VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

VLÁČEK ŘÍZENÝ POČÍTAČEM

MODEL TRAIN CONTROLLED BY COMPUTER

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **MATOUŠ HAVLÍČEK**
AUTHOR

VEDOUCÍ PRÁCE **prof. Dr. Ing. PAVEL ZEMČÍK, dr. h. c.**
SUPERVISOR

BRNO 2026

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Zadání bakalářské práce

172501

Ústav:	Ústav počítačové grafiky a multimédií (UPGM)
Student:	**Havlíček Matouš**
Program:	Informační technologie
Název:	**Vláček řízený počítačem**
Kategorie:	Vestavěné systémy
Akademický rok:	2025/26

Zadání:

1. Prostudujte literaturu a dostupná řešení na téma řízení modelu vláčku počítačem, případně "embedded počítačem". Prostudujte i existující realizace takového řízení.
2. Navrhněte vhodný způsob řízení vláčku a také způsob jeho ovládání uživatelem, přitom zvažte použití mobilního zařízení jako prostředku uživatelského rozhraní.
3. Analyzujte a popište možnosti a dosažitelné vlastnosti navrženého řešení.
4. Navržené řešení implementujte a demonstrujte například na jednoduchém malém kolejišti.
5. Diskutujte dosažené výsledky a možnosti pokračování práce.

Literatura:

- Dle pokynů vedoucího

Při obhajobě semestrální části projektu je požadováno:
Body 1-3 zadání

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce:	**Zemčík Pavel, prof. Dr. Ing., dr. h. c.**
Vedoucí ústavu:	Černocký Jan, prof. Dr. Ing.
Datum zadání:	1.11.2025
Termín pro odevzdání:	13.5.2026
Datum schválení:	7.11.2025

Abstrakt

Tato bakalářská práce se zabývá návrhem a implementací autonomního řídicího systému pro modelovou železnici, založeného na konceptu digitálního dvojčete. Hlavním přínosem tohoto řešení je odstranění tradičních hardwarových senzorů v kolejišti, které jsou nahrazeny bezkontaktním sledováním, s využitím webkamery a metod počítačového vidění. ArUco markery a sledování pohybu jsou použity pro přesnou lokalizaci modelového vlaku v reálném čase. Řídicí software byl vyvinut v Pythonu s využitím knihoven OpenCV a PySide6, přičemž zpracovává obrazová data a prostřednictvím platformy Arduino odesílá DCC instrukce přímo do lokomotivy. Výsledný systém umožňuje plynulou vizualizaci, manuální ovládání a automatizaci jízdních řádů s vysokou přesností bez nutnosti invazivních zásahů do infrastruktury kolejiště.

Abstract

This bachelor thesis focuses on the design and implementation of an autonomous control system for a model railway based on the digital twin concept. The primary contribution of this solution is the elimination of traditional hardware sensors embedded in the tracks, which are replaced by non-contact tracking using a webcam and computer vision methods. ArUco fiducial markers and motion tracking are utilized for precise real-time locomotive localization. The control software, developed in Python using the OpenCV and PySide6 libraries, processes image data and sends DCC-standard instructions to the locomotive via the Arduino platform. The resulting system enables smooth visualization, manual control, and schedule automation with high positional accuracy without the need for intrusive modifications to the railway infrastructure.

Klíčová slova

počítačová simulace, modelové kolejiště, počítačové vidění, digitální dvojče, Python, OpenCV, ArUco markery, Arduino, DCC

Keywords

computer simulation, model train track, computer vision, digital twin, Python, OpenCV, ArUco markers, Arduino, DCC

Citace

HAVLÍČEK, Matouš. *Vláček řízený počítačem*. Brno, 2026. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce prof. Dr. Ing. Pavel Zemčík, dr. h. c.

Vláček řízený počítačem

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana prof. Dr. Ing. Pavla Zemčíka, dr. h. c. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal. Pro jazykovou revizi technické zprávy a asistenci při tvoření práce jsem využil nástroj umělé inteligence Gemini.

.....
Matouš Havlíček
13. května 2026

Poděkování

Děkuji panu profesoru Pavlu Zemčíkovi za odborné vedení mé práce a metodickou pomoc, kterou mi poskytoval i přes velkou pracovní zátěž. Rád bych poděkoval Ing. Tomáši Zemčíkovi, za poskytnutí modelového kolejiště a lokomotivy.

Obsah

Seznam obrázků

Kapitola 1

Úvod

Modelové vláčky fascinují lidi napříč generacemi. Zatímco dříve šlo převážně o oblíbené dětské hračky, dnes kolem nich existuje velká komunita nadšenců, kteří budují rozsáhlé a neuvěřitelně detailní kolejiště. Dnes už ovšem tyto modely neslouží pouze pro zábavu ve volném čase. Často je najdeme i na středních a vysokých školách jako výborné interaktivní učební pomůcky. Na malých, bezpečných modelech si totiž mohou studenti vyzkoušet, jak funguje automatizace, řízení dopravy nebo komunikace mezi počítači, bez rizika škod nebo bezpečnostních následků reálného železničního provozu.

Způsob, jakým tyto modelové železnice ovládáme, se přes léta velmi změnil. Dříve stačilo jen otáčet kolečkem na transformátoru a měnit napětí v kolejích, aby vlak jel rychleji nebo pomaleji. Dnes už je ve světě standardem digitální řízení, kdy do kolejí putují přesné počítačové povely. Díky tomu může na jedné trati jezdit mnoho vlaků nezávisle na sobě. Problémem však stále zůstává vysoká cena a složitost celého systému. Pokud dnes chcete, aby počítač sám řídil vlaky a věděl, kde přesně se na trati nacházejí, musíte si koupit drahé specializované programy a hlavně fyzicky zasáhnout do infrastruktury, tedy rozřezat koleje a zabudovat do nich různé hardwarové senzory a detektory. Existují sice i dostupné bezplatné programy, ale ty stále vyžadují složitou a nákladnou instalaci fyzických snímačů.

Právě tato finanční a technická bariéra mě přivedla k myšlence na směr této práce. Jako studenta informatiky mě velmi zajímá propojování softwaru s reálným fyzickým světem. Chtěl jsem vytvořit něco, co umožní ovládat modely moderně a inteligentně, aniž by člověk musel být expert na slaboproudou elektroniku a musel nevratně zasahovat do samotných kolejí. Využití běžné kamery a zpracování jejího obrazu pomocí počítače mi přišlo jako ideální a velmi perspektivní cesta, jak tento problém vyřešit levně a snadno.

Cílem této práce je navrhnout a naprogramovat systém, který inovativním způsobem propojí počítačovou simulaci s opravdovým fyzickým modelem vlaku. Výsledkem má být aplikace, ve které si uživatel na obrazovce nakreslí svou trať, naplánuje, jak má vlak jezdit, a tento virtuální, plynulý pohyb se okamžitě přenese na skutečný model na stole. Místo toho, abychom do kolejí instalovali drahé senzory, bude na celé kolejiště dohlížet webová kamera. Počítač z jejího obrazu sám rozpozná, kde přesně se vlak nachází. Tím vznikne takzvané digitální dvojče (Digital Twin). Digitální dvojče je koncept, kdy se virtuální svět na monitoru počítače dokonale a v reálném čase synchronizuje s tím, co se děje ve skutečnosti.

Text práce je rozdělen do šesti kapitol. Po tomto úvodu následuje druhá kapitola, která seznamuje se současnými možnostmi řízení modelových železnic a přibližuje, jak dnes fungují různé sledovací systémy z pohledu počítačového vidění. Ve třetí kapitole je popsáno zhodnocení současného stavu a celkový plán práce. Čtvrtá kapitola se podrobně věnuje samotné realizaci a programování – od tvorby uživatelského prostředí, přes výpočet fyziky

pohybu vlaku, až po zpracování obrazu z kamery. Pátá kapitola obsahuje testování vytvořeného systému na reálném hardwaru a zhodnocení toho, jak spolehlivě navržené řešení v praxi funguje. V závěrečné šesté kapitole jsou pak shrnuty dosažené výsledky a naznačené možnosti, jak by se dal tento projekt do budoucna ještě dále rozvíjet.

Kapitola 2

Technologický základ pro kamerové sledování modelové železnice

Tato kapitola popisuje technologie a principy, které jsou důležité pro návrh systému řešeného v této práci. Cílem není vytvořit kompletní encyklopedický přehled celého oboru železničního modelářství nebo počítačového vidění. Kapitola se zaměřuje hlavně na oblasti, které přímo souvisejí s řízením modelové železnice, zjišťováním polohy vozidel, kamerovým sledováním a vytvořením digitálního dvojčete.

Nejdříve je vysvětlen princip řízení modelových železnic, zejména rozdíl mezi analogovým řízením a digitálním standardem DCC. Následuje přehled běžných metod detekce polohy vozidel, protože právě znalost polohy je nutná pro jakoukoliv automatizaci provozu. Dále jsou popsána existující softwarová řešení pro řízení kolejiště a jejich vlastnosti. Poslední část kapitoly se věnuje počítačovému vidění, fiduciozním markerům a principu digitálního dvojčete, které tvoří základ navrhovaného řešení.

2.1 Řízení modelové železnice

Aby bylo možné modelovou železnici automatizovat, je potřeba řešit dvě základní věci. První z nich je samotné řízení vlaků, tedy možnost určovat jejich rychlost, směr jízdy a další funkce. Druhou je zpětná vazba o tom, kde se vozidla na kolejišti právě nacházejí. Tyto dvě části spolu úzce souvisí, ale nejsou stejné. Systém může umět vlak ovládat, ale to ještě neznamená, že přesně ví, kde se vlak nachází.

V praxi se pro řízení modelových železnic používají hlavně dva přístupy. Starší analogové řízení je jednodušší, ale má výrazná omezení při provozu více vlaků. Modernější digitální řízení, zejména standard DCC [2], umožňuje ovládat více vozidel nezávisle na sobě. Samotné DCC ale neřeší přesné sledování polohy vozidel, což je jeden z hlavních důvodů, proč se tato práce zabývá doplňkovým kamerovým sledováním.

2.1.1 Analogové řízení

Tradiční analogové řízení modelové železnice je založené na přímé regulaci stejnosměrného napětí v kolejích. Rychlost motoru lokomotivy závisí na velikosti přivedeného napětí a směr jízdy se mění změnou polarity. Napětí se typicky pohybuje přibližně v rozsahu od nuly do dvanácti, případně šestnácti voltů podle použitého měřítka a konkrétního systému [16].

Tento princip je jednoduchý a pro malé kolejiště může být dostatečný. Jeho hlavní nevýhoda se ale projeví ve chvíli, kdy je na jednom elektricky propojeném úseku více lokomotiv.

Všechny lokomotivy reagují na stejnou změnu napětí, a proto není možné je řídit nezávisle. Pokud má jedna lokomotiva zastavit a druhá pokračovat, musí být trať rozdělena na samostatné elektrické úseky. Tím se zvyšuje složitost zapojení a roste množství kabeláže.

Analogové řízení tedy dobře ukazuje základní problém modelové železnice: samotné napájení kolejí nestačí pro pohodlné řízení více vozidel. Pro složitější provoz je potřeba systém, který dokáže posílat konkrétní příkazy konkrétním lokomotivám.

2.1.2 Digital Command Control

V současné době je nejrozšířenějším digitálním standardem pro řízení modelových železnic systém DCC [2], tedy Digital Command Control. Tento systém původně vyvinul Bernd Lenz v roce 1988 pro evropské výrobce modelů [13]. Architekturu následně převzala organizace National Model Railroad Association (NMRA), která ji v roce 1994 poprvé formálně standardizovala a publikovala ve svém časopise NMRA Bulletin [2]. Technické parametry jsou popsány hlavně v normách S-9.1 a S-9.2 [17, 15].

Základní rozdíl oproti analogovému řízení spočívá v tom, že kolejemi neprotéká plynule měněné stejnosměrné napětí. Místo toho je do kolejí přiváděn digitální signál ve formě obdélníkového střídavého napětí. Toto napětí má dvě funkce současně. Slouží jako zdroj energie pro lokomotivu a její elektroniku, ale zároveň nese digitální datové pakety s příkazy.

Jednotlivé lokomotivy jsou vybaveny dekodérem, který průběžně čte signál z kolejí. Pokud dekodér najde paket určený pro svoji adresu, provede příslušný příkaz. Může jít například o změnu rychlosti, změnu směru jízdy, rozsvícení světel nebo spuštění zvukové funkce. Díky tomu může být na stejné trati více lokomotiv, ale každá reaguje jen na příkazy určené pro ni.

