none"> ■ TUR_STAV_i ■ $Temp_{sim}$ ■ $TUR_VYSTUPNI_VYKON_CILOVY_i$ ■ $TUR_ZMENA_VYKONU_CAS_CILOVY_i$
Výstup	<ul style="list-style-type: none"> ■ $TUR_VYSTUPNI_VYKON_{ti}$ ■ $TUR_SPOTREBA_EL_{ti}$ ■ $TUR_SPOTREBA_GAS_{ti}$ ■ $TUR_ZMENA_VYKONU_GRADIENT_i$

#TUR9	Uveď turbínu i do požadovaného stavu
Účel	Slouží k nastavení příslušných proměnných do požadovaného stavu. V požadavcích uvažujeme jako funkci $NASTAV_STAV(POZADOVANY_STAV)$.
Akce	<p>Podle názvu stavu se dohledá datová struktura náležící k požadovanému stavu ($POZADOVANY_STAV$) jako $STAV$. Pro turbínu i se nastaví:</p> <ul style="list-style-type: none"> ■ $TUR_STAV_i = STAV$ ■ $TUR_TIMESTAMP_STAV_i = t$ ■ $TUR_VYSTUPNI_VYKON_{ti} =$ Pokud $STAV$ je $POWER$ nastaví P_{min}, jinak 0 ■ $TUR_SPOTREBA_EL_{ti} = STAV.EE_STAVU$ ■ $TUR_SPOTREBA_GAS_{ti} = VYPOCITEJ_SPOTREBU_PLYNU(Temp_{sim}, TUR_VYSTUPNI_VYKON_{ti})[\#TUR2]$ <p>Pokud $STAV$ je $POWER$ nastaví:</p> <ul style="list-style-type: none"> ■ $TUR_VYSTUPNI_VYKON_POCATECNI_i = P_{min}$ ■ $TUR_VYSTUPNI_VYKON_CILOVY_i = P_{min}$ ■ $TUR_ZMENA_VYKONU_CAS_i = t$ ■ $TUR_ZMENA_VYKONU_CAS_CILOVY_i = t$ ■ $TUR_ZMENA_VYKONU_GRADIENT_i = 0$ <p>Zkontroluje přes všechny přechody ze stavu $STAV$, zda existuje takový, který by byl podmíněn timeoutem. Pokud ano, označíme ho jako $PRECHOD$. Nastaví se $TUR_DOBA_TIMEOUT_i = PRECHOD.TIMEOUT_STAVU$ a naplánuje se událost pro vykonání přechodu do stavu dle příslušného přechodu (provede se $ZPRACUJ_TIMEOUT(PRECHOD.DO_STAVU)[\#TUR11]$) za dobu $TUR_DOBA_TIMEOUT_i$. Referenci na událost uloží do $TUR_PLANOVA_NA_UDALOST_i$. (Pozn. Pokud $STAV$ je $POWER$ neměl by takový přechod existovat)</p>
Vstup	<ul style="list-style-type: none"> ■ $POZADOVANY_STAV$ (jako název) ■ $Temp_{sim}$
Výstup	<ul style="list-style-type: none"> ■ TUR_STAV_i ■ $TUR_VYSTUPNI_VYKON_{ti}$ ■ $TUR_SPOTREBA_EL_{ti}$ ■ $TUR_SPOTREBA_GAS_{ti}$

#TUR10	Zpracuj povel
Účel	Neplatí pro regulační povel, který řeší zvláštní požadavek. Pokud výchozím stavem je stav poskytující výkon, musí být na minimálním výkonu.
Akce	<p>Pokud turbína i byla ve stavu s plánovanou událostí – timeout, zkontroluje, zda turbína i setrvala ve stavu méně než dle nastaveného timeoutu – tj. $(TUR_STAV_TIMESTAMP_i - t) < TUR_DOBA_TIMEOUT_i$ a zruší naplánovanou událost uloženou v $TUR_PLANOVANA_UDALOST_i$ a odstraní její referenci.</p> <p>Podle názvu stavu TUR_STAV_i se dohledá příslušná datová struktura jako $STAV$. Pokud $STAV$ je $POWER$ zkontroluje se, zda hodnota odpovídá minimálnímu technickému výkonu, tj. $ZJISTI_AKTUALNI_VYKON(i)[\#TUR5] = P_{min}$.</p> <p>Zkontroluje, zda pro aktuální stav TUR_STAV_i existuje přechod X pro požadovaný $POVEL$ – tj. zda najdeme $X.ZE_STAVU = TUR_STAV_i$ a $X.POVEL_PRECHODU = POVEL$. Tento přechod označíme jako $PRECHOD$. Provede změnu stavu pomocí $NASTAV_STAV(PRECHOD.DO_STAVU)[\#TUR9]$.</p>
Vstup	<ul style="list-style-type: none"> ■ $POVEL$ (řetězec názvu povelu) ■ TUR_STAV_i ■ $TUR_DOBA_TIMEOUT_i$ ■ $TUR_STAV_TIMESTAMP_i$ ■ $TUR_PLANOVANA_UDALOST_i$
Výstup	<ul style="list-style-type: none"> ■ Log: $POVEL$
#TUR11	Zpracuj timeout
Účel	V požadavcích uvažujeme jako funkci $ZPRACUJ_TIMEOUT(\text{požadovaný_stav})$.
Akce	<p>Zkontroluje, zda pro aktuální stav TUR_STAV_i existuje přechod X s timeoutem – tj. zda najdeme $X.ZE_STAVU = TUR_STAV_i$, $X.POVEL_PRECHODU = TIMEOUT$ (rezervovaný název povelu), $X.TIMEOUT_PRECHODU = TUR_DOBA_TIMEOUT_i$ a $X.DO_STAVU = POZADOVANY_STAV$.</p> <p>Zkontroluje, zda turbína i setrvala ve stavu dle nastaveného timeoutu – tj. $(TUR_STAV_TIMESTAMP_i - t) = TUR_DOBA_TIMEOUT_i$.</p> <p>Odstraní referenci v $TUR_PLANOVANA_UDALOST_i$.</p> <p>Provede změnu stavu pomocí $NASTAV_STAV(POZADOVANY_STAV)[\#TUR9]$.</p>
Vstup	<ul style="list-style-type: none"> ■ $POZADOVANY_STAV$ ■ TUR_STAV_i ■ $TUR_DOBA_TIMEOUT_i$ ■ $TUR_STAV_TIMESTAMP_i$ ■ $TUR_PLANOVANA_UDALOST_i$
Výstup	

V rámci všech požadavků, pokud došlo k nesplnění předpokladů akce, vzniká chybový stav a provede se jeho zpracování.

7.1.2 Požadavky baterie

Druhá skupina požadavků se zabývá modelováním baterie. Popis je proveden na základě reprezentace baterie pomocí proměnných dle tabulky 7.3. Jsou zde popsány schopnosti jako inicializovat svůj stav, získat hodnotu aktuálního výkonu, zjistit jaké množství energie je možné maximálně uložit nebo poskytnout, upravit výkon baterie. Uvažujeme, že baterie je schopná výkon poskytovat, nebo přijímat okamžitě – odezva bez prodlení. Podrobný popis funkcí následuje v tabulkách.

#BAT1	Inicializace baterie
Účel	Na počátku simulace jsou proměnné baterie inicializovány.
Akce	Nastaví: <ul style="list-style-type: none"> ■ $BAT_VYKON_t = 0$ ■ $BAT_STAV_t = BAT_POCATECNI_STAV$ ■ $BAT_ZMENA_VYKONU_GRADIENT = 0$ ■ $BAT_ZMENA_VYKONU = 0$
Vstup	<ul style="list-style-type: none"> ■ $BAT_POCATECNI_STAV$
Výstup	<ul style="list-style-type: none"> ■ BAT_VYKON_t ■ BAT_STAV_t ■ $BAT_ZMENA_VYKONU_GRADIENT$
#BAT2	Získej aktuální výkon baterie
Účel	Vrátí aktuální výkon baterie. V požadavcích uvažujeme jako funkci $ZJISTI_AKTUALNI_VYKON_BATERIE()$.
Akce	Zavedeme $DOBA_K_DOPOCTU = t - BAT_ZMENA_VYKONU$ $AKTUALNI_VYKON_BATERIE = BAT_VYKON_{BAT_ZMENA_VYKONU} + (DOBA_K_DOPOCTU * BAT_ZMENA_VYKONU_GRADIENT)$
Vstup	<ul style="list-style-type: none"> ■ BAT_ZMENA_VYKONU ■ $BAT_VYKON_{BAT_ZMENA_VYKONU}$ ■ $BAT_ZMENA_VYKONU_GRADIENT$
Výstup	<ul style="list-style-type: none"> ■ $AKTUALNI_VYKON_BATERIE$

#BAT3	Aktuální stav uložené energie
Účel	Vrátí aktuální stav energie v baterii. V požadavcích uvažujeme jako funkci $BAT_AKTUALNI_STAV()$
Akce	<p>Zavedeme $DOBA_K_DOPOCTU = t - BAT_ZMENA_VYKONU$</p> <p>$BAT_VYKON = ZJISTI_AKTUALNI_VYKON_BATERIE()[\#BAT2]$</p> <p>Pokud $BAT_VYKON < 0$ (dobíjí) vrátí:</p> <ul style="list-style-type: none"> ■ $AKTUALNI_STAV = \frac{BAT_VYKON_{BAT_ZMENA_VYKONU} + (DOBA_K_DOPOCTU * (-1) * BAT_VYKON_{BAT_ZMENA_VYKONU} * \eta_{chrg}) + ((DOBA_K_DOPOCTU * (-1) * (DOBA_K_DOPOCTU * BAT_ZMENA_VYKONU_GRADIENT) * \eta_{chrg})/2)}{2}$ <p>Pokud $BAT_VYKON = 0$ vrátí:</p> <ul style="list-style-type: none"> ■ $AKTUALNI_STAV = BAT_VYKON_{BAT_ZMENA_VYKONU}$ <p>Pokud $BAT_VYKON > 0$ (vybíjí) vrátí:</p> <ul style="list-style-type: none"> ■ $AKTUALNI_STAV = \frac{BAT_VYKON_{BAT_ZMENA_VYKONU} + (DOBA_K_DOPOCTU * (-1) * BAT_VYKON_{BAT_ZMENA_VYKONU} * \eta_{dsch}) + ((DOBA_K_DOPOCTU * (-1) * (DOBA_K_DOPOCTU * BAT_ZMENA_VYKONU_GRADIENT) * \eta_{dsch})/2)}{2}$
Vstup	<ul style="list-style-type: none"> ■ BAT_ZMENA_VYKONU ■ $BAT_VYKON_{BAT_ZMENA_VYKONU}$ ■ $BAT_STAV_{BAT_ZMENA_VYKONU}$ ■ $BAT_ZMENA_VYKONU_GRADIENT$ ■ η_{chrg} ■ η_{dsch}
Výstup	<ul style="list-style-type: none"> ■ $AKTUALNI_STAV$

#BAT4	Aktuální maximálně možné množství energie k uložení
Účel	Vrátí množství energie, které lze do baterie vložit do maximální kapacity. Energie surová, která se vkládá před přepočtem nabíjecí účinnosti. V požadavcích uvažujeme jako funkci $BAT_MAX_PRO_ULOZENI()$.
Akce	<p>Vrátí:</p> <ul style="list-style-type: none"> ■ $MNOZSTVI_ENERGIE = (E_{max} - BAT_AKTUALNI_STAV()[\#BAT3])/\eta_{chrg}$
Vstup	
Výstup	<ul style="list-style-type: none"> ■ $MNOZSTVI_ENERGIE$

#BAT5	Aktuální maximálně možné množství energie k dodání
Účel	Vrátí množství energie, které lze z baterie dodat do vyčerpání na minimální kapacitu. Energie, kterou lze použít je po započtení vybíjecí účinnosti. V požadavcích uvažujeme jako funkci <code>BAT_MAX_PRO_DODAVKU()</code> .
Akce	Vrátí: <ul style="list-style-type: none"> ■ $MNOZSTVI_ENERGIE = (BAT_AKTUALNI_STAV()[\#BAT3] - E_{min})/\eta_{dSch}$
Vstup	
Výstup	<ul style="list-style-type: none"> ■ $MNOZSTVI_ENERGIE$

#BAT6	ZMĚNA VÝKONU na příští vteřinu
Účel	Umožní nastavit výkon baterie a tím měnit stav – nabíjí, vybíjí, neutrální. Kladný výkon dodává ven z baterie – vybíjí, záporný výkon ukládá do baterie – nabíjí. Dynamika změny výkonu v příští vteřině se určuje přes gradient. V požadavcích uvažujeme jako funkci <code>NA-STAV_VYKON_BATERIE(požadovaný_výkon, požadovaný_gradient)</code> . Pro správné fungování se předpokládá volání každou vteřinu, aby fungovalo hlídání limitů

