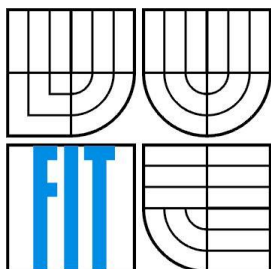


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ  
FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

# SYSTÉM PRO SPRÁVU SIMULÁTORŮ

SIMULATOR MANAGEMENT SYSTEM

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

AUTOR PRÁCE  
AUTHOR

Bc. Ievgen Ragulin

VEDOUCÍ PRÁCE  
SUPERVISOR

Mgr. Karol Rydlo

BRNO 2015

# Název diplomové práce v jazyce práce

## Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Mgr. Karola Rydla. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Ievgen Ragulin  
26.05.2015

## Poděkování

Rád bych poděkoval všem, kteří mně pomohli při vytváření mé diplomové práce. Především Mgr. Karolu Rydlovi za odbornou pomoc, podporu a rady kam směřovat moji práci. Většina praktické části projektu byla vypracována během pracovní stáže Erasmus v Karlsruhe, Německo ve společnosti CAS Software AG, oddělení CAS Aviation. Rád bych poděkoval Martinovi Buchheimu z CAS Software AG za poskytnutou odbornou pomoc.

© Ievgen Ragulin, ROK 2015

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů..*

## **Abstrakt**

Tato práce pojednává o základech simulace letu a systémech pro vizualizaci a správu letových dat. Cílem práce je vytvoření systému pro správu simulátorů letadel, umožňujícího zobrazovat a kontrolovat současný stav simulátorů, letových veličin a navigačních dat.

## **Abstract**

This thesis discusses the basics of flight simulation, systems for flight data visualization and management of flight data. The aim of the work is designing a system for simulator management, providing a capability for displaying and controlling a current state of simulators, flight and navigation data.

## **Klíčová slova**

Laboratoř SimStar, AeroWorks, simulace letu, systém pro správu simulátorů, vizualizace dat v prohlížeči

## **Keywords**

Laboratory SimStar, AeroWorks, flight simulation, simulator management system, data visualization in browser

## **Citace**

Ragulin Ievgen: Systém pro správu simulátorů, diplomová práce, Brno, FIT VUT v Brně, 2015

# Obsah

Obsah.....	1
1 Úvod.....	3
2 Seznam použitých zkratek .....	4
3 Problematika simulace letu .....	6
3.1 Organizace simulátoru letu .....	6
3.1.1 Pohybové rovnice .....	7
3.1.2 Aerodynamický model.....	7
3.1.3 Model motoru .....	7
3.1.4 Získání dat .....	7
3.1.5 Model podvozku .....	7
3.1.6 Model počasí.....	8
3.1.7 Vizuální systém .....	8
3.1.8 Zvukový systém.....	9
3.1.9 Pohybový systém .....	9
3.1.10 Kontrola zatížení.....	10
3.1.11 Zobrazení letových dat.....	10
4 Současné trendy vizualizace letových dat.....	13
4.1 Letové displeje dnešních letadel .....	13
4.1.1 Letové přístroje Boeing 737 NG.....	13
4.1.2 Systém letových instrumentů Garmin G1000.....	15
4.2 Systém EFIS&EMS Integra TL-6624.....	16
4.3 Výhled do budoucna .....	17
4.3.1 Displej vertikální situace (VSD).....	17
4.3.2 SVS .....	18
5 Prostředí leteckého simulátoru.....	20
5.1 Laboratoř EV-97 SportStar RTC .....	20
5.2 Laboratoř Boeing 737-800 NG .....	21
6 Implementace systému pro správu simulátorů.....	22
6.1 Popis systému pro správu simulátorů .....	22
6.1.1 Architektura systému pro správu simulátorů .....	22
6.1.2 Použité technologie.....	23
6.1.3 Odůvodnění zvolených technologií .....	24
6.1.4 Popis databázového modelu.....	24
6.1.5 Podporované prohlížeče.....	26

6.2	Uživatelské rozhraní systému pro správu simulátorů .....	26
6.2.1	Panel řízení letu .....	32
6.3	Popis činnosti aplikace .....	37
6.4	Protokol AWCom .....	37
6.5	Vylepšení výkonu aplikace .....	37
6.6	Vylepšení animace .....	39
7	Provedená testování .....	41
7.1.1	Testování pomocí generátoru falešných simulačních dat .....	41
7.1.2	Testování pomocí lokální instalace X-Plane a v laboratoři SimStar .....	42
8	Porovnání s existujícími systémy .....	44
8.1.1	FSPS Flight Simulator Manager .....	44
9	Možnosti budoucího rozšíření .....	47
10	Závěr .....	49
	Příloha A – komunikační protocol a posílane veličiny .....	53
	Příloha B. Příručka pro použití systému pro správu simulátorů .....	56
	Spouštění systému pro správu simulátorů .....	56
	Editování konfigurace aplikace .....	57
	Překlad kódu systému pro správu simulátorů .....	57
	Příloha C. Databázový model systému pro správu simulátorů .....	58
	Příloha D. Parametry simulovaných letounů .....	63
	Parametry letounu EV-97 Sportstar RTC .....	63
	Parametry letounu 737-800 .....	63

# 1 Úvod

Cílem mojí práce je vytvoření a popis systému pro správu simulátorů letadel, který by umožňoval instruktorovi, operátorovi v reálném čase sledovat přes webový prohlížeč stav simulátorů, vyhodnocovat dovednosti pilotu. Kromě toho, musí mít funkcionalitu pro zobrazení nejen současných, ale taky minulých létů, což může být použité instruktorem nebo pilotem pro zpětnou analýzu a vyhodnocení letu. Vytvořený systém umožňuje:

- přidávání, mazání, editace údajů o modelech letadel (např. Boeing 737, Airbus 320 apd.), modelech motorů a simulátorech
- získávání letových dat ze simulátoru X-Plane
- zobrazení a vizualizace informace o probíhajících simulacích na simulátorech X-Plane v prohlížeči, o současném stavu letových přístrojů, motorů, palivových nádrží, zobrazení na mapě dráhy pohybu simulovaného letounu a zobrazení výškového a rychlostního profilu letu.
- ukládání informací o probíhajících simulacích do databáze, což umožňuje uživatelům nejen vidět současné léty, ale taky minulé pro zpětnou analýzu a vyhodnocení.

V kapitolách 3, 4, 5 se analyzuje problematika simulace letu, popisuje se prostředí leteckého simulátoru, ve kterém byla diplomová práce vypracovaná. V kapitole 6 je popsán vytvořený systém pro správu simulátorů. Kapitola 7 obsahuje informace o provedených testováních. V kapitole 8 je systém porovnán s existujícími podobnými systémy. Kapitola 9 pojednává o možnostech budoucího rozšíření. Práce obsahuje také přílohy A, B, C, D.

Diplomová práce navazuje na semestrální projekt, řešený na FIT VUT v roce 2013-2014. Kapitoly 3, 4, 5 byly převzaty z tohoto semestrálního projektu.

## 2 Seznam použitých zkratek

<b>ACFD</b>	<b>Advanced Civil Flight Deck</b>	<b>Pokročila přístrojová deska pro civilní letouny</b>
<b>AG</b>	<b>Aktiengesellschaft</b>	<b>Akciová společnost</b>
<b>API</b>	<b>Application Programming Interface</b>	<b>Rozhraní pro programování aplikací</b>
<b>AWCom</b>	<b>AeroWorks communication protocol</b>	<b>AeroWorks komunikační protokol</b>
<b>AWT</b>	<b>Abstract Window Toolkit</b>	<b>Knihovna grafických uživatelských prvků</b>
<b>BAC</b>	<b>British Aircraft Corporation</b>	<b>Britská letadlová korporace</b>
<b>CHT</b>	<b>Cylinder head temperature</b>	<b>Teplota hlav válců</b>
<b>DEUTA</b>	<b>Deutsche Tachometer-Werke GmbH</b>	<b>Offenbachová rychloměrová díla s.r.o.</b>
<b>DU</b>	<b>Display Unit</b>	<b>Zobrazovací jednotka</b>
<b>EFIS</b>	<b>Electronic Flight Instrument System</b>	<b>Elektronický letový informační systém</b>
<b>EGT</b>	<b>Exhaust Gas Temperature</b>	<b>Teplota výfukových plynů</b>
<b>EMS</b>	<b>Engine Monitoring System</b>	<b>Systém monitorování motorů</b>
<b>ENG</b>	<b>Engine</b>	<b>Motor</b>
<b>FFT</b>	<b>Fast Fourier Transform</b>	<b>Rychlá Fourierova Transformace</b>
<b>FSPS</b>	<b>Flight Simulator Platform Solutions</b>	<b>Řešení pro platformu letových simulátorů</b>
<b>FPS</b>	<b>Frames Per Second</b>	<b>Snímku za sekundu</b>
<b>GmbH</b>	<b>Gesellschaft mit beschränkter Haftung</b>	<b>Společnosti s ručením omezeným</b>
<b>GUI</b>	<b>Graphical User Interface</b>	<b>Grafické uživatelské rozhraní</b>
<b>HSI</b>	<b>Horizontal Situation Indicator</b>	<b>Indikátor horizontální situace</b>
<b>HTML</b>	<b>HyperText Markup Language</b>	<b>Hypertextový značkový jazyk</b>
<b>HTTP</b>	<b>HyperText Transfer Protocol</b>	<b>Hypertextový přenosový protokol</b>
<b>IP</b>	<b>Internet protocol</b>	<b>Internetový protokol</b>
<b>JIRA</b>	<b>Project management software</b>	<b>Softwarový nástroj na řízení projektu</b>
<b>LCD</b>	<b>Liquid-Crystal Display</b>	<b>Displej z tekutých krystalů</b>
<b>MFD</b>	<b>Multi-function display</b>	<b>Multifunkční displej</b>
<b>NASA</b>	<b>National Aeronautics and Space</b>	<b>Národní úřad pro letectví a</b>

	<b>Administration</b>	<b>kosmonautiku</b>
<b>ND</b>	<b>Navigation Display</b>	<b>Navigační displej</b>
<b>NG</b>	<b>New Generation</b>	<b>Označuje novou generaci např. letounu</b>
<b>OTA</b>	<b>Offenbacher Tachometer-Werke GmbH</b>	<b>Offenbachová tachometrová díla s.r.o.</b>
<b>PFD</b>	<b>Primary Flight Display</b>	<b>Primární letový displej</b>
<b>PHP</b>	<b>PHP Hypertext Preprocessor</b>	<b>PHP hypertextový preprocesor</b>
<b>PRI</b>	<b>Primary</b>	<b>Primární</b>
<b>RTC</b>	<b>Restricted Type Certificate</b>	<b>Certifikát omezeného typu</b>
<b>RIA</b>	<b>Rich Internet Application</b>	<b>Bohatá internetová aplikace</b>
<b>RPM</b>	<b>Revolutions Per Minute</b>	<b>Otáčky za minutu</b>
<b>SM</b>	<b>Simulator Management System</b>	<b>Systém pro správu simulátorů</b>
<b>SQL</b>	<b>Structured Query Language</b>	<b>Strukturovaný dotazovací jazyk</b>
<b>SVG</b>	<b>Scalable Vector Graphics</b>	<b>Škálovatelná vektorová grafika</b>
<b>SVS</b>	<b>Synthetic Vision System</b>	<b>Systém syntetického vidění</b>
<b>TCP</b>	<b>Transmission Control Protocol</b>	<b>Přenosový řídicí protokol</b>
<b>URL</b>	<b>Uniform Resource Locator</b>	<b>Řetězec znaků sloužící k specifikaci umístění informací</b>
<b>VDO</b>	<b>Vereinigte DEUTA - OTA</b>	<b>Sjednocená DEUTA - OTA</b>
<b>VSD</b>	<b>Vertical Situation Display</b>	<b>Displej vertikální situace</b>
<b>XM Radio</b>	<b>No Modulation Radio (satellite radio)</b>	<b>Rádio bez modulace (satelitní rozhlas)</b>

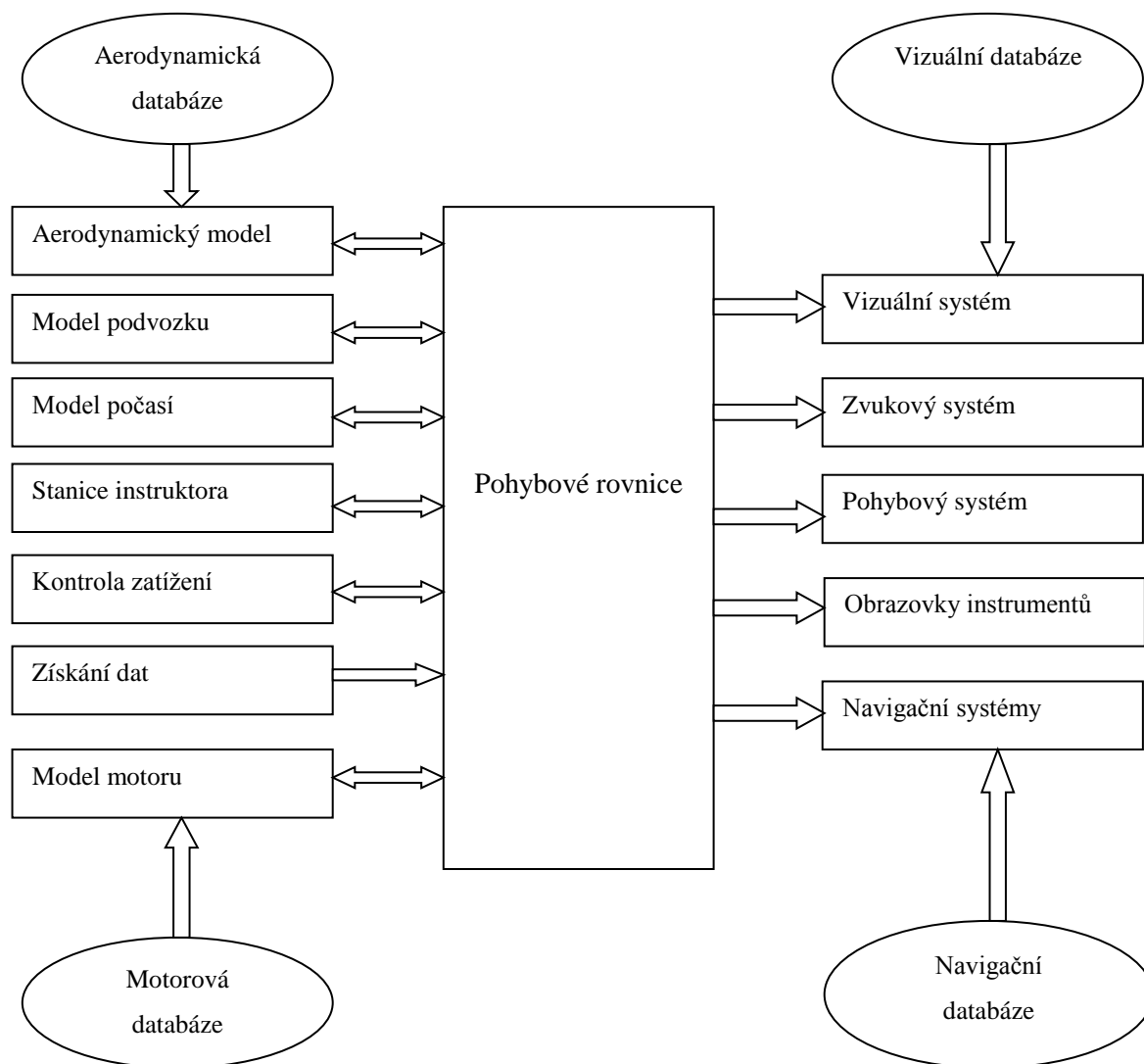


### 3 Problematika simulace letu

V této kapitole je popsána problematika simulace letu, a organizace simulátoru letu.

#### 3.1 Organizace simulátoru letu

David Allerton [1] navrhuje dobrý přehled technik simulací používaných v dnešních simulátorech letu. Nejdůležitější komponenty typického simulátoru jsou uvedeny na Obr. 1.



Obr. 1. Organizace simulátoru letu [1]

### **3.1.1 Pohybové rovnice**

Pohybové rovnice jsou ústředním bodem všech simulátorů letu. Určují stavy simulátoru. Po získání všech vstupů, včetně těch od pilota, simulátor spočítá proměnné, které popisují stav simulovaného letadla, zejména síly, momenty, polohu, nadmořskou výšku, kurs a rychlosti [1].

### **3.1.2 Aerodynamický model**

Aerodynamický model dává možnost vypočítat aerodynamické síly a momenty. Například, koeficient vztlaku může být získán pomocí funkce úhlu náběhu, kde konkrétní aerodynamické koeficienty jsou definované v aerodynamické databázi [1].

### **3.1.3 Model motoru**

Data motoru jsou závislá na stavu letadla. Implementace modelu motoru potřebuje přístup k proměnným, které byly spočítané v leteckém modelu. Model motoru používáme na získání tahu motoru, průtoku paliva, motorového tlaku, rychlosti otáček atd. [1].

### **3.1.4 Získání dat**

Plný letecký simulátor pilotní kabiny je přesnou kopií kabiny letadla. Navíc primární ovladače letu (řídící paky, kormidla, brzdy, klapky, podvozek atd.), obecně každá páka a přepínač musí být propojeny s odpovídajícím modulem simulátoru. Každý ovladač poskytuje digitální nebo analogová data o svém stavu. Současná poloha ovladačů musí být vzorkována pro každý simulační krok a předána odpovídajícímu modulu simulátoru. Plný letecký simulátor může mít několik stovek vstupů [1].

### **3.1.5 Model podvozku**

Během rolování se podvozek dotýká země a vzájemná působení mezi vzletovou a přistávací dráhou, pneumatikami, koly a vzpěrou podvozku vedou k různým dynamikám letadla během letu. Obvykle se poskytuje dodatečný model pro simulace efektu pneumatik a podvozku. Simulace ovládání letadla během vzletu, přistání a rolování je kritickým momentem z hlediska přesnosti. Nouzové situace, které procvičují piloti, zahrnují pozemní incidenty: prasknutí pneumatik, selhání motoru, námraza vzletové a přistávací dráhy, selhání zpětného tahu, přehřátí brzd, aquaplaning a další [1].

V praxi jsou do simulace dodány přechody mezi stavy před vzletem a hned po přistání, kde je aerodynamický vliv podvozku propojený s dynamikou podvozku [1].