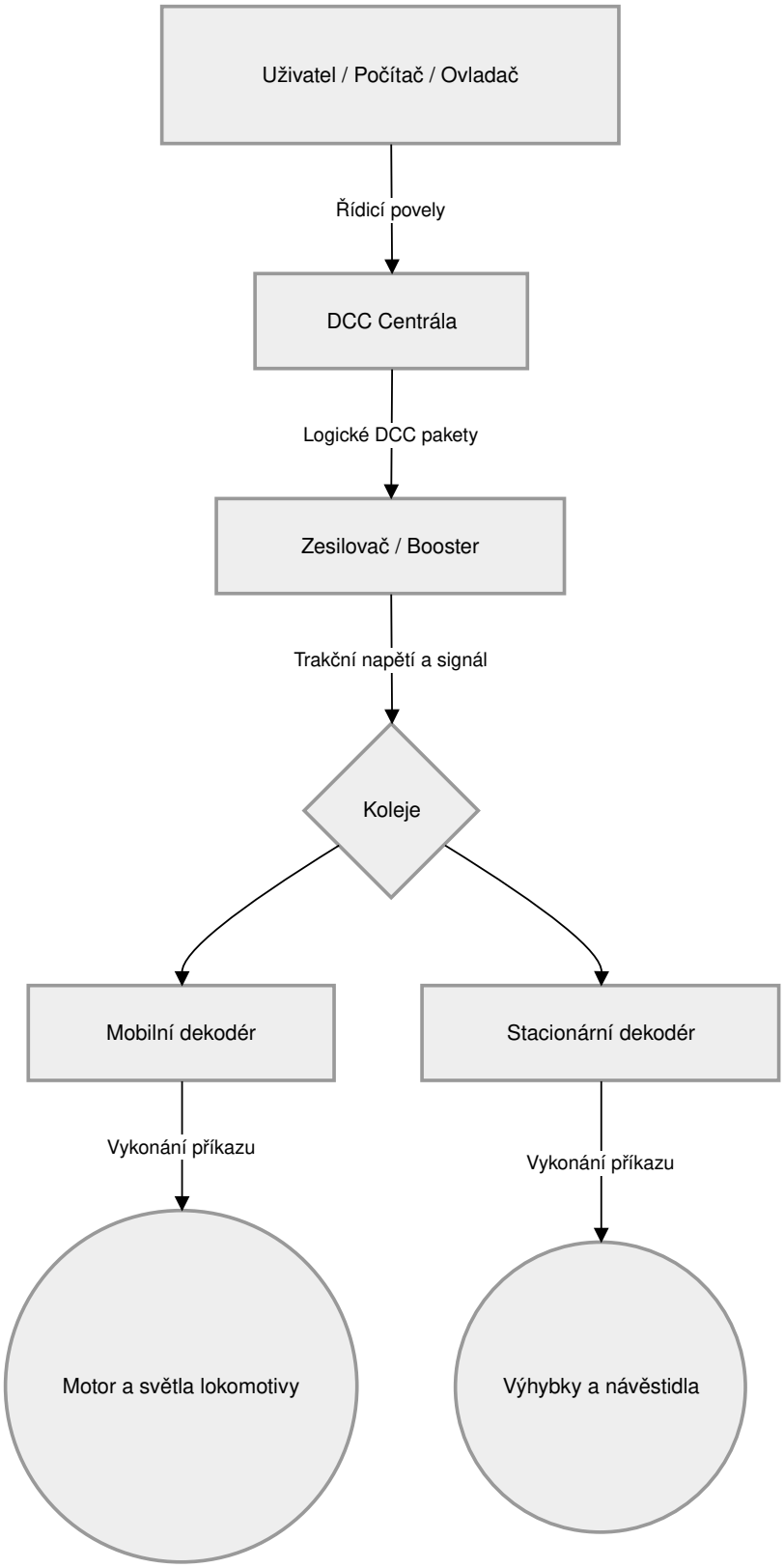
Logické hodnoty v DCC signálu nejsou kódovány změnou amplitudy, ale délkou pulzu. Logická jednička je reprezentována kratší periodou signálu, zatímco logická nula periodou delší. Norma NMRA S-9.1 uvádí nominální délku 58 mikrosekund pro každou polaritu u logické jedničky a minimálně 100 mikrosekund pro každou polaritu u logické nuly [17]. Tento způsob přenosu je vhodný pro prostředí modelové železnice, kde se mohou objevovat krátkodobé výpadky kontaktu mezi koly a kolejnicí.

Pro tuto práci je důležité hlavně to, že DCC velmi dobře řeší řízení vozidel, ale neřeší přesnou informaci o jejich poloze. Centrála může poslat lokomotivě příkaz, aby jela určitou rychlostí, ale ze samotného DCC signálu obvykle neví, kde se lokomotiva v daný okamžik nachází. Tato chybějící zpětná vazba musí být doplněna jiným způsobem.

2.1.3 Architektura systému DCC

Typický DCC systém se skládá z několika hardwarových částí, které spolupracují při řízení provozu.

- **Centrála** přijímá povely od uživatele, ovladače nebo počítače. Tyto povely převádí do formátu DCC paketů a stará se o jejich opakované vysílání.
- **Zesilovač**, často označovaný jako booster, převádí logický DCC signál na trakční napětí a dodává do kolejí potřebný proud.
- **Mobilní dekodér** je umístěn přímo v lokomotivě. Čte pakety z kolejí a vykonává příkazy určené pro danou adresu.
- **Stacionární dekodér** se používá pro pevné prvky kolejiště, například výhybky, návěstidla nebo osvětlení.



Obrázek 2.1: Zjednodušená architektura DCC systému: uživatel nebo počítač posílá povely do centrály, centrála generuje DCC pakety, booster je zesílí a dekodér v lokomotivě provede příkaz.

Datový paket DCC se skládá z několika částí. Obsahuje synchronizační preambuli, adresní bajt, instrukční část a kontrolní součet. Kontrolní součet pomáhá odhalit poškozené pakety, které mohou vzniknout například při rušení nebo krátkém výpadku kontaktu [15]. Díky adresaci dokáže dekodér rozlišit, zda je příkaz určen právě jemu.

Z pohledu této práce je DCC vhodné jako řídicí vrstva, protože umožňuje posílat konkrétní povely konkrétním lokomotivám. Nestačí však jako zdroj informací o poloze. Proto je potřeba doplnit systém o další vrstvu, která bude sledovat skutečný pohyb vozidel.

2.2 Detekce polohy kolejových vozidel

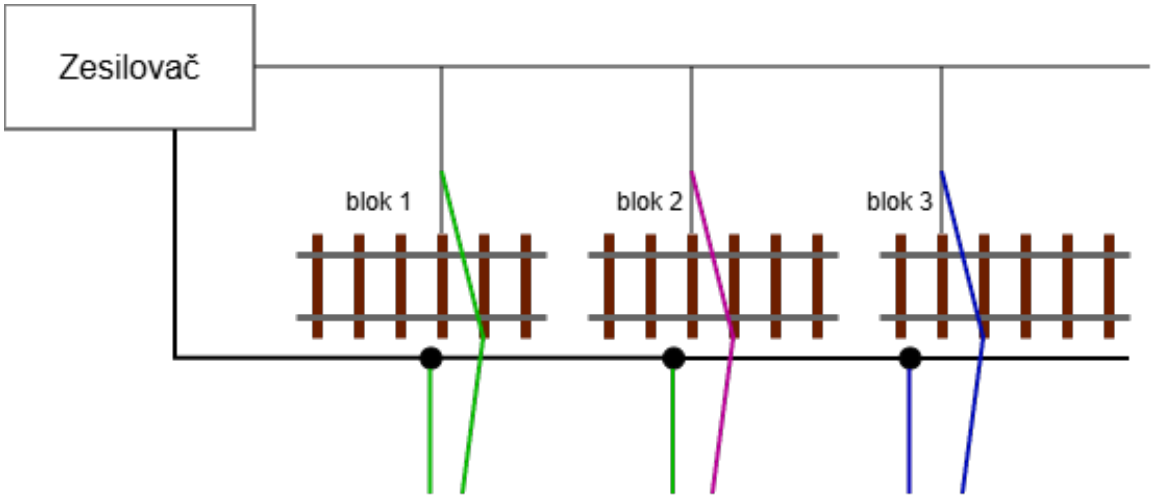
Přesná znalost polohy vozidel je nutná pro bezpečný a automatizovaný provoz. Pokud systém neví, kde se vlak nachází, nemůže spolehlivě rozhodnout, zda je úsek volný, zda může vlak pokračovat, nebo zda hrozí kolize. U modelové železnice se proto běžně používají různé formy zpětné vazby.

DCC je ve své základní podobě převážně jednosměrný systém. Centrála posílá příkazy do kolejí, ale lokomotivy samy od sebe nevracejí přesnou informaci o své poloze. V praxi se proto používají dodatečné senzory, které zjišťují obsazení úseků nebo průjezd konkrétním místem [22].

2.2.1 Bloková detekce obsazení

Jednou z nejběžnějších metod je bloková detekce obsazení. Trať je rozdělena na elektricky oddělené úseky, které se nazývají bloky. Každý blok je napájen přes detekční modul. Pokud se v bloku nachází vozidlo, které odebírá proud, detektor vyhodnotí úsek jako obsazený [22].

Tento princip je v praxi poměrně spolehlivý. Dobře funguje hlavně u lokomotiv, protože ty přirozeně odebírají proud pro motor a elektroniku. U běžných vagónů je situace složitější. Prázdný vagón nemusí odebírat žádný proud, a proto ho detekční systém nemusí vidět. Aby byl takový vagón zjistitelný, musí být doplněn například odporem mezi nápravami.



Obrázek 2.2: Princip blokové detekce obsazení. Kolejiště je rozděleno na izolované úseky a každý úsek je vyhodnocován samostatným detekčním modulem.

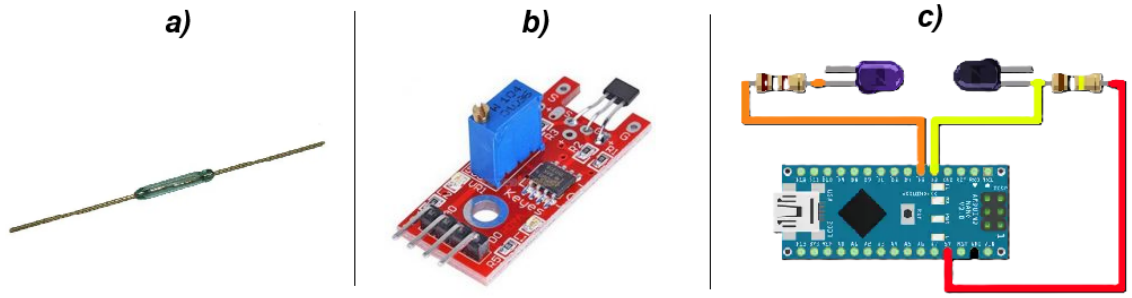
Nevýhodou blokové detekce je potřeba zásahu do fyzické infrastruktury. Koleje musí být rozděleny na izolované úseky a každý blok musí být samostatně připojen k detektoru. U většího kolejiště to znamená velké množství kabeláže, složitější zapojení a náročnější údržbu. Dalším omezením je, že systém obvykle poskytuje pouze informaci o obsazení celého bloku, nikoliv přesnou spojitou polohu vlaku uvnitř tohoto bloku.

2.2.2 Bodové snímače polohy

Vedle blokové detekce se používají také bodové snímače. Ty neříkají, že je obsazený celý úsek, ale zaznamenají průjezd vozidla konkrétním místem. Jsou vhodné například pro kontrolu průjezdu kolem návěstidla, zastavení ve stanici nebo spuštění určité akce.

Mezi běžné bodové snímače patří:

- **Jazýčkové kontakty** jsou malé skleněné součástky s kovovými kontakty uvnitř. Sepnou se při přiblížení magnetu, který bývá umístěn na vozidle.
- **Hallovy sondy** využívají Hallova jevu. Také reagují na magnetické pole, ale neobsahují mechanicky pohyblivé části, což zvyšuje jejich životnost a spolehlivost [23].
- **Optické brány** pracují s přerušením světelného nebo infračerveného paprsku. Když vozidlo projede mezi vysílačem a přijímačem, paprsek se přeruší a systém zaznamená průjezd.



Obrázek 2.3: Příklady hardwarových prvků pro bodovou detekci: a) jazýčkový kontakt, b) modul s Hallovou sondou a komparátorem, c) zapojení infračervené optické brány. Původní obrázky do koláže převzaty z [6], [3], [18].

Bodové snímače mohou být přesné, ale poskytují informaci pouze v konkrétním místě. Pokud vlak projede mezi dvěma snímači, systém jeho přesnou polohu pouze odhaduje. Pro plynulé sledování pohybu by bylo potřeba velké množství snímačů, což opět zvyšuje složitost celého řešení.

2.3 Existující softwarová řešení

Kromě hardwaru existuje také řada softwarových nástrojů, které slouží k řízení modelové železnice. Tyto programy obvykle zobrazují plán kolejiště, zpracovávají informace ze senzorů a posílají příkazy do DCC centrály. Díky tomu lze vytvářet automatizované trasy, řídit provoz ve stanicích nebo zabránit kolizím.

Pro tuto práci je důležité zjistit, jakým způsobem tato řešení pracují s informací o poloze vlaků. Většina běžně používaných systémů je postavena na diskrétní zpětné vazbě ze senzorů. To znamená, že pracují se stavy typu „blok je obsazený“ nebo „senzor byl sepnutý“. Nepředpokládají, že poloha vlaku bude kontinuálně získávána z obrazu kamery.

