



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMOBILNÍHO A DOPRAVNÍHO
INŽENÝRSTVÍ**

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF AUTOMOTIVE ENGINEERING

VIZUALIZACE VÝSLEDKŮ SIMULACE JÍZDNÍ DYNAMIKY VOZIDLA

VISUALIZATION OF THE RESULTS OF VEHICLE DYNAMICS SIMULATIONS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. TOMÁŠ ŠTEFANEC

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. PETR PORTEŠ, Dr.

BRNO 2014

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav automobilního a dopravního inženýrství

Akademický rok: 2013/14

ZADÁNÍ DIPLOMOVÉ PRÁCE

student(ka): Bc. Tomáš Štefanec

který/která studuje v **magisterském studijním programu**

obor: **Automobilní a dopravní inženýrství (2301T038)**

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Vizualizace výsledků simulace jízdní dynamiky vozidla

v anglickém jazyce:

Visualization of the results of vehicle dynamics simulations

Stručná charakteristika problematiky úkolu:

V Simulink 3D Animation vytvořte vizualizační prostředí pro zobrazení jízdy vozidla. Pohyb vozidla je řízen výsledky simulačních výpočtů.

Cíle diplomové práce:

Cílem je vytvořit vizualizační prostředí, které umožní:

- rychlé sestavení scény jízdní zkoušky, zkušební tratě
- rychlé vytvoření vizuální reprezentace vozidel
- zobrazit pohyb vozidla a veličiny charakterizující dynamické vlastnosti vozidla
- prohlížet scénu z různých pohledů



Seznam odborné literatury:

GILLESPIE, T. D. Fundamentals of Vehicle Dynamics. Warrendale: Society of Automotive Engineers, 1992. ISBN 1-56091-199-9.

KARBAN P. Matlab a Simulink. Computer press, a.s. Brno 2006. ISBN 978-80-251-1448-3.

ŽÁRA J. VRML 97. Computer press, a.s. Brno 1999. ISBN 80-7226-143-6x.

Vedoucí diplomové práce: Ing. Petr Porteš, Dr.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2013/14.

V Brně, dne 19.11.2013



prof. Ing. Václav Píštěk, DrSc.
Ředitel ústavu

prof. RNDr. Miroslav Doupovec, CSc., dr. h. c.
Děkan



ABSTRAKT

Diplomová práce je zaměřena na tvorbu vizualizačního prostředí zobrazujícího pohyb vozidla. Dochází zde k propojení virtuální reality s prostředím Matlab/Simulink a jeho toolboxem 3D Animation. Základní parametry vozidla se dají jednoduše editovat pomocí aplikace vytvořené v softwaru Matlab. Společně s pohybem vozu jsou vykreslovány vybrané dynamické účinky. Vizualizace je řízena výsledky simulačních výpočtů nebo měření.

KLÍČOVÁ SLOVA

vozidlo, jízdní dynamika vozidla, jazyk VRML, virtuální realita, Matlab, Simulink 3D Animation

ABSTRACT

The master thesis is focused on creation of a visualization environment that displays a movement of a vehicle. A connection of the virtual reality with Matlab/Simulink and its 3D Animation toolbox is realized. Basic parameters of the vehicle can be easily edited using application created in Matlab. Selected dynamic effects are rendered together with the movement of the vehicle. The visualization is managed by results of the simulation calculations or measurements.

KEYWORDS

vehicle, vehicle dynamics, VRML language, virtual reality, Matlab, Simulink 3D Animation



BIBLIOGRAFICKÁ CITACE

ŠTEFANEC, T. *Vizualizace výsledků simulace jízdní dynamiky vozidla*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2014. 78 s. Vedoucí diplomové práce Ing. Petr Porteš, Dr.



ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že tato práce je mým původním dílem, zpracoval jsem ji samostatně pod vedením Ing. Petra Porteše, Dr. a s použitím literatury uvedené v seznamu.

V Brně dne 30. května 2014

.....

Tomáš Štefanec



PODĚKOVÁNÍ

Tímto bych chtěl poděkovat vedoucímu mé diplomové práce Ing. Petru Portešovi, Dr. za cenné rady, připomínky a poskytnuté materiály, kterými přispěl k vypracování této práce. Děkuji také panu Žárovi za pomoc při řešení problémů s virtuální realitou. Velké díky patří mé rodině za podporu při studiu.



OBSAH

Úvod.....	10
1 Použitý software	11
1.1 Matlab	11
1.1.1 M-soubory	12
1.1.2 Struktury.....	13
1.1.3 Matlab Guide.....	14
1.2 Simulink	15
1.2.1 Toolbox 3D Animation	15
1.3 V-Realm Builder	16
2 Virtuální realita.....	18
2.1 Struktura VRML	19
2.2 Souřadný systém	19
2.3 Informace o virtuálním světě.....	20
2.3.1 Viewpoint	20
2.3.2 Navigationinfo.....	20
2.3.3 Worldinfo	21
2.4 Vytvoření iluze prostoru.....	21
2.4.1 Pozadí.....	21
2.4.2 Bodový zdroj světla.....	22
2.5 Vytváření objektů.....	23
2.5.1 Základní tělesa.....	24
2.5.2 Obecná tělesa.....	24
2.5.3 IndexedFaceSet	24
2.6 Nastavení vzhledu objektů	25
2.7 Pojmenování uzlů.....	26
3 Testy pro osobní vozidla dle normy ISO.....	27
3.1 Double lane-change test – ISO 3888-1.....	27
3.1.1 Test 1	28
3.1.2 Test 2.....	29
3.2 Manévr ISO 3888-2.....	29
3.3 Vytyčení testovací trati.....	30
4 Dynamika vozidla.....	32
4.1 Souřadný systém vozidla.....	32
4.2 Globální souřadný systém	32
4.3 Veličiny trajektorie vozidla.....	33
4.3.1 Rychlost vozidla v horizontální rovině	33
4.3.2 Úhel směrové úchyly v těžišti vozidla	33



4.3.3	Směrový úhel.....	34
4.4	Síly na kolech.....	34
4.4.1	Podélná síla	34
4.4.2	Boční síla.....	35
4.4.3	Třecí kružnice.....	36
4.5	Jednostopý model vozidla	38
5	Vytvoření virtuální scény	41
5.1	Aplikace VRML CREATOR	41
5.2	Funkce vrml	43
5.2.1	Definice vozidla	44
5.2.2	Testovací plocha.....	47
5.2.3	Funkce vrml12.....	47
5.2.4	Funkce vrml3.....	49
5.2.5	Funkce vrml4.....	50
5.3	Zobrazení ovládacích prvků vozidla	51
5.3.1	Model volantu	52
5.3.2	Ukazatel zařazeného rychlostního pole	53
5.3.3	Přístrojová deska	53
5.4	Zobrazení dynamických účinků	54
5.4.1	Rychlost a zrychlení	54
5.4.2	Silové účinky na kolech	56
5.4.3	Výpis hodnot vybraných veličin.....	56
5.5	Pohledy ve scéně.....	58
5.6	Předávání událostí mezi uzly.....	59
5.7	Konečný vzhled scény.....	61
6	Model v prostředí Simulink.....	62
6.1	Příprava vstupních dat.....	62
6.2	Vytvoření signálů	63
6.2.1	Signál translace	64
6.2.2	Signál rotace.....	65
6.2.3	Signál měřítka	65
6.2.4	Signál pro zobrazení rychlostního stupně	66
6.3	Blok VR Sink	67
6.4	VR Text Output.....	67
6.5	Výsledná animace	69
	Závěr	70
	Seznam použitých zkratk a symbolů	73
	Seznam příloh.....	77



ÚVOD

Tato diplomová práce se zabývá možnostmi vizualizace jízdy vozidla, ovládané naměřenými nebo vypočtenými daty. Je zde snaha o rychlé a snadné vytvoření vizuální reprezentace vozidla a jízdní dráhy. Tato snaha vychází z časové náročnosti v situaci, kdy se má sestavit reálný model vozidla. Vozidlo totiž musí být nejdříve vhodně připraveno. Následně musí dojít k nasnímání celého vozu pomocí 3D skeneru. Ve finální etapě se pak jednotlivé naskenované díly skládají dohromady. Pro základní představu o pohybu vozidla však stačí jednoduchý model, který by byl vytvořen v několika málo krocích.

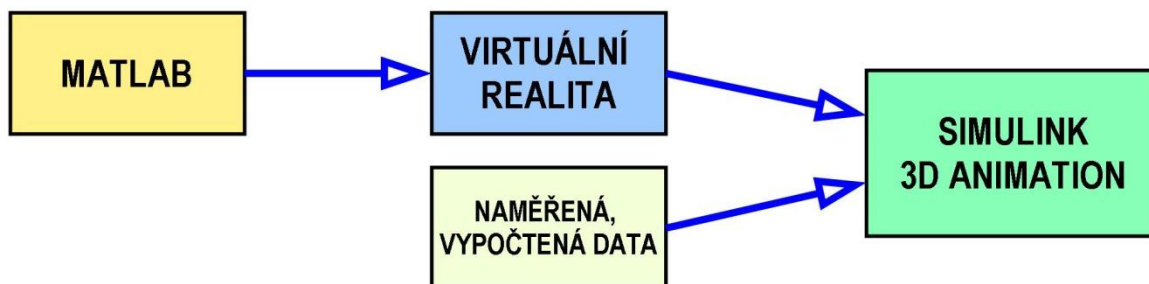
V první části práce je nejdříve ve stručnosti popsán software, který byl využíván při zpracování diplomové práce. Jedná se především o programy Matlab, Simulink a V-Realm Builder. Dále jsou zmíněny a objasněny jízdní testy prováděné dle normy ISO 3888. Modely trati pro tyto testy jsou obsahem praktické části. Poslední díl teoretické části tvoří popis vybraných dynamických veličin. Jedná se především o veličiny a účinky zobrazované později při simulaci pohybu vozidla. Součástí je i rozbor jednostopého modelu vozidla, z něhož pochází výsledky použité jako vstupní data pro řízení simulace.

Druhá část se dělí na dva oddíly. V prvním je vysvětlen postup při návrhu a tvorbě vizualizačního prostředí. Popisuje se zde vytvořená uživatelská aplikace, pomocí níž vzniká virtuální scéna. Aplikace dovoluje zadávat základní parametry vozidla a vybrat některou z předdefinovaných variant trati. Model vozidla je tak z části parametrický, aby se alespoň blížil rozměrům skutečného automobilu. Obsahem je dále naznačení principu tvorby virtuální scény s využitím programu Matlab. Rozebrány jsou jednotlivé funkce obstarávající tvorbu virtuální reality. V poslední části práce dochází díky programu Simulink a jeho toolboxu 3D Animation k interakci mezi vytvořenou scénou a daty vstupujícími jako řídicí signál do simulace pohybu vozu. Model pro řízení pohybu a vykreslování dynamických účinků na vozidlo je sestaven z funkčních bloků programu Simulink. Potřebné signály, dále vstupující jako parametry do konkrétních uzlů scény, jsou vytvářeny v subsystémech. Tvar signálů je vysvětlen spolu s nastavením hlavních bloků modelu. Na závěr je uvedena finální podoba animace jízdy vozidla se všemi doplňkovými prvky.



1 POUŽITÝ SOFTWARE

Tato kapitola slouží pro základní popis softwaru, který jsem využíval při tvorbě této diplomové práce. Jedná se především o tyto programy: MATLAB 2012b, Simulink, V-Realm Builder 2.0. Dále jsem pracoval s programy: VmIPad, PSPad (kontrola chyb, úprava zdrojových kódů), Layout (součást programu ShetchUp), Autodesk Inventor 2013, Adobe Photoshop CS3. Práce v doplňkových programech nebude popsána, neboť nemá bližší souvislost s tématem diplomové práce.



Obr. 1.1 Struktura práce

1.1 MATLAB

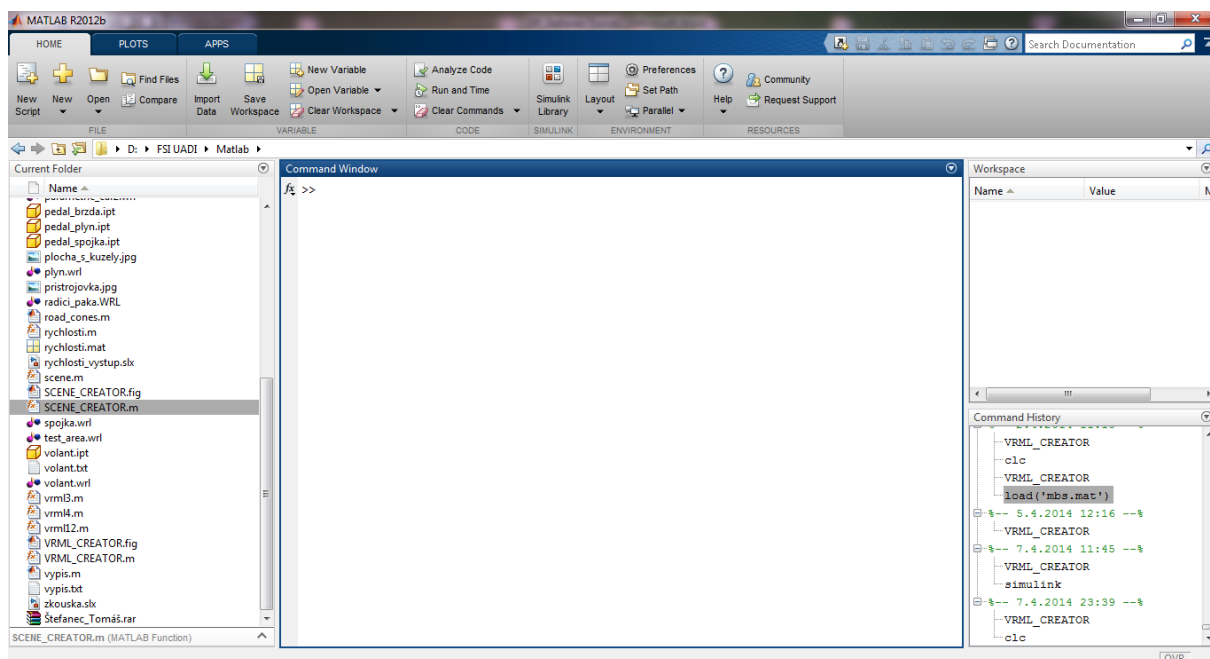
MATLAB, jakožto výpočetní systém, se v posledních letech stal celosvětovým standardem v oblasti výpočtů a simulací ve vědě, výzkumu, průmyslu i ve sféře vzdělávání. Název je zkratkou anglického Matrix Laboratory = maticová laboratoř. Prostředí tohoto programu slouží pro vědeckotechnické výpočty, modelování, tvorbu algoritmů, simulace, měření a zpracování signálů, analýzu dat, atd. Matlab poskytuje uživatelům výpočetní a grafické nástroje, obsáhle knihovny funkcí a také výkonný programovací jazyk čtvrté generace. Za velké plus je považováno velmi rychlé výpočetní jádro s optimálními algoritmy. Obsahuje sofistikované řešiče soustav algebraických rovnic, podporuje komplexní čísla, zvládá práci s velkými řídkými maticemi nebo řešení soustav obyčejných diferenciálních rovnic. [6; 13]

Hlavní rysy MATLABu: [13]

- Rychlé výpočetní jádro
- Programovací jazyk 4. generace
- Rozšiřitelný systém
- Podpora paralelních výpočtů
- Množství aplikačních knihoven
- Působivá 2D a 3D grafika
- Podpora vícerozměrných polí a datových struktur
- Interaktivní průvodce importem dat
- Komunikace s externími měřicími a monitorovacími přístroji v reálném čase
- Rozšiřitelnost modulů
- Integrace s jazykem Java



Matlab je dostupný pro všechny významné platformy – Windows, Linux, Solaris, Mac. V rámci této diplomové práce byl používán v operačním systému Windows 7. Rozložení pracovního prostředí ukazuje obrázek 1.2. Toto prostředí lze rozdělit do několika oblastí. V horní části se nachází 3 záložky – *HOME*, *PLOTS*, *APPS*. Záložka *HOME* obsahuje např. ikony pro nastavení Matlabu, volání nápovědy, import dat, vytváření nových souborů, atd. V záložce *PLOTS* lze zvolit jednotlivé typy grafů pro vykreslení dat. Záložka *APPS* nabízí spuštění přídatných toolboxů, případně jejich doinstalaci. Pod horní lištou se nachází řádek, tzv. *Current Directory*, ve kterém se nastavuje pracovní adresář. Pracovní adresář je místo, kam se ukládají soubory vytvořené v Matlabu. Nastavení tohoto adresáře by proto mělo následovat ihned po spuštění. V levé části obrazovky se nachází okno *Current Folder*, kde vidíme obsah pracovního adresáře. Z tohoto okna lze spouštět jednotlivé skripty a také provádět základní úkony se soubory (mazat, přejmenovat, kopírovat,...). V pravé části pod horní lištou najdeme okno *Workspace* – pracovní plocha. Ukládají se sem všechny vytvořené a používané proměnné. Pod tímto oknem vpravo dole je umístěno okno *Command History* – historie příkazů. Zobrazují se zde všechny použité příkazy. Ty lze znovu vyvolat dvojitým poklikáním myši na daný příkaz. Uprostřed obrazovky najdeme nejdůležitější okno, a to *Command Window* – příkazové okno. V této části zadáváme příkazy, vypisujeme obsah a hodnoty proměnných, případně varování nebo chybová hlášení. Zajišťuje komunikaci s výpočetním jádrem.



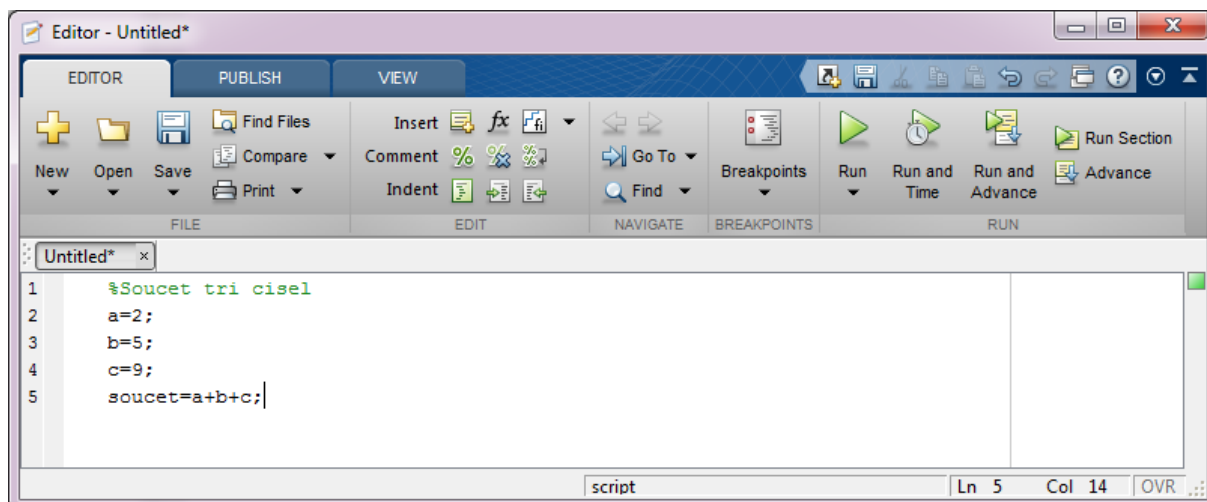
Obr. 1.2 Pracovní prostředí programu Matlab

1.1.1 M-SOUBORY

Uživatel má dvě možnosti jak pracovat s Matlabem. První možností je využití tzv. interaktivního přístupu, kdy se zadávají příkazy přímo do příkazového okna v hlavním okně a zde jsou také spouštěny. Tato možnost je vhodná pouze pro jednoduché výpočty menšího rozsahu, ale také například pro vyzkoušení příkazů a algoritmů. Práci v interaktivním režimu lze přirovnat k práci s kalkulačkou. Nevýhodou při tomto způsobu práce je nemožnost uložení posloupnosti příkazů. To znamená, že v případě ukončení práce v Matlabu nebo náhlého výpadku proudu ztrácíme zapsané příkazy a jejich posloupnost. [6; 8]



Druhou možností je vytvoření tzv. m-souborů, které mají svoji příponu *.m. Příkazy jsou zapisovány do textového souboru, ze kterého jsou pak spouštěny v zapsaném pořadí. M-soubor lze vytvořit v libovolném textovém editoru (bez formátování) nebo pomocí vestavěného editoru, přímo určeného k jejich tvorbě. Editor lze spustit buď příkazem *edit*, nebo kliknutím na ikonu *New Script* v hlavním okně (záložka *Home*). Hlavní výhodou oproti interaktivnímu způsobu je tedy možnost uložení souboru na disk počítače. Příkazy tak lze spouštět opakovaně i po ukončení Matlabu. Dalšími výhodami jsou: zvýraznění chybně zapsané syntaxe příkazu, nástroje pro optimalizaci skriptu, použití příkazů *if*, *switch-case* a cyklů *for*, *while*, atd. [6; 8]



Obr. 1.3 Zapsání příkazů v prostředí vestavěného editoru

M-soubory se dělí na dva typy – *skript* a *funkce*. Skript je nejjednodušší m-soubor, obsahuje popořadě zapsané příkazy a povely dle postupu naší práce. Jedná se tedy o jakýsi seznam příkazů. Spuštění skriptu se provede kliknutím na ikonu *Run* přímo v editoru nebo zapsáním jména skriptu bez přípony v příkazovém okně Matlabu. Funkce je, dá se říct, „chytřejší“ skript. Může být volána s jedním nebo několika vstupními parametry a také může předávat výstupní parametry. Proměnné definované v rámci funkce po provedení příkazů zanikají a nezůstávají uložené v okně *Workspace*, jako je tomu po spuštění jednoduchého skriptu. [6; 8]

1.1.2 STRUKTURY

„Struktura je podobný datový typ jako pole, umožňuje uložit kolekci proměnných různých typů do jedné proměnné. Na rozdíl od pole, kde jsou jednotlivé buňky indexovány pomocí pozic v poli, jsou struktury adresovány pomocí textových identifikátorů.“ [8]

Struktury se vytváří tak, že přiřadíme jednotlivým polím struktury jejich obsah. Jako příklad lze uvést vytvoření struktury se jménem *automobil*. Tato struktura dále obsahuje pole s názvy: *znacka*, *model*, *motor*. Přístup k jednotlivým polím struktury je naznačen na obrázku 1.4. Operace se strukturami jsou dále v této práci využity při zpracování a přípravě vstupní matice pro řízení modelu virtuální reality v prostředí Simulink 3D Animation.



```
>> automobil.znacka='Skoda';
>> automobil.model='Octavia';
>> automobil.motor=1.2;
>> automobil

automobil =

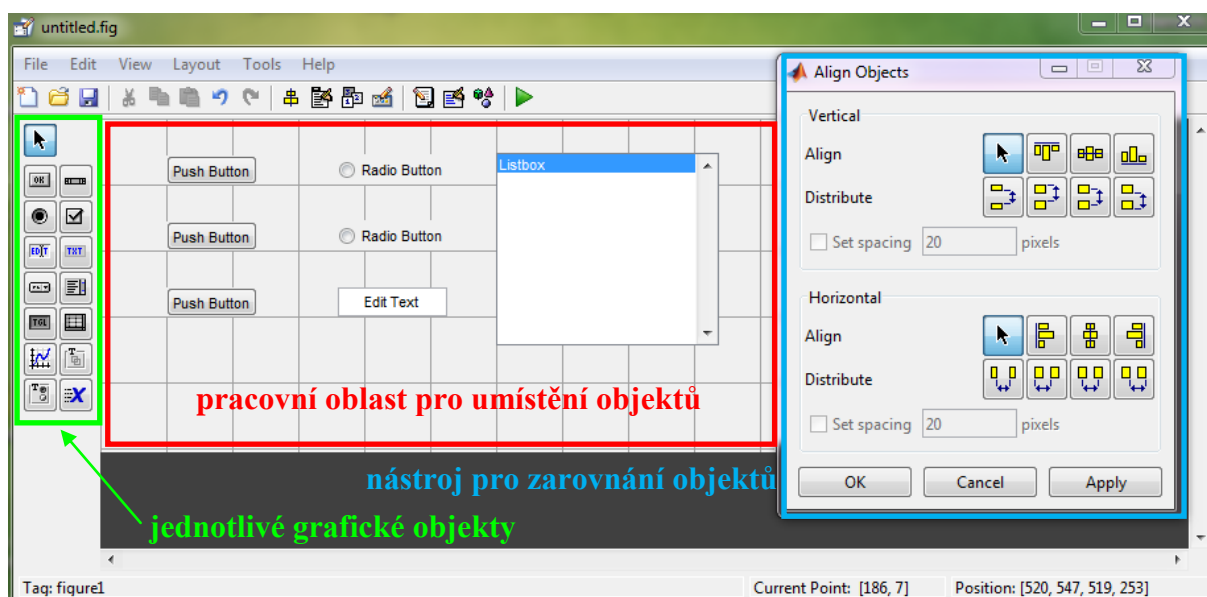
znacka: 'Skoda'
model: 'Octavia'
motor: 1.2000
```

Obr. 1.4 Vytvoření struktury se třemi poli

1.1.3 MATLAB GUIDE

V prostředí Matlabu je možno vytvářet různé grafické objekty a to buď „ručně“, nebo interaktivně. GUIDE je nástroj pro interaktivní tvorbu grafických objektů integrovaný do systému Matlab. Název je zkratkou Graphical User Interface Development Enviroment. Jde o jakýsi automatický generátor grafických objektů. Uživatel pomocí myši umísťuje do plochy jednotlivé objekty (tlačítko, rozbalovací menu, tabulka, graf, textové pole, zaškrťovací políčko, posuvník), nastavuje jejich polohu, vzhled a další vlastnosti. Oproti „ruční“ tvorbě jde o časově nenáročnou metodu. Malou nevýhodou je fakt, že generované zdrojové texty jsou delší a ne vždy zcela optimální na orientování se v nich. Výhodou však je, že uživatel si nemusí pamatovat velkou spoustu příkazů. [8]

Po uložení projektu se v aktivním adresáři objeví dva soubory, jeden s příponou *.fig, druhý s příponou *.m. Do prvního souboru se ukládá samotný vzhled grafického objektu. Druhý obsahuje jeho zdrojový kód včetně odezev na akce učiněné uživatelem (např. kliknutí na tlačítko, výběr v menu, atd.). Tento interaktivní nástroj byl v rámci této diplomové práce využitý pro tvorbu aplikace k zadávání parametrů vozidla. Pracovní prostředí GUIDE je na obrázku 1.5.

