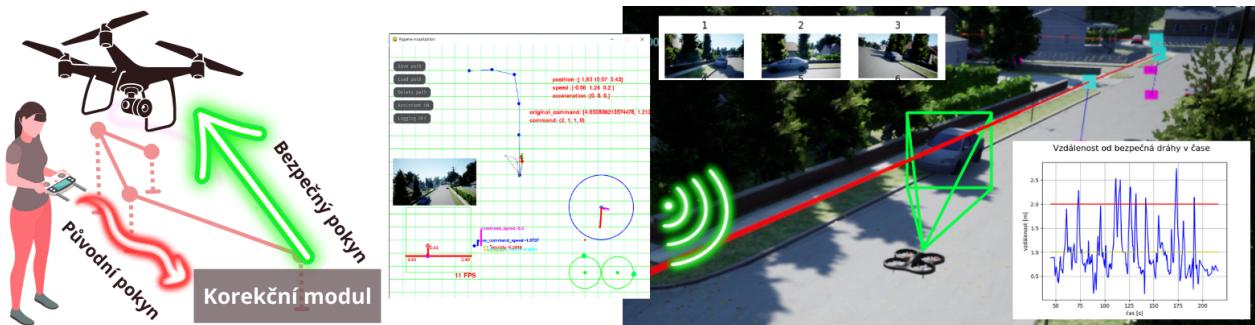


# Bezpečný průzkum dronem s využitím chytrého pohybu po trajektoriích

Adam Ferencz\*



## Abstrakt

Piloti dronu často musí hlídat při plnění průzkumné mise více věcí zároveň. Pilot něco pozoruje, sleduje obraz, natočení dronu, rychlosť a překážky. Cílem této práce je vytvořit aplikaci, která pilotovi ulehčí hlídání bezpečné polohy dronu.

Článek prezentuje novou metodu, která reviduje povely pilota tak, aby se dron držel "bezpečně blízko" předem definované dráze letu. Pilot se tak může více věnovat plnění cílů mise, než přímému a bezpečnému řízení dronu. Navržená metoda byla testována v aplikaci, která umožňuje definovat bezpečnou trajektorii dronu a pro testovací lety využívá integraci simulátoru.

Výstupem jsou testy uživatelské aplikace v simulátoru AirSim, kde uživatel dělá nejdříve misi bez zapnuté korekce a poté s ní. V uživatelských testech, kde byl cíl splnit průzkumnou misi, piloti při letu bez korekce strávili 55% mimo definovanou bezpečnou oblast a při letu s asistentem takto strávili pouze 5%. Průměrné vychýlení zavedení od ideální trajektorie kleslo z 2,29 metrů na 0,92 metru (bezpečná vzdálenost byla při testech do 2 metrů).

**Klíčová slova:** drone — semiautonomus — simulation — AirSim — python

**Přiložené materiály:** Pilotní uživatelský test 1 — Tvorba dráhy — Pilotní uživatelský test 2

\*xferen05@stud.fit.vutbr.cz, Faculty of Information Technology, Brno University of Technology

## 1. Introduction

V poslední době se zvedá zájem o využití dronů v různých komerčních sektorech. Příkladem je filmařství, bezpečnostní dohled nebo prodej realit. Jedním z případů užití dronu je průzkum větší oblasti. Příkladem může být kontrola technického stavu lanovky na horách. Uživatel potřebuje pravidelně kontrolovat stav lanovky a při letu se musí soustředit na pořizované video či fotografie.

Motivací této práce je navrhnout a implementovat způsob jak pilota zbavit části zátěže pomocí asisto-

váního letu po bezpečné dráze. Pilot musí například vždy vědět, jak je dron natočený, protože pokud je natočený obráceně, má pilot v podstatě prohozené ovládání. Kromě letu samotného musí zkoumat přenášený obraz a například podávat hlášení o tom, co vidí. Prostředí, v němž se pohybuje může být v oblasti s lidmi, od kterých si pilot musí držet bezpečnou vzdálenost.

Existují aplikační řešení nabízející funkce s různými letovými módy pro usnadnění letu v různých situacích. Dále existují aplikace využívající rozšířenou

realitu pro odlehčení zátěže pilota. Ty kladou důraz zejména na orientaci v prostoru, čímž nabízí pomoc při vícedronových misích.

Pilot celou dobu řídí dron normálně, ale pokud se začne více soustředit na pozorování obrazu a vychýlí se z předem definované trasy, asistent ho vráti zpět do bezpečné zóny.

**Cílem práce je návrh metody pro korekci řídících povelů pilota** tak, aby se dron držel na předem definované dráze letu. Součástí řešení je:

- reprezentace bezpečné dráhy a celého virtuálního prostoru
- návrh GUI jak pro tvorbu a editaci dráhy, tak i pro samotný bezpečný let a ovládání dronu
- návrh vhodné architektury aplikace, aby bylo možné ji napojit na různé simulátory i reálný dron
- vytvoření aplikace implementující korekci řídících povelů a provedení uživatelského testování

