



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

MODELOVÁNÍ A SIMULACE ŘÍZENÍ PODÉLNÉHO LETU LETOUNU

MODELLING AND SIMULATION OF THE LONGITUDINAL FLIGHT CONTROL

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Dominik Zavadil

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Miroslav Jirgl, Ph.D.

BRNO 2024

Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Dominik Zavadil

ID: 240472

Ročník: 3

Akademický rok: 2023/24

NÁZEV TÉMATU:

Modelování a simulace řízení podélného letu letounu

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je seznámit se s možnostmi modelování podélného letu letounu, následně sestavit simulační model a zhodnotit možnosti jeho řízení.

1. Seznamte se s možnostmi modelování podélného letu letounu a stručně popište.
2. Vyberte vhodnou strukturu modelu a identifikujte její parametry na základě dostupných dat z leteckého simulátoru.
3. Vyhodnoťte základní parametry/charakteristiky systému (modelu).
4. Verifikujte správnost modelu na základě měření odezev na další vstupní signály.
5. Realizujte simulační model pro řízení podélného letu.
6. Proveďte simulaci a zhodnoťte dosažené výsledky.

DOPORUČENÁ LITERATURA:

FOSSSEN, T. I. Mathematical Models for Control of Aircraft and Satellites [online]. 3rd ed. Department of Engineering Cybernetics, NTNU, 2013. Dostupné z: <https://www.scribd.com/document/300312683/Aircraft-Fossen-2013>

Termín zadání: 5.2.2024

Termín odevzdání: 22.5.2024

Vedoucí práce: Ing. Miroslav Jirgl, Ph.D.

Ing. Miroslav Jirgl, Ph.D.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Tato bakalářská práce se zaměřuje na návrh modelu podélného letu letounu, identifikaci jeho parametrů a následnou validaci získaného modelu. Teoretická část popisuje modelování podélného letu pomocí pohybových rovnic, které jsou následně linearizovány a transformovány do stavového popisu. V praktické části se pro identifikaci dvou modelů a jejich následnou validaci používala data změřená na leteckém simulátoru. Pro generování vstupních signálů u validačních dat byl použit joystick emulovaný deskou Arduino Leonardo. Na závěr byl z validovaných modelů sestaven simulační model, který lze použít pro zjištění odezvy letounu na různé typy vstupních signálů.

Klíčová slova

Model, identifikace, podélný let, MATLAB, validace, simulace, Arduino Leonardo

Abstract

This bachelor thesis focuses on the design of a longitudinal flight model of an aircraft, identification of its parameters and validation of the obtained model. The theoretical part describes the modelling of longitudinal flight using equations of motion, which are then linearized and transformed into a state-space representation. In the practical part, data measured on a flight simulator were used to identify two models and for their subsequent validation. A joystick emulated by an Arduino Leonardo board was used to generate input signals in the process of acquiring the validation data. Finally, the validated models were used to build a simulation model that can be used to determine the response of the aircraft to different types of input signals.

Keywords

Model, identification, longitudinal flight, MATLAB, validation, simulation, Arduino Leonardo

Bibliografická citace

ZAVADIL, Dominik. *Modelování a simulace řízení podélného letu letounu*. Brno, 2024. Dostupné také z: <https://www.vut.cz/studenti/zav-prace/detail/160029>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Miroslav Jirgl.

Prohlášení autora o původnosti díla

Jméno a příjmení studenta: *Dominik Zavadil*

VUT ID studenta: *240472*

Typ práce: *Bakalářská práce*

Akademický rok: *2023/24*

Téma závěrečné práce: *Modelování a simulace řízení podélného letu letounu*

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: 22. května 2024

podpis autora

Poděkování

Chtěl bych tímto poděkovat vedoucímu své bakalářské práce Ing. Miroslavu Jirglovi, Ph.D. za vstřícný přístup a pomoc při realizaci a psaní práce.

V Brně dne: 22. května 2024

podpis autora

Obsah

SEZNAM OBRÁZKŮ	9
SEZNAM TABULEK.....	10
ÚVOD	11
1. ZPŮSOBY MODELOVÁNÍ PODÉLNÉHO LETU.....	12
1.1 SOUŘADNICOVÉ SYSTÉMY	12
1.1.1 Letadlový souřadnicový systém.....	13
1.1.2 Zemský souřadnicový systém	13
1.1.3 Aerodynamický souřadnicový systém.....	14
1.1.4 Vztahy mezi zemským a letadlovým souřadnicových systémem	14
1.1.5 Vztahy mezi aerodynamickým a letadlovým souřadnicovým systémem	15
1.2 POPIS TVORBY MODELU PODÉLNÉHO LETU	16
1.2.1 Pohybové rovnice.....	16
1.2.2 Linearizace.....	17
1.2.3 Stavový popis	18
1.2.4 Transformace linearizovaných rovnic do stavového popisu	19
1.3 DYNAMIKA PODÉLNÉHO LETU	20
1.3.1 Rychlé kmity.....	20
1.3.2 Pomalé kmity (fugoidální).....	21
1.3.3 Integrátor.....	21
2. NÁVRH A IDENTIFIKACE MODELU.....	22
2.1 STRUKTURA MODELU	22
2.2 IDENTIFIKACE PARAMETRŮ MODELU	23
2.2.1 Sběr dat.....	23
2.2.2 Postup identifikace.....	24
2.3 DYNAMICKÉ VLASTNOSTI MODELU	26
3. VALIDACE MODELU.....	28
3.1 POUŽITÉ SIGNÁLY	28
3.1.1 Puls	28
3.1.2 Rampa	29
3.1.3 PRBS (Pseudo-Random Binary Signal)	29
3.2 TVORBA APLIKACE PRO GENEROVÁNÍ VSTUPNÍCH SIGNÁLŮ	31
3.2.1 Arduino Leonardo.....	31
3.2.2 Joystick knihovna.....	32
3.2.3 Vlastní program	32
3.3 MĚŘENÍ A ZPRACOVÁNÍ NAMĚŘENÝCH DAT	33
3.3.1 Naměření dat na simulátoru X-PLANE 10.....	33
3.3.2 Extrahování z naměřených dat.....	35
3.3.3 Zpracování extrahovaných dat.....	36
3.3.4 Problém se vzorkováním	37
3.4 POROVNÁNÍ NAMĚŘENÝCH DAT A ODEZEV MODELU	38
3.4.1 Porovnání průběhů pro jednotlivé validační signály.....	40
3.4.2 Porovnání vyváženého a nevyváženého letu	44

3.4.3	<i>Porovnání průběhů s různými tahy</i>	45
4.	SIMULAČNÍ MODEL PRO ŘÍZENÍ PODÉLNÉHO LETU	46
4.1	POPIS SIMULAČNÍHO MODELU	47
4.2	IMPLEMENTACE VÝPOČTU VÝŠKY U 4-STAVOVÉHO MODELU	47
4.3	PRŮBĚH SIMULACE PRO JEDNOTLIVÉ VSTUPNÍ SIGNÁLY	48
4.3.1	<i>Signál z joysticku (Arduina)</i>	48
4.3.2	<i>Signál z Workspace</i>	49
4.3.3	<i>Data změřená na UNOB</i>	49
4.3.4	<i>Ručně poskládaný signál</i>	49
4.4	UKÁZKA SIMULOVANÝCH PRŮBĚHŮ	50
5.	ZÁVĚR	54
	LITERATURA	55
	SEZNAM SYMBOLŮ A ZKRATEK	57

SEZNAM OBRÁZKŮ

1.1	Rozdíl v osách mezi normami GOST a ICAO [1].....	12
1.2	Letadlový souřadnicový systém [2].....	13
1.3	Vztah mezi letadlovým a zemským souřadnicovým systémem, upraveno [2]	13
1.4	Vztah mezi aerodynamickým a letadlovým souřadnicovým systémem, upraveno [2].....	14
1.5	Postupné rotace ze zemského do letadlového souřadnicového systému, upraveno [2]	15
1.6	Síly působící na letadlo a významné úhly mezi osami, upraveno [1].....	16
1.7	Typické rozložení pólů v komplexní rovině [6]	20
1.8	Ukázka působení rychlých kmitů [2]	21
1.9	Ukázka vlivu pomalých kmitů [2].....	21
2.1	Letecký simulátor na UNOB	23
2.2	Simulační obvod použitý k validaci modelu	24
2.3	Porovnání odezev modelů a identifikačních dat z UNOB.....	25
2.4	Vykreslení pólů pomocí příkazu <i>pzmap</i>	26
3.1	Ukázka pulsu a jeho parametrů	29
3.2	Ukázka rampy a jejích parametrů.....	29
3.3	Ukázka PRBS a jeho parametrů	30
3.4	Použitá struktura 8-bitového Fibonacci LFSR	30
3.5	Použitá deska Arduino Leonardo	31
3.6	Vývojový diagram generování signálů v Arduino	33
3.7	Blokové schéma měřicí stanice	34
3.8	Formát naměřených dat.....	35
3.9	Ukázka průběhu Q z neupravené datové sady.....	36
3.10	Ukázka průběhu Q po úpravě dat	36
3.11	Histogram vzorkovací periody	37
3.12	Porovnání resamplovaného a interpolovaného PRBS	38
3.13	Histogram vzorkovacích period po interpolaci	38
3.14	Porovnání pro malý puls	40
3.15	Porovnání pro velký puls.....	41
3.16	Porovnání pro rampu.....	42
3.17	Porovnání pro PRBS	43
3.18	Porovnání dat z nevyváženého letu	44
3.19	Průběh výšky letu pro různé tahy motoru.....	45
4.1	Grafické rozhraní simulace	46
4.2	Simulační model.....	47
4.3	Výpočet výšky u 4-stavového modelu	48
4.4	Propojení mezi joystickem (Arduinem) a Simulinkem	49
4.5	Panel pro tvorbu vlastního signálu	50
4.6	Ukázka použití simulačního modelu	50
4.7	Ukázka simulačního modelu – signál Arduino.....	51
4.8	Ukázka simulačního modelu – ruční signál.....	52
4.9	Výběr typu signálu pro UNOB simulaci	53
4.10	Ukázka simulačního modelu – UNOB signál.....	53

SEZNAM TABULEK

2.1	Póly systémů a jejich parametry	27
-----	---------------------------------------	----

ÚVOD

Modelování a simulace letu letounu mají klíčovou roli v leteckém inženýrství, a to nejen při navrhování nových letounů, ale i u optimalizace stávajících leteckých konstrukcí. Problematika tohoto modelování je velmi komplexní, zejména kvůli používání nelineárních rovnic s velkým množstvím parametrů. V práci byl ale použit značně zjednodušený linearizovaný model, který reprezentuje chování letounu pouze při specifických podmínkách.

Podélný let zahrnuje pohyb letounu ve vertikální rovině. Obě křídla jsou vždy ve stejné výšce, letadlo nebude zatáčet a může se pouze naklánět nahoru a dolů, čímž dokáže měnit svou výšku. V teoretické části jsou uvedeny použité souřadnicové systémy, dále je popsán let letounu nelineárními pohybovými rovnicemi, které jsou následně linearizovány a přepsány do tvaru stavového popisu.

Cílem této práce bylo vytvoření vlastního modelu, který popisuje dynamické chování letounu při podélném letu, následná identifikace jeho parametrů a validace pomocí jiných dat. Na závěr byl z těchto modelů vytvořen simulační model s různými možnostmi řízení.

Tato práce je členěna do čtyřech kapitol. V kapitole 1 je popsána základní teorie týkající se modelování podélného letu letounu a stručný rozbor dynamických vlastností týkajících se podélného letu. Náplní kapitoly 2 je návrh vlastní struktury modelu a provedení identifikace potřebných parametrů z naměřených dat. V kapitole 3 je rozebrána validace získaných modelů pomocí nově naměřených dat na leteckém simulátoru. Na jejím závěru je také zhodnocení dosažených výsledků samotné validace. Kapitola 4 ukazuje simulační model sestavený v Simulinku a jednotlivé možnosti jeho ovládání.

1. ZPŮSOBY MODELOVÁNÍ PODÉLNÉHO LETU

V této kapitole je teoreticky popsán způsob, jakým je možno namodelovat podélný let letadla, a co vše je k tomu třeba znát.

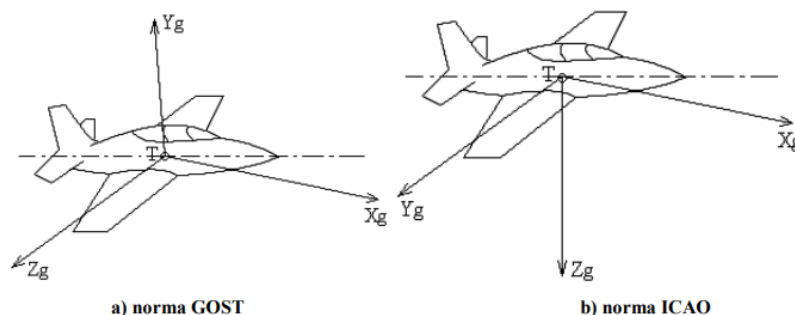
Jsou zde přiblíženy souřadnicové systémy, které hrají roli ve stanovení vazeb mezi důležitými veličinami. Dále je uveden způsob získání pohybových rovnic a jejich následná linearizace a přepis do stavového popisu. Nakonec jsou popsány základní dynamické vlastnosti podélného letu.

1.1 Souřadnicové systémy

Existují různé souřadnicové systémy (zemský, letadlový, aerodynamický aj.). Každý systém v sobě umožňuje popsat důležité veličiny s ním svázané – rychlosti a síly. Je možné popis těchto veličin transformovat z jednoho systému do druhého pomocí tzv. *matic rotací*. [1][2]

Matice rotace je tvořena goniometrickými funkcemi sinus a cosinus. Jejich argumenty jsou Eulerovy úhly, o které se jednotlivé osy systému otočí oproti jejich původní poloze. Jakmile se provedou všechny nutné rotace pro dokončení transformace, vektory rychlosti a síly z původního systému mohou být popsány v nově dosaženém systému. Popis je vytvořen pomocí maticového vynásobení všech postupně použitých matic rotací s vektorem původních rychlostí nebo sil. [2][3][4]

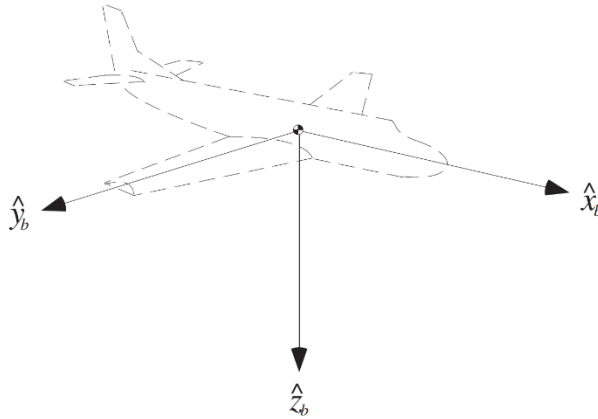
Ve světě se používají dvě normy pro zápis souřadnicových systémů – GOST a ICAO. Rozdíl spočívá v odlišném popsání os a v jiném směru vertikální osy, viz obrázek 1.1. Mezi popisy příslušejícími k jednotlivým normám existuje přepočtení vztah, který je více popsán například v literatuře [1]. V této práci bude použita norma ICAO.



Obrázek 1.1 Rozdíl v osách mezi normami GOST a ICAO [1]

1.1.1 Letadlový souřadnicový systém

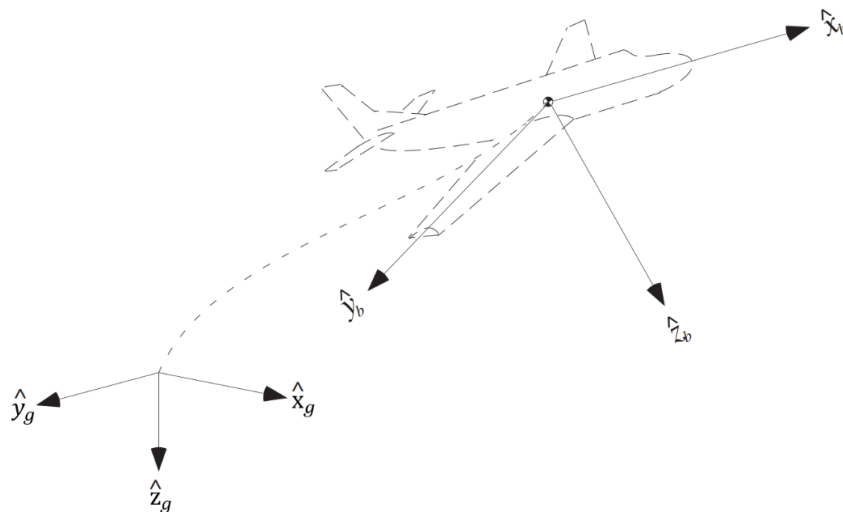
Počátek letadlového souřadnicového systému je umístěn do těžiště letadla. Osa x_b se překrývá s podélnou osou letadla. Osa y_b směřuje podél pravého křídla letadla. Osa z_b vychází z těžiště směrem dolů, kolmo na rovinu danou osami x_b a y_b , viz obrázek 1.2. [1][2]



Obrázek 1.2 Letadlový souřadnicový systém [2]

1.1.2 Zemský souřadnicový systém

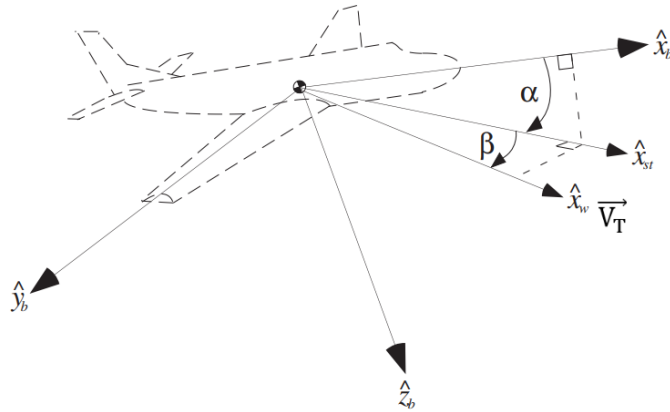
Počátek tohoto souřadnicového systému je pevně spjatý se zemským povrchem a může být umístěn kdekoliv na něm. V případě zápisu NED (North-East-Down) osa x_g směřuje na sever, osa y_g na východ a osa z_g dolů ve směru tíhového zrychlení g , viz obrázek 1.3. [1][2]



Obrázek 1.3 Vztah mezi letadlovým a zemským souřadnicovým systémem, upraveno [2]

1.1.3 Aerodynamický souřadnicový systém

Aerodynamický souřadnicový systém má svůj počátek v těžišti letadla, stejně jako systém letadlový. Osa x_w má stejný směr jako vektor celkové rychlosti V_T , kterou se letadlo (těžiště) pohybuje. Dále osa y_w směřuje na stranu pravého křídla a osa z_w dolů, kolmo na rovinu definovanou osami x_w a y_w , tedy obdobně jako u letadlového systému, viz obrázek 1.4. [1][2]



Obrázek 1.4 Vztah mezi aerodynamickým a letadlovým souřadnicovým systémem, upraveno [2]

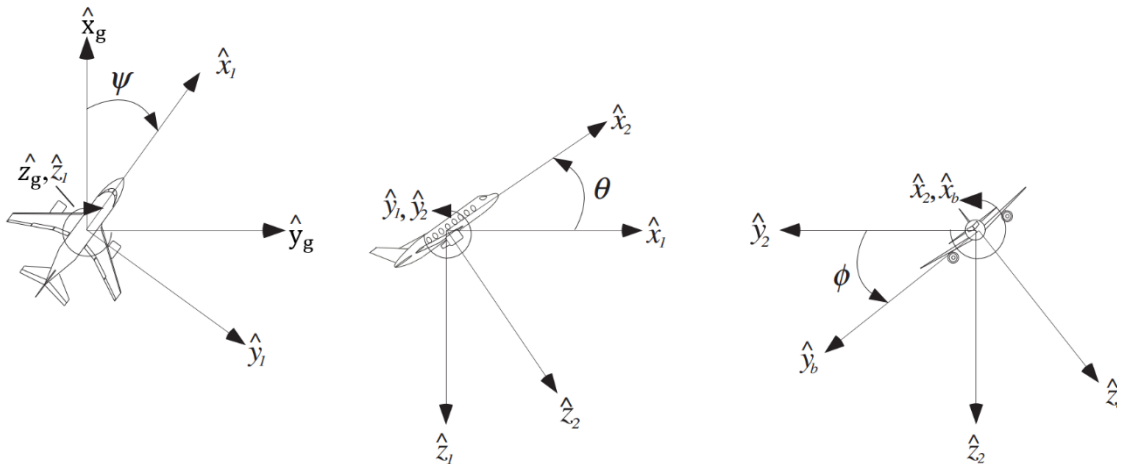
1.1.4 Vztahy mezi zemským a letadlovým souřadnicových systémem

Pro popis natočení letadla vzhledem k zemi jsou použity tzv. *polohové úhly*. Polohové úhly jsou úhly kursu letu ψ , podélného sklonu θ a příčného náklonu Φ . [1]

Jako první se počátek zemského souřadnicového systému přenesse do těžiště letadla (počátky systémů jsou shodné). Dále se souřadnicový systém otáčí podle svých jednotlivých os, viz obrázek 1.5. Tyto natočení jsou popsány pomocí následujících matic rotací (1.1) a (1.2). Pro zkrácení zápisu bude v následujícím textu použito zjednodušení, kde $\sin \theta = s_\theta$ a $\cos \theta = c_\theta$: [2][3][4]

$$\begin{pmatrix} x_b \\ y_b \\ z_b \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_\Phi & s_\Phi \\ 0 & -s_\Phi & c_\Phi \end{pmatrix} \begin{pmatrix} c_\theta & 0 & -s_\theta \\ 0 & 1 & 0 \\ s_\theta & 0 & c_\theta \end{pmatrix} \begin{pmatrix} c_\psi & s_\psi & 0 \\ -s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_g \\ y_g \\ z_g \end{pmatrix} \quad (1.1)$$

$$\begin{pmatrix} x_b \\ y_b \\ z_b \end{pmatrix} = \begin{pmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ -c_\Phi s_\psi + s_\Phi s_\theta c_\psi & c_\Phi c_\psi + s_\Phi s_\theta s_\psi & s_\Phi c_\theta \\ s_\Phi s_\psi + c_\Phi s_\theta c_\psi & -s_\Phi c_\psi + c_\Phi s_\theta s_\psi & c_\Phi c_\theta \end{pmatrix} \begin{pmatrix} x_g \\ y_g \\ z_g \end{pmatrix} \quad (1.2)$$



Obrázek 1.5 Postupné rotace ze zemského do letadlového souřadnicového systému, upraveno [2]

1.1.5 Vztahy mezi aerodynamickým a letadlovým souřadnicovým systémem

Na popis aerodynamiky letadla se používají tzv. *aerodynamické úhly*. Díky těmto úhlům lze popsat transformaci z letadlové do aerodynamického souřadnicového systému, a také vyjádřit, jak se vektor celkové rychlosti letadla \mathbf{V}_T promítne do vektorů rychlostí \mathbf{U} , \mathbf{V} a \mathbf{W} pro jednotlivé osy letadla (\mathbf{x}_b , \mathbf{y}_b a \mathbf{z}_b).