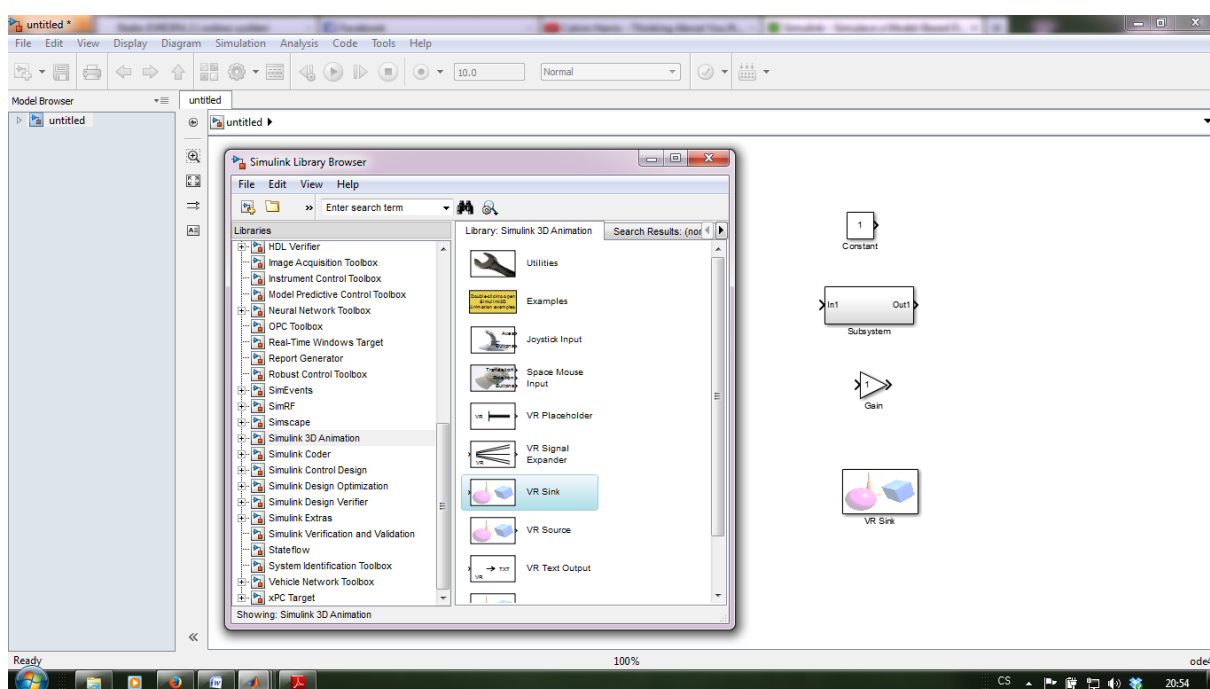


Obr. 1.5 Prostedí Matlab GUIDE



1.2 SIMULINK

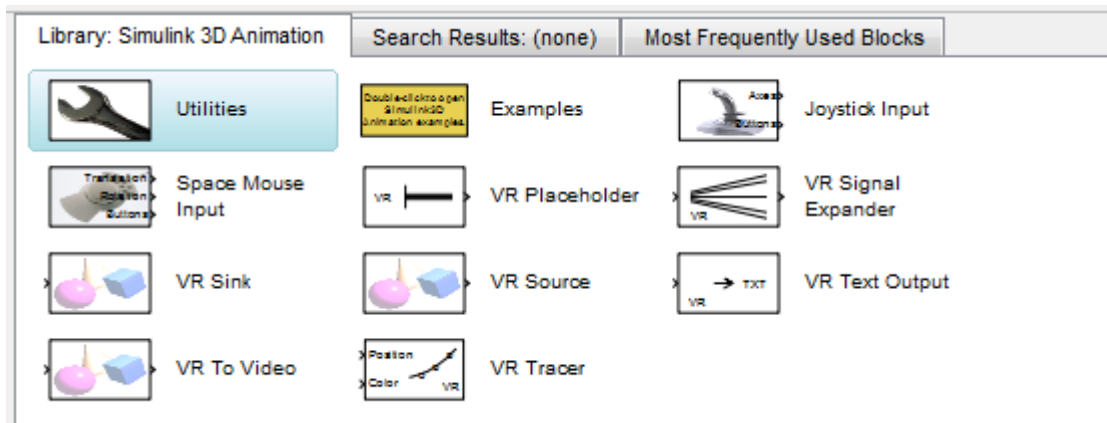
Simulink je jedna z nejznámějších a nejpoužívanějších nadstavb Matlabu. Slouží pro modelování dynamických systémů, jejich simulaci a analýzu. Název vznikl spojením anglických slov SIMULATION a LINK, což znamená simulace a spojení. Tento toolbox dovoluje uživateli snadno tvořit modely soustav v podobě blokových schémat a rovnic. Pomocí umísťování a přesouvání funkčních bloků můžeme vytvářet systémy lineární, nelineární, v čase diskretní nebo spojitý. Jednotlivé funkční bloky nalezneme v knihovně – Simulink Library Browser. Díky otevřené architektuře lze nadefinovat svoje vlastní bloky a rozšířit tak zmíněnou knihovnu předdefinovaných bloků. Součástí jsou také tzv. blocksety. To jsou specializované knihovny pro jednotlivé vědecké a technické obory, které rozšiřují základní knihovnu Simulinku. Seznam dostupných blocksetů najdeme v levé části okna knihoven.



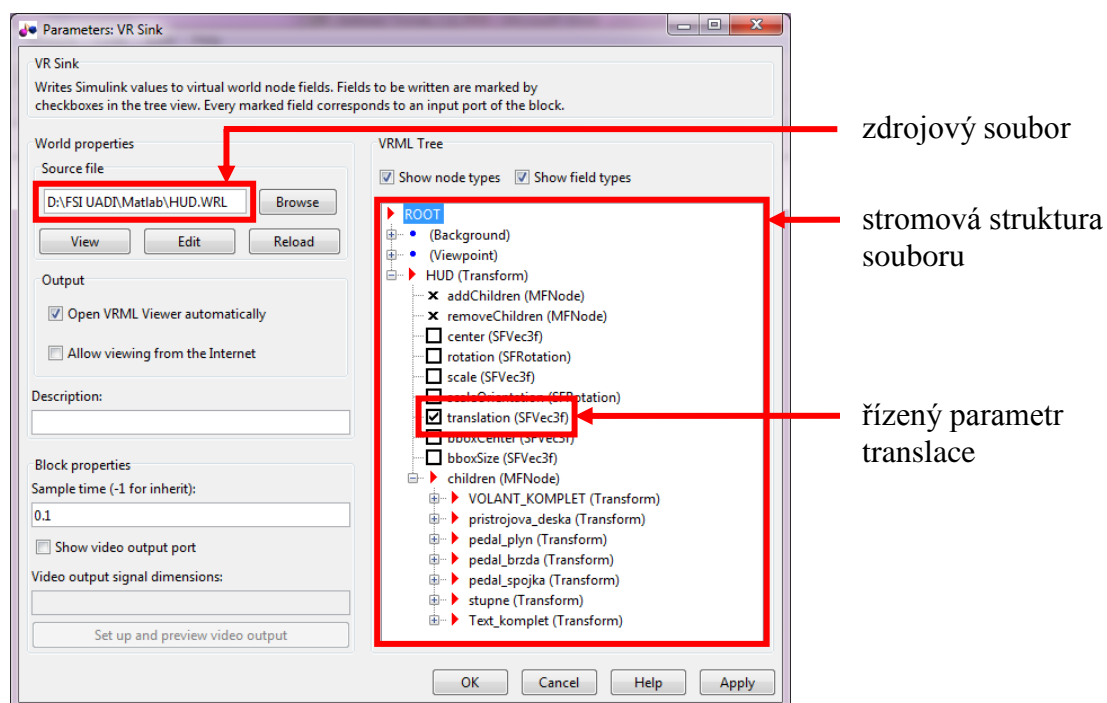
Obr. 1.6 Prostředí Simulinku s oknem knihoven funkčních bloků

1.2.1 TOOLBOX 3D ANIMATION

3D Animation (ve starších verzích Virtual Reality Toolbox) umožňuje vzájemnou obousměrnou komunikaci mezi virtuální realitou a prostředím Simulink. Je založen na standardu jazyka VRML 97, jenž bude popsán v následující kapitole. Propojením dynamických modelů se scénami virtuální reality lze dynamické systémy vizualizovat a uživatel tak získá lepší představu o tom, jak pracují. Simulace je možno pozorovat z různých pohledů, uživatel se může ve virtuálním světě pohybovat a ovládat ho. Pro možnost ovládání je potřeba využít bloky, které slouží pro připojení speciálních periférií, jako např. joystick nebo SpaceMouse (3D polohovací zařízení), atd. [13]



Obr. 1.8 Bloky toolboxu 3D Animation



Obr. 1.7 Parametry bloku VR Sink

1.3 V-REALM BUILDER

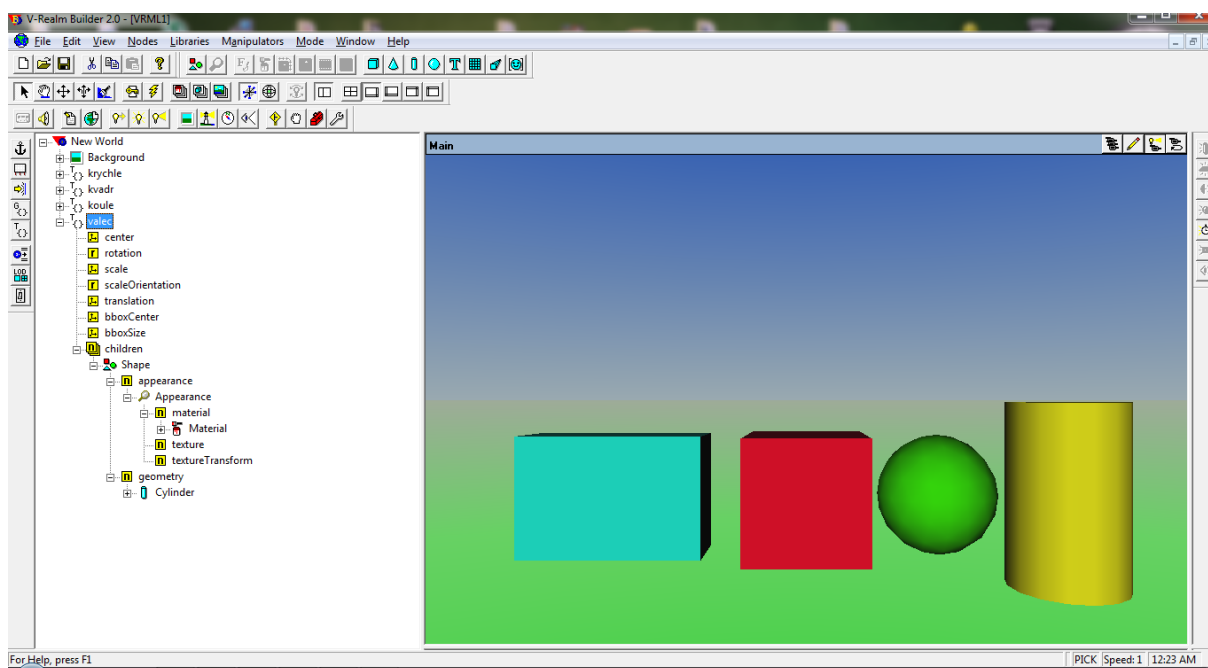
Virtuální světy lze principiálně tvořit dvěma způsoby. První a náročnější je zapisování virtuálních scén pomocí syntaxe jazyka VRML 97 v textovém editoru, např. *Poznámkový blok* v prostředí Windows. V takovém případě je nutné soubor uložit s příponou *.wrl. Existují však i speciální textové editory, např. *VrmlPad*, které dokážou okamžitě označit chyby v zapsaném příkazu a také nabízí seznam možných příkazů po zapsání prvních několika písmen. Tyto funkce usnadňují práci, stále však je tento způsob nejenom časově náročný, ale vyžaduje také dokonalou znalost struktury jazyka VRML.

Druhým způsobem je práce s editory, které vkládají do scény přímo definované objekty. S objekty je možno pohybovat, měnit jejich rozměry, měnit barvy, atd. a uživatel ihned vidí provedené změny. Tento způsob je tedy daleko rychlejší, více srozumitelný, je intuitivní a pro



začátečníky vhodnější. Nevyžaduje hlubší znalosti jazyka VRML, neboť po dokončení a uložení virtuálního modelu editor automaticky generuje zdrojový kód. Vytvořený soubor má příponu *.wrl. Mezi editory, které umožňují pracovat tímto způsobem, se řadí také program V-Realm Builder. Ten je součástí instalačního balíčku Matlabu. Standardně však není nainstalován a je potřeba ho ručně doinstalovat.

Program V-Realm Builder jsem využíval pro ověřování správných pozic jednotlivých prvků virtuální scény a pro případnou korekci zdrojového textu. Pracovní prostředí je zobrazeno na obrázku 1.9.



Obr. 1.9 Okno programu V-Realm Builder



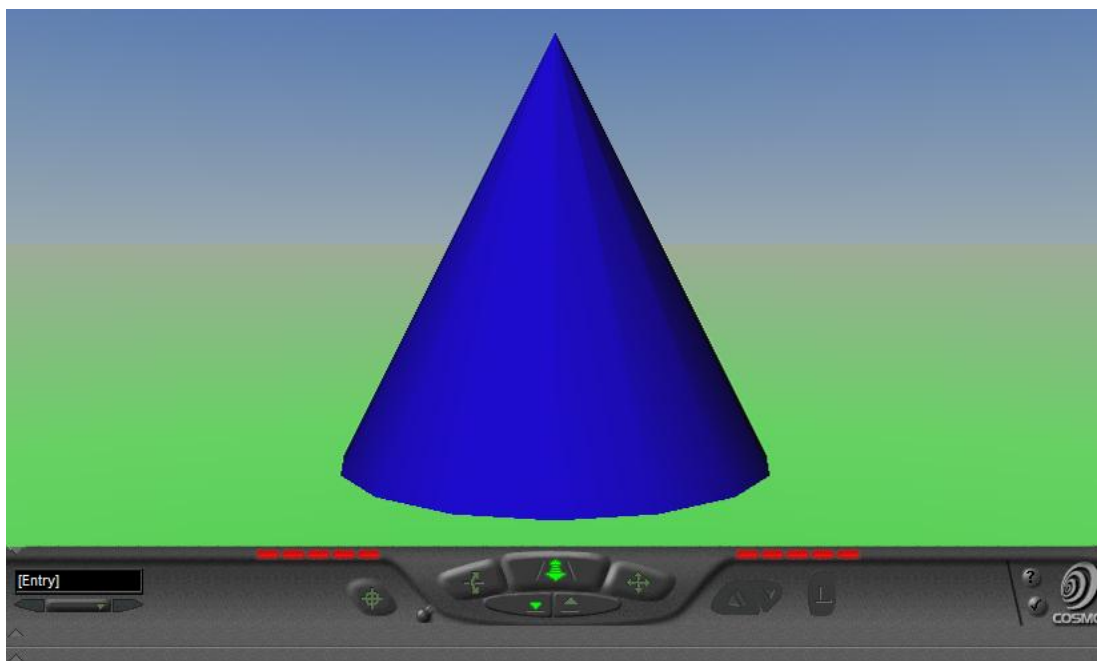
2 VIRTUÁLNÍ REALITA

Pod pojmem virtuální realita je potřeba si představit prostředí, jež umožňuje pracovat v trojrozměrném prostoru vymodelovaném na počítači. Základem jsou postupy obdobné jako v počítačové grafice. Jedná se o vytváření modelů, jejich pohyb ve scéně a zobrazení v reálném čase. Pro umocnění pocitu, že se člověk nachází ve virtuálním světě, jsou vyvinuta speciální zařízení, jako např. různé helmy s displeji, speciální brýle, dotykové ovladače a další. Lze se obejít i bez těchto zařízení, avšak za cenu menšího prožitku. Vystačíme si s obyčejným monitorem počítače, klávesnicí a myší.

Pro popis virtuálních světů slouží jazyk VRML (Virtual Reality Modeling Language). Počátek vývoje jazyka VRML se datuje k roku 1995, kdy vznikla verze VRML 1.0. V roce 1996 vznikla verze 2.0, která však byla v průběhu roku neustále upřesňována a měněna. V roce 1997 získává jazyk jméno VRML 97. Ke konci roku 1997 byl jazyk přijat za ISO standard s označením ISO/IEC 14772-1:1997. Jak je vidět, tak v průběhu vývoje jazyka existovalo několik verzí, které však nebyly jednotné. I to byl jeden z důvodů, proč se v počátcích jazyk moc neuchytil. Intenzivnějšího využití se dosáhlo právě až po sjednocení a zavedení normy, která předepisuje strukturu jazyka VRML. Jazyk VRML je zároveň jakýmsi typem formátu a lze ho přirovnat k obrazovým nebo video formátům.

V době, kdy jazyk vznikl, způsobil převrat v prezentování prostorových dat. Do té doby byla práce s prostorovými daty výhradou drahým a složitým systémům, tzn. ne vždy dobře dostupných. Hlavním využitím mělo být prezentování na internetu, resp. www stránkách. V dnešní době se však člověk na internetu s jazykem VRML téměř nepotká. Jeho technické možnosti jsou dávno překonány.

Soubory virtuální reality je možné zobrazit pomocí prohlížečů. Ty uživateli umožní nejenom prohlédnutí virtuální scény, ale také pohyb v ní. Jelikož mělo být využití VRML především na internetu, tak je velké množství VRML prohlížečů součástí prohlížečů www stránek. Mezi zástupce prohlížečů patří např. Cosmoplayer, WorldView nebo Cortona. V rámci této diplomové práce je virtuální svět zobrazován pomocí prohlížeče, který je součástí toolboxu 3D Animation v programu Simulink. [2; 9]



Obr. 2.1 Vzhled prohlížeče Cosmoplayer



2.1 STRUKTURA VRML

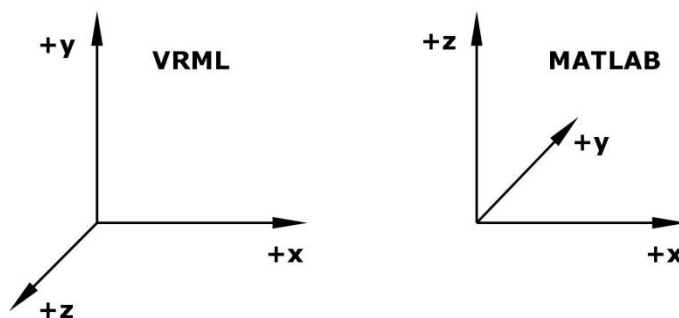
Každý soubor virtuální reality se skládá z několika částí. V úvodu (na prvním řádku) se nachází neměnná hlavička souboru. Obsahuje informace o typu souboru, použité verzi jazyka a způsob kódování znaků. Za hlavičkou následují vlastnosti popisující virtuální svět. Nejrozsáhlejší část popisuje samotná virtuální tělesa a jejich definice. Propojení statických a dynamických prvků je obsahem poslední části. Soubory mohou obsahovat také tzv. komentáře. Ty se uvozují znakem #. Vše za tímto znakem na stejném řádku je bráno pouze jako poznámka (s výjimkou hlavičky). Hlavní části VRML souboru jsou uvedeny v tabulce 2.1.

Tab. 2.1 Členění VRML souboru [9]

#VRML V2.0 utf8	hlavička souboru VRML
WorldInfo { ... } ViewPoint { ... }	úvodní, všeobecné informace o virtuálním světě
Transform { ... } Group { ... } Interpolator { ... }	popis těles, jejich vlastností, definice prvků potřebných pro animace a interakci
ROUTE ... TO ...	propojení dynamických a statických prvků z předchozí části

2.2 SOUŘADNÝ SYSTÉM

Ve virtuálním světě se používá pravotočivý systém souřadnic. Orientace jednotlivých os je znázorněna na obrázku 2.2. Vodorovnou rovinu tvoří osy X a Z, kde kladný směr osy X míří doprava a kladný směr osy Z směrem k nám. Svislý pohyb se uskutečňuje v ose Y, kde směr nahoru značí kladný směr osy Y. V této práci se bude vozidlo pohybovat v rovině XZ. Jednotlivé hodnoty a parametry se zadávají dle normy ISO, tzn. vzdálenost je udávána v metrech, úhly v radiánech a čas v sekundách. Orientace os ve virtuálním světě je odlišná od orientace, kterou používá program Matlab a Simulink. Na to je potřeba pamatovat později při tvorbě scény, případně při tvorbě signálů pro řízení modelu. [6; 9]



Obr. 2.2 Orientace souřadných systémů



2.3 INFORMACE O VIRTUÁLNÍM SVĚTĚ

2.3.1 VIEWPOINT

Ve virtuální realitě je možné pozorovat scénu z různých stanovišť. Nejdříve musí být však definováno a to pomocí příkazu Viewpoint. Ve scéně lze vytvořit několik takových stanovišť a lze se mezi nimi libovolně přepínat. První definované stanoviště se označuje jako výchozí a právě na toto místo je umístěn návštěvník – avatar při otevření souboru virtuální reality. V tomto uzlu se nastavuje poloha pomocí rozměrů XYZ, směr pohledu (orientace), šířka záběru a další viz tabulka 2.2. Defaultně je pohled avatara nastaven směrem do záporné osy Z. [2; 9]

Tab. 2.2 Parametry uzlu Viewpoint [9]

fieldOfView	1.57	šířka záběru, $\pi/2 = 180^\circ$
jump	TRUE	způsob přechodu na jiné stanoviště - skokově
orientation	0 1 0 1.57	směr pohledu; první tři čísla udávají směr osy (zadaný souřadnicemi x, y, z), kolem které se natáčíme, poslední číslo udává hodnotu natočení
position	1 2 3	poloha ve scéně udávaná souřadnicemi x, y, z [m]
decription	PRVNÍ	název stanoviště

2.3.2 NAVIGATIONINFO

V tomto uzlu se definují vlastnosti a parametry avatara. Tyto vlastnosti jsou podobné člověku. Jednotlivé zadávané parametry jsou uvedeny v tabulce 2.3. U parametru *avatarSize* se zapisují 3 rozměry: maximální povolená vzdálenost od překážky, výška očí nad terénem (to, na čem zrovna avatar stojí), maximální výška překročitelné překážky.

Tab. 2.3 Parametry uzlu NavigationInfo[9]

avatarSize	[0.25,1.6,0.75]	rozměrové charakteristiky avatara
headlight	TRUE	čelní světlo - zapnuto
visibilityLimit	0.0	dohled avatara [m], hodnota nula - nekonečno
speed	1.0	rychlost pohybu [$\text{m}\cdot\text{s}^{-1}$]
type	[WALK,ANY]	nastavení způsobu pohybu avatara

Parametr *type* je seznam povolených způsobů ovládání avatara. Existují tři základní metody:

- WALK – běžná chůze, pohyb po podložce, působí gravitace ve směru -Y, jsou zapnuty kolize avatara s předměty
- FLY – stejné jako WALK, avšak bez působení gravitace
- EXAMINE – vhodné pro zkoumání objektů, nepůsobí gravitace, parametr speed se nebere v úvahu, kolize jsou vypnuty

Doplňujícími možnostmi jsou pak způsoby ANY a NONE. [9]



2.3.3 WORLDINFO

Jedná se o informativní uzel. Zvykem je umisťovat ho na začátek zdrojového textu (kódu). Obsahuje dva parametry – *title* a *info*. První parametr smí obsahovat pouze jeden textový řetězec, druhý jich může obsahovat více.

Tab. 2.4 Parametry uzlu WorldInfo [9]

title	Diplomová práce	název souboru VRML
info	Autor: Datum: ÚADI FSI VUT BRNO	další údaje

2.4 VYTVOŘENÍ ILUZE PROSTORU

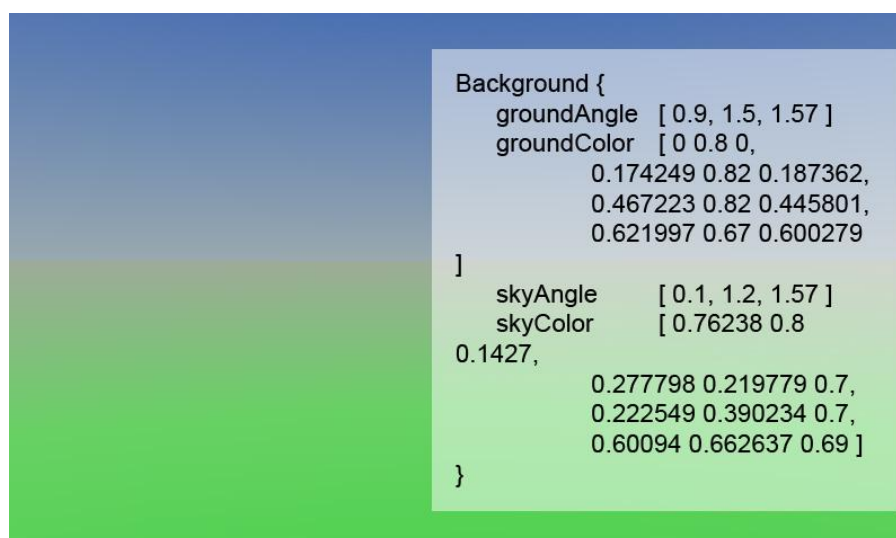
Abychom vytvořili iluzi skutečného trojrozměrného světa, musí být ve scéně vloženy kromě samotných objektů také další prvky. Jedná se především o pozadí a zdroje světla. Bez světla toho totiž, stejně jako v reálném světě, není moc vidět.

2.4.1 POZADÍ

Defaultně je pozadí virtuálního světa nastaveno jako černé. Toto umožňuje změnit uzel *Background*. Dovoluje nastavit pozadí jednobarevné až po barevný přechod měnící se s výškou nad zemí. Pozadí je zcela samostatný prvek a není ovlivňován světelnými zdroji. Pozadí může být vnímáno jako obrovské těleso, uvnitř kterého se nachází avatar. Z geometrického hlediska může být takovým tělesem koule nebo krychle. Koule je používána pro jednoduché pozadí, např. navození iluze oblohy. Krychle je vhodná pro zobrazení panoramatu. Na každou ze šesti stran krychle je nutné zevnitř nanést příslušnou texturu. Obrázky lze použít v běžném obrazovém formátu *.jpg. Jejich rozlišení by mělo být pokud možno co nejvyšší. [9]

Tab. 2.5 Parametry uzlu Background [9]

skyColor	0 0 0	seznam barev pro přechody na sférické obloze
skyAngle	[]	rostoucí posloupnost úhlů pro barvy oblohy
groundColor	[]	seznam barev pro postupné přechody na sférické zemi (podlaze)
groundAngle	[]	rostoucí posloupnost úhlů pro barvy země
frontUrl, leftUrl, rightUrl, backUrl, topUrl, bottomUrl	[]	seznamy adres s umístěním obrázků pro jednotlivé stěny obklopující krychle



Obr. 2.3 Vzhled pozadí a jeho zápis ve VRML

2.4.2 BODOVÝ ZDROJ SVĚTLA

Ve VRML existují čtyři druhy zdrojů světla: již zmíněná čelní svítidla avatara, DirectionalLight (rovnoběžné světelné paprsky), PointLight (bodové světlo) a SpotLight (reflektor, směrový zdroj). S výjimkou čelní svítidly smí virtuální scéna obsahovat více světelných zdrojů. Podmínkou však je, že nesmí být sdruženy do skupin, ale musí být vytvořeny samostatně. V diplomové práci jsem použil bodové zdroje světla pro osvětlení testovací plochy.

U bodového zdroje se nastavuje barva vyzařovaného světla, jeho intenzita, vypnutí nebo zapnutí, poloha zdroje v prostoru, dosah paprsků, útlum se zvyšující se vzdáleností od zdroje. Jednotlivé parametry jsou uvedeny v tabulce 2.6. [9]

Tab. 2.6 Parametry uzlu PointLight [9]

color	1 1 1	barva paprsků v paletě RGB
intensity	1	intenzita paprsků, rozsah 0÷1
ambientIntensity	0	příspěvek zdroje k nepřímému osvětlení světa, rozsah 0÷1
on	TRUE	rostoucí posloupnost úhlů pro barvy země
location	10 10 5	definování polohy světla
radius	200	nastavení dosahu paprsků
attenuation	1 0 0	koeficienty pro výpočet útlumu světla



2.5 VYTVÁŘENÍ OBJEKTŮ

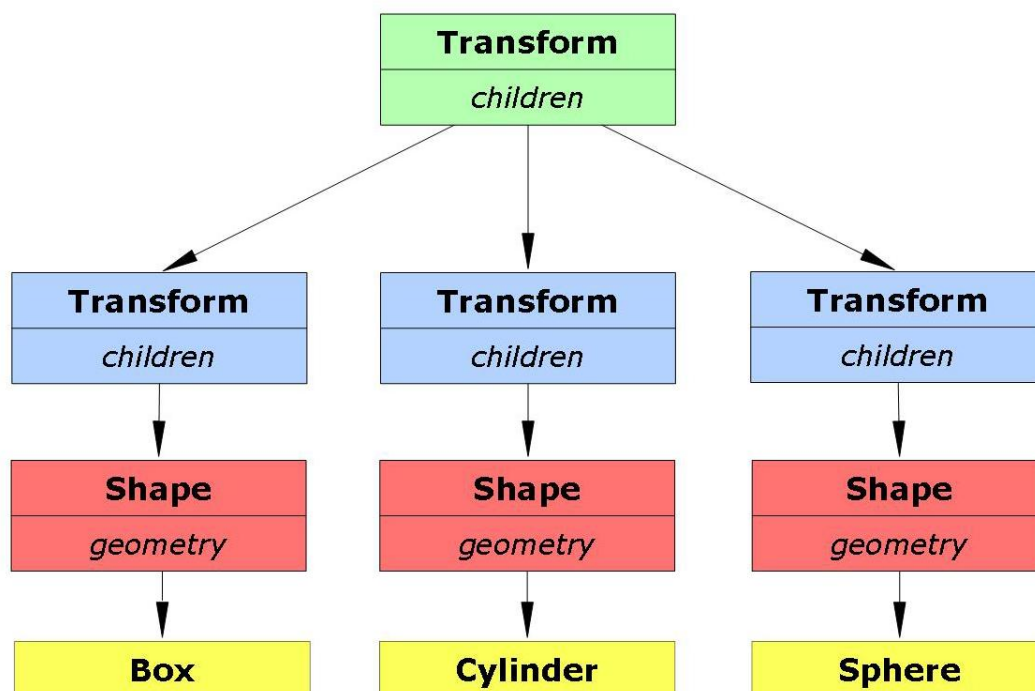
V této části bude vysvětleno jakým způsobem a z jakých těles a prvků lze skládat umělé světy. Na úvod je důležité zmínit, že v roli parametrů uzlů se mohou objevovat jiné uzly. To znamená, že tvůrce může z uzlů vytvářet tzv. stromovou strukturu. Podle polohy uzlů ve struktuře se pak mluví o vztazích rodič – potomek – sourozenec.