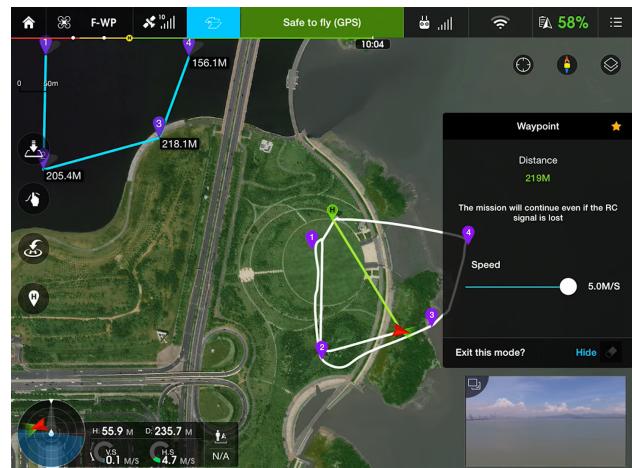
Řešením je aplikace, která modeluje prostředí s bezpečnou dráhou a komunikuje s dronem (simulovalným, nebo reálným). V aplikaci je modul filtrující příkazy od uživatele. Ty jsou nejdříve zkонтrolovány, případně poupraveny a až poté odeslány dronu.

Díky vzniklé aplikaci je možné volně ovládat dron na průzkumné misi a nemuset si tolik hlídat svou polohu. V rámci řešení bylo navrhnuo GUI, ve kterém je možné editovat dráhu a při letu zobrazuje pilotovi návodné vizuální prvky.

## 2. Existující řešení

Existují aplikace pro létání s drony, které se různými způsoby snaží pilotovi ulehčit plnění cíle mise. Při takové misi pilot často zápasí s obtížnou orientací v prostoru. Není pro něj lehké správně odhadnout rychlosť a vzdálenost.

**DJI GO** je oficiální mobilní aplikace pro drony aktuálně největší společnosti v tomto odvětví. Aplikace se dá považovat za state of the art v designu GUI i v letových módech, které nabízí. Kromě základního letového módu, kde "dopředu" znamená let ve směru čela dronu, jsou zde i další tzv. intelligentní letové módy. **Waypoint mód** (viz obrázek 1) umožňuje automatický let po předdefinované dráze. Výhodou je, že si pilot může dráhu připravit předem. Další módy se už soustředí na různé situace, ve kterých se vždy hodí jiné ovládání. **Course Lock**, kde "dopředu" znamená na sever, je vhodný při orientaci podle mapy, oproti klasickému módu, který je dobrý při orientaci podle videa z kamery. **Home lock**, kde "dopředu" znamená pryč od místa startu.



Obrázek 1. Waypoint mód - dron se pohybuje po předem definované misi, případně vykonává předem definované akce.

Aplikace **DroCo** byla vyvinuta pro řízení vícedronových misí. Drony se registrují do centrálního systému DroCo a jsou pak navzájem viditelné. Na obrázku 2 je vidět pohled na dron v režimu třetí osoby. Budovy zobrazované v aplikaci jsou korespondují s budovami reálného světa. Zároveň zde vidí vše podstatné jako je mapa, video stream z dronu a také další drony, se nimiž by měl kooperovat. Je také možné do rozšířeného světa přidávat různé interaktivní prvky misí nebo zóny. Právě tyto prvky pilotovi pomáhají s orientací a bezpečností. Aplikace může komunikovat s ROS drony, DJI drony a nebo se simulovanými drony přes jejich vlastní protokol [1, 2, 3].



Obrázek 2. Ukázka z aplikace Droco [2], dron umístěný do virtuálního prostředí s promítnutým záběrem z jeho kamery.

U existujících řešení chybí mezistupeň mezi automatickým co nejpřesnějším letem po předdefinované trajektorii a letem volným. Výhodou takového řešení je volnost letu pro pilota, kdy pilot letící v bezpečné zóně ani nepozná, že mu do řízení algoritmus zasahuje. Zároveň se ale pilot může spolehnout, že kdyby z nepozornosti udělal špatný řídící pokyn, tak bude zachycen a upraven.

### 3. Návrh řešení

Uživatel ví, jakou dráhu bude létat. Má ji definovanou z mapy, nebo ji extrahuje z předchozího letu. Jeho nejtěžší chvíle při plnění mise nastává, když má sledovat více věcí najednou. Aplikace musí pilotovi ulehčit zatížení tak, aby se mohl méně soustředit na let a více na průzkum, natočení kamery, gymbalu atd. Je třeba vytvořit aplikaci, která umožní zpracovat vstupy od pilota (z jeho ovladače) dříve, než jsou odeslány dronu. Při zpracování těchto vstupů musí mít také aplikace povědomí o tom, v jakém stavu se dron nachází a jaká je bezpečná dráha.

Zároveň je nutné dodat, že není cílem zasahovat přímo do řízení dronu (PID kontroller, který mapuje pokyn na jednotlivé motory). To bude bráno jako uzavřené řešení. Korekční metoda bude revidovat přímo pokyny pilota. Dron bude považován za hmotný bod, kde pokyn pilota znamená přímo zrychlení, které se aplikuje.

Navržená metoda korekce je v aplikaci používána jako modul, jehož vstupem jsou pokyny uživatele z ovladače, stav dronu a definovaná dráha. Výstupem této metody je bezpečný pokyn, který je následně dronu odeslán.

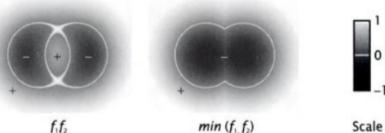
Pilot předem ví, kam letí a proto je schopný popsat jaké území je bezpečné a jaké ne. Aby mohl vytvořit avizovanou **bezpečnou dráhu**, je potřeba **navrhnut vhodnou strukturu**, která ji bude uchovávat.

