



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

**Fakulta biomedicínského inženýrství
Katedra biomedicínské techniky**

**Návrh pohyblivé plošiny a metod hodnocení
pohybu segmentů těla živočichů**

*Design of rotational platform and methods for
evaluation of animals body segments
movements*

Diplomová práce

Studijní program: Magisterský

Studijní obor: Přístroje a metody pro Biomedicínu

Vedoucí práce: Doc. Ing. Patrik Kutílek, PhD.

Bc. David Škoda

Kladno květen 2016

Anotace

Práce se zabývá návrhem a konstrukčním řešením elektrické naklápěcí plošiny navržené pro měření pohybových změn malých živočichů při jejím naklápění v prostoru a metod kvantitativního hodnocení kinematických dat. Za tímto účelem je využito 3 na sebe kolmo umístěných kamer, jež jsou součástí konstrukce plošiny. Zpracování obrazové informace je realizováno v prostředí MatLab®. Dále je v práci popsáno řešení vyskytnuvších se problémů při návrhu jednotlivých realizací podsystémů plošiny, komunikačního rozhraní mezi navrženým softwarem a mechanickou částí plošiny a způsobů hodnocení naměřených dat. Testování návrhů plošiny a algoritmů bylo realizováno na žábách rozmanitých druhů a *gekončících nočních*, přičemž část práce zabývající se hodnocením je zaměřena pouze na gekončíky. Nakonec je v rámci ověření vhodnosti plošiny věnováno několik odstavců charakterizaci reakcí živočichů na pohyb plošiny, a jejich kvantitativnímu srovnání. Projekt je výsledkem spolupráce mezi ČVUT - Fakultou Biomedicínského Inženýrství, Přírodovědnou Fakultou Univerzity Karlovy v Praze a Fakultní nemocnicí Motol a je financován ze SGS15/107/OHK4/1T/17.

Klíčová slova

Naklápěcí plošina, kamerový systém, hodnocení pohybových vzorů;

Anotation

This work describes the design and mechanical properties of an Electronic Revolvable Platform constructed to measure slight movement changes of small animals when tilted in space. In order to do that a Camera System consisting of three orthogonal webcams has been proposed and placed as a part of the revolvable platform. The image processing methods are realized in MatLab®. This thesis describes certain solutions to problems that surfaced while dealing with every aspect of the design of both – the platform, and the SW to control it. The communication between them is also described and last, but not least, the means of quantifying the measured data and comparing methods between species are proposed and described. The measurements have so far been conducted on various species of frogs and *leopard geckos*, while this work focuses mainly on geckos. The purpose of this work is to find a similarity in the behavior patterns of different species on tilting of the platform in space and quantitatively compare the results between them. This project has been realized thanks to cooperation between ČVUT – The Faculty of Biomedical Engineering, Charles’s University in Prague – Faculty of Natural Sciences, and University Hospital Motol in Prague and has been financed by SGS15/107/OHK4/1T/17.

Keywords

Revolvable platform, Camera System, Quantification of movement patterns;

Katedra přírodovědných oborů

Akademický rok: 2015/2016

Z a d á n í d i p l o m o v é p r á c e

Student: **Bc. David Škoda**
Studijní obor: Přístroje a metody pro biomedicínu
Téma: **Návrh pohyblivé plošiny a metod hodnocení pohybu segmentů těla živočichů**
Téma anglicky: Design of rotational platform and methods for evaluation of animals body segments movements

Z á s a d y p r o v y p r a c o v á n í :

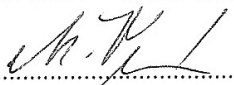
Cílem práce je vytvoření pohyblivé naklápěcí plošiny určené k záznamu pohybu živočichů. Dále je cílem práce vytvoření metod kvantitativního hodnocení kinematických dat pro studium funkce nervové soustavy. Pro pohyblivou naklápěcí plošinu vytvořte algoritmy řízení přesného pohybu a záznamu kinematických dat pohybu segmentů těla živočicha. Při výběru metod vycházejte z metod používaných k měření a hodnocení pohybu lidského těla. Algoritmy přesného řízení plošiny a záznamu pohybu, a jeho hodnocení, implementujte v prostředí Matlab. V rámci práce dále realizujte měření rozličných druhů živočichů ve spolupráci s PřF UK a FM Motol. Využitím naměřených dat proveďte vyhodnocení zjištěných ukazatelů určených navrženými metodami, a proveďte statistická zhodnocení výsledků experimentálních měření kinematických dat pohybu živočichů.


Seznam odborné literatury:

- [1] KUTÍLEK Patrik, Adam ŽIŽKA, Vybrané kapitoly z experimentální biomechaniky, ed. 1, Praha: nakladatelství ČVUT, 2012, ISBN 978-80-01-04993-8
- [2] MADGWICK, Sebastian O.H., Andrew J.L. HARRISON, Paul M. SHARKEY, Ravi VAIDYANATHAN a William S. HARWIN, Measuring motion with kinematically redundant accelerometer arrays: Theory, simulation and implementation, Mechatronics, ročník 23, číslo 5, 2013, 518-529 s., DOI: 10.1016/j.mechatronics.2013.04.003.
- [3] SCIARINI, Lee W. a Denise NICHOLSON, Foundations of Augmented Cognition. Neuroergonomics and Operational Neuroscience, ed. 1, ročník 5638, kapitola 5, 2009, Springer Berlin Heidelberg, ISBN 978-3-642-02811-3
- [4] ZHOU, Huiyu, Thomas STONE, HU, Use of multiple wearable inertial sensors in upper limb motion tracking, Medical Engineering, ročník 30, číslo 1, 2008, 123-133 s., DOI: 10.1016/j.medengphy.2006.11.010.

Vedoucí: doc. Ing. Patrik Kutílek, MSc., Ph.D.
Konzultant: Ing. Stanislav Kušmírek, as. MUDr. Rudolf Černý, CSc. (2.LF UK), doc. RNDr. Daniel Frynta, Ph.D. (PřF UK)

Zadání platné do: 20.08.2017


.....
vedoucí katedry / pracoviště


.....
děkan

V Kladně dne 07.12.2015

Poděkování

Chtěl bych poděkovat svému vedoucímu panu Doc. Ing. Patriku Kutílkovi, Ph.D. za vedení celé práce a pomoc při řešení jednotlivých problémů, které se během našich měření vyskytly, a dále Bc. Jánů Hýblovi, který uskutečnil podstatnou část návrhu a pomohl s konstrukcí popsané naklápěcí plošiny ve spolupráci se SPŠ a VOŠ Kladno a za jeho přístup a kooperaci při řešení softwarové komunikace mezi plošinou a MatLabem®. Poděkování patří též panu Doc. RNDr. Danielu Fryntovi Ph.D. za vymyšlení koncepce projektu a jeho realizaci na půdě naší fakulty. Dík patří rovněž Aniko Kurali Ph.D. za výpomoc při měření se zvířaty a Bc. Tereze Hášové za jejich zajištění z Přírodovědné fakulty a dovoz na místo měření, jakož i asistenci při samotném měření, na které se rovněž aktivně podílely studentky Jana Dušková a Julie Tomišková, a student Martin Eberl, za což rovněž děkuji.

Prohlášení

Prohlašuji, že jsem diplomovou práci s názvem **Návrh pohyblivé plošiny a metod hodnocení pohybu segmentů těla živočichů** vypracoval pod vedením Doc. Ing. Patrika Kutílka, Ph.D. samostatně a použil k tomu úplný výčet citací použitých pramenů, které uvádím v seznamu přiloženém k závěrečné zprávě.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č.121/2000 Sb. O právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

v *Kladně* dne *16.5.2016*

Škoda
.....

podpis

Obsah

1	ÚVOD.....	1
2	SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY	2
2.1	DYNAMICKÉ NAKLÁPĚCÍ PLOŠINY	2
2.2	METODY MĚŘENÍ A HODNOCENÍ POHYBU DROBNÝCH ŽIVOČICHŮ	6
2.3	ZÁVĚR.....	7
3	CÍLE A HYPOTÉZY	8
4	NÁVRH HARDWAROVÉ ČÁSTI PLOŠINY.....	9
4.1	POPIS KONSTRUKČNÍHO ŘEŠENÍ NAKLÁPĚCÍ PLOŠINY	9
4.1.1	POPIS HLAVNÍCH KONSTRUKČNÍCH ČÁSTÍ.....	9
4.1.2	HLAVNÍ ELEKTRONICKÉ PODSYSTÉMY	11
4.2	AKTUÁTORY POHYBU PLOŠINY	13
4.2.1	REŽIM ZAPOJENÍ	13
4.2.2	NASTAVENÍ KROKŮ.....	13
4.2.3	NASTAVENÍ ŘÍDÍCÍHO PROUDU	15
4.2.4	NASTAVENÍ STANDBY PROUDU	15
4.2.5	POHYBOVÉ ROZSAHY MOTORŮ.....	16
4.3	KAMEROVÝ PODSYSTÉM	16
4.4	FIXACE ŽIVOČICHA.....	18
4.5	ZÁVĚR KAPITOLY.....	18
5	ŘÍZENÍ POHYBU PLOŠINY A ZÁZNAMU	20
5.1	METODY ŘÍZENÍ PLOŠINY	20
5.1.1	MANUÁLNÍ KROKOVÁNÍ	20
5.1.2	SEKVENČNÍ ŘÍZENÍ.....	21
5.1.3	AUTOMATICKÁ KALIBRACE.....	23
5.1.4	KOREKCE SINUSOVÉHO ZRYCHLOVÁNÍ A ZPOMALOVÁNÍ	24
5.1.5	POSTUPNÁ AKCELERACE A DECELERACE.....	CHYBA! ZÁLOŽKA NENÍ DEFINOVÁNA.
5.1.6	ZAPÍNÁNÍ A VYPÍNÁNÍ PLOŠINY	25
5.2	METODY ZPRACOVÁNÍ OBRAZOVÉ INFORMACE.....	26
5.2.1	FORMÁT NAHRANÝCH VIDEÍ.....	26
5.2.2	EXTRAKCE SNÍMKŮ	27
5.2.3	POSTPROCESSING	28
5.2.4	ALGORITMY DETEKCE MARKERŮ	30
5.2.5	VÝPOČET ÚHLU.....	35
5.2.6	EXPORT DAT DO VÝSTUPNÍCH GRAFŮ A TABULEK	35

5.2.7	POPIS NAVRŽENÉHO GUI	37
5.2.8	FILTRACE POHYBOVÉHO ZKRESLENÍ.....	40
5.3	KOMUNIKACE MEZI PC PŘI NAHRÁVÁNÍ	41
5.4	ZÁVĚR KAPITOLY	43
6	<u>EXPERIMENTÁLNÍ TESTOVÁNÍ PLOŠINY</u>	44
6.1	VOLBA MĚŘENÝCH ŽIVOČICHŮ	44
6.2	VOLBA SLEDOVANÝCH UKAZATELŮ.....	44
6.3	ZPRACOVÁNÍ NAMĚŘENÝCH DAT	46
7	<u>HODNOCENÍ VÝSLEDKŮ EXPERIMENTÁLNÍ ČÁSTI</u>	48
7.1	UKÁZKA VÝSLEDNÝCH GRAFŮ ČASOVÉ ZÁVISLOSTI ÚHLU PLOŠINY A HLAVY ZVÍŘETE A DIAGRAMŮ VZÁJEMNÉ ÚHLOVÉ ZÁVISLOSTI	48
7.2	UKÁZKA EXPORTOVANÝCH TABULEK.....	50
7.3	HODNOCENÍ FORMOU BOXPLOTŮ.....	CHYBA! ZÁLOŽKA NENÍ DEFINOVÁNA.
8	<u>ZÁVĚR</u>	57
8.1	NAPLNĚNÍ CÍLŮ PRÁCE	57
8.2	OVĚŘENÍ HYPOTÉZ	58
8.3	PERSPEKTIVY DALŠÍHO VYUŽITÍ NAVRŽENÝCH METOD	58
	<u>SEZNAM POUŽITÉ LITERATURY</u>	60
	<u>SEZNAM POUŽITÝCH ZKRATEK.....</u>	62
	<u>SEZNAM OBRÁZKŮ</u>	63
	<u>SEZNAM GRAFŮ</u>	64
	<u>SEZNAM TABULEK</u>	65

1 Úvod

V současné lékařské praxi se vyskytuje mnoho případů, kdy člověk měří odezvu jiného člověka, tj. pacienta, na změnu orientace v prostoru či translační pohyb podložky, na níž subjekt stojí. Uvedená metoda se označuje jako „dynamická postulografie“ a klade si za cíl mimo studium vestibulárního či svalově-kosterního systému také studium nervové soustavy pomocí vyvolávání silových změn působících na subjekt (*Teszler, 2000*). Tyto síly nutí vyšetřovanou osobu udržovat rovnováhu a její nervový systém vysílá signály kosternímu svalstvu, které zpětně zajišťuje takovou reakci těla jako celku, aby subjekt znovunabyl narušené stability. Povětšinou se používají jednoduché plošiny, na nichž subjekt stojí, a které umožňují pohyb nebo náklon v jednom či více směrech, případně kolem jedné či více os. Finančně nejnáročnější řešení zahrnuje složité robotické systémy s několika aktuátory - plošiny s možností náklonu kolem všech 3 os ve 3-D prostoru, popřípadě translační pohyb ve všech 3 osách kartézského systému souřadnic. Mezi takové plošiny patří například systém *6DOF2000E (MOOG, East Aurora, NY, USA)*, který je vybaven rovněž laserovými snímači či celistvým kamerovým systémem pro záznam pohybu vybraných segmentů těla. Zmíněná plošina zajišťuje volnost rotace kolem 3 na sebe kolmých os a rovněž umožňuje měřit navíc i kontaktní síly pod chodidly subjektu (*Luu, 2011*), (*Tsai, 2014*).

Pakliže můžeme uplatnit metody z dynamické postulografie na člověku, potom nám nic nebrání uvedené aplikovat i na zvířata. I takové studie existují, nicméně zpravidla se nejedná o klinické či medicínské aplikace pro veterinární praxi, ale práce, sloužící primárně vědeckým účelům. Nejčastěji se u takovýchto experimentů setkáme s analýzou savců, méně potom obojživelníků, plazů a v neposlední řadě ptáků, jejichž úchyt na plošiny je bezesporu sám o sobě dramatickou záležitostí.

Tato práce je zaměřena na řešení problémů a návrh plošiny pro postulografické studie drobných živočichů. Zabývá se designem a konstrukčním řešením samotné plošiny, jakož i vhodným umístěním kamerového systému. V experimentální části je dopodrobna popsán navržený software k řízení plošiny, způsob záznamu videí z kamerového systému a na jejich základě prováděná analýza poloh segmentů živočicha v prostoru. Nakonec se v práci po návrhu metod věnuje hodnocení a kvantifikaci naměřených dat a jejich statistickému zpracování na měřeném testovacím souboru čítajícím 30 jedinců *gekončika nočního*.

2 Současný stav řešené problematiky

V rámci komentované rešerše současného stavu problematiky jsem se pokusil zjistit aktuální vědecké výsledky vědeckých týmů, které se již zabývaly měřeními na zvířatech za využití naklápěcích nebo jinak pohyblivých plošin. Několik výsledků, reprezentujících současný stav řešené problematiky, uvádím společně s odkazem na podrobnější informace.

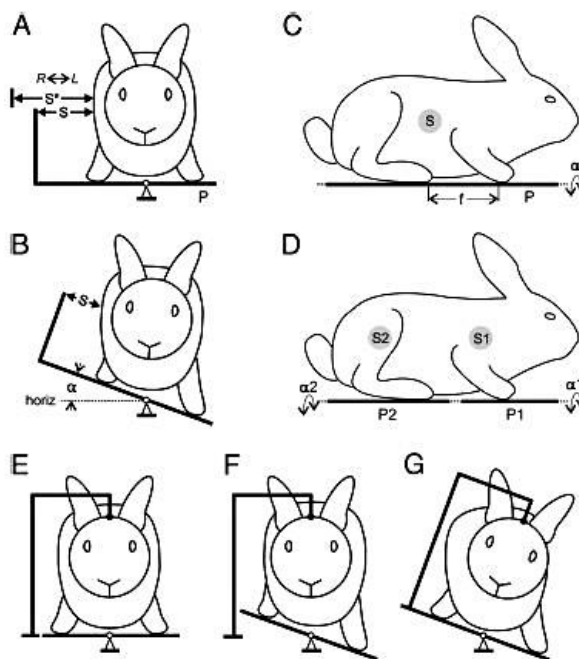
2.1 Dynamické naklápěcí plošiny

Jeden z vůbec prvních experimentů zahrnoval naklápění živočicha v lidské ruce, přičemž záznam pohybové odezvy byl měřen pomocí pořizování fotografických snímků v pevně stanovených intervalech (*Heath, 1969*). S ohledem na tehdy dostupné prostředky byla studie určitě průlomová – v dnešní době však vyvolá na tváři letmý úsměv při pomyslení, že by se stejným způsobem realizovala i v současnosti. Je zřejmé, že si dnes se současnými pokročilými technologiemi můžeme dovolit konstrukci výrobně složitějších robotických naklápěcích systémů a možnost sehnat miniaturní kameru prakticky v každém specializovaném obchodě situaci rovněž usnadňuje.

Modernější metodou je tedy studium pohybu ze záznamu videa, pořízeného kamerovým systémem. Subjekt je osazen speciálními markery (značkami), jejichž polohu v prostoru sledujeme, a je umístěn na pohyblivé plošině, která umožňuje rotaci kolem minimálně jedné osy. Přestože existují systémy, umožňující rotaci kolem více os, studium je zpravidla prováděno při rotaci kolem jediné z nich. Příkladem výše zmíněného řešení je právě studie na králících (*Beloozerova, 2003*). Použité markery byly kontrastní, a byly umístěny na krku, hlavě a podél páteře zvířete. Jejich záznam byl prováděn jednou kamerou ze vzdálenosti 2 m, přičemž analýzu jednotlivých snímků videa bylo nutno provádět ručně z důvodu omezených možností zkonstruované plošiny. V této studii autoři naklápěli zvíře ve frontální rovině. Pohyb plošiny byl uskutečňován buďto po sinusové trajektorii (s udanou frekvencí 0,5 – 1,0 Hz, avšak autoři dále neuvadli, co se zde přesně „frekvencí“ myslí), nebo po lichoběžníkové dráze (rampy) od +20° do -20°. Srst na králících byla v místech umístění markerů vyholena a testování probíhalo buďto za úplné tmy, nebo po zaslepení králíků sádrovým odlitkem.

Dále autoři článku testovali reakci králíka na elektrickou stimulaci vestibulárního aparátu. Uvedeno bylo, že aplikovaný proud o hodnotách 2-3 mA nebyl pro zvíře nebezpečný a nebyl zvířetem pocitově jakkoli vnímán. Elektrody pro aplikaci proudu byly umístěny bilaterálně na mastoideální výběžky lebky. Nakonec byla hlava zvířete zafixována k rámu plošiny a autoři zkoušeli plošinou naklápět za fixace hlavy v jedné poloze, přičemž plošina pod chodidly králíka se naklápěla, a dále při variantě, kdy se rám naklápěl společně s plošinou.

Uvedené metody studie považuji osobně za drastické a výsledky takových experimentů za diskutabilní, protože jejich zkreslení ovlivněním zvířete takto uzpůsobeným měřením dosahuje zcela nepřijatelné úrovně. Schéma experimentu ukazuje obrázek 2.1 vedle.

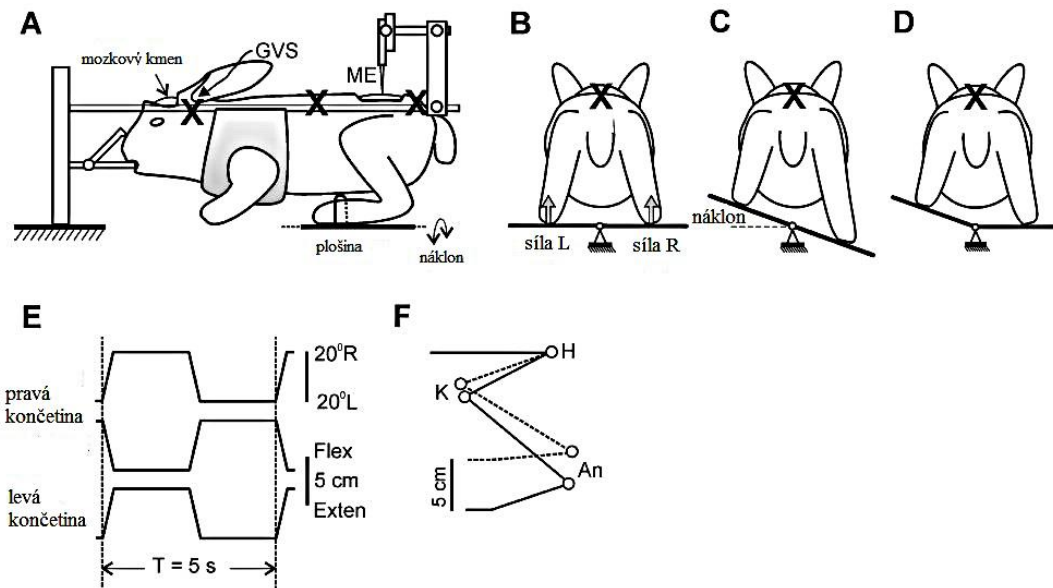


Obrázek 2.1: Návrh uspořádání experimentu na králících podle (Beloozerova, 2003); (A) odezva na stimulaci, (B) odezva na stimulaci a naklopení v prostoru, (E) hlava fixována k rámu, (F) otáčení plošiny při fixaci hlavy k nehybnému rámu, (G) otáčení plošiny při fixaci hlavy k rámu, jež se pohybuje spolu s plošinou.

Rozšířením zmíněné metody bylo přidání silových senzorů pod tělo králíka (Hsu, 2012). V uvedené studii se úhlové odchylky měřily pomocí goniometrů. Celá hlava, páteř a kyčel byly tentokrát zafixovány k nepohyblivému rámu. Přední končetiny byly volně vloženy do závěsné sítě a zadní byly v distálních bodech připevněny ke dvěma nezávislým naklápěcím plošinám – každá noha k jedné. Osa rotace obou plošin byla podél mediální osy králíka. Zvíře bylo podrobena laminektomii na obratlích L4 a L5 a do zadních končetin byly implantovány EMG elektrody pro snímání svalového potenciálu během měření. Autoři zkoumali pohybovou odezvu zadních končetin pomocí snímání EMG potenciálu při různém naklápění obou nezávislých plošin. Uspořádání experimentu je vidět na obrázku 2.2 na další straně.

Pokud jde o možnosti uchycení zvířete k plošině, v současnosti existují dvě varianty – buďto se upevní pomocí mechanických držáků segmentů těla (Lyalka, 2005), což má samozřejmě neblahý dopad na možnosti pohybu zvířete, anebo se povrch plošiny opatří protiskluzovým (drsným) povrchem a zvíře je vycvičeno, aby z plošiny neseskakovalo (Lyalka, 2009). Vycvičit ale nelze všechna zvířata, a tak přestože u některých savců (pes, kočka, králík) tento volnější způsob nečiní problémy, u obojživelníků se s tak poslušnými a chápavými subjekty zpravidla nesetkáváme, a takovýto způsob měření nelze realizovat.

Výše jsem se zmínil o možnosti měření kočky, dovolím si volně navázat studií na *kočce domácí* (Macpherson, 2007), (Deliagina, 2006), realizované na otočné naklápěcí plošině, využívající kamerový systém Vicon (VICON, Motion Systems, Inc, Lake Forest, CA), který se nacházel mimo plošinu a určoval polohu segmentů v prostoru pomocí sledování reflexních markerů na těle kočky. Tento systém byl rovněž vybaven snímači sil pod chodidly, ale navíc umožňoval kromě řízení velikosti úhlů náklonu plošiny též měnit rychlost náklonu ve stupních za sekundu.



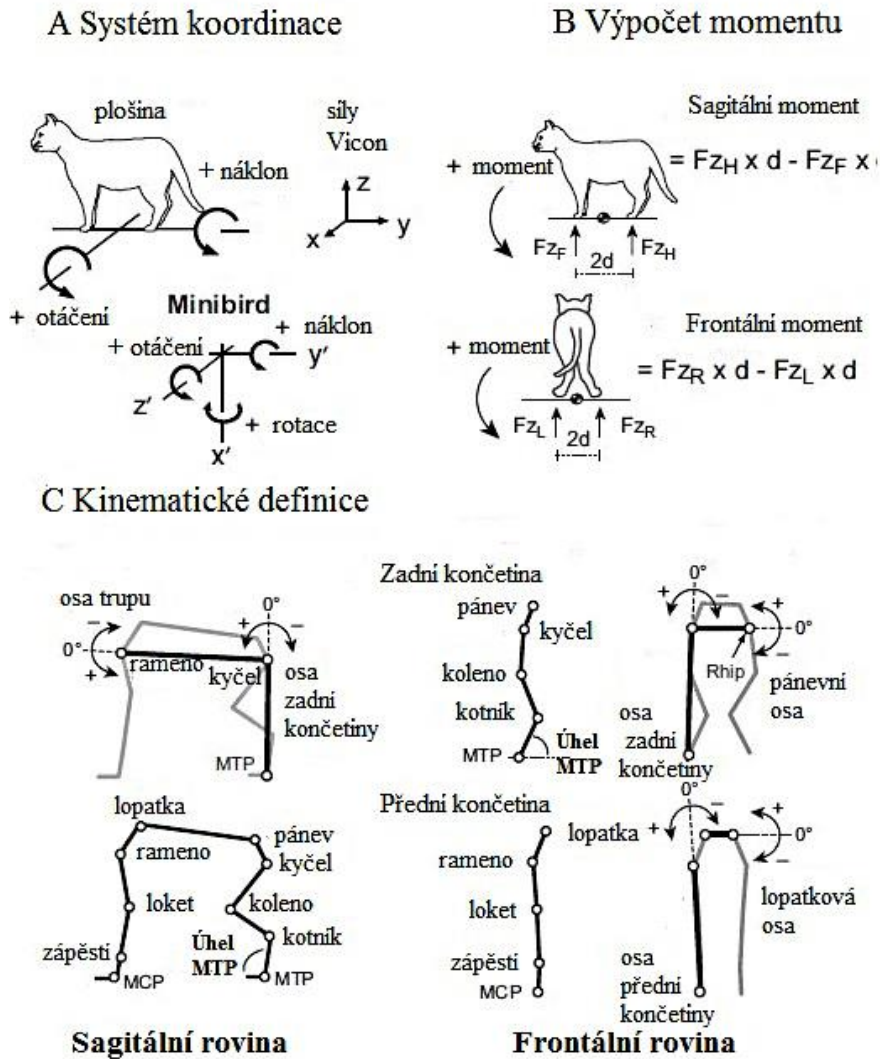
Obrázek 2.2: Uspořádání experimentu rozšířeného měření na králících (Hsu, 2012); (A) fixace hlavy, páteře a kyčlí k nehybnému rámu, (B) rovnovážná poloha, (C) náklon obou plošin, (D) variace náklonu plošin.

Kočky byly dlouhé měsíce způsobem odměn trénovány k tomu, aby na plošině stály v klidu. Poté byly uspány a bylo jim implantováno 16 elektrod do 16 různých svalů (extenzorů a flexorů končetin) pro sledování EMG aktivity. Následně byly podrobeny experimentu, při kterém stály na plošině, která se otáčela v režimu rampy s úhlovou rychlostí dosahující až $40^\circ/\text{s}$. Sledována byla jak EMG odezva ve svalech, tak orientace segmentů v prostoru pomocí *Vicon* systému. Náklony plošiny byly realizovány jak ve frontální tak sagitální rovině kočky. Nákras experimentu ilustruje obrázek 2.3 na vedlejší straně.

Komplikovanější realizaci představuje zmíněný systém doplněný o možnost translačního posuvu (Ting, 2004). Nevýhodou je obtížnější určení přesné polohy živočicha, z důvodu externě umístěného kamerového systému (mimo pohybující se části plošiny). V této studii byly kočky opět trénovány ke klidnému stoji s užitím výše zmíněné metody (Macpherson, 2007), tentokrát však měla kočka pod každou tlapkou samostatnou plošinu se dvěma stupni volnosti v translaci v horizontální rovině a 2 stupni volnosti v rotaci v sagitální a frontální rovině. Po měsících tréninku byly rovněž uspány a byly jim implantovány elektrody zcela totožně, jako u (Macpherson, 2007). Během experimentu byly sledovány kontaktní síly pod tlapkami kočky.

Pokud se však chceme od savců oprostit a začít měřit například na obojživelnících nebo plazech, jedná se již o menší zvířata (pokud zrovna neměříme kajmany (Heath, 1969)), a konstrukční řešení jak plošiny, tak kamerového systému, musí být tomuto faktu patřičně uzpůsobeno. Při měření na malých subjektech (krysa, žába, želva, ještěrka apod.) již zpravidla nebylo, v rozporu s očekáváními, využíváno více než jednoho stupně rotace v prostoru. Většinou se volí pouze rotace kolem vertikální osy živočicha, kdy je snímací kamera umístěna nad živočichem a snímá celé měření seshora (Shinder, 2011). Jinou variantou je rotace kolem jedné z horizontálních os živočicha (Dieringer, 1982).

Dieringer však prováděl ve srovnání s výše popsanými experimenty úplně jiný pokus. Malým žábám promítal ve tmě projektorem na stěnu před nimi horizontální či vertikální bílé pruhy. S žábou na plošině otáčel v transverzální rovině a sledoval pohyby očních bulv.



Obrázek 2.3: Nákres k experimentu na kočce domácí (Macpherson, 2007).

V současnosti jsou z důvodu potřeby základního výzkumu funkce a evoluce vestibulárního systému či nervové soustavy subjektu stále více kladeny nároky na měření ještě menších živočichů o velikosti třeba 5 cm. V minulosti zařízení pro takové experimenty umožňovala rotaci kolem pouze jedné z os (Dieringer, 1983). Standardní Motion Capture (MoCap) systémy již nelze u tak malých živočichů použít, protože jsou markery buďto na zvíře příliš veliké, nebo je třeba posunout kamerový systém do bezprostřední blízkosti živočicha, kde narážíme na limity ostření samotných kamer a nemožnost přesného vyhodnocení polohy ve 3-D prostoru z důvodu konstrukčního řešení rozestupu kamer u komerčně dostupných kamerových systémů.

Je tedy zřejmé, že řešením je užití malých markerů a spojení naklápací plošiny s kamerovým systémem tak, že se kamerový systém stane součástí plošiny, čímž se značně redukuje vzdálenost mezi markery a objektivem kamery. V současnosti takový systém neexistuje. Rovněž je nutné pro kalibrační účely znát přesnou polohu základny plošiny v prostoru s ohledem na dnešní úzus a konvence. Dosud nebyl vyvinut systém, který by tuto

polohu určoval pomocí snímačů typu gyroakcelerometru či podobných – vždy bylo užito ručních úhloměřů případně algoritmického dopočtu polohy z řídicího softwaru, což nemusí vždy odpovídat skutečné poloze v prostoru. Výchozí poloha plošiny tak není nikdy korigována v souladu s dohodnutými konvencemi o přesné orientaci ve 3-D prostoru.

Všechny zmíněné nedostatky odstraňuje námi navržený systém. Naše naklápěcí plošina umožňuje rotaci v prostoru kolem všech tří os a kamerový systém, čítající dohromady 3 na sebe kolmo umístěné kamery, je součástí plošiny a naklápí se společně s ní, což nám umožňuje usuzovat na polohu segmentů v anatomickém souřadném systému i velmi malého živočicha. Uvedený systém byl zkonstruován v souladu s požadavky Přírodovědné Fakulty Univerzity Karlovy a 1. Lékařské Fakulty Univerzity Karlovy v Praze a je popsán v této práci.