Než budou představena základní tělesa, je potřeba se seznámit s rodičovským (příp. skupinovým) uzlem *Transform (Group)*. Pomocí tohoto uzlu lze provádět transformace potomků, kteří jsou zapsáni v parametru *children*. [2; 9]

Tab. 2.7 Seznam základních parametrů uzlu *Transform* [9]

Transform {	scale	1 1 1	nastavení měřítka
	rotation	0 1 0 0	určení osy otáčení a velikosti natočení
	translation	0 0 0	posunutí
	children }		seznam potomků

V případě, kdy měníme polohu rodiče (uzlu *Transform*), mění se zároveň poloha všech jeho potomků (*children*). Potomky uzlu *Transform* mohou být další uzly *Transform*. To se s výhodou využívá při tvorbě objektů složených z více částí nebo pro postupné transformace. Jako příklad poslouží situace, kdy potřebujeme objekt např. nejdříve pootočit a následně posunout. Je však nutné dodržet pořadí těchto transformací, protože kdybychom nejdříve potomka posunuli a následně pootočili, pravděpodobně bychom nedostali žádanou polohu a orientaci.



Obr. 2.4 Znárodnění stromové struktury



2.5.1 ZÁKLADNÍ TĚLESA

V nabídce „stavebních prvků“ najdeme pouze čtyři základní tělesa. Jsou to koule, kvádr, válec a kužel. Při vložení do scény jsou všechna tato tělesa defaultně umístěna tak, že jejich těžiště leží v počátku souřadného systému. Parametry těles a jejich defaultní hodnoty ukazuje tabulka 2.8. [9]

Tab. 2.8 Parametry základních těles [9]

těleso	uzel	parametr	defaultní hodnota	význam
koule	Sphere { }	radius	1	poloměr
kvádr	Box { }	size	2 2 2	délka stran ve směru x,y,z
válec	Cylinder { }	radius height	1 2	poloměr výška
kužel	Cone { }	bottomRadius height	1 2	poloměr podstavy výška

U kužele a válce lze, v případě, kdy to není vyžadováno, vypnout viditelnost spodní, resp. horní podstavy. Snižují se tím nároky na výkon počítače a tento fakt také urychluje manipulaci s tělesy.

2.5.2 OBECNÁ TĚLESA

Je jasné, že při tvorbě virtuálních světů si stěží vystačíme se čtyřmi základními tělesy. Dalšími doplňujícími uzly jsou:

- *IndexedFaceSet* – množina ploch
- *IndexedLineSet* – množina čar
- *Extrusion* – opláštění
- *ElevationGrid* – výšková mapa
- *Text*
- *PoinSet* – množina bodů

Pomocí těchto prvků se dají vymodelovat téměř libovolné tvary. Struktura je již komplikovanější než u základních těles. V diplomové práci jsem použil množiny ploch pro zobrazení vozidla a množiny čar pro zobrazení dynamických účinků.

2.5.3 INDEXEDFACESET

Tento uzel skrývá velké možnosti, neboť dovoluje vytvářet i velmi složité tvary. Zapisuje se do parametru *geometry* uzlu *Shape*. Základem je rozdělení objektu na malé plochy. Abychom mohli množiny těchto ploch vytvořit, musíme znát souřadnice jednotlivých vrcholů ploch. Souřadnice jednotlivých bodů se zapisují do parametru *coord* *Coordinate*. Podle pořadí zápisu bodů získávají body indexy. První zapsaný bod má index 0, druhý 1, atd. Následně se v parametru *coordIndex* zapisují jednotlivé plochy pomocí zmíněných indexů. Každá plocha se ukončuje zápisem čísla -1. Plochy jsou chápány jako jednostranné a aby byly viditelné je třeba jednotlivé body zadávat proti směru hodinových ručiček. V opačném případě plocha viditelná nebude. To se však dá upravit nastavením parametrů *ccw* a *solid* na



hodnotu *FALSE*. Poté budou viditelné obě strany plochy. Detailnější popis a význam všech parametrů uzlu *IndexedFaceSet* uvádí Žára v [9]. Příklad zápisu krychle pomocí šesti ploch:

```

Transform {
  children Shape {
    geometry IndexedFaceSet {
      coord Coordinate {
        point [0 0 0, 1 0 0, 1 0 -1, 0 0 -1,
              0 1 0, 1 1 0, 1 1 -1, 0 1 -1]}
      normal NULL
      texCoord NULL
      ccw FALSE
      colorPerVertex FALSE
      convex TRUE
      normalPerVertex TRUE
      creaseAngle 0
      solid FALSE
      coordIndex [ 0,1,2,3,-1, 4,5,6,7,-1, 0,1,5,4,-1, 1,2,6,5,-1
                  2,3,7,6,-1, 3,0,4,7,-1]
      colorIndex 1
    }
  }
}

```

2.6 NASTAVENÍ VZHLEDU OBJEKTŮ

Vytvořené objekty se bez dalšího nastavování zobrazují jako matná, světle šedá tělesa. Jejich vzhled se nastavuje v parametru *appearance*, který spadá pod uzel *Shape*. Potomkem tohoto parametru je stejnojmenný uzel. Vzhled je možné upravovat dvěma způsoby.

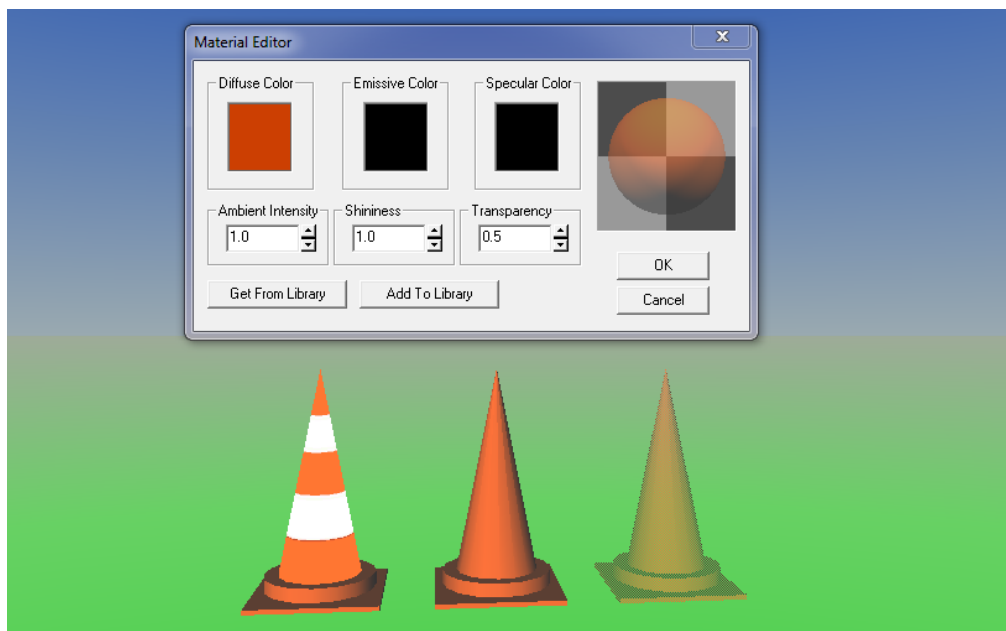
První možností je obarvení tělesa jednou barvou. To se provádí v parametru *material*, konkrétně v jeho potomku – uzlu *Material*. Ačkoliv by název mohl napovídat, že slouží ke zvolení materiálu tělesa, je určen pouze k zadání jeho barvy. Barevných charakteristik se zde vyskytuje hned několik a jejich základní popis je uveden v tabulce 2.9. Parametry mohou nabývat hodnot mezi nulou a jedničkou. [9]

Tab. 2.9 Parametry pro nastavení barvy tělesa [9]

parametr	defaultní hodnota	význam
diffuseColor	0.8 0.8 0.8	základní složení barvy povrchu, složky RGB
ambientIntensity	0.2	jak je barva povrchu zesvětlována jasem prostoru
specularColor	0 0 0	jakou barvu dopadajícího světla povrch odráží
shininess	0.2	ostrost odrazu pro předchozí parametr
emissiveColor	0 0 0	fluorescenční (svítivá) barva povrchu
transparency	0	průhlednost (0 plná viditelnost, 1 těleso zmizí)



Druhou možností je pokrytí povrchu tělesa texturou - vzorkem. Na povrch může být nanesen libovolný obrázek v příslušném obrazovém formátu. Tělesu tak lze dodat přirozený vzhled. Nanášení textury se provádí pomocí uzlu *ImageTexture* parametru *texture*, kde je potřeba vybrat soubor s obrázkem. Další operace s texturou se provádí v uzlu *TextureTransform*, kde se definuje posunutí, natočení a měřítko textury. Dále je možné nastavit opakování se obrázkem ve vodorovném a svislém směru. [9]



Obr. 2.5 Možnosti nastavení vzhledu kužele

2.7 POJMENOVÁNÍ UZLŮ

Ve scéně se mohou vyskytovat objekty, které se několikrát opakují (např. kužely). Nemusí se však jednat pouze o tělesa, platí to například také pro nastavení vzhledu, atd. V těchto případech se doporučuje pojmenovat příslušné uzly příkazem *DEF jméno*. Jakmile máme takto pojmenovaný uzel, lze v další části stromové struktury vytvořit jeho duplikát pomocí příkazu *USE jméno*. Využitím tohoto postupu zmenšujeme velikost vytvořeného souboru a také tento přístup není oproti neustálému vytváření stejných věcí tolik pracný. [9] V případě této diplomové práce je pojmenování uzlů nutností, neboť v další části je potřeba ovládat správné uzly v prostředí Simulinku. Těžko bychom se orientovali mezi desítkami nepojmenovaných uzlů *Transform*.



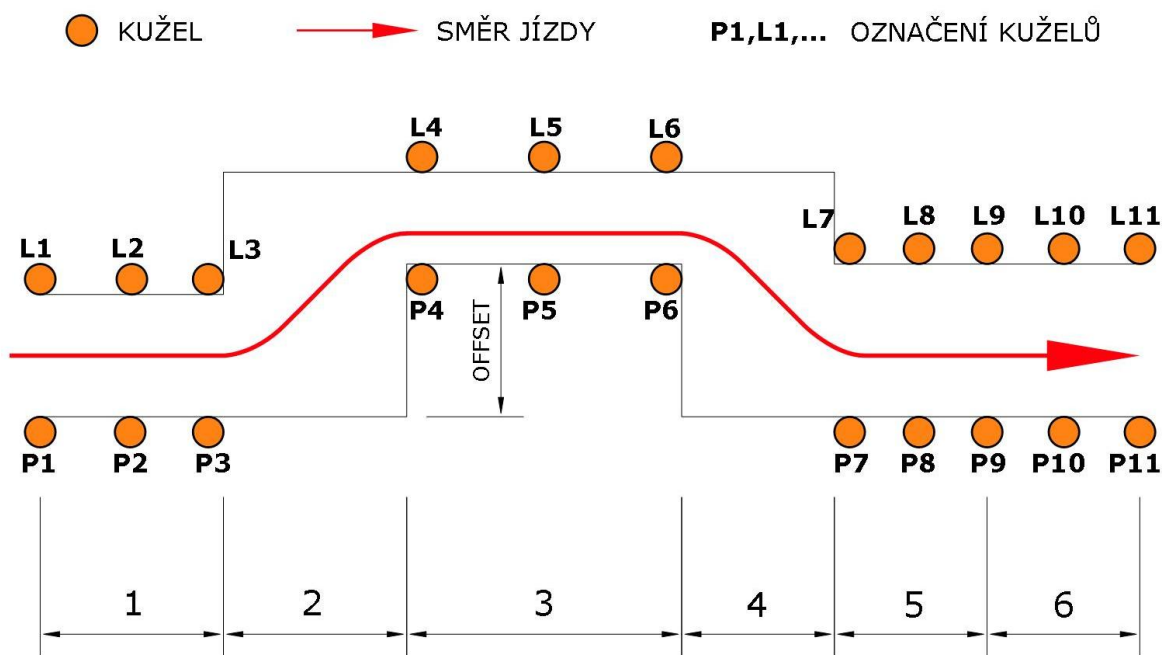
3 TESTY PRO OSOBNÍ VOZIDLA DLE NORMY ISO

V následující části budou popsány principy a příklady testů pro vyšetřování a hodnocení dynamiky osobních automobilů dle normy ISO 3888. Tato norma se skládá ze dvou částí: dvojitá změna jízdního pruhu a vyhýbání se překážce. Testovací procedury obou částí jsou uvedeny níže. Vzhledem k tomu, že testy dle zmíněné normy kvantifikují pouze malou část kompletních charakteristik ovladatelnosti vozidla, nelze získané výsledky testů použít pro legislativní účely.

3.1 DOUBLE LANE-CHANGE TEST – ISO 3888-1

V tomto oddílu ISO 3888 jsou stanoveny rozměry zkušební dráhy pro zkušební metodu s uzavřenou smyčkou k subjektivnímu zjištění chování při dvojitě změně jízdního pruhu, které je jednou z částí dynamiky a ovladatelnosti osobního vozidla. Test je použitelný pro osobní automobily a lehká užitková vozidla do 3,5 tuny. Informace v další části převzaty z [4].

Na obrázku 3.1 je znázorněna testovací trať, rozdělení na jednotlivé úseky, rozmístění a označení vytyčovací kuželů. Délky úseků jsou pevně dány a celková délka tratě je 125 metrů. Šířka úseků je funkcí šířky vozidla. Šířkou je myšlena celková šířka vozu bez zpětných zrcátek. Rozměry tratě jsou uvedeny v tabulce 3.1.



Obr. 3.1 Rozdělení tratě na úseky, rozmístění kuželů



Tab. 3.1 Rozměry tratě [4]

ÚSEK	DÉLKA [m]	OFFSET [m]	ŠÍŘKA [m]
1	15	-	1,1 · šířka vozidla + 0,25
2	30	-	-
3	25	3,5	1,2 · šířka vozidla + 0,25
4	25	-	-
5	15	-	1,3 · šířka vozidla + 0,25
6	15	-	1,3 · šířka vozidla + 0,25

Test spočívá v rychlé změně směru jízdy z jednoho jízdního pruhu do druhého (paralelního k prvnímu) a následném vrácení se do původního pruhu bez překročení hranic tratě. Původně byl jediným hodnoticím kritériem čas potřebný k projetí dráhy. Toto se později ukázalo jako nedostatečné a musely být zahrnuta další kritéria spolu se subjektivním hodnocením řidiče. Detailní vyhodnocení všech možných variant testů poukázalo na určité problémy:

- I když byl test vyvinut pro testování boční dynamiky, zjistilo se, že na hodnocení má velký vliv také podélná dynamika a tento fakt vede k rozptylu ve výsledcích
- Odstranění vlivů podélné dynamiky nevedlo k požadovaným výsledkům v korelaci mezi naměřenými hodnotami a subjektivním hodnocením řidičů

Provedení testu by mělo být umožněno pouze zkušenému řidiči. Průjezd tratí je považován za úspěšný, pokud nedojde k vychýlení žádného z umístěných kuželů ze svého místa. V protokolu ze zkoušky musí být zaznamenáno, na jaký převodový stupeň se vozidlo při testu pohybovalo. Měření vzdálenosti (např. pro stanovení průměrné rychlosti) začíná na začátku úseku 1 a končí na konci úseku 5.

Existují různé modifikace testu. Při každé lze hodnotit jiné veličiny, vždy však musí být ve zprávě uvedeny konkrétní podmínky testu. Níže jsou popsány 2 možné příklady.

3.1.1 TEST 1

Doporučená nájezdová rychlost do úseku 1 je $(80 \pm 3) \text{ km} \cdot \text{h}^{-1}$. Je možné použít vyšší nebo nižší rychlosti, avšak je potřeba je zaznamenat do protokolu. Během pokusu se snažíme držet plynový pedál pokud možno ve stabilní poloze. Mohou být aplikována také další omezení, např. co se řízení týká. Měří a zapisuje se rychlost na konci úseku 5. Tato varianta má několik typických použití:

- Hodnocení, posouzení řízení nebo pohybu vozidla v závislosti na čase
- Hodnocení řidičovy kontroly nad vozem
- Subjektivní hodnocení vozidla



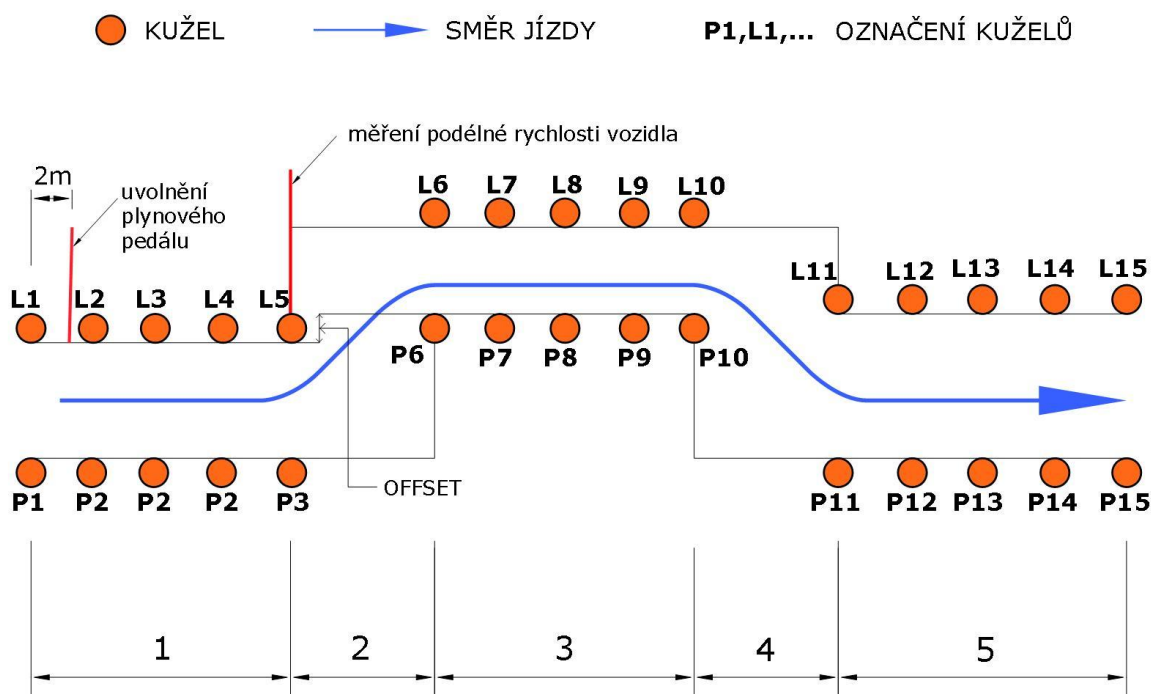
3.1.2 TEST 2

Nájezdová rychlost do úseku 1 je maximální možná, tak aby byl pokus úspěšně dokončen. Jsou možné dvě alternativy polohy plynového pedálu (škrticí klapky), zvolená alternativa musí být opět zaznamenána v protokolu.

3.2 MANÉVR ISO 3888-2

Tento manévr pro změnu jízdního pruhu, původně známý jako tzv. „losí“ test, byl po zpracování a revizi Asociací německého automobilového průmyslu (VDA) převeden na mezinárodní normu ISO 3888-2. Simuluje náhlé vyhnutí se překážce a slouží k hodnocení ovladatelnosti vozidla. Je nedílnou součástí procesu návrhu vozidla a jeho hodnocení.

Nákres testovací trati je na obrázku 3.2, rozměry pak v tabulce 3.2. Celková délka je 61 metrů.



Obr. 3.2 Nákres trati pro manévr ISO 3888-2

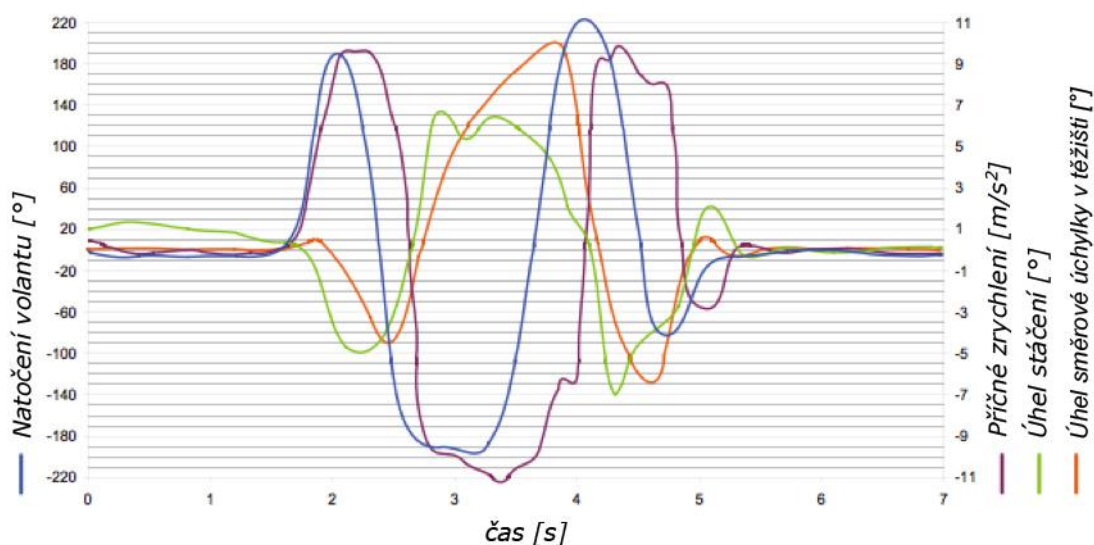
Tab. 3.2 Rozměry dráhy [5]

ÚSEK	DÉLKA [m]	OFFSET [m]	ŠÍŘKA [m]
1	12	-	1,1 · šířka vozidla + 0,25
2	13,5	-	-
3	11	3	šířka vozidla + 1
4	12,5	-	-
5	12	-	1,3 · šířka vozidla + 0,25; nejméně však 3m



Opět by měl manévr provádět zkušený řidič. Jeho úkolem je co nejrychleji projet vytyčenou dráhu bez posunutí kuželů nebo, v horším případě, bez převrácení vozidla. Vozidlo vjíždí na trať na nejvyšší rychlostní stupeň (v případě automatické převodovky poloha D), který zaručuje minimální otáčky motoru 2000 min⁻¹. Po ujetí vzdálenosti 2 metrů řidič uvolní plynový pedál a takto pokračuje vozidlo dále. Měří se rychlost na konci úseku 1 a 5, viz obrázek 3.2. Hodnotí se dosažené rychlosti v jednotlivých bodech a styl projetí trati. Nájezdová rychlost se postupně zvyšuje.

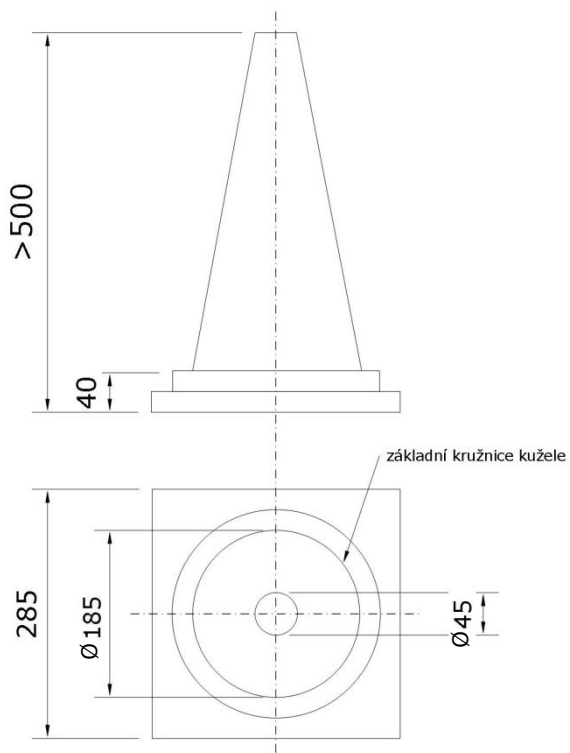
Vzhledem k velkému vlivu řidiče lze výsledky testu použít pouze k subjektivnímu hodnocení. V současné době existují a používají se prostředky eliminující vliv řidiče. Jedná se o systémy, které zcela automaticky řídí celý jízdni manévr. Řídicí robot a řídicí systém dokáže generovat požadovanou rychlost natočení volantu a požadované příčné zrychlení vozidla, které jsou pro zdolání vysoce dynamického testu změny jízdniho pruhu nezbytné. [5; 12; 18]



Obr. 3.3 Příklad měřených dat [12]

3.3 VYTYČENÍ TESTOVACÍ TRATI

V obou případech uvedených testů jsou pro vytyčení testovací trati použity kužely. Požadavky na kužely jsou popsány v normě ISO 3888 [5]. Minimální výška kužele musí být 500 mm. Další rozměry nalezneme na obrázku 3.4. Uvedené hodnoty jsou vyjádřeny v milimetrech. Kužely se umísťují dle obrázku 3.1 a 3.2 tak, aby základní kružnice kužele ležela tečně k okraji dráhy.

