



**MASARYKŮV ÚSTAV
VYŠŠÍCH STUDIÍ
ČVUT V PRAZE**

Diplomová práce

**Tvorba informačního systému pro záznam
a zpracování dat ze systému měření a
regulace mikro turbo-expandéru**

Bc. Radka Vopatová

Oddělení ekonomických studií

Vedoucí práce: Ing. Jiří Kaiser, Ph.D.

7. ledna 2021

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení:	Vopatová	Jméno:	Radka	Osobní číslo:	407139
Fakulta/ústav:	Masarykův ústav vyšších studií (MÚVS)				
Zadávající katedra/ústav:	Oddělení ekonomických studií				
Studijní program:	Projektové řízení inovací				
Studijní obor:	-				

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:
Tvorba informačního systému pro záznam a zpracování dat ze systému měření a regulace mikro turbo-expandéru

Název diplomové práce anglicky:
Implementation of an information system for data processing from the system of instrumentation and control of a micro turbo-expander

Pokyny pro vypracování:

CÍL PRÁCE: Diplomová práce si dává za cíl navrhnout a realizovat informační systém pro účely záznamu a zpracování dat ze systému měření a regulace (MaR) mikroturbíny.
PŘÍNOS PRÁCE: Přínosem diplomové práce je zefektivnění procesu vyhodnocení dat po měření a regulaci výkonu mikroturbíny a s tím i související pracovní výkonnost vývojových inženýrů.
OSNOVA: (1) Úvod; (2) Teoretická část - automatizace, měření výkonnosti pracovníků; (3) Praktická část - popis problému, současný stav řešení, návrh a realizace navrženého řešení; (4) Zhodnocení výsledků - posouzení přínosů a nákladů realizace řešení; (5) Závěr




Seznam doporučené literatury:

(1) SATZINGER, John W. Systems analysis and design in a changing world. Cengage Learning, 2015.
(2) WAGNER, Jaroslav. Měření výkonnosti: jak měřit, vyhodnocovat a využívat informace o podnikové výkonnosti. Grada, 2009.
(3) MILLER, Frederic P., et al., editoři. LabVIEW: Automation, Visual Programming Language, Macintosh, Instrument Control, Microsoft Windows, Unix, Data acquisition, Mac OS X. Dataflow Programming, 2010.

Jméno a pracoviště vedoucí(ho) diplomové práce:
Ing. Jiří Kaiser, Ph.D., ČVUT v Praze, Masarykův ústav vyšších studií

Jméno a pracoviště konzultanta(ky) diplomové práce:
Ing. Václav Novotný, ČVUT v Praze, Fakulta strojní

Datum zadání diplomové práce: 12. 12. 2019 Termín odevzdání diplomové práce: 7. 1. 2021
Platnost zadání diplomové práce: 30. 9. 2021

  
Podpis vedoucí(ho) práce Podpis vedoucí(ho) ústavu/katedry Podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

6. 1. 2021 Vopatová
Datum převzetí zadání Podpis studenta(ky)

Poděkování

Děkuji Ing. Jiřímu Kaiserovi, Ph.D. za vedení této práce, za jeho cenné rady a vstřícný přístup. Rovněž děkuji Ing. Václavu Novotnému za konzultace a jeho drahocenný čas. A v neposlední řadě děkuji rodině, přátelům, své drahé polovičce a všem, kteří mě podpořili při tvorbě této práce a i v samotném studiu.

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.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 2373 odst. 2 zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů, tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu) licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 7. ledna 2021

.....

České vysoké učení technické v Praze
Masarykův ústav vyšších studií

© 2021 Radka Vopatová. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Masarykův ústav vyšších studií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.

Odkaz na tuto práci

Vopatová, Radka. *Tvorba informačního systému pro záznam a zpracování dat ze systému měření a regulace mikro turbo-expandéru*. Diplomová práce. Praha: České vysoké učení technické v Praze, Masarykův ústav vyšších studií, 2021.

Abstrakt

Tato diplomová práce se zabývá návrhem a realizací informačního systému pro účely záznamu a zpracování dat ze systému měření a regulace mikroturbíny. Přínosem práce je zefektivnění procesu vyhodnocení dat pomocí navrženého a implementovaného informačního systému v prostředí MATLAB. Následně jsou vyhodnoceny výhody a nevýhody nového přístupu k řešení problému pomocí ekonomického zhodnocení investice do realizace informačního systému.

Klíčová slova zpracování dat, návrh informačního systém, automatizace, MATLAB, měření a regulace mikro turbo-expandéru, investice

Abstract

This diploma thesis deals with the design and implementation of an information system for the purpose of experimental data acquisition and processing from the instrumentation and control system of a micro turboexpander. The thesis significantly improves the effectivity and productivity of the data processing system designed and implemented in MATLAB. Finally, the advantages and drawbacks of the novel approach are discussed based on standard economic project evaluation methods and also by the non-economical approach.

Keywords data processing, information system design, automation, MATLAB, instrumentation and control of a micro turbo-expander, investment

Obsah

Úvod	1
1 Teoretická část	3
1.1 Automatizace	3
1.1.1 Rozdělení úkolů	4
1.1.2 Úroveň automatizace	5
1.1.3 Taxonomie	6
1.1.3.1 Komputerizace	6
1.1.4 Model	10
1.1.5 Framework	11
1.1.5.1 Primární hodnotící kritéria	13
1.1.5.2 Sekundární hodnotící kritéria	14
1.2 Měření pracovní výkonnosti	16
1.2.1 Produktivita ve výzkumné činnosti	16
1.3 Investice do vývoje informačního systému	18
1.4 Používané jazyky	19
1.4.1 UML	19
1.4.1.1 Use Case Diagram	20
1.4.1.2 Activity Diagram	23
1.4.1.3 Machine State Diagram	25
1.4.2 MATLAB®	26
2 Praktická část	27
2.1 Popis problému a současný stav řešení	27
2.2 Návrh řešení	31
2.2.1 Volba míry automatizace	31
2.2.1.1 Hodnocení kritérií pro daný návrh	34
2.2.2 Návrh systému	35
2.2.2.1 Nahrání souborů	37

2.2.2.2	Určení ustálených stavů	38
2.2.2.3	Vyhodnocení experimentálních dat	40
2.2.2.4	Reset	41
2.3	Realizace	44
2.3.1	Popis matematického modelu	46
2.3.2	Implementace	48
3	Ekonomické zhodnocení	54
3.1	Kapitálové výdaje a příjmy z investice	54
3.2	Diskuze výsledků ekonomického zhodnocení	57
3.3	Další neekonomické přínosy realizace informačního systému . .	59
	Závěr	60
	Literatura	62
A	Seznam použitých zkratk	71
B	Seznam použitých symbolů	73
C	Ukázka .txt souborů vzniklých během měření	74
D	Obsah přiloženého CD	77

Úvod

V dnešní době jsme svědky průběhu čtvrté průmyslové revoluce (Průmysl 4.0). Hromadná a čím dál tím více i sériová výroba již postupně přechází v plně digitalizovanou podobu, kdy od vstupního materiálu po koncový produkt je udrženo tzv. *digitální vlákno*, tzn. stroje si za pomoci manipulátorů předávají výrobek v různých fázích rozpracovanosti a průběžně si uchovávají informaci o jeho pozici, geometrii a proběhlých výrobních operacích. [1] Výrobní postupy pak mohou být konceptem *virtuální továrny* vizualizovány a simulovány paralelně s probíhající výrobou, kdy operátor pouze kontroluje a dohlíží na hladký chod výrobního procesu. Do jisté míry tedy je a nejspíše vždy bude lidský faktor stále přítomný, avšak nežádoucí, neboť je zdrojem chyb a nepřesností. [2] Průmysl bez automatizace dnes v podstatě již neexistuje, nebo jej lidstvo označuje jako řemeslo.

Při vývoji nových prototypů strojů, mechanismů a komponent, je však takovýto postup nemožné využít. Proto je také prototypová výroba natolik nákladná - její míra automatizace je nejnižší ze všech výrobních postupů. Čas strávený vývojem komponent je navíc čas velmi drahý, neboť vývojoví pracovníci jsou často jedni z nejvýše placených ve společnostech. Úspora času, potažmo nákladů, spojených s vývojem nových komponent a jejich prototypovou výrobou je tedy značnou úsporou v rozpočtu společnosti.

V případě výzkumných týmů, které se soustředí na aplikovaný výzkum a experimentální vývoj, který převádí poznatky ze základního výzkumu v koncepty využitelné v praxi, se bez výjimky z podstaty věci vždy jedná o stavbu prototypů a funkčních vzorků. Velké množství času při vývoji nových dílů je vynaloženo zejména a právě na experimentální testování. Při této činnosti je záznam a zpracování měřených experimentálních hodnot jednou z časově nejnáročnějších aktivit.

Za pomoci dnes již dostupných, ale ještě před deseti lety nepředstavitelných, technologií jako jsou aditivní výroba (3D tisk), velmi rychlé výpočetní technologie a 3D modelování, je tak při prototypové výrobě možné mluvit

jako o tzv. *rapid prototypingu*. [3] Aby takovýto prototyping byl ale vsutku „rapid“, je nutné, aby data z měření byla také zpracována automaticky a výzkumný pracovník měl zpětnou vazbu z experimentálního měření co nejdříve k dispozici, jinak dochází ke zbytečné časové prodlevě a zdržení celého procesu vývoje.

Hlavní myšlenkou za touto prací je tedy ulehčit jednotvárnou manuální práci vědeckých pracovníků se zpracováním dat a vytvořit něco, co má reálné využití. Práce se nezabývá pouze koncepčním návrhem systému sběru a zpracování experimentálních dat z měření prototypů mikro turboexpandérů, ale její podstatná část je věnovaná i realizaci systému. V rámci práce je vytvořen tzv. MVP¹, jehož testování a úprava již není obsahem této práce. V samostatné kapitole je diskutována ekonomika investice do realizace takového informačního systému, stejně tak jako další neekonomické přínosy jeho vývoje.

¹Minimum Viable Product

Teoretická část

Tato kapitola slouží k seznámení čtenáře s problematikou, kterou se tato práce zabývá. První část kapitoly je věnována automatizaci, jejím základům a vývoji. Důraz je kladen na vliv automatizace na člověka a její případné negativní a pozitivní důsledky.

Druhá část diskutuje měření pracovní výkonnosti v odvětví výzkumné činnosti a upozorňuje na problémy s tím související.

Třetí část je zaměřená na metody hodnocení investic pro vývoj informačního systému.

Poslední část vysvětluje techniky využívané při tvorbě této práce. Seznamuje čtenáře s jazykem UML², který je aktivně využíván při navrhování systému a jazykem MATLAB³, v kterém je výsledný systém naprogramován.

1.1 Automatizace

Automatizace může být definována jako transformace systému, který plní (částečně nebo úplně) funkci, která dříve byla nebo mohla být provedena (částečně nebo úplně) lidským operátorem. [4]

V dnešní době se pojem automatizace nevztahuje pouze na výrobu, jak tomu bylo v minulosti. Z doby, kdy se automatizace používala čistě k automatizaci manuální práce se lidstvo technologickým vývojem dostalo i k pokrývání kognitivní práce. [5] Počítače jsou využívány k interpretaci a zaznamenávání dat, dokáží dokonce i dělat rozhodnutí a vizualizovat informace. Tyto úkoly jsou považovány za automatizované, včetně přidružených sensorů. Nicméně stále existují systémy, kde je lidská složka důležitou součástí. Příkladem může být řízení letadla, kdy je v neočekávaných situacích zapotřebí rychlého rozhodnutí, které je většinou založené na předchozí zkušenosti člověka. [6]

²Unified Modeling Language

³MATrix LABoratory

Technické aspekty automatizace jsou povětšinou na popředí pozornosti veřejnosti, proto existuje mnoho odborných publikací zabývajících se technickou stránkou automatizace. Na rozdíl od rozsáhlé technické literatury existuje pouze malá, ale rostoucí výzkumná základna zkoumající lidské schopnosti zapojené do práce s automatizovanými systémy. Zkoumání otázek lidské výkonosti je zásadně důležité pro navrhování automatizovaných systémů. Designéři systému musí důkladně zvážit co automatizovat a do jaké míry, tak aby se vyvarovali negativním dopadům na člověka, vzhledem k faktu, že pouze málo věcí nelze automatizovat. [7]

Integrace člověka a stroje se nyní zaměřuje na sdílení úkolů, kontrolu a rozdělení zodpovědností tak, aby se přirozeně doplňovali. Centrální otázkou tedy není pouze jak navrhnout nejlepší automatizovaný systém, ale také jak rozdělit úkoly mezi člověka a stroj. [8]

1.1.1 Rozdělení úkolů

Počáteční přínos do oblasti přidělování úkolů poskytl Fitts, který představil seznam obecných úkolů (Tabulka 1.1), ilustrující kdy člověk exceluje nad strojem a naopak. [9]

Od jeho zavedení však vyvstala řada kritik ohledně použitelnosti seznamu. Jordan například uvedl, že srovnávat schopnosti člověka a stroje je nevhodné, protože pouze stroje mohou být navrženy pro konkrétní předem definované úkoly. Podle něj by myšlenka srovnávání člověka a technologie měla být zahozena, ale přesto by se při navrhování systému mělo brát v potaz, co lidé a stroje dělají nejlépe. Při návrhu by se tedy mělo uvažovat o člověku a stroji jako o doplňujících se entitách, nikoliv jako o konkurenčních entitách. [10]

Podíváme-li se na Fittsův seznam z dnešní perspektivy, jeví se jako zastaralý. Mnoho uvedených schopností v kterých by měl člověk předčit stroj již není pravdivých díky technologickému pokroku a proto není v současnosti použitelný. Uveden je zde pouze jako ilustrace prvních úvah o dělení úkolů mezi člověkem a strojem.

Jak je patrné z Fittsova seznamu, rozhodnutí o rozdělení úkolů mezi člověka a technologické systémy může být založena na různých faktorech. Například podle Williamse může být alokace úkolů rozhodnuta na základě finančních, technických nebo sociálních faktorů. [11]

Přidělení úkolu tomu správnému prvku v systému však není jediné, na co je třeba se zaměřit. Dalším důležitým faktorem je rozhodnutí, jak se bude lidský a automatizovaný prvek systému vzájemně podporovat na různých úrovních automatizace, tak aby byl celý systém robustní a zároveň flexibilní. [8]

Tabulka 1.1: Fittsův seznam[9]

Člověk překonává stroj	Stroj překonává člověka
<ul style="list-style-type: none"> • Schopnost detekovat malé množství vizuální a akustické energie • Schopnost vnímání světelných a zvukových vzorců • Rozhodování a výběr řešení • Schopnost uchovávat velké množství informací po dlouhou dobu a využívat je ve vhodný čas • Schopnost improvizovat a reagovat flexibilně • Schopnost využívat indukativní metody (poznání, které vychází z empiricky zjištěných faktů a dospívá k obecným závěrům) • Schopnost rozhodovat se 	<ul style="list-style-type: none"> • Schopnost rychlé reakce na kontrolní signály a přesné využití velké síly • Schopnost plnění opakujících se úkolů, rutinních úkolů • Schopnost uchovávat malé množství informací a mazat informace • Schopnost využívat deduktivní metody, včetně výpočetních dovedností (vyvozování nových tvrzení při dodržování pravidel logiky) • Schopnost zvládat vysoce komplexní operace, např. dělat hodně činností ve stejnou dobu

1.1.2 Úroveň automatizace

Vztah mezi člověkem a technologií může být popsán jako kontinuum⁴ od aktivity vykonávané zcela manuálně až po aktivity zcela automatizované. Koncept zabývající se spoluprací člověka s automatizovaným systémem nazýváme úroveň automatizace. [12]

Přesto se často stává, že automatizace je považována za přechod od manuálních operací k úplné automatizaci, tj. když jsou operátoři zcela nahrazeni roboty nebo pokročilými stroji. V osmdesátých letech minulého století sice existovaly ambice na vznik plně automatizovaných továren, ale v současné době víme, že nejvíce automatizovaných systémů ve výrobě je semi-automatických - tedy skládají se z kombinací manuálních a automatických úkolů. [13]

Podle CollinsDictionary jsou manuální operace definovány jako práce vykonávané ručně a stroje jsou stejným způsobem definovány jako nástroje, které využívají elektrickou energii nebo motor k vykonání příslušné práce. [14] Po-

⁴spojitý interval stavů systému

jem manuální práce lze tedy definovat jako práci prováděnou bez jakéhokoli nástroje nebo podpory. Proto můžeme poskytnutí dodatečného nástroje nebo jiné podpory uživateli k dosažení úkolu označit za zvýšení úrovně automatizace. Poskytováním další a další podpory můžeme dosáhnout až k plné automatizaci. [12]

Pro ilustraci si můžeme uvést příklad; šroub lze namontovat do konstrukce ručně, což by bylo považováno za nejnižší úroveň automatizace. Pokud ale dodáme dělníkovi manuální nástroj (např. šroubovák) tak zvyšujeme jeho úroveň automatizace. Tuto úroveň můžeme nadále zvyšovat výměnnou manuálního nástroje za elektrický a tak dále až po plně autonomní nástroj na úrovni plné automatizace.

Tento koncept byl zvažována mnoha autory, jejichž definice úrovně automatizace je shrnuta v Tabulce 1.2. Z tabulky je patrné, že nejjednodušší forma automatizace často pracuje ve dvou režimech, manuální nebo automatický.

Na základě provedené rešerše lze však poznamenat, že automatizace není všechno nebo nic. Měla by být spíše považována za kontinuum úrovní automatizace. Kde nejnižší úrovní je plně manuální provedení a nejvyšší úrovní je plná automatizace (bez jakékoli lidské účasti). Lze rovněž tvrdit, že automatizace se může na různých úrovních lišit.

1.1.3 Taxonomie

Dle Frohma je možné rozdělit automatizaci výroby do dvou základních tříd, dle druhu úkolu k automatizaci, mechanizace a komputerizace. Mechanizace je definována jako nahrazení lidské síly a komputerizace je podobným způsobem spojena s kognitivními úkoly, jako je nahrazení lidských smyslových a myšlenkových procesů. Například sběrem, uchováním, analýzou dat a jejich použitím za účelem kontroly výrobního procesu. [15]

Obrázek 1.1 znázorňuje jak s rostoucí úrovní automatizace přechází kognitivní úkoly na komputerizaci a podobně manuální úkoly na mechanizaci.

Ztotožníme-li se s tímto rozdělením, pak se tato práce zabývá komputerizací, a proto se nadále bude věnovat pouze komputerizaci.

1.1.3.1 Komputerizace

Mitalův a Pennathurův výzkum týkající se lidí v pokročilých, dynamických a automatizovaných systémech (např. dálkově ovládaných strojích a robotice), představil řadu taxonomií úrovně automatizace. [16] Stejně jako pokud se jedná o mechanizaci procesu, většina taxonomií v počítačových systémech a systémech orientovaných na informace a jejich řízení, komputerizace také nabízí rozdílné úrovně automatizace od plně manuálního ovládaní až k plné automatizaci. Podle Kabera a kol. by mělo být cílem zachovat v systému jak člověka tak počítač, aby se společně podíleli na aktivním řízení systému, a tím se zlepšovalo porozumění obsluhy. [17]

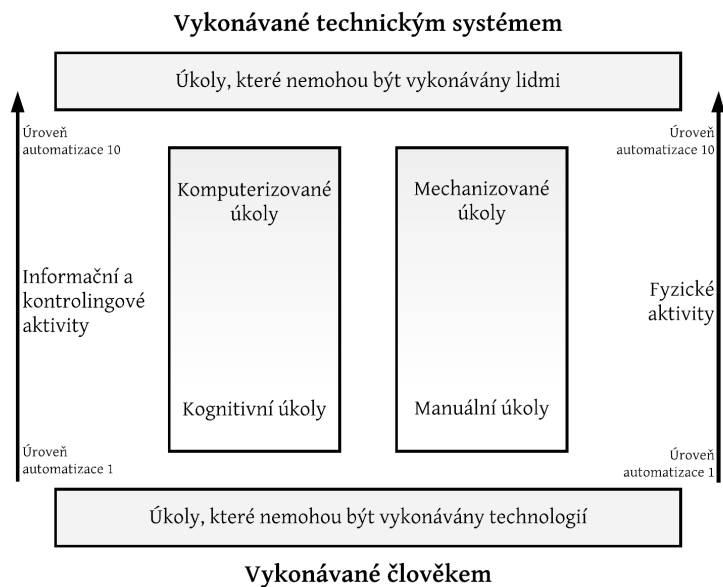
Tabulka 1.2: Definice úrovně automatizace dle různých autorů; převzato z [8]

Autor	Definice úrovně automatizace
Amber a Amber	Míra, do které je lidská energie a kontrola nad produkčním procesem nahrazena strojem.
Sheridan	Míra automatizace zohledňuje zpětnou vazbu tak i sdílení úkolů podle deseti úrovní automatizace (Tabulka 1.3)
Kern a Schumann	Stupeň mechanizace je definován jako technická úroveň v pěti odlišných dimenzích nebo pracovních funkcích.
Billings	Míra automatizace přechází od přímé manuální kontroly do široce autonomních operací, kde je lidská role minimální.
Endsley	Míra automatizace v kontextu expertních systémů je nejlépe aplikovatelná na kognitivní úlohy jako je schopnost reagovat a dělat rozhodnutí na základě systémových informací.
Satchell	Míra automatizace je definována jako sdílení úloh mezi člověkem a strojem s různou úrovní zapojení člověka.
Parasuraman a kol.	Míra automatizace je kontinuum od manuálních po plně automatizované operace.
Groover	Míra automatizace může být definována jako množství lidské práce v systému, která může být vykonávána jak manuálně, semiautomaticky tak i plně automaticky.

Z literatury vyplývá, že existují dva způsoby pohledu na úroveň automatizace. První je klasická desetistupňová taxonomie navržená Sheridanem a Verplanckem zaměřená na interakci člověka s počítačem. [18]

Jak je vidět v Tabulce 1.3, Sheridanova a Verplanckova taxonomie zahrnuje otázky zpětné vazby a relativní sdílení funkcí mezi člověkem a technickým systémem, určování variant, výběrem akce a implementací akce. Tato taxonomie je jednou z nejvíce popisných taxonomií, kterou lze v literatuře nalézt, pokud jde o identifikaci „co“ operátor (člověk) a technický systém (počítač) mají dělat v rámci různých úrovní automatizace a „jak“ by měly spolupracovat. Satchell navíc také zdůrazňuje, že ke sdílení úkolů mezi lidmi a stroji může docházet v mnoha formách a tyto formy lze stratifikovat do různých úrovní v závislosti na zapojení člověka. [19]

Druhým způsobem jak vnímat úroveň automatizace je z pohledu lidského



Obrázek 1.1: Dělení automatizace dle Frohma; upraveno z [15]

Tabulka 1.3: Deset úrovní automatizace [18]

1. Člověk zvažuje alternativy, rozhodne se a provede vybranou akci.
2. Počítač nabídne množinu alternativ, které může člověk při rozhodování ignorovat.
3. Počítač nabídne omezenou množinu alternativ a člověk se rozhodne kterou akci provést.
4. Počítač nabídne omezenou množinu alternativ a jednu doporučí, ale člověk se rozhoduje a provede konečnou akci.
5. Počítač nabídne omezenou množinu alternativ a jednu doporučí, akci po schválení člověka provede.
6. Počítač dělá rozhodnutí, ale dává člověku možnost vetovat rozhodnutí před provedením akce.
7. Počítač rozhoduje a provádí akci, po vykonání akce musí člověka informovat.
8. Počítač rozhoduje a provádí akci, informuje člověka pouze pokud se dotáže.
9. Počítač rozhoduje a provádí akci, informuje člověka pouze pokud je to oprávněné.
10. Počítač rozhoduje a provádí akci pokud předpokládá že by měl, informuje člověka jen pokud je to oprávněné.