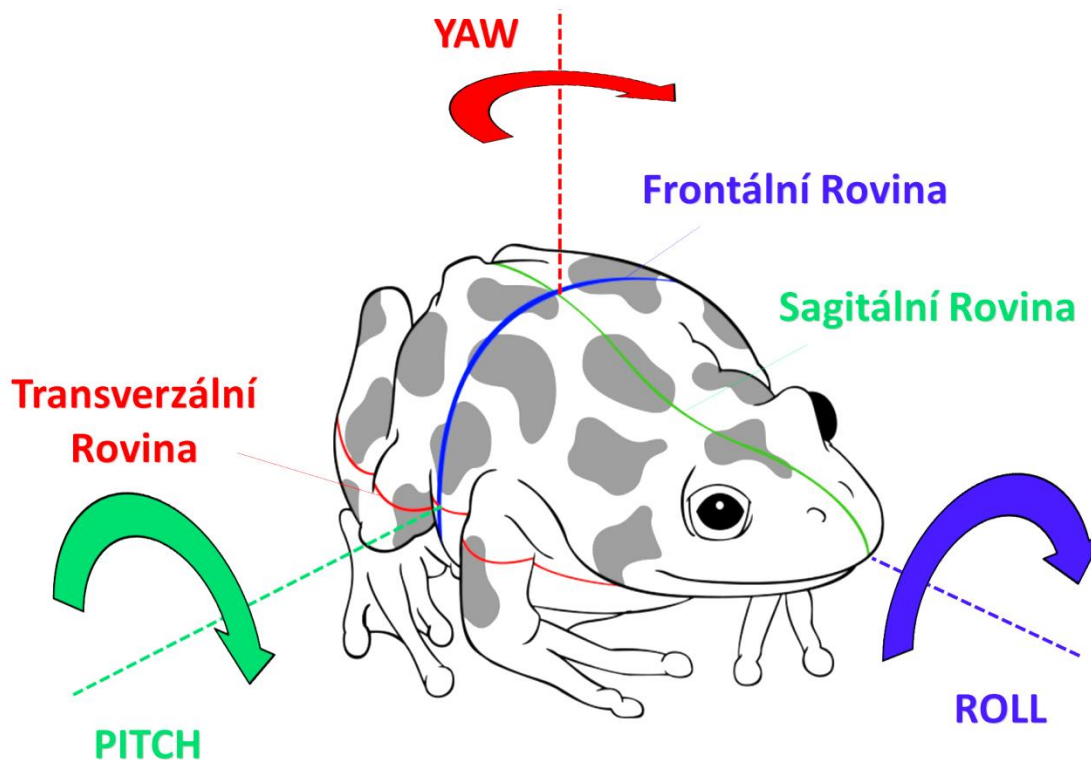
2.2 Metody měření a hodnocení pohybu drobných živočichů

V předcházejících studiích se především setkáme s periodickým rotačním pohybem plošiny během měření, kdy je pevně nastaven rozsah pohybu a rychlost plošiny, jakož i počet period, tedy doba měření. Rozsah pohybu při experimentálních měřeních býval nejčastěji od -20° do $+20^\circ$ (Beloozerova, 2003), (Hsu, 2012), (Lyalka, 2005), (Lyalka, 2009), zatímco úhlové rychlosti pohybu plošiny se volí až $40^\circ/s$ (Ting, 2004). Protože se se sinusovým pohybem plošiny můžeme rovněž v předchozích pracích setkat (Beloozerova, 2003), je na místě jej vyzkoušet také.

Možnosti úchytu zvířete na plošinu zahrnují buďto natrénování zvířete a použití protiskluzového povrchu plošiny (Macpherson, 2007), (Macpherson, 2004), nebo fixaci živočicha k plošině (Beloozerova, 2003), (Hsu, 2012).

Z výše uvedeného plyne, že by bylo vhodné, aby naše plošina měla celkem tři stupně volnosti – 3 rotace okolo na sebe vzájemně kolmých os. Je na místě zde uvést, že této volnosti je dosaženo kaskádovým seřazením tří aktuátorů. To, že na sebe jednotlivé rozměrově nenulové motorové prvky nasedají, způsobuje, že již nelze hovořit o rotacích živočicha kolem jeho anatomických os (jediná rotace, která dané tvrzení ještě splňuje, je v transverzální rovině, poněvadž osa otáčení v tomto případě vždy prochází středem plošiny a tedy i živočichem). Z tohoto důvodu je vhodnější zavést značení Eulerovských rotací dle Tait-Bryena - *YAW* (změna azimutu), *PITCH* (elevace) a *ROLL* (rotace kolem hlavní osy) (Škoda, 2014), viz obrázek 2.4 na vedlejší stránce. Výše zmíněné studie se zaměřily vždy na hodnocení reakcí na rotaci podél pouze jedné osy – i my se zdržíme hodnocení zbylých dvou a jako nejnápravděnější zvolíme *PITCH*, který je rovněž často hodnocen (Beloozerova, 2003).

Pokud jde o současný stav technologií, je možné například určit vývoj zmíněných parametrů v závislosti na periodách a říci, zdali se živočich na periodický pohyb adaptuje, či nikoli. Zajímavě by však mohly vypadat i diagramy vzájemné závislosti úhlu segmentu a úhlu plošiny pro jednotlivé periody. Tyto diagramy by se za předpokladu normálního rozdělení dat daly hodnotit například plochou konvexní obálky (Schubert, 2012) či plochou konfidenční elipsy (Bolbecker, 2011). V případě jakéhokoliv rozdělení je však jisté, že můžeme hodnotit délku trajektorie (Kim, 2009) v takových diagramech pro jednotlivé periody a rovněž sledovat její vývoj.



Obrázek 2.4: Demonstrace anatomických rovin zvířete a sledovaných rotací dle Tait-Bryenovy konvence značení.

2.3 Závěr kapitoly

Jak je vidět, měření s naklápečí plošinou na zvířatech není v naší práci historicky prvním. Neexistuje však celistvý systém pro naklápění a měření pohybové odezvy zároveň. V žádné z předchozích prací nebyl kamerový systém součástí systému naklápečího a nikde dosud nebylo na takovém systému měřeno na živočiších drobných rozměrů. Jelikož se ve světovém měřítku bude jednat o vůbec první experimentální měření podobného charakteru, je zbytečné věnovat se složité analýze vícero segmentů. Jde nám především o ověření funkčnosti systému a navržených metod. Pro naše měření bude v souladu s předcházejícími studiemi jako reprezentativní segment zvolena hlava živočicha a bude provedeno několik testovacích měření pohybu v sagitální rovině k ověření funkčnosti plošiny a vyzkoušení zpracování naměřených dat navrženým *GUI* (Graphical Users Interface – Grafické uživatelské rozhraní). V rámci rešerše jsem totiž nezjistil, že by zatím bylo vytvořeno vhodné GUI pro hodnocení pohybu segmentů drobných živočichů. Stejně tak jsem nenašel, že by byly navrženy vhodné algoritmy automatizovaného záznamu a případně hodnocení pohybu segmentů těla. I touto problematikou se tudíž bude vhodné v návrhu zabývat.

3 Cíle a hypotézy

Cílem je vytvořit naklápěcí plošinu pro malé živočichy se třemi rotačními stupni volnosti. Součástí plošiny bude kamerový systém sestávající ze tří na sebe vzájemně kolmých kamer, sledujících jednotlivé roviny živočicha. K plošině bude vypracované uživatelské rozhraní v MatLabu®, umožňující plošinu precizně řídit a nahrávat z kamer záznamy. Rozhraní rovněž umožní uživateli videa zpracovat a vyexportovat vývoj mezimarkerového úhlu v čase. Měření bude realizováno na výběrovém souboru malých živočichů, poskytnutých Přírodovědnou fakultou Karlovy univerzity v Praze. Účelem měření je srovnat reakce subjektů na pohyb plošiny, a to jak v rámci jednoho druhu, tak i mezidruhově.

Dlouhodobým cílem studie je nalézt ve vzorech chování živočichů společné znaky a prokázat tak (alespoň částečně), že je možná existence dědičné paměti, kdy přestože se jedná o zcela odlišné skupiny živočichů (například obojživelníci versus plazi), všichni na předpřipravené scénáře pohybu plošiny reagují stejně, nebo velmi obdobně.

Hypotézy k verifikaci jsou tedy následující. Zaprvé dokázat, že existují rozdíly v chování měřených subjektů v čase, tj. jejich reakce se mění při periodickém pohybu mezi jednotlivými periodami (subjekt si navyká na pohyb plošiny a uzpůsobuje mu své chování). Zadruhé – prokázat odlišnosti mezi chováním zvířete před stimulací a po stimulaci, jež má napodobovat útok predátora.

4 Návrh hardwarové části plošiny

Následující část se zabývá popisem návrhu naklápěcího a kamerového systému plošiny, jakožto i jeho jednotlivých součástí. Kapitola je rozdělena na stacionární mechanické části, dále aktuátory (motory) a nakonec zvláště je popsán kamerový systém a způsoby fixace živočichů k plošině.

4.1 Popis konstrukčního řešení naklápěcí plošiny

4.1.1 Popis hlavních konstrukčních částí

Základnu navržené a zkonstruované experimentální naklápěcí plošiny (dále jen ERP – Experimental Revolvable Platform) tvoří prostor ohraničený aluminiovým pláštěm (slzičkový plech tloušťky 1 mm), ve kterém se nachází veškerá řídicí a signalizační elektronika. Celkový výkon systému je zhruba okolo 800 W, přičemž řídicí elektronika odebírá 3,0 A z celkového vstupního proudu (při 230 V ze sítě) a zbylých ~0,5 A odebírá signalizace a diody. Schéma uspořádání jednotlivých komponent v základně je možné vidět na obrázku 4.1C na dalším listu.

Zde černě vyznačené moduly nahoře reprezentují převodní elektroniku, dodávanou s jednotlivými motory, mezi uživatelským vstupem a posuvem krokových motorů, červené donuts jsou oddělovací a filtrující toroidní transformátory. Tři zelená pole pod nimi obsahují usměrňovací a filtrující obvody, které napětí z výstupu transformátorů dále zpracovávají do podoby kompatibilní s černými moduly pro řízení motorů. Před každým toroidem je ochranná pojistka proti přepětí. Ochranné pojistky jsou umístěny rovněž za každým toroidem a za každým usměrňovačem. Šedý modul reprezentuje spínaný zdroj pro veškerou zbylou elektroniku v modrém zbývajícím panelu na schématu. Zde se zleva doprava nacházejí – obvod pro periférii a senzory (LEDky na čelním panelu, u dorazových senzorů na plošině apod.), AT90USB logický obvod s mikroprocesorem pro komunikaci mezi motory a MatLabem® a nakonec úplně vpravo se nachází relátka pro postupné připojování toroidů, aby nebyly při okamžitém zapnutí plošiny kladeny příliš vysoké nároky na síť a zařízení tak nevyhazovalo v provozní místnosti jističe. Z uvedeného plyne, že i motory jsou po zapnutí plošiny inicializovány postupně, a to ve sledu od nejnižší uložené po nejvyšší.

Uprostřed základny se vzhůru tyčí nosník, na kterém jsou kaskádně umístěny jednotlivé motory a senzory a na nich se pak nachází naklápěcí plošina s nosným aluminiovým rámem pro umístění nahrávacího kamerového systému.

Čelní panel je opatřen zámkem na klíč ke spuštění plošiny a celkem 9 signalizačními diodami pro jednotlivé motory ve 3 barevných vyvedeních (zelená, žlutá, červená). Dále je zde zapínač dodatečného diodového osvětlení a jeho regulátor (viz oftalmoskopické C-rameno níže) a nouzový vypínač.

Celá konstrukce byla před uzavřením ochranným krytem přetažena gumovou izolační vrstvou pro případ rozlití kapalin rozličného charakteru (IP62) a pro ochranu před usazováním prachu uvnitř konstrukce (IP60). Celý kryt je z důvodu ochrany před vyskytnutím se případného nežádoucího unikajícího proudu na povrch přístroje uzemněn.

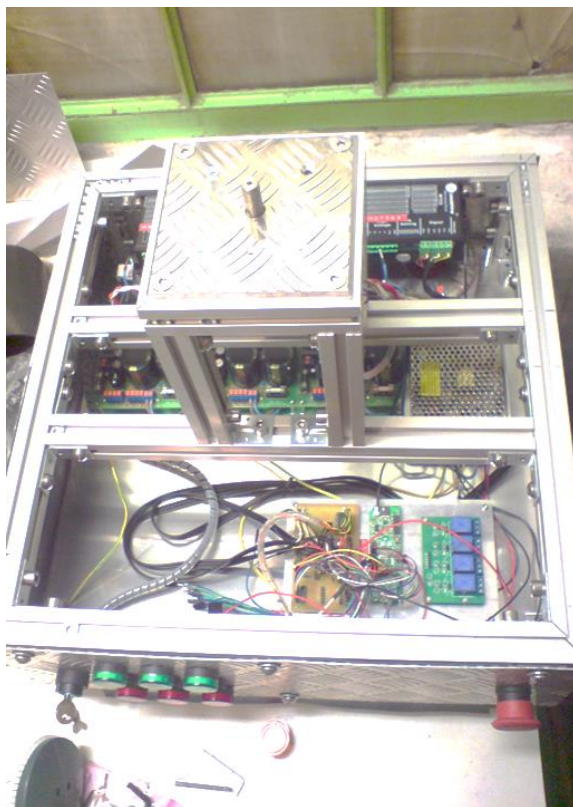
S ohledem na požadavky Fakultní Nemocnice Motol bylo k základně plošiny přidáno oftalmoskopické C-rameno k vyměřování rozsahu zraku studovaných zvířat. Rameno se ze zadu

zvedá a nad plošinou se láme směrem dopředu, kde se kolmo rozdvouje na dvě protichůdné větve, které pokračují dolů a končí lehce nad úrovní plošiny ve výchozí pozici. Z těchto rohových pozic je možné při přiložení oftalmoskopu a otáčení plošiny kolem vertikální osy vyměřit rozsah záběru zraku subjektu.

Vzestupná část ramena rovněž slouží jako základna pro symetricky rozmístěné osvětlovací diody dvou barev (bílá a oranžová), jejichž intenzitu lze regulovat otočným prvkem na čelním panelu plošiny. Na jednom sestupném rameni je umístěn digitální teploměr pro monitoring teploty, jelikož některá zvířata jsou náchylná na teplotu a její změny. Nakonec C-rameno slouží jako závěsná konstrukce pro naklápěcí destičku ve vypnutém stavu.

Samotná platforma se nachází v hierarchii prvků na středovém nosníku ve prostřední základny nejvýše. Tvoří ji rám, ve kterém je ukotvena aluminiová destička o rozměrech 25x20 cm s jedním stupněm volnosti v předozadní translaci. Deska je pokryta suchým zipem, na který nasedá průhledná krabička s měřeným živočichem.

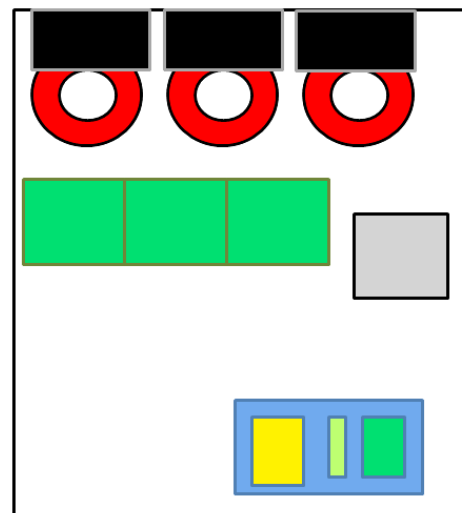
V případě měření živočichů, jejichž chování není ovlivněno magnetickým polem, je k dispozici rozšíření v podobě zajištění kontaktu měřicí krabičky s plošinou čistě na bázi



(A)



(B)



(C)

Obrázek 4.1: (A) Fotografie odkryté základny plošiny, (B) detail převodníků informace z AT90USB na pohyb motorů, skrytých na A za nosníkem pro motory a naklápěcí desku, (C) schéma umístění jednotlivých komponent v základně ERP, popis viz text.

magnetické interakce, což nám umožňuje krabičku s živočichem během měření libovolně otáčet kolem vertikály a zaručit tak kolmost os kamer na jednotlivé roviny zvířete uvnitř krabičky, a to dle požadavků výzkumného pracovníka (více v kapitole 4.4 – *Fixace živočicha*).

4.1.2 Hlavní elektronické podsystémy

Na každý motor připadá jeden jeho ovladač M542, nebo M752 (černé moduly na schématu na obrázku 4.1C). Tyto krabičky převádějí signál z mikrokontroléru na pohyb motorů. Manuálně se na nich posuvníkem dá dle datasheetu nastavit hustota kvantování *mikrokroků* (též *mezikroků*), a to až na 51200 mikrokroků na 360°. Obdobným způsobem lze nastavit vstupní proud, ovlivňující maximální záběr motorů. Jelikož se jedná o bipolární krokové motory, přesnější krok vyžaduje zapojení více cívek (mezikroky). Větší průtokový proud se však manifestuje zahříváním motorů. Je tedy třeba najít vhodný poměr mezi přesností a plynulostí pohybu motoru a výstupní teplotou.

Každý motor vyžaduje stejnosměrné napětí 230 V a proud přibližně 1 A. Z toho důvodu je třeba napájení ze sítě usměrnit a filtrovat. V každém zařízení zároveň nesmí chybět oddělovací transformátor – u naší plošiny k tomuto účelu slouží zmíněné toroidy (obrázek 4.1C – červené pole). O filtraci a další zpracování signálu pro ovladače motorů se starají filtrační a usměrňovací obvody (obrázek 4.1C – zelené pole). Je vhodné podotknout, že jeden z motorů se od zbylých dvou liší a liší se rovněž jeho vstupní proud a ovladač. Jedná se o motor, položený nejnižší. Na tento motor jsou kladeny jiné nároky, než na ostatní – nemusí mít tak silný záběr (tj. moment síly), protože těžiště celé konstrukce se ve většině případů nachází přímo v ose rotace motoru nebo poblíž, a navíc je u něj vyžadován jiný rozsah rotace (viz kapitola 4.2 - *Aktuátory pohybu plošiny*).

Řídící a signalizační elektronika vyžaduje o poznání nižší proudy a celkově si vystačí s přísunem dohromady do 0,5 A. I v tomto případě je třeba napětí ze sítě usměrnit a filtrovat - oddělovací transformátor rovněž nesmí chybět. Tento napájecí zdroj (obrázek 4.1C – šedé pole) napájí jednak veškeré LED na plošině, jednak veškeré pasivní senzory na plošině, a ještě k tomu pokrývá veškeré nároky řídicí jednotky AT90USB, která krom řízení ovladačů jednotlivých motorů přichází signály zpracovává a odesílá na zažádání uživateli.

Řídící mikrokontrolér od firmy Atmel - AT90USB - je typu PAC (Programmable Automation Controller), zaručující multiplatformní kompatibilitu a umožňující distribuci logických smyček mezi více jader, díky čemuž lze řešit komplexní úlohy synchronně bez časové návaznosti (*PAYNE Jeff, 2013*). Na rozdíl od PLC (Programmable Logic Controller), vhodného obecně pro jednodušší úlohy nachází v našem projektu PAC větší uplatnění, protože jeho flexibilita když dojde na charakteristiku I/O (Input/Output) dat nám umožňuje mezi sebou pevně kloubit MatLab, C++, data z akcelerometru, senzorů dorazu, kamer a dalších prvků.

Komunikační modul mezi softwarem pro řízení, programovaným v MatLabu, a ovladači k jednotlivým motorům je tvořen uživatelským programem vytvořeným pro účely řízení plošiny v C++.

Systém senzorů je tvořen celkem 6 dvojicemi magnetických pasivních senzorů. Jeden z páru tvoří jednoduchý magnet, který je vždy stacionárně připevněn k plošině, nebo tělu dalšího motoru tak, že vymezuje jednu z hranic rozsahu pohybu daného motoru. Druhý z páru je tvořen cívkou, na níž je snímáno napětí. Tato cívka je umístěna na pohyblivé části motoru

tak, že v případě dojezdu motoru k jedné z hranic dochází k interakci cívky s magnetickým polem magnetu, ohraničujícího pohyb daného motoru. Změna toku magnetického pole plochou průřezu cívky v čase způsobuje Faradayovu indukci napětí na cívce podle vzorce 1,

$$\varepsilon = -\frac{d\Phi}{dt} = -N \cdot B \cdot \frac{dS}{dt} \quad (1)$$

kde ε značí Faradayovo napětí (V), Φ magnetický tok (Wb), t čas (s), N počet závitů na cívce (-), B magnetickou indukci (Te) a S plochu překryvu plochy průřezu cívky a magnetu (m²).

Jinými slovy – pokud se nám začne na cívce indukovat napětí, znamená to, že se blížíme k hranici dojezdu motoru (zvyšuje se $d\Phi$, tedy dS). Sensory jsou pro všechny tři motory rozmístěny tak, že omezují rozsahy horních dvou od -23,4° do +23,4°, a spodního od -171° do +171°. V případě přetečení hodnoty napětí na senzoru přes určenou mez se motor zastaví a pošle do softwarem nastavené krajní pozice, čímž se zabrání nežádoucím poruchám.

Pro ověření správnosti výchozí pozice plošiny je na spodní stranu naklápěcí destičky instalován akcelerometr. Data z něj nám dávají informaci o naklopení destičky v prostoru, tedy informaci o přímém naklopení horních dvou motorů. Informaci o natočení v horizontální rovině z něj získáme díky jeho kompasovému modulu. Akcelerometr má sloužit pouze informativnímu účelu o správnosti výchozí pozice plošiny a její případné kalibraci.

Kamerový systém je tvořen třemi na sebe kolmo umístěnými komerčně dostupnými HD webkamerami typu Defender G-Lens s možností ostření od 2 cm. Přestože kamery umožňují zaznamenávat videa v 720p rozlišeních, utrpí tím maximální možný *framerate* (počet snímků za sekundu), který je pro nás důležitější. Uvědomíme-li si navíc, že jeden počítač zaznamenává hned tři videa souběžně, bylo nutno záznam PC z důvodu omezených HW prostředků co nejvíce usnadnit – rozlišení videí jsme tedy omezili na použitelné minimum 320x240 px.

Jak již bylo zmíněno výše – součástí oftalmoskopického ramene je série LED, umožňující uživateli kustomizaci nasvícení subjektu. Tato utilita vznikla v momentě, kdy jsme zjistili, že klasické zářivkové osvětlení na síťové frekvenci 50 Hz vrhá z důvodu porušení Nyquistova-Shannonova-Kotělnikova teorému do záznamu videí jasně viditelné blikání (vzorkování videí probíhá při framerate do 30 – často jen 12 FPS (Frames Per Second – ekvivalent *frameratu*)). Provoz v místnosti s více zdroji světla však uměle navyšuje frekvenci blikání zářivek a efekt blikání pro snížení pravděpodobnosti strefení sejmutí vzorku videa do „temného“ momentu téměř vymizí. Pro případy, kdy takový provoz není možný, je na plošině vyvedeno vlastní regulovatelné nasvícení v podobě zmíněných speciálních LED, které tímto neduhem netrpí. Aby bylo dále zamezeno nepříjemným odrazům světla LED od krabiček s měřenými živočichy, bylo nutné za kamery nacházející se mezi subjektem a diodami umístit stínítka.

Poslední z věcí, kterou je třeba zmínit je vyhřívací systém plošiny. Živočichové, se kterými jsme během měření přišli do styku, jsou citliví na teplotu – tu je proto nutné udržovat v pro subjekt přípustných mezích. Pod naklápěcí destičkou je tedy umístěn jednoduchý rezistor, přeměňující část proudu z napájecího zdroje pro elektroniku na teplo. Průtok proudu rezistorem lze regulovat a tím ovlivňovat míru vyhřívání plošiny.

4.2 Aktuátory pohybu plošiny

Jedná se o dva krokové motory typu 60STH88 s kroutícím momentem 4,5 Nm a jeden typu 86HS45 s momentem 3 Nm od společnosti Leadshine technology Co. Ltd., dodávající motory společně s ovladači (dva M542 a jeden M752).

4.2.1 Režim zapojení aktuátorové části

Stator u krokových motorů tvoří válcový plášť, podél něhož jsou proti sobě vyvedeny otvory s vodivými smyčkami. Uvnitř válce se nachází rotor, který si můžeme pro zjednodušení představit jako otočnou cívku nebo magnet (Ve skutečnosti jde o kovové ozubené kolo a celý design je o něco složitější). Orientace cívky je řízena změnami magnetického pole, tvořenými přepínáním proudů smyčkami na statoru. V závislosti na tom, zdali se jedná o unipolární či bipolární zapojení, je rotor otáčen na základě působení jedné či dvou protilehlých smyček statoru. Pořadí aktivních smyček na statoru se střídá v takovém sledu, že se rotor uvnitř otáčí. Pro řízení v bipolárním režimu je tedy nutné zapojit vždy o smyčku navíc, což dvojnásobně zvyšuje proudový odběr – na druhou stranu je rotor v poloze držen oboustranně, což zvyšuje kroutící moment motoru.

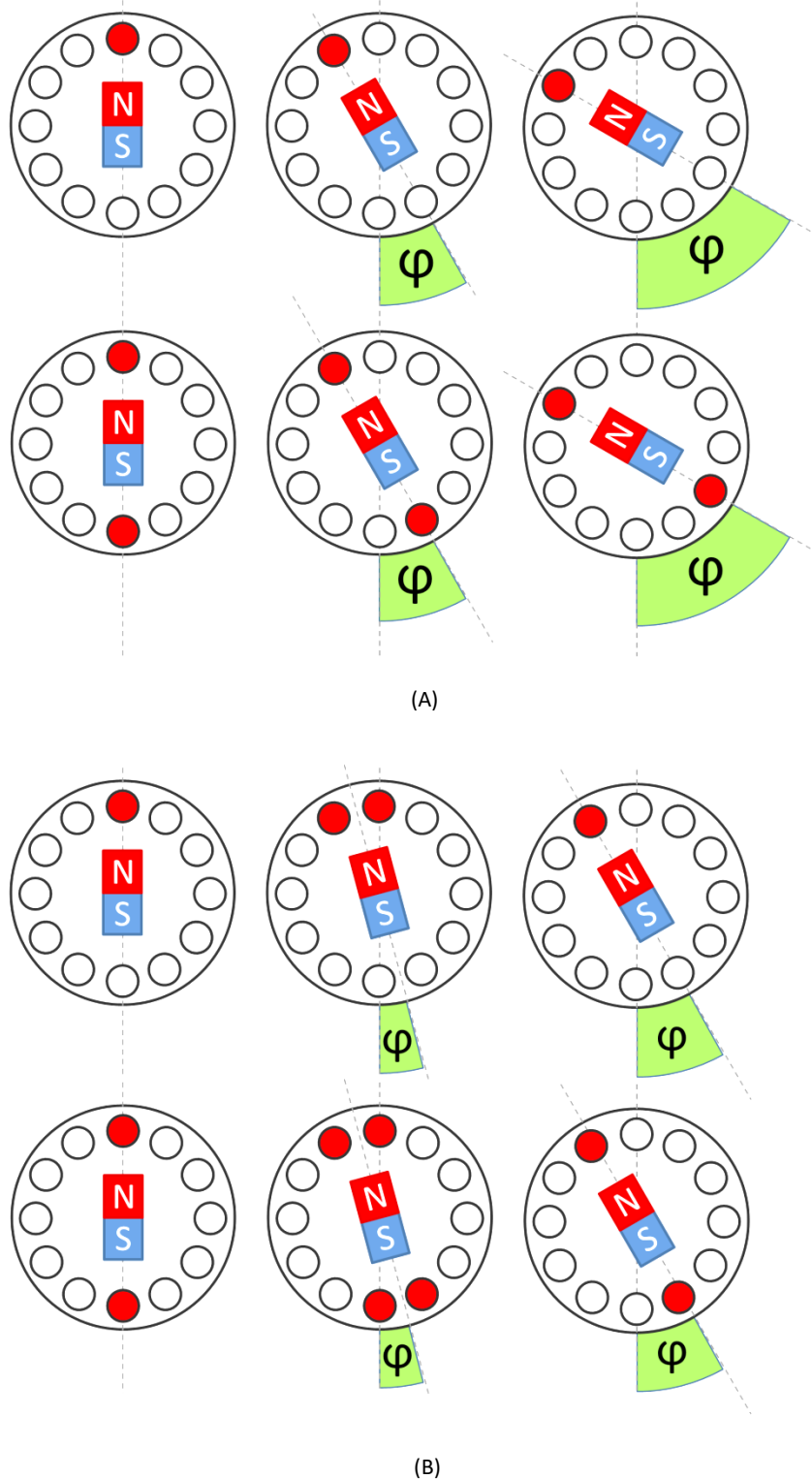
V případě, že popsaný vzor rozšíříme o jednu nebo více sousedních smyček navíc, můžeme rotor zastavit i v poloze mezi dvěma smyčkami na statoru. Takové řízení umožňuje zvýšit plynulost přechodů mezi jednotlivými kroky, ale vyžaduje daleko vyšší nároky na odběr proudu. Tento režim je možné rovněž provozovat v bipolárním či unipolárním zapojení – my používáme bipolární pro naši potřebu vyššího silového momentu. Princip činnosti a rozdíl mezi bipolárním a unipolárním zapojením, stejně jako rozdíl mezi krokem a mezikrokem je vyznačen na obrázku 4.2 na další straně. Rotor je symbolizován magnetem uprostřed statoru a zapojené cívky na obvodu statoru jsou značeny červenými kolečky.

4.2.2 Nastavení krokového pohybu motorů

Krokové motory umožňují díky svému designu zastavit hřídel i mezi cívkami. Těmto polohám, nacházejícím se mimo hlavní kroky, se říká mezikroky (viz obrázek 4.2B, další strana). V mezikrocích je však systém nucen setrvat v labilní rovnováze, což při zatížení být jen samotnou váhou naklápací plošiny není možné a tak pokusíme-li se zastavit plošinu v takové pozici, nutně nám asi sekundu po dojezdu sama spadne do nejbližšího stabilního kroku. Mezikroky nám tak pouze umožňují vyhladit pohyb plošiny díky zhuštění bodů, přes které se plošina na požadovanou polohu dostává. Zároveň čím větší počet mezikroků vyžadujeme, tím větší musíme odebírat proud, protože k dosažení tak citlivého nastavení mezikroků je nutno zapojit hned několik desítek cívek.

Naše motory mají společné minimum hlavního kroku na $1,8^\circ$, což činí 200 kroků na 360° , tedy každý krokový motor sestává z 200 cívek. Počet mezikroků na 360° byl u vrchních motorů nastaven na 25 600, což odpovídá 128 mezikrokům na jeden hlavní krok, tedy například mezi odjezdem plošiny v jedné rovině z polohy 0° na $1,8^\circ$ jede plošina přes 128 hodnot, což navenek působí efektem velmi plynulého pohybu.