### 3.1.6 Model počasí

Atmosféra má významný vliv na výkonnost letadla. Rovnice pro vypočítání tlaku vzduchu, hustoty vzduchu, teploty vzduchu jsou dobře známy a přímočaré pro implementaci. Tyto složky se používají v aerodynamickém modelu a modelu motoru. Kromě toho se modelují:

- Vítr. Vítr hraje důležitou roli při plánování letu operátorem letecké společnosti, takže model větru musí být prostorový a měnící se v čase.
- Turbulence. Turbulence také hraje důležitou roli při výcviku pilotů. Piloti potřebují demonstrovat schopnost ovládnutí letadla za podmínek turbulence. Normálně se turbulence generuje jako množina pseudonáhodných procesů, způsobujících odchylky ve třech lineárních osách letadla.
- Střih větru.
- Nebezpečné letové podmínky, zejména, námraza a hustý déšť. Námraza na trupu letadla a rozbředlý sníh na vzletové a přistávací dráze může zvýšit odpor během vzletu, zatímco sníh a voda na vzletové a přistávací dráze mohou snížit účinnost brzdění a řízení [1].

### 3.1.7 Vizuální systém

Vizuální systém poskytuje řadu kanálů obrazů, které se generují v reálném čase, viditelných z pozice pilota. Z počátku je databáze objektů nahrána do paměti vizuálního systému. Databáze může obsahovat pole, letiště, silnice, jezera, pobřežní čáry, dopravní prostředky, budovy, stromy, lesy a letadla. Existují rozličné standardy pro generování těchto objektů, kde např. OpenFlight je jedním z nejpoužívanějších formátů. V závislosti na vzdálenosti k objektům se zobrazují různé úrovně detailů [1]. Příklad vizuálního systému simulátoru je vidět na Obr. 2



Obr. 2. Google Earth simulátor letu

Když letadlo manévruje, se pozice oka pilota a směr vypočítají pomocí rovnic pohybu a scéna je renderována pro jednotlivé snímky [1].

Každý video výstup má jistý posun. Například, máme tři video výstupy, pokrývající úhel 180°. Přední kanál má ofset 0°, dva zbývajících mají -60°, +60°. V praxi se projektované kanály překrývají o několik stupňů, abychom se vyhnuli viditelným mezerám [1].

### 3.1.8 Zvukový systém

Pro generování zvuků v pilotní kabině se používají dvě metody. První způsob spočívá v nahrávání reálných zvuků letadla uvnitř pilotní kabiny. Nedostatek toho přístupu spočívá v počtu záznamů, nutných pro obsáhnutí všech letových podmínek. Například, zvuky motorů se mohou měnit podle rychlosti, výšky, stavu motorů (otáček a tahu), a tudíž by bylo zapotřebí pokrýt všechny možné kombinace [1].

Alternativním a používanějším způsobem generování zvuků je analyzovat zdroj všech zvuků a vygenerovat pro ně odpovídající zvukovou stopu. Tyto zvuky se zkombinují s bílým šumem. Synteticky generované zvuky můžeme porovnat se skutečnými zvuky pomocí Fourierovy transformace (FFT) [1].

### 3.1.9 Pohybový systém

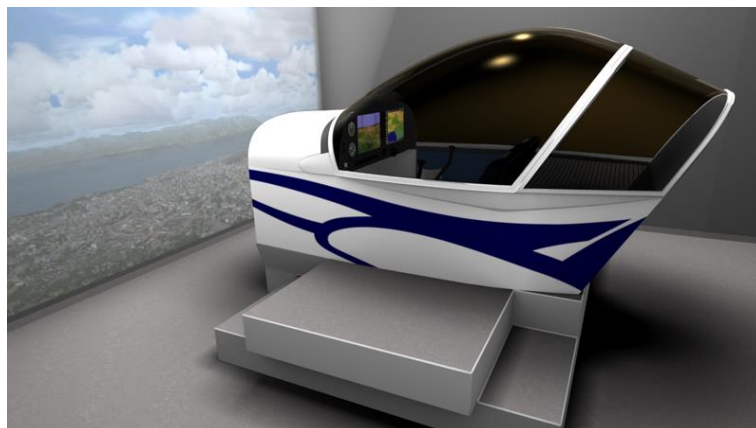
Během manévrování simulovaným letadlem, bude pilot očekávat pocit akcelerace, který by zažil během skutečného letu. Akcelerace se vypočítají v modelu letu a předávají se pohybovému systému. V standardní pohybové platformě obsahující šest hydraulických aktuátorů se každý aktuátor přesouvá na novou pozici, aby zkusil replikovat akcelerace na tělo pilota [1].

Opravdový pohyb samozřejmě nemůže být generován. Nicméně pohybové senzory člověka se dají oklamat. Mozek reaguje na začátek pohybu, ale nemůže detekovat velmi malé míry pohybu. Navíc, kvůli vizuálnímu systému pilot může cítit pohyb, který se skutečně na pohybovou platformu neaplikoval [1].

Simulátor s pohybovým systémem je na Obr. 3. Také existují simulátory bez pohybového systému, příklad je na Obr. 4.



Obr. 3. ATR 42-300 - simulátor s pohybovým systémem [16]



Obr. 4. Simulátor Evektor SportStart bez pohybového systému [17]

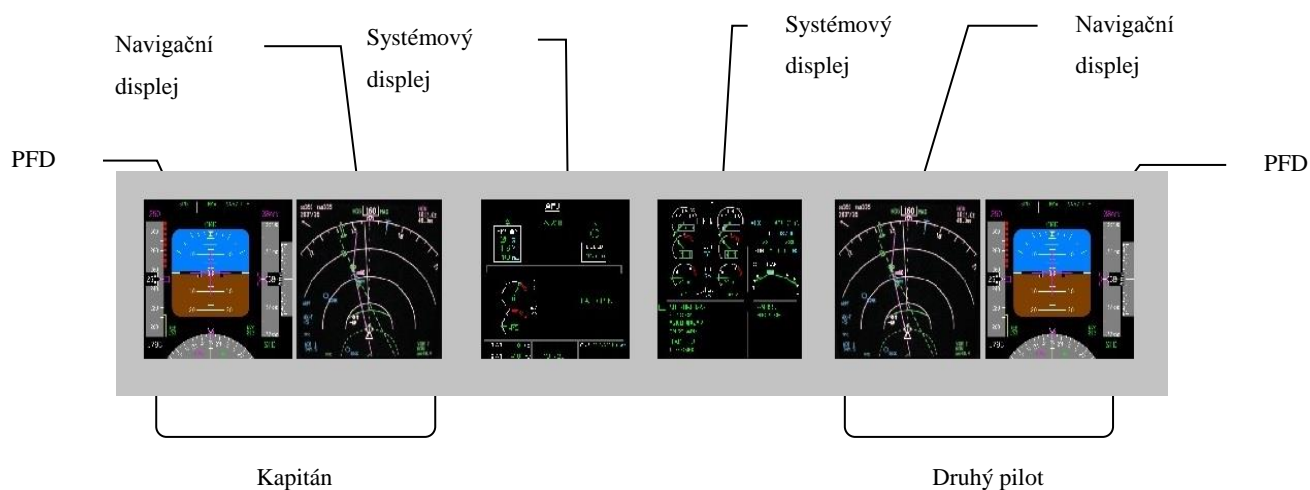
### 3.1.10 Kontrola zatížení

Když letadlo letí, vzdušný proud obtéká povrch ovladačů a to ovlivňuje stupeň zatížení na jejich površích, zejména na primárních letových ovládacích: výškových kormidlech, křídélkách, kormidlech. Síla, kterou musíme aplikovat na ovladač, záleží na rychlosti. Tyto efekty se také musí simulovat a to je funkce systému kontroly zatížení. Kontrola zatížení se provádí spojením aktuátorů s ovladači letu v simulátoru takovým způsobem, aby aktuátor poskytoval odpor k pohybu; odpor se obecně mění s rychlosti [1].

### 3.1.11 Zobrazení letových dat

Avionické systémy slouží pro zobrazení fyzikálních veličin a důležitých parametrů letu. Avionické systémy zahrnují dvě období letu. Před rokem 1980 měla většina letadel mechanické systémy. Mnohé z těchto přístrojů obsahovaly komplikované mechanismy. Od roku 1980 mnohá civilní a vojenská letadla používají elektronické systémy (EFIS). EFIS displeje jsou založeny na počítačové grafice s osmipalcovými monitory, obecně s obrazovkami, které se aktualizují nejméně dvacetkrát za vteřinu (20 Hz) [1].

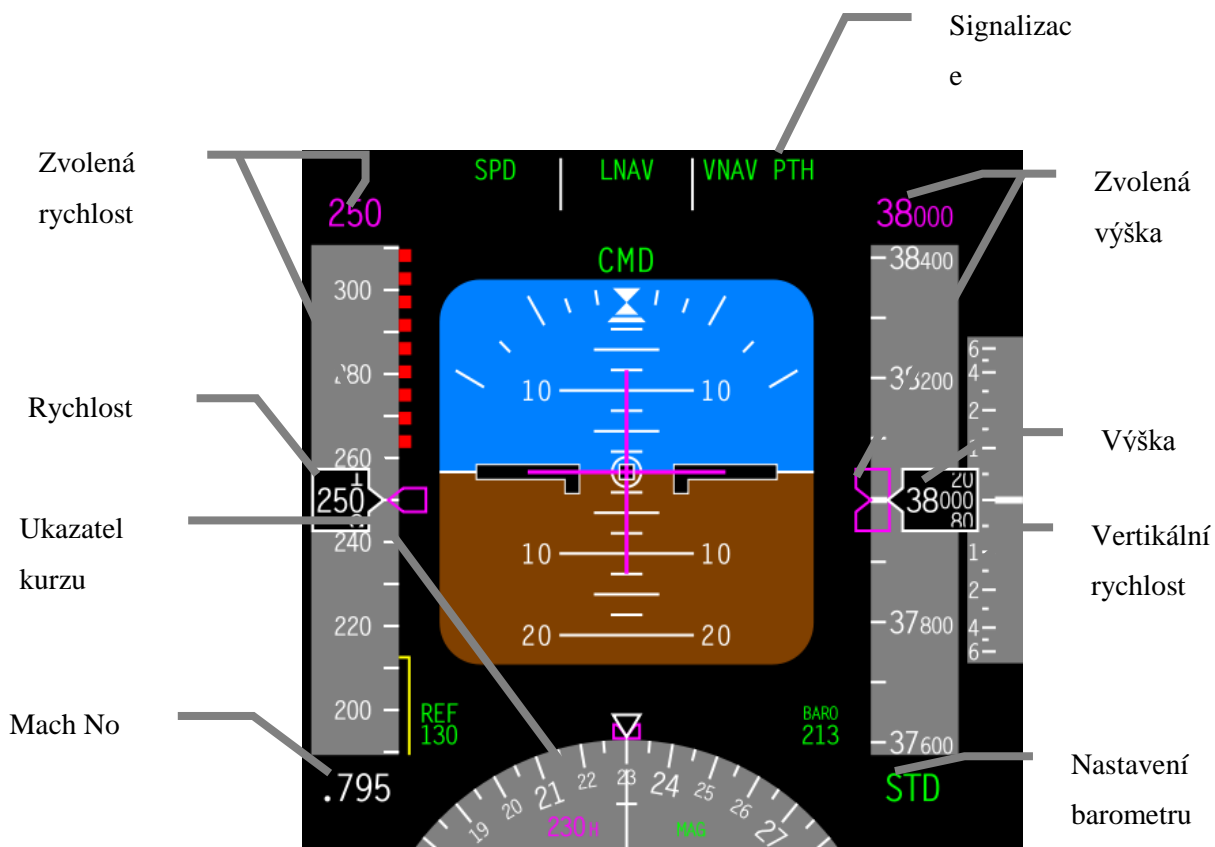
V 70. letech britská vláda podporovala program známý pod názvem Advanced Civil Flight Deck (ACFD), neboli pokročilá přístrojová deska pro civilní dopravní letouny. Tento systém měl demonstrovat použití přístrojové desky s PFD (Primary Flight Display) a ND (Navigation Display) displeji. Rozložení takových displejů je vidět na Obr. 5. Jedním z prvních testovacích strojů byl letoun BAC 1-11, který obsahoval jeden primární letový displej a jeden navigační displej. Tento letoun můžeme považovat za začátek expanze elektro-optických přístrojů do kokpitů moderních letounů.



Obr. 5. Pokročilá civilní přístrojová deska [2]

### 3.1.11.1 Primární letový displej (PFD)

Primární letový displej je moderní letecký instrument pro zobrazení letových dat. Na palubní desce jsou dva primární letové displeje: pro prvního a druhého pilota. Detaily rozložení displejů se mohou lišit v závislosti na letadle, ale většina PFD dodržuje podobné konvence. Popis jednotlivých přístrojů můžeme vidět na Obr. 6



Obr. 6. Hlavní letový displej PFD, Boeing 747-400 [18]

### 3.1.11.2 Navigační displej (ND)

Navigační displej poskytuje navigační, směrové, provozní a povětrnostní informace. Na palubní desce jsou dva navigační displeje: pro prvního a druhého pilota. Příklad navigačního displeje je na Obr. 7.



Obr. 7. Navigační displej Boeing 777 [20]

## 4 Současné trendy vizualizace letových dat

V kapitole 4 jsou popsány současné trendy vizualizace letových dat. Uvedené příklady jsou ukázkou toho, jak probíhá vizualizace letových dat na dnešních letounech

### 4.1 Letové displeje dnešních letadel

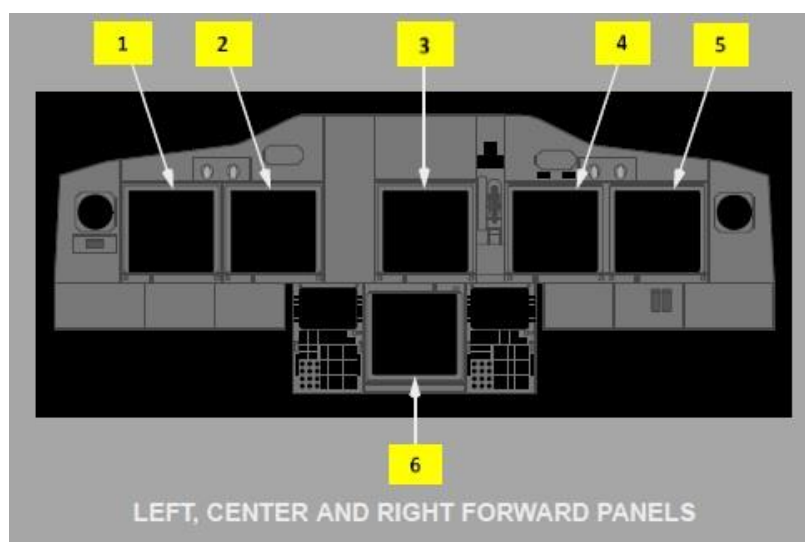
V této kapitole prozkoumáme, jak vypadají letové displeje velkých letadel (Boeing 737) střední třídy (na příkladu displeje Garmin G1000) a menší třídy (na příkladu displeje EFIS&EMS Integra TL-6624).

#### 4.1.1 Letové přístroje Boeing 737 NG

Na příkladu letových instrumentů Boeing 737 podíváme se, jak v současné době vypadají letové instrumenty letadel větších rozměrů.

##### 4.1.1.1 EFIS/Map displejový systém – přehled

Na Obr. 8 je zobrazená schéma displejového systému Boeing 737 NG.



Obr. 8. PFD/ND displejový systém – přehled [6]

- |                                   |                                    |
|-----------------------------------|------------------------------------|
| 1) Vnější displej prvního pilota  | 4) Vnitřní displej druhého pilota. |
| 2) Vnitřní displej prvního pilota | 5) Vnější displej druhého pilota.  |
| 3) Horní displej.                 | 6) Dolní displej.                  |



#### 4.1.1.2 Hlavní displeje

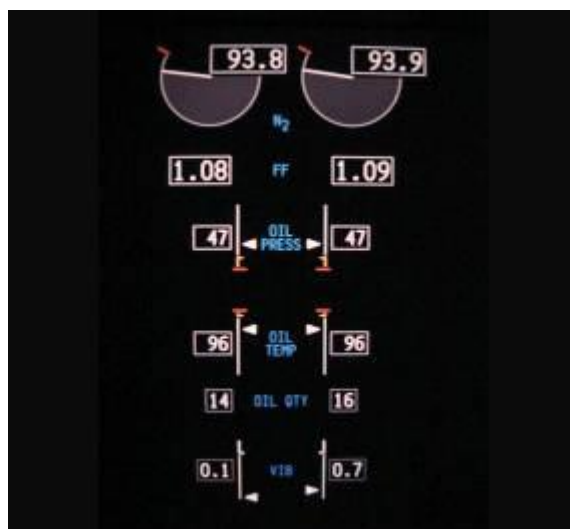
Pilot i kopilot mají k dispozici pro zobrazení letových dat displeje PFD a ND, které jsou blíže popsány v předchozí kapitole.

#### 4.1.1.3 Horní a dolní displeje

Horní displej (viz Obr. 9) je primární displej motorů, dolní displej (viz Obr. 10) je sekundární displej motorů.

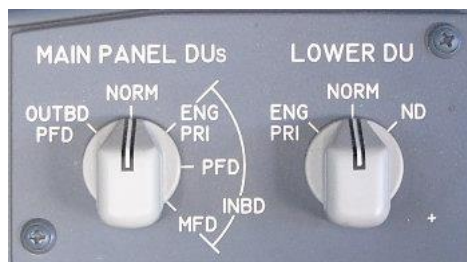


Obr. 9. Horní displej v Boeing 737 NG [49]



Obr. 10. Dolní displej v Boeing 737 NG [49]

V případě poruchy horního displeje lze displej motorů zobrazit na dolním displeji. Můžeme nastavit přepínač Spodní DU v polohu ENG PRI (viz Obr. 11), a tím pádem zobrazit displej motorů na dolním displeji namísto na horním.