Bezpečná zóna  $S$  je definována jako ideální trajektorie dronu (množina úseček definována řídícími body) a maximální povolená vzdálenost od této trajektorie.

$$S = [p_0, p_1, p_2, p_3 \dots p_n], \quad (1)$$

kde  $p_0, p_1, p_2, p_3 \dots p_n$  jsou jednotlivé řídící body. Každý bod obsahuje následující informace:

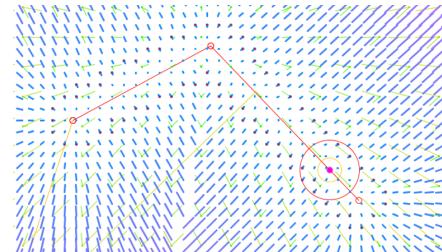
- latitude - zeměpisná šířka
- longitude - zeměpisná délka
- altitude - nadmořská výška



Obrázek 3. Implicitní povrch zjednodušený do 2D [4]

Korekční metoda se na model světa dívá tak, že v každém místě dokáže určit míru nebezpečí pro dron a zároveň určit vektor směřující k nejbližší bezpečné poloze. Toto je inspirované výpočtem implicitních povrchů (implicit surfaces), které pro modelování tělesa využívají vzdálenostní funkci od jeho kostry viz obrázek 3.

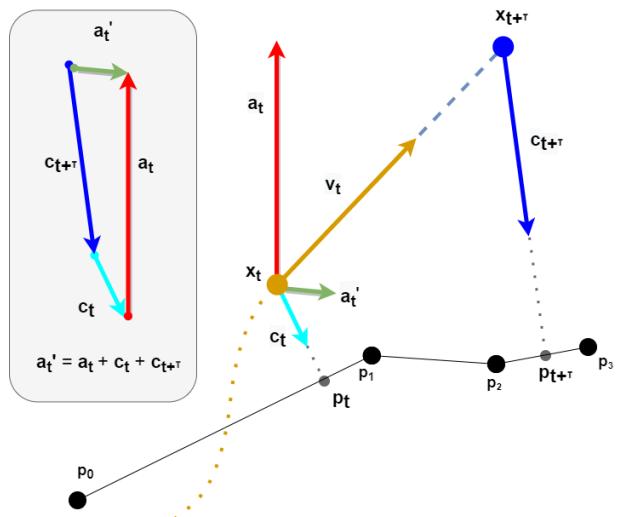
V případě dronu implicitní povrch znamená neproniknutelnou hranici. Zásadní pro výpočet korekce je vektorové pole (viz obrázek 4) vytvořené díky vzdálenostní funkci.



Obrázek 4. 2D vizualizace sil ovlivňujících výsledné řízení dronu na základě jeho polohy vůči bezpečné trajektorii. Obrázek z experimentálního programu napsaném v jazyce Processing.

Obecně lze popsát návrh metody následovně. Pilot dává povely dronu, které určují zrychlení daným směrem. Korekční metoda na základě polohy dronu vůči řídící struktuře (bezpečná dráha dronu definující bezpečnou zónu) a rychlosti dronu upraví řídící povely pilota tak, aby dron neopustil bezpečnou zónu.

Existuje bod v prostoru s pozicí  $x_t$ . Ten představuje ovládaný dron. Letí rychlosť  $v_t$  a působí na něj zrychlení  $a_t$ . Zrychlení  $a_t'$  je pokyn pilota.



Obrázek 5. Korekce s ohledem na rychlosť dronu: predikce polohy v čase  $t + \tau$ , kde  $\tau > \Delta t$ ; grafický vektorový součet pro výpočet bezpečného pokynu (váhové koeficienty zanedbány).

Polohu a rychlosť dronu vyjádříme rovnicemi pro změnu polohy hmotného bodu:

$$\mathbf{x}_{t+1} = \mathbf{t}_k + \mathbf{v}_t \cdot \Delta t + \frac{1}{2} \mathbf{a}_t \cdot \Delta t^2 \quad (2)$$

$$\mathbf{v}_{t+1} = \mathbf{v}_t + \mathbf{a}_t \cdot \Delta t \quad (3)$$

, kde  $\Delta t$  je uplynulý čas od poslední změny polohy.

$$\mathbf{c}_t = r(L_2(x_t, p_t)) \cdot (x_t - p_t) \quad (4)$$

$$r(d) = \frac{(d-2)^3}{10} + 0.5, \quad (5)$$

kde  $p_t$  je nejbližší bod k  $x_t$  ležící na přímce nejbližšího úseku  $S$ ,  $L_2$  je Eukleidovská vzdálenost,  $r$  je normalizační funkce a  $\mathbf{c}_t$  je korekční vektor.

$\mathbf{c}_{t+\tau}$  je korekční síla pro predikovanou polohu, která je počítána stejně (viz výše), ale pro predikovanou polohu  $x_{t+\tau}$ .

Výsledné zrychlení (upravený povel pilota) je pak:

$$\mathbf{a}_t = w_0 \cdot \mathbf{a}_t + w_1 \cdot \mathbf{c}_t + w_2 \cdot \mathbf{c}_{t+\tau}, \quad (6)$$

kde  $w_0$ ,  $w_1$  a  $w_2$  jsou váhy a  $\mathbf{c}_t$  je korekční síla pro aktuální polohu,  $\mathbf{c}_{t+\tau}$  je korekční síla pro predikovanou polohu a  $\mathbf{a}_t$  je výsledné zrychlení.