Tabulka 1.4: Deset úrovní automatizace dle Endsleye a Kabera[22]

Úroveň automatizace	Role			
	Monitorování	Generování	Výběr	Implementace
Manuální kontrola	člověk	člověk	člověk	člověk
Podpora implementace	člověk/počítač	člověk	člověk	člověk/počítač
Dávkové zpracování	člověk/počítač	člověk	člověk	počítač
Sdílená kontrola	člověk/počítač	člověk/počítač	člověk	člověk/počítač
Podpora rozhodování	člověk/počítač	člověk/počítač	člověk	počítač
Smíšené rozhodování	člověk/počítač	člověk/počítač	člověk/počítač	počítač
Rigidní systém	člověk/počítač	počítač	člověk	počítač
Automatizované rozhodování	člověk/počítač	člověk/počítač	počítač	počítač
Dozorovaná kontrola	člověk/počítač	počítač	počítač	počítač
Plná automatizace	počítač	počítač	počítač	počítač

Tabulka 1.5: Čtyři úrovně automatizace[8]

Absence automatizace	Žádná podpora od systému.
Upozornění	Systém poskytuje informace o tom co se stalo a asistuje operátorovi v jeho rozhodování.
Návrh	Systém upozorňuje a navrhuje vhodnou akci, ale nechává operátora rozhodnout a jednat.
Akce	Systém sám vyřeší problém a informuje operátora, který může použít právo veta.

zpracování informací. [20] Zpracování informací člověkem se skládá z následujících čtyř kroků: získání informace, analýza a zobrazení informace, rozhodnutí o akci na základě analýzy a nakonec implementace akce na základě rozhodnutí. [21] Dle Parasuramana lze úkoly v každém z těchto čtyř kroků automatizovat na různé úrovni. Spojením taxonomie Sheridana a Verplanka s modelem lidského zpracování informací, Parasuraman ukázal, že konkrétní systém může zahrnovat automatizaci ve všech čtyřech krocích na různých úrovních. [21]

Endsley a Kaber představili taxonomii, která by mohla poskytnout podporu člověku pomocí expertního systému. Navrhli desetistupňovou taxonomii pro širší použitelnost v oblasti kognitivních a psychomotorických úkolů vyžadujících kontrolu v reálném čase. [22] Tato taxonomie je zachycena v Tabulce 1.4

V porovnání s modelem Endsleye a Kabera a Parasuramana, Lorenz a kol. dospěli k závěru, že automatizované systémy se neliší pouze v závislosti na úrovni automatizace, ale také v závislosti na různých typech automatizace. Navrhli, aby se na úrovně automatizace pohlíželo skrz čtyři stupně automatizace, které jsou zachyceny v Tabulce 1.5. [8]

S dalším způsobem, jak popsat úrovně automatizace, přišel Billings, který nenahlíží na úroveň automatizace jako na jednoduchý přesun od úplného ručního ovládání po plně automatizované řízení. Úrovně automatizace jsou rozdělením úkolů mezi člověka a automatizovaný systém, kde je důležité, že role operátora se může lišit. Může se např. jednat o přímé kontroly nad celým

procesem až po relativně pasivní kontrolní funkci, ve které je úkol operátora řešen technickým systémem. [6]

1.1.4 Model

Jak již bylo zmíněno v sekci 1.1.2, automatizované systémy mohou pracovat na několika úrovních vzhledem k integraci člověka do procesu. Parasuraman navrhl model [21], který k uvedeným 10 úrovním automatizace (Tabulka 1.3) přidružuje 4 stupňový pohled, jak člověk zpracovává informace. Níže popsaný model vychází z modelu navrženého Parasuramanem. [21]

Třídy funkcí při zpracování informace jsou následující:

- Získávání informací
- Analýza informací
- Rozhodování a výběr řešení
- Realizace řešení

Každou z těchto tříd funkcí lze automatizovat na mnoha různých úrovních. Příklad jak lze použít několik úrovní automatizace ve fázi rozhodování je vidět v Tabulce 1.3.

Získávání informací

Automatizace pro získávání informací se vztahuje na snímání a zaznamenávání vstupních dat. Tyto operace jsou ekvivalentní k první fázi zpracování informací člověkem. Na nejnižší úrovni se může taková automatizace skládat ze strategií jak mechanicky pohybovat senzory za účelem skenování a sledování. Mírná úroveň automatizace v této fázi může zahrnovat organizování příchozích informací podle nějakého kritéria, např. podle seznamu priorit se zvýrazněním některých částí informací.

Je třeba si však uvědomit, že jak organizace tak i zvýrazňování dat snižuje viditelnost originálních dat. Zvýrazňování a filtrování může vést k různým dopadům na lidský výkon, jak je popsáno později v sekci o spolehlivosti automatizace.

Analýza informací

Automatizace analýzy informací zahrnuje kognitivní funkce jako je pracovní paměť a inferenční procesy⁵. Při nízké úrovni, lze na příchozí data použít algoritmy, které umožňují extrapolaci dat v čase nebo předpovědi. Vyšší úroveň automatizace v této fázi zahrnuje integraci, ve které je několik vstupních proměnných sloučeno do jedné hodnoty. Složitější formy automatizace analýzy

⁵Proces užívaný k vyvozování nových informací z informací známých.

zahrnují „manažery informací“, kteří poskytují kontextově závislé souhrny údajů uživateli.

Rozhodování a výběr řešení

Třetí fáze, rozhodnutí a výběr opatření, zahrnuje výběr z možných alternativ. Automatizace této fáze zahrnuje různé úrovně rozšíření nebo nahrazení lidského rozhodování a výběru možností strojovým rozhodováním.

Například expertní systémy jsou navrženy s podmíněnou logikou, tedy souborem pravidel, které předepisují volbu, pokud existuje konkrétní podmínka. Různé úrovně automatizace v této fázi jsou nejlépe definovány původní taxonomií navrženou Sheridanem a uvedené v Tabulce 1.3, která definuje kontinuum od systému, který doporučuje akce až k systému, který je sám vykonává. [18]

Realizace řešení

Konečná fáze zahrnuje implementaci vybraného řešení. Automatizace této fáze zahrnuje různé úrovně strojové realizace dle výběru akce a obvykle nahrazuje ruku nebo hlas člověka. Rozdílná úroveň automatizace může být definována pomocí relativního množství manuální ku automatické aktivitě při vykonávání akce.

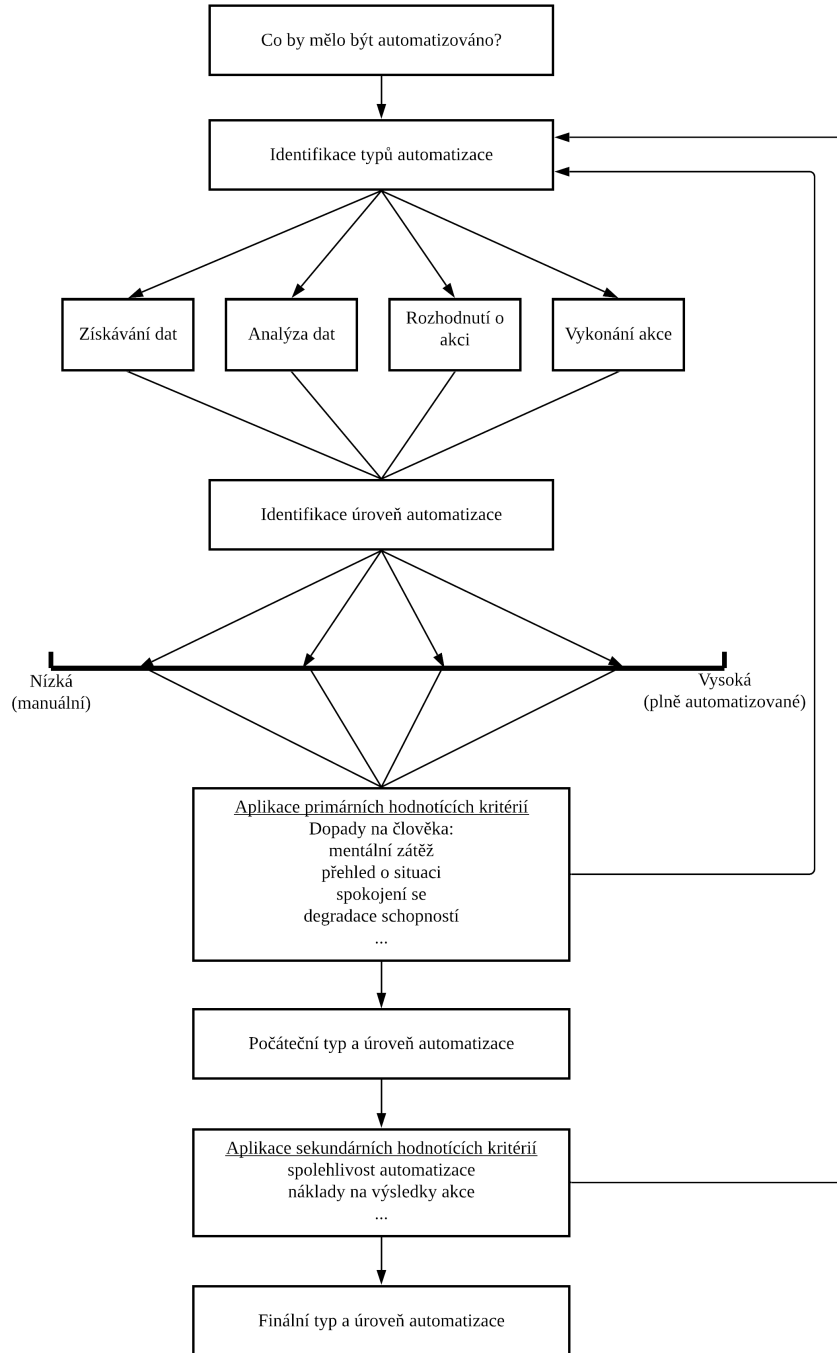
Příkladem může být tiskárna, která má možnost manuálního třídění, automatického třídění, automatického kompletování či automatického sešívání reprezentuje různé úrovně automatizace úloh, které může uživatel zvolit.

Automatizace akce zahrnuje agenty, kteří sledují interakci člověka s počítačem a na základě toho vykonávají určité podúkoly automaticky v závislosti na situaci. Automatizace závislá na kontextu je nazývána adaptivní automatizací.

1.1.5 Framework

Byl navržen framework [21], který pomáhá s vytvořením designu automatizace. Prvním krokem je uvědomit si, že automatizovat je možné každou část jiným dílem. Tento model pomáhá rozhodnout, kterou část – získávání informací, analýzu informací, rozhodování a výběr řešení nebo realizaci řešení - automatizovat a do jaké míry. Jak design navrhnout však není jednoduchou otázkou, vždy je potřeba hledat rovnováhu mezi benefity a náklady.

Navržený čtyř dimenzionální model může sloužit jako odrazový můstek pro finální design. Jak lze vidět na Diagramu 1.2 mnoho úrovní automatizace může být uvažováno pro každý typ automatizace. Autoři navrhli, že každá úroveň automatizace by měla být ohodnocena v souvislosti jaké bude mít následky pro člověka. Ale výkonnost člověka není jediným důležitým faktorem, sekundární hodnotící kritérium zahrnuje spolehlivost automatizace a náklady na rozhodnutí či realizaci akce. Aplikace těchto kritérií je navržena jako průběžný a rekursivní proces.



Obrázek 1.2: Proces návrhu designu pro automatizace; upraveno z [21]

1.1.5.1 Primární hodnotící kritéria

Přes dvě desetiletí se výzkumníci zabývají různými aspekty lidské interakce s automatickým systémem. Tento výzkum, který zahrnuje teoretickou analýzu, experimenty, simulace a modelování, studie v terénu a analýzu incidentů a nehod z reálného provozu, ukázal, že automatizace má jak pozitivní, tak i negativní dopad na výkonnost člověka. [23] Velice stručně budou popsány 4 oblasti: mentální pracovní zátěž, povědomí o situaci, spokojení se a degradace schopností. Samozřejmě existují i další hrozby přicházející s vytížením člověka, ale ty nejsou v uvedeném modelu uvažovány.

Mentální pracovní zátěž

Důkazy naznačují, že dobře navržená automatizace informací může snížit úroveň mentální zátěže pro operátora na úroveň, která je vhodná pro provedení daného úkolu.

Nejjednodušší úrovní organizování informací může být například seznam priorit, které pomohou operátorovi ve výběru informací relevantních pro dané rozhodnutí. Souhrny údajů mohou taktéž pomoci vyloučením časově náročného vyhledávání. Transformace dat, například grafická reprezentace informací může být také prospěšná.

Transformace a integrace nezpracovaných dat do určité formy (grafické nebo jiné), která odpovídá operátorově zastoupení systémových operací bylo shledáno jako užitečný konstrukční princip. [24] Tyto výsledky by však neměly být vykládány tak, že výsledkem je vždy vyvážená pracovní zátěž operátora. Případy, kdy automatizace zapříčinila nárůst mentální zátěže byly také nalezeny. Většinou se jedná o systémy, ve kterých je obtížné automatizaci zahájit a zapojit, čímž se zvýší kognitivní zátěž a pokud je vyžadováno rozsáhlé zadávání dat, tak i fyzická pracovní zátěž operátora. [25] Obecně platí, že účinek automatizace na duševní pracovní zátěž se zrcadlí s účinkem na zlepšování produktivity a účinnosti člověka. [26]

Povědomí o situaci

Automatizace rozhodovací funkce může snížit povědomí obsluhy o systému a určitých dynamických prvcích pracovního prostředí. Lidé mají tendenci méně si všimnout změn v prostředí nebo systémových stavech, když jsou tyto změny pod dohledem jiného agenta (ať už je to agent automatizace nebo jiný člověk), než když provádí změny sami. [27]

Také pokud pomáhá s rozhodnutím člověku expertní systém nebo rozhodnutí rovnou sám udělá, operátor ztrácí přehled o zdrojích informací v systému, protože není sám aktivně zapojen k vyhodnocování dat a konání rozhodnutí. K této situaci může dojít, když je operátor pasivní a jehož úkolem je monitorovat proces a zakročit pouze pokud hrozí chyba či nehoda. [27]

Spokojení se

Pokud je automatizace vysoce, ale nikoliv perfektně, spolehlivá při provádění rozhodnutí, může operátor ztratit ostražitost a nesledovat informační zdroje, a tak nemusí příležitostně správně detekovat pochybení automatizovaného procesu. Tento efekt nadměrné důvěry nebo „spokojení se“ je největší, když je operátor zapojen do více úkolů a monitorování automatizovaného systému je méně zřejmé, oproti tomu, kdy je monitorování automatizovaného systému jediným úkolem operátora. [27]

Automatizace informační analýzy může také vést ke spokojení se pokud algoritmy, z nichž vychází filtrování, predikce nebo integrační operace jsou spolehlivé, ale ne perfektně. [28] Proto se efekt spokojení se může objevovat i při automatizaci získávání a analýzy informací nikoliv jen při fázi rozhodování. [29]

Degradace schopností

Pokud je rozhodovací funkce neustále prováděna automatizovaně, přijde čas kdy lidský operátor nebude již tak schopný jako když sám prováděl tuto funkci. Toto tvrzení podporuje velký soubor výzkumů dokumentujících, že bez používání dochází k zapomínání a úbytku dovedností. [30]

Potenciální náklady, jako je snížené povědomí o situaci, spokojení se a degradace dovedností, společně ukazují, že automatizace na vysoké úrovni může vést k neznalosti operátorů. [31] Všechny tři zdroje zranitelnosti můžou představovat nebezpečí v případě selhání systému, a proto musí být automatizace navržena tak, aby se takové potenciální náklady na lidský výkon nevyskytovaly.

1.1.5.2 Sekundární hodnotící kritéria

Spolehlivost automatizace

Pozitivní vliv automatizace na mentální pracovní zátěž operátora pravděpodobně nebude platit, pokud je automatizace nespolehlivá. Proto je zajištění vysoké spolehlivosti kritickým hodnotícím kritériem při implementaci automatizace. Několik postupů pro odhad spolehlivosti bylo navrženo stejně tak i různé metody pro analýzu spolehlivosti pomocí softwaru. [32] Použití těchto technik může být užitečné, ale pouze pokud jsou jejich výsledky interpretovány obezřetně. Zejména to, co se jeví jako „tvrdá čísla“, například spolehlivost 0,997 nebo střední doba do selhání 100 000 hodin. Je třeba na tyto výsledky pohlížet s jistým skepticismem, protože tyto hodnoty představují odhad průměru, zatímco by bylo potřeba znát i odchylku od průměru, která může být značná. [33]

Navíc spolehlivost automatizovaného systému nemůže být vždy jednoduše definována pomocí pravděpodobnosti. Selhání mohou nastat i bez předvídatel-

né příčiny poruchy v softwaru nebo hardwaru, ale protože modelované předpoklady designérů nejsou v dané operaci splněny. [4]

Spolehlivost automatizace je důležitým determinantem kvůli jejímu vlivu na člověka. Nespolehlivost snižuje důvěru operátorů a může podkopávat potenciální výhody automatizace. Automatizované systémy mohou být nedostatečně využívány nebo vyřazovány kvůli nedůvěře. [4] Příkladem může být kouřový detektor s přecitlivělým čidlem, který neustále spouští falešný alarm, po čase bude ignorován a nebude tak plnit svoji funkci.

Pokud lze u automatizace informací zajistit extrémně vysokou spolehlivost, pak můžeme usilovat o velmi vysoké úrovně automatizace informací. Samozřejmě vysokou spolehlivost nelze garantovat u mnoha případů. Například nepřesnost snímacích senzorů nebo změna priorit operátora může zapříčinit, že algoritmus používaný automatizací bude nevhodný pro dané podmínky. Přesto může být automatické získávání a analýza dat využívána na relativně vysoké úrovni, pokud má operátor přístup k původním datům (např. zvýraznění, ale ne filtrování) a je si vědom úrovně nespolehlivosti, kdy bude věnovat pozornost i původním datům. [34]

Náklady na výsledky rozhodnutí

Vysoká úroveň automatizace může být spojena s potenciálními náklady, snížení povědomí operátora o situaci, spokojení se a degradaci dovedností. To však neznamená, že vysoké úrovně automatizace by neměly být brány v úvahu při rozhodování a realizaci akcí. Posouzení vhodné úrovně automatizace pro rozhodování vyžaduje zvážení nákladů spojených s výstupy rozhodnutí a akce. Náklady na nesprávné či nepřesné rozhodnutí a následné související akce, které lidé a automatické systémy vykonávají se ve většině systémů liší. Mnoho rutinních akcí má předvídatelné důsledky, které zahrnují malé nebo žádné náklady, pokud se akce nevydaří podle plánu. Rizika spojena s výsledkem rozhodnutí lze vyčíslit jako náklady na chybu krát pravděpodobnost této chyby. [21]

Pro rozhodnutí zahrnující relativně malé riziko, pravděpodobně nebude mít velký dopad, i když nastane úplné selhání systému. Takováto rozhodnutí jsou dobrými kandidáty na vysokou úroveň automatizace. Ve skutečnosti, kdyby operátoři museli neustále činit každé z těchto relativně jednoduchých rozhodnutí mohlo by je vytěžovat tak, že by nemohli zastávat jiné důležité funkce.

Automatizace rozhodování na vysoké úrovni může být také odůvodněna ve vysoce časově kritických situacích, kde lidský operátor nemá dostatek času na odpověď a vhodnou reakci. Pokud se ale za neobvyklých okolností očekává, že operátor převezme kontrolu, není vysoká automatizace doporučena vzhledem k negativním dopadům na člověka zmíněným dříve v této práci. [21]

1.2 Měření pracovní výkonosti

Produktivita je většinou definována jako poměr množství vstupů na množství použitelných výstupů (Rovnice 1.1).

$$Produktivita = \frac{vstupy}{vystupy} [-] \quad (1.1)$$

Ikdyž s touto obecnou definicí lze souhlasit, tak literatura věnovaná produktivitě práce ihned odhalí, že její použití není obecně uplatnitelné. [35] Každý obor má svá specifika čímž vznikají různé nároky na měření produktivity. Další podkapitola odhaluje jaké výzvě čelí měření produktivity v odvětví výzkumné činnosti.

1.2.1 Produktivita ve výzkumné činnosti

Obecně můžeme říct, že cílem výzkumné činnosti je získávání nových znalostí. Výzkumná činnost je proces, ve kterém se na vstupu nachází lidské, hmotné a nehmotné zdroje a na výstupu nové znalosti. Mezi hmotné zdroje řadíme vědecké přístroje a materiály, které mohou být využívány při experimentech, nehmotnými zdroji pak nazýváme již získané znalosti či sociální sítě. Výstup, nové znalosti, mají charakter jak hmotné povahy, jako jsou publikace, patenty, prezentace na konferencích, atd., tak i nehmotné povahy, kdy mluvíme o nabytých vědomostech či poradenské činnosti. [36]

Při produkování nových znalostí se tedy jedná o funkci s více vstupy a výstupy. Hlavním ukazatelem efektivity výrobní jednotky je produktivita, jednoduše řečeno, to co je vyprodukováno v daném časovém období na základě dodaných vstupů. Pro výpočet produktivity výzkumu je pak potřeba přijmout několik zjednodušení a předpokladů.

Není možné přesně změřit jakékoliv nové znalosti, které nejsou kodifikovány. I když jsou kodifikovány, čelíme problému identifikace a poměrování jejich různých forem. Bylo prokázáno, že v tzv. tvrdých vědách převládá jako forma výstupu vědeckých činností publikace ve vědeckých časopisech. [37] Databáze jako je Scopus nebo Web of Science jsou rozsáhle používány a testovány v bibliometrických analýzách⁶ a jsou dostatečně transparentní ve smyslu jejich pokrytí a dosahu. Tento ukazatel je přijatelný jedná-li se o přírodní vědy, nikoli však v umění, humanitních vědách a značné části věd společenských.

Jiné formy výstupu, zejména tedy patenty, lze identifikovat v komerčních nebo bezplatných databázích, jako jsou Derwent a Espacenet. S patentem jsou povětšinou spjaté i publikace, které objasňují daný patent. Nelze tedy vyloučit, že kredit za patent není uměle zdvojnásoben díky těmto článkům.

Výzkumné projekty často zahrnují tým výzkumníků, což je zrcadleno ve spoluautorství publikací. Měření produktivity musí tedy zohledňovat dílčí

⁶měření dosahu příspěvků

příspěvky jednotlivých přispěvatelů na výstupu. Příspěvky jednotlivých spoluautorů nemusí být nutně stejné, a tak v některých oblastech pořadí autorů signalizuje míru jejich příspěvku (konvence jak je řadit se liší napříč obory). [38] Všechny ukazatele založené na úplném nebo přímém počítání autorství, kdy jen hlavní autor obdrží kredit, nebo se autorství dělí rovným dílem v oborech, kde je přínos autorů vyjádřen v různém poměru v závislosti na pořadí jmen v publikaci jsou nevhodné. Pro korektní vyjádření produktivity proto nelze tento ukazatel využívat.

Dalším faktorem ovlivňující tento ukazatel spočívá ve skutečnosti, že v každém odvětví se intenzita publikování liší. [39] Aby se předešlo zkreslení v žebříčku produktivity, lze porovnávat pouze výzkumné pracovníky ve stejné oblasti výzkumu. [40] Hodnocení produktivity jednotlivce bez zkreslení by pak znamenalo hodnocení jednotlivce právě v jednom oboru. Z čehož hned můžeme předpokládat dvoustupňový přístup pokud budeme porovnávat pracovníky z různých oborů. Nejprve se provede hodnocení jedince ve vlastním oboru a následně bude tato hodnota agregována, tak aby mohla být porovnána s agregovanou hodnotou výzkumníka z jiného oboru. [36]

V bibliometrii je pojem produktivita založen na počtu publikací a pojem dopad vyjadřován pomocí počtu citací publikací autora. Z mikroekonomického hlediska by takovéto definování produktivity mělo smysl pouze pokud by všechny publikace měly stejnou hodnotu, k čemuž ve skutečnosti ale nedochází, protože různé publikace přináší nové znalosti o různých hodnotách, kde se jejich hodnota měří jejich dopadem na vědecký pokrok. [41]

Manuální zpracování dat z experimentálních měření je jednou z pracovních náplní výzkumných pracovníků, která je do značné míry nekreativní a přímo se nabízí ji delegovat na stroje a počítače. Automatizace sběru a zpracování dat umožní výzkumným pracovníkům věnovat se kreativnějším činnostem s vyšší přidanou hodnotou a zvýšit tak jejich produktivitu. V přeneseném slova smyslu ušetří čas, potažmo peníze, který by musel být věnován na zpracování experimentálních hodnot. [42] Na toto téma se právě zaměřuje praktická část této diplomové práce.