Obr. 11. Boeing 737 DU selektor [7]

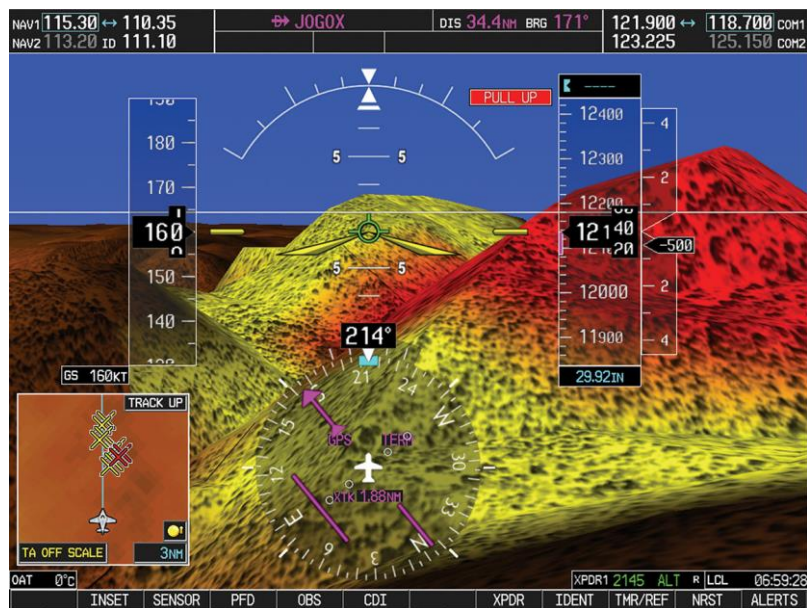
#### 4.1.2 Systém letových instrumentů Garmin G1000

Garmin G1000 je integrovaným systémem letových instrumentů, který se používá ve mnoha dnešních letadlech střední třídy, jako třeba Cessna [8][9], Diamond [9], Beech King Air [9]. G1000 může mít 2 nebo 3 LCD displejů, každý z nich se může používat jako hlavní letový displej (PFD) nebo multifunkční displej (MFD) [9]. Na Obr. 12 je ukázka letových instrumentů G1000 v letadle Cessna 182.



Obr. 12. G1000 v Cesna 182 [9]

Na Obr. 13 je ukázka primárního letového displeje Garmin G1000. PFD ukazuje základní letové instrumenty jako rychlosti letu, výškoměr, ukazatel kurzu, umělý horizont, vertikální rychlost atd.



Obr. 13. Garmin G1000 primární letový displej [9]

Multifunkční displej Garmin G1000 typicky zobrazuje hýbající se mapu po pravé straně a informace o motorech po levé straně. Jiné možné obrazovky zahrnují informace o nejbližších letištích, navigačních pomůckách, XM rádio, programování plánu letu a jiné [9]. Na Obr. 14 je ukázka multifunkčního displeje Garmin G1000.



Obr. 14. Garmin G1000 multifunkční displej [10]

## 4.2 Systém EFIS&EMS Integra TL-6624

Příkladem instrumentů letadel malé třídy je EFIS&EMS Integra TL-6624. EFIS&EMS Integra TL-6624 je multifunkční systém který se používá pro sledování letu a motorů. TL-6624 integruje všechny hlavní letové instrumenty (výškoměr, indikátor vertikální rychlosti, indikátor rychlosti letu, kompas,

chronometr, indikátor úhlu náběhu, vnitřní teplotu vzduchu, vnější teplotu vzduchu). EFIS Integra také zahrnuje navigační systémy (HSI – indikátor horizontální situace) a celosvětové mapy terénu. Také integruje všechny primární motorové instrumenty (otáčky za minutu, indikátor průtoku paliva, indikátory teploty, množství paliva atd.) [12]. Na Obr. 15 je ukázka EFIS&EMS Integra displeje.



Obr. 15. EFIS&EMS Integra TL-6624 [11]

## 4.3 Výhled do budoucna

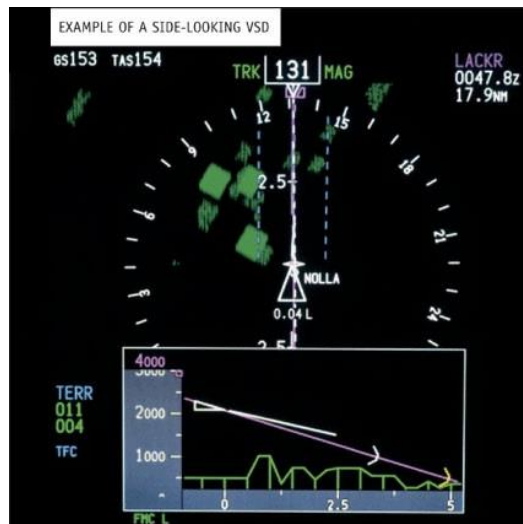
V této kapitole jsou popsány současné trendy vizualizace letových dat.

### 4.3.1 Displej vertikální situace (VSD)

V současné době musí posádka letadla získávat informace z různých zdrojů pro pochopení vertikální situace letu. Tyto zdroje zahrnují barometrické a radiolokační výškoměry, indikátor vertikální rychlosti, ground proximity warning systém, navigační informace, navigační grafy [2].

Pro účely prevence řízeného letu do terénu a nehod při přistání vytvořil Boeing VSD - grafický obraz vertikální dráhy letadla, který zlepšuje celkové povědomí posádky letadla o situaci. VSD se zobrazuje v dolní části navigačního displeje, pokud je vybrán posádkou letadla, [3].

Obr. 16 ukazuje příklad VSD na Boeingu. Základními prvky VSD jsou výšková informace a škála horizontální vzdálenosti, symbol letadla, vektor vertikální dráhy pohybu, vyobrazení terénu, zvolená výška, zvolená vertikální rychlost, navigační pomůcky a rozličné informace, zvolené posádkou letadla [3].



**Obr. 16. Displej vertikální situace [3]**

### 4.3.2 SVS

Současné displeje mají velké množství čísel, písmen a symbolů. Tyto popisy parametrů jsou odvozené od nástrojů používaných vědci a inženýry pro abstrakci principů reality pro matematickou analýzu. Účelům, pro které byly navrženy, fungují dobře, ale pilot potřebuje neustále přemýšlet tak, aby s pomocí 2D informace, která se poskytuje na PFD a ND displejích, vytvořil ve své hlavě 3D/4D obraz reálného světa [2].

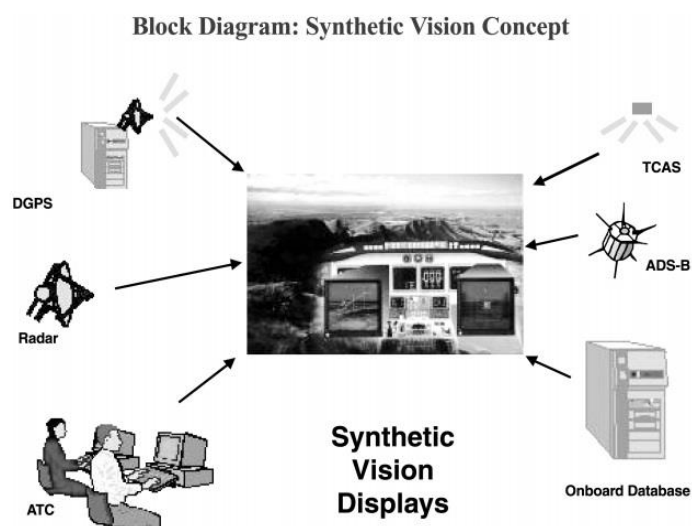
Cílem SVS je poskytovat pilotům intuitivní 3D informace v reálném čase. Díky přesné znalosti současné pozice letadla, databázi terénu a dnešním možnostem počítačové grafiky, je možný vytvořit přesnou 3D reprezentaci terénu, který je vidět z pilotní kabiny [2]. SVS systémy jsou určeny také pro snížení pravděpodobnosti nehod zlepšením situačního a prostorového povědomí pilota za špatné viditelnosti [4]. Obr. 17 ukazuje příklad SVS displeje.





Obr. 17. SVS displej vyvinutý NASA [5]

Obr. 18 ilustruje koncept potenciálního SVS. Tento koncept displeje umožňuje prezentace 3D scén, které obsahují nutnou a dostačující informaci a realismus, aby vygenerovaný obraz byl skoro ekvivalentní reálnému obrazu za denního světla. Symbolická informace, například prezentace umělého horizontu, směru pohybu, indikátory výšky, může být přidána na SVS displej pro zvětšení situačního povědomí [4].



Obr. 18. Možný SVS [4]

Pokud je na displeji zobrazen předpovězený výsledek manévru, dá se říci, že displej je reprezentací 4D scény, protože zahrnuje časovou složku. V 1995 roce Mnichovská univerzita v Německu spolu s VDO Luftfahrgeräte Werk of Frankfurt realizovala program, v důsledku kterého bylo letadlo vybaveno 4D displeji, na kterých se zobrazovala vyhodnocení pojezdové dráhy, přistání, vzletnutí, přiblížení a manévru vyhýbání terénu [2].

## 5 Prostředí leteckého simulátoru

Systém pro správu simulátorů byl implementován v prostředí leteckých simulátorů EV-97 SportStar RTC a Boeing-737 NG. Pro systém pro správu simulátorů je zapotřebí vědět rozsahy vstupních parametrů jednotlivých letounů. V této kapitole je přehled jednotlivých modelů simulátorů a jejich technické parametry.

### 5.1 Laboratoř EV-97 SportStar RTC

Jedná se o laboratoř na VUT FIT kde se nachází kokpit letounu EV-97. Má trup ze skutečného letadla EV-97 SportStar. Obsahuje projekční systém 4m:3m. V této laboratoři se dají řešit projekty v oborech designu letecké elektrotechniky, rozšířené reality, počítačové grafiky, designu systémů řízení letu, hardware designu, modelování a simulace, zpracování signálů a vývoje software. Na Obr. 19 je ukázka skutečného letounu EV-97 SportStar RTC. Na Obr. 20 je simulátor z laboratoří VUT FIT. Příloha D obsahuje parametry tohoto letounu potřebné pro panel řízení letu SM.



Obr. 19. EV-97 SportStar RTC [19][20]



Obr. 20. Simulátor EV-97 Sportstar RTC na VUT FIT [21]

## 5.2 Laboratoř Boeing 737-800 NG

V této laboratoři se nachází kokpit letounu Boeing 737 NG, kde je možné řešit projekty v oblasti počítačové grafiky, vestavěných systémů, sítí a bezpečnosti. Pro vizualizaci slouží cylindrický projekční systém, který má 5 metrů v průměru a 3 metry na výšku. K osvětlení této plochy jsou použity 3 projektory. Pro komunikaci s displeji a simulací je použit protokol AWCCom. Na Obr. 21 je ukázka kokpitu skutečného Boeing 737-800 NG. Na Obr. 22 je simulátor z laboratoří FIT VUT. Příloha D obsahuje parametry tohoto letounu, potřebné pro panel řízení letu SM.



Obr. 21. Kokpit skutečného Boeing-737-800 NG [13].



Obr. 22. Simulátor Boeing-737-800 NG na FIT VUT [21]



## **6 Implementace systému pro správu simulátorů**

Tato kapitola popisuje naimplementovaný systém pro správu simulátorů (SM), jeho architekturu, použité technologie, databázový model, způsob komunikace se simulátory, použité technologie a návod na použití

### **6.1 Popis systému pro správu simulátorů**

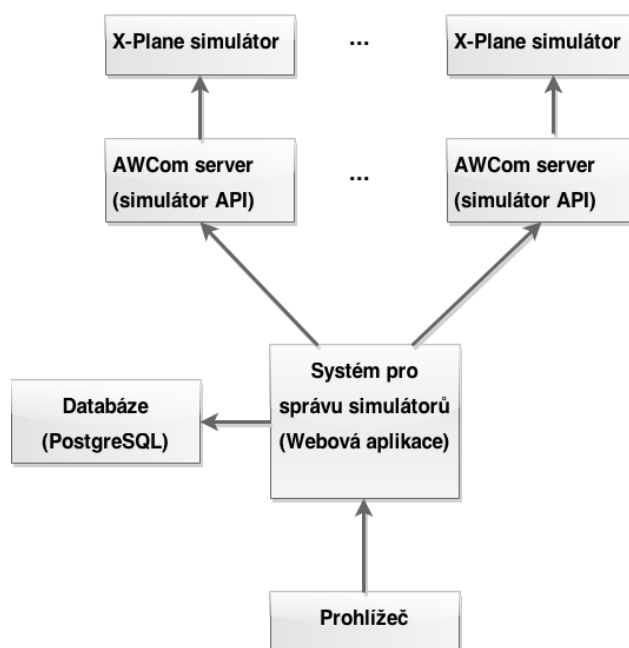
V této kapitole jsou popsány použité technologie, architektura vytvořeného systému, popis databázového modelu a podpora prohlížečů.

#### **6.1.1 Architektura systému pro správu simulátorů**

Systém pro správu simulátorů je vyvinutý v programovacím jazyce Java, pomocí frameworku Vaadin. Aplikace běží na Webovém serveru Apache Tomcat. SM získává data ze simulátorů přes protokol ACom. Jeho popis je uveden v Příloze A. SM používá databáze PostgreSQL pro ukládání konfigurací simulátorů a dat o simulacích. Uživatel přistupuje k SM přes prohlížeč.

Systém pro správu simulátorů získává data z každého simulátoru v samostatném vlákne. První implementace měla jen jedno vlákno, ale po testování aplikace na víc než jednom simulátoru jsme zaznamenali podstatné zhoršení výkonu aplikace. Tím pádem, jednomu simulátoru odpovídá jedno vlákno. Pro tento účel aplikace používá Javový Thread pool, což znamená, že SM nevytváří nové vlákno pokaždé, když potřebuje získat data ze simulátoru, ale vytváří je jen po spouštění aplikace a při změně počtu simulátorů.

Framework Vaadin je užitečný pro webovou část aplikace: pro předávání dat mezi klientskou a serverovou částí aplikace, komunikaci s databází, pro zobrazování standardních prvků uživatelského rozhraní (tabulky, menu, formy, tlačítka, notifikační zprávy).



Obr. 23. Architektura systému pro správu simulátorů

## 6.1.2 Použité technologie

Systém pro správu simulátorů byl vyvinutý pomocí následujících technologií:

- X-Plane 9.00r1 – letový simulátor [28].
- Java 8 [29] – backend aplikace, komunikace ze simulátory přes TCP/IP socket, práce z databází.
- Vaadin 7.3 [30] – framework pro tvorbu aplikace Simulator Manager, pomocí kterého se aplikace v internetovém prohlížeči zobrazuje a chová stejně jako desktopová aplikace (RIA).
- Apache Tomcat 8.0 [31] – webový server potřebný pro běh aplikace.
- JavaScript, HTML5, Canvas – klientská strana aplikace (front end).
- PostgreSQL 9.3 [32] – databázový systém použitý pro ukládání dat.
- TCP/IP socket – komunikace ze simulátory.
- Google Maps API v2 [33] – použité pro zobrazení dráhy letu na mapě.
- Google Charts [34] – použité pro animaci ukazatelů stavu paliva, motorů.
- Highcharts [35] – výškový a rychlostní profily letu.
- GitHub [36] – verzovací systém, plánování práce.
- JIRA [37] – plánování práce během Erasmus stáže v Karlsruhe.
- GreenSock Animation Platform [38] – knihovna pro HTML5 animace.
- DbVizualizer [39] – vizualizace databázového modelu.
- Java AWT [43], Java Swing [44] – knihovny použité pro vývoj pomocné desktopové aplikace pro generování falešných dat simulace.

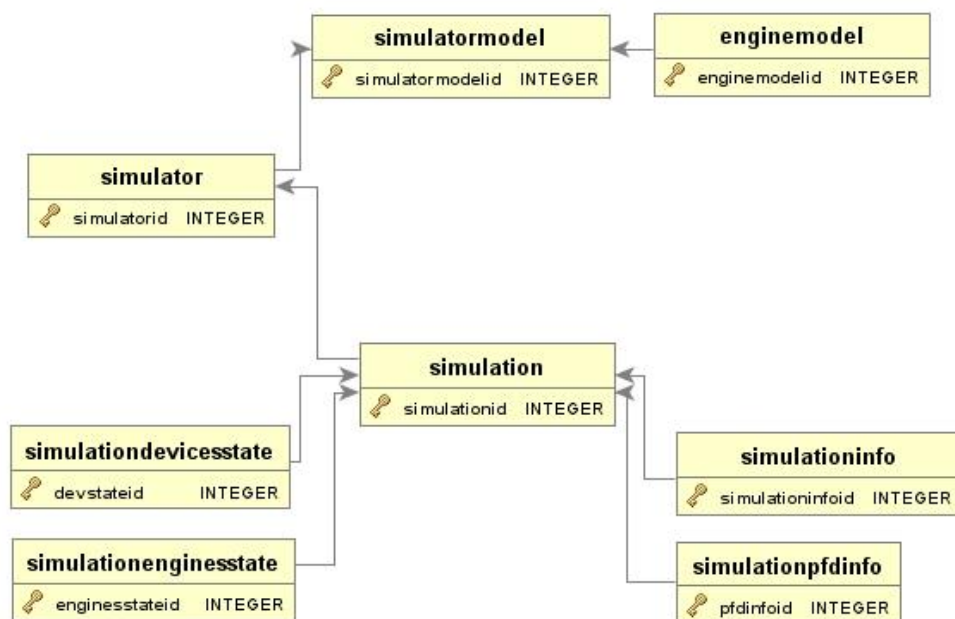
### 6.1.3 Odůvodnění zvolených technologií

Účelem této práce bylo vytvořit systém pro správu simulátorů, který by byl pohodlný pro uživatele, nevyžadoval instalaci, umožňoval používat různé operační systémy a zařízení. Proto bylo rozhodnuto systém vyvinout jako webovou aplikaci. Architekturu aplikace se dá rozdělit na tři části: databáze, serverová část a klientská část.