Grafické sčítání těchto vektorů je zobrazeno na obrázku 5, kde je vidět zásadní podíl korekčního vektoru  $\mathbf{c}_{t+\tau}$  na výsledku korekce.

## 4. Implementace aplikace

Navržená metoda si klade za cíl vhodně filtrovat pokyny pilota, a tím mu pomoci udržet se v bezpečné oblasti při plnění mise. Pro ověření metody jsem navrhl architekturu systému umožňující experimentálně testovat metodu v simulovaném i v reálném prostředí. Základní bloky systému jsou zobrazeny na obrázku 8. Hlavní aplikace, která obsahuje:

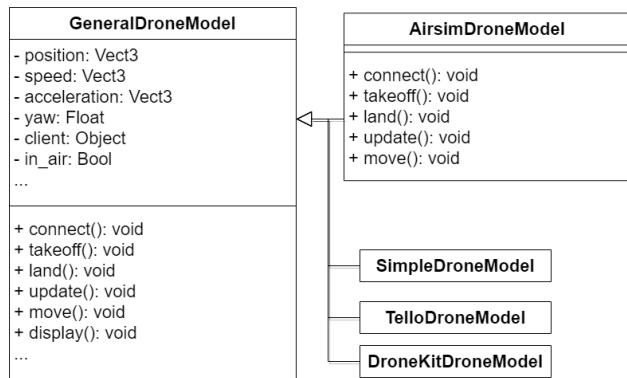
- Adapter ovladače dronu
- Korekční modul
- Model prostředí

Hlavní aplikace komunikuje s blokem Dron. Komunikace probíhá za účelem ovládání a získávání informací o stavu dronu. Dron může být reálný, nebo simulovaný. V případě simulovaného dronu se pak jedná o samostatně běžící program. Komunikaci s blokem Dron, realizuje Adapter ovladače dronu. Ten je implementován jako třída (jiná vždy pro jiný protokol/API) rozšiřující základní abstraktní třídu GeneralDroneModel tak, jak je znázorněno na obrázku 6.

Adapter ovladače dronu si kromě odesílání povelů také uchovává informace o dronu.

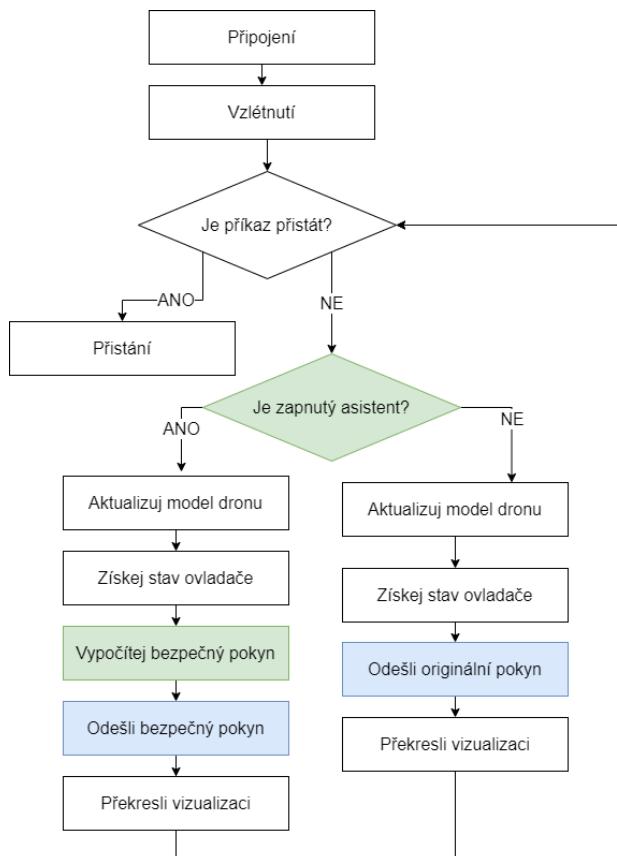
Hlavní aplikace běží ve smyčce a vykonává následující kroky zobrazené na obrázku 7:

1. Aktualizuje si v informace o dronu (poloha, rychlost, orientace).



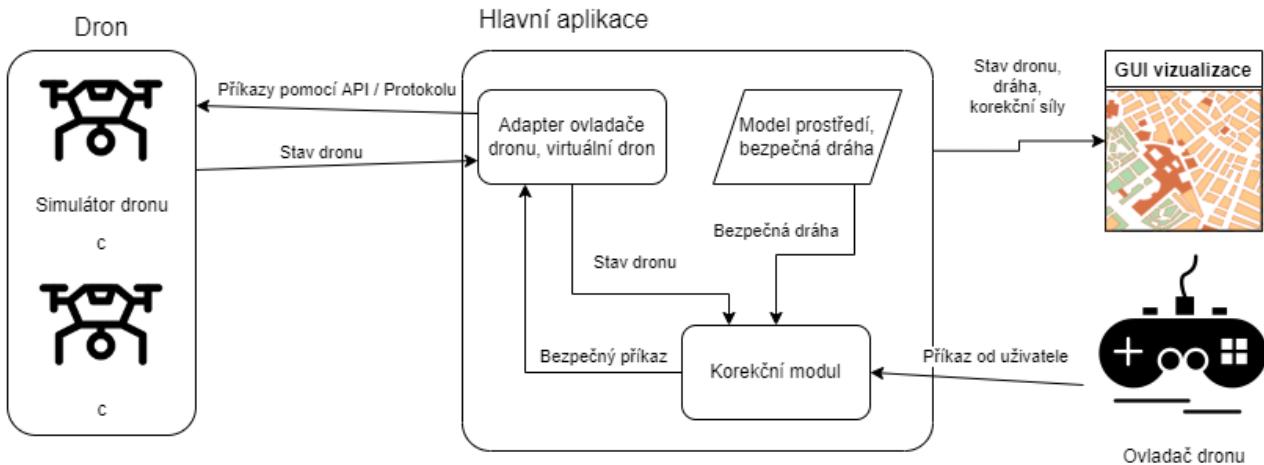
**Obrázek 6.** Třídy modelů dronu, kde konkrétní třída funguje jako adaptér pro komunikaci s externím simulátorem/dronem. Jednotlivé názvy tříd jsou odvozené od různých typů simulátorů nebo dronu použitých v projektu.