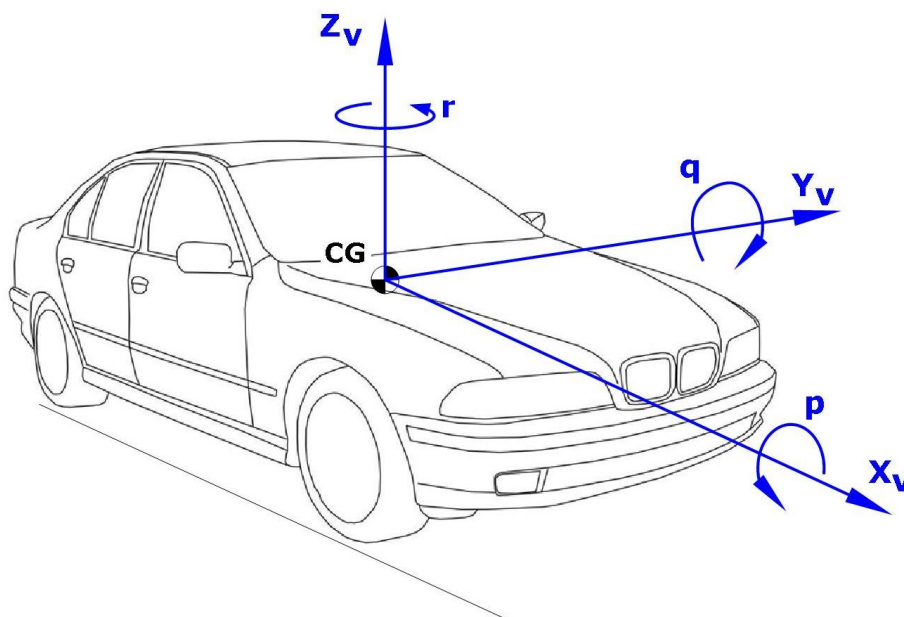


Obr. 3.4 Rozměry kužele



4 DYNAMIKA VOZIDLA

4.1 SOUŘADNÝ SYSTÉM VOZIDLA



Obr. 4.1 Souřadný systém vozidla dle ISO

Ve vozidle jsou pohyby definovány vzhledem k pravotočivému pravoúhlému souřadnému systému, který vychází z těžiště vozu a pohybuje se s ním. Popis obrázku 4.1:

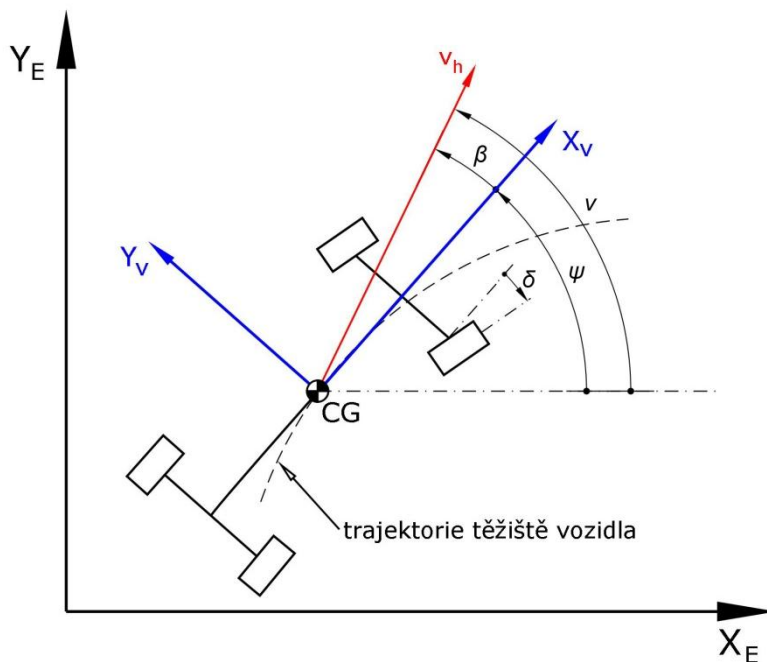
CG	těžiště vozidla
X_v	podélná osa (směr) vozidla
Y_v	příčná osa (směr) vozidla
Z_v	vertikální osa (směr) vozidla
p	rychlost klopení kolem osy X
q	rychlost klonění kolem osy Y
r	rychlost stáčení kolem osy Z

4.2 GLOBÁLNÍ SOUŘADNÝ SYSTÉM

Pozice vozidla a jeho trajektorie je při jízdě definována vzhledem ke globálnímu pravotočivému pravoúhlému souřadnému systému na Zemi. Pozice tohoto systému se volí tak, že na začátku testu nebo manévru jsou globální a souřadný systém vozidla shodné. Na obrázku 4.2 jsou znázorněny jednotlivé souřadnice. Kladný směr osy X znázorňuje pohyb vozidla dopředu, kladný směr osy Y pohyb doleva a kladný směr osy Z pohyb vertikální směrem nahoru. Úhel mezi osami X_E globálního SS a osou X_v SS vozidla se nazývá úhel stáčení vozidla Ψ (v obrázku kladný). Úhel mezi osou X_E globálního SS a vektorem rychlosti vozidla je směrový úhel ν (v obrázku kladný). Úhel mezi osou X_v souřadného systému vozidla a vektorem rychlosti je nazýván úhlem směrové úchylky β (v obrázku kladný). Úhel δ znázorňuje úhel natočení kol (v obrázku záporný). Tento úhel je pro



každé kolo jiný v důsledku Ackermanovy geometrie řízení. Vnitřní kolo se vždy natočí v zatáčce více než kolo vnější. [3]



Obr. 4.2 Vozidlo v globálním souřadném systému

4.3 VELIČINY TRAJEKTORIE VOZIDLA

4.3.1 RYCHLOST VOZIDLA V HORIZONTÁLNÍ ROVINĚ

Horizontální rychlost můžeme rozložit na podélnou a boční složku. Vypočítá se ze vztahu [10]:

$$v_h = \sqrt{v_x^2 + v_y^2}, \quad (1)$$

kde:

v_h	$[\text{m} \cdot \text{s}^{-1}]$	rychlost v horizontálním směru
v_x	$[\text{m} \cdot \text{s}^{-1}]$	podélná rychlost vozidla
v_y	$[\text{m} \cdot \text{s}^{-1}]$	boční (příčná) rychlost vozidla

4.3.2 ÚHEL SMĚROVÉ ÚCHYLKY V TĚŽIŠTI VOZIDLA

$$\beta = \arctan \frac{v_y}{v_x}, [10] \quad (2)$$

kde:

β	[rad]	úhel směrové úchyly
---------	-------	---------------------



4.3.3 SMĚROVÝ ÚHEL

$$v = \psi + \beta, [10] \quad (3)$$

kde:

v	[rad]	směrový úhel
Ψ	[rad]	úhel stáčení vozidla

4.4 SÍLY NA KOLECH

4.4.1 PODÉLNÁ SÍLA

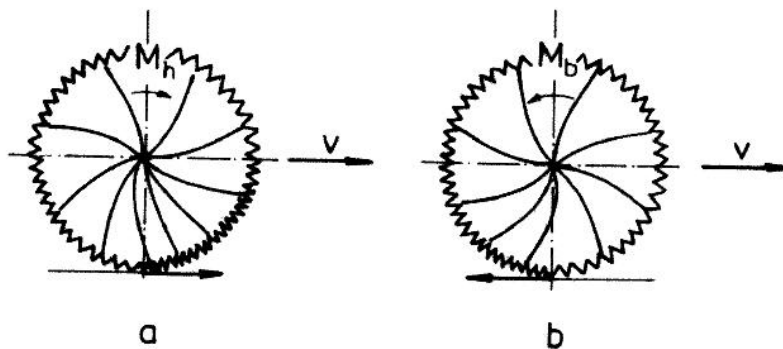
Podélná síla je dána několika parametry. Závisí především na povrchu, po kterém se vozidlo pohybuje, resp. na součiniteli tření mezi pneumatikou a povrchem. Dalším důležitým faktorem je zatížení kola. S rostoucím zatížením vzrůstá maximální podélná síla a zároveň klesá hodnota součinitele tření. Podélnou sílu ovlivňuje také rychlost, kdy se zvyšující se rychlostí klesá hodnota špičkového i kluzného součinitele tření. V dynamice silničních vozidel má významný vliv také skluz pneumatiky. Existují dva druhy skluzu – podélný a příčný. V případě, kdy se bavíme o podélné síle na kole, nás zajímá podélný skluz, který vzniká při akceleraci a brzdění. Skluz je definován jako relativní pohyb mezi pneumatikou a vozovkou, vyjadřuje se v procentech. Při ideálním odvalování kola k tomuto jevu nedochází. Při jízdě se však kola ideálně neodvalují a určité části pneumatiky neustále ve skluzu. [10; 20; 21]

Podélnou sílu lze vypočítat podle vztahu [10]:

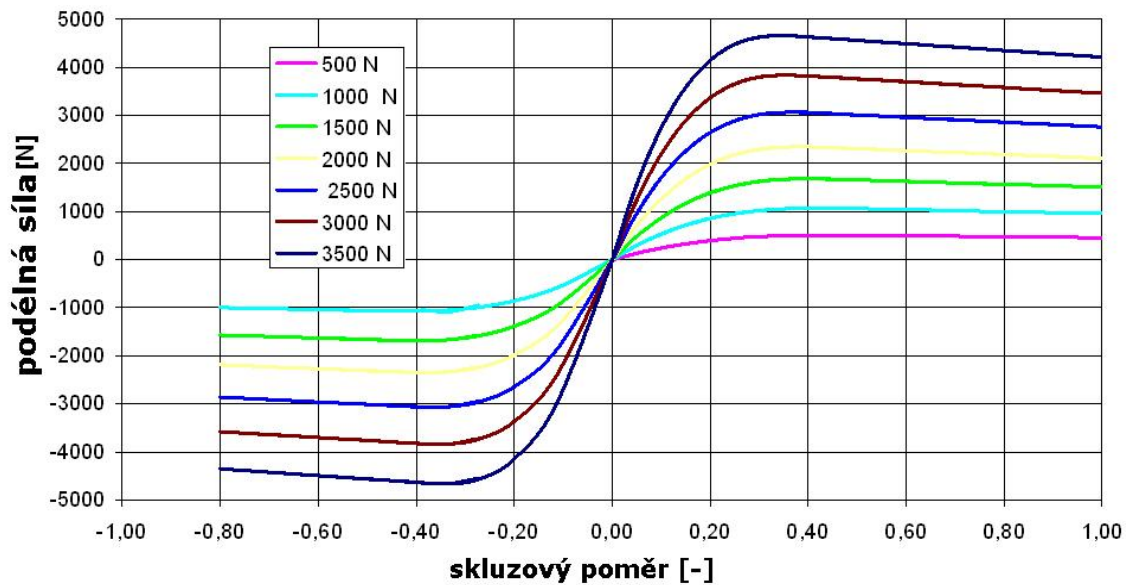
$$F_x = \mu \cdot F_z, \quad (4)$$

kde:

F_x	[N]	podélná síla působící na kolo
μ	[-]	součinitel tření
F_z	[N]	zatížení kola



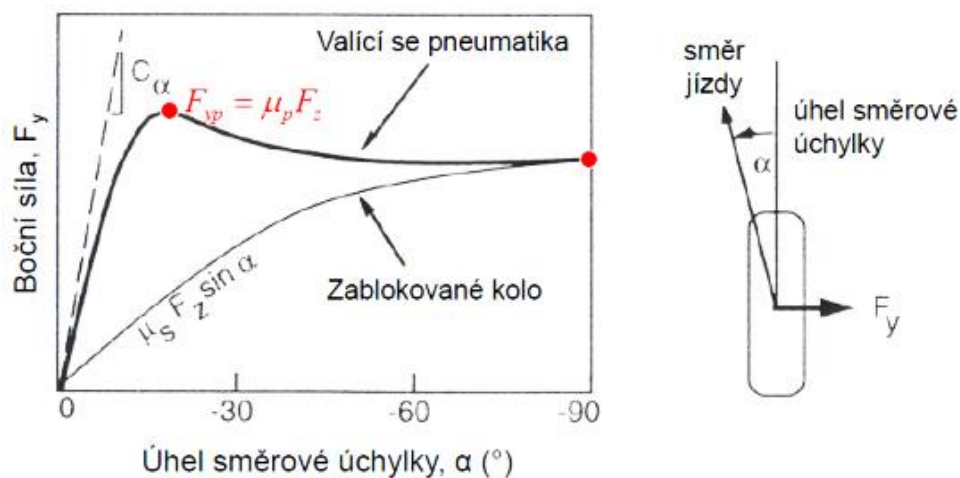
Obr. 4.3 Vznik skluzu: a) poháněné kolo, b) brzděné kolo [21]



Obr. 4.4 Podélná síla pro různá zatížení v závislosti na skluzu [11]

4.4.2 BOČNÍ SÍLA

Stejně jako podélná síla, tak i boční síla závisí na několika dalších okolnostech. Mezi ně patří směrová tuhost pneumatiky, konstrukce pneumatiky (diagonální, radiální), velikost zatížení (zvětšuje boční sílu, snižuje koeficient tření), tlak v pneumatice (má vliv až při velkém zatížení), rozměry pneumatiky, tvar běhounu. Menší vliv na boční sílu má rychlost vozidla a povrch vozovky. [3; 10]



Obr. 4.5 Závislost boční síly na úhlu směrové úchylky [10]



ÚHEL SMĚROVÉ ÚCHYLKY

Úhel směrové úchylky, označovaný α , vzniká při zatáčení vozidla. Je to úhel mezi rovinou otáčení kola a směrem, kterým se kolo pohybuje. Směrová úchylka je vyvolána hlavně vlivem pružnosti pneumatik a také působením odstředivé síly, bočního větru, vlivem naklonené vozovky, atd. Nevzniká pouze na kolech řízené nápravy, ale i na kolech ostatních. [15; 20]

SMĚROVÁ TUHOST PNEUMATIKY

Primární význam v ohledu na zatáčení a stabilní chování vozidla má počáteční sklon křivky znázorňující boční sílu. Sklon křivky při nulovém úhlu směrové úchylky je známý jako směrová tuhost C_α . Ta se vypočítá [10]:

$$C_\alpha = - \left. \frac{\partial F_y}{\partial \alpha} \right|_{\alpha=0}, \quad (5)$$

kde:

C_α	[N·deg ⁻¹]	směrová tuhost pneumatiky
F_y	[N]	boční síla
α	[°]	úhel směrové úchylky pneumatiky

Konvence je taková, že kladný úhel směrové úchylky vytváří zápornou boční sílu. Z toho vyplývá, že směrová tuhost musí být záporná. Díky tomu jsou na obrázku 4.5 vyneseny na vodorovné ose záporné hodnoty úhlu α . [3]

KOEFICIENT ZATÁČENÍ

Vypočítá se jako poměr směrové tuhosti k zatížení:

$$C_{coef} = \frac{C_\alpha}{F_z}, \quad (6)$$

kde:

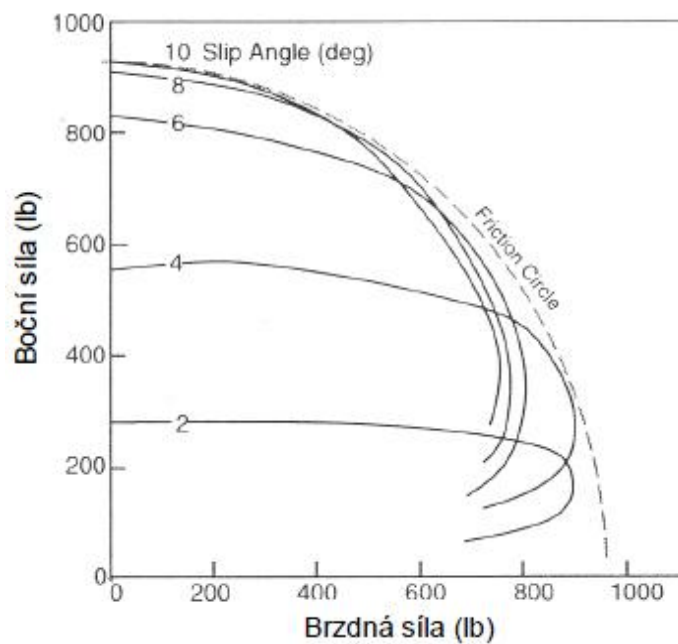
C_{coef}	[(N·deg ⁻¹)·N ⁻¹]	koeficient zatáčení
F_z	[N]	zatížení kola

4.4.3 TŘECÍ KRUŽNICE

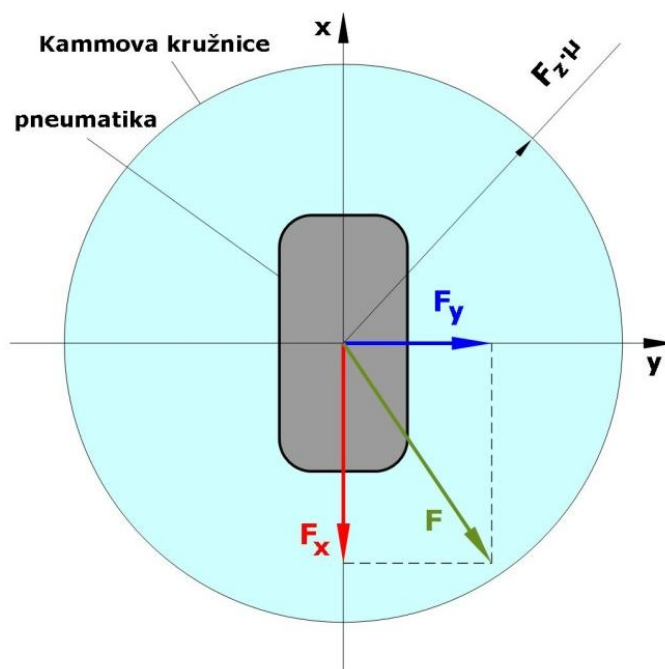
Základní a nutnou podmínkou pro bezpečnou jízdu vozidla je dobrá přilnavost pneumatik (kol) k vozovce, tzv. adheze. Adhezní limity pneumatiky lze znázornit pomocí třecí kružnice, známé také pod názvem Kammova kružnice. Na kolo působí při jízdě obvodové síly (hnací, brzdná), boční síla (při zatáčení, vlivem sklonu vozovky) a svislá síla (zatížení). Pokud působí na pneumatiku obvodové a boční síla zároveň, jejich velikosti se geometricky sčítají, viz vztah (7) [14], a vytvářejí výsledný vektor síly. Hranici přilnavosti vytváří kružnice, jejíž poloměr je dán součinem svislé síly a součinitelem tření (koeficientem přilnavosti) mezi vozovkou a pneumatikou. Překročí-li výslednice sil poloměr Kammovy kružnice, dojde ke ztrátě přilnavosti pneumatiky a dochází ke smýkání. V případě, kdy se kola



nachází ve smyku, vozidlo se stává neovladatelným. Skutečné pneumatiky mají podélnou přilnavost větší než boční. Hranici přilnavosti pak tvoří elipsa. [3; 14]



Obr. 4.6 Závislost boční síly na brzdné síle [10]



Obr. 4.7 Kammova kružnice přilnavosti



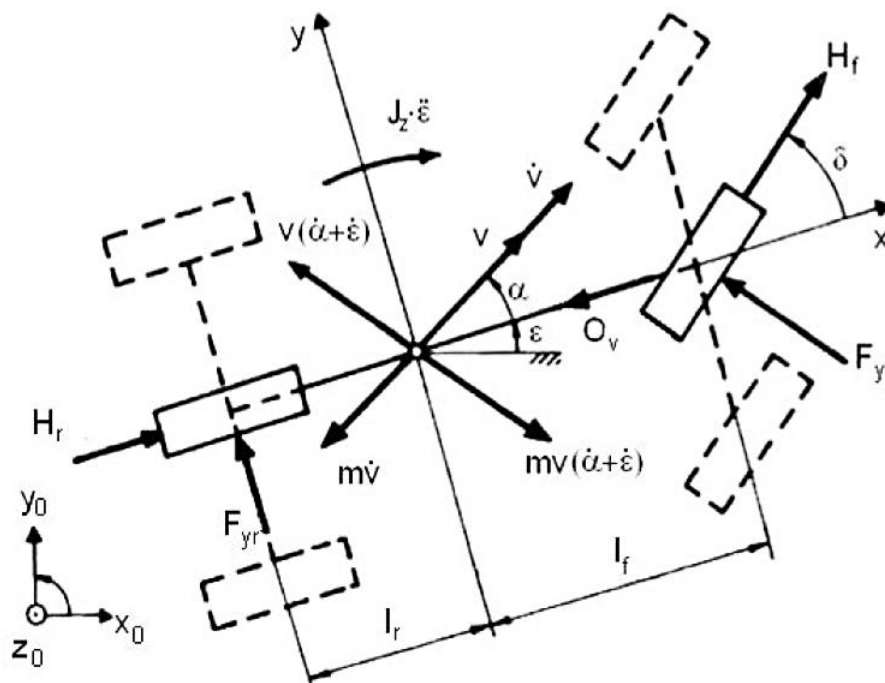
$$F_p = \sqrt{F_x^2 + F_y^2} \leq \mu_p \cdot F_z, [10; 14] \quad (7)$$

kde:

F_p	[N]	výslednice síly přenášené pneumatikou
F_x	[N]	podélná (obvodová) síla
F_y	[N]	zatížení kol
F_z	[N]	svislá síla (zatížení kola)
μ_p	[-]	součinitel tření mezi kolem a vozovkou

4.5 JEDNOSTOPÝ MODEL VOZIDLA

Informace pro popis jednostopého modelu vozidla čerpány z [1; 7; 19]. Popis a výpočet pohybu vozidla představuje náročný úkol. Existuje však několik zjednodušených modelů, které slouží pro simulaci dynamiky vozidel, jako např. lineární jednostopý rovinný model s řízeními předními koly, lineární jednostopý rovinný model se všemi řízenými koly, lineární prostorový model se všemi řízenými koly. Nejjednodušší ze všech modelů a v praxi poměrně často používaný je jednostopý lineární model vozidla. Přes svou relativní jednoduchost dokáže poskytnout dobré výsledky. Hlavními zjednodušeními jsou: těžiště se nachází v rovině vozovky (neuvažuje se tak změna zatížení kol vlivem klopení karoserie), natočení vnějšího a vnitřního kola je stejné. Ačkoliv například dvoustopý model poskytuje přesnější výsledky, je obtížnější na modelování a potřebuje více vstupních hodnot, které se navíc špatně měří nebo zjišťují.



Obr. 4.8 Jednostopý lineární model vozidla [7]



Tento model se využívá pro zjišťování stability a ovladatelnosti automobilu. Pohyb vozu se dá zapsat pomocí soustavy tří pohybových rovnic – ve směru osy X, ve směru osy Y a rovnováha momentů kolem osy Z. Linearizací (tj. uvažováním malých úhlů a dosazením za $\sin(x) \approx 0$, $\cos(x) \approx 1$) pak dostaneme tyto rovnice [1; 7]:

$$-m\dot{v} + H_f + H_r - O_v = 0 \quad (8)$$

$$-mv(\dot{\alpha} + \dot{\varepsilon}) + F_{yf} + F_{yr} = 0 \quad (9)$$

$$-J_z \ddot{\varepsilon} + F_{yf} \cdot l_f - F_{yr} \cdot l_r = 0 \quad (10)$$

kde:

m	[kg]	hmotnost vozidla
v	[m.s ⁻¹]	rychlost vozidla
\dot{v}	[m.s ⁻²]	derivace rychlosti
H _f	[N]	podélná síla na přední nápravě
H _r	[N]	podélná síla na zadní nápravě
O _v	[N]	vzdušný odpor
F _{yf}	[N]	boční síla na přední nápravě
F _{yr}	[N]	boční síla na zadní nápravě
$\dot{\alpha}$	[rad.s ⁻¹]	derivace úhlu směrové úchytky
$\dot{\varepsilon}$	[rad.s ⁻¹]	derivace úhlu stáčení (rychlost stáčení)
$\ddot{\varepsilon}$	[rad.s ⁻²]	úhlové zrychlení stáčení
J _z	[kg.m ²]	moment setrvačnosti ke svislé ose z
l _f	[m]	vzdálenost přední nápravy od těžiště
l _r	[m]	vzdálenost zadní nápravy od těžiště

Boční síly získáme součinem směrových tuhostí pneumatik a úhly směrových úchylek. Pokud budeme uvažovat jízdu vozidla konstantní rychlostí (tedy $\dot{v} = 0$), můžeme potom jeho pohyb vyjádřit pomocí soustavy dvou rovnic [1; 7]:

$$\left(mv + \frac{C_{\alpha f} \cdot l_f - C_{\alpha r} \cdot l_r}{v} \right) \varepsilon + mv\dot{\alpha} + (C_{\alpha f} + C_{\alpha r})\alpha - C_{\alpha f} \cdot \delta = 0 \quad (11)$$

$$J_z \ddot{\varepsilon} + \frac{C_{\alpha f} \cdot l_f^2 - C_{\alpha r} \cdot l_r^2}{v} \varepsilon + (C_{\alpha f} \cdot l_f + C_{\alpha r} \cdot l_r) \cdot \alpha - C_{\alpha f} \cdot l_f \cdot \delta = 0 \quad (12)$$

kde:

ε	[rad]	úhel stáčení
C _{αf}	[m.s ⁻¹]	směrová tuhost obou pneumatik přední nápravy
C _{αr}	[m.s ⁻²]	směrová tuhost obou pneumatik zadní nápravy
δ	[rad]	úhel natočení předních kol



Následuje několik úprav. Vektor rychlosti můžeme rozepsat takto:

$$v = [U, V], \quad (13)$$

kde:

v	$[\text{m}\cdot\text{s}^{-1}]$	vektor rychlosti v těžišti vozidla
U	$[\text{m}\cdot\text{s}^{-1}]$	podélná složka rychlosti
V	$[\text{m}\cdot\text{s}^{-1}]$	příčná složka rychlosti

Výše bylo uvedeno, že uvažujeme pouze malé úhly. Pro úhel směrové úchyly α v těžišti pak platí:

$$\alpha = \frac{V}{U}, \quad (14)$$

Výsledná soustava rovnic pro popis dynamiky vozidla má pak v maticovém zápisu tuto podobu [1; 7]:

$$\begin{bmatrix} \dot{V} \\ \dot{\varepsilon} \end{bmatrix} = \begin{bmatrix} \frac{-C_{\alpha f} - C_{\alpha r}}{Um} & \frac{-C_{\alpha f} \cdot l_f + C_{\alpha r} \cdot l_r - U^2 m}{Um} \\ \frac{-C_{\alpha f} \cdot l_f + C_{\alpha r} \cdot l_r}{UJ_z} & \frac{-C_{\alpha f} \cdot l_f^2 - C_{\alpha r} \cdot l_r^2}{UJ_z} \end{bmatrix} \begin{bmatrix} V \\ \varepsilon \end{bmatrix} + \delta \begin{bmatrix} \frac{C_{\alpha f}}{m} \\ \frac{C_{\alpha f} \cdot l_f}{J_z} \end{bmatrix}, \quad (15)$$

Úhel natočení předních kol δ je budící funkce, veličiny V a ε jsou odezvy systému. Abychom dokázali určit polohu vozidla, musíme složky jeho rychlosti $[U, V]$ přepočítat (transformovat) do základního souřadného systému $[x_0, y_0]$, který svírá se souřadným systémem vozidla úhel ε , viz obrázek 4.8.

$$\begin{bmatrix} U_0 \\ V_0 \end{bmatrix} = \begin{bmatrix} \cos \varepsilon & -\sin \varepsilon \\ \sin \varepsilon & \cos \varepsilon \end{bmatrix} \begin{bmatrix} U \\ V \end{bmatrix}, \quad [1] \quad (16)$$

kde:

U_0	$[\text{m}\cdot\text{s}^{-1}]$	podélná složka rychlosti v základním souřadném systému
V_0	$[\text{m}\cdot\text{s}^{-1}]$	příčná složka rychlosti v základním souřadném systému
ε	[rad]	úhel mezi souřadnými systémy

Výpočet polohy [7]:

$$x_0 = \int_0^t v \cdot \cos(\alpha + \varepsilon) dt \quad (17)$$

$$y_0 = \int_0^t v \cdot \sin(\alpha + \varepsilon) dt \quad (18)$$

kde:

x_0	[m]	poloha ve směru osy x
y_0	[m]	poloha ve směru osy y

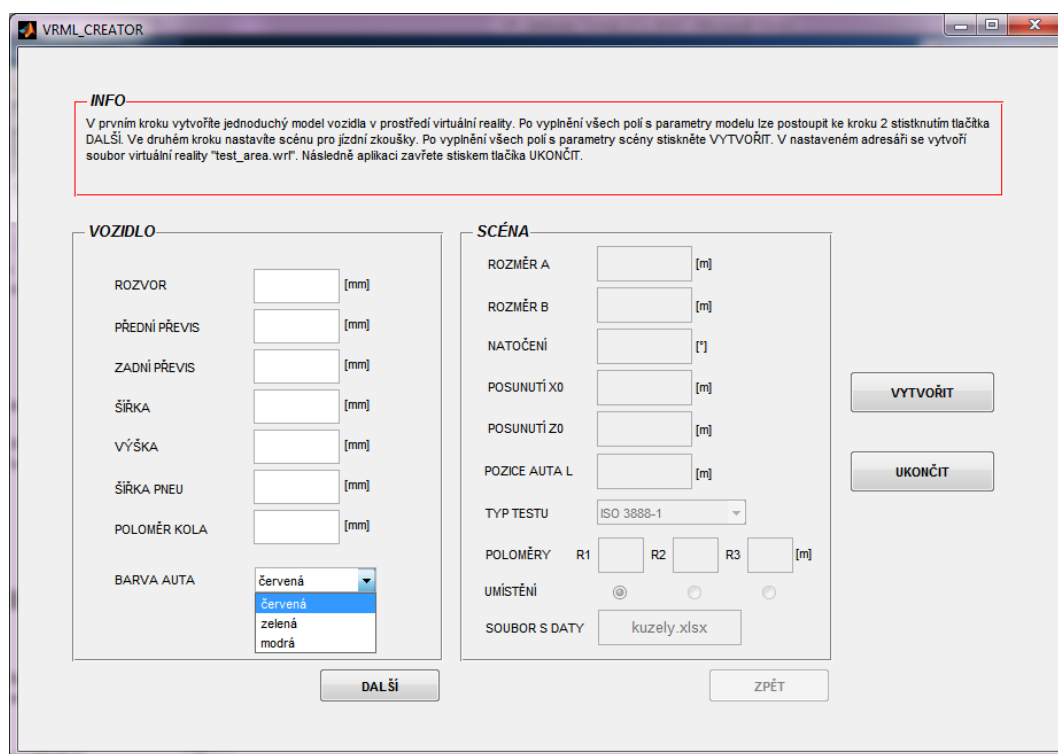


5 VYTVOŘENÍ VIRTUÁLNÍ SCÉNY

V této kapitole bude popsáno prostředí sloužící pro vytvoření virtuálního světa. Cílem je umožnit uživateli volit si parametry – rozměry vozidla a testovací plochy dle potřeby. Výsledkem má být tedy jakýsi parametrický model, vytvořený v několika málo krocích. Nejdříve byla zvolena varianta vytvoření skriptu, jenž by obsahoval jednotlivé parametry uložené do proměnných. Tato možnost však nebyla vhodná, neboť přepisování parametrů přímo ve zdrojovém textu by bylo nepřehledné. Proto byla v programu Matlab pomocí toolboxu GUIDE vytvořena uživatelská aplikace, která usnadňuje a zrychluje práci.

5.1 APLIKACE VRML CREATOR

Aplikace VRML CREATOR slouží ke snadné a rychlé tvorbě souboru virtuální reality. Pro definici stačí pouze několik kliknutí myši a zadání několika rozměrů. Během chvilky tak vznikne scéna dostačující k zobrazení pohybu vozidla.



Obr. 5.1 Prostředí aplikace VRML CREATOR

Okno aplikace je rozděleno na tři hlavní oddíly. V horní části se nachází základní informace o aplikaci a jejím používání. Levá dolní část slouží k zápisu rozměrů vozidla, pravá dolní k volbě a zadávání parametrů scény. Jednotlivé hodnoty se vepisují do editovatelných textových políček. Dále se zde vyskytují rozbalovací (pop-up) menu, textová pole, prepínací tlačítka (radiobutton) a klasická tlačítka (pushbutton).

Po spuštění je aktivní pouze část pro definici vozidla. Parametry scény nelze v tuto chvíli měnit. Nejdříve je nutné korektně vyplnit rozměry vozidla:

- rozvor
- přední převis
- zadní převis



- šířku vozidla
- výšku vozidla
- šířku pneumatik
- poloměr kola (disk + pneumatika).