Pokud se oprostíme od bibliometrického způsobu měření produktivity, například v aplikovaném výzkumu a experimentálním vývoji, měřit produktivitu se ukáže být jako velmi náročné. Roku 1957 se Peter Drucker, někdy označovaný jako otec moderního managementu, zabýval právě touto otázkou - jak v tzv. *znalostní ekonomice* měřit produktivitu. Se standardní definicí produktivity, která je uvedena v Rovnici 1.1 však narazil hned v prvním členu, čitateli, na problém s definicí toho, co je vlastně úkolem pracovníka ve znalostní ekonomice. Na rozdíl od manuálních pracovníků, kdy je očekávaný výstup zřejmý, tomu u takovýchto pracovníků dost často není nebo nemusí být. [42]

Bibliometrie tedy, byť nedokonalá, je pravděpodobně to nejlepší, nejobjektivnější a celosvětově používané, co lidstvo prozatím v ohledu měření produktivity vědecko-výzkumných pracovníků vymyslelo.

1.3 Investice do vývoje informačního systému

Tato kapitola přibližuje čtenáři problematiku ekonomického hodnocení investic, se zaměřením na informační systémy.

Vkládá-li podnikatel, či jako v tomto případě vedoucí výzkumného týmu, prostředky prozíravě a účelově do služeb či produktů, od kterých v budoucnu očekává navýšení dlouhodobé prosperity podniku či skupiny, lze takovéto peněžní výdaje označit za *investice*. Investice by měly z dlouhodobého hlediska přinášet příjmy a mít dlouhodobý pozitivní dopad na chod společnosti. Investice tedy není možné chápat jen jako vynakládání prostředků, ale vždy v souvislosti s nadějí v navýšení budoucí hodnoty společnosti. [43]

Existuje mnoho metod pro hodnocení investic. Základní dělení může být chápáno na ty, které respektují faktor času a rizika a naopak na ty, které jej nerespektují. Takto můžeme rozdělit metody ekonomického hodnocení investic na metody *statické*, které hodnotí pouze finanční toky a metody *dynamické*, které kromě peněžních toků berou v potaz i riziko a čas.

Do kategorie statických metod řadíme například *prostou dobu návratnosti* (PP - Payback Period), *metodu návratnosti investice* (ROI - Return on Investment), *celkový příjem z investice*, *průměrné roční cash flow* nebo varianty těchto metod. Níže jsou popsány ty vybrané, které budou použity pro hodnocení realizovaného informačního systému z praktické části v Kapitole 2.

Jelikož je investice do tvorby tohoto informačního systému vnímána spíše z pohledu realizované úspory času zaměstnanců výzkumného týmu, je vhodnou a srozumitelnou statickou metodou pro investora doba návratnosti investice do zakázky na tvorbu informačního systému. Použitý vztah pro prostou dobu návratnosti PP je znázorněn v Rovnici 1.2.

$$PP = \frac{\text{naklady do investice}}{\text{prumerny ročni zisk}} [\text{let}] \quad (1.2)$$

Investiční náklady jsou jednorázové, jedná o zakázku formou subdodávky pro výzkumný tým. Průměrný roční zisk je realizován formou průměrné roční úspory času, přepočteného přes hodinovou sazbu výzkumného pracovníka na osobní náklady zaměstnance, který strávil určité množství času vyhodnocováním dat z měření. Pro vyjádření *celkového příjmu*, resp. *celkové úspory* a *čisté celkové úspory* je použito následujících vztahů 1.3 a 1.4. Kde CF_i značí cash flow v roce i .

$$CP = \sum_{i=1}^n CF_i [Kč] \quad (1.3)$$

$$NCP = -IN + \sum_{i=1}^n CF_i [Kč] \quad (1.4)$$

Z pohledu *dynamických* metod bude investice hodnocena za pomoci *metody čisté současné hodnoty* (NPV - Net Present Value, viz Rovnice 1.5)

a *vnitřního výnosového procenta* (IRR - Internal Rate of Return, viz Rovnice 1.6). Vzhledem k tomu, že se jedná spíše o drobnou investici s nízkou očekávanou dobou návratnosti, která bude do velké míry odvislá od provedeného počtu experimentů a intenzitou vývoje nových generací turboexpandérů, bude faktor času hrát nižší roli, než u větších dlouhodobějších investic.

$$NPV = -IN + \sum_{i=1}^n \frac{CF_i}{(1+WACC)^i} [Kč] \quad (1.5)$$

$$-IN + \sum_{i=1}^n \frac{CF_i}{(1+IRR)^i} = 0 \quad (1.6)$$

Příčemž IRR je taková diskontní sazba, pro které je čistá současná hodnota rovna nule. Pro přijetí investice by IRR mělo být alespoň rovné či vyšší váženým nákladům na kapitál. Samotný výpočet IRR je proveden prostřednictvím funkce v MS Excel.

Konečně je ještě pro zhodnocení investice využito tzv. *NPV kvocientu* (PI - Profitability Index, viz Rovnice 1.7), který vyjadřuje míru ziskovosti v relativním poměru ku investičním nákladům. Jedná se tedy o podobný indikátor jako je NPV, nicméně je relativizován. Jeho jednotkami tedy jsou násobky investičních nákladů, proto označení kvocient. Investice je pak doporučena, je-li NPV_q (PI) rovno alespoň 1 a s rostoucím kvocientem roste profitabilita investice.

$$PI = \frac{\sum_{i=1}^n \frac{CF_i}{(1+WACC)^i}}{IN} [-] \quad (1.7)$$

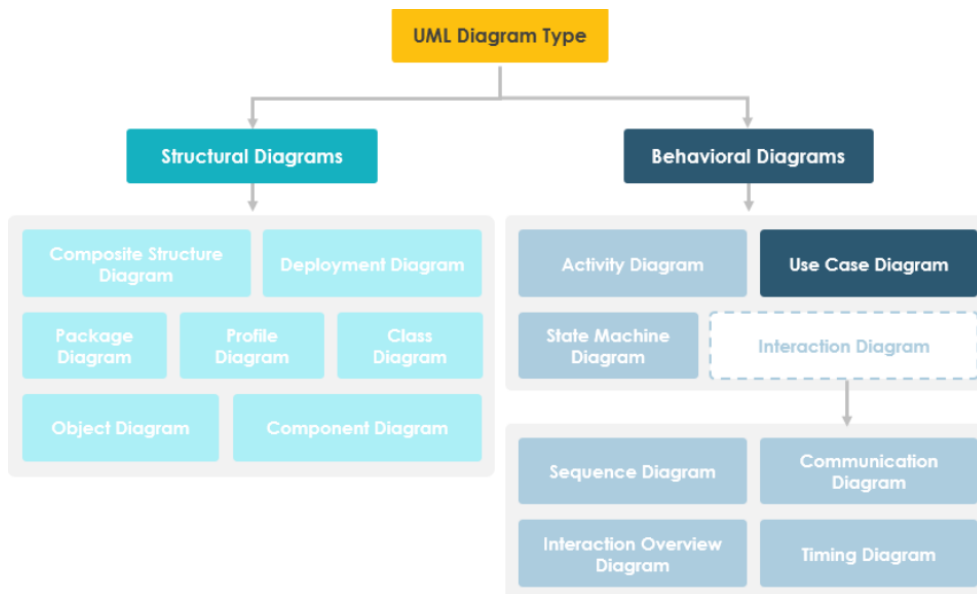
1.4 Používané jazyky

V této sekci je čtenář seznámen s jazyky, které jsou v této práci používány. Nejprve mu je představen modelovací jazyk UML a následně jazyk MATLAB, využitý pro realizaci informačního systému.

1.4.1 UML

UML, neboli Unified Modeling Language, je standardizovaný modelovací jazyk, který se skládá z různých druhů diagramů vyvinutých tak, aby pomáhaly vývojářům specifikovat, vizualizovat, konstruovat a dokumentovat jednotlivé prvky systému, včetně jejich struktury. Nejedná se o jazyk využívaný pouze k návrhu počítačových systémů, nýbrž i k modelování nesoftwareového charakteru. [44]

UML představuje kolekci příkladů dobré praxe, které se uplatnily při modelování velkých a komplexních systémů. Jedná se o důležitou složku vývoje



Obrázek 1.3: Typy UML diagramů; převzato z [46]

objektově orientovaných systémů a procesu vývoje obecně. UML používá především grafické notace k vyjádření podoby softwarového projektu. Jeho používání pomáhá s komunikací, návrhem systému a validací architektury v rámci projektového týmu. [45]

UML 2.0 definuje 13 typů diagramů rozdělených do 3 kategorií - diagramy strukturální, chování a diagramy interakcí. Jejich vzájemné vztahy jsou vidět na Obrázku 1.3, kde jsou rovněž všechny typy diagramů vyjmenovány. [44]

Strukturální diagramy znázorňují statickou strukturu systému a jeho částí na různých úrovních abstrakce a implementace, a jak spolu tyto části souvisí. Jednotlivé elementy strukturálních diagramů mají za úkol reprezentovat koncept systému.

Diagramy chování ukazují dynamické chování objektů v systému, které může být popsáno jako série změn v systému a v čase.

Kategorie *diagramů interakce* je celá odvozená z obecnější kategorie diagramů chování, disponuje tedy stejnými vlastnostmi jako tato nadřazená kategorie.

V této práci jsou z UML kolekce využívány pouze diagramy chování a proto jsou pouze tyto diagramy popsány podrobněji.

1.4.1.1 Use Case Diagram

Use Case Diagram, neboli diagram případů užití je ve většině případů jednoduchý. Nezobrazuje detaily případů užití, ale pouze shrnuje nějaké vztahy mezi jednotlivými případy užití, aktéry a systémem. Diagram rovněž nere-

flektuje skutečné pořadí kroků vykonávaných k dosažení cíle případu užití. [46]

UC⁷ diagramy typicky vznikají v počáteční fázi vývoje a jsou většinou aplikovány za účelem:

- specifikace kontextu systému
- zachycení požadavků na systém
- ověření architektury systému
- řízení implementace a vytváření testovacích scénářů

V diagramu se vyskytují entity, které jsou popsány níže a jejich grafické znázornění můžete nalézt na Obrázcích 1.4, 1.5. Příklad, jak se diagram používá je znázorněn na Diagramu 1.6. Diagram zachycuje, že zákazník (aktér) může v objednávkovém systému objednat zboží, kde úkon Objednání zboží mimo jiné obsahuje úkon Přihlášení do systému. Přihlašování do systému může být rozšířeno o případ užití Nesprávné heslo, což nastane v případě zadání nesprávného hesla během přihlašování.

Aktér

je ten, kdo interaguje s případy užití (se systémem). Každý aktér musí být přiřazen alespoň k jednomu případu užití. Zajišťuje vstupy do systému a od systému očekává výstupy.

Případ užití

je funkce systému (manuální či automatický proces).

Relace (Komunikační asociace)

označuje vztah znázorňující tok informací mezi prvky systému.

Relace «include»

se používá pokud případ užití využívá funkcionality jiného případu užití.

Relace «extend»

rozšiřuje vztah o specifické chování systému za splnění dané podmínky.

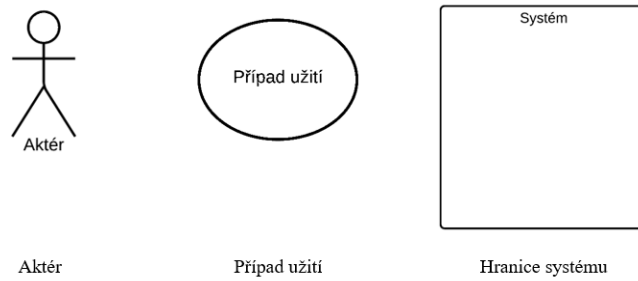
Relace generalizace

popisuje vztah rodič-potomek, kdy potomek je vylepšením rodiče.

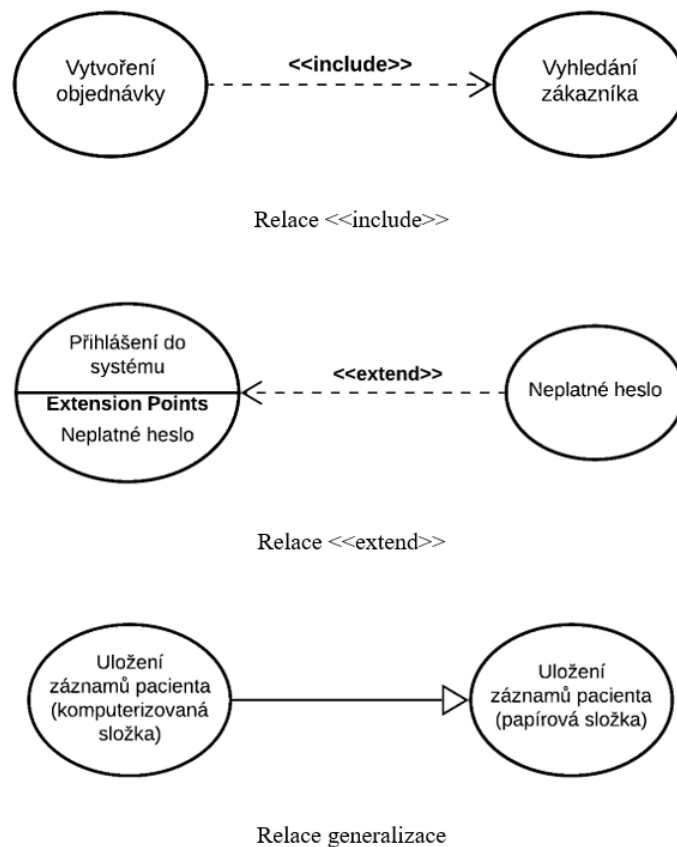
Hranice systému

znázorňují hranice celého systému, v případě komplexního systému mohou ohraničovat jednotlivé moduly systému.

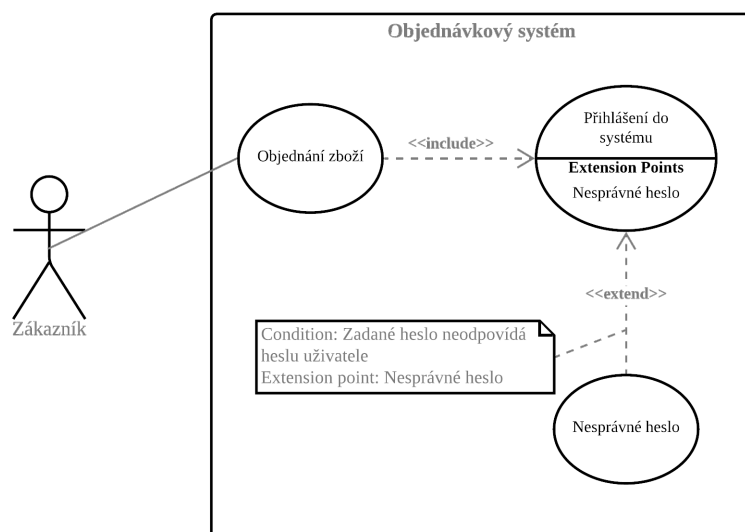
⁷Use Case



Obrázek 1.4: Základní entity Use Case Diagramu



Obrázek 1.5: Relace Use Case Diagramu



Obrázek 1.6: Ukázka použití Use Case Diagramu

1.4.1.2 Activity Diagram

Activity diagram či diagram aktivit je dalším důležitým diagramem chování. Popisuje dynamické aspekty systému, dalo by se říci, že se jedná o vylepšenou verzi procesního diagramu. [47]

Používá se pro popis aktivit, které vedou k cíli dané aktivity. Typicky se jedná o událost, která je dosažena pomocí různých kroků, které je potřeba koordinovat. Také je vhodný pro modelování jak jeden UC interaguje s jiným, obecně pro UC, kde se aktivity překrývají.

Diagram aktivit využívá několik druhů různých uzlů, ty nejpoužívanější jsou zobrazeny na Obrázku 1.7 a společně s ostatními prvky diagramu popsány níže.

Aktivita

se používá k reprezentaci sady akcí.

Akce

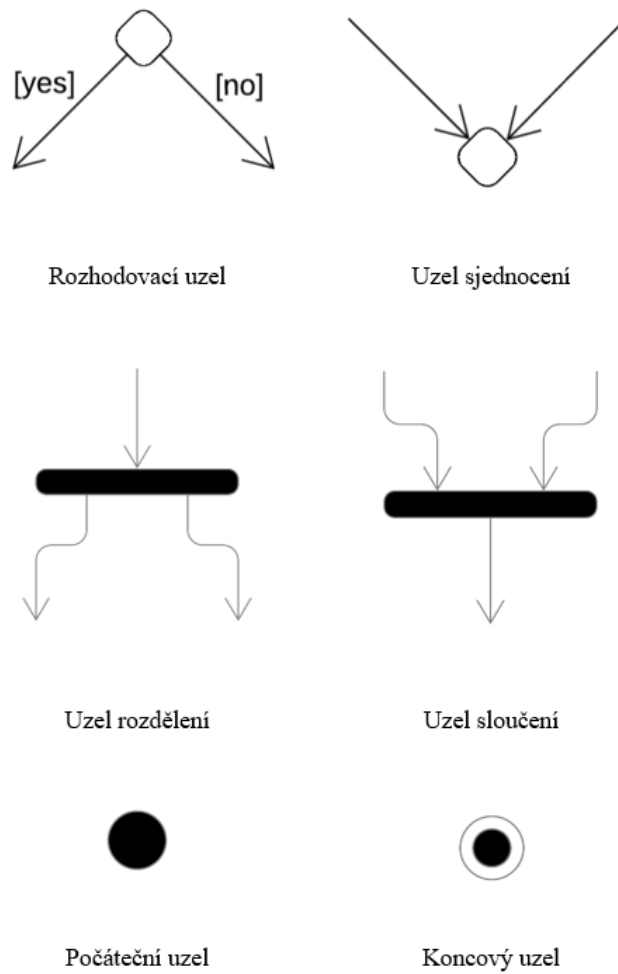
označuje úkol, který má být vykonán.

Kontrolní tok

ukazuje sekvenci vykonávání.

Objektový tok

zobrazuje tok objektu od jedné aktivity (nebo akce) do jiné aktivity (nebo akce).



Obrázek 1.7: Základní uzly v Activity Diagramu

Počáteční uzel

určuje počátek sady akcí nebo aktivit.

Koncový uzel

zastavuje všechny kontrolní i objektové toky aktivity (nebo akce).

Objektový uzel

reprezentuje objekt, který je spojen se sadou objektových toků.

Rozhodovací uzel

reprezentuje podmínku, která zaručuje, že kontrolní či objektový tok bude pokračovat pouze jednou cestou.

Uzel sjednocení

sjednocuje dohromady různé cesty, které vznikly pomocí rozhodovacího uzlu.

Uzel rozdělení

rozděluje chování do paralelních či konkurenčních toků aktivit (nebo akcí).

Uzel spojení

spojuje paralelní nebo konkurenční toky aktivit (nebo akcí).

Plavecká dráha

je způsob jak seskupit aktivity vykonávané stejným aktérem v rámci diagramu aktivit.

1.4.1.3 Machine State Diagram

Machine State Diagram, nebo také diagram stavů systému zachycuje fakt, že chování jednotlivých entit nezávisí pouze na vstupech, ale závisí i na předcházejícím stavu. UML stavový diagram zobrazuje různé stavy entity a způsob, jak systém reaguje na různé události. [48]

Diagram je většinou používán k popisu chování, které je závislé na stavu objektu. Objekt odpovídá rozdílně na stejnou událost v závislosti na tom, v jakém stavu se nachází. Můžeme ho aplikovat na jakýkoliv element, který je ve vztahu s jinou entitou, např. aktérem, případem užití, metodou atd.

Obecně lze říci, že stavový diagram je graf skládající se ze stavů a přechodů mezi stavy. Grafická podoba dále zmíněných prvků je stejná jako u diagramu aktivit.

Stav

je omezení či situace v životním cyklu objektu. Charakterizuje tedy podmínku za které objekt setrvává v daném stavu.

Přechod

popisuje pohyb mezi stavy, kde je na každé přechodové hraně uvedeno, která událost spustila přechod.

Počáteční stav

je znám jako počáteční pseudo-stav, kde přechod z tohoto stavu ukáže skutečný první stav.

Koncový stav

slouží k ukončení přechodů před ukončením systému jako celku.

Událost

reprezentuje událost, která způsobí přechod z jednoho stavu do stavu jiného.

1.4.2 MATLAB®

MATLAB neboli „matrix laboratory“ je programovací platforma navržená speciálně pro inženýry a vědce. Srdcem MATLABu je jazyk MATLAB, jazyk založený na maticích, který umožňuje vyjadřování výpočetní matematiky. [49] Zatímco ostatní programovací jazyky obvykle pracují s čísly jedno po druhém, MATLAB pracuje na celých maticích a polích.

MATLAB kombinuje desktopové prostředí uzpůsobené pro iterativní analýzu a návrhové procesy s programovacím jazykem. [50] Dále také disponuje velkým množstvím vestavěných funkcí, které se každým dnem zvětšuje.

MATLAB může být využit i v rámci jiného programovacího prostředí bez potřeby zapínání desktopové verze aplikace. Momentálně lze využívat z prostředí programovacích jazyků: [51]

- C/C++
- Fortran
- Java
- Python
- COM komponenty a aplikace (včetně mnoha programů psaných pomocí Visual C# .NET a Visual Basic .NET)

Další bezespornou výhodou je interaktivní práce s daty s možností grafického výstupu. Jazyk umožňuje vykreslování grafů s možností jejich další optimalizace. Velké plus přináší jazyku také dobře zpracovaná a rozsáhlá nápověda ve spojení s velkou uživatelskou komunitou sdílející své znalosti. [52]

Různorodé možnosti využití přináší i množství nástrojů specializovaných na jedno konkrétní odvětví. Příkladem může být například prostředí *App Designer*, interaktivní vývojové prostředí pro návrh rozvržení aplikace a programování jejího chování. Poskytuje plně integrovanou verzi editoru MATLAB a velkou sadu interaktivních komponent uživatelského rozhraní. Také nabízí správce rozložení komponent na usnadnění práce s uživatelským rozhraním a možnosti automatického přeformátování, tak aby aplikace detekovala změny velikosti obrazovky a reagovala na ně. Aplikace lze distribuovat vytvořením samostatně spustitelné desktopové nebo webové aplikace přímo z prostředí App Designeru. [53] Právě pomocí tohoto nástroje je vytvořena aplikace v rámci této diplomové práce.

Praktická část

Začátek této kapitoly si klade za cíl seznámit čtenáře s problematikou automatizace zpracování a vyhodnocování dat z experimentálního měření vzduchových turboexpandérů.

Následně je proveden návrh systému pro určitou část z popsaného problému. Nejprve je zvolena míra, do které bude daná část procesu automatizována a následuje návrh samotného systému pomocí UML diagramů.

Závěrem kapitoly jsou popsány kroky a funkce využité v rámci realizace informačního systému. Opomenuto nezůstalo ani grafické rozhraní, které je v kapitole také ukázáno a popsáno.

2.1 Popis problému a současný stav řešení

Na ČVUT jsou vyvíjeny vzduchové turboexpandéry s cílem aplikovat výsledky výzkumu a vývoje v komerčně dodávané biomasové mikrokogenerační jednotce pracující na základě organického Rankinova cyklu (ORC) - tepelného cyklu sloužícího k přeměně tepla na energii mechanickou, resp. elektrickou. [54]

Aby takovýto tepelný cyklus mohl provozně fungovat, musí být vybaven expanzním strojem - komponentou, která je srdcem tepelného cyklu a právě v ní dochází k expanzi pracovního média, při které je potenciální tlaková energie přeměněna v mechanickou energii rotace hřídele stroje, který je přes spojku propojen s elektrickým generátorem, ve kterém je mechanická energie kontinuálně transformována v elektrickou.