Spodní motor je jiného typu – má vyšší záběrový moment a zároveň umožňuje rotaci přes celou kružnici. U něj jsme nastavili citlivost 51200 mezikroků na 360° , což má za následek



Obrázek 4.2: Princip činnosti krokového motoru. (A) – Unipolární posuv po plných krocích (nahore) a bipolární posuv po plných krocích (dole); (B) – Unipolární posuv po mezikrocích (nahore) a bipolární posuv po mezikrocích (dole).

dvojnásobnou plynulost při pohybu vzhledem k vrchním dvěma motorům. Při použití stejné analogie jako výše se plošina při pohybu z 0° na $1,8^\circ$ přesune přes celkem 256 mezipoloh.

4.2.3 Nastavení řídicího proudu motorů

Jedná se o proud, který doplňuje točivý moment motoru působící změnu natočení jeho rotoru v prostoru. Jelikož je každý pohyb plošiny bezpodmínečně provázen fází, kdy plošina zrychluje nebo zpomaluje, z druhého Newtonova zákona („*Pohybuje-li se těleso se zrychlením, je toto zrychlení přímo úměrné působící síle a nepřímo úměrné hmotnosti tělesa*“) plyne, že zde krom sil, nutných k udržení plošiny v nějaké stacionární poloze, vznikají ještě další síly, spojené s akcelerací a decelerací, které je třeba překonat.

Z uvedeného je zřejmé, že řídicí proud musí být vyšší, než proud udržující plošinu ve stacionární pozici. Tento proud lze na motorech nastavit a v případě vrchních dvou motorů je jeho vrcholová hodnota rovna 2,37 A (1,69 A RMS). U spodního motoru je vrcholový proud nastaven na 3,31 A (2,36 A RMS). *RMS* značí *Root Mean Square*, tedy doslova odmocninu ze středního čtverce. Definovat *RMS* lze rovněž jako teplo, které by při 100% disipaci příkonu odevzdal spotřebič užívající střídavý proud, kdyby užíval proud stejnosměrný o velikosti právě *RMS*. Výkon si v teorii signálů můžeme při vhodné volbě konstanty (odpor $R = 1$) představit jako druhou mocninu amplitudy výchylky, buďto napětí, nebo proudu, jak plyne ze vztahu 2. Střední výkon spočteme jednoduše integrací sinusového průběhu druhé mocniny napětí či proudu a podělením výsledku délkou intervalu, přes který jsme integrovali (vzoreček 3). Pokud by někoho mátl, že jsme *RMS* definovali přes energii, zatímco teď tady srovnáváme nějaký výkon, potom upozorňuji, že výkon je energie za čas. Zvolíme-li si čas nějak vhodně (například 1 s), je výkon totožný s energií, tedy oním teplem, užitým při definici *RMS*. Veličina *RMS* tak skutečně dostává svému překladu.

$$P = U \cdot I = U \cdot \frac{U}{R} = \frac{U^2}{R} ; \quad P = U \cdot I = R \cdot I \cdot I = R \cdot I^2 \quad (2)$$

$$RMS = \frac{1}{T} \cdot \int_t^{t+T} u(t)^2 dt = \frac{1}{T} \cdot \int_t^{t+T} i(t)^2 dt = \frac{1}{T} \cdot \int_t^{t+T} A^2 \cdot \sin^2(\omega t + \varphi) dt \quad (3)$$

Zde P značí výkon (W), U napětí (V), I proud (A), R odpor (Ω), *RMS* odmocninu střední hodnoty čtverce výchylky (V nebo A), T periodu (s), t čas (s), $u(t)$ napětí jako funkci času (V), $i(t)$ proud jako funkci času (A), A amplitudu (V nebo A), ω úhlový kmitočet (rad/s) a φ úhlový posun (rad). Po integraci dojdeme k závěru, že je *RMS* rovno maximální amplitudové výchylce, dělené odmocninou ze dvou.

4.2.4 Nastavení standby proudu motorů

V případě, že se plošina nepohybuje, je stále nutno, aby motory vyvíjely určitý kroučící moment pro udržení plošiny v její aktuální poloze. K tomu jsou motory napájeny tzv. „*standby*“ proudem, který je 60 % z řídicího proudu pro daný motor. Tento proud se významně podílí na zahřívání motoru, jelikož zhruba polovinu času během testování či měření na zvířatech plošina stojí na místě. Nesmí tedy být příliš vysoký, ale na druhou stranu ani příliš nízký – riskovali bychom zhroucení plošiny v krajních úvratích.

4.2.5 Pohybové rozsahy motorů

Oba horní motory, naklápějící plošinu v sagitální a frontální rovině zvířete, mají rozsah od $-23,4^\circ$ do $+23,4^\circ$. Teoretický rozsah je od -45° do $+45^\circ$, ale vzhledem k váze nástavby, kterou musí motory v krajních polohách překonávat, byl rozsah omezen téměř na polovinu. Experimentálně jsme prokázali, že při průměrném zatížení plošiny a současném nastavení odběru proudů obou motorů dochází při přesazení 32° na jakoukoliv stranu k vytvoření příliš velké ramene síly, které motory nejsou schopné kompenzovat, a plošina se neudrží v žádané poloze.

Nejnižší motor, rotující s plošinou kolem její vertikály, má teoretický rozsah od -180° do $+180^\circ$, nicméně i zde byl omezen na interval od -171° do $+171^\circ$ z důvodu vyšší bezpečnosti při provozu plošiny (obstrukce obtáčení kabelů kolem konstrukce).

4.3 Kamerový podsystém

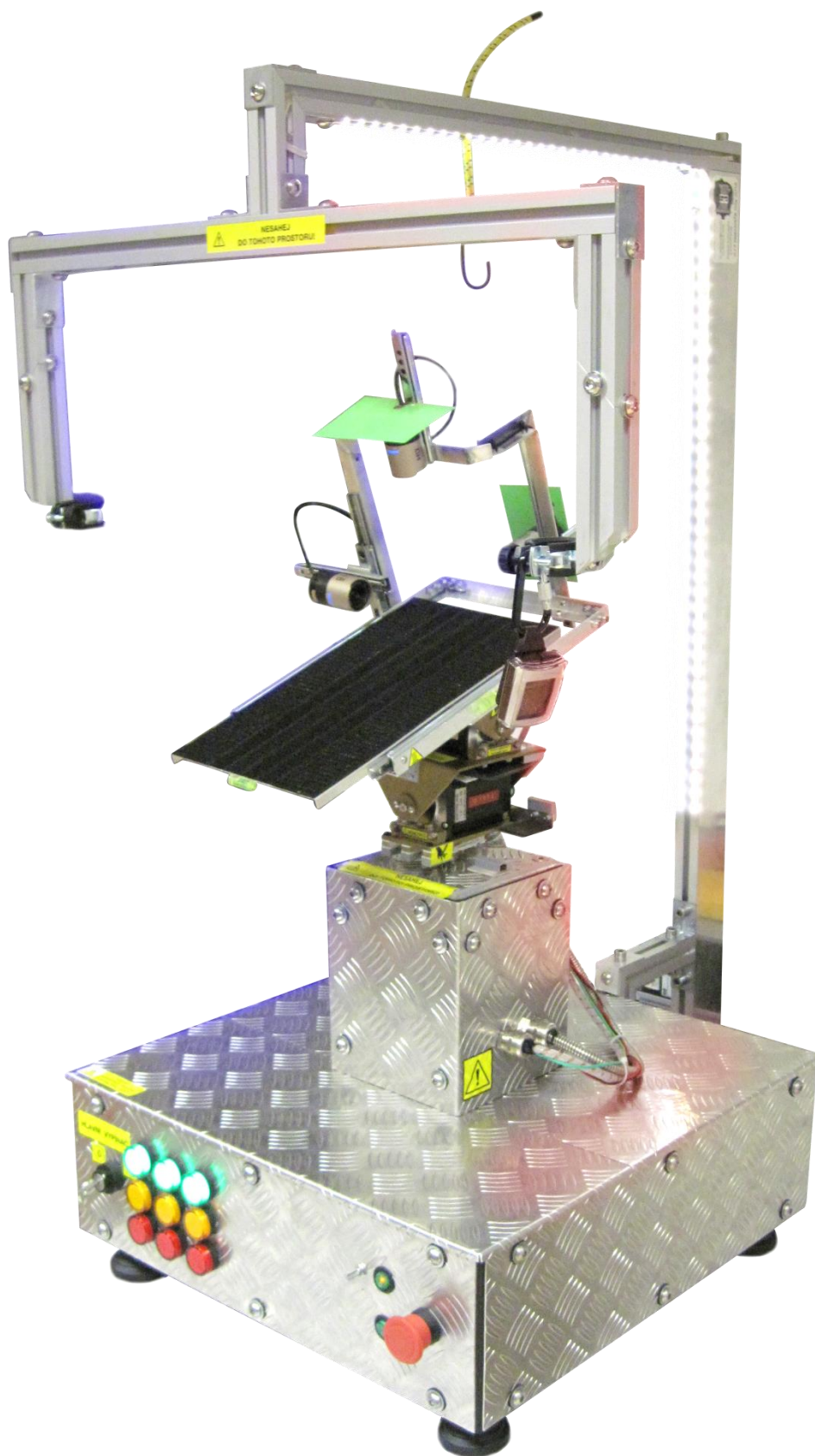
Každá z kamer snímá jednu rovinu živočicha a jejich umístění na hliníkový rám s pojezdy umožňuje dodatečný translační posun každé z kamer, u některých (sagitální rovina) i se dvěma stupni volnosti. Možnost translačního pohybu po ose kamery je zde nutností, protože ostření kamery není automatické a je třeba jej manuálně korigovat otáčením prstence na čele kamery, což při experimentu zkrátka provádět nelze. Jednoduchým nastavením ostřicí vzdálenosti do středu plošiny a následným popojížděním s kamerou v její ose lze rychle docílit zaostření na kteroukoliv část živočicha, přestože daň, kterou za tuto obchůzku platíme, je modifikace velikosti markerů na nahraném videu a tím úprava obtížnosti zpracování záznamu (prakticky se jedná o zoom in/zoom out). Hliníkový rám s pojezdy je určen pouze pro držení kamer v navzájem kolmé poloze, jakákoliv jiná aplikace síly na něj by mohla způsobit ztrátu integrity materiálu. Hliník byl zvolen proto, že se jedná o lehký pevný kov. Jelikož je celý rám i s kamerami součástí naklápěcí desky, bylo nutno rám zkonstruovat z co nejlépejšího materiálu, aby nám hmotnost rámu neovlivnila váhu naklápěcí destičky a nevedla k dalším problémům s nutností navýšení vstupního proudu k motorům za účelem zvýšení jejich kroutícího momentu k udržení celé konstrukce.

Fotku hotové plošiny se všemi výše zmíněnými prvky si můžete prohlédnout na obrázku 4.3 vedle.

Kamerový systém je tedy, jak již bylo zmíněno, součástí plošiny a všechny tři kamery jsou k plošině připevněny tak, že je zaručena vzájemná kolmost jejich os. Na studovaná zvířata jsou upevněny kontrastní barevné markery – do každé roviny dvě tečky barvy, která tvoří pokud možno nejvyšší kontrast vzhledem ke zbarvení těla živočicha. Softwarově je video z měření za dozoru zpracováno a polohy markerů v dané rovině zaznamenány, načež je spočten úhel mezi orientací segmentu těla živočicha a plošiny v prostoru.

Při nanášení markerů bylo užito různých metod, od standardních nalepovacích markerů, které ze slizem pokryté pokožky žab odpadávaly, přes potravinářská barviva, nanesená na bíle obarvený podklad práškového složení, který při kontaktu s vlhkou pokožkou tvořil pevnější strukturu, až po nestandardní nalepovací bílé papírky, na které se barvy nanasly pomocí fixu.

Výstupem z našeho měření v jedné rovině (sagitální) je tedy vývoj úhlů mezi markery, reprezentujících úhel označeného segmentu, v čase. Společně s tímto úhlem jsme schopni měřit pohyb plošiny a obě časové závislosti v návaznosti na sebe analyzovat.



Obrázek 4.3: Fotografický snímek hotové plošiny ve standby režimu.

4.4 Fixace živočicha

Radikální řešení fixace živočicha přímo k plošině, popsané v člancích (*Beloozerova, 2003*) a (*Hsu, 2012*), subjekt příliš omezuje a bezesporu ovlivňuje výsledky měření. Oproti tomu realizaci natrénování gekončků a toho dne chycených žab bychom si asi představovali stěží (*Macpherson, 2007*). Obojživelníky a plazy nelze jednoduše trénovat a jejich bezbariérové umístění na plošinu se během experimentu zkrátka neobejde bez obtíží v podobě nahánění živočicha po prostorech laboratoře. Je tedy nasnadě, že potřebujeme kombinaci obou řešení – živočich musí mít dostatečnou volnost k tomu, aby mohl provádět pohybovou kompenzaci a nebylo tím narušeno jeho fyzické zdraví, ale zároveň mu musí být zabráněno v případném úniku z plošiny či významné změně jeho orientace v prostoru vůči kamerám.

Jedním z prvních řešení byl molitanový svěrák s otvorem pro hlavu, na níž jsme kompenzaci sledovali. Zvířeti jsme obvykle nechali přední končetiny s hlavou na jedné straně svěráku a zbytek na druhé. Tělo bylo uzavřeno v kapse pomocí gumiček, kterými jsme svěrák na obou stranách stáhli. Tento způsob fixace skýtal možnost měřit subjekt nerušeně v jedné rovině, avšak zvíře se záhy po fixaci cítilo nekomfortně a projevovalo tendence k úniku. Zvláště u žab, které se před měřením vždy nafoukly, toto představovalo značný problém, protože po krátké době od zahájení měření se relativně uklidnily a vyfoukly, čímž zmenšily svůj původní objem natolik, že jim nečinilo problémy se ze svěráku vymanit. Pokud pomíneme zmíněné nedostatky našeho prvního řešení, stále tu byl problém s živočichy, jejichž pokožka je fyziologicky pokryta slizem, který molitanový svěrák efektivně vstřebával a tím pokožku živočichů nebezpečně vysoušel.

Pouze koncepčním řešením zůstala možnost přidělení poutek ze suchého zipu pro tělo živočicha přímo k plošině. Jedná se ale o řešení, které zůstalo opravdu pouze „na papíře“, jelikož problémy molitanového svěráku řeší jen ve značně omezené míře. Návrh nám však vnukl myšlenku pokrytí celého prostoru naklápěcí desky suchým zipem a vymyšlení fixačního mechanismu na bázi suchého zipu.

Finálním řešením se staly plastové průhledné krabičky slepené potravinářským lepidlem pro živočichy různých rozměrů. Každá krabička je zespod opatřena suchým zipem, který přilnul k povrchu naklápěcí destičky. Krabičky jsou zhotoveny v několika velikostních variantách, přičemž jejich volba odvisí na velikosti těla měřeného živočicha. Zvíře má uvnitř většinou jen tolik prostoru, aby se nemohlo v krabičce otočit – naneštěstí někteří gekončci se ukázali jako nadmíru flexibilní a párkrát nám dokázali, jak hluboce jsme se v posudku jejich obratnosti zmýlili. Pro zajištění návaznosti měření i po modifikaci polohy živočicha vzhledem ke kamerovému systému bylo řešení rozšířeno o připevnění magnetu k naklápěcí destičce a vložení magnetu dovnitř krabičky, což neovlivní fixaci krabičky k plošině a navíc nám to umožní pozici krabičky vzhledem ke kameře měnit během nahrávání v případě, že se zvíře chová neposlušně a v krabičce se různě natáčí.

4.5 Závěr kapitoly

V kapitole bylo popsáno umístění jednotlivých hlavních komponentů celého systému na těle ERP. U každého z hlavních prvků byl stručně popsán jeho funkční význam. Dále bylo v kapitole shrnuto proudové a krokové nastavení použitých aktuátorů jakož i vysvětlen princip krokových motorů obecně. Dále byl popsán kamerový podsystém pro záznam pohybu

segmentů živočicha na plošině a navrhnutá metoda fixace živočicha k plošině. Výše uvedený popis komplexního řešení systému s naklápěcí plošinou dokazuje, že se jedná o náročný návrh, a než bylo dosaženo současného stavu, bylo třeba provést řadu modifikací - především proběhlo několikeré přearanžování uložení motorů a kamer tak, aby bylo dosaženo požadovaných rozsahů pohybu plošiny a nedošlo k přetížení motorů. Samotný návrh hardwaru však musí být doplněn vhodným softwarem, aby bylo zabezpečeno bezproblémové řízení podsystémů celého systému a zpracování měřených dat.

5 Řízení pohybu, záznamu a hodnocení dat

Následující kapitola obsahuje popis mnou navržené metodiky, algoritmy pro řízení plošiny a popis navržených GUI pro řízení plošiny. Dále je v kapitole popsána komunikace mezi kamerami, PC a plošinou.

5.1 Metody řízení plošiny

5.1.1 Manuální krokování

5.1.1.1 Funkční význam a princip

Během návrhu a odlaďování ovládání plošiny bylo nutné v prvé řadě zavést možnost manuálního příkazu k natáčení motoru o určitý úhel v jednom směru rotace. Za tímto účelem se v hlavním řídicím GUI nachází pole, do kterého se vyplňuje krok motoru. Poté se uživatel rozhodne, se kterým motorem chce o daný krok na kterou stranu pootočit a podle toho klikne na jednu ze 6 šipek v GUI (2 pro každý směr a 3 motory).

Informace jsou řídicímu mikroprocesoru odesílány v diametrálně odlišném formátu, než ve kterém je zadáván úhel a směr rotace. Zprv, mikroprocesor s úhly vůbec nepracuje. Celý rozsah pohybu motoru má rozkvantován na počet hodnot odpovídající součinu *zadaného rozsahu ve stupních a počtu mikrokroků na jeden stupeň*. Náš nastavený krok je tedy nutné pro každý motor před odesláním přeložit na počet mikrokroků, kterému pootočení odpovídá. Směr pohybu ovlivní pouze to, zdali výsledek od současné polohy odečteme, nebo jej k ní přičteme.

Samotná poloha ovšem nestačí, neboť je třeba zadat také rychlost, jaké má požadovaný motor během rotace dosáhnout. Ta je v uživatelském rozhraní nastavována opět do pole k tomu určeného, a to ve stupních za sekundu. Mikroprocesor však vyžaduje jako vstup časovou hodnotu, udávající frekvenci s jakou má motoru posílat vstupní signál. Tato frekvence a střída poté určují rychlost naklápění motoru a jeho sílu. Je tedy nutné údaj o rychlosti ve stupních za sekundu z GUI převést na čas. Tento převod probíhá zcela empiricky a výsledná rovnice byla nalezena pomocí experimentálního měření s gyroakcelerometrem od firmy Xsens metodikou užití zpětné vazby.

Vzhledem k tomu, že v dosavadních experimentech autoři článků zkoušeli různé varianty nastavení rychlosti, viz *kapitola 2.1 – dynamické naklápěcí plošiny*, můžeme si teoreticky dovolit to samé. Ideální by bylo měřit se stejnou rychlostí, jako již bylo měřeno jinde. Bohužel existují rychlosti, při kterých může pohyb konstruktivně interferovat s plošinou za přítomnosti oscilací (hlubší pojednání o problému je v *kapitole 5.1.4 - Korekce sinusového zrychlování a zpomalování*) a tyto rychlosti tedy musíme bezpodmínečně vyřadit, protože by vzniklé oscilace narušovaly celý experiment. Nakonec bylo s Přírodovědnou Fakultou Univerzity Karlovy v Praze oboustranně dohodnuto užití bezpečných rychlostí o hodnotách $2,5^\circ/s$ a $5,0^\circ/s$.

5.1.1.2 Návrh GUI ovládání aktuátorů

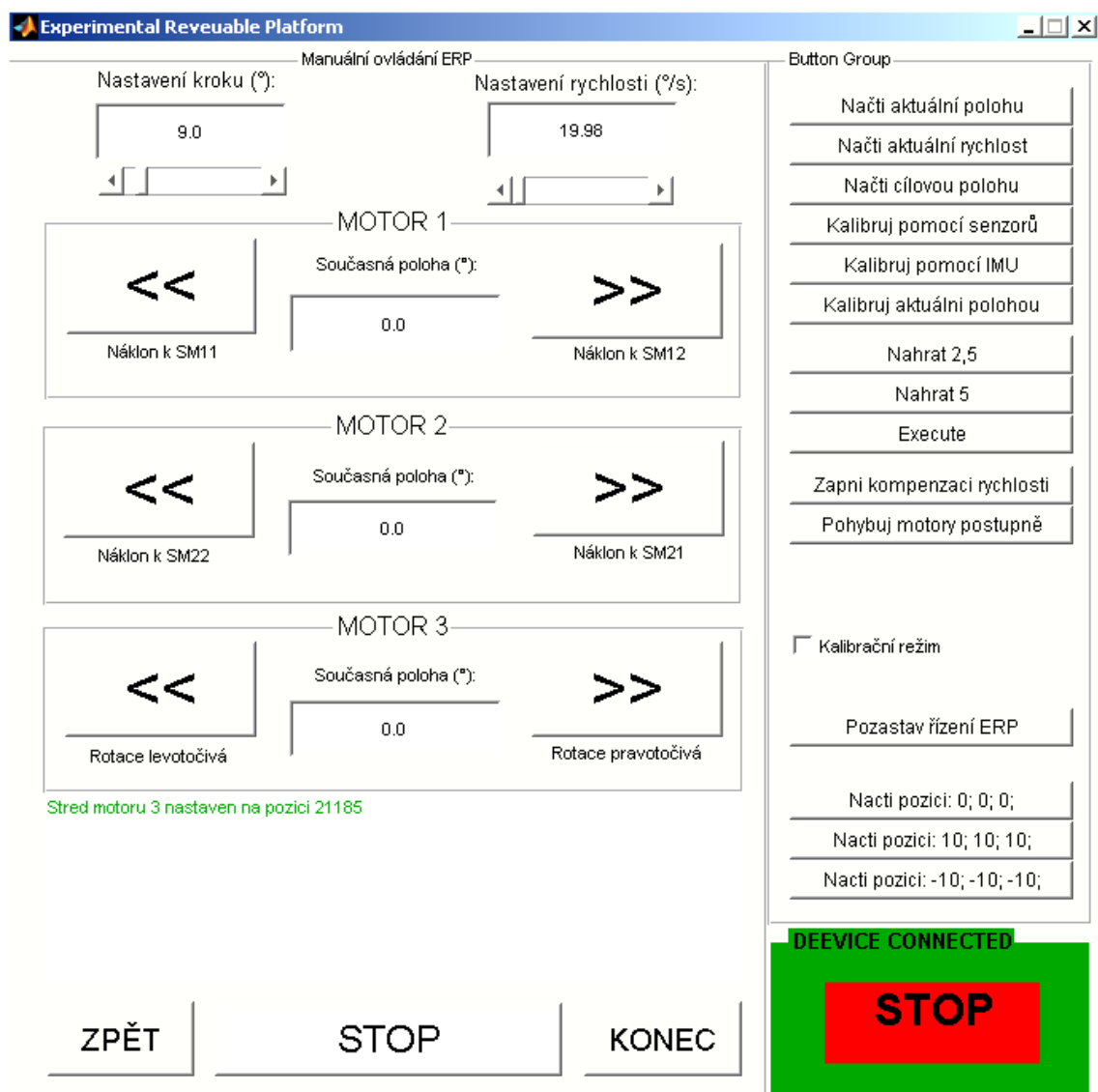
Navržené GUI uvedené na obrázku 5.1 vedle bylo vytvořeno v souladu s požadavky na způsob řízení plošiny a kamerového systému. Rozmístění jednotlivých prvků je navrženo a rozšířeno podle předchozích experimentů, uskutečňovaných během vývoje plošiny.

Obrázek 5.1 níže ukazuje GUI pro manuální ovládání aktuátorů. V hlavičce se nastavuje již v kapitole 5.1.1.1 zmíněný krok a rychlost posuvu motorů, dále je GUI rozdělena na několik dalších podčástí – tři z nich pod sebou ukazují polohu jednotlivých motorů vzhledem k výchozí kalibrované poloze a tlačítka se šipkami po stranách umožňují uživateli s jednotlivými motory potočit na kteroukoliv stranu o udaný krok udanou rychlostí. Pod nimi se nachází dialogový box, informující uživatele o událostech, jež během řízení probíhají, včetně upozornění na případné výskyty chyb. Význam panelu s tlačítky vpravo bude dále popsán v příslušících kapitolách.

5.1.2 Sekvenční řízení

5.1.2.1 Funkční význam a princip

Na návrh vlastního řízení krokových motorů pomocí GUI je navázána implementace složitějších funkcí, jako je možnost nahrání komplexnější sekvence pohybu jednotlivých motorů



Obrázek 5.1: Ukázka z GUI pro manuální řízení motorů.

plošiny. Tato sekvence by byla po odeslání na UART (řídící mikrokontrolér) uložena do FLASH paměti mikrokontroléru a následně plošinou vykonána automaticky, bez nutnosti zásahu uživatele.

Za tímto účelem jsem navrhl GUI, umožňující uživateli jednak vytvářet scénáře pohybu, jednak je ukládat ve formátu *.xls (*.xlsx) na disk a nakonec načítat. Hlavní myšlenkou byla možnost externího uživatele předpřipravít si scénář mimo laboratoř a pak jej jednoduše načíst do programu a vykonat. Program sám rozpozná, v jakém formátu jsou data v importované tabulce uložena a snaží se je přečíst, korigovat a uložit do náhledové tabulky v GUI.

Každý scénář musí nutně obsahovat tři datové sekce pro tři motory. Tyto sekce mohou být v externím scénáři umístěny prakticky téměř libovolně – musí pouze respektovat logické pořadí motorů (zleva či shora – 1., 2., 3.). Sekce mohou být vedle sebe v souboru pro čtení uspořádané horizontálně, či vertikálně, mohou či nemusí být spojeny do jedné tabulky, mohou mezi sebou mít prakticky libovolně velké mezery a mohou či nemusí obsahovat popisky sloupečků.

Každá sekce musí obsahovat minimálně dvě datové sady – nutnou sadou je poloha ve stupních, povinně volitelnou sadou je buďto rychlost, s jakou se uživatel chce na danou polohu dopravit, anebo čas, za který se sem chce dopravit. V závislosti na zvolené možnosti v GUI (řízení pomocí polohy a rychlosti, nebo polohy a času) jsou sady správně načteny do programu a poslední údaj (rychlost, nebo čas) je dopočten. Software rovněž kontroluje správnost zadaných údajů, aby nedošlo k přesahu maximálního rozsahu plošin, případně přesahu maximální rychlosti plošiny. V případě výskytu neplatných dat jsou data zaokrouhlena na nejbližší možnou hodnotu a zbytek údajů v tabulce dopočten na základě nových hodnot. Před konverzí a odesláním na UART jsou všechna data v tabulkách ještě jednou pro jistotu zkontrolována.

S ohledem na dosud provedené výzkumy a požadavky Přírodovědné Fakulty Univerzity Karlovy v Praze bylo však prezentované GUI odsunuto do pozadí, jelikož výzkum vyžaduje v rámci jeho obhajitelnosti v kontextu s předešlými pracemi pouze periodické scénáře. Do GUI na obrázku 5.1 na předchozí stránce tedy byly na pravý panel zavedeny tlačítka pro nahrání zatím dvou použitelných scénářů rampových přechodů z $-21,6^\circ$ do $+21,6^\circ$ a zpět s možností selekce rychlosti $2,5^\circ/s$ a $5,0^\circ/s$. Oba scénáře jsou dimenzovány tak, že zaberou bezmála 3 minuty, čímž jsou měření normalizována. Naposled nahraný scénář je inicializován tlačítkem *Execute* (provést).

Než jsme tyto dva scénáře implementovali jako součást řídicího GUI, byl na žábách testován rovněž scénář sinusového pohybu, kdy se rychlost v čase měnila tak, že minimum amplitudy bylo na $2,5^\circ/s$ a maximum na $5,0^\circ/s$. Sinus probíhal vždy pouze jednosměrně, na druhou stranu jsme se vraceli po rampovém přechodu, a každý sinus či návrat bylo nutné manuálně zadat – výsledná měření byla tedy spíše experimentálního charakteru a sloužila spíše k testování možností systému. Z důvodu hrubé kvantizace celkového rozsahu čítajícího $43,2^\circ$ po $3,6^\circ$ (tedy 12 úseků) se však sinusový scénář ukázal jako méně vhodný, a nebyl dále využit.

5.1.2.2 Návrh GUI ovládání aktuátorů

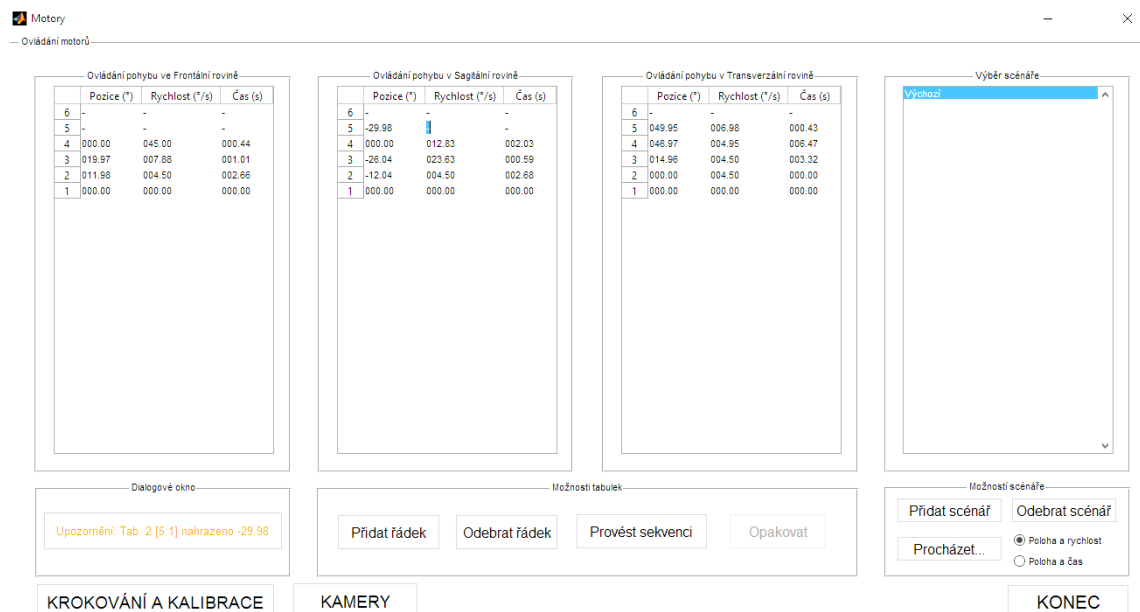
Na obrázku 5.2 na vedlejší stránce je možné spatřit ukázkou původního GUI pro sekvenční řízení aktuátorů. GUI je opět rozděleno do několika podsekcí (panelů). Tři z nich obsahují tabulkové scénáře pro jednotlivé motory, každý scénář sestává z informace o poloze, rychlosti,

se kterou se chceme na konkrétní úhel natočení dostat, a doby, jakou nám to má trvat. Stačí vyplnit polohu a libovolný z dalších dvou údajů pro automatický dopočet třetího. V případě zpětné modifikace polohy se však nabízí dvě varianty řešení – buďto opravíme údaj o rychlosti při zachování času, nebo opravíme údaj o čase při zachování rychlosti. Pro účel tohoto výběru je vpravo dole možnost zatržení řízení buďto polohou a rychlostí, která upřednostňuje rychlost před časem, anebo řízení polohou a časem, která upřednostňuje čas před rychlostí. Výběr určuje též chování softwaru při načítání scénářů z externích souborů. V případě pouze dvousloupcových sad dat tak očekává na místě druhého sloupce buďto rychlost, nebo čas. V případě kompletních trojsloupcových sad dat potom v závislosti na této volbě jeden ze zbýlých dvou sloupců zanedbává a dopočítává.