Úhel náběhu α je získán rotací kolem osy \mathbf{y}_b a nachází se mezi aerodynamickou osou \mathbf{x}_w a rovinou, na které leží osa letadla \mathbf{x}_b . Touto rotací je také získána poloha osy \mathbf{z}_w .

Následně je provedena další rotace, tentokrát kolem již zmíněné osy \mathbf{z}_w , tak aby osa \mathbf{x}_w kopírovala směr vektoru celkové rychlosti letadla \mathbf{V}_T . Úhel, o který se musel souřadnicový systém otočit, je úhel vybočení β . Postup rotací a vyznačené úhly lze vidět na obrázku 1.4 a jsou popsány rovnicemi (1.3) a (1.4). [1][2][3]

$$\begin{pmatrix} \mathbf{x}_w \\ \mathbf{y}_w \\ \mathbf{z}_w \end{pmatrix} = \begin{pmatrix} c_\alpha & 0 & s_\alpha \\ 0 & 1 & 0 \\ -s_\alpha & 0 & c_\alpha \end{pmatrix} \begin{pmatrix} c_\beta & s_\beta & 0 \\ -s_\beta & c_\beta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{x}_b \\ \mathbf{y}_b \\ \mathbf{z}_b \end{pmatrix} \quad (1.3)$$

$$\begin{pmatrix} \mathbf{x}_w \\ \mathbf{y}_w \\ \mathbf{z}_w \end{pmatrix} = \begin{pmatrix} c_\alpha c_\beta & s_\beta & c_\beta s_\alpha \\ -c_\alpha s_\beta & c_\beta & -s_\alpha s_\beta \\ -s_\alpha & 0 & c_\alpha \end{pmatrix} \begin{pmatrix} \mathbf{x}_b \\ \mathbf{y}_b \\ \mathbf{z}_b \end{pmatrix} \quad (1.4)$$

Za pomoci inverzního postupu je možné vyjádřit vektory rychlostí \mathbf{U} , \mathbf{V} a \mathbf{W} z vektoru celkové rychlosti letadla \mathbf{V}_T a úhlů α a β , jak je napsáno v rovnicích (1.5) až (1.8). [3]

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = \begin{pmatrix} c_\alpha c_\beta & -c_\alpha s_\beta & -s_\alpha \\ s_\beta & c_\beta & 0 \\ c_\beta s_\alpha & -s_\alpha s_\beta & c_\alpha \end{pmatrix} \begin{pmatrix} V_T \\ 0 \\ 0 \end{pmatrix} \quad (1.5)$$

$$U = V_T \cos(\alpha) \cos(\beta) \quad (1.6)$$

$$V = V_T \sin(\beta) \quad (1.7)$$

$$W = V_T \sin(\alpha) \cos(\beta) \quad (1.8)$$

1.2 Popis tvorby modelu podélného letu

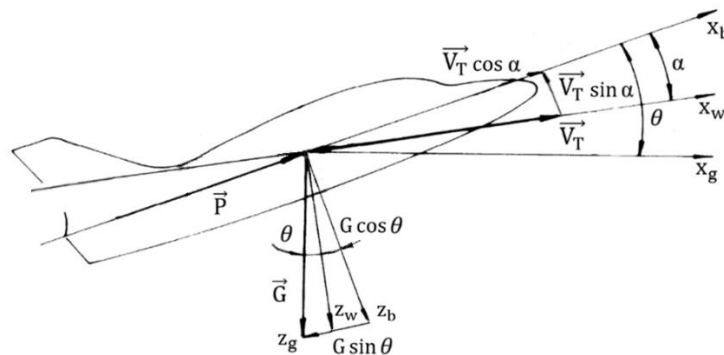
Při tvorbě modelu se vychází z pohybových rovnic, které zachycují závislosti sil a momentů v jednotlivých osách. Jsou zde vyjádřeny vlivy jednotlivých úhlů, rychlostí a dalších parametrů, které působí na stav letadla.

Rovnice, které vyplynou z fyzikálních zákonů, jsou nelineární. Jejich analýza a práce s nimi je možná, ale jednodušší je dané rovnice linearizovat kolem pracovního bodu a s přijatelnou chybou mít možnost využít metody pro práci s lineárními systémy.

Po linearizaci je možné transformovat rovnice do stavového popisu. Ten je tvořen čtyřmi maticemi, které reprezentují sledované stavy letadla, jejich vzájemné působení na sebe a vlivy vstupních signálů na stavy. [1][2][3]

1.2.1 Pohybové rovnice

Na začátku tvorby pohybových rovnic se počátky letadlového, zemského a aerodynamického souřadnicového systému položí do těžiště letadla. Dále se všechny systémy natočí tak, aby osy y byly ztotožněné. Následkem toho je rovina x_b - z_b souhlasná s rovinou x_w - z_w . To znamená, že ve stejné rovině leží i vektor rychlosti letadla V_T , viz obrázek 1.6. Toto uspořádání souřadnicových systémů je prvním předpokladem pro modelování podélného letu. [1]



Obrázek 1.6 Síly působící na letadlo a významné úhly mezi osami, upraveno [1]

Výsledné obecné pohybové rovnice jsou (1.9) až (1.14). Postup je podrobněji rozveden v literatuře [3].

$$m(\dot{U} + QW - RV + g \sin(\Theta)) = X \quad (1.9)$$

$$m(\dot{V} + UR - WP - g \cos(\Theta) \sin(\Phi)) = Y \quad (1.10)$$

$$m(\dot{W} + VP - QU - g \cos(\Theta) \cos(\Phi)) = Z \quad (1.11)$$

$$I_x \dot{P} - I_{xy}(\dot{R} + PQ) + (I_z - I_y)QR = L \quad (1.12)$$

$$I_y \dot{Q} + I_{xz}(P^2 - R^2) + (I_x - I_z)PR = M \quad (1.13)$$

$$I_z \dot{R} - I_{xz}\dot{P} + (I_y - I_x)PQ + I_{xz}QR = N \quad (1.14)$$

Pro úplné vyjádření všech souvislostí, které se na popisovaných silách projeví, je ještě nutné rovnice doplnit o závislosti sil a momentů na dalších parametrech letounu. Ty se popisují pomocí aerodynamických koeficientů zahrnujících např. koeficient aerodynamického odporu, vztlaku apod. V této práci jsou však uvedeny jen obecné rovnice, které jsou pro přibližné vykreslení problematiky dostatečné. [2][3]

1.2.2 Linearizace

Nabyté pohybové rovnice jsou nelineární a pro další zpracování je vhodné je linearizovat. Linearizace se provádí pomocí Taylorova rozvoje 1. řádu kolem pracovního bodu a metodou malých odchylek. Výsledkem tohoto postupu jsou stavy \mathbf{x} , jakožto součty ustálených hodnot \mathbf{x}_0 (určují počáteční stav systému) a odchylek od ustálených hodnot $\Delta\mathbf{x}$: [1][3]

$$\mathbf{x} = \mathbf{x}_0 + \Delta\mathbf{x} \quad (1.15)$$

Jelikož se předpokládá, že odchylky jsou malé, můžou se zanedbat jejich násobky a mocniny. Výsledné lineární rovnice jsou sice jednodušší, ale s přijatelnou přesností budou platit jen pro zvolené okolí pracovního bodu a pro krátké doby zkoumání. [1][2]

U modelování letadla je obvyklé, že lze oddělit podélný a stranový popis letu. Hlavním předpokladem je, že délka letadla je mnohem větší než jeho šířka a výška. Dále se předpokládá, že podélná rychlost (v ose \mathbf{x}_w) je podstatně větší než příčná (osa \mathbf{y}_w) a vertikální (osa \mathbf{z}_w). Nakonec se spoléhá na to, že odchylky veličin typických pro stranový pohyb (ΔV , ΔP , ΔR , $\Delta\theta$) jsou zanedbatelné. [3]

Při zkoumání ustáleného podélného letu se počítá s konstantní rychlostí U_0 , výškou h_0 , úhlem náběhu α_0 a hmotností m_0 . Veličiny, které jsou během zkoumání

podélného letu nulové, jsou V_0 , P_0 , R_0 , β_0 , Φ_0 a Ψ_0 . Ustálené hodnoty a odchylky sil podstatných pro podélný let jsou pak popsány rovnicemi (1.16) až (1.21). [3]

$$m(Q_0W_0 - R_0V_0 + g \sin(\Theta_0)) = X_0 \quad (1.16)$$

$$m(P_0V_0 - Q_0U_0 - g \cos(\Theta_0)\cos(\Phi_0)) = Z_0 \quad (1.17)$$

$$((P_0^2 - R_0^2)I_{xz} + (I_x - I_z)P_0R_0) = M_0 \quad (1.18)$$

$$m[\Delta\dot{U} + Q_0\Delta W + W_0\Delta Q + g \cos(\Theta_0)\Delta\Theta] = \Delta X \quad (1.19)$$

$$m[\Delta\dot{W} - U_0\Delta Q - Q_0\Delta U + g \sin(\Theta_0) \cos(\Phi_0)\Delta\Theta] = \Delta Z \quad (1.20)$$

$$I_y\Delta\dot{Q} = \Delta M \quad (1.21)$$

1.2.3 Stavový popis

V dnešní době je k dispozici výkonná výpočetní technika, což zjednodušuje výpočty pohybových rovnic. Jelikož počítače velmi dobře zvládají numerické počty s maticemi při řešení lineárních dynamických systémů, je výhodné pro tuto práci využít stavový popis. [4]

Systém popsaný soustavou lineárních diferenciálních rovnic lze přepsat do tzv. stavového popisu, kde jsou rovnice (1.22) a (1.23) stavovými rovnicemi systému. Stavový popis se pak skládá z matic konstant **A**, **B**, **C** a **D**, které popisují vazby mezi stavovými proměnnými, působení řídicích signálů na stavy a způsoby, jakými se stavy projeví na výstupu systému. Počet stavů udává řád systému. [4][5]

$$\Delta\dot{\mathbf{x}}(t) = \mathbf{A} \cdot \Delta\mathbf{x}(t) + \mathbf{B} \cdot \Delta\mathbf{u}(t) \quad (1.22)$$

$$\Delta\mathbf{y}(t) = \mathbf{C} \cdot \Delta\mathbf{x}(t) + \mathbf{D} \cdot \Delta\mathbf{u}(t) \quad (1.23)$$

Kde: **A** je stavová matice systému

B je matice řízení

C je matice výstupů

D je matice přímého vlivu vstupů na výstupy

$\Delta\mathbf{x}(t)$ jsou stavy systému

$\Delta\mathbf{u}(t)$ jsou řídicí signály

$\Delta\mathbf{y}(t)$ jsou výstupy systému

1.2.4 Transformace linearizovaných rovnic do stavového popisu

Pomocí dříve vyjádřených lineárních pohybových rovnic (1.16) až (1.21) lze získat koeficienty matic **A**, **B** a **C** pro žádaný stavový popis, který sleduje podstatné veličiny pro podélný let. Je použit 4-stavový model, kde stavy jsou podélná rychlost U , vertikální rychlost W , úhlová rychlost změny podélného sklonu Q ($\dot{\theta}$) a úhel podélného sklonu θ . Uvažované ovládací prvky jsou tah motoru δ_T a výchylka výškového kormidla δ_E . Vztah mezi linearizovanými silami a stavovými veličinami vychází z rovnic (1.9) až (1.14).

Tento postup je možný, když jsou známy přesné číselné parametry letadla, určující hodnoty konstant v maticích **A**, **B** a **C**. V případech, kdy známé nejsou, je možné dané koeficienty získat pomocí metod identifikace z naměřených dat, čímž se bude zabývat praktická část této práce. [1][2][3][6]

Obecný 4-stavový popis je vyjádřen rovnicí (1.24). Matice **D** je nulová a matice **C** je za předpokladu měřitelnosti odchylek na hlavní diagonále jednotková. Takže výsledné stavy se přímo přepíší na výstup systému. [3][4][5]

$$\begin{pmatrix} \Delta \dot{U} \\ \Delta \dot{W} \\ \Delta \dot{Q} \\ \Delta \dot{\theta} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \Delta U \\ \Delta W \\ \Delta Q \\ \Delta \theta \end{pmatrix} + \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \delta_T \\ \delta_E \end{pmatrix} \quad (1.24)$$

V určitých aplikacích, kdy je důležité znát výšku letu, se výška zavádí jako pátá stavová proměnná. První čtyři stavy jsou stejné jako v rovnici (1.24). Rovnice výšky je odvozena z rovnice pro sílu Z (1.11). [3]

$$\Delta a_z = \Delta \dot{W} - U_0 \Delta Q \quad (1.25)$$

$$\Delta \dot{h} = -\Delta a_z = U_0 \Delta Q - \Delta \dot{W} \quad (1.26)$$

Vztah se integruje za předpokladu $\Delta \dot{h}(0) = U_0 \Delta \theta(0) - \Delta W(0) = 0$, poté vychází:

$$\Delta \dot{h} = U_0 \Delta \theta - \Delta W = U_0 (\Delta \theta - \Delta \alpha) \quad (1.27)$$

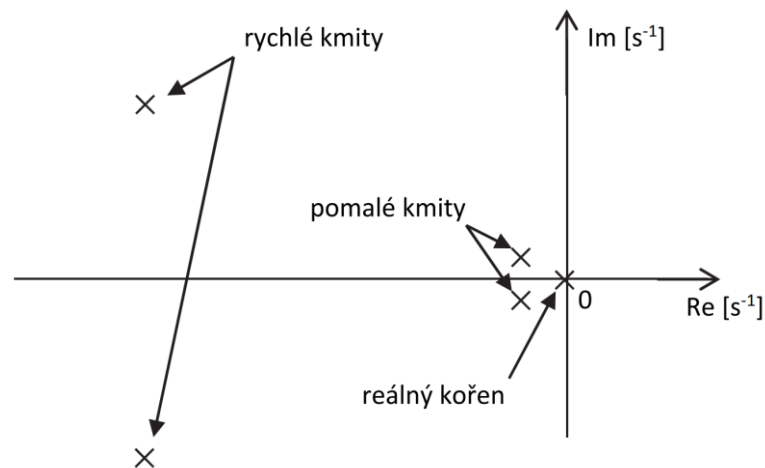
Výsledný 5-stavový model je popsán rovnicí (1.28). [3]

$$\begin{pmatrix} \Delta \dot{U} \\ \Delta \dot{W} \\ \Delta \dot{Q} \\ \Delta \dot{\theta} \\ \Delta \dot{h} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & 0 \\ a_{21} & a_{22} & a_{23} & a_{24} & 0 \\ a_{31} & a_{32} & a_{33} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & U_0 & 0 \end{pmatrix} \begin{pmatrix} \Delta U \\ \Delta W \\ \Delta Q \\ \Delta \theta \\ \Delta h \end{pmatrix} + \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \delta_T \\ \delta_E \end{pmatrix} \quad (1.28)$$

1.3 Dynamika podélného letu

Dynamika podélného letu ukazuje, jaká bude odezva systému/letadla na řídicí signály. Ty mohou být účelně vyvolány pilotem nebo způsobeny vnějšími vlivy, například poryvem větru či turbulencemi. Podoba odezvy a její dynamické vlastnosti lze spočítat pomocí charakteristického polynomu, který je společný pro všechny dílčí přenosy (vlivy vstupních signálů na stavy) nacházející se v modelu letadla.

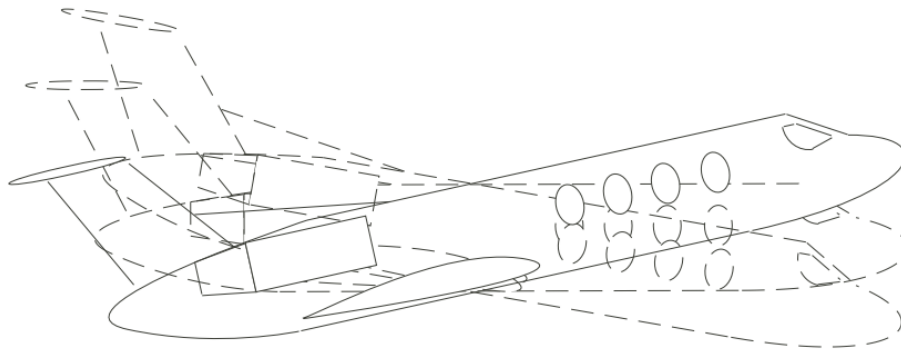
Pro podélný let jsou typické dva páry komplexně sdružených pólů. Pokud se do modelu zahrne i výška letu, přidá se i jeden reálný pól blízko počátku (integrátor). Pólpáry se projeví jako zakmitání na charakteristikách stavových veličin při změně řídicího signálu. Módy kmitání se dělí na rychlé a pomalé, kde rychlé kmity mají svůj pólpár dále od imaginární osy a pomalé kmity jej mají blíže, jak je vidět na obrázku 1.7. [1][3][4]



Obrázek 1.7 Typické rozložení pólů v komplexní rovině [6]

1.3.1 Rychlé kmity

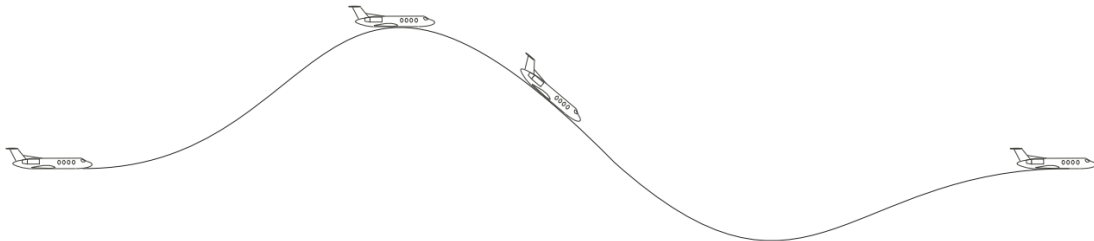
Rychlé kmity definuje naklání letadla kolem jeho těžiště, viz obrázek 1.8. Projevují se při narušení rovnovážného stavu na průbězích úhlu náběhu α , rychlosti změny podélného sklonu Q a podélném sklonu θ . Během tohoto jevu letadlo neustále letí konstantní aerodynamickou rychlostí. Obecně se perioda rychlých kmitů pohybuje řádově v jednotkách sekund a samotné kmitání bývá vcelku dobře tlumeno. [1][2][4]



Obrázek 1.8 Ukázka působení rychlých kmitů [2]

1.3.2 Pomalé kmity (fugoidální)

Pomalé kmity mají dominantnější vliv na podélný let než kmity rychlé. Periodu mají v řádu desítek až stovek vteřin a jsou špatně tlumené. Při letu se v letadle neustále přelévá kinetická energie. Při klesání se zvětšuje až do bodu, kdy její nárůst způsobí, že letadlo začne stoupat. Následně letadlo stoupá do té doby, než se kinetická energie „vybije“ a letadlo začne znovu klesat. Celý cyklus se opakuje a je postupně tlumen aerodynamickým odporem. Jelikož jsou tyto odpory navrženy tak, aby byly malé, je malé i tlumení. Letadlo tedy letí prostorem, zatímco se mu neustále mění úhel podélného sklonu θ a aerodynamická rychlost V_T , což pak ovlivňuje i výšku letu h , viz obrázek 1.9. [1][3][4]



Obrázek 1.9 Ukázka vlivu pomalých kmitů [2]

1.3.3 Integrátor

Integrátor reprezentuje zisk výšky, při změně výchylky výškového kormidla. Takže po skončení všech přechodových dějů se bude letadlo nacházet v nové výšce. [6]

2. NÁVRH A IDENTIFIKACE MODELU

Tato kapitola je zaměřena na návrh vhodné struktury modelu a následnou identifikaci jeho výpočetních koeficientů (matice **A**, **B** a **C**) z naměřených dat. K tomuto úkolu byla využita data naměřená na stacionárním leteckém simulátoru na UNOB (Univerzita obrany).