Pro mikro ORC jednotky je standardně používaný objemový expandér. Tento stroj je snazší co do návrhu a provozu, avšak dosahuje nižších provozních účinností a také má nízkou životnost. Proto se výzkumní pracovníci z oblasti energetického inženýrství mikrokogeneračních zařízení zaměřují na vývoj mikroexpanzních strojů odlišné koncepce, a to sice dynamických lopatkových strojů (turboexpandérů). Ty fungují na jiném fyzikálním principu a slibují přínos ve smyslu navýšení účinnosti transformace energie, prodloužení životnosti stroje a také snížení ceny expandéru, který je nejvyšší cenovou

položkou na mikrokogeneračních ORC zařízeních. Pokud by vývoj takovýchto turboexpandérů dosáhl komercializace, umožnilo by to plošné tržní nasazení těchto jednotek, a tedy celospolečenský nárůst přeměny primární obnovitelné energie v elektřinu. [55]

Hlavním úkolem experimentálního měření je vyhodnocení účinnosti prototypů mikro turboexpandéru stejné geometrie, avšak vyrobených za pomoci různých aditivních technologií (3D tisku) a opatřených různými typy povrchových úprav a dokončovacích operací. Zkoumány byly vlivy, jako je například vliv výsledné drsnosti povrchu na účinnost stroje, dosahované přesnosti různých aditivních technologií nebo (ne)homogenita výrobku a její efekt na dynamiku stroje. Za účely experimentu byla vybudována experimentální trať pro provoz turbín se stlačeným vzduchem.

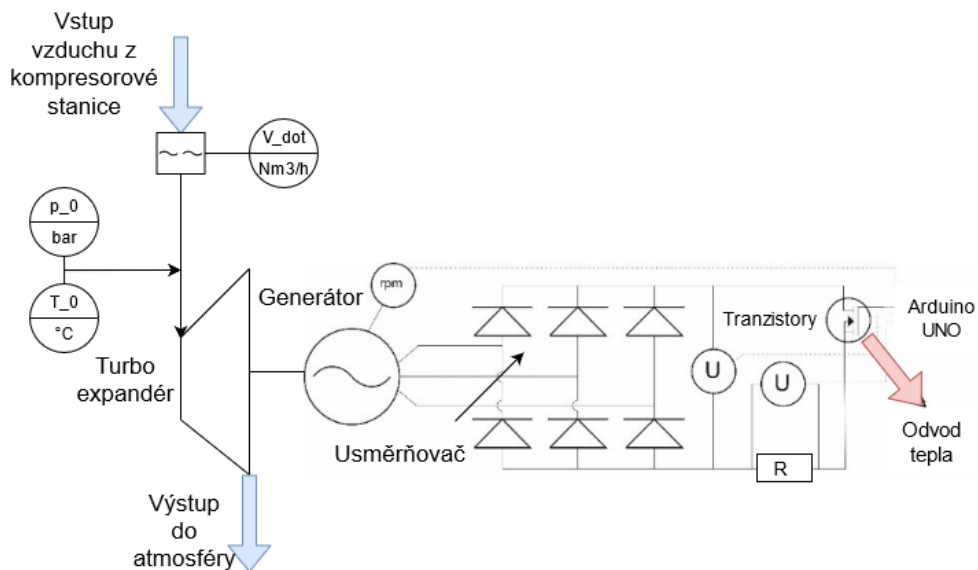
Experimentální měření spočívalo v postupném navyšování a snižování otáček stroje od 3000 rpm⁸ do 12 000 rpm s krokem 500 při daném tlaku stlačeného vzduchu na vstupu, čímž pro jeden tlak bylo získáno 36 ustálených stavů, pro něž byla matematickým modelem následně vyhodnocována účinnost turboexpandéru. *Ustálený stav* je takový stav, kdy se proměnné, které definují chování systému nemění. V případě měřených proměnných v prováděném experimentu však vždy hodnoty oscilují okolo požadované hodnoty a navíc jsou zatíženy šumem vlivem nedokonalého elektrického stínění signálu. Z tohoto důvodu jsou vyhodnocované stavy ve skutečnosti spíše *kvazi-ustálené*. Nicméně pro usnadnění čitelnosti snad čtenář dovolí, aby autorka používala dále v rámci práce termín *ustálený stav*. Jelikož byl experiment opakován pro čtyři různé tlakové poměry, bylo pro jednu kombinaci sestavy stator-rotor obdrženo 144 ustálených stavů. Celkem bylo změřeno 20 různých konfigurací, což činí celkový počet ustálených stavů, které byly vyhodnocovány roven 2 880. [55]

Experimentální trať je zobrazena na Obrázku 2.1. Na vstupu je stlačený upravený vzduch z kompresorové stanice, který je na požadovaném tlaku vháněn do turboexpandéru, kde koná práci zpracováváním tlakového spádu mezi tlakem vzduchu a atmosférou. Před vstupem do turboexpandéru jsou změřeny parametry proudícího média - objemový tok vzduchu, jeho teplota a tlak. Zároveň je v laboratoři měřena atmosférická teplota a tlak, čímž lze získat z látkových vlastností maximální potenciál energie obsažený v proudě stlačeného vzduchu. Turboexpandér je hřídelí, na které jsou měřeny otáčky stroje, spojen s BLDC⁹ generátorem. Detaily o experimentální trati jsou mimo jiné popsány v [56].

Elektrický výkon z generátoru je pak usměrněn v usměrňovači a následně je změřeno jeho napětí. Proud je měřen z úbytku napětí na rezistoru o známém odporu. Elektrický proud po tomto měření teče do tranzistoru, který je provozován v lineárním režimu a reguluje tím „mařené množství výkonu“, respektive zátěž generátoru. Proud dále putuje do systému odvodu tepla, který se sestává

⁸Rotation per minute

⁹Bezkartáčový stejnosměrný motor



Obrázek 2.1: Schéma experimentální tratě pro měření účinnosti turboexpandérů; upraveno z [56]

z rezistorů a vodního chlazení z PC¹⁰. Otevření tranzistoru je řízeno externě za pomoci programu v ArduinoUNO¹¹. Ten umožňuje sběr a zápis dat ze systému měření elektrického výkonu turboexpandéru a zároveň regulaci výkonu právě prostřednictvím řízení otevření tranzistoru. Software v Arduino je schopen regulovat zátěž buďto v režimu na konstantní otáčky, kdy se snaží nastavit pomocí jednoduchého regulačního obvodu s negativní zpětnou vazbou a *PID regulátorem* držet konstantní otáčky stroje, nebo otevřenou smyčkou v režimu na konstantní proud. PID regulátor je spojitý regulátor systémů složený z proporcionální, integrální a derivační složky. Princip funkce regulátoru je založen na regulaci tzv. regulační odchylky - rozdílem mezi aktuální a požadovanou veličinou a výpočtu akční veličiny, která provádí zásah do samotné regulované soustavy. [57].

Sběr dat je navíc komplikovaný tím, že probíhá na dvou oddělených zařízeních. Data ohledně elektronického výkonu stroje jsou ukládána prostřednictvím Arduino a posílána do PC, zatímco data charakterizující tok stlačeného vzduchu jsou sbírána přes Labview do laboratorního počítače. Kompilace dat přes časový záznam pak musel být prováděn ručně.

Příklad vizualizace naměřených raw dat je ukázán na Obrázcích 2.2 a 2.3. Obrázek 2.2 znázorňuje průběh otáček a výkonu stroje v čase. Z tohoto grafu operátor vyhodnocoval naznačené ustálené stavy při stávajícím způsobu řešení. Na Obrázku 2.3 je vidět právě výše zmiňovaný mód řízení stroje „Constant

¹⁰Personal computer

¹¹Arduino Uno je vývojová deska s mikrokontrolérem

2.1. Popis problému a současný stav řešení



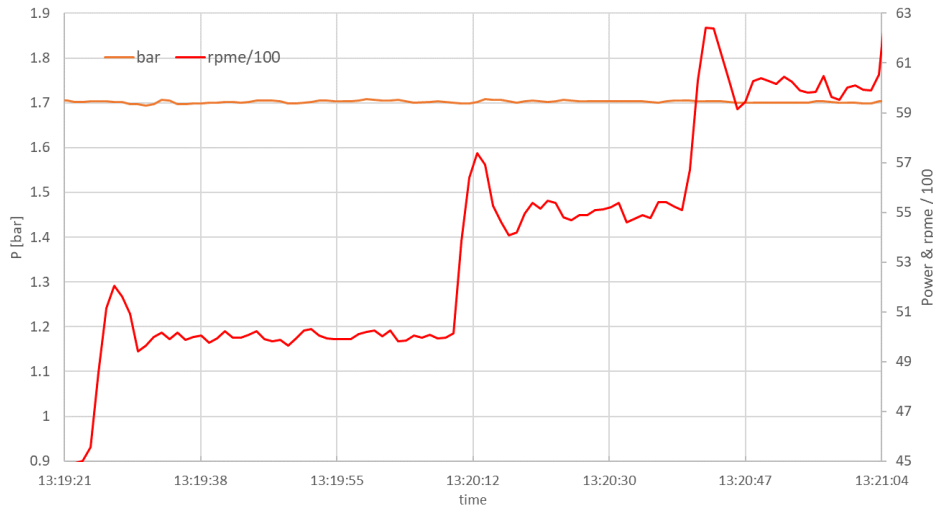
Obrázek 2.2: Příklad raw dat pro jeden tlakový poměr - otáčky a elektrický výkon - a znázornění ustáleného stavu

RPM“, ve kterém je pomocí zpětnovazební smyčky s PID regulátorem řízena zátěž generátoru tak, aby se točil na požadovaných otáčkách. Na grafu je mimo jiné vidět funkce PID regulátoru, která minimalizuje regulační odchylku a provádí zásah do řízené soustavy.

Současný stav vyhodnocování těchto raw dat byl poloautomatický, a to sice takový, že ustálené stavy byly z časové stopy vybírány manuálně za pomoci vizuální kontroly operátorem vyčítáním z grafu otáček stroje v čase. Po volbě počátku a konce ustálených stavů se naměřené hodnoty mezi těmito časovými značkami zprůměrovaly v čase a proběhl výpočet za pomoci matematického modelu, který spočetl stavové vlastnosti proudu vzduchu před a za turbínou a zejména přepočtl změřený elektrický výkon na mechanický výkon stroje. Porovnáním naměřeného mechanického výkonu stroje s teoreticky maximálně možným dosažitelným výkonem obsaženým v proudu vzduchu pak byla získána účinnost stroje. Ta je funkcí mnoha parametrů, ale pro jednu geometrii a jednu sestavu zejména závisí na otáčkách a tlakovém poměru před a za turbínou. Tyto křivky byly vyneseny manuálně v MS Excel do grafů a operátor zhodnotil kvalitu sestavy, resp. aditivní technologie výroby z povahy účinnostních charakteristik a z maximální dosažené účinnosti stroje. Tento proces je pro lepší pochopení zobrazen na Diagramu 2.4.

V budoucnu bude tato experimentální metoda dále používána pro testování nových generací turboexpandérů, avšak časová náročnost spojená s manuálním vyhodnocováním dat z měření se ukázala být nezvladatelná.

Nejnáročnější položkou při této činnosti je manuální výběr ustálených



Obrázek 2.3: Detail vzorku dat z měření - ukázka regulace otáček stroje v módu Constant RPM

stavů, kdy operátor ručně za pomoci odečítání z grafu volil časovou stopu počátku a konce ustáleného stavu a hodnoty mezi těmito limitami se následně zprůměrovaly a byly použity pro výpočet matematickým modelem a vyhodnocení účinnosti stroje. Pro jednu sestavu turbíny je časová náročnost vyhodnocení dat rovna přibližně dvěma a půl hodinám času. Při dvaceti sestavách tedy operátor stráví padesát hodin, neboli více jak jeden pracovní týden času pouze výběrem ustálených stavů a vykreslováním dat do grafů. Bylo tedy zřejmé, že pro budoucí potřeby musí být tento proces alespoň do určité míry automatizován. To je mimo jiné cílem této závěrečné práce.

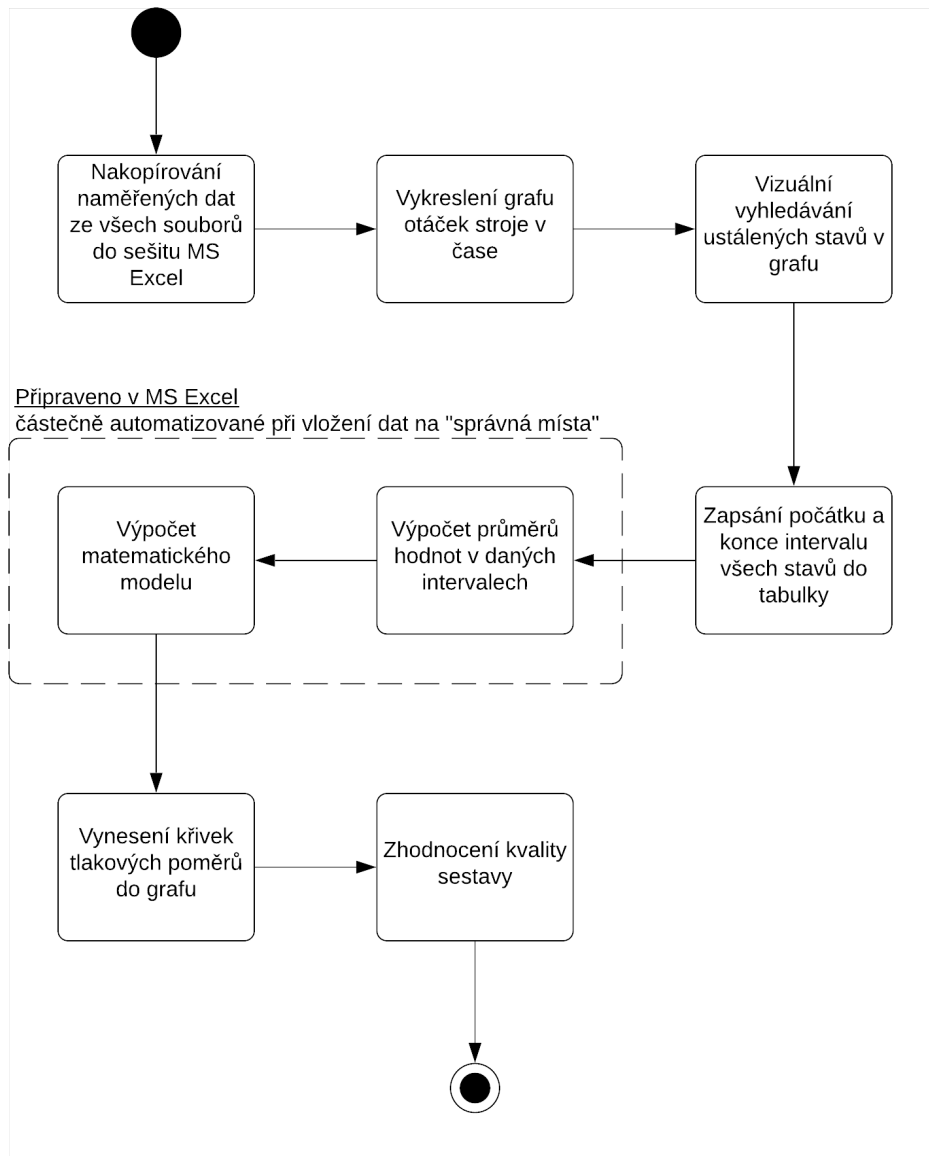
2.2 Návrh řešení

Tato sekce se zabývá návrhem systému, kde je nejprve analyzováno co by mělo být automatizováno a do jaké míry a následně je pomocí UML diagramů navrhnut samotný systém.

2.2.1 Volba míry automatizace

Správně zvolit míru automatizace není jednoduchý úkol, je zapotřebí brát v úvahu mnoho proměnných. Podkladem pro volbu úrovně automatizace autorka použila framework popsany v Sekci 1.1.5 a zobrazený na Diagramu 1.2.

Prvním krokem je určit co by mělo být automatizováno. Existuje několik úkonů v procesu, které by byly bývali dobrými kandidáty pro automatizaci.



Obrázek 2.4: Současný stav řešení pomocí flow chart

Analyzujeme-li proces krok po kroku, získáme následující body možné automatizace:

- Měření a zápis - automatizace změny počtu otáček za minutu
- Přenos naměřených dat - automatické sdílení výsledků pomocí cloudu
- Kompletace dat - automatické sjednocení více souborů z měření a vyloučení duplicit
- Zpracování dat - automatické výpočty a nalezení ustálených stavů
- Zobrazení dat - automatické vykreslení grafů

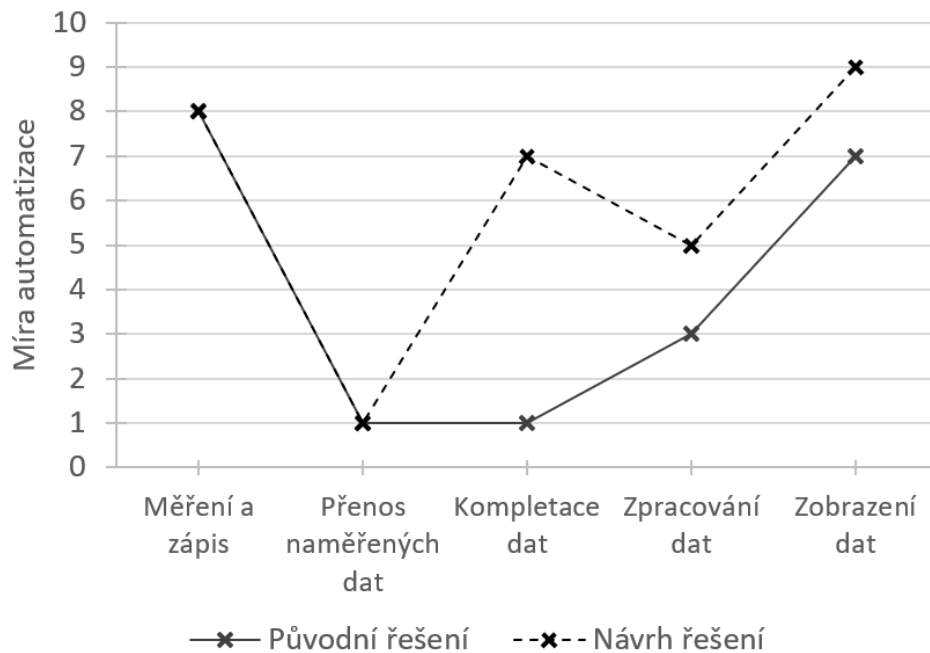
U všech zmíněných bodů, by bylo možno dosáhnout vysoké míry automatizace či za ideálních podmínek dokonce plné automatizace. Zda by bylo zavedení plné automatizace vhodným a ekonomicky opodstatněným řešením daného problému však není meritem této práce. Jak je zřejmé z popisu současného řešení výše, nejpalčivějším problémem je pro výzkumné pracovníky zdoluhavé manuální vybírání ustálených stavů, které bude nadále podrobeno zkoumání z perspektivy automatizace.

Vyznačování ustálených stavů zahrnuje kompletaci dat, tedy vzájemné propojení souborů přicházejících z prostředí Labview a Arduino. Tyto data jsou následně zpracovávána a na základě těchto dat jsou určovány ustálené stavy a následně vypočítávány požadované hodnoty, které jsou zobrazovány do grafu. Automatizace bude tedy aplikována na kompletaci dat, zpracování dat a jejich zobrazení.

Všechny tyto úkony spadají do části *Získávání informací* v modelu, ale i tak může být každá část automatizována na různé úrovni. V tomto kroku je třeba identifikovat stupně automatizace pro jednotlivé úkony a následně zhodnotit tyto návrhy na základě primárních a sekundárních hodnotících kritérií a v případě potřeby změnit jak „to, co automatizovat“ tak i „jak to automatizovat“.

Sjednocení souborů z různých zdrojů je možné automatizovat na vysoké úrovni, člověk musí pouze dodat potřebné soubory. Zpracování dat může probíhat automatizovaně, nicméně v případě vyhodnocování ustálených stavů je potřeba asistence člověka. Algoritmus nemusí být natolik spolehlivý, aby detekoval stavy správně v každé situaci. Proto má uživatel právo veta a možnost ustálené stavy zadat ručně, což by se dalo nazvat jako napůl cesty k plné automatizaci - tedy úrovni 5. Grafické znázornění dat může být čistě v režii automatizace.

Úroveň automatizace pro jednotlivé úkoly byla určena na stupnici 0 až 10, kde 0 vyjadřuje žádnou automatizaci a 10 plnou automatizaci. Ohodnocen je jak původní stav, tak i navržená změna, obojí je zaznamenáno v Grafu 2.5.



Obrázek 2.5: Znázornění stupňů automatizace v původním a novém řešení

2.2.1.1 Hodnocení kritérií pro daný návrh

Mentální pracovní zátěž by měla být signifikantně snížena, protože operátor nemusí trávit dlouhé hodiny zpracováním dat a zapojuje se pouze při kontrole vyznačených ustálených stavů, což by mělo mít taktéž pozitivní vliv na produktivitu člověka.

Povědomí o situaci by mělo být zachováno díky potřebě vizuální kontroly ustálených stavů operátorem. Zároveň jsou operátorovi k dispozici původní data a vizualizovaná data pomocí grafu. Hrozí však, že při vysoké spolehlivosti návrhů může operátor ztratit ostražitost.

Spokojení se s navrženými stavy je vždy rizikem. Bude záležet na pracovitosti a poctivosti operátora, do jaké míry bude kontrolovat validitu stavů. Operátor bude obeznámen s úrovní nespolehlivosti, tak aby dokázal vyhodnotit reálnou potřebu kontroly.

Degradace schopností v tomto případě nehrozí, protože se jedná o automatizaci jednotvárné práce, u které není zapotřebí téměř žádné speciální dovednosti.

Z primárních hodnotících kritérií vyplývá, že největší riziko plyne z možnosti nadměrné důvěry operátora v automatickou detekci stavů. Toto riziko nelze

zcela eliminovat, ale může být částečně potlačeno znalostí nespolehlivosti systému a díky přístupu k původním datům.

Spolehlivost automatizace je velkou neznámou, v tuto chvíli nelze s určitostí říct jak dobře či špatně bude algoritmus vyhodnocovat ustálené stavy. Proto je důležité, že má operátor přístup k původním datům a věnuje jim dostatečnou pozornost.

Náklady na výsledky rozhodnutí v případě chybné detekce ustáleného stavu by neměli být vysoké díky vizuální kontrole operátora. Náklady by měli vznikat pouze v minimálním počtu případů, tedy pravděpodobnost vzniku chyby je nízká.

S přihlédnutím ke všem hodnotícím kritériům, se autorka rozhodla daný návrh realizovat. Odhalená rizika by se měla vyskytovat s nízkou pravděpodobností a jejich dopad by neměl být nijak závažný vzhledem k faktu, že se jedná o výzkum, kde je vysoká pravděpodobnost odhalení chyby ještě před reálným využitím experimentů.

2.2.2 Návrh systému

Tato část práce je zaměřena na návrh systému v závislosti na požadavcích, co a jak má systém vykonávat. Tyto požadavky byly určeny na základě očekávání budoucích uživatelů a diskuse uvedené v sekci 2.2.1.

Návrh je proveden pomocí UML jazyka, neboli unifikovaného modelovacího jazyka, který slouží především k vizualizaci navrhovaného systému. [58] Nejedná se tedy o programovací jazyk, ale o způsob zachycení chování a struktury systému pomocí speciálních diagramů.