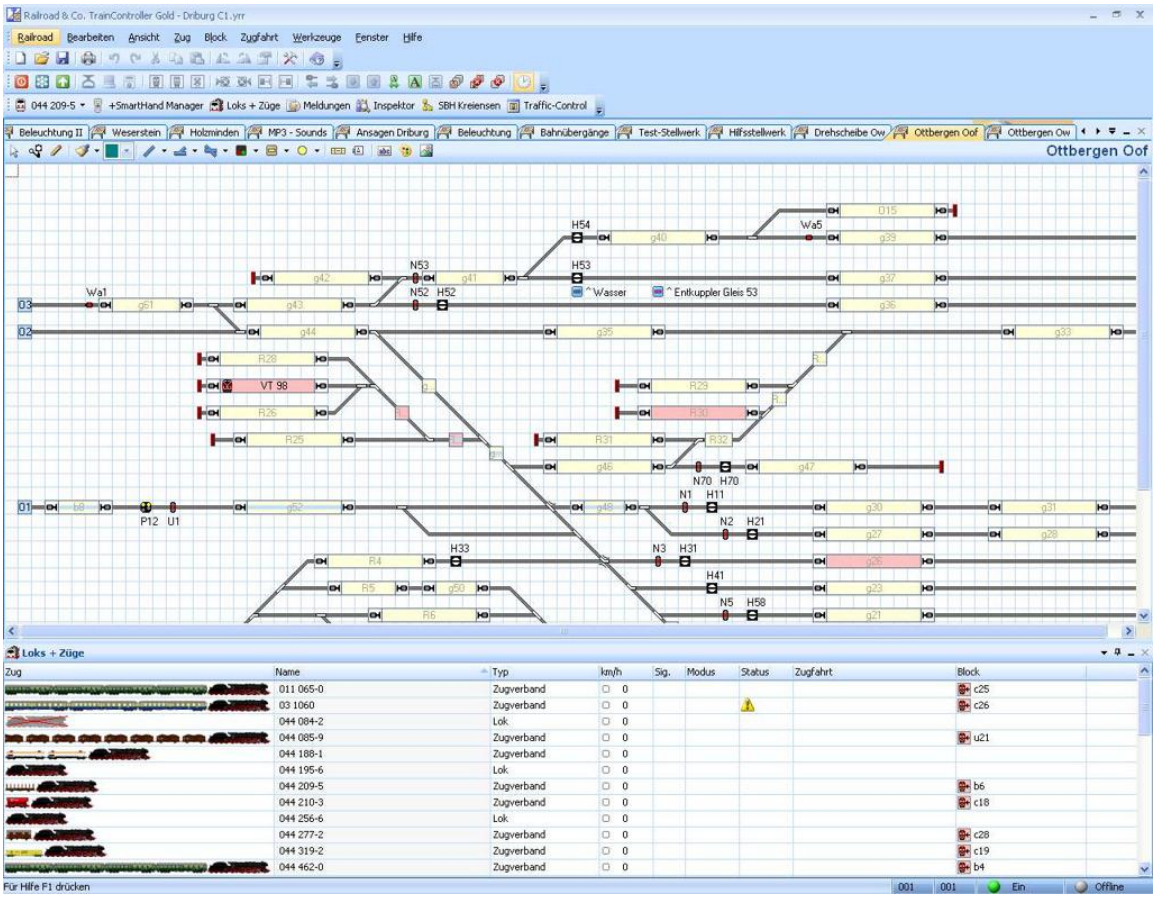
2.3.1 Komerční dispečerské systémy

Mezi známá komerční řešení patří například TrainController [7], [4] nebo Win-Digipet [19]. Tyto programy umožňují vytvořit virtuální plán kolejiště a propojit ho s fyzickou železnicí. Uživatel může definovat bloky, výhybky, návěstidla, trasy a pravidla provozu.

Typickým příkladem je TrainController, který pracuje s blokově orientovanou logikou. Kolejiště je ve softwaru rozděleno na logické úseky, které odpovídají fyzickým blokům vybaveným detekcí obsazení. Program na základě těchto informací rozhoduje, kam může vlak jet, které úseky musí být rezervovány a kdy je potřeba vlak zastavit.

Důležitou součástí těchto systémů je prevence konfliktních stavů. Pokud například dva vlaky jedou proti sobě po jednokolejné trati, software musí zabránit situaci, kdy se navzájem zablokují a žádný z nich nemůže pokračovat. Takový stav se označuje jako deadlock. Komerční dispečerské systémy proto pracují s rezervací cest a průběžně kontrolují, zda je plánovaná trasa bezpečná.

Silnou stránkou komerčních systémů je jejich propracovanost a množství funkcí. Slabší stránkou z pohledu této práce je jejich závislost na tradiční zpětné vazbě. Systém obvykle očekává data z bloků nebo bodových senzorů, nikoliv souřadnice získané kamerou.



Obrázek 2.4: Ukázka grafického rozhraní dispečera v programu TrainController [8].

2.3.2 Otevřená řešení

V oblasti otevřeného softwaru je významným projektem JMRI [11], tedy Java Model Railroad Interface. JMRI je platforma napsaná v jazyce Java a obsahuje několik nástrojů pro práci s modelovou železnicí. DecoderPro slouží hlavně ke konfiguraci dekodérů, zatímco PanelPro umožňuje vytvářet ovládací panely a vizualizace kolejiště.

Dalším známým řešením je Rocrail [20]. Jeho architektura je založena na rozdělení na serverovou část a klientské rozhraní. Výpočetní část může běžet na jednom zařízení, zatímco uživatelé se k ní připojují z jiných zařízení. Rocrail podporuje automatizaci provozu, práci s bloky, trasami a různými digitálními systémy.

Otevřená řešení mají výhodu v tom, že jsou dostupná zdarma a často umožňují rozsáhlejší úpravy. Přesto i zde platí, že základní způsob práce s polohou vlaků vychází hlavně z klasických senzorů. Software zpracovává informace o obsazení bloků nebo sepnutí kontaktů. Neposkytuje běžně hotový systém pro plynulé kamerové sledování vozidel pomocí markerů.

2.3.3 Zpracování polohových dat v existujících systémech

Společným jmenovatelem většiny současných dispečerských systémů, ať už se jedná o komerční produkty nebo otevřené projekty, je jejich úzká provázanost s fyzickou detekční infrastrukturou. Tyto programy jsou softwarově navrženy primárně pro zpracování diskrét-

ních událostí. Architektura systému očekává skokové změny stavů, typicky informace typu „senzor byl sepnut“ nebo „izolovaný úsek je obsazen“.

V praxi to znamená, že systémy nezískávají z kolejiště spojitou informaci o přesných souřadnicích vozidla. Pokud dispečerský software potřebuje zastavit vlak přesně u nástupiště uvnitř delšího bloku, neprovádí přímé měření jeho reálné polohy. Místo toho využívá metodu interpolace (odhadu). Poloha vlaku je matematicky dopočítávána na základě předem zkalibrovaných rychlostních profilů konkrétní lokomotivy a času, který uplynul od minutí posledního fyzického snímače.

Tento přístup je v oboru železničního modelářství historicky standardizován. Poskytuje spolehlivé výsledky za předpokladu, že je kolejiště vybaveno robustní, dostatečně hustou a precizně zapojenou sítí fyzických detektorů popsaných v předchozích podkapitolách.

2.4 Počítačové vidění

Počítačové vidění je oblast informatiky, která se zabývá zpracováním obrazu a získáváním informací z fotografií nebo videa. V této práci slouží jako alternativa k hardwarovým senzorům. Místo toho, aby bylo kolejiště rozděleno na velké množství elektrických bloků, je pohyb vozidel sledován kamerou.

Použití kamery má několik výhod. Jeden snímač může pokrýt větší část kolejiště a není nutné instalovat senzor do každého úseku trati. Kamera také umožňuje získat více informací než jen stav obsazeno/neobsazeno. Při vhodném zpracování obrazu lze určit nejen přítomnost objektu, ale také jeho polohu, identitu a orientaci.

Současně je ale potřeba počítat s problémy, které jsou pro kamerové systémy běžné. Patří mezi ně změny osvětlení, stíny, odlesky, zkreslení objektivu nebo částečné zakrytí sledovaného objektu. Návrh detekční metody proto musí být dostatečně robustní.

2.4.1 Knihovna OpenCV

Pro implementaci algoritmů počítačového vidění se často používá knihovna OpenCV. Jde o otevřenou knihovnu, která obsahuje velké množství funkcí pro zpracování obrazu, detekci objektů, práci s kamerou, kalibraci a další úlohy. Původně vznikla ve společnosti Intel a dnes patří mezi nejpoužívanější nástroje v oblasti počítačového vidění [5].

OpenCV je napsána hlavně v jazycích C a C++, ale nabízí rozhraní i pro další jazyky, například Python. To je praktické pro prototypování, protože Python umožňuje rychle testovat jednotlivé kroky zpracování obrazu. V kontextu této práce je OpenCV vhodné zejména pro práci s kamerovým obrazem, převody barevných prostorů, detekci markerů a výpočet polohy objektů.

Pro navrhovaný systém je důležité, že OpenCV neřeší pouze samotné načtení obrazu z kamery. Poskytuje také nástroje pro kalibraci kamery, odhad geometrie scény a detekci ArUco markerů. Díky tomu lze na jednom technologickém základu postavit celý sledovací řetězec.

2.4.2 Barevná detekce

Jedním z nejjednodušších způsobů detekce objektu v obraze je sledování konkrétní barvy. Kamera pořídí snímek, obraz se převede do vhodného barevného prostoru a následně se vyberou pixely, které odpovídají zadanému rozsahu barvy. Výsledkem je binární maska, ve které jsou vyznačené oblasti pravděpodobně patřící sledovanému objektu.

Běžný obraz z kamery bývá reprezentován v modelu RGB. Tento model popisuje barvu pomocí červené, zelené a modré složky. Pro detekci barevných objektů ale často není ideální, protože změna osvětlení ovlivňuje všechny tři složky najednou. Proto se používá například barevný prostor HSV, který odděluje odstín barvy od sytosti a jasu.

V úvodní fázi vývoje byl zvažován přístup založený na detekci barevné značky umístěné na vozidle. Tento přístup je jednoduchý a výpočetně nenáročný. Praktické testy ale ukázaly, že je příliš citlivý na světelné podmínky [1]. Lokální stíny, odlesky nebo změna intenzity osvětlení způsobovaly rozpad detekované masky a ztrátu sledování.

Z tohoto důvodu nebyla barevná detekce zvolena jako hlavní metoda. Pro demonstrační nebo velmi kontrolované prostředí by mohla být použitelná, ale pro spolehlivější sledování modelové železnice je vhodnější metoda, která dokáže objekt nejen najít, ale také jednoznačně identifikovat.

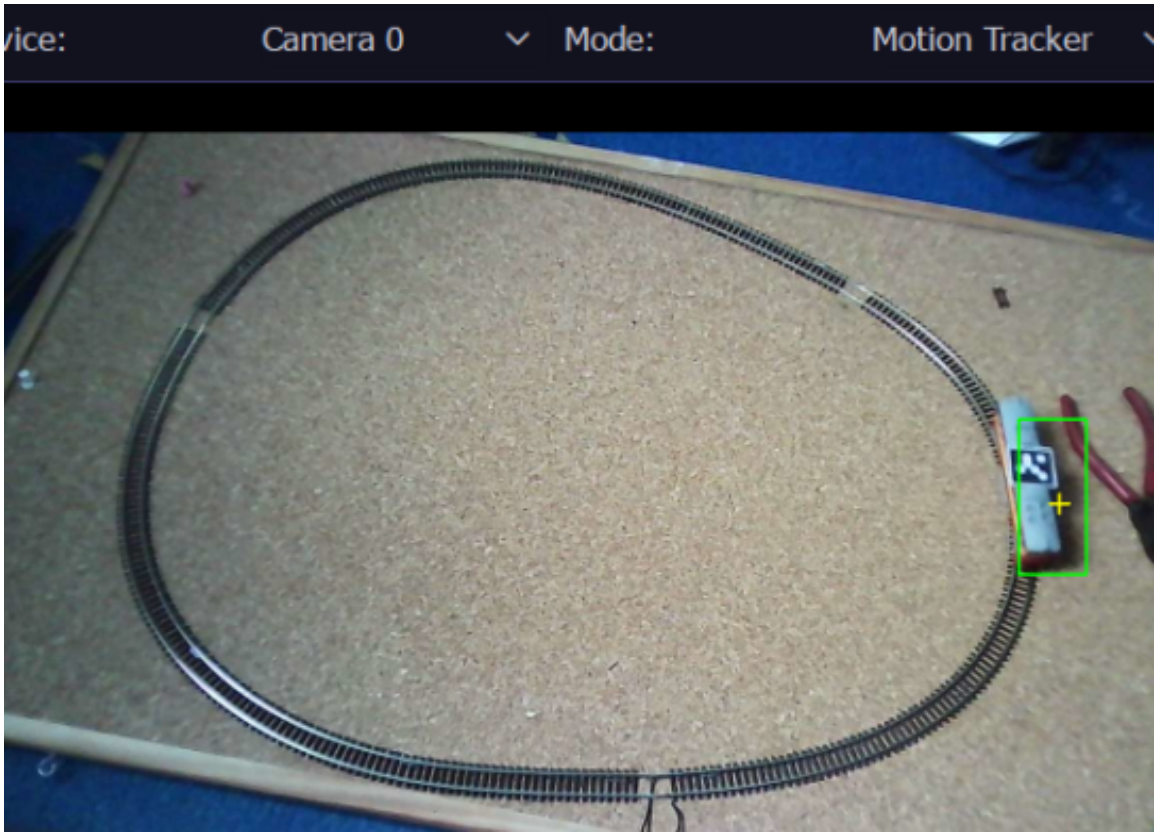
2.4.3 Detekce pohybu a odečítání pozadí

Zatímco barevná detekce se zaměřuje na statické vizuální vlastnosti objektu, v kontextu železničního provozu je často užitečnější zaměřit se na samotný pohyb. Techniky odečítání pozadí (Background Subtraction) slouží k separaci pohybujících se objektů (popředí) od statické scény (pozadí). Výsledkem tohoto procesu je binární maska, která jasně definuje, ve kterých částech obrazu dochází k pohybu.

