

4.4.5. Nástroj pro emulaci prostředí simulátoru

Ověření návrhu mělo podle původního plánu proběhnout po instalaci modulu VZ do lokomotivního simulátoru. Práce na výstavbě se ovšem zpozdily, zejména kvůli celosvětové pandemii choroby Covid-19 v roce 2020 a s ní spojenými omezeními, a v době před termínem odevzdání této práce nebyl simulátor provozuschopný. Z toho důvodu bylo nutné vytvořit řešení, kterým by bylo možné v hlediska modulu VZ napodobovat prostředí simulátoru.

Protože ve vzniklé situaci nezbyvalo mnoho času na komplexní návrh tohoto emulátoru, rozhodl se autor vytvořit SW část emulátoru jako program ve skriptovacím jazyce Python¹³, který se vyznačuje flexibilitou a jednoduchostí.

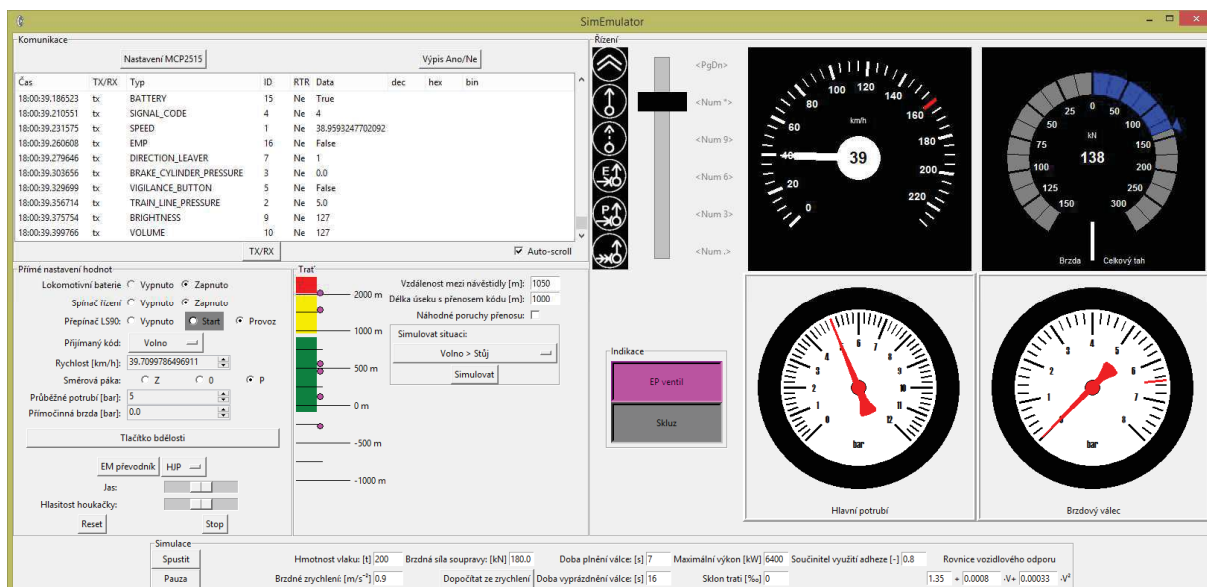
Veškerá datová komunikace mezi modulem VZ a zbytkem simulátoru probíhá přes rozhraní CAN bus připojené k modulu pomocí osmižilového UTP kabelu s konektorem RJ45. Připojení sběrnice CAN bus ke standardnímu osobnímu počítači, na kterém bude emulátor spouštěn, nebylo možné bez specializovaného hardwaru, byla proto použita vývojová deska Arduino UNO, převádí komunikaci ze sběrnice CAN na rozhraní USB. Deska Arduino je připojena k modulu s CAN kontrolérem MCP2515, tedy stejným, který je osazen na základní desce modulu VZ. S tímto modulem komunikuje MCU Arduino desky přes rozhraní SPI.

4.4.5.1. Program SimEmulator

Software emulující pro modul VZ prostředí lokomotivního simulátoru byl nazván SimEmulator. Vzhledem k faktu, že tento SW nebude po zprovoznění simulátoru již využíván, není v této práci jeho návrh popsán ve velkém detailu a jsou uvedeny pouze jeho hlavní funkcionality nutné pro komunikaci s modulem.

Program SimEmulator má grafické uživatelské ovládací rozhraní, vytvořené pomocí knihovny tkinter. Po spuštění aplikace je uživatel vyzván k výběru COM portu, na kterém má program komunikovat s Arduino deskou. Hlavní okno aplikace, které se otevře po úspěšném připojení k desce, je rozděleno na několik oddílů, ve kterých jsou sdruženy ovládací prvky pro jednotlivé funkční celky programu. Jednotlivé oddíly jsou popsány níže. Vzhled GUI programu ukazuje Obrázek 17.