Dialogový box vlevo dole, upozorňující uživatele na korigované hodnoty či prázdná pole nebo informace o úspěšném načtení, uložení scénáře a jeho odeslání na UART, nesmí ani zde chybět. Vpravo si pak můžeme všimnout listu se scénáři, mezi kterými můžeme přecházet. Na obrázku 5.2 níže je uveden pouze výchozí scénář, pomocí tlačítek níže však můžeme scénáře libovolně přidávat, mazat a měnit jejich název. Tlačítkem dole uprostřed „Provést sekvenci“ provedeme kontrolu dat, konverzi na pro UART stravitelný formát a odeslání na UART k provedení.

5.1.3 Automatická kalibrace

Po každém zapnutí plošiny se předpokládá odchylka od výchozí tj. nulové polohy, povětšinou velice významná. Je tedy na místě po každém zapnutí provést automatickou kalibraci výchozí polohy.



Obrázek 5.2: Navržené GUI pro sekvenci řízení motorů, umožňující vytvářet, ukládat a ukládat uživatelem definované scénáře pohybu plošiny.

5.1.3.1 Kalibrace pomocí dorazových senzorů

Automatická kalibrace pomocí dorazových senzorů probíhá tak, že systém postupně jede s každým motorem na obě strany tak dlouho, dokud nenarazí na dorazový senzor a počítá si, přes kolik mikrokroků se při tom otočil. Jelikož známe celkový rozsah každého z motorů v mikrokrocích, jsme schopni z počtu přejetých mikrokroků mezi oběma dorazy určit polohu, ze které jsme kalibraci zahajovali, a vypočítat, o kolik mikrokroků se posunout na kterou stranu, abychom dosáhli středové polohy rozsahu pro daný motor. Uvedené se postupně opakuje pro všechny tři motory v pořadí odshora dolů. Výsledkem automatické kalibrace je tedy poměrně přesné uvedení plošiny do rovnovážné výchozí polohy. Polohu je nicméně třeba dokalibrovat jiným způsobem – z tohoto důvodu je kalibrace za pomoci dat pouze z řídicího softwaru nedostatečná, jak bylo nastíněno už v rešerši.

5.1.3.2 Akcelerometr

Pro dodatečnou kontrolu je na spodní straně naklápěcí desky umístěn trojosý akcelerometr ArduinoModul, měřící náklon horních dvou motorů a rotaci spodního motoru v prostoru (snímač úhlové rychlosti - L3G4200D, digitální akcelerometr - ADXL345, kompas - HMC5883L). Akcelerometr slouží k ověření správnosti výchozí polohy a její korekci. Metodou zpětné vazby posílá motorům tak dlouho informace o natočení v prostoru, dokud se plošina nenachází v žádané výchozí poloze.

5.1.3.3 Vodováhy

Protože se občas může ERP systém i akcelerometr mýlit, a protože jsme před jeho implementací potřebovali pozici plošiny kalibrovat bez něj, zavedli jsme do pravého panelu na obrázku 5.1 ze strany 21 možnost manuální kalibrace. Principiálně to funguje tak, že manuálně pohybujeme s každým motorem tak dlouho, dokud vodováhy na okrajích naklápěcí plošiny neukazují středovou polohu. Jakmile jsme spokojeni s polohami bublinek vzhledem k ryskám na vodovahách, stiskneme tlačítko „*Nastavit jako výchozí polohu*“, čímž se přepíše aktuální polohy v GUI všech motorů na 0° a mikrokrokové polohy motorů celého systému na středy jednotlivých rozsahů.

5.1.4 Korekce sinusového zrychlování a zpomalování

Při odlaďování rotačního pohybu naklápěcí desky pomocí manuálního krokování jsme narazili na zajímavý fenomén vyskytující se u všech motorů, jehož původ jsme nebyli schopni lokalizovat, nicméně jehož výskyt bylo třeba potlačit.

Zašleme-li libovolný motor z jedné okrajové polohy do druhé, je možné pozorovat zvláštní sinusový charakter rychlosti pohybu motoru, přičemž amplituda tohoto sinu se jeví vyšší při krajních úvratích motorů, tedy v místech, kde je třeba vyvinout největší moment síly k rozpoohybování motoru. Je tedy nasnadě, že uvedený jev nějak souvisí se zatížením motoru, zkusili jsme tedy navýšit řídicí proud a změnit zapojení motorů na bipolární, abychom co nejvíce navýšili kroutící moment motoru. Přestože jsme jev těmito opatřeními utlumili, nezbavili jsme se jej zcela, a zůstal stále dobře pozorovatelným. Bylo tedy třeba navrhnout jiný způsob korekce.

Pomocí gyroakcelerometru od firmy Xsens jsme naměřili charakter amplitudy tohoto parazitního sinu jako funkci rychlosti v závislosti na vzdálenosti od středové polohy motoru. V ovládacím softwaru jsme navrhli protifunkce pro jednotlivé motory, které po sečtení s uvedeným fenoménem vytvoří destruktivní interferenci a zůstane pouze stejnosměrná rychlostní složka. Uvedené opatření by pravděpodobně nefungovalo v případě, kdy by fáze parazitního sinu byla funkcí vzdálenosti od středu motoru – my však přišli na to, že sinus vždy začíná tam, kde začíná pohyb motoru, což nás přivádí k myšlence, že se jedná o vadu samotných motorů. Za těchto okolností jsme tedy schopni parazitní sinus na libovolné dráze poměrně dobře kompenzovat. Kompenzace není 100%, jelikož jsme pracovali s korekční protifunkcí naměřenou při pohybu plošiny z jednoho dorazu ke druhému. Pokud tedy s plošinou pohybujeme v jiných mezích, je stále místy (především u krajních úvratí) patrné mírné kolísání rychlosti. Vzhledem k původnímu charakteru tohoto fenoménu se však již jedná o zanedbatelný bias.

Je samozřejmostí, že po této korekci bylo třeba znovu testovat převodní funkci mezi uživatelem zadanými $^{\circ}/s$ v GUI a skutečnou úhlovou rychlostí plošiny, a uzpůsobit ji.

5.1.5 Decelerace a akcelerace na požadovanou rychlost

Dalším problémem, který bylo nutno vyřešit, byla postupná akcelerace a decelerace na začátku a konci pohybu plošiny. Bez tohoto zásahu se plošina dostává z nulové rychlosti na požadovanou v řádech setin vteřiny, což je problém především u výše zmíněných rampových scénářů, kdy plošina dojede na jeden doraz a ihned se rozjede na druhý. Setrvačná síla, působící v okamžiku změny směru pohybu plošiny na živočicha, nabývá v tomto okamžiku nejvyšších hodnot a zcela narušuje měření.

Do GUI bylo za účelem řešení tohoto problému zavedeno postupné zrychlení a zpomalení plošiny. To se musí dít na malém, avšak zanedbatelném úseku trajektorie. Experimentálně jsme usoudili, že dráha odpovídající $1,8^{\circ}$ (128 mikrokrokům) postačuje k posupné akceleraci, či deceleraci. Na definovaném úseku je ale možné provést pouze jednu z těchto variant – odsud plyne, že chceme-li s plošinou pohybovat a zároveň využívat tuto utilitu, navýší se nám minimální krok plošiny na dvojnásobek, tedy $3,6^{\circ}$ ($1,8^{\circ}$ na rozjezd a $1,8^{\circ}$ na zastavení).

5.1.6 Zapínání a vypínání plošiny

Výše bylo zmíněno, že pokud vypneme plošinu, nuluje se záběrový moment motorů a celá konstrukce se zřítí, což je pro motory velice nebezpečné. Z toho důvodu je třeba při každém, byť nouzovém, vypnutí plošiny naklápěcí desku držet. Pokud je plošina vypnutá a nehodláme ji v nejbližší době zapínat, je v nejvyšším bodě oftalmoskopického ramene vyvedena díra, kterou je provlečen úchytný hák, kterým se zahákne horní posuv kamerového systému a plošina tak setrvává sama téměř ve výchozí poloze.

V původním konceptu byl sepsán skript pro automatické bezpečné vypínání plošiny. Při stisknutí příslušného tlačítka plošina automaticky postupně s motory došla na vytyčené dorazy a poté se vypla, čímž bylo zamezeno pádu konstrukce z velké výšky. Bohužel naše rezervy na jednotlivých dorazech způsobí, že i tak má plošina před vypnutím motorů asi 5 cm prostoru k pádu, což je vzhledem k váze celé nástavby nepřijatelné. Dlouhodobým setrváváním v takto

dosažené poloze je navíc namáhán materiál konstrukce - je tedy nutno motory stejně něčím podložit. Z toho důvodu se skript neujal a my přešli na závěsné řešení.

5.2 Metody zpracování obrazové informace

Následující kapitola popisuje použité metody při vytváření GUI pro nahrávání videí, detekci markerů a export úhlů. Zároveň jsou zde zmíněny některé hlavní problémy s nahráváním a jejich efektivní řešení či potlačení.

5.2.1 Formát nahraných videí

Při výběru vhodného formátu nahraných videí bylo vybíráno mezi klasickým *.avi a nekomprimovaným *.avi. Oba typy se mezi sebou diametrálně liší velikostí výstupního souboru, za což může komprimace typu „*delta frames*“, která je u nekomprimovaného typu potlačena. Princip *delta frames* spočívá v pamatování si pouze změny následujícího snímku vzhledem k aktuálnímu, čímž odpadá potřeba pamatovat si dlouhé údaje o každém z pixelu snímku videa a snižuje se tak významně velikost souboru. Vzhledem k typické uživatelské aplikaci, jakou je jednoduché přehrávání videa od začátku do konce, je toto ideálním řešením, protože systém má vždy před přechodem na další snímek přímý přístup k předchozímu, který zrovna zobrazuje.

Při zpracovávání videí toto ovšem představuje problém, protože je často nutné ve videu uskutečňovat posun po desítkách či stovkách snímků. U standardních volně dostupných i placených přehrávačů je video ihned po inicializaci rozsegmentováno na několik částí a údaje o každém snímku na začátku každého segmentu jsou uloženy do paměti, takže pokud se uživatel rozhodne ve videu popojít o sérii snímků, systém ke zobrazení požadovaného snímku využije údaje z posledního známého snímku na začátku aktuálního segmentu, který neleží nikterak daleko a snímek je tak zobrazen v relativně krátkém čase.

Při načítání požadovaného snímku v MatLabu® toto ale realizovat nelze. Segmentace videa a uložení počátečních snímků každého segmentu do paměti nelze využít k urychlení přístupu k pozdějším snímkům ve videu – program při dotazu na poslední snímek videa musí procházet každý snímek počínaje tím na samotném začátku videa. U dlouhých videí, obsahujících vyšší počet snímků, se čekací doba vzhledem k nutné frekvenci posuvu ve videu stává nepřípustně dlouhou (například 30 sekund).

Problém řeší nekomprimované *.avi. Zde však rapidně narůstá velikost výstupního souboru. Bavíme se o jednotkách GB na 3 minuty záznamu pro jedno video. Vzhledem k tomu, že vždy nahráváme ze tří kamer současně a videa jsou v rámci postprocessingu normalizována tak, že všechna obsahují nejvyšší počet snímků pro dané měření (viz kapitola 5.2.3 – *Předzpracování videí*), celková velikost jednoho třiminutového měření je okolo 10 GB. Pro ilustraci – při měření jedné série gekončíků o 15 kusech a 4 měření na jeden kus vychází celková velikost nahraných videí až na 600 GB.

Pro urychlení celého procesu ukládání videí byl pořízen SSD (Solid State Drive) disk, který má ale omezenou kapacitu na 120 GB. Nahraná videa je tak třeba ihned po ukončení nahrávání přesunout na úložní HDD (Hard Disk Drive) o kapacitě 1 TB. I taková kapacita by ale byla vyčerpána během dvou měření (a to nezmiňuji zálohování dat). Bylo tedy nutné přijít s řešením.

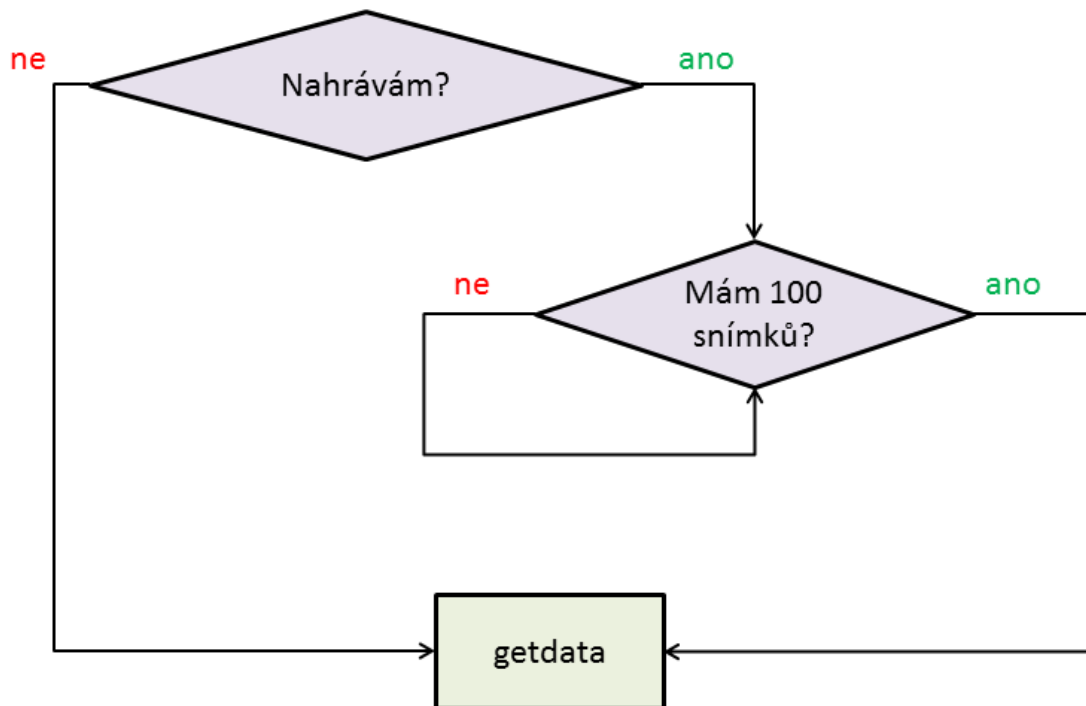
Za tím účelem byla navržena metoda, využívající externího programu pro změnu typu komprimace videosouboru. Samotný nahrávací skript v MatLabu® nahrává v komprimovaném klasickém *.avi, čímž šetří volné místo na disku. Pokud chceme ale videa zpracovávat a zjišťovat úhly, je třeba komprimaci videí před jejich nahráním do programu externím programem změnit na nekomprimované *.avi. Proces změny v použitém volně dostupném *virtualDubu* trvá pro jeden tříminutový záznam asi 10 sekund a vzhledem k době nutné ke zpracování jednoho videa (kolem 10 minut) se tak jedná o zanedbatelnou časovou přírážku. Rád bych zde podotkl, že se jedná o bezztrátovou kompresi založenou na metodě eliminace redundantních dat, tedy pokud zvolíme zachování 100% kvality videa, neutrpí tím video jakoukoliv újmu na kvalitě – vyhodnocení záznamu tím tedy není nikterak dotčeno.

5.2.2 Extrakce snímků

K extrakci nahraných snímků z webkamer v MatLabu® slouží příkaz „*getdata*“. Standardně však software veškeré nahrané snímky ukládá do mezipaměti a nároky na systémové prostředky tak s časem rostou až do doby, než je program automaticky ukončen z důvodu přetečení dostupné operační paměti. Problém lze poměrně jednoduše vyřešit vhodným nastavením vlastnosti založeného videoobjektu s názvem „*DiskLogger*“. Ten lze nastavit buďto na logování pouze do paměti, pouze na disk, nebo do paměti i na disk. Pokud jej nastavíme na logování pouze na disk, je problém téměř vyřešen definitivně, jediné riziko je, že systém nestíhá zaznamenané snímky z kamer ukládat na disk rychleji, než je kamery pořizují – v takovém případě postupně během nahrávání stále rostou požadavky na paměť, přestože o dost pomaleji. Během nahrávání je tak třeba sledovat ve správci úloh náročnost procesu MatLab® na paměť a pokud vidíme, že hodnota roste, je třeba zajistit nižší sběr snímků u kamer například snížením osvětlení nahrávané scény.

Během nahrávání bylo vhodné zajistit zpětnou odezvu kamerového systému v podobě zobrazování stávajícího frameratu jednotlivých kamer. Zpočátku byl orientačně framerate kamer odhadnut pouze při startu softwaru a to průměrově ze 40 pokusně nahraných snímků, přičemž prvních deset se ignorovalo z důvodu automatického nastavování clony u kamer vzhledem k osvětlení scény (kolísání frameratu). Později jsme zjistili, že se framerate vyvíjí a mění v průběhu měření a bylo vhodné tyto změny mezi jednotlivými nahrávanými sledovat. K tomu opět poslouží příkaz *getdata*, který kromě snímků z kamer umí vrátit i systémový čas, ve kterém byly jednotlivé snímky pořízeny. Z časové difference mezi jednotlivými snímky můžeme určit vývoj frameratu v čase a po skončení nahrávání jednoduše zjistit medián frameratu. Medián byl zvolen pro lepší reprezentaci skutečného frameratu nežli poskytuje aritmetický průměr z důvodu ignorování náhlých skokových krátkých změn. Problémem byla pouze restrikce příkazu *getdata* na nastavení vlastnosti videoobjektu *LoggingMode* na *Memory*, což nás vrací zpět k problému s přetékáním paměti. *Getdata* totiž neumožňuje extrahovat jakoukoliv informaci při nastavení *LoggingMode* videoobjektu na *Disk*.

Navrhl jsem tedy kompromis v řešení, spočívající v nastavení *LoggingMode* na *Disk&Memory*, přičemž každých 100 nahraných snímků se automaticky spouští funkce, dotazující se pouze na časová razítka, která si ukládám průběžně do vektoru (*FramesAcquiredFcnCount* = 100 a *FramesAcquiredFcn* volá *getdata*). Příkazem *getdata* se totiž vyčistí operační paměť od extrahovaných snímků. Jelikož v tu dobu běží ukládání videa i do souborů na disk, výstupem je video na disku a vektor času záznamu jednotlivých snímků v



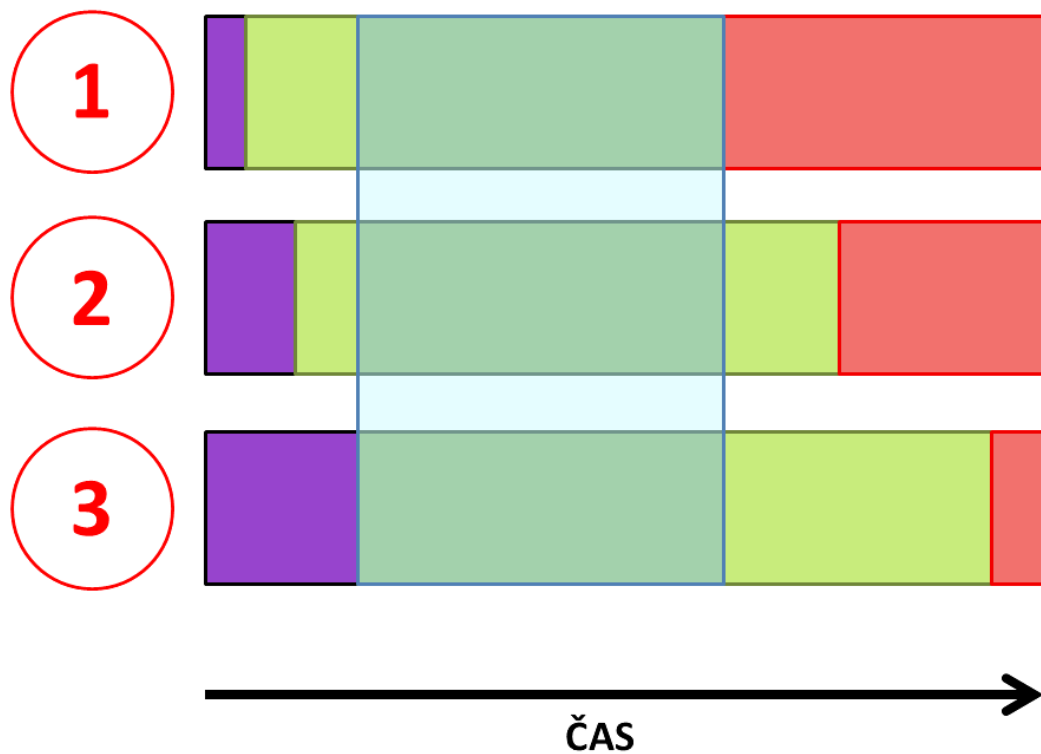
Obrázek 5.3: Schéma algoritmu pro extrakci časových razítek k určení FPS a vykreslení grafů

paměti, který je po ukončení nahrávání zkompletován (extrakce ze zbývajících snímků z paměti – po stisknutí STOP tlačítka je zavolána *FramesAcquiredFcn* ještě jednou) a uložen do textového souboru k nahraným videím pro pozdější zpracování formou grafu. Jednoduchý algoritmus je znázorněn na schématu na obrázku 5.3 výše.

Po skončení každého nahrávání tak máme v paměti uložená časová razítka z předchozího měření, není tedy problém zjistit poslední framerate a uživateli jej pro informaci zobrazit v GUI. Podotýkám, že každý videosoubor může mít pouze jednu hodnotu framerate, stejnou pro celou délku záznamu. Náš framerate ale konstantní není, proto nahraná videa nemusí běžet v reálném čase. Hodnota framerate v nahraných videích je pouze orientační – pro hodnocení vývoje úhlů v čase slouží záznamy časových razítek z výše zmíněného textového souboru (viz následující kapitola 5.2.3 – *Předzpracování videí*).

5.2.3 Předzpracování videí

V první řadě bylo nutné vyřešit problém se zpožděním zapínání a vypínání nahrávání jednotlivých kamer. Jelikož není možné v MatLabu® kamery zapnout nebo vypnout záraz, ale pouze postupně formou *for* cyklu, dochází u kamer k jistým zpožděním či předstihům. Odstartování měření má většinou konstantní prodlevu blížící se 1 sekundě mezi zapnutím každé z kamer. Jejich vypínání je však zpožděno různě v závislosti na počtu snímků dané kamery v paměti, které ještě zbývá uložit do souboru na disk. Kamery s vyšším frameratem mají tak tendenci ukončovat nahrávání později, protože nahrají větší počet snímků za sekundu a zůstane jim tak v paměti po vypnutí více snímků k zaznamenání do souboru na disk. Je logické, že z hlediska měření nás zajímá pouze ten časový úsek, ve kterém zaznamenávaly všechny tři kamery, jak ilustruje obrázek 5.4 na vedlejší straně.

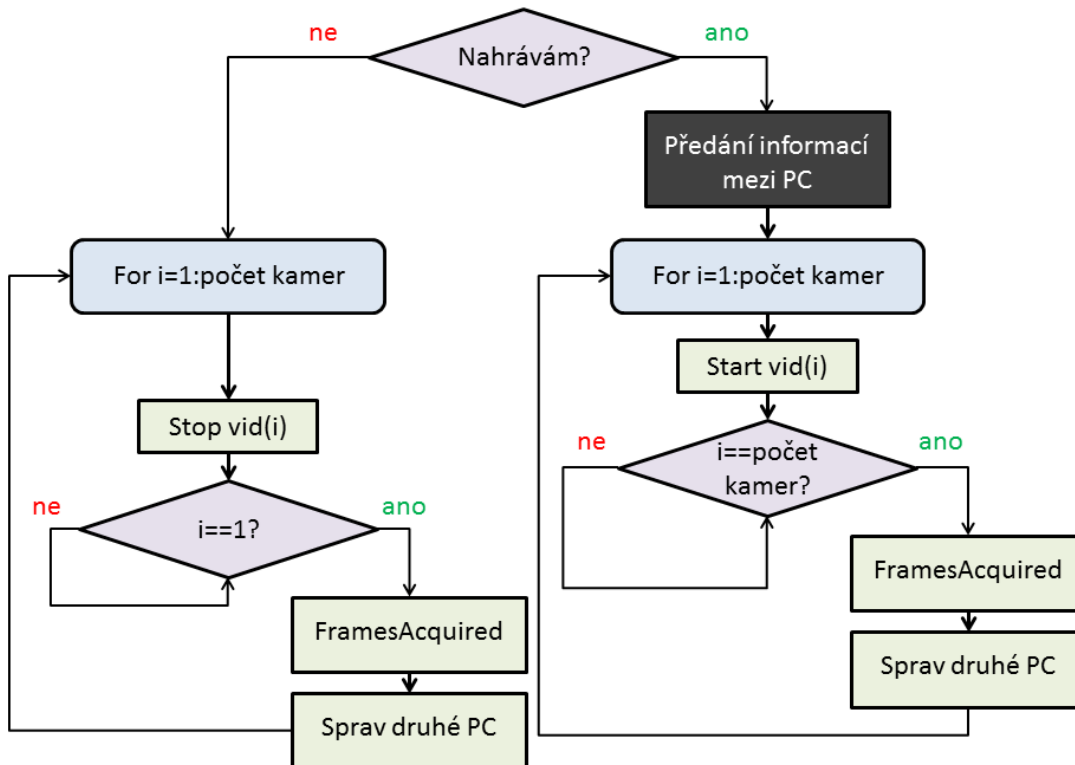


Obrázek 5.4: Ilustrace zpoždění při zahájení a ukončení nahrávání. Čísly v kroužcích jsou označeny kamery v pořadí, v jakém se zapínají, fialové zóny reprezentují časové úseky před zahájením měření, zelené během měření a červené po ukončení měření. Modře ohraničená zóna reprezentuje úsek, který nás zajímá a je třeba vyextrahovat.

Úseky je možné nasegmentovat za předpokladu znalosti časových razítek. Rád bych zde ještě podotkl, že varianta řešení přes dotaz na *AcquiredFrames*, vracející počet dosud zaznamenaných snímků danou kamerou, v době zapnutí a vypnutí kamer není realizovatelná. Přestože příkaz vrací rozumné údaje v době vypínání kamer, při startu ukazuje u všech kamer 0, což by znamenalo naprosto synchronní spuštění všech tří kamer zároveň, což v praxi nepozorujeme.

V první řadě je tedy v rámci postprocessingu nutné videa ořezat dle návodu výše. V době zapnutí poslední kamery mi tedy stačí vyextrahovat počet dosud zaznamenaných snímků z předchozích dvou kamer a v době vypnutí první z kamer provedu to samé se všemi zbylými kamerami. Tím získám údaje o tom, od kterého po který snímek mám nahraná videa ponechat a zbytek smazat. Údaje jsou společně s dopočteným frameratem zapsány do textového souboru ve sdíleném adresáři *Cutoff.txt*. Celou situaci znázorňuje diagram na obrázku 5.5 na další straně.

Druhým krokem je časová synchronizace videí. Chceme-li v MatLabu® přehrávat všechna tři videa zároveň, je nejprve nutné normalizovat jejich počet snímků. Každé video musí mít stejný počet snímků, což ale vzhledem k jejich rozdílným frameratům není zajištěno. Řešení je takové, že je všem videím přiřazen počet snímků odpovídající videu s nejvyšším počtem snímků z daného měření (tj. žádná ztráta použitelných dat). Určité snímky ve videích s nižším frameratem jsou na základě podílového faktoru, tvořeným podílem frameratu aktuálního videa a nejmenšího frameratu z měření, duplikovány tak, že při současném přehrávání všech tří videí



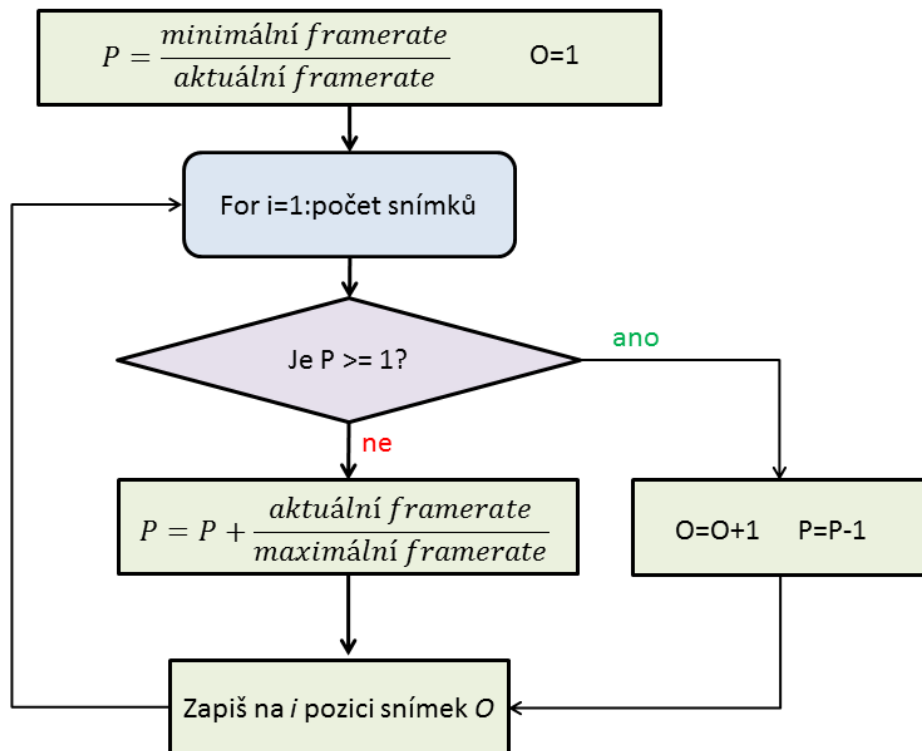
Obrázek 5.5: Diagram algoritmu pro zjišťování použitelného úseku videa. „Blackbox“ v pravé rozhodovací větvi bude pro svou komplexnost popsán na separátním diagramu dále v textu. Pravá větev označuje zahájení nahrávání, kdy čekám na zapnutí poslední z kamer, poté se dotážu na počet dosud zaznamenaných snímků předešlých kamer (*FramesAcquired* box) a vydám pokyn druhému PC k zahájení záznamu pohybu plošiny (*Sprav druhé PC* box). Levá větev slouží k extrakci téže informace v případě vypnutí nahrávání, dotazuje se ale všech kamer a box *Sprav druhé PC* zde ukončuje záznam pohybu plošiny.

je zajištěna časová synchronizace. Základní myšlenku ilustruje diagram na obrázku 5.6 na vedlejší straně. Tento postup byl testován na nahrávání displeje zapnutých stopek, přičemž časový údaj byl viditelný ze všech tří kamer, a byla ověřena jeho funkčnost. Případný nesouhlas délek po postprocessingu o jeden snímek je pak řešen jednoduchým oříznutím tohoto snímku od konce.

Takto upravená videa jsou připravena pro načtení v jiné části grafického uživatelského rozhraní, sloužícího k detekci markerů a nalezení úhlů.

