

Diplomová práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra radioelektroniky

Návrh řízení zdrojů elektrické energie zohledňující nová pravidla na trhu s energiemi

Bc. Martin Henych

Školitel: Doc. Ing. Přemysl Šůcha, Ph.D.

Školitel–specialista: Prof. Ing. Pavel Sovka, Cs.C.

Květen 2022

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Henych** Jméno: **Martin** Osobní číslo: **465388**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávací katedra/ústav: **Katedra radioelektroniky**
Studijní program: **Otevřené elektronické systémy**
Studijní obor: **Integrované elektronické systémy**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Návrh řízení zdrojů elektrické energie zohledňující nová pravidla na trhu s energiemi

Název diplomové práce anglicky:

Scheduling of Energy Resources Considering New Regulations of the Electricity Market

Pokyny pro vypracování:

Rozvrhování zdrojů energie je téma intenzivně studované jak v odborné literatuře, tak v průmyslové praxi. Motivací pro tuto práci jsou aktuální změny pravidel na trhu s energiemi. Cílem je popsat nová pravidla pro poskytování základních služeb v dodávání elektrické energie do sítě a navrhnout řízení vybraných zdrojů energie pro jejich poskytování. Zadání se skládá z těchto úkolů:

- 1) Seznamte se s pravidly pro poskytování elektrické energie na trhu a popište jednotlivé služby.
- 2) Proveďte rešerši literatury týkající se rozvrhování zdrojů elektrické energie.
- 3) Navrhněte algoritmus řízení vybraných zdrojů pro poskytování základních služeb. Použijte metodu celočíselného lineárního programování přičemž se zaměřte na efektivitu modelu.
- 4) Algoritmus otestujte na reálných datech a vyhodnoťte dosažené výsledky. Dále vyhodnoťte výpočetní náročnost algoritmu.

Seznam doporučené literatury:

- [1] Juan I. Pérez-Díaz et al., 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 et al., Optimized Operation of Local Energy Community Providing Frequency Restoration Reserve, IEEE Access, Volume 8, page 180558-180575, 2020.

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: **01.02.2022**

Termín odevzdání diplomové práce: **20.05.2022**

Platnost zadání diplomové práce: **30.09.2023**

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

doc. Ing. Stanislav Vitek, Ph.D.
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 odborné vedení mé diplomové práce a konzultace problematiky optimalizace. Dále můj velký dík patří prof. Ing. Pavlu Sovkovi Cs.C. za konzultace v oblasti predikce časových řad a inspiraci pro návrh prediktoru. Děkuji Ing. Miroslavovi Konečnému a Ing. Robertu Pecovi z OSC a.s. za zadání tohoto náročného úkolu a podporu během jeho řešení. Děkuji všem svým kolegům z OSC a.s. se kterými jsem konzultoval veškeré technické detaily, týkající se problematiky přenosové soustavy. Rád bych poděkoval i svým přátelům Bc. Petru Ungarovi a Ing. Martinu Hofmanovi za podporu v začátcích studia informatiky a energetiky. Děkuji svým rodičům za psychickou a materiální podporu během studia. A nakonec můj hlavní a největší dík patří prodejci lyží a příležitostnému kuchaři Jiřímu Babicovi, že je mi tak velkou životní inspirací.

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, 18. května 2022

Bc. Martin Henych

Abstrakt

Práce se zabývá řízením a optimalizací distribuovaných energetických systémů. Cílem práce je návrh algoritmu, pro optimální řízení distribuovaného systému v reálném čase. První část práce se zabývá obecným popisem trhu s energiemi, zavedením systémového popisu jednotek, charakterizováním řídicích signálů a formulací optimalizačního problému. V druhé části je na základě této studie sestaven MILP model, řešící optimalizační zpětně při znalosti všech požadavků na systém. Při řízení v reálném čase, ale nemáme k dispozici budoucí požadavky a úloha tak obsahuje neurčitosti. Neurčitosti byly vyřešeny heuristikami upravující MILP model pro provoz v reálném čase a odhadováním budoucích požadavků. K odhadování požadavků byly zkonstruovány dva prediktory. První prediktor využívá statistického ARIMA modelu a druhý prediktor byl vytvořen jako LSTM neuronová síť. Následné experimenty testují na reálných datech online algoritmus s ARIMA prediktorem a LSTM prediktorem a porovnávají jeho efektivitu vůči optimálnímu řešení nalezené zpětnou optimalizací pomocí offline algoritmu.

Klíčová slova: agregační blok, optimalizace, ILP, ARIMA, LSTM, distribuovaný systém, energetika, aFRR, mFRR, strojové učení

Školitel: Doc. Ing. Přemysl Šůcha,
Ph.D.
Centrum průmyslové informatiky,
Jugoslávských partyzánů,
Praha 6

Abstract

The thesis is focused on the optimization of the Virtual Power Plant (VPP). Its aim is to design an algorithm for optimal control of VPP in real-time. First part describes the energy market, the behaviour of VPP, rules for providing Frequency Restoration Reserve services and formulates our optimization task. The second part of the thesis is focused on the design of the control algorithm. Initially is formulated formal mathematical description of the problem and subsequently provided MILP model. It represents an offline algorithm, that finds an optimal solution on a closed time interval if we know the trajectories of input signals. As demands for the reserve power activations are not known in real-time, it brings additional uncertainties to be solved. To deal with them is used rolling window method with a modified MILP model and with created predictors using ML algorithms and statistic methods. The developed algorithm is then tested with the real data from multiple VPPs and the output is compared with the optimal solution obtained from an offline algorithm.

Keywords: virtual power plant, optimization, ILP, ARIMA, LSTM, distributed system, energetics, aFRR, mFRR, machine learning

Title translation: Scheduling of Energy Resources Considering New Regulations of the Electricity Market

Obsah

1 Úvod	1	9 Predikce pomocí Statistického Modelu	53
2 Pravidla trhu s energiemi	3	9.1 Úvod do ARIMA modelu	53
2.1 Princip trhu	3	9.2 Identifikace modelu	54
2.2 Účastníci trhu s energiemi	4	9.3 Estimace parametrů modelu	55
2.3 Stanovení dodávek plánované a regulační energie	5	9.4 Návrh prediktoru	57
3 Energetické jednotky na straně poskytovatele	7	9.5 Výsledný prediktor	60
3.1 Výrobní jednotka	7	10 Predikce pomocí Neuronové Sítě	63
3.2 Akumulátor	9	10.1 Úvod do neuronových sítí	63
3.3 Charakteristika zdrojů dle typu energie	11	10.1.1 Matematický popis a základní princip	63
3.3.1 Tepelné elektrárny	11	10.1.2 Aktivační funkce	66
3.3.2 Přehradní vodní elektrárny	12	10.1.3 Ztrátová funkce	67
3.3.3 Přecherpávací vodní elektrárna	13	10.1.4 Konvoluční vrstva	68
3.3.4 Baterie	13	10.2 Rekurentní neuronové sítě	69
3.3.5 Obnovitelné zdroje	14	10.2.1 LSTM architektura	70
3.4 Agregáčnı blok	14	10.2.2 „Many to One“ a "Many to Many" architektury	72
4 Podpůrné služby	17	10.3 Návrh prediktoru	73
4.1 FCR	18	10.3.1 „Many to One“	73
4.2 aFRR	19	10.3.2 „Many to many“	75
4.3 mFRR	23	10.4 Výsledný prediktor	75
4.4 RR	26	10.4.1 Porovnání LSTM prediktoru s ARIMA prediktorem	77
4.5 Shrnutí podpůrných služeb	27	11 Testování modelů	81
5 Souhrn vybraných pracı věnující se našemu optimalizačnímu problému	29	11.1 Testování výpočetní náročnosti algoritmu	81
5.1 Ucelený souhrn problematiky	29	11.2 Testování online modelu	84
5.2 Původní optimalizace virtuální elektrárny	30	11.3 Testování dynamických vlastností	89
5.3 Současné koncepty	30	11.4 Testování na reálných datech	94
5.4 Shrnutí	32	11.4.1 Agregáčnı blok E.ON	94
6 Optimalizační problém	33	11.4.2 Agregáčnı blok ČEZ	98
7 MILP model soustavy	37	11.5 Diskuze nad výsledky experimentů	103
7.1 Úvod do ILP	37	12 Závěr	105
7.2 Požadavky na model	38	12.1 Shrnutí	105
7.3 Proměnné a parametry modelu	39	12.2 Náměty na budoucí práci	106
7.4 Omezující podmínky modelu	39	A Literatura	107
7.5 Kriteriałnı funkce modelu	42		
8 Online řešení	43		
8.1 Rolling window	43		
8.2 Úprava MILP modelu	45		
8.2.1 Počáteční podmínky	45		
8.2.2 Limitní požadavky	46		
8.2.3 Úprava kriteriałnı funkce	50		

Obrázky

2.1 Diagramový výkon	6	8.3 Ukázka predikovaného a plně rozšířeného limitního průběhu . . .	50
2.2 Porovnání diagramového výkonu a výkonu odebíraného ze sítě	6	9.1 Požadavky aktivace služby aFRR	57
3.1 Porovnání systémové odezvy systému s omezeným a neomezeným trendem	9	9.2 Původní a přefiltrované požadavky spolu s limitními požadavky na odezvu	58
3.2 Schéma agregačního bloku tvořeného dvěma uhelnými elektrárnami, paroplynovou elektrárnou, přečerpávací elektrárnou třemi vodními elektrárny a bateriovou stanicí	15	9.3 Závislost energie signálu na stupni diference	59
4.1 Ukázka omezení zadávacího pásma vlivem FCR a průběh výkonu na výrobní jednotce	19	9.4 Zobrazení A: diferencovaného signálu B: jeho autokorelační funkce C: jeho parciální-autokorelační funkce	60
4.2 Ukázka požadavku na aFRR, odezva systému a meze pro splnění požadavku	20	9.5 Chyba jednotlivých kroků predikce	61
4.3 Znázornění povelů pro aktivaci a deaktivaci $aFRR_t$ a skutečná hodnota P_t^{DG} a výsledný výkon P_t	21	9.6 Porovnání predikovaného skutečného průběhu aFRR	61
4.4 Znázornění povelů pro aktivaci $aFRR_t$ mezi jednotlivými obchodními intervaly	21	10.1 Schéma neuronu	64
4.5 Reakce na požadavek aFRR z hlediska agregačního bloku tvořeného dvěma výrobními jednotkami	22	10.2 Neuronová síť se 2 skrytými vnitřními vrstvami	65
4.6 Znázornění požadavku na mFRR s odezvou systému a mezemi pro splnění požadavku	23	10.3 Aktivační funkce	67
4.7 Aktivace a deaktivace mFRR	24	10.4 Znázornění chybových funkcí	68
4.8 Reakce na požadavek mFRR z hlediska agregačního bloku tvořeného dvěma výrobními jednotkami	25	10.5 Schéma neuronové sítě s konvolučními a skrytými vrstvami	69
4.9 Průběh aktivace a deaktivace služby RR	26	10.6 Schéma rekurentní neuronové sítě	70
4.10 Poskytování podpůrných služeb v čase spolu s odchylkou frekvence	27	10.7 LSTM buňka	71
7.1 Ukázka ILP problému	38	10.8 „Many to One“ realizace	72
8.1 Zobrazení principu metody posuvného okna	44	10.9 Many to Many realizace	72
8.2 Ukázka predikovaného a limitního průběhu	49	10.10 Závislost chyby predikce na volbě parametrů	74
		10.11 Porovnání predikce pro 4 kroky prediktoru založeném na „Many to One“ principu predikce	74
		10.12 Porovnání přesnosti predikce pro různé architektury „Many to Many“ prediktoru	75
		10.13 Porovnání predikce pro 4 kroky prediktoru založeném na „Many to Many“ principu predikce	76
		10.14 Porovnání přesnosti predikce „Many to One“ a „Many to Many“ architektur	76
		10.15 Porovnání MSE prediktorů ARIMA a LSTM pro jednotlivé kroky predikce	77

10.16 Schéma architektury rekurentní sítě „Many to One“	78	11.13 Rozložení výkonů na akumulátory agregčního bloku při jeho řízení pomocí online algoritmu s ARIMA prediktorem	92
10.17 Schéma architektury rekurentní sítě „Many to Many“	79	11.14 Optimální rozložení výkonů na výrobní jednotky agregčního bloku nalezené pomocí zpětné optimalizace offline algoritmem	92
11.1 Závislost výpočetní náročnosti na $ \Omega^T $ a $ \Omega^{AB} $	83	11.15 Optimální rozložení výkonů na akumulátory agregčního bloku nalezené pomocí zpětné optimalizace offline algoritmem	93
11.2 Průběh požadavků služby $aFRR_t^a$ a $mFRR_t^a$ pro agregční blok ČEZ	85	11.16 Průběh požadavků služby $aFRR_t^a$ a $mFRR_t^a$ pro agregční blok E.ON	95
11.3 Rozložení výkonů pro jednotlivé výrobní jednotky nalezené pomocí online algoritmu s LSTM prediktorem	85	11.17 Rozložení výkonů pro jednotlivé vodní elektrárny a kogenerační jednotky agregčního bloku E.ON řízeného online algoritmem s LSTM prediktorem	95
11.4 Rozložení výkonů pro jednotlivé akumulátory nalezené pomocí online algoritmu s LSTM prediktorem ...	86	11.18 Rozložení výkonů pro jednotlivé vodní elektrárny a kogenerační jednotky agregčního bloku E.ON řízeného online algoritmem s ARIMA prediktorem	96
11.5 Rozložení výkonů pro jednotlivé výrobní jednotky nalezené pomocí online algoritmu s ARIMA prediktorem	86	11.19 Optimální rozložení výkonů pro jednotlivé vodní elektrárny a kogenerační jednotky agregčního bloku E.ON nalezené offline algoritmem	96
11.6 Rozložení výkonů pro jednotlivé akumulátory nalezené pomocí online algoritmu s ARIMA prediktorem .	87	11.20 Průběh požadavků služby $aFRR_t^a$ a $mFRR_t^a$ pro agregční blok ČEZ	99
11.7 Optimální rozložení výkonů pro jednotlivé výrobní jednotky agregčního bloku nalezené pomocí zpětné optimalizace offline algoritmem	87	11.21 Rozložení výkonů pro jednotlivé elektrárny agregčního bloku ČEZ při jeho řízení pomocí online algoritmu s LSTM prediktorem	99
11.8 Optimální rozložení výkonů pro jednotlivé akumulátory agregčního bloku nalezené pomocí zpětné optimalizace offline algoritmem ...	88	11.22 Rozložení výkonů pro jednotlivé přečerpávací vodní elektrárny a baterie agregčního bloku ČEZ s LSTM prediktorem	100
11.9 Průběh požadavků služby $aFRR_t^a$ a $mFRR_t^a$	90	11.23 Rozložení výkonů pro jednotlivé elektrárny agregčního bloku ČEZ při jeho řízení pomocí online algoritmu s ARIMA prediktorem	100
11.10 Rozložení výkonů na výrobní jednotky agregčního bloku při jeho řízení pomocí online algoritmu s LSTM prediktorem	90		
11.11 Rozložení výkonů na akumulátory agregčního bloku při jeho řízení pomocí online algoritmu s LSTM prediktorem	91		
11.12 Rozložení výkonů na výrobní jednotky agregčního bloku při jeho řízení pomocí online algoritmu s ARIMA prediktorem	91		

11.24 Rozložení výkonů pro jednotlivé přečerpávací vodní elektrárny a baterie agregčního bloku ČEZ s ARIMA prediktorem	101
11.25 Optimální rozložení výkonů pro jednotlivé elektrárny agregčního bloku ČEZ nalezené pomocí zpětné optimalizace offline algoritmem ..	101
11.26 Optimální rozložení výkonů pro jednotlivé přečerpávací vodní elektrárny a baterie agregčního bloku ČEZ nalezené pomocí zpětné optimalizace offline algoritmem ..	102

Tabulky

4.1 Služby výkonové rovnováhy	18
7.1 Proměnné MILP modelu	39
7.2 Parametry MILP modelu	40
8.1 Přidané parametry do modelu ..	45
8.2 Seznam proměnných v případě plné aktivace kladné regulační energie ..	46
8.3 Seznam proměnných v případě plné aktivace záporné regulační energie ..	47
8.4 Horní a dolní omezení v čase $t \in \Omega^{TF}$ v případě plné aktivace kladné (resp. záporné) regulační energie	47
10.1 Hodnoty MSE jednotlivých kroků „Many to One“ a „Many to Many“ architektury	76
10.2 Porovnání MSE prediktorů ARIMA a LSTM pro jednotlivé kroky predikce	77
11.1 Parametry výrobních jednotek při dynamickém testování	84
11.2 Parametry akumulátorů při dynamickém testování	84
11.3 Náklady na provoz	88
11.4 Parametry výrobních jednotek při dynamickém testování	89
11.5 Parametry akumulátorů při dynamickém testování	89
11.6 Náklady na provoz	93
11.7 Výrobní jednotky v agregčním bloku E.ON	94
11.8 Náklady na provoz	97
11.9 Výrobní jednotky v agregčním bloku ČEZ	98
11.10 Akumulátory v agregčním bloku ČEZ	98
11.11 Náklady na provoz	102

Kapitola 1

Úvod

V současné době je lidská civilizace již plně závislá na dodávkách elektrické energie. Elektrická energie je pro nás již nepostradatelnou komoditou a přerušení jejích dodávek by mělo na civilizovanou společnost fatální důsledky. Lidstvo neustále přichází s novými technologiemi výroby nebo akumulace elektrické energie. Nové technologie výroby se snaží efektivně využít tzv. zelené zdroje, které by umožnily výrobu elektrické energie s nižším negativním dopadem na životní prostředí. Tyto zdroje ovšem přináší jistou nevýhodu, jelikož nejsou plně říditelné a jejich výroba tak přináší určitý stupeň neurčitosti. Tato neurčitost může způsobit odchylku od plánovaných dodávek elektrické energie, kterou je třeba regulovat pomocí říditelných zdrojů. Nové technologie akumulace pak přináší možnost efektivněji hospodařit s elektrickou energií a dnes jsou již akumulátory nezbytnou součástí distribuční soustavy elektrické energie. S novými technologiemi přichází i stále větší spotřeba elektrické energie, která sebou přináší i svá úskalí.

V rámci celosvětové rostoucí spotřeby elektrické energie dochází ke stále většímu tlaku na řízení zdrojů energie a udržení stability dodávek elektrické energie. S rostoucí nároky se neustále snižují intervaly zadávání jednotlivých požadavků, což vnáší do řízení jednotlivých zdrojů značnou složitost. V rámci zvyšující se frekvenci příchozích požadavků dochází postupně na energetickém trhu k vytváření nových pravidel pro dodávky elektrické energie a provoz jednotlivých zdrojů. V rámci nových pravidel se objevuje možnost virtuálního distribuovaného zdroje složeného z více reálných zařízení. Tento virtuální zdroj v energetické soustavě tak spojuje několik oddělených reálných zdrojů do jednoho celku a je s ním nadále zacházeno jako s jediným zdrojem elektrické energie.

Jednotlivé zdroje tvořící virtuální zdroj již nepodléhají přímo pokynům centrálního řízení, a jejich řízení se tak odehrává čistě v rámci distribuovaného virtuálního zdroje. V rámci virtuální jednotky má tedy vlastník této distribuované soustavy volnost v požadavcích na jednotlivé jednotky, pokud virtuální jednotka splní limitní požadavky dané pravidly dodávek elektrické energie. Tato volnost přináší možnost optimalizovaného řízení z hlediska maximalizace zisku majitele soustavy distribučních jednotek. Optimalizace

tak může spočívat v efektivním rozvrhování jednotlivých zdrojů pro dané časové intervaly, nebo pomocí efektivního řídicího algoritmu, který dokáže optimálně přerozdělovat výkonové požadavky v rámci distribuovaného zdroje.

Cílem této diplomové práce je návrh optimalizačního algoritmu, který je schopen efektivně řídit distribuovanou soustavu zdrojů elektrické energie s cílem maximalizace zisku poskytovatele elektrické energie. Práce představuje prvotní studium problematiky a zabývá se zde pouze plně říditelnými zdroji elektrické energie. Neuvažují se zde neříditelné obnovitelné zdroje jako jsou větrné nebo fotovoltaické elektrárny. V rámci práce si popíšeme pravidla dodávek elektrické energie, systémový popis jednotlivých zdrojů, pravidla pro dodávky regulační energie. Následně využijeme tyto poznatky pro sestavení optimalizační úlohy, která bude schopna nalézt optimální rozložení výkonů pro distribuovaný zdroj. Následně navrhne algoritmus, který bude schopen efektivního řízení tohoto distribuovaného zdroje v reálném čase a jím nalezená rozložení porovnáme s rozložením nalezeným pomocí offline optimalizační úlohy. Algoritmus bude testovaný na datech z reálného provozu, která nám pro tuto práci poskytla společnost OSC a.s.

Kapitola 2

Pravidla trhu s energiemi

V rámci této kapitoly se zaměříme na popis trhu s energiemi. Uvedeme si jednotlivé účastníky trhu, jejich práva a povinnosti a zároveň nastíníme problematiku rozvodné sítě, jejíž stabilitu je potřeba dodržovat. V závěru kapitoly vysvětlíme princip nových pravidel trhu, které nám umožňují zavést takzvaný agregační blok, jehož optimalizace bude hlavním předmětem této práce.

2.1 Princip trhu

Elektrická energie je specifickou komoditou, jelikož je přímo odebíraná ze sítě, což znamená že o její potřebě se rozhoduje v reálném čase. Hlavním problémem elektrické energie je velmi obtížná skladovatelnost (jedinou možností jsou akumulátory omezené kapacity), což znamená, že vyrobená energie musí být ihned spotřebována. Druhým specifickým elektrické energie je způsob její dopravy, jelikož v síti není možné přímo určit výrobce a odběratele.

Z důvodu prakticky nemožného uskladnění, je třeba pro zajištění stability rozvodné sítě zajistit aby byl odebíraný výkon ze sítě roven rozdílu výkonu vyráběného a výkonových ztrát. V rámci bodové sítě, lze tuto podmínku je možno vyjádřit pomocí jednoduché rovnice

$$P_t^{G-} = P_t^{G+} - K \frac{df}{dt} - P_t^{LOSS}. \quad (2.1)$$

P_t^{G-} je výkon odebíraný ze sítě v čase t , P_t^{G+} je výkon dodávaný do sítě v čase t , K je výkonové číslo soustavy, f je frekvence v síti a P_t^{LOSS} jsou výkonové ztráty v čase t . V případě, že do sítě bude dodáván vyšší výkon než je odebíraný, tak by docházelo k přebytku výkonu, což se projeví zvyšováním frekvence v rozvodné síti. Naopak při odebírání vyššího výkonu, než je do sítě dodáván by znamenalo nedostatek výkonu a snižování frekvence v síti. Oba tyto stavy jsou nežádoucí a bez zásahu mohou vést postupně až k výpadku dodávky elektřiny a rozpadu sítě (tzv. blackout). Je tedy potřeba zajistit bilanci mezi výrobou a spotřebou elektrické energie.

Posledním významným účastníkem je **provozovatel přenosové soustavy** (dále jen provozovatel). Provozovatel je zodpovědný za stabilitu rozvodné sítě. Přímo komunikuje s poskytovatelem a odběratelem a na základě odběru dává povely poskytovateli k dodávkám výkonu. Provozovatel má ve své soustavě výše uvedený dispečink, který je zodpovědný za správné řízení dodávek regulační energie. Provozovatel tak dává poskytovateli požadavky na dodávky výkonu, aby zajistil stabilitu rozvodné sítě. V ČR je provozovatelem skupina ČEPS.

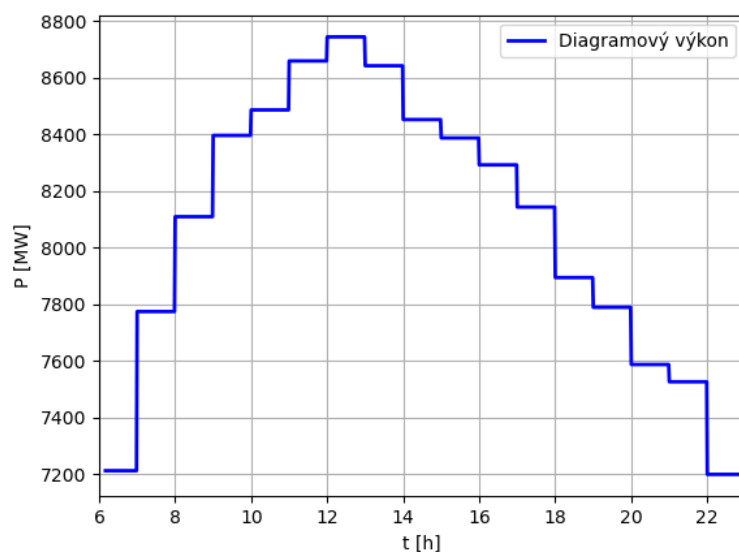
2.3 Stanovení dodávek plánované a regulační energie

Výsledné stanovení dodávek elektrické energie se odehrává na denním trhu s elektrickou energií. Pro každý den jsou stanovené obchodní intervaly (v ČR se jedná o 96 čtvrt hodinových intervalů) pro které se naplňuje dodávka výkonu. V rámci obchodního intervalu tak jde o předběžný odhad, kolik energie se v daném intervalu spotřebuje. Tyto odhady spočívají v poptávce na straně odběratele a také dle dlouhodobých statistik. Pro každý obchodní interval se na energetické burze vytváří aukce na které se sjednávají obchody pro výrobce i odběratele.

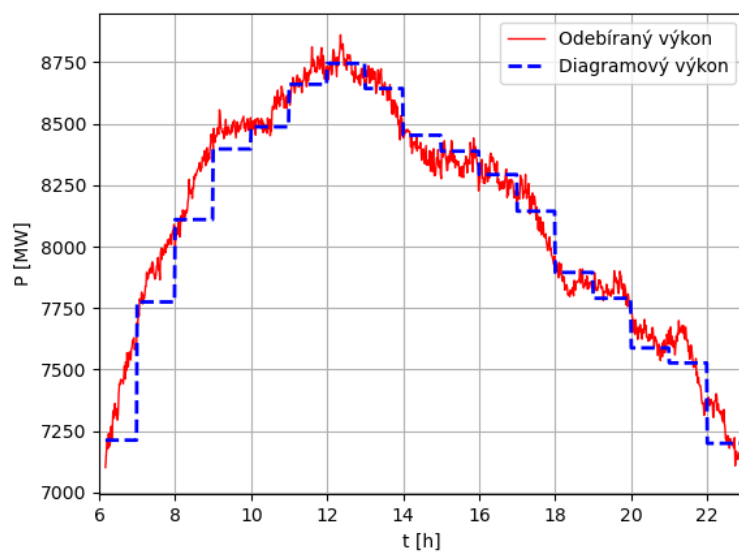
Zmíněný odhad dodávaného výkonu se nazývá **Diagramový výkon** (P_t^{DG}). V časovém horizontu se jedná o nespojitou schodovitou funkci, kde pro každý obchodní interval je tento výkon konstantní. Při praktické realizaci dodávek se diagramový výkon mění spojitě po rampovém přejezdu, jehož průběh je symetrický na rozmezí dvou obchodních intervalů. Na základě trhu se vyhodnotí rozdělení mezi jednotlivé výrobce a ti se pak v daném obchodním intervalu zavazují tento výkon do sítě dodávat. Na obrázku 2.1 je zobrazena ukáзка průběhu diagramového výkonu pro jednotlivé obchodní intervaly.

Na denním trhu se rovněž obchoduje i s regulační energií. Princip regulační energie spočívá v zálohách, které se dle potřeby aktivují a slouží k regulaci odchylky vzniklé vyšším nebo nižším odběrem výkonu, než odpovídá smluvněmu diagramovému výkonu. Obdobně jako u diagramového výkonu se i u regulační energie obchoduje v rámci jednotlivých obchodních intervalů. Regulační energie se prodává ve formě takzvaných podpůrných služeb (více se o těchto službách dozvíme v příslušné kapitole). Existuje více typů podpůrných služeb, na jejichž základě má provozovatel přenosové soustavy právo dávat požadavky na dodávky výkonu poskytovateli elektrické energie. Podle toho jakou službu poskytovatel v daném obchodním intervalu poskytuje určuje charakter požadavků provozovatele a frekvenci jejich zadávání. Problematiku podpůrných služeb rozebereme v kapitole 4.

Pro lepší představu si ještě uvedeme jak vypadá skutečný výkon dodávaný do sítě na základě aktuálního odběru ze sítě. Tato situace je zobrazena na obrázku 2.2. Vidíme, že skutečný odebíraný výkon se liší v každém obchodním

**Obrázek 2.1:** Diagramový výkon

intervalu, ale diagramový výkon nám slouží jako dobrý odhad a střední hodnota odebíraného výkonu odpovídá střední hodnotě diagramového výkonu. Odchytky způsobené vyšším či nižším odběrem byly kompenzovány pomocí regulační energie.

**Obrázek 2.2:** Porovnání diagramového výkonu a výkonu odebíraného ze sítě

Kapitola 3

Energetické jednotky na straně poskytovatele

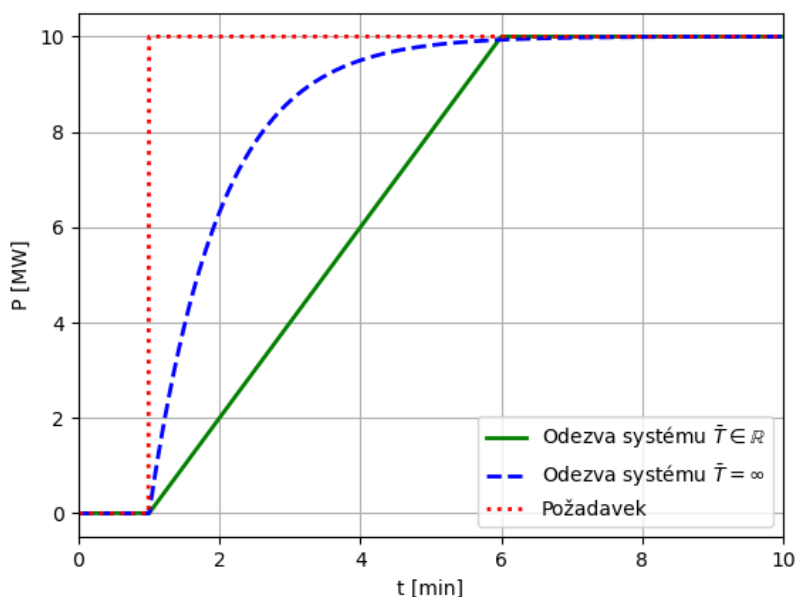
V následující kapitole si uvedeme princip základních typů jednotek poskytovatele. Uvedeme si jejich obecné technické limity a systémový popis. Zároveň si uvedeme i základní informace o jednotlivých zdrojích z hlediska způsobu výroby elektrické energie. Představíme si základní výhody, nevýhody a případná rizika jednotlivých zdrojů. V závěru kapitoly si vysvětlíme pojem fiktivní elektrárny, která nám umožňuje spojit výše uvedené zdroje dohromady a způsob jejího řízení.

3.1 Výrobní jednotka

Pod pojmem výrobní jednotka rozumíme libovolný zdroj elektrické energie, který umíme řídit a který dodává elektrickou energii do sítě. Výrobní jednotka je schopna elektrickou energii pouze vyrábět. Což znamená že její výkon dodaný do sítě bude vždy kladný (v opačném případě by se jednalo o spotřebič). Pod pojmem výrobní jednotka si lze představit například elektrárnu. Nutno podotknout, že elektrárna obsahuje několik výrobních jednotek (bloky, turbíny) a výsledný výkon dodávaný elektrárnou je pak superpozicí jednotlivých výkonů dodávaných těmito výrobními bloky.

Ve výrobní jednotce dochází pomocí přeměny nějakého jiného druhu energie (tepelné, mechanické, atd.) na energii elektrickou pomocí generátoru. Pokud se jedná o energii tepelnou, pak zdroj energie představuje palivo (plyn, uhlí, biomasa nebo štěpný materiál), jehož zpracováním dochází ke zvyšování tepla a konání práce. Vztah výkonu a ceny provozu nám pak u takovéto elektrárny uvádí nákladová křivka, která charakterizuje kolik paliva a za jakou cenu při provozu spotřebujeme. Obecně budeme uvažovat lineární charakteristiku výrobní jednotky a její nákladovou křivku pak můžeme vyjádřit pomocí rovnice

$$C_t = a + b \cdot P_t \quad (3.1)$$



Obrázek 3.1: Porovnání systémové odezvy systému s omezeným a neomezeným trendem

výrobních jednotek, kde doba jejich aktivace je menší než perioda příchozích výkonových požadavků.

Velikosti jednotlivých technických parametrů jsou silně ovlivněny tím, jaký typ výrobní jednotky se jedná. V části 3.3 budeme jednotlivé typy stručně charakterizovat a uvedeme si jejich obecné vlastnosti. V následující sekci se zaměříme na charakteristiku akumulátorů.

3.2 Akumulátor

Akumulátor je označení pro jednotku, která je v sobě schopna uchovávat elektrickou energii omezené kapacity. Obdobně jako u výrobní jednotky, zde dochází k přeměně elektrické energie na jiný druh energie, který již jsme schopni uchovávat (mechanická, chemická, atd.) a zároveň musíme být schopni udělat i opačnou přeměnu, tedy z akumulované energie vyrobit energii elektrickou. Akumulátor tak dokáže vlivem těchto přeměn elektrickou energii vyrábět (resp. spotřebovávat) a dokáže tak do sítě dodávat (resp. ze sítě odebírat) energii. Akumulátor je tedy schopen se v soustavě jednotek chovat jako zdroj elektrické energie (v případě poskytování akumulované energie), nebo jako spotřebič (v případě akumulace energie), přičemž množství vyrobené (resp. akumulované) energie je limitováno jeho kapacitou.

Akumulátor je tedy schopen pracovat ve 2 režimech. V režimu poskytování

Akumulátor je tedy jednotka, která je schopná ze sítě výkon odebírat, ale i dodávat na základě výše uvedených technických omezení. Poslední důležité omezení naší definice akumulátoru je, že nebudeme uvažovat současné nabíjení a vybíjení. Akumulátor tedy můžeme v jeden okamžik pouze nabíjet nebo pouze vybíjet. U některých typů akumulátorů může provoz obou režimů vyžadovat náklady definované pomocí nákladové křivky obdobně jako (3.1) máme

$$C_t^c = a + b \cdot P_t^c, \quad (3.14)$$

$$C_t^d = a + b \cdot P_t^d, \quad (3.15)$$

kde C_t^c představuje cenu nabíjení v čase t a C_t^d představuje cenu vybíjení v čase t . Nyní se pojďme zaměřit na charakteristiku výrobních jednotek a akumulátorů dle typů zdroje energie, ze kterého vzniká energie elektrická, nebo se do něj elektrická energie přeměňuje pro akumulaci.

■ 3.3 Charakteristika zdrojů dle typu energie

Doposud jsme hovořili o jednotkách v soustavě poskytovatele v obecném rámci. Zavedli jsme pojem zdroj a akumulátor. Uvedli jsme si jejich základní systémový popis, princip fungování a jejich obecné technické limity, ale nespécifikovali jsme co ve skutečnosti tyto zdroje představuje. V reálném systému se pod pojmem výrobní jednotka jedná o výkonný blok elektrárny, nebo přímo o elektrárnu jako celek. Akumulátor je pak jakýkoliv systém schopný přeměny elektrické energie na energii v akumulované formě a zároveň je schopen i zpětné přeměny. V našem případě se jedná o bateriovou stanici nebo o přečerpávací vodní elektrárnu, ale v současné době již existuje více forem akumulátorů, které se začínají v energetice využívat. Jednotlivé typy zdrojů a akumulátorů odpovídají našemu systémovému popisu, ale je dobré si zde uvést jejich základní charakteristiku pro lepší pochopení jejich dynamiky a jejich následnému využití v soustavě jednotek poskytovatele.

■ 3.3.1 Tepelné elektrárny

Pod pojmem tepelná elektrárna definujeme obecně elektrárnu, která využívá k výrobě elektrické energie přeměnu tepla. Toto teplo vzniká zpracováním paliva a to buď pomocí jaderné reakce, kde dochází k štěpení radioaktivního materiálu, nebo spalováním. V případě štěpení hovoříme o jaderných elektrárnách a v případě spalování se jedná o uhelné elektrárny, paroplynové, elektrárny zpracující biomasu, nebo ve speciálním případě i kogenerační elektrárna. Všechny tepelné elektrárny potřebují ke své činnosti palivo, které je potřeba dodávat a od ceny spotřebovaného paliva se následně odvíjí i cena provozu elektrárny. Každá tepelná elektrárna je tak charakterizována svou nákladovou křivkou.

Vodní elektrárny mají ale obrovskou výhodu z hlediska své dynamiky. Jelikož vodní elektrárna využívá k výrobě energie přeměny z mechanické energie, tak je možné elektrárnu velmi rychle najíždět a odstavovat spuštěním nebo zastavením toku. Oproti tepelným elektrárnám zde nedochází k tepelné výměně, která je oproti změně toku dosti pomalá. Nevýhodou vodních elektráren je omezená zásoba paliva v podobě obsahu vodní nádrže. Vodní elektrárna je tak limitována omezeným časem, po který je schopná výkon dodávat a pak se musí čekat na doplnění nádrže na požadovanou úroveň. V rámci našeho problému nebudeme toto omezení uvažovat. Předpokládáme, tedy že elektrárna má pro naše potřeby dostatek vody na výrobu elektrické energie po celou dobu provozu.

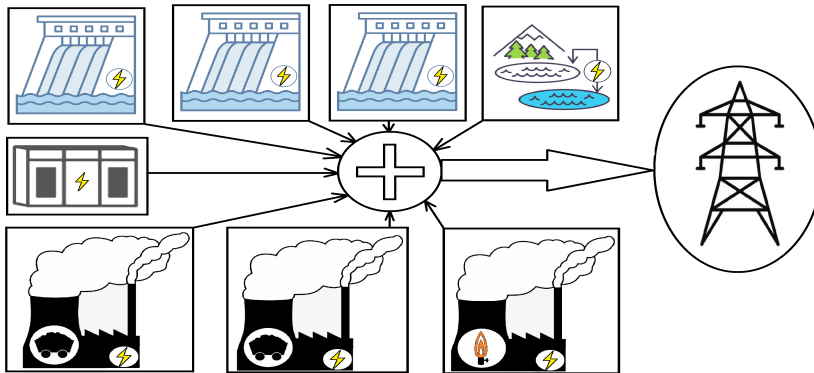
■ 3.3.3 Přečerpávací vodní elektrárna

Přečerpávací vodní elektrárna slouží v soustavě jednotek poskytovatele jako akumulátor energie. K výrobě elektrické energie využívá stejný systém jako přehradní vodní elektrárna, ale její výhodou je, že může díky čerpání vody v opačném směru energii za síť i odebírat. Princip fungování elektrárny spočívá v principu výměny vody mezi dvěma nádržemi, kde jedna je položena ve vyšší nadmořské výšce. Pokud dochází k čerpání vody z dolní nádrže do nádrže horní, tak elektrárna koná práci a odebírá výkon ze sítě. Opačný chod se nazývá turbínování při kterém naopak dochází ke konání práce, kdy tekoucí voda z horní nádrže do dolní roztáčí turbínu.