2. Zpracuje stav ovladače, tedy vstupy od uživatele.
3. Pokud je zapnutý asistent, použije výše zmíněné aktuální hodnoty pro vykonání korekčního algoritmu.
4. Adapter posílá pokyn dronu vhodným protokolem.
5. Je vykresleno GUI aplikace.



**Obrázek 7.** Vývojový diagram hlavního cyklu programu.

Po těchto krocích se vše opakuje, dokud není aplikace ukončena, nebo dron nedostane pokyn přistát.



Obrázek 8. Návrh architektury aplikace

Ovládání dronu je velmi specifické. Pro testování je potřeba, aby způsob ovládání vycházel ze standardních módů řízení a rozložení pokynů na ovladači. Ovladač samotný je také zásadní, protože řízení samotou klávesnicí by nepřineslo dostatečnou přesnost. Existuje více možností jak rozložit pokyny na ovladač. V tomto případě byl zvolen ”Evropský režim”, který se často používá u pilotů dronů, ale například u letadel není obvyklý<sup>1</sup>. Tento stejný mód je použit také u originální aplikace pro Tello dron<sup>2</sup>, která byla inspirací.

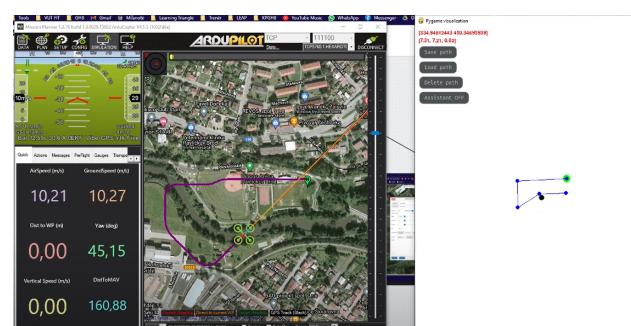
Bylo potřeba zvolit vhodný jazyk, který umožňuje experimentální implementaci představené metody a zároveň vytvoření interaktivního GUI s možností vizualizace. Pro testování musel být zvolen takový simulátor, do něhož je možné zasahovat zvenčí. Je třeba, aby měl dostatečně realistický vlastní PID řídící kontrolér. Z pohledu otestování aplikace na uživatelích je nutné, aby nabízel 3D prostředí a možnost získání záběru z kamery dronu. Účelem aplikace je totiž především být schopný bezpečně navigovat dron, který se vzdálí z dohledu (tedy kamera je hlavní orientační prvek) a efektivně s ním pořídit snímky nebo záběry.

Celá aplikace je programovaná v jazyce python. Knihovna tvořící uživatelské rozhraní je PyGame a její nadstavba PyGameGui. Díky této knihovně je možné připojit Xbox ovladač, který se rozložením shoduje s tradičními ovladači dronů.

Z více možností simulátorů, na něž lze aplikaci napojit, je primární simulátor AirSim [5]. Umožňuje totiž umístění simulovaného dronu do virtuálního světa a poskytuje obraz kamery dronu. Simulátor běží na Unreal enginu s API pro python. Jeho fyzikální model dronu (tedy i simulátor jeho kontroleru) je dostačující. Prostředí, ve kterém se s dronom léta, je zobrazeno na

obrázku 12.

Mimo primární simulátor AirSim lze řešení napojit na simulátor typu SITL (Software In The Loop) od ArduPilot. Ten realisticky modeluje kontrolér dronu a pomocí knihovny DroneKit je možné ho napojit. Simulátor typu SITL v kombinaci s vizualizačním programem Mission planner (viz obrázek 9) byl použit k úvodnímu ladění algoritmu (viz obrázek 9). Aplikaci lze využít také s fyzickým dronem DJ Tello pomocí dostupného SDK<sup>3</sup>. Při testování se ukázalo, že Tello není dostatečně spolehlivé.



Obrázek 9. Testování se simulátorem SITL v Mission Planneru.