V praxi je však statické pozadí málokdy ideální. Změny denního světla, pohyb stínů vrhnutých projíždějícím vlakem nebo šum snímače mohou způsobit falešné detekce. Z tohoto důvodu se využívají adaptivní algoritmy, které si udržují a průběžně aktualizují statistický model pozadí pro každý pixel obrazu.

Jedním z běžně používaných algoritmů pro tuto úlohu je MOG2 (Mixture of Gaussians 2), jehož autorem je Zoran Zivkovic [24]. Tento algoritmus modeluje stav každého pixelu pomocí směsi Gaussových rozdělení. Hlavní výhodou algoritmu MOG2 je jeho schopnost automaticky volit optimální počet Gaussových křivek pro daný pixel, což mu umožňuje lépe se adaptovat na dynamické změny ve scéně.

Další zásadní vlastností MOG2, která je užitečná při sledování vozidel, je nativní detekce stínů. Algoritmus dokáže rozeznat pixely, u kterých nedošlo ke změně textury, ale pouze k poklesu jasu. Takové pixely označí jako stín (v masce typicky reprezentované šedou barvou), čímž zabrání tomu, aby byl stín mylně klasifikován jako součást pohybujícího se vlaku.



Obrázek 2.5: Ukázka využití detekce pohybu pomocí algoritmu MOG2.

2.4.4 Fiduciozní markery

Spolehlivější možností je použití fiduciozních markerů. Jedná se o uměle vytvořené vizuální značky, které jsou navržené tak, aby je algoritmus dokázal rychle a přesně najít v obraze. Vypadají podobně jako zjednodušené QR kódy, ale jejich hlavní účel není přenášet velké množství dat. Slouží hlavně k identifikaci objektu a odhadu jeho polohy.

V této práci je využíván standard ArUco [9]. ArUco marker má čtvercový tvar, černý okraj a vnitřní binární vzor. Černý okraj pomáhá algoritmu najít obrys markeru a jeho rohy. Vnitřní vzor kóduje identifikátor markeru, díky kterému lze rozlišit jednotlivá vozidla. Každý vlak tak může nést jiný marker a systém pozná, který vlak právě vidí.

Výhodou ArUco markerů je, že kromě samotné detekce umožňují také odhad orientace. Pokud jsou známé parametry kamery a fyzická velikost markeru, lze ze souřadnic jeho rohů vypočítat polohu a natočení vůči kameře. To je pro sledování vozidel velmi užitečné, protože systém nemusí znát pouze přibližné místo, ale může pracovat i se směrem pohybu.

Oproti barevné detekci jsou markery robustnější. Nejsou závislé pouze na jedné barvě a díky vnitřnímu kódu umožňují jednoznačnou identifikaci. Nevýhodou je, že marker musí být na vozidle fyzicky umístěn a musí být pro kameru viditelný. Pokud je zakrytý nebo příliš malý, detekce může selhat.

Obrázek 2.6: Příklad ArUco markeru a jeho detekce v obraze. Detekční algoritmus určí rohy markeru, přečte jeho identifikátor a může odhadnout jeho polohu vůči kameře.

2.4.5 Odhad polohy a orientace

Samotné nalezení markeru v obraze nestačí. Pro řízení modelové železnice je potřeba převést obrazovou informaci na polohu ve skutečném nebo virtuálním prostoru. K tomu slouží kalibrace kamery a geometrické výpočty.

Kalibrace kamery popisuje vlastnosti použité optiky. Zahrnuje například ohniskovou vzdálenost, střed obrazu a zkreslení čočky. Bez kalibrace může být obraz zkreslený, a proto by vypočítaná poloha markeru nebyla přesná. Po kalibraci lze lépe určit vztah mezi body v obraze a jejich odpovídající polohou v prostoru.

Při znalosti velikosti markeru a parametrů kamery lze použít algoritmus pro řešení úlohy Perspective-n-Point. Tento algoritmus hledá prostorovou pozici objektu podle známých bodů na objektu a jejich projekce v obraze. U ArUco markeru jsou těmito body jeho čtyři rohy. Výsledkem je odhad posunutí a rotace markeru vůči kameře.

V navrhovaném systému se tato informace dá použít k aktualizaci polohy vlaku v digitálním dvojčeti. Kamera tedy neposkytuje pouze obraz, ale stává se zdrojem dat o reálném pohybu vozidel.

2.5 Digitální dvojče

Digitální dvojče je virtuální reprezentace reálného fyzického systému, která je průběžně aktualizována daty z reality. Nejde tedy jen o běžný statický model. Důležité je propojení mezi fyzickým objektem a jeho digitální kopií. Digitální model má co nejlépe odpovídat skutečnému stavu fyzického systému.

Pojem digitální dvojče je často spojován s prací Michaela Grievese a s oblastí řízení životního cyklu výrobků. Později se tento koncept rozšířil i do dalších oblastí, například průmyslu, robotiky, dopravy nebo simulací [10]. Obecně lze říct, že digitální dvojče pomáhá sledovat, analyzovat a řídit fyzický systém pomocí jeho softwarové reprezentace.

2.5.1 Základní části digitálního dvojčete

Plnohodnotné digitální dvojče se obvykle skládá ze tří základních částí:

1. **Fyzická entita** je reálný objekt nebo systém. V této práci jde o fyzické kolejiště, lokomotivy, výhybky a další prvky modelové železnice.

2. **Virtuální entita** je softwarový model fyzického systému. V tomto případě jde o digitální reprezentaci kolejiště, polohy vlaků a jejich pohybu.

3. **Datové propojení** zajišťuje výměnu informací mezi fyzickou a virtuální částí. Do virtuálního modelu proudí data z kamery a z virtuálního modelu mohou zpět odcházet řídicí příkazy [21].

V kontextu modelové železnice může digitální dvojče sloužit jako prostředník mezi realitou a řízením. Fyzické vlaky se pohybují po kolejích, kamera sleduje jejich skutečnou polohu a software tuto informaci promítá do virtuální mapy. Díky tomu může systém porovnávat očekávanou a skutečnou polohu vozidel.

2.5.2 Digitální dvojče v kyber-fyzikálních systémech

V moderních kyber-fyzikálních systémech (CPS) neslouží digitální dvojče pouze jako pasivní vizualizace, ale tvoří jádro pro takzvanou uzavřenou regulační smyčku [21]. Propojení mezi fyzickou a virtuální částí je v těchto případech obousměrné a nepřetržité [12].

Směrem z fyzického světa do softwaru proudí telemetrická a senzorická data, která v reálném čase aktualizují stav virtuální entity. Software na základě těchto přesných dat modeluje chování systému, predikuje budoucí stavy a následně generuje řídicí příkazy. Tyto příkazy jsou odesílány zpět do fyzického světa, kde je vykonávají akční členy (motory, ventily, relé).

Kritickým parametrem pro úspěšné fungování takové regulační smyčky je systémová latence (celkové zpoždění). Pokud je zpracování senzorických dat nebo následná komunikace příliš pomalá, virtuální model začne za realitou zaostávat. Systém pak pracuje se zastaralými daty, což může vést k nepřesným nebo nesprávným řídicím zásahům. Rychlost a determinismus datového propojení jsou proto při návrhu digitálního dvojčete pro řídicí účely naprosto klíčové [14].

2.6 Shrnutí kapitoly

Tato kapitola shrnula teoretické poznatky a technologické principy z oblasti řízení modelových železnic a zpracování obrazu. Nejprve byl popsán standard Digital Command Control (DCC), který na rozdíl od tradičního analogového přístupu umožňuje nezávislé digitální adresování a řízení více hnacích vozidel na jedné trati.

Následně byly rozebrány konvenční metody získávání zpětné vazby, reprezentované především blokovou detekcí obsazení a bodovými snímači. Z provedené rešerše vyplývá, že tyto systémy poskytují řídicímu softwaru diskrétní události o poloze vozidel a jejich nasazení je technicky podmíněno fyzickými úpravami samotné infrastruktury kolejiště. Přehled existujících komerčních i open-source dispečerských aplikací (jako jsou TrainController, iTrain či JMRI) potvrdil, že současné systémy jsou architektonicky primárně navrženy právě pro spolupráci s touto hardwarovou senzorikou.

Poslední část textu byla věnována teoretickému základu alternativních přístupů k lokalizaci vozidel. Byly představeny principy počítačového vidění, nástroje knihovny OpenCV a techniky detekce objektů, s důrazem na porovnání barevného prahování a robustnějšího sledování pomocí fiduciozních značek ArUco. Společně s definicí konceptu digitálního dvojčete tak byly popsány všechny klíčové technologie nezbytné pro porozumění moderním kyber-fyzikálním systémům.

Kapitola 3

Zhodnocení současného stavu a plán práce

Tato kapitola kriticky reflektuje technologie a existující systémy popsané v teoretickém úvodu. Na základě vlastního zhodnocení a konzultací s vedoucím práce jsou zde identifikovány hlavní nedostatky současných řešení. V reakci na ně je definován plán práce, stanoveny systémové požadavky a navržena konkrétní hardwarová a softwarová architektura, která tyto nedostatky odstraňuje.

3.1 Kritické zhodnocení dosavadního stavu

Dostupná řešení pro automatizované řízení modelových železnic jsou po technické stránce propracovaná, ale z analýzy provedené v předchozí kapitole vyplývá jeden zásadní limit vzhledem k cílům této práce. Tyto systémy předpokládají, že kolejiště je pevně zabudované a uživatel je ochoten investovat značný čas i finanční prostředky do instalace rozsáhlé senzorické infrastruktury.

Ačkoliv standard Digital Command Control (DCC) vyřešil problém nezávislého řízení vozidel, vrstva zpětné vazby pro zjišťování polohy zůstává závislá na invazivních elektromechanických konceptech.

Jak shrnuje Tabulka 3.1, existující řídicí aplikace jsou vnitřně svázány s tímto hardwarovým modelem. Zpracovávají výhradně diskrétní události, typicky informaci o tom, že lokomotiva vjela do elektricky izolovaného bloku.

Tabulka 3.1: Kritické porovnání vlastností existujících softwarových řešení

Řešení	**Typ**	**Zpětná vazba**	**Omezení pro tuto práci**
TrainController	Komerční	Fyzické bloky a senzory	Vysoká závislost na HW infrastruktuře
iTrain	Komerční	Fyzické bloky a senzory	Neumožňuje nativní vizuální sledování
Win-Digipet	Komerční	Fyzické bloky a senzory	Nákladná pořizovací cena systému
JMRI	Otevřená	Senzory, panely	Chybí modul pro plynulou lokalizaci
Rocrail	Otevřená	Fyzické bloky, trasy	Poloha je pouze logicky odvozována

Na základě tohoto zhodnocení lze konstatovat, že na trhu chybí dostupné řešení, které by dokázalo nahradit komplexní sítě hardwarových senzorů a poskytovalo by spojitou informaci o přesných souřadnicích vozidla na trati. V důsledku tohoto nedostatku je precizní zastavení vlaku u nástupiště v komerčních systémech často řešeno nepřímo pomocí kalibrovaných rychlostních profilů.

3.2 Návrh koncepce a systémové požadavky

Cílem této práce je vytvořit systém, který se vůči výše popsanému stavu vymezuje a odstraňuje nutnost fyzických zásahů do kolejiště. Koncepce je založena na zachování standardu DCC pro odesílání povelů, avšak s kompletním nahrazením hardwarové zpětné vazby bezkontaktním optickým systémem.

Na základě tohoto cíle byly definovány následující systémové požadavky, které musí navrhované řešení splňovat (definované zadání systému):

- **Bezkontaktní lokalizace:** Systém musí být schopen sledovat polohu hnacích vozidel výhradně pomocí kamery, bez použití kolejových obvodů.
- **Spojitá detekce polohy:** Software musí pracovat s kontinuálními souřadnicemi a úhlem natočení vlaku, nikoliv pouze s diskrétními bloky.
- **Synchronizace přes fyzikální engine:** Aplikace musí vizuální data okamžitě vyhodnotit a autonomně odeslat korekční příkaz fyzickému modelu s minimální latencí.
- **Nízká pořizovací cena hardwaru:** Systém musí fungovat s běžně dostupnými open-source mikrokontroléry, které nahradí drahou komerční DCC centrálu.
- **Uživatelské rozhraní:** Součástí musí být grafická aplikace (digitální dvojče), která uživateli umožní stavbu trati a zobrazení reálné polohy vlaků v čase.