5.2.4 Algoritmy detekce markerů

V této části představím funkci umožňující z každého snímku videa separovat dvě distinktivní barevné značky. Prvním krokem je možnost uživatele nastavit si barvu, na kterou se má SW zaměřit a tuto barvu následně v obrázku hledat. Barva je tvořena třemi podsložkami RGB (Red – červená, Green – zelená a Blue – modrá), přičemž každá nabývá hloubky hustoty tónů (též valéru) od 0 do 255. Nastavovat hledanou barvu pomocí tří sliderů je ale zbytečně náročné, proto jsem zavedl poněkud interaktivnější metodu, kdy uživatel vidí náhled k danému snímku a může si jednoduše myší nakliknout pixel s požadovanou barvou a SW automaticky přiřadí RGB hodnoty tak, aby odpovídaly barvě nakliknutého pixelu. Hledat pouze jednu kombinaci RGB hodnot by ale bylo chybou, jelikož přestože v reálu mají oba markery



Obrázek 5.6: Schéma časové synchronizace pomocí duplikací snímků ve videích; P je podílový faktor a O slouží jako počítadlo.

podobnou barvu, většinou není přesně na tón stejná. Byla tedy zavedena možnost stanovení tzv. *tolerance*.

Tolerance je toleranční rozmezí, specifikované uživatelem. Stanovuje jednostrannou odchylku od požadovaných RGB tónů, která ještě bude vnímána jako shoda. Pokud například uživatel hledá RGB tóny 120, 50 a 30 a má toleranci nastavenou na 10, SW označí všechny pixely v obrázku, mající RGB tóny v následujících rozsazích: 110-130, 40-60 a 20-40.

Z nalezených pixelů je možné vytvořit binární masku detekce, která se uživateli zobrazuje hned vedle náhledu aktuálního snímku, aby jasně viděl, co SW detekoval. Dalším úkolem bylo rozdělit množinu nalezených pixelů na 2 oblasti a jejich středy prohlásit za reprezentanty markerů. K tomu posloužila funkce *k-means*.

Algoritmus funkce *k-means* lze popsat nejlépe s pomocí obrázků. U tohoto algoritmu je na počátku nutné stanovit počet výsledných množin, což nám nečiní žádný problém, jelikož my víme, že markery hledáme vždy dva. V první iteraci algoritmus náhodně rozhodí středy těchto množin do obrázku (lze je nastavit manuálně). Nyní spočte euklidovské vzdálenosti každého detekovaného pixelu od těchto středů a podle jejich blízkosti ke středům je rozdělí do dvou množin (každé množině přísluší jeden střed). Poté spočte těžiště každé z množin dle fyzikálního vzorce 4

$$T_x = \frac{1}{N} \sum_{i=1}^N x_i \cdot S_i ; \quad T_y = \frac{1}{N} \sum_{i=1}^N y_i \cdot S_i , \quad (4)$$

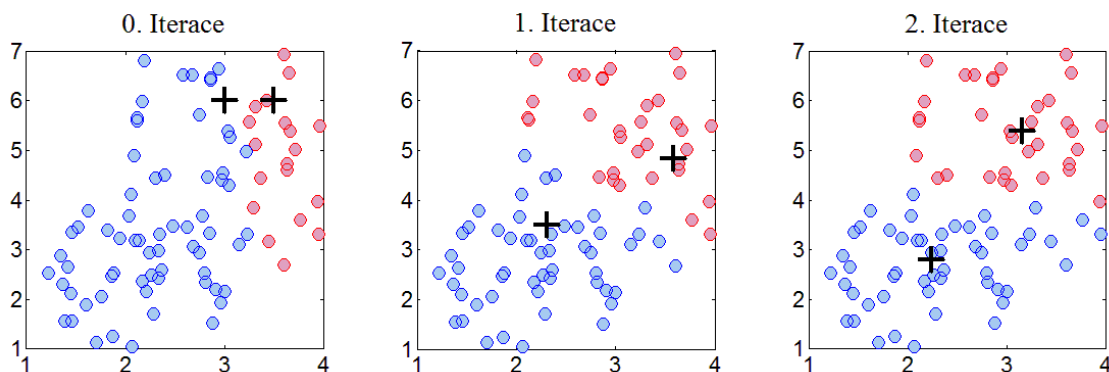
kde T_x a T_y označují xové a yové souřadnice těžiště množiny, N je počet pixelů příslušících dané množině, x_i a y_i jsou souřadnice i tého pixelu z množiny a S_i je jeho plocha. Protože se bavíme o pixelech, můžeme snadno nahlédnout, že plocha pixelu bude vždy jednotková, celý vzorec se nám výhodně zredukuje na vzorec pro střední hodnotu xových a yových souřadnic. Do těchto pozic nastaví *k-means* algoritmus nové středy množin. V dalších iteracích opakuje tyto tři kroky stále dokola, dokud se pozice středů signifikantně mění, jak ukazuje obrázek 5.7 níže.

V našem uživatelském rozhraní se středy určují náhodně pouze, pokud neznáme pozice středů na předchozím snímku, v opačném případě použijeme tyto středy jako výchozí, čímž algoritmus doiteruje v kratší době.

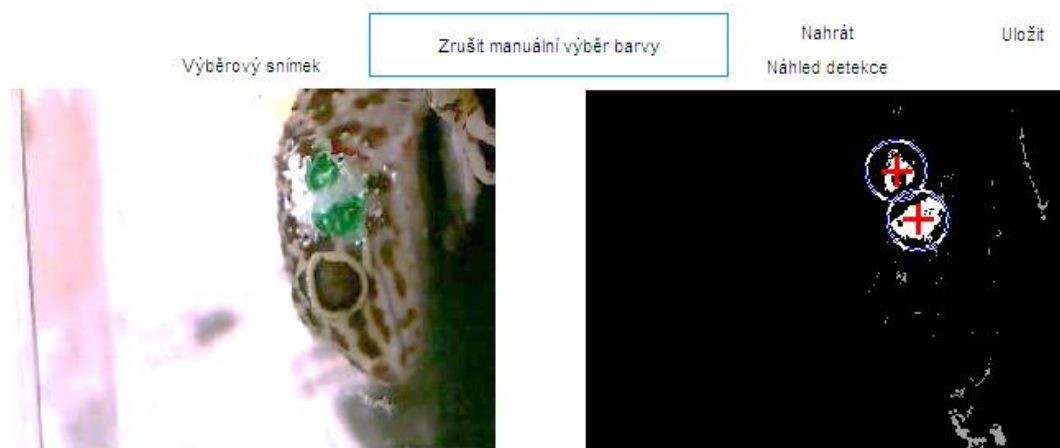
Bohužel bylo pokusem zjištěno, že *k-means* v MatLabu® občas dělá chyby a najde střed jedné z množin na zcela nesmyslném místě (přibližně 1/30 případů). Aby bylo této chybě zabráněno, byly do SW implementovány dva obranné mechanismy. Prvním z nich je kontrola mezimarkerové vzdálenosti, která sleduje vzdálenost mezi středy markerů na předchozím snímku a porovnává ji s tou aktuální. Pokud dojde k přesažení 20% tolerance, je algoritmus spuštěn znovu. V případě, že znovu dojde k přesažení, je algoritmus spuštěn ještě jednou, a to naposled. Pokud i poté dojde k přesažení, jsou pozice markerů uznány za správné a dále přichází na řadu buďto druhý obranný mechanismus, nebo uživatelská korekce.

Druhým obranným mechanismem je použití tzv. poloměru oříznutí. Tento algoritmus vyžaduje pro svou funkci znalost středů markerů z předchozího snímku. Uživatel si definuje poloměr oříznutí v jednotkách pixelů. Algoritmus na stávajícím snímku vynesou kružnice o stanoveném poloměru ze středů z předchozího snímku a pouze pixely uvnitř vymezených kružnic podstupují všechny výše zmíněné procesy, pixely mimo tyto kružnice nejsou v úloze vyhodnocení brány v potaz, jak ukazuje obrázek 5.8 z GUI na vedlejší straně.

Již byla řeč o minimálně 5 nastaveních, které se významně podílí na detekci markerů na každém snímku. Jsou to tři tóny RGB, tolerance a poloměr oříznutí (kontrolní mez pro mezimarkerovou vzdálenost je nastavena pevně softwarem). Jelikož se všech pět údajů v čase



Obrázek 5.7: Princip *k-means* algoritmu krok po kroku – první dvě iterace. Černé křížky označují vypočtené středy množin předpokládaných dvou skupin bodů. Modrou a červenou barvou separovaná kolečka potom značí body příslušící jednotlivým množinám.



Obrázek 5.8: Důvod implementace poloměru oříznutí – bílé pixely uvnitř kružnic jsou pixely, se kterými se počítá při detekci středů, šedé pixely jsou v tomto výpočtu ignorovány, protože leží mimo oblast poloměru oříznutí.

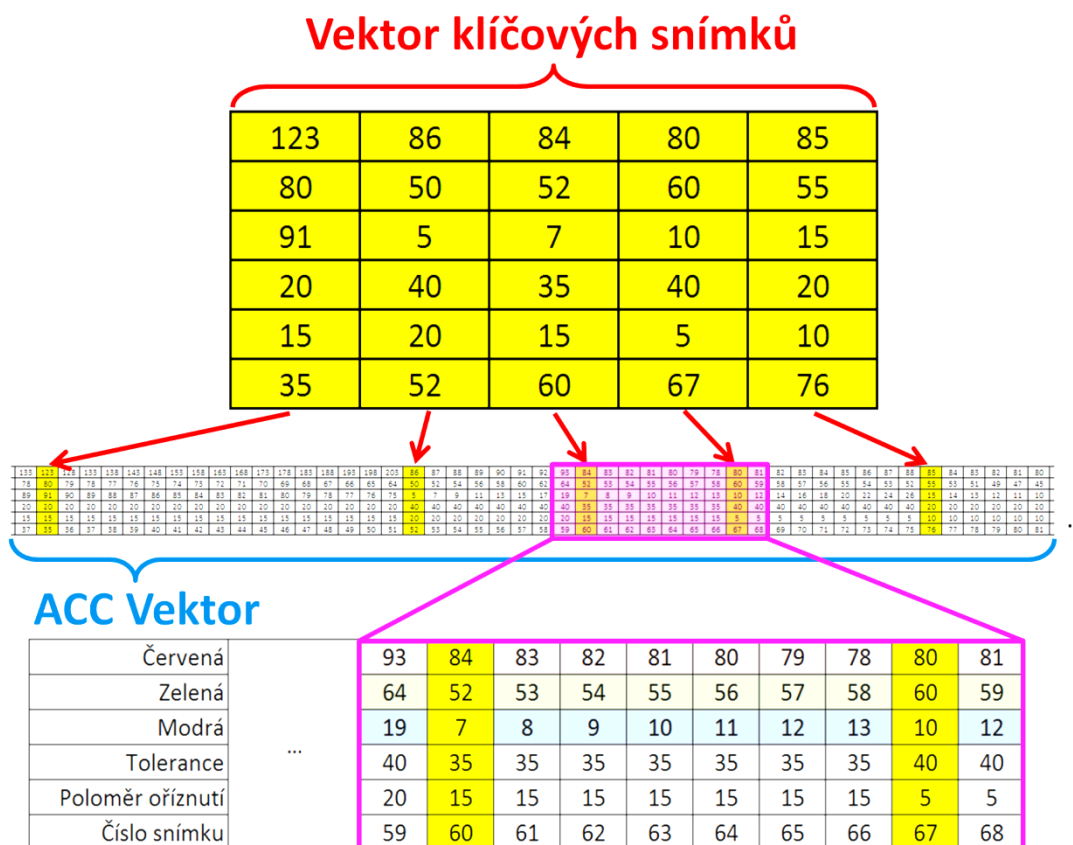
mění dle uvážení uživatele, vyhodnocujícího video, bylo nutné zavést feature, kterou jsem označil názvem „vektor klíčových snímků“. S pojmem klíčové snímky se jistě mnozí setkali v programech pro práci s videi a tvorbu animací. Na pozicích klíčových snímků jsou uloženy vlastnosti nějaké vrstvy nebo předmětu (pozice na x, y, rotace, průhlednost apod.). Na všech snímcích mezi těmito klíčovými snímky se obvykle tzv. prokládá (*tweening*), většinou lineárně, tzn., že výše zmíněné vlastnosti postupně přecházejí z jednoho nastavení do druhého.

Implementace lineárního přechodu mezi klíčovými snímky by v našem případě byla obtížná a zcela nepřínosná, proto jsem zvolil jednodušší variantu bez přechodu. Všechny hodnoty jsou zachovány až do dalšího klíčového snímku, tzn., buďto než uživatel nějakou z hodnot změní, nebo do konce videa. Toto uspořádání přináší krom jiného také tu výhodu, že v případě pádu skriptu není nutné analýzu opakovat zcela od začátku – SW totiž umožňuje i v případě nečekaného selhání skriptu vektor klíčových snímků uložit do textové podoby na disk a zpětně jej načíst, takže uživatel o žádné změny nepřijde.

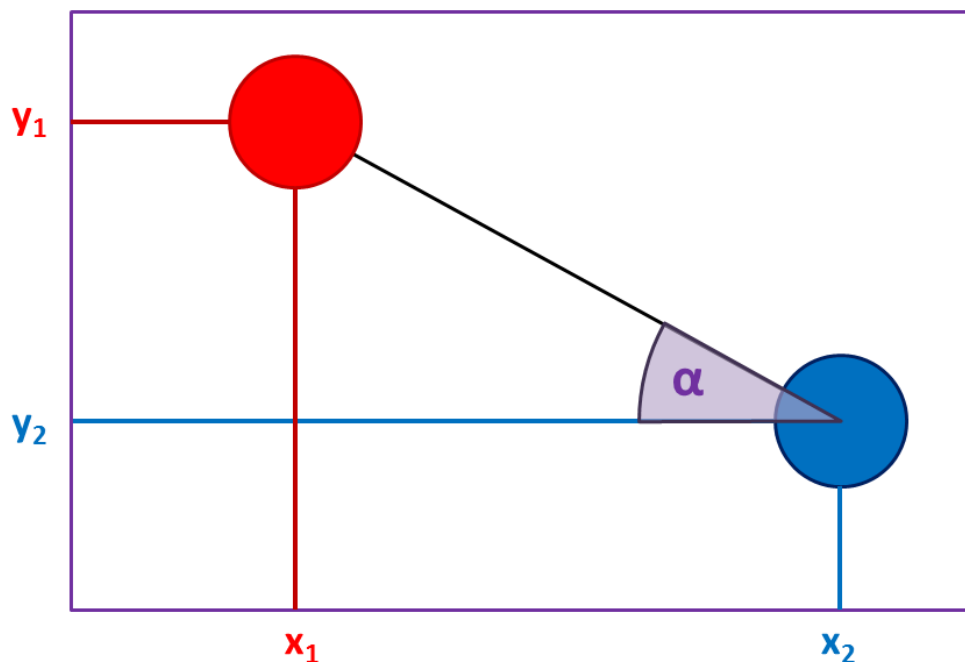
Závěrem představuji implementaci algoritmu adaptace barev markerů, kterou jsem nazval ACC (Adaptive Color Change). Barvy markerů se totiž v závislosti na jejich osvětlení mění rapidně a uživatel tak musí velmi často barvu znovu naklikávat (změna osvětlení, změna polohy plošiny, náklonu, stín obsluhy, reflexe diod, odrazy světla od barevných povrchů...). Tuto potřebu z velké části eliminuje ACC algoritmus.

Princip je následující: Software před zobrazením středů na dalším snímku vyše tzv. sondy, kdy s aktuálním nastavením RGB, tolerance a poloměru oříznutí zkusí, jaké pixely najde na dalším snímku videa. Z těchto pixelů si vytáhne medián tónů R, G a B a pro následující snímek automaticky přenastaví hledanou barvu na tyto hodnoty. Teprve potom spustí detekci pro následující snímek naostro s upravenými hodnotami a takto pokračuje až do vybraného snímku. Výsledkem je, jak název napovídá, adaptace softwaru na změnu barev markerů v čase. SW se snaží sledovat, jakým směrem se barva markerů mění, a přizpůsobuje tomu nastavení tónů RGB tak, aby uživatel nemusel co 10 snímků korigovat hledanou barvu. Samozřejmě se detekce ani tak neobejde zcela bez zásahu obsluhy. V případě kontinuální změny barvy markeru mají ACC algoritmem detekované pixely tendenci postupně „técť“ směrem od středů markerů, až v kritickém okamžiku začne chytat okolí. K tomu však dochází zpravidla až po cca 300+ snímcích.

Vzhledem k principu funkce ACC algoritmu je nasnadě, že bude třeba mít k dispozici matici, obsahující údaje o RGB hodnotách pro každý snímek videa. Na pozadí za vektorem klíčových snímků tedy běží ještě ACC vektor, který shromažďuje údaje o RGB, toleranci a poloměru oříznutí pro všechny snímky videa. ACC vektor je podřízený vektoru klíčových snímků, se kterým přímo pracuje uživatel, tzn., že pokud ACC vektor narazí na klíčový snímek, modifikuje si hodnoty podle něj a nepřepisuje jej. Komunikace mezi oběma vektory (maticemi) funguje bezproblémově a ACC vektor je možné zrekonstruovat jen na základě znalosti vektoru klíčových snímků, který si uživatel může uložit na disk v textové podobě, a nahraného videa. Nevýhodou je občas dlouhá prodleva mezi zobrazením náhledu detekce. Aby bylo možné náhled detekce zobrazit, je třeba zprocesit všechny snímky od posledního známého klíčového snímku. Největší komplikace tak například nastanou, pokud má uživatel klíčový pouze první snímek videa a například omylem posune slider na konec videa. ACC algoritmus musí projít všechny snímky od začátku až do požadovaného snímku, což zabere spoustu času. Aby byla čekací doba v budoucnu zkrácena, je již dopočítaných známých hodnot využito, a v případě například uživatelova návratu na snímek, kde již ACC hodnoty existují, je náhled detekce ihned zobrazen za pomoci těchto údajů. Při restartu GUI je však ACC vektor smazán a čekací doby resetovány.



Obrázek 5.9: Znárodnění hierarchie vektoru klíčových snímků a ACC vektoru; Žlutě jsou vyznačeny klíčové snímky a všechno mezi tvoří společně s klíčovými snímky ACC vektor. Růžově vyznačený úsek byl přiblížen, aby bylo vidět, jak se můžou v ACC vektoru mezi dvěma klíčovými snímky měnit hodnoty R, G, a B, přičemž Tolerance a Poloměr oříznutí zůstává až do dalšího klíčového snímku konstantní.



Obrázek 5.10: Princip výpočtu úhlu mezi markery. Červené a modré kolečko označují středy markerů, fialový rámeček ilustruje velikost rastru, α je hledaný úhel mezi markery a vswlice a horizontály označují xové a ysouřadnice středů markeru.

Poslední věc, kterou je třeba zmínit je tlačítko pro vypuštění barvy markeru. Občas se subjekt příliš rychle pohne a markery se příliš rozmažou, nebo nám na chvíli zvíře vyjde ze záběru, takže jeden nebo více markerů ztratíme. V takových případech nemá smysl cokoliv detekovat a je vhodné detekci pro několik dalších snímků vypustit.

5.2.5 Výpočet úhlů

Jakmile máme na obrázku nalezené středy markerů, nalezení úhlu je triviální matematickou záležitostí. Úhly se spočtou pomocí tangenty, přičemž odvěsny trojúhelníku tvoří rozdíl xových a yových souřadnic markerů, viz obrázek 5.10 nahoře a vzorec 5 níže.

$$\alpha = \text{tg} \left(\frac{y_2 - y_1}{x_2 - x_1} \right) \quad (5)$$

Zde α značí úhel mezi markery, x je xová souřadnice a y yová souřadnice markeru.

5.2.6 Export dat do výstupních grafů a tabulek

K exportu detekovaných úhlů slouží GUI na obrázku 5.15 v kapitole 5.2.7 – *Popis navrženého GUI*, strana 41. Hlavním výsledkem celého výše popsaného procesu je textový dokument, obsahující pro každý snímek z videa spočtený úhel mezi markery.

V tuto chvíli byly popsány metody a postupy, jejichž pro nás významným výstupem jsou dva textové soubory – záznam časových razítek pro video (kdy jsme který snímek daného videa pořídili) a dokument, obsahující úhly mezi markery pro každý jednotlivý snímek videa. K vynesení grafu závislosti úhlu zvířete a úhlu plošiny v čase nám chybí úhel plošiny. Ten je v momentě zahájení nahrávání zaznamenáván na počítači, kterým se ovládá plošina. Podrobnější popis naleznete v *kapitole 5.3 - Komunikace mezi PC a plošinou během nahrávání*.

Za účelem exportu veškerých statisticky významných veličin a obrázků grafů jakož i textových souborů, obsahujících všechny relevantní informace ke zreprodukování výsledků, bylo do nahrávacího GUI na obrázku 5.11 vedle přidáno tlačítko „Export“.

Tlačítko uživatele posílá do dalšího okna, kde je dotázán na 2 až 3 soubory, v závislosti na tom, zdali zaškrtně checkbox o potvrzení vlastnictví textového dokumentu s časovými razítky. Tento dokument totiž často nebývá zapotřebí a při prvních měřeních k dispozici dokonce vůbec nebyl. V případě, že měření probíhalo za víceméně konstantního frameratu, případně framerate fluktoval se stejnou odchylkou na obě strany od střední hodnoty, respektive mediánu, je možné úhel zvířete s úhlem plošiny časově sesynchronizovat jednoduchým roztažením dat na celou šířku delšího z obou záznamů. Tímto pravděpodobně dochází ke zkreslení celkového času měření, nicméně je zachována relevance průběhů vůči sobě samým.

Vlastnictví časových razítek však definitivně určuje přesnost časové synchronizace v grafech. Po výběru dokumentu s exportovanými úhly, průběhem naklápění plošiny a případně tedy časovými razítky, je uživateli v náhledovém okně zobrazen výstupní graf s označenými úseky, které software byl schopen interpolovat. Mezi jednotlivými interpolacemi lze vybírat v menu a nakliknutá interpolace je v grafu zvýrazněna. Uživateli je povoleno vybírat zatím ze tří typů interpolace pro označený úsek – lineární, spline a kubická. Případně může interpolaci zcela vynechat, pokud je vidět, že nahrazuje příliš dlouhý úsek a mohla by zanechat chybu do exportovaných dat. Krom toho umožňuje GUI uživateli ještě měnit pozici legendy, která je občas nevhodně umístěna a není přes ni vidět graf.

Když je uživatel s náhledem grafu spokojen, stiskne tlačítko pokračovat. Toto GUI jsem navrhl pro účely zpracování měření z periodických scénářů pohybu plošiny. Nelze jej využít pro hodnocení neperiodických měření. Software se pokusí graf rozsegmentovat na jednotlivé periody a pokud se mu to povede, zobrazí uživateli pro kontrolu v náhledovém okně graf po segmentaci s vyznačenými úseky, ohraničujícími jednotlivé periody. Poté jsou z jednotlivých period extrahovány informace o veličinách, popsaných v *kapitole 6.2 - Volba sledovaných ukazatelů* u obrázku 6.1 dále (str. 45), a zapsány do textového souboru nesoucího název původního souboru s přídomkem „* - údaje“. Časová osa, průběh pohybu plošiny a průběh úhlu u zvířete je pro každou periodu zapsán do dalšího textového dokumentu, opět respektující původní název, s přídomkem „ - periody“. Zároveň je sejmuto screenshot grafu ve formátu *.bmp a screenshoty diagramů vzájemných závislostí úhlu plošiny a úhlu zvířete s vypočtenou délkou trajektorie v názvu grafu pro všechny periody.

Nakonec je uživatel dotázán na soubor typu *.xls (*.xlsx), do něž mají být veškeré statistické hodnoty pro daný subjekt zapsány. Software automaticky zapisuje v předdefinovaném formátu hned na první z listů excel souboru. Nejprve však vždy vyšle sondu pro zjištění, zdali aktuální list již neobsahuje nějaké informace, a pokud ano, najde jejich polohu a kategoricky umístí nová data až za ně, nechaje mezi nimi mezeru pro snadnější orientaci. Uvedená metoda eliminuje nutnost neustálé záměny listu či souboru s každým

hodnoceným subjektem z důvodu automatického přepisu indukovaným samotným MatLabem® a zároveň umožňuje mít všechna data k určité sadě měření pohromadě, což šetří čas při jejich kompletaci a přeuspořádání.

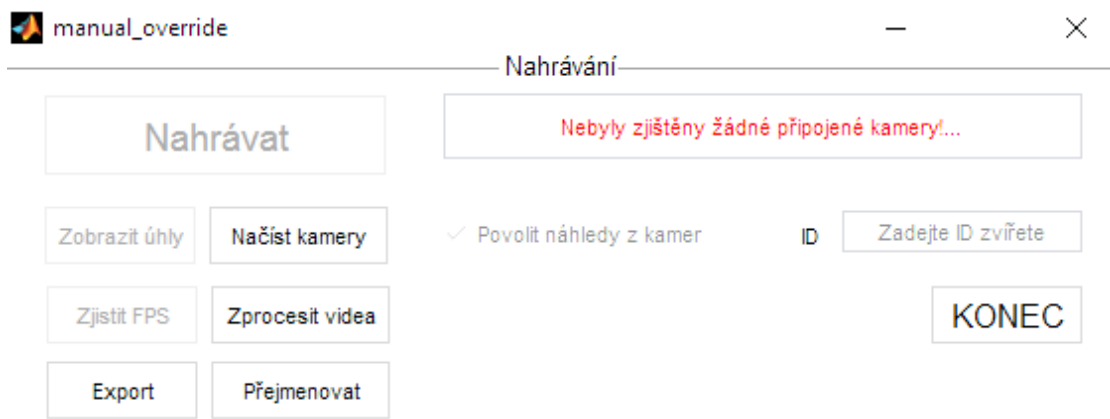
5.2.7 Popis navrženého GUI

Na obrázku 5.11 níže je patrné uspořádání GUI pro nahrávání videí. SW je schopen fungovat i bez připojených kamer, jak je na obrázku patrné, a umožňuje uživateli první předzpracování nahraných videí. Tlačítka pracující s kamerami jsou v této fázi inaktivní. Dialogové okno zobrazuje uživateli vše, co je podstatné, například při zpracovávání videí zobrazuje postup formou dvou řádků, na jednom ukazuje postup ořezávání videí (hotovo/celkový počet) a na druhém postup časové synchronizace ve stejném stylu. Zároveň uživatele informuje při selhání komunikace mezi počítači nebo o příliš nízkém či vysokém frameratu kamer. Po zahájení nahrávání vypisuje že bylo nahrávání zahájeno a po vypnutí jasně informuje, kdy se co vypíná a co SW dělá (například přesun videí z SSD na HDD, viz výše).

Nahrávání není povoleno, dokud uživatel nevyplní políčko s ID subjektu – ID je totiž součástí názvu nahraných videí a veškerých textových souborů k videu vztaženým. Primárním atributem však zůstává doba zahájení měření (ve formátu rok_měsíc_den hodina:minuta:sekunda).

GUI dovoluje uživateli zapnout náhled ke kamerám, a standardně zůstává toto nastavení zapnuté, protože je často zapotřebí náhledy sledovat při ukotvování subjektu na plošinu a při samotném měření.

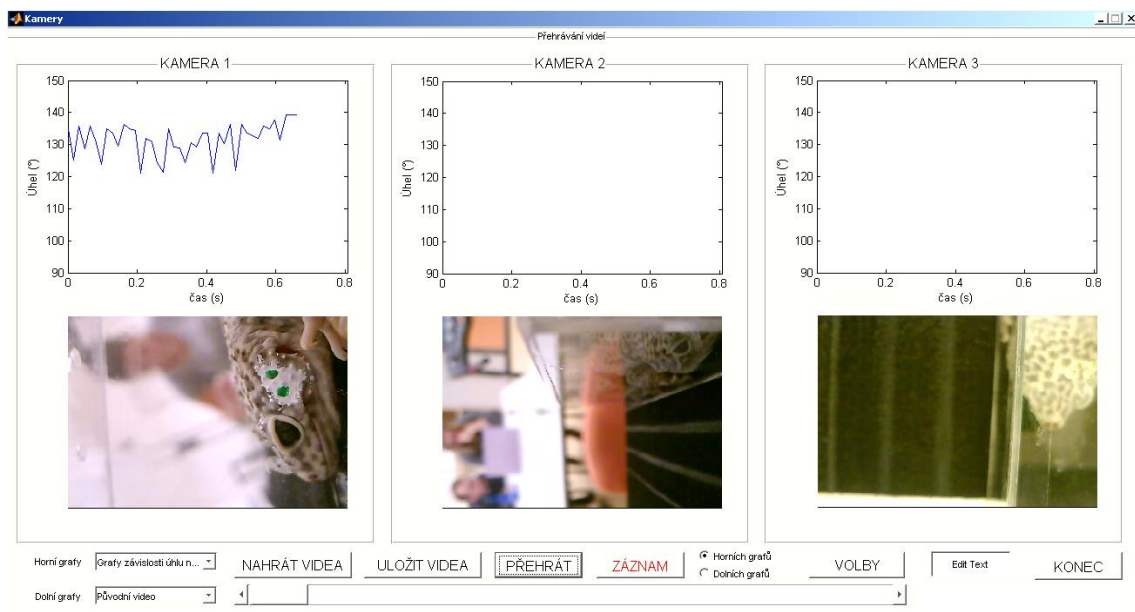
Tlačítko „načíst kamery“ poslouží v případě výpadku některé kamery, stejnou funkci bude mít ale i restart GUI. „Zobrazit úhly“ vyžaduje některé další textové soubory, jako záznam pohybu plošiny pro dané měření a vyexportované úhly z videa, a poskytuje náhled výsledného grafu. „Zjistit FPS“ umožňuje narychlo zjistit hrubý odhad fps. Po stisknutí tlačítka „Zprocesit videa“ je uživatel vyzván k výběru všech nahraných videí, které si přeje zprocesit, a postprocessing je zahájen. Tlačítko „Přejmenovat“ vzniklo z důvodu občasného chybného zadání ID subjektu. Jelikož jsou k videím vztaženy další textové soubory nacházející se na různých lokacích v počítači, toto tlačítko umožňuje hromadné přejmenování všech spřažených souborů. Funkce tlačítka „Export“ byla popsána v kapitole výše.



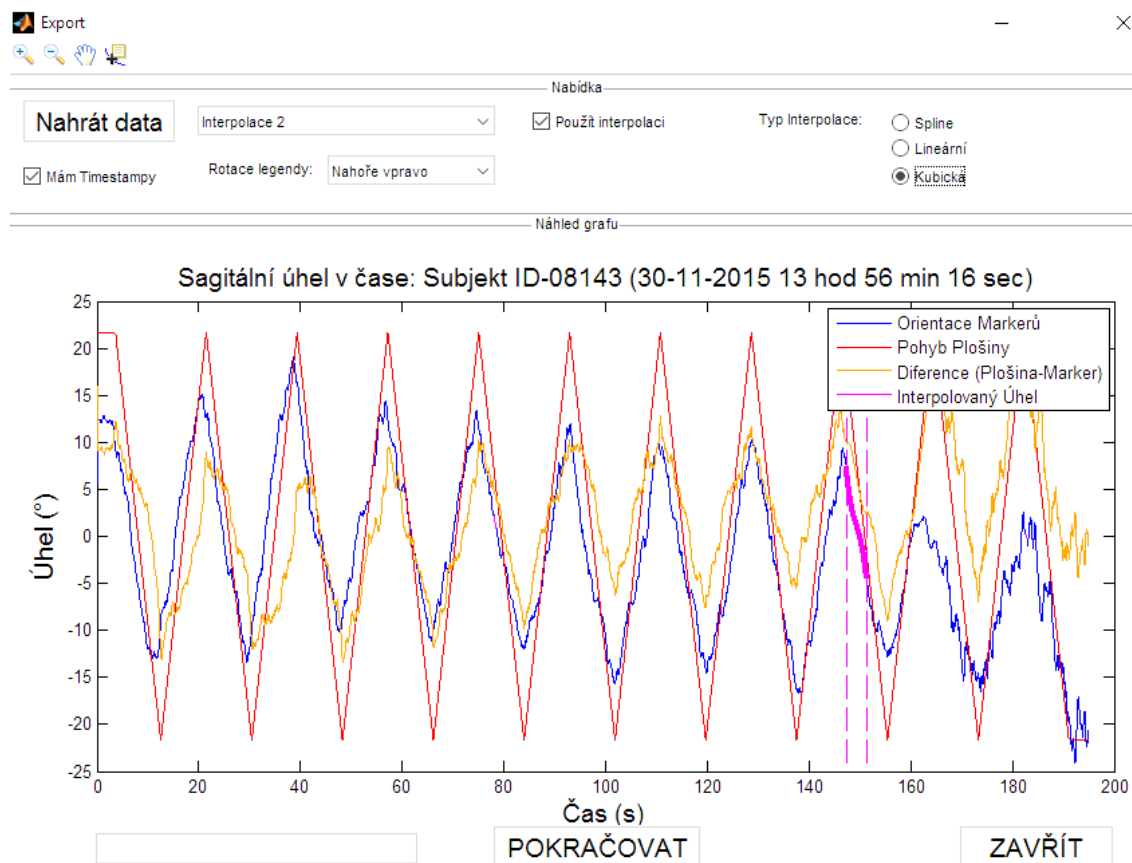
Obrázek 5.11: Ukázka GUI pro nahrávání z kamer po dokončení postprocessingu videí bez připojených kamer.

