

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta Elektrotechnická
Katedra řídící techniky



Diplomová práce

Vizualizace dynamických modelů ve Virtual Reality Toolboxu v Matlabu

Lukáš JOZÍFEK

Vedoucí práce: Ing. Jiří Roubal, Ph.D.

Studijní program: Elektrotechnika a informatika magisterský M2612

Obor: Kybernetika a měření

2009

Prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v přiloženém seznamu.

V Praze dne.....9.1.2009.....

.....*František*.....
podpis

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra řídicí techniky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: Lukáš Jozínek

Studijní program: Elektrotechnika a informatika (magisterský), dobíhající
Obor: Kybernetika a měření, blok KM1 - Řídicí technika

Název téma: Vizualizace dynamických modelů ve Virtual Reality Toolboxu v Matlabu

Pokyny pro vypracování:

1. Seznamte se s Virtual Reality toolboxem v Matlabu a sepište stručný návod pro začátečníky v prostředí Virtual Reality Toolboxu.
2. Vytvořte model systému kulička na tyči (odvoďte matematicko fyzikální model, vytvořte simulinkový model s maskou pro zadávání fyzikálních konstant a parametrů, vytvořte virtuální realitu k modelu).
3. Navrhněte řízení pro tento model.
4. Vytvořte verzi modelu kulička na tyči pro porovnání s laboratorním modelem v laboratoři K26.
5. Vytvořte model systému dvou inverzních kyvadel na vozíku.
6. Navrhněte řízení pro tento model.
7. Model dvou inverzních kyvadel vytvořte ve dvou verzích (kyvadla ovlivňují/neovlivňují pohyb vozíku).
8. Vytvořte model Wattova odstředivého regulátoru.
9. Vytvořte demo regulační smyčku s Wattovým regulátorem z předchozího bodu (regulační smyčka má vstupní poruchovou veličinu).

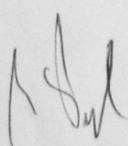
Seznam odborné literatury:

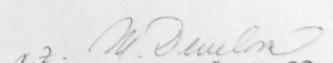
<http://cs.felk.cvut.cz/webis/en/people/zara.html>
<http://www.cgg.cvut.cz/LaskavyPruvodce/>
<http://www.knihy.cpress.cz/Book.asp?ID=158>
<http://www.mathworks.com/access/helpdesk/help/toolbox/vr/>

Vedoucí: Ing. Jiří Roubal, Ph.D.

Platnost zadání: do konce letního semestru 2008/2009

prof. Ing. Michael Šebek, DrSc.
vedoucí katedry



z.d. 
doc. Ing. Boris Šimák, CSc.
děkan

V Praze dne 20. 9. 2008

Poděkování

Rád bych na tomto místě poděkoval především mému vedoucímu diplomové práce Ing. Jiřímu Roubalovi, Ph.D. za jeho ochotu a trpělivost.. Dále bych rád poděkoval svým rodičům za podporu na studiu a všem, bez kterých by tato práce nemohla vzniknout.

Abstrakt

Tato práce se zabývá tvorbou virtuálních modelů v programu Matlab, pomocí virtual reality toolboxu. Model je tvořen dynamickým systémem v simulinku a trojrozměrnou scénou. Ta je popsána pomocí jazyka VRML (Virtual reality modeling language). Pro výuku návrhu regulátorů jsou vytvořeny modely „kulička na tyči“ a „dvě kyvadla na vozíku“. Pro tyto modely jsou navrženy regulátory PID (proporcionálně-integračně-derivační) a LQG (kvadraticky optimální řízení). Jako poslední model je zde ukázkový regulátor „Wattův odstředivý regulátor“, který stabilizuje otáčky parního stroje pomocí tlaku páry.

Abstract

This work deals with creating virtual models in Matlab program using the Virtual Reality Toolbox. Model is composed by Simulink dynamics system and three dimensional scene. It is written in VRML (Virtual reality modelling language). For controller design learning are created two models: „ball and beam“ and „two inverted pendulums“. These models are driven by PID and LQG controllers. Last model is „flyball governor“ that controls the speed of a steam engine by regulating the amount of steam.