- PostgreSQL pro databáze. Kvůli tomu, že simulátory posílají podobná data (databázové schéma se nemění), je vhodné používat relační databázi pro ukládání dat. Místo PostgreSQL SM by mohl použít jakýkoliv jiný databázový systém, například MySQL.
- Java 8 pro serverovou část. Vytvořený SM neobsahuje nějaké specifické funkce, které by se dali naimplementovat jenom v Javě. Naopak, serverová část aplikace by mohla být vyvinutá pomocí jakéhokoli programovacího na to určeného: Python, PHP, Ruby, Node.js atd. Nicméně, Java byla vybrána kvůli poskytované stabilitě, multiplatformovosti, rozsáhlému množství knihoven, aktivní komunitě, osobní preferenci a výborné podpoře multithreadingu.
- Vaadin 7.3 jako základní framework pro propojení serverové a klientské části aplikace. Odpověď na otázku “proč byl použit právě tento framework” zní podobně jako v předchozím odstavci: pro vývoj SM by se mohl použít jakýkoliv jiný framework. Vaadin poskytuje sadu standardních uživatelských prvků, usnadňuje jejich propojení z databázi, poskytuje řešení komunikaci mezi klientskou a serverovou částí aplikace. Tento framework je zajímavý také tím, že poskytuje možnost psát kód v Javě, který je pak automaticky přeložen do JavaScriptu.
- JavaScript a HTML5 Canvas. JavaScript je v podstatě standardem pro vývoj RIA (Rich Internet Applications), proto nebyla na výběr žádná jiná alternativa [42]. Pro vizualizaci primárního letového displeje bylo rozhodnuto používat HTML5 Canvas (namísto SVG) kvůli tomu, že PFD obsahuje poměrně velké množství prvků a má komplikovanou logiku, která by se pomocí SVG programovala obtížně.
- HTTP jako komunikační protokol mezi serverovou a klientskou částí. Další možnosti by bylo používat WebSocket protokol, fungující přes TCP, který poskytuje lepší výkon, i když je podporován jen v nejmodernějších prohlížečích [50]. Rychlost předávání zpráv mezi serverem a klientem v SM je kritická, proto je používání WebSocketu místo http, jedním z možných způsobů vylepšení SM.

### 6.1.4 Popis databázového modelu

SM používá databázový systém PostgreSQL 9.3 pro ukládání modelů letadel, modelů motorů, konfiguraci simulátorů, informací o proběhlých simulacích. Obr. 24 znázorňuje databázový model SM.



**Obr. 24. Databázový model systému pro správu simulátorů**

Příloha C obsahuje materiály, na které se v této kapitole odkazuje. Obr. 47 znázorňuje databázový model SM.

Tabulka „simulatoremodel” (viz Obr. 48, příloha C) obsahuje informace o modelech simulátorů, o jejich konfiguraci a charakteristikách.

Tabulka „enginemodel“ (viz Obr. 49, příloha C) obsahuje informace o modelech motorů pro některý model simulátoru, o jejich charakteristikách, o minimálních a maximálních povolených hodnotách pro jednotlivé charakteristiky. Jeden záznam v této tabulce obsahuje informace o jednom modelu motoru.

Tabulka „simulator” (viz Obr. 50, příloha C) obsahuje informace o konfiguraci simulátorů, přidávaných uživatelem do SM: kterému modelu simulátoru odpovídá tento simulátor, na kterém stroji a portu běží tento simulátor, název simulátoru.

Tabulka „simulation” (viz Obr. 51, příloha C) obsahuje informace o právě běžících nebo již minulých simulacích.

Tabulky „simulationinfo“, „simulationdevicesstate“, „simulationpfdinfo“ (Obr. 53, Obr. 54, Obr. 55 v příloze C) obsahují informace o souřadnicích, stavu hlavního letového displeje a jiných zařízeních během jednotlivých simulací.

Tabulka „simulationenginesstate“ (viz Obr. 51) obsahuje informace o stavech motorů během jednotlivých simulací. Hodnoty se zapisují do pole, přičemž počet hodnot v tom poli odpovídá počtu motorů simulátoru.

Obrázky znázorňující databázový model byly vytvořené pomocí nástroje DbVisualizer [39].

### 6.1.5 Podporované prohlížeče

Vytvořená webová aplikace pro správu simulátoru se dá použít v následujících prohlížečích:

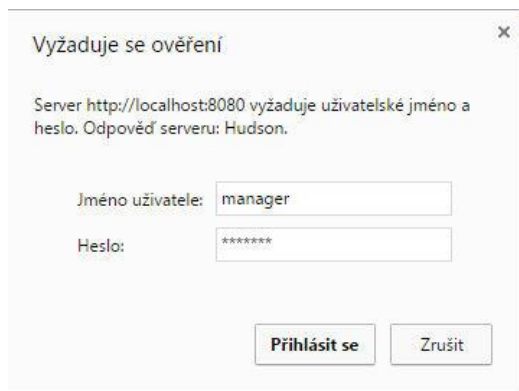
- Chrome 42 nebo novější (pro Windows)
- Chromium 45 nebo novější (pro Linux)

Aplikace byla testována také v prohlížeči Firefox a na mobilních zařízeních Android, kde je aplikace také funkční, ale panel řízení letu je výrazně zpomalený v porovnání s prohlížeči Chrome a Chromium. Přidání lepší podpory pro mobilní zařízení je jedním z možných směrů vylepšení aplikace.

## 6.2 Uživatelské rozhraní systému pro správu simulátorů

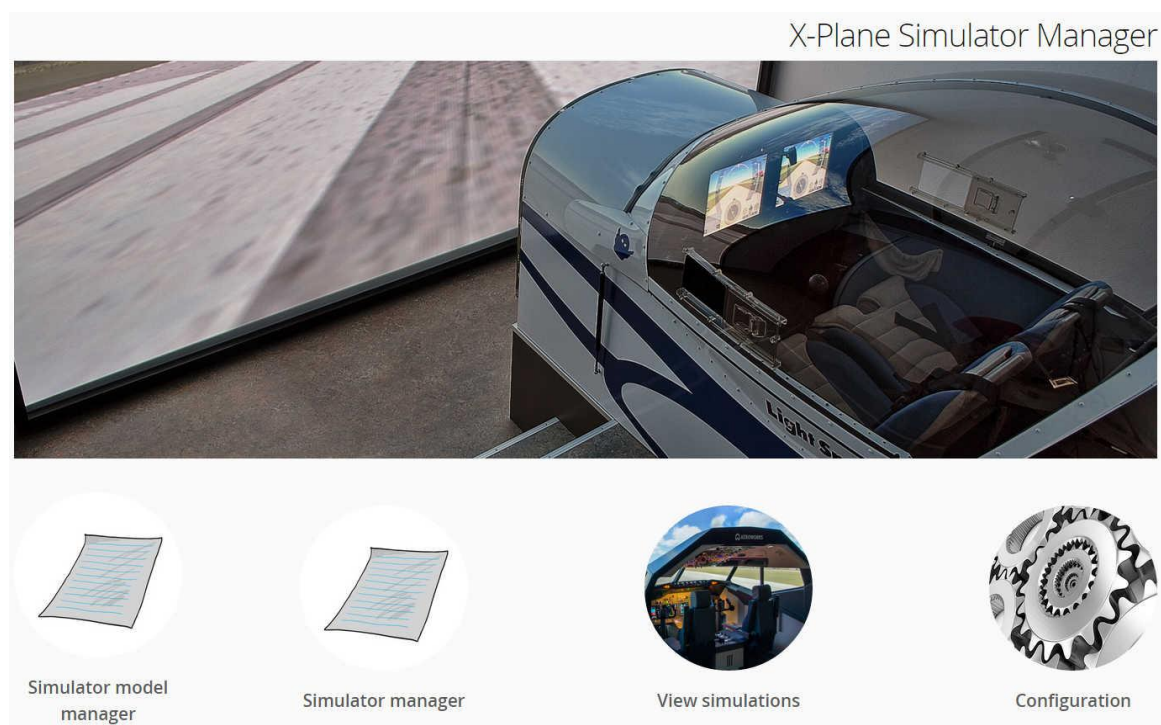
V této kapitole je popsáno jak se systém používá. Tento popis je zde uveden pro lepší pochopení vytvořeného systému, nikoliv pro vyčerpávající seznam jeho funkcí.

Pro začátek použití aplikace uživatel přejde na stránku <http://localhost:8080/SimulatorManager/> (místo localhost a 8080 je potřeba uvést server a port na kterém aplikace skutečně běží, detaily viz v Příloze B). Na vyzvání je potřeba vyplnit uživatelské jméno a heslo, formulář je zobrazený na Obr. 25 (detaily viz v příloze B).



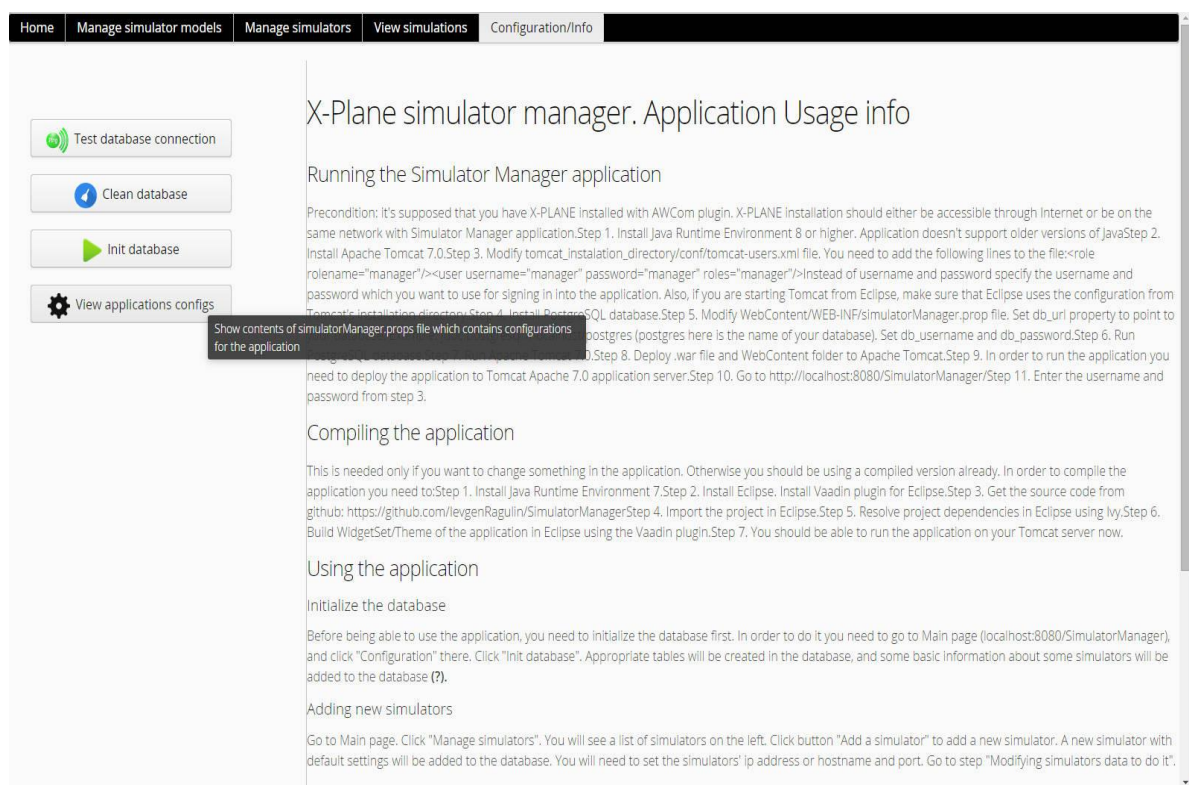
Obr. 25. Přihlašování do systému pro správu simulátorů

Následně se uživateli zobrazí úvodní stránka, na které si bude moci zvolit jednu ze čtyř možností: spravování modelu simulátoru, spravování simulátorů, zobrazení běžících/přehrávání již proběhlých simulací, konfigurace SM. Úvodní stránka je zobrazena na Obr. 26.



**Obr. 26. Úvodní stránka systému pro správu simulátorů**

Po zvolení položky menu “Configuration” (viz Obr. 27), se zobrazí stránka, na které lze otestovat připojení na databáze, vyčistit všechna data z databáze, inicializovat databázi, zobrazit aktuální konfigurační soubor. Po pravé straně je zobrazeny návod k použití aplikace.



**Obr. 27. Stránka konfigurování aplikaci**

Po zvolení položky menu “Simulator model manager” se uživateli zobrazí stránka pro přidávání a editaci modelů simulátoru, viz Obr. 28. Na této stránce uživatel nastaví modely simulátorů a modely jejich motorů: maximální a minimální množství paliva v nádržích, zapnutí nebo vypnutí jednotlivých hodnot pro motory (různé motory posílají různá data) atp. Nastavení na této stránce ovlivňuje zobrazení jednotlivých indikátorů na palubní desce na stránce “View simulations”. Pokud uživatel přidá na této stránce 4 motory, bude pak stránka “View simulations” zobrazovat informace o 4 motorech. Hodnoty „min, low, high, max” určují jakou barvou jsou označené úseky na indikátorech, zobrazujících tyto hodnoty (pokud máme nedostatek paliva v nádrži, zobrazíme to červenou barvou).

Obr. 28. Spravování modelů simulátorů v SM

Při pokusu uložení nevalidní konfigurace (například, když chybí nastavení maximálního množství paliva v nádržích), systém zobrazí chybovou hlášku s popisem vzniklého problému, viz Obr. 29. SM ověřuje zda model simulátoru obsahuje nejméně jeden motor, zda pro všechny zapnuté vlastnosti modelu motoru jsou nastavené minimální, nízká, vysoká a maximální hodnota.



Obr. 29. Validace na stránce spravování modelů simulátorů v SM



Důležitým nastavením motoru na této stránce je “Engine id (AWCom)”, které určuje kterému motoru posílanému přes AWCom odpovídá nakonfigurovaný motor v SM. Například, pokud “Engine id (AWCom)” = 1, to znamená, že tento model simulátoru odpovídá 1. motoru, v simulátoru X-Plane.

Vytvořené modely simulátorů jsou pak dostupné k výběru na stránce “Manage simulators”.

Pro pochopení aplikace je důležité, že modely motorů, nastavené na této stránce, nemusí obsahovat všechny hodnoty, které posílá protokol AWCom. Je zapotřebí natavit jen ty hodnoty, které pak budou uživatele zajímat na panelu řízení letu. Doporučeným postupem je zapínat nejmenší možný počet hodnot pro každý motor, protože velký počet ukazatelů zatěžuje a zpomaluje vizualizace na panelu řízení letu. Vylepšení výkonu těchto ukazatelů je jedním z možných směrů vývoje aplikace.

Dále má uživatel možnost přejít na stránku „Manage simulators“ (viz Obr. 30), kde se přidávají a editují jednotlivé simulátory. Tady je potřeba nastavit IP adresu/název hostitele serveru, na kterém běží server AWCom<sup>1</sup> pro tento simulátor. Mezi další možnosti patří nastavení kterému modelu simulátoru odpovídá tento simulátor.

**Obr. 30. Stránka pro spravování simulátorů v SM**

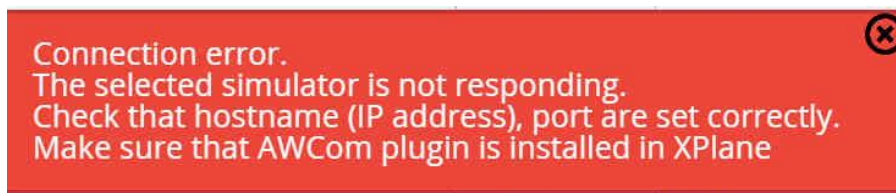
Dále můžeme nastavit, zda je simulátor aktivní. Pokud simulátor je označený jako aktivní, SM se snaží průběžně (ve výchozím nastavení co 1000 ms, což můžeme změnit přes konfigurační soubor) připojovat k simulátoru, získávat data o simulacích, ukládat je do databáze. Pokud simulátor není označený jako aktivní, SM se nepřipojuje k simulátoru a neukládá data o simulaci do databáze.

<sup>1</sup> AWCom server většinou běží na stejném stroji jako simulátor. Pomocí AWCom serveru SM získává data o běžících simulacích. Detaily viz v kapitole o komunikačním systému AWCom



Při pokusu uložit data na stránce spravování simulátorů, SM ověří, zda se může připojit na uvedenou adresu. Pokud se SM nepodaří získat data z uvedené adresy, uživateli se zobrazí chybová hláška (viz Obr. 31).

Další věc, kterou SM ověřuje při pokusu uložit data na stránce spravování simulátorů, je zda data posílána simulátorem odpovídají zvolenému modelu simulátoru. Pokud získaná data neodpovídají konfiguraci tohoto modelu (například model simulátoru v SM má nakonfigurované 3 palivové nádrže, ale získána ze simulátoru data svědčí o tom, že simulátor má jen 2 palivové nádrže), uživateli se zobrazí chybová hláška. Na Obr. 32 je znázorněná chybová hláška, kterou zobrazuje systém tehdy, když nakonfigurovany model simulátoru v SM neodpovídá datům, která posílá X-plane simulátor. Chybová hláška obsahuje informace o chybě a o tom jak problém opravit.



**Obr. 31. Validace na stránce pro spravování simulátorů v SM – chyba komunikace**



**Obr. 32. Validace na stránce pro spravování simulátoru v SM - chyba konfigurace**

Po dokončení nastavení simulátoru, má uživatel možnost přejít na stránku “View simulations” (viz Obr. 33). V tabulce nahoře je seznám simulátorů, v tabulce dole je seznam běžících nebo minulých simulací na zvoleném simulátoru. Po zvolení simulace uživatel může stisknout tlačítko “View chosen simulation” a přejít na panel řízení letu. Dále se zde nachází funkcionalita pro spravování simulace. Uživatel může:

- odmazávat simulace z databáze
- ukončovat právě běžící simulaci, což znamená, že data ze zvoleného simulátoru se již nebudou ukládat v databázi do této simulace, ale vytvoří se nová simulace, do které se nová data bude přidávat.

HomeManage simulator modelsManage simulatorsView simulationsConfiguration/Info

Select a simulator

Is simulator active	Simulator id	Simulator name
true	1	New67921000

Simulation sessions on the selected simulator

Simulation id	Simulator id	Is on	Is paused	Started time	Latest update
21	1	false	false	27.4.2015 8:40:00	27.4.2015 10:04:41
20	1	false	false	27.4.2015 8:32:36	27.4.2015 8:40:00
19	1	false	false	27.4.2015 8:23:55	27.4.2015 8:32:36
18	1	false	false	27.4.2015 8:15:33	27.4.2015 8:23:55
17	1	false	false	27.4.2015 8:13:42	27.4.2015 8:15:33
16	1	false	false	26.4.2015 15:46:49	27.4.2015 0:19:44
15	1	false	false	26.4.2015 15:43:20	26.4.2015 15:46:49

Delete chosen simulation

View chosen simulation

**Obr. 33. Stránka pro výběr simulace: běžící nebo minulé**

Po stisknutí tlačítka “View chosen simulation” se zobrazí panel řízení letu (viz Obr. 34). Na tomto panelu je zobrazeny PFD, výškový a rychlostní profil letu, stav řídicí páky, výchylek, klappek, brzd, podvozků, palivových nádrží, motorů. Nachází se zde také mapa, na které je zobrazená dráha letu a současná poloha simulovaného letadla. Panel umožňuje zobrazovat probíhající simulace, a také přehrávat minulé simulace.