Pokračování na další stránce

Pokračování požadavku

Akce	<p>Pokud je baterie povolována na změnu výkonu, zkontrolují se limity na výkon.</p> <ul style="list-style-type: none"> ■ Pro $POZADAVEK_VYKON > 0$ (vybíjí), tak $POZADAVEK_VYKON + POZADAVEK_GRADIENT_VYKONU \leq P_{max}^{dsch}, BAT_AKTUALNI_STAV()[\#BAT5] \geq E_{min}$ ■ Pro $POZADAVEK_VYKON < 0$ (nabíjí), tak $POZADAVEK_VYKON + POZADAVEK_GRADIENT_VYKONU \leq P_{max}^{chrg}, BAT_AKTUALNI_STAV()[\#BAT5] \leq E_{max}$ <p>Nastaví:</p> <ul style="list-style-type: none"> ■ $BAT_STAV_t = BAT_AKTUALNI_STAV()[\#BAT5]$ ■ $BAT_VYKON_t = NOVY_VYKON = POZADAVEK_VYKON$ ■ $BAT_ZMENA_VYKONU_GRADIENT = NOVY_GRADIENT = POZADAVEK_GRADIENT_VYKONU$ ■ $BAT_ZMENA_VYKONU_GRADIENT = t$ <p>Pokud vybíjíme $POZADAVEK_VYKON > 0$, musíme ověřit, zda v příští sekundě máme dostatek potřebné energie pro určený výkon a gradient:</p> <ul style="list-style-type: none"> ■ $(POZADAVEK_VYKON + (POZADAVEK_GRADIENT_VYKONU/2))/3600 < BAT_MAX_PRO_DODAVKU()[\#BAT5]$ ■ Pokud nerovnost neplatí, musíme dopočítat gradient a výkon tak, aby v příští sekundě baterie dosáhla minima. ■ Zavedeme $DOSTUPNA_E = BAT_MAX_PRO_DODAVKU()[\#BAT5] * 3600$ ■ Pokud $DOSTUPNA_E = 0$: <ul style="list-style-type: none"> ■ Upravíme $NOVY_VYKON = 0, NOVY_GRADIENT = 0$ ■ Pokud $POZADAVEK_VYKON/2 \leq DOSTUPNA_E$: <ul style="list-style-type: none"> ■ Upravíme $NOVY_GRADIENT = 2 * (DOSTUPNA_E - POZADAVEK_VYKON)$ ■ Jinak: <ul style="list-style-type: none"> ■ Upravíme $NOVY_GRADIENT = 2 * DOSTUPNA_E, NOVY_GRADIENT = -2 * DOSTUPNA_E$ <p>Pokud nabíjíme $POZADAVEK_VYKON < 0$, musíme ověřit, zda v příští sekundě můžeme uložit přebytek energie pro určený výkon a gradient:</p> <ul style="list-style-type: none"> ■ $(POZADAVEK_VYKON + (POZADAVEK_GRADIENT_VYKONU/2))/3600 < BAT_MAX_PRO_ULOZENI()[\#BAT4]$ ■ Pokud nerovnost neplatí, musíme dopočítat gradient a výkon tak, aby v příští sekundě baterie dosáhla minima. ■ Zavedeme $DOSTUPNA_KAPACITA = BAT_MAX_PRO_ULOZENI()[\#BAT4] * 3600$ ■ Pokud $DOSTUPNA_KAPACITA = 0$: <ul style="list-style-type: none"> ■ Upravíme $NOVY_VYKON = 0, NOVY_GRADIENT = 0$ ■ Pokud $POZADAVEK_VYKON \leq 2 * DOSTUPNA_KAPACITA$: <ul style="list-style-type: none"> ■ Upravíme $NOVY_GRADIENT = -2 * (DOSTUPNA_KAPACITA - POZADAVEK_VYKON)$ ■ Jinak: <ul style="list-style-type: none"> ■ Upravíme $NOVY_GRADIENT = -2 * DOSTUPNA_KAPACITA, NOVY_GRADIENT = 2 * DOSTUPNA_KAPACITA$ <p style="text-align: center;">49</p> <p>Pokud byl nastaven $NOVY_VYKON$, nebo $NOVY_GRADIENT$, upravíme příslušnou charakteristiku baterie a nové hodnoty vrátíme.</p>
------	---

Pokračování na další stránce

Vstup	<ul style="list-style-type: none"> ■ <i>POZADAVEK_VYKON</i> ■ <i>POZADAVEK_GRADIENT_VYKONU</i>
Výstup	<ul style="list-style-type: none"> ■ <i>BAT_VYKON_i</i> ■ <i>BAT_STAV_i</i> ■ <i>BAT_ZMENA_VYKONU_GRADIENT</i> ■ <i>NOVY_VYKON</i> ■ <i>NOVY_GRADIENT</i>

7.1.3 Požadavky simulace

Popsanou reprezentaci technologií doplníme požadavky zabývající se dalšími aspekty v simulaci. Zabývají se inicializací dle vstupní konfigurace simulace, podoby povelů na turbíny a jejich zpracování, záznamem hodnot, které budeme chtít sledovat, poskytováním požadavků na služby atp. Popis navazuje na poskytnuté vstupy – vstupní konfigurace (zaveden0 tabulkou 7.1), data o službách (popsáno kapitolou 6.3), konfigurace stavového diagramu, který obsahuje popis stavů a hran, který jsme uvedli tabulkou 6.1, 6.3. Požadavky jsou rozepsány v následujících tabulkách.

#SIM1	Výchozí stav turbín v simulaci
Účel	Na základě konfiguračního souboru jsou jednotlivé turbíny na počátku simulace uvedeny do jejich výchozího stavu. Výchozí stav se řídí podle hodnoty v poli $B_{power}[]$ pro každou turbínu.
Akce	<p>Na počátku je vytvořeno N turbín, kde N je popsáno pomocí konfiguračního parametru B_{count}. Každá turbína i je inicializována na základě $B_{power}[i]$ z pole hodnot.</p> <ul style="list-style-type: none"> ■ Pokud je hodnota v rozsahu větší nebo rovno minimální technický výkon P_{min} nastaví do <i>POWER</i> stavu: <ul style="list-style-type: none"> ■ Realizováno akcí požadavku #TUR4, $B_{power}[i]$ jako <i>POCATECNI_VYKON</i> ■ Pokud je hodnota v rozmezí $[0, P_{min})$ je turbína i uvedena do <i>READY</i> stavu: <ul style="list-style-type: none"> ■ Realizováno akcí požadavku #TUR1 ■ Pokud je hodnota záporná, nebo neuvedena, je turbína i uvedena do <i>OFF</i> stavu: <ul style="list-style-type: none"> ■ Realizováno akcí požadavku #TUR1
Vstup	<ul style="list-style-type: none"> ■ B_{count} ■ $B_{power}[]$
Výstup	

#SIM2	Výchozí stav baterie
Účel	Na základě konfiguračního souboru na počátku simulace je nastaven výchozí stav baterie.
Akce	Inicializace je realizována akcí požadavku #BAT1
Vstup	Konfigurační soubor
Výstup	Inicializované proměnné
#SIM3	Nízkoúrovňové povely
Účel	Turbína umožňuje volat nízkoúrovňové povely vycházející ze stavového diagramu. Lze vytvořit základní sekvenci povelů pro účely testování a ověření bude možné skrze soubor povelů (formát viz. tabulka 1), který čte a vykonává manuální řízení.
Akce	<p>Povel je určen následující strukturou:</p> <ul style="list-style-type: none"> ■ t – čas povelu ■ i – index turbíny (indexováno od 1) ■ název povelu – např. A, B, \dots ■ Pro regulační povel následují doplňující parametry P – požadovaný výkon, G – gradient změny výkonu <p>Veškeré povely uvedené v souboru se načtou a v odpovídajícím čase jsou uvolněny do simulace skrze rozhraní řízení. [zpracování pomocí #TUR9, regulační povel #TUR7]</p>
Vstup	
Výstup	<ul style="list-style-type: none"> ■ Log: Povel
#SIM4	Vysokoúrovňový povel <i>OFF</i>
Účel	Pro řízení s nižší granularitou lze použít abstraktnější povel, který je zkratkou za případnou skupinu povelů. Povel <i>OFF</i> převede turbínu <i>READY</i> stavu v minimálním čase. Ukázka zápisu viz. tabulka 2.
Akce	<p>Povel je určen následující strukturou:</p> <ul style="list-style-type: none"> ■ t – čas povelu ■ i – index turbíny (indexováno od 1) ■ <i>OFF</i> ■ Rozšiřující doplňující parametry G – gradient (v <i>POWER</i> stavu nejdříve sníží výkon) <p>Provede:</p> <ul style="list-style-type: none"> ■ na základě v jakém stavu se turbína nachází, vyhledá ve vypočítané mapě cest příslušnou cestu vedoucí do <i>READY</i> stavu a její sekvenci příkazů [způsob výpočtu a struktura mapy v #SIM10] ■ naplánuje vykonání všech povel, dle předem vypočtených časů aktivace povelu [zpracování pomocí #TUR9]
Vstup	
Výstup	<ul style="list-style-type: none"> ■ Log: Povel <i>OFF</i>

#SIM5	Vysokoúrovňový povel <i>ON</i>
Účel	Pro řízení s nižší granularitou lze použít abstraktnější povel, který je zkratkou za případnou skupinu povelů. Povel <i>ON</i> převede turbínu <i>POWER</i> stavu v minimálním čase. Ukázka zápisu viz. tabulka 2.
Akce	<p>Povel je určen následující strukturou:</p> <ul style="list-style-type: none"> ■ t – čas povelu ■ i – index turbíny (indexováno od 1) ■ <i>ON</i> ■ Rozšiřující doplňující parametry P – požadovaný výkon, G – gradient (po vstoupení do <i>POWER</i> provádí změnu výkonu) <p>Provede:</p> <ul style="list-style-type: none"> ■ na základě v jakém stavu se turbína nachází, vyhledá ve vypočítané mapě cest příslušnou cestu vedoucí do <i>POWER</i> stavu a její sekvenci příkazů [způsob výpočtu a struktura mapy v [#SIM10] ■ naplánuje vykonání všech povelů dle předem vypočtených časů aktivace povelu [zpracování pomocí #TUR9]
Vstup	
Výstup	<ul style="list-style-type: none"> ■ Log: Povel <i>ON</i>

#SIM6	Vysokoúrovňový povel <i>POWER</i>
Účel	Pro řízení s nižší granularitou lze použít abstraktnější povely, které jsou zkratkou za skupinu povelů. Povel <i>POWER</i> provede změnu výkonu v <i>POWER</i> stavu. Ukázka zápisu viz. tabulka 2.
Akce	<p>Povel je určen následující strukturou:</p> <ul style="list-style-type: none"> ■ t – čas povelu ■ i – index turbíny (indexováno od 1) ■ <i>POWER</i> ■ P – požadovaný výkon ■ G – gradient změny výkonu <p>Provede:</p> <ul style="list-style-type: none"> ■ v čase t regulační povel s parametry P a G [regulační povel v #TUR7]
Vstup	
Výstup	<ul style="list-style-type: none"> ■ Log: Povel <i>POWER</i>

#SIM7	Logování
Účel	Během simulace bude docházet k logování hodnot a událostí během simulace pro možný výstup různých charakteristik.
Akce	<p>Data ukládány ve dvou režimech</p> <ul style="list-style-type: none"> ■ body na křivce – tzn. mezi body na grafu může být křivka, každý bod na křivce představuje hodnotu v daném čase ■ změna hodnoty – tzn. bod reprezentuje změnu, hodnota trvá do další změny, v grafu v čase t je hodnota reprezentována konstantní vodorovnou křivkou <p>Bude probíhat záznam následujících hodnot (kdy a jakých je uvedeno v rámci příslušných požadavků jako výstup):</p> <ul style="list-style-type: none"> ■ $spotreba_plynu_{ti}$ – (body na křivce) ■ $spotreba_energie_{ti}$ – (body na křivce) ■ $stav_{ti}$ – (změna hodnoty) ■ $vykon_{ti}$ – (body na křivce) ■ $gradient_{ti}$ – (změna hodnoty) ■ $vykon_baterie_t$ – (body na křivce) ■ $stav_baterie_t$ – (body na křivce) ■ FCR_t – (body na křivce) ■ $aFRR_t$ – (body na křivce) ■ $mFRR_t$ – (body na křivce) ■ P_t^{AGR} – (změna hodnoty) ■ $poveley_{ti}$ – (změna hodnoty) ■ chybové události
Vstup	
Výstup	Data sledovaných metrik
#SIM8	Požadavky SVR
Účel	Na základě vstupního souboru o službách, který obsahuje jednotlivé struktury požadavků na podpůrné služby po řádcích (viz. kapitola 6.3), ve výchozím čase struktury zpřístupní data simulaci.
Akce	<ul style="list-style-type: none"> ■ Načte vstupní data ■ Pro každou strukturu vytvoří událost s časem $t = vychozi_cas$, v uvedený čas se data zpřístupní simulaci a řízení rovněž. Provede se log požadavku na jednotlivé služby.
Vstup	<ul style="list-style-type: none"> ■ SVR (soubor s daty)
Výstup	<ul style="list-style-type: none"> ■ Události poskytující data o požadavcích

#SIM9	Validace konfigurace dynamického stavového diagramu
Účel	Vstupní soubor obsahující konfiguraci stavového diagramu turbíny pro správnou funkci musí splňovat určité nároky na omezení. Před spuštěním simulace konfigurace bude dodržení omezení zkontrolováno.
Akce	<ul style="list-style-type: none"> ■ Načte vstupní data ■ Ověří pravidla: <ul style="list-style-type: none"> ■ Napříč stavy existuje pouze jeden <i>POWER</i> stav ■ Přejechod, který má uvedený timeout nesmí vycházet z <i>POWER</i> stavu ■ Z každého stavu může vycházet maximálně 1 přechod, který má uvedený timeout ■ V případě porušení nespustí simulaci a předá informaci uživateli, co bylo porušeno
Vstup	<ul style="list-style-type: none"> ■ Stavový diagram turbíny (soubor s daty)
Výstup	
#SIM10	Napočítání cest pro povel <i>ON/OFF</i> pro dané stavy
Účel	Před zahájením simulace při její inicializaci analyzuje vstupní dynamický stavový diagram. Pro každý stav provede hledání nejkratší cesty z hlediska času, při povelu <i>ON</i> do <i>POWER</i> stavu a povelu <i>OFF</i> do <i>READY</i> stavu. Uloží do paměti sadu povelů spolu s informací, v jakém čase je třeba spustit, rovněž celkový čas dosažení požadovaného stavu.
Akce	<ul style="list-style-type: none"> ■ Načte vstupní dynamický stavový diagram ■ Provede algoritmus pro hledání nejkratší cesty ■ Pro každý stav <i>S</i>, uloží vypočtenou vzdálenost do <i>POWER</i> stavu <i>P</i> a zrekonstruuje cestu <i>S</i>-><i>P</i>, kde pokud použitá hrana představuje povel (není timeout hranou), zaznamená název povelu a čas jeho použití, stejně provede i pro nalezení cesty do <i>READY</i> stavu (pozn. cesta nemusí existovat, v tom případě příslušný povel <i>ON/OFF</i> nelze volat) ■ Vznikne mapa, náčrt datové struktury: <ul style="list-style-type: none"> ■ Stav <i>S</i> -> {"ON" -> {"commands": [{"A", 0}, {"B", 30}, ...], "time": 50}, "OFF" -> {"commands": [{"D", 0}, {"E", 10}, [], ...], "time": 30}} ■ Pozn. jednotky času v s ■ Pokud cesta není, pole jsou prázdná, tak stav <i>S</i> -> {"ON" -> {"commands": null, "time": ∞}, "OFF" -> {"commands": null, "time": ∞}}
Vstup	<ul style="list-style-type: none"> ■ Stavový diagram turbíny (soubor s daty)
Výstup	<ul style="list-style-type: none"> ■ Mapa v paměti pro každý stav a povel <i>ON</i>, <i>OFF</i>