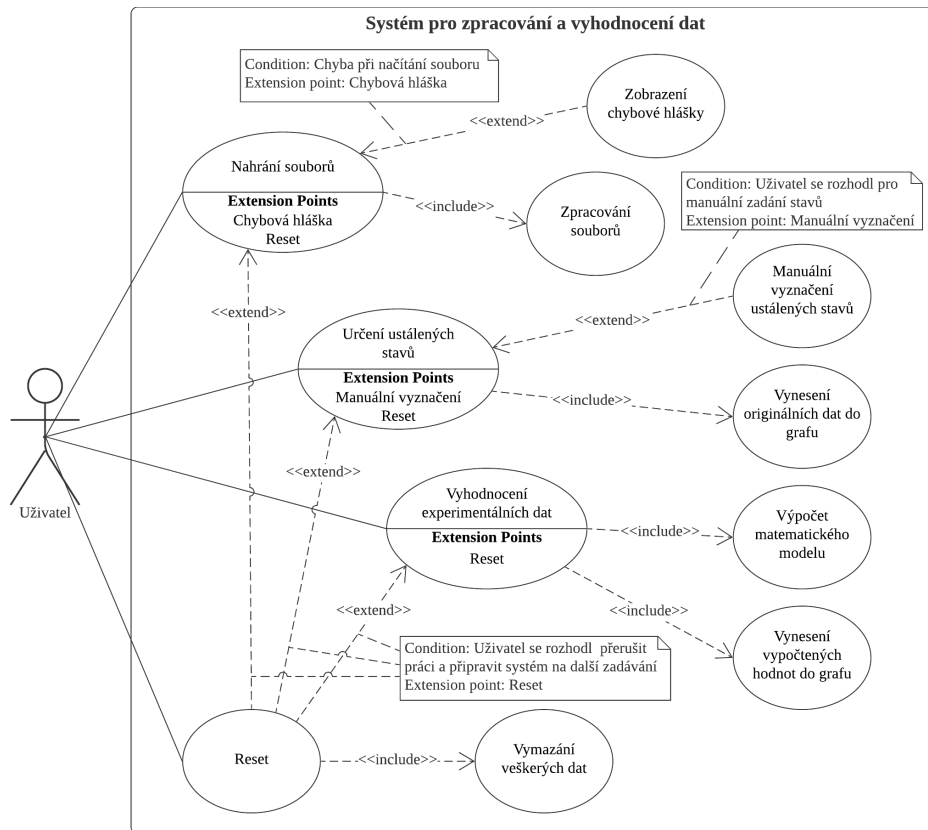
K vymezení interakce člověka se systémem je použit diagram případů užití, tzv. use case diagram. Tento diagram nezobrazuje detaily jednotlivých případů užití, pouze shrnuje vztahy mezi aktéry, případy užití a systémem. Diagram nezachycuje pořadí kroků k naplnění případu, této problematice se autorka věnuje v následných sekcích práce.

V případě systému pro zpracování a vyhodnocení dat (Diagram 2.6) máme pouze jednoho aktéra - uživatele systému. Komunikace mezi člověkem a systémem probíhá napřímo, jedná se tedy o konečnou aplikaci pro uživatele a bude zapotřebí navrhnout uživatelsky přívětivé GUI¹².

Hlavním případům užití z Diagramu 2.6 jsou věnovány samostatné podkapitoly, kde jsou popsány a znázorněny graficky. Jedná se o následující:

- Načtení souborů
- Určení ustálených stavů

¹²Graphical User Interface

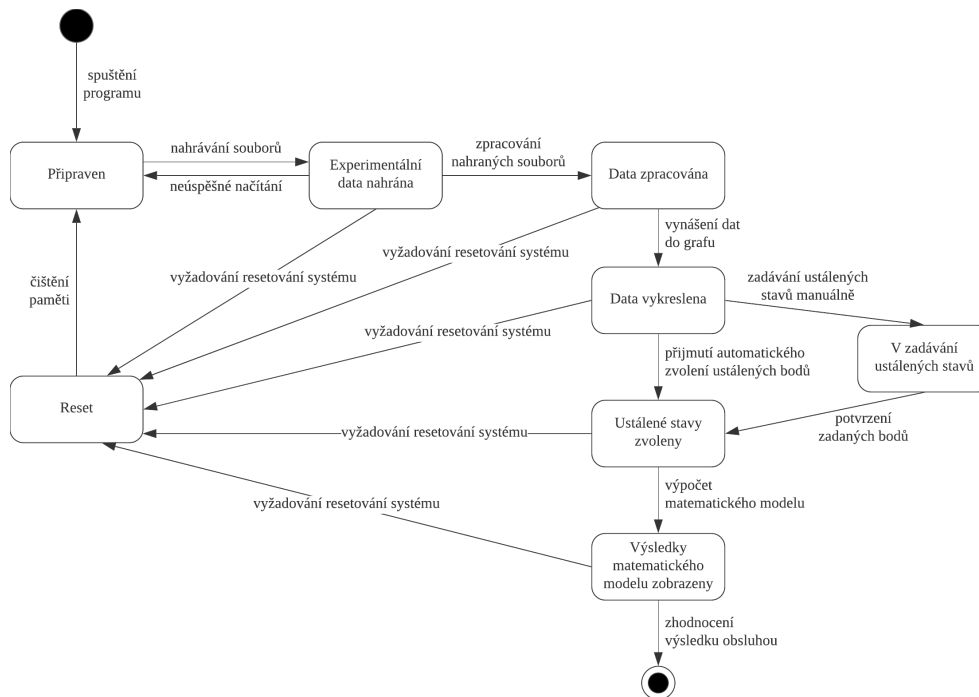


Obrázek 2.6: Diagram případů užití systému pro zpracování a vyhodnocení dat

- Vyhodnocení experimentálních dat
- Reset

Stavy, kterých je systém schopen nabývat jsou identifikovány na stavovém Diagramu 2.7. Základní stav, tj. ten stav ve kterém se systém nachází po zapnutí je nazván *Připraven* a existují 3 možné cesty jak se do tohoto stavu dostat - po zapnutí programu, pokud nastane chyba při načítání souborů se vstupními daty nebo na žádost uživatele. Následuje stav *Experimentální data nahrána*, kterého lze dosáhnout pouze úspěšným načtením souborů. V případě úspěchu může systém postoupit do stavu *Data zpracována*, kde jsou data ze všech souborů spárována a uložena na základě časové značky.

Uživatel si může spárovaná data nechat vykreslit a přivést tak systém do stavu *Data vykreslena*. V tomto stavu jsou do grafu vyneseny hodnoty experimentálních dat, na které byl aplikován algoritmus na hledání ustálených stavů. Tyto ustálené stavy jsou označeny, aby mohl uživatel zhodnotit jejich správnost. Pokud uživatel není spokojený s automaticky generovanými inter-



Obrázek 2.7: Stavový digram - systém

valy, přechází systém do *V zadávání ustálených stavů*, kde si pomocí myši může intervaly upravit. Jakmile uživatel potvrdí korektnost intervalů znázorňující ustálené stavy, systém přechází do *Ustálené stavy zvoleny*.

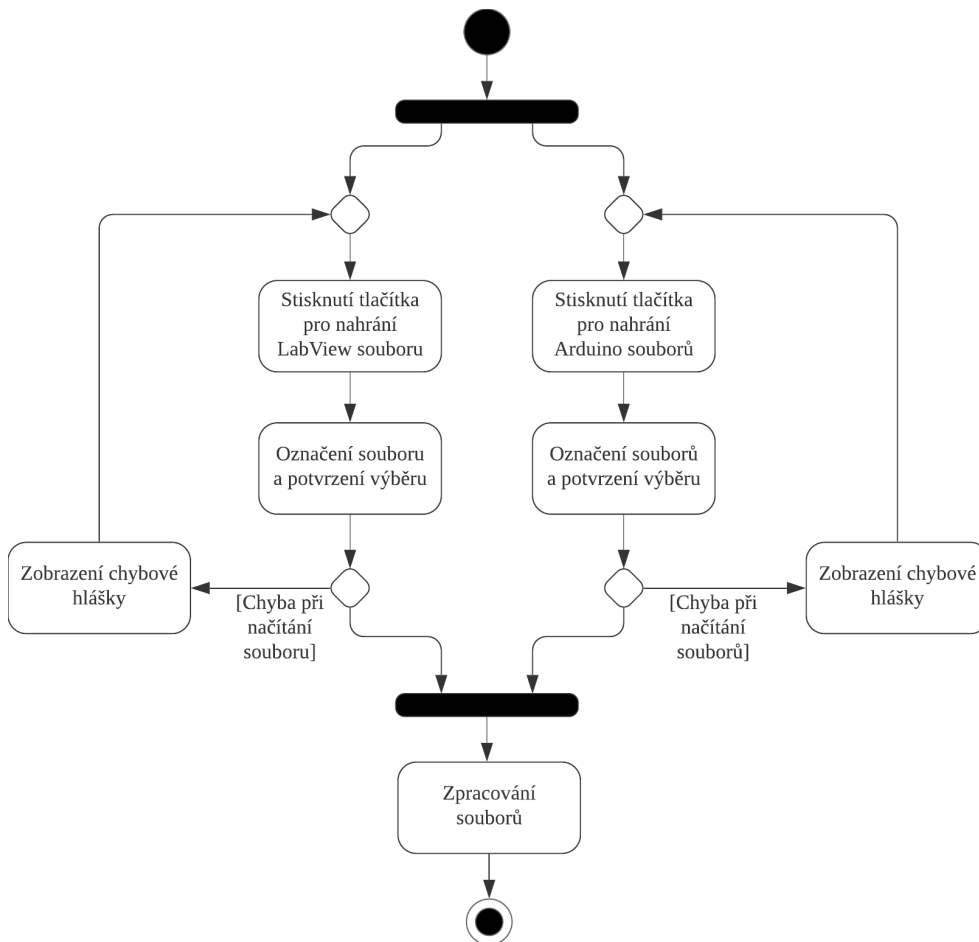
Nejsložitější úkol programu, výpočet matematického modelu, nás následně dovede do stavu *Výsledky matematického modelu zobrazeny*. Jedná se o stav, který uživateli poskytne nepostradatelné informace k zhodnocení výsledků měření a všechny předcházející kroky jsou vykonávány se záměrem dostat se do tohoto stavu.

Téměř v každém stavu se může uživatel z libovolného důvodu rozhodnout začít práci od začátku, což je umožněno s využitím stavu *Reset*, který systém vyčistí od dat a docílí stavu *Připraven*.

2.2.2.1 Nahrání souborů

Prvním krokem k využití programu k účelu, pro který je navrhnut je načtení souborů z měření. Konkrétně soubory vytvořené pomocí systému Labview a Arduino. Na pořadí, ve kterém jsou soubory načítány nezáleží, což je na Diagramu 2.8 znázorněno pomocí paralelních větví. Nicméně je potřeba počkat na načtení všech souborů, aby bylo umožněno zpracování dat a uložení do vnitřní paměti programu.

Jednotlivé kroky a kdo je vykonává jsou popsány v Tabulce 2.1. Sekvence



Obrázek 2.8: Diagram případu užití UC001: Nahrání souborů

kroků 1 až 4 a kroků 5 až 8 ze základního scénáře jsou vzájemně zaměnitelné. Alternativní scénář nastává v případě neúspěšného nahrání souboru, kdy je zobrazena chybová hláška a je zapotřebí nějaké kroky opakovat. V každém kroku základního scénáře může nastat rozšíření o alternativní scénář, kdy se uživatel rozhodl uvést program do stavu připravenému k dalšímu zadávání.

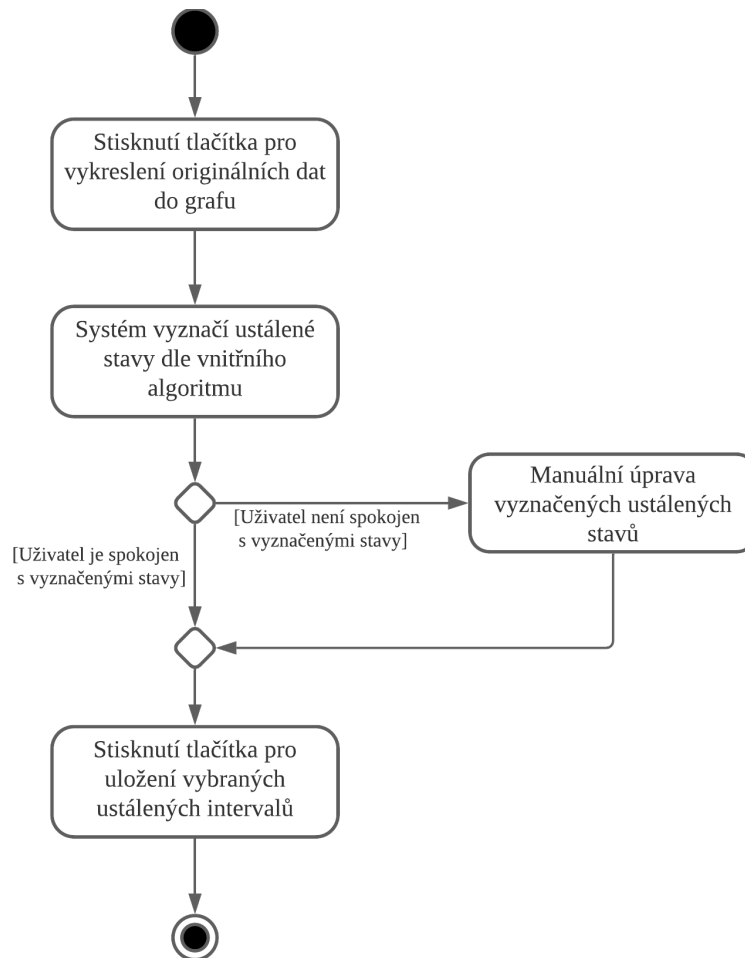
2.2.2.2 Určení ustálených stavů

Vymezení ustálených stavů je pro následný výpočet kritické, proto je zapotřebí věnovat mu dostatečnou pozornost. V rámci zvýšení produktivity operátora a urychlení procesu určování ustálených stavů jsou navrženy 2 možnosti jak k tomuto úkolu přistupovat.

Program sám detekuje ustálené stavy pomocí implementovaného algoritmu (Sekce 2.3.2) a nabídne je uživateli. Ten se rozhoduje zda návrh přijme, tak

Tabulka 2.1: Popis případu užití UC001: Nahrání souborů

Název případu užití	Nahrání souborů		
Identifikace případu užití	UC001		
Cíl případu užití	Načíst všechny soubory vytvořené během měření.		
Aktér	Uživatel		
Vstupní podmínky	Program je otevřený a připravený k použití.		
Základní scénář¹³	Krok	Role	Akce
	1	uživatel	Stiskne tlačítko pro načtení Labview souboru.
	2	system	Zobrazí průzkumník souborů.
	3	uživatel	Vybere požadovaný soubor a potvrdí výběr souboru.
	4	system	Nahraje soubor do paměti.
	5	uživatel	Stiskne tlačítko pro načtení Arduino souborů.
	6	system	Zobrazí průzkumník souborů.
	7	uživatel	Vybere požadované soubory a potvrdí výběr souboru.
	8	system	Nahraje soubory do paměti.
	9	system	Zpracuje načtené soubory jak je popsáno v Sekci 2.3.
Alternativní scénář	Krok	Role	Akce
	4.1	system	Zobrazí chybovou hlášku o neúspěšném pokusu nahrát soubor.
	4.2	uživatel	Vrací se ke kroku 1 hlavního scénáře.
	8.1	system	Zobrazí chybovou hlášku o neúspěšném pokusu nahrát soubory.
	8.2	uživatel	Vrací se ke kroku 5 hlavního scénáře.
	1-9	uživatel	Může v každém kroku scénáře přejít na případ užití UC004 v Tabulce 2.4.
Výstupní podmínky	Soubory jsou nahrány a zpracovány programem.		



Obrázek 2.9: Diagram případu užití UC002: Určení ustálených stavů

jak je nebo stavy manuálně upraví. Tento proces je znázorněn na Obrázku 2.9 a popsán v Tabulce 2.2 jako alternativní scénář. I v tomto případě užití může nastat rozšíření základního scénáře v každém jeho kroku a navrácení systému do počátečního stavu.

2.2.2.3 Vyhodnocení experimentálních dat

Diagram 2.10 zobrazuje proces zahrnující výpočet a zobrazení výsledků matematického modelu. Vykreslení výsledků do grafu a vyplnění tabulek jsou úkony, které se mohou provést v libovolném pořadí v závislosti na konečné implementaci.

Základní scénář je k nahlédnutí v Tabulce 2.3. Stejně jako v předchozích případech i zde může v každém kroku základního scénáře uživatel přerušit práci s programem a resetovat ho do výchozího stavu.

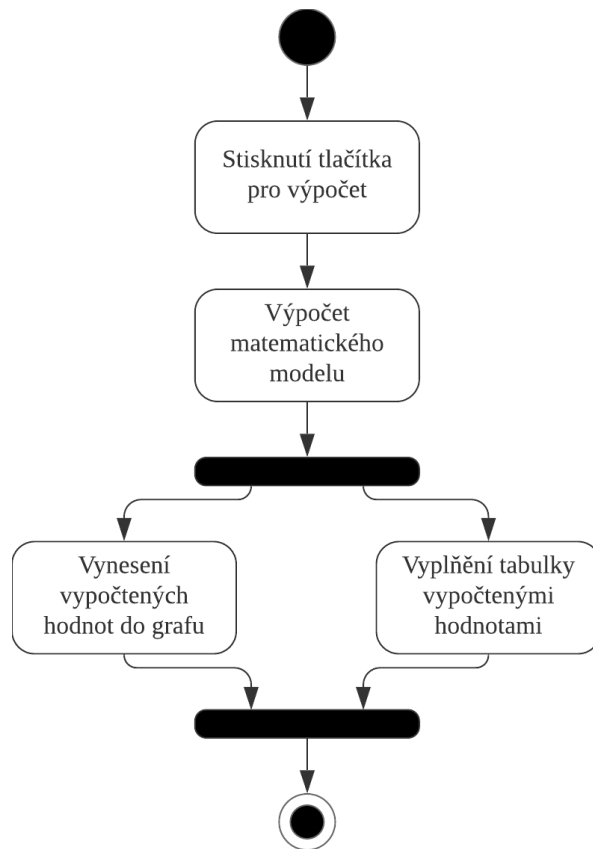
Tabulka 2.2: Popis případu užití UC002: Určení ustálených stavů

Název případu užití	Určení ustálených stavů		
Identifikace případu užití	UC002		
Cíl případu užití	Vybrat a uložit do paměti mezní body ustálených stavů.		
Aktér	Uživatel		
Vstupní podmínky	Soubory z měření jsou úspěšně nahrané a zpracované v rámci programu.		
Základní scénář	Krok	Role	Akce
	1	uživatel	Stiskne tlačítko pro vykreslení originálních dat.
	2	system	Určí ustálené stavy pomocí algoritmu (Sekce 2.3.2) a vykreslí originální data s vypočtenými ustálenými stavy do grafu.
	3	uživatel	Potvrdí vybrané ustálené intervaly stisknutím tlačítka.
	4	system	Uloží body vymezující ustálené stavy do paměti.
Alternativní scénář	Krok	Role	Akce
	3.1	uživatel	Pokud není spokojený s navrženými ustálenými stavy, opraví je manuálně.
	1-4	uživatel	Může v každém kroku scénáře přejít na případ užití UC004 v Tabulce 2.4.
Výstupní podmínky	Body vymezující ustálené stavy jsou identifikovány a uloženy do paměti programu.		

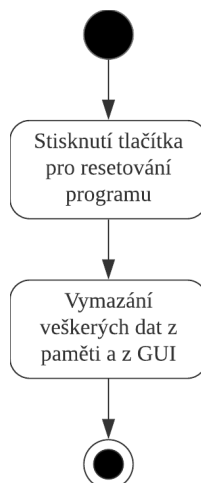
2.2.2.4 Reset

Tento případ užití rozšiřuje téměř všechny ostatní případy užití. Jedná se o dobrovolné přerušování práce uživatele a resetování prostředí programu, tak aby bylo možné začít novou práci s programem. Proces je vyobrazen na Diagramu 2.11 a popsán v Tabulce 2.4.

Reset umožňuje uživateli přerušování z mnoha důvodů, např. pokud uživatel zapomene nahrát všechny soubory nebo se dopustí jiné chyby, nemusí tak ukočit celý program, ale stačí prostředí jen resetovat do výchozího stavu. Tato funkcionality je důležitá z pohledu uživatelské přívětivosti programu.



Obrázek 2.10: Diagram případu užití UC003: Vyhodnocení experimentálních dat



Obrázek 2.11: Diagram případu užití UC004: Reset

Tabulka 2.3: Popis případu užití UC003: Vyhodnocení experimentálních dat

Název případu užití	Vyhodnocení experimentálních dat												
Identifikace případu užití	UC003												
Cíl případu užití	Získání účinnostních charakteristik.												
Aktér	Uživatel												
Vstupní podmínky	Soubory z měření jsou úspěšně zpracované, stejně tak body vymezující ustálené stavy.												
Základní scénář	<table border="1"> <thead> <tr> <th>Krok</th> <th>Role</th> <th>Akce</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>uživatel</td> <td>Stiskne tlačítko pro výpočet matematického modelu</td> </tr> <tr> <td>2</td> <td>system</td> <td>Vypočte matematický model popsaný v Sekci 2.3.1.</td> </tr> <tr> <td>3</td> <td>system</td> <td>Zobrazí vypočtené hodnoty v grafu a vypíše je do tabulek.</td> </tr> </tbody> </table>	Krok	Role	Akce	1	uživatel	Stiskne tlačítko pro výpočet matematického modelu	2	system	Vypočte matematický model popsaný v Sekci 2.3.1.	3	system	Zobrazí vypočtené hodnoty v grafu a vypíše je do tabulek.
Krok	Role	Akce											
1	uživatel	Stiskne tlačítko pro výpočet matematického modelu											
2	system	Vypočte matematický model popsaný v Sekci 2.3.1.											
3	system	Zobrazí vypočtené hodnoty v grafu a vypíše je do tabulek.											
Alternativní scénář	<table border="1"> <thead> <tr> <th>Krok</th> <th>Role</th> <th>Akce</th> </tr> </thead> <tbody> <tr> <td>1-3</td> <td>uživatel</td> <td>Může v každém kroku scénáře přejít na případ užití UC004 v Tabulce 2.4.</td> </tr> </tbody> </table>	Krok	Role	Akce	1-3	uživatel	Může v každém kroku scénáře přejít na případ užití UC004 v Tabulce 2.4.						
Krok	Role	Akce											
1-3	uživatel	Může v každém kroku scénáře přejít na případ užití UC004 v Tabulce 2.4.											
Výstupní podmínky	Experimentální výsledky jsou vypočteny a zobrazeny uživateli v tabulce a částečně zobrazeny v grafu.												

Tabulka 2.4: Popis případu užití UC004: Reset

Název případu užití	Reset									
Identifikace případu užití	UC004									
Cíl případu užití	Připravení programu na zpracování nových dat.									
Aktér	Uživatel									
Vstupní podmínky	Program je otevřený.									
Základní scénář	<table border="1"> <thead> <tr> <th>Krok</th> <th>Role</th> <th>Akce</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>uživatel</td> <td>Stiskne tlačítko na resetování systému.</td> </tr> <tr> <td>2</td> <td>system</td> <td>Vymaže veškerý obsah z paměti včetně grafů a tabulek z GUI.</td> </tr> </tbody> </table>	Krok	Role	Akce	1	uživatel	Stiskne tlačítko na resetování systému.	2	system	Vymaže veškerý obsah z paměti včetně grafů a tabulek z GUI.
Krok	Role	Akce								
1	uživatel	Stiskne tlačítko na resetování systému.								
2	system	Vymaže veškerý obsah z paměti včetně grafů a tabulek z GUI.								
Výstupní podmínky	V paměti nejsou uložena žádná data a program je připraven k použití.									