Stanovení struktury vychází z teoretické části a pro identifikaci bylo využito prostředí MATLAB a Simulink od společnosti MathWorks. MATLAB posloužil k identifikaci modelu a následnému grafickému zpracování výsledků a Simulink byl využit k simulacím s identifikovaným modelem.

2.1 Struktura modelu

Model je navržen tak, aby popisoval chování letadla pouze při podélném letu, během kterého je zkoumána i výška, v jaké se letadlo nachází. Jediné síly a momenty, které na letadlo působí, jsou síly \mathbf{X} (v ose \mathbf{x}_b) a \mathbf{Z} (v ose \mathbf{z}_b) a moment \mathbf{M} (kolem osy \mathbf{y}_b). Ostatní síly a momenty, typické pro stranový pohyb, jsou nulové. Obě křídla jsou vždy ve stejné výšce, letadlo nebude nikdy zatáčet a může se pouze naklánět nahoru a dolů, čímž dokáže měnit svou výšku. [3]

Jsou navrženy dva stavové popisy – jeden 4-stavový a druhý 5-stavový. Vybrané společné stavové veličiny, definující podélný let, jsou úhel náběhu α , aerodynamická rychlost v , rychlost změny podélného sklonu Q (pitch rate) a podélný sklon θ . U 5-stavového popisu se navíc uvažuje s výškou h , jejíž závislost bude nutné identifikovat z naměřených dat. V případě 4-stavového modelu bude závislost výšky h dopočítávána pomocí identifikovaných stavů a upravené rovnice (1.27). Výpočet a z něj vyplývající zapojení v Simulinku je blíže popsáno v kapitole 4.2.

Dva řídicí signály, použity k ovládní letadla, jsou výchylka výškového kormidla δ_E a výchylka tahu motoru δ_T . Výchylka výškového kormidla δ_E je ovládána pohybem kniplu. Při zanedbání dynamiky mezi těmito prvky lze prohlásit, že výchylka kormidla δ_E je přímo rovna výchylce kniplu.

Výsledná struktura 5-stavového modelu je popsána rovnicemi (2.1). Pro 4-stavový model jsou rovnice, až na přítomnost stavu výšky h , stejné. Hodnoty koeficientů matic **A**, **B** a **C** budou následně zjištěny pomocí identifikace z dostupných dat změřených na UNOB.

$$\begin{pmatrix} \Delta \dot{\alpha} \\ \Delta \dot{v} \\ \Delta \dot{Q} \\ \Delta \dot{\theta} \\ \Delta \dot{h} \end{pmatrix} = \mathbf{A} \cdot \begin{pmatrix} \Delta \alpha \\ \Delta v \\ \Delta Q \\ \Delta \theta \\ \Delta h \end{pmatrix} + \mathbf{B} \cdot \begin{pmatrix} \delta_E \\ \delta_T \end{pmatrix} \quad \begin{pmatrix} \Delta \alpha \\ \Delta v \\ \Delta Q \\ \Delta \theta \\ \Delta h \end{pmatrix} = \mathbf{C} \cdot \begin{pmatrix} \Delta \alpha \\ \Delta v \\ \Delta Q \\ \Delta \theta \\ \Delta h \end{pmatrix} \quad (2.1)$$

2.2 Identifikace parametrů modelu

K identifikaci bylo použito prostředí MATLAB, přesněji jeho rozšíření s názvem System Identification Toolbox.

2.2.1 Sběr dat

Data, ze kterých byl identifikován předpis modelu, byla naměřena na UNOB využitím dostupného stacionárního leteckého simulátoru, viz obrázek 2.1. Ten byl ovládán elektronicky řízeným potenciometrem, který sloužil jako joystick, a vybaven softwarem X-PLANE 10, což je poloprofesionální letecký simulátor. Tento software umožňuje sběr provozních a letových dat s frekvencí 20 Hz. [7]



Obrázek 2.1 Letecký simulátor na UNOB

Využitím uvedeného simulátoru tak byla změřena odezva letounu na impulzovou změnu výchylky kniplu. Data byla získána za definovaných podmínek. Letoun byl uveden do ustáleného stavu a výchozí podmínky při měření byly:

- Letoun: King Air C90B,
- výchozí výška letu: $h_0 = 2900$ ft,
- výchozí rychlost letu: $v_0 = 170$ mph,
- Machovo číslo: $M_0 = 0.23$,
- počáteční tah motoru $\delta_{T0} = 600$ lb.
- Všechny úhly a jejich derivace byly ve výchozím stavu přibližně nulové.

Naměřená data obsahují sedm průběhů, kde každý průběh odpovídá jedné stavové nebo řídicí veličině v navržených modelech (2.1). Data dále zahrnují samostatný časový vektor, který obsahuje časy měření.

2.2.2 Postup identifikace

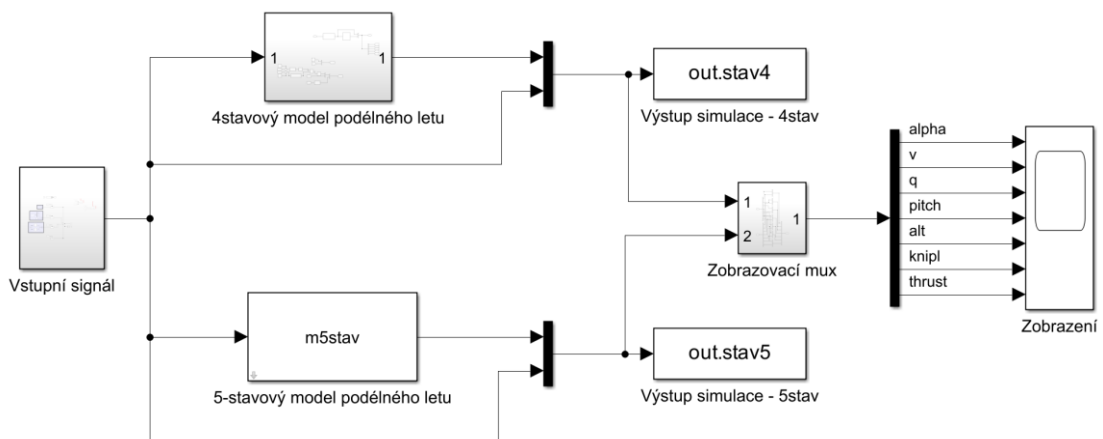
Pro samotné nalezení stavového popisu z naměřených dat byl využit příkaz *ssest* (může být také použito rozhraní *systemIdentification*). Identifikace byla provedena jak pro 5-stavový, tak i pro 4-stavový model.

Funkce *ssest* požadovala data, ze kterých má daný model odvodit. To bylo provedeno převodem dat do formátu *iddata*. U tohoto datového typu bylo nutné specifikovat vzorkovací periodu a také určit, která část dat reprezentuje řídicí veličiny, a která stavové. Dále bylo u *ssest* nutné specifikovat řád výstupního modelu. V případě této práce jde o 5. nebo 4. řád. Nakonec byl ještě přidán parametr *opt*, který v sobě nese nastavení samotné identifikace.

Proměnná *opt* je typu objekt *ssestOptions*. Objekt *ssestOptions* v sobě má uložené veškeré nastavení identifikace a jednotlivé parametry se mění pomocí tečkové notace a přepsání vnitřní proměnné objektu. V tomto případě byl změněn počáteční stav estimace na nulový, a také bylo nastaveno vynucení stabilního modelu. Použité příkazy, konkrétně pro získání 5-stavového modelu, jsou: [8]

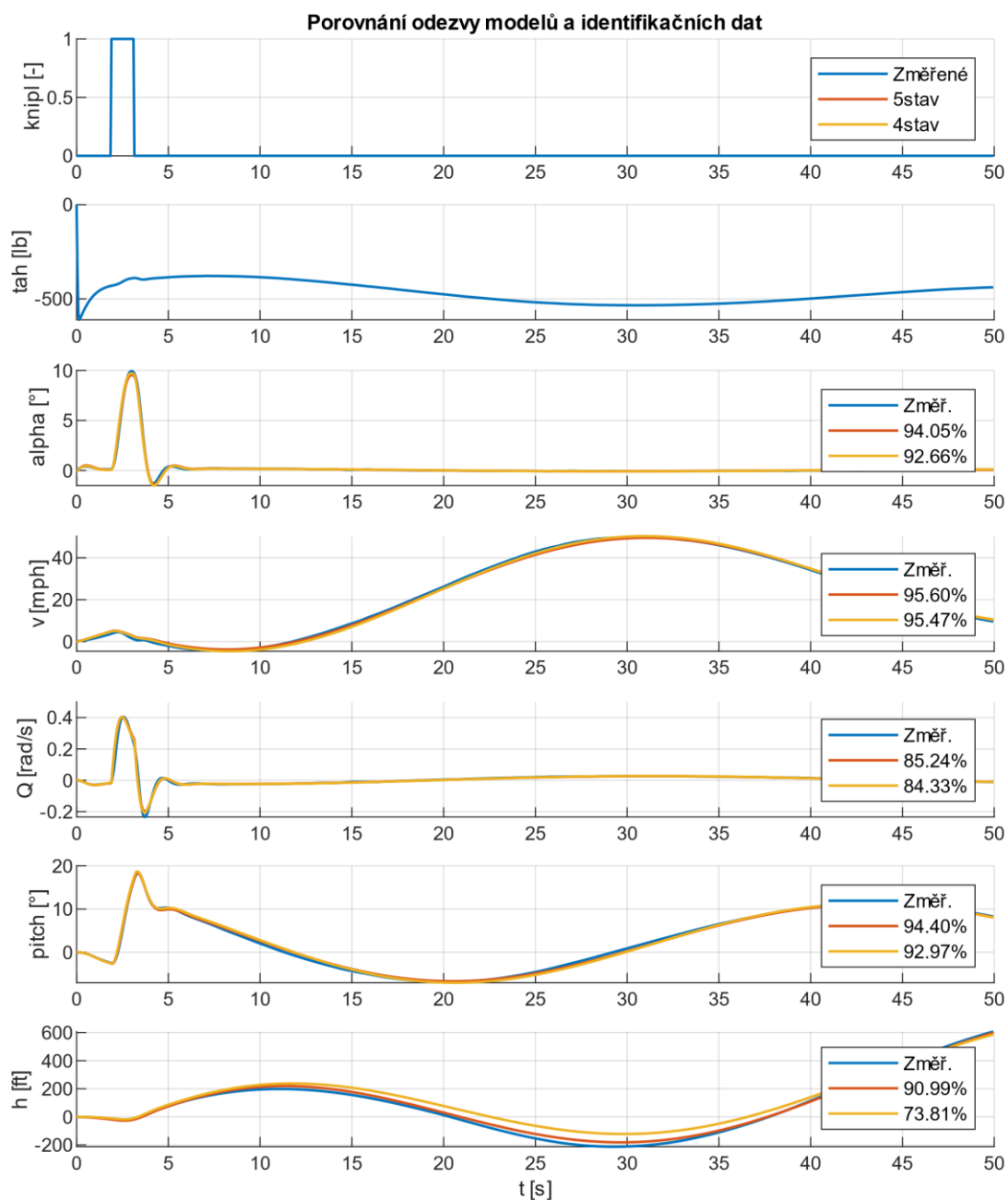
```
opt.InitialState = 'zero';  
opt.EnforceStability = true;  
  
model = ssest(data, 5, "OutputName", ["alpha" "v" "Q" "pitch" "h"],  
"InputName", ["knipl" "tah"], opt);
```

Výsledkem funkce *ssest* jsou modely, které reprezentují vliv řídicích veličin na stavové veličiny, a také vzájemné závislosti mezi stavovými veličinami – stavové popisy. Pro ověření, zda byly modely dobře identifikovány, byla odsimulována a zobrazena jejich odezva na řídicí veličiny použité při měření na UNOB. Odsimulovaná data byla následně porovnána s původně naměřenými na UNOB. Simulace byla realizována v prostředí Simulink pomocí zapojení na obrázku 2.2.



Obrázek 2.2 Simulační obvod použitý k porovnání

Na obrázku 2.3 je zobrazeno porovnání mezi průběhy identifikovaných modelů a změřenými daty. Z jednotlivých grafů a přesnosti proložení fit, získané pomocí funkce *compare*, lze vidět, že aproximace naměřených dat využitím identifikovaného modelu je poměrně přesná, což do jisté míry odráží kvalitu identifikovaného modelu. [8]



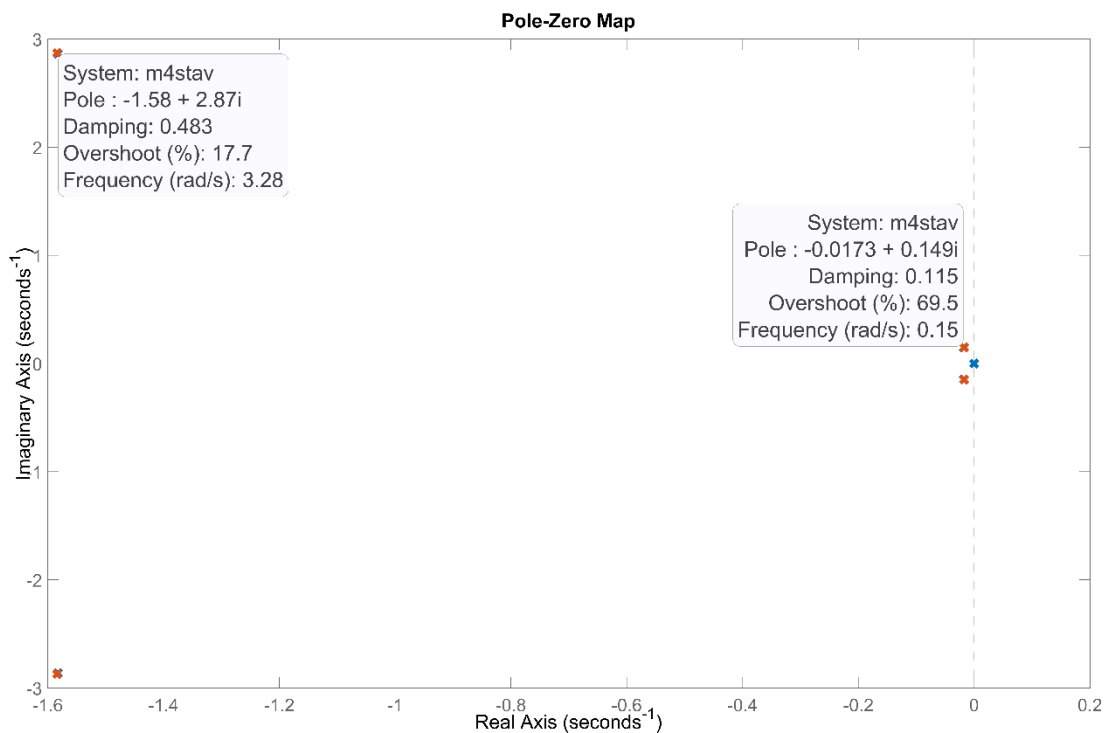
Obrázek 2.3 Porovnání odezvy modelů a identifikačních dat z UNOB

2.3 Dynamické vlastnosti modelu

U nalezených modelů je možné hlouběji prozkoumat jejich dynamické vlastnosti, které definují chování letadla při letu. Jednou z typických vlastností všech modelů podélného letu, tedy nezávisle na typu modelovaného letadla, je kmitání stavových veličin při změně řídicího signálu. [1][2][6]

Bude-li uvažován let, kde se může měnit výška a nejsou přítomny žádné vnější vlivy, např. turbulence nebo poryvy větru, tak se dynamické vlastnosti dají vyčíst z kořenů charakteristického polynomu identifikovaného systému. [1][2]

Pro zjištění rozmístění těchto kořenů – pólů systému – pro jednotlivé modely byl použit příkaz *pzmap* v prostředí MATLAB, jehož výsledek je zobrazen na obrázku 2.4. Z něho lze vidět, že pro oba modely pozice pólů odpovídají teoretickému předpokladu. Pro oba modely jsou zde dva komplexně sdružené pólpáry, ukazující na rychlé a pomalé kmity, a u 5-stavového modelu je navíc jeden pól tak blízko počátku, že se při delší zkoumaného jevu dá považovat za integrátor ($2,98 \cdot 10^{-7} \approx 0$).



Obrázek 2.4 Vykreslení pólů pomocí příkazu *pzmap*

Pólové dvojice a parametry, definující pomalé a rychlé kmity, byly sepsány do tabulky 2.1. Z tabulky je vidět, že póly odpovídající čtyřem společným stavům jsou v podstatě totožné.

Tabulka 2.1 Póly systémů a jejich parametry

MODEL	KMITY	PÓLY	ω [rad/s]	T [s]	ε [-]	γ [s]
5-STAV	pomalé	$-0,0178 \pm j0,1471$	0,15	42,17	0,12	38,93
	rychlé	$-1,58 \pm j2,86$	3,27	1,92	0,48	0,44
	integrátor	$2,98 \cdot 10^{-7}$	-	-	-	-
4-STAV	pomalé	$-0,0173 \pm j0,149$	0,15	42,17	0,115	40,06
	rychlé	$-1,58 \pm j2,87$	3,28	1,92	0,48	0,44

Kde, ω je frekvence kmitů,

T je perioda kmitů,

ε je poměrné tlumení,

γ je poločas utlumení (za jak dlouho se amplituda kmitů zmenší na polovinu). [9]

3. VALIDACE MODELU

Po získání identifikovaného modelu je důležité ověřit jeho funkčnost pro jiné signály než jen pro ty, pomocí kterých byl identifikován – tomuto procesu se říká validace. Pro samotnou validaci bylo nutné naměřit nové průběhy podélného letu. K tomu byl stejně jak v případě identifikace využit letecký simulátor na UNOB, viz kapitola 2.2.1.

Pro možnost validace modelu byly aplikovány různé vstupní signály popsané níže. Pro získání požadovaného tvaru signálů a opakovatelnosti měření bylo použito automatizované generování vstupních signálů (reprezentujících výchylky kniplu) pomocí desky Arduino Leonardo. Odezva na tyto řídicí signály se měřila a zaznamenávala do textového souboru ve formátu CSV (Comma-Separated Values). Změřená data pak byla zpracována a porovnána s odezvou získanou z identifikovaných modelů.

3.1 Použité signály

Navržená struktura použitých stavových popisů uvažuje dva ovládací prvky, a to tah motoru δ_T a výchylku výškového kormidla δ_E . Je ale třeba vzít v potaz, že tah motoru je pevně svázán se stavy letadla, a tak mu nejde určit jiný externí průběh. Oba modely také počítají s počátečním tahem motoru $\delta_{T0} = 600$ lb a pro jiné hodnoty se nedá počítat s jejich platností. Pro ilustraci je v pozdější kapitole 3.4.3 ukázán vliv různých hodnot tahu motoru na podélný let. S touto úvahou se tím pádem bude určovat pouze tvar signálu pro výchylku výškového kormidla δ_E .