■ 7.1.4 UI aplikace

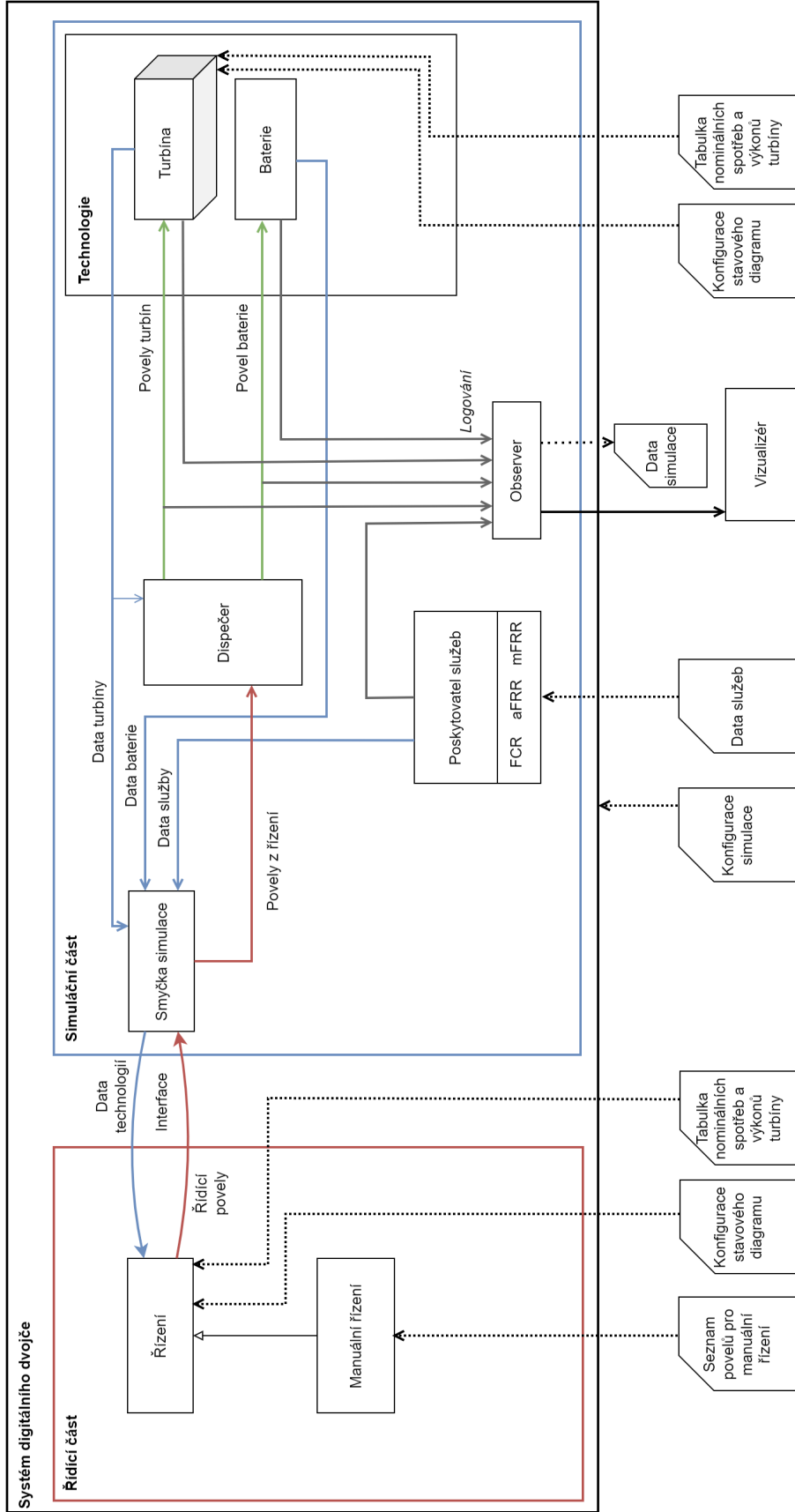
Uživatelské rozhraní systému není nijak komplexní, výstup dat z digitálního dvojče bude proveden pomocí grafů. Požadavek #UI1 uvádí výčet požadovaných grafů.

#UI1	Výstupy grafů simulace
Účel	V rámci programu bude možné provést simulaci, po ukončení simulace dojde k vyhodnocení dat a jejich vykreslení do grafů.
Akce	<p>Grafy — výčet, jaká data budou v daném grafu reprezentována</p> <ul style="list-style-type: none"> ■ Graf výkonů – výkony po jednotlivých turbínách a rovněž křivka jejich součtu, barevně oddělené za turbíny ■ Graf služeb – požadavky na výkon po jednotlivých službách i jejich součet, barevně oddělené pro jednotlivé služby ■ Bilance požadavku a dodávky – agregovaný požadavek na plnění služeb, součtový z výkon turbín a výkon baterie, skutečná dodávka (složená z výkonu turbín a baterie), 0/1 křivka, zda docházení ke správnému plnění, úroveň nabití baterie ■ Spotřeba elektrické energie po turbínách a v součtu ■ Spotřeba plynu po turbínách a v součtu
Vstup	
Výstup	

7.2 Architektura

Výše zmíněné požadavky doplníme architektonickou koncepcí systému, jejíž grafické zpracování si můžeme prohlédnout na obrázku 7.1. Architekturu systému digitálního dvojče můžeme logicky rozdělit na dvě hlavní části - simulační a řídicí část. Simulační část má na starosti načtení vstupů, inicializaci všech objektů spolu s předáním potřebných vstupních dat, následně obsluhu běhu celé simulace včetně aktualizace objektů představující technologie (turbíny, baterie), průběžné logování potřebných dat. Druhou částí je řízení, které vnáší život do celého systému, aby zařízení pracovalo – tj. produkovalo výstupní výkon ze zařízení pomocí turbín a baterie podle předepsaného žádaného výkonu.

Vstupem digitálního dvojče jsou soubory – požadavky služeb, konfigurace, popis stavového diagramu, tabulky nominálních hodnot výkonů a spotřeb, povely pro manuální řízení. Data služeb (ve formátu z kapitoly 6.3) jsou všechna předem načtena a příslušné povely na služby rozvrženy na příslušné časy, kdy mají být v simulaci uvolněny. Pro udržování informace, jaký je aktuální žádaný výkon pro všechny služby a jaký byl poslední požadavek na služby popsáný průběhem žádaného výkonu slouží komponenta nazvaná poskytovatel služeb. Stav poskytovatele služeb je aktualizován s každým příchodem nového povelu služby (jeho uvolnění simulaci probíhá dle požadavku #SIM8). Konfigurace neboli konfigurační soubor (ve formátu JSON) obsahuje různé potřebné parametry pro nastavení digitálního dvojče a chování komponent technologií. Obsahuje parametry turbín, baterie a simulace, které byly popsány tabulkami 6.5, 6.6, 6.8. Komponenta turbína funguje dle zavedených požadavků #TUR1–#TUR11. Pro své fungování potřebuje stavový diagram, který je popsán v souboru konfigurace stavového diagramu. Pro zjištění maximálního výkonu a spotřeb ve stavech určených variabilní spotřebou využije soubor s tabulkou nominálních spotřeb a výkonů turbíny. Komponenta baterie funguje dle popisu požadavků #BAT1–#BAT6. Komponenta dispečera umožňuje delegovat a realizovat požadavky pro turbíny, které obdrží ve formátu popsáném požadavky #SIM3–#SIM6. Obdržené povely převádí na volání funkcí, které turbína poskytuje. Pro vysokoúrovňové povely zabezpečuje navíc logiku, aby došlo k zabezpečení efektu dle návrhu (např. povel *ON* s cílovým výkonem – provede start a po jeho dokončení zahájí změnu výkonu). Povely pro baterku jsou předávány na komponentu baterky bez další logiky. Poslední významnou komponentou v simulační části je tzv. smyčka simulace. Její funkčnost je zahájena po inicializaci všech komponent. K její aktivaci dochází každou vteřinu (v čase simulace) od počátku až do konce simulace. Její roli uvedeme na následovně.



Obrázek 7.1: Architektura digitálního dvojče systému

V každém vykonání provede sběr aktuálních dat (z turbín, baterie a poskytovatele služeb), vytvoří z nich vstup pro řízení jehož strukturu popisuje tabulka 6.9 a intenzivně jsme se ním zabývali v kapitole 6.5. Tato data jsou předána řízení a to má příležitost provést své potřebné výpočty a zpět vrátí řídicí povely. Tyto povely jsou poté v simulační části předány dispečerovi, který zabezpečí jejich realizaci na technologiích. Celý průběh můžeme sledovat na obrázku 7.1, sběr dat a jejich předání řízení ilustrují modré šipky, zpětné předání řídicích povelů do simulační části a následně dispečerovi ilustrují červené šipky. Zbývající zelené šipky ilustrují interpretaci povelů na technologii. Napříč těmito procesy probíhá sběr dat do komponenty observer v duchu požadavku #SIM7.

Řídicí a simulační část mají mezi sebou pevně určené rozhraní, které je navrženo striktně, aby mezi sebou nemohli interagovat jinak než přes toto určené datové rozhraní. Počítá se, že bude možné implementovat více druhů řízení bez potřeby velkých zásahů do systému. Drobný zásah bude potřeba jen pro vytvoření nové třídy, která bude dědit ze společného rozhraní řízení a zavedení schopnosti rozpoznat unikátní řetězec, při jehož uvedení v konfiguraci, dojde k nasazení do běhu odpovídající implementace. Předpokládá se, že řídicí třídu může implementovat někdo, kdo nezná celý rozsah simulační části a nemusí potřebovat do ní zasahovat. Rozhraní řídicí třídy je definováno pomocí konstruktoru, který není z datového pohledu tak striktní, jsou v něm předány sdílené reference na datové objekty konfigurací přístupné i simulační části. Předpokládá se, že řídicí část z těchto struktur bude pouze číst a aspekty možného ovlivnění, zde byly upozaděny s tím, že vývojář řízení bude zodpovědný a nebude škodit (vždy někdo spřízněný s projektem výzkumu). Rozhraní řídicí třídy rovněž obsahuje metodu pro výpočet řídicích povelů, která je volána každou vteřinu v čase simulace uvedenou komponentou smyčka simulace. Na základě vstupu je úkolem poskytnout řídicí povely. Pro účely vydání povelu jsou vytvořeny třídy, které umožňují popsat povel v rozsahu požadavků #SIM3–#SIM6. Výstupem je tedy seznam povelů určených turbínám, které mají okamžitou platnost. Jejich předání k turbínám a realizace probíhá bez prodlení. Navíc je předán řídicí povel pro baterii.

V této práci bylo do tohoto konceptu implementováno manuální řízení, které funguje následovně. V rámci konstrukturu dojde ke čtení souboru obsahující seznam povelů pro manuální řízení. Tento soubor se nachází ve složce, která je předána jako argument programu digitálního dvojče. Tyto povely si manuální řízení uloží do paměti. Jakmile je volána funkce pro poskytnutí aktuálních povelů, ze vstupních dat se vezme čas v simulaci a z paměti se vyberou všechny povely odpovídající tomuto času. Tyto povely jsou vydány zpět simulační části a odstraněny z paměti. Zároveň je potřeba dodat povel pro baterii. Zde je použita strategie udržení rovnováhy. Povel na baterii je dopočítán tak, aby kompenzoval rozdíl mezi výkonem turbín a žádaným výkonem služby. Baterie tak může dodávat výkon, nabíjet nebo zcela nic, záleží na situaci vstupu. Povel může být vydán i mimo rozsah maximálního možného výkonu, nebo nemusí stačit energie v baterii. V tom případě zafunguje přímo návrh komponenty baterie a je realizován výsledný povel dle aktuálních možností.

To znamená, pokud chceme vyšší výkon než baterie umí poskytnout, poskytne jen maximální. Chceme poskytnout výkon, ale v baterce není energie, není dodán žádný výkon. V rámci těchto situací pak typicky dochází k neplnění požadavku na výkon. Na tyto možnosti je brán zřetel v rámci požadavku týkající se baterie #BAT6.

Poslední komponentu, kterou zmíníme je vizualizace. Po ukončení simulace zvoleného období se data uložená v observeru vezmou a vytvoří se grafy popsané požadavkem #UI1 a dojde k jejich uložení na souborový systém ve formátu .html[29]. Jejich zobrazení je pak možné v internetovém prohlížeči. Zároveň dojde k uložení logu ze systému jehož příklad spolu s grafickými výstupy si ukážeme v kapitole 8 zabývající se experimenty.

7.3 Zvolené technologie

Výběr technologií byl zcela přímočarý. Jako programovací jazyk byl zvolen Python[30] především díky existenci knihovny pro tvorbu simulací SimPy. Tato knihovna byla snadná na pochopení jejich konceptu a dobře aplikovatelná na náš problém. Jazyk python v obecnosti je jednoduchý a rychlý na použití, umožňuje použití opravdu velkého množství knihoven a je velice populární díky své flexibilitě využití v mnoha oblastech. Naproti tomu je správné zmínit, že Python není nejrychlejší jazyk a tento fakt jsme při volbě měli na zřetel. Nicméně v případě potřeby zrychlení je možné zapojit a využít rozšíření napsaná v jazyce C a C++. Řada známých knihoven je rovnou takto implementována – NumPy, Pandas, Tensorflow... Rovněž existuje možnost využít just-in-time kompilace pomocí kompilátoru Numba[31], kde stačí nad částí kódu přidat anotace. Poté místo interpretace částí kódu standardním způsobem, při spuštění dojde ke kompilaci zvoleného kódu a jeho vykonání je poté rychlejší. Na závěr uvedeme, že v rámci této práce nebylo potřeba využít dodatečných optimalizací rychlosti. Rychlost výpočtu výsledku vzhledem k simulovaným časovým horizontům byla dostatečná našim potřebám. V kapitole 8 zabývající se experimenty se rychlostním srovnáním budeme zabývat.