2.3 Realizace

Výstupem z experimentálního měření účinnosti turboexpandéru je několik textových souborů. Jeden textový (ukázka souboru v příloze, Obrázek C.1) soubor je vygenerován z celého měření systémem Labview, který pro každou vteřinu měření zaznamenává:

- aktuální čas [hh:mm:ss]
- otáčky turbíny za minutu [$ot. \cdot m^{-1}$]
- točivý moment [Nm]
- výkon [W]
- objemový průtok vzduchu DN 20 [$m_N^3 \cdot h^{-1}$]
- objemový průtok vzduchu průtok DN 40 [$m_N^3 \cdot h^{-1}$]
- převodník stlačeného vzduchu 1 [bar]
- převodník stlačeného vzduchu 2 [bar]
- převodník stlačeného vzduchu 3 [bar]
- převodník stlačeného vzduchu 4 [bar]
- převodník stlačeného vzduchu 5 [bar]
- převodník stlačeného vzduchu 6 [bar]
- teplotu stlačeného vzduchu [°C]
- teplotu okolní atmosféry [°C]
- teplotu rosného bodu [°C]

Na každém řádku souboru se tedy nachází 15 naměřených hodnot. Výjimku tvoří pouze první 2 řádky, které tvoří hlavičku dokumentu - název hodnoty a její jednotky.

Tento soubor je třeba nahrát do navrženého systému a zpracovat, tak jak je uvedeno dále. Nejprve se odebere hlavička souboru a pak je každý řádek rozdělen na 15 hodnot. Protože se ale jedná o textový soubor, jsou data nahrána jako řetězce (datový typ string), což není vhodný datový typ pro matematické operace. Číselné hodnoty je třeba převést na datový typ, který podporuje matematické operace v pohyblivé desetinné čárce (double) a čas je z formátu hh:mm:ss překalkulován na sekundy. Takto upravené hodnoty jsou uloženy do matice, ve stejném pořadí, ve kterém byly hodnoty v souboru. Výsledná matice má tedy rozměr 15 krát počet záznamů, kde každý sloupec vyjadřuje danou proměnnou pro jednotlivý záznam.

Na rozdíl od souboru generovaného pomocí systému Labview, Arduino při každém přerušení měření vygeneruje textový soubor (ukázka souboru v příloze, Obrázek C.2) s naměřenými hodnotami a při opětovném zahájení měření začne zapisovat do souboru nového. Ze strany Arduina se tedy může očekávat více souborů vstupujících do programu. Hodnoty jsou stejně jako u předešlého systému zaznamenávány po vteřinách, kde jsou zapsány hodnoty:

- aktuální čas [dd:mm:yyyy hh:mm:ss]
- napětí [V]
- elektrický proud [A]
- elektrický výkon [W]
- otáčky turbíny měřené z elektrického proudu [$ot. \cdot m^{-1}$]
- otáčky turbíny měřené optickým senzorem [$ot. \cdot m^{-1}$]

I v tomto případě přináší textový soubor několik komplikací a tedy i v tomto případě je nutné odebrat hlavičku souboru a data převádět na vhodný datový typ. Z časové značky je odstraněno datum a čas je převeden na sekundy stejně jako u časové hodnoty z Labview.

Tento proces úpravy dat je proveden se všemi doloženými textovými soubory a následně jsou data seříděna podle časové značky a uložena do matice. Jedná se o matici s 6 sloupci a počtem řádků rovným počtu záznamů v souboru. Hodnoty ve sloupcích matice reprezentují výše zmíněné veličiny: aktuální čas (převedeno na sekundy), napětí, elektrický proud, elektrický výkon, otáčky turbíny měřené z elektrického proudu a otáčky turbíny měřené optickým senzorem. Krok řazení je kritický pro další zpracování dat, v případě špatného pořadí by byli výsledky nevalidní a tudíž nepoužitelné.

Dalším krokem, kterým je třeba se zabývat je detekce tzv. ustáleného stavu. Tento stav se určuje na základě otáček turbíny měřené z elektrického proudu. Ustálený stav je prvně detekován automaticky pomocí algoritmu v době zpracování souborů. Takto zvolené stavy jsou vykresleny do grafu závislosti počtu otáček na čase, kde je operátor vizuálně kontroluje. Pokud je operátor nespokojen s volbou ustálených stavů, přichází na řadu manuální zadávání, které je obdobné jako v současném řešení. Operátor aktivuje proces zadávání a pomocí kurzoru vyznačuje postupně sekvenci začátek - konec ustáleného stavu. Díky interaktivnímu prostředí však operátor nemusí dohledávat konkrétní záznamy k danému bodu, to je zajištěno uvnitř programu pomocí detekce kliknutí myši v grafu. Toto pseudo-manuální určování ustálených stavů není ideální, ale i v tomto případě se jedná o progres oproti původnímu řešení.

Když jsou všechny ustálené intervaly určeny, nastává čas pro výpočet matematického modelu. Tento model je popsán v Sekci 2.3.1 včetně všech použitých matematických vzorců.

Posledním úkolem programu je předání vypočtených informací operátorovi, aby mohl vyhodnotit výsledky. Všechny vypočtené hodnoty jsou vyplněny do tabulky, která se nachází na druhé záložce v GUI. V oblasti grafu se vykreslí závislost účinnosti na otáčkách pro různé tlakové poměry a do tabulky lokalizované pod grafem se vypíše pro každý tlakový poměr jeho maximální hodnota výkonu a účinnosti.

Pro komfort uživatele je přidána funkcionalita, která dokáže vymazat veškerý obsah a vrátit aplikaci do stavu před začátkem práce. Předchází se tím problému, že by bylo potřeba pro každé nové zadávání program restartovat.

2.3.1 Popis matematického modelu

Po volbě ustálených stavů probíhá výpočet naměřených dat matematickým modelem pro zhodnocení účinnosti stroje. Matematický model byl vytvořen v předchozím experimentálním měření a byl realizován v prostředí MS Excel s VBA¹⁴ funkcí *GoalSeek* pro iterativní hledání řešení. V rámci této práce byl zjednodušen, značně urychlen (zejména iterační výpočet) a implementován do prostředí MATLAB s grafickou nadstavbou.

Zjednodušeně je matematický model popsán v následující pasáži. Tento model je převzat ze současného stavu řešení, avšak implementován do MATLABu s několika vylepšeními numerického charakteru pro urychlení výpočtu.

Pro vyhodnocení teoreticky maximálně dosažitelného výkonu obsaženého v proudu vzduchu je nutné za pomoci změřeného objemového průtoku vzduchu určit hmotnostní tok z rovnice 2.1.

$$\dot{m}_{\text{air}} = \dot{V}_{\text{air}} \cdot \rho_{\text{norm}} \left[\text{kg} \cdot \text{s}^{-1} \right] \quad (2.1)$$

Látkové vlastnosti stlačeného vzduchu jsou určeny s pomocí knihovny látkových vlastností REFPROP¹⁵ od amerického Národního institutu standardů a technologie (NIST) implementované do prostředí MATLAB přes Python wrapper. [59] Ze změřeného tlaku a teploty před a za turboexpandérem tak můžou být určeny potřebné látkové veličiny - *enthalpie* a *entropie* proudu. Viz Rovnice 2.2, 2.3, 2.4.

$$h_1 = \text{Enthalpy}(p_1; T_1) \left[\text{kJ} \cdot \text{kg}^{-1} \right] \quad (2.2)$$

$$s_1 = \text{Enthalpy}(p_1; T_1) \left[\text{kJ} \cdot \text{kg}^{-1} \cdot \text{K}^{-1} \right] \quad (2.3)$$

$$h_{3_{is}} = \text{Enthalpy}(p_3; s_1) \left[\text{kJ} \cdot \text{kg}^{-1} \right] \quad (2.4)$$

¹⁴Visual Basic for Applications

¹⁵NIST Reference Fluid Thermodynamic and Transport Properties Database

Z rozdílů látkových vlastností vzduchu před a za turboexpandérem při ideální expanzi vzduchu a hmotnostního toku je určen teoreticky maximálně dosažitelný výkon z Rovnice 2.5.

$$P_{is} = \dot{m}_{air} \cdot (h_1 - h_{3is}) \cdot 1000 [W] \quad (2.5)$$

Tento spočtený teoretický výkon je pak porovnáván s mechanickým výkonem turboexpandéru. Nicméně měřeno je elektrické napětí a proud a je nutné se mechanickému výkonu dopočíst přes účinnost generátoru a systému pro měření elektrického výkonu. Výpočet je znázorněn v Rovnici 2.6.

$$P_{el_{meas}} = U \cdot I [W] \quad (2.6)$$

Dále je ze známého úbytku napětí na diodách usměrňovače určena ztráta na usměrňovači dle Rovnice 2.7.

$$P_{rect} = dU_{rect} \cdot I [W] \quad (2.7)$$

Za účelem zjištění účinnosti generátoru musí být nejprve spočten kroučící moment stroje dle Rovnice 2.8.

$$T = \frac{(P_{el_{meas}} + P_{rect})}{\omega} [N \cdot m] \quad (2.8)$$

Výpočet elektrické účinnosti generátoru η_g závisí na rychlosti otáček za minutu. Pro rozdílná rozpětí otáček je výpočet proveden pomocí jiného polynomu, což je shrnuté v Tabulce 2.5. Polynomy závislosti účinnosti použitého generátoru Turnigy SK3-6364-245 na otáčkách pro dané rozsahy kroučících momentů byly získány za pomoci PHP nástroje s názvem TorqueCalc z webu¹⁶ určeného modelářům využívající křivky účinnosti pro návrh motorů a generátorů zejména pro RC modely. Tyto polynomy jsou obsaženy ve funkci `find_eta_g()`, která na základě počtu otáček vybírá vhodný polynom a vrací vypočtenou hodnotu.

Se znalostí účinnosti generátoru je vyhodnocen mechanický výkon stroje jakožto podíl elektrického výkonu po odečtení ztrát systému měření a zátěže a účinnosti generátoru - Rovnice 2.9.

$$P_{mech} = \frac{(P_{el_{meas}} + P_{rect})}{\eta_g} [W] \quad (2.9)$$

Díky tomu tak konečně může být zhodnocena pro daný ustálený stav účinnost turboexpandéru z podílu mechanického výkonu a teoreticky maximálního výkonu proudu vzduchu z Rovnice 2.10.

$$\eta_{is} = \frac{P_{mech}}{P_{is}} [-] \quad (2.10)$$

¹⁶<https://www.ecalc.ch/torquecalc.php>

Tabulka 2.5: Výpočet η_g pro různý počet otáček turbíny za minutu

Otáčky turbíny za minutu	Polynom pro určení účinnosti elektrického generátoru
(2750, 3250>	$579,36 \cdot M^3 - 605,77 \cdot M^2 + 252,43 \cdot M + 36,178$
(3250, 3750>	$502,54 \cdot M^3 - 556,00 \cdot M^2 + 243,16 \cdot M + 36,547$
(3750, 4250>	$505,48 \cdot M^3 - 557,73 \cdot M^2 + 244,12 \cdot M + 36,221$
(4250, 4750>	$576,50 \cdot M^3 - 616,00 \cdot M^2 + 259,99 \cdot M + 34,584$
(4750, 5250>	$514,01 \cdot M^3 - 563,65 \cdot M^2 + 246,53 \cdot M + 35,35$
(5250, 5750>	$512,15 \cdot M^3 - 561,32 \cdot M^2 + 246,21 \cdot M + 34,968$
(5750, 6250>	$528,89 \cdot M^3 - 573,53 \cdot M^2 + 249,53 \cdot M + 34,239$
(6250, 6750>	$535,17 \cdot M^3 - 577,65 \cdot M^2 + 250,81 \cdot M + 33,632$
(6750, 7250>	$545,02 \cdot M^3 - 583,62 \cdot M^2 + 252,33 \cdot M + 32,97$
(7250, 7750>	$549,28 \cdot M^3 - 585,81 \cdot M^2 + 253,05 \cdot M + 32,326$
(7750, 8250>	$567,64 \cdot M^3 - 597,69 \cdot M^2 + 255,79 \cdot M + 31,521$
(8250, 8750>	$577,44 \cdot M^3 - 603,04 \cdot M^2 + 256,94 \cdot M + 30,805$
(8750, 9250>	$602,62 \cdot M^3 - 618,99 \cdot M^2 + 260,28 \cdot M + 29,936$
(9250, 9750>	$610,87 \cdot M^3 - 623,80 \cdot M^2 + 261,38 \cdot M + 29,18$
(9750, 10250>	$638,94 \cdot M^3 - 640,00 \cdot M^2 + 264,36 \cdot M + 28,311$
(10250, 10750>	$643,74 \cdot M^3 - 639,54 \cdot M^2 + 263,74 \cdot M + 27,65$
(10750, 11250>	$5886,4 \cdot M^3 - 2650,3 \cdot M^2 + 511,89 \cdot M + 17,134$
(11250, 11750>	$663,78 \cdot M^3 - 650,52 \cdot M^2 + 265,59 \cdot M + 26,839$
(11750, 12750>	$5886,4 \cdot M^3 - 2650,3 \cdot M^2 + 511,89 \cdot M + 17,134$

Tyto hodnoty účinnosti se pak vypisují v tabulkách a v grafu v grafickém uživatelském rozhraní, jak lze vidět na Obrázcích 2.13 a 2.14 .

Ve zmíněných vzorcích nejsou použity všechny naměřené hodnoty, protože ne všechny hodnoty jsou validní k tomuto druhu experimentu. Ukládány jsou však všechny hodnoty pro případ, aby mohl být program s drobnou úpravou použit i pro výpočet jiných veličin.

2.3.2 Implementace

K tvorbě grafického rozhraní aplikace je použito vestavěné prostředí *App Designer*, které poskytuje možnost grafického vytvoření uživatelského rozhraní, které následně automaticky generuje pro tento návrh MATLAB kód. Tento před-připravený kód grafického rozhraní lze následně modifikovat přidáváním vlastních funkcí k jednotlivým prvkům tak, aby se docílilo očekávaného chování aplikace.

Tělo aplikace je rovněž napsáno pomocí jazyka MATLAB. Během implementace jsou používány knihovní funkce jazyku, jako příklad může být uvedena funkce `plot()` ke generování grafů, funkce `size()` k určení velikosti matice či funkce `sortrows()`, která seřadí řádky matice dle daného kritéria. Popisu implementovaných funkcí se věnuje tato kapitola níže.

```
function [LabviewData, lines] = LabViewData(filename)
```

Na vstupu funkce `LabViewData()` se nachází soubor vygenerovaný při měření prostředím Labview. V těle funkce je soubor otevřen a zpracován, což zahrnuje i převádění datových typů, jak je zmíněno výše. Výstupem funkce je matice hodnot, která se posléze používá k výpočtům a počet záznamů v souboru, který slouží jako pomocná proměnná v dalších volaných funkcích.

```
function [ArduinoData, lines] = ArduinoData(filename,
        ArduinoData)
```

Stejně jako předchozí funkce má funkce `ArduinoData()` za úkol zpracovat soubor na vstupu. Narozdíl od předchozí funkce však na vstupu dostává také matici vyplněnou hodnotami z předchozího volání této funkce. Jedná-li se o první volání funkce, matice na vstupu je prázdná. Toto opatření je nutné, aby bylo možno nahrávat více než jeden soubor s naměřenými daty. Výsledkem volání funkce je matice doplněná o zpracovaná data seřazená podle časové značky z načteného souboru a počet záznamů v souboru.

```
function [time] = findIntervals(ArduinoData)
```

Funkce `findIntervals()` pracuje s maticí, která je výstupem funkce `ArduinoData()`. V těle funkce je použita MATLAB funkce, pro detekci ustálených stavů, založená na algoritmu *sliding window*. [60] Jedná se o algoritmus implementovaný v knihovně MATLAB, proto zde nebude popisován.

```
function [time] = setIntervals(ArduinoData)
```

Funkce `setIntervals()` pracuje s maticí, která je výstupem funkce `ArduinoData()`. Stanovuje ustálené intervaly, které jsou jejím výstupem ve formě řady, kde je uložena sekvence údajů - začátek, konec - každého ustáleného stavu. Délka řady podělená dvěma tedy odpovídá počtu ustálených stavů.

```
function [ArduinoDataCalculated, firstNext] =
        ArduinoData_calculate(time01, time02, ArduinoData, first,
        last)
```

`ArduinoData_calculate()` je funkce, která pro zadaný časový interval vypočte průměr všech hodnot získaných ze souborů generovaných pomocí ArduinoUNO. Jelikož se jedná o data setříděná, funkce vrací index posledního záznamu v daném časovém intervalu, aby se při výpočtu hodnot z dalšího intervalu navázalo na předchozí procházení matice a nebyla potřeba hledat interval vždy od začátku matice.

```
function [LabviewDataCalculated, firstNext] =
    LabviewData_calculate(time01, time02, LabViewData, first,
        last)
```

Funkce `LabviewData_calculate()` pracuje na totožném principu jako předchozí funkce, pouze pracuje s maticí připravenou na základě souboru z Labview. Vypočítává hodnoty ze zadaného časového intervalu, který je uveden na vstupu funkce, které následně ve formě řady hodnot vrací na výstupu funkce. I zde je ukládána a vracena hodnota posledního zpracovaného záznamu pro vyšší efektivitu.

```
function [eta_g] = find_eta_g(RPMe, torque_c)
```

Jedná se o pomocnou funkci, která na základě otáček stroje za minutu a točivého momentu vrací hodnotu účinnosti generátoru pro daná vstupní data. Výpočty jsou založené na polynomech uvedených v Tabulce 2.5.

```
function [DataCalculated] = Calculate(LabviewDataCalculated,
    ArduinoDataCalculated)
```

Na vstupu funkce `Calculate()` jsou matice vypočtených hodnot z Arduino a Labview souborů pomocí funkcí `ArduinoData_calculate()` a `LabviewData_calculate()`. S pomocí těchto dat jsou vypočtena výsledná data z experimentálního měření pomocí matematického modelu popsaného v Sekci 2.3.1. Výsledná data jsou seřazena podle tlakového poměru jako primárního kritéria a otáček turbíny jako kritéria sekundárního.

```
function [DataCalculated] = fillForIntervals (LabviewData,
    linesL, ArduinoData, linesA, time)
```

Funkce `fillForIntervals()` kombinuje 3 výše zmíněné funkce: `ArduinoData_calculate()`, `LabviewData_calculate()` a `Calculate()`. Jejím úkolem je zkompletovat a propojit data z Arduina a data z prostředí Labview na základě časových intervalů. Následně pomocí funkce `Calculate()` připravit matici s daty vypočtenými matematickým modelem popsaným v Sekci 2.3.1.

```
function [PRchange] = SplitAndPlot(DataCalculated, lines)
```

`SplitAndPlot()` má za úkol zjistit kolik různých poměrů tlaku se ve vypočtených datech vyskytuje. Data jsou seskupena podle tlakového poměru a tak je výstupem funkce řada indexů, na kterých začíná sada s rozdílným tlakovým poměrem.

```
function [resultTable] = createResultTable(DataCalculated,
    PRchange)
```

Vytvoření matice, která následně zobrazuje uživateli právě jeden záznam pro každý tlakový poměr má na starosti funkce `createResultTable()`. Vybrán je vždy takový záznam, který má spočtený maximální mechanický výkon a maximální účinnost v rámci stejného tlakového poměru.

Všechny výše zmíněné implementované funkce jsou vhodně zasazeny do šablony vygenerované pomocí *App Designeru*. Spojením vzniklo GUI, které je vyobrazeno na Obrázku 2.12. GUI se skládá ze dvou záložek, kde jedna slouží k interakci s operátorem a na druhé jsou vyneseny vypočtené hodnoty.

V levé části GUI jsou lokalizovány všechny *tlačítka*, konkrétně se jedná o následující:

Labview file

otevřítá průzkumník souborů za účelem nahrání souboru vzniklého během měření pomocí programu Labview.

Arduino file

otevřítá průzkumník souborů za účelem nahrání souborů vzniklých během měření pomocí zařízení ArduinoUNO.

Load&Show data

potvrzuje, že operátor zadal všechny soubory a je možné soubory zpracovat. Slouží rovněž k vykreslení dat, ze kterých se následně vyčítají ustálené stavy.

Set intervals

je použito v případě, že operátor chce zadat ustálené stavy manuálně pomocí kurzoru. Tlačítko zapíná režim zadávání bodů v grafu, který je ukončen stisknutím klávesy Enter.

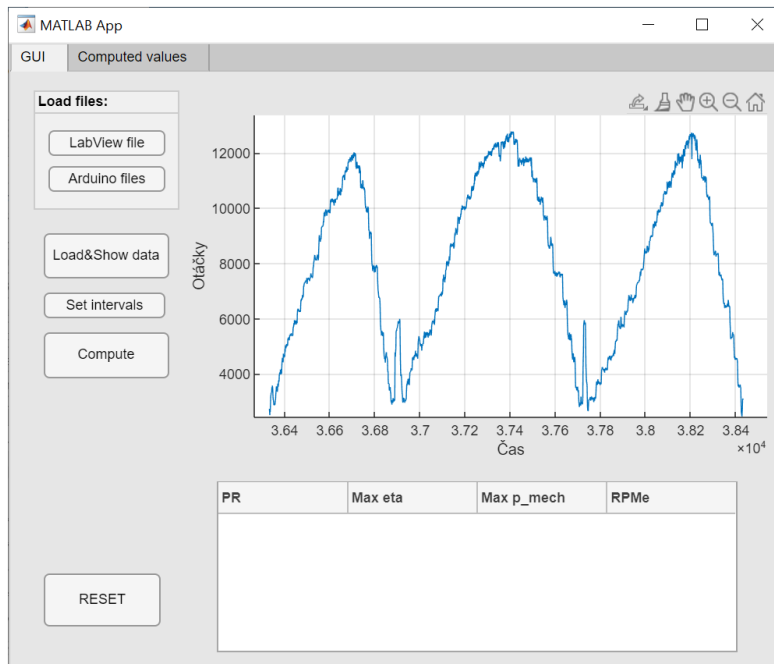
Compute

vypočte pro všechny ustálené stavy požadované hodnoty a následně na ně aplikuje matematický model. Vypočtené hodnoty se následně vykreslí do grafu a vypíší do tabulky pod grafem a tabulky na druhé záložce aplikace.

Reset

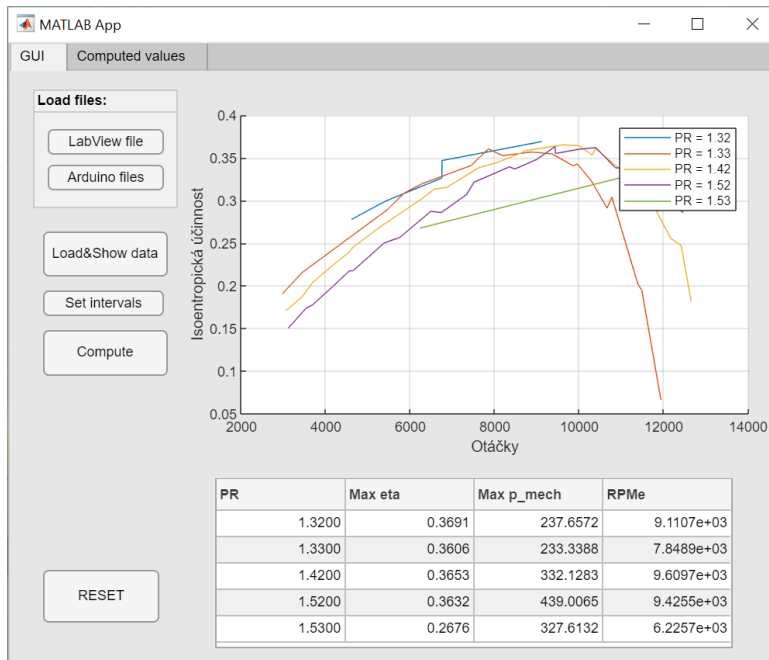
tlačítko vymaže paměť programu a připraví jej do stavu, ve kterém se program nachází po zapnutí.

Vedle tlačítek je v GUI oblast grafu v pravém horním rohu a pod ní tabulka na připravená na vynesení výsledků. Výsledný stav aplikace je vidět na Obrázku 2.13, v oblasti grafu se vykreslí závislost účinnosti na otáčkách



Obrázek 2.12: Ukázka implementovaného GUI

pro různé tlakové poměry a v tabulce pod grafem jsou vypsány maximální dosažené hodnoty výkonu a účinnosti pro různé tlakové poměry. Na druhé záložce, která je zobrazená na Obrázku 2.14, jsou vypsány všechny napočítané hodnoty pro případ, že by je operátor potřeboval.