Obsah

1.Úvod.....	1
2.Seznámení se s Virtual Reality Toolboxem	2
2.1.Virtual Reality Toolbox.....	2
2.2.Vytvoření virtuální scény - jazyk VRML.....	3
2.3.Zápis a čtení dat virtuální scény.....	8
2.4.Simulink a maska.....	10
3.Kulička na tyči.....	12
3.1.Model tyče.....	13
3.2.Model kuličky.....	15
3.3.Kompletní model.....	17
3.4.Identifikace reálného modelu.....	21
3.5.PID regulace kuličky na tyči.....	22
4.Vozík se dvěma kyvadly.....	24
4.1.Model vozíku.....	26
4.2.Model kyvadla.....	27
4.3.Kompletní model.....	29
4.4.Regulace inverzních kyvadel.....	31
5.Wattův odstředivý regulátor.....	33
5.1.Kompletní model.....	35
6.Závěr.....	37
Literatura.....	38

Seznam ilustrací

Obrázek 2.2.1: Souřadná soustava pro posun a rotaci.....	3
Obrázek 2.2.2: Příklad VRML kódu a jeho zobrazení.....	3
Obrázek 2.3.1: Dialog bloku VR Sink.....	8
Obrázek 2.4.1: Editor masky.....	10
Obrázek 3.1: Laboratorní model kuličky na tyči.....	12
Obrázek 3.1.1: Model tyče.....	13
Obrázek 3.2.1: Schématický nákres kuličky na tyči.....	15
Obrázek 3.2.2: Kulička v kolejničce.....	15
Obrázek 3.3.1: Vnitřní simulinkové schéma bloku „Ball and beam“.....	17
Obrázek 3.3.2: Simulinkové schéma systému tyče.....	18
Obrázek 3.3.3: Zapojení s PID reguátorem.....	18
Obrázek 3.3.4: Simulinkové schéma systému kuličky.....	18
Obrázek 3.3.5: Simulinkové schéma virtuální reality.....	19
Obrázek 3.3.6: Virtuální a reálný model kuličky na tyči.....	19
Obrázek 3.3.7: Nastavení parametrů modelu pomocí masky.....	20
Obrázek 3.4.1: Porovnání simulace a modelu.....	21
Obrázek 3.5.1: Zapojení modelu kuličky na tyči s PID regulátorem.....	22
Obrázek 3.5.2: Odezva na vstupní signál.....	23
Obrázek 3.5.3: Návrh PID pomocí GMK	23
Obrázek 4.1: Vozík se dvěma kyvadly.....	24
Obrázek 4.1.1: Schéma vozíku (blok cart).....	27
Obrázek 4.2.1: Schéma kyvadla neovlivňující vozík.....	28
Obrázek 4.2.2: Schéma kyvadla ovlivňující vozík.....	28
Obrázek 4.3.1: Model se dvěma kyvadly neovlivňující vozík.....	29
Obrázek 4.3.2: Model se dvěma kyvadly ovlivňující vozík.....	29
Obrázek 4.3.3: Blok virtuální reality.....	30
Obrázek 4.3.4: Nastavení parametrů modelu.....	30
Obrázek 4.4.1: LQR se sledováním konstantní reference.....	31
Obrázek 4.4.2: LQG řízení dvou kyvadel.....	32
Obrázek 5.1: Wattův regulátor.....	33
Obrázek 5.2: Schéma Wattova regulátoru.....	35
Obrázek 5.1.1: Zapojení kompletního modelu.....	35
Obrázek 5.1.2: Simulace ventilu.....	36
Obrázek 5.1.3: Nastavení regulátoru.....	36
Obrázek 6.1: Virtuální a reálný model kuličky na tyči.....	37

1. Úvod

Regulace a řízení je důležitým pojmem v mnoha odvětví lidské činnosti. Abychom se naučili řídit složité děje, je třeba začít na jednoduchých příkladech.