Princip akumulace energie spočívá v uchovávání vody v horní nádrži. Když přijde požadavek na dodávání výkonu a přečerpávací vodní elektrárna disponuje dostatečným množstvím vody v horní nádrži, tak se zpřístupní turbínování a elektrárna začne dodávat výkon do sítě. Stav nabití zde odpovídá množství vody v horní nádrži a limituje čas, ve kterém je elektrárna schopna dodávat (resp. odebírat) výkon. Obdobně jako přehradní vodní elektrárna má i přečerpávací vodní elektrárna velmi dobrou dynamiku v podobě vysoké hodnoty trendu. Oproti přehradní vodní elektrárně však může přečerpávací vodní elektrárna disponovat velmi velkým výkonem, jelikož zde se jedná o umělou konstrukci nádrží a nejsme zde limitováni lokací vodního toku. Přečerpávací vodní elektrárnu jde ovšem zkonstruovat i na elektrárně přehradní, pokud se pod přehradou bude nacházet nádrž, ze které bude možno vodu čerpat zpět do přehrady.

■ 3.3.4 Baterie

Baterie rovněž slouží v soustavě jednotek poskytovatele jako akumulátor elektrické energie. Využívají principu přeměny elektrické energie na energii chemickou, kterou jsme schopni ukládat v elektrolytických člancích. Tyto baterie fungují jako klasické baterie, které známe z běžného života, jenom mají mnohonásobně vyšší kapacitu a jsou schopné dodávat podstatně vyšší výkon. Princip akumulace elektrické energie tak spočívá v nabíjení baterie, tedy ve zvyšování jejího elektrochemického potenciálu. Stejně jako přečerpávací



Obrázek 3.2: Schéma agregáčnho bloku tvořenho dvěma uhelnmi elektrárnami, paroplynovou elektrárnou, přečerpávac elektrárnou třmi vodnmi elektrárnami a bateriovou stanic

S rostoucí frekvenc, se kterou pcházjí požadavky na aktivaci regulačnch zloh vak dolo v energetice k monosti využívat tzv. Agregáčn bloky (Pozn.: V literatue se Agregáčn blok tak vyskytuje pod pojmem Virtuln elektrrna). Agregáčn blok pedstavuje systm vrobnch jednotek a akumulátr, které v soustav dispečinku pedstavuj jednu vrobn jednotku. Jedn se tak o distribuovan systm, na kter pchz jeden požadavek a k jeho splnn je poteba agregovat vkon jednotlivch prvk tohoto systmu. Agregáčn blok tak dv poskytovateli monost rozhodnout o provozu svch vrobnch jednotek. M tedy monost vhodn perozdlit vkon mezi soustavu svch jednotek tak aby bylo vyhovno požadavkm dispečinku na agregáčn blok. Agregáčn blok tedy umoňuje poskytovateli monost optimalizace, tak aby maximalizoval svj zisk, za podmnky, že spln sjednan dodvky elektrické energie. Znzornn propojen vice vrobnch jednotek a akumulátr do jednoho agregáčnho bloku je na obrzku 3.2

Z hlediska diagramovho vkonu, kter je znm na kad obchodn interval se jedn o deterministick problm, kter je snadno řeiteln pomocí linernho celočselnho programovn (pipadn kvadratickho celočselnho programovn, pokud jsou nkladov kivky kvadratick). Problm spočv v poskytovn regulačnch zloh, jejich povaha je stochastick. Objem regulačnch zloh se toti odv od rozdlu plnovan spoteby a aktuln spoteby, která zpsobuje odchylku frekvence v rozvodn sti a vyžaduje okamžitou regulaci. Velikost tto odchylky nejsme schopni dopedu určit a rozdl plnovan spoteby a aktuln spoteby nm tak vytv nejistoty. Regulačn zlohy se poskytuj ve form podprnch slueb, které se dl do ty skupin a budou vysvtleny v nsledujc kapitole. Na zklad charakteru tchto podprnch slueb nsledn navrhnme algoritmus rozdlovn vkonovch požadavk, tak aby byly splnny podmnky na dodn požadovanho vkonu a zrove se pokusme maximalizovat zisk poskytovatele.

Kapitola 4

Podpůrné služby

V kapitole 2 jsme se zmínili o diagramovém výkonu a trhu s elektřinou. Diagramový výkon nám představuje hrubý odhad výkonu, který bude potřeba do sítě dodat v daném časovém intervalu. Byť je tento odhad mnohdy velmi přesný tak pro zajištění stability rozvodné sítě nestačí. Z dlouhodobého hlediska jsme schopni odhadnout střední hodnotu odebíraného výkonu ze sítě, ale jeho aktuální hodnota bude téměř vždy odlišná. Vzniká nám tedy odchylka od diagramového výkonu, kterou je třeba regulovat dodáním (resp. odebráním) určitého množství výkonu do sítě (resp. ze sítě). Toto množství energie je potřeba mít v požadované době k dispozici. Hovoříme tak o regulačních zálohách, které je možné dle potřeby aktivovat, abychom regulovali výkonovou odchylku. Zálohy poskytujeme ve formě podpůrných služeb. Máme více typů služeb zajišťující dodávky regulačních záloh, které se dělí podle množství energie a doby, do které je nutné požadovanou regulační zálohu poskytnout.

Podpůrné služby tak slouží k zajištění výkonové rovnováhy v síti. V kapitole 2 jsme uvedli, že stejně jako v případě diagramového výkonu se i v případě regulačních záloh jejich poskytování vztahuje na sjednané obchodní intervaly, kdy se poskytovatel zavazuje k jejich dodání a v případě nedodání je pokutován. V případě opakovaného nedodání regulačních záloh může dojít i k odebrání licence poskytovateli. Podpůrné služby se dělí do 2 kategorií, první kategorií jsou "služby zajištění výkonové rovnováhy" a druhou kategorií jsou "služby obnovení dodávek výkonu". Druhá kategorie slouží k co nejrychlejšímu obnovení dodávek elektrické energie v případě blackoutu, ovšem v rámci problematiky této diplomové práce se touto skupinou zabývat nebudeme. První kategorie služeb se dělí na 4 skupiny, které si v rámci této kapitoly popíšeme.

Jedná se o Automatickou regulaci frekvence (FCR), Regulaci výkonové rovnováhy s automatickou aktivací (aFRR), Regulaci výkonové rovnováhy s automatickou aktivací (mFRR) a Zálohy pro náhradu (RR). Rozdělení služeb zajištění výkonové rovnováhy je uvedeno v tabulce 4.1.

Název	Popis služby	Doba aktivace
FCR	Automatická regulace frekvence.	Okamžitá
aFRR	Regulace výkonové rovnováhy s automatickou aktivací	7.5 minuty
mFRR	Regulace výkonové rovnováhy s manuální aktivací	12.5 minuty
RR	Zálohy pro náhradu	30 minut

Tabulka 4.1: Služby výkonové rovnováhy

V rovnici (2.1) jsme uvedli základní podmínku zajištění stability sítě. Výkon dodávaný do sítě P_t pak tedy bude odpovídat následujícímu vztahu

$$P_t = P_t^{DG} + P_t^{FCR} + P_t^{aFRR} + P_t^{mFRR} + P_t^{RR}. \quad (4.1)$$

P_t^{DG} odpovídá diagramovému výkonu v čase t , P_t^{FCR} představuje dodávky regulačních záloh službou FCR v čase t , P_t^{aFRR} odpovídá regulačním zálohám dodaných službou aFRR v čase t , P_t^{mFRR} jsou regulační zálohy dodané službou aFRR v čase t a P_t^{RR} jsou regulační rezervy dodané službou RR v čase t . Velikost výkonových záloh pro podpůrné služby v dané elektrizační soustavě se stanovuje pomocí mnoha parametrů na základě statistických dat o průběhu spotřeby a výroby elektrické energie. Pro naše účely není stanovení objemu důležité a čtenáře pro tyto účely odkážeme na Kodex Přenosové soustavy ČEPS [2] kde je tato problematika plně popsána.

4.1 FCR

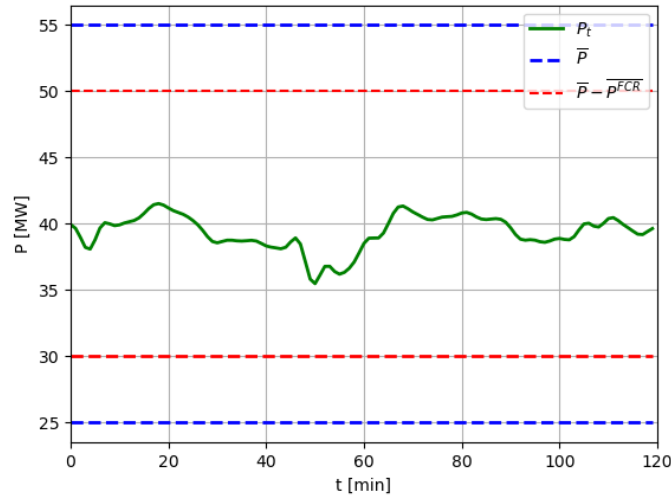
FCR představuje plně automatickou regulaci, která spočívá v přesně definované změně výkonu způsobené aktuální změnou frekvence. Jedná se tedy o zpětnovazební P regulátor. Aktuální změnu výkonu udává rovnice

$$P_t^{FCR} = -\frac{100}{S} \frac{P_n}{f_n} \Delta f_t = -K \Delta f_t. \quad (4.2)$$

Kde P_t^{FCR} představuje požadovanou změnu výkonu, Δf odpovídá aktuální změně frekvence a člen S představuje statistiku korektoru frekvence. P_n odpovídá nominálnímu výkonu výrobní jednotky a f_n je jmenovitá hodnota frekvence, v našem případě 50 Hz. Maximální hodnota FCR nastává při změně frekvence o 200 mHz od zadané hodnoty. Poskytovatel je povinen zajistit požadovanou regulační energii pro FCR do 30 sekund od aktivace. Změna výkonu je omezená maximální velikostí FCR, která představuje celkový objem držených záloh. Platí tak vztah

$$P_t^{FCR} \in \langle -\bar{P}^{FCR}, \bar{P}^{FCR} \rangle, \quad (4.3)$$

kde \bar{P}^{FCR} představuje maximální objem záloh FCR a při jejím dosažení se jedná o plnou aktivaci FCR. Minimální velikost zálohy pro \bar{P}^{FCR} je 3 MW a maximální velikost je 10 MW na jednotku poskytovatele.



Obrázek 4.1: Ukázka omezení zadávacího pásma vlivem FCR a průběh výkonu na výrobní jednotce

FCR je plně automatická služba, která má za cíl korigovat nejmenší odchylky od plánovaného výkonu. Jejich průběh je plně stochastický a nejedná se o stacionární proces. FCR je zajišťována technickou konstrukcí elektrárny pomocí obvodů zajišťující automatickou regulaci frekvence. Z hlediska našeho problému je pro nás důležitá jen její maximální velikost \bar{P}^{FCR} , která nám bude ovlivňovat velikost pásma ve kterém budeme moct výkon zadávat, ale pouze v rámci daného bloku tedy

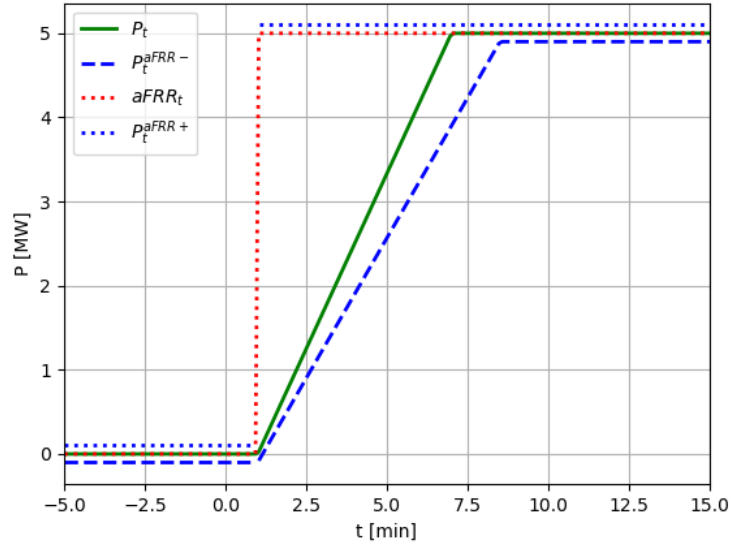
$$\bar{P}^* = \bar{P} - \bar{P}^{FCR}, \quad (4.4)$$

$$\underline{P}^* = \underline{P} + \bar{P}^{FCR}. \quad (4.5)$$

Kde \bar{P}^* představuje maximální možný zadávaný výkon a \underline{P}^* je minimální možný zadávaný. Znázornění omezení zadávacího pásma je na obrázku 4.1. Pro přehlednější značení si zavedeme úmluvu. Pokud bude v následujícím textu uvedeno \bar{P} (resp. \underline{P}) tak máme na mysli \bar{P}^* (resp. \underline{P}^*). Dále se však touto službou již zabývat nebudeme.

4.2 aFRR

Tato služba představuje poskytování záloh s automatickou aktivací. Oproti FCR zde probíhá aktivace služby a to na základě požadavků regulátoru frekvence a salda předávaných výkonů. Služba se realizuje změnou hodnoty dodávaného výkonu a její velikost je dána algoritmem regulátoru dispečinku ČEPS. Změna výkonu může být kladná ($aFRR^+$) nebo záporná ($aFRR^-$) a poskytovatel elektrické energie je zavázán k její realizaci do 7.5 minut od požadavku. Tato doba se nazývá doba plné aktivace služby aFRR. V



Obrázek 4.2: Ukázka požadavku na aFRR, odezva systému a meze pro splnění požadavku

rámci doby aktivace služby se musí hodnota postupně aktivovaného výkonu pohybovat v intervalu

$$P_t^{aFRR} \in \langle P_t^{aFRR-}, P_t^{aFRR+} \rangle, \quad (4.6)$$

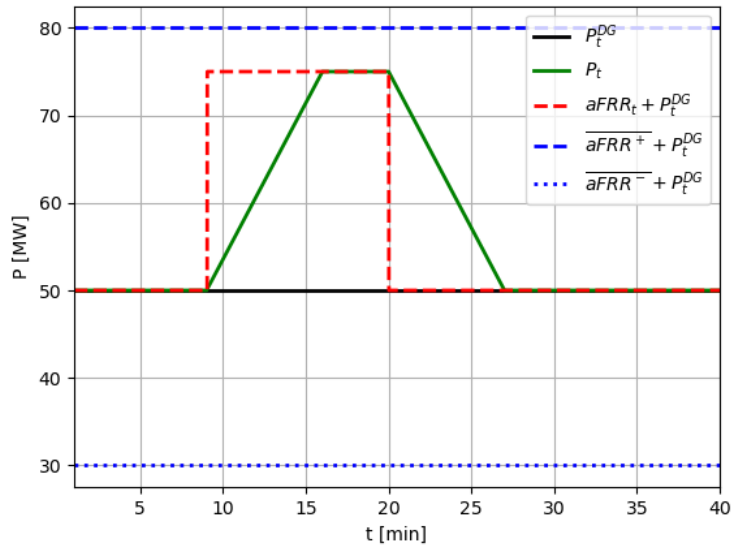
kde

$$P_t^{aFRR-} = \min \left\{ aFRR_t, \frac{aFRR_t - P_{t_a}}{7.5} (t - t_a) + P_{t_a} + \frac{aFRR_t - P_{t_a}}{7.5} t_a \right\} - \epsilon, \quad (4.7)$$

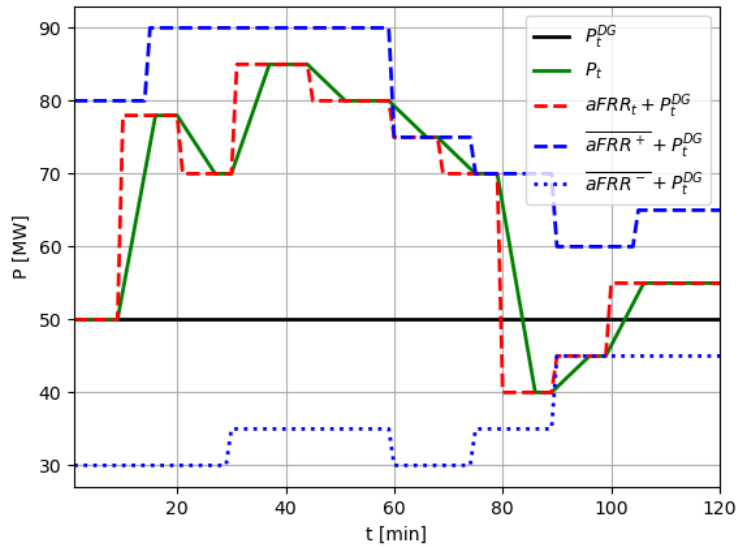
$$P_t^{aFRR+} = \max \left\{ aFRR_t, \frac{aFRR_t - P_{t_a}}{7.5} (t - t_a) + P_{t_a} + \frac{aFRR_t - P_{t_a}}{7.5} t_a \right\} + \epsilon. \quad (4.8)$$

Údaj $aFRR_t$ zde představuje požadavek na aktivaci služby aFRR, P_{t_a} je výkon jednotky v čase aktivace t_a a ϵ představuje povolenou odchylku. Za nedodržení těchto limitů je poskytovatel penalizován. Průběh aktivace požadavku spolu se znázorněním limitního intervalu je na obrázku 4.2.

Vymezené zálohy pro poskytování služby aFRR se pohybují v intervalu $\langle \overline{aFRR}^-, \overline{aFRR}^+ \rangle$. Velikost \overline{aFRR}^+ (res. \overline{aFRR}^-) je dohodnutá na energetické burze a vztahuje se vždy k danému obchodnímu intervalu, po který bude služba poskytována. Stejně jako požadavek na aktivaci může přijít i požadavek na deaktivaci služby. V takovémto případě musí dojít k návratu dodávaného výkonu na původní hodnotu. Situace je znázorněna na obrázku 4.3. Důležité je si uvědomit, že plocha vymezená křivkami P_t^{DG} a $aFRR_t + P_t^{DG}$ má stejný obsah jako plocha vymezená křivkami P_t^{DG} a P_t . První obsah představuje požadavek na dodávku regulační energie do sítě a druhý obsah odpovídá dodanému množství energie. Tyto obsahy musí být stejné. V případě rozdílnosti



Obrázek 4.3: Znázornění povelů pro aktivaci a deaktivaci $aFRR_t$ a skutečná hodnota P_t^{DG} a výsledný výkon P_t

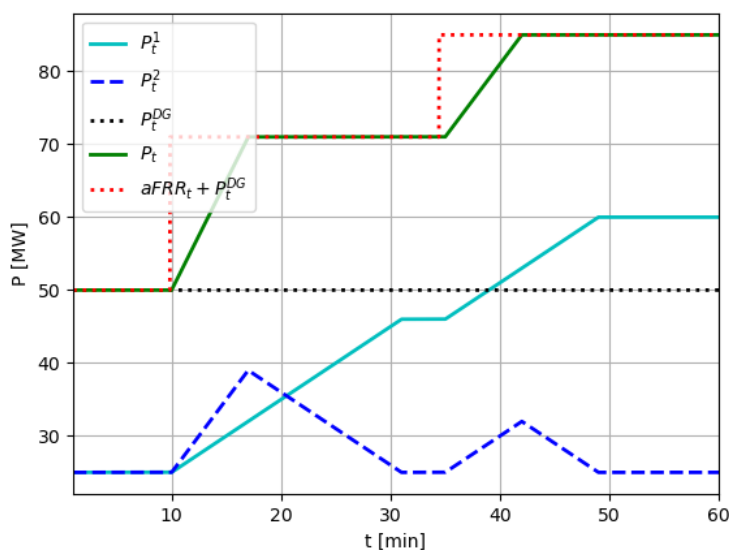


Obrázek 4.4: Znázornění povelů pro aktivaci $aFRR_t$ mezi jednotlivými obchodními intervaly

je poskytovatel pokutován za nesplnění dodávek regulační energie. Velikost limitních hodnot intervalu $(\overline{aFRR}^-, \overline{aFRR}^+)$ se může mezi jednotlivými obchodními intervaly měnit. Požadavek na aFRR, ale nikdy nemůže být vyšší (resp. nižší) než je limitní hodnota \overline{aFRR}^+ (resp. \overline{aFRR}^-). Tato situace je znázorněna na obrázku 4.4.

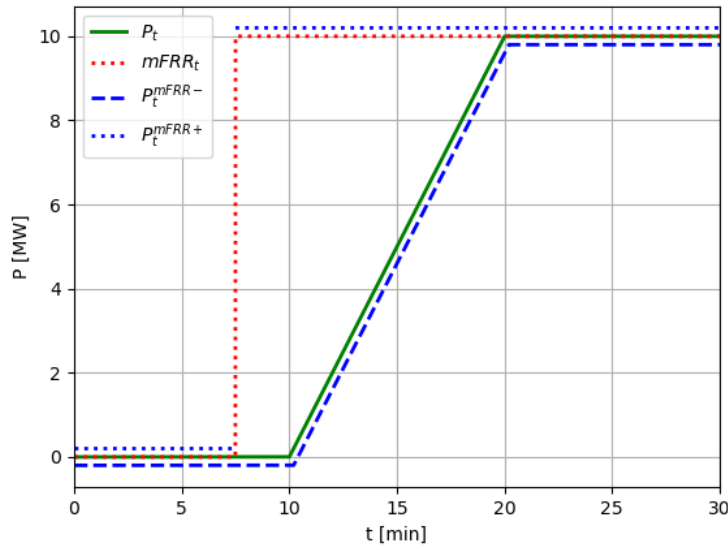
Zpravidla se při aktivaci aFRR využívá maximální zatěžování (\bar{T}), které stanovuje maximální rychlost změny výkonu. Maximálního zatěžování se využívá z důvodu co nejrychlejšího splnění požadavku aFRR. V případě poskytování aFRR na jedné jednotce, je potřeba rozmezí $(\overline{aFRR}^-, \overline{aFRR}^+)$ tak, abychom byli schopni vždy splnit jakýkoliv aktivační požadavek. U aFRR nevíme kdy může požadavek přijít. Nevíme ani jeho případnou velikost, pouze interval ve kterém se požadavek může pohybovat.

V případě agregačního bloku tvořeného více než jednou výrobní jednotkou přichází jednotný požadavek na aFRR, který daný agregační blok musí poskytnout. Rozložení uvnitř výkonu je tak vhodné optimalizovat. Uvažujme nyní příklad, kde se v soustavě nacházejí 2 zdroje, které dohromady vytváří diagramový výkon. Jeden ze zdrojů je rychlý, ale jeho využití na vyšší výkon je drahé. Druhý zdroj je pomalý, ale jeho využívání při vyšších výkonech je výhodnější než v případě zdroje rychlého. Přejde-li požadavek na aFRR pak je nutné vhodně rozložit požadavek mezi zdroje. Obrázek 4.5 zobrazuje optimální rozložení pro tuto ukázkovou situaci.



Obrázek 4.5: Reakce na požadavek aFRR z hlediska agregačního bloku tvořeného dvěma výrobními jednotkami

Zde je důležité si všimnout, že pomalý zdroj 1 využívá při změně požadavku svůj maximální trend \bar{T}^1 . Naopak rychlý zdroj 2 nastavuje svou rychlost změny, tak aby za daný časový interval došlo k dosažení požadovaného výkonu celého agregačního bloku P_t a následně se vrací zpět na původní výkon se změnou odpovídající trendu \bar{T}^1 .



Obrázek 4.6: Znázornění požadavku na mFRR s odezvou systému a mezemi pro splnění požadavku

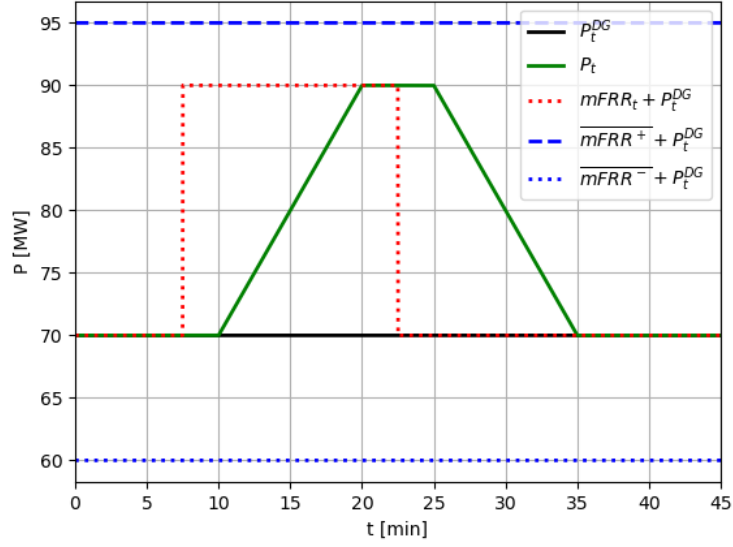
4.3 mFRR

Tato služba představuje manuálně aktivované regulační zálohy, které je poskytovatel povinen dodat do času t od požadavku na její aktivaci. Velikost t nabývá hodnot 5 minut a 12.5 minuty. mFRR tak dělíme na $mFRR_5$ a $mFRR_{12.5}$, v energetické terminologii se často objevují pod názvy "pětiminutová záloha" a "dvanácti a půl minutová záloha". Hodnota času t odpovídá maximální povolené době aktivace těchto služeb. V rámci této práce se službou $mFRR_5$ zabývat nebudeme.

Pro $mFRR_{12.5}$ se doba t skládá z 2.5 minut přípravy a 10 minut rampování na požadovanou hodnotu. Pro přehlednější značení udělme následující úmluvu. V následujícím textu označením mFRR máme na mysli $mFRR_{12.5}$. Stejně jako u aFRR je i pro službu mFRR vymezena rezerva výkonu, který se může poskytnout pro regulaci odchylky, tento interval označme jako $\langle mFRR^- ; mFRR^+ \rangle$. Interval opět nemusí být nutně symetrický. Službu mFRR dále dělíme podle časového okamžiku, kdy může přijít aktivační signál a to na mFRR s plánovanou aktivací a mFRR s přímou aktivací podle doby, kdy může přijít aktivační požadavek.

V ideálním případě by fáze přípravy a fáze rampování měla proběhnout v dohodnutých časových úsecích a náběh na požadovanou hodnotu by měl trvat 10 minut. Stejně jako v případě aFRR i zde platí pro výkon P_t^{mFRR} omezení

$$P_t^{mFRR} \in \langle P_t^{mFRR-}, P_t^{mFRR+} \rangle. \quad (4.9)$$



Obrázek 4.7: Aktivace a deaktivace mFRR

Kde P_t^{mFRR+} (resp. P_t^{mFRR-}) představují maximální (resp. minimální) hodnotu výkonové odezvy v čase t . Pro $t \in \langle t_a, t_a + 2.5 \rangle$ platí

$$P_t^{mFRR-} = \min \{mFRR_t, P_{t_a}\} - \epsilon, \quad (4.10)$$

$$P_t^{mFRR+} = \max \{mFRR_t, P_{t_a}\} + \epsilon, \quad (4.11)$$

a pro $t \in \langle t_a + 2.5, t_a + 12.5 \rangle$ platí

$$P_t^{mFRR-} = \min \left\{ mFRR_t, \frac{mFRR_t - P_{t_a}}{10} (t - t_a - 2.5) + P_{t_a} + \frac{mFRR_t - P_{t_a}}{10} (t_a - 2.5) \right\} - \epsilon, \quad (4.12)$$

$$P_t^{mFRR+} = \max \left\{ mFRR_t, \frac{mFRR_t - P_{t_a}}{10} (t - t_a - 2.5) + P_{t_a} + \frac{mFRR_t - P_{t_a}}{10} (t_a - 2.5) \right\} + \epsilon. \quad (4.13)$$

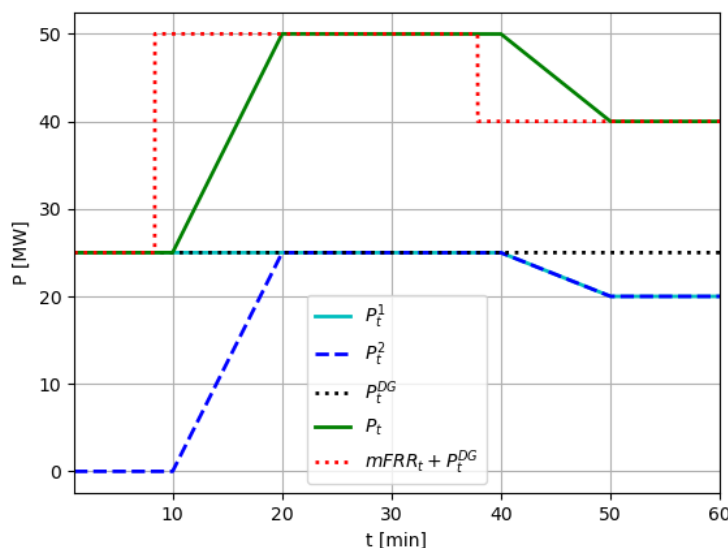
Výraz $mFRR_t$ zde představuje požadavek na aktivaci služby aFRR, P_{t_a} je výkon jednotky v čase aktivace t_a a ϵ představuje povolenou odchylku. Situace je zobrazena na obrázku 4.6.

Pojďme se nyní zaměřit na rozdíl mezi službami mFRR s přímou a mFRR s plánovanou aktivací. Služba s plánovanou aktivací může být aktivována pouze v určeném okamžiku (7.5 minuty před začátkem čtvrt hodinového obchodního intervalu), zatímco služba s přímou aktivací může být aktivována kdykoliv

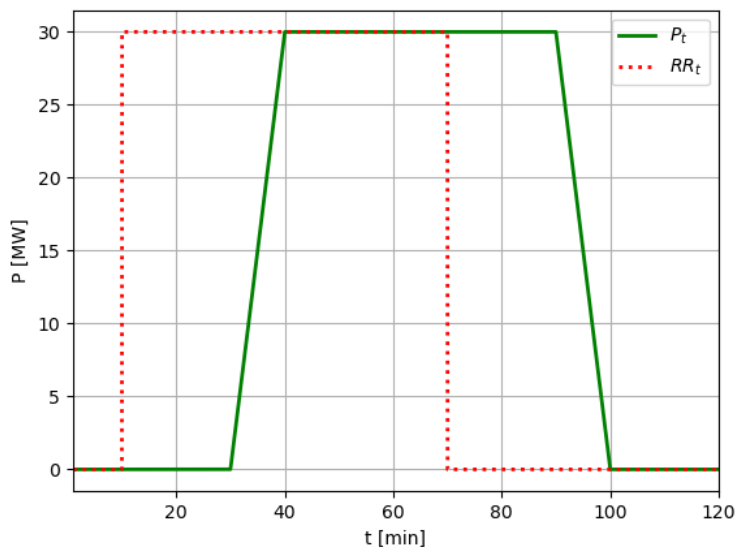
během poskytování (pozn.: Vzhledem k charakteru aktivace regulačních záloh, máme na mysli 7.5 minut před skutečným začátkem poskytování služby a 7.5 minut do konce poskytování této služby). V rámci této diplomové práce budeme uvažovat pouze službu s plánovanou aktivací.

Stejně jako u aFRR může přijít i požadavek na deaktivaci této služby. Průběh má stejný charakter jako v případě aktivace služby, tedy v ideálním případě 2.5 minuty příprava a 10 minut rampování na původní hodnotu diagramového výkonu. Požadavek na deaktivaci služby chodí rovněž 7.5 minuty před začátkem dalšího čtvrt hodinového obchodního intervalu a v případě, že na další obchodní interval již tuto službu neposkytujeme tak se služba deaktivuje automaticky. Ukázka aktivace a deaktivace služby spolu s nabízenými zálohami je zobrazena na obrázku 4.7. Důležité je poznamenat, že obdobně jako u aFRR je plocha vymezená křivkami P_t^{DG} a $mFRR_t + P_t^{DG}$ má stejný obsah jako plocha vymezená křivkami P_t^{DG} a P_t . Tento obsah určuje množství energie, který je dodán do sítě. Plochy opět představují požadavek na regulační energii a skutečnou dodávku. Opět za nesplnění dodávek regulační energie je poskytovatel pokutován.

V rámci agregačního bloku obsahující více než jednu jednotku má opět poskytovatel možnost optimálně rozdělit požadavek na mFRR mezi jednotky. Uvažujme opět modelový příklad, kde agregační blok obsahuje 2 zdroje. Nyní ovšem uvažujme případ kdy oba zdroje mají stejné technické parametry, ale na dosavadním výkonu se podílel pouze jeden, který pracoval provozním maximu a druhý byl deaktivovaný. Požadavek na mFRR způsobil aktivaci zdroje 2,



Obrázek 4.8: Reakce na požadavek mFRR z hlediska agregačního bloku tvořeného dvěma výrobními jednotkami



Obrázek 4.9: Průběh aktivace a deaktivace služby RR

který byl doteď deaktivovaný. Aktivace zdroje umožní splnění požadavku, ale během dalšího obchodního intervalu, kde dojde ke snížení požadavku na mFRR se snižuje výkon obou zdrojů. Situace je znázorněna na obrázku 4.8.

4.4 RR

RR jsou zálohy pro náhradu, realizované poskytnutím sjednané regulační energie P^{RR+} nebo P^{RR-} jednotkou do 30 minut od příkazu dispečinku provozovatele rozvodné sítě. Služba je aktivována v souladu s parametry akceptovaných volných nabídek regulační energie poskytovatele a lze jí aktivovat na pevnou čtvrt hodinu nebo násobek pevně stanovené čtvrt hodiny (minimálně 15 minut, maximálně 60 minut). Na rozdíl od mFRR se tato služba odvíjí od situace a možností poskytovatele elektrické energie, oproti povinnosti službu poskytnout v případě mFRR. Průběh služby je však obdobného charakteru jako mFRR. Rozdíl je v době aktivace kde v případě RR se jedná o 30 minut. Doba aktivace se dělí na 20 minut přípravy a 10 minut rampování. Požadavky na náběh jsou stejného principu jako v případě mFRR. Deaktivační doba obou služeb je stejná jako doba aktivační a to 30 minut. RR je poslední fáze regulace pro zajištění výkonové rovnováhy. Průběh aktivace RR a její deaktivace je zachycen na obrázku 4.9.

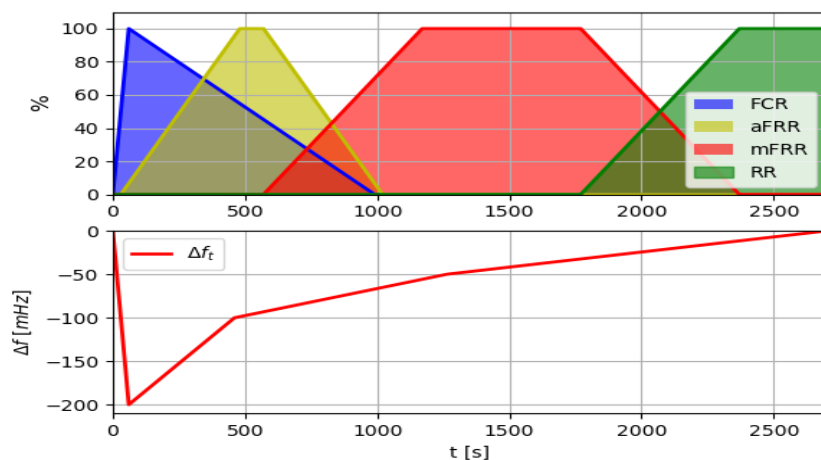
V rámci této diplomové práce se ale poskytováním služby RR zabývat nebudeme. V textu je uvedena stejně jako služba FCR z důvodu obeznámení autora se všemi druhy služeb zajištění výkonové rovnováhy. Pro podrobný popis podpůrných služeb odkážeme čtenáře na Kodex přenosové soustavy ČEPS [1], konkrétně na jeho druhou část [2] týkající se podpůrných služeb.

4.5 Shrnutí podpůrných služeb

V této kapitole jsme udělali stručný úvod do problematiky služeb zajištění výkonové rovnováhy. Ve starší literatuře se obvykle služba FCR nazývala jako "Primární Regulace", služba aFRR jako "Sekundární Regulace" a služba mFRR (všechny druhy uvedené v sekci 4.3) jako "Terciální regulace" nebo jako "Minutová Záloha". Toto pojmenování mělo svůj význam. Drobné odchylky se vždy kompenzují automaticky pomocí služby FCR, v případě, že se vyskytne větší odchylka je potřeba dodat vyšší regulační energii do sítě, aby regulační zálohy poskytované službou FCR mohly být co nejbližší středu intervalu $\langle -\bar{P}^{FCR}, \bar{P}^{FCR} \rangle$ a služba tak plnila svůj regulační účel. Tuto regulační energii představují zálohy jsou dodány pomocí služby aFRR.

Obdobná situace nastává i pro službu aFRR kde se ke kompenzaci využívají zálohy poskytované službou mFRR. Pro aFRR jsou celkové zálohy vyčleněné na soustavu všech výrobních jednotek České Republiky pevně stanoveny provozovatelem přenosové soustavy. Množství poskytovaných regulačních záloh je pak rozloženo rovnoměrně mezi všechny poskytovatele této služby, podle jejich procentuálního zastoupení celkového množství poskytovaných záloh vzhledem k celkovým zálohám. Pozn.: Nově od 1. dubna 2022 se využívá žebříčkové rozdělení kde k aktivaci potřebných záloh dochází na základě cenového žebříčku jednotlivých poskytovatelů.

Pro kompenzaci mFRR již slouží služba RR. Aktivace mFRR již není rovnoměrně rozložena mezi všechny poskytovatele jako v případě aFRR, ale k jejímu rozdělení se využívá specifického algoritmu na straně dispečinku přenosové soustavy. Způsob poskytování regulačních záloh je zobrazen na obrázku 4.11.



Obrázek 4.10: Poskytování podpůrných služeb v čase spolu s odchylkou frekvence

Kapitola 5

Souhrn vybraných prací věnující se našemu optimalizačnímu problému

Problematika Agregáčního bloku se v zahraniční literatuře vyskytuje pod pojmem Virtuální elektrárny (Virtual Power Plant). Z hlediska historie byl tento pojem poprvé zaveden v roce 2005, kdy bylo v Německu spojeno dohromady 29 zdrojů energie, které se v soustavě provozovatele rozvodné sítě sjednotily do jediné virtuální výrobní jednotky a provozovatel sítě tak měl k dispozici jen jednu jednotku, kterou mohl povelovat.

Sjednocení více výrobních jednotek a akumulátorů do jednoho agregáčního bloku je dnes na trhu s energiemi pro mnoho poskytovatelů výhodné, zejména v případě, že disponují i nespolehlivými obnovitelnými zdroji jako jsou fotovoltaické panely, nebo větrné elektrárny, které není možno přímo regulovat. Tyto zdroje není možné přímo řídit, ale v kombinaci s výše uvedenými říditelnými výrobními jednotkami je možné jejich nespolehlivost kompenzovat. Tato kompenzace dává poskytovateli možnost účastnit se trhu s regulačními zálohami, jelikož díky svým kompenzačním jednotkám je schopen poskytovat stabilní zdroj elektrické energie a je schopen garantovat splnění případných dodávek regulačních záloh. Nespolehlivost obnovitelných zdrojů tak přináší do řízení neurčitosti a metody jsou potom založené na principu stochastického programování, kde se s neurčitostmi pracuje.

5.1 Ucelený souhrn problematiky

Z hlediska ucelené literatury existuje v dnešní době mnoho odborných knih zabývajících se virtuálními elektrárnami. Jsou zde popsány metodiky zavedení a také vhodné strategie pro možnost uplatnění těchto distribuovaných energetických systémů na trhu s energiemi. Popis virtuální elektrárny spolu s popisem optimalizace na trhu s elektrickou energií a regulačními zálohami je obsažen v [3]. Dále [4] je nově vydaná kniha zaměřující se na moderní popis virtuální elektrárny a možností statistického vyhodnocování možných scénářů z hlediska nespolehlivých obnovitelných zdrojů. Jedním z prvních ucelených popisů virtuální elektrárny je obsažen v [5]

chastické programování a cílem je opět najít optimální nabídky dodávaného výkonu a poskytování regulační energie.