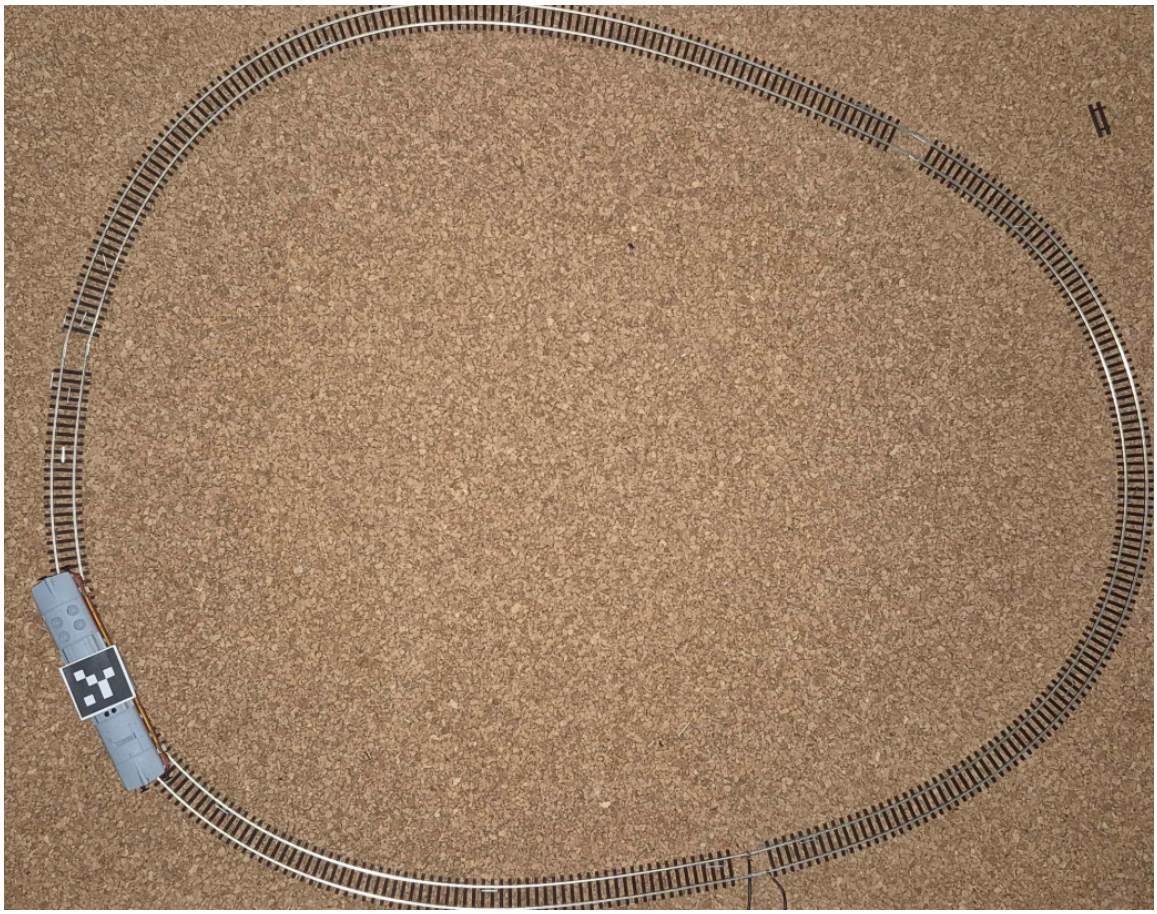
Úspěšné naplnění těchto požadavků vytvoří platformu, která dokáže lokomotivu plynule navigovat a zastavit na předem určeném místě pouze na základě obrazové zpětné vazby.

3.3 Požadovaná architektura a prostředky

Aby bylo dosaženo stanovených cílů, musí být systém koncepčně rozdělen na hardwarovou část, zabezpečující samotný pohyb vozidel, a softwarovou aplikaci, zodpovědnou za zpracování obrazu a řídicí logiku.

3.3.1 Fyzický model a řídicí řetězec

Základním prvkem infrastruktury se stane reálný model lokomotivy. Pro účely prokazatelnosti konceptu je navrženo využití standardního DC modelu, který bude dovybaven mobilním DCC dekodérem a na jehož střechu bude umístěn fiduciozní marker pro optické sledování. Příklad takové experimentální sestavy je zachycen na Obrázku 3.1.



Obrázek 3.1: Experimentální sestava lokomotivy s umístěným markerem, demonstrující minimální zásah do fyzického modelu

Toto uspořádání demonstruje, že systém lze nasadit na běžně dostupné modely bez nutnosti jejich složitých vnitřních hardwarových modifikací, čímž je splněn požadavek na minimální invazivnost řešení. Řídicí vrstva má být řešena pomocí mikrokontroléru (např. platformy Arduino) propojeného s počítačem. Mikrokontrolér ve spojení s výkonovým modulem převezme funkci plnohodnotné DCC centrály, což je přímá odpověď na požadavek nízké pořizovací ceny hardwaru.

3.3.2 Softwarové moduly a požadavky na nástroje

Splnění požadavků na počítačové vidění v reálném čase a interaktivní vizualizaci vyžaduje pečlivou volbu softwarových nástrojů. Celý systém je navržen pro implementaci v programovacím jazyce Python, který dominuje v oblasti zpracování obrazu. Návrh počítá s využitím následujících technologií:

- **Modul počítačového vidění:** Místo výpočetně náročných neuronových sítí je navrženo využití knihovny OpenCV a jejího modulu pro detekci ArUco markerů. Tento deterministický přístup by měl poskytnout absolutní souřadnice i úhel natočení s minimální zátěží procesoru.
- **Modul uživatelského rozhraní:** Pro vizualizaci digitálního dvojčete je požadován nativní desktopový framework (např. PySide6 / Qt). Nativní GUI umožní efektivní vykreslování vektorových prvků (Bézierových křivek) přímo nad obrazem z kamery.
- **Komunikační modul:** Přenos povelů z počítače do mikrokontroléru má být realizován pomocí sériového protokolu z open-source projektu DCC++ EX. Tento formát byl zvolen pro svou textovou povahu a snadnou asynchronní integraci.

Kapitola 4

Realizace a architektura systému

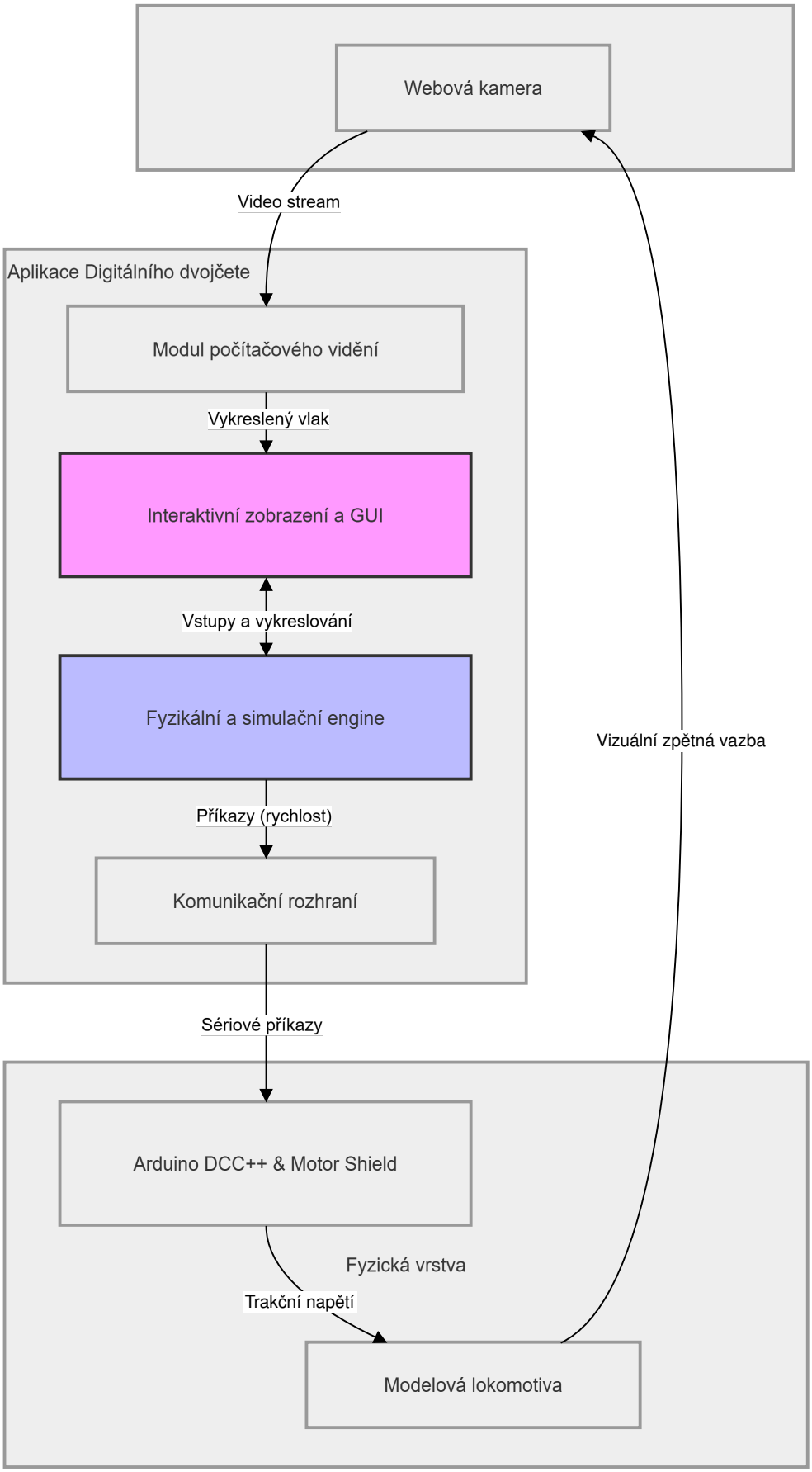
Tato kapitola představuje technické jádro předkládané práce. Přechází od teoretických konceptů a definovaných systémových požadavků k samotnému inženýrskému návrhu a softwarové implementaci digitálního dvojčete. Je kladen důraz na celkovou architekturu systému, logické rozdělení do funkčních bloků a princip jejich obousměrné komunikace.

Cílem této kapitoly je poskytnout ucelený pohled na to, jakým způsobem spolupracují nástroje počítačového vidění, vektorové grafiky, matematické simulace a řídicího hardwaru při vytváření funkční uzavřené regulační smyčky.

4.1 Celková architektura a návrh systému

Při vývoji komplexního kyber-fyzikálního systému, který musí v reálném čase obsluhovat grafické uživatelské rozhraní, provádět výpočetně náročné zpracování video streamu a paralelně udržovat stabilní sériovou komunikaci s externím hardwarem, je kriticky důležité zvolit robustní softwarovou architekturu. Pro tuto aplikaci byl na platformě Python zvolen modulární přístup, který je silně inspirován architektonickým vzorem MVC (Model-View-Controller) a je rozšířen o asynchronní zpracování událostí.

Celý systém je rozdělen na tři základní vrstvy, jejichž vzájemné propojení a datové toky ilustruje blokové schéma na Obrázku 4.1. Jedná se o vrstvu snímací, vrstvu fyzického řízení a centrální softwarovou aplikaci.



Obrázek 4.1: Celkové blokové schéma navržené architektury propojující fyzický hardware se softwarovými moduly aplikace.

4.1.1 Dekompozice softwarové vrstvy

Centrální aplikace běžící na osobním počítači představuje hlavní řídicí část systému. Aby byl zdrojový kód dlouhodobě udržitelný a nedocházelo k zamrzání uživatelského rozhraní, je aplikace vnitřně dekomponována do čtyř nezávislých funkčních bloků:

- **Interaktivní zobrazení (GUI):** Zastřešuje vizualizaci a uživatelskou interakci. Obsahuje editor vektorové mapy a prolíná logická data se surovým obrazem z kamery.
- **Modul počítačového vidění:** Izolovaný proces vyhrazený pro stahování snímků z kamery, jejich analýzu a lokalizaci lokomotiv pomocí fiduciozních značek.
- **Virtuální fyzikální engine:** Jádro systému simulující kinematiku vlaku. Udržuje v paměti logický stav trati, plánuje brzdné dráhy a vypočítává sub-pixelový posun modelů.
- **Hardwarové komunikační rozhraní:** Abstraktní vrstva zodpovědná za překlad vypočítaných logických rychlostí do formátu DCC a jejich odeslání do řídicího mikrokontroléru.

4.1.2 Asynchronní model a správa vláken

Základním kamenem funkčnosti této architektury je striktní rozdělení výpočetní zátěže do nezávislých pracovních vláken (multithreading). Monolitický návrh, kde by se v jedné nekonečné smyčce analyzoval obraz, počítala fyzika a odesílala data přes sériový port, by nevyhnutelně vedl k takzvanému uváznutí (bottleneck). Pokud by například zpracování složitého snímku z kamery trvalo déle, pozastavil by se výpočet fyziky a vlak by ztratil plynulost pohybu.