Pokud se v tuto chvíli zobrazuje právě probíhající simulace, data na tomto panelu se obnovují co  $T$  ms (standardní hodnota  $T=300$  ms, je to nastavitelný přes konfigurační soubor). Pokud uživatel obnoví stránku, například zmačknutím klávese F5, program načte data o této simulaci z databáze a na základě těchto dat nakreslí dráhu letu na mapě, nakreslí grafy výšky a rychlosti.

Pokud panel zobrazuje záznam ukončené simulace, klepnutím na časový bod na grafu výšky nebo rychlosti se všechny prvky na tomto panelu nastaví na hodnoty toho časového okamžiku, což poskytuje uživateli možnost zpětně si prohlížet a analyzovat ukončené simulace.

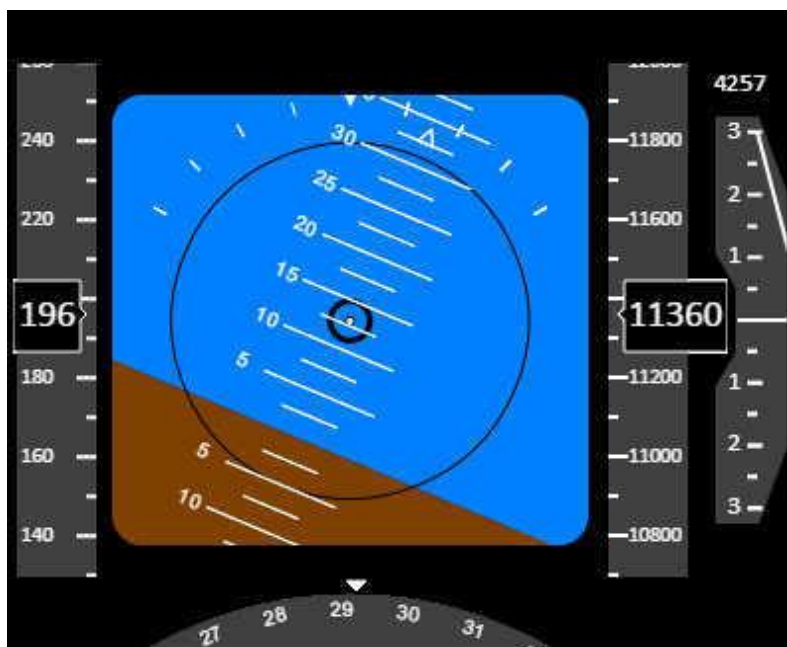


Obr. 34. Panel řízení letu

## 6.2.1 Panel řízení letu

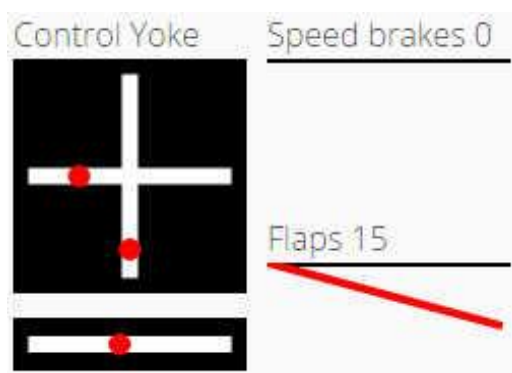
V předchozí kapitole byl krátce popsány panel řízení letu (viz Obr. 34). V této kapitole každý prvek tohoto panelu je popsán detailněji.

V levém horním rohu se nachází hlavní letový displej (PFD), viz obrázek Obr. 35. Tento displej je zjednodušenou podobou PFD použitého v Boeing 737. Význam hodnot na tomto displeji je stejný, jako v Boeing 737, viz kapitola 4.1.1.



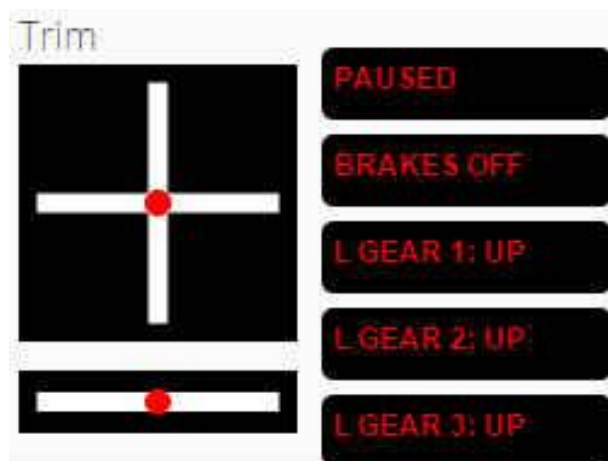
**Obr. 35. Panel řízení letu – PFD**

Po pravé straně od PFD je zobrazená informace o stavu řídicí páky, směrovky, klapek a brzdících klapek (viz Obr. 36). Klapky se zobrazují červenou barvou pokud jsou zapnuté a pokud je rychlost letadla vyšší, než maximální povolená rychlost na klapkách pro tento model letadla (tato rychlost je nastavitelná na stránce pro spravování modelů simulátorů).



**Obr. 36. Panel řízení letu: směrovka, řídicí páka, klapky a vzdušná brzda**

Kromě toho, po pravé straně od PFD je informace o stavu výchylek směrového kormidla, výškovky, levého křídýlka, brzd, podvozků (viz Obr. 37). Také je zde zobrazeno, zda simulace v tuto chvíli běží nebo je pozastavená.



**Obr. 37. Panel řízení letu: odchylky, pauza, brzdy, podvozky**

Po pravé straně od panelů letových instrumentů je umístěná mapa, zobrazující současnou polohu, směr a dráhu létu letadla (viz Obr. 38). Pokud panel řízení letu zobrazuje již minulou simulaci, vybrání časového bodu na výškovém nebo rychlostním grafu přesune obrázek letadla na mapě na tu pozici, ve které se letadlo nacházelo v konkrétní okamžik času. Mapa byla vytvořena pomocí Google Maps API [33].



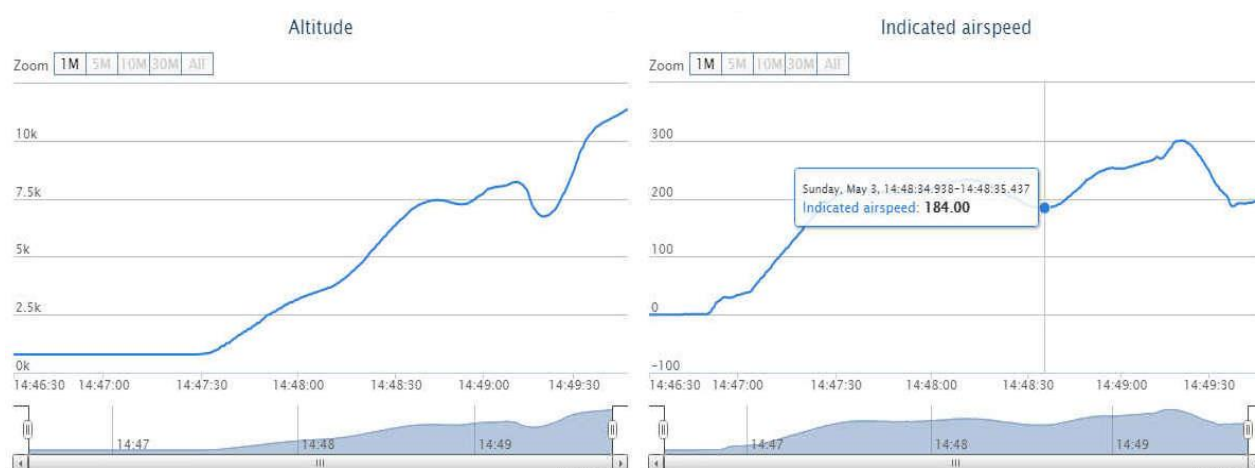
**Obr. 38. Panel řízení létu: mapa [33]**

Dole pod hlavním letovým displejem jsou umístěné grafy (viz Obr. 39), zobrazující výškový a rychlostní profil letu. Taky můžeme nastavit, aby na grafu byl zobrazen jen určitý časový úsek. Například, pokud by uživatel chtěl vědět, na jaké výšce se simulované letadlo nacházelo během posledních pěti minut, může to nastavit kliknutím na „5M“ v levém horním rohu grafu nebo pomocí



přetahování myši začátku a konce časové osy do správné polohy. Při klepnutí na bod na křivce na grafu se zobrazí okénko, obsahující výšku nebo rychlost simulovaného letounu v odpovídající časový okamžik.

Pokud panel řízení letu zobrazuje již minulou simulaci, zmačknutí časového bodu na výškovém nebo rychlostním grafu nastaví všechny letové instrumenty na hodnoty konkrétního časového okamžiku. Výškový a rychlostní profily letu byly vytvořené pomocí knihovny Highcharts [35].



**Obr. 39. Panel řízení letu: výškový a rychlostní profil letu [35]**

Dole pod grafy jsou umístěny ukazatele stavu paliva (viz Obr. 40), zobrazující množství paliva v levé, centrální a pravé nádrži<sup>2</sup>. Na stránce spravování modelů simulátorů je možno nastavit kolik palivových nádrží letadlo má. Povinně je také potřeba nastavit minimální, nízký, vysoký a maximální objem paliva v nádržích, což pak ovlivní počet a vzhled ukazatelů, zobrazených na panelu řízení letu. Tabulka 1 popisuje závislost mezi nastavením minimální, nízké, vysoké a maximální hodnoty a barvou odpovídajícího úseku.

Rozmezí hodnot	Barva úseku na ukazateli stavu paliva
minimální - nízká	červená
nízká - vysoká	zelená
vysoká – maximální	žádná

**Tabulka 1. Závislost mezi nastavením minimální, nízké, vysoké a maximální hodnoty a barvou odpovídajícího úseku ukazatelů stavu paliva**

<sup>2</sup> Hodnoty -1, -2 jsou chybové stavy, které znamenají, že letadlo nebo komunikační protokol nepodporuje tyto hodnoty. Detailněji viz v kapitole o komunikačním systému AWCom

• Fuel tanks



Obr. 40. Panel řízení letu: palivové nádrže [34]

Dole pod budíky s informací o palivových nádržích je zobrazen stav motorů letadel. Panel může obsahovat 1 až 8 motorů. Seznam všech parametrů a jednotky jsou uvedené v Příloze A. Na stránce spravování modelů simulátorů je možný nastavit, které parametry se zobrazí na této stránce. Pro zvolené parametry je potřeba nastavit minimální, nízkou, vysokou a maximální hodnotu, což pak ovlivní vzhled budíků. Tabulka 2 popisuje závislost mezi nastavením minimální, nízké, vysoké a maximální hodnoty a barvou odpovídajícího úseku.

Rozmezí hodnot	Barva úseku na ukazateli stavu motoru
minimální - nízká	žadná
nízká - vysoká	zelená
vysoká – maximální	červená

Tabulka 2. Závislost mezi nastavením minimální, nízké, vysoké a maximální hodnoty a barvou odpovídajícího úseku ukazatelů stavu motorů

• Engine 0



Obr. 41. Panel řízení letu: motory [34]

## 6.3 Popis činnosti aplikace

Po spuštění simulace na některém simulátoru systém pro správu simulátorů (dále SM) přidává do databáze záznam o začátku simulace do tabulky *simulationinfo*. Po získávání dat, SM rozparsuje data a uloží je do databáze.

Systém pro správu simulátorů co  $T_1$  ms ( $T_1$  je nastavitelné přes konfigurační soubor), získává data ze simulátorů a uschovává data o současném stavu simulátorů v operační paměti. Když uživatel otevírá v prohlížeči panel řízení letu, SM tak nevolá odpovídající simulátor, ale používá data, již uložené v operační paměti.

Taky, SM co  $T_2$  ms ( $T_2$  je nastavitelné), ukládá data ze simulátorů do databáze.

## 6.4 Protokol AWCom

AWCom je protokol vyvinutý v rámci skupiny Aeroworks, Mgr. Karolem Rydlem [41]. Protokol AWCom se používá jako rozhraní, pomocí kterého se dá přistupovat k simulátorům. Systém pro správu simulátoru získává data ze simulátoru pomocí tohoto protokolu. SM komunikuje z AWCom serverem přes TCP/IP sockety.

Pro získávání simulačních dat SM pošle příkaz “GET:DATA:END” AWCom serveru. AWCom server pak pošle odpověď ve formátu **klíč:hodnota**.

Pro získávání informací o stavu motorů SM pošle příkaz “GET:ENGINE:END” AWCom serveru, který pak pošle odpověď ve formátu **Eiklíč:hodnota**, kde *i* je pořadí motoru. Nejmenší možná hodnota *i* je 0, největší možná hodnota je 7, protože podporovaný počet motorů je 8.

Seznam klíčů, jejich význam a jednotky posílaných hodnot jsou uvedené v Příloze A.

## 6.5 Vylepšení výkonu aplikace

V této kapitole je popsáno jakými způsoby se vylepšoval výkon a plynulost animace systému pro správu simulátorů.

- Paralelní zpracování První implementace aplikace získávala a ukládala data ze všech simulátorů ve stejném vlákne. Ukázalo se, že v případech kdy byl některý ze simulátorů vypnutý, aplikace se výrazně zpomalila. Proto bylo rozhodnuto, že SM bude mít na obsluhování komunikace s každým simulátorem jedno vlákno. Po startu SM se zjistí podle databáze počet nakonfigurovaných simulátorů a vytvoří se odpovídající počet vláken. Pokud se některý simulátor smaže nebo přidá, změníme taky počet vláken. Ukázalo se, že po přidání vícera vláken výkon SM, již nebyl ovlivněn počtem simulátorů.



- Vylepšení komunikace mezi klientem a serverem. Framework Vaadin [30], který byl použit při vývoji aplikace, usnadňuje komunikace mezi klientskou (JavaScript) a serverovou (Java) částí aplikace. Usnadňuje ji natolik, že komunikaci programátor při použití `AbstractJavaScriptComponent` [42] vůbec nemusí řešit: stačí nadefinovat Java třídu, ve které se ukládá sdílený stav serverové a klientské části aplikace, Vaadin se pak postará o komunikaci: když se nějaká hodnota v objektu této třídy změní, Vaadin ten nový stav pošle buď klientovi, nebo serveru. Takový přístup občas není vhodný, hlavně v případě panelu řízení letu, kde probíhá vizualizace letu a data o stavu simulátoru se posílají poměrně často (výchozí nastavení - co 1 vteřinu). V následujících případech byla komunikace mezi klientskou a serverovou částí byla upravená:
- Data se posílají, jen pokud se něco změnilo. V původní implementaci SM server před nastavením dat do sdíleného stavu ověřoval, zda se nová data, obdržená ze simulátoru se liší od dat, které JavaScript klient již má. Pak, pokud se data lišila, SM server obnovoval sdílený stav. Problém toho přístupu byl v tom, že Vaadin posílá data nejen když server změní sdílený stav, ale i když serverová část aplikace čte data z třídy stavu (to jsem zjistil pomocí debugovacího nástroje Frameworku Vaadin, který zobrazuje jednotlivé zprávy obdržené klientem). Kvůli tomu se zbytečně posílalo velké množství dat, což výrazně ovlivňovalo výkonnost aplikace. Řešením bylo pro každou JavaScript komponentu naimplementovat jednoduchou třídu, která duplikuje sdílený stav. Ta třída se pak používá serverem pro ověření, jestli klientská část JavaScript komponenty má stejná data jako ta, co byla obdržená od simulátoru, nebo jestli má zastaralá data. Pokud zjistíme, že klientská část má neaktuální data, obnoví se sdílený stav v serverové části SM a Vaadin pak tyto data odešle.
- Kratší názvy proměnných. Bylo zjištěno, že Vaadin utrácel hodně času (přibližně 500 ms v původní implementaci) na parsování dat, posílaných serverem o stavu simulací, a že při zmenšení objemu posílaných dat bylo možné zmenšit čas potřebný pro zpracovávání dat několikanásobně pomocí použití kratších názvů proměnných ve sdíleném stavu.
- Vymazat data ze sdíleného stavu po přečtení. Další změnu, která byla nutná kvůli specifičnosti používaného frameworku Vaadin, bylo potřeba provést v komponentě pro zobrazování výškového a rychlostního profilů letu. Po prvním volání stránky na zobrazení běžící simulace SM získává data z databáze ohledně dráhy letu letounu, výšce a rychlosti letounu během letu a na základě těchto dat vytvoří rychlostní, výškový profily letu, nakreslí dráhu letu letadla na mapě. Data se nastavují do sdíleného stavu, který se automaticky přeposílá frameworkem Vaadin klientské části aplikace. Vzhledem k tomu, že data se ukládají do databáze skoro každou vteřinu a simulace občas běží hodiny, množství těch dat může být poměrně veliké. V tom případě není vhodné, aby se tato data přeposílali přes síť tam a zpátky. Řešením bylo smazat na klientské stránce data o historii letu, výšky a rychlosti

ze sdíleného stavu hned po jejich přečtení. Tím pádem se celá výšková a rychlostní historie letu posílají přes síť jen jednou.

- Nastavení použitého frameworku Vaadin do „production mode“ [45]. Ve výchozím nastavení Vaadin používá „debug mode“, který loguje hodně informací do konzole prohlížeče, což zpomaluje běh aplikace.