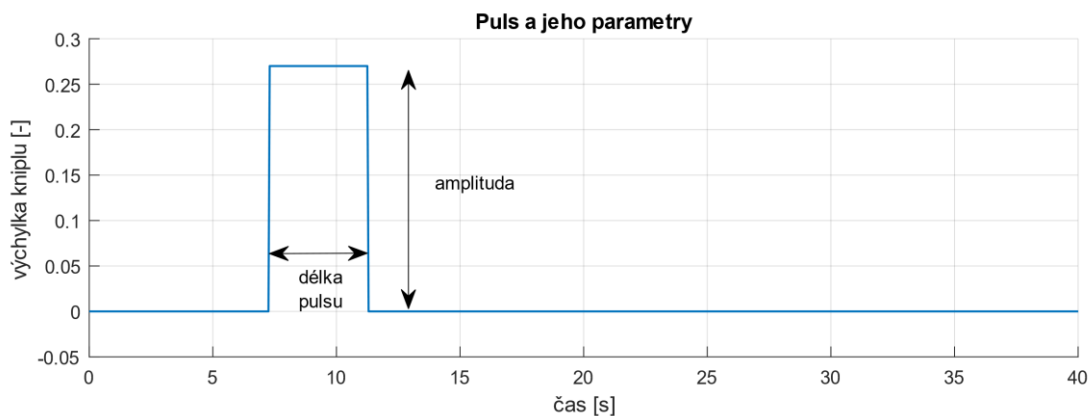
Při tvorbě signálů je důležité, aby jejich projev byl měřitelný, ale zároveň, aby nezpůsobil příliš velké odchylky v měřených stavech, čímž by opustil oblast pracovního bodu, pro který jsou modely navrženy. Díky tomu, že je měření prováděno na simulátoru, kde není žádný šum, odpadá problém s odstupem signál-šum, kvůli kterému by malé signály nemohly být použity.

Zvolené signály jsou puls s různými amplitudami, rampa a PRBS.

3.1.1 Puls

Puls je neperiodický signál o určité amplitudě a šířce, viz obrázek 3.1. Aplikací pulsu na systém se získá impulsová charakteristika systému, která společně se skokovou charakteristikou popisuje chování systému. [10]

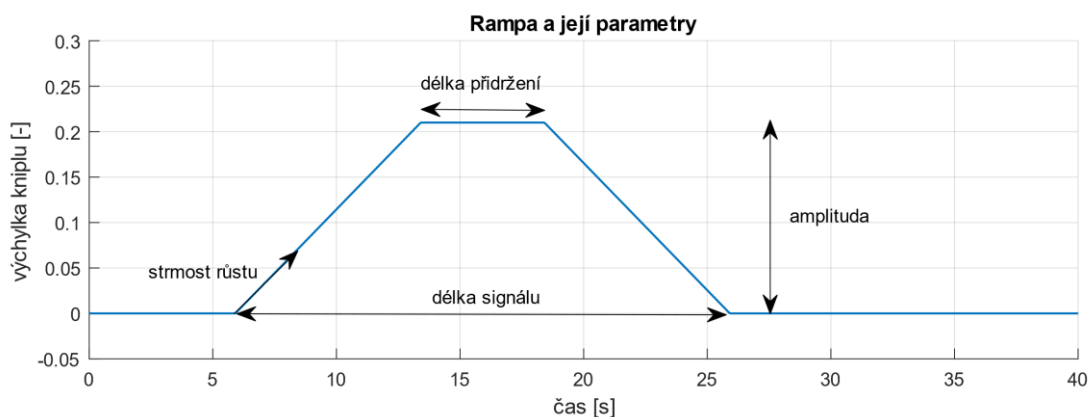
Puls byl také aplikován jako vstupní signál dat použitých na identifikaci.



Obrázek 3.1 Ukázka pulsu a jeho parametrů

3.1.2 Rampa

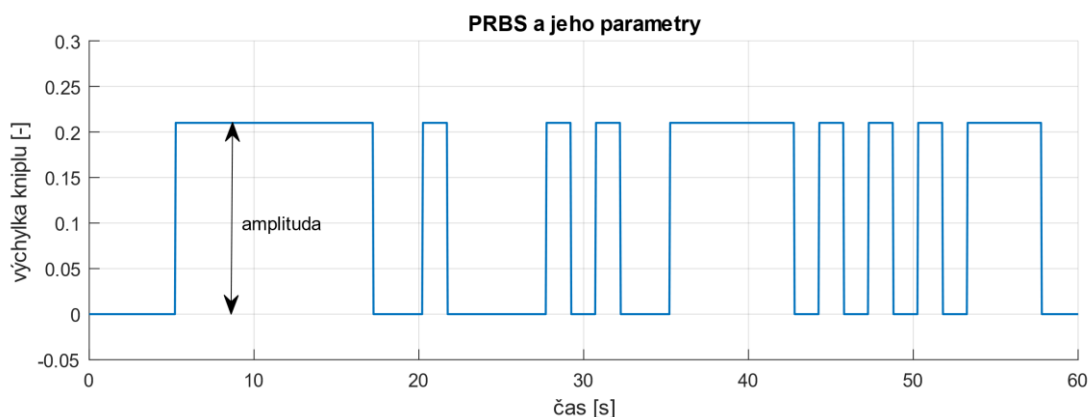
Rampa použitá v této práci začíná lineárním růstem pokračujícím až do ustálené hodnoty, ve které určitou dobu setrvává, a následně začne lineárně klesat zpět na počáteční hodnotu, viz obrázek 3.2.



Obrázek 3.2 Ukázka rampy a jejích parametrů

3.1.3 PRBS (Pseudo-Random Binary Signal)

Poslední použitý signál je tzv. pseudo-náhodná binární posloupnost, která je zobrazena na obrázku 3.3. Je to sekvence nul a jedniček, jejíž posloupnost určuje posuvný N-bitový registr (Linear-Feedback-Shift Register – LFSR). Existují dvě základní uspořádání LFSR – Fibonacci a Galois. V této práci je použit Fibonacci LFSR a níže je jeho stručný popis.

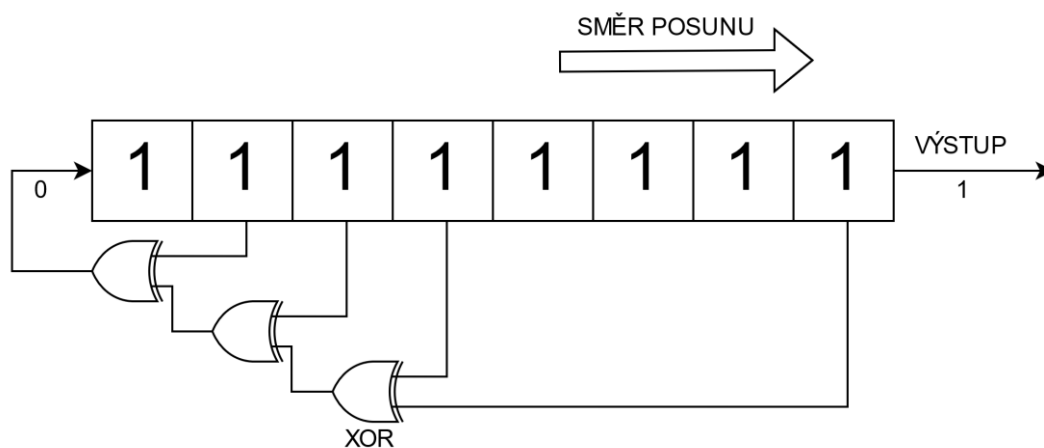


Obrázek 3.3 Ukázka PRBS a jeho parametrů

Jak se registr posouvá, tak se hodnota, kterou by „vysunul“, bere jako výstupní. Nová hodnota, která se do registru „nasune“, se spočítá z vybraných bitů aplikací operace XOR.

Délka sekvence, která se bude periodicky opakovat, je dána výběrem bitů, ze kterých se „nasunovaná“ hodnota počítá. Při správném výběru lze dosáhnout délky posloupnosti až $2^N - 1$ (N je počet bitů LFSR). Počáteční inicializační hodnota registru musí být nenulová, jinak operace XOR nic nezmění. [11][12]

Pro tuto práci byl použit 8-bitový registr s maximální délkou posloupnosti 255 čísel, viz obrázek 3.4. Inicializační hodnota registru byla 0xFF. Doba mezi posuny registru byla jedna vteřina.



Obrázek 3.4 Použitá struktura 8-bitového Fibonacci LFSR

3.2 Tvorba aplikace pro generování vstupních signálů

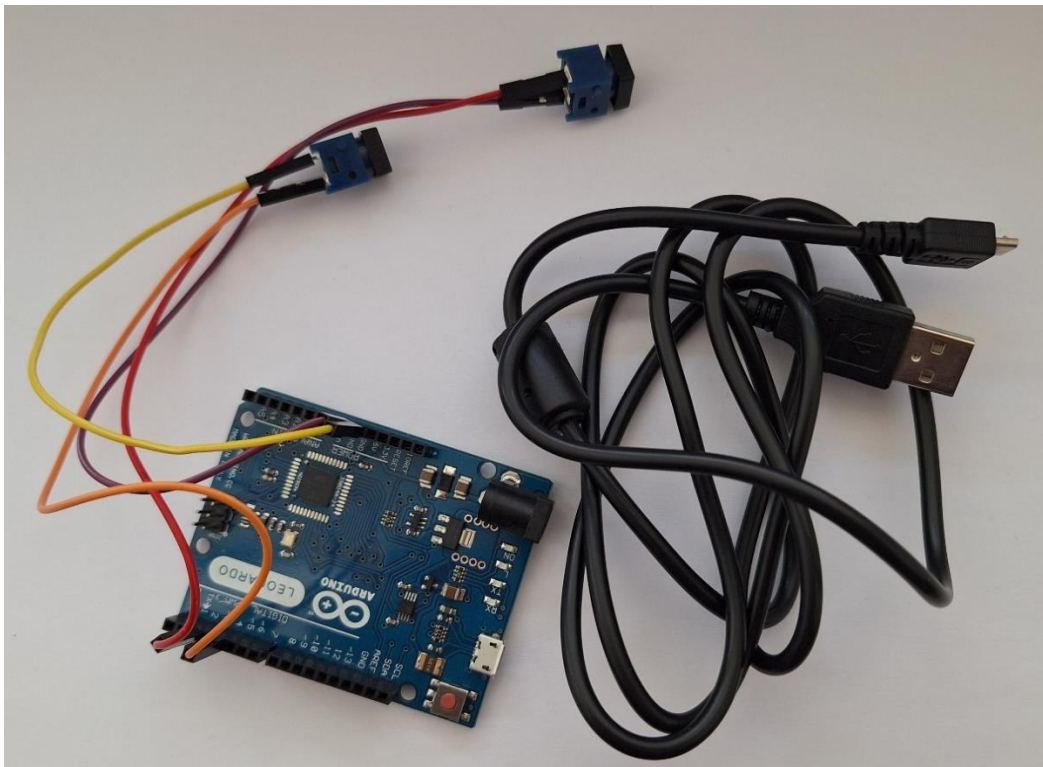
Ovládání simulátoru X-PLANE 10 na UNOB je realizováno pomocí různých periférií jako jsou joystick, letecké pedály, plynové pedály apod. Všechny periférie jsou pomocí mapování, přímo v softwaru X-PLANE 10, přiřazeny jednotlivým ovládacím prvkům na letadle. Ovládání výchylky výškového kormidla δ_E je ve výchozím stavu realizováno jednou osou hlavního kniplu (joysticku). [7]

Pro generování signálů byla vybrána deska Arduino Leonardo, jelikož programování v Arduino je dobře přístupné a také protože přímo na desku Leonardo existuje knihovna, která dovolí této desce emulovat funkčnost joysticku. V důsledku toho je možné ovládání výškového kormidla předat Arduino a určovat tak jeho průběh.

3.2.1 Arduino Leonardo

Arduino Leonardo, ukázáno na obrázku 3.5, je deska s mikrokontrolérem, digitálními I/O piny, 16 MHz krystalovým oscilátorem, micro-USB připojením a dalšími funkcionalitami. Samotná implementace programu probíhá v prostředí Arduino IDE.

V prostředí Arduino IDE se programuje v Arduino kóde, který je založený na jazyce C++. Jednoduše se v něm spravují přidané knihovny a dá se přes něj rovnou nahrávat kód do připojených desek. Obsahuje také vestavěnou funkci Serial Plotteru a Serial Monitoru, díky kterým lze jednoduše kontrolovat běh programu. [13]



Obrázek 3.5 Použitá deska Arduino Leonardo

3.2.2 Joystick knihovna

Tato knihovna umožňuje Arduino Leonardo emulovat funkcionalitu joysticku se všemi možnostmi, které joystick může mít. Lze ho následně najít v připojených perifériích a simulátor X-PLANE 10 je schopný na něho namapovat ovládací prvky letadla.

Knihovna obsahuje několik funkcí, umožňujících programovat vychýlení joysticku v X, Y, Z osách, velké množství tlačítek atd. [14]

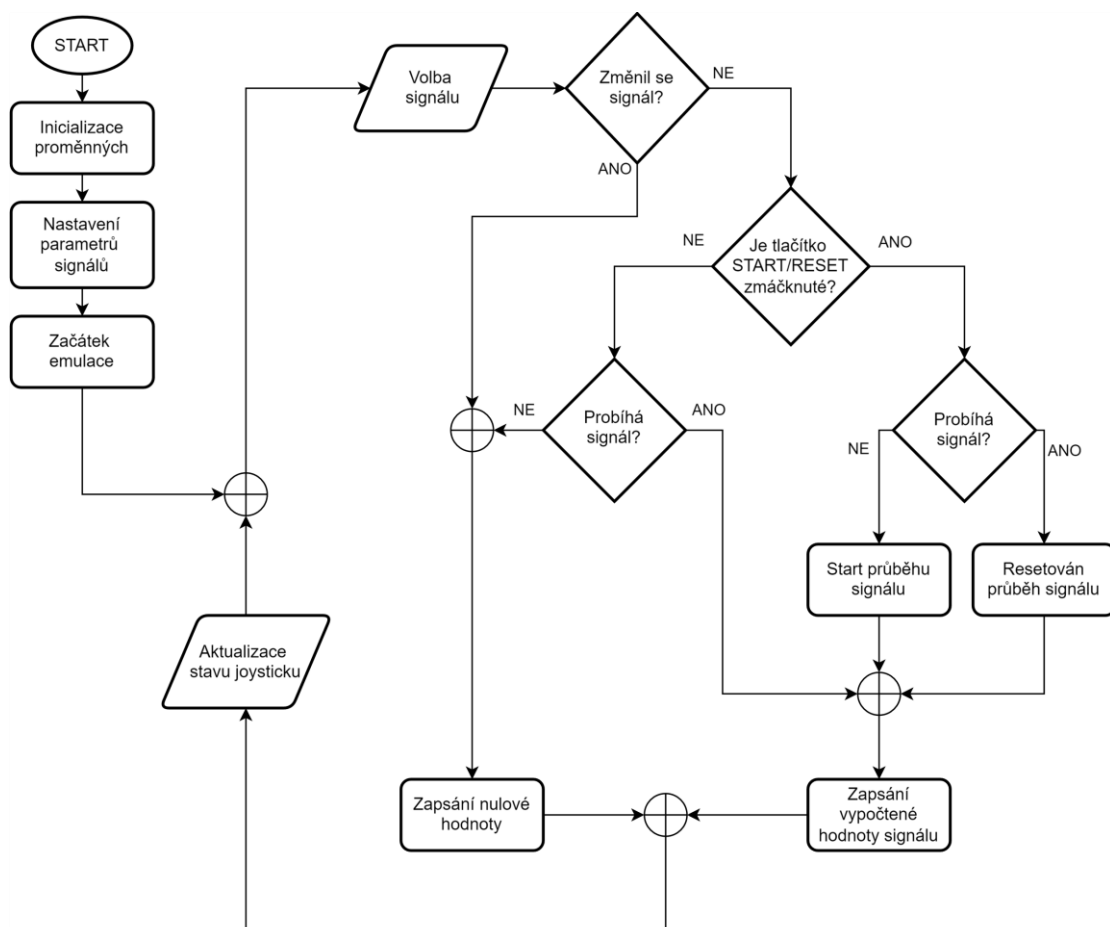
3.2.3 Vlastní program

Na začátku programu je provedena inicializace všech potřebných proměnných. Dále se nastaví, co samotný joystick bude umět – v případě podélného letu postačí pohyb v Y ose – a příkaz pro zapnutí emulace joysticku.

V hlavní smyčce programu je řešeno, jak za běhu programu mezi signály přepínat, odstartovat průběh signálu, nebo signál zastavit. K tomu slouží dvě tlačítka připojená na digitální I/O piny desky. Podle zvoleného tvaru signálu se pak počítá jeho průběh a vypočtené hodnoty se následně propisují jako stav emulovaného joysticku, který by v případě reálného joysticku odpovídal výchylce jeho polohy.

Pro každý signál zvolený v kapitole 3.1 je naprogramována funkce, která daný průběh počítá. Výstupem každé funkce je parametrizovatelný průběh vybraného signálu (lze měnit amplituda, doba trvání, strmost růstu apod.). Samotné funkce lze skládat za sebe a tím tvořit součty dvou průběhů.

Pro lepší představu je na obrázku 3.6 nakreslený vývojový diagram reprezentující chod programu.



Obrázek 3.6 Vývojový diagram generování signálů v Arduino

3.3 Měření a zpracování naměřených dat

Pro měření byl využit stejný simulátor jako pro první soubor dat, ze kterých byly identifikovány modely. Jedná se o stacionární letecký simulátor vybavený softwarem X-PLANE 10 a jeho stručný popis lze nalézt v kapitole 2.2.1. Ke generování vstupních signálů posloužila již výše popsaná deska Arduino Leonardo.

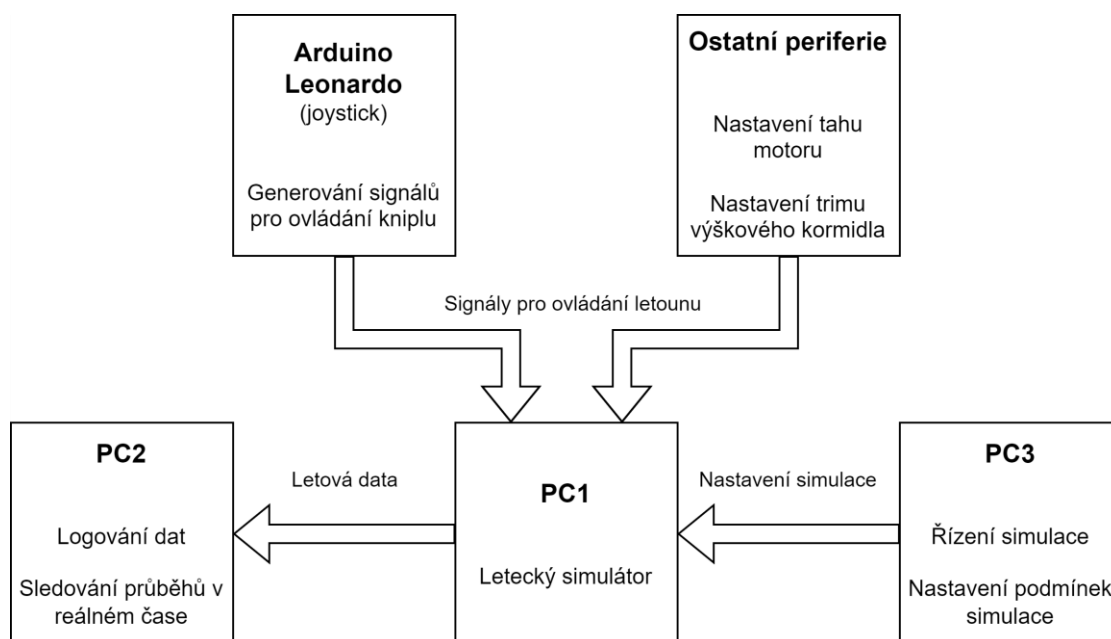
Z naměřených dat v CSV formátu byly extrahovány průběhy podstatných veličin pro porovnání s odezvou identifikovaných modelů. Následovalo zpracování těchto průběhů do podoby, ve které mohly být využity pro verifikaci modelů. Pro veškeré zpracování dat bylo použito prostředí MATLAB, společně se Simulinkem.

3.3.1 Naměření dat na simulátoru X-PLANE 10

Na prvním počítači byl spuštěn letecký simulátor X-PLANE 10. Ke stejnému počítači byla přes USB připojena Arduino deska, která obstarávala generování signálů, které během simulace ovládaly knipl.

Počítač se simulátorem odesílal naměřená data přes komunikační modul do druhého počítače, na kterém probíhalo logování dat do textového souboru, jehož formát lze vidět na obrázku 3.8. Zde bylo také možné sledovat v reálném čase veličiny právě měřeného průběhu.

Byl zde i třetí počítač sloužící jako řídicí stanice. Díky ní bylo možné upravovat nastavení podmínek simulace ve smyslu chování počasí, vlivu podnebí, oblačnosti apod. V případě této práce byly všechny externí vlivy počasí vypnuty, tak aby bylo dosaženo co nejideálnějších podmínek, a to z důvodu opakovatelnosti měření. Blokové schéma celé měřicí stanice je zobrazeno níže, na obrázku 3.7.