SimPy je knihovna pro diskretně-událostní simulace, která využívá generátorů v Pythonu. Pomocí klíčového slova *yield* v kombinaci událostí s *timeoutem* (časovým určením za jak dlouho má dojít ke zpracování události), dojde k přerušení vykonávání, událost se uloží do fronty do doby než nastane požadovaný čas a běh kódu se vrátí do místa, odkud událost byla vyvolána. Aby bylo možné provádět více čekání najednou, lze vykonávání kódu větvit na procesy, mezi kterými jednovláknový běh různě přepíná dle vytvářených událostí a jejich zpracování. Pokud je více událostí plánováno na jeden časový okamžik, jejich zpracování probíhá v pořadí, jak události na daný čas vznikali (koncept FIFO). Na toto chování je potřeba brát zřetel, aby zpracování událostí a jejich efekty probíhali, jak potřebujeme. Pořadí ve frontě lze ovlivnit nastavením priority události. Poté na jeden časový okamžik vzniká více FIFO front, které jsou postupně zpracovávány dle priority. V rámci této koncepce je emulován čas skrze *float* čítač a lze vždy určit v jakém časovém okamžiku simulovaného času se nacházíme. V případě, že vytváříme proces,

můžeme si jeho referenci uložit a kdykoli v případě potřeby vykonávání toho procesu ukončit. Zmíněný popis fungování knihovny je úplným základem, jsou k dispozici i složitější koncepty, pro potřeby implementace systému však postačili tyto základy.

Jedním z požadavků byla možnost vizualizace vzniklých dat z digitálního dvojce do grafů. Pro tento účel byla zvolena knihovna plotly[32]. Umožňuje provést vykreslení grafu a jeho uložení do souboru ve formátu .html. Graf je poté možné zpětně kdykoliv načíst v internetovém prohlížeči a navíc graf si zachová interaktivitu jako například zobrazení určitého výseku dat ze spojnicového grafu, přiblížení, zobrazení jen některých křivek, zjištění hodnoty po najetí a další.

Jeden ze souborů jako vstup je ve formátu .xlsx – sešit aplikace Excel, pro potřeby načtení byla využita knihovna openpyxl[33]. Jak je vidět, nebylo potřeba využití velkého počtu technologií, při vývoji byl kladen důraz na řádný objektový návrh, využití definice datových typů pro vstupy, výstupy funkcí a atributy tříd, aby kód byl dobře čitelný. *Jazyk Python nebyl zpočátku typovaný, tento koncept byl přidán do jazyka později.*

Kapitola 8

Experimenty - ilustrace řešení

Nyní se budeme zabývat ukázkou vytvořeného digitálního dvojče. Na jednoduchém vstupu si ukážeme výslednou formu vstupních souborů a výstupy z digitálního dvojče. Budeme se zabývat tím, co se během simulace odehrálo a jak je to ve výsledcích prezentováno. Následně ukážeme výsledek chování na větším scénáři, který nebudeme studovat do hloubky, ale poslouží pro představu, jak může vypadat výstup při trochu realističtějším použití. Celou kapitolu uzavřeme experimentem zabývajícím se škálovatelností systému, uvidíme jak dlouho trvá výpočet v závislosti na délce simulace a počtu turbín v simulaci. Podotkneme, že veškeré vstupy jsou pouze ilustrační a nevychází z reálných hodnot, v některých případech jimi mohou být nejvýše inspirována. Při vytváření vstupů jsme se pokusili vytvořit takové, které dávají nějaký logický smysl. Nad digitálním dvojčetem probíhají reálné experimenty, avšak žádná data není možné uveřejnit. Veškeré uvedené experimenty byly provedeny na stolním počítači s operačním systémem Windows 10 Pro, procesorem Intel Core i5-760 s taktem 2.80 GHz, RAM 16 GB s frekvencí 1333 MHz, Python verze 3.10.

8.1 Ilustrace jednoduchého příkladu

Pro začátek si ukážeme krátkou simulaci na horizontu 20 minut, která bude obsahovat konfiguraci zařízení o třech turbínách. První bude zcela vypnutá, druhá bude inicializována do výkonového stavu a po celou dobu bude na stabilním výkonu, poslední turbínu budeme řídit pomocí manuálního řízení, aby byl vidět celý průběh turbíny skrze stavový diagram. Pro začátek představíme vstupy. Jako první představíme uvažované služby. Využijeme všech 3 služeb, služba mFRR bude od počátku na 4 MW a v 15. minutě (900s) dorazí požadavek na 0 MW. Pro aFRR budeme mít v 5. minutě (300s) požadavek na 2 MW a ve 12. minutě (720s) je aktivace odvolána na 0 MW (pozn. pro dobu plné aktivace uvažujeme 5 minut – 300s). Pro FCR máme demonstrativní ukázkou vteřinových požadavků mezi 10. a 30. vteřinou. V reálném provozu by požadavky na FCR zpravidla chodily každou vteřinu. Zápis zmíněného v souboru můžeme vidět ve výpisu 3.

1	2023-04-01 00:00:00							
2	mFRR	2	0	4	900	4		
3	FCR	2	10	0.000000	11	0.666672		
4	FCR	2	11	0.666672	12	0.716673		
5	FCR	2	12	0.716673	13	0.866673		
6	FCR	2	13	0.866673	14	1.016673		
7	FCR	2	14	1.016673	15	1.216673		
8	FCR	2	15	1.216673	16	1.416672		
9	FCR	2	16	1.416672	17	1.616673		
10	FCR	2	17	1.616673	18	1.716672		
11	FCR	2	18	1.716672	19	1.866672		
12	FCR	2	19	1.866672	20	1.966673		
13	FCR	2	20	1.966673	21	1.866672		
14	FCR	2	21	1.866672	22	1.716672		
15	FCR	2	22	1.716672	23	1.616673		
16	FCR	2	23	1.616673	24	1.416672		
17	FCR	2	24	1.416672	25	1.216673		
18	FCR	2	25	1.216673	26	1.016673		
19	FCR	2	26	1.016673	27	0.866673		
20	FCR	2	27	0.866673	28	0.716673		
21	FCR	2	28	0.716673	29	0.666672		
22	FCR	2	29	0.666672	30	0.000000		
23	aFRR	2	300	0	630	2		
24	aFRR	2	720	2	1020	0		
25	mFRR	3	900	4	1050	4	1800	0

Výpis 3: Ukázka vstupu podpůrných služeb ve formátu popsaném v kapitole 6.3

Pro popis turbíny poskytneme dva soubory. Stavový diagram zvolíme v podobě, kterou jsme si ukázali na obrázku 6.2 v kapitole 6.1. Tento diagram je popsán ve formátu JSON, datová reprezentace vychází z prezentovaných myšlenek, kterými jsme se zabývali v příslušné kapitole. Soubor obsahuje definici stavů (*states*) a hran (*edges*). Každý stav je popsán názvem (*name*), spotřebou energie (*ee*) a plynu (*gas*). Hrana je popsána výchozím (*from*) a cílovým (*to*) stavem, názvem příkazu (*command*) a v případě potřeby doplněno dobou *timeoutu* (*timeout*). Nadále jsou uvedeny významné stavy (*key_states*) pro určení *OFF*, *READY* a *POWER* stavu a významný povel pro regulaci výkonu (*key_command.regularory*). Popis diagramu si můžeme konkrétně prohlédnout ve výpisu 4.

```

1  {
2    "states": [
3      {"name": "S0", "ee": 0, "gas": 0},
4      {"name": "S1", "ee": 2, "gas": 0},
5      {"name": "S2", "ee": 10, "gas": 0},
6      {"name": "S3", "ee": 150, "gas": 0},
7      {"name": "S4", "ee": 50, "gas": "VAR_POWER"},
8      {"name": "S5", "ee": 30, "gas": "VAR_POWER"},
9      {"name": "S6", "ee": 25, "gas": "VAR_POWER"},
10     {"name": "S7", "ee": 25, "gas": "VAR_POWER"},
11     {"name": "S8", "ee": 40, "gas": 2},
12     {"name": "S9", "ee": 20, "gas": 1},
13     {"name": "S10", "ee": 60, "gas": "VAR_POWER"}
14   ],
15   "edges": [
16     {"from": "S0", "to": "S1", "command": "S0S1"},
17     {"from": "S1", "to": "S0", "command": "S1S0"},
18     {"from": "S1", "to": "S2", "command": "S1S2"},
19     {"from": "S2", "to": "S1", "command": "S2S1"},
20     {"from": "S2", "to": "S3", "command": "A"},
21     {"from": "S3", "to": "S4", "command": "TIMEOUT",
22     ↪ "timeout": 15},
23     {"from": "S4", "to": "S5", "command": "TIMEOUT",
24     ↪ "timeout": 30},
25     {"from": "S5", "to": "S6", "command": "B"},
26     {"from": "S5", "to": "S9", "command": "E"},
27     {"from": "S6", "to": "S7", "command": "TIMEOUT",
28     ↪ "timeout": 5},
29     {"from": "S7", "to": "S8", "command": "D"},
30     {"from": "S8", "to": "S6", "command": "B"},
31     {"from": "S8", "to": "S9", "command": "TIMEOUT",
32     ↪ "timeout": 60},
33     {"from": "S9", "to": "S10", "command": "F"},
34     {"from": "S10", "to": "S5", "command": "TIMEOUT",
35     ↪ "timeout": 20},
36     {"from": "S9", "to": "S2", "command": "TIMEOUT",
37     ↪ "timeout": 90}
38   ],
39   "key_states": {
40     "off": "S0",
41     "ready": "S2",
42     "power": "S7"
43   },
44   "key_command": {
45     "regulatory": "C"
46   }
47 }

```

Výpis 4: Ukázka vstupu stavového diagramu dle obrázku 6.2 ve formátu JSON

Druhý soubor obsahuje nominální hodnoty spotřeb a výkonů. Pro vstupní data využijeme již zmíněné hodnoty uvedené při popisu problematiky v kapitole 6.1. Vstupní soubor je ve formátu .xlsx (sešit programu Excel). Na prvním listu nazvaném POWER uvedeny nominální výkony dle tabulky 6.4, jak můžeme vidět i na obrázku 8.1, na druhém listu nazvaném CONSUMPTION jsou hodnoty nominálních spotřeb dle tabulky 6.2, jak můžeme vidět i na obrázku 8.2.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	POWER	0	10	20	30	40	50	60	70	80	90	100	
2		15	0	600	1 200	1 770	2 310	2 810	3 250	3 610	3 930	4 230	4 430
3		20	0	570	1 140	1 682	2 195	2 670	3 088	3 430	3 734	4 019	4 209
4		25	0	542	1 083	1 597	2 085	2 536	2 933	3 258	3 547	3 818	3 998
5													

Obrázek 8.1: Ukázka vstupu nominálních výkonů v souboru ve formátu .xlsx

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	POWER	0	10	20	30	40	50	60	70	80	90	100	
2		15	0	600	1 200	1 770	2 310	2 810	3 250	3 610	3 930	4 230	4 430
3		20	0	570	1 140	1 682	2 195	2 670	3 088	3 430	3 734	4 019	4 209
4		25	0	542	1 083	1 597	2 085	2 536	2 933	3 258	3 547	3 818	3 998
5													

Obrázek 8.2: Ukázka vstupu nominálních spotřeb plynu v souboru ve formátu .xlsx

Dalším vstupním soubor je tzv. konfigurace, která se zabývá popisem parametrů turbíny, baterie, simulace a nastavení digitálního dvojče. Problematikou jsme se zabývali v kapitolách 6.1, 6.2, 6.4. V následujícím výpisu na obrázku 5 je vyobrazena ukázka vstupního souboru. Parametry jsou rozděleny do tří oddílů. První se zabývá parametry turbíny uvedené tabulkou 6.5, druhý parametry baterky uvedené tabulkou 6.6, třetí parametry simulace a digitálního dvojče uvedené tabulkou 6.8. Nad rámeček formátu je ke každé proměnné doplněno pomocí komentáře označení parametru, jak bylo zavedeno. Jelikož není možné v kódu uvést horní a spodní indexy graficky, jsou popsány následovně. Spodní index následuje za znakem podtržítka ve složených závorkách, horní index za znakem stříšky ve složených závorkách.

```

1  {
2    "tur": {
3      "max_grad_up": 60, // P_{max-up}~{grad}
4      "max_grad_down": 60, // P_{max-down}~{grad}
5      "p_min": 120, // P_{min}
6      "max_grad_change": 60 // P_{max-change}~{grad}
7    },
8    "bat": {
9      "max_dchrg_pwr": 6, // P_{max}~{dsch}
10     "max_chrg_pwr": 6, // P_{max}~{chrg}
11     "min_cap": 1, // E_{min}
12     "max_cap": 6, // E_{max}
13     "eff_chrg": 95, // eta_{chrg} -- účinnost
14     "eff_dchrg": 92 // eta_{dsch} -- účinnost
15   },
16   "sim": {
17     "agt_count": 3, // B_{count}
18     "agt_power": [-1, 4000, 0], // B_{power}[]
19     "bat_energy": 3, // E_{init}
20     "time_start": "2023-04-01 00:00:00", // T_{start}~{sim}
21     "time_end": "2023-04-01 00:20:00", // T_{end}~{sim}
22     "temperature_default": 20.0, // Temp_{sim}
23     "ee_price": 6600, // Cost_{EE}
24     "gas_price": 2500, // Cost_{GAS}
25     "control": "manual" // výběr řízení -- manuální
26   }
27 }

```

Výpis 5: Ukázka vstupu konfigurace ve formátu JSON

Posledním souborem, kterým vstupy uzavřeme je seznam povelů manuálního řízení. Podobu vstupu můžeme vidět na obrázku 6. U turbíny zahájíme start ve 4. minutě a později vydáme povel *B*. Ve výkonovém stavu v 5. minutě navyšujeme výkon na 2 MW, ve 12. minutě snižujeme na 1 MW a v 15. minutě snižujeme na minimální výkon pro odpojení. Odpojení je zahájeno povel *D* v 15. minutě a 20. vteřině. V průběhu vypínací sekvence se rozhodneme vydat

povel F pro restart, ale později se rozhodneme restart ukončit povel E a turbína dorazí nakonec do *READY* stavu.

```

1  {
2    "commands": [
3      ["2023-04-01 00:04:00", 3, "A"],
4      ["2023-04-01 00:04:50", 3, "B"],
5      ["2023-04-01 00:05:00", 3, "C", 2000, 20],
6      ["2023-04-01 00:12:00", 3, "C", 1000, -20],
7      ["2023-04-01 00:15:00", 3, "C", 120, -60],
8      ["2023-04-01 00:15:20", 3, "D"],
9      ["2023-04-01 00:16:30", 3, "F"],
10     ["2023-04-01 00:17:00", 3, "E"]
11   ]
12 }
```

Výpis 6: Ukázka vstupu povelů turbíně pro manuální řízení dle kapitoly 6.5.