Mnoho prací se v rámci optimalizace zabývalo návrhem dvoustupňového algoritmu, který bude mít za cíl maximalizovat profit poskytovatele. V první fázi došlo na základě statistických analýz k vytvoření scénářů provozu na následující den a úloha byla vyřešena opět metodou stochastického programování a druhá fáze se soustředila na krátkodobou optimalizaci během následujícího dne. Tuto problematiku řešily práce [23], [24] a [25]. Tyto algoritmy se dle prací jeví jako efektivní, ale v rámci online algoritmu neuvažují optimální rozdělení z hlediska poskytování podpůrných služeb na základě přijatých požadavků z dispečinku. Jejich cílem je opět maximalizace profitu na trhu s energiemi, stochastické vlastnosti vnáší do této problematiky využití nespolehlivých zdrojů elektrické energie a je tak potřeba na základě predikcí vygenerovat scénáře pro jejich fungování. Druhá fáze algoritmu tak spočívá v optimalizaci prodeje dodávek diagramového výkonu a regulačních záloh na denním trhu. Tento přístup využívá i diplomová práce [26], kde se autor zabývá optimalizací virtuální elektrárny na trhu s energiemi.

Problematiku poskytování podpůrné služby mFRR s plánovanou aktivací diskutují autoři v [27]. Autor zde opět využívá 2 stavový algoritmus a ve své soustavě jednotek uvažuje nespolehlivé obnovitelné zdroje a akumulátory. Opět se zde využívá stochastické lineární programování, ovšem k jeho vyřešení používá Sample Average Approximation. První část algoritmu nalezne optimální množství regulační energie a druhá část algoritmu se již zaměřuje na obchodování s energií na denním trhu. V rámci strategie využívá autor scénáře, kdy dojde k plné aktivaci regulačních záloh a dále scénáře, kdy k aktivaci nedojde. Práce [28] se zaměřuje na optimalizaci poskytování služby aFRR virtuální elektrárnou tvořenou soustavou vodních elektráren. Autor hledá strategii na dlouhodobý časový horizont, kde opět využívá stochastického programování na základě údajů o stavu vodních hladin v soustavě jeho vodních elektráren. Návrh optimalizačního algoritmu zohledňující poskytování služby mFRR s plánovanou aktivací je diskutován v [29]. Opět se zde jedná o dvoustavový algoritmus, kde v prvním stavu dojde pomocí stochastického programování k nalezení optimálních hodnot pro diagramový výkon a regulační zálohy. V první fázi se opět pracuje se scénáři, které vycházejí ze statistických dat, které určují jejich pravděpodobnost. Druhá fáze algoritmu pak využívá MILP model, který se spouští v rámci periody obchodního intervalu odpovídající periodě aktivačního signálu služby mFRR a na základě predikovaných hodnot nastavuje optimální výkonové požadavky uvnitř virtuální elektrárny. Autor v práci neuvádí, kde získává virtuální elektrárna predikované hodnoty. Virtuální elektrárna zde v tomto případě představuje mikro síť se spolehlivými zdroji a odběrateli a také ze soustavy akumulátorů.

Kapitola 6

Optimalizační problém

V předchozích kapitolách jsme si zavedli základní informace o trhu s energiemi a uvedli jsme si jednotlivé účastníky a jejich práva a povinnosti v rámci dodávání (resp. odebírání energií) do rozvodné sítě (resp. z rozvodné sítě). V kapitole 3 jsme si uvedly typy jednotek v soustavě poskytovatele a jejich charakteristiku z hlediska systémového popisu a také dle typů zdrojů z hlediska výroby a akumulace elektrické energie. V závěru této kapitoly jsme si uvedli pojem agregáčního bloku, který funguje jako distribuovaná výrobní jednotka a sestává se z několika výrobních jednotek a akumulátorů. V kapitole 4 jsme charakterizovali podpůrné služby, které slouží k poskytování regulačních záloh pro zajištění stability rozvodné sítě. Charakterizovali jsme jejich mezní parametry a limitní časové intervaly, spolu s limitními průběhy dodávek regulačních záloh do rozvodné sítě. Uvedli jsme i triviální rozdělení, pokud by jednu službu poskytovaly 2 výrobní jednotky současně.

Cílem této práce je nalézt optimalizační algoritmus pro rozdělení dodávek výkonu mezi jednotlivými jednotkami soustavy poskytovatele, které budou spojeny do jednoho agregáčního bloku. Agregáční blok se tak bude chovat pro dispečink rozvodné sítě, jako jedna výrobní jednotka, což dispečinku umožňuje snadnější rozdělování požadavků mezi jednotlivé výrobní jednotky připojené do sítě. Agregáční blok, je tedy povinen splnit požadavky dispečinku na dodávky diagramového výkonu a poskytování výkonových rezerv, ale vlastní rozdělení těchto požadavků uvnitř agregáčního bloku je již na poskytovateli výkonu, vlastníci agregáční blok. Cílem vhodného přerozdělení je tedy maximalizovat zisk za poskytování diagramového výkonu a regulačních záloh.

Náš algoritmus musí tedy nalézt optimální rozložení v reálném čase při denním provozu agregáčního bloku. V rámci našeho algoritmu uvažujeme poskytování diagramového výkonu a podpůrných služeb aFRR a mFRR s plánovanou aktivací. Mezní limity pro službu FCR uvažujeme v rámci omezení technických limitů elektrárny a nadále budeme rozumět interval $\langle \underline{P}, \bar{P} \rangle$ jako mezní hodnoty výrobní jednotky (resp. akumulátoru) od které byly odečteny meze \bar{P}^{FCR} .

Celkový výkon dodávaný agregačním blokem v čase t , tak bude odpovídat následující rovnici

$$P_t = \sum_{i \in \Omega^{ER}} P_{i,t} + \sum_{k \in \Omega^{ESS}} \frac{1}{\eta_k} P_{i,t}^d - \sum_{i \in \Omega^{ESS}} \eta_k P_{i,t}^c \quad \forall t \in \Omega^T. \quad (6.1)$$

Kde výraz $P_{i,t}$ je dodávaný výkon výrobní jednotkou $i \in \Omega^{ER}$ v čase $t \in \Omega^T$. $P_{i,t}^d$ je výkon akumulátoru $i \in \Omega^{ESS}$ v čase $t \in \Omega^T$ při vybíjení, $P_{i,t}^c$ je výkon akumulátoru $i \in \Omega^{ESS}$ v čase $t \in \Omega^T$ při nabíjení a η_k je účinnost akumulátoru $i \in \Omega^{ESS}$. Výraz Ω^{ER} reprezentuje množinu všech výrobních jednotek agregačního bloku, Ω^{ESS} představuje množinu všech akumulátorů agregačního bloku a Ω^T je množina reprezentující časový horizont na kterém hledáme optimální výkonová rozložení. Označíme-li pak množinu reprezentující agregační blok jako Ω^{AB} pak pro ni platí

$$\Omega^{AB} = \Omega^{ER} \cup \Omega^{ESS} \quad (6.2)$$

Výkon dodávaný agregačním blokem, rovněž musí splňovat následující dvě nerovnice

$$P_t \leq P_t^{DG} + P_t^{aFRR+} + P_t^{mFRR+} + \epsilon \quad \forall t \in \Omega^T, \quad (6.3)$$

$$P_t \geq P_t^{DG} + P_t^{aFRR-} + P_t^{mFRR-} - \epsilon \quad \forall t \in \Omega^T. \quad (6.4)$$

Kde P_t^{DG} představuje požadavek na diagramový výkon agregačního bloku, P_t^{aFRR+} (resp. P_t^{aFRR-} , P_t^{mFRR+} , P_t^{mFRR-}) představuje horní mez pro výkon požadovaný pro službu aFRR (resp. dolní mez pro aFRR, resp. horní mez pro mFRR, resp. dolní mez pro mFRR) v čase $t \in \Omega^T$, definovanou v kapitole 4 a ϵ představuje povolenou odchylku v rámci dodávek diagramového výkonu.

Naším cílem je tedy maximalizovat zisk I (income) poskytovatele, který je definován podle následující rovnice

$$I = \sum_{t \in T} I_t^{DG} + I_t^{aFRR} + I_t^{mFRR} - \sum_{i \in \Omega^{ER}} C_{i,t} - \sum_{i \in \Omega^{ESS}} C_{i,t}^c + C_{i,t}^d, \quad (6.5)$$

kde I_t^{DG} je zisk za poskytovaný diagramový výkon v čase $t \in \Omega^T$, I_t^{aFRR} (resp. I_t^{mFRR}) je zisk za poskytování záloh pro službu aFRR (resp. mFRR). $C_{i,t}$ je cena za provoz výrobní jednotky $i \in \Omega^{ER}$ v čase $t \in \Omega^T$. $C_{i,t}^c$ (resp. $C_{i,t}^d$) je cena za nabíjení (resp. vybíjení) akumulátoru $i \in \Omega^{ESS}$ v čase $t \in \Omega^T$. Parametr $t \in \Omega^T$ pak představuje časový horizont, pro který hledáme optimální rozdělení.

Jelikož v rámci naší optimalizace nemáme za cíl ovlivnit sjednaný objem dodávek diagramového výkonu a množství poskytovaných záloh pro službu aFRR (resp. mFRR), přechází náš optimalizační úkol do minimalizace nákladů poskytovatele

$$\min C, \quad (6.6)$$

kde pro C platí rovnice

$$C = \sum_{t \in T} \sum_{i \in \Omega^{ER}} C_{i,t} + \sum_{i \in \Omega^{ESS}} C_{i,t}^c + C_{i,t}^d \quad (6.7)$$

Jelikož parametr $C_{i,t}$ (resp. $C_{i,t}^c$ a $C_{i,t}^d$) odpovídá hodnotě nákladové křivky výrobní jednotky (resp. akumulátoru) $i \in \Omega^{ER}$ (resp. $i \in \Omega^{ESS}$) v čase $t \in \Omega^T$, můžeme formálně zapsat optimalizační problém jako

$$\max C(P_{k,t}, P_{l,t}^d, P_{l,t}^c) \quad \forall k \in \Omega^{ER}, \forall l \in \Omega^{ESS} \forall t \in \Omega^T \quad (6.8)$$

Naším cílem tedy bude nalézt takové hodnoty $P_{k,t}$, $P_{l,t}^d$ a $P_{l,t}^c$ $\forall k \in \Omega^{ER}, \forall l \in \Omega^{ESS} \forall t \in \Omega^T$, tak aby bylo vyhověno požadavkům definovaných pomocí nerovnic (6.3) a (6.4) a technických provozních technických limitů všech jednotek definovaných v kapitole 3 za účelem maximalizovat zisk poskytovatele (6.5).

Vzhledem k charakteru požadavků se jedná o optimalizační problém, který je možné formulovat pomocí celočíselného lineárního programování (ILP). V našem případě se bude jednat o ILP, kde připouštíme reálné hodnoty nalezených parametrů, tedy o smíšené celočíselné programování (MILP). Naším cílem tedy bude vytvořit MILP model agregačního bloku, kde účelová funkce bude reprezentována jako minimalizace nákladů poskytovatele a množina řešení bude omezena technickými parametry elektráren a podmínkou zajištění optimality v síti, kterou nám reprezentují nerovnice (6.3) a (6.4). Tento model si zdefinujeme v kapitole 7.

MILP model je schopný najít optimální výkonové rozložení na daném časovém intervalu při známých požadavcích na diagramový výkon a regulační zálohy. Model tak je schopen fungovat v offline režimu, kdy jsou všechny požadavky známy. Při online provozu se nám vyskytuje problém z hlediska charakteru požadavků na regulační zálohy. Zatímco známe požadavky na diagramový výkon, tak požadavky na aFRR a mFRR vnášejí do našeho problému neurčitost a v rámci návrhu online algoritmu se s tím budeme muset vypořádat. Pro online algoritmus budeme muset zvolit vhodnou reprezentaci požadavků v čase.

V našem případě budeme požadavky reprezentovat pomocí časových řad. Naš online algoritmus tak bude vždy hledat optimální rozložení výkonových požadavků pro aktuální časový okamžik. V rámci jednoho kroku algoritmu budeme hledat optimální rozložení pro daný úsek této časové řady, kde budeme znát minulé hodnoty a pro ně patřičná rozdělení a naším cílem bude nalézt optimální rozdělení výkonových požadavků pro aktuální časový okamžik. K tomu ovšem budeme muset predikovat časový vývoj této časové řady a pro tento účel budeme potřebovat vhodný prediktor. Návrh online algoritmu si představíme v kapitole 8. Na návrh prediktorů se pak zaměříme v kapitolách 9 a 10.

Kapitola 7

MILP model soustavy

V předchozí kapitole jsme formalizovali optimalizační problém. V této kapitole se zaměříme na jeho offline variantu. Offline varianta bude představovat situaci, kdy máme zadaný časový interval Ω^T a jsou nám známé požadavky na diagramový výkon P_t^{DG} a podpůrné služby $aFRR_t^a$ a $mFRR_t^a$ kde $t \in \Omega^T$. Offline model následně nalezne optimální rozložení výkonů na jednotkách v rámci agregačního bloku, tak aby těmto požadavkům vyhověl. Optimalizační problém formulujeme jako úlohu celočíselného lineárního programování a offline model tak bude MILP model [31] reprezentující chování agregačního bloku na Ω^T .

V první části této kapitoly uděláme rychlý úvod do celočíselného lineárního programování a následně si zavedeme MILP model soustavy, kde jeho kritériální funkcí budou celkové náklady na výrobu v rámci Ω^T a jeho omezující podmínky budou představovat technické limity jednotek a podmínky výkonové rovnováhy definované v předchozích kapitolách.

7.1 Úvod do ILP

Celočíselné lineární programování je odvětví matematické optimalizace, kde jako proměnné uvažujeme celá čísla a kritériální funkce spolu s omezujícími podmínkami jsou lineární. Oproti klasickému se jedná o NP-úplný problém [32]. Na jeho vyřešení tedy neexistuje algoritmus s polynomiální složitostí.

Standardní forma ILP úlohy je zadána jako

$$\max \mathbf{c}^T \cdot \mathbf{x}, \quad (7.1)$$

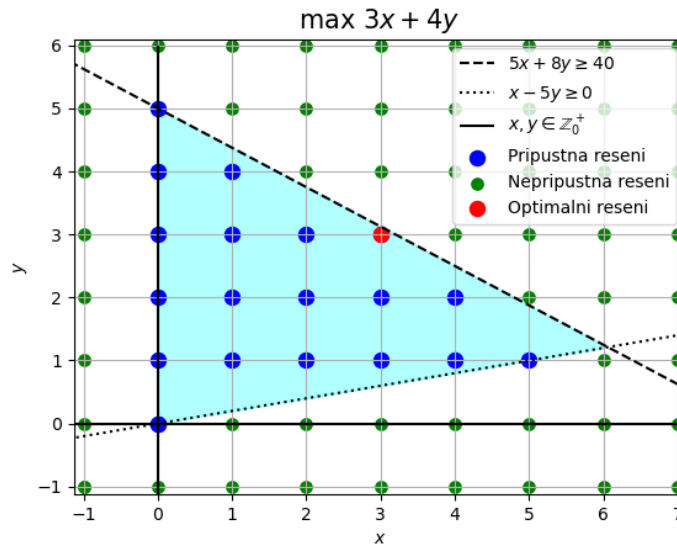
za podmínek

$$\mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}, \quad (7.2)$$

$$\mathbf{x} \geq 0, \quad (7.3)$$

$$\mathbf{x} \in \mathbb{Z}^n. \quad (7.4)$$

Vektor \mathbf{x} zde představuje vektor proměnných. Pokud bychom uvažovali místo (7.4) podmínku $\mathbf{x} \in \mathbb{R}^n$, jednalo by se o úlohu lineárního programování, která



Obrázek 7.1: Ukázka ILP problému

má polynomiální komplexitu.

Vzhledem k požadavku na celočíselnost řešení se nám rozpadne souvislost přípustné množiny (znázorněno na obrázku 7.1.), na které hledáme přípustné řešení a úloha se tak stává NP obtížná. Úloha celočíselného lineárního programování je již dlouho známý problém a pro její vyřešení dnes existují optimalizované solvery, které jsou tuto úlohu schopny řešit velice rychle. V rámci této práce budeme používat solver od firmy Gurobi [33].

Celočíselné lineární programování dnes představuje velice silný nástroj v matematické optimalizaci. Jelikož se jedná o dobře zvládnutý kombinatorický problém, tak se využívá pro řešení řady kombinatorických úloh tak, že se úloha převede na ILP úlohu. Následně se pomocí optimalizovaného solveru nalezne řešení. ILP tak dokáže efektivně vyřešit například problém batohu nebo problém obchodního cestujícího. V této práci jsme ILP zvolili z důvodu nelinearity, které nám do úlohy přináší technické limity jednotlivých bloků a diskrétní povaha požadavků diagramového výkonu P_t^{DG} a podpůrných služeb $aFRR_t^a$ a $mFRR_t^a$.

7.2 Požadavky na model

Cílem našeho modelu bude minimalizovat náklady C na výrobu elektrické energie na straně provozovatele. Model bude pracovat s daným časovým intervalem Ω^T , na kterém nám budou známy požadavky na Diagramový výkon P_t^{DG} a požadavky na aktivaci služeb $aFRR_t^a$ a $mFRR_t^a$. Bude se tedy jednat o offline optimalizaci, ale model následně využijeme i při tvorbě online algoritmu. Offline model bude rozdělovat výkonové požadavky uvnitř agregačního bloku a to pro jednotlivé časové okamžiky $t \in \Omega^T$ uvnitř daného

Název	Typ	Popis proměnné
$P_{i,t}$	Reálné číslo	Požadavek na výkon výrobního bloku $i \in \Omega^{ER}$ v čase $t \in \Omega^T$
$x_{i,t}$	Binární hodnota	Informace je-li výrobní blok $i \in \Omega^{ER}$ v čase $t \in \Omega^T$ aktivní
$y_{i,t}^a$	Binární hodnota	Čas $t \in \Omega^T$ kdy je výrobní blok $i \in \Omega^{ER}$ aktivován
$y_{i,t}^d$	Binární hodnota	Čas $t \in \Omega^T$ kdy je výrobní blok $i \in \Omega^{ER}$ deaktivován
$SOC_{i,t}$	Reálné číslo	Stav nabití akumulátoru $i \in \Omega^{ESS}$ v čase $t \in \Omega^T$
$P_{i,t}^c$	Reálné číslo	Nabíjecí výkon akumulátoru $i \in \Omega^{ESS}$ v čase $t \in \Omega^T$
$P_{i,t}^d$	Reálné číslo	Vybíjecí výkon akumulátoru $i \in \Omega^{ESS}$ v čase $t \in \Omega^T$
$c_{i,t}$	Binární hodnota	Informuje je-li akumulátor $i \in \Omega^{ESS}$ v čase $t \in \Omega^T$ nabíjen
$d_{i,t}$	Binární hodnota	Informuje je-li akumulátor $i \in \Omega^{ESS}$ v čase $t \in \Omega^T$ vybíjen
P_t^g	Reálné číslo	Výkon odebíraný ze sítě v čase $t \in \Omega^T$

Tabulka 7.1: Proměnné MILP modelu

intervalu. Časový okamžik představuje minimální periodu se kterou se mohou měnit vstupní požadavky. V našem případě se bude jednat o periodu 1 minuta.

Jelikož se jedná o NP obtížnou úlohu, klademe si za cíl pokrýt co nejdelší časový úsek pro daný počet jednotek zahrnutých do agregačního bloku. Model nebude uvažovat obnovitelné zdroje, které nelze přímo ovládat. V rámci jednotek zahrnutých do agregačního bloku tak budeme uvažovat výrobní jednotky (tepelné elektrárny a vodní elektrárny) a akumulátory (přečerpávací vodní elektrárny a bateriové stanice). Jejich systémový popis jednotek nám bude určovat omezující podmínky modelu.

7.3 Proměnné a parametry modelu

V rámci našeho modelu agregačního bloku budeme uvažovat množinu Ω^{ER} zahrnující všechny jednotky agregačního bloku, množinu Ω^{ESS} zahrnující všechny akumulátory výrobního bloku a diskrétní množinu časových okamžiků Ω^T , reprezentující časový interval.

V tabulce 7.1 jsou uvedeny proměnné se kterými bude náš MILP model pracovat. Jedná se proměnné, které mohou nabývat reálných hodnot a také celočíselné proměnné, které nám zde slouží jako řídicí proměnné, udávající informaci o stavu zdrojů. Tabulka 7.2 uvádí seznam parametrů, které bude náš model zahrnovat.

7.4 Omezující podmínky modelu

Technické limity

Jelikož hlavní proměnné, které nás budou zajímat jsou požadavky na výkony v rámci výrobních jednotek agregačního bloku. Pro každou výrobní jednotku

Název	Popis parametru
\bar{P}_i	Maximální výkon výrobní jednotky $i \in \Omega^{ER}$
\underline{P}_i	Minimální výkon výrobní jednotky $i \in \Omega^{ER}$
T_i	Maximální výkonový trend výrobní jednotky $i \in \Omega^{ER}$
\overline{SOC}_i	Maximální stav nabití akumulátoru $i \in \Omega^{ESS}$
\underline{SOC}_i	Minimální stav nabití akumulátoru $i \in \Omega^{ESS}$
\bar{P}_i^c	Maximální nabíjecí výkon akumulátoru $i \in \Omega^{ESS}$
\bar{P}_i^d	Maximální vybíjecí výkon akumulátoru $i \in \Omega^{ESS}$
\underline{P}_i^c	Minimální nabíjecí výkon akumulátoru $i \in \Omega^{ESS}$
\underline{P}_i^d	Minimální vybíjecí výkon akumulátoru $i \in \Omega^{ESS}$
T_i^c	Maximální nabíjecí výkonový trend akumulátoru $i \in \Omega^{ESS}$
T_i^d	Maximální vybíjecí výkonový trend akumulátoru $i \in \Omega^{ESS}$
η_i	Účinnost nabíjení (resp. vybíjení) akumulátoru $i \in \Omega^{ESS}$
P_t^{DG}	Požadovaný diagramový výkon v čase $t \in \Omega^T$
P_t^{aFRR+}	Horní mez pro průběh aktivace aFRR $t \in \Omega^T$
P_t^{aFRR-}	Dolní mez pro průběh aktivace aFRR $t \in \Omega^T$
P_t^{mFRR+}	Horní mez pro průběh aktivace mFRR $t \in \Omega^T$
P_t^{mFRR-}	Dolní mez pro průběh aktivace mFRR $t \in \Omega^T$
C_i^a	Cena aktivace výrobního bloku $i \in \Omega^{ER}$
C_i^d	Cena deaktivace výrobního bloku $i \in \Omega^{ER}$
a_i, b_i	Koeficienty nákladové křivky výrobního bloku $i \in \Omega^{ER}$
a_i^c, b_i^c	Koeficienty nákladové křivky nabíjení akumulátoru $i \in \Omega^{ESS}$
a_i^d, b_i^d	Koeficienty nákladové křivky vybíjení akumulátoru $i \in \Omega^{ESS}$
C_t^g	Cena jednotkového výkonu odebraného ze sítě v čase $t \in \Omega^T$
ϵ	Povolená výkonová odchylka

Tabulka 7.2: Parametry MILP modelu

platí následující vztah

$$x_{i,t} \cdot \underline{P}_i \leq P_{i,t} \leq x_{i,t} \cdot \bar{P}_i \quad \forall i \in \Omega^{ER} \quad \forall t \in \Omega^T. \quad (7.5)$$

Tato rovnice nám udává rozmezí výkonu bloku $i \in \Omega^{ER}$ pro časový okamžik $t \in \Omega^T$. Je-li blok v aktivním provozu ($x_{i,t} = 1$), pak se jeho hodnota může pohybovat v intervalu $\langle \underline{P}_i, \bar{P}_i \rangle$. Je-li blok odstaven ($x_{i,t} = 0$), pak je jeho hodnota rovna 0.

Dalším požadavkem na výkon je vztah

$$-x_{i,t} \cdot T_i \leq P_{i,t} - P_{i,(t-1)} \leq x_{i,t} \cdot T_i \quad \forall i \in \Omega^{ER} \quad \forall t \in \Omega^T. \quad (7.6)$$

Tato podmínka nám omezuje rychlost změny výkonu pro danou výrobní jednotku. Vidíme že maximální změna výkonu mezi časovými okamžiky t a $(t-1)$ je podmíněna maximálním zatěžovacím trendem bloku $i \in \Omega^{ER}$.

Následující vztah nám udává hodnoty pro časové okamžiky, kdy výrobní blok aktivujeme a deaktivujeme:

$$x_{i,t} - x_{i,(t-1)} \leq y_{i,t}^a \quad \forall i \in \Omega^{ER} \quad \forall t \in \Omega^T, \quad (7.7)$$

$$x_{i,(t-1)} - x_{i,t} \leq y_{i,t}^d \quad \forall i \in \Omega^{ER} \quad \forall t \in \Omega^T. \quad (7.8)$$

Nyní se zaměříme na akumulátory. Každý akumulátor je v čase $t \in \Omega^T$ stav nabit na určitou hodnotu, kterou udává stav nabití a pro jeho rozmezí platí

$$\underline{SOC}_i \leq SOC_{i,t} \leq \overline{SOC}_i \quad \forall i \in \Omega^{ESS} \quad \forall t \in \Omega^T. \quad (7.9)$$

Rovnice nám tedy udává interval hodnot energie kterou jsme schopni v akumulátoru i v čase $t \in \Omega^T$ uložit. Pro stav nabití dále platí následující rovnice

$$SOC_{i,t} = SOC_{i,(t-1)} + \eta_i \cdot P_{i,t}^c - \frac{P_{i,t}^d}{\eta_i} \quad \forall i \in \Omega^{ESS} \quad \forall t \in \Omega^T. \quad (7.10)$$

Tato rovnice nám udává jak se bude měnit stav nabití akumulátoru $i \in \Omega^{ESS}$ mezi časovými okamžiky t a $(t-1)$. Pro nabíjení a vybíjení rovněž platí následující vztahy

$$c_{i,t} \cdot \underline{P}_i^c \leq P_{i,t}^c \leq c_{i,t} \cdot \overline{P}_i^c \quad \forall i \in \Omega^{ESS} \quad \forall t \in \Omega^T, \quad (7.11)$$

$$d_{i,t} \cdot \underline{P}_i^d \leq P_{i,t}^d \leq d_{i,t} \cdot \overline{P}_i^d \quad \forall i \in \Omega^{ESS} \quad \forall t \in \Omega^T, \quad (7.12)$$

$$0 \leq P_{i,t}^c - P_{i,(t-1)}^c \leq c_{i,t} \cdot T_i^c \quad \forall i \in \Omega^{ESS} \quad \forall t \in \Omega^T, \quad (7.13)$$

$$0 \leq P_{i,t}^d - P_{i,(t-1)}^d \leq d_{i,t} \cdot T_i^d \quad \forall i \in \Omega^{ESS} \quad \forall t \in \Omega^T, \quad (7.14)$$

$$c_{i,t} + d_{i,t} \leq 1 \quad \forall i \in \Omega^{ESS} \quad \forall t \in \Omega^T. \quad (7.15)$$

Rovnice (7.11) a (7.12) nám udávají technické limity pro nabíjení a vybíjení akumulátoru. Rovnice (7.13) a (7.14) nám říkají, jakou maximální změnu výkonu lze při nabíjení (resp. vybíjení) realizovat. (Pozn.: Rovnice (7.13) a (7.14) je vhodné uvažovat pouze v případě, že akumulátor představuje přečerpávací vodní elektrárna, pro baterii jsou trendy obvykle velmi vysoké). Rovnice (7.15) nám udává důležitý vztah a to že akumulátor lze v jednom časovém okamžiku pouze nabíjet nebo vybíjet, ale obě varianty jsou najednou nepřípustné.

■ Podmínky výkonové rovnováhy

Nyní se zaměříme na podmínky pro splnění výkonové rovnováhy. Pro přehlednější značení si zavedeme následující pomocné proměnné

$$P_t^{EV} = \sum_{i \in \Omega^{ER}} P_{i,t} \quad \forall t \in \Omega^T, \quad (7.16)$$

$$P_t^C = \sum_{i \in \Omega^{ESS}} \eta_i \cdot P_{i,t}^c \quad \forall t \in \Omega^T, \quad (7.17)$$

$$P_t^D = \sum_{i \in \Omega^{ESS}} \frac{P_{i,t}^d}{\eta_i} \quad \forall t \in \Omega^T, \quad (7.18)$$

Rovnice (7.16) nám udává celkové množství vyrobeného výkonu všemi výrobními bloky v čase $t \in \Omega^T$. Rovnice (7.17) uvádí celkový výkon, který se využívá k nabíjení akumulátorů v čase $t \in \Omega^T$. Rovnice (7.18) udává celkový výkon, který vzniká při vybíjení akumulátorů.

Model musí splňovat podmínky výkonové rovnováhy, které jsme schopni zapsat pomocí následujících rovnic

$$P_t^{EV} + P_t^D - P_t^g - P_t^C \leq P_t^{DG} + P_t^{aFRR+} + P_t^{mFRR+} + \epsilon, \quad (7.19)$$

$$P_t^{EV} + P_t^D - P_t^g - P_t^C \geq P_t^{DG} + P_t^{aFRR-} + P_t^{mFRR-} - \epsilon, \quad (7.20)$$

$$P_t^g \geq 0 \quad (7.21)$$

Rovnice (7.19) a (7.20) představují požadavek na splnění výkonové rovnováhy vzhledem k aktivovaným požadavkům na služby aFRR a mFRR. Rovnice (7.17) představuje záruku dosažení požadovaných rezerv při aktuálním diagramovém výkonu.

7.5 Kriteriaální funkce modelu

Kriteriaální funkce představuje náklady poskytovatele. Před zavedením kriteriaální funkce si pro přehlednější značení zavedeme několik pomocných proměnných. První proměnná nám bude představovat cenu provozu výrobní jednotky. Vztah pro jednu výrobní jednotku udává následující rovnice:

$$C_i^{ER} = \sum_{t \in \Omega^T} y_{i,t}^a C_i^a + \sum_{t \in \Omega^T} y_{i,t}^d C_i^d + \sum_{t \in \Omega^T} (a_i \cdot x_{i,t} + b_i \cdot P_{i,t}). \quad (7.22)$$

První člen této rovnice udává celkovou cenu aktivací výrobního bloku v daném časovém intervalu, druhý člen je cena deaktivací a poslední člen, představuje cenu aktivního provozu. Celková cena je potom součet všech cen jednotlivých výrobních bloků tedy

$$C^{ER} = \sum_{i \in \Omega^{ER}} C_i^{ER}. \quad (7.23)$$

Dále si potřebuje zavést celkovou cenu za akumulátoru pro kterou platí vztah

$$C_i^{ESS} = \sum_{t \in \Omega^T} (a_i^d \cdot d_{i,t} + b_i^d \cdot P_{i,t}^d) + \sum_{t \in \Omega^T} (a_i^c \cdot c_{i,t} + b_i^c \cdot P_{i,t}^c). \quad (7.24)$$

První člen této rovnice představuje celkovou cenu akumulátoru i nabíjení v časovém intervalu a druhý člen představuje cenu vybíjení. V poslední řadě potřebujeme celkovou cenu nakoupené energie, kterou odebíráme z rozvodné sítě pro kterou platí následující vztah

$$C^G = \sum_{t \in \Omega^T} C_t^g \cdot P_t^g. \quad (7.25)$$

Naše kriteriaální funkce pak bude ve tvaru

$$\min \sum_{i \in \Omega^{ER}} C_i^{ER} + \sum_{i \in \Omega^{ESS}} C_i^{ESS} + C^G \quad (7.26)$$

Cílem je tedy minimalizovat celkové náklady agregačního bloku.

Kapitola 8

Online řešení

V minulé kapitole jsme si zavedli MILP model, který nalezne optimální rozložení výkonu na soustavě jednotek poskytovatele. Tento model je offline řešením našeho problému, jelikož je schopen nalézt optimální řešení až při známých požadavcích na Diagramový výkon a služby mFRR s plánovanou aktivací a aFRR. Model nalezne řešení na zadaném časovém intervalu, kde teoreticky nemáme omezení na jeho délku (pozn.: jediné omezení nám zde přináší výpočetní náročnost, kterou se budeme zabývat v předposlední kapitole). Pro reálný provoz ovšem budeme potřebovat způsob jak nalézt optimální řešení v reálném čase pro zadaný časový okamžik. Budeme tedy potřebovat online model, který bude schopen najít přípustné řešení v reálném čase. V této kapitole si zavedeme online model, který bude pracovat v reálném čase, využijeme při tom offline modelu, definovaného v předchozí kapitole.

8.1 Rolling window

Náš online model bude využívat metody posuvného okna (Rolling window) [34]. Uvažujme množinu Ω^T , která nám představuje všechny časové okamžiky pro které hledáme rozložení výkonů. Ω^T je tedy časový horizont, na kterém pracuje offline model a pro který v jednom kroku nalezne optimální rozložení výkonů. Offline řešení zná výkonové požadavky P_t^{DG} , $aFRR_t^a$ a $mFRR_t^a$ pro $\forall t \in \Omega^T$. Nyní si rozeberme povahu jednotlivých požadavků. P_t^{DG} je nám známé dopředu, jelikož se jedná o smluvené dodávky diagramového výkonu na jednotlivé obchodní intervaly. Jedná se tak o deterministický signál. $mFRR_t^a$ jsou požadavky na mFRR s plánovanou aktivací, které se mohou měnit pouze 7.5 minuty před začátkem následujícího obchodního intervalu. Zároveň platí

$$mFRR_t^a \in \langle \overline{mFRR}^-, \overline{mFRR}^+ \rangle \quad \forall t \in \Omega^T. \quad (8.1)$$

Kde \overline{mFRR}^+ (resp. \overline{mFRR}^-) je maximum kladné (resp. záporné) regulační energie, která je pro daný obchodní interval poskytována. V rámci optimalizace tak víme, že tato hodnota se může měnit pouze v daných časových okamžicích, představující polovinu obchodního intervalu. $aFRR_t^a$ jsou požadavky na službu aFRR. Podobně jako u mFRR s plánovanou aktivací platí

$$aFRR_t^a \in \langle \overline{aFRR}^-, \overline{aFRR}^+ \rangle \quad \forall t \in \Omega^T. \quad (8.2)$$

Kde \overline{aFRR}^+ (resp. \overline{aFRR}^-) je maximum kladné (resp. záporné) regulační energie, která je pro daný obchodní interval poskytována. Na rozdíl od mFRR s plánovanou aktivací se tento požadavek může měnit kdykoliv. Tento signál je tedy stochastické povahy.

Vzhledem ke stochastické povaze aFRR a mFRR tak neznáme hodnoty všech výkonových požadavků dopředu. Známe pouze hodnoty požadavků v aktuálním časovém okamžiku t_a a hodnoty v minulosti $t < t_a$. V každém kroku online algoritmu je potřeba nalézt výkonová rozdělení pro časový okamžik t_a . Pro nalezení tohoto rozdělení využijeme Rolling window. Tato metoda bude využívat smíšeného celočíselného lineárního programování stejně jak offline řešení, ovšem pouze na časové množině $\Omega^{Tw} \subset \Omega^T$, která obsahuje $2m + 1$ prvků. Označme

$$\Omega^{TP} = \{t_a - m, t_a - m + 1, \dots, t_a - 2, t_a - 1\}, \quad (8.3)$$

$$\Omega^{TA} = \{t_a\}, \quad (8.4)$$

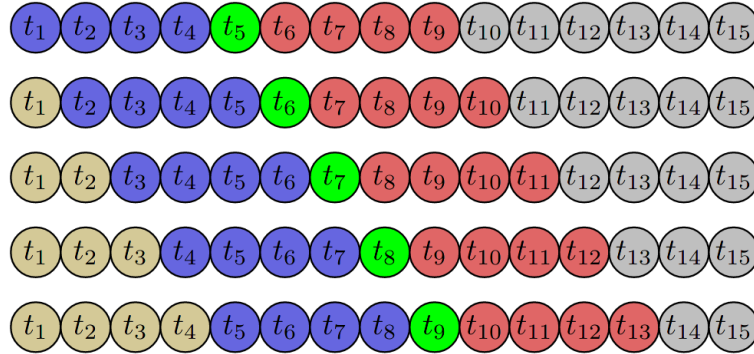
$$\Omega^{TF} = \{t_a + 1, t_a + 2, \dots, t_a + m\}, \quad (8.5)$$

potom

$$\Omega^{Tw} = \Omega^{TP} \cup \Omega^{TA} \cup \Omega^{TF}. \quad (8.6)$$

Kde Ω^{TP} představuje množinu minulých časových okamžiků $t < t_a$, Ω^{TA} představuje aktuální časový okamžik t_a a Ω^{TF} je množina následujících časových okamžiků $t > t_a$. Ω^{Tw} nám tak představuje časové okno, jehož středem je t_a .

Pro $\forall t \in \Omega^{TP}$ známe výkonové požadavky i rozdělení výkonů na jednotky agregačního bloku, jelikož jsme je již stanovili v minulých krocích algoritmu. Pro t_a známe výkonové požadavky a hledáme rozdělení výkonů. Pro $\forall t \in \Omega^{TF}$ výkonové požadavky neznáme, ovšem pro správné fungování algoritmu je budeme muset predikovat (Pozn.: v následujících 2 kapitolách si podrobně rozebereme prediktory). Výkonové požadavky pro $\forall t \in \Omega^{TP}$ nám tak představují počáteční podmínky modelu. MILP model najde optimální rozdělení na celé množině Ω^{Tw} , ovšem výstupem pro nás je pouze výkonové rozdělení pro t_a . Princip metody posuvného okna je na obrázku 8.1. Řádky reprezentují



Obrázek 8.1: Zobrazení principu metody posuvného okna

Název	Popis parametru
P_{i,t_p}^*	Výkon výrobní jednotky $i \in \Omega^{ER}$ v čase $t_p \in \Omega^{TP}$
x_{i,t_p}^*	Výrobní jednotka $i \in \Omega^{ER}$ je v čase $t_p \in \Omega^{TP}$ aktivní
y_{i,t_p}^{a*}	Výrobní jednotka $i \in \Omega^{ER}$ je v čase $t_p \in \Omega^{TP}$ aktivována
y_{i,t_p}^{d*}	Výrobní jednotka $i \in \Omega^{ER}$ je v čase $t_p \in \Omega^{TP}$ deaktivována
SOC_{i,t_p}^*	Stav nabití akumulátoru $i \in \Omega^{ESS}$ v čase $t_p \in \Omega^{TP}$
P_{i,t_p}^{c*}	Nabíjecí výkon akumulátoru $i \in \Omega^{ESS}$ v čase $t_p \in \Omega^{TP}$
P_{i,t_p}^{d*}	Vybíjecí výkon akumulátoru $i \in \Omega^{ESS}$ v čase $t_p \in \Omega^{TP}$
c_{i,t_p}^*	Akumulátor $i \in \Omega^{ESS}$ je v čase $t_p \in \Omega^{TP}$ nabíjen
d_{i,t_p}^*	Akumulátor $i \in \Omega^{ESS}$ je v čase $t_p \in \Omega^{TP}$ vybíjen
$P_{t_p}^{g*}$	Výkon odebíraný ze sítě v čase $t_p \in \Omega^{TP}$

Tabulka 8.1: Přidané parametry do modelu

kroky algoritmu. Zeleně jsou znázorněny t_a pro jednotlivé kroky algoritmu Modře jsou znázorněny $t \in \Omega^{TP}$ pro jednotlivé kroky algoritmu a Červeně jsou znázorněny $t \in \Omega^{TF}$ pro jednotlivé kroky algoritmu. Šedou barvou jsou potom znázorněny časové okamžiky, se kterými algoritmus v daném kroku nepracuje a pro které neznáme rozložení výkonů a žlutě jsou znázorněny časové okamžiky, se kterými algoritmus pracoval v minulých krocích a pro které již máme rozložení výkonů.