Obrázek 3.7 Blokové schéma měřicí stanice

Během samotného měření bylo velmi důležité uvést letoun do tzv. vyváženého letu, nebo se mu aspoň co nejvíce přiblížit. Jedná se o stav, při kterém letoun nestoupá ani neklesá a není zasahováno do jeho ovládání. V této situaci jsou úhly významné pro podélný let a jejich úhlové rychlosti velmi blízké nule, což je velmi podstatné pro samotnou linearizaci modelu a porovnatelnost s navrhnutými modely.

Navození vyváženého letu bylo uskutečněno pomocí změny trimu výškového kormidla. Změna trimu způsobí změnu v aerodynamických silách, které na kormidlo působí, a tím pádem tak lze ovlivnit i samotné stoupání letadla. Jelikož je tato změna v podstatě akčním zásahem, bylo třeba před aplikací signálu z Arduina počkat na odeznění přechodových jevů, které by jinak změřenou odezvu zkreslily. Změna trimu měla za následek změnu vlivu výchylky výškového kormidla na stavy letadla, v důsledku čehož musely být patřičně upraveny konstanty v maticích **B** u obou identifikovaných modelů. [15]

Samotné měření pro jeden tvar signálu začalo zapnutím simulátoru, čímž se založil nový logovací soubor. Dále se letadlo pomocí trimu uvedlo do stavu vyváženého letu, načež byl Arduino vydán příkaz ke generování signálu. Po odměření dostatečně dlouhého časového vzorku byly pozice a stav letadla vyresetovány. Tento proces byl pro každý typ signálu několikrát zopakován. Výsledkem byl jeden datový soubor s určitým počtem naměřených odezev pro jeden tvar signálu, ve kterém je uloženo velké množství veličin týkajících se letu letadla, přičemž důležité z nich budou vybrány procesem extrakce.

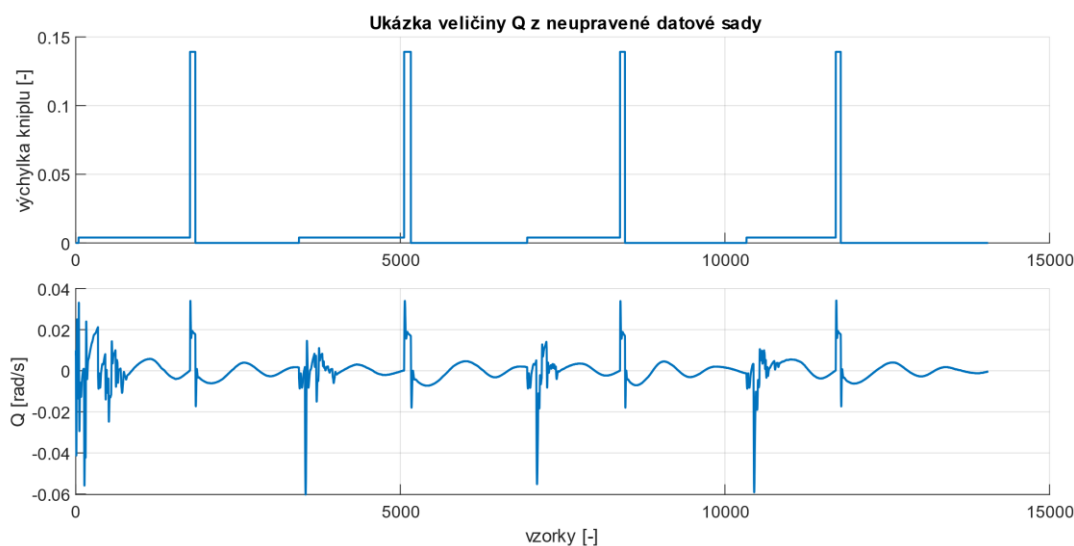
	A	B	C	D	E	F	G	H	I	J	K	L	M
	real_time	Var...	totl_time	Var...	missn_time	Var...	timer_time	Var...	zulu_time	Var...	local_time	Var...	hobbs_time
	Number	▼C...▼Number	▼C...▼Number	▼C...▼Number	▼C...▼Number	▼C...▼Number	▼C...▼Number	▼C...▼Number	▼C...▼Number	▼C...▼Number	▼C...▼Number	▼C...▼Number	▼C...▼Number
1	_real_time		_totl_time		missn_time		timer_time		zulu_time		local_time		hobbs_time
2	0.51935		0.05025		0.05025		0.00000		10.62050		11.62050		73.89634
3	2.37487		0.10050		0.10050		0.00000		10.62051		11.62051		73.89635
4	3.46083		0.15075		0.15075		0.00000		10.62053		11.62053		73.89637
5	3.49906		0.18898		0.18898		0.00000		10.62054		11.62054		73.89639
6	3.53685		0.22678		0.22678		0.00000		10.62055		11.62055		73.89639
7	3.57490		0.26483		0.26483		0.00000		10.62056		11.62056		73.89640
8	3.61519		0.30511		0.30511		0.00000		10.62057		11.62057		73.89641
9	3.65456		0.34449		0.34449		0.00000		10.62058		11.62058		73.89642
10	3.69218		0.38211		0.38211		0.00000		10.62059		11.62059		73.89642
11	3.73018		0.42011		0.42011		0.00000		10.62060		11.62060		73.89643

Obrázek 3.8 Formát naměřených dat

3.3.2 Extrahování z naměřených dat

V první řadě bylo nutné nalogovaný soubor importovat do MATLABu, aby se s ním mohlo dále pracovat. V importovaných datech, která jsou v podobě tabulky, bylo třeba vyhledat průběhy veličin odpovídajících stavovým veličinám identifikovaných modelů – úhel náběhu α , rychlost letu v , úhel podélného sklonu θ , rychlost změny úhlu podélného sklonu Q a výšku letu h –, řídicí signály – výchylka kniplu δ_E a tah motoru δ_T – a nakonec časový vektor s okamžiky měření.

Pro tuto úlohu byl napsán jednoduchý skript, který v importovaném souboru požadované veličiny hledá. Pro každý datový soubor pak vznikne tabulka, ve které jsou pouze hodnoty odpovídající stavům, řídicím signálům nebo času. Na obrázku 3.9 je vykreslená podoba části extrahovaných dat.

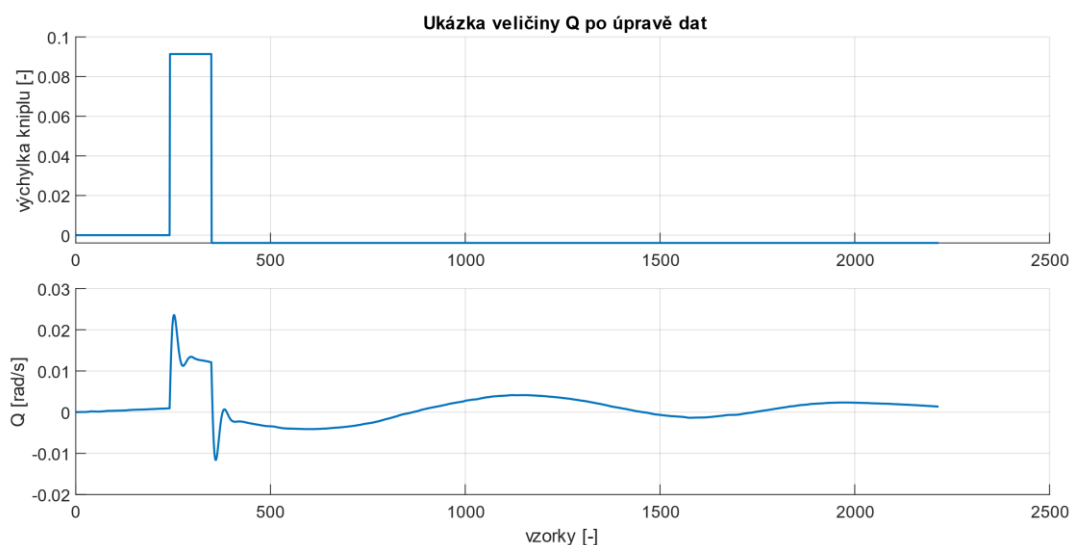


Obrázek 3.9 Ukázka průběhu Q z neupravené datové sady

3.3.3 Zpracování extrahovaných dat

Všechny datové soubory bylo nutné po extrakci dále upravit. První úpravou bylo počáteční ořezání. Hodnoty na začátku a na konci jednotlivých měření nebyly validní, a tak musely být odstraněny. Pro započítí nového měření se musel simulátor restartovat a při jeho inicializaci pak vznikaly chyby na počátcích naměřených datových sad. Chyby na konci měření obdobně souvisely s restartováním simulátoru.

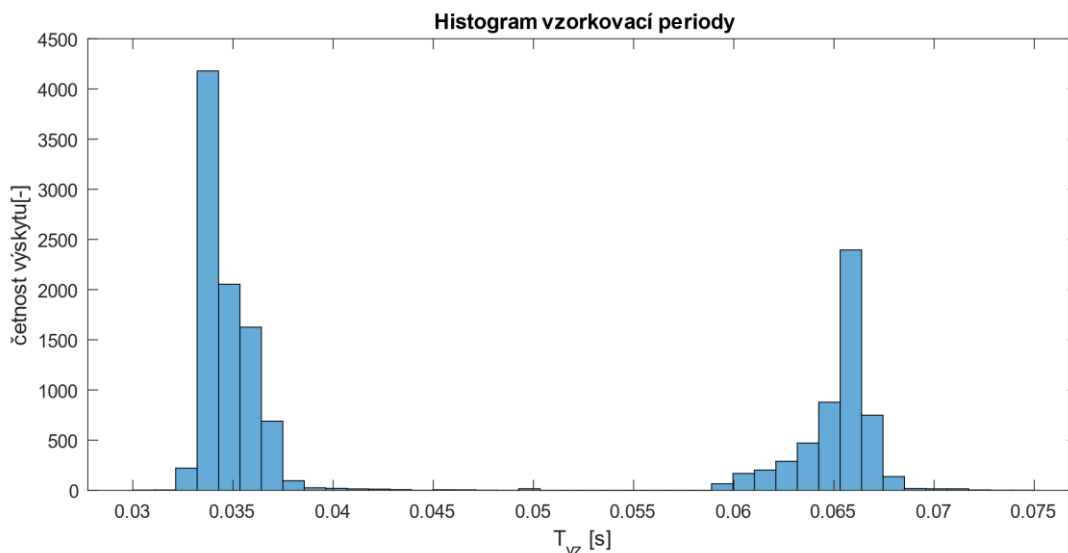
Po ořezání byla datová sada – jeden dlouhý graf – rozdělena na dílčí průběhy. Každý dílčí průběh byl následně upraven tak, aby jednotlivé signály začínaly v nule, viz obrázek 3.10. Tyto průběhy pak byly připraveny na porovnávání s odezvami modelů.



Obrázek 3.10 Ukázka průběhu Q po úpravě dat

3.3.4 Problém se vzorkováním

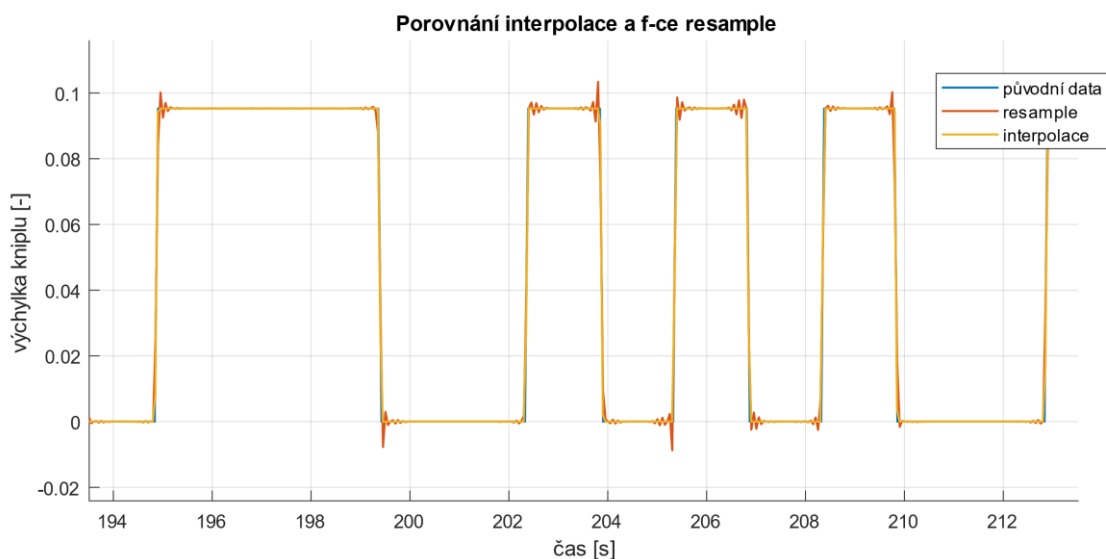
Během zpracovávání naměřených dat se objevil problém se vzorkováním. Ačkoliv bylo řečeno, že simulátor loguje data s frekvencí 20 Hz odpovídající periodě vzorkování 0,05 s, bylo po přezkoumání hodnot časových vektorů naměřených signálů zjištěno, že vzorkování není rovnoměrné. Pro přiblížení je na obrázku 3.11 vykreslen histogram rozdílů dvou po sobě jdoucích hodnot časového vektoru. Zde by měla správně jako nejčtetnější hodnota vycházet perioda 0,05 s.



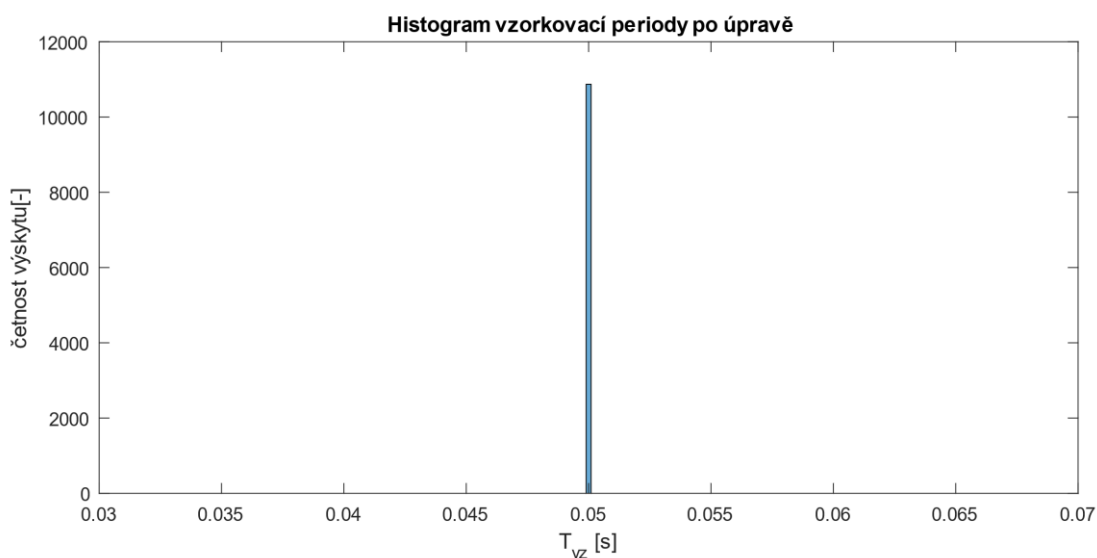
Obrázek 3.11 Histogram vzorkovací periody

Tato chyba ve vzorkování nemá sice významný dopad na grafické zobrazování průběhů, ale problém se projeví u samotné identifikace a následné operace porovnání (např. u funkce *compare*). Ta požaduje, aby hodnoty dvou průběhů na stejné pozici v časovém vektoru odpovídaly stejnému času. Pokud toto není splněno, porovnávají se hodnoty, které nastaly v jiný časový okamžik, což vede k nesmyslným výsledkům.

Problém lze vyřešit pomocí funkce *resample* nebo interpolace. V tomhle případě byla zvolena náprava pomocí interpolace, jelikož funkce *resample* obsahuje i určitý druh filtru, a tudíž znehodnocovala obdélníkové signály, které byly používány jako řídicí signály, viz obrázek 3.12. Histogram interpolovaného časového vektoru je možné vidět na obrázku 3.13. [8]



Obrázek 3.12 Porovnání resamplovaného a interpolovaného PRBS



Obrázek 3.13 Histogram vzorkovacích period po interpolaci

3.4 Porovnání naměřených dat a odezev modelu

Zpracovaná a převzorkovaná data bylo nyní možné porovnat s odezvou identifikovaných modelů. Pro zjištění odezvy modelů bylo vytvořeno zapojení v Simulinku, které je blíže popsáno v kapitole 3.4.3. Pro určení podobnosti průběhů byla využita funkce *compare*. Tato funkce spočítá chybu mezi odezvou modelů a měřeními daty, tzv. NRMSE (Normalised Root Mean Square Error), a z ní určí procenta fitu – podobnosti průběhů. Pro každou veličinu ve všech datových sadách je jejich fit procento zobrazeno v legendě.

Níže jsou zobrazeny grafy porovnání pro jednotlivé typy generovaných signálů popsaných v kapitole 3.1. Každý graf obsahuje průběh změřeného signálu a další dva průběhy, které odpovídají odezvě 4-stavového a 5-stavového modelu. Na konci této kapitoly lze také nalézt graf porovnání vlivu tahu motoru na výšku letu, viz obrázek 3.19, a graf, který ukazuje nefunkčnost identifikovaných modelů v případě nedodržení počátečních podmínek, viz obrázek 3.18.

Z grafů je na první pohled vidět podstatný rozdíl mezi 4-stavovým a 5-stavovým modelem – odlišný průběh výšky letu. Tento rozdíl je způsoben jiným přístupem, kterým jednotlivé modely výšku počítají, a uvažováním jiných počátečních podmínek.

Všechna data, která byla použita pro identifikaci nebo porovnání, byla nulována. Zároveň se předpokládá, že všechny úhly ve chvíli změny výchylky kniplu byly nulové (podmínka vyváženého letu). Dokonale vyváženého letu je však téměř nemožné dosáhnout, tím pádem opravdové hodnoty úhlů nulové nebyly. Nulováním počátečních úhlů může dojít k zanedbání faktu, že letadlo už před aplikací výchylky výškového kormidla stoupalo. Jelikož je zisk výšky charakterizován jako integrátor, tak příspěvek počátečního růstu výšky může velmi ovlivnit celkovou naintergovanou hodnotu výšky letu.

5-stavový model předpokládá nulovou hodnotu počátečních podmínek, díky čemuž nepočítá s vlivem opravdových počátečních hodnot úhlů na průběh výšky letu. Tento model je v tomto případě nevhodný pro správné zobrazení výšky, protože nedokáže počítat s nenulovými počátečními podmínkami.