Posledním parametrem je barva vozidla. Z rozbalovacího menu lze vybrat tři barvy – červená, zelená a modrá. Defaultně je nastavena červená barva. Jednoduchým způsobem se dá přidat více barev. Jednotlivé položky menu mohou být přidány pomocí toolboxu Guide. Těmto položkám se následně přiřadí barvy složkami RGB (R-red, G-green, B-blue) ve zdrojovém textu aplikace (soubor *VRML_CREATOR.m*), v oddílu *pushbutton1_Callback*.

Posun k dalšímu kroku je možný až po vyplnění všech textových polí. Jejich obsah je omezen pouze na číselné hodnoty. V případě, že pole obsahuje písmena, je uživatel na toto upozorněn. Příklad použité konstrukce příkazů:

```
E1=get(handles.edit1,'String');      % uložení obsahu pole do proměnné
T1=isletter(E1);                    % zjištění výskytu písmen
a=length(E1);                       % zjištění délky řetězce

for i=1:a                            % cyklus pro kontrolu každého znaku řetězce
if T1(i)~=1
    warndlg('Neplatná hodnota. Zadejte pouze číselné hodnoty v desítkové soustavě.',
    'UPOZORNĚNÍ');                  % vytvoření okna s upozorněním
    set(handles.edit1,'String',''); % vymazání obsahu pole
    break
end
end
```

Ve zdrojovém kódu aplikace jsou zakomponovány i další omezující podmínky. Jedná se o stanovení mezí, minimálních nebo maximálních rozměrů jednotlivých částí. Vytvářený model by totiž při neuvážené volbě parametrů mohl vykazovat chyby a nemusel by se správně zobrazovat. Jednou z podmínek je například hodnota rozvoru vozu vůči velikosti poloměru kola. Další podmínky určují minimální hodnoty předního a zadního převisu. Minimální šířka vozidla je nastavena na jeden metr, šířka pneumatiky na 400 milimetrů a poloměr kola na 500 milimetrů. V případě potřeby je možné tyto okrajové podmínky změnit ve zdrojovém textu v části *pushbutton2_Callback*.

Kliknutím na tlačítko *Další* se zpřístupní část pro nastavení scény a naopak se uzamkne první část. Vrátit se k případné úpravě v první části je možné pomocí tlačítka *Zpět*. Mezi veličiny scény patří:

- rozměry testovací plochy
- počáteční natočení souřadného systému kuželů
- počáteční pozice prvního kužele na testovací ploše
- počáteční vzdálenost vozidla před prvním kuželem
- volba jízdního testu
- poloměry pro kruhový test
- název souboru pro uživatelský test

I v této sekci lze zadávat pouze číselné hodnoty. Meze testovací plochy nejsou stanoveny, je však doporučeno volit raději větší rozměry. Rozsah natočení souřadného



systému kuželů je 0° až 90° . V rozbalovacím menu se nachází čtyři možné varianty testovacího manévru. První dva testy jsou definované normou ISO 3888. Třetí položkou je kruhový test, pro který je potřeba zadat poloměry kružnice zobrazovaných na ploše. Poloměry musí být zadány od nejmenšího po největší. Poslední variantou je uživatelský test. Jedná se o trať vytyčenou kužely a rozmístění jednotlivých kuželů je definováno tabulkou v souboru programu MS Excel (*.xlsx). Název souboru, ze kterého se data načítají, se zapisuje do posledního textového políčka. Orientaci při zadávání proměnných usnadňuje zobrazení zakótovaného náčrtku v samostatném okně.

Posledním krokem zůstává samotné vytvoření virtuální scény. K tomu dojde stisknutím tlačítka *Vytvořit*. Při této akci se nejdříve uloží do proměnných hodnoty z textových polí. Při volbě některé z variant testu dle ISO 3888 je kontrolována pozice vozidla vůči testovací ploše. Pokud se nachází průsečík osy zadní nápravy a podélné osy vozu mimo plochu, zobrazí se okno s upozorněním a požadavkem na změnu parametrů. Syntaxe pro kontrolu pozice vypadá následovně:

```
switch test
  case 1
    car(1,1)=x0-(0.5*((1.2*width)+0.25)*sind(natoceni))-
      -((car_pos+fov+wheelbase)*cosd(natoceni));
    car(1,2)=z0+(0.5*((1.2*width)+0.25)*cosd(natoceni))-
      -((car_pos+fov+wheelbase)*sind(natoceni));
    if (car(1,1)<0 | car(1,2)<0)
      warndlg('Upravte kombinaci parametrů x0, z0, L a úhel natočení. Poloha
        vozidla se nachází mimo testovací plochu.','UPOZORNĚNÍ');
    else
      vrml12(wheelbase,fov,rovh,width,height,w_radius,w_width,color,area_x,area_z,
        natoceni,x0,z0,car_pos,test)
      warndlg('Vše proběhlo v pořádku. Byl vytvořen model scény s vozidlem. Nyní můžete
        aplikaci zavřít.','POTVRZENÍ');
    end
  case 2
    :
    :
end
```

Jakmile je vše v pořádku, spustí se příslušná funkce, která obstará vytvoření souboru virtuální reality dle zadaných hodnot. Pro každý typ testu je vytvořena konkrétní funkce. Jednotlivé funkce budou popsány dále. Po úspěšném vytvoření se objeví zpráva a aplikaci je možné zavřít stisknutím tlačítka *Ukončit*.

5.2 FUNKCE VRML

Výše zmíněná aplikace dovoluje nastavit jeden ze čtyř nadefinovaných testů. Vytváření virtuální reality pro různé testy s využitím jedné funkce či skriptu by bylo komplikované. Proto byly vytvořeny 3 funkce: vrml12, vrml3 a vrml4. Funkce zahrnují definici vozidla, popis testovací plochy a okolí, rozmístění kuželů, příp. kružnic, geometrii prvků doplňujících samotné vozidlo (volant, pedály, přístrojová deska, znázornění řadicí páky), definici prvků pro zobrazení dynamických účinků (soubor čar znázorňujících síly, zrychlení, rychlost, Kammovy kružnice) a nastavení pohledů pro prohlížení scény.



PŘÍKAZ FPRINTF

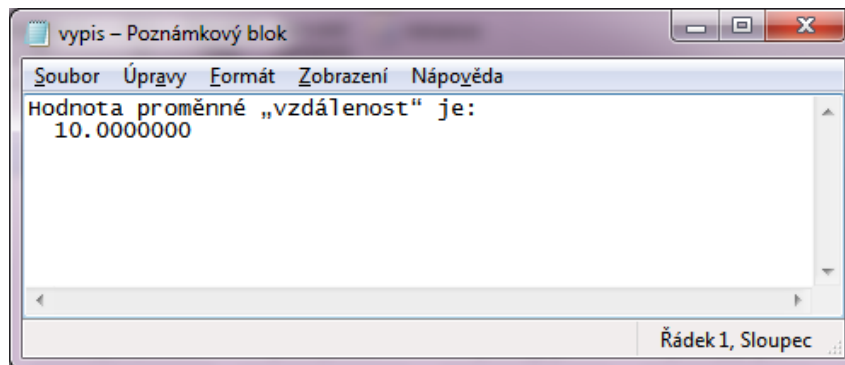
K vytvoření virtuálního světa je zapotřebí zápis v jazyce VRML 97. To znamená, že pomocí Matlabu musí dojít k zápisu VRML kódu do textového souboru. K tomu slouží příkaz (funkce) *fprintf*. Konkrétní soubor musí být nejdřív zpřístupněn a to příkazem *fopen*. Pro dokončení je nutné soubor zase uzavřít příkazem *fclose*. Při zpřístupňování a zápisu lze pracovat ve třech módech [6]:

- ‚r‘ otevře soubor pouze pro čtení
- ‚w‘ umožněn zápis, pokud soubor neexistuje, založí se, pokud existuje, přepíše se
- ‚a‘ umožněno přidávání, pokud soubor neexistuje, založí se, pokud existuje, data se připojí na konec souboru.

Parametry příkazu *fprintf* jsou: popisovač (identifikátor) souboru, formát zapisovaných dat a vlastní proměnné. Formát ovlivňuje podobu zápisu proměnné a její délku. Formátovací řetězec začíná vždy znakem procenta, následují dvě čísla oddělená tečkou. Zápis *%8.4f* značí, že se bude zapisovat proměnná ve tvaru desetinného čísla, která má celkem 8 pozic včetně desetinné čárky, z toho 4 cifry se nachází za desetinnou čárkou. Do výpisu je možné umístit také určité speciální znaky, např.: \n (skočí na nový řádek), \t (tabulátor), \\ (zpětné lomítko), \“ (apostrofov), %% (procento). [6] Příklad výpisu hodnoty proměnné:

```
vzdalenost=10;
fid=fopen('vypis.txt','w');           %vytvoreni identifikatoru
fprintf(fid, 'Hodnota proměnné „vzdálenost“ je: \r\n'); %výpis do souboru
fprintf(fid, %12.7f \r\n,vzdalenost); %výpis proměnné
fclose(fid)                           %uzavreni souboru
```

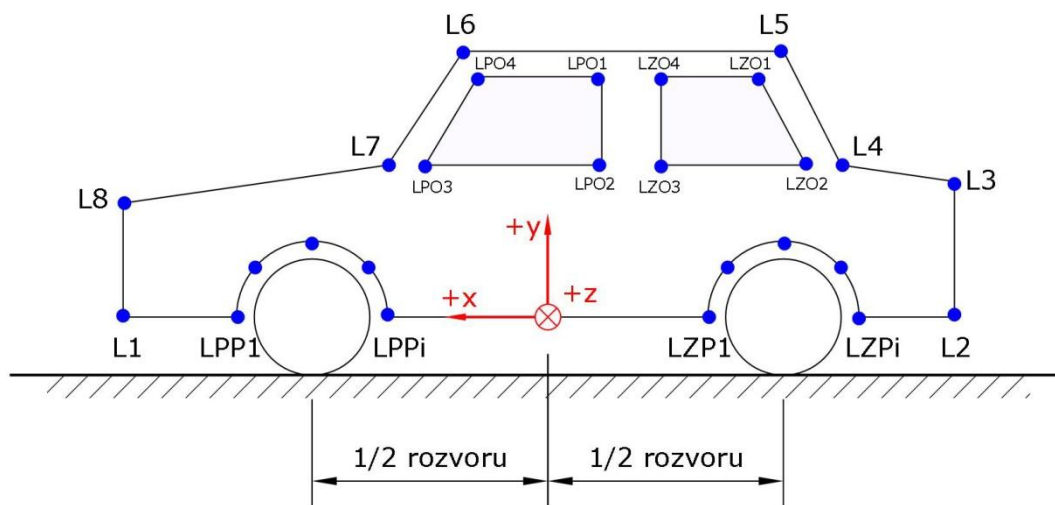
Výsledek:



Obr. 5.2 Vytvoření textového souboru příkazem *fprintf*

5.2.1 DEFINICE VOZIDLA

V rámci každé funkce je vypočítávána geometrie vozidla, resp. poloha jednotlivých bodů. Cílem není vytvořit reálný model. To by bylo značně složité a vyžadovalo by to zdlouhavou optimalizaci kódu. Proto byla vytvořena pouze základní reprezentace vozidla, která je pro zobrazení pohybu vozidla dostačující. Model vozu je sestaven převážně z prvků *IndexedFaceSet*. Souřadnice jednotlivých bodů, potřebných pro zápis ploch, jsou počítány pomocí základních matematických operací s využitím goniometrických funkcí.



Obr. 5.3 Boční pohled na vozidlo s vyznačenými body

Boční tvar modelu a poloha bodů jsou naznačeny na obrázku 5.3. Celkově se prototyp skládá z třiceti osmi ploch a ze čtyř válcových těles. Kompletní model obsahuje také volant a přístrojovou desku. Tvorba těchto prvků bude vysvětlena v dalších kapitolách.

Vztažným bodem pro definici souřadnic je počátek souřadného systému automobilu, jenž je umístěn v průsečíku podélné osy a příčné osy půlící vzdálenost mezi nápravami. Orientace kladných směrů jednotlivých os viz obrázek 5.3.

Nejdříve se získají souřadnice předního podběhu LPPi, resp. zadního podběhu LZPi. Příslušné body jsou na oblouku rozmístěny s krokem 10° , tj. 19 pozic. Toto rozdělení zcela dostačuje pro vykreslování. Krok je možné změnit, ale muselo by dojít k úpravě větší části zdrojového souboru funkce. Poloměr podběhu je oproti poloměru kola zvětšen o 40 mm. Další pozice bodů už jsou dopočítávány ze známých bodů a ze zadaných hodnot. Ve výpočtu jsou nastaveny na pevně určité hodnoty úhlů, které ovlivňují jednotlivá umístění. Úhly se odměřují od vodorovné roviny. Hodnoty jsou určeny takto:

- úhel kapoty 8°
- úhel kufru 3°
- úhel čelního skla 53°
- úhel zadního skla 55°

V případě nutnosti se dají i tato čísla změnit. Nemělo by se tak ale dít v příliš velkém rozsahu, aby bylo zaručeno správné vykreslení reprezentace vozidla.

Po výpočtu souřadnic bodů na levém boku dojde k uložení bodů pravého boku pouze změnou souřadnice z. Nakonec jsou definovány zbývající pozice oken (na levém boku, na pravém boku, čelní, zadní), ploch znázorňujících přední a zadní světla a body pro „uzavření“ modelu ze spodní části. Tím jsou všechna nutná umístění určena. V poslední části dochází k vypisování příkazů pro vznik virtuálního objektu. Příklad zápisu kódu pro vykreslení plochy vozu:

```
fprintf(fid, 'DEF vozidlo Transform { \r\n');
fprintf(fid, 'translation %12.7f %12.7f %12.7f \r\n', car(1,1), w_radius, -car(1,2));
fprintf(fid, 'rotation 0 1 0 %12.7f \r\n', natoc);
```

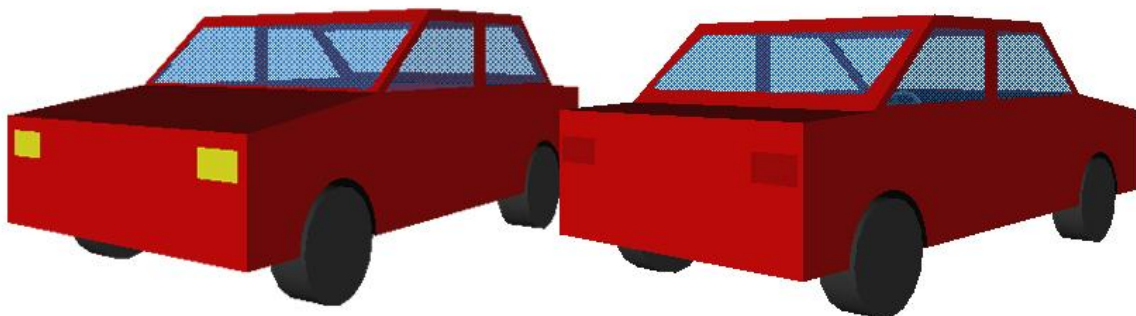


```

fprintf(fid, 'children [ \r\n');
fprintf(fid, 'DEF vozidlo_pohyb Transform { \r\n');
fprintf(fid, 'children [ \r\n');
fprintf(fid, 'DEF carbody Transform { \r\n');
fprintf(fid, 'children [ \r\n');
:
:
fprintf(fid, 'geometry IndexedFaceSet { \r\n');
fprintf(fid, 'color Color { \r\n');
fprintf(fid, 'color [ 0.801 0.801 0.801, %7.5f %7.5f %7.5f] \r\n',color(1),color(2),color(3));
fprintf(fid, '}' \r\n');
fprintf(fid, 'coord Coordinate { \r\n');
fprintf(fid, 'point [ %12.7f %12.7f %12.7f, \r\n',lslip(1),lslip(2),lslip(3));
for w=1:l
    fprintf(fid, '%12.7f %12.7f %12.7f, 'lfww(w,1),lfww(w,2),lfww(w,3));
end
:
:
fprintf(fid, 'normal NULL \r\n');
fprintf(fid, 'texCoord NULL \r\n');
fprintf(fid, 'ccw FALSE \r\n');
fprintf(fid, 'colorPerVertex FALSE \r\n');
fprintf(fid, 'convex FALSE \r\n');
fprintf(fid, 'normalPerVertex TRUE \r\n');
fprintf(fid, 'creaseAngle 0 \r\n');
fprintf(fid, 'solid FALSE \r\n');
fprintf(fid, 'coordIndex [ ');
for N=0:46
    fprintf(fid, '%5.0f, 'N);
end

```

%zápis indexu bodu plochy



Obr. 5.4 Model vozidla

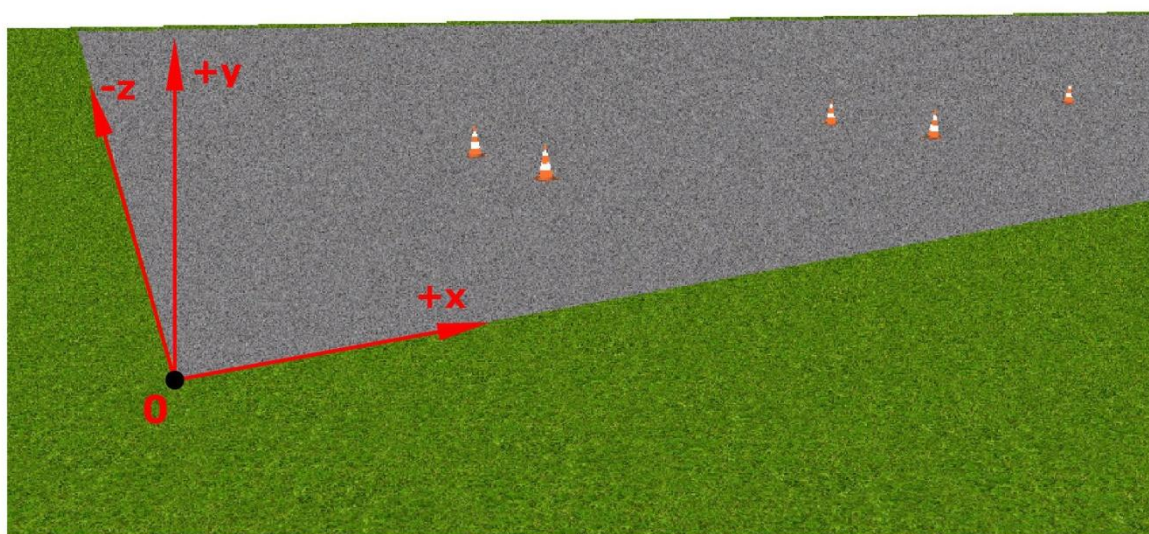


5.2.2 TESTOVACÍ PLOCHA

Plocha, po které se vůz pohybuje, je rovinná a je tvořena základním tělesem ve tvaru kvádrů. Rozměry podstavy vychází z parametrů zadaných v aplikaci. Výška kvádrů je nastavena na 50 milimetrů. Celý kvádr je pokryt texturou asfaltu. Hustota (měřítko) textury se upravuje automaticky podle velikosti plochy, takže dráha vypadá vždy stejně.

Okolí je vytvořeno také pomocí kvádrů. Rozměry v rovině x-z jsou v každém směru třikrát větší než rozměry testovací plochy. Na těleso je nanášena textura znázorňující trávník. Oba kvádry jsou vůči sobě vhodně posunuty, aby nedocházelo k překrývání a chybě ve vykreslování povrchů. Poloha souřadného systému dráhy je vidět na obrázku 5.5.

V závislosti na zvoleném jízdním manévru dále scéna obsahuje buď kužely rozmístěné po ploše, nebo tři kruhy (kruhový test).



Obr. 5.5 Orientace souřadného systému testovací dráhy

5.2.3 FUNKCE VRML12

Tato funkce obstarává vytvoření scény pro jízdní manévry definované normou ISO 3888, viz kapitola 3.1 a 3.2. Struktura a algoritmus se přizpůsobuje spuštění přes aplikaci VRML CREATOR. Funkci lze samozřejmě spustit i ručně v příkazovém okně Matlabu. V tom případě je však potřeba porozumět významu jednotlivých vstupních proměnných. Ty jsou stručně popsány v nápovědě k funkci. Zápis funkce se vstupy:

vrml12(wheelbase,fov,rovh,width,height,w_radius,w_width,color,area_x,area_z,natoeni,x0,z0,car_pos,test)

Celkem vstupuje z aplikace do funkce 15 parametrů. V úvodu se nachází výpočet geometrie vozidla (pro všechny funkce stejný). Následuje část vztahující se ke scéně testovací plochy. V maticích jsou pro obě varianty testu pevně určeny pozice jednotlivých kuželů rozmístěných napravo od vozu. Pozice levých kuželů je dále dopočítána z geometrie vozidla, resp. z jeho šířky. Na základě hodnoty vstupní proměnné *test* je pomocí použití podmínky *IF* vybrána správná varianta testu ISO. Pro tyto manévry nabývá parametr *test* hodnot 1 nebo 2.



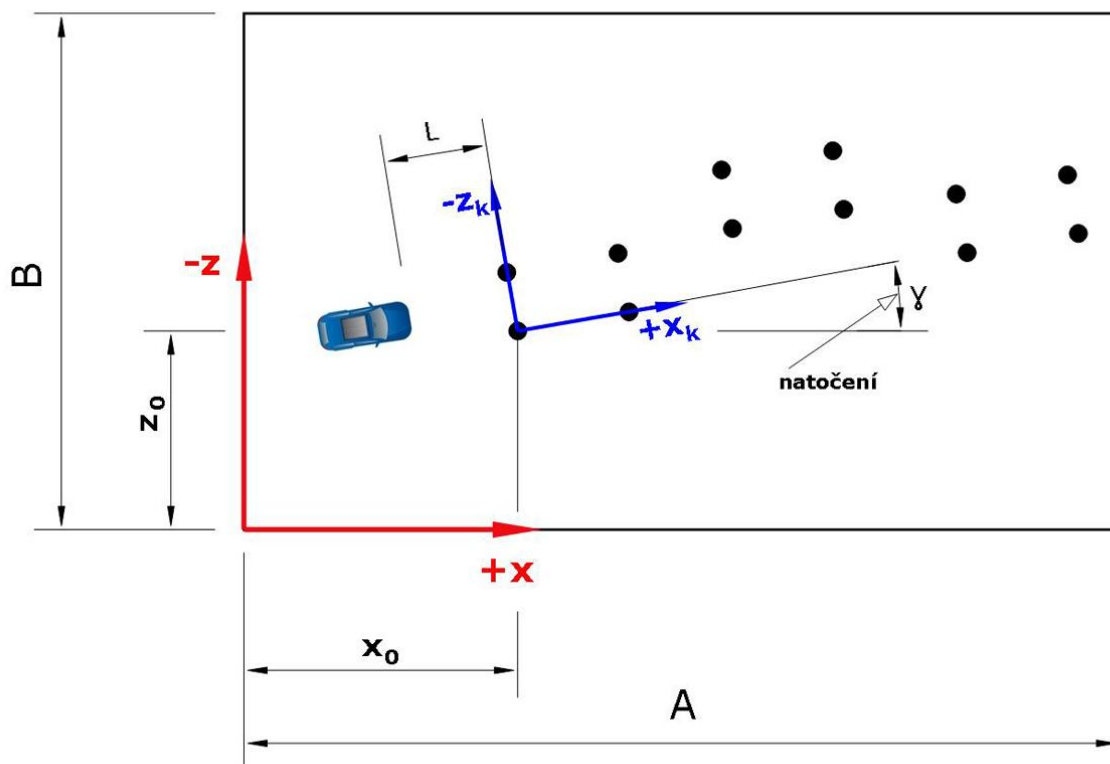
U ISO 3888-2 algoritmus kontroluje šířku posledního úseku tratě. Dle normy musí být minimálně 3 metry široká. Pokud je podmínka splněna, kužely jsou umístěny dle tabulky 3.2. Jestliže šířka vychází menší, nastaví se levé kužely v posledním úseku právě 3 metry od pravých. Kontrola šířky je zapsána takto:

```

:
cones_loc.l_cones(11:15,1)=cones_loc.r_cones(11:15,1);
cones_loc.l_cones(11:15,3)=0;

%ověření podmínky - poslední usek musí být minimálně 3m široký
if ((1.3*width)+0.25)>=3
    cones_loc.l_cones(11:15,2)=(cones_loc.r_cones(11:15,2)+((1.3*width)+0.25))+0.185;
else
    cones_loc.l_cones(11:15,2)=(cones_loc.r_cones(11:15,2)+3)+0.185;
end

```



Obr. 5.6 Rozložení scény testu dle ISO 3888

Na obrázku 5.6 je znázorněno a zakótováno rozložení scény. Rozměry x_0 a z_0 určují polohu počátku souřadného systému kuželů. Tímto počátkem prochází svislá osa prvního pravého kužele. Orientaci souřadného systému ovlivňuje úhel natočení (kolem svislé osy). Rozměr L nastavuje výchozí polohu vozidla vůči kuželům. Vyjadřuje vzdálenost od středu prvních kuželů po přední část vozu. I přes to, že jsou jednotlivé prvky umístěny v záporné části globální osy z , parametry se uvádí kladné. Převod na záporné hodnoty řeší automaticky přímo funkce `vrml12`. Umístění vozidla na ploše:



```

natoc=(pi/180)*natoceni;
%pozice vozidla ve scéně
if test==1
    car(1,1)=x0-(0.5*((1.1*width)+0.25+0.185)*sin(natoc))-
        -((car_pos+fov+0.5*wheelbase)*cos(natoc));
    car(1,2)=z0+(0.5*((1.1*width)+0.25+0.185)*cos(natoc))-
        -((car_pos+fov+0.5*wheelbase)*sin(natoc));
elseif test==2
    car(1,1)=x0-(0.5*((1.1*width)+0.25+0.185)*sin(natoc))-
        -((car_pos+fov+0.5*wheelbase)*cos(natoc));
    car(1,2)=z0+(0.5*((1.1*width)+0.25+0.185)*cos(natoc))-
        -((car_pos+fov+0.5*wheelbase)*sin(natoc));
end

```

Příkaz pro zápis v jazyce Vrml:

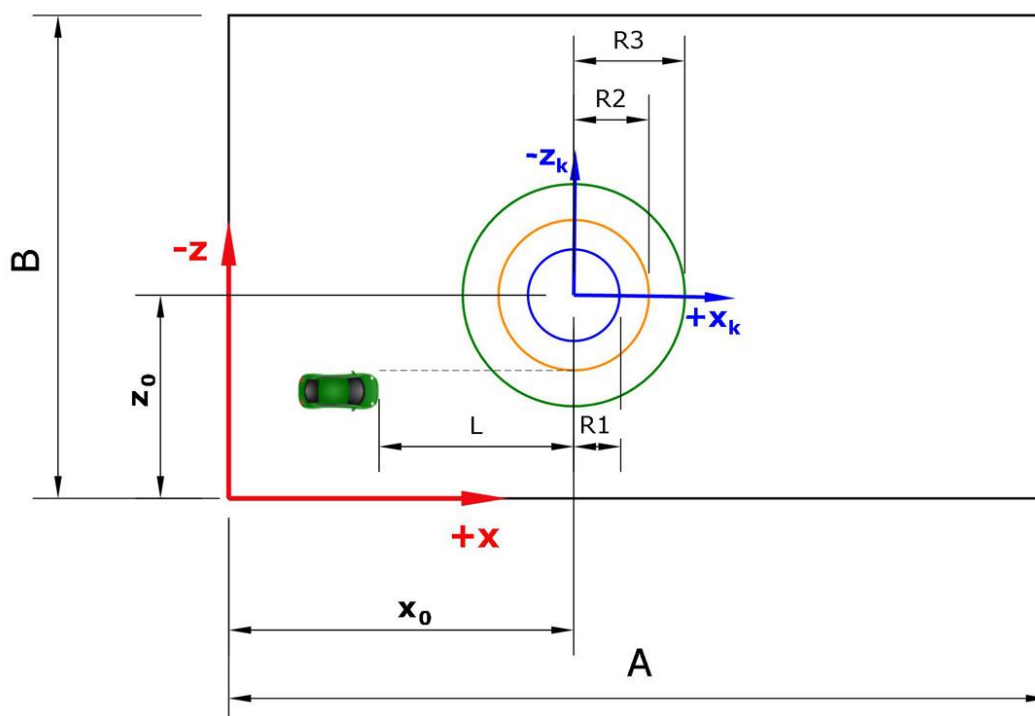
```

fprintf(fid, 'DEF vozidlo Transform { \r\n');
fprintf(fid, 'translation %12.7f %12.7f %12.7f \r\n', car(1,1), w_radius, -car(1,2));
fprintf(fid, 'rotation 0 1 0 %12.7f \r\n', natoc);
:

```

5.2.4 FUNKCE VRML3

Přípravu scény pro tzv. kruhový test zajišťuje funkce *vrml3*. Při této konfiguraci se zde nenachází žádné kužely. Na podložce, po které se vozidlo pohybuje, jsou vytvořeny 3 soustředné kruhy vytvořené prvkem *IndexedFaceSet*. Rozmístění ukazuje obrázek 5.7.