Pro vizualizaci a výpočty vzdáleností od dráhy je třeba transformovat polohové údaje mezi následujícími formami:

- GPS souřadnice
- relativní metry od centrálního bodu
- poloha ve 2D pohledu shora ve vizualizaci

V grafickém uživatelském rozhraní je potřeba vyvářet, editovat a spravovat mise (uložit, načíst, vymazat). Je potřeba, aby uživatel své mise mohl nahrávat a ukládat pro pozdější použití. Dále musí uživatelské prostředí zobrazovat všechny prvky potřebné k letu. Tedy konkrétně je třeba zobrazit polohu dronu i polohu

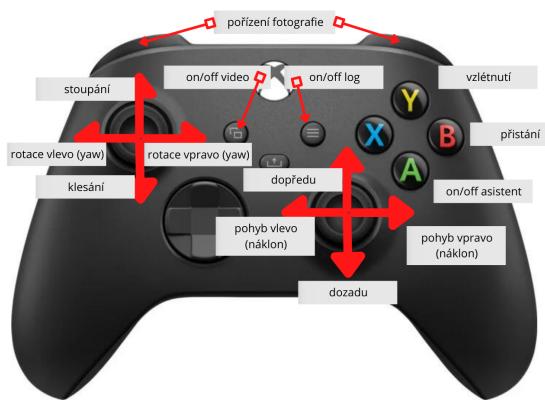
<sup>1</sup>[www.getfpv.com/learn/fpv-essentials/choosing-right-transmitter-mode/](http://www.getfpv.com/learn/fpv-essentials/choosing-right-transmitter-mode/)

<sup>2</sup>[www.ryzerobotics.com/tello](http://www.ryzerobotics.com/tello)

<sup>3</sup>Software Development Kit

bezpečné dráhy, video stream z dronu, vizualizační prvky, které napoví k jak silné korekci dochází (míru nebezpečí). Zároveň musí aplikace získávat velké množství pokynů od uživatele: pokyny pro dron, spouštění logování, přepínání asistenta, zapínání kamery, pokyn pro vzlet, pokyn pro přistání a pokyn pro pořízení fotografie.

Navrhoji pro ovládání využít myš v případě práce s dráhou a Xbox kontrolér. Misi bude možné načíst ze seznamu v modálním okně. U přepínačů asistenta a logování navrhoji vytvoření interaktivních tlačítek v GUI a jejich umístění i na Xbox kontroler, který bude mimo to využit pro pokyny letu, pořizování fotografií a zapínání videa viz obrázek 10. Poloha páček ovladače bude do GUI vizualizována, aby měl uživatel vizuální odezvu na jejich rozsah. Klávesnice je použita pouze na debug. Reálná aplikace by jí neměla.



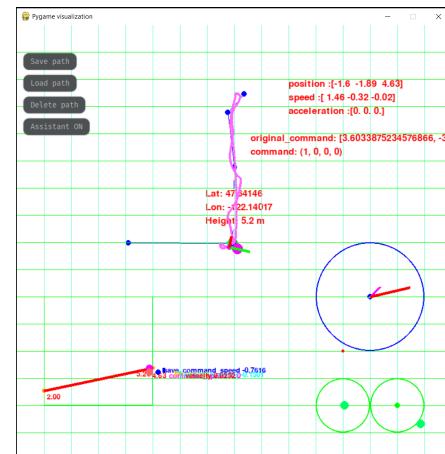
**Obrázek 10.** Popis ovládacích prvků na ovladači Xbox.

Bezpečnou dráhu lze importovat nebo vytvořit klikáním myši přímo v aplikaci ideálně podle zobrazené historie předchozího letu. Mise se ukládají jako soubory json, aby je bylo možné později načíst. v GUI je možné dráhu editovat (přesouvat kontrolní body).

Uživateli je potřeba dát najevo, jak moc je v nebezpečí. Tuto informaci nese jeho poloha samotná a také velikosti korekčních vektorů. Uživateli je zobrazen pohled shora, kde vidí polohu dronu v závislosti ke dráze. Dále vidí, jakou má dron orientaci a rychlosť (včetně směru letu). Tyto prvky fungují jako minimapa.

Aby uživatel dostal vhodnou informaci i o výšce, navrhoji vizualizovat pohled "zboku", kde jsou vždy zobrazené krajní body aktuálně vybraného nejbližšího segmentu bezpečné dráhy. Vůči nim je zobrazen dron a jeho spojnice s nejbližším bodem segmentu. Díky tomu je vidět, jak je dron vysoko a zároveň, zda dráha stoupá/klesá nebo je monotonní.

Uživateli se také vizualizují korekční vektory



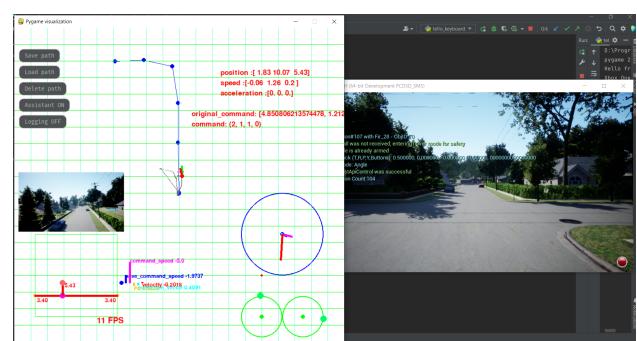
**Obrázek 11.** Samostatné okno GUI - mapa, vytvořená dráha, dron, historie polohy, vizualizace měřítka (jeden čtverec je 10 metrů), vizualizace korekčních vektorů.

a to jak vertikální, tak horizontální. Díky tomu vidí, v jaké míře je mu do letu zasahováno, respektive míru nebez-pečí.

## 5. Uživatelské testování

Cílem uživatelského testování je zjistit zda provedení průzkumné mise s pomocí letového asistenta (korekčního algoritmu) je jednodušší a hlavně bezpečnější. Subjektivní metrikou je míra snížení zátěže. Objektivními metrikami jsou hodnoty vzdálenosti od bezpečné trajektorie. Ty určují míru nebezpečí.