## 6.6 Vylepšení animace

V první implementaci SM panel řízení letu měl frekvence přibližně 6 FPS, proto animace nebyla plynulá, navíc celá aplikace zatěžovala procesor a grafickou kartu uživatele. Proto byla provedená následující vylepšení:

- Místo použití jednoho HTML5 Canvasu, používáme 8 Canvasů, a když se nějaká hodnota změní, nepřekresluje celý panel řízení letu, ale jen jeho jednu část.
- Místo překreslování primárního letového displeje v každém snímku, prvně zjistíme, jestli se hodnoty úhlu klopení, klonění, výšky a rychlosti změnili, a SM překresluje odpovídající Canvas jen v tom případě, když se hodnota změnila na hodnotu  $> \alpha$ .  $\alpha$  je definovány zvlášť pro každý Canvas, protože různé prvky panelu řízení letu vyžadují různé míry přesnosti.
- V původní implementaci se animace implementovala na nízké úrovni, kde se v SM počítalo, na kolik je potřeba v příštím kroku animace zvětšit animovanou hodnotu apod. Ukázalo se, že lepším řešením je používat pro animování knihovnu (v tomto případě byla použita knihovna GreenSock [38]), která vyřeší některé věci za programátora. Knihovna se mimo jiné snaží o nejlepší frekvence obnovování obrazovky, ale pokud vysoká frekvence není možná, animace se nezpomalí ve snaze zobrazit všechny snímky s velkým zpožděním, ale místo toho bude ukazovat nejaktuálnější snímky. Tato knihovna je užitečná pro aproximace animovaných hodnot, vypočítání kroku animace (tzn. hodnoty, na kterou bude nastavený animovaný prvek v dalším snímku).
- V původní implementaci byla snaha posílat data a obnovovat panel řízení letu co nejčastěji (co 300 ms), a ta data pak mezi sebou aproximovat. Ukázalo se ale, že když se data posílají příliš často, zatěžuje procesor uživatele a kvůli tomu animace není tak plynulá. Problém byl vyřešený tím, že data se posílají méně často (asi co 1s), a pak se aproximují pro vytvoření plynulejší animace. V případě, když rychlejší odezva je pro uživatele důležitější než plynulá animace, dá se frekvence obnovování panelu řízení letu nastavit přes konfigurační soubor na menší hodnotu.
- V původní implementaci server posílal data o poloze letadla JavaScript klientovi se stejnou frekvencí, jako pro ostatní prvky panelu řízení letu. Ukázalo se, že v případě, když Google mapa obsahuje velký počet souřadnic letadla, to hodně zatěžuje tu komponentu, a celý panel

řízení letu je pak zpomalený. Řešením bylo přidávat nové souřadnice na mapu méně často (jednou za 6 vteřin je výchozím nastavení, což se dá změnit přes konfigurační soubor).

Výsledkem vylepšení, popsanych v této kapitole, bylo zvýšení FPS panelu řízení letu do přibližně 30 snímků za minutu, plynulejší animace, menší zatíženost procesoru a grafické karty uživatele.

## 7 Provedená testování

V této kapitole je popsán jak se systém pro správu simulátorů testoval.

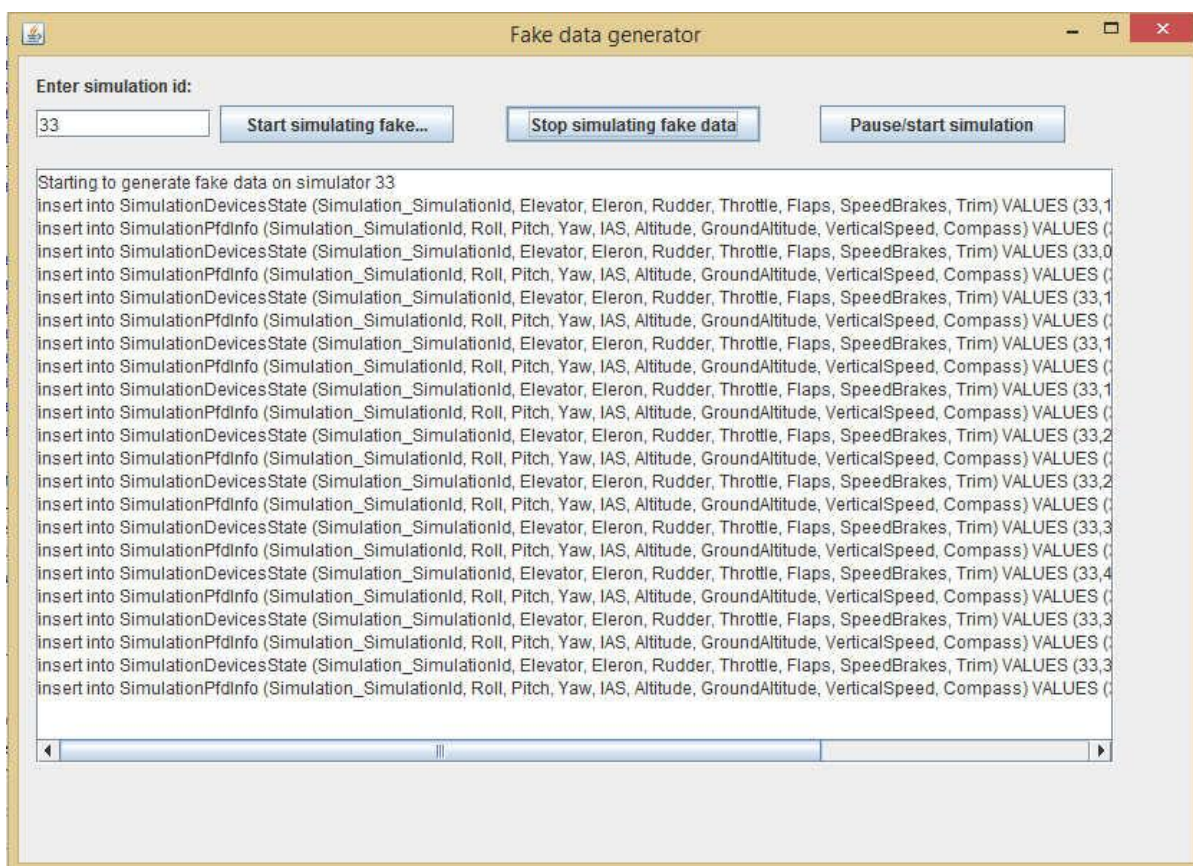
### 7.1.1 Testování pomocí generátoru falešných simulačních dat

Na začátku vývoje SM aplikace nepodporovala získávání dat ze simulátorů, proto bylo zapotřebí vyvinout nějaký jednoduchý generátor simulačních dat. Taková aplikace byla naimplementovaná v programovacím jazyce Java pomocí knihoven AWT [43] a Swing [44]. Po spuštění aplikace jednou za vteřinu vygeneruje náhodná simulační data a uloží je do databáze. SM pak ty data použije na panelu řízení letu pro vizualizace simulaci. Na Obr. 42 je ukázka této aplikace. Na Obr. 43 je schéma propojení systému SM a generátoru dat.

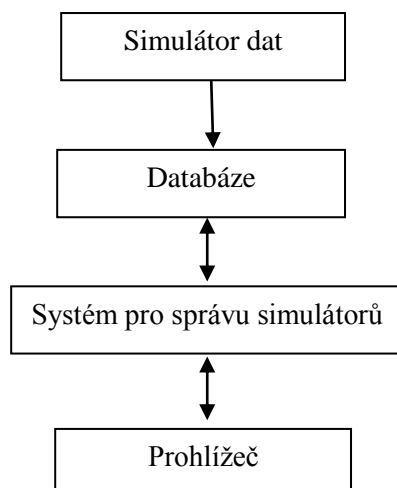
Tento generátor falešných dat už se nepoužívá, protože SM získává data z X-Plane simulátorů napřímo.

Odkaz na zdrojový kód generátoru simulačních dat:

<https://github.com/IevgenRagulin/FakeDataGeneratorForSM>.



Obr. 42. Uživatelské rozhraní generátoru falešných simulačních dat



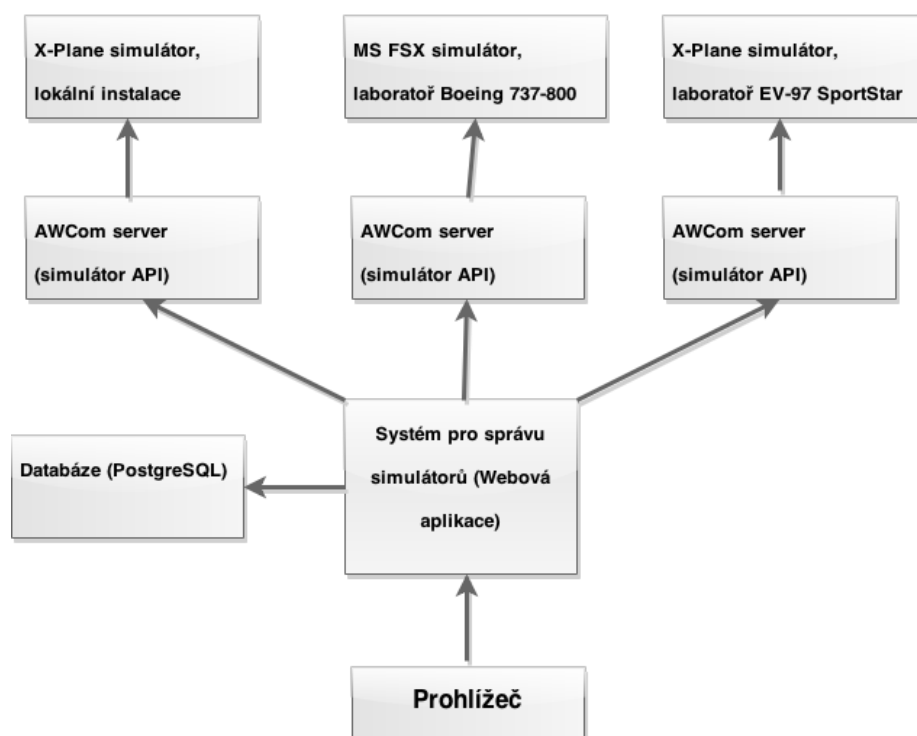
Obr. 43. Schéma propojení generátoru dat a SM

### 7.1.2 Testování pomocí lokální instalace X-Plane a v laboratoři SimStar

Během vývoje systém pro správu simulátorů se testoval pomocí lokální instalace X-Plane (byla použita verze 9.01rc), která běží na stejném stroji jako SM, a prohlížeče Google Chrome. Dále byl SM testován v laboratoři SimStar na simulátorech letounů Evektor SportStar a Boeing 737 NG. Na Obr. 44 je uvedená schéma propojení SM a simulátorů.

SM byl otestován dobrovolníky, kteří poznamenali, že panel řízení letu je vhodný nejen pro vyhodnocování a analýzu letu, ale vzhledem k dostačujícímu množství informací, může být použit pro vlastní řízení letadla. Navíc, v některých případech panel řízení letu SM poskytuje detailnější a srozumitelnější informace o letové situaci, než palubní deska v simulátoru X-Plane. Díky provedeným testováním se podařilo odhalit prvky uživatelského rozhraní, které byly matoucí pro uživatele. Také byly nalezené některé další defekty, které byly následně upraveny. Některá vylepšení provedená po testování:

- Zvětšení snímkové frekvence panelu řízení letu z 6 do 30 snímku za vteřinu.
- Přidání podpory češtiny při konfiguraci letadla.
- Přidání dialogových oken, upřesňujících, zda uživatel skutečně chce provést některé kritické činnosti, jako například mazání modelů simulátorů z databáze.



Obr. 44. Schéma propojení SM a simulátorů

## 8 Porovnání s existujícími systémy

V této kapitole je provedená analýza řešení, které poskytují obdobnou funkcionalitu jako SM: Jediným nalezeným konkurentem je FSPS – Flight Simulator Manager [48].

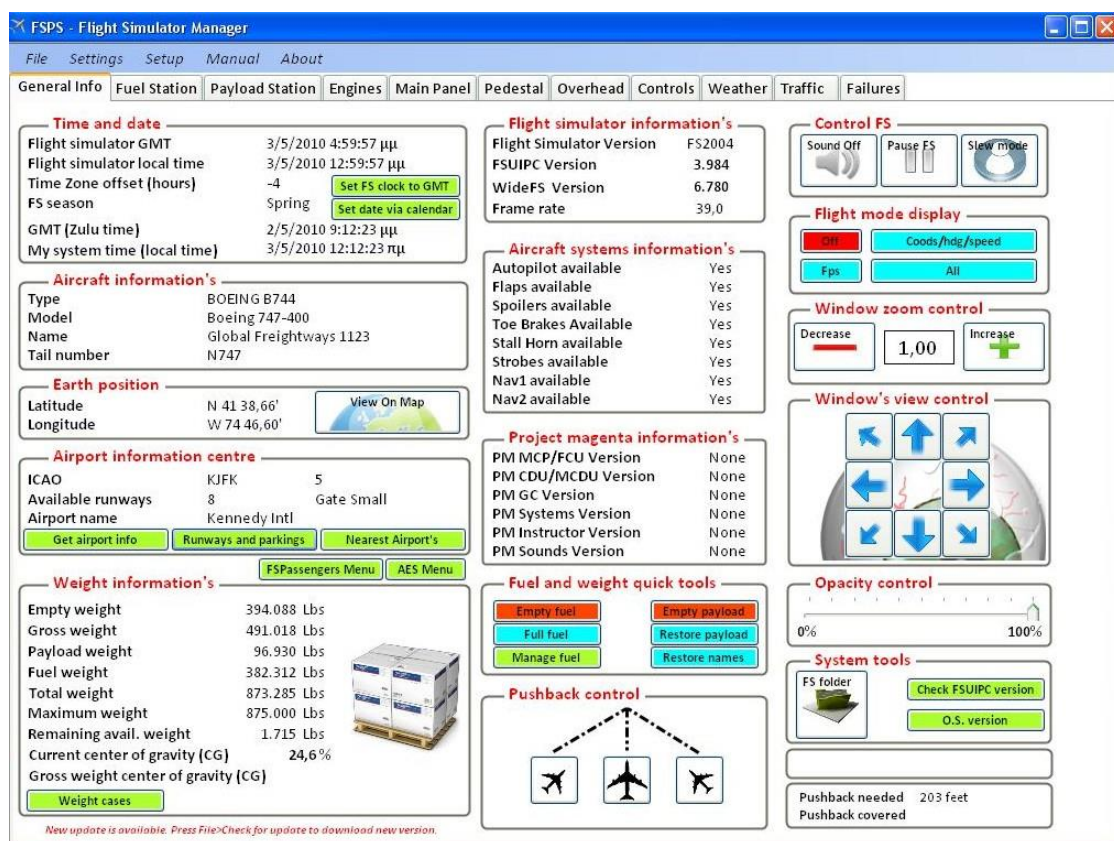
### 8.1.1 FSPS Flight Simulator Manager

FSPS Flight Simulator Manager [48] je systém, který se používá se simulátory Microsoft Flight Simulator. Je to desktopová aplikace, pomocí které se uživatel může připojit na běžící simulátor. V této aplikaci se zobrazují informace o stavu motorů, palivových nádrží, stavu ovládacích prvků, poloze, simulovaném letounu, současném letišti, provozu, nastaveních autopilotu, letišti atd. Jinými slovy, tato aplikace poskytuje víc informací, než samotný panel řízení letu v simulátoru. Informace se v FSPS většinou zobrazuje v textové podobě. Kromě toho FSPS obsahuje funkci odesílání příkazů na simulátor.

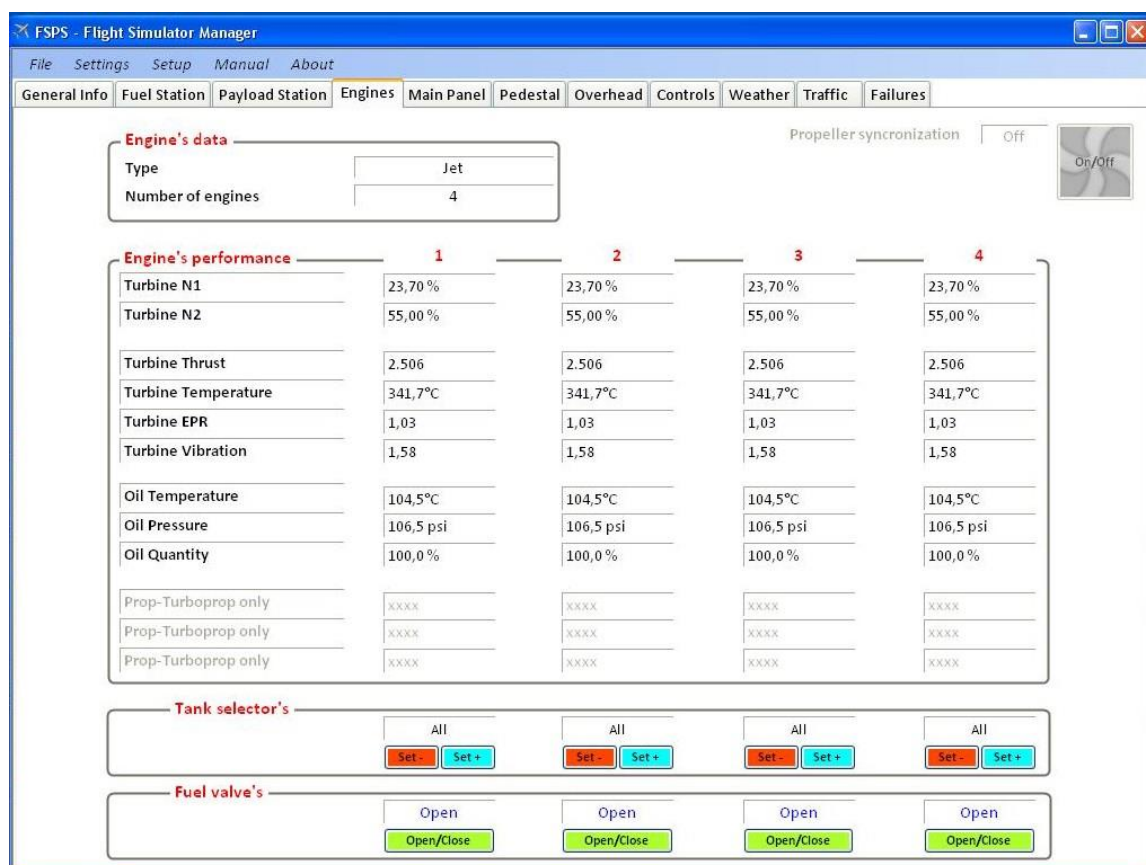
Flight Simulator Manager by mohl být přeložený do češtiny jako “správce simulátoru” (důležité tady je jednotné číslo na konci). Na rozdíl od vytvořeného systému v této diplomové práci, FSPS je pluginem, který rozšiřuje základní možnosti simulátoru a slouží jako pomůcka při letu pro pilota. SM, vytvořený v této práci slouží pro spravování simulátorů a simulačních dat.

Na Obr. 45 je zobrazený hlavní panel FSPS. Ten panel obsahuje informace o simulovaném letounu a jeho konfiguraci, vaze, letišti.

Dalším rozdílem FSPS a SM je to, že data, poskytnuté FSPS jsou vyčerpávající, ale jsou zobrazená v textové podobě. Namísto toho SM data vizualizuje, pomocí 2D animace. Na Obr. 46 je vidět panel motorů v FSPS. Pro porovnání, panel motorů v SM je na Obr. 41.