Obr. 5.7 Podoba scény pro kruhový test



Vnější poloměry jednotlivých kruhů, resp. mezikruží zadá uživatel. Vnitřní poloměr je vypočítáván z vnějšího zmenšením hodnoty o 500 milimetrů. Pro využití prvku IndexedFaceSet je potřeba znát souřadnice všech bodů. Body jsou vytvořeny rozdělením kruhu po 5° (vznikne 72 bodů) a použitím cyklu *for* s příslušnými vztahy:

```
krok=5;
stupne=(0:krok:360);           %rozdělení kružnice
alfa=(stupne*pi)/180;         %přepočet stupňů na radiány
%výpočet souřadnic kružnice
poc_kroku=size(stupne);
kruh1(1:(poc_kroku(2)-1),1:3)=0;
kruh2(1:(poc_kroku(2)-1),1:3)=0;
for i=1:(poc_kroku(2)-1)
    kruh1(i,:)=[r1*cos(alfa(i)),0,r1*sin(alfa(i))];
    kruh2(i,:)=[r2*cos(alfa(i)),0,r2*sin(alfa(i))];
end
```

Střed kruhů definují opět souřadnice x_0 a z_0 . Vozidlo se umísťuje tečně ke kruhům tak, že osy ve směru x vozidla a globálního souřadného systému jsou rovnoběžné. Ke kterému kruhu bude vozidlo ustaveno, je zvoleno pomocí přepínacích prvků (radiobuttonů) v okně aplikace VRML CREATOR:

```
if umisteni==[1,0,0]           %stav prvků (on, off, off)
    car(1,1)=x0-(car_pos+fovh+0.5*wheelbase); %určení počáteční pozice vozidla tečně k
                                        %poloměru R1
    car(1,2)=-z0+R1+(0.5*width);
elseif umisteni==[0,1,0]      %stav prvků (off, on, off)
    car(1,1)=x0-(car_pos+fovh+0.5*wheelbase); %určení počáteční pozice vozidla tečně k
                                        %poloměru R2
    car(1,2)=-z0+R2+(0.5*width);
elseif umisteni==[0,0,1]      %stav prvků (off, off, on)
    car(1,1)=x0-(car_pos+fovh+0.5*wheelbase); %určení počáteční pozice vozidla tečně k
                                        %poloměru R3

    car(1,2)=-z0+R3+(0.5*width);
end
```

5.2.5 FUNKCE VRML4

Poslední z vytvořených funkcí dovoluje uživateli navrhnout si svůj vlastní test. Rozmístění jednotlivých kuželů na ploše je zapsáno v souboru programu MS Excel. Součástí této práce je vzorová tabulka, viz obrázek 5.8. Souřadnice se nachází ve sloupcích A-C. Ve sloupci A se definuje poloha ve směru x , ve sloupci B poloha ve směru z a ve sloupci C poloha ve směru osy y (výšková souřadnice). Pro pohyb ve vodorovné rovině musí být výškové souřadnice nastaveny na hodnotu 0. Na prvních dvou řádcích je nutné popsat pozici prvního pravého a prvního levého kužele. Jde o vytvoření jakési „vstupní brány“, od které se odměřuje počáteční pozice vozu. Jednotlivé vzdálenosti mezi kužely se vztahují k prvnímu kuželu [$x=0, z=0$].

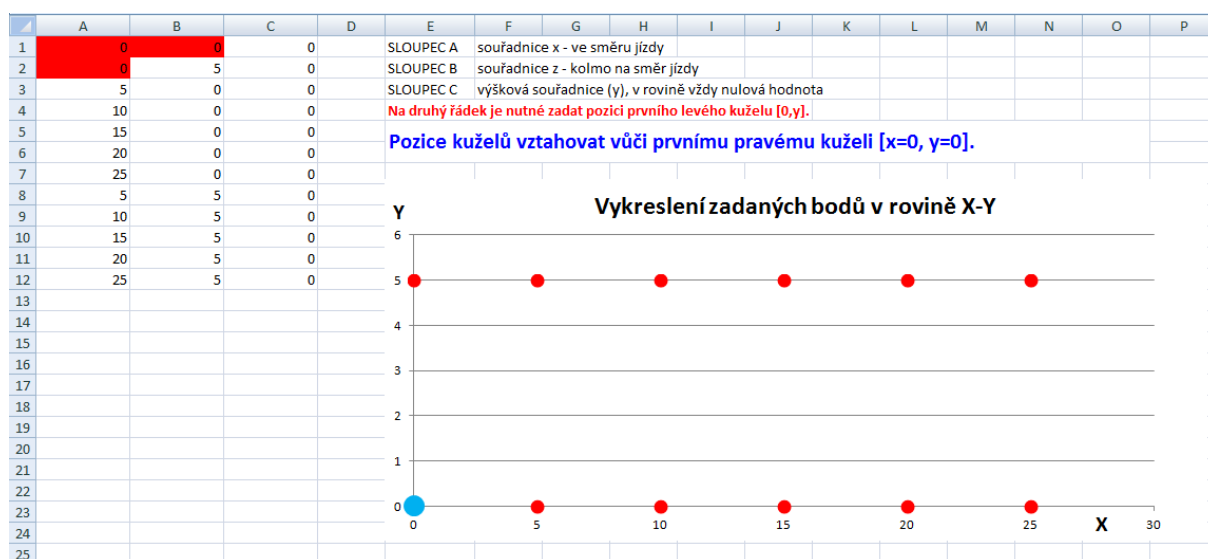


Funkce provede automaticky import dat ze sloupců A-C a uloží je do proměnné *cones*. Rozsah sloupců je možné měnit ve zdrojovém textu:

```
cones=xlread(name,'A:C'); %import
cones(:,2)=cones(:,2)*(-1); %úprava pro orientaci ve virtuálním světě
[a,b]=size(cones);
```

%umístění auta

```
rozdil=cones(1,2)-cones(2,2);
car(1,1)=x0-(car_pos+(wheelbase/2)+fov);
car(1,2)=-z0-abs((rozdil/2));
```



Obr. 5.8 Zápis souřadnic v MS Excel

5.3 ZOBRAZENÍ OVLÁDACÍCH PRVKŮ VOZIDLA

Ke snadnější představě o pohybu a ovládání vozidla obsahuje scéna některé, k tomu vhodné, zobrazovací prvky. Patří mezi ně:

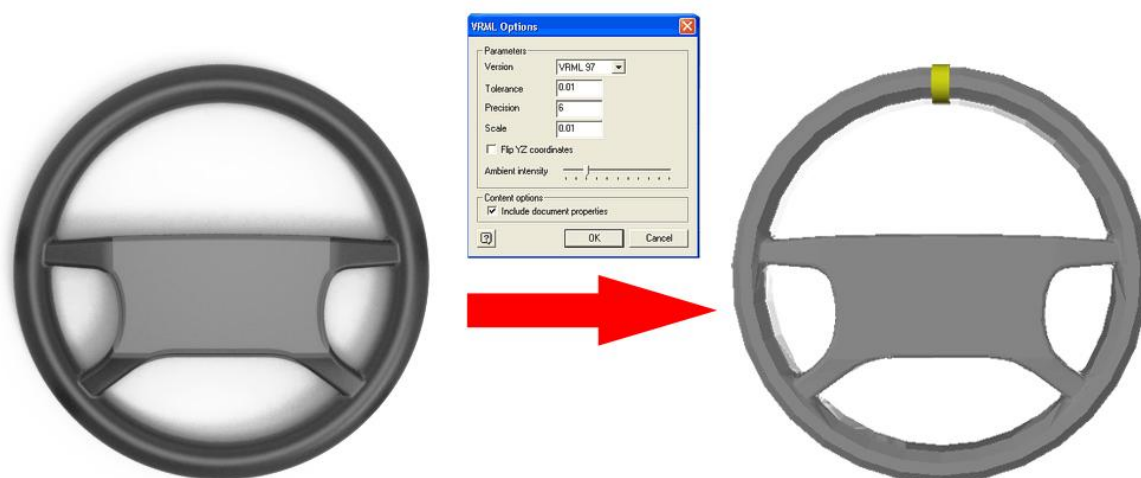
- model volantu
- model přístrojové desky
- model pedálů (plynový, brzdový, spojkový)
- ukazatel zařazeného rychlostního stupně

Zatáčení vozu je reprezentováno natáčením volantu, otáčky a rychlost lze sledovat na otáčkoměru a rychloměru na přístrojové desce, přidávání plynu, brzdění a řazení vyobrazuje pohyb soustavy pedálů. Zařazený rychlostní stupeň je znázorněn na válcovém tělese s obrazovou texturou.



5.3.1 MODEL VOLANTU

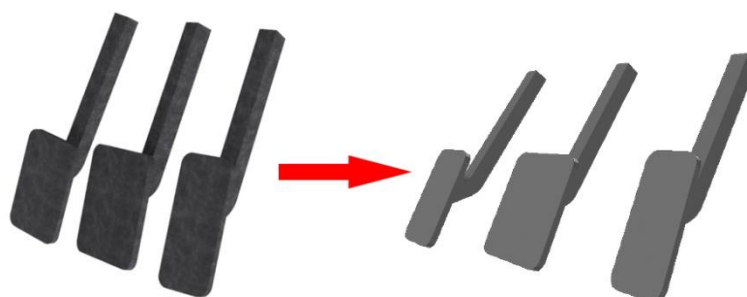
Pomocí základních konstrukčních prvků ve virtuální realitě by nebylo možné model sestavit. Jedinou možností tak zůstalo použití soustavy ploch. Proto byl model volantu vymodelován v programu Autodesk Inventor Professional 2013. Následně pomocí doplňkového modulu VRML Translator proběhl export do jazyka VRML. Prvotní návrh volantu byl zapsán na přibližně 40 tisících řádcích, což by značně převyšovalo počet řádků zbylé části souboru. Následně tedy došlo ke zredukování, pro reprezentaci volantu, nepotřebných rádiusů, zkosení, apod. Byla nastavena také menší přesnost modelu při exportu. Výsledný vrml zápis byl pomocí editoru PSPad doplněn o zápis příkazu *fprintf*. Volant je opatřen žlutou středovou značkou, vytvořenou jako základní válcové těleso. Konečný model volantu je znázorněn na obrázku 5.9.



Obr. 5.9 3D a VRML model volantu

MODEL PEDÁLŮ

Sestava pedálů byla vytvořena stejným postupem, jako tomu bylo u volantu. Pedály se skládají z jednoduchých tvarů, což opět vychází z požadavku na rozumně dlouhý textový zápis těles ve VRML. Podoba pedálů je na obrázku 5.10.

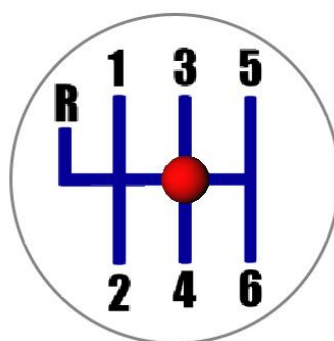


Obr. 5.10 Sestava pedálů



5.3.2 UKAZATEL ZAŘAZENÉHO RYCHLOSTNÍHO POLE

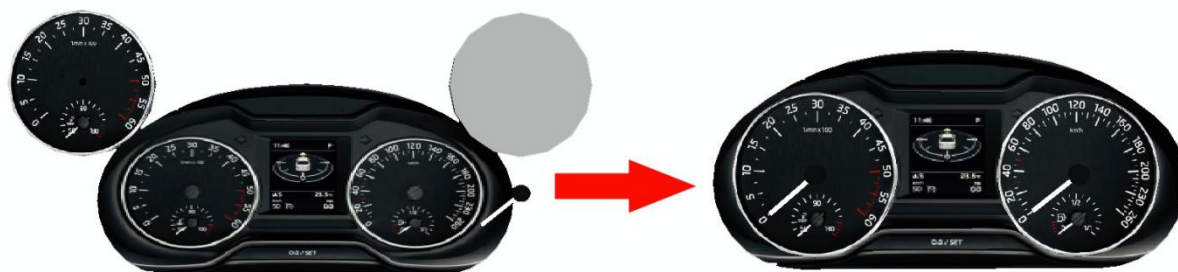
K informování uživatele o zařazeném rychlostním stupni je určen jednoduchý ukazatel na obrázku 5.11. Sestává z válcového tělesa o malé výšce, obrazové textury a koule. Textura znázorňující polohu rychlostních stupňů byla vytvořena pomocí programu Adobe Photoshop. Červená kulička se skokově pohybuje v závislosti na zařazené rychlosti. Její přesné pozice pro jednotlivé stupně byly stanoveny zkoušením a postupným zpřesňováním ustavení. Souřadnice kuličky jsou uloženy v modelu pro řízení pohybu v programu Simulink. Výchozí pozice kuličky je v poloze neutrální.



Obr. 5.11 Podoba ukazatele rychlostních stupňů

5.3.3 PŘÍSTROJOVÁ DESKA

Tvar přístrojové desky byl modelován obdobně jako volant a pedály v programu Inventor s následným exportem do VRML. Povrch tohoto tělesa je pokryt texturou vyobrazující přístrojovou desku s otáčkoměrem a tachometrem vozu Škoda Octavia III. Textura byla získána z [16; 17]. Obrázek prošel grafickou úpravou tak, aby co nejvíce kopíroval vytvořené těleso. I tak došlo k jisté deformaci textury. Věrnějšímu zobrazení přispělo přidání dvou válcových těles, představující rychloměr a tachometr. Na ně byla doplněna textura stupnic obou ukazatelů. Uprostřed těchto válců byla dotvořena tělesa tvaru hranolu reprezentující ručičky, otáčející se kolem středu válců.



Obr. 5.12 Celek přístrojové desky



5.4 ZOBRAZENÍ DYNAMICKÝCH ÚČINKŮ

Při pohybu vozidla se mimo samotný pohyb vykreslují vybrané veličiny dynamiky vozu. Mezi tyto zobrazované prvky se řadí:

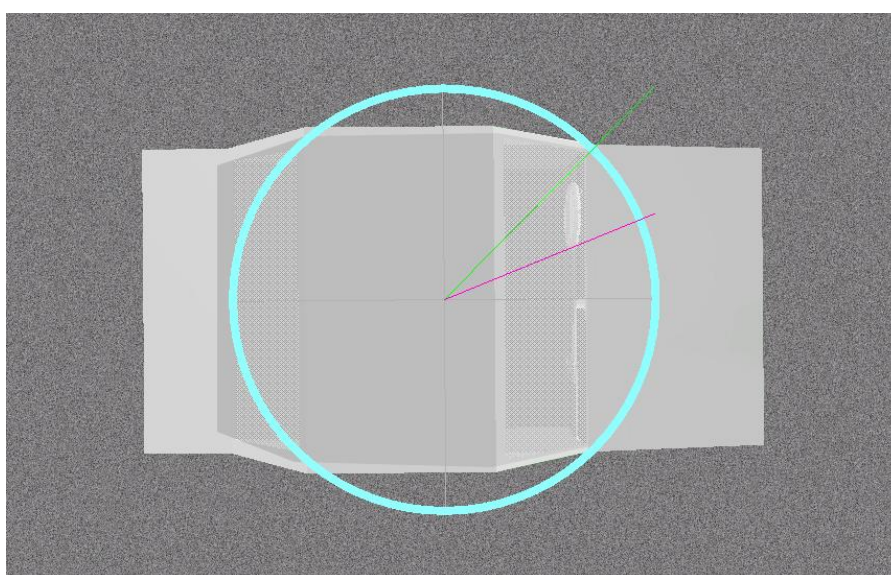
- výslednice podélné a boční síly na kolech
- svislé zatížení kol
- Kammovy kružnice
- úhly směrových úchylek
- velikost a směr rychlosti v těžišti vozu
- velikost a směr zrychlení vozu
- hranice přetížení 1g

K zobrazení bylo zapotřebí zvolit vhodný přepočet jednotek dynamických veličin k základní délkové jednotce používané ve virtuální realitě, tj. k metru. Jsou použity tyto přepočty:

• silové účinky	1 N	=	0,0001 m
• rychlost	1 km·h ⁻¹	=	0,01 m
• zrychlení	1 m·s ⁻²	=	0,1 m
• přetížení	1 g	=	0,981 m

5.4.1 RYCHLOST A ZRYCHLENÍ

S pohybem vozidla je svázán pohyb mezikruží světle modré barvy, které se vždy nachází v uvažovaném středu vozu ve výšce 0,5 metru nad střechou. Uprostřed mezikruží je vytvořen osový kříž. Vnitřní poloměr mezikruží představuje hranici přetížení 1g a zároveň rychlost 98 km·h⁻¹. Ze středu kříže vycházejí dvě čáry. Růžová čára znázorňuje velikost a směr horizontální rychlosti vozu. Horizontální rychlost je výslednicí podélné a příčné rychlosti. Zelenou čarou je značeno výsledné zrychlení v těžišti, které je opět výslednicí podélného a příčného zrychlení. Podobu tohoto prvku ukazuje obrázek 5.13.



Obr. 5.13 Zobrazení rychlosti a zrychlení



INDEXED LINE SET

Zobrazení účinků (síly, rychlost, zrychlení) pomocí čar je v podstatě jedinou možnou variantou. Změnou měřítka v jednotlivých osách lze dosáhnout změny délky a orientace čar. U žádného jiného tělesa nebo prvku by takové řízení délky a směru nebylo uskutečnitelné. Změnou měřítek by docházelo k nechtěným deformacím a změně tloušťek těles. Čáry a množiny čar se tvoří pomocí položky *IndexedLineSet*. Nevýhodou čar je fakt, že uživatel nemá možnost nastavit jejich tloušťku. Tloušťka totiž odpovídá velikosti jednoho obrazového bodu monitoru. Ke správnému zobrazení čar přispívá jejich umístění těsně nad povrch jiných těles. Zabráníme tak nechtěnému překrytí. [2; 9] Seznam parametrů uzlu *IndexedLineSet* ukazuje tabulka 5.1.

Tab. 5.1 Parametry *IndexedFaceSet* [9]

coord	prázdný uzel	seznam bodů v uzlu <i>Coordinate</i>
color	prázdný uzel	seznam barev v uzlu <i>Color</i>
coordIndex	[]	posloupnosti indexů vrcholů jednotlivých čar, ukončení číslem -1
colorIndex	[]	posloupnosti indexů barev pro vrcholy nebo čáry
colorPerVertex	TRUE	barvy v parametru <i>colorIndex</i> se vztahují na vrcholy, FALSE – na čáry

Zapsání čar středového kříže skládajícího se ze čtyř čar:

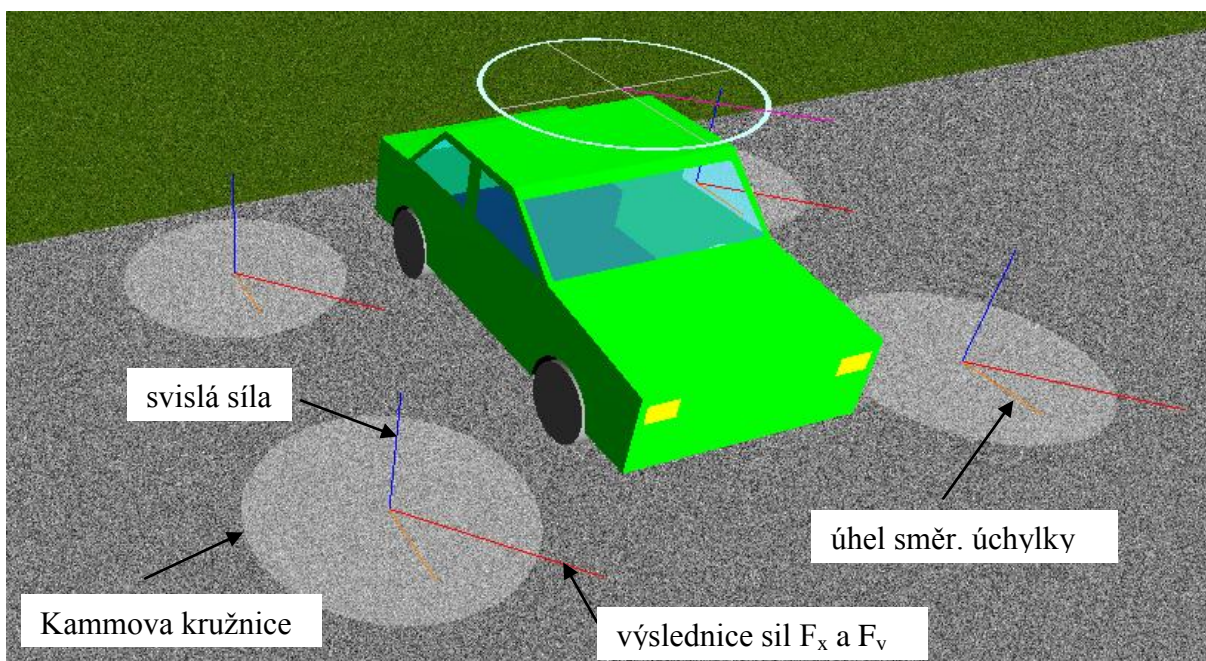
```
fprintf(fid, 'DEF kriz_stredovy Transform { \r\n');
fprintf(fid, 'children IndexedLineSet { \r\n');
fprintf(fid, 'color Color { \r\n');
fprintf(fid, 'color 0.7 0.7 0.7 \r\n');           %barva čar
fprintf(fid, '} \r\n');
fprintf(fid, 'coord Coordinate { \r\n');
fprintf(fid, 'point [ 0 0 0, \r\n');           %zápis souřadnic bodů
fprintf(fid, '0.981 0 0, \r\n');
fprintf(fid, '0 0 0.981, \r\n');
fprintf(fid, '-0.981 0 0, \r\n');
fprintf(fid, '0 0 -0.981 ] \r\n');
fprintf(fid, '} \r\n');
fprintf(fid, 'colorPerVertex FALSE \r\n');
fprintf(fid, 'coordIndex [ 0, 1, -1, 0, 2, -1, 0, 3, \r\n');   %zápis indexů čar
fprintf(fid, '-1, 0, 4, -1 ] \r\n');
fprintf(fid, 'colorIndex [ 0, 0, 0, 0 ] \r\n');
fprintf(fid, '} \r\n');
fprintf(fid, '} \r\n');
```



5.4.2 SILOVÉ ÚČINKY NA KOLECH

Každému ze čtyř kol vozidla přísluší množina tří čar a jedné plochy. Dvě čáry značí síly působící na kolo, třetí čára má konstantní délku a její vychýlení prezentuje úhel směrové úchylky kol dané nápravy. Červená čára přísluší výslednici podélné (F_x) a příčné (F_y) síly, modrá čára svislému zatížení (F_z) a oranžová úhlu směrové úchylky kola. Plocha tvaru kruhu znázorňuje kružnici přilnavosti, tzv. Kammovu kružnici. Průhlednost plochy je nastavena na 70%. Poloměr kružnice je závislý na zatížení kola ve svislém směru a hodnotě součinitele tření mezi vozovkou a pneumatikou. Jakmile koncový bod výslednice sil F_x a F_y překročí hranici kruhu, dojde ke ztrátě přilnavosti a následnému smyku kola. Zatížení vychází z výsledků simulace nebo z reálného měření. Součinitel tření vstupuje do modelu v Simulinku jako konstanta.

Soustava těchto prvků je umístěna po obou stranách vozidla tak, aby bylo zřejmé, ke kterému kolu patří. Nachází se 100 milimetrů nad rovinou vozovky.



Obr. 5.14 Prvky zobrazující dynamické účinky

5.4.3 VÝPIS HODNOT VYBRANÝCH VELIČIN

Posledním z prvků, jež mají za úkol informovat uživatele o nejdůležitějších veličinách dynamiky vozidla, je textové pole. Propojením textového pole se vstupními daty můžeme zobrazovat aktuální hodnoty vybraných veličin.

Samotné textové pole je doplněno o podklad (obdélníková plocha) světle modré barvy pro lepší čitelnost textu. Text je rozepsán na 7 řádků:

- souřadnice X, Y, Z
- podélná rychlost
- úhel směrové úchylky
- příčné zrychlení
- podélné zrychlení



Na prvních třech řádcích se zobrazují souřadnice vozidla, postupně X, Y a Z, vzhledem k souřadnému systému, který je shodný se souřadným systémem vozu před začátkem pohybu.

Textové pole je tvořeno uzlem *Text*. Nedovoluje velké úpravy a efekty textu. Jedná se v podstatě o jednoduchý plochý text. Základní parametry, které nastavujeme, najdeme v tabulce 5.2. Dále u textu lze například měnit velikost písma, vybrat z 3 základních typů písma, nastavit styl, řádkování, atd. [9]

Tab. 5.2 Základní parametry uzlu *Text* [9]

string	[]	posloupnost textových řetězců
fontstyle	prázdný uzel	styl písma v uzlu <i>Fontstyle</i>
maxExtent	0.0	nezáporná hodnota omezující maximální rozměr nápisu ve směru daném stylem písma, hodnota 0 ruší všechna omezení
length	[]	seznam nezáporných délek pro každý z řetězců, hodnota 0 znamená libovolnou délku

Vytvoření textového pole:

```
fprintf(fid, 'geometry DEF TEXT_hodnoty Text { \r\n');
fprintf(fid, 'string [ "Souradnice Xe:", \r\n');
fprintf(fid, "'Souradnice Ye:", \r\n');
fprintf(fid, "'Souradnice Ze:", \r\n');
fprintf(fid, "'Podelna rychlost:", \r\n');
fprintf(fid, "'Uhel smerove uchylky:", \r\n');
fprintf(fid, "'Pricne zrychleni:", \r\n');
fprintf(fid, "'Podelne zrychleni:" ] \r\n');
fprintf(fid, 'fontStyle FontStyle { \r\n');
fprintf(fid, 'family "TYPEWRITER" \r\n');
fprintf(fid, 'size 0.05 \r\n');
fprintf(fid, 'style "BOLD" \r\n');
fprintf(fid, '} \r\n');
```

Obr. 5.15 Textové pole pro výpis hodnot



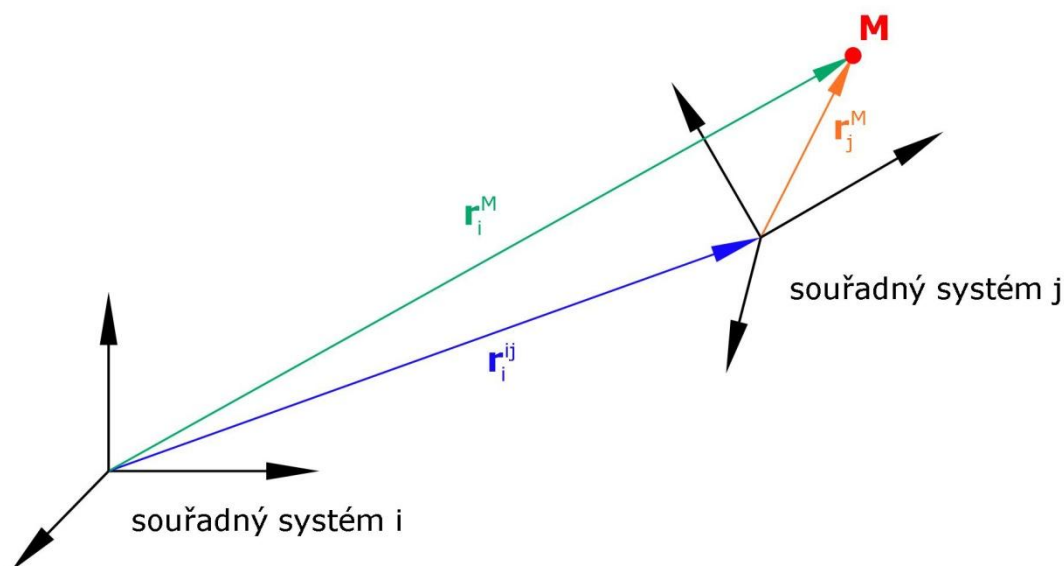
5.5 POHLEDY VE SCÉNĚ

Jedním z požadavků byla možnost prohlížet jízdu vozidla z více pohledů. Tyto pohledy, tzv. *viewpointy*, jsou definovány přímo ve zdrojovém kódu pro vytváření scény. Sledování pohybu vozu je možné z osmi základních stanovišť:

- start
- pohled zezadu
- pohled zepředu
- pohled na pravý bok
- pohled na levý bok
- pohled shora
- detail na přední část vozu
- pohled řidiče

U každého *viewpointu* se zadává pozice vzhledem k základnímu systému souřadnic. Další položkou je nastavení orientace. Bez změny orientace míří avatarův pohled automaticky do záporné osy z. Při změně hodnoty se pohled překlápí kolem definované osy o zadaný úhel.