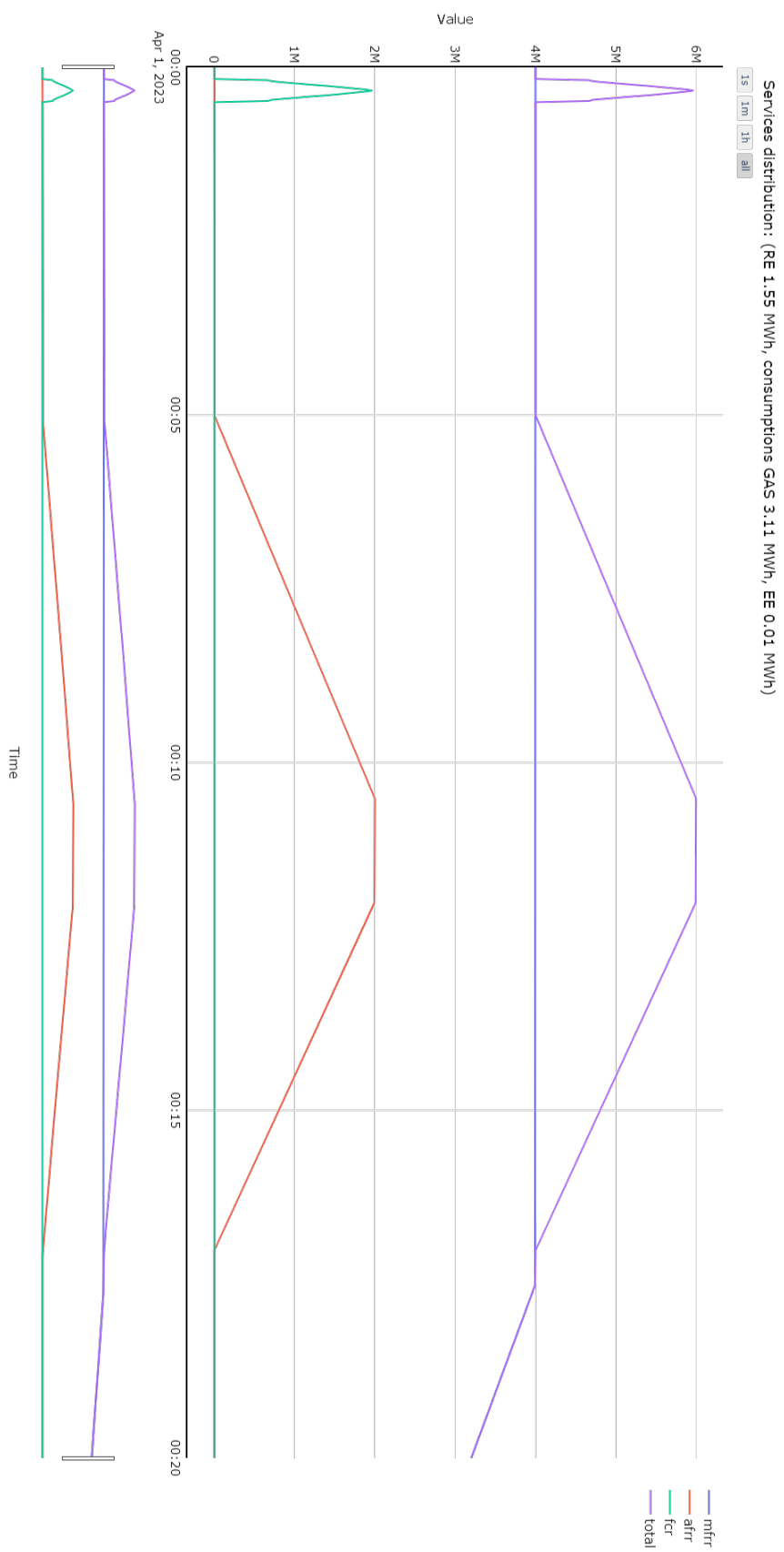
Popsali jsme všechny vstupní soubory a nyní se podíváme podobu výstupních diagramů z digitálního dvojče. Na obrázku 8.3 zabývající se zobrazením průběhu služeb, můžeme vidět službu FCR znázorněnou zelenou křivkou. Na počátku máme krátký intenzivní záchvěv skoro do 2 MW. Pro služba aFRR znázorněnou červenou vidíme táhlý náběh od 5. minuty a po chvíli ve stejném tempu sestup. Služba mFRR znázorněna modrou barvou je od počátku na 4 MW a v 15. minutě začíná klesat. Součet výkonu všech služeb je znázorněn fialovou barvou.

Graf zabývající se výkony jednotlivých turbín je znázorněn na obrázku 8.4. Máme zde 3 turbíny, modrou křivkou je znázorněna turbína 1, která je neaktivní. Turbína 2 znázorněna červenou křivkou je na stabilním výkonu 4 MW. Turbína 3 znázorněna zelenou nejdříve nedodává výkon, protože byla v *READY* stavu a poté jsme prováděli její start. Ve 4. minutě a 50. vteřině začíná turbína dodávat výkon, vidíme skok na minimální technický výkon. Po chvíli je vydán povel a výkon začíná růst. Ve 12. minutě byl vydán povel ke snížení výkonu a my vidíme první pokles na 1 MW, kde nějakou dobu setrvává. V 15. minutě opět snižujeme výkon na minimální technický a v 15. minutě a 20. vteřině vydáváme povel k vypnutí a turbína přestává dodávat výkon, což vidíme skokem na 0 MW. V grafu jsou ještě vykresleny dvě skupiny bodů, kde jedna je pro vydané povelů a druhá pro změny stavů na časové ose. Výstupní soubor po najetí umožňuje zobrazit informace. V této podobě jsme o tuto informaci ochuzeni, bude doplněna výpisem z logu běhu digitálního dvojče na závěr.

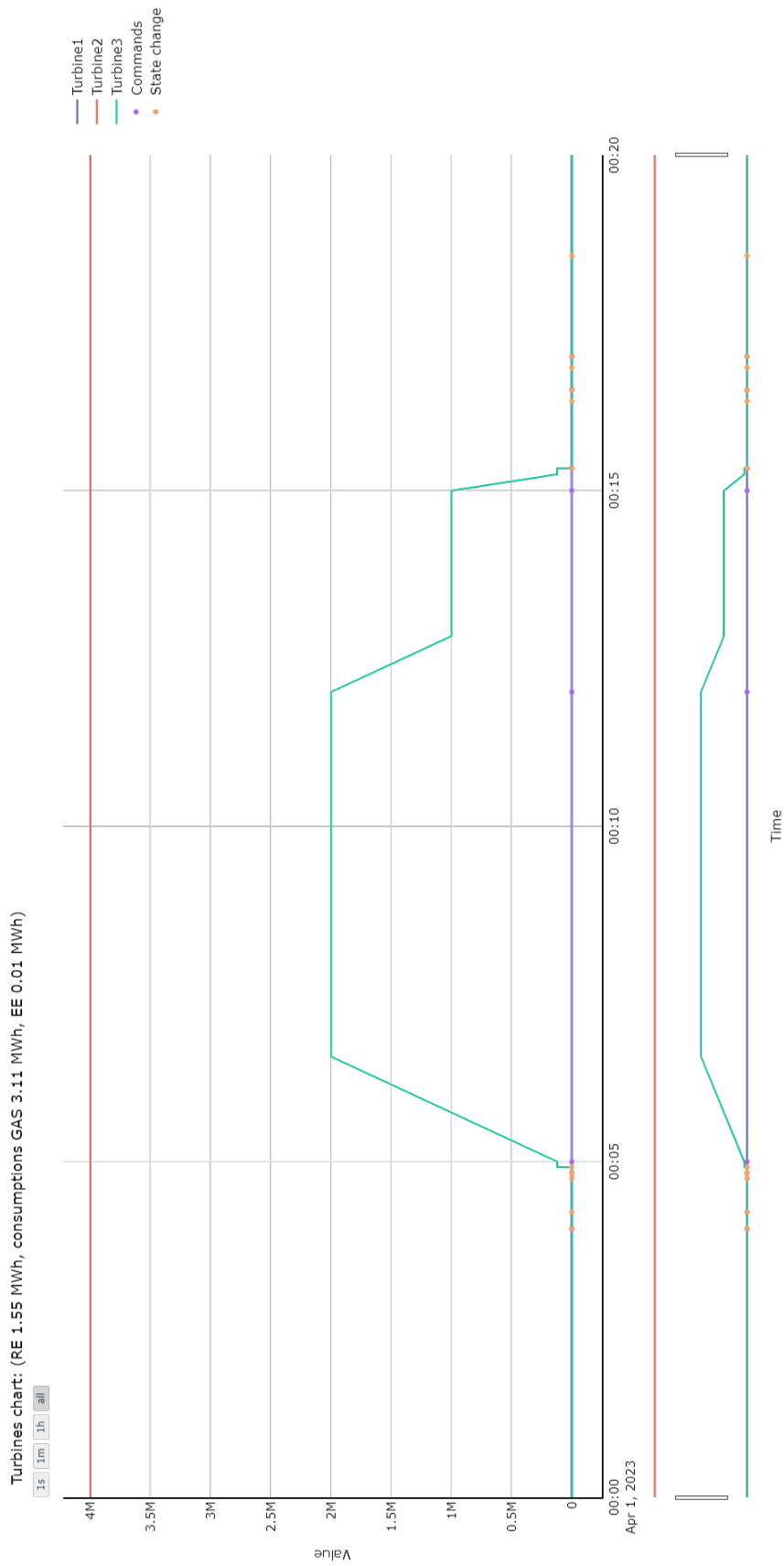
Ústředním grafem je bilance. Tento graf může vidět na obrázku 8.5 a zobrazuje agregace hlavních informací – výkon dodávaný baterkou (modrá křivka), úroveň nabití baterie (červená křivka), žádaný výkon služeb (zelená křivka), součet výstupního výkonu turbín (fialová křivka), výstupní výkon zařízení

(oranžová křivka). Díky cíli poskytnout výkon, který je žádán, zelená křivka pro žádaný výkon služeb není vidět a je zcela překryta oranžovou křivkou výstupního výkonu zařízení. Řízení pro baterii kompenzuje rozdíl mezi žádaným výkonem a výstupním výkonem turbín. Výkyvem mezi 10. a 30. vteřinou způsobeným požadavky na FCR baterie kompenzuje tento chybějící žádaný výkon. Po 5. minutě můžeme vidět, že nám roste výkon z turbín způsobený turbínou 3, přebytky jsou ukládány do baterie.

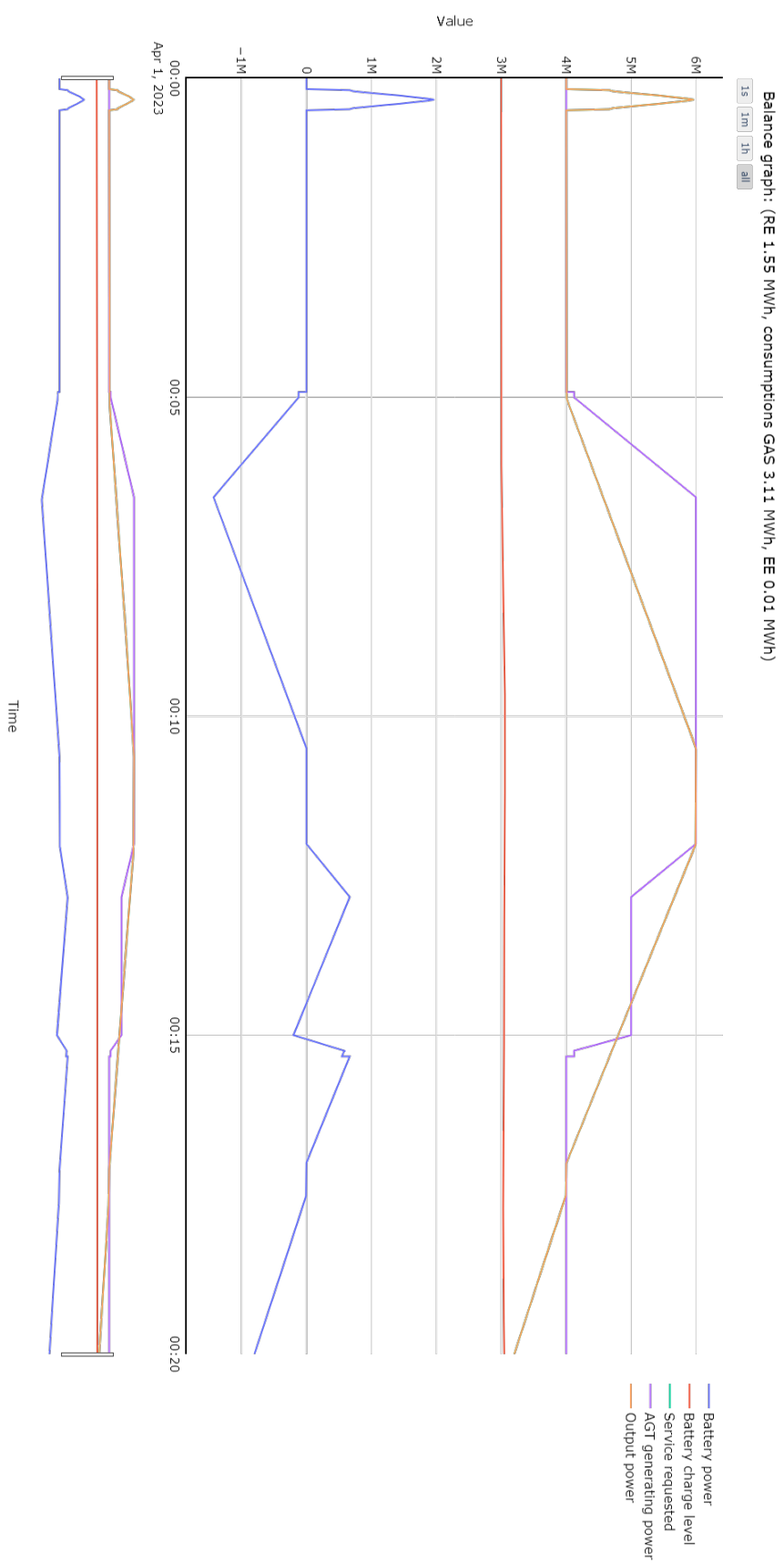
Další dva grafy se zabývají spotřebou komodit. Na obrázku 8.6 vidíme vývoj příkonu elektrické energie za jednotlivé turbíny a celkový součet. Příkon elektrické energie je dán stavem, tyto změny jsou skokové, proto graf má schodovitou charakteristiku. Na obrázku 8.7 vidíme vývoj příkonu plynu rovněž za jednotlivé turbíny a v celkovém součtu. Spotřeba plynu je dána stavem a případně výkonem. Díky tomu můžeme částečně vidět schodovitou charakteristiku, která je ovšem doplněna dynamickým průběhem. U turbíny 3 znázorněné zelenou barvou vidíme prohnutou křivku od 5. minuty. V ten moment jsme započali navyšovat výkon a v každou vteřinu je snímána aktuální hodnota spotřeby. Díky nelineární nákladové charakteristice turbíny, která byla připravena v rámci syntetických dat nominálních spotřeb, je tento projev viditelný. Opět grafy obsahují body změn stavů a vydání povelů. Dodatečné informace nevidíme, můžeme si však všimnout, že výskyt bodů alespoň odpovídá zlomům nebo skokům na křivce. Tyto chybějící informace doplňujeme výpisem logu ve výpisu 7. Zde můžeme vidět informaci o inicializaci všech turbín a baterky. Záznamy o volání a zpracování povelů, zahájení změny a dosažení stavů, změnách výkonů. Upozorníme na čas 16:30, kde se nachází záznam, že došlo k přerušení timeoutu. Turbína se nacházela ve stavu, který měl být opuštěn po uplynutí timeoutu. Stav byl opuštěn dříve pomocí vydaného povelu a plánovaná událost změny stavu vynucené timeoutem byla zrušena. Na konci logu můžeme vidět informace o spotřebách včetně ceny, dle zadaných vstupů a množství dodané regulační energie v rámci služeb.