8.2 Úprava MILP modelu

Pro správné fungování Rolling window algoritmu budeme muset upravit MILP model definovaný v předchozí kapitole. Model musí být schopen pracovat s počátečními podmínkami, které představují již nalezená rozdělení výkonů a také musí být schopen pracovat i s nepřesností predikovaných požadavků $aFRR_t^a$ a $mFRR_t^a$ kde $t \in \Omega^{TF}$. Zároveň jsme limitováni rozsahem časového okna, což znamená, že budeme muset upravit model tak aby vhodně pracoval s energií uloženou v akumulátorech.

8.2.1 Počáteční podmínky

Přidání počátečních podmínek do modelu jsme schopni realizovat pomocí přidání parametrů do našeho modelu. Přidané parametry jsou uvedeny v tabulce 8.1. Vidíme, že parametry jsou hodnoty jednotlivých proměnných pro časové okamžiky $t \in \Omega^{TP}$. Do modelu tak musíme přidat omezující podmínky, které zajistí splnění počátečních podmínek

$$P_{i,t_p} = P_{i,t_p}^* \quad \forall i \in \Omega^{ER} \quad \forall t_p \in \Omega^{TP}, \quad (8.7)$$

$$x_{i,t_p} = x_{i,t_p}^* \quad \forall i \in \Omega^{ER} \quad \forall t_p \in \Omega^{TP}, \quad (8.8)$$

$$y_{i,t_p}^a = y_{i,t_p}^{a*} \quad \forall i \in \Omega^{ER} \quad \forall t_p \in \Omega^{TP}, \quad (8.9)$$

$$y_{i,t_p}^d = y_{i,t_p}^{d*} \quad \forall i \in \Omega^{ER} \quad \forall t_p \in \Omega^{TP}, \quad (8.10)$$

$$SOC_{i,t_p} = SOC_{i,t_p}^* \quad \forall i \in \Omega^{ESS} \quad \forall t_p \in \Omega^{TP},, \quad (8.11)$$

$$P_{i,t_p}^c = P_{i,t_p}^{c*} \quad \forall i \in \Omega^{ESS} \quad \forall t_p \in \Omega^{TP},, \quad (8.12)$$

$$P_{i,t_p}^d = P_{i,t_p}^{d*} \quad \forall i \in \Omega^{ESS} \quad \forall t_p \in \Omega^{TP},, \quad (8.13)$$

$$c_{i,t_p} = c_{i,t_p}^* \quad \forall i \in \Omega^{ESS} \quad \forall t_p \in \Omega^{TP},, \quad (8.14)$$

$$d_{i,t_p} = d_{i,t_p}^* \quad \forall i \in \Omega^{ESS} \quad \forall t_p \in \Omega^{TP},, \quad (8.15)$$

$$P_{t_p}^g = P_{t_p}^{g*} \quad \forall t_p \in \Omega^{TP},. \quad (8.16)$$

Proměnné s časovými indexy t_p nemusíme dále uvažovat v účelové funkci, jelikož se z pohledu optimalizace nyní již jedná o konstanty.

8.2.2 Limitní požadavky

Nepřesnost predikovaných požadavků $aFRR_t^a$ a $mFRR_t^a$ může způsobit, že online model nebude v některém z kroků výpočtu schopen najít přípustné řešení, jelikož predikce požadavků v předchozím kroku se příliš liší od skutečné hodnoty požadavku v aktuálním kroku. Pro tento případ je důležité si definovat nejhorší možný případ průběhu požadavků na služby aFRR a mFRR s plánovanou aktivací.

Nejhorší situace nastane pokud v následujícím kroku dojde k požadavku na plnou aktivaci regulačních záloh. Pojmem plné aktivace máme na mysli požadavek na maximální kladnou regulační energii (resp. maximální zápornou regulační energii). Tento požadavek způsobí velký výkonový trend, kterého musí být agregační blok schopen dosáhnout a proto je potřeba v aktuálním časovém okamžiku volit takové rozložení, abychom byli schopni tuto situaci splnit. Do modelu si tak zavedeme pomocné proměnné, které nám budou reprezentovat výkonová rozložení pro případ plné aktivace kladné energie (resp. záporné energie). Seznam proměnných pro případ plné aktivace kladné regulační energie je uveden v tabulce 8.2. a tabulce 8.3 je uveden seznam

Název	Typ	Popis proměnné
$\hat{P}_{i,t}$	Reálné číslo	Požadavek na výkon výrobního bloku $i \in \Omega^{ER}$ v čase $t \in \Omega^{TF}$
$S\hat{O}C_{i,t}$	Reálné číslo	Stav nabití akumulátoru $i \in \Omega^{ESS}$ v čase $t \in \Omega^{TF}$
$\hat{x}_{i,t}$	Binární hodnota	Informace je-li výrobní blok $i \in \Omega^{ER}$ v čase $t \in \Omega^{TF}$ aktivní
$\hat{y}_{i,t}^a$	Binární hodnota	Čas $t \in \Omega^{TF}$ kdy je výrobní blok $i \in \Omega^{ER}$ aktivován
$\hat{y}_{i,t}^d$	Binární hodnota	Čas $t \in \Omega^{TF}$ kdy je výrobní blok $i \in \Omega^{ER}$ deaktivován
$\hat{P}_{i,t}^c$	Reálné číslo	Nabíjecí výkon akumulátoru $i \in \Omega^{ESS}$ v čase $t \in \Omega^{TF}$
$\hat{P}_{i,t}^d$	Reálné číslo	Vybíjecí výkon akumulátoru $i \in \Omega^{ESS}$ v čase $t \in \Omega^{TF}$
$\hat{c}_{i,t}$	Binární hodnota	Informuje je-li akumulátor $i \in \Omega^{ESS}$ v čase $t \in \Omega^{TF}$ nabíjen
$\hat{d}_{i,t}$	Binární hodnota	Informuje je-li akumulátor $i \in \Omega^{ESS}$ v čase $t \in \Omega^{TF}$ vybíjen
\hat{P}_t^g	Reálné číslo	Výkon odebíraný ze sítě v čase $t \in \Omega^{TF}$

Tabulka 8.2: Seznam proměnných v případě plné aktivace kladné regulační energie

Název	Typ	Popis proměnné
$\tilde{P}_{i,t}$	Reálné číslo	Požadavek na výkon výrobního bloku $i \in \Omega^{ER}$ v čase $t \in \Omega^{TF}$
$S\tilde{O}C_{i,t}$	Reálné číslo	Stav nabití akumulátoru $i \in \Omega^{ESS}$ v čase $t \in \Omega^{TF}$
$\tilde{x}_{i,t}$	Binární hodnota	Informace je-li výrobní blok $i \in \Omega^{ER}$ v čase $t \in \Omega^{TF}$ aktivní
$\tilde{y}_{i,t}^a$	Binární hodnota	Čas $t \in \Omega^{TF}$ kdy je výrobní blok $i \in \Omega^{ER}$ aktivován
$\tilde{y}_{i,t}^d$	Binární hodnota	Čas $t \in \Omega^{TF}$ kdy je výrobní blok $i \in \Omega^{ER}$ deaktivován
$\tilde{P}_{i,t}^c$	Reálné číslo	Nabíjecí výkon akumulátoru $i \in \Omega^{ESS}$ v čase $t \in \Omega^{TF}$
$\tilde{P}_{i,t}^d$	Reálné číslo	Vybíjecí výkon akumulátoru $i \in \Omega^{ESS}$ v čase $t \in \Omega^{TF}$
$\tilde{c}_{i,t}$	Binární hodnota	Informuje je-li akumulátor $i \in \Omega^{ESS}$ v čase $t \in \Omega^{TF}$ nabíjen
$\tilde{d}_{i,t}$	Binární hodnota	Informuje je-li akumulátor $i \in \Omega^{ESS}$ v čase $t \in \Omega^{TF}$ vybíjen
\tilde{P}_t^g	Reálné číslo	Výkon odebíraný ze sítě v čase $t \in \Omega^{TF}$

Tabulka 8.3: Seznam proměnných v případě plné aktivace záporné regulační energie

Název	Popis parametru
\hat{P}_t^{aFRR+}	Horní mez pro průběh aktivace aFRR v případě plné aktivace kladné regulační energie
\hat{P}_t^{aFRR-}	Dolní mez pro průběh aktivace aFRR v případě plné aktivace kladné regulační energie
\tilde{P}_t^{aFRR+}	Horní mez pro průběh aktivace aFRR v případě plné aktivace záporné regulační energie
\tilde{P}_t^{aFRR-}	Dolní mez pro průběh aktivace aFRR v případě plné aktivace záporné regulační energie
\hat{P}_t^{mFRR+}	Horní mez pro průběh aktivace mFRR v případě plné aktivace kladné regulační energie
\hat{P}_t^{mFRR-}	Dolní mez pro průběh aktivace mFRR v případě plné aktivace kladné regulační energie
\tilde{P}_t^{mFRR+}	Horní mez pro průběh aktivace mFRR v případě plné aktivace záporné regulační energie
\tilde{P}_t^{mFRR-}	Dolní mez pro průběh aktivace mFRR v případě plné aktivace záporné regulační energie

Tabulka 8.4: Horní a dolní omezení v čase $t \in \Omega^{TF}$ v případě plné aktivace kladné (resp. záporné) regulační energie

proměnných pro případ záporné regulační energie.

V rámci plné aktivace kladné (resp. záporné) regulační energie si ještě zavedeme pomocné parametry do modelu, které nám budou reprezentovat průběh limitních požadavků. Tyto parametry jsou uvedeny v tabulce 8.4. Do modelu nám pak přibudou omezující podmínky, které nám zajistí splnitelnost plné aktivace. V čase t_a tak budeme uvažovat následující omezení

$$-x_{i,(t_a+1)} \cdot T_i \leq P_{i,(t_a+1)} - P_{i,t_a} \leq -x_{i,(t_a+1)} \cdot T_i \quad \forall i \in \Omega^{ER}, \quad (8.17)$$

$$-\hat{x}_{i,(t_a+1)} \cdot T_i \leq \hat{P}_{i,t_a+1} - P_{i,t_a} \leq -\hat{x}_{i,(t_a+1)} \cdot T_i \quad \forall i \in \Omega^{ER}, \quad (8.18)$$

$$-\tilde{x}_{i,(t_a+1)} \cdot T_i \leq \tilde{P}_{i,(t_a+1)} - P_{i,t_a} \leq -\tilde{x}_{i,(t_a+1)} \cdot T_i \quad \forall i \in \Omega^{ER}, \quad (8.19)$$

$$-c_{i,(t_a+1)} \cdot T_i^c \leq P_{i,(t_a+1)}^c - P_{i,t_a}^c \leq -c_{i,(t_a+1)} \cdot T_i^c \quad \forall i \in \Omega^{ESS}, \quad (8.20)$$

$$-\hat{c}_{i,(t_a+1)} \cdot T_i^c \leq \hat{P}_{i,(t_a+1)}^c - P_{i,t_a}^c \leq -\hat{c}_{i,(t_a+1)} \cdot T_i^c \quad \forall i \in \Omega^{ESS}, \quad (8.21)$$

$$-\tilde{c}_{i,(t_a+1)} \cdot T_i^c \leq \tilde{P}_{i,(t_a+1)}^c - P_{i,t_a}^c \leq -\tilde{c}_{i,(t_a+1)} \cdot T_i^c \quad \forall i \in \Omega^{ESS}, \quad (8.22)$$

$$-d_{i,(t_a+1)} \cdot T_i^d \leq P_{i,(t_a+1)}^c - P_{i,t_a}^c \leq -d_{i,(t_a+1)} \cdot T_i^d \quad \forall i \in \Omega^{ESS}, \quad (8.23)$$

$$-\hat{d}_{i,(t_a+1)} \cdot \mathbf{T}_i^d \leq \hat{P}_{i,(t_a+1)}^c - P_{i,t_a}^c \leq -\hat{d}_{i,(t_a+1)} \cdot \mathbf{T}_i^d \quad \forall i \in \Omega^{ESS}, \quad (8.24)$$

$$-\tilde{d}_{i,(t_a+1)} \cdot \mathbf{T}_i^d \leq \tilde{P}_{i,(t_a+1)}^c - P_{i,t_a}^c \leq -\tilde{d}_{i,(t_a+1)} \cdot \mathbf{T}_i^d \quad \forall i \in \Omega^{ESS}, \quad (8.25)$$

$$SOC_{i,(t_a+1)} = SOC_{i,t_a} + \eta_i P_{i,(t_a+1)}^c - \frac{P_{i,(t_a+1)}^d}{\eta_i} \quad \forall i \in \Omega^{ESS}, \quad (8.26)$$

$$S\hat{O}C_{i,(t_a+1)} = SOC_{i,t_a} + \eta_i \hat{P}_{i,(t_a+1)}^c - \frac{\hat{P}_{i,(t_a+1)}^d}{\eta_i} \quad \forall i \in \Omega^{ESS}, \quad (8.27)$$

$$S\tilde{O}C_{i,(t_a+1)} = SOC_{i,t_a} + \eta_i \tilde{P}_{i,(t_a+1)}^c - \frac{\tilde{P}_{i,(t_a+1)}^d}{\eta_i} \quad \forall i \in \Omega^{ESS}. \quad (8.28)$$

Zároveň musí pro limitní pùběhy platit podmínky výkonové rovnováhy. Označme

$$\hat{P}_t^{EV} = \sum_{i \in \Omega^{ER}} \hat{P}_{i,t} \quad \forall t \in \Omega^{TF}, \quad (8.29)$$

$$\hat{P}_t^C = \sum_{i \in \Omega^{ESS}} \eta_i \cdot \hat{P}_{i,t}^c \quad \forall t \in \Omega^{TF}, \quad (8.30)$$

$$\hat{P}_t^D = \sum_{i \in \Omega^{ESS}} \frac{\hat{P}_{i,t}^d}{\eta_i} \quad \forall t \in \Omega^{TF}, \quad (8.31)$$

$$\tilde{P}_t^{EV} = \sum_{i \in \Omega^{ER}} \tilde{P}_{i,t} \quad \forall t \in \Omega^{TF}, \quad (8.32)$$

$$\tilde{P}_t^C = \sum_{i \in \Omega^{ESS}} \eta_i \cdot \tilde{P}_{i,t}^c \quad \forall t \in \Omega^{TF}, \quad (8.33)$$

$$\tilde{P}_t^D = \sum_{i \in \Omega^{ESS}} \frac{\tilde{P}_{i,t}^d}{\eta_i} \quad \forall t \in \Omega^{TF}, \quad (8.34)$$

potom pro zajištění výkonové rovnováhy musí platit

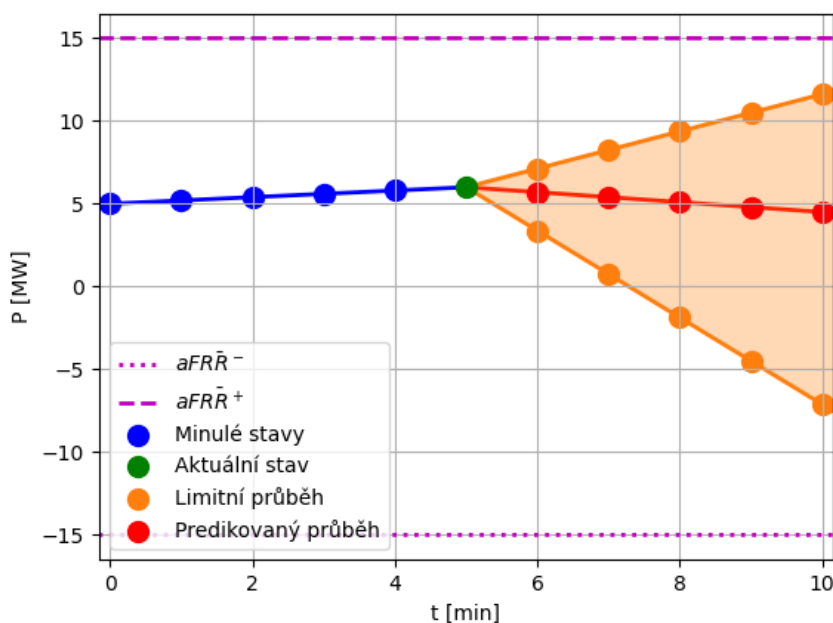
$$\hat{P}_t^{EV} + \hat{P}_t^D - \hat{P}_t^C - \hat{P}_t^g \leq P_t^{DG} + \hat{P}_t^{aFRR+} + \hat{P}_t^{mFRR+} + \epsilon \quad \forall t \in \Omega^{TF}, \quad (8.35)$$

$$\hat{P}_t^{EV} + \hat{P}_t^D - \hat{P}_t^C - \hat{P}_t^g \geq P_t^{DG} + \hat{P}_t^{aFRR-} + \hat{P}_t^{mFRR-} - \epsilon \quad \forall t \in \Omega^{TF}, \quad (8.36)$$

$$\tilde{P}_t^{EV} + \tilde{P}_t^D - \tilde{P}_t^C - \tilde{P}_t^g \leq P_t^{DG} + \tilde{P}_t^{aFRR+} + \tilde{P}_t^{mFRR+} + \epsilon \quad \forall t \in \Omega^{TF}, \quad (8.37)$$

$$\tilde{P}_t^{EV} + \tilde{P}_t^D - \tilde{P}_t^C - \tilde{P}_t^g \geq P_t^{DG} + \tilde{P}_t^{aFRR-} + \tilde{P}_t^{mFRR-} - \epsilon \quad \forall t \in \Omega^{TF}. \quad (8.38)$$

Tyto rovnice nám zajistí takové rozložení výkonù v čase t_a , aby bylo schopno splnit i krajní případy představující požadavky na plnou aktivaci kladné (resp. záporné) regulační energie. Pro proměnné, realizující případ plné aktivace kladné (resp. záporné) regulační energie platí technické limity definované rovnicemi (7.5) až (7.15). Tyto průběhy nám reprezentují omezující podmínky a jejich výkonová rozložení nevstupují do kritériální funkce. Optimalizace probíhá pouze pro rozložení na predikovaném průběhu.



Obrázek 8.2: Ukázka predikovaného a limitního průběhu

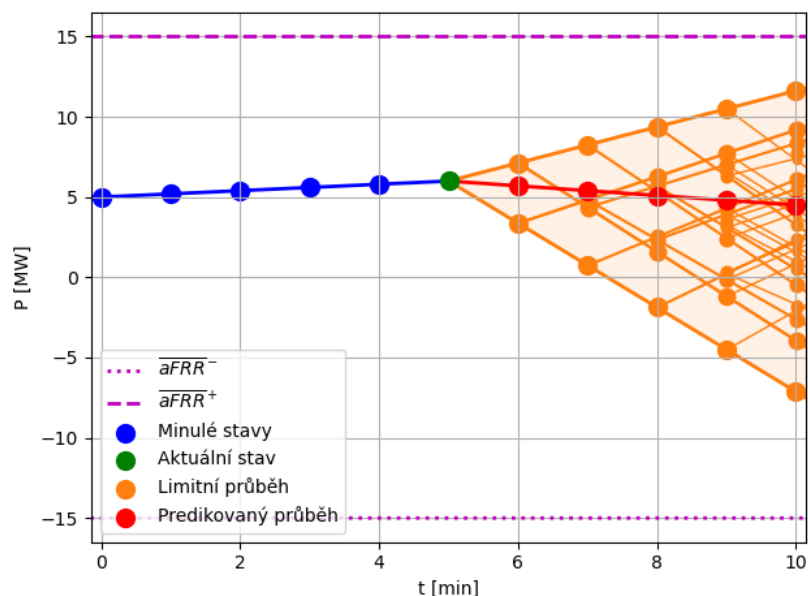
Ukázka průběhu predikovaných hodnot a hodnot v limitním případě plné aktivace je na obrázku 8.2. Modře jsou zobrazeny stavy pro které již bylo nalezeno výkonové rozložení. Zeleně je zobrazen aktuální stav t_a pro který hledáme výkonové rozložení. Toto je ovšem ovlivněno predikovaným průběhem (červená) a limitním průběhem (oranžová). Limitní průběh nám tak vymezuje oblast, pro kterou dokážeme v případě nepřesnosti predikce nalézt v následujícím kroku přípustné řešení.

Optimální rozložení tak v každém kroku uvažujeme pouze pro predikovaný průběh. Pomocí přidání podmínek garantující splnění limitních požadavků jsme si definovali heuristiku, která nám sice nezajistí přímou robustnost, ale vede na uspokojivé výsledky. Jak se, ale v dalších kapitolách popisující návrh prediktorů dozvíme, tak jsme schopni predikovat průběh požadavků velmi přesně.

■ Úplné rozšíření

Výše uvedená heuristika, lze vylepšit. Toto vylepšení spočívá v úplném rozšíření, kdy v každém kroku limitního průběhu budeme uvažovat plnou aktivaci opačné polaroty. Toto vylepšení nám způsobí rozvětvení limitních průběhů asymptoticky odpovídající $\mathcal{O}(2^k)$, kde k odpovídá počtu predikovaných vzorků. Pro tato větvičí se řešení platí obdobné vztahy jako (8.17) až (8.38) a je potřeba doplnit odpovídající seznam proměnných a parametrů.

Z důvodu úspory místa a ušetření výpočetního výkonu uděláme úmluvu, že se tímto úplným rozšířením zabývat nebudeme. V kapitole 11 zaměřující se na testování modelu si ukážeme že výše uvedená heuristika nám pro naše



Obrázek 8.3: Ukázka predikovaného a plně rozšířeného limitního průběhu

účely postačí. Toto plné rozšíření tak bude představovat námět pro další práci. Ukázka průběhu predikovaných hodnot a hodnot v limitním případě plné aktivity pro úplné rozšíření je na obrázku 8.3.

8.2.3 Úprava kriteriální funkce

Jelikož jsme v rámci online algoritmu limitováni šířkou posuvného okna, tak nejsme schopni vidět celý časový horizont. MILP model tak může mít tendenci využívat energii uloženou v akumulátorech z důvodu levnějších nákladů. Akumulátory mají obvykle vyšší trend než výrobní jednotky, ale jejich schopnost vyrábět výkon je omezena jejich kapacitou a proto se nevyplatí je využívat jako zdroje pro splnění dodávek diagramového výkonu.

Rovněž se může mnohdy vyplatit využít k zajištění dodávek diagramového výkonu, jen menší počet zdrojů pracujících na vyšších výkonech. Z hlediska minimalizace nákladů pro aktuální časový okamžik t_a je to tak správně, ale tato optimalizace nás může dostat do problémů, kdy nebudeme mít k dispozici zdroje, tak aby byly schopny splnit požadavky při vyšším trendu. Jako příklad si můžeme uvést situaci, kdy došlo k vybití akumulátorů a tím pádem není k dispozici rychlý zdroj, aby pokryl požadavek s vysokým trendem. Rovněž v případě výrobních jednotek může být problém, pokud je vyžadován vysoký kladný trend a na aktuálních dodávkách se podílelo jen omezené množství výrobních jednotek, které pracovaly na plný výkon. V tomto případě nezvládnou zbylé jednotky splnit požadavky na vysoký trend.

Abychom byli schopni splnit i požadavky s vysokým trendem, je výhodné

udržovat výkony výrobních jednotek v polovině intervalu $\langle \underline{P}, \bar{P} \rangle$. Stejně tak je dobré udržovat akumulátory na poloviční kapacitě, která odpovídá polovině intervalu $\langle \underline{SOC}, \overline{SOC} \rangle$. Pro udržení výkonů výrobních jednotek a kapacit akumulátorů na požadovaných hodnotách budeme muset upravit kriteriální funkci MILP modelu, aby penalizovala odchylky od těchto hodnot.

Označme

$$P_i^\square = \underline{P}_i + \frac{\bar{P}_i - \underline{P}_i}{2}, \quad (8.39)$$

$$SOC_i^\square = \underline{SOC}_i + \frac{\overline{SOC}_i - \underline{SOC}_i}{2}. \quad (8.40)$$

potom si pro přehlednější značení zavedeme pomocné proměnné

$$\Delta P_{i,t} = |P_{i,t} - P_i^\square|, \quad (8.41)$$

$$\Delta SOC_{i,t} = |SOC_{i,t} - SOC_i^\square|. \quad (8.42)$$

V rovnicích (8.41) a (8.42) se ale vyskytují absolutní hodnoty. Z hlediska MILP modelu toto nelze přímo vyjádřit, jelikož MILP model pracuje pouze s lineárními funkcemi a lineárními omezujícími podmínkami. Tyto proměnné tak budeme muset vyjádřit pomocí následujících nerovností

$$\Delta P_{i,t} \geq P_{i,t} - P_i^\square, \quad (8.43)$$

$$\Delta P_{i,t} \geq P_i^\square - P_{i,t}, \quad (8.44)$$

$$\Delta SOC_{i,t} \geq SOC_{i,t} - SOC_i^\square, \quad (8.45)$$

$$\Delta SOC_{i,t} \geq SOC_i^\square - SOC_{i,t}. \quad (8.46)$$

Pro přehlednější značení si ještě zavedeme pomocné proměnné

$$\Delta P = \sum_{t \in \Omega^{Tw}} \sum_{i \in \Omega^{ER}} \Delta P_{i,t}, \quad (8.47)$$

$$\Delta SOC = \sum_{t \in \Omega^{Tw}} \sum_{i \in \Omega^{ESS}} \Delta SOC_{i,t}. \quad (8.48)$$

Výsledná kriteriální funkce našeho online modelu pak bude ve tvaru

$$\min \sum_{i \in \Omega^{ER}} C_i^{ER} + \sum_{i \in \Omega^{ESS}} C_i^{ESS} + C^G + \mu^P \cdot \Delta P + \mu^{SOC} \Delta SOC. \quad (8.49)$$

μ^P a μ^{SOC} představují multiplikatívni adaptivní koeficienty. Jelikož naším cílem je kriteriální funkci minimalizovat, bude se solver snažit minimalizovat jednotlivé hodnoty $\Delta P_{i,t} \forall i \in \Omega^{ER} \forall t \in \Omega^T$ a $\Delta SOC_{i,t} \forall i \in \Omega^{ESS} \forall t \in \Omega^T$ za podmínek definovaných nerovnicemi (8.43) až (8.46).

Online model je tak multikriteriální úloha celočíselného lineárního programování. Snaha o udržení minimální odchylky stavu nabití akumulátorů od poloviny kapacity a výkonu výrobní jednotky od poloviny zadávacího

intervalu představují pomocná kritéria optimalizace. Online řešení se tak snaží minimalizovat náklady, ale bere v úvahu i pomocná kritéria. Koefficienty μ^P a μ^{SOC} nám udávají váhu pomocných kritérií. Volba těchto koeficientů je zásadní pro správné fungování online algoritmu. V případě malých hodnot μ^P a μ^{SOC} nebudou mít pomocná kritéria v kriteriální funkci téměř žádnou váhu. Naopak pokud volíme příliš velké hodnoty μ^P a μ^{SOC} , tak se model snaží docílit minimální odchylky a původní cíl minimalizace bude mít menší vliv na optimalizaci. V našem případě tyto hodnoty experimentálně. Jejich volba se bude odvíjet od sestavy agregačního bloku jak uvidíme v kapitole 11. Toto pomocné kritérium nám obecně nezajistí robustnost řešení, ale v našem řešení představuje heuristiku, která vede k uspokojivým výsledkům.

Na základě této heuristiky a heuristiky definované v sekci 8.2.2. by tak měl online algoritmus být schopen efektivně řídit jednotky agregačního bloku aby vyhověl vstupním požadavkům P_t^{DG} , $aFRR_t^a$ a $mFRR_t^a$ kde $t \in \Omega^T$. V následujících dvou kapitolách se zaměříme na návrhy prediktorů a následně provedeme experimenty, kde budeme tento algoritmus s jednotlivými prediktory testovat.

Kapitola 9

Predikce pomocí Statistického Modelu

V minulé kapitole jsme zavedli online řešení, využívající Rolling Window. V rámci Rolling Window jsme si uvedli, že pro daný čas t_a je potřeba najít optimální rozložení výkonových požadavků v jednotky agregacího bloku, kde pro časové okamžiky $t > t_a$ neznáme přesnou hodnotu všech požadavků. Požadavky P_t^{DG} nám jsou známy dopředu a jedná se tak o deterministický signál. Požadavky $mFRR_t^a$ se mohou měnit jen v polovině každého obchodního intervalu a uvážíme-li (8.1) a princip zajištění heuristiky popsané v části 8.2.2 a 8.2.3, tak pro tento signál prediktor potřebovat nebudeme a časovém okamžiku, kde může dojít ke změně hodnoty signálu budeme počítat s nejhorším scénářem, tedy plnou aktivací, který nám zajistí heuristika popsaná v části 8.2.2.

Poslední signál $aFRR_t^a$ má stochastickou povahu a jeho hodnoty je třeba predikovat. Pro predikování těchto hodnot navrheme 2 prediktory. Prvním z nich je prediktor pomocí neuronové sítě, který si rozebereme v následující kapitole. V této kapitole se zaměříme na poněkud starší metodu. Tato metoda využívá statistického modelu ARIMA. Jedná se o metodu využívanou ve statistice, ekonomii a k analýze časových řad. V této kapitole si postupně uvedeme metody identifikace modelu, estimaci parametrů modelu a konstrukci prediktoru. V závěru kapitoly si uvedeme výsledky prediktoru na testovacích datech a zhodnotíme jeho přesnost.

9.1 Úvod do ARIMA modelu

Model ARIMA je zobecněním modelu ARMA, který se využívá k modelování stacionárních stochastických procesů. Na rozdíl od modelu ARMA lze model ARIMA využít i k modelování nestacionárních procesů. Označení modelu vychází z jeho anglického názvu **A**uto-**R**egresive **I**ntegrated **M**oving **A**verage. Model se tak skládá ze tří částí. Autoregresní (AR) část nám udává, že část aktuální hodnoty časové řady lze vyjádřit jako lineární kombinaci p předchozích hodnot. Hodnota p pak udává řád AR části. Integrační část nám představuje diferenci časové řady. Řád integrační části je pak stupeň difference d . Poslední částí je klouzavý průměr (MA), která určuje, že část rezidua časové řady, lze vyjádřit jako lineární kombinaci q předchozích reziduí. Spojením všech složek

dohromady dostáváme proces ARIMA řádu (p, d, q) .

V případě $d = 0$ se jedná o proces ARMA řádu (p, q) . Tento proces lze zapsat pomocí následující rovnice

$$x_t = \phi_1 x_{(t-1)} + \dots + \phi_p x_{(t-p)} + \epsilon_t - \theta_1 \epsilon_{(t-1)} - \dots - \theta_q \epsilon_{(t-q)}. \quad (9.1)$$

Kde $\{x_i\}_{i=t}^{t-p}$ jsou hodnoty časové řady a $\{\epsilon_i\}_{i=t}^{t-q}$ jsou hodnoty reziduí, která lze modelovat pomocí bílého šumu. Hodnoty $\{\phi_i\}_{i=0}^p$ a $\{\theta_i\}_{i=0}^q$ jsou pak multiplikativní koeficienty ARMA modelu splňující $\phi_i \in \mathbb{R} \forall i \in \{1, \dots, p\}$ a $\theta_i \in \mathbb{R} \forall i \in \{1, \dots, q\}$. Tento vztah lze pomocí operátoru posunutí z vyjádřit jako

$$(1 - \phi_1 z - \phi_1 z^2 - \dots - \phi_p z^p) x_t = (1 - \theta_1 z - \theta_1 z^2 - \dots - \theta_q z^q) \epsilon_t, \quad (9.2)$$

což lze zapsat jako

$$\phi(z) x_t = \theta(z) \epsilon_t, \quad (9.3)$$

kde $\phi(z)$ je polynom řádu p a $\theta(z)$ je polynom řádu q . ARMA model soustavy lze též reprezentovat jako IIR filtr s přenosovou funkcí

$$H(z) = \frac{\theta(z)}{\phi(z)}, \quad (9.4)$$

a výslednou časovou řadu tak dostaneme jako přefiltrování bílého šumu, který je vstupem do soustavy. Hodnoty $\{\phi_i\}_{i=0}^p$ a $\{\theta_i\}_{i=0}^q$ jsou pak koeficienty tohoto filtru. Zde Následné zobecnění ARMA na ARIMA představuje přidání diference pro časovou řadu x_t , pro ARIMA řádu (p, d, q) platí

$$\phi(z) \nabla^d x_t = \theta(z) \epsilon_t, \quad (9.5)$$

nebo dosazením do (9.1)

$$x_t = \phi_1 \nabla^d x_{(t-1)} + \dots + \phi_p \nabla^d x_{(t-p)} + \epsilon_t - \theta_1 \epsilon_{(t-1)} - \dots - \theta_q \epsilon_{(t-q)}. \quad (9.6)$$

Vztah (9.6) popisující ARIMA model je velmi podobný vztahu (9.1). Zde je důležité si povšimnout, že ARIMA (p, d, q) model funguje na stejném principu jako ARMA model, ovšem nepracuje s původním signálem x_t ale s jeho diferencí řádu d . Pro hlubší pochopení této problematiky odkážeme čtenáře na [37]

9.2 Identifikace modelu

Pro sestavení modelu je potřeba model identifikovat, tedy zjistit řád modelu (p, d, q) . Jako první je potřeba identifikovat řád integrační části d . Identifikaci provedeme postupnou diferencí signálu reprezentující časovou řadu a pro každou diferencí vypočítáme energii signálu. Pro diferencí signálu platí

$$\nabla x_t = x_t - x_{(t-1)}, \quad (9.7)$$

$$\nabla^2 x_t = \nabla x_t - \nabla x_{(t-1)} = x_t - 2x_{(t-1)} + x_{(t-2)}, \quad (9.8)$$

tedy obecně

$$\nabla^d x_t = \sum_{k=0}^d \binom{d}{k} (-1)^{d-k} x_{t-k}. \quad (9.9)$$

Řád d pak bude odpovídat stupni difference, pro který vykazoval diferencovaný signál nejmenší energii.

$$d = \arg \min_{\hat{d}} \sum_{t=-\infty}^{\infty} \|\nabla^{\hat{d}} x_t\|. \quad (9.10)$$

Následně provedeme identifikaci ARMA modelu, kterou provedeme na diferencovaném signálu. Optimální řád modelu (p, q) lze určit pomocí penaltových kritérií. My využijeme Akaikovo informační kritérium $AIC(p, q)$ [35] dané vztahem

$$AIC(p, q) = N \ln \frac{\sigma_w^2}{N - p - q} + 2(p + q), \quad (9.11)$$

kde N je počet vzorků a σ_w^2 je rozptyl chyby estimace. Toto kritérium má sice tendenci nadhodnocovat řád, ale v našem případě kdy $N \gg (p + q)$ jeho použití vyhovuje.

Zde je dobré poznamenat, že identifikace ARMA model řádu (p, q) se v praxi velmi často dělá alternativní metodou [36]. Tato metoda vychází z autokorelační a parciální-autokorelační funkce. Pro stanovení řádu d se využívá parciální-autokorelační funkce a podle jejích nenulových koeficientů stanovíme řád d . Řád q se stanovuje podle nenulových koeficientů autokorelační funkce. Tato metoda vychází z Akaikeho informačního kritéria a čtenáře pro tento případ odkážeme na [37].

9.3 Estimace parametrů modelu

Po stanovení řádu modelu, tedy nalezení hodnot (p, d, q) . Je potřeba nalézt hodnoty koeficientů $\{\phi_i\}_{i=0}^p$ a $\{\theta_i\}_{i=0}^q$. Zde je situace stejná jako pro ARMA model. Zde si uvedeme ještě jednu poznámku. Využijeme-li poznatků [38], lze ARMA model řádu (p, q) transformovat na AR model řádu (∞) . Využijme vztahu (9.4) a zapišme jej jako

$$\frac{\theta(z)}{\phi(z)} = \frac{1}{\Phi(z)}, \quad (9.12)$$

kde koeficienty $\{\Phi_i\}_{i=0}^{\infty}$ polynomu $\Phi(z)$ jsou koeficienty hledaného modelu AR (∞) . Úpravou (9.10) dostáváme

$$\Phi(z)\theta(z) = \phi(z), \quad (9.13)$$

Pro koeficienty $\{\phi_i\}_{i=0}^p$ pak platí

$$\phi_n = \Phi_n * \theta_n. \quad (9.14)$$

Koeficienty $\{\phi_i\}_{i=0}^p$ pak vzniknou konvolucí koeficientů $\{\theta_i\}_{i=0}^p$ a $\{\Phi_i\}_{i=0}^\infty$. Pro nalezení koeficientů $\{\Phi_i\}_{i=0}^\infty$ je potřeba provést zpětnou konvoluci, pomocí které dostáváme

$$\Phi_i = \begin{cases} 1 & i = 0 \\ -\sum_{k=1}^q \theta_k \Phi_{(i-k)} + \phi(k) & 1 \leq i \leq p \\ -\sum_{k=1}^q \theta_k \Phi_{(i-k)} & i > p \end{cases} \quad (9.15)$$

V praxi pak neuvažujeme AR (∞), ale pouze AR (m) kde m je takový řád, který dobře vystihne charakter modelu ARMA (p, q). Obvykle se volí

$$m = 2(p + q) + 1. \quad (9.16)$$

Výsledný model se kterým budeme pracovat tedy nebude ARIMA (p, d, q), ale ARIMA ($m, d, 0$). Převod ARMA ARMA (p, q) na AR (m) má své opodstatnění, jelikož pro estimaci koeficientů autoregresního modelu lze využít lineární metody, zatímco v případě ARMA (p, q) je potřeba využít nelineární iterativní metody a zároveň je snadnější i výsledná realizace prediktoru.

Jak jsme si uvedli v části 9.1, tak ARMA model představuje filtr s přenosovou funkcí definovanou v (9.4). Impulsová odezva tohoto filtru je potom

$$h_t = \mathcal{Z}^{-1} \{H(z)\}. \quad (9.17)$$

Potom pro $\forall t$ platí

$$x_t + \sum_{i=1}^p \phi_i x_{(t-i)} = \sum_{i=0}^q \theta_i \epsilon_{(t-i)}. \quad (9.18)$$

Provedme nyní korelaci s časovou řadou x_t a dostaneme

$$R_t^x + \sum_{i=1}^p \phi_i R_{(t-i)}^x = \sigma_w^2 c_t, \quad (9.19)$$

kde R_t^x je autokorelační funkce signálu x_t a

$$c_t = \sum_{i=0}^q \theta_i h_{t-i}^* = \sum_{i=0}^{q-t} \theta_{(i+t)} h_i^*. \quad (9.20)$$

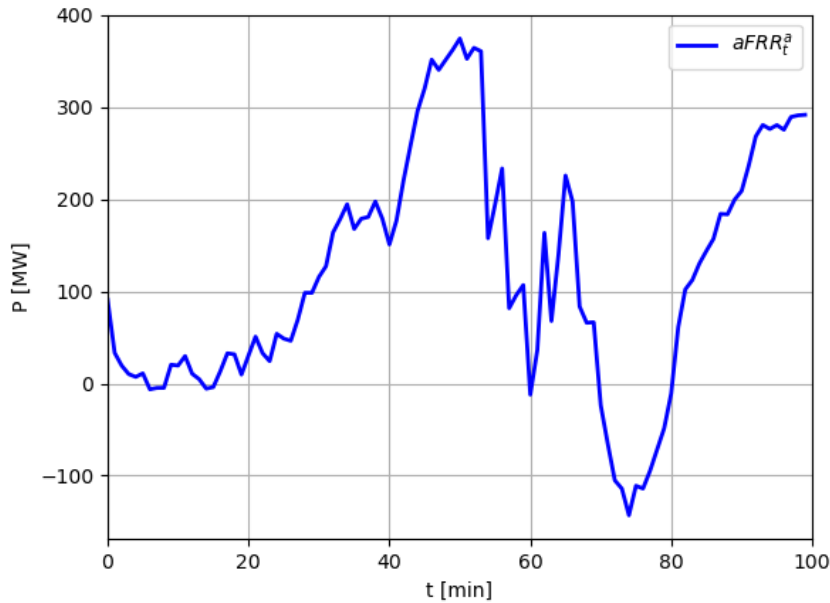
Pro AR (m) přejde (9.17) do tvaru

$$R_t^x + \sum_{i=1}^m \Phi_i R_{(t-i)}^x = \begin{cases} \sigma_w^2 \theta_0 & t = 0 \\ 0 & t > 0. \end{cases} \quad (9.21)$$

Odtud dostáváme soustavu lineárních Yule-Walkerových [39] rovnic

$$\begin{bmatrix} R_0^x & R_{(m-1)}^x & R_{(m-2)}^x & \cdots & R_1^x \\ R_1^x & R_0^x & R_{(m-1)}^x & \cdots & R_2^x \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R_{(m-1)}^x & R_{(m-2)}^x & R_{(m-3)}^x & \cdots & R_0^x \end{bmatrix} \cdot \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_m \end{bmatrix} = \begin{bmatrix} R_0^x \\ R_1^x \\ \vdots \\ R_{(m-1)}^x \end{bmatrix}, \quad (9.22)$$

jejímž vyřešením dostáváme koeficienty $\{\Phi_i\}_{i=0}^m$.