Obrázek 2.13: Zobrazení výsledků v GUI

The figure shows a MATLAB App GUI with a table of computed values. The table has columns for PR, v_dot, p_ad, p_em, T, voltage, current, power, and RPMe. The data is organized by PR values, with multiple rows for each PR value showing different operating points.

PR	v_dot	p_ad	p_em	T	voltage	current	power	RPMe
1.3200	77.8180	1.3057	0.9864	22.1014	14.1362	7.8045	110.4483	4.609E
1.3200	78.3114	1.3076	0.9869	22.0752	17.1556	6.9049	118.6458	5.410E
1.3200	78.0983	1.3066	0.9868	22.1018	22.4924	5.4255	122.0151	6.750E
1.3200	78.4025	1.3043	0.9865	22.1064	22.2256	5.9750	132.0686	6.751E
1.3200	78.4060	1.3052	0.9865	22.0507	30.6936	4.0298	123.6904	9.110E
1.3300	79.8386	1.3096	0.9864	22.1311	8.0620	9.2180	74.3155	
1.3300	80.5626	1.3077	0.9864	22.1059	9.7440	8.8846	86.9442	3.433E
1.3300	79.9335	1.3090	0.9864	22.0879	17.1216	6.8272	116.6833	5.454E
1.3300	77.9877	1.3079	0.9869	22.0739	18.7843	6.3623	120.1678	5.866E
1.3300	77.3072	1.3077	0.9868	22.0707	20.4205	5.8797	121.1827	6.291E
1.3300	78.1381	1.3085	0.9868	22.1055	24.8837	4.9976	125.3254	7.438E
1.3300	78.2065	1.3081	0.9864	22.0610	26.4665	4.9814	131.2046	7.848E
1.3300	77.6807	1.3087	0.9867	22.1402	27.5036	4.4996	124.4393	8.192E
1.3300	77.8070	1.3109	0.9868	22.1418	30.1020	3.9984	120.3875	8.869E
1.3300	77.7283	1.3102	0.9867	22.1387	32.0491	3.5440	114.3223	9.358E
1.3300	77.4908	1.3096	0.9866	22.1576	33.9834	3.0217	102.6893	9.858E
1.3300	78.9773	1.3083	0.9864	22.0382	34.0935	3.1025	105.7643	9.951E
1.3300	77.2471	1.3084	0.9866	22.1149	35.6928	2.5459	90.8714	1.028E
1.3300	75.8707	1.3072	0.9866	22.0318	37.1316	2.0294	74.6900	1.065E
1.3300	77.3379	1.3078	0.9865	22.0258	37.3737	2.0944	78.2933	1.077E

Obrázek 2.14: Zálůžka v GUI se všemi vypočtenými výslednými hodnotami.

Ekonomické zhodnocení

Tato kapitola se věnuje vyhodnocení výdajů a příjmů z investice pomocí ukazatelů zmíněných v Sekci 1.3 a následně ilustruje tyto výsledky pomocí citlivostní analýzy. Závěrem se také věnuje dalším neekonomickým přínosům realizace informačního systému.

3.1 Kapitálové výdaje a příjmy z investice

Po definici základních ukazatelů pro ekonomické hodnocení investic bude nyní popsán ekonomický model diskutující náklady a výnosy z investice a jejich bilance v čase. Investice bude ekonomicky následně zhodnocena za pomoci ukazatelů ze Sekce 1.3.

Dobu životnosti investice, a zároveň i období, po které bude investice do návrhu a realizace informačního systému hodnocena, je zvolena jakožto 4 roky. Důvodem je soulad s dobou realizace vědecko-výzkumného projektu na vývoj mikro turboexpandérů, po který minimálně bude vyvinutý software používán pro vyhodnocení experimentálních dat z provozu prototypů. Hodnocení je tedy prováděno z pozice investora - vedoucího pracovníka výzkumného týmu - který si na zakázku objednává řešení informačního systému pro sběr a vyhodnocení experimentálních dat z měření.

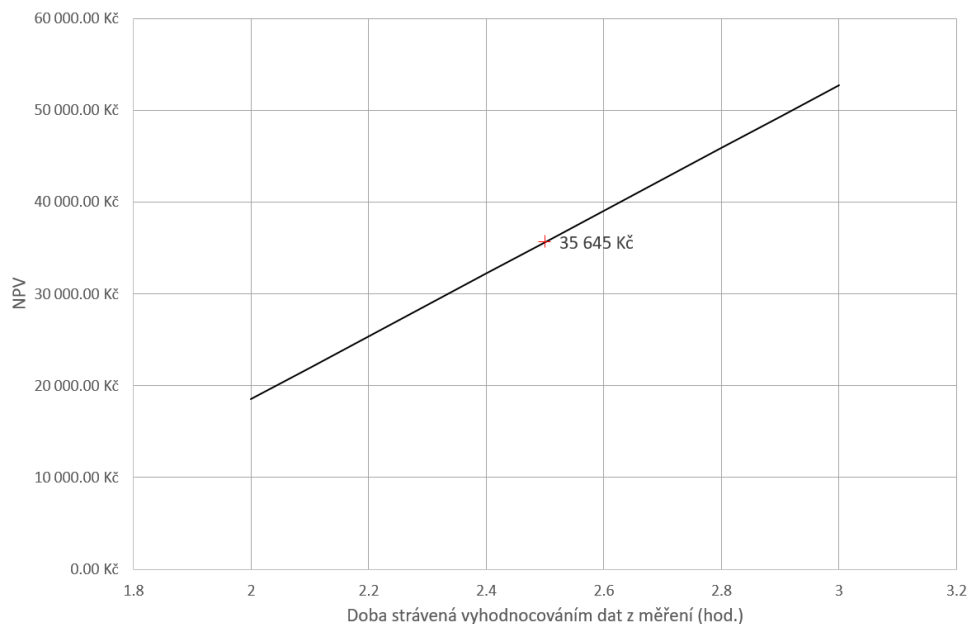
Jelikož pro zafinancování budou použity finance spojené s dotací na výzkumný projekt, které byly výzkumnému týmu přiděleny *ex ante*¹⁷, lze investici přisoudit velmi nízké riziko, a stejně tak relativně nízké vážené náklady na kapitál. Při realizaci je přihlédnuto k možnostem klienta, který má neomezený celoakademický přístup k interaktivnímu programovému prostředí MATLAB. Jak již bylo diskutováno v rámci kapitoly 1.4.2, MATLAB má nespornou výhodu v rychlé práci s velkými maticemi dat prostřednictvím uživatelský přívětivého GUI. Navíc je tento skriptovací programovací jazyk s dostatečnou kvalitou dokumentace možno i do budoucna upravovat a modulárně jej používat

¹⁷příjemce obdrží platbu předem

i pro jiná experimentální měření. Z tohoto hlediska tedy neplynou klientovi žádné náklady spojené s pořizováním prostředí, ve kterém bude software spouštěn.

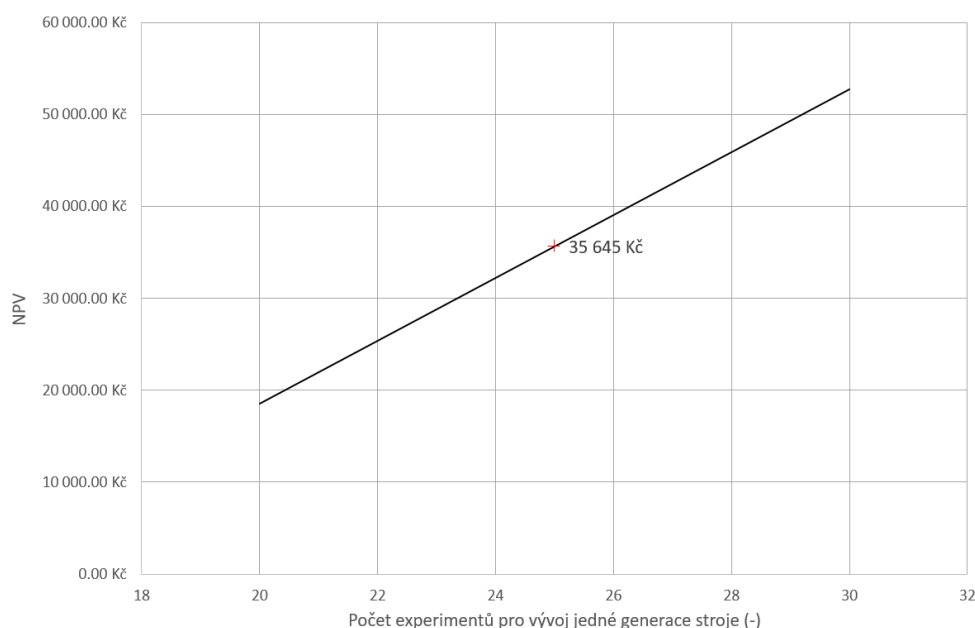
Hlavními náklady, které vytvářejí cenu zakázky, jsou tedy z největší části osobní náklady na čas strávený návrhem a realizací softwaru dle požadavků klienta. Na základě předchozích zkušeností byla tvorba tohoto informačního systému oceněna na 50 000 Kč, a to sice kromě realizace i včetně komunikace s klientem, tvorby dokumentace, odlazování bugů a fixování problémů.

Na druhou stranu pro výzkumné centrum dle vyjádření hlavního řešitele projektu znamená plně automatizované řešení úsporu 2,5 hodiny jednotvárné práce výzkumného pracovníka na jednom experimentálním měření, která by jinak mohla být využita za podstatně produktivnějším účelem. Na Diagramu 2.4 se úspora 2,5 hodin promítá do prvních čtyř aktivit, největší úspora času však vzniká u vizuálního vyhledávání ustálených stavů v grafu a s tím spojené zapisování začátků a konců ustálených intervalů. Řečí čísel to pro zaměstnavatele v superhrubé mzdě znamená zhruba 1000 Kč úsporu na jednom provedeném experimentu. Jelikož lze očekávat, že s narůstajícím počtem vyhodnocených experimentálních měření by částečně klesala doba strávená vyhodnocováním dat, neboť by výzkumník měl již tou dobou značnou praxi s manuální volbou ustálených stavů a iterativním výpočtem, byla provedena také citlivostní analýza na dobu strávenou vyhodnocováním jednoho experimentálního měření, viz Obr. 3.1.



Obrázek 3.1: Citlivostní analýza faktoru doby vyhodnocování dat z experimentálního měření na NPV

3.1. Kapitálové výdaje a příjmy z investice



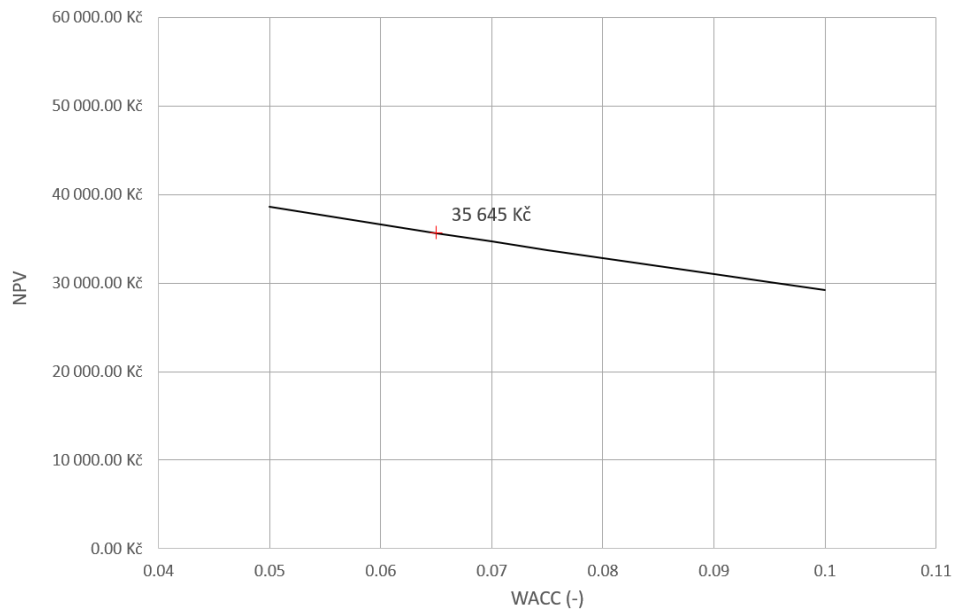
Obrázek 3.2: Citlivostní analýza faktoru počtu experimentů pro vývoj jedné generace turbostroje na NPV

Je tedy jasné, že celková úspora a tedy i ekonomická výhodnost investice do tvorby informačního systému bude hrubě odviset od počtu provedených experimentů v následujících čtyřech letech od realizace tohoto informačního systému. Řešitelem výzkumného projektu bylo uvedeno, že očekává na každou generaci prototypu mikro turboexpandéru zhruba 25 experimentálních měření, což je stejně jako bylo doposud provedeno za pomoci manuálního vyhodnocování a jednoduchého modelu v prostředí MS Excel. Citlivostní analýza na faktor počtu potřebných provedených experimentů pro vývoj jedné generace prototypu je znázorněna na Obrázku 3.2.

Během doby realizace projektu mají být vyvinuty čtyři generace prototypu, přičemž ta poslední má být finální předkomerční verzí. Celkem lze tedy očekávat nejméně 100 provedených experimentů ve čtyřech letech, s rovnoměrnou distribucí vývoje generací mezi jednotlivými lety v době realizace projektu. Každý rok by tedy vyhodnocováním dat člen řešitelského týmu strávil $25 \cdot 2,5 = 62,5$ hodin, což činí v superhrubé mzdě 25 000 Kč. Tato roční celková úspora ponížena o vážený průměr nákladů na kapitál bude porovnávána vůči investičním nákladům a v rámci Sekce 3.2 vyhodnocena za pomoci výše zmiňovaných indikátorů ekonomického zhodnocení investice.

Vážený průměr nákladů na kapitál (WACC) byl odhadnut z informací obdržených od vedoucího výzkumného týmu - projekt je hrazen ze sta procent z podpory z mezinárodních (norských) fondů podporujících vědu a výzkum.

3.2. Diskuze výsledků ekonomického zhodnocení



Obrázek 3.3: Citlivostní analýza faktoru WACC na NPV

Podpora je přitom vyplácena *ex ante* a poté výzkumné centrum disponuje podporou jakožto vlastním kapitálem. Náklady spojené s vlastním kapitálem jsou míněné jako požadovaná výnosnost investic. Jelikož investice není hrazena z cizího kapitálu, nevstupuje do stanovení diskontní míry úroková sazba. Na základě těchto informací a po konzultaci s vedoucím řešitelského týmu byla zvolena hodnota $WACC = 6.5 \%$, přičemž byla opět provedena citlivostní analýza na výši váženého průměru nákladů na kapitál a její dopad na čistou současnou hodnotu investice, viz Obr. 3.3.

3.2 Diskuze výsledků ekonomického zhodnocení

V Tabulce 3.1 je znázorněno cash flow v letech od počátku investice do konce doby realizace výzkumného projektu. Jak je z tabulky vidět, investice je provedena v čase nula a všechny budoucí toky peněz ve sledovaném časovém úseku jsou již pouze kladné ve smyslu příjmů, resp. úspor.

Již v druhém roce po provedení investice do realizace informačního systému je kumulované cash flow na nule, a tedy *prostá doba návratnosti* investice je rovna 2 letům, tedy $PP = 2$ roky. Z kumulovaného diskontovaného cash flow v posledním roce hodnocení investice je evidentní výše *čisté současné hodnoty*, tedy $NPV = 35\,645$ Kč. Za pomoci vestavěné funkce prostředí MS Excel „MIRA.VYNOSNOSTI“ bylo vypočteno *vnitřní výnosové procento* investice jako $IRR = 34.9 \%$. Dále byla vypočtena i *diskontovaná doba návratnosti* za

3.2. Diskuze výsledků ekonomického zhodnocení

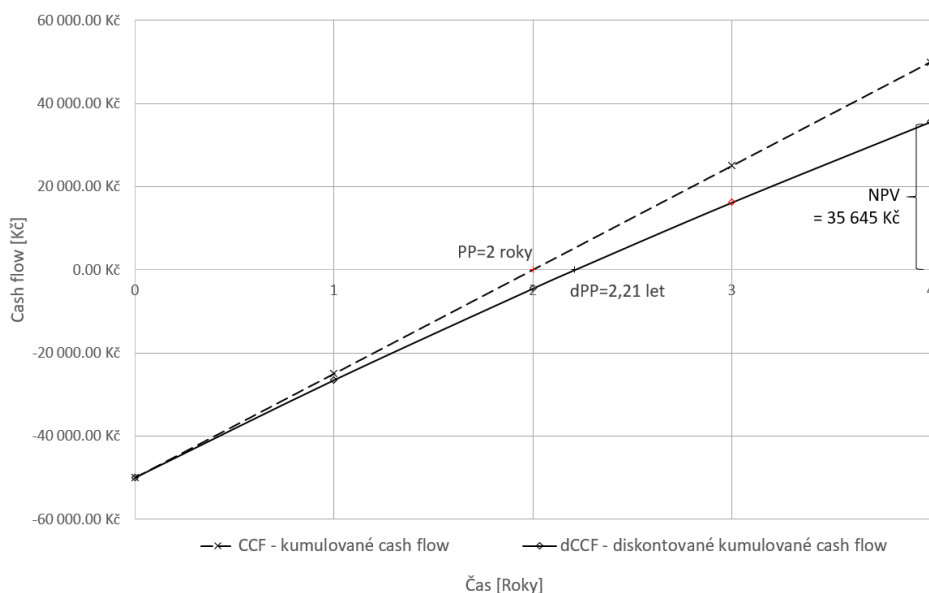
Tabulka 3.1: Průběh cash flow a diskontovaného cash flow v letech hodnocení investice

Roky	0	1	2	3	4
CF	-50 000.00 Kč	25 000.00 Kč	25 000.00 Kč	25 000.00 Kč	25 000.00 Kč
CCF	-50 000.00 Kč	-25 000.00 Kč	0.00 Kč	25 000.00 Kč	50 000.00 Kč
dCF	-50 000.00 Kč	23 474.18 Kč	22 041.48 Kč	20 696.23 Kč	19 433.08 Kč
dCCF	-50 000.00 Kč	-26 525.82 Kč	-4 484.34 Kč	16 211.89 Kč	35 644.97 Kč

pomocí funkce „Hledání řešení“ při hledání řešení rovnice 3.1.

$$dCCF = 0 [-] \quad (3.1)$$

Času, ve kterém tato rovnice platí, je tato diskontovaná doba návratnosti rovna. Konečně byl spočten také NPV kvocient $PI = 1.713$, ten značí, že investovaná částka se za dobu hodnocení investice nejenže vrátí, ale také se zhodnotí o dalších 71 %. Všechny ukazatele tedy hodnotí ekonomickou výhodnost investice jako značně pozitivní. Ostatně tyto ukazatele jsou také znázorněny v grafu toku peněz v čase na Obrázku 3.4.



Obrázek 3.4: Průběh cash flow a diskontovaného cash flow v letech hodnocení investice

Jak již bylo zmíněno výše, pro tři kritéria:

- WACC
- Doba strávená vyhodnocováním dat z měření

- Počet experimentů pro vývoj jedné generace stroje

byly provedeny citlivostní analýzy s ohledem na vliv faktorů na NPV. To umožní investorovi snadněji se orientovat ve vlivu proměnných vstupujících do ekonomického modelu investice a případných dopadech na její (ne)výhodnost. Grafy citlivostních analýz jsou zobrazeny na Obrázcích 3.3, 3.2 a 3.1.

3.3 Další neekonomické přínosy realizace informačního systému

Mimo v Kapitole 3.2 diskutované ekonomické přínosy vykazuje realizace projektu vývoje informačního systému i mnohé neekonomické přínosy.

Tím na první pohled zřejmým a z podstaty věci jasným, je ušetření výzkumných pracovníků od vykonávání jednotvárné nepříjemné práce. Nejen že při monotónní práci se s postupem času může čím dál tím více projevovat lidský faktor, a může tak docházet k chybám při manuálních vstupech ustálených stavů nebo i při vyhodnocování dat, ale také pracovníci ztrácí při velkém množství dat potřebných k vyhodnocení motivaci pro práci. Namísto únavné monotónní činnosti pak výzkumní pracovníci mohou věnovat svůj produktivní čas kreativnějším činnostem s potenciálním celospolečenským přínosem, než manuální zpracování dat. To se v roce 2021 navíc jeví už i jako funkčně i morálně zastaralé a má-li český výzkum a vývoj dosahovat světové úrovně, musí být maximum úsilí věnováno právě kreativním činnostem.

Rozvrstvení časové úspory v průběhu roku je značně nehomogenní, neboť experimenty jsou prováděny zpravidla všechny naráz a následně je rychle vyžadován výsledek, aby výzkumný pracovník mohl adekvátně upravit geometrii stroje přímo na místě a plně tak využít koncept *rapid prototypingu* v kombinaci s aditivními technologiemi výroby hlavních komponent lopatkového stroje.

Dříve bylo nutné manuálně vyhodnocovat experimentální data z měření, což bylo vždy prováděno až následně s několikanásobným odstupem a nebylo možné ihned reagovat na výsledky. S využitím informačního systému tedy bude moci výzkumný pracovník ihned v laboratoři vyhodnotit, jaká geometrie lopatkování stroje je výhodná s ohledem na výslednou účinnost stroje, provést adekvátní změny v 3D modelu, a pokud je tato možnost k dispozici, ihned poslat upravený model lopatkových kol, po převedení do požadovaného formátu, do 3D tiskárny. Tato upravená kola pak znovu upnout do sestavy a provést další sérii měření s tlakovým vzduchem. Díky okamžité zpětné vazbě z informačního systému se tak celý proces vývoje zrychlí a zkomfortní.

Závěr

Začátek diplomové práce je věnován tématu automatizace, jejímu vývoji v čase, způsobům, kterými byla popsána a především dopadům automatizace na člověka. Není opomenut ani způsob, jakým je možné zanalyzovat proces a navrhnout jeho částečné či celkové zautomatizování s ohledem na člověka.

Dalším rozebíraným tématem je měření pracovní výkonnosti a jeho úskalí v odvětví vědy a výzkumu. Následované ekonomickým hodnocením zaměřeným na informační systémy, kde diskutuje statické i dynamické hodnotící metody investic.

Čtvrtá a poslední kapitola v teoretické části práce seznamuje čtenáře se základy jazyků použitých v praktické části diplomové práce. Jako první je zmíněn modelovací jazyk UML, jeho běžné použití a členění. Důraz je kladen na diagramy chování, které jsou využity při zpracování této práce. Druhým zmíněným jazykem je jazyk MATLAB, u kterého se čtenář dozví jeho primární užití a výhody, které jeho použití přináší.

Úvod praktické části se věnuje vysvětlení problému, který bude následně řešen a popisuje současný stav jeho řešení.

Následně je popsán návrh informačního systému, nejprve je diskutován současný stav automatizace a navržena místa, kde je možné zvednout úroveň automatizace. Samotný návrh je posléze založen na modelu popsaném v teoretické části práce. Další návrhová část se věnuje modelování systému pomocí UML diagramů, převážně identifikací případů užití a stavů, kterých může systém nabývat.

Po sekci návrhu je popsáno, jak je systém implementován. Prvně je popsán proces abstraktně, kde je zachyceno vnitřní chování programu a je vypíchnut použitý matematický model. Realizační část je uzavřena popisem implementovaných funkcí a ukázkou grafického uživatelského rozhraní.

Poslední kapitola diplomové práce je zasvěcena ekonomickému hodnocení, kde se metody hodnocení investic aplikují na danou problematiku. Hodnocení investice do realizace vývoje informačního systému je prováděno z pozice vedoucího výzkumného týmu, který bude tento software využívat v rámci vývoje

prototypů turbostrojů.