Z tohoto důvodu je systém postaven na architektuře řízené událostmi (Event-Driven Architecture). Hlavní vlákno operačního systému je vyhrazeno výhradně pro obsluhu uživatelského rozhraní a plynulé překreslování vektorového plátna. Ostatní moduly (vidění, fyzika, komunikace) operují ve vlastních procesech.

Aby nedocházelo ke kolizím při přístupu do sdílené paměti (tzv. souběh dat neboli race condition), je meziprocesová komunikace realizována pomocí front a asynchronních signálů, které nativně poskytuje použitý framework Qt. Pracovní vlákno počítačového vidění tak neupravuje souřadnice přímo v grafickém rozhraní, ale pouze do systému emituje datový paket s novými souřadnicemi. Ostatní moduly si tento paket asynchronně převezmou a zpracují ho ve chvíli, kdy jsou na to připraveny.

4.1.3 Datový tok v uzavřené regulační smyčce

Životní cyklus dat a vzájemná součinnost vrstev tvoří plně uzavřenou regulační smyčku, která zajišťuje nepřetržitou synchronizaci digitálního dvojčete s realitou.

Cyklus je inicializován v momentě, kdy uživatel v grafickém rozhraní stanoví cílovou rychlost vlaku. Tento logický povel přebírá *Fyzikální engine*, který na vektorovém modelu trati zahájí plynulou matematickou akceleraci vozidla. Vypočítané spojité hodnoty rychlosti jsou průběžně odesílány do *Komunikačního rozhraní*, jež je konvertuje do paketů a předává je fyzické centrále tvořené mikrokontrolérem Arduino a výkonovým štítem.

Centrála vpraví do kolejí střídavý DCC signál, na který okamžitě reaguje palubní dekodér a uvede fyzickou lokomotivu do pohybu. Tuto kinetickou změnu v reálném světě

následně zachytí webová kamera. *Modul počítačového vidění* ze snímku extrahuje polohová data a odešle je zpět do hlavní aplikace. Regulační smyčka se tím uzavírá, přičemž fyzikální engine může tyto získané reálné souřadnice využít ke korekci případných nepřesností ve své vlastní matematické simulaci.

4.2 Interaktivní zobrazení železnice

Uživatelské rozhraní (GUI) představuje nejviditelnější vrstvu celého systému. Neslouží pouze jako pasivní zobrazovač, ale plní roli komplexního řídicího pultu a editačního softwaru. Vzhledem k požadavku na interaktivní manipulaci s vektorovými objekty a jejich dynamické prolínání se živým obrazem z kamery byl pro vývoj tohoto modulu zvolen moderní aplikační framework PySide6 (oficiální vazba pro knihovnu Qt for Python).

4.2.1 Architektura grafické scény a pohledu

Ústředním technologickým prvkem vizualizačního modulu je implementace architektury *Graphics View Framework*. Historické nebo jednodušší dispečerské aplikace často využívají takzvaná rastrová plátna, kde je nutné při každé změně stavu (například při posunu vlaku o jeden pixel) přepočítat a překreslit celou matici bodů na obrazovce. Takový přístup je pro aplikace běžící v reálném čase vysoce neefektivní.

Naproti tomu zvolená architektura frameworku Qt striktně odděluje data od jejich prezentace a využívá objektově orientovaný přístup. V paměti počítače je udržována takzvaná *Scéna* (QGraphicsScene), která tvoří dvourozměrný virtuální prostor obsahující strukturovaný strom prvků (uzly, křivky, indikátory vlaků). Každý prvek je na scéně nezávislou entitou s vlastními lokálními souřadnicemi, z-indexem a metodami pro detekci kolizí či obsluhu kliknutí myší.

Uživateli je pak zprostředkován *Pohled* (QGraphicsView), což je v podstatě vizuální okno, přes které se na tuto scénu dívá. Vykreslovací jádro systému je vysoce optimalizované – pokud se virtuální vlak v rámci scény posune, systém neobnovuje celou obrazovku, ale překreslí pouze ohraničující obdélník (bounding box) samotného vlaku a jeho bezprostředního okolí. Tento princip výrazně snižuje zátěž procesoru a umožňuje aplikaci plynule vizualizovat dění na kolejišti i na méně výkonných zařízeních.

4.2.2 Topologický model vektorové tratě

Pro zajištění navigace vozidel musí systém disponovat jednoznačným matematickým popisem kolejiště. K tvorbě této topologie byl v rámci rozhraní vyvinut interaktivní editor pracující na principech teorie orientovaných grafů. Fyzická trať je abstrahována do dvou logických struktur:

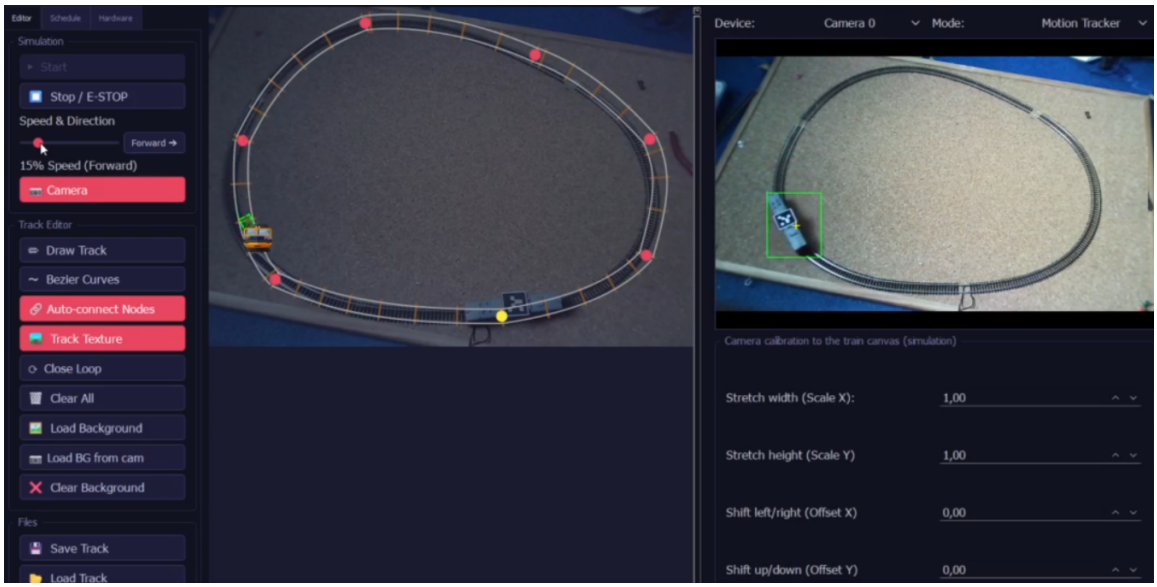
- **Uzly:** Reprezentují diskrétní orientační body. Typicky se jedná o středové body železničních stanic, zakončení slepých kolejí nebo spojovací uzly u rozvětvení (výhybek). Každý uzel disponuje unikátním identifikátorem a globálními souřadnicemi (x, y).
- **Hrany:** Představují povolené jízdní cesty mezi dvěma uzly. Vzhledem k charakteru reálného kolejiva, které se skládá z oblouků a přechodnic, nelze hrany reprezentovat pouhými přímkami. Systém proto využívá parametrické kubické Bézierovy křivky.

V grafickém editoru uživatel interaktivně buduje tyto struktury tím, že na plátno umisťuje uzly a propojuje je hranami. Následně má možnost pomocí zobrazených kotevních bodů

(tečen) tvarovat zakřivení tratě tak, aby co nejpřesněji lícovala s geometrií fyzických kolejí zachycených kamerou na pozadí.

4.2.3 Vizuální kompozice a prolínání realit

Jedním z největších úskalí při propojování fyzického světa s jeho digitálním modelem je přehlednost. Aby byl uživatel schopen jasně rozlišit fyzický vlak zachycený kamerou od jeho virtuální predikce, aplikuje systém třívrstvou kompozici obrazu (viz Obrázek 4.2):



Obrázek 4.2: Uživatelské rozhraní aplikace zobrazující třívrstvou kompozici: Snímek z kamery (podklad), vektorová trať (střední vrstva) a dynamické indikátory vozidel (vrchní vrstva).

1. **Kamerový podklad** (spodní vrstva) — Snímek z webové kamery slouží jako vizuální šablona při návrhu a kalibraci trati. Po dokončení kalibrace lze podklad skrýt nebo snížit jeho průhlednost.

2. **Vektorová trať** (střední vrstva) — Matematický graf kolejiště obsahující nakreslené křivky, uzly a popisky stanic.

3. **Indikátory vozidel** (vrchní vrstva) — Nejdůležitější vrstva digitálního dvojčete. Obsahuje pohybující se prvky. Vykresluje se zde virtuální vlak (generovaný fyzikálním enginem) a současně indikátor reálného vlaku (generovaný z dat počítačového vidění).

Správná funkce systému se projeví vizuálním splynutím obou indikátorů na obrazovce. Pokud virtuální model sleduje trajektorii fyzického vozidla s minimální odchylkou, uzavřená regulační smyčka digitálního dvojčete funguje korektně.

4.2.4 Prostorová kalibrace a transformace

Obraz pořízený kamerou (tzv. *Camera Space*) a vnitřní matematický souřadnicový systém vektorového plátna (*Canvas Space*) se v reálném nasazení téměř nikdy neshodují. Drobná

změna ohniskové vzdálenosti, naklonění stativu nebo optické zkreslení čočky způsobí, že vektorová mapa vytvořená včera dnes nebude lícovat se zapnutou kamerou.

Místo toho, aby uživatel musel složitě přerýsovávat celou vektorovou mapu, disponuje modul uživatelského rozhraní nástrojem pro lineární prostorovou kalibraci (afinní transformaci). Tato kalibrace funguje jako matematický můstek mezi kamerou a plátnem. Nástroj uživateli zpřístupňuje dvě sady kalibračních parametrů:

- **Měřítko (Scale S_x, S_y):** Parametry reprezentující zvětšení nebo zmenšení matice obrazu. Kompenzují případný ořez video streamu nebo změnu rozlišení.
- **Posun (Offset O_x, O_y):** Parametry určující absolutní posun počátku souřadnicové soustavy. Tyto hodnoty slouží k přesnému lícování po pohnutí s kamerou.

Všechna surová poziční data z kamery $P_{cam} = [x_s, y_s]$ jsou před vložením do vektorového plátna prohnána převodní rovnicí a přepočítána na finální prezentační souřadnice $P_{canvas} = [x_c, y_c]$:

$$x_c = x_s \cdot S_x + O_x \tag{4.1}$$

$$y_c = y_s \cdot S_y + O_y \tag{4.2}$$

Tento matematický model zajišťuje větší flexibilitu aplikace. Systém může běžet nad kamerou s rozlišením 720p stejně jako 4K, aniž by došlo k porušení logiky navigace vnitřního enginu.

4.3 Snímání scény a počítačové vidění

Zatímco fyzikální engine v hlavní aplikaci vypočítává ideální teoretickou polohu vozidel, modul počítačového vidění slouží jako nezávislá zpětnovazební smyčka. Jeho jediným úkolem je nepřetržitě zjišťovat skutečnou fyzickou polohu modelů na kolejišti a poskytovat tyto souřadnice hlavní aplikaci ke korekci. Modul je vystavěn nad open-source knihovnou OpenCV a běží ve zcela izolovaném asynchronním vlákně, aby výpočetně náročná analýza matice obrazu neblokovala překreslování uživatelského rozhraní.

4.3.1 Lokalizace pomocí fiduciozních značek

Sledování generického trojrozměrného objektu (modelu lokomotivy) vůči členitému a barevně komplexnímu pozadí modelového kolejiště představuje složitou úlohu, která je vysoce náchylná k chybám vlivem stínů a změn okolního osvětlení. Jak bylo argumentováno v rešeršní části, prostá detekce barev se pro tento účel ukázala jako nedostatečně robustní. Systém proto využívá sledování pomocí fiduciozních značek standardu ArUco.

Zpracování každého zachyceného snímku z video streamu prochází striktní analytickou pipelinou (řetězcem úloh). Obraz je nejprve konvertován z barevného prostoru BGR do stupňů šedi (Grayscale), což snižuje objem zpracovávaných dat v operační paměti a urychluje následné výpočty. Následně je na obraz aplikováno adaptivní prahování pro vytvoření binární masky se zdůrazněnými hranami.

Algoritmus v masce detekuje uzavřené kontury a filtruje ty, které neodpovídají čtyřúhelníkům. Na nalezené kandidáty je aplikována perspektivní transformace (homografie), která kompenzuje prostorové zkreslení a narovná obraz značky pro čtení. Pokud dekódovaná

vnitřní matice markeru odpovídá hledanému identifikátoru, systém vypočítá geometrický střed značky a získá přesný vektor rotace vůči ose kamery s přesností na zlomky pixelu.

Obrázek 4.3: Výstup modulu počítačového vidění: identifikovaný ArUco marker s vyznačeným středovým bodem a hranami.