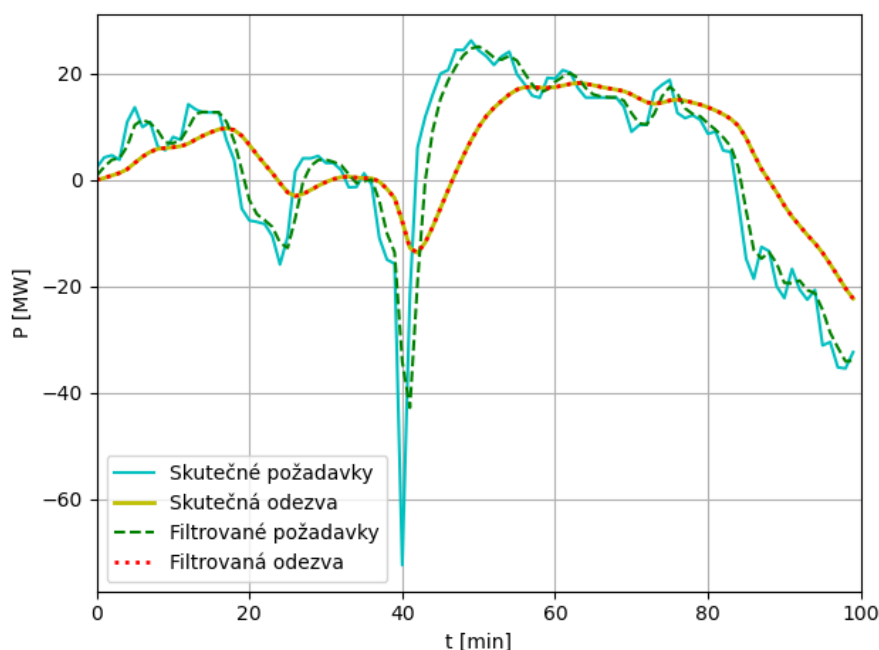
Obrázek 9.1: Požadavky aktivace služby aFRR

9.4 Návrh prediktoru

Nyní se zaměříme na výsledný ARIMA prediktor. Naším úkolem bude predikovat hodnoty požadavků $aFRR_t^a$ pro $t > t_a$. K dispozici máme data o požadavcích z let 2015 až 2021. Z hlediska signálového popisu se jedná o diskretní signál se vzorkovací periodou 1 minuta. K dispozici tak máme více než 3.5 milionu vzorků. Hodnoty jsme získali od OSC a.s. stejně jako všechna ostatní data uvedená v této diplomové práci. Jedná se o požadavky na celou soustavu ČR. Zde je důležité si uvést, že je nutné predikovat požadavky na celou elektrizační soustavu. V rámci poskytování služby aFRR existují 2 způsoby přerozdělování požadavků mezi poskytovatele.

První způsob se nazývá „Pro-rata“ [40] a využívá poměrného rozdělování požadavků mezi jednotlivé poskytovatele, podle toho kolik % představují jeho zálohy v celkových zálohách vymezených na soustavu. Druhým způsobem je systém CMOL [41] kde se využívá cenového žebříčku za poskytování regulačních záloh a dochází k postupné aktivaci, začínající od nejlevnějších poskytovatelů. Pro oba případy pak existuje jednoduchý způsob dopočítání výsledného požadavku na agregační blok.

Ukázka požadavků na aktivaci je uvedena na obrázku 9.1. Vidíme, že signál vykazuje šumové vlastnosti. Jelikož jsme v rámci požadavku povinni splnit limity definované ve (4.7) a (4.8), můžeme si dovolit tento signál filtrovat pomocí mírného IRR filtru řádu 2 typu dolní propust, čímž zlepšíme pre-



Obrázek 9.2: Původní a přefiltrované požadavky spolu s limitními požadavky na odezvu

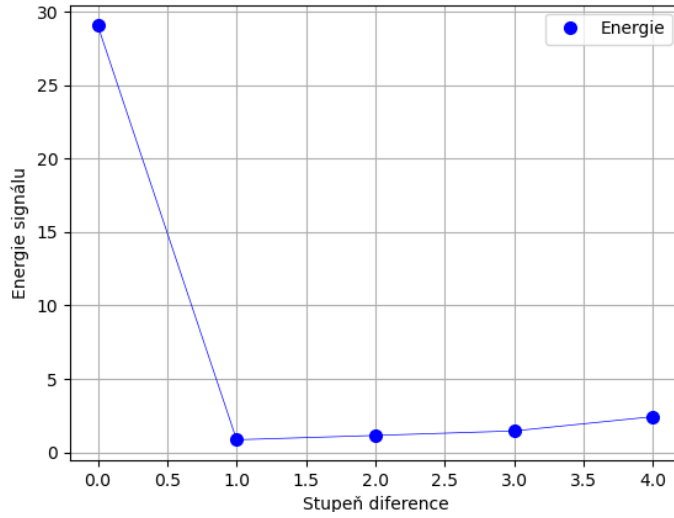
dikovatelnost signálu. Ukázka původního a přefiltrovaného signálu spolu s limitními požadavky je uvedena na obrázku 9.2. Vidíme, že filtrace je validní krok a limitní hodnota požadavků se prakticky nezmění.

Tato filtrace způsobí zvýšení korelace vzorků a proto výsledný model ARIMA model bude mít vyšší řády (p,q) . Pozn.: Pokud bychom modelovali signál reprezentující požadavky před filtrací. Dostali bychom model ARIMA $(1,1,0)$ a výsledná predikce vzorků $x_{(t+i)}$ pro $i > 1$, by tak byla velmi nepřesná, jelikož by hodnota následujícího vzorku vycházela jen z hodnoty vzorku předchozího.

V rámci identifikace si musíme zvolit vhodnou část signálu, na které provedeme identifikaci modelu. Volíme délku 1 týden, která odpovídá 10080 vzorků. Vybraný týden jsme zvolili 11.3.2019 až 17.3.2019. Ze statistických analýz vyplývá, že signál vykazuje určitou periodicitu, která odpovídá délce jednoho dne. V rámci dnů tedy má průběh aktivace aFRR velmi podobný průběh.

Na zvoleném signálu nejprve provedeme postupné diferencování a pro každý stupeň difference vypočítáme energii podle (9.10). Signál jsme diferencovali 4x (tzn. $d \leq 4$) a minimum energie jsme dosáhli pro $d = 1$, poté energie začala narůstat. Závislost energie na diferencii signálu je na obrázku 9.3.

Pomocí diferencí jsme identifikovali řád $d = 1$. Nyní zbývá identifikovat řády p a q . Zde budeme podle části 9.2 postupovat jako v případě ARMA



Obrázek 9.3: Závislost energie signálu na stupni diference

procesu. Budeme tedy pracovat s první diferencí našeho signálu a budeme potřebovat jeho parciální-autokorelační funkci pro stanovení parametru d a autokorelační funkci pro stanovení parametru q .

Dle parciální-autokorelační funkce a autokorelační funkce jsme identifikovali řády $q = 2$ a $p = 5$. Podle sekce 9.3, ale tento model převedeme na model AR řádu (m) kde řád m volíme podle (9.15) jako $m = 15$. Následně podle (9.20) dopočítáme koeficienty $\{\Phi_i\}_{i=0}^m$.

Po dopočítání koeficientů $\{\Phi_i\}_{i=0}^m$ máme hotovou identifikaci modelu a estimaci jeho parametrů. Nyní můžeme sestavit prediktor, kde využijeme (9.1) a pro predikovanou hodnotu platí

$$\hat{x}_{(t+1)} = 1 + \Phi_1(x_t - x_{(t-1)}) + \Phi_2(x_{(t-1)} - x_{(t-2)}) + \dots + \Phi_m(x_{(t-m+1)} - x_{(t-m)}) + \theta, \quad (9.23)$$

kde

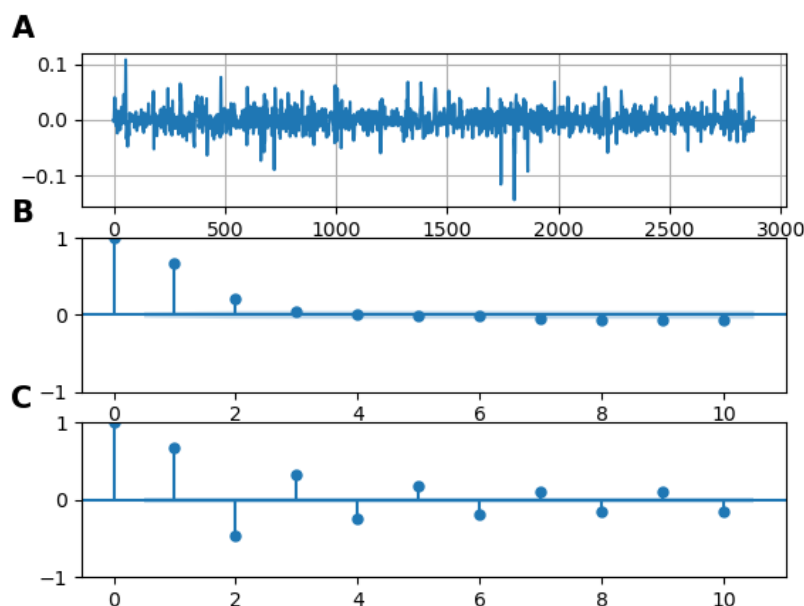
$$\theta = 1 - \sum_{i=1}^m \Phi_i. \quad (9.24)$$

Dostáváme tak predikci první hodnoty za horizontem. Pro správné fungování online algoritmu potřebujeme 4 hodnoty za horizontem, tedy $\hat{x}_{(t+2)}$, $\hat{x}_{(t+3)}$ a $\hat{x}_{(t+4)}$. Tyto hodnoty dopočítáme rekurentně z již predikovaných hodnot

$$\hat{x}_{(t+2)} = 1 + \Phi_1(\hat{x}_{(t+1)} - x_t) + \Phi_2(x_t - x_{(t-1)}) + \dots + \Phi_m(x_{(t-m+2)} - x_{(t-m+1)}) + \theta, \quad (9.25)$$

$$\hat{x}_{(t+3)} = 1 + \Phi_1(\hat{x}_{(t+2)} - \hat{x}_{(t+1)}) + \Phi_2(\hat{x}_{(t+1)} - x_t) + \dots + \Phi_m(x_{(t-m+3)} - x_{(t-m+2)}) + \theta, \quad (9.26)$$

$$\hat{x}_{(t+4)} = 1 + \Phi_1(\hat{x}_{(t+3)} - \hat{x}_{(t+2)}) + \Phi_2(\hat{x}_{(t+2)} - \hat{x}_{(t+1)}) + \dots + \Phi_m(x_{(t-m+2)} - x_{(t-m+1)}) + \theta. \quad (9.27)$$



Obrázek 9.4: Zobrazení A: diferencovaného signálu B: jeho autokorelační funkce C: jeho parciální-autokorelační funkce

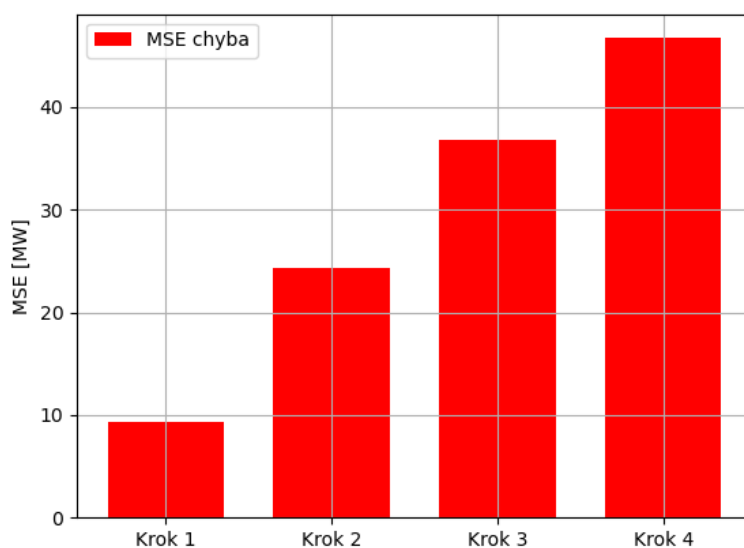
9.5 Výsledný prediktor

V minulé sekci, jsme provedli identifikaci modelu a estimaci jeho parametrů. Nyní nám zbývá otestovat jeho přesnost na testovacích datech. Pro testování jsme vybrali data odpovídající průběhu požadavků za rok 2021. Tato data nám tak vytváří testovací množinu Ω^N . Predikovali jsme pomocí navrženého ARIMA modelu 4 hodnoty za horizontem podle rovnic (9.21), (9.23), (9.24) a (9.25).

Přesnost jednotlivých kroků jsme stanovily pomocí střední kvadratické chyby jako

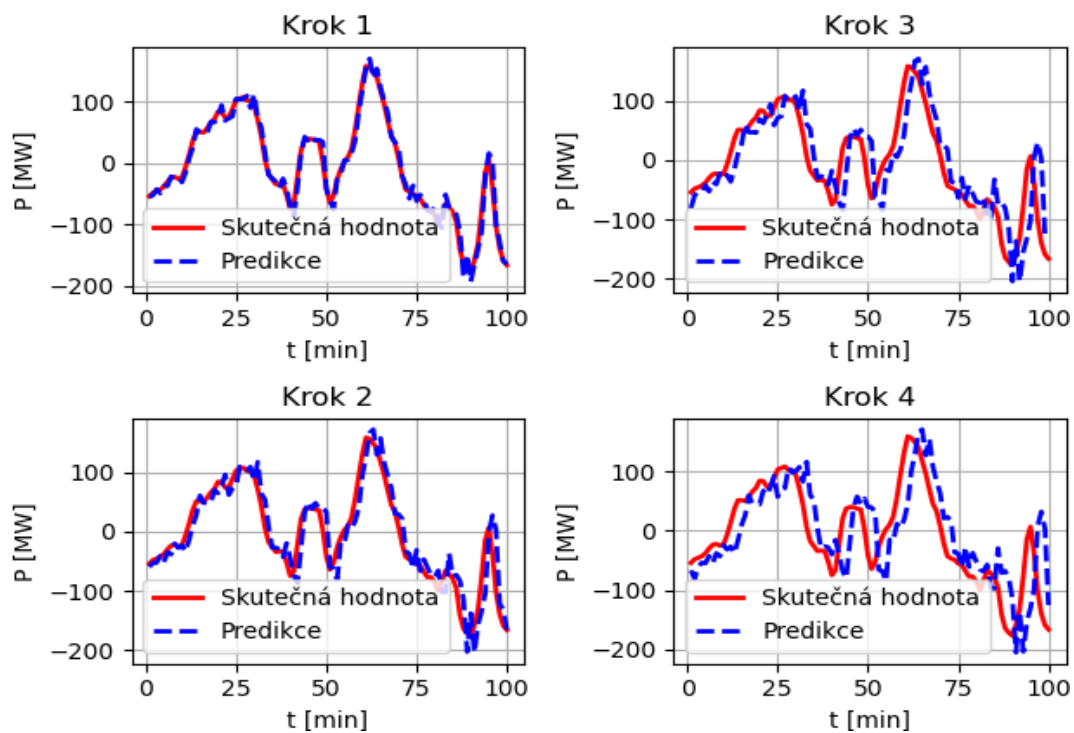
$$MSE_i = \sqrt{\sum_{k \in \Omega^N} \frac{(\hat{x}_{(k+i)} - x_{(k+i)})^2}{N}} \quad i \in \{1, 2, 3, 4\}, \quad (9.28)$$

výraz $\hat{x}_{(k+i)}$ představuje predikovanou hodnotu a $x_{(k+i)}$ je skutečná hodnota. Střední kvadratická chyba jednotlivých kroků predikce je zobrazena na obrázku 9.5. Střední kvadratická chyba čtvrtého kroku je $\pm 46.89 MW$, což je necelých 5% z celkového rozsahu aFRR $\pm 1000 MW$. Na obrázku 9.6 je zobrazeno porovnání skutečného průběhu signálu průběhu predikce pro jednotlivé kroky. Vzhledem k povaze signálu reprezentujícího požadavky na aFRR lze výsledky predikce považovat za uspokojivé. Model ARIMA představuje lineární prediktor, který nám dává algoritmus jak vytvořit predikci



Obrázek 9.5: Chyba jednotlivých kroků predikce

průběhu požadavků. Tento prediktor nám bude sloužit jako referenční prediktor hodnot. V následující kapitole sestavíme prediktor pomocí rekurentní Neuronové sítě. Dle současných poznatků vykazují neuronové sítě lepší vlastností než ARIMA model. Bohužel neuronová síť představuje černou skříňku, proto je dispozici referenční prediktor, kdyby neuronová síť selhala.



Obrázek 9.6: Porovnání predikovaného skutečného průběhu aFRR

Kapitola 10

Predikce pomocí Neuronové Sítě

V kapitole 8 jsme si představili online algoritmus a uvedli jsme, že pro jeho správné fungování musíme v každém kroku zajistit predikci průběhu služby aFRR. V minulé kapitole jsme se zabývali lineárním ARIMA modelem, pomocí kterého dokážeme predikovat hodnoty za časovým horizontem jako lineární kombinaci hodnot minulých. V této kapitole se budeme zabývat návrhem prediktoru pomocí rekurentní neuronové sítě. Nejprve si představíme základní principy hlubokého učení a následně navrheme prediktor a zhodnotíme jeho kvalitu.

10.1 Úvod do neuronových sítí

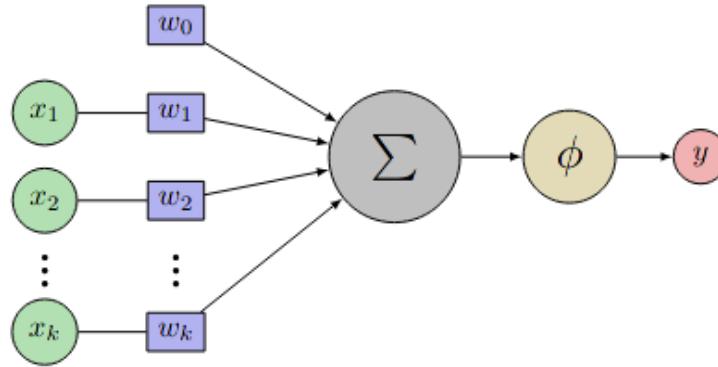
Neuronové sítě představují výpočetní nástroj umělé inteligence. Jejich vzorem jsou biologické struktury stejného názvu. V dnešní době nachází velké uplatnění v problematice rozpoznávání, zpracování obrazu, kompresi, analýze audiosignálů, k predikci vývoje časových řad a mnoha dalších aplikacích. Sítě jsou založeny na architekturách propojování výpočetních vrstev tvořených elementárními buňkami nazývanými neurony, mající několik vstupů, ale jen jeden výstup. Motivace k jejich vzniku byl problém klasifikace lineárně neseperabilních dat. První neuronové sítě byly inspirovány Perceptronovým algoritmem [42] a Logistickou regresí (viz původní název "Multi-layer perceptron").

Pozn.: Tato část slouží jako velmi rychlý úvod do problematiky. Uvedené funkce tvoří výťah z velkého množství používaných funkcí v neuronových sítích. Jsou zde uvedené jen věci nezbytné pro pochopení postupu tvorby prediktoru.

10.1.1 Matematický popis a základní princip

Neuronové sítě jsou tvořeny vzájemně propojenými umělými neurony [43]. Umělý neuron lze reprezentovat jako funkci n proměnných mající výstup y :

$$y = f(\mathbf{x}). \quad (10.1)$$



Obrázek 10.1: Schéma neuronu

Funkce $f(\mathbf{x})$ je obecně nelineární, lze ji ovšem reprezentovat následujícím způsobem

$$f(\mathbf{x}) = \phi(\mathbb{A}(\mathbf{x})). \quad (10.2)$$

Kde $\mathbb{A}(\mathbf{x})$ je afinní zobrazení a $\phi(x)$ je diferencovatelná nelineární funkce. Jelikož $\mathbb{A} : \mathbb{R}^n \rightarrow \mathbb{R}$ můžeme psát $\mathbb{A}(\mathbf{x})$ jako

$$\mathbb{A}(\mathbf{x}) = \mathbf{w}^T \cdot \mathbf{x} = \sum_{i=1}^k w_i \cdot x_i + w_0. \quad (10.3)$$

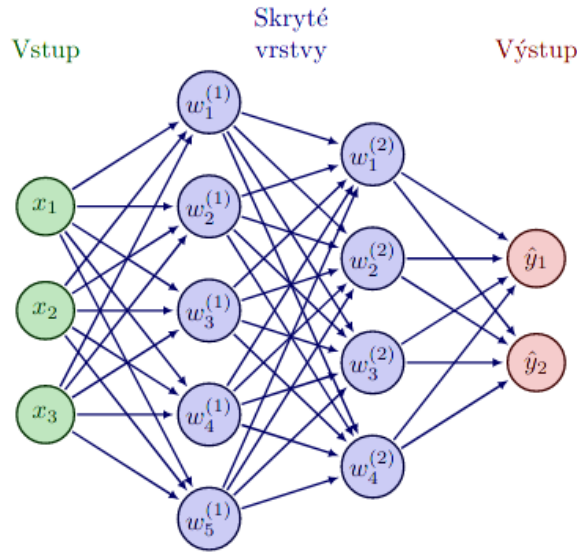
Hodnoty $w^T = \{w_0, \dots, w_k\}$, představují parametry neuronu (pozn.: někdy také parametry vrstvy). Grafické znázornění neuronu $y = f(\mathbf{x})$ je na obrázku 10.1. Z obrázku můžeme vidět podobnost mezi skutečným biologickým neuronem, který tvoří několik synapsí spojujících se dohromady. Neuronová síť je pak tvořena vrstvami neuronů, které jsou vzájemně propojené [44]. Neuronová síť přijímá vstupní vektor, který následně projde přes všechny vrstvy sítě a na poslední výstupní vrstvě dostáváme výslednou hodnotu, kterou neuronová síť vypočítala. Matematicky tak můžeme neuronovou síť s k vrstvami zapsat pomocí operátoru skládání funkcí \circ jako

$$F(\mathbf{x}) = f_k \circ f_{k-1} \circ \dots \circ f_2 \circ f_1(\mathbf{x}) = \hat{\mathbf{y}}, \quad (10.4)$$

kde $f_i(\mathbf{x}) : \mathbb{R}^{n_i} \rightarrow \mathbb{R}^{m_i}$ $i \in \{1, \dots, k\}$ jsou funkce vnitřní přenosové vrstvy sítě. Neuronová síť je tedy zobrazení $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$. Na obrázku 10.2. je zobrazeno schéma neuronové sítě se 2 skrytými vnitřními vrstvami.

Výstupem sítě je vektor $\hat{\mathbf{y}}$, který je ovlivněn parametry jednotlivých vrstev sítě. Pro správné fungování neuronové sítě je potřeba správně nastavit jednotlivé parametry. K tomuto problému se využívá gradientního algoritmu [45]. Proces nastavování parametrů sítě se nazývá učení. Při učení máme k dispozici vstupní data a k nim správné výsledky. Po vyhodnocení výstupu $\hat{\mathbf{y}}$ se data porovnají s požadovanou hodnotou \mathbf{y} pomocí chybové funkce

$$\mathcal{L} = J(\hat{\mathbf{y}}, \mathbf{y}). \quad (10.5)$$



Obrázek 10.2: Neuronová síť se 2 skrytými vnitřními vrstvami

Výraz \mathcal{L} zde představuje chybu výpočtu neuronové sítě a $J : \mathbb{R}^m \rightarrow \mathbb{R}$ je funkce pro stanovení chyby (loss funkce). Velikost této chyby pak ovlivní hodnoty parametrů jednotlivých vrstev sítě pomocí gradientní algoritmu, kde dochází ke zpětné propagaci chyby. Pro každou vrstvu $i \in \{1, 2, \dots, k\}$ platí řetězové pravidlo [46]

$$\nabla \mathbf{w}_i = \frac{\partial \mathcal{L}}{\partial \mathbf{w}_i} = \frac{\partial \mathcal{L}}{\partial \hat{\mathbf{y}}} \cdot \frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{y}_k} \cdots \frac{\partial \mathbf{y}_{i+1}}{\partial \mathbf{y}_i} \cdot \frac{\partial \mathbf{y}_i}{\partial \mathbf{w}_i}, \quad (10.6)$$

kde \mathbf{y}_i představuje výstup $f_i(\mathbf{x}_i)$ vrstvy i , \mathbf{w}_i jsou parametry vrstvy i a $\nabla \mathbf{w}_i$ je gradient dopropagovaný do vrstvy i . Uvažujme nyní množinu dat $\mathcal{T} = \{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \dots, (\mathbf{x}_N, \mathbf{y}_N)\}$, pokud provedeme výpočet všech $\hat{\mathbf{y}}_i$ a vypočítáme chybu $\mathcal{L}_j = J(\hat{\mathbf{y}}_j, \mathbf{y}_j)$ kde $j \in \{1, \dots, N\}$ pak pro správné nastavení hodnot minimalizující celkovou chybu neuronové sítě platí

$$\mathbf{w}_i^{t+1} = \mathbf{w}_i^t - \mu \sum_{j=1}^N \frac{\partial \mathcal{L}_j}{\partial \mathbf{w}_i}, \quad (10.7)$$

kde \mathbf{w}_i^t jsou parametry vrstvy i se kterými byly počítány hodnoty $\hat{\mathbf{y}}_i$ a \mathbf{w}_i^{t+1} jsou nové parametry vrstvy i . Tento proces se nazývá „stochastic gradient descent“ [47]. Jedná se o stochastický učící se proces, u kterého není zaručena konvergence do globálního minima u multi-modálních funkcí. Rychlost konvergence a chyba v ustáleném stavu je ovlivněna volbou konstanty μ , která se nazývá rychlost učení (learning rate) [48].

Toto byla základní matematická formulace principu neuronové sítě. Vidíme, že je zde velká podobnost s perceptronem což je lineární klasifikátor, který nalezne podprostor definující separaci dat do daných tříd. Zároveň vidíme

podobnost s logistickou regresí, která využívá gradientní metodu učení parametrů logistické sigmoidy pro nalezení pravděpodobnostního rozložení dat a tím pádem i jejich klasifikaci pomocí Bayesovského přístupu.

10.1.2 Aktivační funkce

V rovnici (10.2) jsme uvedli funkci $\phi(x)$. Tato funkce je obecně nelineární a jejím hlavním účelem je vnést do jinak lineárního algoritmu nelinearitu [49], která by měla způsobit potlačení (resp. navýšení) hodnot, které negativně (resp. pozitivně) ovlivňují výsledek. V rámci gradientního algoritmu se její parametry nemění a požadavek na její diferencovatelnost nám umožní zpětnou propagaci gradientu pro libovolné hodnoty.

Při prvním návrhu neuronové sítě (tehdy vícevrstvého perceptronu) se uvažovalo o funkci

$$\phi(x) = \text{sign}(x). \quad (10.8)$$

Ta ovšem nebyla spojitá v bodě $x = 0$. Pro aproximaci tvaru této funkce se tak využily funkce hyperbolický tangens [48]

$$\phi(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (10.9)$$

a logistický sigmoid [50]

$$\phi(x) = \sigma(x) = \frac{1}{1 + e^{-x}}. \quad (10.10)$$

Tyto funkce fungovaly v rámci neuronových sítí dobře a používají se běžně i v dnešních architekturách. Pro jejich gradienty platí

$$\frac{d \tanh(x)}{dx} = \frac{4}{(e^x + e^{-x})^2}, \quad (10.11)$$

$$\frac{d\sigma(x)}{dx} = \sigma(x)(1 - \sigma(x)). \quad (10.12)$$

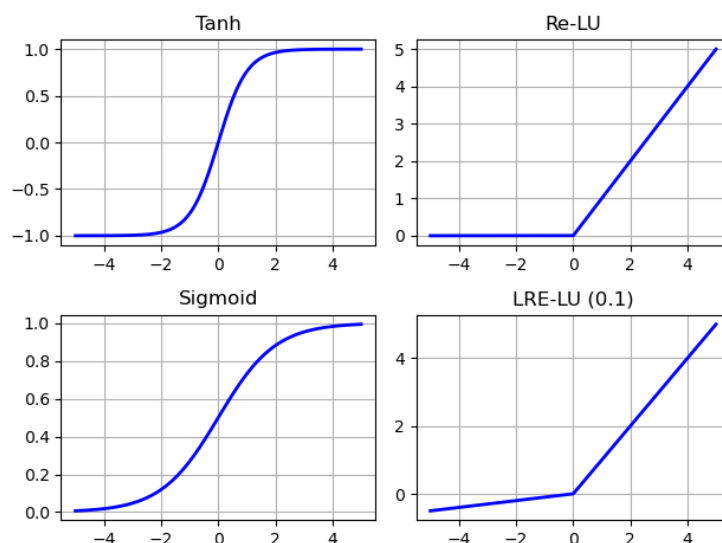
Pro tyto funkce tak nastává problém v případě zpětné propagace hodnot, které jsou v absolutní hodnotě větší než 1, jelikož gradient těchto funkcí je pro tyto hodnoty téměř nulový. Tento jev se nazývá „vanishing gradient problem“ [51]

Vzhledem k charakteru funkcí (10.9) a (10.10) vidíme, že cílem je potlačit záporné hodnoty, čímž dosáhneme požadované nelinearity a síť si v rámci učení dokáže nastavit správné hodnoty jednotlivých vrstev. Pro tento účel se zavádí nové aktivační funkce zvané Re-LU (rectified linear unit) [52] a LRe-LU (Leaky Re-LU) [53] definované následovně

$$\phi(x) = \text{ReLU}(x) = \max\{x, 0\}, \quad (10.13)$$

$$\phi(x) = \text{LReLU}(x) = \max\{x, 0\} + \min\{\alpha x, 0\} \quad \alpha \in (0, 1). \quad (10.14)$$

Tyto funkce již nevykazují vanishing gradient problem a v moderních neuronových sítích se velmi často používají. Znázornění aktivačních funkcí je na obrázku 10.3.



Obrázek 10.3: Aktivační funkce

10.1.3 Ztrátová funkce

V minulé části jsme si zavedli funkci $J(\hat{\mathbf{y}}, \mathbf{y})$, která slouží ke stanovení chyby \mathcal{L} výpočtu neuronové sítě. Zároveň jsme si nastínili gradientní algoritmus pro nastavení parametrů sítě, aby chyba byla minimální. Cílem učení je najít takové parametry neuronové sítě, abychom dosáhli co možná nejmenší chyby výstupních hodnot. Proces učení je tak možné zadefinovat jako optimalizační úlohu

$$\min_{(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{T}} J(F(\mathbf{x}_i), \mathbf{y}_i). \quad (10.15)$$

Hledáme tedy takové parametry \mathbf{w}^* splňující

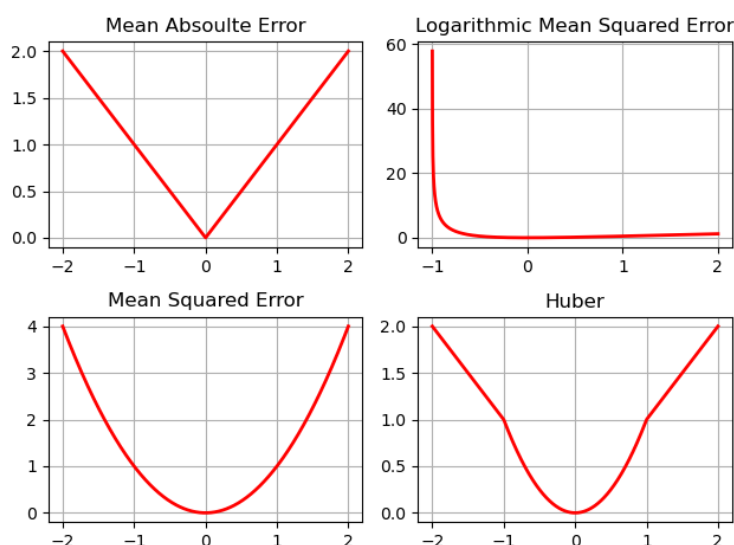
$$\mathbf{w}^* = \arg \min_w \sum_{(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{T}} J(F(\mathbf{x}_i), \mathbf{y}_i). \quad (10.16)$$

V této optimalizační úloze nám funkce $J(\hat{\mathbf{y}}, \mathbf{y})$ představuje účelovou funkci. Nalezení optimálních parametrů tak bude ovlivněno charakterem funkce. Vzhledem k náhodné povaze signálů se velmi často používá střední kvadratická chyba (Mean Squared Error)

$$J(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{N} \|\hat{\mathbf{y}} - \mathbf{y}\|^2 = \frac{1}{N} \sum_{i=0}^N (\hat{y}_i - y_i)^2. \quad (10.17)$$

Střední kvadratická chyba má však nevýhodu pro velké odchylky, jelikož velikost chyby roste s druhou mocninou. Další často používanými chybovými funkcemi jsou střední kvadratická logaritmická chyba (Mean Logarithmic Squared Error)

$$J(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{N} \sum_{i=0}^N \left(\ln \left(\frac{y_i + 1}{\hat{y}_i + 1} \right) \right)^2, \quad (10.18)$$



Obrázek 10.4: Znázornění chybových funkcí

a střední absolutní chyba (Mean Absolute Error)

$$J(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{N} |\hat{\mathbf{y}} - \mathbf{y}| = \frac{1}{N} \sum_{i=0}^N |\hat{y}_i - y_i|. \quad (10.19)$$

Tyto chybové funkce naopak vykazují vyšší hodnoty pro chyby blízké 0 oproti střední kvadratické chybě. Poslední chybovou funkcí, kterou si uvedeme je Huberova funkce [54], která je kombinací střední kvadratické chyby a střední absolutní chyby, definovaná jako

$$J(\hat{\mathbf{y}}, \mathbf{y}) = \min \left\{ \frac{|\hat{\mathbf{y}} - \mathbf{y}|}{N}, \frac{\|\hat{\mathbf{y}} - \mathbf{y}\|^2}{N} \right\}. \quad (10.20)$$

Znázornění uvedených chybových funkcí je uvedeno na obrázku 10.4.

10.1.4 Konvoluční vrstva

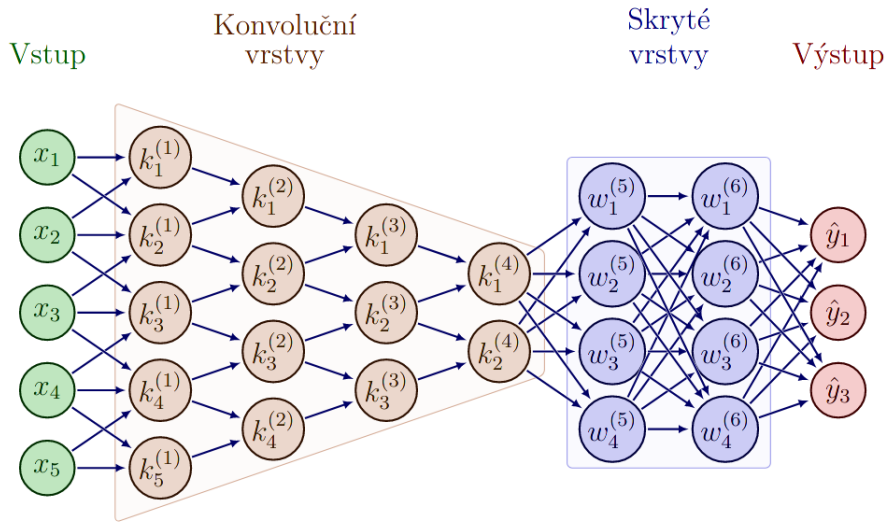
Konvoluce je dobře známý operátor z teorie signálů a systémů. V neuronových sítích se využívá konvoluční filtr k extrakci příznaků [55]. V dnešních moderních architekturách se využívá velkého množství konvolučních vrstev k extrahování co největšího množství příznaků ze vstupu. Konvoluční vrstvy se z toho důvodu využívají jako první vrstvy sítě, z důvodu extrahování příznaků.

Diskrétní konvoluci značíme

$$\mathbf{y} = \mathbf{x} * \mathbf{k}, \quad (10.21)$$

kde \mathbf{x} je vstupní vektor \mathbf{k} je vektor reprezentující konvoluční vrstvu a \mathbf{y} je výstupní vektor. Pro složky výstupního vektoru platí

$$y_n = \sum_{i=1}^K x_{n-i} \cdot k_i, \quad (10.22)$$



Obrázek 10.5: Schéma neuronové sítě s konvolučními a skrytými vrstvami

kde K je počet koeficientů k . Při učení sítě je potřeba správně nastavit koeficienty \mathbf{k} , aby síť dokázala na vstupu správně extrahovat příznaky. Schéma neuronové sítě obsahující konvoluční a skryté vrstvy je zobrazeno na obrázku 10.5.

10.2 Rekurentní neuronové sítě

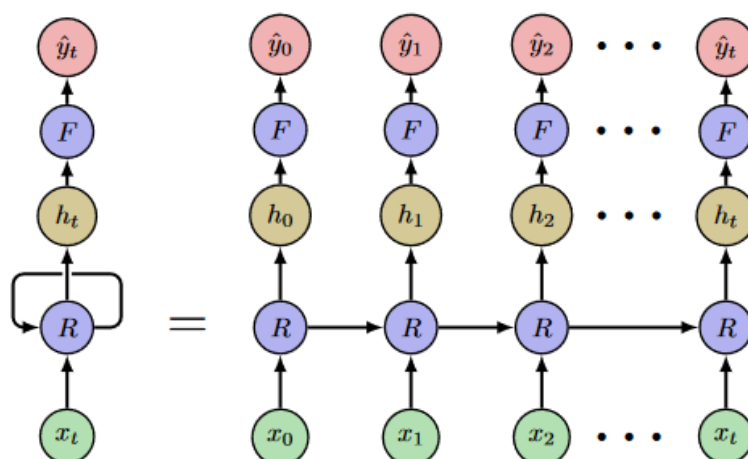
Princip neuronových sítí uvedený v předchozí části předpokládá pouze dopředné šíření výpočtu a neuvažoval, že by stávající výsledek mohl jakýmkoliv způsobem ovlivnit následující výpočet (během jedné epochy učení). Původní neuronové sítě tak neuvažovaly závislost mezi po sobě následujícími daty a jejich cílem bylo pouze správně vyhodnotit aktuální data.