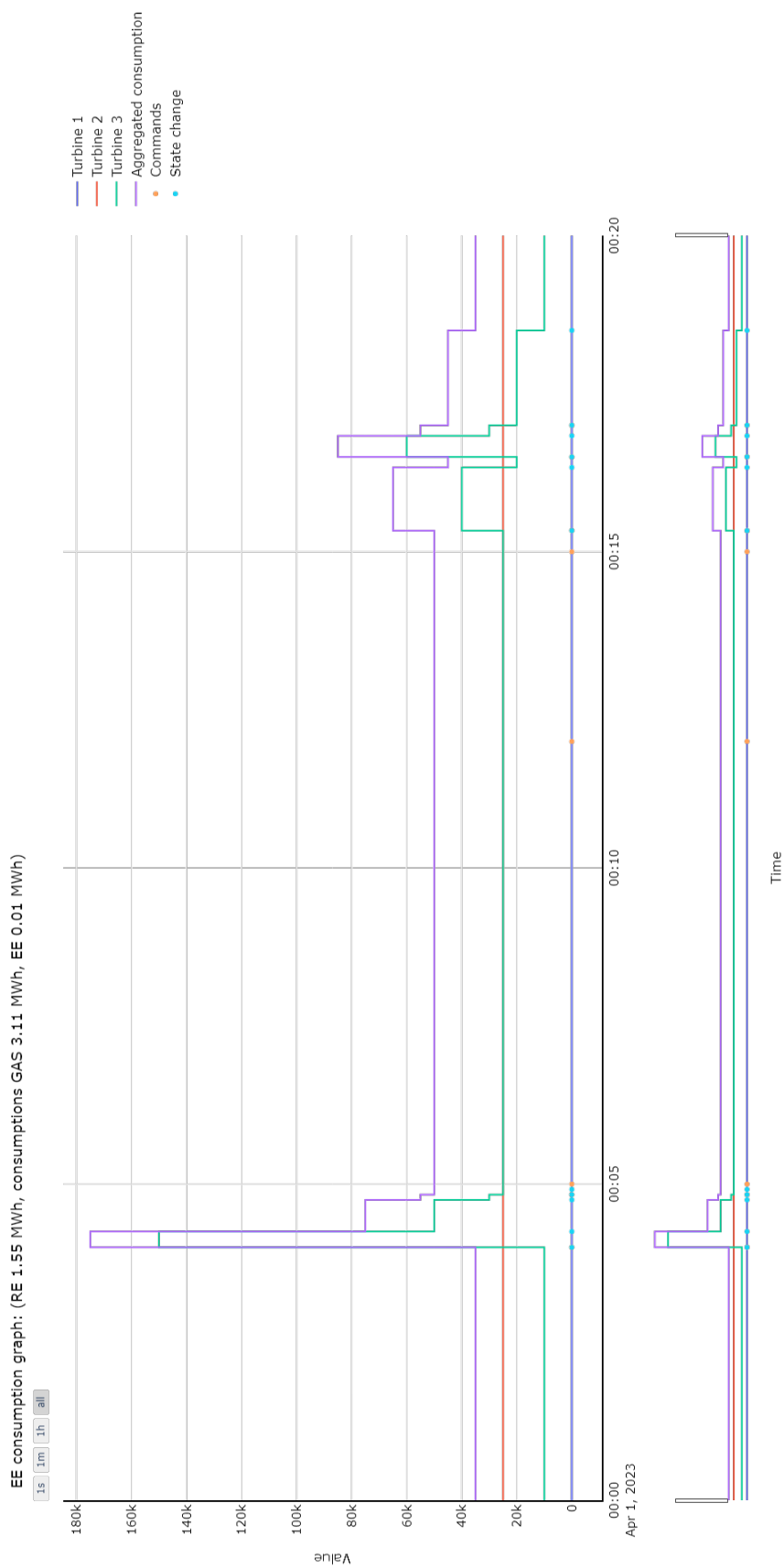
Obrázek 8.3: Ukázka výstupu – graf služeb



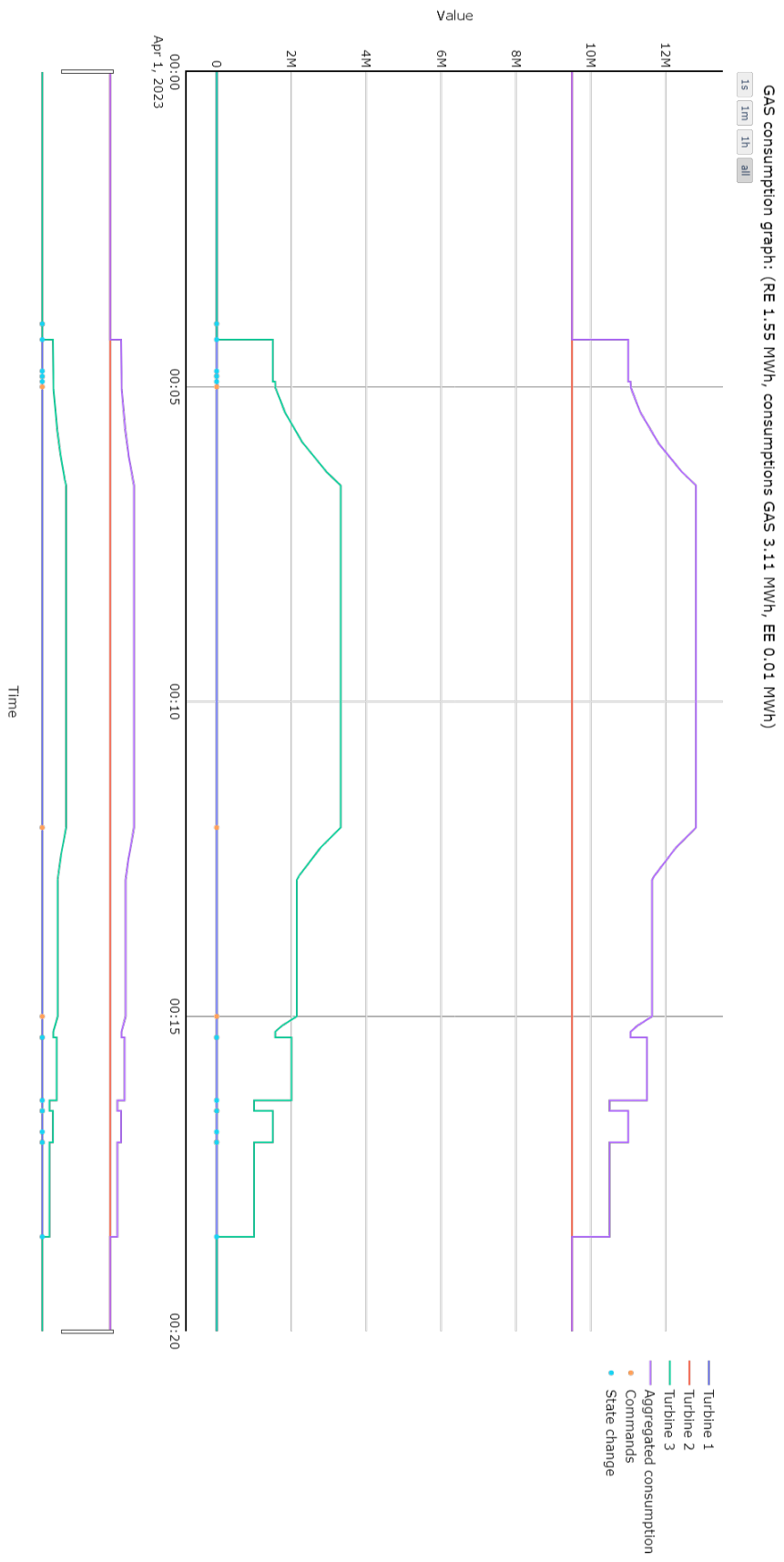
Obrázek 8.4: Ukázka výstupu – graf výkonů turbín



Obrázek 8.5: Ukázka výstupu – graf bilance



Obrázek 8.6: Ukázka výstupu – graf spotřeb elektrické energie



Obrázek 8.7: Ukázka výstupu – graf spotřeb plynu

```

2023-04-01 00:00:00 (0) MessageLogLevel.INFO: Turbine 1 initiated into state S0
↳ with required output power "0.0"
2023-04-01 00:00:00 (0) MessageLogLevel.INFO: Turbine 2 initiated into state S7
↳ with required output power "4000.0"
2023-04-01 00:00:00 (0) MessageLogLevel.INFO: Turbine 3 initiated into state S2
↳ with required output power "0.0"
2023-04-01 00:00:00 (0) MessageLogLevel.INFO: Battery initiated on 3000000 Wh
2023-04-01 00:04:00 (240) MessageLogLevel.INFO: Turbine 3 command "A" called
2023-04-01 00:04:00 (240) MessageLogLevel.INFO: Turbine 3 handling command "A"
2023-04-01 00:04:00 (240) MessageLogLevel.INFO: Turbine 3 changing state S2->S3
2023-04-01 00:04:00 (240) MessageLogLevel.INFO: Turbine 3 reached state S3
2023-04-01 00:04:15 (255) MessageLogLevel.INFO: Turbine 3 changing state S3->S4
2023-04-01 00:04:15 (255) MessageLogLevel.INFO: Turbine 3 reached state S4
2023-04-01 00:04:45 (285) MessageLogLevel.INFO: Turbine 3 changing state S4->S5
2023-04-01 00:04:45 (285) MessageLogLevel.INFO: Turbine 3 reached state S5
2023-04-01 00:04:50 (290) MessageLogLevel.INFO: Turbine 3 command "B" called
2023-04-01 00:04:50 (290) MessageLogLevel.INFO: Turbine 3 handling command "B"
2023-04-01 00:04:50 (290) MessageLogLevel.INFO: Turbine 3 changing state S5->S6
2023-04-01 00:04:50 (290) MessageLogLevel.INFO: Turbine 3 reached state S6
2023-04-01 00:04:55 (295) MessageLogLevel.INFO: Turbine 3 changing state S6->S7
2023-04-01 00:04:55 (295) MessageLogLevel.INFO: Turbine 3 reached state S7
2023-04-01 00:05:00 (300) MessageLogLevel.INFO: Turbine 3 regulation command "C"
↳ called
2023-04-01 00:05:00 (300) MessageLogLevel.INFO: Turbine 3 changing power output
↳ from "120.0" to "2000.0" with gradient "20.0"
2023-04-01 00:06:34 (394.0) MessageLogLevel.INFO: Turbine 3 reached requested
↳ power output "2000.0"
2023-04-01 00:12:00 (720) MessageLogLevel.INFO: Turbine 3 regulation command "C"
↳ called
2023-04-01 00:12:00 (720) MessageLogLevel.INFO: Turbine 3 changing power output
↳ from "2000.0" to "1000.0" with gradient "-20.0"
2023-04-01 00:12:50 (770.0) MessageLogLevel.INFO: Turbine 3 reached requested
↳ power output "1000.0"
2023-04-01 00:15:00 (900) MessageLogLevel.INFO: Turbine 3 regulation command "C"
↳ called
2023-04-01 00:15:00 (900) MessageLogLevel.INFO: Turbine 3 changing power output
↳ from "1000.0" to "120.0" with gradient "-60.0"
2023-04-01 00:15:14.666667 (914.6666666666666) MessageLogLevel.INFO: Turbine 3
↳ reached requested power output "120.0"
2023-04-01 00:15:20 (920) MessageLogLevel.INFO: Turbine 3 command "D" called
2023-04-01 00:15:20 (920) MessageLogLevel.INFO: Turbine 3 handling command "D"
2023-04-01 00:15:20 (920) MessageLogLevel.INFO: Turbine 3 changing state S7->S8
2023-04-01 00:15:20 (920) MessageLogLevel.INFO: Turbine 3 reached state S8
2023-04-01 00:16:20 (980) MessageLogLevel.INFO: Turbine 3 changing state S8->S9
2023-04-01 00:16:20 (980) MessageLogLevel.INFO: Turbine 3 reached state S9
2023-04-01 00:16:30 (990) MessageLogLevel.INFO: Turbine 3 command "F" called
2023-04-01 00:16:30 (990) MessageLogLevel.INFO: Turbine 3 handling command "F"
2023-04-01 00:16:30 (990) MessageLogLevel.INFO: Turbine 3 changing state S9->S10
2023-04-01 00:16:30 (990) MessageLogLevel.INFO: Turbine 3 reached state S10
2023-04-01 00:16:30 (990) MessageLogLevel.INFO: Turbine 3 timeout has been
↳ interrupted in state change
2023-04-01 00:16:50 (1010) MessageLogLevel.INFO: Turbine 3 changing state
↳ S10->S5
2023-04-01 00:16:50 (1010) MessageLogLevel.INFO: Turbine 3 reached state S5
2023-04-01 00:17:00 (1020) MessageLogLevel.INFO: Turbine 3 command "E" called
2023-04-01 00:17:00 (1020) MessageLogLevel.INFO: Turbine 3 handling command "E"
2023-04-01 00:17:00 (1020) MessageLogLevel.INFO: Turbine 3 changing state S5->S9
2023-04-01 00:17:00 (1020) MessageLogLevel.INFO: Turbine 3 reached state S9
2023-04-01 00:18:30 (1110) MessageLogLevel.INFO: Turbine 3 changing state S9->S2
2023-04-01 00:18:30 (1110) MessageLogLevel.INFO: Turbine 3 reached state S2
Electricity consumption 0.01 MWh, total price 92.49
Gas consumption 3.11 MWh, total price 7773.18
Regulation energy produced 1.55 MWh

```

Výpis 7: Ukázka výstupu logu z běhu digitálního dvojče.