4.3.2 Alternativní metoda: Detekce pohybu (Motion Tracking)

Jako druhá, plnohodnotná možnost lokalizace vozidel byl do systému implementován modul pro sledování pohybu založený na algoritmu odečítání pozadí (Background Subtraction). Zvolen byl konkrétně model MOG2[24] (Mixture of Gaussians). Tento režim funguje zcela nezávisle na detekci ArUco značek a představuje užitečnou alternativu pro situace, kdy na modely nelze či není žádoucí fyzicky umístit žádné markery. Výhodou tohoto řešení je plná autonomie – systém nevyžaduje žádný počáteční zásah uživatele ani manuální výběr oblasti zájmu.

Zpracování obrazu prochází dedikovanou analytickou pipelinou. Po spuštění modulu proběhne nejprve krátká inicializační fáze (warm-up interval v délce 30 snímků), během které algoritmus sleduje statickou scénu kolejiště a adaptivně sestavuje matematický model pozadí. Po této kalibraci začne systém od každého nového snímku tento model pozadí odečítat, čímž vznikne binární maska obsahující pouze pixely s detekovaným pohybem. Pro odstranění vizuálního šumu jsou na masku aplikovány morfologické operace (otevření a zavření pomocí eliptického jádra).

Následně systém v této čisté masce detekuje souvislé kontury a odfiltruje ty, jejichž plošný obsah nedosahuje minimální prahové hodnoty (eliminace drobných stínů). Aby algoritmus udržel kontext a zabránil nežádoucímu přeskočení sledování na jiný pohybující se objekt (například na ruku uživatele zasahujícího do kolejiště), implementuje funkci prostorové paměti. Pokud se na scéně nachází vícero kandidátů na vozidlo, modul nebere nutně ten plošně největší, ale vypočítá geometrická těžiště (momenty) všech kontur a jednoznačně vybere tu, jejíž euklidovská vzdálenost od minulé známé polohy vlaku je nejmenší.

Jelikož samotná pohybující se kontura (na rozdíl od ArUco markeru) nenese nativní informaci o absolutní rotaci objektu, je úhel natočení vozidla aproximován dynamicky.

4.3.3 Řešení nespojitosti úhlové rotace

Zatímco matematická filtrace lineárních souřadnic je triviální, aplikace vzorce EMA na úhel natočení vlaku naráží na logický problém nespojitosti kruhu. Pokud se fyzický vlak otočí například z hodnoty 359° na hodnotu 1°, absolutní matematický rozdíl činí $-358°$. Standardní filtrační rovnice by v takovém případě začala indikátor otáčet přes celý kruh zpět, což by způsobilo masivní vizuální anomálii.

K řešení tohoto stavu byla do filtračního bloku implementována ochrana založená na modulární aritmetice. Algoritmus neustále normalizuje rozdíl mezi posledním vyhlazeným a nově přijatým úhlem do rozsahu -180° až 180°, čímž garantuje, že systém pro interpolaci vždy zvolí nejkratší možnou úhlovou cestu napříč počátkem.

4.4 Virtuální fyzikální engine

Zatímco GUI zajišťuje pasivní zobrazení, skutečné logické rozhodování a plánování tras probíhá uvnitř simulačního a fyzikálního enginu. Jedná se o dedikovaný softwarový modul, který vyhodnocuje stav nakreslené trati, aplikuje pravidla definovaná uživatelským zásahem a vypočítává přesnou kinematiku virtuálního vlaku. Jeho úkolem je zajistit plně plynulý fyzikální pohyb vektorového objektu tak, aby vygenerovaný akcelerační profil mohl být s vysokou přesností replikován fyzickým modelem kolejišti.

4.4.1 Diskrétní časová simulace (Fixed Time-Step)

Pro zajištění deterministické povahy fyzikálních výpočtů, které musí být naprosto nezávislé na výkonu procesoru uživatele nebo na obnovovací frekvenci jeho monitoru, využívá engine architekturu s fixním časovým krokem. Simulace neběží v nekonečném asynchronním běhu, ale je spouštěna precizním časovačem v pravidelných intervalech (tzv. tick). Pro potřeby této aplikace byl zvolen interval 40 milisekund (25 Hz). Toto rozlišení poskytuje dostatečnou granularitu pro plynulý výpočet kinetiky a zároveň negeneruje zbytečnou nadměrnou zátěž pro sériovou linku při odesílání dat do hardwaru.

4.4.2 Kinetika pohybu a prediktivní brzdění

V každém časovém kroku simulace je fyzikální stav vozidla definován třemi klíčovými proměnnými: aktuální rychlostí ($v_{current}$), cílovou rychlostí definovanou uživatelem ($v_{commanded}$) a interní bezpečnou rychlostí (v_{target}). Vektorová kinetika v systému nerespektuje okamžité skoky. Změní-li uživatel požadavek na rychlost z nuly na maximum, engine hodnoty aproximuje pomocí integrace konstantního zrychlení a nebo zpomalení d.

Nejkritičtějším mechanismem tohoto modulu je algoritmus prediktivního brzdění (Lookahead). Virtuální vlak se nepohybuje v prázdném prostoru, ale po hranách orientovaného grafu. V každém ze 25 kroků za sekundu prohledává engine topologii trasy před vlakem. Jakmile systém detekuje, že v navazujícím logickém uzlu je naplánováno zastavení nebo čeká nesprávně přehozená výhybka, musí autonomně převzít řízení.

Systém průběžně měří zbývající logickou vzdálenost k cílovému uzlu. Pokud tato vzdálenost klesne pod nastavenou mez brzdné dráhy, engine automaticky přepíše uživatelský

povel, přiřadí hodnotě cílové rychlosti nulu a začne model plynule zpomalovat předem definovaným krokem (decelerací), čímž zajistí bezpečné zastavení přesně v oblasti stanice.

4.4.3 Arc-length parametrizace a sub-pixelový posun

Při transformaci teoretické kinetiky do grafického zobrazení bylo nutné vyřešit zásadní problém s deformací rychlosti na Bézierových křivkách. Tyto křivky trpí matematickým nedostatkem známým jako problém nestejnoměrné parametrizace délky oblouku (Arc-length parameterization).

Parametr času t podél křivky nenarůstá rovnoměrně vůči fyzické délce. Pokud by se vlak posouval prostou iterací přes tento parametr, na rovných úsecích by vlak vizuálně zrychloval a v prudkých obloucích masivně zpomaloval, ačkoliv fyzikální rychlost enginu by zůstávala konstantní.

Řešení tohoto problému spočívá v oddělení logické vzdálenosti od geometrie křivky. Algoritmus aproximuje oblouk na sérii pevných bodů a mezi těmito body měří skutečnou euklidovskou vzdálenost (pomocí Pythagorovy věty). Následně systém zavádí takzvaný subpixelový akumulátor.

Protože vypočítaná dráha pro jeden simulační krok v délce 40 ms často představuje pouze zlomky pixelu, nelze vlak posunout přímo na další bod v matici. Rychlost se proto sčítá v akumulátoru. K fyzickému posunu objektu po plátně dojde až ve chvíli, kdy hodnota v akumulátoru převýší změřenou fyzickou vzdálenost ke zkoumanému bodu. Případný přebytek dráhy je vyřešen lineární interpolací polohy (Lerp), která eliminuje i to nejjemnější optické trhání na monitoru.

4.5 Hardwarové propojení a řízení motorů

Rozhraní mezi čistě matematickým světem aplikace a reálným fyzickým kolejištěm tvoří komunikační modul a hardwarová řídicí centrála. Změny stavu vygenerované fyzikálním enginem nemají význam, dokud nejsou úspěšně transformovány na elektrické napětí v kolejnicích.

Jak bylo definováno v návrhu, systém neovládá motory lokomotiv přímo pomocí spojitého napětí (analogově), ale využívá plně digitální standard DCC. Z osobního počítače putují logická data ve formě textových příkazů skrz sběrnici USB do mikrokontroléru Arduino (firmware DCC++ EX). Tento mikrokontrolér, osazený výkonovým obvodem (Motor Shield), překládá příkazy do střídavého PWM signálu a zesiluje je na úroveň potřebnou pro napájení trati (typicky 12 V pro modelovou velikost H0).

4.5.1 Kompenzace nelinearity motoru (Lookup Table)

Při počáteční integraci systému byl odhalen kritický nesoulad mezi simulací a realitou. Zatímco fyzikální engine počítá zrychlení jako absolutně lineární a geometricky přesné, fyzický stejnosměrný motor uvnitř lokomotivy se chová nelineárně. Stupeň jedna v DCC systému nepředstavuje 1 % celkové rychlosti.

Pokud by systém předával lineární povely z enginu přímo do kolejí, virtuální vlak by u nástupiště zastavil včas, ale fyzický model by se setrvačností projel daleko za hranici logické stanice. Pro synchronizaci těchto dvou entit bylo nevyhnutelné zařadit do řídicího řetězce kalibrační převodní tabulku (Lookup Table - LUT).

Tato tabulka softwarově kompenzuje fyzikální nedokonalosti reálného hardwaru. Mapuje spojitou, teoretickou rychlost enginu (v pixelech na časový krok) na konkrétní diskrétní kroky DCC dekodéru v rozsahu 0 až 126. Pokud fyzikální engine spočítá, že aktuální rychlost je 70%, LUT algoritmus využije interpolaci k výpočtu přesného odpovídajícího stupně pro konkrétní lokomotivu (např. krok 45), aby vizuální indikátor na kameře přesně odpovídal stavu simulace.

4.5.2 Bezpečnostní rutiny a reverzace směru

Posledním zásadním mechanismem komunikačního modulu je softwarová ochrana trakčního ústrojí vozidel. Ačkoliv moderní DCC dekodéry disponují vnitřní logikou pro plynulé brzdění a rozjezd, v kontextu digitálního dvojčete je nezbytné tuto ochranu vynutit již na úrovni simulačního enginu. Pokud by se směr jízdy v DCC paketu změnil okamžitě, došlo by k desynchronizaci virtuálního modelu s reálným strojem a v případě starších hardwarových komponent by kinetická energie setrvačníků mohla poškodit ozubené převody modelu.

Aby k této situaci nedošlo, obsahuje systém ochranu průchodu nulou (Zero-Crossing). Pakliže je vydán povel ke změně směru, fyzikální engine neprovede okamžitou reverzaci, ale zahájí plynulou matematickou deceleraci. Rychlost je v každém časovém kroku simulace (40 ms) snižována, dokud nedosáhne nuly. Teprve v tomto momentě, kdy je zaručeno, že virtuální i fyzický model stojí, povolí systém odeslání paketu s opačnou polaritou a zahájí opětovnou akceleraci. Tento přístup garantuje dokonalé vizuální splynutí obou entit a zároveň slouží jako redundantní pojistka pro ochranu mechanických částí lokomotivy.

Kapitola 5

Testování a vyhodnocení systému

Tato kapitola se věnuje závěrečnému zhodnocení implementovaného digitálního dvojčete. Cílem je ověřit, zda výsledný systém splňuje všechny předem definované požadavky ze zadání (viz Kapitola 3), a následně zhodnotit jeho stabilitu, výkonnost a uživatelskou přívětivost v reálném provozu.

5.1 Vyhodnocení systémových požadavků

Pro objektivní posouzení úspěšnosti předkládaného řešení byly dosažené výsledky konfrontovány s původní specifikací. Lze konstatovat, že všechny klíčové body zadání byly naplněny:

- **Bezkontaktní lokalizace:** *Splněno.* Systém nevyužívá žádné kolejové obvody ani bodové detektory. Poloha vozidel je spolehlivě získávána výhradně pomocí kamerového snímače a detekce ArUco markerů.
- **Spojitá detekce polohy:** *Splněno.* Aplikace úspěšně opustila koncept diskrétních bloků. Vlak je v systému reprezentován jako entita se spojitými prostorovými souřadnicemi a úhlem natočení.
- **Nízká pořizovací cena hardwaru:** *Splněno.* Komerční řídicí centrály byly úspěšně nahrazeny dostupnou platformou Arduino s obvodem Motor Shield a open-source firmwarem DCC++ EX, čímž došlo k radikálnímu snížení nákladů na infrastrukturu.
- **Uživatelské rozhraní:** *Splněno.* Byl vyvinut interaktivní grafický nástroj v PySide6, který umožňuje pohodlnou tvorbu vektorové trati a slouží jako vizuální digitální dvojče s možností překryvu živého videa.
- **Synchronizace přes fyzikální engine:** *Splněno.* Byla implementována uzavřená regulační smyčka. Engine autonomně řídí kinematiku (včetně prediktivního brzdění a odesílá DCC příkazy.