Rekurentní neuronová síť je speciálním typem neuronové sítě. Na rozdíl od klasické sítě využívá informace z obdržené z předchozích dat a využívá je k ovlivnění výpočtu. Rekurentní neuronová síť tak využívá principu zpětné vazby [56]. Zpětnovazební smyčka je ovšem v architektuře neuronové sítě problematická, z důvodu zpětné propagace gradientu. Tento problém, lze vyřešit aproximací nekonečné zpětnovazební smyčky pomocí periodického opakování struktury [57]. Počet opakování základní struktury se nazývá stupeň rekurence. V rámci vyhodnocování si síť uchovává vnitřní stav \mathbf{h}_t , který následně ovlivní výstupní hodnotu $\hat{\mathbf{y}}_t$

$$h_t = R(\mathbf{x}_t, \mathbf{h}_{t-1}, \mathbf{w}), \quad (10.23)$$

$$\mathbf{y}_t = F(\mathbf{h}_t, \mathbf{v}), \quad (10.24)$$

kde \mathbf{w} a \mathbf{v} představují parametry rekurentní sítě. Struktura neuronové sítě popsaná stavovými rovnicemi (10.23) a (10.24) je na obrázku 10.6. Rekurentní



Obrázek 10.6: Schéma rekurentní neuronové sítě

neuronové sítě se v dnešní době využívají v širokém spektru problémů jako je analýza textu, rozpoznávání hlasu nebo právě pro predikci. Rekurentní neuronová síť tak bude základním stavebním prvkem našeho prediktoru. Nevýhodou uvedené rekurentní sítě je, že informace o minulých vstupech je podmíněna stupněm rekurence [58]. Pro správně fungující prediktor je dobré mít k dispozici co nejvíce informací o předchozích vzorcích, zavést tedy jistou paměťovou buňku.

10.2.1 LSTM architektura

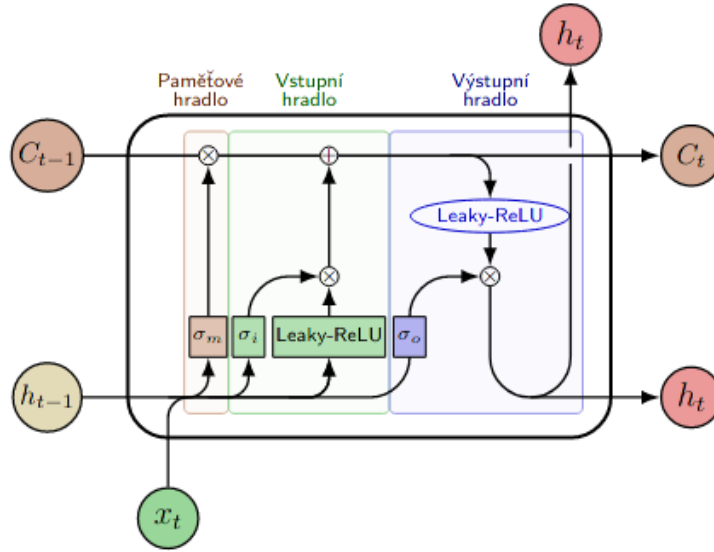
Požadavek na paměť uvnitř sítě se podařilo určitým způsobem dosáhnout pomocí LSTM architektury [59]. LSTM je zkratka z anglického názvu Long Short-Term Memory, což lze volně přeložit jako dlouhá krátkodobá paměť. LSTM architektura využívá stejného principu jako rekurentní síť, ale zde je funkce vnitřních stavů $R(x_t, h_{t-1}, w)$ reprezentována LSTM buňkou.

LSTM buňka vyhodnocuje kromě vnitřního stavu sítě i svůj vlastní stav C_t , který nám představuje paměť. Zde je potřeba uvědomit si rozdíl mezi vnitřním stavem h_t a vlastním stavem C_t . Vnitřní stav nese informaci pouze o vstupní hodnotě x_t , ale vlastní stav buňky představuje agregaci dosavadních hodnot. Tento stav tedy uchovává informaci, ohledně toho jaké hodnoty již viděl, kdežto vnitřní stav přenáší pouze informaci o tom jakou hodnotu právě vidí. Vnitřní stav a vlastní jsou propojeny pomocí hradel spolu se vstupní hodnotou. LSTM buňka obsahuje 3 hradla [60], paměťové hradlo, vstupní hradlo a výstupní hradlo. Schéma LSTM buňky je na obrázku 10.7.

Paměťové hradlo zpracovává informace ze vstupu a vnitřního stavu

$$\sigma_m = \sigma(w_{h,m} \cdot h_{t-1} + w_{x,m} \cdot x_t + w_{m,0}). \quad (10.25)$$

Vstupní hradlo funguje na velmi podobném principu jako paměťové hradlo,



Obrázek 10.7: LSTM buňka

tedy zpracovává vstup x_t a hodnotu vnitřního stavu h_{t-1}

$$\sigma_i = \sigma(w_{h,i} \cdot h_{t-1} + w_{x,i} \cdot x_t + w_{i,0}), \quad (10.26)$$

ale výstup σ_i je ještě násoben hodnotou σ_a , která se nazývá aktivátor vstupu

$$\sigma_a = \text{LReLU}(w_{h,a} \cdot h_{t-1} + w_{x,a} \cdot x_t + w_{a,0}). \quad (10.27)$$

Jedná se o ten stejný výpočet, ale s jinou aktivační funkcí v našem případě Leaky-ReLU. Po vyhodnocení výstupu paměťového hradla σ_m a vstupního hradla $\sigma_i \cdot \sigma_a$ se provede výpočet vlastního stavu

$$C_t = C_{t-1} \cdot \sigma_m + \sigma_i \cdot \sigma_a, \quad (10.28)$$

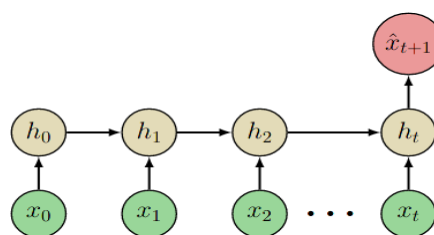
a následně se vyhodnotí hodnota výstupního hradla

$$\sigma_o = \sigma(w_{h,o} \cdot h_{t-1} + w_{x,o} \cdot x_t + w_{o,0}) \quad (10.29)$$

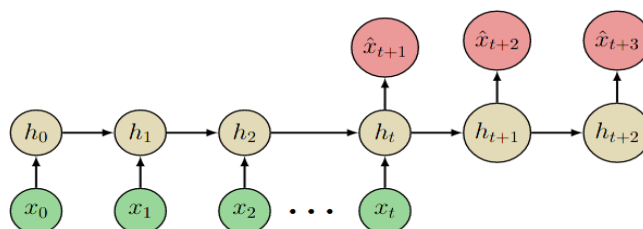
a vypočítá se vnitřní stav

$$h_t = \text{LReLU}(C_t) \cdot \sigma_o. \quad (10.30)$$

LSTM architektura má v rámci rekurentních sítí velké využití [61][62]. Hlavní výhoda spočívá ve vnitřní agregaci informace, což umožňuje síti zpracovat velmi dlouhý řetězec dat s mnohonásobně menším stupněm rekurence. Rovněž díky hradlům se v síti dobře šíří zpětně propagovaný gradient [63] (v klasické rekurentní síti hrozí velká akumulace gradientu) a díky své poměrně malé velikosti se síť i rychle učí.



Obrázek 10.8: „Many to One“ realizace



Obrázek 10.9: Many to Many realizace

10.2.2 „Many to One“ a „Many to Many“ architektury

Jak již bylo zmíněno výše, tak rekurentní sítě se využívají na zpracování posloupnosti dat. Z tohoto důvodu jsou vhodné pro analýzu časové řady z hlediska predikce budoucích hodnot na základě hodnot minulých. Díky LSTM buňkám, které jsou schopny si uchovávat vlastní stavy o dosavadních datech je možné určit predikci příštího vzorku dle znalosti několika posledních hodnot. Tato predikce může být jednokroková, kde dochází k predikci pouze jedné hodnoty, nebo vícekroková kde se predikuje delší časový horizont.

V rámci predikce jednoho vzorku dle několika minulých mluvíme o predikci „Many to One“ [64]. Tato predikce je tedy funkcí předchozích hodnot, což lze zapsat jako

$$\hat{y}_t = \hat{x}_{t+1} = F(x_t, x_{t-1}, \dots, x_{t-k}). \quad (10.31)$$

Architektura „Many to One“ bývá často realizována pomocí standardní rekurentní neuronové sítě, nebo pomocí rekurentní sítě s LSTM buňkami. Výstupem je jedna hodnota, která vznikne obvykle použitím standardních skrytých vrstev do kterých vstupují vnitřní proměnné. Ukázka primitivní „Many to One“ architektury je na obrázku 10.8.

„Many to many“ architektura se pak využívá k predikci delšího časového horizontu najednou dle posloupnosti několika minulých hodnot. Výstupní vektor je opět funkcí předchozích hodnot

$$\hat{\mathbf{y}}_t = (\hat{x}_{t+1}, \hat{x}_{t+2}, \dots, \hat{x}_{t+m}) = F(x_t, x_{t-1}, \dots, x_{t-k}). \quad (10.32)$$

Architektury „Many to Many“ [65] využívají jak klasických rekurentních sítí a LSTM buněk, kde se výstupní vektor vypočítá z vnitřních proměnných

pomocí skrytých vrstev, tak se využívá i složitějších technik rekurentních sítí jako je například použití enkodéru [66]. V případě použití enkodéru máme na výstupu více „Many to One“ výstupů, kterým jsou vstupy určeny pomocí enkodéru vnitřních stavů.

10.3 Návrh prediktoru

Vzhledem k úloze potřebuje predikovat 4 časové hodnoty za horizontem. V rámci návrhu jsme zkusili navrhnout prediktor na principu „Many to One“ i prediktor „Many to Many“. Jednokrokový prediktor „Many to One“ jsme pak předělali na vícekrokový pomocí predikce následujících hodnot z již predikovaných. K dispozici jsme měli data požadavků v roce 2015 až 2021. Dohromady jsme tak měli k dispozici 3 629 000 hodnot.

10.3.1 „Many to One“

Velikost vstupu

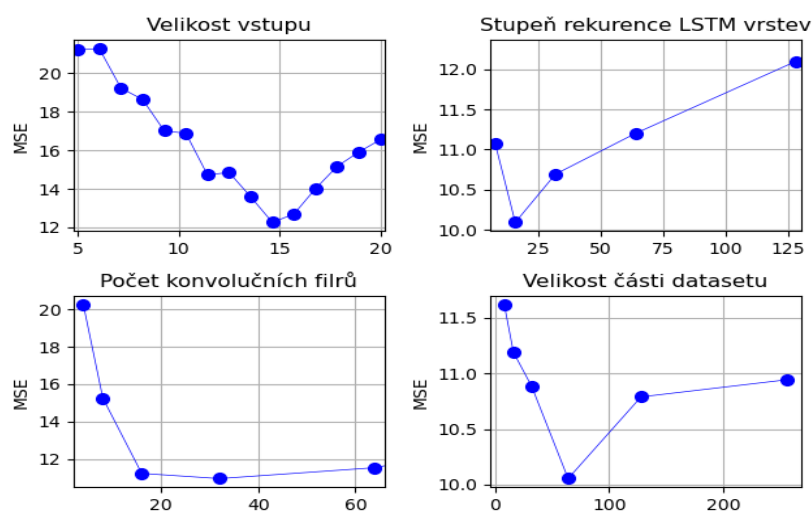
Prvním problémem byla volba velikosti vstupního vektoru. Velikost jsme volili na základě chyby jednokrokového elementárního prediktoru tvořeného jednou LSTM vrstvou. Testovali jsme velikost vstupu pro celočíselné hodnoty z intervalu $\langle 5, 20 \rangle$. Nejlepší vlastnosti vykazovala síť pro velikost vstupu $k_{in} = 15$. Zobrazeno na 10.10.

Rekurence

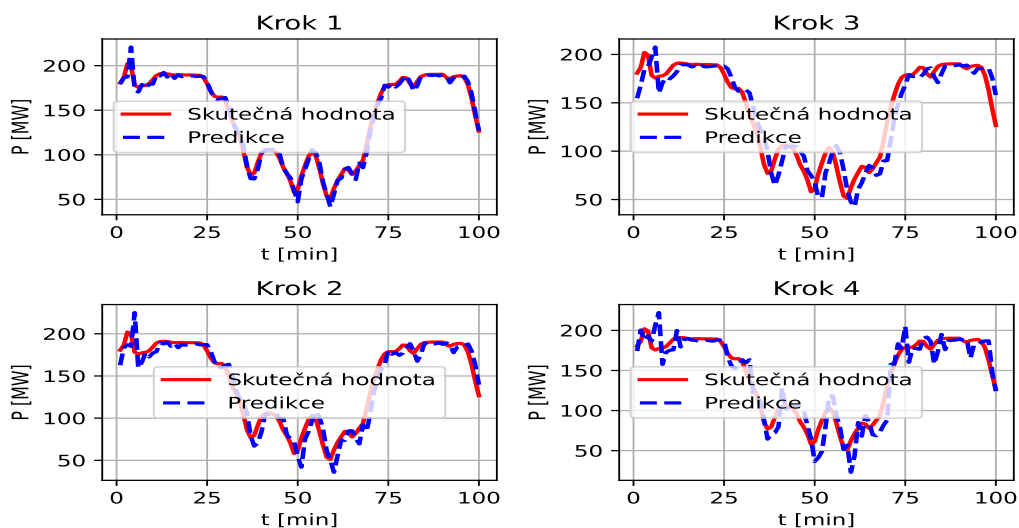
Dále bylo potřeba zvolit příslušnou architekturu neuronové sítě. V rámci predikce na principu „Many to One“, jsme vytvořili rekurentní síť s LSTM buňkami. Pro jednu rekurentní LSTM vrstvu jsme však nedosáhli dobrých výsledků. Přidáním dalších 2 LSTM vrstev se, ale přesnost predikce velmi zlepšila. Zde bylo důležité zvolit stupeň rekurence. Opět jsme testovali několik hodnot. Tentokrát jsme, ale vzhledem k vlastnostem neuronových sítí volili mocniny 2. Nejlepší hodnoty predikce jsme dosáhli pro stupeň rekurence 16. Zobrazeno na 10.10.

Konvoluční vrstva

Pro získání více příznaků jsme před LSTM část sítě přidali konvoluční vrstvy. Zde bylo opět potřeba zvolit vhodný počet konvolučních vrstev. Postupně jsme tak navyšovali počet konvolučních vrstev a v každé vrstvě jsme přidávali dvojnásobný počet filtrů než ve vrstvě předchozí. První vrstva obsahovala 4 konvoluční filtry. Nejlepší výsledky predikce jsme získali pro 4 konvoluční vrstvy, kde poslední vrstva obsahovala 32 filtrů. Za LSTM vrstvou jsme ještě umístili klasickou skrytou vrstvou pro lepší vyhodnocení výsledků. Nejlépe vycházela skrytá vrstva, která obsahovala 8 neuronů a za ní následovala výstupní vrstva, která vracela 1 hodnotu.



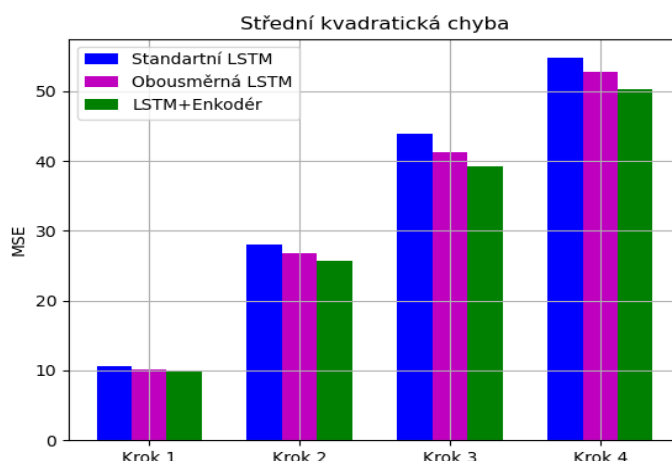
Obrázek 10.10: Závislost chyby predikce na volbě parametrů



Obrázek 10.11: Porovnání predikce pro 4 kroky prediktoru založeném na „Many to One“ principu predikce

Parametry učení

Posledním krokem bylo zvolit parametry učení, tedy velikost části datasetu, na které se provede optimalizační krok, optimalizační metodu a chybovou funkci. Nejlepší velikost testovací části vycházela 64 vzorků. Jako optimalizační metodu jsme volili ADAM [67]. Jako chybovou funkci jsme nejprve volili střední kvadratickou chybu, následně jsme ale zjistili, že lepších výsledků dosáhneme pomocí Huberovy chybové funkce. Neuronovou síť jsme natrénovali na 100 epochách s postupně se snižující konstantou μ . Závislost jednotlivých parametrů sítě a parametrů učení je na obrázku 10.10. Před učením jsme provedli normalizaci dat, po predikci jsme provedli zpětnou transformaci.



Obrázek 10.12: Porovnání přesnosti predikce pro různé architektury „Many to Many“ prediktoru

10.3.2 „Many to many“

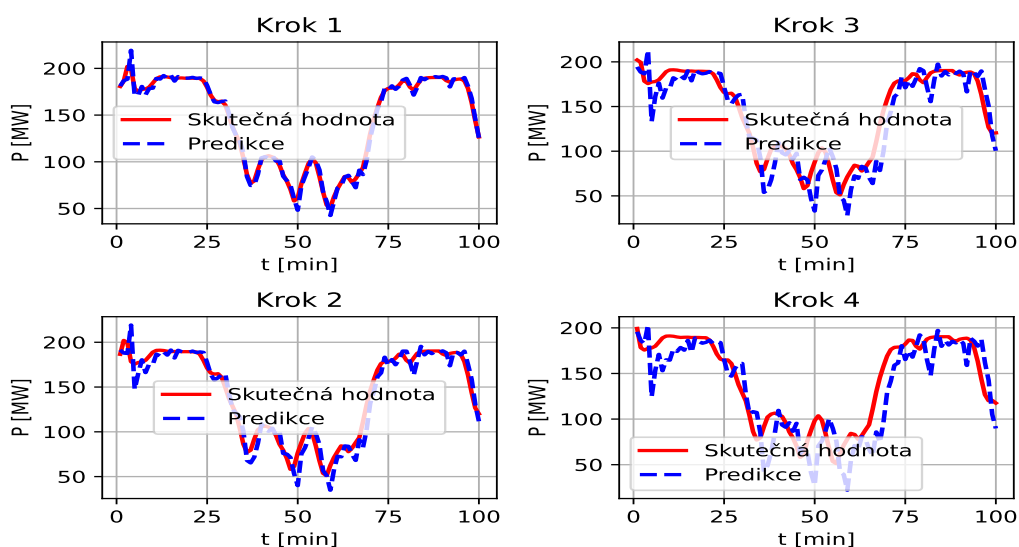
Po natrénování jednokrokového prediktoru jsme vytvořili vícekrokový prediktor, kde jsme využívali k následující predikci již predikované hodnoty. Pro lepší výsledky predikce jsme přidali za prediktor nelineární funkci, která způsobovala omezení gradientu predikce a v určitých případech aproximovala predikci pomocí odmocniny. Ukázka predikce pomocí prediktoru založeného na „Many to One“ přístupu je zobrazena na obrázku 10.11. Schéma architektury navržené sítě je na obrázku 10.16.

Při návrhu prediktoru "Many to Many" jsme vycházeli z „Many to One“ přístupu. Velikost vstupu jsme volili 15 hodnot. Predikovali jsme 4 hodnoty za horizontem. V rámci architektury jsme zkoušeli klasickou LSTM [68] s více výstupy, Obousměrnou LSTM [69] s více výstupy a LSTM s enkodérem [70]. U LSTM obousměrné LSTM jsme přidali 1 skrytou vrstvu s šestnácti neurony, čímž jsme dosáhli lepších výsledků. V rámci architektury s enkodérem, jsme použili původní „Many to One“ prediktor, ale na jeho výstup jsme přidali enkodér a následně jsme přidali další LSTM vrstvu.

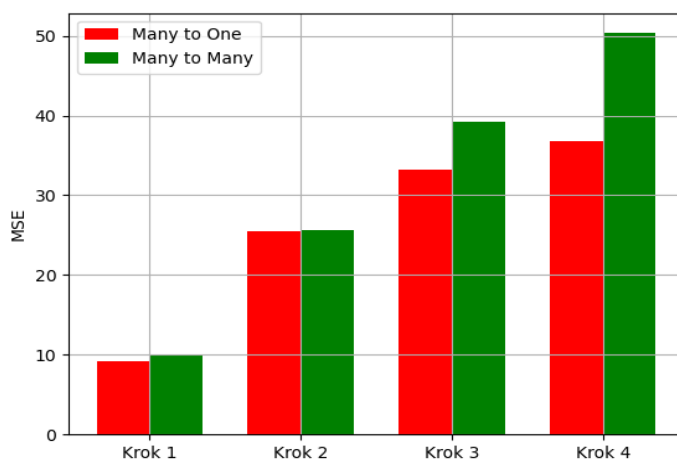
Porovnání výsledků pro výše uvedené architektury je znázorněno v grafu na obrázku 10.12. Z grafu vidíme, že nejlepší výsledky má architektura s enkodérem. Ukázky jednotlivých kroků vícekrokové predikce jsou na obrázku 10.13. Schéma architektury „Many to Many“ je na obrázku 10.17.

10.4 Výsledný prediktor

V minulé sekci jsme navrhli 2 prediktory. Prediktor „Many to One“ na základě vstupních dat predikoval, jednu hodnotu. Vícekrokový prediktor jsme pak navrhli jako rekurentní opakování tohoto prediktoru. Prediktor „Many to Many“ udělal všechny predikce najednou. Porovnání MSE těchto prediktorů



Obrázek 10.13: Porovnání predikce pro 4 kroky prediktoru založeném na „Many to Many“ principu predikce



Obrázek 10.14: Porovnání přesnosti predikce „Many to One“ a „Many to Many“ architektur

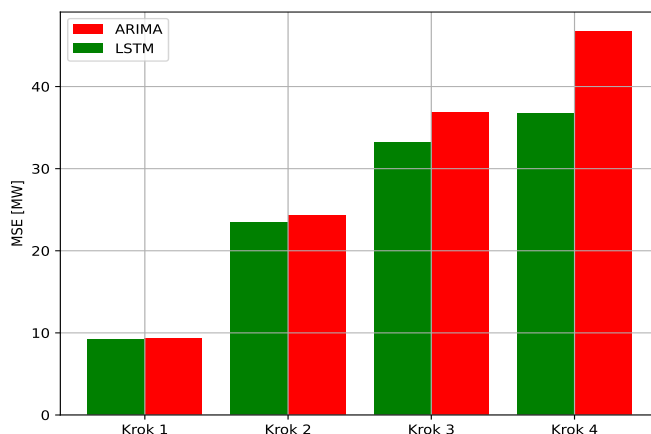
Architektura	MSE krok 1	MSE krok 2	MSE krok 3	MSE krok 4
„Many to One“	9.21 MW	25.52 MW	33.23 MW	36.74 MW
„Many to Many“	9.88 MW	25.65 MW	39.24 MW	50.34 MW

Tabulka 10.1: Hodnoty MSE jednotlivých kroků „Many to One“ a „Many to Many“ architektury

je uvedeno v tabulce 10.1 a v grafu na obrázku 10.14. Z grafu jasně vyplývá, že lepších výsledků dosahuje prediktor založený na principu „Many to One“. Pro náš online algoritmus tak využijeme tento prediktor, jehož chyba $\pm 36.74 \text{ MW}$ ve čtvrtém kroku jsou necelá 4% celkového rozsahu aFRR $\pm 1000 \text{ MW}$

Architektura	MSE krok 1	MSE krok 2	MSE krok 3	MSE krok 4
LSTM	9.21 MW	25.52 MW	33.23 MW	36.74 MW
ARIMA	9.36 MW	26.33 MW	36.86 MW	46.74 MW

Tabulka 10.2: Porovnání MSE prediktorů ARIMA a LSTM pro jednotlivé kroky predikce



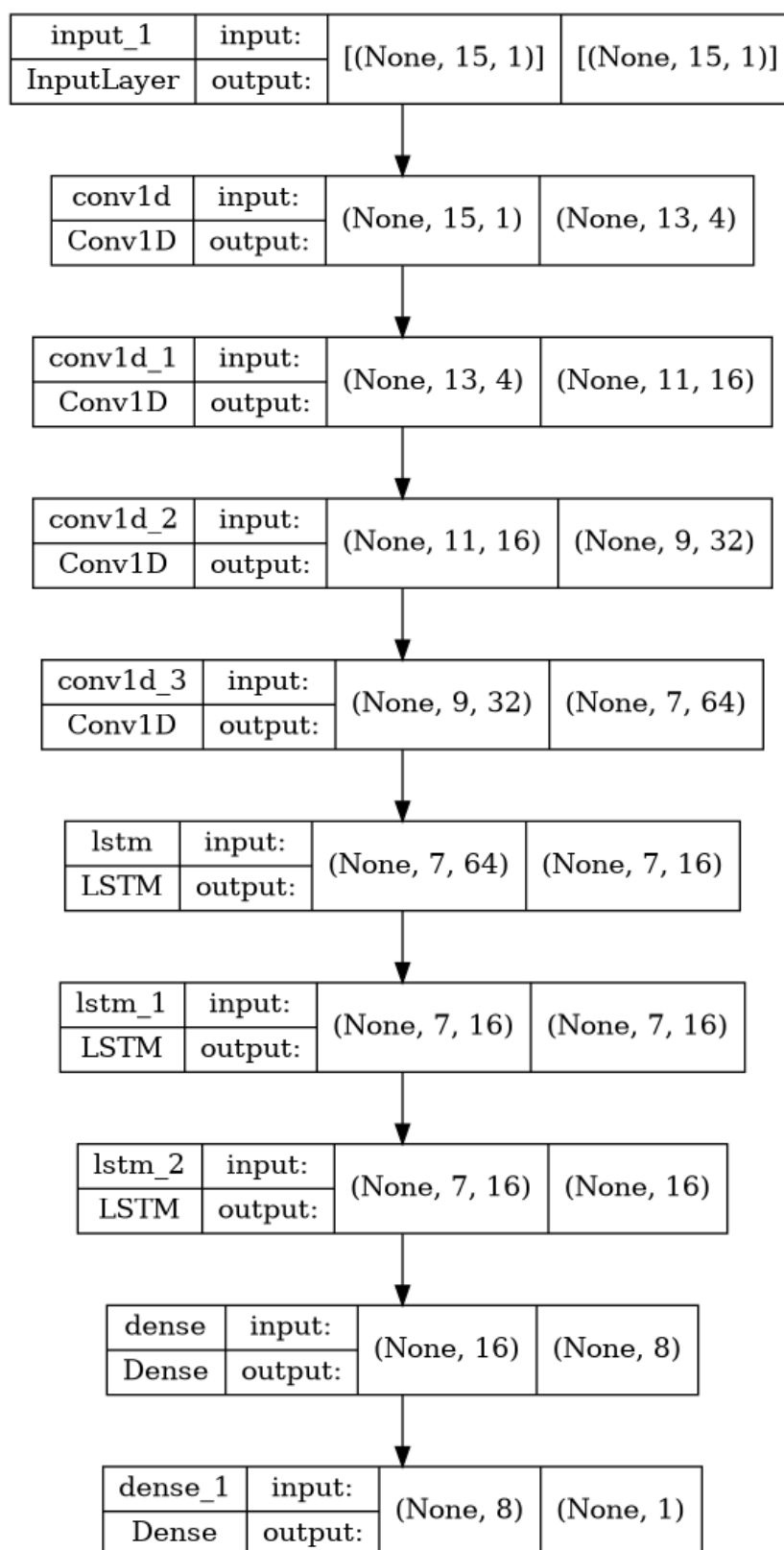
Obrázek 10.15: Porovnání MSE prediktorů ARIMA a LSTM pro jednotlivé kroky predikce

10.4.1 Porovnání LSTM prediktoru s ARIMA prediktorem

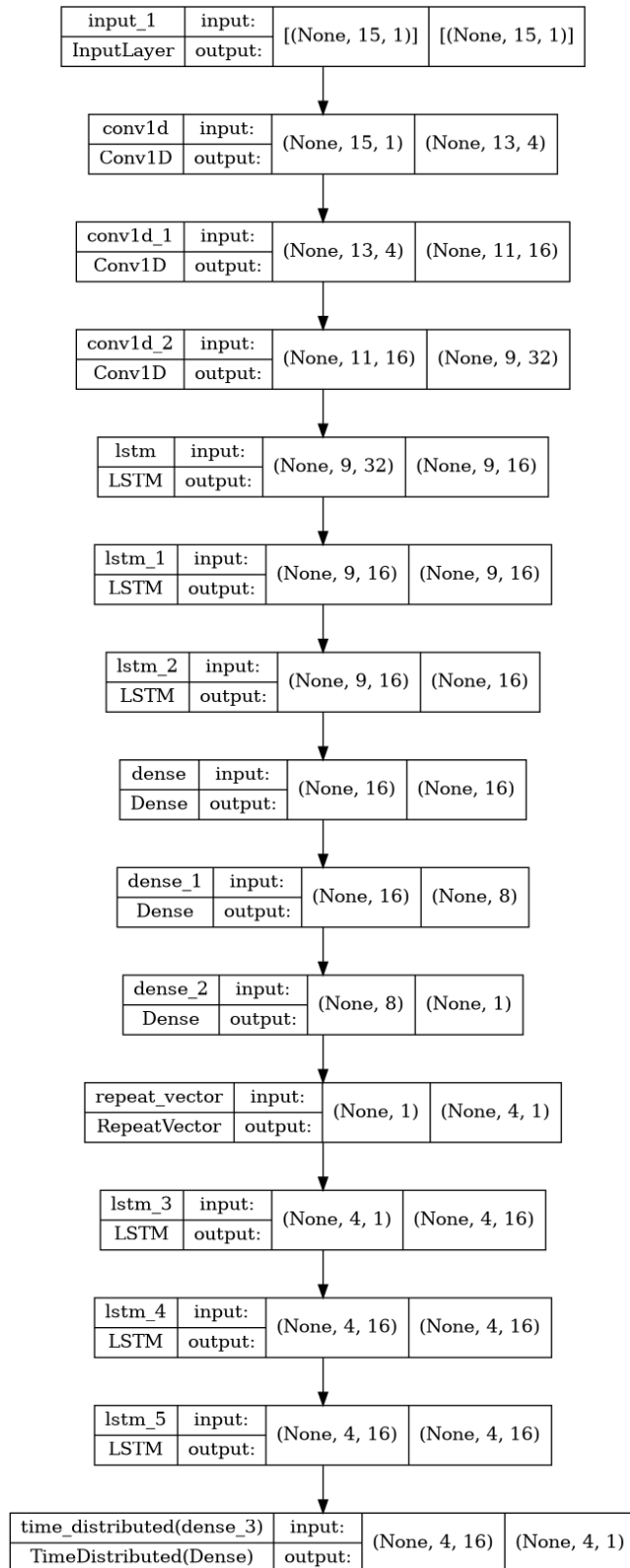
Prediktor využívající LSTM architekturu rekurentní sítě je nelineárním prediktorem. Oproti tomu ARIMA model je autoregresní model, kde výsledná predikovaná hodnota vzniká jako lineární kombinace předchozích hodnot. Prediktor s LSTM architekturou byl navíc natrénovaný na mnohem větším datasetu a jeho proces učení je mnohem komplexnější než stanovení koeficientů ARIMA modelu.

V případě ARIMA modelu dokážeme odhadnout, jak tento prediktor bude reagovat v případě, že nastane limitní situace, na kterou jsme v trénovacím datasetu nenarazili. Tím pádem máme algoritmus, jehož chování bezpečně známe. To v případě neuronové sítě zaručit nemůžeme. Nevýhodou neuronové sítě je její komplexita a tento prediktor nám tak vytváří černou skříňku do které z hlediska algoritmu nevidíme a nevíme, jak se bude chovat v případě dat, která jsou velmi odlišná od trénovacích dat.

Hodnoty MSE pro jednotlivé kroky ARIMA prediktoru a LSTM prediktoru jsou uvedeny v tabulce 10.2. a zobrazeny na obrázku 10.15. Z obrázků můžeme vidět, že LSTM prediktor vykazuje menší MSE pro všechny čtyři kroky predikce a využití neuronové sítě je tak na místě. V následující kapitole se zaměříme na experimenty, kde budeme testovat online model s oběma prediktory zvlášť a výsledky následně porovnáme vyhodnotíme tak efektivitu těchto prediktorů.



Obrázek 10.16: Schéma architektury rekurentní sítě „Many to One“



Obrázek 10.17: Schéma architektury rekurentní sítě „Many to Many“

Kapitola 11

Testování modelů

V kapitole 7 jsme zavedli MILP model, který dokáže nalézt optimální rozložení na zadaném časovém intervalu Ω^T . Následně jsme v kapitole 8 zavedli online algoritmus, který využívá upraveného MILP modelu a funguje na principu posuvného okna, kde má k dispozici jen omezený časový úsek $\Omega^{\hat{T}}$ a hledá optimální rozložení v čase t_a , který je středem intervalu $\Omega^{\hat{T}}$. Ke správnému fungování tohoto algoritmu je potřeba predikovat hodnoty služby aFRR, která nám do úlohy přináší stochastický charakter. Predikci hodnot realizujeme pomocí rekurentní neuronové sítě, kterou jsme si představili v minulé kapitole. Neuronová síť se ovšem spoléhá na gradientní algoritmy a může tedy nastat situace, kdy vstupem budou data velmi odlišná od dat z trénovacího datasetu a neuronová síť tak může selhat. Proto využíváme lineární referenční ARIMA prediktor, který je schopen predikovat hodnoty za horizontem jako lineární kombinaci předchozích hodnot.

V této kapitole se zaměříme na testování offline algoritmu, který nalezne na Ω^T optimální rozložení výkonů uvnitř agregačního bloku na základě známých požadavků a následně budeme testovat online algoritmus, který nalezne rozložení v reálném čase. Výsledné průběhy výkonových rozložení porovnáme a zhodnotíme kvalitu online algoritmu vůči optimálnímu průběhu. Testování rozdělíme do několika kategorií. Nejprve budeme testovat výpočetní náročnost algoritmu vzhledem k velikosti $|\Omega^{\hat{T}}|$ a počtu jednotek agregačního bloku, tedy velikosti množiny $|\Omega^{AB}| = |\Omega^{ER}| + |\Omega^{ESS}|$. Poté si ověříme správnost našeho algoritmu na modelovém případě a následně otestujeme jeho spolehlivost algoritmu při velmi dynamických změnách požadavků na podpůrné služby aFRR a mFRR. V poslední řadě budeme testovat algoritmus na datech z reálného provozu.

11.1 Testování výpočetní náročnosti algoritmu

Nejprve otestujeme výpočetní náročnost našeho algoritmu. Výpočetní náročnost budeme testovat vzhledem k počtu jednotek tvořící agregační blok Ω^{AB} a vzhledem k délce časového intervalu Ω^T . Začneme na velikosti agregačního bloku tvořeného šesti jednotkami, z čehož jsou 4 výrobní jednotky a 2 akumu-

látory a velikosti časového intervalu $|\Omega^T| = 9$ což velikosti našeho časového okna definovaného v kapitole 8.

Postupně budeme navyšovat velikost Ω^T o jednotky minut a budeme zvyšovat počet jednotek tvořící agregační blok, přičemž zachováme poměr 2 : 1 počtu výrobních jednotek a počtu akumulátorů. Parametry výrobních jednotek a akumulátorů budeme generovat následovně pomocí rovnoměrných rozdělení $\mathcal{U}(a, b)$

$$\bar{P}_i = \mathcal{U}(10, 20) \quad \forall i \in \Omega^{ER}, \quad (11.1)$$

$$\underline{P}_i = 1 \quad \forall i \in \Omega^{ER}, \quad (11.2)$$

$$T_i = \mathcal{U}(1, 4) \quad \forall i \in \Omega^{ER}, \quad (11.3)$$

$$C_i^a = \mathcal{U}(200, 400) \quad \forall i \in \Omega^{ER}, \quad (11.4)$$

$$C_i^d = \mathcal{U}(200, 400) \quad \forall i \in \Omega^{ER}, \quad (11.5)$$

$$a_i = \mathcal{U}(50, 100) \quad \forall i \in \Omega^{ER}, \quad (11.6)$$

$$b_i = \mathcal{U}(20, 70) \quad \forall i \in \Omega^{ER}, \quad (11.7)$$

$$\overline{SOC}_i = \mathcal{U}(300, 1000) \quad \forall i \in \Omega^{ESS}, \quad (11.8)$$

$$\underline{SOC}_i = 0 \quad \forall i \in \Omega^{ESS}, \quad (11.9)$$

$$\bar{P}_i^c = \mathcal{U}(10, 50) \quad \forall i \in \Omega^{ESS}, \quad (11.10)$$

$$\underline{P}_i^c = 0 \quad \forall i \in \Omega^{ESS}, \quad (11.11)$$

$$\bar{P}_i^d = \bar{P}_i^c \quad \forall i \in \Omega^{ESS}, \quad (11.12)$$

$$\underline{P}_i^d = 0 \quad \forall i \in \Omega^{ESS}, \quad (11.13)$$

$$T_i^c = \mathcal{U}(5, 15) \quad \forall i \in \Omega^{ESS}, \quad (11.14)$$

$$T_i^d = T_i^c \quad \forall i \in \Omega^{ESS}, \quad (11.15)$$

$$\eta_i = 1 \quad \forall i \in \Omega^{ESS}, \quad (11.16)$$

$$a_i^c = \mathcal{U}(80, 150) \quad \forall i \in \Omega^{ESS}, \quad (11.17)$$

$$b_i^c = \mathcal{U}(50, 100) \quad \forall i \in \Omega^{ESS}, \quad (11.18)$$

$$a_i^d = 0 \quad \forall i \in \Omega^{ESS}, \quad (11.19)$$

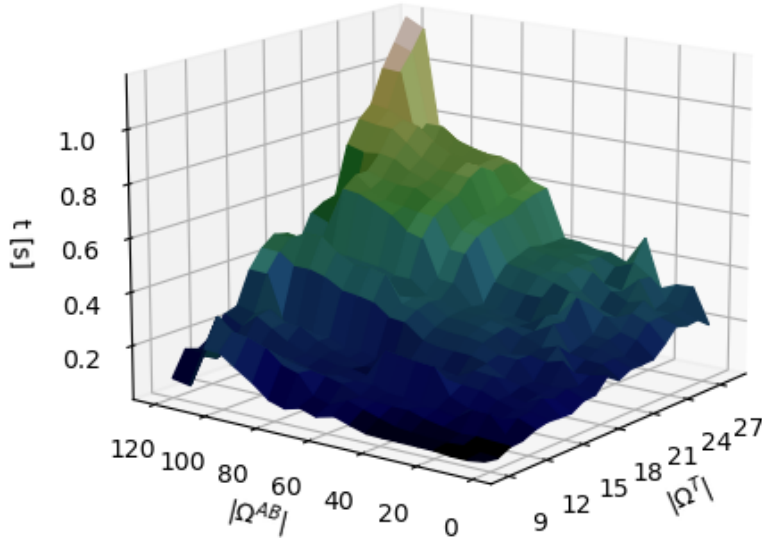
$$b_i^d = 0 \quad \forall i \in \Omega^{ESS}. \quad (11.20)$$

Požadavky na diagramový výkon budeme generovat jako

$$P_t^{DG} = \frac{1}{|\Omega^{ER}|} \sum_{i \in \Omega^{ER}} \frac{\bar{P}_i + 1}{2} \quad \forall t \in \Omega^T, \quad (11.21)$$

požadavky na službu mFRR s plánovanou aktivací budeme generovat pomocí rovnoměrného rozdělení jako

$$mFRR_t^a = K \cdot \mathcal{U}(-5, 5) \quad \forall t \in \Omega^T, \quad (11.22)$$



Obrázek 11.1: Závislost výpočetní náročnosti na $|\Omega^T|$ a $|\Omega^{AB}|$

kde

$$K = |\Omega^{ER}| + |\Omega^{ESS}|. \quad (11.23)$$

Požadavky na aFRR vytvoříme z harmonického průběhu s přidáním Gausovským šumem.

$$aFRR_t^a = K \left(0.1 \cdot \sin \left(2\pi \frac{t}{10} \right) + 0.05 \cdot \mathcal{N}(0, 1) \right) \quad \forall t \in \Omega^T, \quad (11.24)$$

kde $\mathcal{N}(0, 1)$ značí Gausovské rozdělení s parametry $\mu = 0$, $\sigma = 1$. Generování těchto výkonových požadavků nám tak zajistí podobné provozní podmínky a testování složitosti bude v pořádku.