Informační systém, který byl v rámci diplomové práce vyvinut, bude sloužit k zefektivnění procesu sběru a vyhodnocení dat z experimentálního měření a urychlí tak vývoj nových generací prototypů turbo expandérů. Na základě provedeného ekonomického hodnocení, ale i analýzy dalších neekonomických přínosů, vyplývá, že investice do realizace takového informačního systému je příhodná. Kromě úspory vynaloženého úsilí a času k vyhodnocení dat dojde také k úspoře peněz vyjádřené čistou současnou hodnotou projektu 35 645 Kč, která je generována právě prostřednictvím předpokládané úspory času.

Při návrhu informačního systému bylo navíc dbáno, aby byl návrh v co nejvyšší možné míře modulární a mohl být v budoucnu použit pro vyhodnocení experimentálních dat i z jiných měření. Pouhou změnou modulu matematického modelu se zachováním struktury a koncepce kódu včetně jejího grafického rozhraní tak může software sloužit i pro odlišnou aplikaci vyhodnocení experimentálních dat s dynamickými změnami sledované proměnné.

Literatura

- [1] Digitální vlákno jako "prováděcí směrnice" digitální transformace. Dostupné z: <https://www.aveng.cz/digitalizace-podniku/digitalni-vlakno-jako-provadeci-smernice-digitalni-transformace>{_}1
- [2] Optimalizace výrobních procesů a virtuální továrna. Dostupné z: <https://www.technickytydenik.cz/rubriky/serialy/ze-zivota-vyzkumneho-centra/optimalizace-vyrobnich-procesu-a-virtualni-tovarna>{_}38350.html
- [3] Novotny, V.; Spale, J.; Stunova, B. B.; aj.: 3D Printing in Turbomachinery: Overview of Technologies, Applications and Possibilities for Industry 4.0. In *Turbo Expo: Power for Land, Sea, and Air*, ročník 58677, American Society of Mechanical Engineers, 2019, str. V006T24A021.
- [4] Parasuraman, R.; Riley, V.: Humans and automation: Use, misuse, disuse, abuse. *Human factors*, ročník 39, č. 2, 1997: s. 230–253.
- [5] Drucker, P. F.: *Management challenges for the 21st century*. Routledge, 2007.
- [6] Billings, C. E.: *Aviation automation: the search for a human-centered approach*. Mahwah, N.J: Lawrence Erlbaum Associates Publishers, 1997, ISBN 9780805821260 9780805821277.
- [7] Lindström, V.; Winroth, M.: Aligning manufacturing strategy and levels of automation: A case study. *Journal of Engineering and Technology Management - JET-M*, ročník 27, č. 3-4, 2010: s. 148–159, ISSN 09234748, doi:10.1016/j.jengtecman.2010.06.002.
- [8] Frohm, J.; Lindström, V.; Stahre, J.; aj.: Levels of Automation in Manufacturing. *Ergonomia - an International journal of ergonomics and human factors*, ročník 30, č. 3, 2008, ISSN 0137-4990.

-
- [9] Fitts, P. M.; Viteles, M.; Barr, N.; aj.: Human engineering for an effective air-navigation and traffic-control system, and appendixes 1 thru 3. Technická zpráva, OHIO STATE UNIV RESEARCH FOUNDATION COLUMBUS, 1951.
- [10] Jordan, N.: Allocation of functions between man and machines in automated systems. *Journal of applied psychology*, ročník 47, č. 3, 1963: str. 161.
- [11] Williams, T.: PERA and GERAM: Establishment of the Place of the Human in Enterprise Integration. In *Proceedings of the 14th International Federation of Automatic Control (IFAC) World Congress, Beijing, China, 1999*.
- [12] Lindström, V.; Winroth, M.: Aligning manufacturing strategy and levels of automation: A case study. *Journal of Engineering and Technology Management - JET-M*, ročník 27, č. 3-4, 2010: s. 148–159, ISSN 09234748, doi:10.1016/j.jengtecman.2010.06.002.
- [13] Jones, J. A.; Widjaja, T. K.: Electronic human factors guide for aviation maintenance. *Proceedings of the Human Factors and Ergonomics Society*, ročník 1, č. JANUARY, 1995: s. 71–74, ISSN 10711813.
- [14] Manual definition and meaning: Collins English Dictionary. Dostupné z: <https://www.collinsdictionary.com/dictionary/english/manual>
- [15] Frohm, J.; Bellgran, M.; Lindström, V.: A model for parallel levels of automation within manufacturing. *18th International Conference on Production Research*, , č. February 2017, 2005.
- [16] Pennathur, A.: Advanced technologies and humans in manufacturing workplaces: An interdependent relationship. *International Journal of Industrial Ergonomics*, ročník 33, 04 2004: s. 295–313, doi:10.1016/j.ergon.2003.10.002.
- [17] Kaber, D. B.; Onal, E.; Endsley, M. R.: Design of automation for tele-robots and the effect on performance, operator situation awareness, and subjective workload. *Human factors and ergonomics in manufacturing & service industries*, ročník 10, č. 4, 2000: s. 409–430.
- [18] Sheridan, T.: Computer control and human alienation. 1980.
- [19] Satchell, P.: *Innovation and automation*. Routledge, 2018.
- [20] Wickens, C. D.; Hollands, J. G.; Banbury, S.; aj.: *Engineering psychology and human performance*. Psychology Press, 2015.

-
- [21] Parasuraman, R.; Sheridan, T. B.; Wickens, C. D.: A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans.*, ročník 30, č. 3, 2000: s. 286–297, ISSN 10834427, doi:10.1109/3468.844354.
- [22] Endsley, M. R.: Level of automation effects on performance, situation awareness and workload in a dynamic control task. *Ergonomics*, ročník 42, č. 3, 1999: s. 462–492.
- [23] Woods, D. D.; Sarter, N. B.: Evaluating the impact of new technology on human-machine cooperation. In *Verification and validation of complex systems: Human factors issues*, Springer, 1993, s. 133–158.
- [24] Vicente, K. J.; Rasmussen, J.: Ecological interface design: Theoretical foundations. *IEEE Transactions on systems, man, and cybernetics*, ročník 22, č. 4, 1992: s. 589–606.
- [25] Kirlik, A.: Modeling strategic behavior in human-automation interaction: Why an "aid" can (and should) go unused. *Human factors*, ročník 35, č. 2, 1993: s. 221–242.
- [26] Landauer, T. K.: *The trouble with computers: Usefulness, usability, and productivity*. MIT press, 1995.
- [27] Kaber, D. B.; Omal, E.; Endsley, M.: Level of automation effects on telerobot performance and human operator situation awareness and subjective workload. *Automation technology and human performance: Current research and trends*, 1999: s. 165–170.
- [28] Parasuraman, R.; Molloy, R.; Singh, I. L.: Performance consequences of automation-induced 'complacency'. *The International Journal of Aviation Psychology*, ročník 3, č. 1, 1993: s. 1–23.
- [29] Crocoll, W. M.; Coury, B. G.: Status or recommendation: Selecting the type of information for decision aiding. In *Proceedings of the human factors society annual meeting*, ročník 34, SAGE Publications Sage CA: Los Angeles, CA, 1990, s. 1524–1528.
- [30] McMillan, G. R.; Beevis, D.; Salas, E.: *Applications of human performance models to system design*, ročník 2. Springer Science & Business Media, 2013.
- [31] Wickens, C. D.: Designing for situation awareness and trust in automation. *IFAC Proceedings Volumes*, ročník 28, č. 23, 1995: s. 365–370.
- [32] Parnas, D. L.; Van Schouwen, A. J.; Kwan, S. P.: Evaluation of safety-critical software. *Communications of the ACM*, ročník 33, č. 6, 1990: s. 636–648.

- [33] Council, N. R.; aj.: *The future of air traffic control: Human operators and automation*. National Academies Press, 1998.
- [34] Wickens, C. D.; Conejo, R.; Gempler, K.: Unreliable automated attention cueing for air-ground targeting and traffic maneuvering. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, ročník 43, SAGE Publications Sage CA: Los Angeles, CA, 1999, s. 21–25.
- [35] Measuring Productivity - OECD Manual. *Measuring Productivity - OECD Manual*, 2001, doi:10.1787/9789264194519-en.
- [36] Abramo, G.; D'Angelo, C. A.: How do you define and measure research productivity? *Scientometrics*, ročník 101, č. 2, 2014: s. 1129–1144, ISSN 15882861, doi:10.1007/s11192-014-1269-8.
- [37] Moed, H. F.: *Citation analysis in research evaluation*, ročník 9. Springer Science & Business Media, 2006.
- [38] Pontille, D.: Signature et travail d'attribution. *Réseaux*, ročník n° 127-128, 01 2004: s. 253–275, doi:10.3917/res.127.0253.
- [39] Butler, L.: Assessing University Research: A Plea for A Balanced Approach. *Science & Public Policy - SCI PUBLIC POLICY*, ročník 34, 10 2007: s. 565–574, doi:10.3152/030234207X254404.
- [40] Abramo, G.; Andrea D'Angelo, C.; Di Costa, F.: Assessment of sectoral aggregation distortion in research productivity measurements. *Research Evaluation*, ročník 17, č. 2, 2008: s. 111–121.
- [41] Glänzel, W.: Seven Myths in Bibliometrics. About facts and fiction in quantitative science studies. *Proceedings of WIS*, ročník 2, 06 2008, doi: 10.1080/09737766.2008.10700836.
- [42] Natalie A. Townsend, OD, and John J. McSoley, O.: The Do ' s and Don ' ts of measuring IOP. *Review of optometry*, 2008: s. 1–9.
- [43] Scholleová, H.: *Ekonomické a finanční řízení pro neekonomy: 2., aktualizované a rozšířené vydání*. Grada Publishing a.s., 2012, ISBN 9788024777177. Dostupné z: <https://books.google.cz/books?id=siFgAgAAQBAJ>
- [44] What is UML? - ArchiMetric. Dostupné z: <https://www.archimetric.com/what-is-uml/>
- [45] Visual Paradigm: What is Unified Modeling Language (UML)? 2017. Dostupné z: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/>

-
- [46] Visual Paradigm: What is Use Case Diagram? 2019. Dostupné z: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>
- [47] Visual Paradigm: What is Activity Diagram? 2019. Dostupné z: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/>
- [48] Visual Paradigm: What is State Machine Diagram? 2018. Dostupné z: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-state-machine-diagram/>
- [49] What is MATLAB? Dostupné z: <https://nl.mathworks.com/discovery/what-is-matlab.html>
- [50] MATLAB. Dostupné z: <https://nl.mathworks.com/products/matlab.html>
- [51] Using MATLAB with Other Programming Languages. Dostupné z: <https://nl.mathworks.com/products/matlab/matlab-and-other-programming-languages.html>
- [52] Vopatová, R.: *Analýza a popis vzorku pneumatiky*. B.S. thesis, České vysoké učení technické v Praze. Vypočetní a informační centrum., 2018.
- [53] The MathWorks, I.: Develop Apps Using App Designer. Dostupné z: <https://www.mathworks.com/help/matlab/app-designer.html>
- [54] Maščuch, J.: Organický Rankinův cyklus v decentralizované mikrokogeneraci. 2014.
- [55] Weiß, A. P.; Novotný, V.; Popp, T.; aj.: Customized ORC micro turbo-expanders - From 1D design to modular construction kit and prospects of additive manufacturing. *Energy*, ročník 209, 2020: str. 118407, ISSN 0360-5442, doi:<https://doi.org/10.1016/j.energy.2020.118407>. Dostupné z: <http://www.sciencedirect.com/science/article/pii/S0360544220315140>
- [56] Novotny, V.; Vitvarova, M.; Kolovratnik, M.; aj.: Design and Manufacturing of a Metal 3D Printed kW Scale Axial Turboexpander. ASME International, jun 2019, ISBN 9780791858714, doi:10.1115/gt2019-91822.
- [57] Hofreiter, M.: *Základy automatického \dot{L}^{TM} řízení*. Druhé vydání, 2018, ISBN 978-80-01-06380-4, 166 s.
- [58] Fowler, M.: *Destilované UML*. Grada Publishing as, 2009.

- [59] Lemmon, E. W.; Huber, M. L.; McLinden, M. O.: NIST Standard Reference Database 23: Reference Fluid Thermodynamic and Transport Properties-REFPROP, Version 9.1, National Institute of Standards and Technology. 2013, doi:<http://dx.doi.org/10.18434/T4JS3C>. Dostupné z: <https://www.nist.gov/srd/refprop>
- [60] MatWorks: Sliding Window Method and Exponential Weighting Method. Dostupné z: <https://uk.mathworks.com/help/dsp/ug/sliding-window-method-and-exponential-weighting-method.html>

Seznam obrázků

1.1	Dělení automatizace dle Frohma; upraveno z [15]	8
1.2	Proces návrhu designu pro automatizace; upraveno z [21]	12
1.3	Typy UML diagramů; převzato z [46]	20
1.4	Základní entity Use Case Diagramu	22
1.5	Relace Use Case Diagramu	22
1.6	Ukázka použití Use Case Diagramu	23
1.7	Základní uzly v Activity Diagramu	24
2.1	Schéma experimentální tratě pro měření účinnosti turboexpandérů; upraveno z [56]	29
2.2	Příklad raw dat pro jeden tlakový poměr - otáčky a elektrický výkon - a znázornění ustáleného stavu	30
2.3	Detail vzorku dat z měření - ukázka regulace otáček stroje v módu Constant RPM	31
2.4	Současný stav řešení pomocí flow chart	32
2.5	Znázornění stupňů automatizace v původním a novém řešení	34
2.6	Diagram případů užití systému pro zpracování a vyhodnocení dat	36
2.7	Stavový digram - systém	37
2.8	Diagram případu užití UC001: Nahrání souborů	38
2.9	Diagram případu užití UC002: Určení ustálených stavů	40
2.10	Diagram případu užití UC003: Vyhodnocení experimentálních dat	42
2.11	Diagram případu užití UC004: Reset	42
2.12	Ukázka implementovaného GUI	52
2.13	Zobrazení výsledků v GUI	53
2.14	Záložka v GUI se všemi vypočtenými výslednými hodnotami.	53
3.1	Citlivostní analýza faktoru doby vyhodnocování dat z experimentálního měření na NPV	55
3.2	Citlivostní analýza faktoru počtu experimentů pro vývoj jedné generace turbostroje na NPV	56

3.3	Citlivostní analýza faktoru WACC na NPV	57
3.4	Průběh cash flow a diskontovaného cash flow v letech hodnocení investice	58
C.1	Ukázka souboru generovaného z Labview prostředí	75
C.2	Ukázka souboru generovaného z Arduina	76

Seznam tabulek

1.1	Fittsův seznam[9]	5
1.2	Definice úrovně automatizace dle různých autorů; převzato z [8] . .	7
1.3	Deset úrovní automatizace [18]	8
1.4	Deset úrovní automatizace dle Endsleye a Kabera[22]	9
1.5	Čtyři úrovně automatizace[8]	9
2.1	Popis případu užití UC001: Nahrání souborů	39
2.2	Popis případu užití UC002: Určení ustálených stavů	41
2.3	Popis případu užití UC003: Vyhodnocení experimentálních dat . .	43
2.4	Popis případu užití UC004: Reset	43
2.5	Výpočet η_g pro různý počet otáček turbíny za minutu	48
3.1	Průběh cash flow a diskontovaného cash flow v letech hodnocení investice	58

Seznam použitých zkratk

BLDC Bezkartáčový stejnosměrný motor/generátor

CCF Kumulované cash flow

CF Cash flow

CP Celkový Příjem

dCCF Kumulované diskontované cash flow

dCF Diskontované cash flow

dPP Diskontovaná doba návratnosti

GUI Graphical User Interface

IN Investiční náklady

IRR Internal Rate of Return

MATLAB MATrix LABoratory

NCP Čistý celkový příjem

NPV Net Present Value

ORC Organický Rankinův Cyklus

PC Personal Computer

PP Doba návratnosti

REFPROP NIST Reference Fluid Thermodynamic and Transport Properties Database

ROI Return On Investment

RPM Rotation Per Minute

UC Use Case

UML Unified Modeling Language

VBA Visual Basic for Applications

WACC Weighted Average Cost of Capital

Seznam použitých symbolů

- η_{is} Účinnost turboexpandéru [%]
 P_{mech} Mechanický výkon turboexpandéru [W]
 $P_{el_{meas}}$ Měřený elektrický výkon [W]
 P_{rect} Ztrátový výkon na usměrňovači [W]
 P_{is} Teoretický maximální výkon [W]
 η_g Účinnost elektrického generátoru [%]
 T Točivý moment [Nm]
 ω Úhlová frekvence [$rad \cdot s^{-1}$]
 dU_{rect} Úbytek elektrického napětí na usměrňovači [V]
 I Elektrický proud [A]
 U Elektrické napětí [V]
 \dot{m}_{air} hmotnostní tok vzduchu [$kg \cdot s^{-1}$]
 \dot{V}_{air} objemový tok vzduchu [$m^3 \cdot s^{-1}$]
 ρ_{norm} normální hustota vzduchu [$kg \cdot m^{-3}$]
 h_1 entalpie stlačeného vzduchu na vstupu [$kJ \cdot kg^{-1}$]
 s_1 entropie stlačeného vzduchu na vstupu [$kJ \cdot kg^{-1} \cdot K^{-1}$]
 $h_{3, is}$ entalpie stlačeného vzduchu na výstupu při ideální expanzi [$kJ \cdot kg^{-1}$]
 T_1 teplota stlačeného vzduchu na vstupu [$^{\circ}C$]
 p_1 tlak stlačeného vzduchu na vstupu [$bar(a)$]

Ukázka .txt souborů vzniklých během měření

Zeitstempel	Drehzahl U/min Nm	Drehmoment Nm	Leistung Nm/h	Durchfluss DN 20		Durchfluss DN 40		Druck 1 Druck 2 Druck 3 Druck 4 Druck 5 Druck 6	Temperatur Thermoelement	Temperatur Leitung	Taupunkttemperatur
				bar	bar	bar	bar				
10:04:12	-12,026882	-0,042776	0,895529	0,055929	0,023665	0,988247	0,986300	0,979567	21,702578	21,974832	-14,208991
10:04:13	-18,023591	-0,042487	0,896192	0,056573	-0,023665	0,988203	0,988223	0,980112	21,716376	21,980239	-14,219443
10:04:14	-31,481227	-0,041646	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:15	-37,788251	-0,041562	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:16	-37,788251	-0,041562	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:17	-37,788251	-0,041562	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:18	-39,507746	-0,042651	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:19	-39,507746	-0,042651	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:20	-37,729416	-0,040873	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:21	-37,729416	-0,040873	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:22	-32,250235	-0,040661	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:23	-31,871868	-0,042816	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:24	-31,871868	-0,042816	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:25	-45,611745	-0,043801	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:26	-38,113920	-0,042816	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:27	-37,729416	-0,042816	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:28	-37,729416	-0,042816	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:29	-37,729416	-0,042816	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:30	-30,69338	-0,043469	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:31	-29,870886	-0,043473	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:32	-29,870886	-0,043473	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:33	-29,870886	-0,043473	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:34	-29,870886	-0,043473	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:35	-33,788251	-0,042539	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:36	-33,788251	-0,042539	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:37	-33,788251	-0,042539	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:38	-33,788251	-0,042539	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:39	-33,788251	-0,042539	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:40	-34,076629	-0,041394	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:41	-34,076629	-0,041394	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:42	-39,173282	-0,042539	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:43	-31,871868	-0,043144	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:44	-30,519968	-0,043096	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:45	-14,082426	-0,042487	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:46	-17,972465	-0,042567	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:47	-53,454075	-0,043308	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:48	-50,610296	-0,043068	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:49	-34,076629	-0,041394	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:50	-34,11133	-0,041574	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:51	-61,376405	-0,043637	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:52	-55,020991	-0,043445	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:53	-18,023591	-0,042323	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:54	-20,042237	-0,042627	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:55	-29,870886	-0,041995	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:56	-30,856409	-0,042079	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:57	-32,250235	-0,042539	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:58	-32,250235	-0,042539	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610
10:04:59	-37,729416	-0,043308	0,896534	0,056573	-0,023665	0,982171	0,988110	0,979879	21,685638	21,966895	-14,202610

Obrázek C.1: Ukázka souboru generovaného z Labview prostředí

Time	Voltage	Current	Power	RPMe	RPMo		
24.01.2020 10:05:32			7,192	9,457	68,014744	2739,14285714286	2720
24.01.2020 10:05:34			6,438	9,47	60,96786	2512 2554	
24.01.2020 10:05:34			6,844	9,442	64,621048	2662,57142857143	2665
24.01.2020 10:05:36			6,728	9,472	63,727616	2623,42857142857	2621
24.01.2020 10:05:36			7,25	8,951	64,89475	2694,28571428571	2761
24.01.2020 10:05:38			8,7	8,745	76,0815 3184	3123	
24.01.2020 10:05:39			9,164	8,76	80,27664	3244,28571428571	3258
24.01.2020 10:05:40			9,454	8,805	83,24247	3322 3310	
24.01.2020 10:05:41			9,454	8,716	82,401064	3330 3327	
24.01.2020 10:05:42			9,512	8,76	83,32512	3335,14285714286	3354
24.01.2020 10:05:43			9,976	8,716	86,950816	3508,28571428571	3478
24.01.2020 10:05:44			10,034	8,745	87,74733	3405,14285714286	3482
24.01.2020 10:05:45			10,208	8,997	91,841376	3580,28571428571	3536
24.01.2020 10:05:46			9,976	8,922	90,040824	3537,42857142857	3491
24.01.2020 10:05:47			9,628	9,07	87,32596	3420,28571428571	3356
24.01.2020 10:05:48			8,932	9,011	81,531528	3182,57142857143	3188
24.01.2020 10:05:49			8,642	9,056	78,261952	3078,57142857143	3106
24.01.2020 10:05:50			8,294	9,323	76,960026	3049,14285714286	3006
24.01.2020 10:05:51			8,004	9,307	74,012988	2958,57142857143	2924
24.01.2020 10:05:52			7,772	9,247	71,867684	2876,28571428571	2887
24.01.2020 10:05:53			7,714	9,277	71,562778	2870,28571428571	2868
24.01.2020 10:05:54			7,83	9,247	72,40401	2915,42857142857	2903
24.01.2020 10:05:55			7,946	9,263	73,603798	2920,28571428571	2953
24.01.2020 10:05:56			7,772	9,248	71,875456	2872,28571428571	2875
24.01.2020 10:05:57			7,888	9,263	73,066544	2925,14285714286	2907
24.01.2020 10:05:58			8,468	9,262	79,256072	3018 3087	
24.01.2020 10:05:59			8,526	9,011	76,827786	3036 3062	
24.01.2020 10:06:00			9,164	8,953	82,045292	3165,14285714286	3245
24.01.2020 10:06:01			9,454	8,745	83,18244	3319,42857142857	3339
24.01.2020 10:06:02			9,802	8,849	86,737898	3408 3413	
24.01.2020 10:06:03			10,092	8,745	88,25454	3552 3514	
24.01.2020 10:06:04			10,092	8,79	88,70868	3481,42857142857	3537
24.01.2020 10:06:05			10,034	8,864	88,941376	3458,85714285714	3506
24.01.2020 10:06:06			9,57	8,73	84,34128	3382,28571428571	3383
24.01.2020 10:06:07			9,512	8,745	83,18244	3358,57142857143	3385
24.01.2020 10:06:08			9,976	8,745	87,24012	3408 3505	
24.01.2020 10:06:09			10,15	8,789	89,20835	3630,28571428571	3541
24.01.2020 10:06:10			10,556	8,82	92,739332	3648,85714285714	3619
24.01.2020 10:06:11			10,15	8,804	89,3606 3544,57142857143	3489	
24.01.2020 10:06:12			10,44	8,404	87,73776	3601,42857142857	3588

Obrázek C.2: Ukázka souboru generovaného z Arduina

Obsah přiloženého CD

readme.txt	stručný popis obsahu CD
src	
├── impl.....	zdrojové kódy implementace
└── thesis	zdrojová forma práce ve formátu L ^A T _E X
text	
├── thesis.pdf	text práce ve formátu PDF
└── poster.pdf	poster ve formátu PDF