GUI, které se zobrazí po kliknutí na tlačítko „Export“ popsané výše, je vidět na obrázku 5.13A na vedlejší straně. Vlevo nahoře je tlačítko pro nahrání dat (úhly, pohyb plošiny a případně časová razítka), pod ním je checkbox pro potvrzení vlastnictví časových razítek. Vpravo od něj je menu pro selekci interpolovaného úseku v náhledu grafu a pod ním možnost výběru umístění legendy. Uprostřed nahoře je checkbox pro vypnutí/zapnutí označeného interpolovaného úseku a vpravo od něj se nachází výběr typu interpolace vybraného úseku. Uprostřed GUI je vidět náhled grafu před segmentací. Kontrolní vykreslení po segmentaci je patrné na obrázku 5.13B. Dole vlevo je dialogový box zobrazující veškeré procesy a postupy uživateli a tlačítkem „Pokračovat“ uživatel potvrdí správnost grafu a na disk jsou uloženy všechny screenshoty a dokumenty popsané výše.

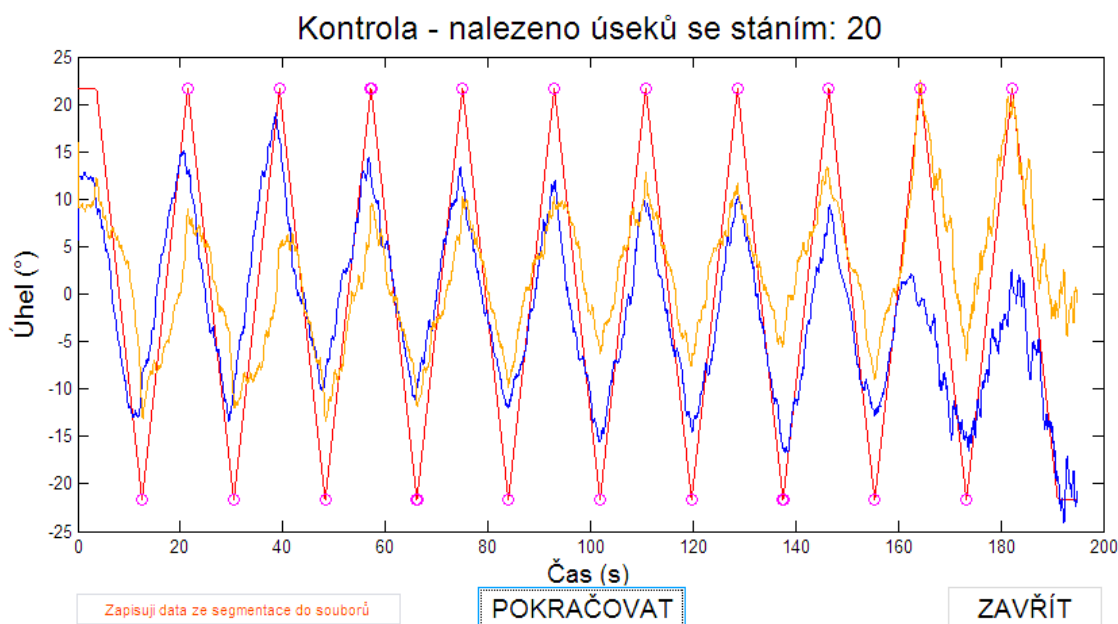
Na obrázku 5.12 níže je možné vidět, jak vypadá okno pro současné přehrávání 3 videí. Horní a dolní menu umožňuje uživateli zvolit, co se ve třech vedlejších oknech bude přehrávat. Na výběr jsou možnosti: nic, originální video, video s barevně označenými markery, binární maska detekce a úhly v čase. Na obrázku 5.12 jsou zrovna nahoře vybrány úhly v čase a dole je originální video. V horních posledních dvou oknech není vidět úhel z důvodu absence markerů (měření bylo zaměřeno jen na sagitální rovinu). Před finálním exportem úhlů je video z prvních dvou oken otočeno o 90° tak, aby se horizontální rovina shodovala se zemskou horizontálou. Zobrazené úhly na obrázku 5.12 jsou ještě před touto úpravou. Z tlačítek stojí za povšimnutí tlačítka „Volby“ a „Uložit videa“, která obě odkazují na další části GUI. Tlačítko pro záznam zde slouží pro prezentační účely – umožňuje z videí vyseknout třeba jen určitý úsek, nebo nahrát část vývoje grafů v čase do separátního videosouboru.



Obrázek 5.12: Ukázka GUI pro současné přehrávání tří videí a náhledy k úhlům.

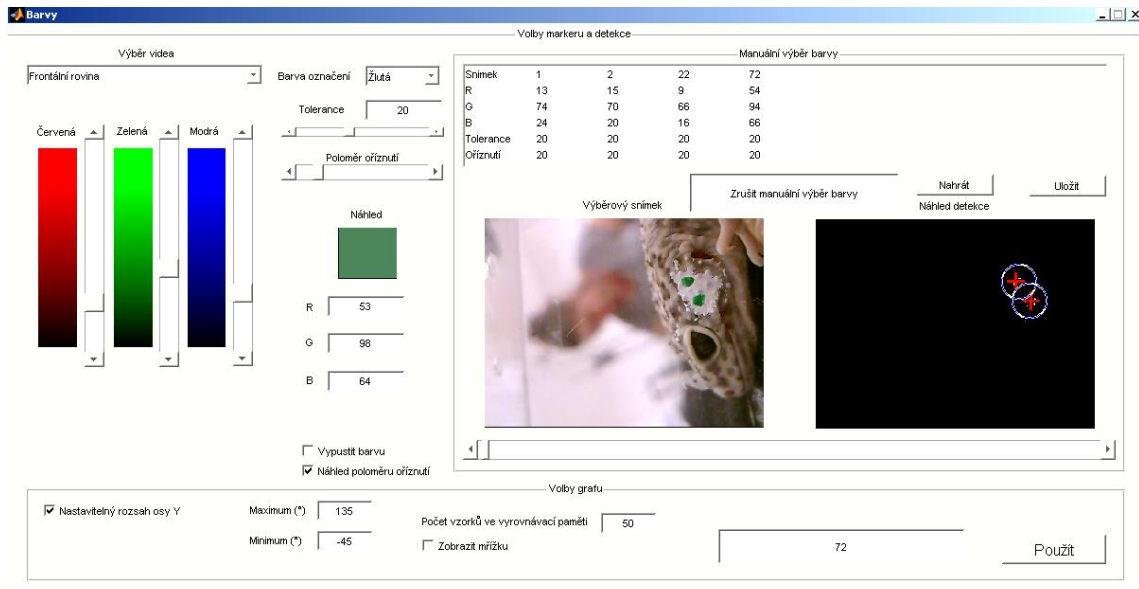


(A)



(B)

Obrázek 5.12: Ukázka GUI pro selekci a úpravu automatických interpolací a export grafických a statistických údajů; (A) - před segmentací, (B) - kontrolní vykreslení po segmentaci.



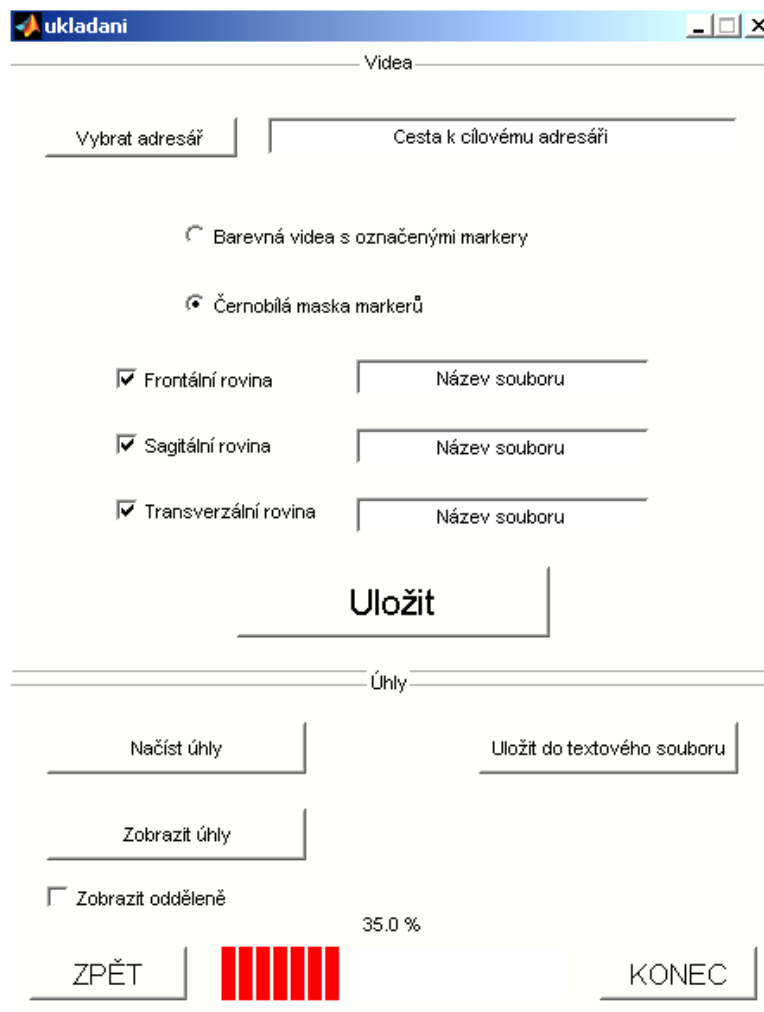
Obrázek 5.13: Ukázka z GUI pro nastavení detekce markerů.

Nejdůležitější procesy se odehrávají v GUI, které se zobrazí po stisknutí tlačítka „Volby“, viz obrázek 5.14 nahoře. Zde menu vlevo nahoře slouží k přepínání mezi videi, RGB slidery nemají žádný zvláštní význam, stejně jako editboxy s tóny, které s náhledem pouze informují o vybrané barvě. V levé části GUI je ale možné nastavit toleranci a poloměr oříznutí, o nichž byla řeč v kapitole 5.2.4 – *Algoritmy detekce markerů*. Níže si můžeme všimnout checkboxu pro vypuštění barvy a pro náhled poloměru oříznutí na binární masce detekce. Tento náhled je na obrázku 5.14 zapnutý a vpravo je možné vidět, jak kružnice v masce vypadají. Co již moc dobře vidět není, jsou detekované pixely, které uvnitř kružnic září bíle a mimo kružnice šedě. Slider pod náhledy slouží k posunu mezi snímky ve videu a široký editbox vpravo dole zobrazuje číslo snímku, na kterém se uživatel nachází, což přichází vhod při debuggingu. Tabulka vpravo nahoře je tabulka vektoru klíčových snímků. Pod ní jsou tlačítka na její export či import. V dolní části GUI se nachází možnosti související s GUI na předchozím obrázku, jedná se zejména o rozsah yové osy u vývoje úhlů v čase, zobrazení mřížky nebo nastavení šířky oken.

Na obrázku 5.15 na vedlejší straně je vidět GUI, které se zobrazí po stisknutí tlačítka „Uložit videa“. V zásadě celá horní část GUI se zabývá přeuložením modifikovaných videí na disk pro prezentační účely. Dolní část umožňuje na základě zhotoveného či importovaného vektoru klíčových snímků načíst úhly. Po stisknutí „Načíst úhly“ se uživateli zobrazí nahrávací obdélník, jak je na obrázku 5.15 patrné. Po nahrání úhlů se uživateli zobrazí možnost jejich zobrazení či uložení do textového souboru.

5.2.8 Filtrace pohybového zkreslení

Veškeré algoritmy pro korekci obrazu z hlediska pohybového zkreslení jsou výpočetně, tj. také časově, náročné a ty méně náročné vyžadují znalost převažujícího směru zkreslení. Tuto znalost většinou k dispozici explicitně nemáme a museli bychom ji SW podávat přes speciální volby. Vzhledem k časové náročnosti tedy tento typ filtrace není softwarem podporován. Je pro nás jednodušší místa, kde se subjekt rychle pohnul, jednoduše vynechat a poté tyto úseky ve finálním exportu interpolovat kubickým splinem. Většinou se jedná o úseky pokrývající širší do 5 snímků, tedy interpolace hodnoty velmi spolehlivě nahrazuje.



Obrázek 5.14: Ukázka GUI pro export úhlů a videí.

5.3 Komunikace mezi PC a plošinou během nahrávání

Tato podkapitola popisuje komunikaci mezi nahrávacím a ovládacím PC v momentě, kdy je inicializováno a ukončeno nahrávání.

Domnívám se, že jsem dosud explicitně nezmínil, že je celý systém řízen ze dvou různých počítačů. Jeden slouží pouze k nahrávání videí a komunikaci s řídicím počítačem, druhý slouží k řízení plošiny a komunikaci s nahrávacím počítačem. Nahrávací počítač označíme zkratkou *CMS PC* (Camera Measurement System PC) a řídicí *ERP PC* (Electronic Revolving Platform PC). Podívejme se podrobněji na to, co se děje po stisknutí tlačítka „Nahrávat“ na CMS PC.

Než bude řeč o detailech ohledně komunikace, je na místě upozornit na 2 důležité věci. Zaprvé – veškeré soubory jsou pojmenovány tak, že jejich název obsahuje jednak zadané ID měřeného subjektu a jednak přesné datum a čas zahájení měření – ID subjektu však zná pouze CMS PC. Zadruhé – při zahájení nahrávání na CMS PC dojde k výraznému zpomalení chodu systémového času na CMS PC, a přestože se čas po nějaké době sám za asistence připojení k internetu zase opraví, tato doba může trvat až několik minut, a to pouze za předpokladu, že během ní není zahájeno další měření. Je tedy zřejmé, že v tomto okamžiku disponujeme na

začátku každého měření dvěma různými časovými údaji, které lze pro pojmenování videí využít – chybný údaj z CMS PC a správný údaj z ERP PC.

Z uvedeného vyplynula nutnost nějak před nahráváním předat ERP PC údaj o ID subjektu a zpětně poslat CMS PC správný čas a datum pro vytvoření videosouborů. Problém je vyřešen následovně.

V okamžiku zahájení nahrávání nejprve CMS PC do sdíleného adresáře zapíše do předpřipraveného komunikačního dokumentu údaj o tom, že chceme nahrávat a za něj připiše ID subjektu. Poté se na CMS PC spouští timer s periodou 0,1 sekund, trvající celkem 2 sekundy. Během této doby je čten jiný komunikační soubor, ve kterém se záhy objeví čas a datum, ke kterému se hned dostaneme.

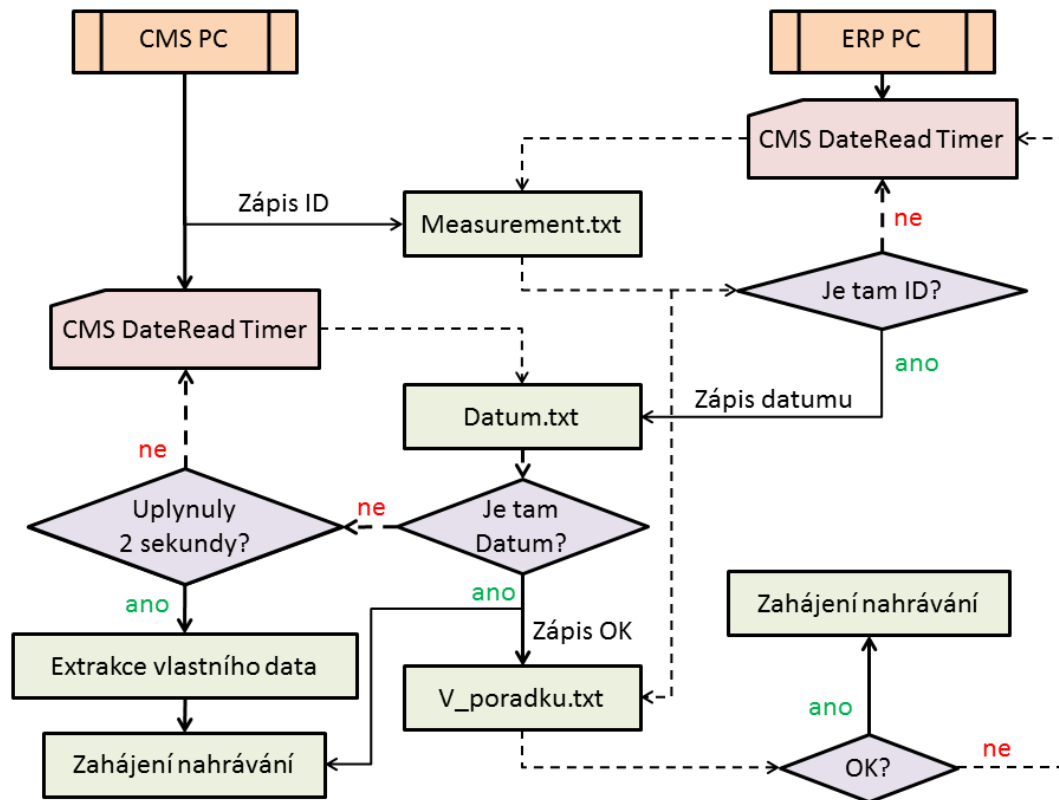
Na ERP PC běží neustále timer s periodou 0,01 sekund, který k souboru s údaji o tom, zdali se měří, přistupuje 100x za vteřinu a kontroluje jeho obsah. V momentě, kdy je do souboru vypsáno, že bylo zahájeno nahrávání, si přečte ID subjektu a od sebe extrahuje systémový čas a datum a zapíše jej do komunikačního souboru s datem. Zároveň u sebe v rámci svého timeru začne kontrolovat třetí komunikační soubor, obsahující zprávu o tom, že je vše připraveno a přichystáno a můžeme skutečně začít měřit.

Pokud se v souboru s datem datum neobjeví do dvou sekund, je timer na CMS PC zastaven, uživatel je formou dialogového okna spraven o pravděpodobně narušeném spojení mezi CMS a ERP PC a CMS PC si pro videa extrahuje čas a datum od sebe (předpokládáme, že si chceme jen nahrát pokusné záznamy a nejde nám teď o pohyb plošiny – toto nám umožní nahrávat videa a odlaďovat software i s vypnutým ERP PC). Pokud do dvou sekund zjistíme datum v komunikačním souboru, vezmeme si jej a nyní CMS PC nic nebrání v předpřipravení videosouborů k nahrávání, což je operace, která může trvat i několik sekund. Jakmile je vše připraveno k nahrávání, je o tom těsně před zapnutím poslední kamery do třetího komunikačního souboru zapsán záznam.

ERP PC po detekci tohoto záznamu zakládá vlastní textový dokument, obsahující ID subjektu a odeslaný čas zahájení měření, do kterého začne ukládat pohyb plošiny. V rámci svého timeru nepřestává kontrolovat první komunikační soubor s informací o tom, zdali měříme, nebo ne.

V momentě stopnutí nahrávání je těsně před vypnutím první z kamer tento soubor CMS PC přepsán údajem o ukončení nahrávání, ERP PC si jej přečte, ukončí u sebe záznam pohybu plošiny a odešle jej k nahraným videím na CMS PC. CMS PC ještě dokončí drobné úpravy, jako zápis mediánů fps a snímků k ořezání do „*cutoff.txt*“, jak bylo zmíněno výše, a poté všechny soubory přesune z SSD na HDD a informuje uživatele o tom, že může začít další nahrávání.

Celý proces jsem se pokusil ilustrovat posledním, asi nejkompexnějším, diagramem na obrázku 5.16 na vedlejší stránce. Paralelně běžící časové smyčky timerů jsem zde označil přerušovanými šipkami a funkce timer spouštějící jsou v obdélníčcích s uříznutým rohem. Obdélníčky ve sloupci uprostřed si můžeme představit jako zmíněné komunikační textové soubory.



Obrázek 5.15: Schéma komunikace mezi CMS PC a ERP PC při zahájení nahrávání.

5.4 Závěr kapitoly

V kapitole bylo představeno a probráno, jak se videa nahrávají, proč jsme použili konkrétní formát videa, jak je nutno videa po nahrání zpracovat a jak odhadujeme framerate. Dále byla značná pozornost věnována algoritmům detekce markeru, kde byla řeč o nastavení RGB valérů, toleranci, poloměru oříznutí, *k-means*, kontrole mezimarkerové vzdálenosti a v neposlední řadě byla též zmíněna důležitost vektoru klíčových snímků a ACC vektoru jakož i tlačítka pro vypuštění barvy detekce. Dále jsme si představili funkce a použití mnou navrženého GUI, abychom viděli, jak rozhraní v praxi vypadá, a popsali si funkce některých důležitých tlačítek. Nakonec byla popsána komunikace mezi jednotlivými PC během nahrávání.

Výše uvedený navržený software, společně s hardwarem plošiny, bude otestován a využit pro měření konkrétních živočichů, aby byla ověřena funkčnost mých návrhů v praxi. Zmíním však zde, že i během vývoje byla aplikovatelnost návrhů průběžně ověřována měřením jednotlivých živočichů (plazů), aby se předešlo problémům v SW a HW v průběhu vědecky důležitých měření.

6 Experimentální testování plošiny

6.1 Volba měřených živočichů

Volba konkrétních druhů živočichů a vlastní měření probíhalo ve spolupráci s Přírodovědnou fakultou UK, která primárně požaduje studium pohybových aktivit živočichů jako odezvu na změnu polohy těla živočicha. Při úplně prvních měřeních proběhlo několik koncepčních měření na *gekončících nočních*, během kterých jsme zjistili, že je náš dosavadní systém upevnění zvířat na plošinu zcela nedostačující a je třeba jej podstatně vylepšit. Zároveň jsme odhalili několik jiných nedostatků, týkajících se nastavení kamer a pohybu plošiny. První měření na gekonech tedy proběhly, avšak do jakékoliv statistiky tyto nahrávky zahrnuty nejsou.

Ve druhé fázi, nedlouho po prvním měření, jsme testovali systém a navržené metody na žábách, konkrétně 2 kusech *ropuchy obecné*. Ani tyto nahrávky jsme ve finálním vyhodnocení neuvažovali, a systém v rámci přeuspořádání pro zlepšení kvality záznamů dostal významných změn.

Třetí série měření zahrnovala okolo 10 kusů *gekončků nočních*. Přestože se mělo jednat o první ostré měření, kvůli nalezení několika chyb v konstrukci systému a nahrávání videí, nekompletního záznamu pohybu plošiny a stále ještě ne zcela dořešenému upevnění zvířat na plošinu byla i tato série měření při statistickém hodnocení zanedbána.

Až 4. skupina měřených živočichů, sestávající z 11 *ropuch obecných (Bufo Bufo)* byla proměřena tak, že se výsledky z měření staly pro naše účely prezentovatelnými. Následné měření obsahovalo na 6 rozličných druhů žab (*Bufo Bufo*, *Bufo Regularis*, *Phrynomerus Microps*, *Pyxicephalus Edulis*, *Dyscophus Guineti* a *Phrynomerus Bifasciatus*) a měření s těmito obojživelníky bylo zakončeno 14 dalšími kusy *ropuchy obecné*. Na těchto sériích byla zjištěna nutnost dispozice předpřipravenými automatizovanými scénáři pro budoucí měření. Zároveň jsme zde poprvé testovali reakci na sinusový pohyb plošiny.

Po sérii měření se žábami bylo odladěno ještě několik problémů s časovou synchronizací dat, kterou bylo pro naměřené žáby nutno opravit ručně záznam od záznamu, a která teď probíhá automatizovaně na základě časových razítek z kamerových záznamů. Zároveň po zkušenostech se žábami byly do ovládacího SW implementovány jednoduché automatizované scénáře s rampami, tj. pohybem od dorazu k dorazu a nazpět. Tyto scénáře probíhají zcela jednotně pro všechna měření – od této chvíle tak nelze zpochybnit normalizaci měření pro všechny subjekty.

Poté jsme naměřili, pro tuto práci poslední, sérii, tentokrát s *gekončičky nočními*, čítající celkem na 30 kusů. Poslední měření bylo specifické již zavedenou metodikou, kdy scénáře jednak probíhaly při každém měření stejně, čímž byly zajištěny stejné podmínky měření pro každý subjekt, a zároveň se každé zvíře měřilo čtyřikrát – pro dvě různé rychlosti (2,5°/s a 5,0°/s) a před stimulací a po trojnásobné stimulaci dřevěnou špejlí skrze otvory v krabici do kořene ocasu, což mělo imitovat útok predátora, přičemž každé měření trvalo 3 minuty.

6.2 Volba sledovaných ukazatelů

Jelikož se plošina pohybuje s periodou, je možné pro každou periodu vybrat určité oblasti či body, ve kterých extrahujeme hodnoty úhlu z pohybu segmentu zvířete. Tyto

hodnoty je vhodné normovat aktuální hodnotou náklonu plošiny, abychom mohli procentuálně vyjádřit míru kompenzace pohybu plošiny. Jak ukazuje obrázek 6.1 níže, my jsme každou periodu rozdělili na dvě půlperiody – A a B. V každé půlperiodě jsme sejmuli úhel segmentu v době, kdy se plošina zastavila, a v době, kdy se plošina opět rozjela (Δ_1 a Δ_2 ve $^\circ$) a podělili je úhly plošiny ve stejných okamžicích (Ω_1 a Ω_2 ve $^\circ$), čímž jsme dostali kvantifikovatelný poměr, který jsme označili jako δ_1 a δ_2 (%). Další z významných veličin je rozsah pohybu ROM (Range of Movement), který je též vhodné vyjádřit po normalizaci na rozsah pohybu plošiny v daném úseku. Na každou půlperiodu připadá jeden ROM, který je spočten z veličin na obrázku 6.1 dle vzorce 6 a 7.

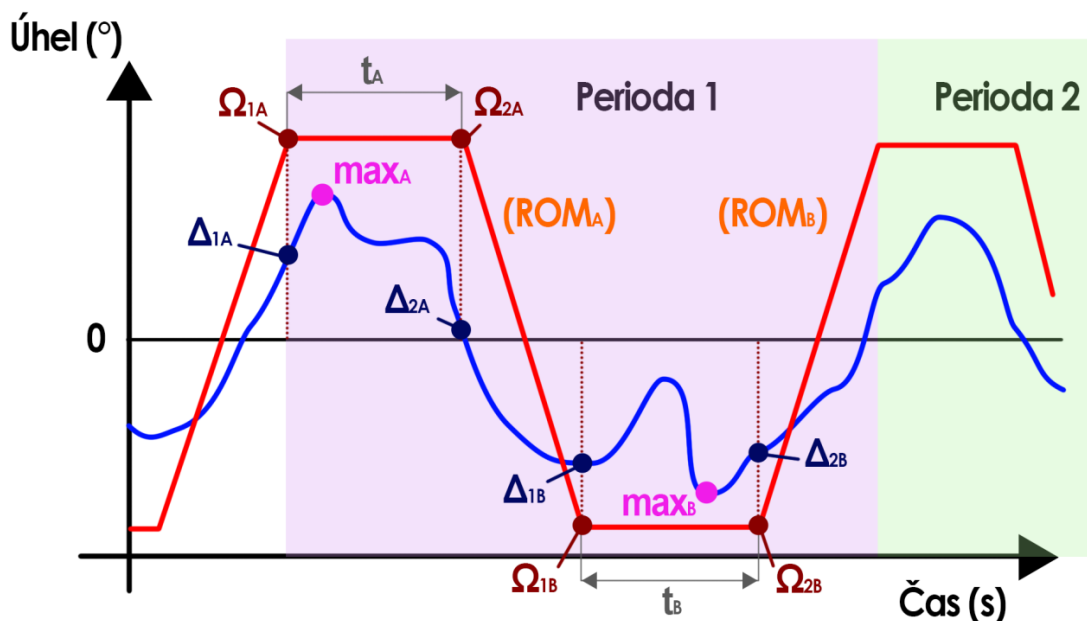
$$ROM_A = 100 \cdot \frac{\Delta_{2A} - \Delta_{1B}}{\Omega_{2A} - \Omega_{1B}} \quad (\%) \quad (6)$$

$$ROM_B = 100 \cdot \frac{\Delta_{2B} - \Delta_{1A}}{\Omega_{2B} - \Omega_{1A}} \quad (\%) \quad (7)$$

Pro některá měření významnou veličinou by rovněž mohla být doba, po kterou plošina před dalším pohybem stála t_A a t_B (s).

V diagramech vzájemné závislosti úhlu zvířete a plošiny počítáme délku trajektorie na každou z period.

Zajímavým a bezesporu vhodným ukazatelem je korelace pohybu zvířete s křivkou pohybu plošiny v sestupném či vzestupném segmentu grafu. Abychom se vyhnuli problémům s nepřesnými hodnotami obou signálů, je na místě použít Spearmanův korelační koeficient, zahrnující v sobě pouze výsledek korelace mezi vlastními diferenciemi jednotlivých signálů, tedy v potaz je brán pouze sklon křivky, nikoliv absolutní hodnoty. Uvedené by se dalo nahradit



Obrázek 6.1: Příklad sledovaných veličin ze záznamu pohybu zvířete v čase. Červenou je pohyb plošiny a modrou pohyb zvířete.

regresí bodů, tvořících pohyb zvířete v těchto úsecích, lineární přímkou za užití metody nejmenších čtverců, a porovnání výsledného úhlu sklonu přímky s úhlem plošiny. Obě metody by měly dávat podobné výsledky, přičemž Spearmanova korelace bude přesnější, jelikož u regrese nahrazujeme nelineární úsek signálu lineární přímkou.

Nakonec dle požadavků Přírodovědné Fakulty Univerzity Karlovy v Praze byl spočten rozdíl úhlu plošiny a úhlu segmentu a výslednou křivku jsme zanesli do již získaných dat. Nabízí se možnost hodnotit délku trajektorie i pro diagramy, kde na horizontální ose zůstane pohyb plošiny, ale na vertikální budeme vynášet právě zmíněnou diferenci.

6.3 Zpracování naměřených dat

Protože byla víceméně celá problematika zpracování naměřených dat popsána v předešlých kapitolách, věnuji tuto podkapitolu stručnému přehledu všech procesů nad pořízenými videi, vedoucích ke konkrétním výsledkům, jejichž hodnocením se zabývá následující kapitola.