Veškeré vstupy a výstupy této ukázky jsou součástí přílohy. Vidíme, že na malou ukázkou prezentace vstupů a výstupů je množství informací celkem obsáhlé.

8.2 Ilustrace rozsáhlejší simulace

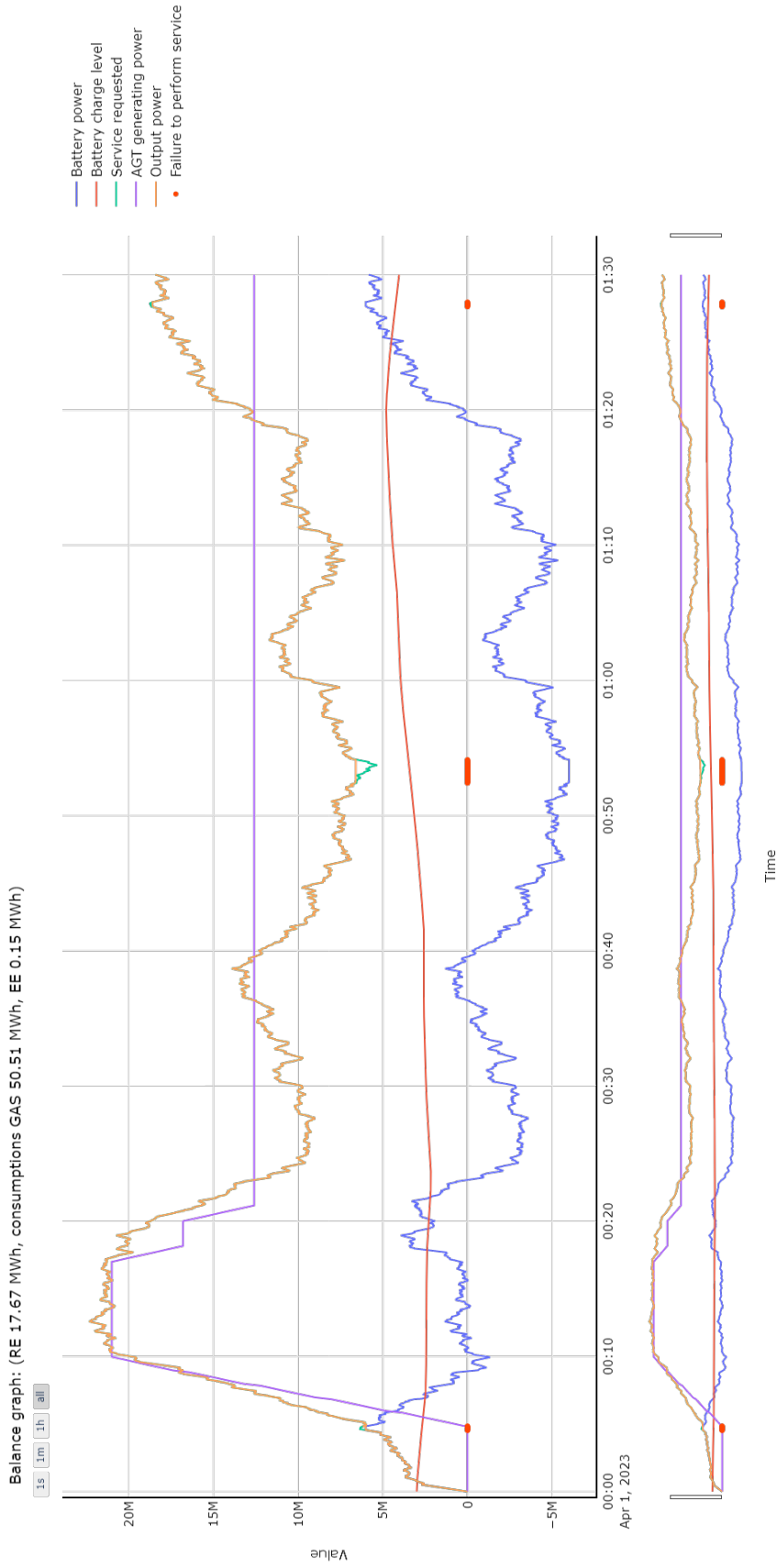
V předchozí části jsme si ukázali fungování digitálního dvojče na malém příkladu. Nyní budeme ilustrovat chování na delším časovém horizontu. Mnoho zachováme ve stejné podobě jako předchozí ukázka – stavový diagram turbín, nominální výkony a spotřeby. Použijeme průběh inspirovanými reálnými daty. Konkrétní průběh služeb, ale neuvedeme. Ukážeme pouze průběh součtu žádaného výkonu všech služeb dohromady. Povelů pro řízení byly nastaveny od ruky, tak aby většina požadavků byla plněna. Zanechali jsme některé části neplnění, aby byl vidět i tento případ. Konfigurace je shodná s předchozím příkladem. Odlišně byla nastavena délka simulace na 90 minut, počet turbín na 5 a všechny začínající ve stavu *READY*. Z celé ukázky ukážeme to nejzajímavější a tím jsou vstupní povelů pro řízení, graf bilance a výkonů turbín. Povelů pro řízení nalezneme ve výpisu 8. Komentář již neposkytneme, čtenář by měl být schopen jejich významu již porozumět. Na obrázku 8.8

```

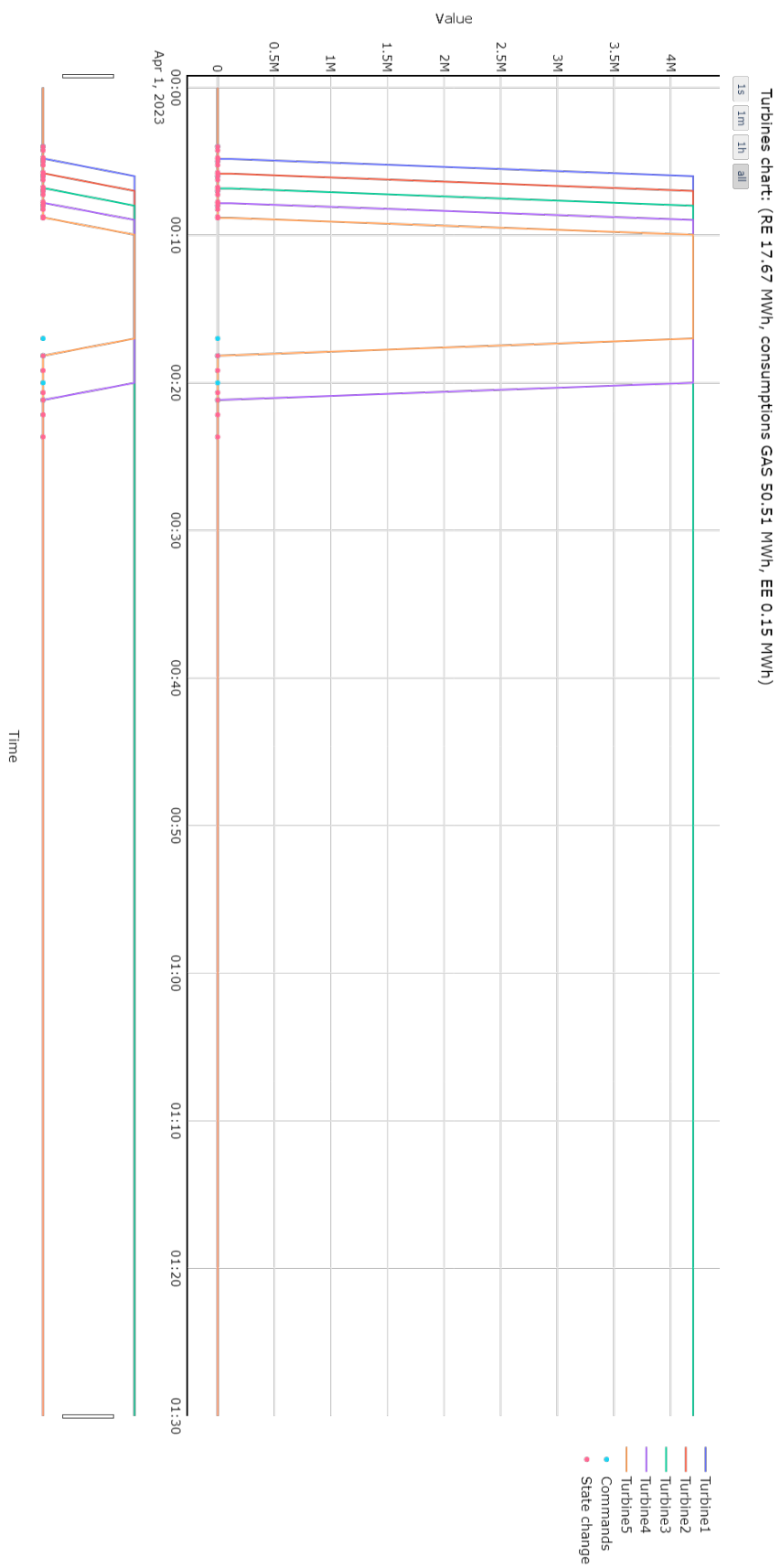
1  {
2  "commands": [
3    ["2023-04-01 00:04:00", 1, "ON", 4200, 60],
4    ["2023-04-01 00:05:00", 2, "ON", 4200, 60],
5    ["2023-04-01 00:06:00", 3, "ON", 4200, 60],
6    ["2023-04-01 00:07:00", 4, "ON", 4200, 60],
7    ["2023-04-01 00:08:00", 5, "ON", 4200, 60],
8    ["2023-04-01 00:17:00", 5, "C", 120, -60],
9    ["2023-04-01 00:18:10", 5, "OFF"],
10   ["2023-04-01 00:20:00", 4, "C", 120, -60],
11   ["2023-04-01 00:21:10", 4, "OFF"]
12 ]
13 }
```

Výpis 8: Ukázka vstupu povelů turbínám pro manuální řízení, rozsáhlejší scénář

můžeme vidět průběh jednotlivých charakteristik. Na obrázku upozorníme, že nově se zde vyskytují červené body na horizontální ose, které symbolizují nesoulad výstupního výkonu zařízení a žádaného výkonu v rámci služeb. V daných místech je vidět zelená křivka požadavku na služby mimo průběh výstupního výkonu zařízení zobrazeného oranžově. V tomto případě jsme řídicí povelů záměrně zadali nedokonale, aby tato vlastnost vyhodnocení výsledku byla vidět. Důvodem proč k tomu dochází je ten, že výrobní výkon turbín se nedostatečně blíží k žádanému výkonu a baterie díky svým limitům výkonu tento rozdíl není schopna kompenzovat. Navíc ještě uvádíme graf výkonů turbín na obrázku 8.9.



Obrázek 8.8: Ukázka výstupu – graf bilance, rozsáhlejší scénář



Obrázek 8.9: Ukázka výstupu – graf výkonů turbín, rozsáhlejší scénář

8.3 Výkonnost digitálního dvojče

Ukázali jsme si na dvou příkladech, jak vypadá chování digitálního dvojče. Nyní nás bude zajímat, jakou škálovatelnost vytvořený systém má. S jakou rychlostí dokáže různé časové úseky simulovat a jestli se mění výkonost s počtem uvažovaných turbín. Experiment bude prováděn na datech služeb maximální délky jednoho týdne, které není možné zveřejnit, ale která vychází z reálnějších experimentů. Znalost přesného průběhu pro experiment není podstatný. Poznamenejme jen, že maximální možný požadavek na FCR je ± 10 MW, pro aFRR a mFRR je shodně možný požadavek do 10MW. Charakteristika turbín a baterie bude shodná s již prezentovanými vstupy. Budeme pouze měnit délku simulace a počet turbín. Výchozí stav při inicializaci pro všechny turbíny bude stav *READY*. Aby vytváření řídicích povelů nebylo složité, využijeme automatizovaného řídicího algoritmu, který není předmětem této práce. To nám dodá seznam povelů, které můžeme načíst a provést pomocí manuálního řízení. Nasimulujeme tím případ reálnější použití, avšak vynecháme vlivu složitější implementace automatizovaného řízení na rychlost výpočtu simulace. Měříme čas od spuštění simulace včetně inicializace komponent do jejího konce. Součástí není čtení dat ze souborů a ukládání výsledků do souborů, vytváření výstupních grafů.

Konkrétní měření jsme provedli pro případ 5 a 10 turbín pro časové horizonty 1, 2, 4, 8, 16 hodin a 1 – 7 dní. Výsledky si můžeme prohlédnout v tabulce 8.1. Naměřené hodnoty jsou uvedeny ve vteřinách. Při porovnání časů u všech

x	1h	2h	4h	8h	16h	1d	2d	3d	4d	5d	6d	7d
5t	1,43	2,72	5,24	10,78	21,93	32,01	64,25	96,36	132,18	161,25	200,49	229,72
10t	1,73	3,37	7,03	12,92	26,31	38,81	77,95	117,83	157,30	197,50	239,03	277,05

Tabulka 8.1: Tabulka rychlostí výpočtu simulace dle délky simulovaného období a počtu turbín v simulaci

délek simulací pro 5 a 10 turbín je vidět, že nárůst je něco okolo 20% procent. Při srovnání časů vzhledem k měnící se délce simulace můžeme pozorovat lineární růst s koeficientem $1 - 1,05$ pro delší časová období. Pro kratší časové úseky koeficient vychází vyšší, avšak časy absolutně jsou krátké a projeví se znatelně společný overhead pro inicializaci vůči celkovému běhu. Pro určení poměru mezi simulovaným časem a dobou běhu, pro 5 turbín při nejdelším 7 denním scénáři vychází délka výpočtu 0,37 ms na simulaci 1 s reálného světa. Na závěr lze zhodnotit, že se podařilo vytvořit simulátor hybridního zdroje, který je dobře škálovatelný. Délka běhu úměrně roste velikosti vstupu (délce simulovaného období). Výsledky z porovnání 5 a 10 turbín naznačují, že nárůst není příliš znatelný. Z pohledu počtu turbín, studium více jak deseti pro nás nemá význam, uvažujeme simulace zdrojů nejvýše do tohoto počtu.