¹³ Python je vysokoúrovňový interpretovaný programovací jazyk. Jednou z jeho specifických charakteristik je, že používá odsazení jako prvek syntaxe. Jednotlivé bloky kódu (např. tělo cyklu, funkce...) nejsou ohraničovány klíčovými slovy, ani závorkami, ale směrodatné je jejich odsazení od začátku řádku. Python je objektově orientovaný dynamický jazyk s důrazem na přehlednost a pochopitelnost kódu. [50]



Obrázek 17 Uživatelské rozhraní programu SimEmulator

4.4.5.1.1. Komunikace

V oddílu Komunikace se vypisují zprávy přijaté a odeslané programem. Pro každou zprávu je uvedeno, zda šlo o příchozí či odchozí zprávu, čas, její identifikátor na sběrnici CAN bus, datová část a příznak, zda šlo o „remote frame“. Zprávy může uživatel filtrovat na odchozí a příchozí. V tomto oddílu GUI je také umístěno tlačítko vyvolávající podokno, v němž je možné provést konfiguraci CAN kontroléru MCP2515 připojeného k Arduino desce.

4.4.5.1.2. Přímé nastavení hodnot

Oddíl Přímé nastavení hodnot byl využíván zejména v počáteční fázi vývoje modulu VZ. Je využit pro nastavování hodnot veškerých výstupních proměnných, zejména těch, pro které nebyl vytvořen speciální ovládací prvek. Tlačítko bdělosti zobrazené v tomto oddílu lze kromě myši obsloužit také stiskem mezerníku. Na tomto panelu je dále umístěno tlačítko, kterým lze zastavit simulaci a program ukončit.

4.4.5.1.3. Trať

V oddílu Trať je zobrazena jednoduchá simulace pohybu HV po infrastruktuře. Hlavní část tvoří stupnice v rozsahu +2000 m až -1000 m, vedle níž se pomocí barevných obdélníků symbolizuje, v jaké vzdálenosti od vozidla dojde ke změně přenášeného kódu VZ. (Záporná část stupnice slouží pro určení, jak daleko již vozidlo došlo při přijímání aktuálního kódu). Pomocí tlačítka je možné na trať vložit úseky simulující příjezd vlaku k návestidlu v poloze Stůj (tj. přenášení zeleného, žlutého a červeného kódu v po sobě následujících traťových oddílech). Délku těchto úseků je možné nastavit vyplněním hodnoty v příslušném textovém poli. Dále může uživatel zvlášť zadat hodnotu do pole „Vzdálenost mezi návestidly“. Je-li tato

hodnota větší než délka úseku s přenosem kódu, bude simulován na vzdálenosti odpovídající rozdílu hodnot výpadek příjmu kódu při průjezdu kolem návěstidla.

Zaškrtně-li uživatel políčko „Náhodné poruchy přenosu“, budou při vytváření simulovaných oddílů vloženy také krátké body, při průjezdu jimiž se bude přijímaný kód náhodně měnit. Tyto poruchy jsou zobrazeny jako fialová kolečka v místě odpovídajícím jejich poloze na trati.

Kliknutím levým tlačítkem myši na zobrazený úsek přenosu kódu se změní jeho kód na více povolující, tj. v posloupnosti: červená, žlutá, žluté mezikruží, zelená, poté případně zpět na červenou. Dvojklikem levého tlačítka je na místo kurzoru vložen nový úsek s přenosem kódu odpovídající nastavené délce. Pravé tlačítko myši vkládá na pozici kurzoru výše popsanou poruchu přenosu, stisknutí prostředního tlačítka myši nad některým zobrazeným prvkem ho z trati odstraní.

4.4.5.1.4. Řízení

Panel řízení simuluje rychloměr, indikátor tažné/brzdné síly lokomotivy a dvojici manometrů, z nichž jeden ukazuje hodnotu tlaku v hlavním potrubí přímočinné brzdy a druhý hodnotu tlaku v brzdovém válci přímočinné brzdy. Dále je v tomto oddíle zobrazen posuvník symbolizující hlavní jízdní páku sloužící k řízení vozidla. HJP je možné ovládat přetahováním posuvníku do vyznačených poloh nebo stiskem klávesy uvedené u dané polohy. Vzhled ovládacích prvků je inspirován přístroji na ovládacím pultu elektrických jednotek řady 440 (RegioPanter) a odvozených typů. Polohy HJP jsou od spodu: rychlobrzda (R), brzda pneumatická (BP), brzda elektrodynamická (BE), výběh (0), jízda (J), souhlas (S). Aretované jsou polohy R, 0 a J. [30]

4.4.5.1.5. Simulace

Poslední částí GUI je panel Simulace určený pro nastavování parametrů simulace jízdy vlaku, která vychází z pohybové rovnice vlaku (1)

$$\vec{F} + \vec{O}_v + \vec{O}_t + \vec{O}_z = \vec{0} \quad (1)$$

kde

\vec{F} ... tažná síla hnacích vozidel [N]

\vec{O}_v ... vozidlový odpor [N]

\vec{O}_t ... traťový odpor [N]

\vec{O}_z ... odpor zrychlení [N]

$\vec{0}$... nulový vektor

Brzdění vlaku je z pohledu této rovnice ekvivalentní se zvyšováním vozidlového odporu (díky tření mezi koly a brzdovými špalky).