5.2 Výkonnostní a funkční testování

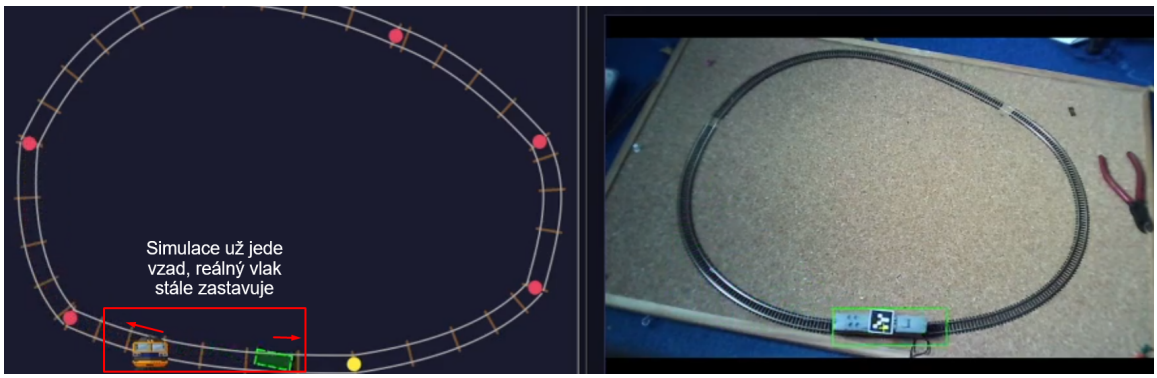
Následně byl systém podroben praktickým zkouškám na experimentálním kolejišti. Testování bylo zaměřeno na stabilitu asynchronního běhu, kalibraci hardwaru a spolehlivost kamerové detekce.

Během souvislého provozu systém neprokázal žádné úniky paměti (memory leaks) ani zablokování hlavního vlákna (zamrznutí GUI). Nastavený fixní krok fyzikálního enginu (40 milisekund) se ukázal jako optimální. Aplikace stíhala v reálném čase plynule překreslovat indikátory na plátně, zatímco fronta příkazů na sériové lince nebyla zahlcována nadbytečnými DCC pakety.

Pro správnou funkčnost regulační smyčky bylo nezbytné provést kalibraci převodní tabulky (LUT). Fyzický model lokomotivy byl testován při sérii různých diskrétních stupňů rychlosti DCC. Během těchto jízd modul počítačového vidění kontinuálně měřil skutečnou pixelovou rychlost modelu napříč plátnem. Získaná data posloužila k sestavení přesné nelineární křivky, která nyní spolehlivě mapuje logickou simulační rychlost enginu na adekvátní reálné chování elektromotoru.

Dalším kritickým testovacím scénářem bylo ověření chování systému při náhlé změně směru jízdy (tzv. reverzaci). Zpočátku matematický model předpokládal téměř okamžité zastavení vozidla. Praktické testy však ukázaly, že fyzický model lokomotivy vyžaduje vlivem mechanické setrvačnosti a interního zpoždění palubního dekodéru delší časový úsek k úplnému zastavení. Pokud simulace změnila směr dříve, než fyzický vlak skutečně stál, docházelo k rozsynchronizování digitálního dvojčete.

Na základě tohoto měření byla v simulaci upravena (prodloužena) ochranná prodleva pro průchod nulou (Zero-Crossing). Průběh tohoto ladění ilustruje Obrázek 5.1. Tímto inženýrským zásahem bylo dosaženo mnohem plynulejšího chování, kdy virtuální indikátor kopíruje setrvačnost reálného vozidla.



Obrázek 5.1: Grafické porovnání průběhu rychlosti při náhlé reverzaci směru. Původní teoretický předpoklad byl na základě měření upraven prodloužením zastavovací prodlevy, aby odpovídal setrvačnosti fyzického modelu.

Při testování modulu počítačového vidění byla zjištěna vysoká spolehlivost lokalizace při běžném osvětlení. Praktické testy nicméně odhalily, že extrémní světelné podmínky a zejména prudké odlesky na lesklém povrchu tištěného ArUco markeru mohou způsobovat krátkodobé výpadky v detekci, které musí asynchronní filtr vyhlazovat. Na základě tohoto zjištění je pro ostré nasazení systému doporučeno tisknout fiduciozní značky výhradně na matný materiál pohlcující světlo.

5.3 Uživatelské testování

Vzhledem k tomu, že systém je navržen i pro uživatele bez hlubokých znalostí z oblasti teorie grafů či počítačového vidění, bylo nezbytné ověřit celkovou srozumitelnost aplikace.

Proběhlo proto uživatelské testování s třemi laickými uživateli, kteří se na vývoji softwaru nijak nepodíleli.

Hlavním testovacím scénářem bylo ověření nástrojů pro tvorbu topologie. Uživatelé měli za úkol v editoru narýsovat funkční model kolejiště pomocí umisťování uzlů a propojování a tvarování Bézierových křivek. Cílem bylo zjistit, zda je pro ně celková koncepce ovládání intuitivní a zda jim vizuální logika aplikace dává smysl.

Testování naznačilo, že základní princip tvorby trati byl pro uživatele srozumitelný. Tvorba topologického grafu pomocí vizuálního natahování tečen byla hodnocena jako přirozená a nevyžadovala zdlouhavé vysvětlování. Z testování vyplynulo, že aplikace úspěšně abstrahuje složitou matematickou reprezentaci do přívětivého uživatelského rozhraní, ve kterém se běžný uživatel dokáže rychle orientovat.

Kapitola 6

Závěr

Cílem této práce bylo navrhnout a vytvořit plně funkční systém digitálního dvojčete pro modelovou železnici, který dokáže řídit vlaky výhradně pomocí kamerové zpětné vazby bez použití fyzických senzorů v kolejišti. Mohu s potěšením konstatovat, že tento hlavní cíl i všechny dílčí body zadání byly úspěšně splněny.

V první fázi práce (shrnuté v Kapitolách 2 a 3) jsem analyzoval a vyhodnotil současné metody detekce polohy. Jak je v textu explicitně uvedeno, mým cílem nebylo vytvořit historickou encyklopedii všech existujících systémů, ale provést cílenou rešerši technologií bezprostředně souvisejících s mým řešením. Na základě této analýzy jsem jasně identifikoval hardwarové a cenové nevýhody tradičních kolejových obvodů a navrhl vlastní koncepci založenou na počítačovém vidění.

Následná realizace (Kapitola 4) propojila svět čistého softwaru a matematiky se světem fyzického hardwaru. Podařilo se mi naprogramovat uživatelské rozhraní pro tvorbu vektorových map trati, implementovat detekci obrazu pomocí ArUco značek, sledování pohybu a zajistit plynulou obousměrnou komunikaci s řídicím mikrokontrolérem. Zde jsem prakticky zjistil, že i ten nejlépe napsaný kód musí nakonec čelit nedokonalostem reality – ať už jde o stíny a odlesky v obraze, nebo o nelineární chování elektromotoru lokomotivy. Hledání těchto řešení mi přineslo velké zkušenosti v oblasti asynchronního programování a algoritmů zpracování obrazu.

Výsledkem je stabilní aplikace, která dokáže v reálném čase sledovat vlak a přesně mapovat jeho polohu na virtuální plátno (viz vyhodnocení v Kapitole 5). Namísto složitého zapojování desítek detektorů a tahání metrů kabelů dnes k plné automatizaci provozu stačí jedna webová kamera, běžný napájecí zdroj a elektronika Arduino. Samotná centrální aplikace, prokázala během praktických experimentů vysokou stabilitu.

Ačkoliv své původní cíle považuji za splněné, nabízí se několik atraktivních směrů pro další rozvoj aplikace. Z hlediska nových funkcí a maximálního uživatelského komfortu bych systém rád rozšířil o algoritmy pro plně automatickou detekci kolejiva z obrazu. To by umožnilo samočinné vygenerování a překreslení topologie trati na vektorové plátno zcela bez nutnosti ručního rýsování. Dále by bylo velmi přínosné doplnit aplikaci o komunikační modul pro automatické ovládání výhybek a návěstidel, čímž by se otevřela cesta ke tvorbě pokročilých dispečerských scénářů a bezkoliznímu provozu více lokomotiv současně. Vzhledem k tomu, že je celá architektura programu navržena striktně modulárně, je systém na integraci těchto i dalších inovativních prvků plně připraven.

Literatura

[1] AJMAL, A. et al. A Comparison of RGB and HSV Colour Spaces for Visual Attention Models. *International Conference on Image and Vision Computing New Zealand*. Auckland, New Zealand: IEEE, 2018. Dostupné z: https://www.researchgate.net/publication/328759615_A_Comparison_of_RGB_and_HSV_Colour_Spaces_for_Visual_Attention_Models.

[2] AMES, S. Recommended Practices for Digital Command Control. *NMRA Bulletin*. Soddy Daisy, TN: National Model Railroad Association, January 1994, s. 32.

[3] ArduinoModules.info. *KY-024 Linear Magnetic Hall Module [online obrázek]*. 2026. Dostupné z: https://arduinomodules.info/ky-024-linear-magnetic-hall-module/. [cit. 2026-05-11].

[4] BERKHOUT, X. *ITrain 5: Manual*. 2021. Dostupné z: https://berros.eu/download/5-0/iTrain%205%20manual.pdf. [cit. 2026-05-07].

[5] BRADSKI, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000, sv. 25, č. 11, s. 120–125.

[6] Comus Group. *Reed Switches [online obrázek]*. 2026. Dostupné z: https://www.directindustry.com/prod/comus-group/product-37554-1064401.html. [cit. 2026-05-11].

[7] Freiwald Software. *RAILROAD & CO. TrainController Users Guide*. 2024. Dostupné z: https://www.freiwald.com/software/UsersGuide.pdf. [cit. 2026-05-02].

[8] Freiwald Software. *TrainController - Ukázka rozhraní dispečera [online obrázek]*. 2026. Dostupné z: http://deer.he.net/~freiwald/images/layout/driburg/2.jpg. [cit. 2026-05-11].

[9] GARRIDO JURADO, S.; MUÑOZ SALINAS, R.; MADRID CUEVAS, F. J. a MARÍN JIMÉNEZ, M. J. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 2014, sv. 47, č. 6, s. 2280–2292.

[10] GRIEVES, M. Origins of the Digital Twin Concept. *Florida Institute of Technology*, 2016.

[11] JMRI Project. *Java Model Railroad Interface*. 2024. Dostupné z: https://www.jmri.org/.

[12] KRITZINGER, W.; KARNER, M.; TRAAR, G.; HENJES, J. a SIHN, W. Digital Twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine*. Elsevier, 2018, sv. 51, č. 11, s. 1016–1022.

[13] LENZ ELEKTRONIK GMBH. *1979-2019: 40 years of Lenz Elektronik*. 2019. Dostupné z: https://www.lenz-elektronik.de/en/Lenz/1979-2019-40-years-of-Lenz-Elektronik/.

[14] MTOWE, D. P.; LONG, L. a KIM, D. M. Low-Latency Edge-Enabled Digital Twin System for Multi-Robot Collision Avoidance and Remote Control. *Sensors*. MDPI, 2025, sv. 25, č. 15, s. 4666.

[15] NATIONAL MODEL RAILROAD ASSOCIATION. *S-9.2 Communications Standards for DCC*. 2004. Dostupné z: https://www.nmra.org/sites/default/files/s-92-2004-07.pdf.

[16] NATIONAL MODEL RAILROAD ASSOCIATION. *S-9 ELECTRICAL*. Soddy Daisy, TN: National Model Railroad Association, 2025. Dostupné z: https://www.nmra.org/sites/default/files/standards/sandrp/Elec/S/s-9_electrical.pdf.

[17] NATIONAL MODEL RAILROAD ASSOCIATION. *S-9.1 Electrical Standards for Digital Command Control*. 2025. Dostupné z: https://www.nmra.org/sites/default/files/standards/sandrp/DCC/S/s-9.1_electrical_standards_for_digital_command_control.pdf.

[18] OZEKI SYSTEMS. *How to setup an optical gate on Arduino Nano [online obrázek]*. 2026. Dostupné z: https://ozeki.hu/p_3027-how-to-setup-an-optical-gate-on-arduino-nano.html. [cit. 2026-05-11].

[19] PETERLIN, P. *Win-Digipet 2021 Premium Edition: Manual*. 2021. Dostupné z: https://www.windigipet.de/files/Manual_2021.pdf. [cit. 2026-05-07].

[20] ROCRAIL. *Rocrail - Innovative Model Railroad Control System*. 2024. Dostupné z: https://wiki.rocrail.net/.

[21] TAO, F.; ZHANG, H.; LIU, A. a NEE, A. Y. C. Digital Twin in Industry: State-of-the-Art. *IEEE Transactions on Industrial Informatics*, 2019, sv. 15, č. 4, s. 2405–2415.

[22] TEXAS INSTRUMENTS. *Isolated Current Sense Reference Design for Traction Inverter*. Dallas, TX, 2018. Dostupné z: https://www.ti.com/lit/pdf/tidudr5. [cit. 2026-05-05].

[23] ZHOU, B. et al. Integrated magnetic Hall effect sensors. *RSC Advances*, 2022. Dostupné z: https://pmc.ncbi.nlm.nih.gov/articles/PMC8695063/.

[24] ZIVKOVIC, Z. Improved adaptive Gaussian mixture model for background subtraction. *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* Cambridge, UK: IEEE, 2004, sv. 2, s. 28–31. Dostupné z: https://www.researchgate.net/publication/4090386_Improved_Adaptive_Gaussian_Mixture_Model_for_Background_Subtraction.