Pro testování bude použit simulátor AirSim. Hlavní kamera simulátoru je nastavena na režim pohledu třetí osoby. Je umístěna na zemi tak, jako kdyby uživatel řídil dron v realitě. Pokud se dron vzdálí od kamery, uživatel ho stěží vidí a musí se spolehnout na předsta-vené GUI, ve kterém vidí obraz z kamery dronu. Kamera dronu nedisponuje gimbalem. Pohled testovacího uživatele je zobrazen na obrázku 12.



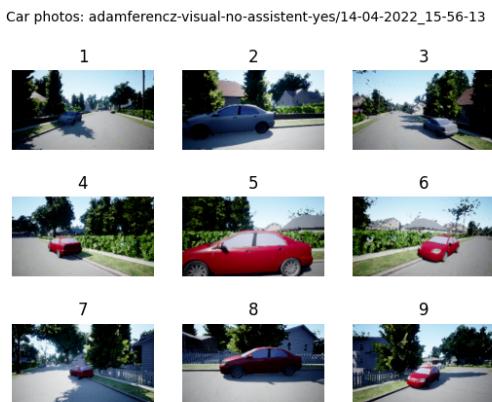
**Obrázek 12.** Experimentální GUI pro uživatelské testy v simulátoru AirSim.

Pro otestování představených aspektů je třeba, aby uživatel plnil pozorovací misi. Proto navrhoji následující testovací protokol:



**Obrázek 13.** Pohled shora na testovací misi.

Uživatel bude mít za úkol letět misi (viz. obrázek 13), při které je cílem vyfotografovat auta parkující na ulici a to vždy ze tří směrů: šikmo zepředu, z boku a šikmo ze zadu, jak to je zobrazeno na obrázku 14. Uživatel ví, že bezpečná dráha je uprostřed silnice a měl by se jí vždy držet (při letu s asistentem i bez). Po nařízení všech tří aut se tester vrací po bezpečné dráze na místo startu.

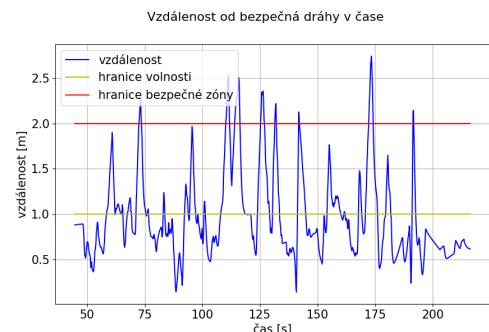


**Obrázek 14.** 9 fotografií aut potřebných ke splnění mise. Fotografie číslo 5 bude penalizována, jelikož není zachyceno celé vozidlo.

V grafu 15 jsou vidět dvě důležité hranice. První je "hranice volnosti" určující volnou oblast, ve které ještě není aplikován asistent (minimálně jeho aktuální korekční síla je nulová), druhou je "hranice bezpečné zóny". Pokud se za ní dron dostane, vyskytne se v nebezpečné zóně, což se zapíše do výsledku testu.

Z každého letu je pořízen komplexní log letových i programových údajů a summarizace statistických údajů. Pro vyhodnocení jsou vybraná následující kritéria:

1. Čas celého jednoho letu
2. Průměrná vzdálenost od ideální trajektorie
3. Poměr času stráveného mimo bezpečnou zónu (v procentech)
4. Počet chybných fotografií (chybí zcela, nebo nesplňuje požadavky)



**Obrázek 15.** Graf vzdálenosti dronu od ideální trajektorie v průběhu testovacího letu s asistentem.

Uživateli je nejdříve představeno ovládání. Seznámí se s chováním dronu krátkým volným letem. Poté si nanečisto proletí dráhu a vyzkouší si fotografování. Vyzkouší si také, jaké je, když na něj při letu působí asistent.

Před každým testovacím letem si uživatel podle pokynů připraví veškerá nastavení a s dronem vzletí. Když je dron ve vzduchu, spouští logování a zahajuje test. Po návratu test ukončuje vypnutím logování a resetuje program pro další let.

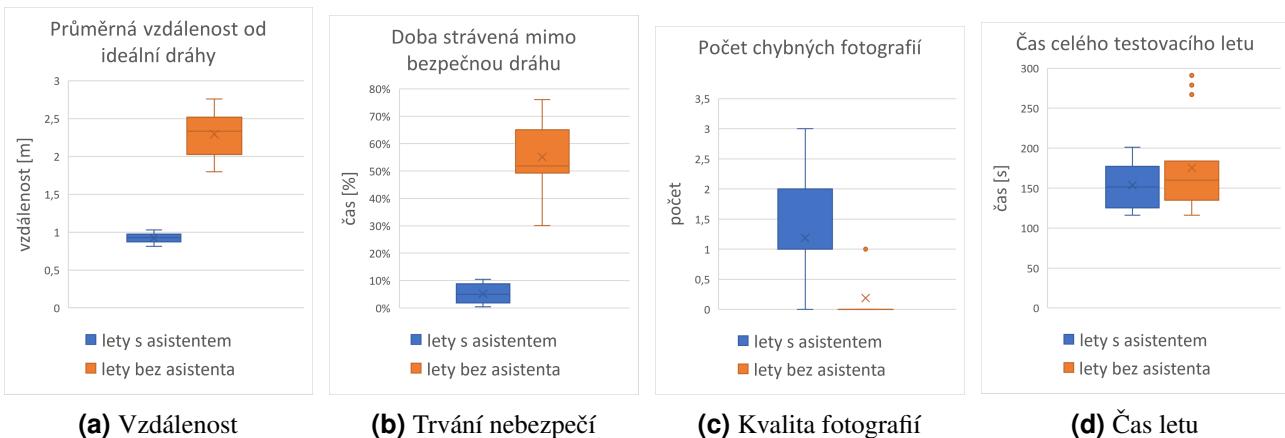
Takto uživatel letí 4 lety. Nejdříve bez asistenta a poté s asistencem.