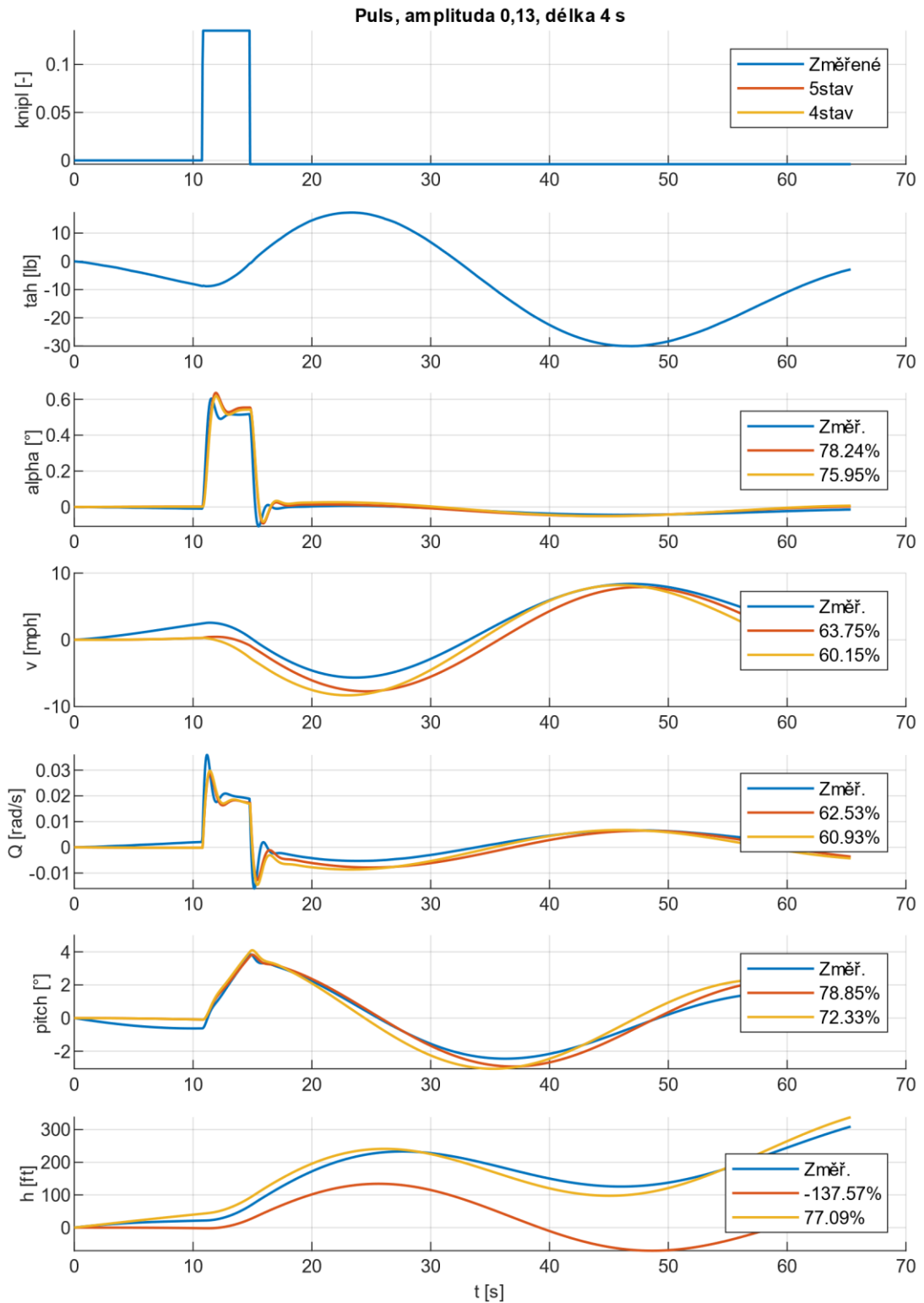
4-stavový model stav výšky letu počítá dle rovnice (4.1), která počáteční stav úhlů respektuje, a tudíž dokáže výšku lépe reprezentovat. To je patrné i z grafů porovnání, které jsou zobrazeny níže.

Nenulové úhly nejsou pro případ čtyř stavových veličin, společných pro oba modely, zase tak podstatné, což je možné pozorovat na průbězích, jejichž shoda s původně naměřenými daty se pohybuje nad 60 %.

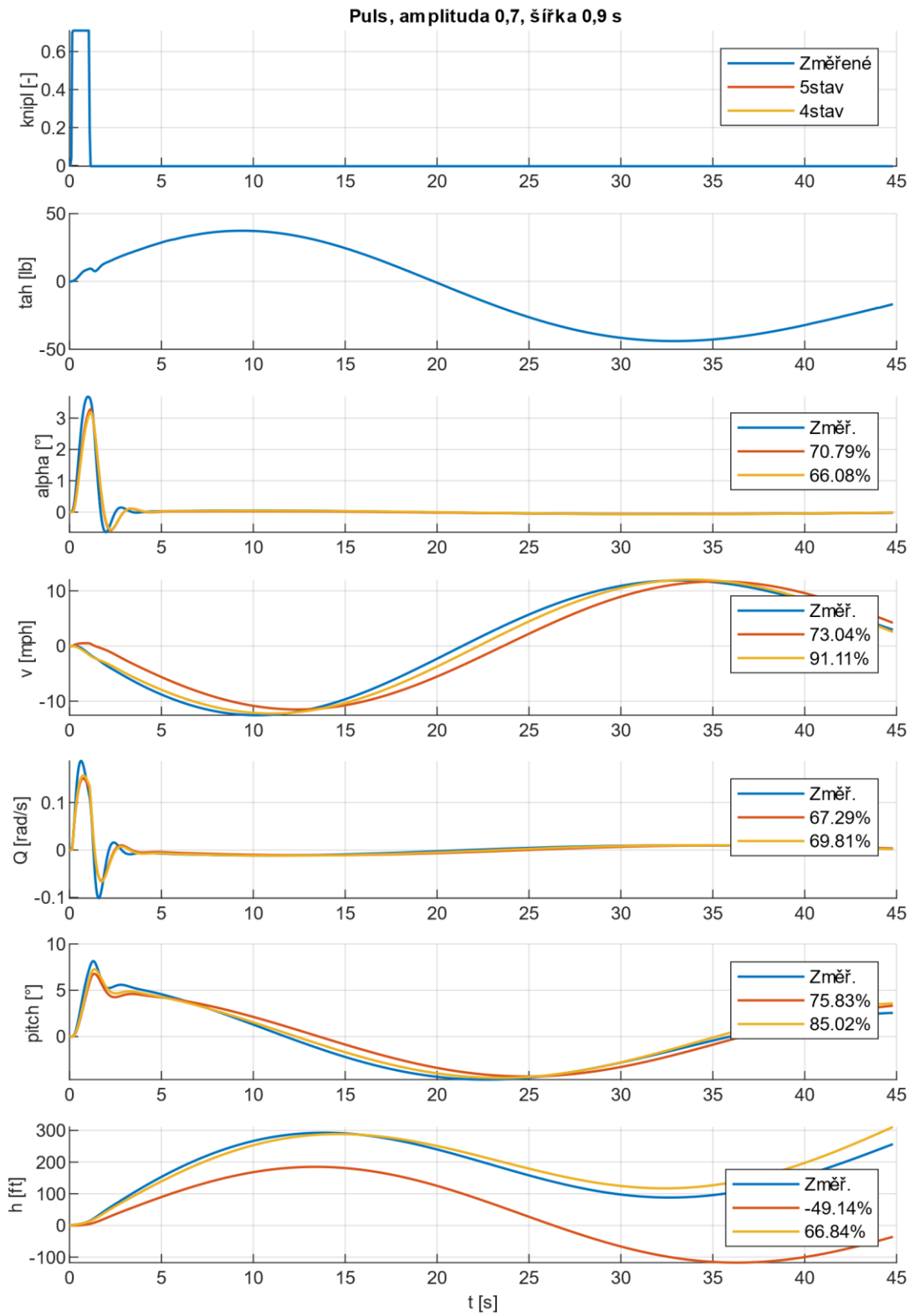
Pro každý typ signálu bylo provedeno několik měření. Pro vykreslení byla vybrána měření, která nejlépe reprezentují datovou sadu daného signálu. Níže zobrazené grafy porovnání odpovídají validačním signálům:

- Puls, amplituda 0,13, délka 4 s (viz obrázek 3.14),
- Puls, amplituda 0,7, délka 0,9 s (viz obrázek 3.15),
- Rampa, amplituda 0,09, délka 20 s, přidržení 5 s (viz obrázek 3.16),
- PRBS, amplituda 0,095 (viz obrázek 3.17).

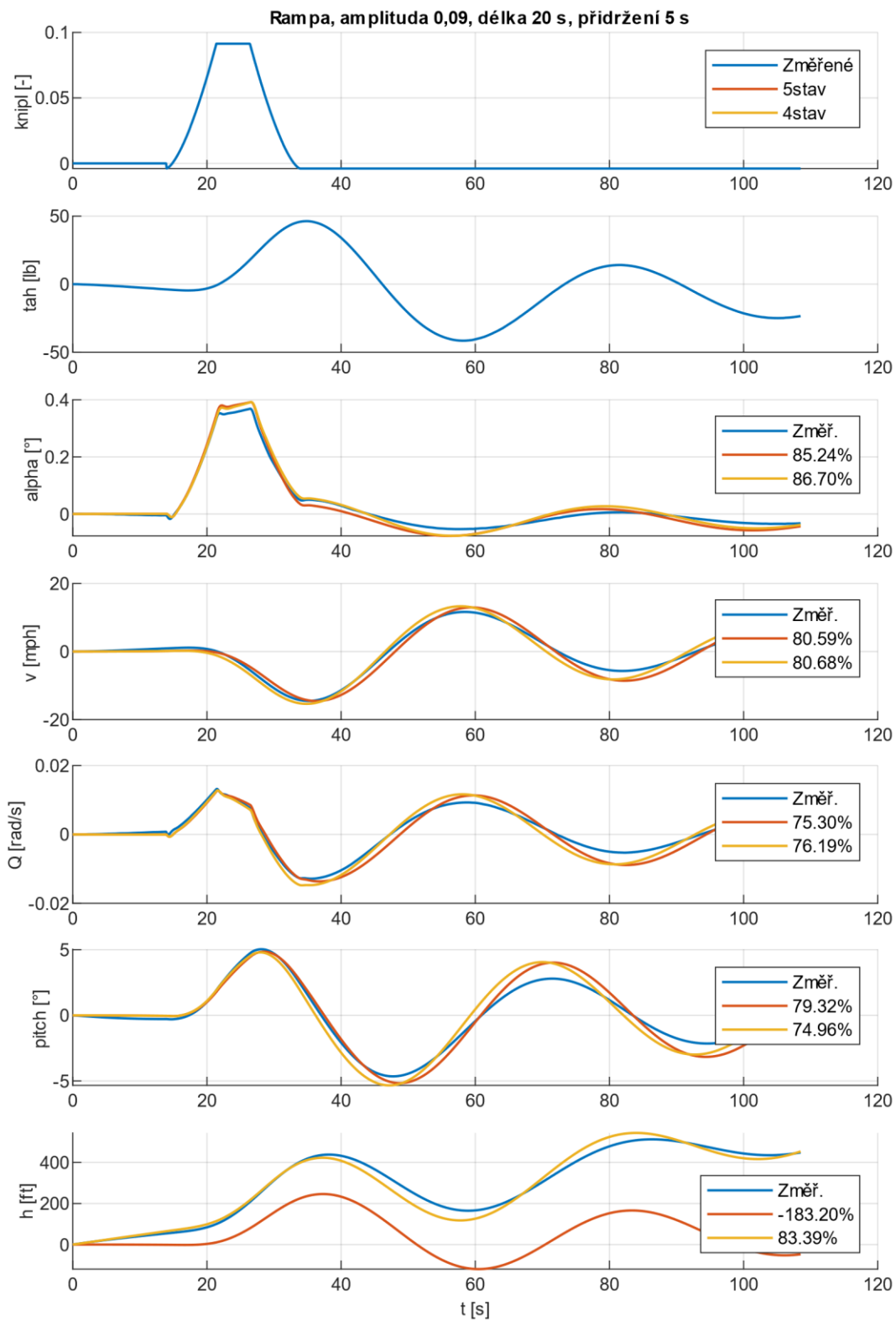
3.4.1 Porovnání průběhů pro jednotlivé validační signály



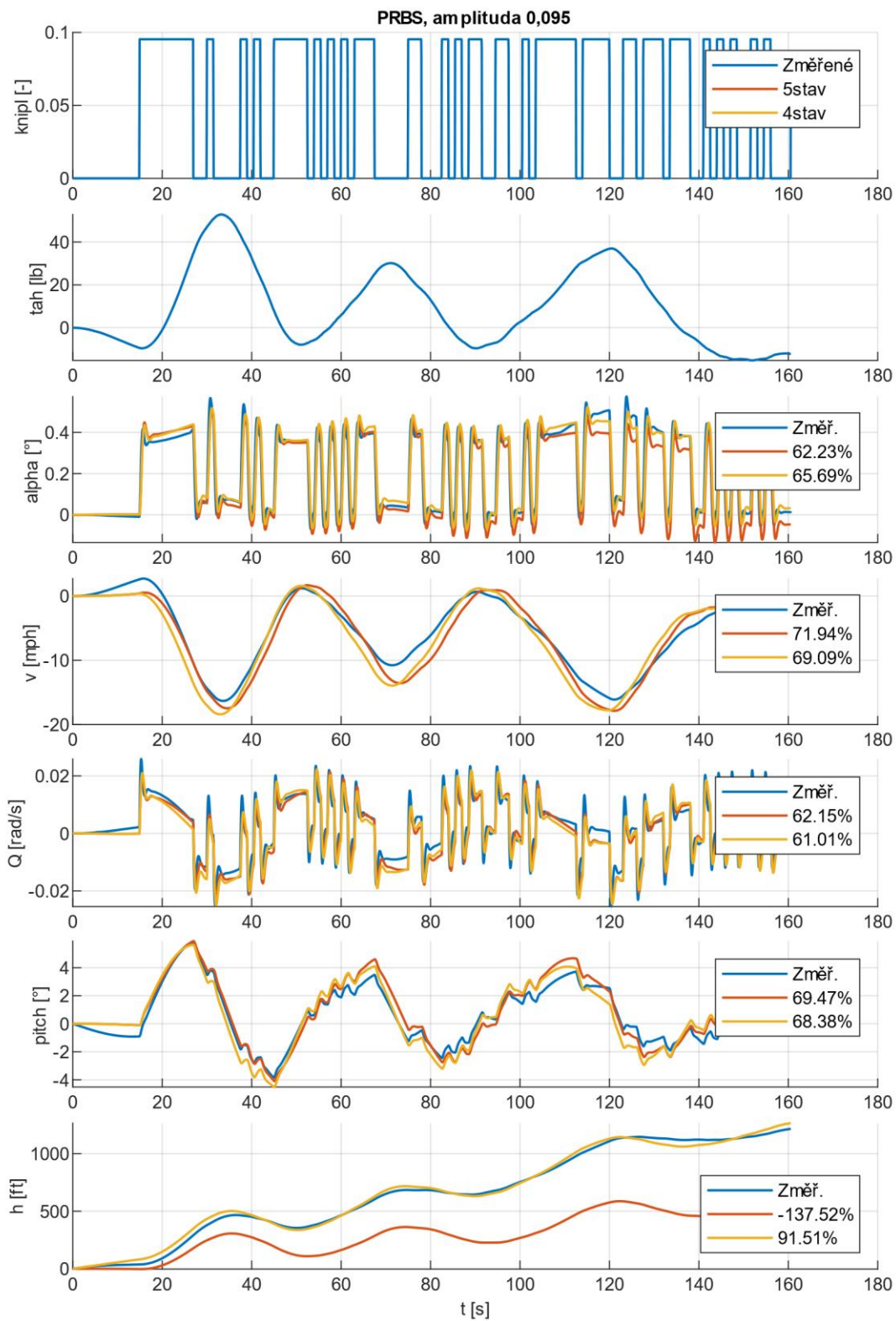
Obrázek 3.14 Porovnání pro malý puls



Obrázek 3.15 Porovnání pro velký puls



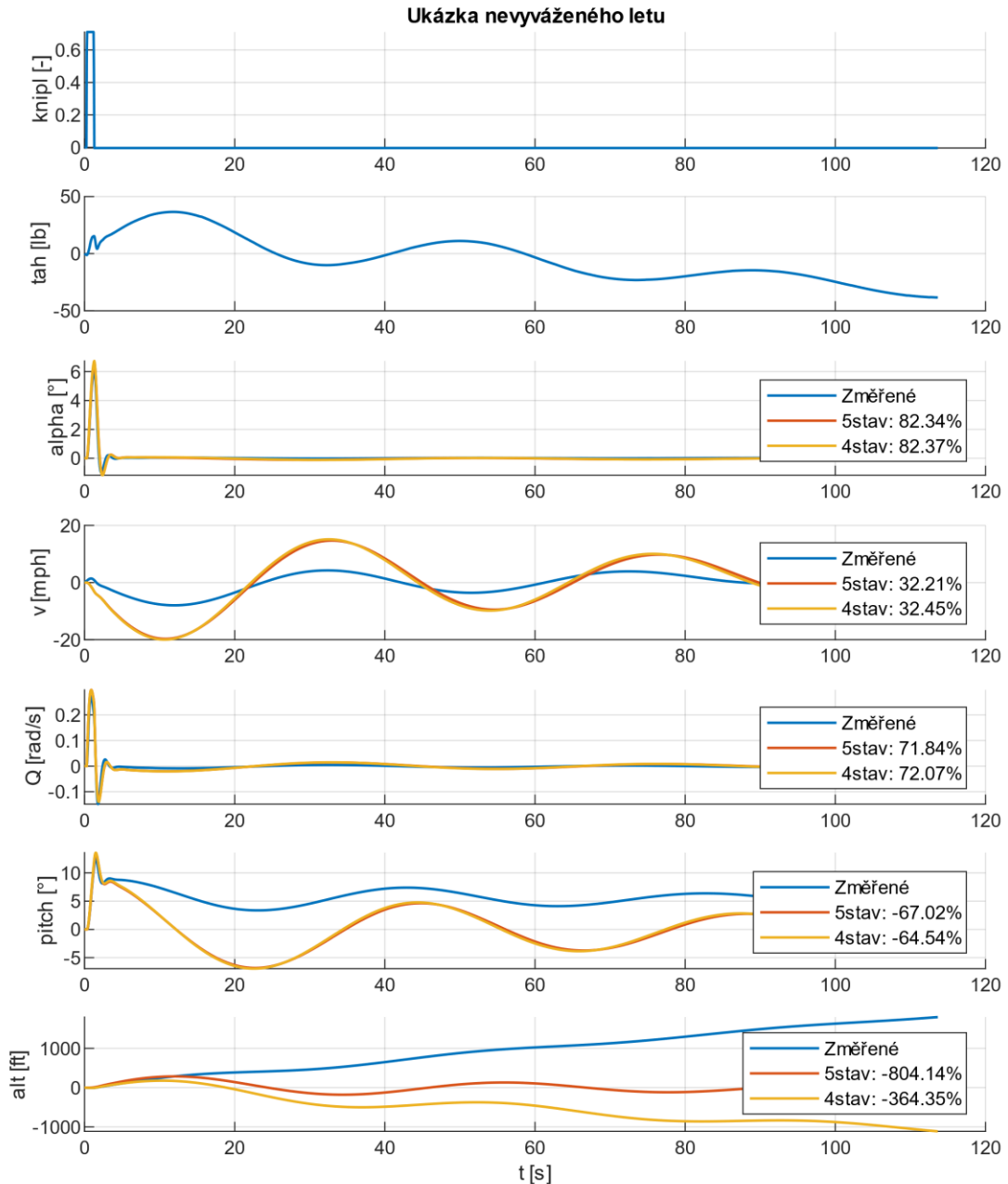
Obrázek 3.16 Porovnání pro rampu



Obrázek 3.17 Porovnání pro PRBS

3.4.2 Porovnání vyváženého a nevyváženého letu

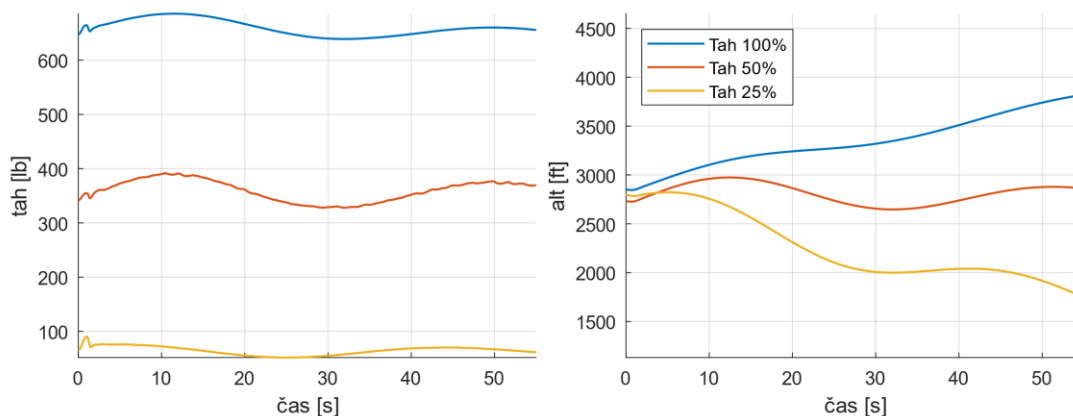
Na obrázku 3.18 je zobrazen průběh nevyváženého letu a odezvy jednotlivých modelů. Během nevyváženého letu letoun samovolně stoupá, přestože nedostal jakýkoliv povel od ovládacích prvků. V tomto případě nelze uvažovat nulový počáteční úhel podélného sklonu θ . Stoupání je totiž zapříčiněno kladným úhlem θ (v grafu pitch), což vyplývá z rovnice pro výpočet výšky (1.27). Počáteční podmínky se pak nenachází v okolí pracovních bodů navržených modelů, a proto bylo třeba let vyvážit.