Úkolem této práce je vytvořit virtuální modely pro výuku návrhu regulátorů v programu Matlab a Simulink s využitím Virtual Reality toolboxu. Tyto modely mají za úkol zjednodušit seznámení se studenta s návrhem a použitím regulátorů na reálných modelech. Reálné laboratorní modely nohou být často nedostupné nebo přístup k nim časově omezený. Virtuální modely přenáší laboratoř ke studentovi i s výhodou možnosti pozorovat průběh regulace vizuálně. Tato vizualizace je mnohem přehlednější než grafy a tabulky. Další výhodou virtuálních modelů je možnost měnit jejich fyzické parametry. Těchto parametrů by nemělo být mnoho, aby bylo nastavení ještě přehledné.

Důležitá je také interakce modelu s uživatelem. Ve vizualizaci lze kliknout na požadované místo regulace a tento podmět je odeslán regulátoru.

Pro každý model je nutné odvodit jeho matematicko-fyzikální popis. Z něho vytvořit schéma v programu Simulink a připojit trojrozměrnou scénu pomocí Virtual Reality Toolboxu. Pro popis 3D scény je použit jazyk VRML.

2. Seznámení se s Virtual Reality Toolboxem

2.1. Virtual Reality Toolbox

Tato kapitola se zabývá Virtual Reality Toolboxem. Tento toolbox rozšiřuje program Matlab o možnost pracovat s trojrozměrným pohyblivým světem neboli virtuální realitou. Pro popis scény využívá jazyk VRML, který je standardem pro internetovou trojrozměrnou grafiku. O VRML pojednává další kapitola.

Nejdříve se krátce seznámíme s programem Matlab a jeho nadstavbou Simulink. Matlab je aplikace pro zpracování výpočtů a příkazů zadané pomocí příkazové řádky. Výpočty lze uložit jako posloupnost příkazů do souboru jako tzv. m-file. Podobně lze napsat i vlastní funkce, které se dále použijí při výpočtech.

Simulink je grafická nadstavba Matlabu. Každá funkce je reprezentovaná pomocí bloku se vstupy a výstupy. Grafické řešení je přehlednější a vhodnější pro dynamické systémy. Simulink umožňuje sloučení více bloků do jednoho subsystému. Pro zjednodušené nastavení parametrů funkcí lze vybrané parametry nahradit proměnnou. Tyto proměnné lze měnit pomocí dialogu, který se přiřadí k bloku subsystému. Tento dialog je nazván maska, protože zakrývá vnitřní zapojení subsystému a uživatel nastavuje pouze vybrané proměnné.

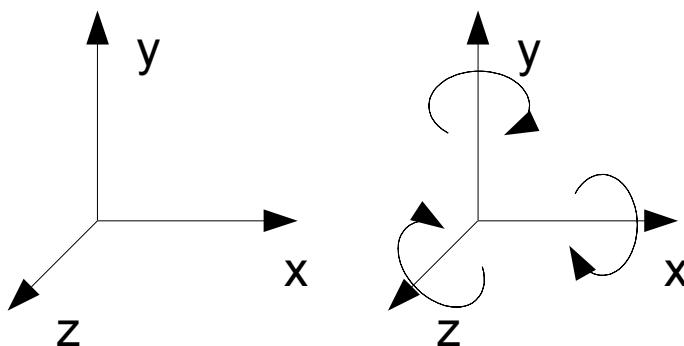
Matlab byl původně navržen pro výpočty matic, ale díky možnosti volného rozšíření o další výpočetní nástroje se z něj stal univerzální výpočetní program. Nástroje jsou rozřazeny do několika knihoven podle oboru využití (modelování dynamických systémů, finanční nástroje atd.). Knihovny neboli toolboxy jsou většinou totožné pro Matlab i Simulink, ale není to pravidlem.