8.4 Zhodnocení výsledku

Na závěr této kapitoly provedeme zhodnocení představených výsledků a doplníme poznámku k testování. Dle formulovaných požadavků se podařilo celé zadání splnit. Podařilo se namodelovat funkci turbín, které mění svůj stav skrze stavový diagram, dokáží ve výkonovém stavu regulovat dodávaný výkon. Baterie rovněž plní svou funkci, umí ukládat i poskytovat energii v rámci své kapacity. Digitální dvojče umí zpracovávat požadavky na služby dle vstupních dat a předávat řízení. Veškerá data jsou v průběhu sbírána a úspěšně vizualizovaná do grafů. Prezentované řídicí rozhraní bylo využito pro potřeby manuálního řízení, ale rovněž vzniklé digitální dvojče již umožnilo vznik dalších řídicích algoritmů mimo tuto práci. Například řízení na bázi MILP se známým horizontem žádaného výkonu. Toto řízení díky vytvořenému modelu spočítá optimální strategii pro řízení, ze které jsou vytvořeny řídicí povely, které je možné provést pomocí digitálního dvojče. Tento algoritmus může posloužit jako benchmark pro optimální cenu řízení na krátkých horizontech. Vývoj tohoto algoritmu byl předmětem jiné diplomové práce, která bude k nalezení na DSpace Českého vysokého učení technického[34]. Dále v rámci digitálního dvojče byl vyvíjen heuristický řídicí algoritmus, který je prototypem pro budoucí implementaci reálného řídicího algoritmu.

V rámci práce jsme se prozatím nezabývali otázkou testování. Jelikož vzniklé dílo není malého rozsahu a prioritou byl vývoj funkcionalit a ladění propojení s dalšími řídicími algoritmy, testování probíhalo pouze manuálně během vývoje. Nebylo možné vyčlenit dostatek času na samostatný blok testování například pomocí nějakých automatizovaných testů. Především se prováděla kontrola chování skrze produkované výstupy při vývoji nových funkcionalit. Rovněž v rámci neustálého užívání digitálního dvojče pro další výzkum byla příležitost zachytit a opravit uniklé chyby.

Kapitola 9

Závěr

Tato práce se zabývala návrhem digitálního dvojče hybridního zdroje elektrické energie pro poskytování podpurných služeb (dodávky regulační energie) zabezpečující stabilitu elektrické sítě. Celá práce započala studiem problematiky, co to jsou podpurné služby, jaký mají dopad na zadání práce – vytvoření schopnosti porozumění doméně. V další etapě probíhalo seznámení se s technologiemi, které bude digitální dvojče napodobovat – turbíny, bateriový systém a bylo potřeba vyřešit, jak tyto komponenty modelovat, aby bylo možné postihnout jejich reálné chování. Vytvořený model sice některé věci zanedbává, nicméně podařilo se vytvořit dostatečnou reprezentaci pro ilustraci chování technologií i celého zařízení. Vytvoření digitálního dvojče bylo dílčí úlohou jiného náročnějšího úkolu, který není předmětem této práce a to je vývoj robustního řízení prezentovaného hybridního zdroje, kde řízení se musí vypořádat s otázkou neurčitosti. Studium problému výrazně usnadnila existence právě vytvořeného digitálního dvojče díky možnosti testovat chování různých návrhů řídicích algoritmů a vizualizovat dění hybridního zdroje.

Kromě faktu, že se podařilo splnit zadání popsané pomocí požadavků v rámci jednání se zadavatelem a s výsledkem díla byl spokojen, digitální dvojče splnilo svůj účel v rámci svých plynoucích přínosů. Jsme schopni ilustrovat fungování zdroje na základě řídicího algoritmů nasazeného na zařízení, které zatím neexistuje. Jsme schopni ověřit chování na testech, které by v reálném provozu stáli velké množství peněz vzhledem k velkým spotřebám plynu a jeho ceně. Rovněž testy typu, které bychom potřebovali vyzkoušet není možné provádět, nemůžeme si dovolit jen tak do sítě uvolnit 10 MW výkonu, protože počínání tohoto typu by samo o sobě vytvářelo nerovnováhu v síti. V budoucnu, pokud takový zdroj vznikne je cílem rozšíření digitálního dvojče o schopnost napojení na reálná data z provozu a jejich možnou diagnostiku. To přispěje v rámci zpětné analýzy chování z provozu, například pokud došlo k chybě, k porozumění z čeho vznikla a možnosti otestování opravy řízení, které může být nasazeno zpět do reálného zařízení.

Na výsledném produktu digitálního dvojče je prováděn další výzkum řízení. Již se podařilo vytvořit koncept řízení založeném na výpočtu nad modelem MILP, které poskytne optimální řešení z finančního pohledu při předem známém vývoji žádaného výkonu[34]. Toto řízení slouží pro srovnání jako optimální nákladové řešení při vývoji dalších algoritmů, které již budoucí vývoj

požadavků služeb nemají k dispozici. Jedná se zejména o řešení založená na heuristickém a podmínkovém přístupu. Digitální dvojče posloužilo a poslouží pro další výzkum a napomáhá k vývoji nového hybridního zdroje a jeho případné správě.

Příloha A

Literatura

- [1] Ondřej Mamula and David Hrycej. Replacement of ancillary service resources lost during decarbonization. In *2022 14th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pages 145–152. IEEE, 2022.
- [2] Michael Grieves. Digital twin: manufacturing excellence through virtual factory replication. *White paper*, 1(2014):1–7, 2014.
- [3] Eric VanDerHorn and Sankaran Mahadevan. Digital twin: Generalization, characterization and implementation. *Decision Support Systems*, 145:113524, 2021.
- [4] Ludvig Wingren and Jonas Johnsson. Battery energy storage systems as an alternative to gas turbines for the fast active disturbance reserve, 2018.
- [5] Jahedul Islam Chowdhury, Nazmiye Balta-Ozkan, Pietro Goglio, Yukun Hu, Liz Varga, and Leah McCabe. Techno-environmental analysis of battery storage for grid level energy services. *Renewable and Sustainable Energy Reviews*, 131:110018, 2020.
- [6] Joohyun Cho and Andrew N. Kleit. Energy storage systems in energy and ancillary markets: A backwards induction approach. *Applied Energy*, 147:176–183, 2015.
- [7] Lluc Canals Casals and Beatriz Amante García. Second-life batteries on a gas turbine power plant to provide area regulation services. *Batteries*, 3(1):10, 2017.
- [8] F.A.V. Biggins, S. Homan, J.O. Ejeh, and S. Brown. To trade or not to trade: Simultaneously optimising battery storage for arbitrage and ancillary services. *Journal of Energy Storage*, 50:104234, 2022.
- [9] Mostafa Naemi, Dominic Davis, and Michael J. Brear. Optimisation and analysis of battery storage integrated into a wind power plant participating in a wholesale electricity market with energy and ancillary services. *Journal of Cleaner Production*, 373:133909, 2022.

- [10] Jin Tan and Yingchen Zhang. Coordinated control strategy of a battery energy storage system to support a wind power plant providing multi-timescale frequency ancillary services. *IEEE Transactions on Sustainable Energy*, 8(3):1140–1153, 2017.
- [11] Mostafa Naemi, Dominic Davis, and Michael J. Brear. Optimisation and analysis of battery storage integrated into a wind power plant participating in a wholesale electricity market with energy and ancillary services. *Journal of Cleaner Production*, 373:133909, 2022.
- [12] Hristiyan Kanchev, Bruno Francois, and V Lazarov. Unit commitment by dynamic programming for microgrid operational planning optimization and emission reduction. In *International Aegean Conference on Electrical Machines and Power Electronics and Electromotion, Joint Conference*, pages 502–507. IEEE, 2011.
- [13] Ushnik Mukherjee, Sean Walker, Azadeh Maroufmashat, Michael Fowler, and Ali Elkamel. Power-to-gas to meet transportation demand while providing ancillary services to the electrical grid. In *2016 IEEE smart energy grid engineering (SEGE)*, pages 221–225. IEEE, 2016.
- [14] Lingxi Zhang, Stephen Clegg, and Pierluigi Mancarella. Modeling of electrolyzers in hydrogen vehicle refueling stations for provision of ancillary services. In *IREP2017 Symposium*, 2017.
- [15] Mareike Jentsch, Tobias Trost, and Michael Sterner. Optimal use of power-to-gas energy storage systems in an 85% renewable energy scenario. *Energy Procedia*, 46:254–261, 2014. 8th International Renewable Energy Storage Conference and Exhibition (IRES 2013).
- [16] Thumeera R. Wanasinghe, Leah Wroblewski, Bui K. Petersen, Raymond G. Gosine, Lesley Anne James, Oscar De Silva, George K. I. Mann, and Peter J. Warrian. Digital twin for the oil and gas industry: Overview, research trends, opportunities, and challenges. *IEEE Access*, 8:104175–104197, 2020.
- [17] Siavash H. Khajavi, Naser Hossein Motlagh, Alireza Jaribion, Liss C. Werner, and Jan Holmström. Digital twin: Vision, benefits, boundaries, and creation for buildings. *IEEE Access*, 7:147406–147419, 2019.
- [18] Bin Xu, June Wang, Xinping Wang, Zhihong Liang, Liming Cui, Xiao Liu, and Anthony Y Ku. A case study of digital-twin-modelling analysis on power-plant-performance optimizations. *Clean Energy*, 3(3):227–234, 11 2019.
- [19] Michel R. Miyazaki, Asgeir J. Sørensen, Nicolas Lefebvre, Kevin K. Yum, and Eilif Pedersen. Hybrid modeling of strategic loading of a marine hybrid power plant with experimental validation. *IEEE Access*, 4:8793–8804, 2016.

- [20] Jianxi Yu, Nils Petersen, Pei Liu, Zheng Li, and Manfred Wirsum. Hybrid modelling and simulation of thermal systems of in-service power plants for digital twin development. *Energy*, 260:125088, 2022.
- [21] Emma Söderäng, Saana Hautala, Maciej Mikulski, Xiaoguo Storm, and Seppo Niemi. Development of a digital twin for real-time simulation of a combustion engine-based power plant with battery storage and grid coupling. *Energy Conversion and Management*, 266:115793, 2022.
- [22] Thien-An Nguyen-Huu, Trung Thai Tran, Minh-Quan Tran, Phuong H. Nguyen, and Jg Sloopweg. Operation orchestration of local energy communities through digital twin: A review on suitable modeling and simulation approaches. In *2022 IEEE 7th International Energy Conference (ENERGYCON)*, pages 1–6, 2022.
- [23] Jiaxuan Han, Qiteng Hong, Mazheruddin H. Syed, Md Asif Uddin Khan, Guangya Yang, Graeme Burt, and Campbell Booth. Cloud-edge hosted digital twins for coordinated control of distributed energy resources. *IEEE Transactions on Cloud Computing*, pages 1–15, 2022.
- [24] a.s. ČEPS. <https://www.ceps.cz/cs/pro-verejnost>.
- [25] a.s. ČEPS. Kodex přenosové soustavy – část ii., podpůrné služby (pps). https://www.ceps.cz/download-data/?format=pagefile&path=10972/modules/files/88367_kodex-ps-cast-ii-od-12-9-22.pdf.
- [26] Martin Henych. Návrh řízení zdrojů elektrické energie zohledňující nová pravidla na trhu s energiemi. <https://dspace.cvut.cz/handle/10467/100970>.
- [27] Json standard. <https://www.json.org/json-en.html>.
- [28] Simpy – knihovna pro simulace v jazyce python. <https://simpy.readthedocs.io/en/latest/>.
- [29] Html standard. <https://html.spec.whatwg.org/>.
- [30] Python – oficiální stránka programovacího jazyka. <https://www.python.org/>.
- [31] Numba knihovna. <https://numba.pydata.org/>.
- [32] Plotly – knihovna pro tvorbu grafů v jazyce python. <https://plotly.com/python/>.
- [33] Openpyxl – knihovna pro soubory formátu xlsx. <https://openpyxl.readthedocs.io/en/stable/>.
- [34] Bc. Jitka Stoklasová. Algoritmus pro řízení hybridních zdrojů elektrické energie s neurčitým požadavkem. Bude k nalezení na <https://dspace.cvut.cz/>, 2023.



Příloha B

Seznam použitých zkratk

DT - digital twin - digitální dvojče

SVR - služba výkonové rovnováhy

RE - regulační energie

FCR - zálohy pro automatickou regulaci frekvence

aFRR - zálohy pro regulaci výkonové rovnováhy s automatickou regulací

mFRR - zálohy pro regulaci výkonové rovnováhy s manuální aktivací

FIFO - First In First Out - první vznikl, první zpracován

MILP - Mixed Integer Linear Programming - celočíselné lineární programování

Příloha C

Elektronické přílohy

Součástí práce jsou uvedené přílohy v následujícím seznamu rozdělené do složek.

Složka **basic** – jednoduchý příklad

tur_datasheet.xlsx – vstupní soubor obsahující nominální hodnoty výkonů a spotřeb plynu turbíny

tur_state_machine.json – vstupní soubor popisující stavový diagram turbíny

sim_config.json – vstupní soubor konfigurace simulace

commands.json – vstupní soubor obsahující seznam povelů turbínám pro manuální řízení

01_services_fig.html – výstupní soubor, graf služeb

02_turbines_fig.html – výstupní soubor, graf výkonů turbín

03_balance_fig.html – výstupní soubor, graf bilance

04_ee_fig.html – výstupní soubor, graf příkonů elektrické energie turbín

05_gas_fig.html – výstupní soubor, graf příkonů plynu turbín

log.txt – výstupní soubor, log informací z digitálního dvojče

Složka **complex** – rozsáhlejší příklad simulace

tur_datasheet.xlsx – vstupní soubor obsahující nominální hodnoty výkonů a spotřeb plynu turbíny

tur_state_machine.json – vstupní soubor popisující stavový diagram turbíny

sim_config.json – vstupní soubor konfigurace simulace

commands.json – vstupní soubor obsahující seznam povelů turbínám pro manuální řízení

02_turbines_fig.html – výstupní soubor, graf výkonů turbín

03_balance_fig.html – výstupní soubor, graf bilance

04_ee_fig.html – výstupní soubor, graf příkonů elektrické energie turbín

05_gas_fig.html – výstupní soubor, graf příkonů plynu turbín

log.txt – výstupní soubor, log informací z digitálního dvojče