Testování provedli 4 uživatelé: autor článku (muž, 23 let, menší zkušenost s létatím s druhy) a tři studenti střední školy (věk 15, 15 a 17 let, žádné zkušenosti s ovládáním dronu, ale rychlé porozumění principu řízení).

Výsledky testování ukázaly viz tabulka 1 a grafy 16, že s použitím asistenta značně klesá míra nebezpečí a to z 55% na 5%. Průměrná velikost vychýlení od dráhy je pro let s asistencem o 1,37 metrů nižší. Při testování uživatelům nejvíce dělalo problém pořídit kvalitní fotografie, pokud byli nuteni dostat se při letu s asistencem na kraj bezpečné zóny a tam "zápasit" s asistencem. Proto let s asistencem obsahuje v průměru jednu a více chybných fotografií. U letu bez pilota si uživatel dron umístil vždy jak potřeboval (na úkor bezpečnosti), tudíž je chybná fotografie ojedinělá viz graf 16c. Poznatky z testování budou použity na úpravu GUI a stabilizaci dronu při foci v krajních polohách.

	s asistencem	bez asistenta
Vzdálenost	0,92 m	2,29 m
Trvání nebezpečí	5,16 %	55,16 %
Kvalita fotografií	1,18 chyb	0,18 chyb
Čas letu	153,43 s	175,50 s

**Tabulka 1.** Tabulka průměrných hodnot měřených pro všechny lety.



Obrázek 16. Krabicový graf pro výstupní hodnoty uživatelských testů.

## 6. Závěr

Byla navržena nová metoda, která má za cíl snížit mentální zátěž pilota při plnění mise. Ta funguje na principu filtrování pokynů pilota na základě definované bezpečné dráhy, polohy, rychlosti dronu a původního pokynu pilota. Výstupem filtru je zkorigovaný pokyn, který je posán dronu místo původního.

Pro účely testování a vyhodnocení byla implementována experimentální aplikace. Ta umožnuje využít externí simulátor AirSim (nebo jiný simulátor či reálný dron) k uživatelským testům.

Experimenty ukázaly, že pilot, který nepoužívá pro plnění mise zapnutou asistenci, má odchylku od bezpečné dráhy o 1,37 metrů větší. Odchylka od bezpečné dráhy byla 2,29 m při bezpečné šířce 2 metry. Fotky bez asistenta jsou sice subjektivně lepší, ale je to za cenu zvýšeného rizika. Uživatel se zapnutým asistentem dosahoval průměrné odchylky 0,92 m za stejných podmínek. Jako negativum se ukázala situace, kdy pro pořízení fotky uživatel musel umístit dron na okraj zóny, v tu chvíli z důvodu korigování, bylo pořízení fotky nepříjemné.

Největším přínosem práce je princip, který dává uživateli volnost pohybu a přitom mu hlídá bezpečnost. Zároveň aplikace pracuje s mnoha různými nástroji a knihovnami z odvětví řízení dronu.

Díky získané zkušenosti uživatelských testů, bude do algoritmu přidána možnost "focus". Díky níž bude moct uživatel tlacítkem aktivovat zpomalený mód letu, kde letové pokyny jsou přesnější, pomalejší a volná zóna (bez zasahování asistenta) se posune až téměř na hranici bezpečné zóny.

Do budoucna by bylo vhodné provést experimenty s reálným dronem v reálném prostředí. Dále najít nestabilní situace navržené metody a zlepšit v nich stabilitu. Bylo by možné rozšířit řešení o další řídící prvky definující bezpečný prostor letu. Bezpečné území by tak mohlo být komplexnější. Zároveň v různých

místech by mohla být různá omezení.

## Poděkování

Velmi děkuji panu Ing. Vítězslavu Beranovi Ph.D. za jeho rady, které mi pomohly při tvorbě této práce. Dále děkuji Michaele Nepovímové a Tomášovi Novákovi za korekci textu. A všem účastníkům uživatelského testování.

## Literatura

- [1] Róbert Hubinák. Aplikace pro efektivní řízení dronu s využitím rozšířené virtuality, 2020.
- [2] Kamil Sedlmajer. Uživatelské rozhraní pro řízení dronu s využitím rozšířené virtuality. Master's thesis, Brno University of Technology, Faculty of Information Technology, 2019.
- [3] Kamil Sedlmajer, Daniel Bambušek, and Vítězslav Beran. Effective remote drone control using augmented virtuality. In *Proceedings of the 3rd International Conference on Computer-Human Interaction Research and Applications 2019*, pages 177–182. SciTePress - Science and Technology Publications, 2019.
- [4] Jules Bloomenthal, Chandrajit Bajaj, Jim Blinn, Marie-Paule Cani, Brian Wyvill, Alyn Rockwood, and Geoff Wyvill. *Introduction to implicit surfaces*. Morgan Kaufmann, 1997.
- [5] Shital Shah, Debadatta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and service robotics*, pages 621–635. Springer, 2018.