K Virtual Reality Toolboxu lze přistupovat pomocí příkazů Matlabu nebo jej lze ovládat pomocí Simulinku. V této práci jsou použity oba přístupy. Na zobrazení trojrozměrné scény je použit blok Simulinku VR Sink a k nastavení virtuální scény se přistupuje pomocí příkazů Matlabu.

2.2. Vytvoření virtuální scény - jazyk VRML

V této kapitole jsou popsány základní pojmy jazyka VRML, tvorba trojrozměrné scény a příprava pro přidání do Virtual Reality Toolboxu.

VRML (Virtual Reality Modeling Language) je objektový jazyk se stromovou strukturou určený pro popis trojrozměrné grafické scény [1]. Stromová struktura 3D scény je reprezentovaná uzly zapisovanými způsobem `jméno_uzlu {vnitřní obsah}`. Uvnitř uzlu se mohou nacházet další uzly a atributy uzelů. To umožňuje jednoduše sestavovat několik objektů do skupin a pro tyto skupiny definovat společné parametry. Tím lze jednoduše určit i závislost pohyblivých částí na sobě. Například poloha kuličky je definovaná vzhledem k tyče a je nezávislá na úhlu natočení tyče, ale natáčí se s tyčí.



Obrázek 2.2.1: Souřadná soustava pro posun a rotaci

```
#VRML V2.0 utf8
DEF kvadr Transform {
    translation     0.5 0 0
    rotation       0 1 1 0.785
    children Shape {
        geometry Box {size 2 2 2}
        appearance Appearance {
            material Material {
                diffuseColor 0.8 0 0
            }
        }
    }
}
```

Obrázek 2.2.2: Příklad VRML kódu a jeho zobrazení

Na začátku souboru je vždy povinná hlavička, která určuje verzi a formát jazyka. Za hlavičkou následuje úvodní informace o virtuálním světě. Jsou to obecné informace a nadpis

okna (`WorldInfo`), zájmová stanoviště na světě (`Viewpoint`) a způsob procházení světem (`NavigationInfo`). Uzel `viewpoint` nastavuje polohu, úhel pohledu pozorovatele a jeho zorný úhel.

Základní uzly jsou skupinové. Tvoří je uzel skupina (`Group`), který pouze sjednocuje uzly a transformace (`Transform`). Uzel transformace je nejpoužívanější, protože obsahuje nastavení velikosti (`scale`), polohy (`position`) a natočení (`rotation`) podřízeného objektu. Aby bylo možné měnit tyto nastavení vzhledem k jinému bodu, než střed objektu, jsou definované další parametry. Posunutí osy otáčení (`center`) a orientace změny měřítka (`scaleOrientation`).

Vlastní objekt je definován v uzlu tvaru (`Shape`). Ten obsahuje geometrické tvary (`Geometry`) a vzhled (`Appearance`) objektu.

Geometrické tvary mohou být jednoduchá tělesa jako koule (`Sphere`), válec (`Cylinder`), krychle (`Box`) a kužel (`Cone`) nebo libovolný objekt složený z polygonů (`IndexedFaceSet`), výšková mapa (`elevationGrid`) nebo objekt vzniklý posunem profilu po trajektorii tzv. opláštění (`Extrusion`). Další uzly pro zobrazení čar (`IndexedLineSet`), bodů (`PointSet`) a textu (`Text`).

Nastavením vzhledu lze měnit barvu (`material`) nebo povrchovou texturu (`texture`). Uzlem materiálu se nastavuje barva (`diffuseColor`), intenzita (`ambientIntensity`), rozmanost odrazu (`shininess`) a jeho barva (`specularColor`). Dále pak vyzařovaná barva (`emissiveColor`) a průhlednost (`transparency`).

Textura může být obrázek (`ImageTexture`), video ve formátu MPEG1 (`MovieTexture`) nebo opakující se vzorek (`PixelTexture`).