Při nahrávání videí je tedy nutno počítat s prodlevou mezi zapínáním a vypínáním jednotlivých kamer. Kvantifikovatelná část záznamu je pouze ta, kdy běžely současně všechny tři kamery. Tomu je přizpůsoben záznam pohybu plošiny, běžící odděleně, avšak synchronně, na ERP PC. Na plošině je umístěno zvíře s namarkovanou hlavou tak, že markery označená rovina leží pokud možno kolmo na osu kamery, z níž pořizujeme pro nás důležité video. Na ERP PC je spuštěn scénář rampového pohybu plošiny a pozici krabičky se zvířetem vzhledem ke kameře je díky volnému magnetickému kotevnímu mechanismu během měření možno uzpůsobovat tak, že je zaručena autenticita vyšetřované roviny.

Do videí během nahrávání nijak zasahovat nelze – jejich ořezání se tedy děje až po jejich nahrání, a to formou speciálního procesu, který po výběru jednotlivých videí inicializuje uživatel až po dokončení všech měření. Data, potřebná pro ořezání a sjednocení frameratu, obsahuje textový dokument, aktualizující se po každém úspěšně provedeném měření. Tento dokument je vstupem výše zmíněného procesu, přičemž výstupem jsou tři videa, shodující se v celkovém počtu snímků a okamžiku zahájení a skončení nahrávání.

Časovou synchronizaci snímků mezi zahájením a ukončením nahrávání zajišťuje záznam časových razítek v době nahrávání jednotlivých videí. Tento textový dokument je využit až o něco dále při exportu grafů a statistických ukazatelů, po dokončení vyhodnocení úhlů.

Vyhodnocení úhlů v čase je prováděno z jiného uživatelského rozhraní. K detekci je využito především tří hlavních principů:

- Manuální selekce požadované barvy pro detekci z náhledu snímku videa
- Nastavení tolerančního rozmezí pro danou barvu
- Nastavení poloměru oříznutí k vyloučení pixelů, ležících mimo oblast zájmu

Každý snímek videa, na němž byla provedena modifikace libovolného z výše zmíněných parametrů, je uložen do vektoru klíčových snímků, odkud lze celý proces vyhodnocení zreprodukovat. Pro zajištění co nejmenší frekvence nutnosti zásahu uživatele do procesu vyhodnocení úhlů je na pozadí ukládán ACC vektor, který je rozšířením myšlenky vektoru klíčových snímků. Tato utilita je unikátní tím, že mezi uživatelem založenými klíčovými snímky

vysílá sondu a hledá detekované pixely, vyhovující parametrům posledního předchozího klíčového snímku. Z RGB složek nalezených pixelů tvoří medián pro každou ze tří barevných složek a modifikuje jím požadovanou barvu detekce pro další snímek v pořadí při zachování zbylých dvou parametrů posledně známého klíčového snímku (tolerance a poloměr oříznutí). Vektor klíčových snímků je ve všech směrech nadřazený ACC vektoru a detekce markerů na klíčových snímcích je prováděna přesně ze zadaných hodnot – ACC vektor se tedy uplatňuje pouze mezi klíčovými snímky, nicméně možnost jeho reprodukce na základě znalosti pouze vektoru klíčových snímků je stále zachována. Vektor klíčových snímků je nepovinným výstupem detekce úhlu mezi markery v každém videu.

Po skončení detekce je v jiné části téhož GUI proveden zápis úhlů v čase do textových souborů. Tyto soubory, společně se soubory se záznamem pohybu plošiny a soubory obsahujícími časová razítka, vstupují do dalšího GUI, ve kterém je možné zhlédnout náhled k výstupnímu grafu a upravit interpolované úseky. Jedná se o místa, kde záznam úhlu z různých důvodů chyběl, nicméně šířka mezery mezi daty nedosahuje nějak rozumně neinterpolovatelných mezí. Jednotlivé interpolace uživatel buďto upravuje, nebo vypouští.

Po odsouhlasení modifikací interpolací je záznam segmentován na jednotlivé periody a z každé periody jsou extrahovány hodnoty úhlu plošiny a zvířete, případně hodnoty času, z přesně definovaných oblastí. Sledované parametry tvoří procentuální podíl úhlu zvířete ku úhlu plošiny v momentech zastavení a rozjetí plošiny, dále procentuální rozsah pohybu (*ROM*), což je podíl rozsahu úhlu zvířete a rozsahu úhlu plošiny ve vzestupné, či sestupné fázi pohybu plošiny, a nakonec délka trajektorie v diagramu vzájemné závislosti úhlu plošiny a úhlu zvířete pro každou z period. Sledování dalších parametrů, jako je doba stání plošiny mezi jednotlivými rampami nebo maximum či minimum z úhlu zvířete v době stání plošiny, nemají pro své krátké trvání ve vyhodnocení gekončků smysl (setiny sekundy). Při vyhodnocování žab však smyslu nabývají, protože tam byl každý pohyb plošiny iniciován ručně s různě dlouhou prodlevou.

Všechny výsledky, ať už pro gekončíky či žáby, jsou exportovány softwarem do excel tabulek a následně jsou podrobeny statistickému rozboru.

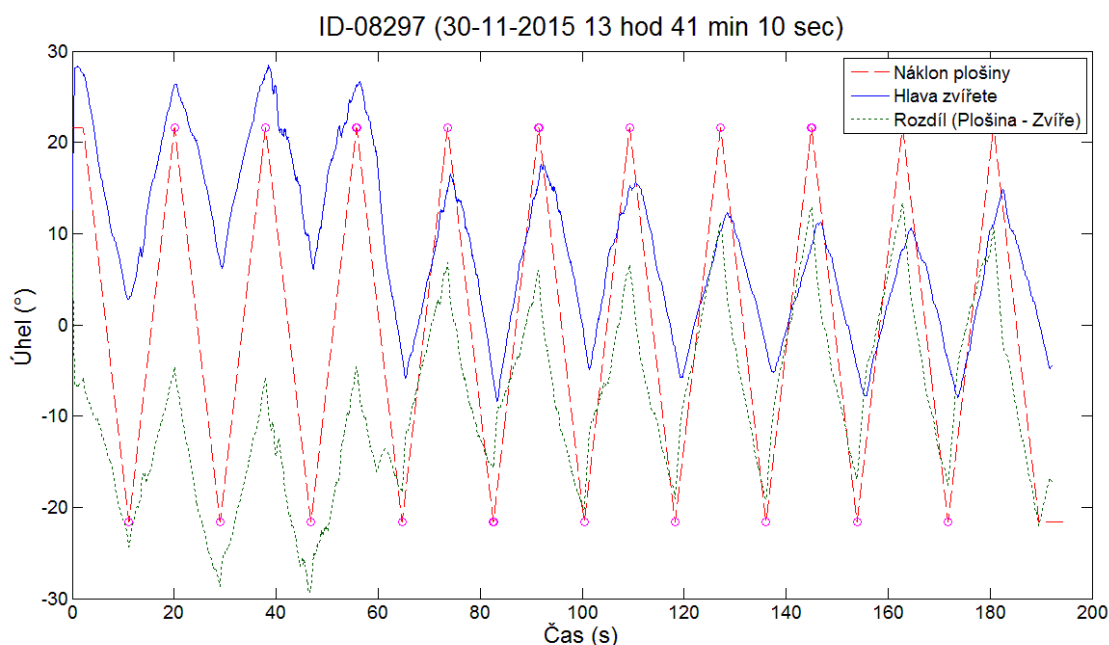
7 Hodnocení výsledků experimentální části

S ohledem na vysoký počet výstupů z měření ve vlastní práci nebudu uvádět všechny výsledné grafy a tabulky. Na příloženém CD jsou uvedeny veškeré výsledky z provedených měření na gekončících nočních - zde tedy uvedu pouze některé reprezentativní výsledky pro vysvětlení základních postupů hodnocení.

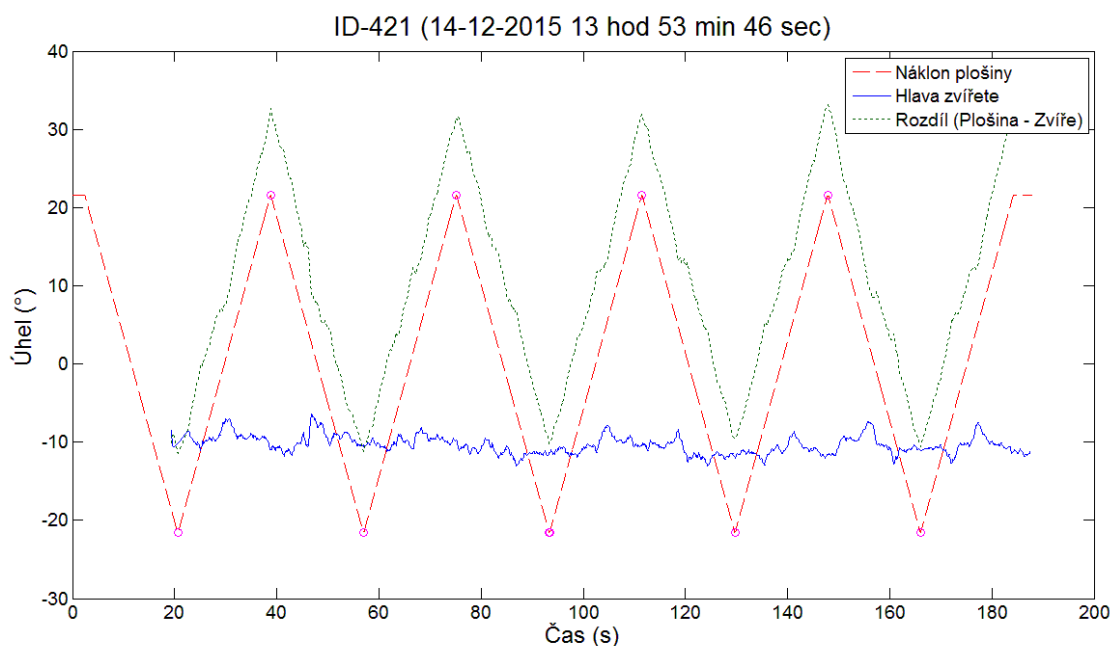
7.1 Ukázka výsledných diagramů a výstupů z GUI

Pro ukázkou jak vypadá poměrně vyvedený graf závislosti úhlu hlavy zvířete v čase a graf pohybu plošiny v čase jsem vybral gekončika s identifikačním číslem 08297. Jedná se o měření ze dne 30. 11. 2015, 13:41:10, před stimulem, rychlost posuvu plošiny $5^\circ/\text{s}$. V grafu 7.1 je zřetelně patrné, jak se může měnit stejnosměrná složka pohybu zvířete i při zachování míry kompenzace pohybu plošiny. Z tohoto důvodu je vhodnější jako ukazatel využít Spearmanův korelační koeficient. Tečkovaná zelená čára dole (graf 7.2), značící rozdíl obou úhlů, by měla odpovídat 100% kompenzaci v době, kdy se pohybuje v blízkosti hodnoty 0° . Z grafu 7.1 však vidíme, že díky počátečnímu naklonění hlavy živočicha v prostoru (posun stejnosměrné složky) dosahuje úhel hlavy ještě vyšších hodnot, nežli úhel plošiny. Pokud bychom se tedy omezili na hodnocení míry kompenzace pouze z odečtu (tečkovaná čára), dopustili bychom se chyby. Z tohoto důvodu není možné hodnotit míru kompenzace podle jednoduchých kritérií, jakými je rozdíl úhlů a odchylka od nulového úhlu.

Graf 7.2 vedle na rozdíl od předchozího grafu ilustruje situaci, kdy gekončík pohyb nikterak nekompensuje. Existují záznamy, kdy většina úhlu hlavy zcela chybí – v takových případech jsme ještě za výsledky ilustrované grafem 7.2 rádi.

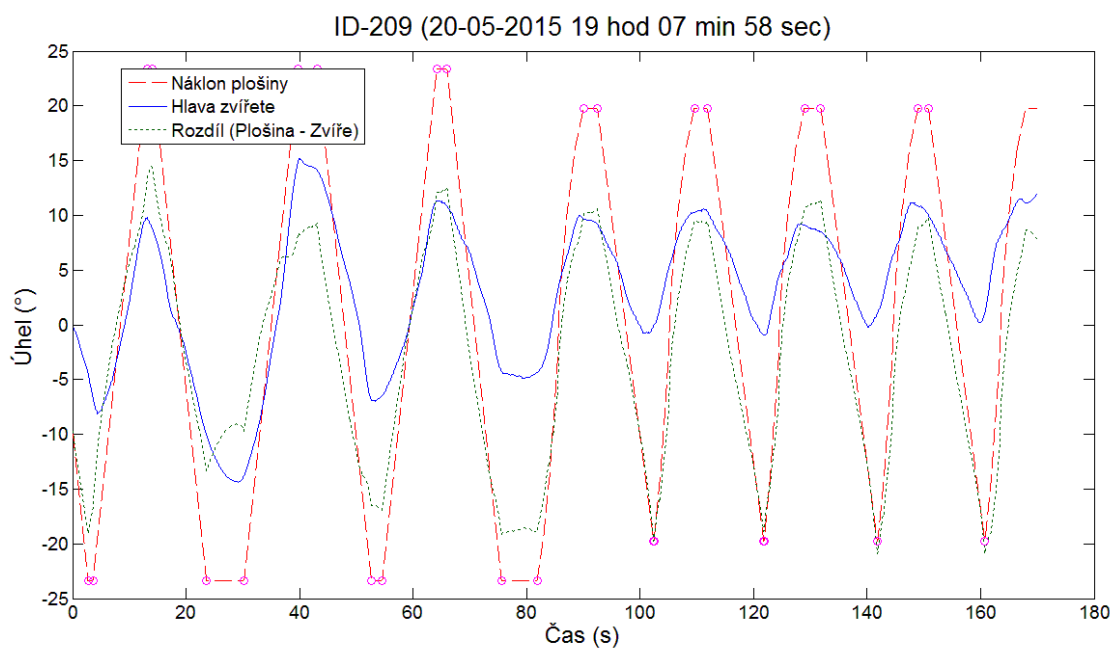


Graf 7.1: Ukázka měření gekončika s identifikačním číslem 08297, měření ze dne 30. 11. 2015, čas 13:41:10, před stimulem, úhlová rychlost rotace plošiny $5,0^\circ/\text{s}$.

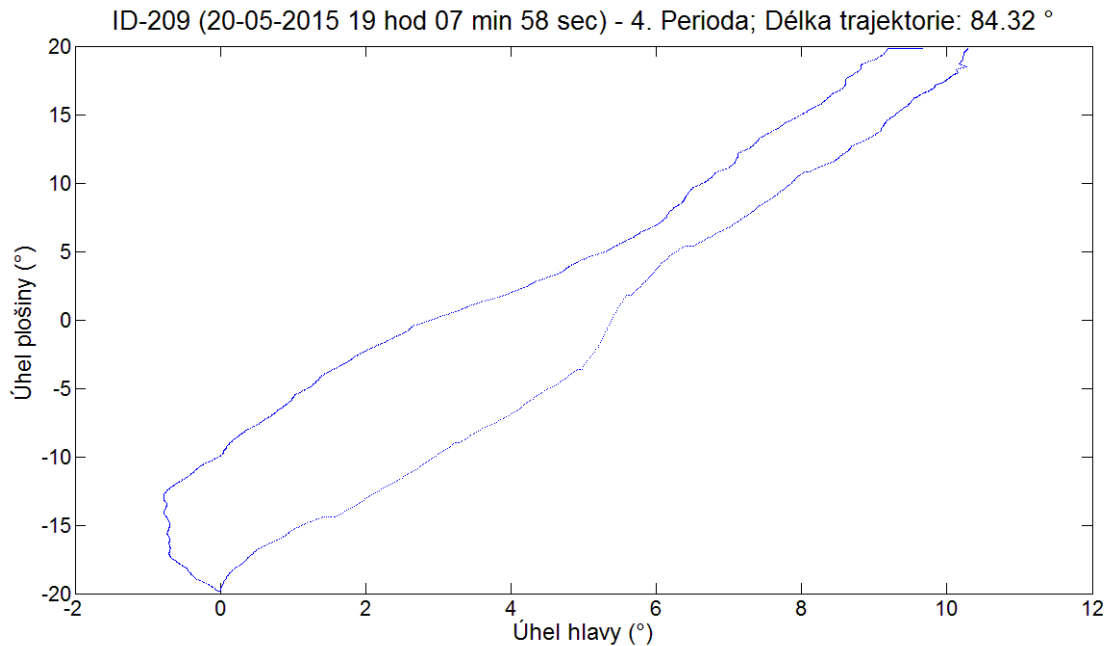


Graf 7.2: Ukázka měření gekončíka s identifikačním číslem 421, měření ze dne 14. 12. 2015, čas 13:53:46, po stimulu, úhlová rychlost rotace plošiny $2,5^{\circ}/s$.

Abychom byli ve svých popisech výstupů z měření kompletní, je třeba uvést též výsledky z měření exotických žab – k tomu poslouží výsledný graf 7.3 níže. Na něm je vidět ze začátku část záznamu z rampy s rychlostí $5,0^{\circ}/s$ a poté 5 sinusových úseků, kdy vzestupná fáze v grafu probíhala po sinusově se měnící rychlosti od $2,5^{\circ}/s$ do $5,0^{\circ}/s$, a sestupná fáze probíhala po rampě $5,0^{\circ}/s$.



Graf 7.3: Ukázka měření *Dyscophus Guineti* s identifikačním číslem 209, měření ze dne 20. 5. 2015, čas 19:07:58, úhlová rychlost rotace plošiny proměnlivá.



Graf 7.4: Ukázka grafu vzájemné závislosti úhlu hlavy zvířete na úhlu pohybu plošiny pro 4. periodu ze záznamu výše (graf 7.3).

Ve všech grafech je vidět kruhové značení barvy magenta, vymezující stacionární oblasti plošiny, tedy úseky, kdy plošina stojí na místě. Na základě této segmentace je záznam rozdělen na jednotlivé periody, což nám umožňuje vyhodnotit pro každou periodu důležité ukazatele, jakým je například délka trajektorie v diagramu vzájemné závislosti úhlu plošiny a úhlu zvířete, jak ukazuje graf 6.4 pro již zmíněný exemplář žáby výše.

Pro úplnost ještě uvádím zmíněný graf závislosti pohybu plošiny na pohybu hlavy živočicha (graf 6.4), ze kterého je vidět přizpůsobení pohybu hlavy živočicha vzhledem k pohybu plošiny. Sklon křivky naznačuje, že roste-li úhel naklonění plošiny, roste též úhel hlavy živočicha, což značí kompenzaci. Z tvaru křivky a výsledné délky trajektorie je rovněž možné posuzovat na hysterezi, tj. zdali živočich reaguje na pohyb plošiny se zpožděním, či nikoli.

7.2 Ukázka exportovaných tabulek

Jak bylo zmíněno dříve, Exportní část GUI je zodpovědná za segmentaci a extrakci veškerých dat z průběhu a jejich zápis do *.xls (*.xlsx) souboru. Příkladem tabulky s výsledky pro výše zmíněnou žábu je tabulka 7.1 níže.

Tabulka 7.1: Ukázka exportovaných ukazatelů pro ID-209 (20-05-2015 19 hod 07 min 58 sec); Značení sledovaných veličin odpovídá značení na obrázku 6.1, Délka trajektorie Z-P značí délku trajektorie v diagramu vzájemné závislosti úhlu Zvířete a úhlu Plošiny a Délka trajektorie Z-D značí délku trajektorie v diagramu vzájemné závislosti úhlu Zvířete a Diference úhlu Plošiny a Zvířete. Indexy u popisku sloupečků označují jednotlivé periody záznamu.

	P ₁	P ₂	P ₃	P ₄	P ₅	P ₆	P ₇	P ₈
δ_{1A} (%)		41,30	64,00	48,38	48,83	51,91	46,03	55,00
δ_{2A} (%)		37,93	60,32	46,20	46,40	52,22	43,06	50,51
ROM _A (%)		40,39	44,81	32,13	23,29	28,42	18,93	22,53
δ_{1B} (%)	18,74	42,85	29,30	18,06	0,19	4,63	-5,19	-5,45
δ_{2B} (%)	28,16	58,57	27,59	18,77	0,05	4,65	-5,64	-5,78
ROM _B (%)	34,73	61,28	37,99	32,55	25,98	25,34	24,68	
Max _A (%)		41,30	64,99	48,38	48,83	53,49	46,03	55,01
Max _B (%)	28,16	61,41	29,75	21,11	0,19	4,65	-5,19	-5,45
t _A (s)		0,76	3,28	1,80	2,44	2,30	2,87	1,82
t _B (s)	0,85	6,63	1,89	6,27	0,03	0,03	0,07	0,03
Délka trajektorie Z-P (°)		112,60	104,94	98,98	84,32	84,67	84,10	84,63
Délka trajektorie Z-D (°)		80,88	75,08	75,90	68,08	66,98	69,68	69,15

Je na první pohled vidět, že některá data chybí. To je způsobeno chybějícím údajem u úhlu zvířete v definovaných bodech periody. Tento neduh by byl částečně odstraněn již zmíněnou metodou korelace či regresní křivky při použití metody nejmenších čtverců, dovolující využít zbytku bodů v okolí.

Výsledky měření všech zvířat je vhodné pro další zpracování umístit do jednotné tabulky, obsahující všechny ukazatele pro všechna zvířata ze série měření se společným znakem (čas měření, rychlost pohybu, stimul apod.) ve všech periodách záznamu. Protože se vždy jedná o dlouhou tabulku, uvedu zde pouze ukázkou výpisu pro 4. periodu ze série měření gekončků uskutečněnou dne 14. 12. 2015, jedná se o rychlost 2,5°/s a měření po stimulu. Tato ukázkou mi přišla nejvhodnější z důvodu nejhustšího obsahu dat pro relativně nízký počet použitelných subjektů z dané série měření (jinými slovy - vejde se na stránku).

Tabulka 7.2: Ukázka sumarizace výsledků pro měření na gekončících ze dne 14. 12. 2015, úhlová rychlost rotace plošiny 2,5 °/s, po stimulu – vidět je pouze 4. perioda. Popis veškerých užitých zkratk je shodný s předešlou tabulkou.

ID	166	421	989	990	08177	08280	08043	08776	
...	...								
P ₄	ROM _A (%)	67,79	3,85	63,45	32,62	3,69	32,03	40,31	
	ROM _B (%)		1,04	62,34	34,29	175,59	26,00	11,19	
	Max _A (%)	25,78	-47,62	56,53	0,70	53,26	70,53	5,93	
	Max _B (%)	110,18	55,62	73,15	64,76	-45,82	-6,29	74,74	
	Délka trajektorie Z-P (°)	140,39	105,76	146,53	173,04	179,28	109,94	38,41	131,77
	Délka trajektorie Z-D (°)	151,11	114,41	161,64	209,89	215,03	107,54	40,32	138,10
...	...								

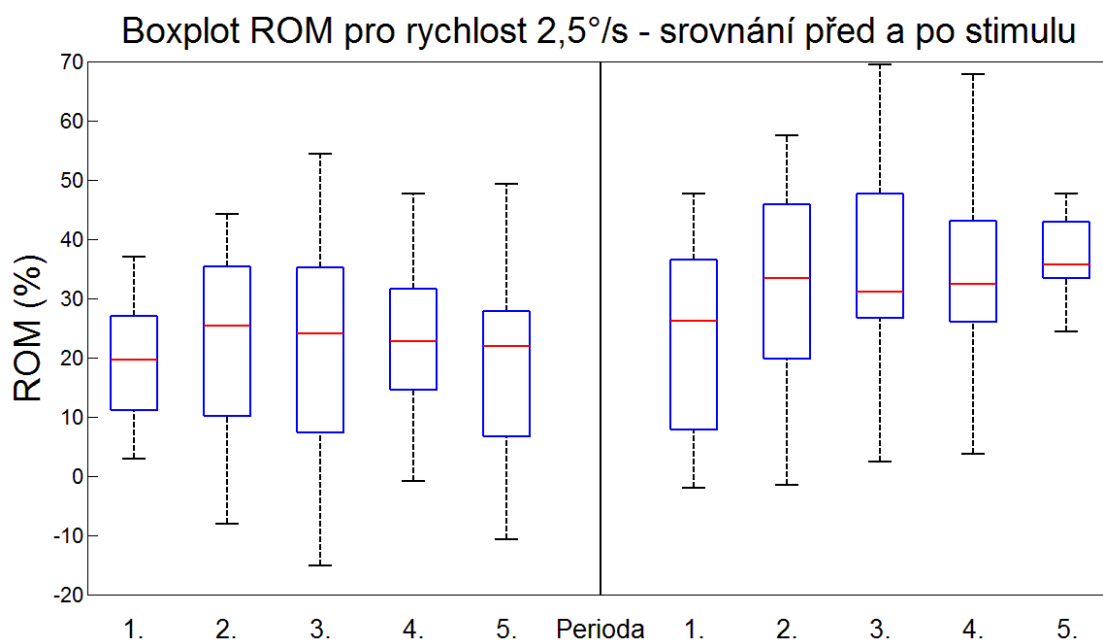
7.3 Statistické hodnocení dat

Během předchozích experimentů se ukázalo, že zřejmě nejvhodnější vizualizaci výsledků poskytne funkce boxplot, vykreslující pro všechny subjekty výběrového souboru pro každou periodu a každý parametr mezikvartilové rozpětí a mediány. Ze zaznamenaných hodnot nemá zřejmě příliš smysl hodnotit veličiny Max_A a Max_B , protože ty se mění v závislosti na posunu stejnosměrné složky úhlu hlavy zvířete při měření. Tento problém však nemusíme řešit u hodnocení ROM , který je pouze poměrem rozsahů pohybu zvířete a plošiny. Ukázku z boxplotu ROM pro rychlost $2,5^\circ/s$ a $5,0^\circ/s$ prezentují grafy 7.5 a 7.6 níže a vedle.

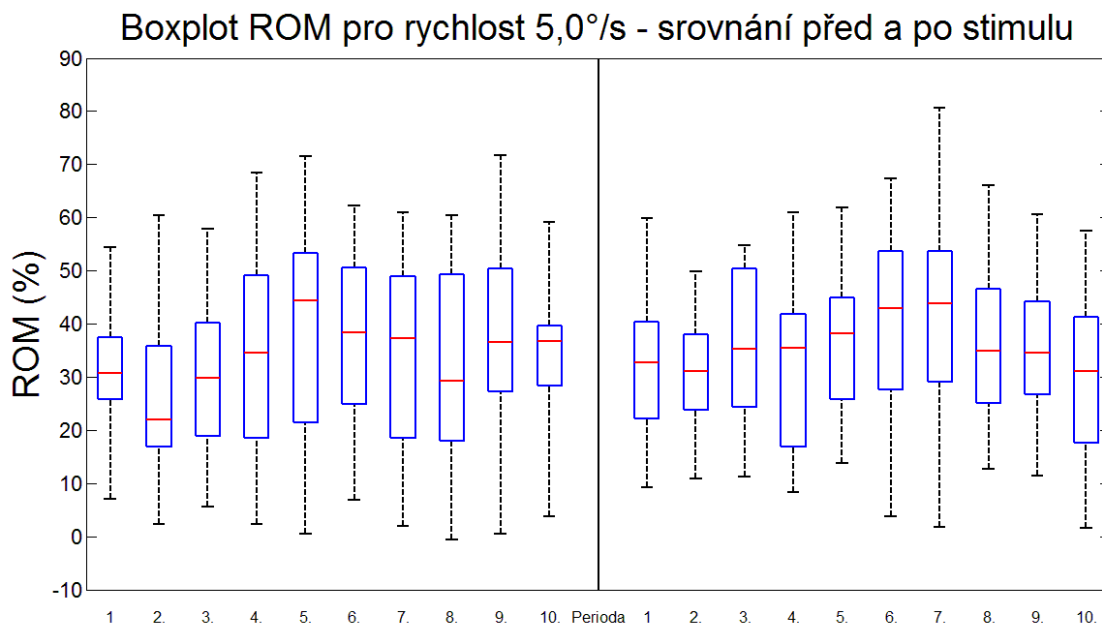
Uvedené grafy shrnují dohromady jak ROM_A tak ROM_B . Mohli bychom se ptát, zdali neexistují rozdíly mezi ROM na jednu a na druhou stranu pohybu plošiny, z mého pohledu však ne. Důkazem jsou boxploty pro ROM_A a ROM_B pro obě dvě rychlosti přiložené na CD.

Co je zde patrné je mírné navýšení mediánů po stimulaci subjektu při rychlost $2,5^\circ/s$. Zvíře nejspíš zpozorní a přidá na kompenzaci. Je však zvláštní, že z výsledků pro vyšší rychlost toto explicitně neplyne a uvedený jev bychom v grafu hledali jen stěží. Je možné, že zvíře mělo zkušenost z měření při nižší rychlosti, realizovaného předtím, a vědělo, že mu po stimulaci skutečně nebezpečí nehrozí, tudíž kompenzovalo méně než v předešlém případě.

Při pohledu na graf 7.6 a oba další grafy, týkající se veličiny ROM přiložené na CD, je však patrný vlnkovitý charakter průběhu mediánů v závislosti na periodě. Ze začátku se zdá, že si zvíře přivyká na pohyb plošiny a nekompenzuje, v průběhu měření kompenzovat začíná a po 6. periodě zase přestává, možná z důvodu přivyknutí periodickému pohybu plošiny a dostavení únavy (přibližně 90 sekund od začátku měření, to už jsme unavení i my). Nezřídka se nám stalo, že gekončící v procesu měření zavřeli oči a počali usínat – únava je tedy definitivně faktorem, ovlivňující výsledky.



Graf 7.5: Ukázka boxplotu sledované veličiny ROM pro úhlovou rychlost rotace plošiny $2,5^\circ/s$ – před a po stimulaci subjektu.



Graf 7.6: Ukázka boxplotu sledované veličiny ROM pro úhlovou rychlost rotace plošiny 5,0°/s – před a po stimulaci subjektu.

Boxploty veličin Max_A a Max_B , neuvedené v tomto textu, jsou na přiloženém CD.

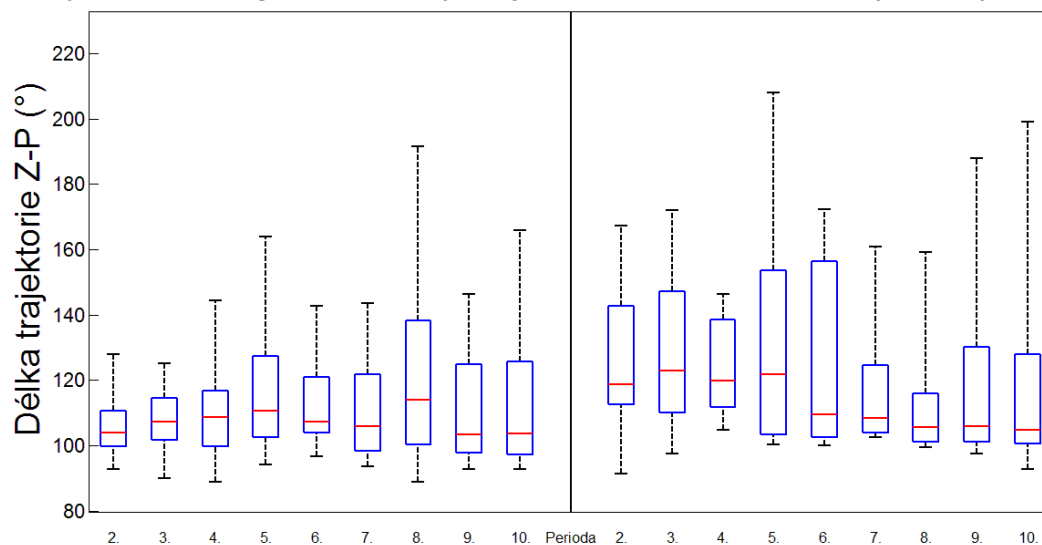
Zajímavou kapitolou jsou délky trajektorie v diagramech pohyb plošiny proti pohybu zvířete. Tady totiž nastal problém s nestejně dlouhými daty pro všechny periody. Ne vždy se nám totiž povedlo naměřit kontinuální záznam pro celou periodu – často jsme odměřili jen velmi krátký úsek a v takových případech je zjevné, že se spočtená délka trajektorie bude diametrálně lišit od ostatních, nicméně tento fakt nepůjde z výsledných dat v tabulce nijak vyčíst.