Kvůli stochastickému generování instancí opakujeme test 10x. Výsledný čas výpočtu pro jednotlivé velikosti $|\Omega^T|$, $|\Omega^{AB}|$ pak odpovídá mediánu jednotlivých výpočetních časů. Výpočetní náročnost je zobrazena na obrázku 11.1. Výpočetní náročnost asymptoticky odpovídala $\mathcal{O}(e^{|\Omega^T| + |\Omega^{AB}|})$. Toto odpovídá komplexitě NP obtížné úlohy, kterou představuje MILP.

V případě největší instance $|\Omega^{AB}| = 120$ a $|\Omega^T| = 29$ byl výpočetní čas $t = 1.15s$. Toto je pro výpočty v reálném čase již poměrně dlouhá doba. V rámci predikce tedy nemá cenu uvažovat příliš dlouhé intervaly.

i	\bar{P}_i	\underline{P}_i	T_i	a_i	b_i	C_i^a	C_i^d
1	100	50	2	100	20	∞	∞
2	100	50	2	100	20	∞	∞
3	100	50	5	200	40	∞	∞
4	100	50	5	200	40	∞	∞

Tabulka 11.1: Parametry výrobních jednotek při dynamickém testování

i	$\overline{\text{SOC}}_i$	$\underline{\text{SOC}}_i$	\bar{P}_i^c	\underline{P}_i^c	\bar{P}_i^d	\underline{P}_i^d	T_i^c	T_i^d	a_i^c	b_i^c	a_i^d	b_i^d
1	40000	0	300	0	300	0	30	30	300	80	0	0
2	40000	0	300	0	300	0	30	30	300	80	0	0

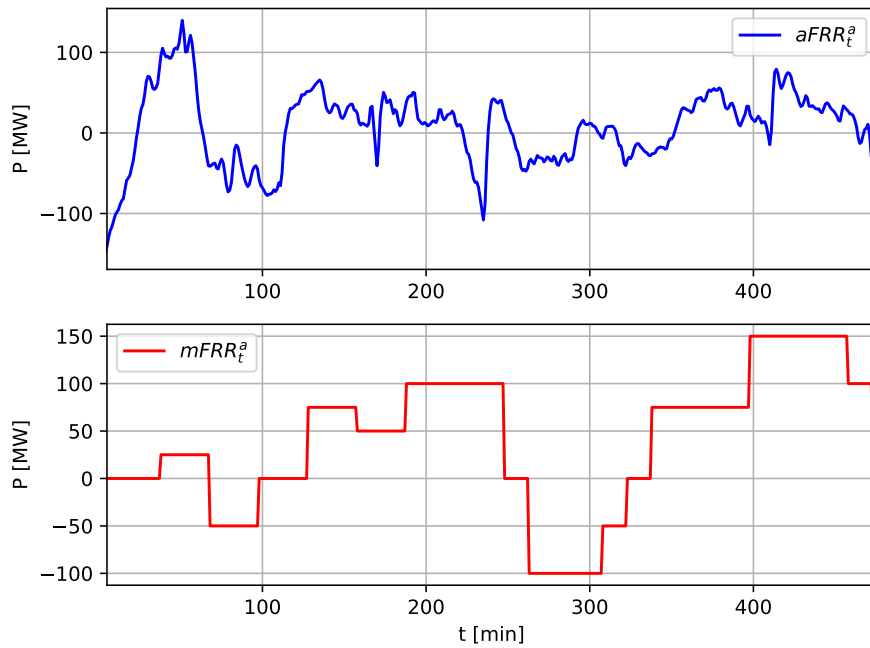
Tabulka 11.2: Parametry akumulátorů při dynamickém testování

11.2 Testování online modelu

V této části otestujeme online model s ARIMA prediktorem a online model s LSTM prediktorem a jejich výsledné rozdělení výkonů uvnitř agregačního bloku porovnáme s optimálním rozdělením nalezené pomocí offline modelu. Pro testování jsme si vytvořili agregační blok, který tvoří 4 výrobní jednotky a 2 akumulátory s parametry uvedenými v tabulce 11.1. a 11.2.

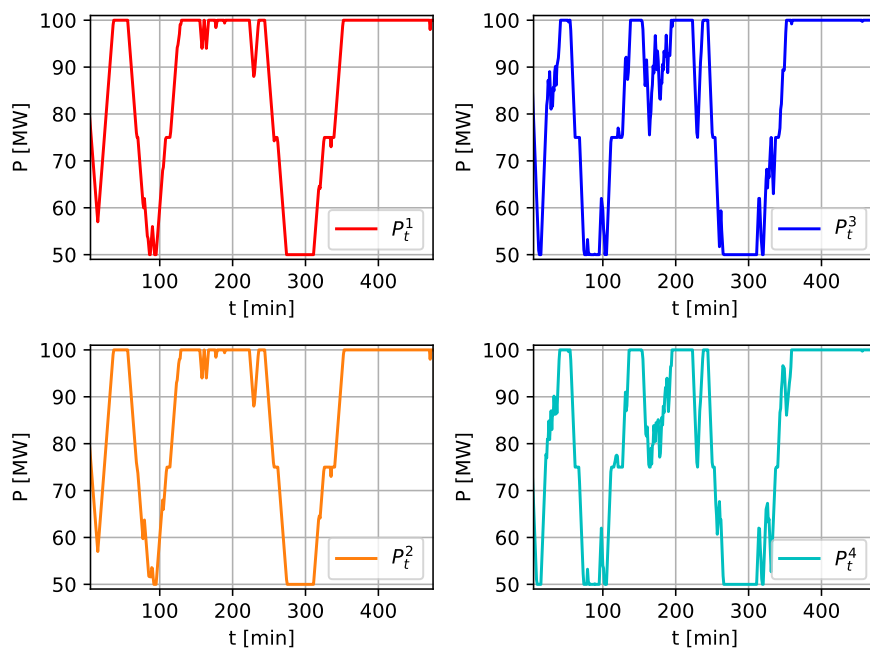
Průběh výkonových požadavků vychází z reálných dat. Délku testovacího intervalu Ω^T jsme volili 6 hodin. Velikost Diagramový výkon volíme konstantní $P_t^{DG} = 300MW \forall t \in \Omega^T$. Průběh požadavků $aFRR_t^a$ a $mFRR_t^a$ je zobrazen na obrázku 11.2. Výsledné rozložení výkonů na výrobní jednotky a akumulátory agregačního bloku pro jednotlivé algoritmy je zobrazeno na obrázcích 11.3 až 11.8. V tabulce 11.3 jsou uvedeny výsledné náklady pro jednotlivé algoritmy.

V rámci tohoto experimentu se snažíme vytvořit ideální agregační blok a generovat výkonové požadavky, které by odpovídaly reálnému agregačnímu bloku s podobnými technickými parametry. Záměrně volíme dva typy výrobních jednotek s různými hodnotami T_i pro $i \in \{1, 2, 3, 4\}$ a akumulátory s vyššími koeficienty pro nabíjecí proces. Tato volba nám ovlivní průběhy výsledných výkonových rozdělení pro online a offline algoritmus. Pokud jsou naše heuristiky správné, měl by online algoritmus najít dražší řešení, jelikož by se měl snažit udržovat akumulátory na poloviční kapacitě.

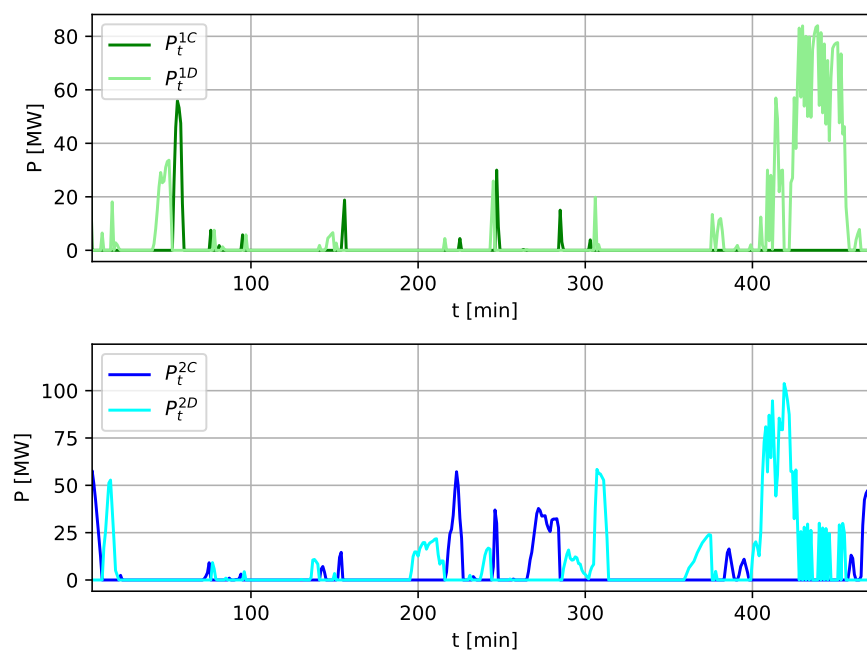


Obrázek 11.2: Průběh požadavků služby $aFRR_t^a$ a $mFRR_t^a$ pro agregační blok ČEZ

Online algoritmus s LSTM prediktorem

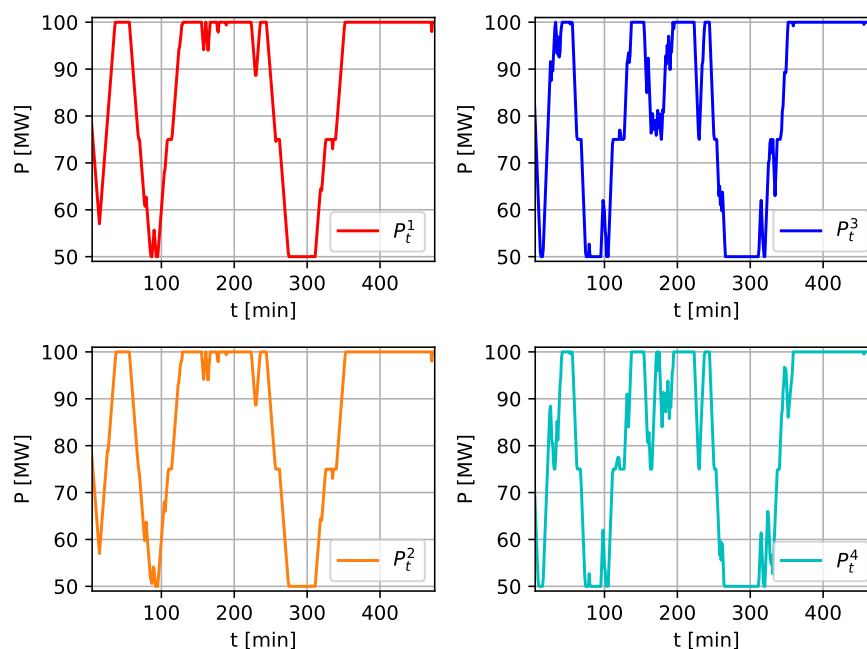


Obrázek 11.3: Rozložení výkonů pro jednotlivé výrobní jednotky nalezené pomocí online algoritmu s LSTM prediktorem

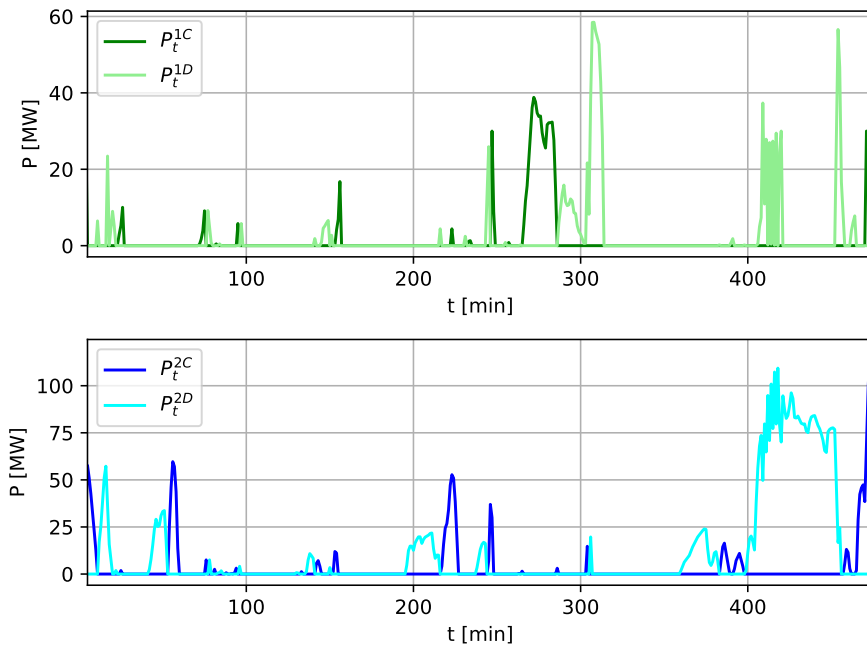


Obrázek 11.4: Rozložení výkonů pro jednotlivé akumulátory nalezené pomocí online algoritmu s LSTM prediktorem

■ Online algoritmus s ARIMA prediktorem

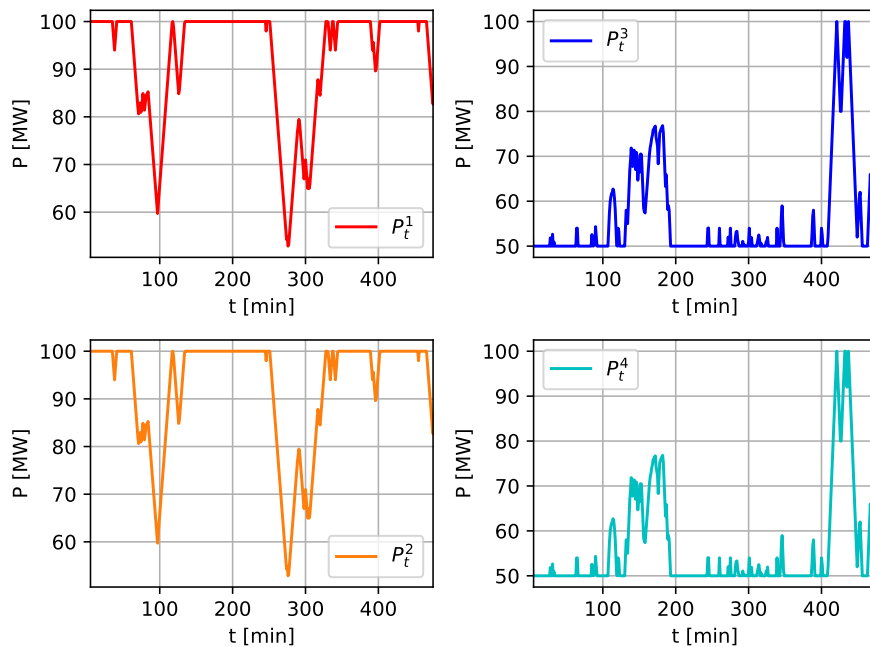


Obrázek 11.5: Rozložení výkonů pro jednotlivé výrobní jednotky nalezené pomocí online algoritmu s ARIMA prediktorem

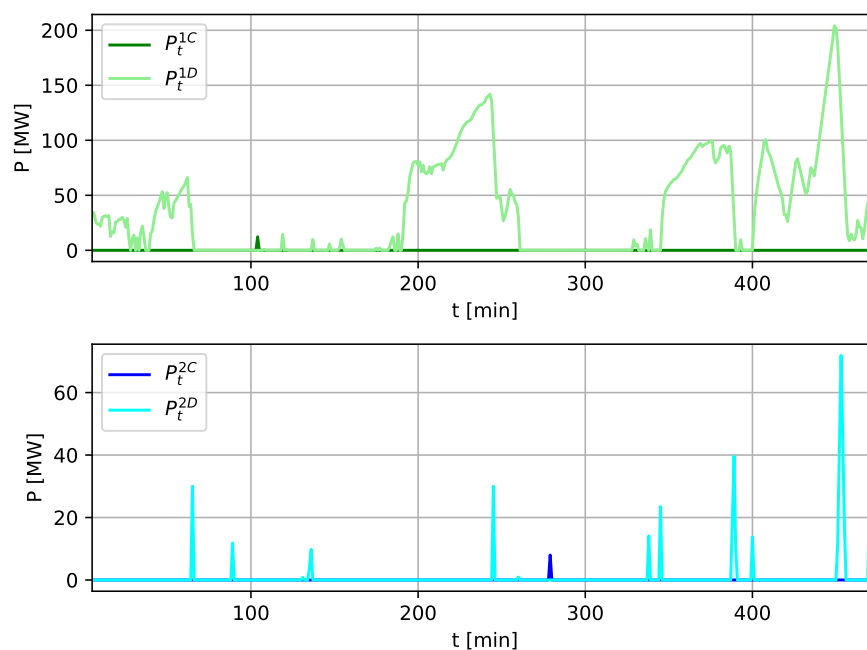


Obrázek 11.6: Rozložení výkonů pro jednotlivé akumulátory nalezené pomocí online algoritmu s ARIMA prediktorem

Offline algoritmus



Obrázek 11.7: Optimální rozložení výkonů pro jednotlivé výrobní jednotky agregačního bloku nalezené pomocí zpětné optimalizace offline algoritmem



Obrázek 11.8: Optimální rozložení výkonů pro jednotlivé akumulátory agregačního bloku nalezené pomocí zpětné optimalizace offline algoritmem

Algoritmus	Prediktor	Náklady	Zdražení vůči optimu
Offline	-	57 840.965 Kč	-
Online	LSTM	61 902.252 Kč	7.02 %
Online	ARIMA	61 946.679 Kč	7.10%

Tabulka 11.3: Náklady na provoz

Výsledné náklady jsou v případě řízení pomocí online algoritmu vyšší oproti optimálním. Zde se jedná o malé navýšení, které představuje zhruba 7 % optimální ceny, což můžeme v tomto případě považovat za dobrý výsledek. V rámci online algoritmu jsme dosáhli lepších výsledků pro verzi s LSTM prediktorem. Mezi online algoritmem s ARIMA prediktorem a online algoritmem s LSTM prediktorem však jde jen o malý rozdíl v celkových nákladech. Z průběhů výkonových rozložení vidíme, že online algoritmus se snaží udržovat akumulátory na poloviční kapacitě a tak dochází v průběhu testování i k jejich nabíjení. Oproti tomu v případě offline algoritmu dochází pouze k vybíjení.

Na modelovém příkladu jasně vidíme že optimalizace pomocí online algoritmu vychází hůře oproti zpětné optimalizaci pomocí offline algoritmu. Pokud by tomu tak nebylo znamenalo by to, že jeden z algoritmů je špatně. Online algoritmus si musí poradit s nejistotami, které nám do reálného průběhu vnáší podpůrné služby. Navíc má online algoritmus omezený rozsah pouze na šířku časového okna a musí tak efektivně využívat energii uloženou v akumulátorech.

i	\bar{P}_i	\underline{P}_i	T_i	a_i	b_i	C_i^a	C_i^d
1	100	30	2.5	75	50	∞	∞
2	100	30	5	150	100	∞	∞
3	100	30	7.5	225	150	∞	∞
4	100	30	10	300	200	∞	∞

Tabulka 11.4: Parametry výrobních jednotek při dynamickém testování

i	$\overline{\text{SOC}}_i$	$\underline{\text{SOC}}_i$	\bar{P}_i^c	\underline{P}_i^c	\bar{P}_i^d	\underline{P}_i^d	T_i^c	T_i^d	a_i^c	b_i^c	a_i^d	b_i^d
1	5000	0	150	0	150	0	10	10	600	300	50	25
2	5000	0	150	0	150	0	10	10	600	300	50	25
3	5000	0	150	0	150	0	10	10	600	300	50	25
4	10000	0	150	0	150	0	15	15	600	300	50	25

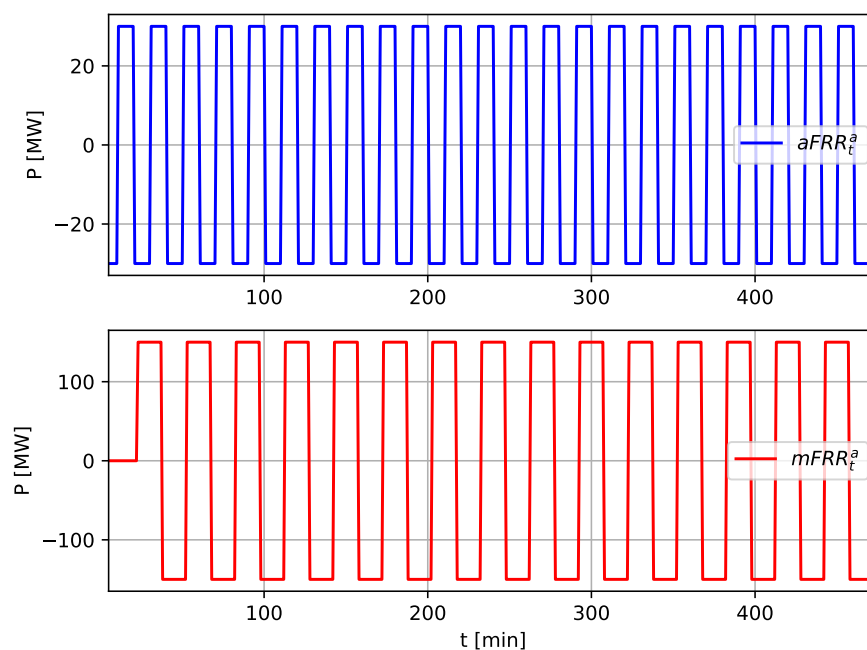
Tabulka 11.5: Parametry akumulátorů při dynamickém testování

11.3 Testování dynamických vlastností

V této části se zaměříme na testování modelu v případě velké dynamiky požadavků na aktivaci podpůrných služeb. Zde již dojde i k testování našeho online modelu. Budeme testovat, jak si online model dokáže poradit s rychle se měnícími požadavky $aFRR_t^a$ a $mFRR_t^a$. Dynamiku požadavků jsme volili tak aby docházelo ke střídavým aktivacím maximálních kladných a maximálních záporných rezerv pro služby aFRR a mFRR.

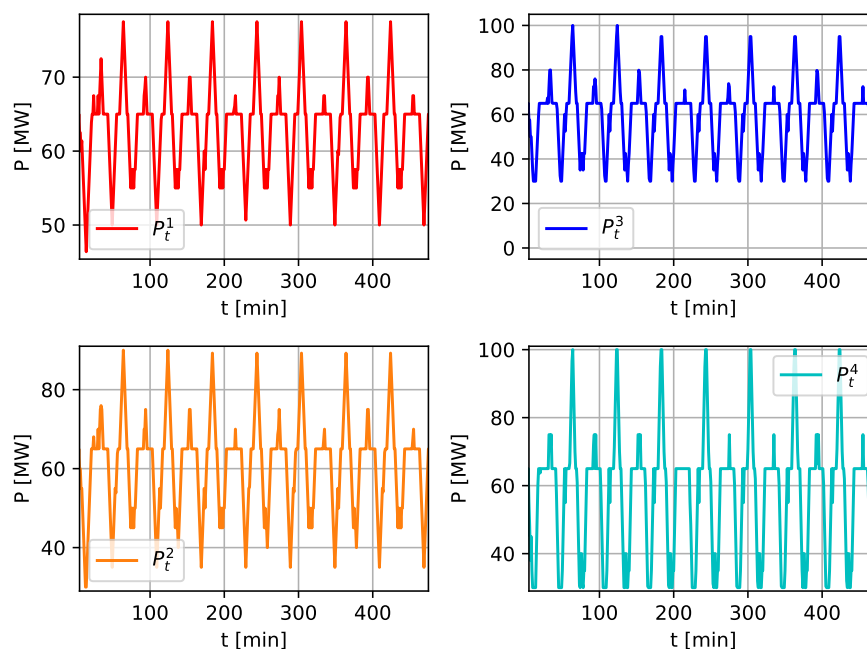
Náš blok jsme sestavili ze 4 výrobních jednotek a 4 akumulátorů. Parametry výrobních jednotek a akumulátorů jsou uvedeny v tabulkách 11.4 a 11.5. Průběh požadavků $aFRR_t^a$ a $mFRR_t^a$ je zobrazen na obrázku 11.9. Hodnota požadavku na diagramový výkon bude konstantní $P_t^{DG} = 300 \text{ MW}$, $\forall t \in \Omega^T$. Délku testovacího intervalu Ω^T jsme volili 6 hodin. Testování jsme provedli pro online algoritmus s predikcí požadavků realizovanou LSTM prediktorem a algoritmus s predikcí realizovanou ARIMA prediktorem. Následně jsme našli optimální výkonová rozložení pomocí zpětné optimalizace offline algoritmem.

Testování provádíme pro obě verze online algoritmu, tedy s LSTM prediktorem a ARIMA prediktorem. Následně pomocí offline algoritmu nalezneme optimální výkonové rozložení odpovídající daným výkonovým požadavkům. Celkové náklady na provoz, při řízení agregáčního bloku pomocí online algoritmů s různými prediktory a porovnání s náklady při optimálním rozložení, jsou uvedeny v tabulce 11.6. Na obrázcích 11.10 až 11.15. jsou zobrazeny průběhy výkonů jednotek tvořící agregáční blok, nalezené pomocí jednotlivých algoritmů. Z grafů vidíme, jak jednotlivé algoritmy využívají výrobní jednotky ke splnění požadavků a jak hospodaří s energií uloženou v akumulátorech. Velký rozdíl v průběhu výkonových rozložení je patrný mezi online algoritmy a offline algoritmem, který využívá zpětné optimalizace a dokáže tak efektivně využít energii uloženou v akumulátorech.

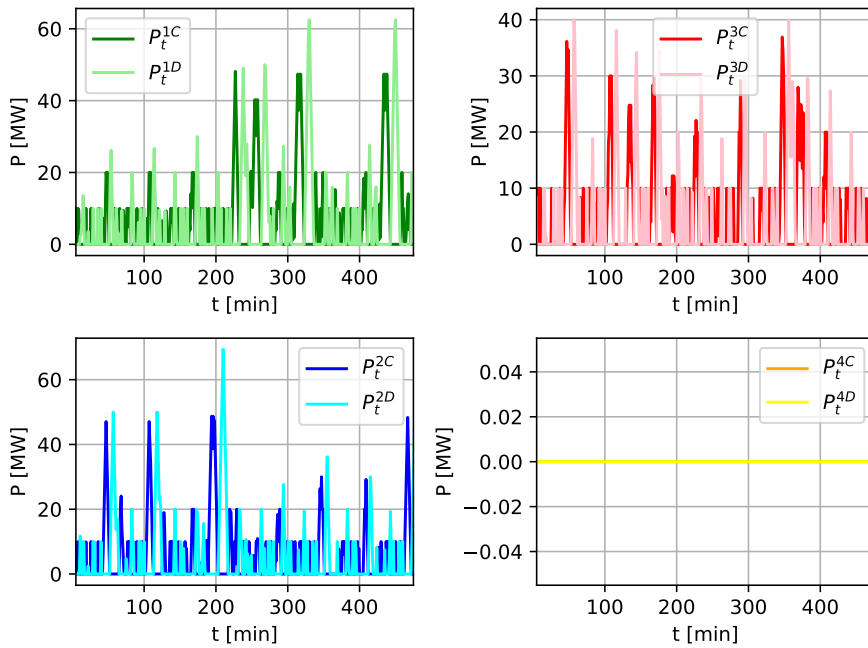


Obrázek 11.9: Průběh požadavků služby $aFRR_t^a$ a $mFRR_t^a$

Online algoritmus s LSTM prediktorem

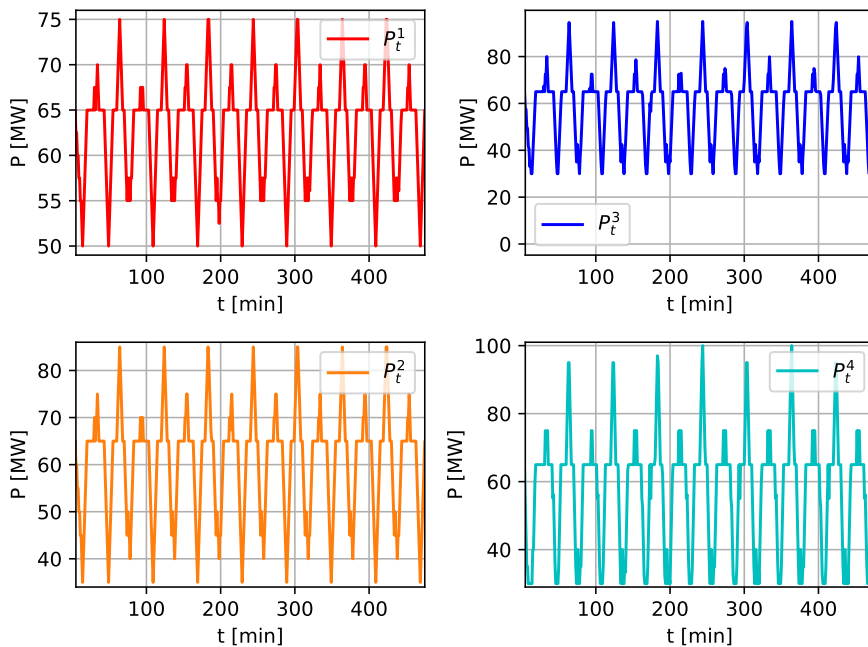


Obrázek 11.10: Rozložení výkonů na výrobní jednotky agregčního bloku při jeho řízení pomocí online algoritmu s LSTM prediktorem

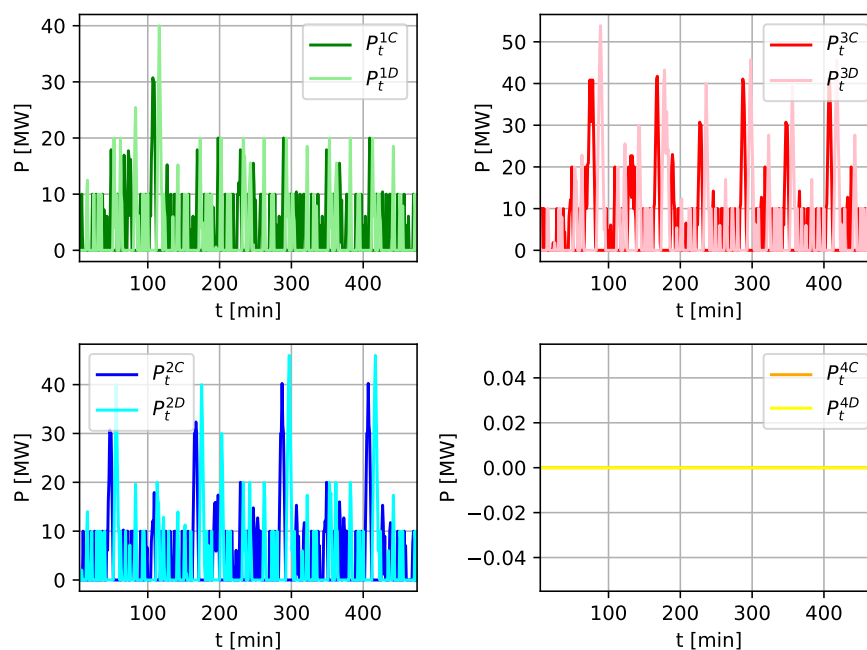


Obrázek 11.11: Rozložení výkonů na akumulátory agregičního bloku při jeho řízení pomocí online algoritmu s LSTM prediktorem

■ Online algoritmus s ARIMA prediktorem

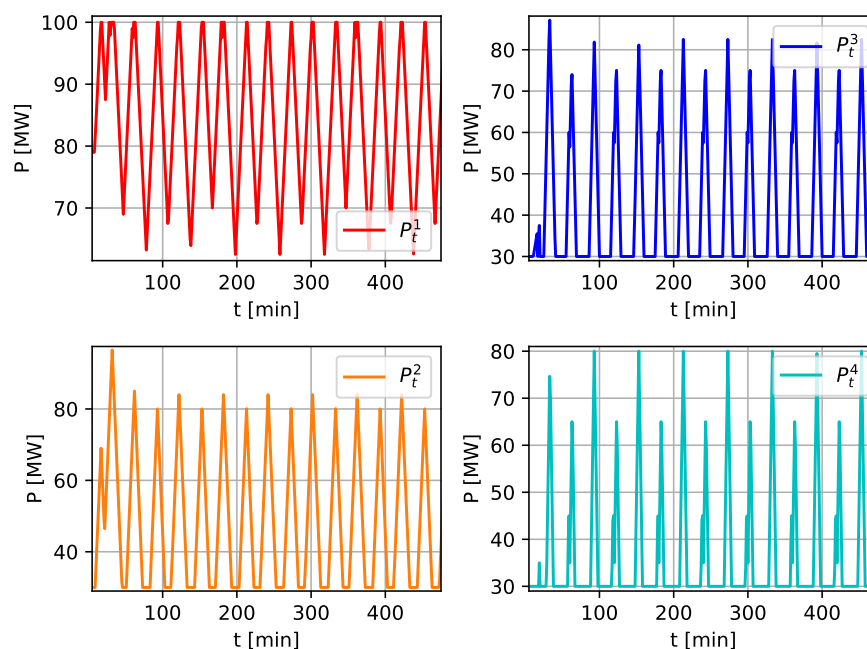


Obrázek 11.12: Rozložení výkonů na výrobní jednotky agregičního bloku při jeho řízení pomocí online algoritmu s ARIMA prediktorem

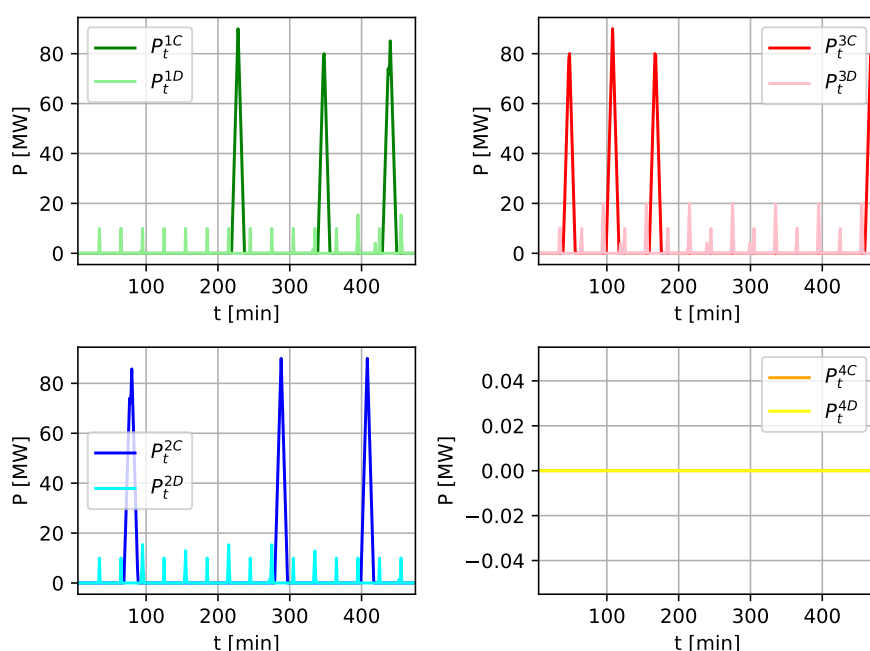


Obrázek 11.13: Rozložení výkonů na akumulátory agregačního bloku při jeho řízení pomocí online algoritmu s ARIMA prediktorem

Offline algoritmus



Obrázek 11.14: Optimální rozložení výkonů na výrobní jednotky agregačního bloku nalezené pomocí zpětné optimalizace offline algoritmem



Obrázek 11.15: Optimální rozložení výkonů na akumulátory agregčního bloku nalezené pomocí zpětné optimalizace offline algoritmem

Algoritmus	Prediktor	Náklady	Zdražení vůči optimu
Offline	-	201 417.654 Kč	-
Online	LSTM	310 318.852 Kč	54.33 %
Online	ARIMA	309 454.733 Kč	53.63 %

Tabulka 11.6: Náklady na provoz

Z tabulky 11.6. vyplývá, že pro rychlé změny požadavků $aFRR_t^a$ dostáváme lepší výsledky z hlediska ceny provozu pro algoritmus s ARIMA prediktorem. Toto může způsobeno tím, že během trénování neuronové sítě se takovýto průběh nevyskytoval a síť s ním tak neumí pracovat. Z obrázků je patrné, že offline algoritmus našel velmi odlišné rozložení oproti online algoritmům. Offline algoritmus má k dispozici celkový průběh $aFRR_t^a$ a $mFRR_t^a \forall t \in \Omega^T$ a tak dokáže efektivně využít akumulátory. Online algoritmus je limitován na šířku posuvného okna $\Omega^{\hat{T}}$ a v rámci naší heuristiky definované v kapitole 8 se snaží udržovat akumulátory na polovičním stavu nabití, což offline algoritmus nedělá.

Cílem tohoto experimentu bylo ukázat, že online algoritmus je schopen nalézt přípustné řešení. Tedy, že dokáže fungovat i v limitních podmínkách, které představují střídající se polarity aktivace plných regulačních záloh. V tomto případě nem selhávají prediktory, jelikož v rámci trénovacích dat, jsme se s takovou situací nesetkali. Naše heuristika si s tím ale dokáže poradit.