Nejdříve byly pohledy vytvářeny pro nulové natočení vozidla vůči ose x. Natočením však vznikl problém a jednotlivé pohledy již neodpovídaly požadovaným pozicím. Proto bylo nutné využít vztahy pro transformaci polohy. Relativní pozice vůči vozidlu je známá a zůstává neměnná. Polohy *viewpointů* jsou tedy přepočítávány vzhledem k počátku základního souřadného systému.



Obr. 5.16 Transformace polohy



Vztah pro přepočítání polohy [22]:

$$r_i^M = C_{ij} \cdot r_j^M + r_i^{ij}, \quad (19)$$

kde:

r_i^M	[m]	polohový vektor, poloha bodu M v souřadném systému i
r_j^M	[m]	polohový vektor, poloha bodu M v souřadném systému j
r_i^{ij}	[m]	polohový vektor, poloha souřadného systému j v souřadném systému i
C_{ij}	[-]	matice směrových kosinů

Souřadný systém vozidla se natáčí kolem svislé osy. Pro tento pohyb má matice směrových kosinů tvar [22]:

$$C_{ij} = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Ukázka zápisu viewpointu v Matlabu:

```
C_oc=[cos(-natoc),-sin(-natoc),0;sin(-natoc),cos(-natoc),0;0,0,1]; %matice smerovych kosinu
r_ss=[car(1,1);-car(1,2);0]; %poloha pocatku SS auta vuci pocatku zakladniho SS
r_c1=[-3;7;0]; %poloha viewpointu v SS auta
r_o1=C_oc*r_c1+r_ss; %poloha bodu vuci zakladnimu SS
fprintf(fid, 'DEF START Viewpoint { \r\n');
fprintf(fid, 'orientation 0 -1 0 %12.7f \r\n',(uhel-natoc));
fprintf(fid, 'position %12.7f 1.60 %12.7f \r\n',r_o1(1),r_o1(2));
fprintf(fid, 'description "start" \r\n');
fprintf(fid, ' } \r\n');
```

5.6 PŘEDÁVÁNÍ UDÁLOSTÍ MEZI UZLY

Parametry uzlů ve VRML můžeme rozdělit do několika skupin. Jsou to parametry statické (*field*), parametry schopné přijímat události (*eventIn*), parametry schopné vysílat události (*eventOut*) a parametry schopné přijímat i vysílat události (*exposedField*). Těchto vlastností parametrů se využívá při tvorbě dynamických scén. Jako příklad může posloužit situace, kdy avatar přichází do určitého prostoru a při vstupu se automaticky rozsvítí světlo nebo začne hrát hudba apod.

K vysílání a přijímání událostí jinými uzly je určen příkaz *ROUTE*. Struktura zápisu tohoto příkazu vypadá následovně [9]:

```
ROUTE A.position_changed TO B.set_translation
```

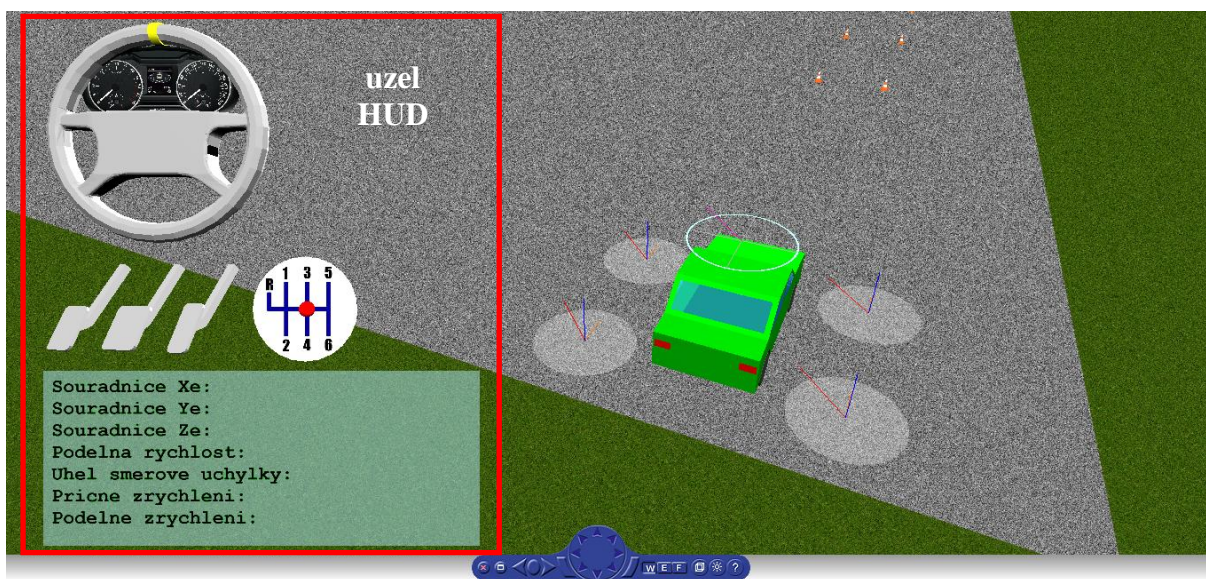
V příkladu je vysílána událost z uzlu A do uzlu B. Při změně hodnoty parametru *position* uzlu A dojde k automatické změně hodnoty parametru *translation* uzlu B.

V diplomové práci je příkaz *ROUTE* použitý pro synchronizaci pohybu kamer (pohledů) s pohybem vozidla. To by bylo možné provést také připojením vstupního signálu k parametrům *position*, příp. *orientation* příslušných pohledů. Zvolená varianta je však jednodušší a nenavyšuje se tak zbytečně počet řízených parametrů.



Ve druhém případě využití dochází k předávání informací o pohybu a orientaci avatara uzlu HUD (Head Up Display). Tento uzel obsahuje model volantu, pedálů, ukazatele zařazeného rychlostního stupně a textové pole pro výpis hodnot. Požadavkem je, aby se celý uzel pohyboval společně s avatarem, to znamená, aby se neměnila relativní poloha a velikost tohoto uzlu vůči avatarovi. K tomuto účelu bylo nutné využít další prvek virtuální reality a to *ProximitySensor* (detektor přítomnosti). Senzor (v práci pojmenovaný *avatar*) pracuje s virtuálním neviditelným kvádrem, jehož rozměry se nastavují v parametru *size*. Detektor sleduje, zda se avatar pohybuje v daném kvádru. Pokud tomu tak je, lze vysílat informace o jeho poloze a natočení uzlům typu Transform. Je potřeba volit rozměry kvádrů adekvátně ke konkrétní aplikaci. Pro účel zobrazení pohybu vozidla je zapsán rozměr $1 \cdot 10^6$ m ve všech třech osách. K docílení požadovaného zobrazení uzlu HUD si musíme představit, že se na tento uzel avatar dívá z bodu $[0, 0, 0]$. Tomuto faktu je potřeba uzpůsobit polohu uzlu. Ta je nastavena pro zobrazení v levé části obrazovky. Zápis detektoru přítomnosti s následným vysíláním události:

```
fprintf(fid, 'DEF avatar ProximitySensor { \r\n');
fprintf(fid, 'size 1000000 1000000 1000000 \r\n');
fprintf(fid, 'enabled TRUE \r\n');
fprintf(fid, '}' \r\n');
fprintf(fid, 'ROUTE avatar.orientation_changed TO HUD.set_rotation \r\n');
fprintf(fid, 'ROUTE avatar.position_changed TO HUD.set_translation \r\n');
```

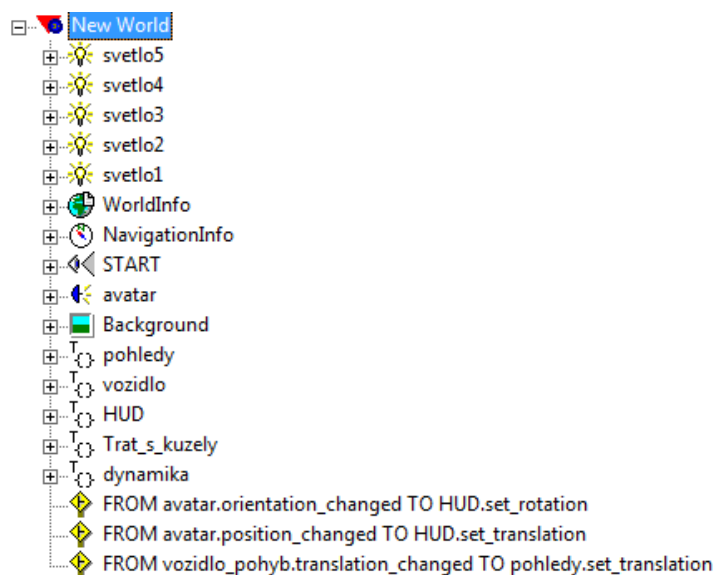


Obr. 5.17 Umístění uzlu HUD

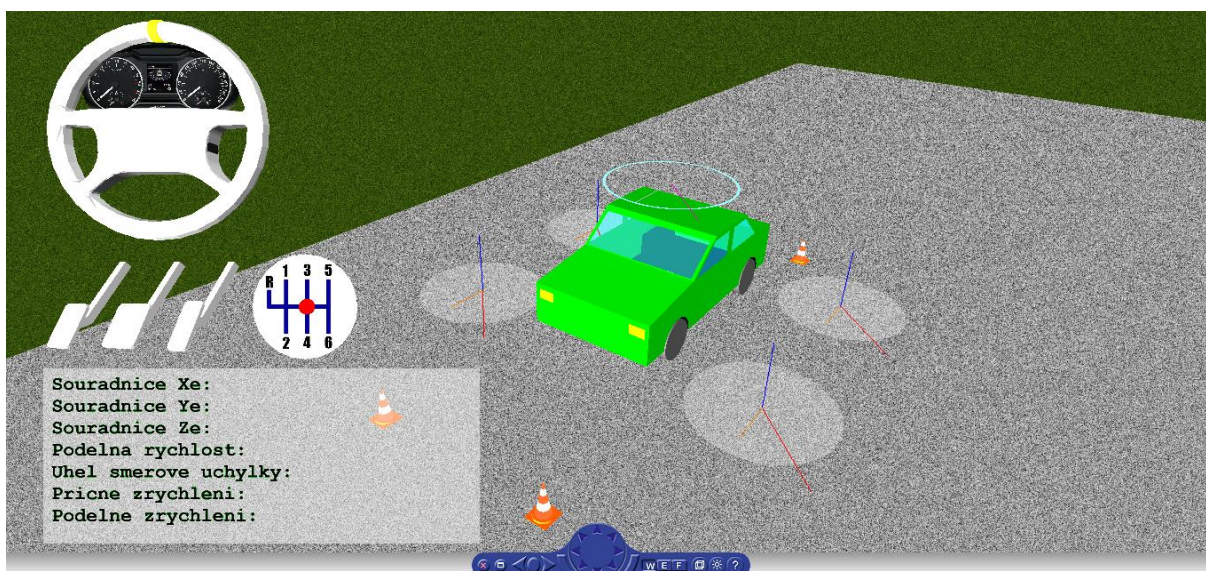


5.7 KONEČNÝ VZHLED SCÉNY

Na obrázku 5.18 se nachází stromová struktura souboru. Soubor virtuální reality obsahuje: pozadí, zdroje světla, detektor přítomnosti, pohledy, geometrie vozidla, uzel HUD, podoba tratě, prvky pro zobrazení dynamiky vozidla a příkazy pro vysílání a přijímání událostí příslušnými uzly. Pro všechny uvažované varianty testů zůstává struktura stále stejná. Jediný rozdíl je v obsahu uzlu *Trat* (příp. *Trat_s_kuzely*). Soubor je uložen pod názvem *test_area.wrl*



Obr. 5.18 Stromová struktura vytvořené scény



Obr. 5.19 Finální podoba scény



6 MODEL V PROSTŘEDÍ SIMULINK

V poslední části bude popsáno sestavení modelu pro řízení pohybu vozu v souboru *test_area.wrl*. V době zpracování diplomové práce nebyly bohužel k dispozici data z reálného měření. Jako alternativa slouží výsledky simulace jednostopého modelu pro skokové natočení a impuls na volantu. Tyto výsledky mně byly poskytnuty vedoucím práce. Data byla následně uložena do matic v potřebných tvarech. V prostředí Matlab/Simulink je vytvořen model skládající se z jednotlivých bloků a subsystémů, které upravují a doplňují vstupní hodnoty. Tyto signály pak vstupují přes blok VR Sink k řízeným uzlům. Dochází tak k rozpočítání, do této chvíle statické, scény.

6.1 PŘÍPRAVA VSTUPNÍCH DAT

Dodány byly dva soubory dat, uložených ve strukturách, s názvy *INV.mat* (impuls na volantu) a *SNV.mat* (skokové natočení volantu). Součástí byla i data, která nebyla pro účel této práce potřebná. Použity tedy byly pouze vybrané veličiny.

Data se načítají pomocí bloku *From file* ze souboru typu *.mat. To znamená, že jednotlivé hodnoty musí být uloženy v matici. Podmínkou je, aby v prvním řádku matice byl zapsán časový okamžik (vzorkovací čas). Poskytnutá data jsou zaznamenána s krokem 0,005s. Oba manévry trvaly 6s. V dalších řádcích jsou již uloženy aktuální hodnoty jednotlivých veličin. S využitím sestaveného skriptu *nacteni_dat.m* došlo k výběru a seřazení dat do požadovaného tvaru. Dále proběhlo uložení výsledného výběru do matic *manevrINV.mat* a *manevrSNV.mat*. Matice je velikosti 31x1201, tzn: 31 řádků a 1201 sloupců. Data jsou zapsána v tomto pořadí:

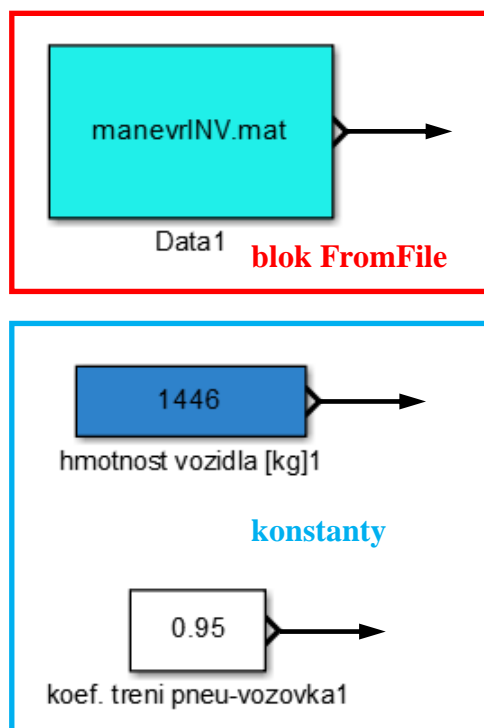
- | | |
|---|----------------------|
| 1. Časový okamžik | [s] |
| 2. Úhel směrové úchylky v těžišti vozidla | [°] |
| 3. Derivace úhlu směrové úchylky | [°·s ⁻¹] |
| 4. Úhel stáčení vozidla | [°] |
| 5. Stáčivá rychlost | [m·s ⁻¹] |
| 6. Derivace stáčivé rychlosti | [m·s ⁻²] |
| 7. Podélná rychlost vozidla | [m·s ⁻¹] |
| 8. Příčná rychlost vozidla | [m·s ⁻¹] |
| 9. Horizontální rychlost vozidla | [m·s ⁻¹] |
| 10. Podélné zrychlení vozidla | [m·s ⁻²] |
| 11. Příčné zrychlení vozidla | [m·s ⁻²] |
| 12. Natočení volantu | [°] |
| 13. Natočení předních kol | [°] |
| 14. Svislé zatížení levého předního kola | [N] |
| 15. Svislé zatížení pravého předního kola | [N] |
| 16. Svislé zatížení levého zadního kola | [N] |
| 17. Svislé zatížení pravého zadního kola | [N] |
| 18. Boční síla působící na kola přední nápravy | [N] |
| 19. Boční síla působící na kola zadní nápravy | [N] |
| 20. Podélná síla působící na kola zadní nápravy | [N] |
| 21. Podélná síla působící na kola zadní nápravy | [N] |
| 22. Úhel směrové úchylky kol přední nápravy | [°] |
| 23. Úhel směrové úchylky kol zadní nápravy | [°] |
| 24. Souřadnice X vozidla | [m] |
| 25. Souřadnice Y vozidla | [m] |



26. Souřadnice Z (výšková) vozidla	[m]
27. Natočení plynového pedálu	[°]
28. Natočení brzdového pedálu	[°]
29. Natočení spojkového pedálu	[°]
30. Otáčky motoru	[min ⁻¹]
31. Zařazený rychlostní stupeň	[-]

Jelikož nebyly k dispozici výsledky z měření, nebyla k dispozici také hodnota natočení pedálu plynu, brzdy ani spojky, otáčky motoru a zařazený rychlostní stupeň. Dle hodnoty rychlosti lze soudit, že při manévru nedocházelo ani k akceleraci, ani brzdění, ani řazení. Pro názornost animace však byly použity smyšlené hodnoty natočení pedálu spojky, otáček motoru a zařazeného rychlostního stupně. Naplnění příslušných řádků hodnotami proběhlo v Matlabu pomocí příkazu *linspace*.

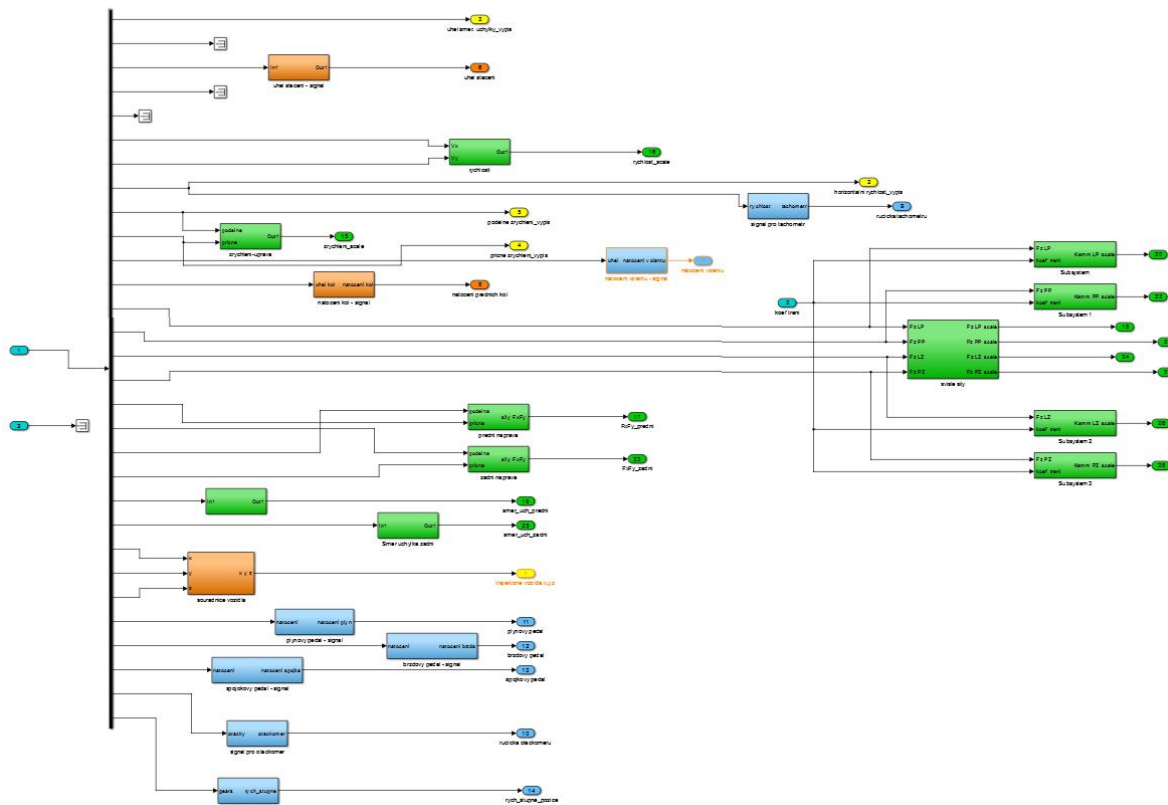
Do modelu vstupuje kromě výše vypsanych veličin také hmotnost vozidla a koeficient tření mezi pneumatikou a vozovkou. Simulace byla prováděna pro automobil o hmotnosti 1446 kg. Hodnota součinitele tření byla zvolena 0,95.



Obr. 6.1 Vstupní bloky

6.2 VYTVOŘENÍ SIGNÁLŮ

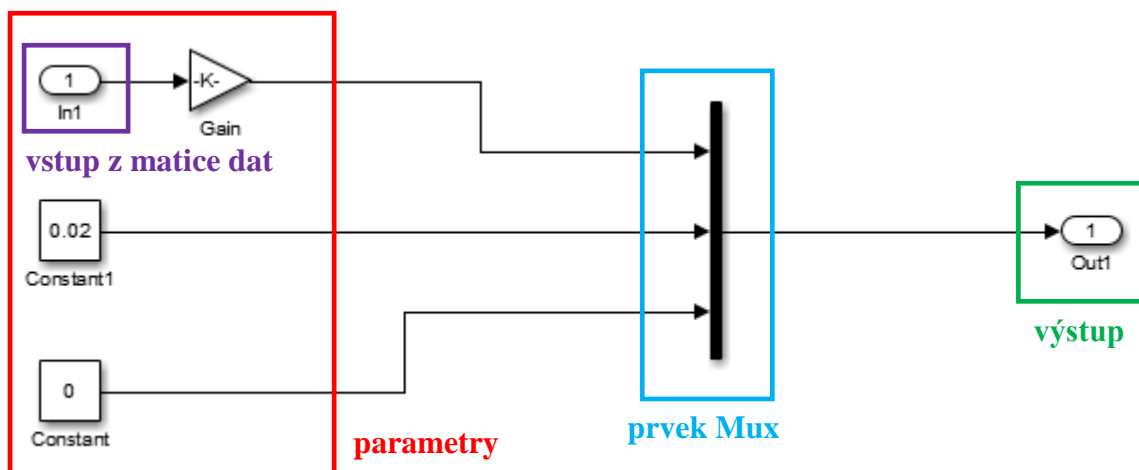
Hlavní a nejdůležitější část modelu v Simulinku tvoří subsystém nazvaný *Vytvoření signálu*. Tento blok se nachází napravo od vstupních parametrů a má zelenou barvu. Uvnitř subsystému jsou vytvářeny potřebné signály vstupující dále do virtuální reality. V podstatě se zde vyskytují 3 druhy signálů – signál pro translaci, rotaci a měřítko. Každý z těchto signálů musí mít předepsaný tvar, který je popsán v normě ISO pro VRML. Tvar každého z uvedených parametrů byl popsán v kapitole 2.5. Tvorba jednotlivých druhů signálů bude vysvětlena v následující části.



Obr. 6.2 Obsah subsystému - Vytvoření signálu

6.2.1 SIGNÁL TRANSLACE

Signál je určen pro parametr translation uzlu Transform a slouží k přepisu jeho hodnot. Musí se skládat ze tří složek – pozice ve směru X, ve směru Y a ve směru Z. Vektor se tvoří blokem Mux. Ukázka signálu translace je na obrázku 6.3.

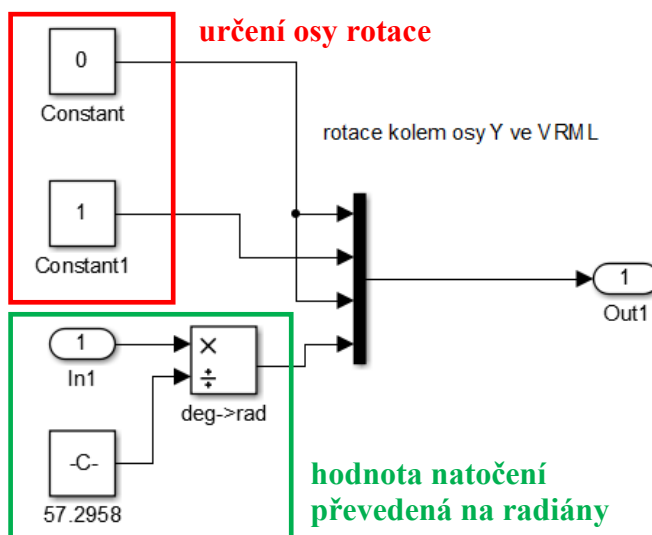


Obr. 6.3 Signál translace



6.2.2 SIGNÁL ROTACE

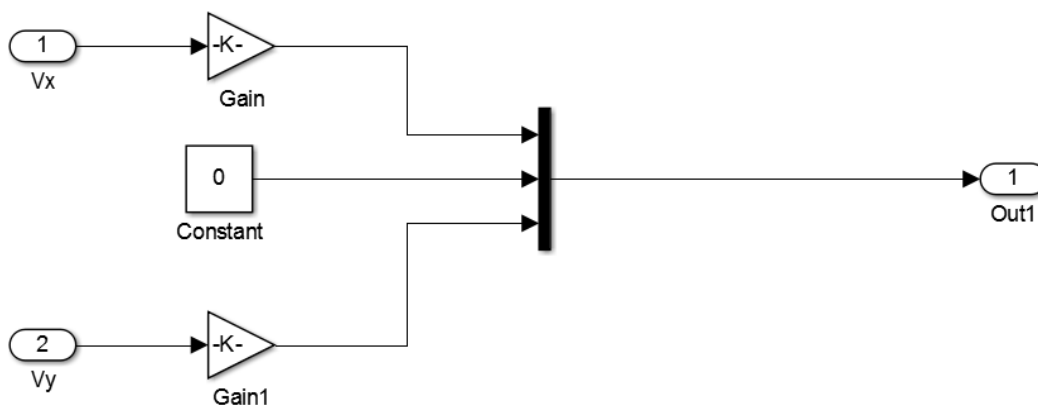
Signál rotace je vysílán do parametru rotation uzlu Transform, kde přepisuje uložené hodnoty. Skládá se ze 4 složek – první 3 udávají směr osy rotace a čtvrtá složka je úhel natočení v radiánech. Příklad signálu rotace je na obrázku 6.4.



Obr. 6.4 Signál pro rotaci

6.2.3 SIGNÁL MĚŘITKA

Signál, který vstupuje do parametru scale uzlu Transform. Jde o tří složkový vektor udávající hodnotu měřítka postupně v osách X, Y a Z. Signál měřítka se využíván pro ovládání prvků zobrazují dynamické účinky. Příklad signálu zobrazuje obrázek 6.5.



Obr. 6.5 Signál pro změnu měřítka



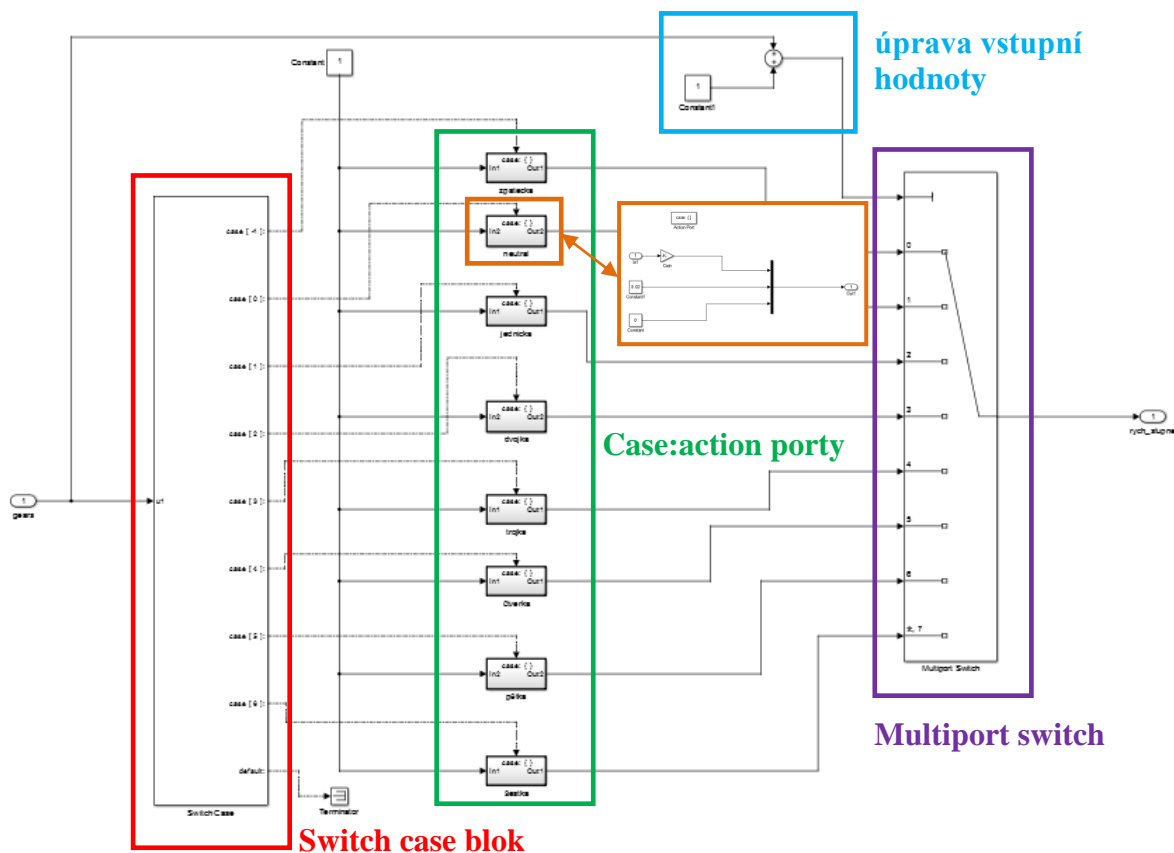
6.2.4 SIGNAL PRO ZOBRAZENÍ RYCHLOSTNÍHO STUPNĚ

Tento signál je v podstatě signál pro translaci. Jeho tvorba je však odlišná od zbylých signálů translace. Signálem je ovládána poloha kuličky. Její pozice je závislá na vstupní hodnotě uložené ve v matici v posledním řádku. Vstup může nabývat hodnot: -1 (zpátečka), 0 (neutrál), 1-6 (rychlostní stupně pro pohyb dopředu).