Obr. 45. Hlavní panel v FSPS Flight Simulator Manager [47]



Obr. 46. Panel motorů v FSPS Flight Simulator Manager [47]



Tabulka 3 obsahuje shrnutí porovnání Systému pro správu simulátoru, vytvořeného v této práci a FSPS.

	<b>SM (Systém pro správu simulátorů)</b>	<b>FSPS Flight Simulator Manager [48]</b>
<b>Podporované simulátory</b>	X-Plane, FXS. Podle podpory AWCCom	Microfost Flight Simulator
<b>Architektura</b>	Klient-serverová aplikace, klient používá prohlížeč	Desktopová aplikace
<b>Podporované operační systémy</b>	Windows, Linux, Android (částečně)	Windows
<b>Účel</b>	Spravování simulátorů, simulačních dat, poskytování možnosti ukládat a zpětně analyzovat minulé simulace	Poskytování dodatečných informací o probíhající simulaci, rozšíření funkcionality letového simulátoru
<b>Poskytované informace ze simulátoru</b>	Stav motorů, palivových nádrží, stav ovládacích prvků, informace o poloze, rychlosti, nadmořské výšce simulovaného letounu	Stav motorů, palivových nádrží, stav ovládacích prvků, informace o poloze, rychlosti, nadmořské výšce simulovaného letounu, o simulovaném letounu, současném letišti, provozu, nastaveních autopilotu, letišti
<b>Poskytuje možnost odesílání příkazů na simulátor</b>	Ne	Ano
<b>Poskytuje možnost ukládat záznam simulace do databáze</b>	Ano	Ne
<b>Poskytuje možnost přehrávat minulé simulace</b>	Ano	Ne
<b>Vizualizace letových dat</b>	Ano. Animovaný primární letový displej, ukazatele stavu paliva, motorů, ovládací prvky	Ne

**Tabulka 3. Porovnání SM a FSPS**

## 9 Možnosti budoucího rozšíření

V této kapitole jsou popsány možné směry budoucího vývoje systému pro správu simulátorů:

- Přidání možnosti odesílání příkazů na simulátor. Jedním z možných způsobů použití SM je při výuce budoucích pilotů. SM by byl použit instruktorem, který by při pomoci SM mohl vzdáleně sledovat průběh simulovaného letu. Možnost odesílání příkazů by v tom případě byla užitečná pro nastavení parametrů simulace pro připravení vhodné simulační situace. Příklady možných příkazů, které by se mohli odesílat na simulátor: nastavení počasí, změna množství paliva, přesunutí letadla na zadanou pozici, nastavení poškození letadla (např., nefungující motor), změna modelu letadla.
- Plynulejší animace primárního letového displeje. I když animace je dostatečně plynulá v současné verzi aplikace, pořád je prostor pro vylepšení.
- Vylepšení přehrávače záznamu již proběhlé simulace. V současnou chvíli přehrávač záznamu funguje tak, že na panelu řízení letu v SM uživatel klepne na některý bod na výškovém nebo rychlostním profilu letu a tím se stav panelu řízení letu nastaví na data, odpovídající tomuto časovému okamžiku. Možným vylepšením by bylo přidávání tlačítek “start” a “pause”. Po klepnutí na start, by SM průběžně (jednou za n vteřin) načítal data z databáze a zobrazoval je na panelu řízení letu. Po klepnutí na “pause” přehrávání záznamu letu by se vše pozastavilo.
- Vylepšení výkonu ukazatelů stavu paliva a motorů. Panel řízení letu teoretický může obsahovat až 144 ukazatelů stavu motorů (18 pro každý motor, celkem může mít simulátor až 8 motorů), což je velké množství při ohledu na to, že data se na těchto ukazatelích obnovují jednou za vteřinu. Pokud je ukazatelů hodně, aplikaci to zatěžuje a zpomaluje. Animace je pak méně plynulá, proto je výkon těch ukazatelů kritický pro aplikaci. Jedním z možných způsobů vylepšení výkonu indikátorů je cachování dat na klientské stráně (JavaScript) a překreslování ukazatelů jen v případě kdy nová hodnota je podstatně větší (menší), než nacachována hodnota.
- Vylepšení rozložení prvků na panelu řízení letu. V současné době se všechny ukazatele stavu motorů na panelu řízení letu nachází vedle sebe. Přívětivější pro uživatele by bylo, kdyby tyto ukazatele byly rozdělené do podskupin: výkon, teploty, elektrika.
- Rozšíření možnosti pro spravování letových dat. Tato práce byla více zaměřena na vizualizaci letových dat, než na jejich spravování, i když některé základní možnosti spravování dat poskytuje také: uživatel může najít libovolnou simulaci v historii simulací a pomocí přehrávače simulací ji přehrát. V případě, že na jednom simulátoru za den proběhne 10 simulací, budeme mít za rok na tomto simulátoru přes 3000 simulací. Najít určitou simulaci pak může být docela obtížné, proto by byly užitečné možnosti nastavování názvu pro určitou simulaci (“pilot Petr Švejk letí přes Atlantik”). Dále vyhledávání a filtrování simulace podle doby trvání, času začátku a konce či názvu. Také by byla užitečná možnost provedení skupinových operací nad

simulacemi, například: “smazat všechny simulace, proběhlé v dubnu 2015”. Dalším možným vylepšením v tomto směru by bylo přidání automatických skriptů, které by na základě některých nastavení průběžně mazali určitá data z databáze (například, simulace, kde doba trvání je méně než 5 vteřin).

- Přidání lepší podpory pro mobilní zařízení a jiné prohlížeče, než Chrome. V tuto chvíli SM funguje v prohlížečích Chrome a Chromium. V prohlížeči Firefox a na mobilních zařízeních Android aplikace sice funguje, ale panel řízení letu je zpomalený.
- V současnou chvíli SM podporuje simulátory X-Plane [28] a Microsoft Flight Simulator. Jedním z možných směrů budoucího vývoje je přidání podpory pro další simulátory, jako například: Flight Gear.
- Vylepšení rychlosti komunikace mezi klientskou a serverovou částí aplikace. V současnou chvíli klientská strana SM komunikuje z Vaadin serverem přes protokol HTTP. Z ohledu na obsah a frekvence odesílání zpráv vhodným by bylo analyzovat možnost použití novější technologie WebSocket, která poskytuje možnost rychlejšího odesílání zpráv [46].

## 10 Závěr

Vytvořený systém pro správu simulátorů letadel umožňuje instruktorovi či operátorovi v reálném čase sledovat přes webový prohlížeč stav simulátorů a vyhodnocovat dovednosti pilotů. Dále je umožněno:

- Přidávat, mazat, editovat údaje o modelech letadel (např. Boeing 737, Airbus 320 apod.), modelech motorů a simulátorech.
- Zobrazovat a vizualizovat probíhající simulace na simulátorech X-Plane ve webovém prohlížeči. Vytvořený panel řízení letu vizualizuje současný stav letových přístrojů, motorů, palivových nádrží, zobrazuje dráhu pohybu simulovaného letounu na mapě, vytváří výškový a rychlostní profily letu v reálném čase.
- Ukládat informace o probíhajících simulacích do databáze. To uživatelům nabízí možnost nejen vidět současné lety, ale taky ty minulé pro zpětnou analýzu a vyhodnocení.
- Spravovat letová data.

V kapitole 9 jsou popsány možnosti budoucího rozšíření systému.

# Literatura

- [1] ALLERTON, David. *Principles of flight simulation*. Chichester, U.K.: Wiley, 2009, xx, 471 p. Aerospace series (Chichester (England)). ISBN 04-707-5436-2.
- [2] JUKES, Malcolm. *Aircraft display systems*. Reston: AIAA, c2004, xxi, 338 s. ISBN 15-634-7657-6.
- [3] AERO: *Vertical situation display for improved flight safety and reduced operating costs* [online]. ColorGraphics, 2002 [cit. 2014-01-11]. Dostupné z: [http://www.boeing.com/commercial/aeromagazine/aero\\_20/vsd\\_story.html](http://www.boeing.com/commercial/aeromagazine/aero_20/vsd_story.html)
- [4] SPITZER, Cary R. *The avionics handbook*. Boca Raton: CRC Press, c2001, 1 v. (various pagings). ISBN 08-493-8348-X.
- [5] NASA. *No More Flying Blind* [online]. 2005 [cit. 2014-01-11]. Dostupné z: [http://www.nasa.gov/vision/earth/improvingflight/svs\\_reno.html](http://www.nasa.gov/vision/earth/improvingflight/svs_reno.html)
- [6] THE BOEING COMPANY. *737 Flight Crew Operations Manual* [online]. 2005 [cit. 2014-01-14]. Dostupné z: <http://www.smartcockpit.com/plane/BOEING/B737NG.html>
- [7] BRADY, Chris. NG Flight Instruments. In: *THE BOEING 737 TECHNICAL SITE* [online]. 1999 [cit. 2014-01-14]. Dostupné z: <http://www.b737.org.uk/flightinsts.htm>
- [8] KAN, OLATHE. Garmin® G1000® Certified for Cessna CitationJet. In: *Garmin: blogs* [online]. 2010 [cit. 2014-01-14]. Dostupné z: <http://garmin.blogs.com/pr/2010/07/garmin-g1000-certified-for-cessna-citationjet.html>
- [9] GARMIN. *Garmin* [online]. 2014 [cit. 2014-01-14]. Dostupné z: <http://www.garmin.com/en-US>
- [10] Garmin G1000. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2014-01-14]. Dostupné z: [http://en.wikipedia.org/wiki/Garmin\\_G1000](http://en.wikipedia.org/wiki/Garmin_G1000)
- [11] TL-6624 EFIS & EMS INTEGRA. *Avionics&Pilot supplies* [online]. 2014 [cit. 2014-01-14]. Dostupné z: <http://www.aeroshop.eu/efis-ems-integra-tl-6624.html>
- [12] EFIS & EMS INTEGRA TL-6624. *TLELECTRONIC* [online]. 2014 [cit. 2014-01-14]. Dostupné z: <http://www2.tl-elektronik.cz/>
- [13] Flightsimulator. *Flightsimulator* [online]. 2004 [cit. 2014-01-14]. Dostupné z: <http://www.petershemsida.net/flightpage1.html>
- [14] SportStar RTC. Technická data. *Evektor aerotechnik* [online]. 2014 [cit. 2014-01-15]. Dostupné z: <http://www.evektoraircraft.com/cs/letadla/sportstar-rtc/technicka-data>
- [15] 737-800 Technical Characteristics. *Boeing* [online]. 2014 [cit. 2014-01-15]. Dostupné z: [http://www.boeing.com/boeing/commercial/737family/pf/pf\\_800tech.page](http://www.boeing.com/boeing/commercial/737family/pf/pf_800tech.page)
- [16] ATR 42 - FULL FLIGHT SIMULATOR. *BalticAviationAcademy* [online]. 2014 [cit. 2014-01-15]. Dostupné z: <http://www.balticaa.com/en/facilities-and-services/full-flight-simulators-ffs/atr-42---full-flight-simulator/>
- [17] AeroWorks. *AeroWorks* [online]. 2014 [cit. 2014-01-15]. Dostupné z: <http://merlin.fit.vutbr.cz/AeroWorks/>
- [18] Electronic flight instrument system. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2014-01-15]. Dostupné z: [http://en.wikipedia.org/wiki/Electronic\\_flight\\_instrument\\_system](http://en.wikipedia.org/wiki/Electronic_flight_instrument_system)
- [19] Aircraft D-ERTC Photo. *Airport-data.com* [online]. 2004- [cit. 2014-01-15]. Dostupné z: <http://www.airport-data.com/aircraft/photo/000761964.html>
- [20] 777 Navigation Display: by jerome meriweather. *Jerome Meriweather* [online]. 2014 [cit. 2014-01-16]. Dostupné z: <http://www.meriweather.com/flightdeck/777/fwd/nd.html>
- [21] AEROWORKS, VUT-FIT. *Foto simulátorů Boeing 737 NG a EV-97 SportStar*.

- [22] *B737 NG Fuel System*. 2013. B737 NG Fuel System [online]. International Aviation Theory Centre [cit. 2015-05-11]. B 737 NG Ground School. Dostupné z: <http://www.slideshare.net/theoryce/b737-ng-fuel-system-20650855>
- [23] *Boeing 737-800*. Flugzeuginfo [online]. [cit. 2015-05-11]. Dostupné z: [http://www.flugzeuginfo.net/acdata\\_php/acdata\\_7378\\_en.php](http://www.flugzeuginfo.net/acdata_php/acdata_7378_en.php)
- [24] *BOEING 737*. U.S. Division Academy [online]. [cit. 2015-05-11]. Dostupné z: <http://ivaous.org/academy/index.php/pilotmenu/atp-airline-transport-pilot/boeing-737-800>
- [25] *Boeing 737 limitations*. BRADY, Chris. Boeing 737 Technical Site [online]. [cit. 2015-05-11]. Dostupné z: <http://www.b737.org.uk/limitations.htm>
- [26] *Boeing 737 Technical Specifications* [online]. BRADY, Chris. [cit. 2015-05-11]. Dostupné z: <http://www.b737.org.uk/techspecsdownload.htm>
- [27] *Boeing 737 Centre Instrument Panel*. BRADY, Chris. *Boeing 737 Technical Site* [online]. [cit. 2015-05-11]. Dostupné z: <http://www.b737.org.uk/panelcentreinst.htm>
- [28] *X-Plane 10 Global. The World's Most Advanced Flight Simulator* [online]. [cit. 2015-05-11]. Dostupné z: <http://www.x-plane.com/>
- [29] *Java 8 Central*. Oracle [online]. [cit. 2015-05-11]. Dostupné z: <http://www.oracle.com/technetwork/java/javase/overview/java8-2100321.html>
- [30] *Vaadin - user interface components for business apps* [online]. [cit. 2015-05-11]. Dostupné z: <https://vaadin.com/home>
- [31] *Apache Tomcat* [online]. [cit. 2015-05-11]. Dostupné z: <http://tomcat.apache.org/>
- [32] *PostgreSQL* [online]. [cit. 2015-05-11]. Dostupné z: <http://www.postgresql.org/>
- [33] *Google Maps API* [online]. [cit. 2015-05-11]. Dostupné z: <https://developers.google.com/maps/>
- [34] *Google Charts* [online]. [cit. 2015-05-11]. Dostupné z: <https://developers.google.com/chart/>
- [35] *Interactive JavaScript charts for your webpage* [online]. [cit. 2015-05-11]. Dostupné z: <http://www.highcharts.com/>
- [36] *GitHub* [online]. [cit. 2015-05-11]. Dostupné z: <https://github.com/>
- [37] *JIRA - Issue & Project Tracking Software* [online]. [cit. 2015-05-11]. Dostupné z: <https://www.atlassian.com/software/jira>
- [38] *GreenSock | GSAP: The new standard for HTML5 animation* [online]. [cit. 2015-05-11]. Dostupné z: <http://greensock.com/>
- [39] *Database Management Software Tools: DbVisualizer* [online]. [cit. 2015-05-11]. Dostupné z: <https://www.dbvis.com/>
- [40] *EVEKTOR - AEROTECHNIK, A.S. 2009. Aircraft Operating Instructions for SportStar light sport aircraft* [online]. Kunovice [cit. 2015-05-11]. Dostupné také z: <http://www.aerodynamicaviation.com/documents/Evektor%20Sportstar%20Max%20POH.pdf>
- [41] RYDLO, Karol. 2012. *Vizualizace vzdušného prostoru ve 3D*. Brno. Diplomová práce. Masaryková univerzita.
- [42] *AbstractJavaScriptComponent - Vaadin documentation*. 2014. *Vaadin 7.4.1 API* [online]. [cit. 2015-05-12]. Dostupné z: <https://vaadin.com/api/7.4.1/com/vaadin/ui/AbstractJavaScriptComponent.html>
- [43] *Package java.awt. Java™ Platform, Standard Edition 7 API Specification* [online]. [cit. 2015-05-18]. Dostupné z: <http://docs.oracle.com/javase/7/docs/api/java/awt/package-summary.html>
- [44] *Trail: Creating a GUI With JFC/Swing. The Java™ Tutorials* [online]. [cit. 2015-05-18]. Dostupné z: <http://docs.oracle.com/javase/tutorial/uiswing/>
- [45] 11.3. *Debug Mode and Window. Book of Vaadin 7* [online]. [cit. 2015-05-21]. Dostupné z: <https://vaadin.com/book/vaadin7/-/page/advanced.debug.html>
- [46] *SOCKET.IO IS OPEN-SOURCE (MIT). Socket.IO* [online]. [cit. 2015-05-21]. Dostupné z: <http://socket.io/>

- [47] FSPS - FLIGHT SIMULATOR MANAGER. *SimMarket* [online]. [cit. 2015-05-21]. Dostupné z: [http://secure.simmarket.com/fsps-flight-simulator-manager-%28en\\_4845%29.phtml](http://secure.simmarket.com/fsps-flight-simulator-manager-%28en_4845%29.phtml)
- [48] Flight Simulator Manager. *FSPS : Flight Simulator add-on software* [online]. [cit. 2015-05-21]. Dostupné z: [http://www.thefsp.com/freeware/31-fsps-flight-simulator-manager.html?search\\_query=fsps&results=25](http://www.thefsp.com/freeware/31-fsps-flight-simulator-manager.html?search_query=fsps&results=25)
- [49] Power Plant. *The Boeing 737 Technical Site* [online]. [cit. 2015-05-24]. Dostupné z: <http://www.b737.org.uk/powerplant.htm>

# Příloha A – komunikační protocol a posílane veličiny

Táto příloha popisuje komunikační protokol AWCCom [41], použitý pro komunikace ze simulátory.