Traťový odpor je s přihlédnutím k účelu tohoto programu zjednodušen pouze na odpor ze sklonu trati \vec{O}_s .

$$O_s = g \cdot m \cdot \sin\left(\arctg\left(\frac{s}{1000}\right)\right) \quad (2)$$

kde

g ... tíhové zrychlení [$kg \cdot m \cdot s^{-2}$]

m ... hmotnost vlaku [kg]

s ... sklon trati [%]

Vozidlový odpor je složen z valivého odporu kol po kolejnicích, aerodynamického odporu a dalších vlivů. Při výpočtech se pro jednotlivé typy vozidel experimentálně sestavují rovnice měrných vozidlových odporů ve tvaru

$$o_v = a + b \cdot V + c \cdot V^2 \quad (3)$$

kde

o_v ... měrný vozidlový odpor [N/kN]

V ... rychlost [$km \cdot h^{-1}$]

a, b, c ... koeficienty charakteristické pro daný typ vozu

Skutečný odpor vozu pak lze získat vynásobením měrného odporu hmotností vozidla v tunách.

Odpor ze zrychlení je důsledkem druhého Newtonova zákona.

V sekci Simulace je dále možné nastavit parametry brzdy vlaku. V praxi je brzdný účinek charakterizován veličinou brzdné procento, pro účely emulátoru bylo však použito přímé vyjádření síly, kterou vlak brzdí při dosažení tlaku 3,8 bar v brzdových válcích.

Po spuštění simulace program SimEmulator vypočítává s krokem 0,001 s hodnoty simulačních veličin, které jsou v intervalu 0,1 s odesílány modulu VZ. Přijatou informaci o kontrolním světle ovládacího panelu LS-90 vyjadřuje podbarvení políčka Start položky Přepínač LS-90 v oddílu Přímé nastavení hodnot, při příjmu zprávy o otevření EP ventilu emulátor zadá požadavek na maximální brzdění EDB a odvětrá hlavní potrubí průběžné brzdy.

Zdrojový kód programu SimEmulator je přílohou č. 4 této práce. Samotný program je zapsán v souboru main.py, soubory registers.csv a bits.csv obsahují informace pro konfiguraci CAN kontroléru MCP2515 příslušnou funkcí programu. Formát CAN zpráv je popsán v souboru can_config.json.

Autor by na tomto místě chtěl podotknout, že si je vědom možných nedostatků v implementaci programu SimEmulator. Vzhledem k pravděpodobnému krátkému využití a

omezeným časovým možnostem jeho vývoje, byl však uznán jako dostatečný pro potřeby této bakalářské práce.

4.4.5.2. Deska Arduino UNO

Deska Arduino UNO provádí převod komunikace z USB portu počítače na sběrnici CAN bus. Součástí převodu je také mikrokontroler ATmega16U2 osazený na desce, jehož úkolem je převod z USB na UART¹⁴ rozhraní s 5 V logikou, které dále vstupuje do hlavního čipu desky ATmega328P.

Data zasílaná pomocí sériového portu mají strukturu odpovídající způsobu, jakým MCU přes rozhraní SPI předává zprávy k odeslání CAN kontroléru. Posloupnost přenášených bytů ukazuje Tabulka 7. Je-li identifikátor zprávy standardní, což je indikováno hodnotou 0 bitu EXIDE v registru SIDL, byty 3 a 4 se nepřenášejí. MCU je kvůli uspořádání paměti obvodu MCP2515 registry EID8 a EID 0 v tomto případě nevynechává, pouze do nich zapíše nulovou hodnotu. Vzhledem k přenosové rychlosti sériového portu, která je podstatně nižší než u rozhraní SPI, je snížení objemu přenášených dat žádoucí. V opačném směru probíhá převod mezi CAN a sériovým portem stejným způsobem.

Bity C a CC v registru SIDL mají být při zápisu do MCP2515 nastaveny na hodnotu 0. Jsou-li při přijetí zprávy z programu SimEmulator nastaveny na hodnotu 1, vyhodnotí to MCU jako zvláštní případ, kdy PC neposílá požadavek na odeslání zprávy po sběrnici CAN, ale jde o instrukci pro desku Arduino. Kód instrukce je v takovém případě přijat v předchozím bytu SIDH. Instrukce s kódem 50 (odpovídajícím ASCII kódu znaku „2“) přes SPI přečte a na sériový port odešle postupně hodnoty všech registrů obvodu MCP2515, naopak instrukce 51 (ASCII kód „3“) zapisuje následujících 127 bytů přijatých z PC jako konfiguraci kontroléru. Tyto dvě instrukce umožňují nepřímé nastavení MCP2515 z připojeného počítače.

Arduino UNO využívá pro detekci příchozí CAN zprávy signál přerušení generovaný CAN kontrolérem stejně jako MCU na základní desce modulu VZ.

Soubor projektu z prostředí Arduino IDE se zdrojovým kódem SW použité desky Arduino_UART2CAN.ino je přílohou číslo 5 této práce.

¹⁴ Zkratka UART označuje doslova „univerzální asynchronní přijímač-vysílač“ (universal asynchronous receiver-transmitter). [51]