Obrázek 3.18 Porovnání dat z nevyváženého letu

3.4.3 Porovnání průběhů s různými tahy

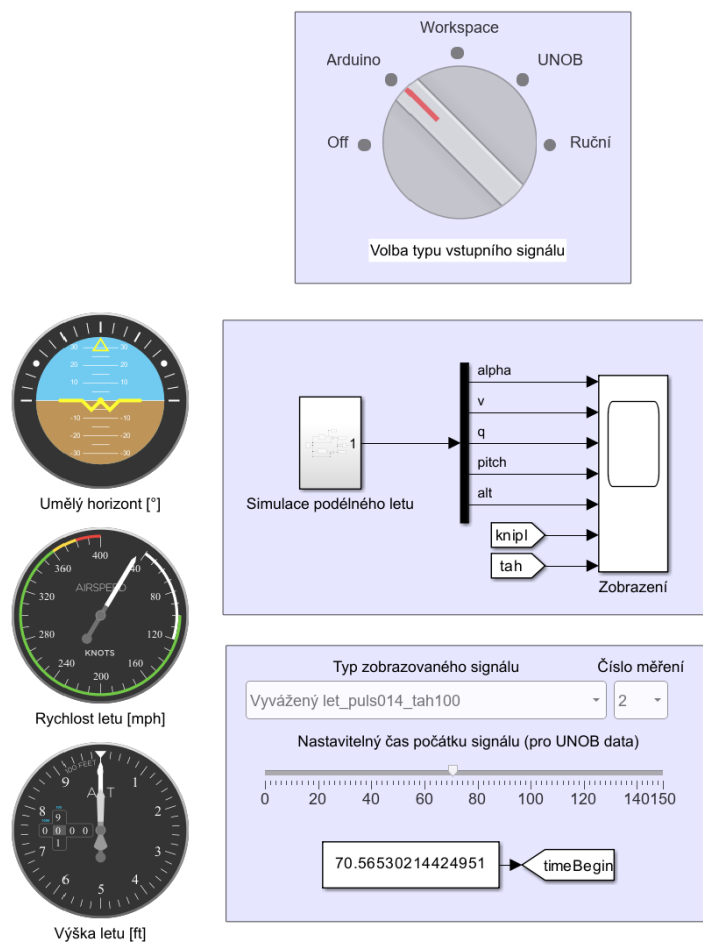
Pro demonstraci vlivu tahu motoru, jakožto druhého ze vstupů, je na obrázku 3.19 graf různých měřených průběhů výšky v závislosti na různém tahu motoru. Procenta v legendě označují výchylku na periférii ovládající tah motoru v simulátoru. Z grafů je vidět, že samotný tah motoru má vliv na směr a velikost změny výšky letu, což je dáno fyzikálními vlastnostmi.



Obrázek 3.19 Průběh výšky letu pro různé tahy motoru

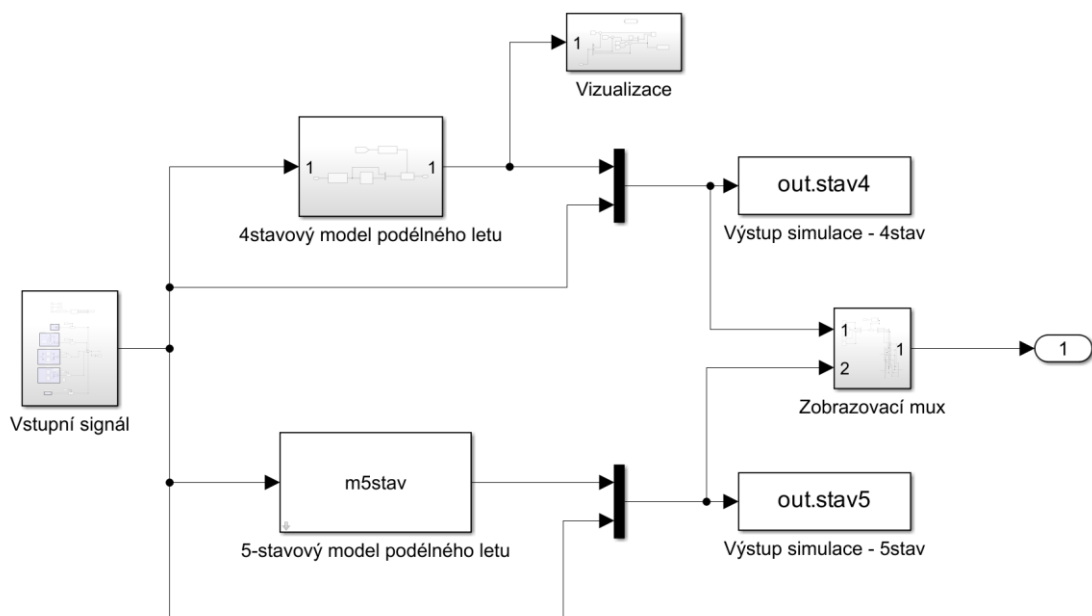
4. SIMULAČNÍ MODEL PRO ŘÍZENÍ PODÉLNÉHO LETU

Pro získávání odezev identifikovaných modelů byl využit nástroj MATLAB/Simulink. V něm byl sestaven simulační model, kterému byly přidány různé možnosti řízení. Je zde možnost vykreslit odezvu na již změřená data, na průběh předaný přes Workspace, na libovolně sestavený signál – např. pomocí step bloků – nebo je zde možnost připojit Arduino desku a využít na ní implementované generování signálů popsané v kapitole 3.1.



Obrázek 4.1 Grafické rozhraní simulace

Grafické rozhraní na obrázku 4.1 obsahuje tři panely a tři ukazatele. Na prvním panelu se vybírá zdroj vstupního signálu použitého pro simulaci. Druhý panel obsahuje samotný simulační model, viz obrázek 4.2, a *Scope*, na kterém se zobrazuje průběh simulovaných veličin. Třetí panel funguje pouze pro simulaci dat naměřených na UNOB. Volí se na něm, jakou naměřenou sadu použít a od jakého časového okamžiku ji simulovat. Ukazatele slouží pouze jako vizualizační prvek.



Obrázek 4.2 Simulační model

4.1 Popis simulačního modelu

V simulačním modelu jsou implementovány čtyři způsoby, jakými vybudit odezvu z identifikovaných modelů. Výstup obou modelů je společně se vstupními signály posílán do *Scope* pro okamžité zobrazení a zároveň pomocí bloků *To Workspace* do *Workspace* pro další případné zpracování.

Blok „Zobrazovací mux“ slouží pro sloučení stejných veličin, tak aby bylo dosaženo přehledného zobrazení v bloku *Scope*. V případě simulace dat z UNOB se naměřená data k simulovaným signálům připojí právě v tomto bloku.

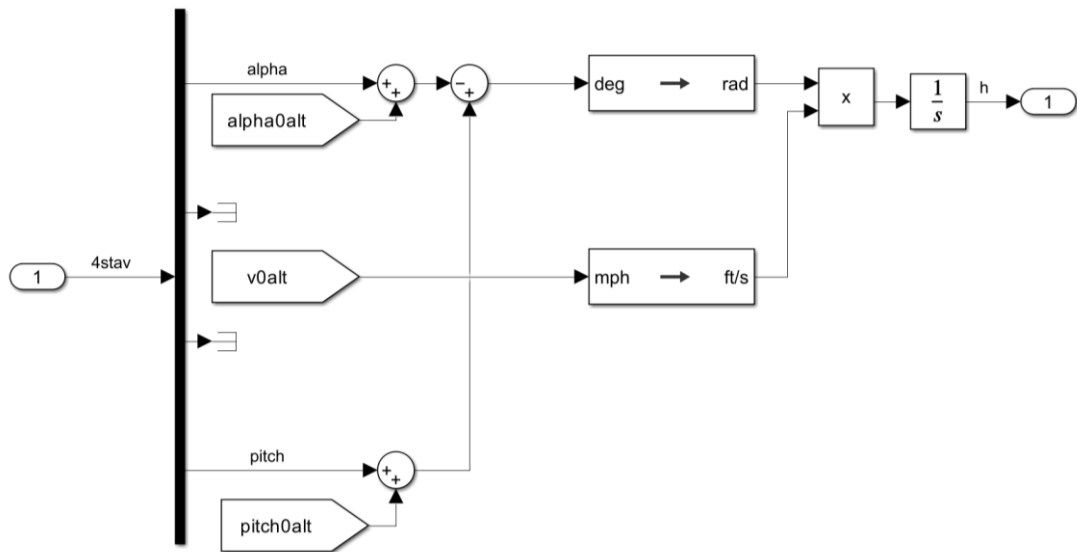
Blok „Vizualice“ zajišťuje výpočet hodnot pro tři ukazatele přítomné v grafickém rozhraní. Počítaná hodnota je získávána jako součet počátečních podmínek simulace a odchylek generovaných ze 4-stavového modelu. Ten je vhodnější díky lepší reprezentaci výšky letu letadla, viz diskuse v kapitole 3.4.

4.2 Implementace výpočtu výšky u 4-stavového modelu

Pro výpočet výšky letu u 4-stavového modelu byla implementována upravená rovnice (1.27), kde původní rovnice byla obohacena o přičtení počátečních podmínek u úhlu podélného sklonu θ a úhlu náběhu α :

$$\Delta \dot{h} = v_0 [(\Delta \theta + \theta_0) - (\Delta \alpha + \alpha_0)] \quad (4.1)$$

Zapojení v Simulinku reprezentující tuto rovnici je na obrázku 4.3.



Obrázek 4.3 Výpočet výšky u 4-stavového modelu

4.3 Průběh simulace pro jednotlivé vstupní signály

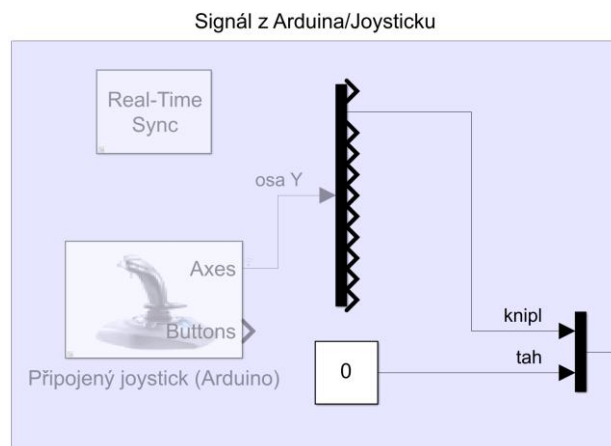
První krok, který se musí vykonat před jakýmkoliv používáním simulátoru, je jeho inicializace. K tomuto úkolu byl vytvořen script v prostředí MATLAB. Po jeho spuštění jsou z příslušných souborů načteny používané modely, nadefinované počáteční podmínky a je nastavena délka simulačního kroku. Ve scriptu je také nutno určit, zda se bude používat jakýkoliv připojený joystick (Arduino).

Po inicializaci stačí už jen vybrat vstupní signál a spustit simulaci.

4.3.1 Signál z joysticku (Arduino)

Po zapnutí patřičně nastaveného inicializačního scriptu se v simulaci odkomentuje blok zaopatřující komunikaci mezi připojeným joystickem a Simulinkem, viz obrázek 4.4. Kdyby byl tento blok odkomentován v případě nepřipojeného joysticku, nebylo by možné simulaci spustit.

Inicializace zároveň aktivuje blok *Real-Time Synchronization*, který synchronizuje časování simulace s reálným časem. Simulovaná doba i trvání simulace v reálném čase jsou pak shodné. Během probíhající simulace Simulink pravidelně aktualizuje stav připojeného joysticku a dokáže tak zaznamenat průběh jeho výchyly.



Obrázek 4.4 Propojení mezi joystickem (Arduinem) a Simulinkem

Podle hodnoty výchylky je počítána odezva modelů, kterou lze společně s výchylkou pozorovat pomocí bloku „Zobrazení“ v grafickém rozhraní simulačního modelu. Počáteční podmínky jsou $v_0 = 170 \text{ mph}$, $h_0 = 2900 \text{ ft}$ a ostatní jsou rovny nule.

4.3.2 Signál z Workspace

Při zvolení tohoto signálu jsou počáteční podmínky a simulované průběhy výchylky výškového kormidla δ_E a tahu motoru δ_T brány z Workspace. Tento signál byl použit pro výpočet procent fitu a vykreslování grafů porovnání v kapitole 3.4.

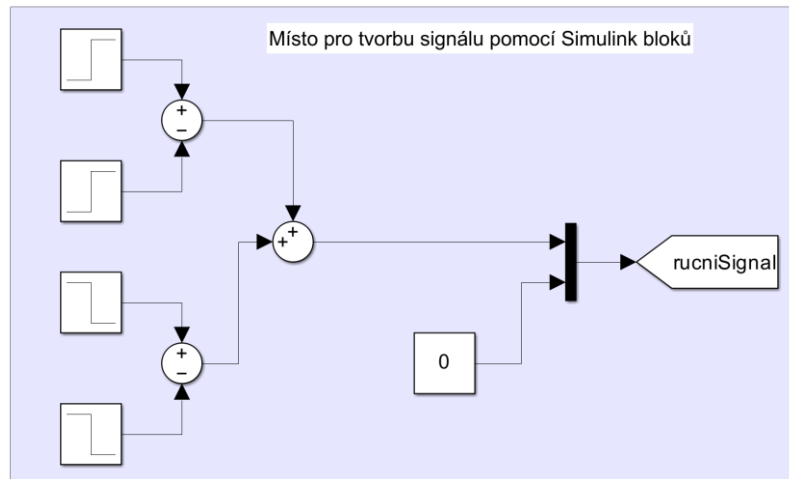
4.3.3 Data změřená na UNOB

Signály z UNOB reprezentují data naměřená na statickém leteckém simulátoru, na jejichž základě se vykreslí zvolené signály i příslušné odezvy modelů. Během inicializace je načtena matice, ve které jsou uloženy hodnoty všech měření provedených na simulátoru na UNOB.

V grafickém rozhraní simulačního modelu, viz obrázek 4.1, je možné navolit si, jaký typ signálu se má simulovat, kolikáté měření má být použito a od jakého času se mají měřená data vykreslovat. Počáteční podmínky odpovídají vybranému signálu ve zvoleném časovém okamžiku. Samotná simulace pak běží do té doby, dokud jsou dostupná naměřená data, poté se zastaví.

4.3.4 Ručně poskládaný signál

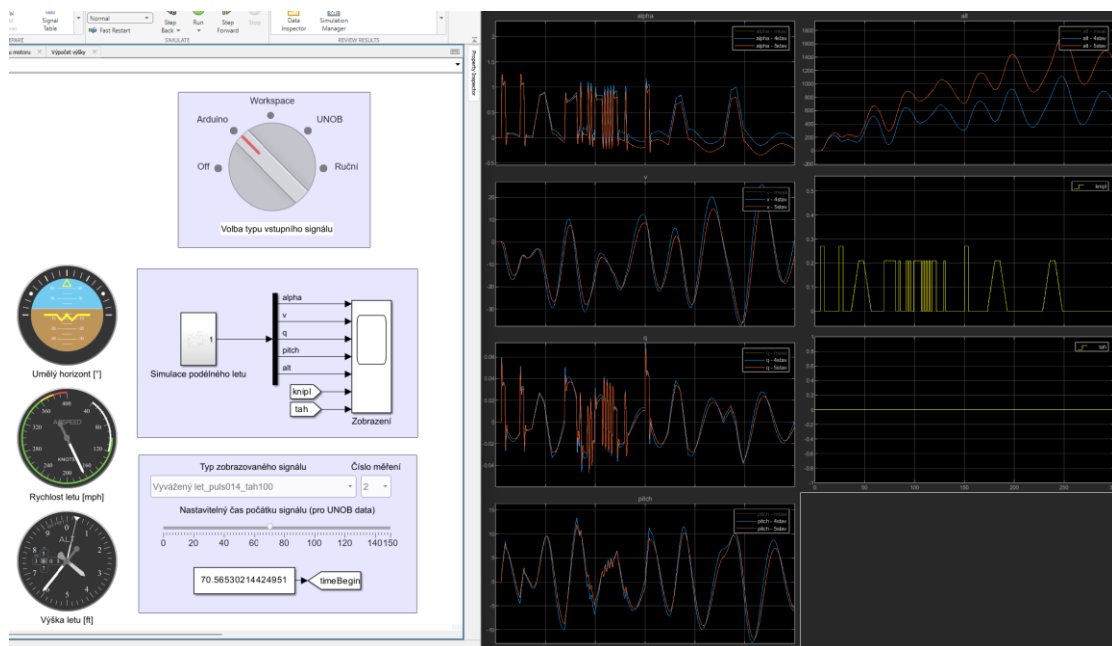
Pro tvorbu vlastního signálu je v tomto případě využít Simulink. Za pomoci bloků, které generují signál, jako jsou *Step*, *Pulse Generator*, *Sine Wave* apod., je možné poskládat libovolný signál a použít ho pro získání odezvy modelů jako vstupní. Pro jednoduché propojení poskládaného signálu a simulačního modelu je vedle panelů hlavního grafického rozhraní místo pro tvorbu vlastního signálu, viz obrázek 4.5. Průběh vytvořeného signálu je pomocí bloku *Goto* předán simulačnímu modelu. Počáteční podmínky jsou v tomto případě $v_0 = 170 \text{ mph}$, $h_0 = 2900 \text{ ft}$ a ostatní jsou rovny nule.



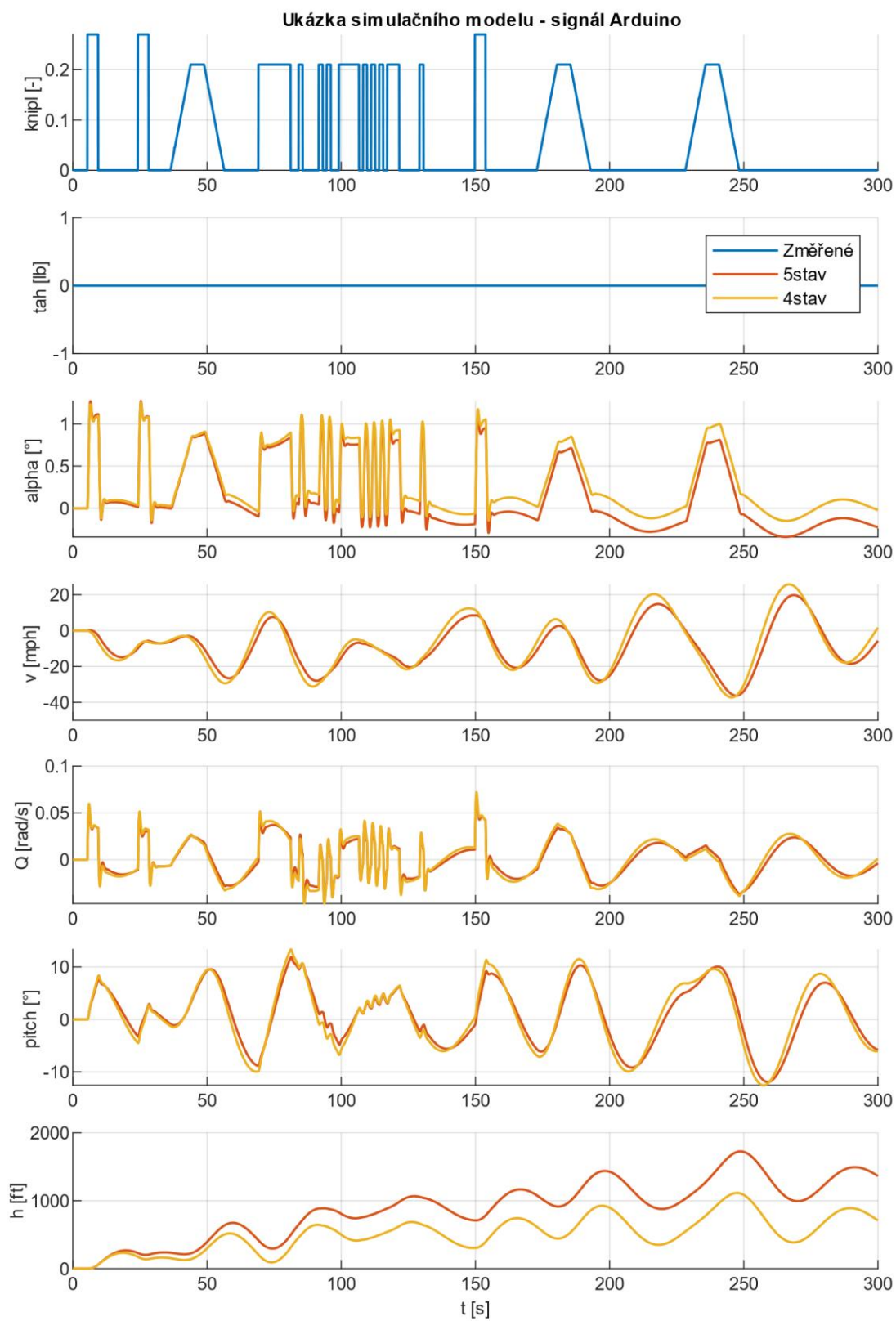
Obrázek 4.5 Panel pro tvorbu vlastního signálu