Navrhl jsem řešení, jak by se tato nekompletní data dala alespoň částečně zužitkovat. Jedním z exportovaných souborů je textový soubor se všemi průběhy v čase, rozdělený do sloupců po periodách. Z něj je možné odečíst délku každé z period a zároveň použitelnou délku signálu. Jejich poměrem získám číslo, kterým lze získanou délku trajektorie z dané periody roznásobit a dostat tak přibližně správnou hodnotu, jakou bychom mohli očekávat při vlastnictví záznamu z celé periody. Jedná se však o metodu, jež zcela určitě nejde použít v případě dlouhého úseku chybějících dat – proto jsem zavedl vlastní omezení na horní hranici poměru, která nesmí přesahovat číslo 5, což by odpovídalo tomu, že máme záznam jen z jedné pětiny periody, nebo méně.

Podotýkám, že většina spočtených trajektorií měla tento poměr roven číslu 1, jak ukazují tabulky na přiloženém CD (listy s předznamenáním „korekce“), což odpovídá kompletnímu záznamu z periody. Ostatní případy, spadající do rozsahu v rozmezí 1-5, byly korigovány příslušným poměrem a po označení byly zaneseny do výchozích tabulek k vynášení grafů boxplotu. Bez této korekce bychom náš již tak malý výběrový soubor museli ještě více zredukovat a boxploty by se staly neprůkaznými.

Pro prezentaci použiji boxplot trajektorie v diagramu zvíře-plošina pro rychlost 5,0°/s, jelikož diagram zvíře-diference vypadá prakticky úplně stejně a oproti rychlosti 2,5°/s je zde více period ke sledování vývoje. Ukázka viz graf 7.7:

Boxplot Délka trajektorie Z-P pro rychlost 5,0°/s - srovnání před a po stimulu



Graf 7.7: Ukázka boxplotu sledované veličiny *Délka trajektorie* v diagramu úhel zvířete-úhel plošiny pro úhlovou rychlost rotace plošiny 5,0°/s – před a po stimulaci subjektu.

Zmíněný “sinusový” fenomén je lehce viditelný i zde. Navzdory původnímu očekávání se však nezdá, že by se délka trajektorie významně měnila s narůstajícím časem.

Pro hodnocení jsem nejprve zvolil subjektivní přístup, protože mně osobně přijde expertní analýza výsledků stejně důležitá, ne-li dokonce důležitější, nežli hodnocení pomocí statistických metod, které popíšu vzápětí.

Vyjdeme z již nastíněných boxplot grafů. Každý sloupeček lze nazývat sadou dat a pro každou sadu lze určit, zdali má normální či nenormální rozdělení. Za tímto účelem jsem využil Shapiro-Wilkův test. Obdobně lze využít Jarque-Bera test – ten však není dle současné literatury pro datovou sadu našeho typu tak vhodný, jako jeho nástupce (THADEWALD, 2007). Na základě normality dat pak lze hledat signifikantní rozdíly mezi středními hodnotami (normální rozdělení), nebo mediány (nenormální rozdělení) jednotlivých sad.

Ukázalo se, že naše sady obsahují pouze normálně rozdělená data. V tomto případě lze srovnávat střední hodnoty jednotlivých sad a sledovat, zdali na zvolené hladině významnosti (5 %) nedochází k výrazným odchylkám. Tomuto účelu poslouží párový *t-test*. Protože ale umí pracovat vždy jen se dvěma sadami, a protože my v rámci jednoho boxplotu disponujeme často i 20 sadami, je nutné *t-test* provést křížově všemi možnými kombinacemi. Ze statistického hlediska se jedná o operace kombinace bez opakování, popsané vzorečkem 8,

$$C(n, k) = \binom{n}{k} = \frac{n!}{(n-k)!k!} \quad (8)$$

kde n značí počet členů (v našem případě počet datových sad – sloupců v boxplot grafu) a k značí hodnotu skupiny (v našem případě tvoříme dvojice ke srovnávání). Počet výsledků po otestování *t-testem* pro 20 sad činí po dosazení do vzorce 8 190, což není málo. Vzniká otázka, jak takové výsledky vizualizovat. Jednou z možností, je vytvoření symetrické matice, na jejíž

obou osách budou vyneseny jednotlivé sady. Na hlavní diagonále se u takové matice vždy vyskytují hodnoty testování mezi dvěma identickými sadami. Výsledkem testování je závěr pro nulovou hypotézu, která v tomto případě tvrdí, že se střední hodnoty testovaných sad významně neliší. Hypotézu lze na zvolené hladině významnosti zamítnout v případě, že výsledek testování vyjde 1 (případně číslo p vyjde o dost nižší, než je hladina významnosti 0,05).

Ukázku výsledné matice pro boxplot parametru *ROM*, rychlost 2,5 °/s, společně s výsledky testu normality pomocí Shapiro-Wilkova a Jarque-Bera testu můžete vidět v tabulce 7.3 níže.

Obdobné formátování co se týče barev je použito i ve zdrojovém excel souboru na přiloženém CD. V symetrické matici nás bude zajímat především první kvadrant, reprezentující výsledky srovnání skupin před a po stimulu. Zde vidíme, že dochází často k rozpoznání významné odchylky, což značí rozdíly v chování zvířete před a po stimulaci. Ve druhém kvadrantu bychom hledali rozdíly mezi periodami pro zvířata jen před stimulací a obdobně ve čtvrtém kvadrantu jen po stimulaci. Občas nacházíme v datech i zde odchylky – ve velké míře se jedná o rozdíly mezi některou z prvních a posledních period, což dává smysl. Občas se ale ukáže rozdíl i mezi dvěma sousedícími periodami, což může být faktor náhody, vylučitelný jen zvětšením výběrového souboru. Třetí kvadrant je celý vybarven šedě, stejně jako celá část pod hlavní diagonálou, a obsahuje pouze duplicitní položky.

Tabulka 7.3: Výsledky p-čísla Shapiro-Wilkova (S-W) a Jarque-Bera (J-B) testu normality dat pro sjednocený parametr *ROM* při rychlosti 2,5 °/s před a po stimulu. Hodnoty větší než 0,05 ve výsledku normality znamenají, že se jedná o normální rozdělení dat, menší že se jedná o nenormální rozdělení. Nižší část tabulky ukazuje symetrickou matici *t-testů*, kde zkratka *pre* označuje měření před stimulací a *post* po stimulaci. Číslo na konci označuje periodu. Číslo větší než 0,05 zde neznačí žádný významný rozdíl mezi daty a menší značí významný rozdíl.

Test normality	Pre - 1	Pre - 2	Pre - 3	Pre - 4	Pre - 5	Post - 1	Post - 2	Post - 3	Post - 4	Post - 5
S-W test	0,7212	0,0653	0,7921	0,2543	0,8757	0,7693	0,6413	0,5938	0,3270	0,8461
J-B test	0,5000	0,1406	0,5000	0,4237	0,5000	0,5000	0,5000	0,5000	0,5000	0,5000

Párový t-test	Pre - 1	Pre - 2	Pre - 3	Pre - 4	Pre - 5	Post - 1	Post - 2	Post - 3	Post - 4	Post - 5
Pre - 1		0,5780	0,8522	0,2133	0,7468	0,7944	0,0638	0,0303	0,0270	0,0627
Pre - 2	0,5780		0,6562	0,8273	0,8217	0,9328	0,0956	0,0219	0,0300	0,1721
Pre - 3	0,8522	0,6562		0,6652	0,8900	0,5963	0,0336	0,0068	0,0287	0,0421
Pre - 4	0,2133	0,8273	0,6652		0,2301	0,9096	0,0903	0,0304	0,2301	0,0666
Pre - 5	0,7468	0,8217	0,8900	0,2301		0,8173	0,0234	0,0003	0,0492	0,0054
Post - 1	0,7944	0,9328	0,5963	0,9096	0,8173		0,5418	0,4535	0,1906	0,2162
Post - 2	0,0638	0,0956	0,0336	0,0903	0,0234	0,5418		0,6143	0,5700	0,3279
Post - 3	0,0303	0,0219	0,0068	0,0304	0,0003	0,4535	0,6143		0,6717	0,9393
Post - 4	0,0270	0,0300	0,0287	0,2301	0,0492	0,1906	0,5700	0,6717		0,5848
Post - 5	0,0627	0,1721	0,0421	0,0666	0,0054	0,2162	0,3279	0,9393	0,5848	

K provedení všech testů jsem využil sepsané skripty, umístěné externě mimo GUI v příslušném adresáři na CD. Jarque-Bera test nebo t-test jsou již součástí Matlabu®, modifikovaný anova test pro přehledné zobrazování výsledků testování je výsledkem vlastního inovačního přístupu a funkci pro realizaci Shapiro-Wilkova testu jsem převzal z internetu z *fileexchange* přes *mathworks*. Veškeré výsledky z testů stejně jako tabulky k boxplotům a skripty pro jejich vykreslení lze nalézt v téže složce na přiloženém CD.

8 Závěr

8.1 Naplnění cílů práce

„Cílem práce je vytvoření pohyblivé naklápěcí plošiny určené k záznamu pohybu živočichů“

V rámci řešení jsem při návrhu a výrobě naklápěcí plošiny kooperoval s panem Bc. Jánem Hýblem ze SPŠ Kladno. Vybral jsem vhodné kamery a asistoval s nastavením krokových motorů. Aktivně jsem se účastnil všech zkušebních a odlaďovacích měření a pomohl přijít s řešeními vyskytnuvších se SW problémů. Plošina je momentálně k dispozici na FBMI ČVUT, je ve finální fázi vývoje, odlaďují se již jen maličkosti. Do budoucna se plánuje její převoz do Prahy na Přírodovědnou Fakultu Univerzity Karlovy, kde s ní budou dále pracovat zaměstnanci a studenti PŘF UK.

„Dále je cílem práce vytvoření metod kvantitativního hodnocení kinematických dat pro studium funkce nervové soustavy.“

Při výběru sledovaných veličin pro hodnocení jsem vyšel z mé předešlé bakalářské práce, kde jsem aktivně využil hodnocení formou délky trajektorie v diagramech vzájemných závislostí úhlů. Navrhl jsem další veličiny, charakteristické pro navržené scénáře pohybu plošiny, které je možno rovněž studovat, a ze kterých je možné usuzovat na kompenzační pohyby živočicha, jakými je například rozsah pohybu *ROM* normalizovaný na rozsah pohybu plošiny. Ozkoušel jsem scénáře periodického pohybu plošiny při různých rychlostech, přejaté z cizích článků.

„Pro pohybovou naklápěcí plošinu vytvořte algoritmy řízení přesného pohybu a záznamu kinematických dat pohybu segmentů těla živočicha.“

Ve spolupráci s Bc. Jánem Hýblem jsme vytvořili komunikační protokol pro rozhraní MatLab a C++ (řízení PAC mikrokontroléru plošiny). V MatLabu jsem vytvořil grafické uživatelské rozhraní, sloužící k ovládání pohybu motorů, dále jsem vytvořil rozhraní, sloužící k nahrávání, vytváření a ukládání celých scénářů pohybu jednotlivých motorů, které úzce komunikuje s FLASH pamětí mikrokontroléru a umožňuje nastavenou sekvenci pohybů provést po startu automaticky, bez nutnosti uživatelského zásahu.

Vytvořil jsem v MatLabu rozhraní, pro načítání připojených kamer, zobrazování náhledů a nahrávání videí. V rámci navrženého softwaru jsem se nejvíce věnoval zpracování naměřených videí a mým největším přínosem byla časová synchronizace nahraných videí s reálným časem při měření, dále synchronizace dat ze záznamu pohybu plošiny a z videí a nakonec algoritmy detekce mezimarkerového úhlu a její zjednodušení v rámci dalšího navrženého rozhraní, sloužícího k detekci.

„Při výběru metod vycházejte z metod používaných k měření a hodnocení pohybu lidského těla.“

Na základě rešerše jsem navrhl vhodné metody a algoritmy výpočtu parametrů pohybu. Užil jsem segmentace periodického pohybu živočicha či plošiny, kdy jsem pro každou periodu extrahoval ze záznamu rozsahy pohybu, maxima v oblastech peaků a diagram vzájemných úhlových závislostí. Navrženým algoritmem jsem vypočetl délky trajektorií v diagramu a do budoucna navrhl užití korelačních metod a porovnávání úhlové odchylky mezi směrnici spádu

plošiny a regresní přímkou, proloženou daty s pohybem živočicha, na základě metody nejmenších čtverců. K hodnocení jsem užil „krabicových grafů“ boxplot, užívaných běžně ve statistice pro hodnocení statistických veličin výběrového souboru. Všechny zmíněné parametry se běžně analyzují v praxi například při vyšetřování chůzového cyklu.

„Algoritmy přesného řízení plošiny a záznamu pohybu, a jeho hodnocení, implementujte v prostředí MatLab.“

Jak již bylo zmíněno, na základě rešerše jsem navrhl vhodné algoritmy řízení plošiny ve spolupráci s Bc. Jánem Hýblem, a implementoval do řídicího HW plošiny.

„V rámci práce dále realizujte měření rozličných druhů živočichů ve spolupráci s PŘF UK a FN Motol.“

Měření živočichů proběhlo ve spolupráci s PŘF UK (doc. RNDr. Daniel Frynta Ph.D.). Měření se uskutečnila na gekončících nočních a 6 druhích žab. Ve spolupráci s Dr. Černým jsme měřili s pomocí oftalmoskopického ramene zrakový rozsah ropuchy obecné.

„Využitím naměřených dat proveďte vyhodnocení zjištěných ukazatelů určených navrženými metodami, a proveďte statistická zhodnocení výsledků experimentálních měření kinematických dat pohybu živočichů.“

Vyhodnocení statistických veličin sledovaných ukazatelů jsem prováděl v prostředí MatLab formou boxplotu. Výstupem z mé práce jsou excel tabulky, přiložené na CD, některé skripty pro manipulaci s daty a obrázky grafů. Ačkoliv se předpokládá hodnocení dat především spolupracovníky z PŘF UK, v mé práci jsem také prezentoval hodnocení naměřených dat a dílčí zjištění.

8.2 Ověření Hypotéz

Ze statistického i expertního hodnocení plyne, že existují odlišnosti v chování zvířete před a po stimulaci, jak ukazuje například prezentovaná tabulka 7.3. Nebyla však jednoznačně potvrzena hypotéza o rozdílech v čase – z expertního hlediska nepozorují žádné změny v charakteru rozložení dat a statisticky vyšly rozdíly mezi jednou z prvních a jednou z posledních period jen velice zřídka a stejně často, jako rozdíly mezi dvěma sousedními periodami, což můžu po konfrontaci s vizuálním hodnocením s klidným svědomím oboje prohlásit za falešně pozitivní výsledky.

Nemyslím si však, že by hypotéza neplatila – časové rozdíly určitě existují, jak ukazuje v kapitole 7.3 – *Statistické hodnocení dat* zmíněný sinusový fenomén. Jejich odhalení pouze vyžaduje aplikaci statistických metod na větší výběrový soubor.

8.3 Perspektivy dalšího využití navržených metod

Rozvoj a další využití mých návrhů může směřovat do oblasti modifikací HW či SW. Pokud jde o SW a metody zpracování dat, nabízí se několik možností. Vzhledem k nutnosti dispozice naměřenými daty ve vybraných oblastech periody záznamu pohybu plošiny by bylo vhodné za účelem zisku některých kvantifikovatelných dat metody rozšířit tak, že nebudou závislé na bodech průběhu záznamu, nýbrž na celé části průběhu záznamu v rámci periody pohybu plošiny. Prvním krokem k využití celého úseku k hodnocení kompenzace namísto jen několika málo bodů, bylo hodnocení formou délky trajektorie v diagramu vzájemných úhlových

závislosti. Možnosti zde však nekončí, bylo by vhodné se nyní zaměřit na hodnocení pomocí Spearmanova korelačního koeficientu, případně úhlové odchylky směrnice spádu plošiny a regresní přímky, proložené daty na základě metody nejmenších čtverců.

Výše zmíněné metody nepracují s jedním bodem v záznamu, ale se skupinou bodů, a umožnily by nám lépe charakterizovat míru kompenzace živočicha a eliminovat nyní chybějící políčka ve výsledných tabulkách z důvodu absence záznamu v tom daném bodě pohybu plošiny.

V práci jsme experimentálně testovali rampové scénáře a sinusový pohyb plošiny – vše v sagitální rovině živočicha. Je zde prostor k rozšíření scénářů na dosud nezkoumané varianty a výzkum můžeme rovněž rozšířit o zbylé dvě studované roviny. Náš experiment z důvodu komparability výsledků stavěl na základě již napsaných článků jiných odborníků v tomto oboru – to nás však neomezuje v tom, vymyslet si vlastní metodiku měření a zavést ji poprvé do světové praxe.

Navržené algoritmy k detekci markerů na základě barevné informace v obraze může sloužit rozličným účelům. Kdykoliv se naskytne potřeba v barevném videu detekovat a sledovat pohyb skupiny pixelů o podobné barvě, lze k tomu použít kostru navržených algoritmů. Algoritmus je momentálně nastaven na sledování dvou markerů a počítání úhlu mezi nimi – při osahání si jeho základů však není pro pokročilého uživatele MatLabu® problém skript za pár týdnů překonfigurovat na detekci jedné či více než dvou skupin.

Pokud jde o využití a modifikace HW, nabízí se možnosti volby rychlejších kamer systému, dále použití "silnějších" aktuátorů pro měření rozměrnějších živočichů atp. Ale pokud jde o naplnění požadavků PŘF UK - ty jsou splněny. Do budoucna se předpokládá měření řady druhů živočichů, a tomu odpovídající navazující výzkum.

Seznam použité literatury

BELOOZEROVA, I. N., P. V. ZELENIN, L. B. POPOVA, G. N. ORLOVSKY, S. GRILLNER a T. G. DELIAGINA, 2003. Postural control in the rabbit maintaining balance on the tilting platform. *Journal of Neurophysiology* [online]. 12., roč. 90, č. 6, s. 3783–3793. ISSN 0022-3077. Dostupné z: doi:10.1152/jn.00590.2003

BOLBECKER, Amanda R., S. Lee HONG, Jerilyn S. KENT, Mallory J. KLAUNIG, Brian F. O'DONNELL a William P. HETRICK, 2011. Postural control in bipolar disorder: increased sway area and decreased dynamical complexity. *PLoS One* [online]. roč. 6, č. 5, s. e19824. ISSN 1932-6203. Dostupné z: doi:10.1371/journal.pone.0019824

DELIAGINA, Tatiana G., Grigori N. ORLOVSKY, Pavel V. ZELENIN a Irina N. BELOOZEROVA, 2006. Neural bases of postural control. *Physiology (Bethesda, Md.)* [online]. 6., roč. 21, s. 216–225. ISSN 1548-9213. Dostupné z: doi:10.1152/physiol.00001.2006

DIERINGER, N. a W. PRECHT, 1982. Compensatory head and eye movements in the frog and their contribution to stabilization of gaze. *Experimental Brain Research*. roč. 47, č. 3, s. 394–406. ISSN 0014-4819.

DIERINGER, N., S. L. COCHRAN a W. PRECHT, 1983. Differences in the central organization of gaze stabilizing reflexes between frog and turtle. *Journal of comparative physiology* [online]. 12., roč. 153, č. 4, s. 495–508. ISSN 0340-7594, 1432-1351. Dostupné z: doi:10.1007/BF00612604

HEATH, James Edward, R. Glenn NORTH CUTT a Robert P. BARBER, 1969. Rotational optokinesis in reptiles and its bearing on pupillary shape. *Zeitschrift für vergleichende Physiologie* [online]. 3., roč. 62, č. 1, s. 75–85. ISSN 0044-362x, 1432-1351. Dostupné z: doi:10.1007/BF00298043

HEATH, James Edward, R. Glenn NORTH CUTT a Robert P. BARBER, 1969. Rotational optokinesis in reptiles and its bearing on pupillary shape. *Zeitschrift für vergleichende Physiologie* [online]. 3., roč. 62, č. 1, s. 75–85. ISSN 0044-362x, 1432-1351. Dostupné z: doi:10.1007/BF00298043

HSU, L.-J., P. V. ZELENIN, G. N. ORLOVSKY a T. G. DELIAGINA, 2012. Effects of galvanic vestibular stimulation on postural limb reflexes and neurons of spinal postural network. *Journal of Neurophysiology* [online]. 7., roč. 108, č. 1, s. 300–313. ISSN 1522-1598. Dostupné z: doi:10.1152/jn.00041.2012

KIM, G., M. FERDJALLAH a G. F. HARRIS, 2009. Fast Differential Analysis of Center of Pressure Data in Normal Children and Children with Cerebral Palsy. In: Anthony J. MCGORON, Chen-Zhong LI a Wei-Chiang LIN, ed. 25th Southern Biomedical Engineering Conference 2009, 15 – 17 May 2009, Miami, Florida, USA [online]. B.m.: Springer Berlin Heidelberg, IFMBE Proceedings, 24, s. 341–342 [vid. 2. duben 2016]. ISBN 978-3-642-01696-7. Dostupné z: http://link.springer.com/chapter/10.1007/978-3-642-01697-4_118

LUU, Billy L., Thomas P. HURYN, H. F. Machiel VAN DER LOOS, Elizabeth A. CROFT a Jean-Sébastien BLOUIN, 2011. Validation of a robotic balance system for investigations in the control of human standing balance. *IEEE transactions on neural systems and rehabilitation engineering: a publication of the IEEE Engineering in Medicine and Biology Society* [online]. 8., roč. 19, č. 4, s. 382–390. ISSN 1558-0210. Dostupné z: doi:10.1109/TNSRE.2011.2140332

LYALKA, V. F., G. N. ORLOVSKY a T. G. DELIAGINA, 2009. Impairment of Postural Control in Rabbits With Extensive Spinal Lesions. *Journal of Neurophysiology* [online]. 4., roč. 101, č. 4, s. 1932–1940. ISSN 0022-3077. Dostupné z: doi:10.1152/jn.00009.2008

LYALKA, V. F., P. V. ZELENIN, A. KARAYANNIDOU, G. N. ORLOVSKY, S. GRILLNER a T. G. DELIAGINA, 2005. Impairment and recovery of postural control in rabbits with spinal cord lesions. *Journal of Neurophysiology* [online]. 12., roč. 94, č. 6, s. 3677–3690. ISSN 0022-3077. Dostupné z: doi:10.1152/jn.00538.2005

- MACPHERSON, Jane M., Dirk G. EVERAERT, Paul J. STAPLEY a Lena H. TING, 2007. Bilateral Vestibular Loss in Cats Leads to Active Destabilization of Balance During Pitch and Roll Rotations of the Support Surface. *J Neurophysiol* [online]. 9.4., roč. 97, č. 1, s. 4357-4367. ISSN 0022-3077, 1522-1598. Dostupné z: doi:10.1152/jn.01338.2006
- PAYNE, Jeff., 2013. *PLC vs. PAC | Control Engineering* [online] [vid. 8. duben 2016]. Dostupné z: <http://www.controleng.com/single-article/plc-vs-pac/44448cf771be09bff7115c621633bd94.html>
- SHINDER, Michael E. a Jeffrey S. TAUBE, 2011. Active and passive movement are encoded equally by head direction cells in the anterodorsal thalamus. *Journal of Neurophysiology* [online]. 8., roč. 106, č. 2, s. 788–800. ISSN 1522-1598. Dostupné z: doi:10.1152/jn.01098.2010
- SCHUBERT, Patric, Marietta KIRCHNER, Dietmar SCHMIDTBLEICHER a Christian T. HAAS, 2012. About the structure of posturography: Sampling duration, parametrization, focus of attention (part I). *Journal of Biomedical Science and Engineering* [online]. roč. 05, č. 09, s. 496–507. ISSN 1937-6871, 1937-688X. Dostupné z: doi:10.4236/jbise.2012.59062
- ŠKODA, David. Metody hodnocení polohy horní končetiny pomocí Xsens systému [online]. Praha, 2014 [cit. 2016-04-02]. Bakalářská práce. České vysoké učení technické v Praze, Fakulta biomedicínského inženýrství. Dostupné z: <http://theses.cz/id/3iypq2/>
- TESZLER, C. B., J. BEN-DAVID, L. PODOSHI a E. SABO, 2000. ITJ - The International Tinnitus Journal - Sonovestibular Symptoms Evaluated by Computed Dynamic Posturography. *The International Tinnitus Journal*. roč. 6, č. 2, s. 140–153.
- THADEWALD, Thorsten a Herbert BÜNING, 2007. Jarque–Bera Test and its Competitors for Testing Normality – A Power Comparison. *Journal of Applied Statistics* [online]. 1.1., roč. 34, č. 1, s. 87–105. ISSN 0266-4763. Dostupné z: doi:10.1080/02664760600994539
- TING, Lena H. a Jane M. MACPHERSON, 2004. Ratio of shear to load ground-reaction force may underlie the directional tuning of the automatic postural response to rotation and translation. *Journal of Neurophysiology* [online]. 8., roč. 92, č. 2, s. 808–823. ISSN 0022-3077. Dostupné z: doi:10.1152/jn.00773.2003
- TSAI, Yi-Ching, Lin-Fen HSIEH a Saiwei YANG, 2014. Age-related changes in posture response under a continuous and unexpected perturbation. *Journal of Biomechanics* [online]. 22.1., roč. 47, č. 2, s. 482–490. ISSN 1873-2380. Dostupné z: doi:10.1016/j.jbiomech.2013.10.047

Seznam použitých zkratek

ACC:	Adaptive Color Change
ANOVA:	Analysis of Variance test
CMS:	Camera Measurement Systém
ERP:	Electric Revolvable platform
Feature:	Synonymum k utilitě
FLASH:	Typ mazatelné paměti mikrokontroléru
FPS:	Frames per Second
FrameRate:	Počet snímků za sekundu (analogie k FPS)
GUI:	Graphical User Interface
HDD:	Hard Disk Drive
HW:	HardWare
I/O:	Input/Output
LED:	Light Emitting Diode
MoCap:	Motion Capture
PAC:	Programmable Automation Controller
PITCH:	Elevace (pohyb v sagitální rovině živočicha)
PLC:	Programmable Logic Controller
PřF UK	Přírodovědná Fakulta Univerzity Karlovy
RGB:	Barevné kanály pixelu (červená, zelená, modrá)
ROLL:	Rotace kolem hlavní osy (pohyb ve frontální rovině živočicha)
SSD:	Solid State Drive
SW:	SoftWare
UART:	Řídící mikrokontrolér
YAW:	Změna azimutu (pohyb v transverzální rovině živočicha)
ε :	Faradayovo indukované napětí na cívce (V)
Φ :	Magnetický tok plochou průřezu cívky (Wb)

Seznam obrázků

<i>Obrázek 2.1: Návrh uspořádání experimentu na králících podle (Beloozerova, 2003).....</i>	<i>3</i>
<i>Obrázek 2.2: Uspořádání experimentu rozšířeného měření na králících (Hsu, 2012).....</i>	<i>4</i>
<i>Obrázek 2.3: Nákras k experimentu na kočce domácí (Macpherson, 2007).</i>	<i>5</i>
<i>Obrázek 2.4: Demonstrace anatomických rovin zvířete a sledovaných rotací.....</i>	<i>7</i>
<i>Obrázek 4.1: Schémata a fotografie základny plošiny.....</i>	<i>10</i>
<i>Obrázek 4.2: Princip činnosti krokového motoru.....</i>	<i>14</i>
<i>Obrázek 4.3: Fotografický snímek hotové plošiny ve standby režimu.</i>	<i>17</i>
<i>Obrázek 5.1: Ukázka z GUI pro manuální řízení motorů.....</i>	<i>21</i>
<i>Obrázek 5.2: Navržené GUI pro sekvenční řízení motorů.....</i>	<i>23</i>
<i>Obrázek 5.3: Schéma algoritmu pro extrakci časových razítek k určení FPS a vykreslení grafů.....</i>	<i>28</i>
<i>Obrázek 5.4: Ilustrace zpoždění při zahájení a ukončení nahrávání.....</i>	<i>29</i>
<i>Obrázek 5.5: Diagram algoritmu pro zjišťování použitelného úseku videa.....</i>	<i>30</i>
<i>Obrázek 5.6: Schéma časové synchronizace pomocí duplikací snímků ve videích.....</i>	<i>31</i>
<i>Obrázek 5.7: Princip k-means algoritmu.....</i>	<i>32</i>
<i>Obrázek 5.8: Demonstrace poloměru oříznutí.....</i>	<i>33</i>
<i>Obrázek 5.9: Znázornění hierarchie vektoru klíčových snímků a ACC vektoru.....</i>	<i>34</i>
<i>Obrázek 5.10: Princip výpočtu úhlu mezi markery.....</i>	<i>35</i>
<i>Obrázek 5.11: Ukázka GUI pro nahrávání z kamer.....</i>	<i>37</i>
<i>Obrázek 5.12: Ukázka GUI pro selekci interpolací a export veličin.....</i>	<i>39</i>
<i>Obrázek 5.14: Ukázka z GUI pro nastavení detekce markerů.....</i>	<i>40</i>
<i>Obrázek 5.15: Ukázka GUI pro export úhlů a videí.....</i>	<i>41</i>
<i>Obrázek 5.16: Schéma komunikace mezi CMS PC a ERP PC při zahájení nahrávání.....</i>	<i>43</i>
<i>Obrázek 6.1: Příklad sledovaných veličin ze záznamu pohybu zvířete v čase.....</i>	<i>45</i>

Seznam grafů

<i>Graf 7.1: Ukázka měření gekončíka s identifikačním číslem 08297</i> -----	48
<i>Graf 7.2: Ukázka měření gekončíka s identifikačním číslem 421</i> -----	49
<i>Graf 7.3: Ukázka měření <i>Dyscophus Guineti</i> s identifikačním číslem 209</i> -----	49
<i>Graf 7.4: Ukázka grafu vzájemné závislosti úhlu hlavy zvířete na úhlu pohybu plošiny</i> -----	50
<i>Graf 7.5: Ukázka boxplotu sledované veličiny ROM, rychlost 2,5°/s</i> -----	52
<i>Graf 7.6: Ukázka boxplotu sledované veličiny ROM, rychlost 5,0°/s</i> -----	53
<i>Graf 7.7: Ukázka boxplotu délky trajektorie</i> -----	54

Seznam tabulek

<i>Tabulka 7.1: Ukázka exportovaných ukazatelů pro ID-209 (20-05-2015 19 hod 07 min 58 sec)</i> -----	51
<i>Tabulka 7.2: Ukázka sumarizace výsledků pro měření na gekončících ze dne 14. 12. 2015</i> -----	51
<i>Tabulka 7.3: Výsledky p-čísla Shapiro-Wilkova (S-W) a Jarque-Bera (J-B) testu normality dat</i> -----	55