Název	Maximální výkon	Počet
Vranov 1	6.3 MW	3x
Vranov 2	1.125 MW	1x
Vír 1	6.44 MW	1x
Vír 2	1.15 MW	1x
Soběnov	615 kW	2x
Kogenerační jednotka	730 kW	27x

Tabulka 11.7: Výrobní jednotky v agregačním bloku E.ON

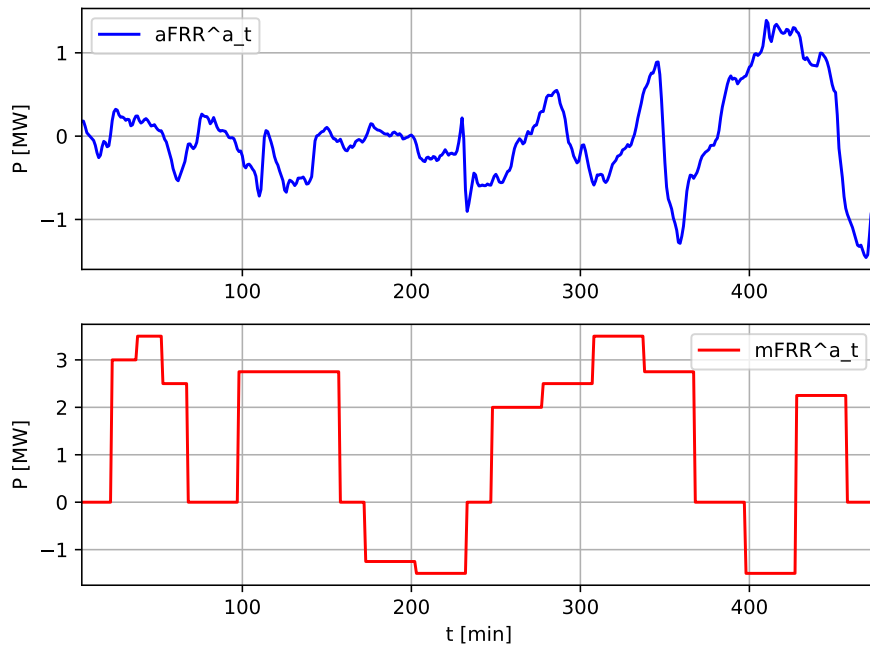
11.4 Testování na reálných datech

V minulých sekcích jsme testovali výpočetní náročnost offline algoritmu a následně jsme vyhodnocovali efektivitu online algoritmu na modelovém příkladu, kde jsme porovnávali jeho efektivitu vůči optimálnímu průběhu a následně jsme testovali jeho robustnost při velmi dynamických požadavcích $aFRR_t^a$ a $mFRR_t^a$. Algoritmus se ukázal být efektivní a robustní. Nyní otestujeme jeho efektivitu na datech z reálného provozu. Data z reálného provozu nám byla poskytnuta firmou OSC a.s. K dispozici máme technické parametry 2 agregačních bloků. Prvním agregačním blokem je skutečný agregační blok skupiny E.ON. Druhým agregačním blokem bude sestava elektráren skupiny ČEZ, kde se pokusíme agregační blok vytvořit a na porovnat zhodnotit jeho případnou efektivitu. Data, která nám byla pro testování poskytnuta zde nezveřejňujeme, jelikož se jedná o soukromý majetek společností ČEZ a E.ON. Uvedeme si zde jen o jaké typy zařízení jde a výsledné průběhy spolu s celkovými náklady na provoz agregačních bloků.

11.4.1 Agregační blok E.ON

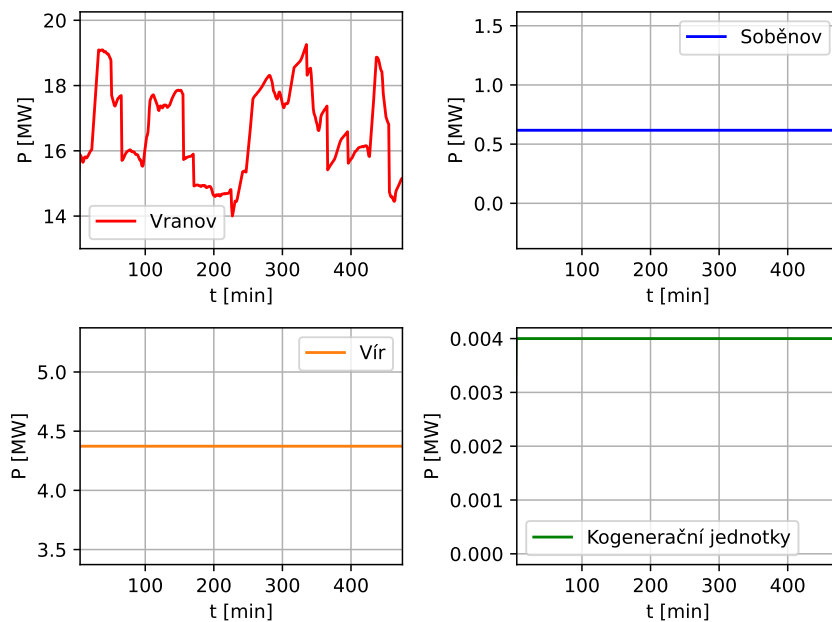
Prvním agregačním blokem, který budeme testovat je agregační blok skupiny E.ON. Tento agregační blok se skládá z 35 výrobních jednotek uvedených v tabulce 11.7. Tento agregační blok neobsahuje akumulátory. Výrobní jednotky zde představují bloky vodních elektráren Vranov, Vír a Soběnov a potom 27 kogeneračních jednotek. Agregační blok zde vytváří jednu distribuovanou výrobní jednotku o celkovém maximálním výkonu 48 MW. Celkový maximální výkon je tedy menší než maximální výkon jednoho bloku uhelné elektrárny.

Blok budeme testovat na reálných požadavcích P_t^{DG} , $aFRR_t^a$ a $mFRR_t^a$ kde $\forall t \in \Omega^T$. Velikost Ω^T je 6 hodin. Velikost $P_t^{DG} = 21MW \forall t \in \Omega^T$. Průběhy $aFRR_t^a$ a $mFRR_t^a$ jsou zobrazeny na obrázku 11.16. Testování jsme prováděli pro online algoritmus s LSTM prediktorem a online algoritmus s ARIMA prediktorem. Následně jsem nalezená rozložení výkonu porovnali s optimálním rozložením výkonu nalezené pomocí offline algoritmu. Průběhy rozdělení výkonů pro jednotlivé algoritmy jsou uvedeny na obrázcích 11.17 až 11.19. Celkové náklady na provoz jsou uvedeny v tabulce 11.8. Pro přehlednost v grafech uvádíme jen sumaci výkonových rozložení v rámci elektráren.



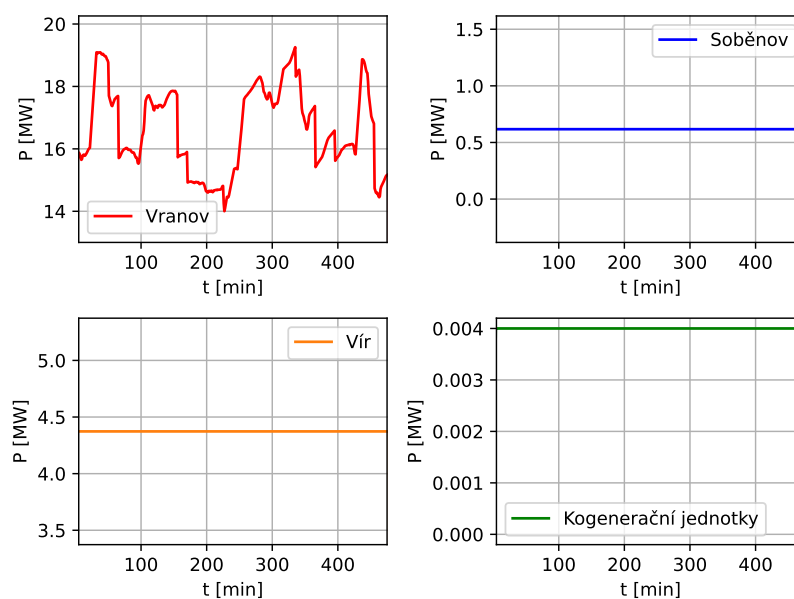
Obrázek 11.16: Průběh požadavků služby $aFRR_t^a$ a $mFRR_t^a$ pro agregační blok E.ON

■ Online algoritmus s LSTM prediktorem



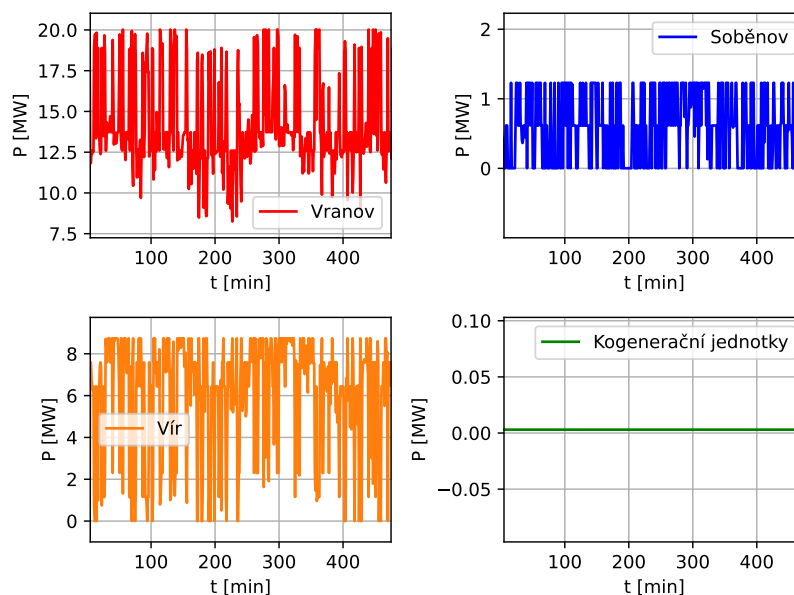
Obrázek 11.17: Rozložení výkonů pro jednotlivé vodní elektrárny a kogenerační jednotky agregačního bloku E.ON řízeného online algoritmem s LSTM prediktorem

Online algoritmus s ARIMA prediktorem



Obrázek 11.18: Rozložení výkonů pro jednotlivé vodní elektrárny a kogenerační jednotky agregačního bloku E.ON řízeného online algoritmem s ARIMA prediktorem

Offline algoritmus



Obrázek 11.19: Optimální rozložení výkonů pro jednotlivé vodní elektrárny a kogenerační jednotky agregačního bloku E.ON nalezené offline algoritmu

Algoritmus	Prediktor	Náklady	Zdražení vůči optimu
Offline	-	4765.350 Kč	-
Online	LSTM	4902.918 Kč	2.88 %
Online	ARIMA	4902.921 Kč	2.89 %

Tabulka 11.8: Náklady na provoz

Z obrázků můžeme vidět, že průběh optimálního rozložení a průběh nalezený pomocí online algoritmu se od sebe velmi liší. Nalezené optimální řešení vykazuje velkou dynamiku. Tento poznatek bude potřeba dále uvážit a zavést úpravy offline algoritmu, v podobě dalších kritérií zohledňující dynamiku jednotek. Výsledné náklady se liší minimálně. Online algoritmus nám tak přináší velmi uspokojivé výsledky, jedinou nevýhodou zde představuje, že oba algoritmy využívají k pokrytí požadavků primárně levnější vodní elektrárny z důvodu jejich levného provozu. Tyto elektrárny ovšem bývají primárně využívány pro regulaci špičkových výkonů a na pokrytí diagramového výkonu se zpravidla nepoužívají.

Výsledky jasně ukazují, že v budoucnu bude potřeba zavést separátní model pro vodní elektrárny. Tento model musí brát do úvahy vodohospodářskou situaci a efektivně hospodařit s vodou v přehradě. Současný model, ale funguje správně jelikož upřednostňuje výrobu energie levnější formou, kterou představuje vodní elektrárna oproti dražším kogeneračním jednotkám. Z tohoto důvodu jsme dosáhli tak nízkých nákladů na provoz agregačního bloku.

Název	Maximální výkon	Počet
Mělník A	110 MW	2x
Mělník B	60 MW	4x
Počerady	210 MW	4x
Ledvice A	110 MW	1x
Ledvice B	660 MW	1x
Orlík	91 MW	4x
Štěchovice	11.25 MW	2x
Diesellový generátor	2 MW	20x

Tabulka 11.9: Výrobní jednotky v agregačním bloku ČEZ

Název	Maximální nabíjecí výkon	Maximální vybíjecí výkon	Počet
Dlouhé stráně	325 MW	315 MW	2x
Dalešice	120 MW	120 MW	4x
Baterie	4 MW	4 MW	20x

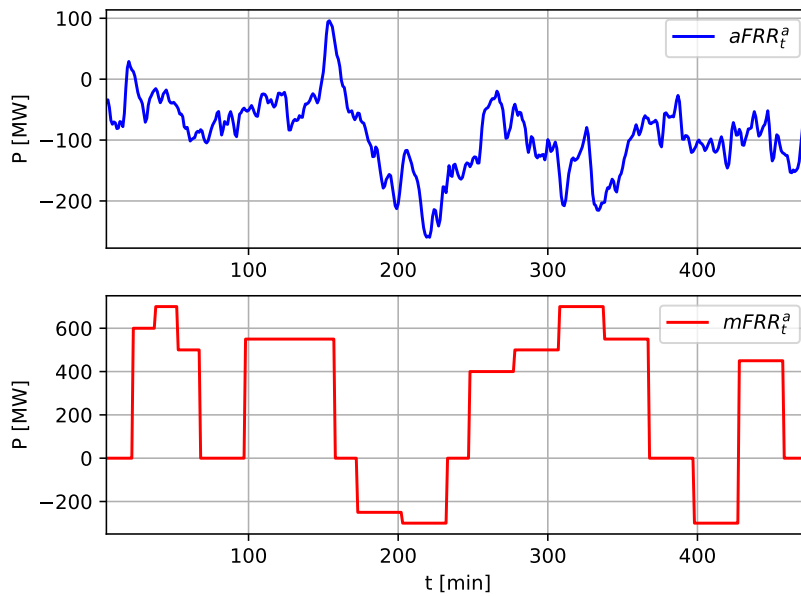
Tabulka 11.10: Akumulátory v agregačním bloku ČEZ

11.4.2 Agregační blok ČEZ

Druhým agregačním blokem bude sestava vybraných elektráren, kogeneračních jednotek a bateriových stanic ČEZ. Ve skutečnosti se nejedná o reálný agregační blok, ale v rámci této diplomové práce, jsme se rozhodli zhodnotit efektivitu spojení daných jednotek do jednoho agregačního bloku. Výrobní jednotky (resp. akumulátory) zařazené do agregačního bloku ČEZ jsou v tabulce 11.9 (resp v tabulce 11.10).

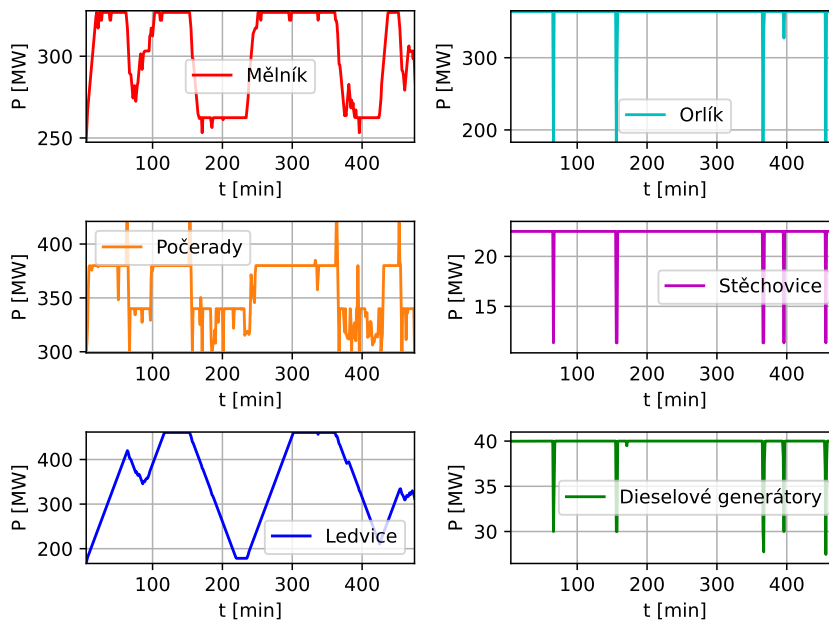
Agregační blok ČEZ tak celkem obsahuje 38 výrobních jednotek a 26 akumulátorů což je dohromady 64 jednotek. Výrobní jednotky představují bloky elektráren a diesellové generátory ČEZ. Akumulátory jsou pak přečerpávací elektrárny a bateriové stanice. Do agregačního bloku jsou zařazené elektrárny Mělník, Ledvice, Počerady, Orlík, Štěchovice, Dlouhé stráně a Dalešice. Potom 20 malých diesellových generátorů a 20 baterií. Velikost testovacího intervalu Ω^T je 6 hodin. Hodnota $P_t^{DG} = 1.4 \text{ GW} \forall t \in \Omega^T$. Průběh požadavků $aFRR_t^a$ a $mFRR_t^a$ je zobrazen na obrázku 11.20.

Testování jsme opět prováděli pro online model s LSTM a ARIMA prediktorem. Nalezená výkonová rozložení jsme porovnali s optimálním výkonovým rozložením nalezené pomocí zpětné optimalizace pomocí offline algoritmu. Porovnání nákladů na provoz je uvedeno v tabulce 11.11. Průběh rozložení výkonů pro jednotlivé algoritmy na jednotlivé jednotky agregačního bloku jsou na obrázcích 11.21 až 11.26. Pro přehlednost je uvádíme jako sumaci v rámci elektráren.

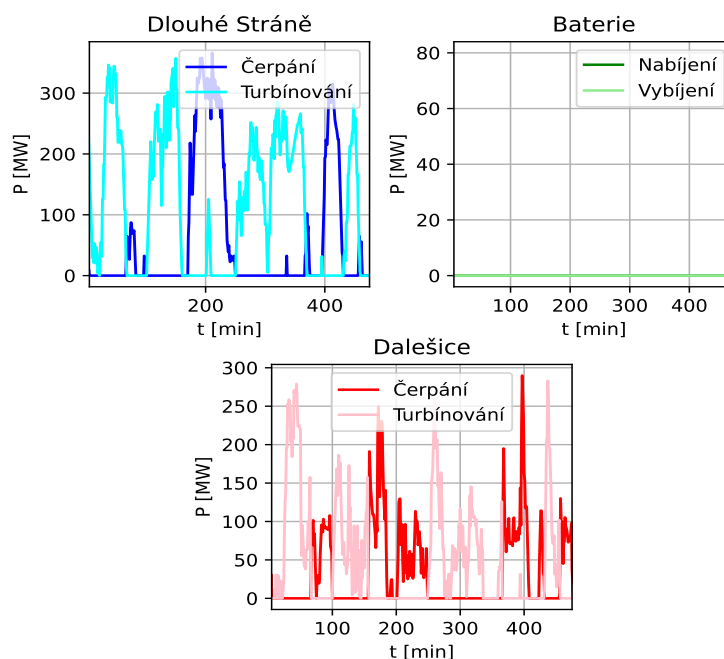


Obrázek 11.20: Průběh požadavků služby $aFRR_t^a$ a $mFRR_t^a$ pro agregační blok ČEZ

■ Online algoritmus s LSTM prediktorem

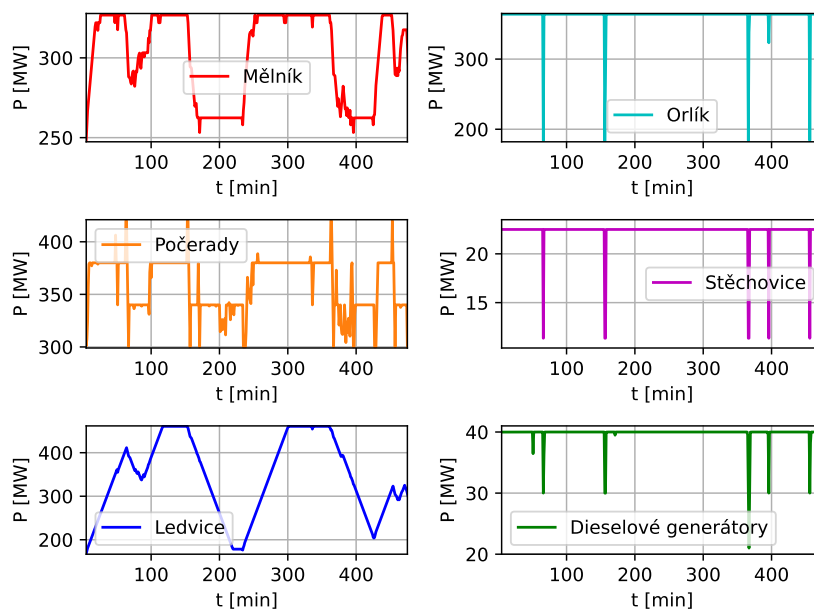


Obrázek 11.21: Rozložení výkonů pro jednotlivé elektrárny agregačního bloku ČEZ při jeho řízení pomocí online algoritmu s LSTM prediktorem

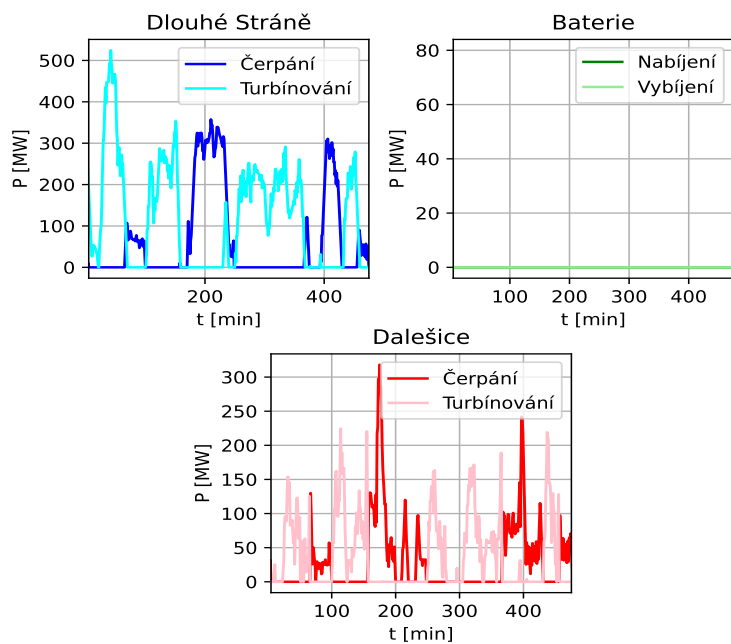


Obrázek 11.22: Rozložení výkonů pro jednotlivé přečerpávací vodní elektrárny a baterie agregačního bloku ČEZ s LSTM prediktorem

Online algoritmus s ARIMA prediktorem

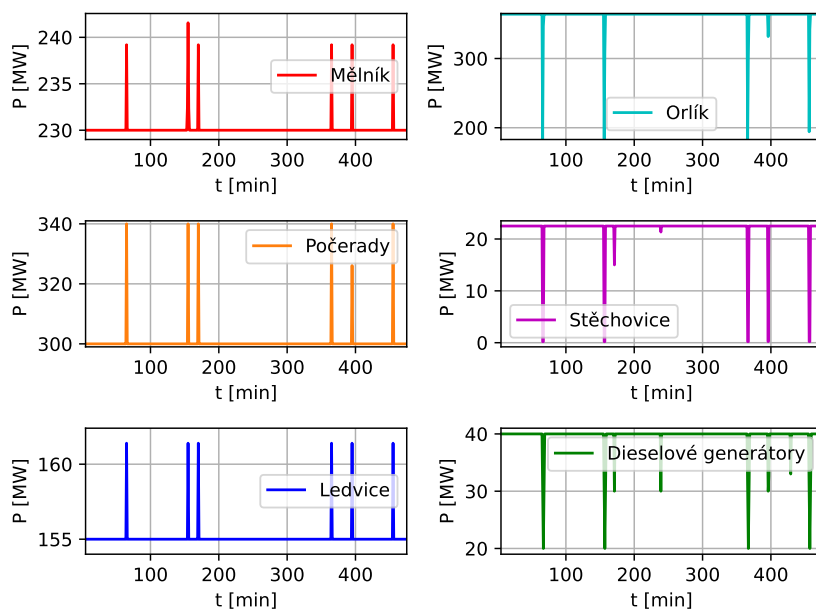


Obrázek 11.23: Rozložení výkonů pro jednotlivé elektrárny agregačního bloku ČEZ při jeho řízení pomocí online algoritmu s ARIMA prediktorem

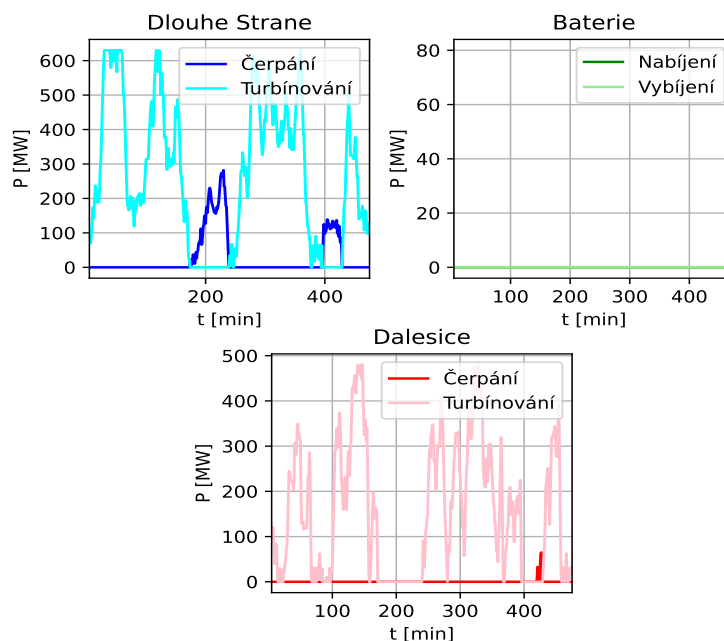


Obrázek 11.24: Rozložení výkonů pro jednotlivé přečerpávací vodní elektrárny a baterie agregačního bloku ČEZ s ARIMA prediktorem

Offline algoritmus



Obrázek 11.25: Optimální rozložení výkonů pro jednotlivé elektrárny agregačního bloku ČEZ nalezené pomocí zpětné optimalizace offline algoritmem



Obrázek 11.26: Optimální rozložení výkonů pro jednotlivé přečerpávací vodní elektrárny a baterie agregáčního bloku ČEZ nalezené pomocí zpětné optimalizace offline algoritmem

Algoritmus	Prediktor	Náklady	Zdražení vůči optimu
Offline	-	145 072 070.569 Kč	-
Online	LSTM	198 262 163.534 Kč	36.66 %
Online	ARIMA	198 464 964.341 Kč	36.80 %

Tabulka 11.11: Náklady na provoz

Z průběhů je patrné, že online algoritmus se opět snaží udržet akumulátory na poloviční kapacitě. Offline algoritmus v rámci nalezení optimálního rozložení tento problém řešit nemusí, jelikož zná požadavky $aFRR_t^a$ a $mFRR_t^a$, $\forall t \in \Omega^T$. V tomto případě vychází lépe online algoritmus s LSTM prediktorem, což se dalo dle předchozích kapitol očekávat. Rozdíl je však velmi malý. Náklady na provoz při použití online algoritmu nám tak vzrostou o necelých 37% vůči optimálním nákladům nalezené zpětnou optimalizací.

Stejně jako u agregáčního bloku E.ON, si zde povšimněme, že online algoritmy i offline algoritmus využívají plný výkon vodních elektráren Orlické přehradě a Štěchovicích z důvodu jejich levného provozu. Tato situace je pro reálný provoz již poměrně nebezpečná. Pro hodnoty výkonů generované elektrárnou Orlické přehradě by došlo k rapidnímu zvýšení hladiny řeky a rychlému úbytku vody v Orlické přehradě. Do budoucna je tedy třeba navrhnout separátní model pro vodní elektrárnu.

11.5 Diskuze nad výsledky experimentů

Cílem této sekce je provést celkové zhodnocení našich experimentů. V rámci experimentů jsme vyhodnocovali výpočetní náročnost našeho algoritmu z hlediska velikosti instance. Následně bylo testováno chování algoritmu v limitním případě, kde bylo cílem ověřit, že online algoritmus je schopen limitním požadavkům v reálném čase vyhovět a následně jsme algoritmus testovali na generovaných datech simulující reálný provoz a na datech z reálného provozu.

Výpočetní náročnost

V sekci zaměřující se na výpočetní náročnost algoritmu, jsme dle experimentů zjistili, že výpočetní náročnost exponenciálně narůstá se zvětšující se instancí. Toto odpovídá NP obtížné úloze, kterou představuje MILP model. Online algoritmus je tak vhodný pro řízení v reálném čase pro velikosti posuvného okna $|\Omega^T| \leq 20$. Nemá cenu uvažovat predikce na více než 20 kroků dopředu. Pro tuto velikost časového okna je online algoritmus schopen nalézt výsledné rozložení pro velikost agregačního bloku $|\Omega^{AB}| \leq 120$ v čase $t \leq 0.5s$. Budeme-li uvažovat výpočetní a komunikační procesy, které je potřeba během reálného provozu udělat, je tento čas přijatelný.

Limitní případ

Cílem tohoto experimentu, bylo otestovat online algoritmus, jak si dokáže poradit v limitním případě. Online algoritmus dokázal nalézt přípustné řešení. V tomto experimentu jsme zaznamenali největší odchylku v celkových provozních nákladech. Tato odchylka byla způsobena heuristikami, zajišťujícími efektivní využití energie uložené v akumulátorech v reálném čase. Rovněž zde byl problém v případě prediktorů, jelikož limitní požadavky byly odlišné od trénovacích dat. Povaha vstupních signálů tak byla jiná než povaha signálů pro které byly prediktory navrženy. Spolehlivost řešení tak závisela na heuristikách definovaných v kapitole 8. Tyto heuristiky si ale s těmito limitními případy dokázaly poradit a online algoritmus tak v tomto testování obstál, byť s vyššími cenovými náklady oproti optimu.

Reálný provoz

V případě reálného provozu jsme zaznamenali zajímavé výsledky. V první řadě je důležité poznamenat, že efektivita online algoritmu závisela na podílu akumulátorů na celkových dodávkách výkonu. Využití energie uložené v akumulátorech vychází levněji než její výroba ve výrobních jednotkách (pokud neuvažujeme vodní elektrárnu, kde je to téměř stejně drahé). Toto způsobuje, že offline algoritmus, který provádí zpětnou optimalizaci na daný časový interval Ω^T se znalostí všech výkonových požadavků, má tendenci

tuto energii vhodně využít, ale již nedoplňuje akumulátor na původní kapacitu.

Oproti offline algoritmu má online algoritmus navíc heuristiky, které se snaží udržovat akumulátory na poloviční kapacitě a to způsobuje v případě využití energie její zpětné doplnění ve vhodných případech. Nabíjení akumulátoru je ovšem mnohdy dražší než provoz výrobní jednotky a tak online algoritmus vykazuje vyšší provozní náklady.

Dále je potřeba poznamenat, že v případě provozu na reálných systémech docházelo k velkému využívání energie z vodních elektráren. Z hlediska matematické optimalizace je toto korektní krok, jelikož výroba energie ve vodní elektrárně je podstatně levnější než výroba energie v elektrárně tepelné. V případě reálného provozu by ovšem došlo k rapidnímu snížení hladiny v nádrži a porušení všech vodohospodářských pravidel.

Naše úloha byla zaměřena na krátký časový interval a cílem bylo primárně navrhnout algoritmus. V další práci je ovšem třeba tuto skutečnost brát v úvahu a modelovat každý druh elektrárny zvlášť. V případě vodní elektrárny bude potřeba vytvořit model, který bude zohledňovat stav nádrže a vodohospodářská pravidla. Tím pádem dojde i k úpravě kriteriální funkce online algoritmu a tím pádem ke změně naší heuristiky. Rovněž bude potřeba zavést lepší model pro velké výrobní jednotky (tepelné a jaderné elektrárny), kde nemůžeme uvažovat, že jejich aktivace a deaktivace proběhne během jedné minuty jako je tomu u menších zdrojů. Tyto aktivace a deaktivace probíhají v řádu hodin a o jejich provedeních nemůžeme rozhodovat v reálném čase. Do budoucna tak bude potřeba navrhnout algoritmus, který bude provádět první fázi optimalizace, ve které dojde k rozvrhování těchto velkých výrobních jednotek na jednotlivé obchodní intervaly.

■ Závěrečné shrnutí experimentů

Výsledky experimentů ukazují, že online algoritmus je schopen řídit distribuovanou soustavu jednotek tvořící agregáčn \acute{y} blok. Všechny experimenty nám poskytly přípustná řešení. Online algoritmus vykazoval dle očekávání vyšší cenu výrobních nákladů vůči optimálnímu provozu získaného pomocí zpětné optimalizace offline algoritmem. Z hlediska výpočetní náročnosti je možné algoritmus využít pro řízení v reálném provozu. Heuristiky zavedené v kapitole 8 se ukázaly jako validní v případě testování limitních požadavků.

Experimenty ovšem ukázaly, že některé věci je potřeba vylepšit. Do budoucna bude potřeba přidat kritéria zohledňující dynamiku výrobních jednotek a akumulátorů. Dále je potřeba diverzifikovat současné modely výrobních jednotek a akumulátorů, tak aby lépe odpovídaly fyzikální realitě a navrhnout algoritmus, který bude řešit první fázi optimalizace, která bude spočívat v rozvrhování jednotek na jednotlivé obchodní intervaly. Jinak považujeme výsledky experimentů za uspokojivé.

Kapitola 12

Závěr

12.1 Shrnutí

Cílem této diplomové práce bylo popsat nová pravidla pro poskytování základních služeb v dodávání elektrické energie do sítě a navrhnout řízení vybraných zdrojů energie pro jejich poskytování. Kapitoly 2, 3 a 4 se věnovaly popisem pravidel trhu s energiemi, systémovým popisem vybraných zdrojů elektrické energie a popisu služeb výkonové rovnováhy. Na základě informací obsažených v těchto kapitolách jsme definovali náš optimalizační problém, který je popsán v kapitole 6. Kapitola 7 zavádí MILP model agregačního bloku obsahující soustavu vybraných jednotek a dokáže pro tuto soustavu najít optimální rozložení výkonů pro dané výkonové požadavky. V následující kapitole jsme zavedli online model, který řídí agregační blok v reálném čase a popsali jsme si problémy, které nám přináší optimalizace v reálném čase z hlediska neurčitosti průběhu požadavků na podpůrné služby. Aby se online model s těmito neurčitostmi vypořádal, potřebuje ke svému fungování prediktor průběhu požadavků. Návrhem prediktorů jsme se zabývali v kapitole 9, kde jsme si představili prediktor využívající statistický ARIMA model a v kapitole 10 jsme navrhli prediktor využívající neuronové sítě.

Následně jsme testovali navržené algoritmy na reálných datech. V rámci testování jsme využívali offline algoritmus, který nám stanovil optimální rozložení a plnil tak funkci reference. Tato rozložení jsme pak porovnávali s rozloženími nalezenými pomocí online algoritmů využívající prediktory. Lepších výsledků jsme dosahovali pro online algoritmus s LSTM prediktorem, až na jeden limitní případ, kdy jsme vnutili modelu velkou dynamiku v podobě střídavé polaritě plné aktivace regulační energie rezervované pro podpůrné služby. Zde jsme dosáhli lepších výsledků pro ARIMA prediktor.

V rámci testování jsme pozorovali, že offline algoritmus často využívá ke splnění dodávek energii uloženou v akumulátorech a neuvažuje její další využití mimo testovací interval. Oproti tomu online algoritmus využívající přidáného kritéria se snaží udržovat akumulátory na poloviční kapacitě, což způsobuje vyšší náklady na provoz. Tato strategie se však jeví jako vhodná. Akumulátory mají velkou dynamiku a jsou tak schopny velmi rychle měnit

svůj dodávaný (resp. odebíraný) výkon do sítě (resp. ze sítě). Udržíme-li akumulátor na poloviční kapacitě, pak jej jsme schopni využít v případě velké změny výkonu, kterou bychom nebyli schopni pokrýt jen s využitím výrobní jednotky, která má dynamiku nižší.

12.2 Náměty na budoucí práci

Tato diplomová práce představuje úvodní studium dané problematiky. Popsané systémové modely jednotek jsou velmi obecné a bude potřeba je upravit, tak abychom se co nejvíce přiblížili fyzikální realitě. Model zároveň neuvažuje neříditelné obnovitelné zdroje jako jsou fotovoltaické panely, nebo větrné elektrárny. Zde bude ovšem potřeba vytvořit další prediktory, které budou predikovat jejich výrobu a na základě toho navrhnout zpětnovazební korekci jejich odchylek od plánu. Zároveň je třeba představit přesnější model vodních elektráren, kde jsme limitováni kapacitou vodní nádrže a také vylepšit model bateriových stanic, kde musíme uvážit postupná poškození baterií způsobená opakovaným nabíjením a vybíjením.

Z hlediska podpůrných služeb bude potřeba uvažovat i mFRR s přímou aktivací a sestavit pro ni prediktor tak jako pro službu aFRR. Dále bude potřeba vylepšit LSTM prediktor, jelikož současná verze uvažuje pouze předchozí hodnoty aFRR a neuvažuje korelace s jinými veličinami. Aktivace podpůrných služeb se odvíjí od systémového modelu přenosové soustavy. Pro lepší predikci bude vhodné soustavu identifikovat a následně vytvořit její systémový model a predikci realizovat pomocí Kalmánova filtru, který by v budoucích pracích mohl nahradit současný ARIMA prediktor. Rovněž je na místě zvážit optimalizaci rizika a povolit malé odchylky než udržovat výrobní jednotky na polovičním výkonu což je pro reálný provoz mnohdy neefektivní.

Současný model rovněž uvažuje aktivace jednotek, které proběhnou v rámci periody příchozích požadavků. Z hlediska dlouhodobé optimalizace bude potřeba uvažovat i aktivace a deaktivace velkých výrobních bloků, které probíhají i v řádech hodin. Rovněž bude potřeba uvážit jiná technická omezení jako je například minimální doba odstávky, nebo maximální doba nasazení výrobní jednotky. Tuto problematiku bychom mohli řešit pomocí algoritmu využívající 2 stupně optimalizace. Druhý stupeň by představovala optimalizace v reálném čase, která by vycházela z našeho online algoritmu. První fáze by se zabývala dlouhodobou optimalizací spočívající v plánování odstávek jednotlivých bloků a obchodní strategie pro objem diagramového výkonu a poskytování regulačních záloh. Těmito tématy se budeme zabývat v navazujícím výzkumu.

Příloha A

Literatura

- [1] *KODEX PŘENOSOVÉ SOUSTAVY*. ČEPS 2021
- [2] *KODEX PŘENOSOVÉ SOUSTAVY Část II. Podpůrné služby (PpS)*. ČEPS 2021
- [3] L. Baringo, M. Rahimiyan *Virtual Power Plants and Electricity Markets: Decision Making Under Uncertainty*. Springer Int. Pub., 2020
- [4] A. Zangeneh, M. Moeini-Aghtaie *Scheduling and Operation of Virtual Power Plants*. Elsevier - Health Sciences Division, 2022
- [5] F. Kazempour, A. Yazdani, R. Iravani *Virtual Power Plants: A Robust Control Structure*. LAP Lambert Academic Publishing, 2016
- [6] E. Nasrolahpour, J. Kazempour, H. Zareipour, W. D. Rosehart. *A bilevel model for participation of a storage system in energy and reserve markets*. IEEE Transactions on Sustainable Energy, 9(2):582–598, April 2018. doi: 10.1109/TSTE. 2017.2749434.
- [7] J. Wang, H. Zhong, Q. Xia, Z. Ma, Z. Wang, X. Wu. *Robust bidding strategy for microgrids in joint energy, reserve and regulation markets*. 2017 IEEE Power Energy Society General Meeting, pages 1–5, 2017.
- [8] Z. Zhang, J. Wang, T. Ding, X. Wang. *A two-layer model for microgrid real-time dispatch based on energy storage system charging/discharging hidden costs*. IEEE Transactions on Sustainable Energy, 8(1):33–42, Jan 2017.
- [9] Z. Saez-de-Ibarra, A. Milo, H. Gaztanaga, V. Debusschere, and S. Bacha. *Co-optimization of storage system sizing and control strategy for intelligent photo-voltaic power plants market integration*. IEEE Transactions on Sustainable Energy, 7(4):1749–1761, Oct 2016.
- [10] M. Chazarra, J. García-González, J. Pérez-Díaz, M- Arteseros. *Stochastic optimization model for the weekly scheduling of a hydropower system in day-ahead and secondary regulation reserve markets*. Electric Power Systems Research, 130:67–77, 01 2016.

- [69] A. Graves, S. Fernández, J. Schmidhuber. *Bidirectional LSTM networks for improved phoneme classification and recognition*. Artificial Neural Networks: Formal Models and Their Applications–ICANN 2005. Springer Berlin Heidelberg, 799-804, 2005
- [70] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*. 2014