4.4 Ukázka simulovaných průběhů

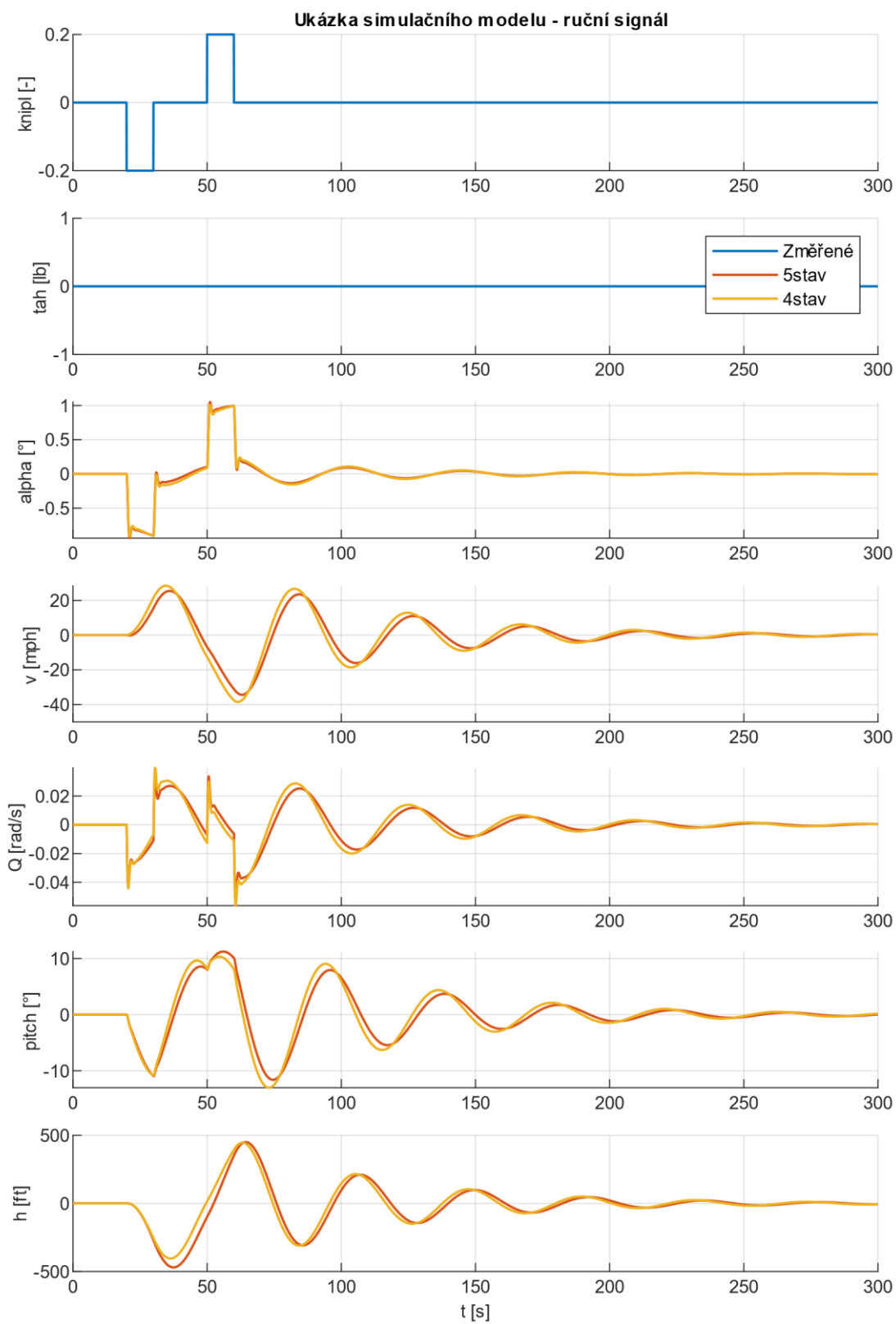
Pro ukázkou bylo odsimulováno řízení pomocí Arduina, ručně poskládaného signálu, a nakonec pomocí zvolených UNOB dat. Pro Arduino byly postupně aplikovány všechny tvary naprogramovaných signálů. Ručně poskládaný signál vychází ze zapojení na obrázku 4.5, které reprezentuje puls dolů a nahoru. Nastavení vybraný signálu je možné vidět na obrázku 4.9. Výstupy lze zobrazit buď přímo ve *Scope* v Simulinku, viz obrázek 4.6, nebo pak přehledněji v samostatném okně v MATLABu, viz obrázky 4.7, 4.8 a 4.10.



Obrázek 4.6 Ukázka použití simulačního modelu



Obrázek 4.7 Ukázka simulačního modelu – signál Arduino



Obrázek 4.8 Ukázka simulačního modelu – ruční signál

Typ zobrazovaného signálu Číslo měření

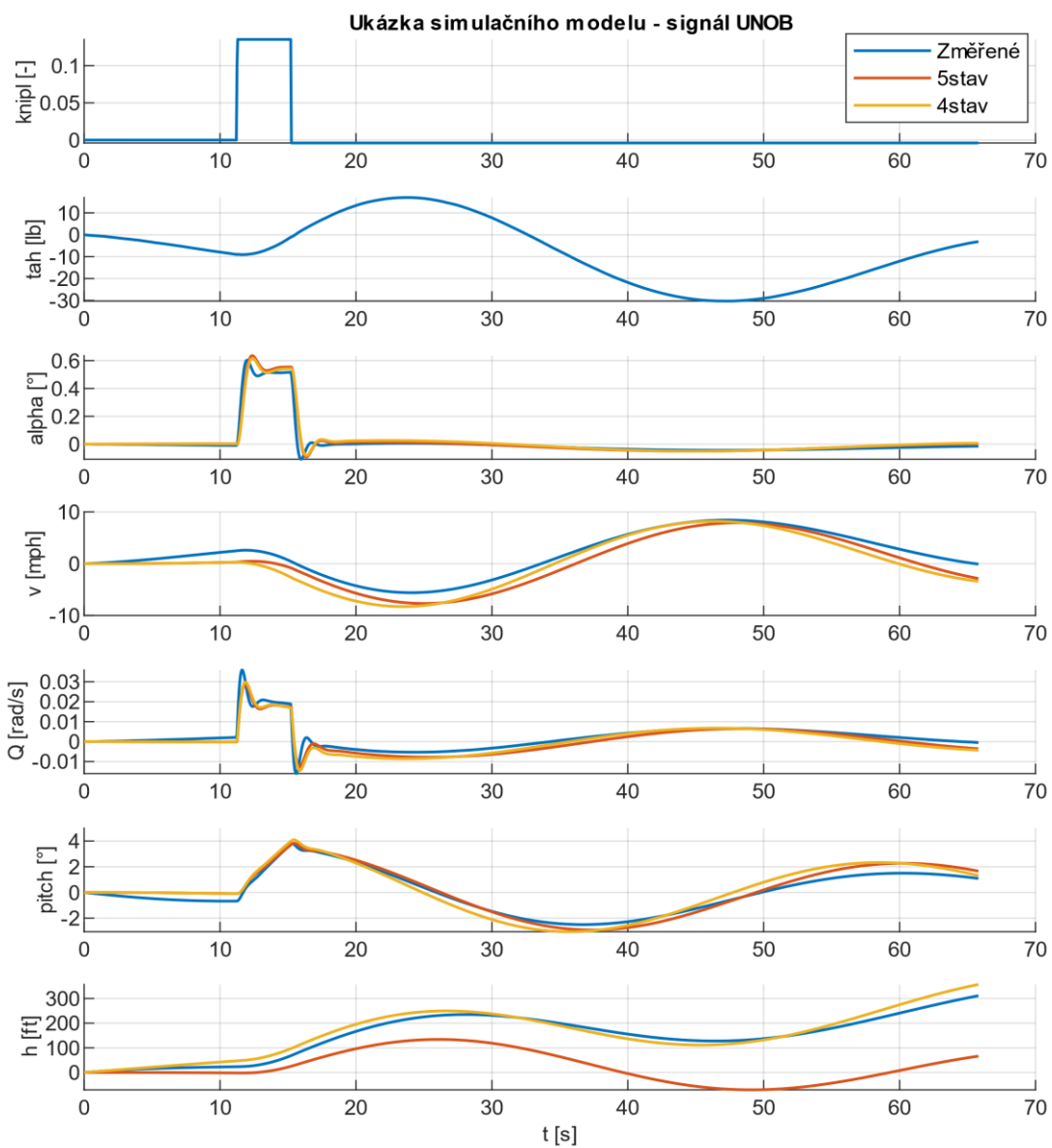
Vyvážený let_puls014_tah100 2

Nastavitelný čas počátku signálu (pro UNOB data)

0 20 40 60 80 100 120 140 150

65.88693957115009 timeBegin

Obrázek 4.9 Výběr typu signálu pro UNOB simulaci



Obrázek 4.10 Ukázka simulačního modelu – UNOB signál

5. ZÁVĚR

V první části práce byly shrnuty relevantní informace k modelování podélného letu letadla a následně byly tyto znalosti uplatněny při návrhu vlastních modelů v prostředí MATLAB. Dále se práce zabývala identifikací parametrů navržených modelů a popsáním dynamických vlastností modelů získaných. Získané modely pak byly podrobeny procesu validace, který rozhodl o jejich funkčnosti. Na závěr byl z validovaných modelů sestaven simulační model, který lze použít pro zjištění odezvy letounu na různé typy vstupních signálů.

V teoretické části je popsán postup při návrhu modelu. Jsou zde zmíněny souřadnicové systémy, a také objasněn postup pro sestavení pohybových rovnic popisujících pohyb letadla, linearizace těchto rovnic a transformace do stavového popisu. Na konci první kapitoly je rozvedena základní dynamika podélného letu, ze které vyplývá, že na stavových veličinách se bude projevovat tzv. pomalé a rychlé kmitání.

Pro praktickou část práce, tedy tvoření modelu, identifikaci i validaci, bylo využito prostředí MATLAB s jeho rozšířením System Identification Toolbox a pro simulace byl použit Simulink.

Ve druhé kapitole byly navrženy dva modely reprezentující podélný let, 4-stavový a 5-stavový, kterým odpovídá rovnice (2.1). Pro tyto modely pak byly identifikovány potřebné koeficienty z dříve naměřených dat získaných z leteckého simulátoru na UNOB. Porovnání odezvy modelu s identifikačními daty lze vidět na obrázku 2.3. Dynamické vlastnosti obou modelů, popsané v kapitole 2.3, odpovídají teoretickým poznatkům. Tato skutečnost byla potvrzena pomocí příkazu *pzmap*, jehož výstup je vykreslen na obrázku 2.4. Tabulka 2.1 pak shrnuje jednotlivé údaje o vykreslených pólech systémů.

Ve třetí kapitole je popsán proces validace, pro který bylo nutné naměřit nová data na simulátoru na UNOB. Pro generování signálů, na které byla měřena odezva, byla využita deska Arduino Leonardo. Grafy porovnání a zhodnocení jsou shrnuty v kapitole 3.4.

Během zpracovávání naměřených dat se objevil problém s nerovnoměrným vzorkováním, který byl opraven pomocí interpolace. Data byla dále porovnána s odpovídající odezvou identifikovaných modelů, viz kapitola 3.4. Z porovnání vyplývá, že 5-stavový model nedokáže reprezentovat průběh výšky pro jiné počáteční podmínky než nulové, na rozdíl od modelu 4-stavového, který toto zvládá. Shoda ve zbylých čtyřech stavových veličinách se pohybuje nad 60 %.

V poslední kapitole je popsán simulační model, na kterém je možnost simulovat odezvy identifikovaných modelů na různé vstupní signály. Mezi implementovanými zdroji signálu je libovolný joystick (nebo Arduino Leonardo), signál přicházející z Workspace, ručně poskládaný signál v Simulinku a je zde také možnost odsimulovat si odezvu na jakákoli data změřená na UNOB.

LITERATURA

- [1] JALOVECKÝ, Rudolf. *Palubní systémy řízení letu*. Online. Brno: Univerzita obrany, 2014. ISBN 978-80-7231-500-0. Dostupné z: <https://www.digitalniknihovna.cz/mzk/view/uuid:c974f860-7e49-11ea-b147-005056827e51?page=uuid:7711085e-4659-4231-8aa3-737131224ee8>. [cit. 2024-05-20].
- [2] PETERS, Mark a KONYAK, Michael A. *The Engineering Analysis and Design of the Aircraft Dynamics Model*. Online. Atlantic City, 2012. Dostupné z: https://www.faa.gov/sites/faa.gov/files/about/office_org/headquarters_offices/ang/AircraftDynamicsModel.pdf. [cit. 2023-12-31].
- [3] FOSSEN, Thor I. *Mathematical Models for Control of Aircraft and Satellites*. Online. 3rd ed. Department of Engineering Cybernetics, NTNU, 2013. Dostupné z: <https://www.scribd.com/document/300312683/Aircraft-Fossen-2013>. [cit. 2023-12-31].
- [4] COOK, M. V. *Flight dynamics principles: [a linear systems approach to aircraft stability and control]*. Online. 2nd ed. Burlington, MA: Butterworth-Heinemann/Elsevier, 2007. ISBN 978-0-7506-6927-6. Dostupné z: <https://aerocastle.files.wordpress.com/2012/04/flightdynamicsprinciples.pdf>. [cit. 2024-05-20].
- [5] ŠTECHA, Jan a HAVLENA, Vladimír. *Teorie dynamických systémů*. Praha: České vysoké učení technické, 2002. ISBN 80-01-01971-3.
- [6] JIRGL, Miroslav. *Analýza modelů chování pilota při řízení letu letounu*. Online, Dizertační práce, vedoucí doc. Ing. Zdeněk Bradáč, Ph.D. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2016. Dostupné z: https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=131942. [cit. 2023-12-31].
- [7] ROBERTS, Jennifer. *Manuál k X-PLANE 10*. Online. 2015. Dostupné z: https://www.x-plane.com/files/manuals/X-Plane_10_Desktop_manual.pdf. [cit. 2024-05-20].
- [8] THE MATHWORKS, INC. *MATLAB Help Center*. Online. 2023. Dostupné z: https://www.mathworks.com/help/?s_tid=gn_supp. [cit. 2023-12-31].

- [9] PEET, Matthew M. *Spacecraft and Aircraft Dynamics: Lecture 11: Longitudinal Dynamics*. Online. Illinois Institute of Technology, 2010. Dostupné také z: <https://control.asu.edu/Classes/MMAE441/Aircraft/441Lecture11.pdf>.
- [10] JURA, Pavel. *Skripta Signály a systémy: Část 2: Spojité systémy*. Online prezentace. 3rd. 2017. [cit. 2024-05-20]. (dostupné po přihlášení)
- [11] PETYOVSKÝ, Petr. *Přednášky BPC-LOS: Logické obvody a systémy 9*. Online. 2023. (dostupné po přihlášení).
- [12] *Linear Feedback Shift Registers*. Online. Wayback Machine. 2001, 2010-05-04. Dostupné z: https://web.archive.org/web/20181001062252/http://www.newwaveinstruments.com:80/resources/articles/m_sequence_linear_feedback_shift_register_lfsr.htm#LFSR%20Implementations. [cit. 2024-05-20].
- [13] ARDUINO. *Arduino*. Online. 2006. Dostupné z: <https://www.arduino.cc>. [cit. 2024-05-20].
- [14] HEIRONIMUS, Matthew. *Arduino Joystick Library*. Online. GitHub. 2008. Dostupné z: <https://github.com/MHeironimus/ArduinoJoystickLibrary>. [cit. 2024-05-20].
- [15] FEDERAL AVIATION ADMINISTRATION. *Pilot's Handbook of Aeronautical Knowledge: Chapter 6, Flight Controls*. Online. 3rd. Oklahoma City: United States Department of Transportation, Federal Aviation Administration, Airman Testing Standards Branch, 2023. Dostupné z: https://www.faa.gov/regulations_policies/handbooks_manuals/aviation/faa-h-8083-25c.pdf. [cit. 2024-05-20].

SEZNAM SYMBOLŮ A ZKRATEK

Zkratky:

CSV	Comma-Seperated Values
FEKT	Fakulta elektrotechniky a komunikačních technologií
GOST	Gosstandart (ruské normy)
ICAO	International Civil Aviation Organization
LFSR	Linear-Feedback-Shift Register
NED	North-East-Down
NRMSE	Normalised Root Mean Square Error
PRBS	Pseudo-Random Binary Signal
UNOB	Univerzita obrany
VUT	Vysoké učení technické v Brně

Zkrácené zápisy (x reprezentuje libovolnou veličinu, zbylé znaky určují význam):

s_x	zjednodušení pro $\sin(x)$
c_x	zjednodušení pro $\cos(x)$
x	stav systému
x_0	ustálená hodnota veličiny x
Δx	odchylka od ustálené hodnoty x_0
\dot{x}	první derivace x
\ddot{x}	druhá derivace x

Symbole:

A	stavová matice systému
a_{ij}	koeficienty matice A (i udává řádek, j udává sloupec)
a_z	zrychlení v ose z [$m \cdot s^{-2}$]
B	matice řízení
b_{ij}	koeficienty matice B (i udává řádek, j udává sloupec)
C	matice výstupů
D	matice přímého vlivu vstupů na výstupy
g	vektor tíhového zrychlení [$m \cdot s^{-2}$]
h	výška letu [ft], [m]
I_x	moment setrvačnosti vzhledem k ose x [$kg \cdot m^2$]
I_y	moment setrvačnosti vzhledem k ose y [$kg \cdot m^2$]
I_z	moment setrvačnosti vzhledem k ose z [$kg \cdot m^2$]
I_{xy}	kvadratický moment setrvačnosti vzhledem k xy [$kg \cdot m^4$]
I_{xz}	kvadratický moment setrvačnosti vzhledem k xz [$kg \cdot m^4$]
K_R	statické zesílení

m	hmotnost [kg]
M	moment síly kolem osy \mathbf{y}_b [$N \cdot m$]
M_0	ustálená hodnota Machova čísla
\dot{P}	zrychlení změny úhlu příčného náklonu [$rad \cdot s^{-2}$],
Q	rychlost změny úhlu podélného sklonu [$rad \cdot s^{-1}$]
\dot{Q}	zrychlení změny úhlu podélného sklonu [$rad \cdot s^{-2}$]
\dot{R}	zrychlení změny úhlu kursu letu [$rad \cdot s^{-2}$]
T	perioda kmitání [s]
U	vektor podélné rychlosti [$m \cdot s^{-1}$]
\dot{U}	podélné zrychlení [$m \cdot s^{-2}$]
$\Delta \mathbf{u}(t)$	řídící signály
V	vektor příčné rychlosti [$m \cdot s^{-1}$]
\dot{V}	příčné zrychlení [$m \cdot s^{-2}$]
v	aerodynamická rychlost letadla [$m \cdot s^{-1}$]
V_T	vektor aerodynamické rychlosti [$m \cdot s^{-1}$]
W	vektor vertikální rychlosti [$m \cdot s^{-1}$]
\dot{W}	vertikální zrychlení [$m \cdot s^{-2}$]
X	vektor podélné síly [N]
\mathbf{x}	stav
\mathbf{x}_0	ustálená hodnota stavu \mathbf{x} (platí pro všechny veličiny s indexem)
$\Delta \mathbf{x}$	odchylka od ustálené hodnoty \mathbf{x}_0 (platí pro všechny veličiny s Δ)
$\Delta \dot{\mathbf{x}}(t)$	derivované stavy systému
$\Delta \mathbf{x}(t)$	stavy systému
\mathbf{x}_b	osa x letadlového souřadnicového systému
\mathbf{x}_g	osa x zemského souřadnicového systému
\mathbf{x}_w	osa x aerodynamického souřadnicového systému
Y	vektor příčné síly [N]
$\Delta \mathbf{y}(t)$	výstupy systému
\mathbf{y}_b	osa y letadlového souřadnicového systému
\mathbf{y}_g	osa y zemského souřadnicového systému
\mathbf{y}_w	osa y aerodynamického souřadnicového systému
Z	vektor vertikální síly [N]
\mathbf{z}_b	osa z letadlového souřadnicového systému
\mathbf{z}_g	osa z zemského souřadnicového systému
\mathbf{z}_w	osa z aerodynamického souřadnicového systému
α	úhel náběhu [$^\circ$], [rad]
β	úhel vybočení [$^\circ$], [rad]

γ	poločas utlumení amplitudy [s]
δ_T	výchylka tahu motoru
δ_{T0}	ustálená hodnota tahu motoru
δ_E	výchylka výškového kormidla (kniplu)
ε	tlumení
θ	úhel podélného sklonu [$^\circ$], [rad]
$\dot{\theta}$	rychlost změny úhlu podélného sklonu [rad · s ⁻¹]
Φ	úhel příčné náklonu [$^\circ$], [rad]
ψ	úhel kursu letu [$^\circ$], [rad]
ω	frekvence kmitání [rad · s ⁻¹]

Příloha A - Zdrojové kódy a MATLAB

Zdrojové kódy pro Arduino Leonardo, veškeré skripty z MATLABu a výsledný simulační model jsou uloženy na přiloženém USB.