Zkratka	Jednotka	Název anglický	Název český	Doplňující popis
LON	°	Longitude	Zeměpisná délka	
LAT	°	Latitude	Zeměpisná šířka	
PIT	°	Pitch angle	Úhel klopení	
BAN	°	Bank angle	Úhel klonění	
HEA	°	Heading	Kurz magnetický	
ALTS	m	Altitude standard	Výška	
GAL	m	Ground altitude	Výška země	
IAS	m/s	Indicated airspeed	Indikovaná rychlost	
TAS	m/s	True airspeed	Skutečná dopředná rychlost	
VS_	ft/min	Vertical speed	Vertikální rychlost	
DTR	°	Rudder trim	Výchylka směrového kormidla	
DTE	°	Elevator trim	Výchylka výškovky	
DTA	°	Aileron trim	Výchylka křídylka	
CR_	°	Rudder angle	Úhel směrového kormidla	
CA_	°	Aileron angle	Úhel křídylka	
CE_	°	Elevator angle	Úhel výškovky	
TT_		True track	Skutečný kurz	
FLP	°	Flaps angle	Úhel klapky	
SBRK	°	Speed brakes	Úhel aerodynamických	



			brzd	
<b>BRK</b>	0, 1	Brakes	Brzdy	0 – není zabrzděný 1 – je zabrzděný
<b>LG_1</b>	-1, 0, 1, 2	Landing gear 1	Podvozek 1	0 – nahoře 1 – v pohybu 2 – dole -1 – není podporovan
<b>LG_2</b>	-1, 0, 1, 2	Landing gear 2	Podvozek 2	Stejně, jako u LG_1
<b>LG_3</b>	-1, 0, 1, 2	Landing gear 3	Podvozek 3	Stejně, jako u LG_1
<b>SSPD</b>	0, 1	Is on pause	Zda je na pauze	0 – simulator je pozastaven 1 – simulator běží
<b>Hodnoty, popisující stav motorů. Každá hodnota se posílá pro každý motor zvlášť (např. pro motor 0 a hodnotu RPM, posílány klíč vypadá tak: „E0RPM“)</b>				
<b>RPM</b>	rev/min	Revolutions Per Minute	Otáčky motoru	
<b>PWR</b>	Watt	Power	Výkon motoru	
<b>PWP</b>	%	Power in Percents	Výkon motoru v procentech	
<b>MP</b>	Pa	Manifold Pressure	Plnicí tlak	
<b>EGT1</b>	°C	Exhaust Gas Temperature 1	Teplota výfukových plynů 1	
<b>EGT2</b>	°C	Exhaust Gas Temperature 2	Teplota výfukových plynů 2	
<b>CHT1</b>	°C	Cylinder Head Temperature 1	Teplota hlav válců	
<b>CHT2</b>	°C	Cylinder Head Temperature 2	Teplota hlav válců	
<b>EST</b>	°C	Engine Suction	Teplota sání	

		Temperature	motoru	
<b>FF</b>	l/hour	Fuel Flow	Průtok paliva	
<b>FP</b>	Pa	Fuel Pressure	Tlak paliva	
<b>OP</b>	Pa	Oil Pressure	Tlak oleje	
<b>OT</b>	°C	Oil Temperature	Teplota oleje	
<b>N1</b>	%	RPM in percents	RPM v procentech	
<b>N2</b>	%	RPM in percents	RPM v procentech	
<b>VIB</b>		Vibration	Vibrace	
<b>VLT</b>	V	Voltage	Napětí	
<b>AMP</b>	A	Amperage	Proud	
<b>LFU</b>	l	Left fuel tank	Levá palivá nádrž	
<b>RFU</b>	l	Right fuel tank	Pravá palivá nádrž	
<b>CFU</b>	l	Central fuel tank	Prostřední palivá nádrž	

# Příloha B. Příručka pro použití systému pro správu simulátorů

V této příloze je popsány jak systém pro správu simulátorů se spouští, překládá.

## Spouštění systému pro správu simulátorů

Předpoklad: je potřeba mít nainstalovány X-Plane s pluginem AWCom. Stroj z X-Plane simulátorem má být přístupný buď přes Internet nebo se nacházet ve stejné síti z SM. Aplikace potřebuje mít přístup na Internet pro funkčnost map a budíků, protože obojí používají Google API.

1. Stáhněte si soubor SimulatorManager.war z GitHubu anebo jej naleznete na dodaném CD, který je součástí této práce, ve složce “deployable”.

<https://github.com/IevgenRagulin/SimulatorManager/tree/master/deployable>.

2. Nainstalujte Java Runtime Environment verze 8.

<http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>

3. Nainstalujte si Apache Tomcat 8.0.

<https://tomcat.apache.org/download-80.cgi>

4. Změňte soubor tomcat\_installation\_directory/conf/tomcat-users.xml. Je potřeba přidat do souboru tyto řádky:

```
<role rolename="manager"/>
<user password="user" roles="manager" username="user"/>
```

Místo „user“ je potřeba uvést přihlašovací jméno a heslo, které byste chtěli používat pro přihlášení do aplikace.

5. Nainstalujte databázový systém PostgreSQL 9.3, nainstalujte databázový systém. Nejlepší pro výkon by bylo, pokud by databázový systém běžel na stejném stroji nebo alespoň ve stejné síti z SM.

<http://www.enterprisedb.com/products-services-training/pgdownload#windows>

6. Vytvořte databázi v PostgreSQL. Pokud chcete, aby systém podporoval češtinu, je potřeba, aby kolace databáze byla UTF8 nebo jiná, která podporuje češtinu.

7. Otevřete soubor

`SimulatorManager.war/WebContent/WEB-INF/simulatorManager.prop.`

- Nastavte **db\_url** na URL, které odpovídá vaší instalaci databáze. Například `jdbc:postgresql://localhost/postgres` (postgres je tady je název vaší databáze)
- Nastavte **db\_username** a **db\_password**, což jsou uživatelské jméno a heslo pro přístup k databázi.

8. Spusťte Apache Tomcat 8.0.
9. Překopírujte soubor SimulatorManager.war, stahnutý v prvním kroku, do složky  
tomcat\_installation\_directory/wtpwebapps.
10. Otevřte stránku <http://localhost:8080/SimulatorManager/> ve vašem prohlížeči.  
Doporučenými prohlížeči jsou Chrome a Chromium. Ve Firefoxu aplikace sice funguje, ale panel řízení letu je subjektivně pomalý.
11. Zadejte uživatelské jméno a heslo, které jste přidali do souboru tomcat-users.xml.

## Editování konfigurace aplikace

Konfigurační soubor aplikace se nachází na této cestě:

`SimulatorManager.war/WebContent/WEB-INF/simulatorManager.prop`

V tomto souboru se nastavují přihlašovací údaje pro databáze, frekvence, ze kterou SM získává data ze simulátorů, ukládá data do databáze, obnovuje uživatelské rozhraní, výchozí hodnoty pro nové modely simulátorů, motorů apod.

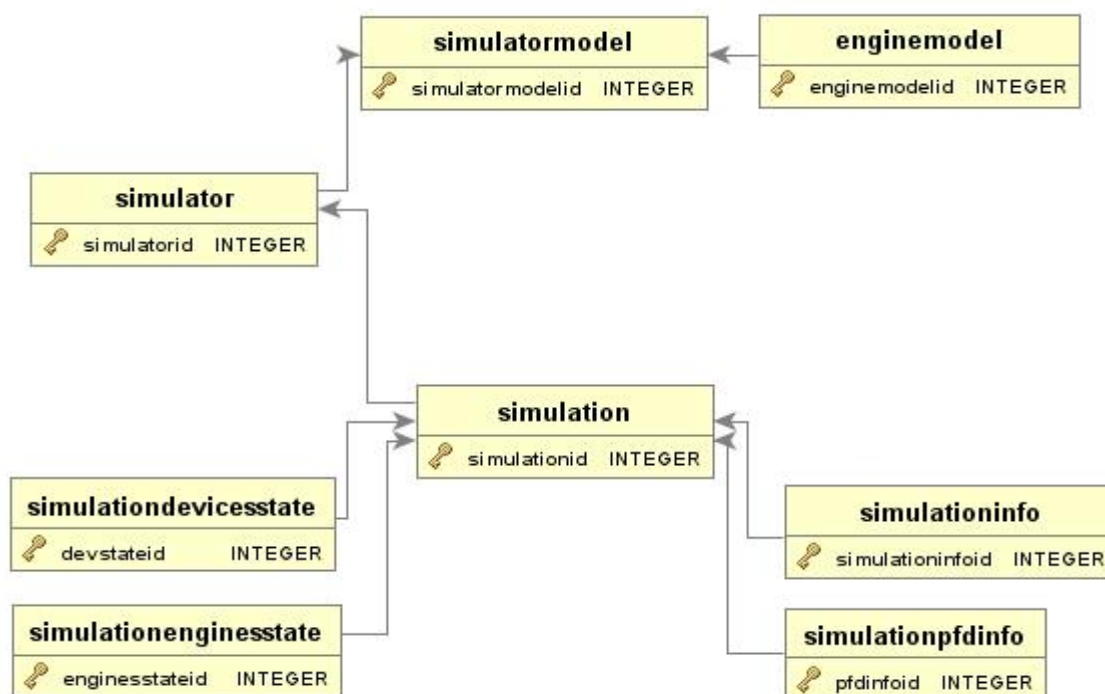
## Překlad kódu systému pro správu simulátorů

Překlad kódu je potřebný jen v tom případě, pokud sami chcete něco změnit v kódu aplikace. Pokud nechcete nic měnit, stačí používat již přeloženou verzi aplikaci. Pro překlad aplikaci je potřeba:


1. Nainstalovat si Java Runtime Environment 8.
2. Nainstalovat si Eclipse 4.4.0.
3. Nainstalovat do Eclipse plugin Vaadin.
4. Stáhnout zdrojové kody z GitHubu <https://github.com/IevgenRagulin/SimulatorManager>
5. Naimportovat si projekt do Eclipse.
6. Pomoci buildovacího nástroje Ivy, který je součástí již nainstalovaného Eclipse pluginu Vaadin, stáhnout všechny potřebné závislosti pro projekt.
7. Přeložit WidgetSet/Theme aplikace v Eclipse pomocí Vaadin pluginu.
8. Spustit aplikaci na Tomcat Serveru.

## Příloha C. Databázový model systému pro správu simulátorů


V této příloze jsou materiály k popisu databázového modelu, používaného systémem pro správu simulátorů. SM používá databázový systém PostgreSQL 9.3 pro ukládání modelů letadel, modelů motorů, konfiguraci simulátorů, informací o proběhlých simulacích. Popis databázového modelu je v kapitole 6.1.3, tady jsou jen materiály k tomu popisu.



Obr. 47. Databázový model systému pro správu simulátorů


simulatormodel		
	simulatormodelid	INTEGER
	simulatormodelname	CHARACTER VARYING(255)
	minspeed	INTEGER
	maxspeed	INTEGER
	maxspeedonflaps	INTEGER
	minspeedonflaps	INTEGER
	hasgears	BOOLEAN
	numberoflandinggears	INTEGER
	lfu	BOOLEAN
	minlfu	REAL
	lowlfu	REAL
	highlfu	REAL
	maxlfu	REAL
	cfu	BOOLEAN
	mincfu	REAL
	lowcfu	REAL
	highcfu	REAL
	maxcfu	REAL
	rfu	BOOLEAN
	minrfu	REAL
	lowrfu	REAL
	highrfu	REAL
	maxrfu	REAL
	timestamp	TIMESTAMP(6) WITHOUT TIME ZONE

Obr. 48. Databázová tabulka simulatormodel


enginemodel			
 enginemodelid	INTEGER	est	BOOLEAN
simulatormodelid	INTEGER	minest	REAL
enginemodelorder	INTEGER	lowest	REAL
rpm	BOOLEAN	highest	REAL
minrpm	REAL	maxest	REAL
lowrpm	REAL	ff_	BOOLEAN
highrpm	REAL	minff	REAL
maxrpm	REAL	lowff	REAL
pwr	BOOLEAN	highff	REAL
minpwr	REAL	maxff	REAL
lowpwr	REAL	fp_	BOOLEAN
highpwr	REAL	minfp	REAL
maxpwr	REAL	lowfp	REAL
pwp	BOOLEAN	highfp	REAL
lowpwp	REAL	maxfp	REAL
highpwp	REAL	op_	BOOLEAN
minpwp	REAL	minop	REAL
maxpwp	REAL	lowop	REAL
mp_	BOOLEAN	highop	REAL
minmp	REAL	maxop	REAL
lowmp	REAL	ot_	BOOLEAN
highmp	REAL	minot	REAL
maxmp	REAL	lowot	REAL
et1	BOOLEAN	highot	REAL
minet1	REAL	maxot	REAL
lowet1	REAL	n1_	BOOLEAN
highet1	REAL	minn1	REAL
maxet1	REAL	lown1	REAL
et2	BOOLEAN	highn1	REAL
minet2	REAL	maxn1	REAL
lowet2	REAL	n2_	BOOLEAN
highet2	REAL	minn2	REAL
maxet2	REAL	lown2	REAL
ct1	BOOLEAN	highn2	REAL
minct1	REAL	maxn2	REAL
lowct1	REAL	vib	BOOLEAN
highct1	REAL	minvib	REAL
maxct1	REAL	lowvib	REAL
ct2	BOOLEAN	highvib	REAL
minct2	REAL	maxvib	REAL
lowct2	REAL	vlt	BOOLEAN
highct2	REAL	minvlt	REAL
maxct2	REAL	lowvlt	REAL
		highvlt	REAL
		maxvlt	REAL
		amp	BOOLEAN
		minamp	REAL
		lowamp	REAL
		highamp	REAL
		maxamp	REAL
		timestamp	TIMESTAMP(6) WITHOUT TIME ZONE

Obr. 49. Databázová tabulka enginemodel




simulator		
	simulatorid	INTEGER
	simulatormodelid	INTEGER
	simulatorname	CHARACTER VARYING(255)
	hostname	CHARACTER VARYING(255)
	port	INTEGER
	active	BOOLEAN
	timestamp	TIMESTAMP(6) WITHOUT TIME ZONE


Obr. 50. Databázová tabulka simulator

simulation		
	simulationid	INTEGER
	simulatorid	INTEGER
	issimulationon	BOOLEAN
	issimulationpaused	BOOLEAN
	simulationstartedtime	TIMESTAMP(6) WITHOUT TIME ZONE
	latestupdateime	TIMESTAMP(6) WITHOUT TIME ZONE
	timestamp	TIMESTAMP(6) WITHOUT TIME ZONE


Obr. 51. Databázová tabulka simulation

simulationenginesstate		
	simulationenginesstateid	INTEGER
	simulationid	INTEGER
	engines_num	INTEGER
	rpm	REAL[]
	pwr	REAL[]
	pwp	REAL[]
	mp_	REAL[]
	et1	REAL[]
	et2	REAL[]
	ct1	REAL[]
	ct2	REAL[]
	est	REAL[]
	ff_	REAL[]
	fp_	REAL[]
	op_	REAL[]
	ot_	REAL[]
	n1_	REAL[]
	n2_	REAL[]
	vib	REAL[]
	vlt	REAL[]
	amp	REAL[]
	timestamp	TIMESTAMP(6) WITHOUT TIME ZONE


Obr. 52. Databázová tabulka simulationenginesstate

simulationinfo	
 simulationinfoid	INTEGER
simulationid	INTEGER
longitude	DOUBLE PRECISION
latitude	DOUBLE PRECISION
timestamp	TIMESTAMP(6) WITHOUT TIME ZONE

Obr. 53. Databázová tabulka simulationinfo

simulationdevicesstate	
 simulationdevicesstateid	INTEGER
simulationid	INTEGER
elevator	DOUBLE PRECISION
eleron	DOUBLE PRECISION
rudder	DOUBLE PRECISION
throttle	DOUBLE PRECISION
flaps	DOUBLE PRECISION
speedbrakes	DOUBLE PRECISION
brakes	BOOLEAN
issimulationpaused	BOOLEAN
elevatortrim	DOUBLE PRECISION
ailerontrim	DOUBLE PRECISION
ruddertrim	DOUBLE PRECISION
landinggear_1	INTEGER
landinggear_2	INTEGER
landinggear_3	INTEGER
lfu	REAL
rfu	REAL
cfu	REAL
timestamp	TIMESTAMP(6) WITHOUT TIME ZONE

Obr. 54. Databázová tabulka simulationdevicesstate

simulationpfdinfo	
 simulationpfdinfoid	INTEGER
simulationid	INTEGER
roll	DOUBLE PRECISION
pitch	DOUBLE PRECISION
heading	DOUBLE PRECISION
truecourse	DOUBLE PRECISION
ias	DOUBLE PRECISION
altitude	DOUBLE PRECISION
groundaltitude	DOUBLE PRECISION
verticalspeed	DOUBLE PRECISION
timestamp	TIMESTAMP(6) WITHOUT TIME ZONE

Obr. 55. Databázová tabulka simulationpfdinfo

# Příloha D. Parametry simulovaných letounů

Data, uvedená v této příloze, se používají pro konfigurování modelu simulátoru v SM.

## Parametry letounu EV-97 Sportstar RTC

Následující data jsou převzata z [14], [40]

<b>Objem palivových nádrží</b>	<b>120 l</b>
<b>Nepřekročitelná rychlost</b>	270 km/h
<b>Maximální rychlost na klapkách</b>	212 km/h
<b>Rychlost stoupání</b>	7.5 m/s
<b>Dolet</b>	1 300 km

### Pohonná jednotka

<b>Motor</b>	<b>Rotax 912 ULS</b>
<b>Spotřeba paliva</b>	15 l / hr
<b>Maximální RPM</b>	5800
<b>Maximální permanentní RPM</b>	5500
<b>Maximální teplota oleje</b>	130 °C
<b>Optimální teplota oleje</b>	88 – 110 °C
<b>Minimální tlak oleje</b>	12 PSI
<b>Maximální tlak oleje</b>	102 PSI
<b>Optimální tlak oleje</b>	29 – 73 PSI
<b>Minimální tlak paliva</b>	2.2 PSI
<b>Maximální CHT</b>	135 °C

## Parametry letounu 737-800

<b>Motory</b>	<b>CFMI CFM56-8</b>
<b>Maximální vzdálenost</b>	5,765 km [15]
<b>Typická rychlost jízdy</b>	0.785 mach [15]
<b>Maximální rychlost</b>	946 km/h [23]
<b>Maximální rychlost na klapkách</b>	350 km/h [24]

<b>Objem levé palivové nádrže</b>	4876 l [22]
<b>Objem pravé palivové nádrže</b>	4876 l [22]
<b>Objem prostřední palivové nádrže</b>	16273 l [22]
<b>Maximální EGT</b>	950 [25]
<b>L.G. Průtok paliva</b>	2187 kg/h [26]
<b>Minimální tlak oleje</b>	13 psi [25]
<b>Maximální teplota oleje</b>	155 °C [25]
<b>Maximální N1</b>	104 °C [25]
<b>Maximální N2</b>	105 °C [25]
<b>Maximální vibrace</b>	5.0 [27]