Signál je vytvářen v subsystému *rychlostní stupně*. Přivedená vstupní hodnota vstupuje do bloku *Switch case*. V tomto bloku jsou definovány přípustné vstupní hodnoty. Počet definovaných tzv. case podmínek určuje počet výstupů z bloku. V tomto případě je tedy 8 uživatelsky definovaných výstupů plus 1 defaultní.

Pro každý výstup je vložen blok *case:Action Port*, se kterým je propojen. V těchto blocích je již tvořen „klasický“ signál pro translaci. Pomocí konstant jsou udány pozice kuličky pro jednotlivé rychlostní stupně. Výstupem z bloku *case:action port* jsou tři složkové vektory.

V posledním kroku se přivádí 8 těchto vektorů na vstupy bloku *Multiport Switch*. Jeho úkolem je v závislosti na vstupní hodnotě rychlostního stupně přivést na výstup odpovídající vektor. Vstupy do bloku jsou číslovány od 0, tudíž bylo nutné vhodně upravit vstupní hodnotu. To je provedeno v horní části subsystému. Uvažujme, že je zařazen například první rychlostní stupeň. Hodnota 1 vstupuje do bloku *Switch case* a zároveň v upravené formě do bloku *Multiport Switch*. *Switch case* zpřístupní cestu k příslušnému *case:action port* a vytvoří se odpovídající vektor udávající pozici. V ostatních sedmi *case:action portech* vznikne vektor o složkách $[0, 0, 0]$. Současně dojde v přepínači k propojení výstupu se vstupem označeným hodnotou 2. Na výstupu se objeví vektor $[-0.037, -0.04, 0]$.



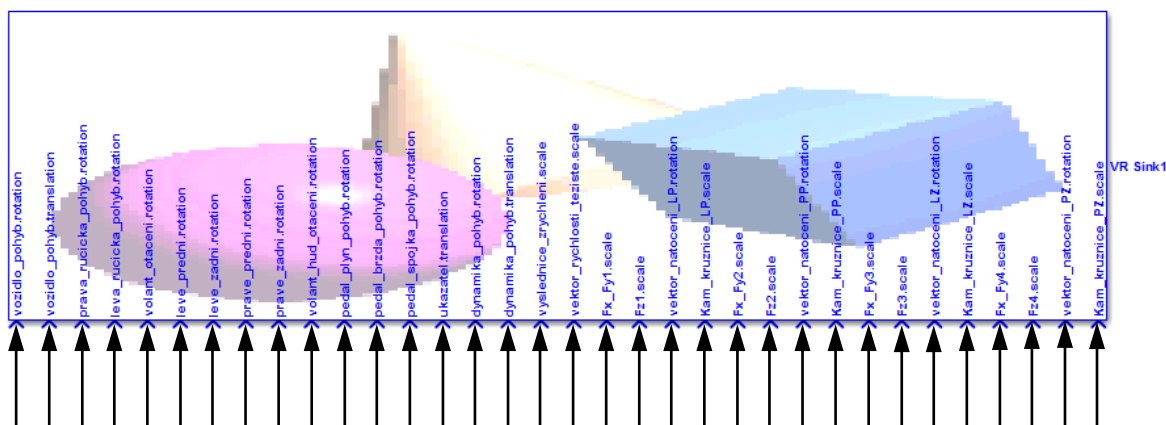
Obr. 6.6 Subsystém pro signál ovládající pozici ukazatele rychlostního stupně



6.3 BLOK VR SINK

Po vytvoření všech potřebných signálů již nic nebrání jejich propojení s parametry ovládaných uzlů v souboru virtuální reality. Tuto akci zajišťuje blok VR Sink, jež je součástí toolboxu Simulink 3D Animation.

Parametry bloku byly uvedeny v kapitole 1.2 na obrázku 1.7. V levé části je nastaven zdrojový soubor, v tomto případě *test_area.wrl*. V pravé části se zobrazí jeho stromová struktura. U jednotlivých parametrů se nachází zaškrtnávací políčko. Jeho zaškrtnutím říkáme, že chceme řídit právě daný parametr. Počet takto označených parametrů udává počet vstupů do bloku VR Sink. Řídit lze pouze uzly pojmenované ve VRML příkazem DEF. Parametrům musí být přiveden signál správného typu (velikosti).



Obr. 6.7 Blok VR Sink se znázorněnými vstupy

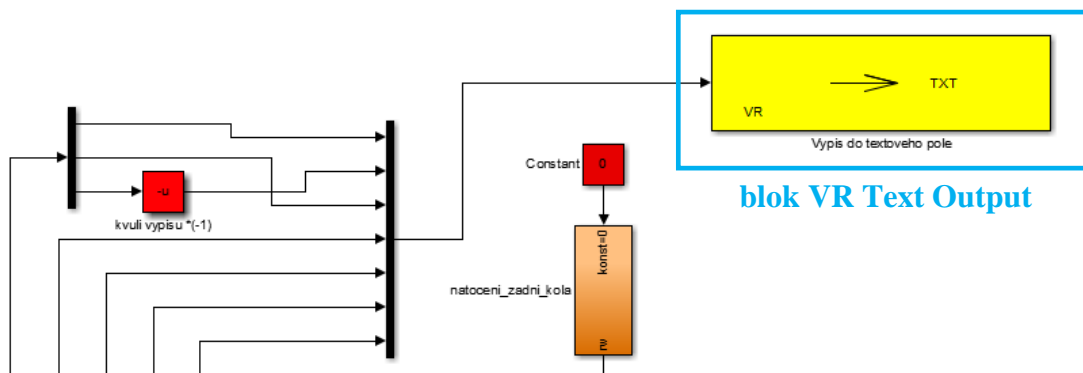
6.4 VR TEXT OUTPUT

Dalším použitým blokem z toolboxu 3D Animation je blok VR Text Output, který umožňuje prepisování uzlu obsahujícího textové pole. V diplomové práci je využit pro výpis aktuálních hodnot vybraných veličin do textového pole uzlu HUD.

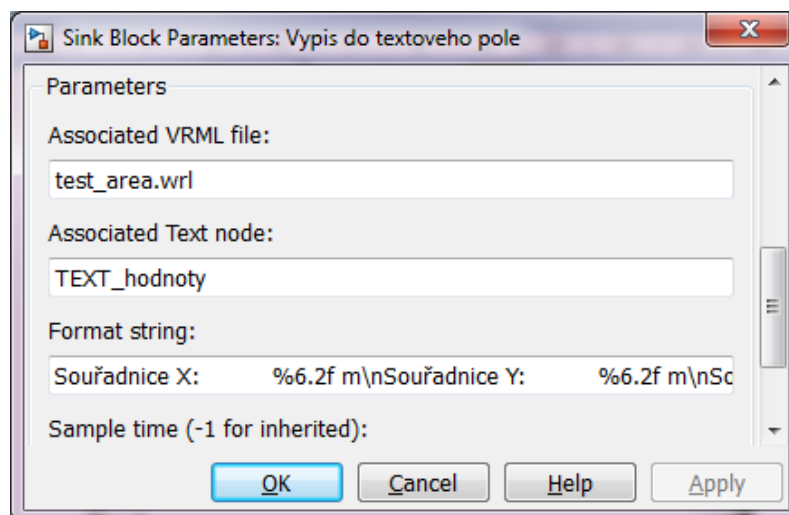
Blok je v práci zvýrazněn žlutou barvou. V jeho parametrech se nastaví název souboru, se kterým se má propojit, název textového uzlu (konkrétně *Text_hodnoty*) a formátovací řetězec. Formátovací řetězec má obdobný zápis jako při použití příkazu `fprintf` v Matlabu. Na místo, kde chceme vložit hodnoty ze vstupní matice, se zapíše znak procenta `%` následovaný informací o počtu číslic a formátu zobrazení hodnoty. Zápis řetězce:

Souřadnice X: `%6.2f m`
Souřadnice Y: `%6.2f m`
Souřadnice Z: `%6.2f m`
Horizontální rychlost: `%6.2f m/s`
Úhel směrové úchylky: `%6.2f deg`
Příčné zrychlení: `%6.2f m/s^2`
Podélné zrychlení: `%6.2f m/s^2`

Zápis `\n` značí přechod na nový řádek. Počet znaků procenta uvedených v řetězci určuje počet vstupních hodnot. Hodnoty vstupují do bloku ve formě vektoru, v diplomové práci tedy vektor se 7 hodnotami. První hodnota se vypíše na pozici prvního znaku procenta, atd. Signály jsou přivedeny ze subsystému *Vytvoření signálu* a následně převedeny na vektor prvkem *MUX*, viz obrázek 6.8.



Obr. 6.8 Přivedení signálu do bloku VR Text Output



Obr. 6.9 Parametry bloku VR Text Output



ZÁVĚR

Cílem této diplomové práce bylo vytvořit prostředí, které dokáže zobrazit jízdu vozidla na základě vložených dat. Data mohou pocházet přímo z reálného měření na vozidle nebo to mohou být výsledky simulačních výpočtů. Podobné úkoly již byly v minulosti řešeny. Jejich malou nevýhodou však je, že obsahují modely vozů a tratí, jež jsou poměrně časově složité na vytvoření. Požadavkem tedy bylo urychlit etapu vytvoření reprezentace vozidla a jízdni tratě.

Model vozidla a tratě je tvořen ve virtuální realitě. Tu popisuje programovací jazyk VRML 97. Bylo tedy nutné se s tímto jazykem dobře seznámit. Jedním z úkolů bylo sestavit základní model vozidla s editovatelnými parametry. Pro zadávání parametrů byla naprogramována aplikace pomocí jednoho z toolboxů programu Matlab. Ta spouští výpočet geometrie vozu dle zadaných rozměrů a zapisuje vozidlo do virtuálního světa. Zápis umožňuje použití speciálního příkazu, který dokáže vypisovat data do textového souboru. Společně s vozidlem je vytvářena okolní scéna, tzn. testovací dráha. V současné době jsou připraveny tři varianty dráhy. První vychází z normalizovaného testu ISO 3888, druhá je určena k tzv. kruhovému testu a třetí varianta je volitelná. Uživatel může zadat pozice vytyčovací kuželů v tabulce programu MS Excel. Takto vytvořený soubor s virtuální realitou lze dále ovládat za pomoci softwaru Matlab/Simulink.

V prostředí programu Simulink byl z funkčních bloků sestaven model umožňující propojení dat uložených v matici právě s vytvořeným souborem virtuální reality. Přivedením upravených signálů ke správným uzlům dojde k rozpočívání do té chvíle statického modelu vozidla. Zároveň se vypisují a vykreslují vybrané dynamické účinky. Je možné sledovat také natáčení volantu, pohyb pedálů a ukazatel rychlostního stupně. Vytvoření signálu pro zobrazení zařazeného rychlostního stupně je možné více způsoby. Kromě zmíněného v kapitole 6.2.4 byla uvažována také možnost zapsání jednotlivých pozic ukazatele přímo do matice dat a to pomocí přiloženého skriptu *gears.m*. V souboru stačí přepsat název proměnné a skript spustit. V tomto případě by nebylo potřeba relativně složitého subsystému (obrázek 6.6). Vstupní signály by bylo možné rovnou posílat do určených uzlů ve stromové struktuře souboru virtuální reality.

Vytvořené prostředí je uzpůsobené pro zobrazení jízdy vozidla po rovině. Může být využito pro zpětnou vizualizaci a získání lepší představy o tom, jak probíhal provedený testovací manévry nebo zkouška. Díky vykreslování dynamických účinků lze sledovat chování automobilu při různých situacích.

Je představeno základní rozhraní, které může být v budoucnu v případě potřeby rozšířeno. Lze dodefinovat další varianty testů nebo například vymodelovat závodní okruh. Jistě by bylo užitečné také doplnit možnost zvolit z různých typů karoserií. Vstupní hodnoty pro animaci lze jednoduše měnit nahráním nové matice s daty. Zajímavá je však i myšlenka ovládání modelu pomocí vhodného hardware vybavení.



POUŽITÉ INFORMAČNÍ ZDROJE

- [1] BLAŽÁK, Ondřej.: *Analýza měření jízdní dynamiky vozidel s využitím matematického modelování*. Brno. 2010. Dizertační práce na FSI VUT v Brně na Ústavu automobilního inženýrství. 92 s. Vedoucí dizertační práce doc. Ing. Zdeněk Kaplan, CSc.
- [2] GERGELITSOVÁ, Šárka. *VRML v příkladech*. 1. vyd. Praha: BEN - technická literatura, 2004, 223 s. ISBN 80-730-0138-1.
- [3] GILLESPIE, T. D. *Fundamentals of Vehicle Dynamics*. 1. vyd. Warrendale: Society of Automotive Engineers, 1992, 495 s. ISBN 15-609-1199-9.
- [4] ISO 3888-1. *Passenger cars - Test track for a severe lane-change manoeuvre: Part 1: Double lane-change*. Geneva (Švýcarsko): ISO copyright office, 1999.
- [5] ISO 3888-2. *Passenger cars - Test track for a severe lane_change manoeuvre: Part 2: Obstacle avoidance*. Geneva (Švýcarsko): ISO copyright office, 2002.
- [6] KARBAN, Pavel. *Výpočty a simulace v programech Matlab a Simulink*. Brno: Computer Press, 2006, 220 s. ISBN 80-251-1301-9.
- [7] VLK, František. *Dynamika motorových vozidel /*. vyd. 2. Brno: Prof.Ing.František Vlk,DrSc., nakladatelství a vydavatelství, 2006, 432 s. ISBN 80-239-0024-2.
- [8] ZAPLATÍLEK, K., DOŇAR, B. *MATLAB: tvorba uživatelských aplikací*. 1. vyd. Praha: BEN, 2004, 215 s. ISBN 80-730-0133-0.
- [9] ŽÁRA, Jiří. *VRML 97: laskavý průvodce virtuálními světy*. Vyd. 1. Praha: Computer Press, 1999. ISBN 80-722-6143-6.
- [10] PORTEŠ, Petr. *Dynamika vozidel*. (přednáška) Brno: ÚADI FSI VUT v Brně, 2012.
- [11] BREUER, Thorsten. *Advanced Dynamic Motorbike Analysis and Driver Simulation*. [online]. [cit. 2014-01-22]. Obrázek ve formátu GIF. Dostupné z: <http://www.ika.rwth-aachen.de/forschung/veroeffentlichung/1998/1998-6/index.php>
- [12] CONSTANT, Charlie. *The moose test (or VDA test)*. [online]. 2012-12-15 [cit. 2014-01-20]. Dostupné z: <http://www.car-engineer.com/the-moose-test-or-vda-test/>
- [13] HUMUSOFT. *MATLAB, Simulink*. [online]. 2014 [cit. 2014-04-19]. Dostupné z: <http://www.humusoft.cz/produkty/matlab/>
- [14] SAJDL, Jan. *Kammova kružnice přilnavosti*. [online]. [cit. 2014-01-20]. Dostupné z: <http://cs.autolexicon.net/articles/kammova-kruznice-prilnavosti/>
- [15] SAJDL, Jan. *Ackermannova podmínka*. [online]. [cit. 2014-01-20]. Dostupné z: <http://cs.autolexicon.net/articles/ackermannova-podminka/>



- [16] VEHI.CO, *ISO Double Lane Change Test*. [online]. [cit. 2014-01-20]. Dostupné z: <http://www.vehico.com/index.php/en/applications/iso-lane-change-test>
- [17] ŠKODA AUTO. *Interiér – Škoda Octavia*. [online]. [cit. 2014-04-15]. Obrázek ve formátu JPG. Dostupné z: <http://new.skoda-auto.com/cs/models/HotspotDetail?HotspotName=I2%20-%20Dashboard&WebID=411a062a-e81f-4f32-8016-b68c22443a7e&Page=interior>
- [18] ŠKODA AUTO. *Interiér – Škoda Octavia*. [online]. [cit. 2014-04-15]. Obrázek ve formátu JPG. Dostupné z: <http://www.skoda-auto.bh/models/HotspotDetail?HotspotName=I4%20-%20P%20C5%99%C3%ADstrojov%C3%A1%20deska%20a%20multifuk%C4%8Dn%C3%AD%20volant&WebID=e55963b0-5648-4d39-a20e-751fe7fc4286&Page=interior&view=Interior>
- [19] BRABEC, P., VOŽENÍLEK, R. *Jednostopý a dvoustopý model vozidla*. [online]. 2002. [cit. 2014-01-18]. Dostupné z: <http://www3.fs.cvut.cz/web/fileadmin/documents/12241-BOZEK/publikace/2002/Jednostopy%20a%20dvoustopy%20model%20vozidla.pdf>
- [20] KRŮZA, Libor. *Konstrukce zařízení na měření směrové tuhosti pneumatik*. [online] Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2011. 107 s. [cit. 2014-05-01]. Vedoucí diplomové práce Ing. Petr Porteš, Dr. Dostupné z: https://dspace.vutbr.cz/xmlui/bitstream/handle/11012/4283/DP_2011_Kruza_Libor.pdf?sequence=1&isAllowed=y
- [21] SZTWIERTNIA, Česlav. *Stabilita vozidla při jízdě v zatáčce*. [online]. Pardubice: Univerzita Pardubice, Dopavní fakulta Jana Pernera. 2006. 70 s. [cit. 2014-05-01]. Vedoucí bakalářské práce Juraj Slamka. Dostupné z: <http://library.upce.cz/Sources/D15706.pdf>
- [22] PORTEŠ, Petr. *Virtuální prototypy*. (přednáška) Brno: ÚADI FSI VUT v Brně, 2012.



SEZNAM POUŽITÝCH SYMBOLŮ

C_{coef}	$[(\text{N} \cdot \text{deg}^{-1}) \cdot \text{N}^{-1}]$	koeficient zatáčení
C_{ij}	$[-]$	matice směrových kosinů
C_{α}	$[\text{N} \cdot \text{deg}^{-1}]$	směrová tuhost pneumatiky
C_{af}	$[\text{m} \cdot \text{s}^{-1}]$	směrová tuhost obou pneumatik přední nápravy
C_{ar}	$[\text{m} \cdot \text{s}^{-2}]$	směrová tuhost obou pneumatik zadní nápravy
F_P	$[\text{N}]$	výslednice síly přenášené pneumatikou
F_x	$[\text{N}]$	podélná síla působící na kolo
F_y	$[\text{N}]$	boční síla
F_{yf}	$[\text{N}]$	boční síla na přední nápravě
F_{yr}	$[\text{N}]$	boční síla na zadní nápravě
F_z	$[\text{N}]$	svislé zatížení kola
H_f	$[\text{N}]$	podélná síla na přední nápravě
H_r	$[\text{N}]$	podélná síla na zadní nápravě
J_z	$[\text{kg} \cdot \text{m}^{-2}]$	moment setrvačnosti ke svislé ose z
l_f	$[\text{m}]$	vzdálenost přední nápravy od těžiště
l_r	$[\text{m}]$	vzdálenost zadní nápravy od těžiště
m	$[\text{kg}]$	hmotnost vozidla
O_v	$[\text{N}]$	vzdušný odpor
r_i^{ij}	$[\text{m}]$	polohový vektor, poloha souřadného systému j v souřadném systému i
r_i^M	$[\text{m}]$	polohový vektor, poloha bodu M v souřadném systému i
r_j^M	$[\text{m}]$	polohový vektor, poloha bodu M v souřadném systému j
U	$[\text{m} \cdot \text{s}^{-1}]$	podélná složka rychlosti
U_0	$[\text{m} \cdot \text{s}^{-1}]$	podélná složka rychlosti v základním souřadném systému
v	$[\text{m} \cdot \text{s}^{-1}]$	rychlost vozidla
\dot{v}	$[\text{m} \cdot \text{s}^{-2}]$	derivace rychlosti
V	$[\text{m} \cdot \text{s}^{-1}]$	příčná složka rychlosti
V_0	$[\text{m} \cdot \text{s}^{-1}]$	příčná složka rychlosti v základním souřadném systému
v_h	$[\text{m} \cdot \text{s}^{-1}]$	rychlost v horizontálním směru
v_x	$[\text{m} \cdot \text{s}^{-1}]$	podélná rychlost vozidla
v_y	$[\text{m} \cdot \text{s}^{-1}]$	boční (příčná) rychlost vozidla



x_0	[m]	poloha ve směru osy x
y_0	[m]	poloha ve směru osy y
α	[rad]	úhel směrové úchylky pneumatiky
$\dot{\alpha}$	[rad.s ⁻¹]	derivace úhlu směrové úchylky
β	[rad]	úhel směrové úchylky
δ	[rad]	úhel natočení předních kol
ε	[rad]	úhel stáčení, úhel mezi souřadnými systémy
$\dot{\varepsilon}$	[rad.s ⁻¹]	derivace úhlu stáčení (rychlost stáčení)
$\ddot{\varepsilon}$	[rad.s ⁻²]	úhlové zrychlení stáčení
μ	[-]	součinitel tření
μ_p	[-]	součinitel tření mezi kolem a vozovkou
ν	[rad]	směrový úhel
Ψ	[rad]	úhel stáčení vozidla

SEZNAM POUŽITÝCH ZKRATEK

3D	tří-rozměrný
area_x	rozměr plochy v ose x
area_z	rozměr plochy v ose z
car_pos	výraz pro pozice vozidla ve scéně
fov_h	označení předního převisu automobilu
GUIDE	Graphical User Interface Development Environment
HUD	Head Up Display
INV	impulz na volantu
ISO	International Organization for Standardization
LPP	levý přední podběh
LZP	levý zadní podběh
MS	Microsoft
RGB	barevný model, R- červená, G- zelená, B- modrá
rov_h	označení zadního převisu automobilu



SNV	skokové natočení volantu
SS	souřadný systém
VDA	Asociace německého automobilového průmyslu
VRML	Virtual Reality Modeling Language
w_radius	označení poloměru kola
w_width	označení šířky pneumatiky



SEZNAM OBRÁZKŮ

Obr. 1.1	Struktura práce	11
Obr. 1.2	Pracovní prostředí programu Matlab	12
Obr. 1.3	Zapsání příkazů v prostředí vestavěného editoru	13
Obr. 1.4	Vytvoření struktury se třemi poli	14
Obr. 1.5	Prostředí Matlab GUIDE	14
Obr. 1.6	Prostředí Simulinku s oknem knihoven funkčních bloků	15
Obr. 1.7	Parametry bloku VR Sink	16
Obr. 1.8	Bloky toolboxu 3D Animation	16
Obr. 1.9	Okno programu V-Realm Builder	17
Obr. 2.1	Vzhled prohlížeče Cosmoplayer	18
Obr. 2.2	Orientace souřadných systémů	19
Obr. 2.3	Vzhled pozadí a jeho zápis ve VRML	22
Obr. 2.4	Znázornění stromové struktury	23
Obr. 2.5	Možnosti nastavení vzhledu kužele	26
Obr. 3.1	Rozdělení tratě na úseky, rozmístění kuželů	27
Obr. 3.2	Nákres trati pro manévr ISO 3888-2	29
Obr. 3.3	Příklad měřených dat [12]	30
Obr. 3.4	Rozměry kužele	31
Obr. 4.1	Souřadný systém vozidla dle ISO	32
Obr. 4.2	Vozidlo v globálním souřadném systému	33
Obr. 4.3	Vznik skluzu: a) poháněné kolo, b) brzděné kolo [21]	34
Obr. 4.4	Závislost boční síly na úhlu směrové úchylky [10]	35
Obr. 4.4	Podélná síla pro různá zatížení v závislosti na skluzu [11]	35
Obr. 4.6	Závislost boční síly na brzděné síle [10]	37
Obr. 4.7	Kammova kružnice přilnavosti	37
Obr. 4.8	Jednostopý lineární model vozidla [7]	38
Obr. 5.1	Prostředí aplikace VRML CREATOR	41
Obr. 5.2	Vytvoření textového souboru příkazem fprintf	44
Obr. 5.3	Boční pohled na vozidlo s vyznačenými body	45
Obr. 5.4	Model vozidla	46
Obr. 5.5	Orientace souřadného systému testovací dráhy	47
Obr. 5.6	Rozložení scény testu dle ISO 3888	48
Obr. 5.7	Podoba scény pro kruhový test	49
Obr. 5.8	Zápis souřadnic v MS Excel	51
Obr. 5.9	3D a VRML model volantu	52
Obr. 5.10	Sestava pedálů	52
Obr. 5.11	Podoba ukazatele rychlostních stupňů	53
Obr. 5.12	Celek přístrojové desky	53
Obr. 5.13	Zobrazení rychlosti a zrychlení	54
Obr. 5.14	Prvky zobrazující dynamické účinky	56
Obr. 5.15	Textové pole pro výpis hodnot	57
Obr. 5.16	Transformace polohy	58
Obr. 5.17	Umístění uzlu HUD	60
Obr. 5.18	Stromová struktura vytvořené scény	61
Obr. 5.19	Finální podoba scény	61
Obr. 6.1	Vstupní bloky	63
Obr. 6.3	Signál translace	64
Obr. 6.2	Obsah subsystému - Vytvoření signálu	64
Obr. 6.4	Signál pro rotaci	65
Obr. 6.5	Signál pro změnu měřítka	65
Obr. 6.6	Subsystém pro signál ovládající pozici ukazatele rychlostního stupně	66
Obr. 6.7	Blok VR Sink se znázorněnými vstupy	67
Obr. 6.8	Přivedení signálu do bloku VR Text Output	68
Obr. 6.9	Parametry bloku VR Text Output	68
Obr. 6.10	Výsledný vzhled animace	69
Obr. 6.11	Propojení funkčních bloků modelu	69



SEZNAM TABULEK

<i>Tab. 2.1 Členění VRML souboru [9]</i>	<i>19</i>
<i>Tab. 2.2 Parametry uzlu Viewpoint [9]</i>	<i>20</i>
<i>Tab. 2.3 Parametry uzlu NavigationInfo [9]</i>	<i>20</i>
<i>Tab. 2.4 Parametry uzlu WorldInfo [9]</i>	<i>21</i>
<i>Tab. 2.5 Parametry uzlu Background [9]</i>	<i>21</i>
<i>Tab. 2.6 Parametry uzlu PointLight [9]</i>	<i>22</i>
<i>Tab. 2.7 Seznam základních parametrů uzlu Transform [9]</i>	<i>23</i>
<i>Tab. 2.8 Parametry základních těles [9]</i>	<i>24</i>
<i>Tab. 2.9 Parametry pro nastavení barvy tělesa [9]</i>	<i>25</i>
<i>Tab. 3.1 Rozměry tratě [4]</i>	<i>28</i>
<i>Tab. 3.2 Rozměry dráhy [5]</i>	<i>29</i>
<i>Tab. 5.1 Parametry IndexedFaceSet [9]</i>	<i>55</i>
<i>Tab. 5.2 Základní parametry uzlu Text [9]</i>	<i>57</i>



SEZNAM PŘÍLOH

Příložené CD obsahuje následující přílohy v elektronické podobě:

- texture složka s obrázky
- VRML_CREATOR.m spouštěcí soubor aplikace
- VRML_CREATOR.fig soubor s grafickou podobou aplikace
- vrml12.m funkce pro vytváření scény
- vrml3.m funkce pro vytváření scény
- vrml4.m funkce pro vytváření scény
- pohyb_vozidla.slx model v Simulinku
- test_area.wrl soubor s virtuální scénou
- manevrINV.mat upravený soubor s výsledky simulací
- manevrSNV.mat upravený soubor s výsledky simulací
- kuzely.xlsx vzorový soubor
- nacteni_dat.m soubor pro uložení dat
- gears.m soubor pro vytvoření pozic ukazatele
- ilustracni_obrazek.jpg
- plocha_s_kuzely.jpg