Pro osvícení scény jde vybrat ze čtyř druhů světel. Zdroj rovnoběžných paprsků (`DirectionalLight`), který osvětuje scénu pod úhlem daným vektorem. Bodový zdroj (`PointLight`) se chová jako obyčejná žárovka s postupně se snižující intenzitou světla. Reflektor (`Spotlight`) je zdrojem kuželového světla s klesající intenzitou kolmo na směr svícení. Speciálním případem světla je čelní svítidla (`Headlight`), která je přichycena na kameře a svítí ve směru pohledu.

Jazyk VRML dovoluje přehrát zvukový soubor ve formátu WAVE a to s proměnnou intenzitou a směrem od zdroje zvuku ve scéně.

Na dokreslení okolí scény je zde mlha (`fog`), která vyzdvihne objekty v popředí a pozadí scény (`background`).

Pro opakované použití složeného modelu má jazyk VRML prototypy, které mají vlastní proměnné. Ty nastavují vybrané parametry uvnitř prototypu. Proměnné můžou být typu `eventIn`, `eventOut`, `exposedField` a `field`. Vstupní událost `eventIn` přijímá změny daného datového typu, např. `SFRotation`. Výstupní události jsou vysílány přes `eventOut`. Univerzální vstupní výstupní proměnná pro události je `exposedField`. Konstantní hodnota při inicializaci je typu `field`.

```
#VRML V2.0 utf8
PROTO kyvadlo
[   exposedField SFRotation naklon 0 1 0 0]
{   Transform {
        rotation IS naklon
        center 0 .5 0
        children [Shape { geometry Cylinder {
                height 1
                radius .02
            }
        }
        Transform {
            translation 0 -.7 0
            rotation 1 0 0 1.57
            children Shape { geometry Cylinder {
                height .1
                radius .3
            }
        }
    }
}
]
}
DEF Kyv kyvadlo {naklon 0 0 1 1} # zde se vytvoří objekt
```

Dynamika a interakce s uživatelem je zajištěna pomocí senzorů a událostí. Událost se generuje pokaždé, když se proměnný parametr změní.

Manipulátory jsou součástí senzorů, ale jejich použití je specializované na ovládání pomocí myši. Válkový manipulátor (`CylinderSensor`) převádí pohyb myši na rotaci válce a ukládá se do události `rotation_changed`. Lze nastavit minimální a maximální úhel otočení. Rovinný manipulátor (`PlaneSensor`) snímá pohyb myši na ploše a převádí je na lokální polohu manipulátoru jako událost `translation_changed`. Kulový manipulátor (`SphereSensor`) detekuje otáčení koule, která obepíná objekty ve stejném uzlu. Tato rotace je vrácena událostí `rotation_change`.

Senzory reagují na různé parametry scény. Senzor kolize (`Collision`) detekuje náraz mezi objekty. Pokud je objekt složitý, tak detekce nárazu je výpočetně náročná, proto lze pro detekci použít jednodušší tvar pomocí uzlu proxy. Detektor přítomnosti (`ProximitySensor`) generuje událost při vstupu kamery do prostoru ohraničeného imaginárním kvádrem o velikosti `size`. Pokud je kamera uvnitř oblasti, tak se pomocí událostí zjistí poloha (`position_changed`) a orientace (`orientation_changed`) kamery. Důležitým typem je časovač (`TimeSensor`), který vyšle událost v daný čas (`startTime`) nebo periodicky za interval daný parametrem `cycleInterval`. Periodické opakování se zapíná parametrem `loop`. Dalším důležitým senzorem je detektor pohybu (`TouchSensor`). Ten detekuje pouze pohyb nad objektem (`isOver`) nebo kliknutí na něj (`isActive`).

Pro animace závislé na čase je třeba převést časový údaj na potřebný parametr objektu. K tomu se používají interpolátory, které tento převod provádí. Jedná se o převodní tabulku, ze které se lineárně interpolují výstupní hodnoty.

K propojení zdrojů událostí (`eventOut`) a příjemců (`eventIn`) slouží parametr `ROUTE`. Ten kopíruje událost z jednoho uzlu do druhého.

Ukázka použití interpolátoru orientace, časovače a detektoru pohybu. Při kliknutí se kyvadlo jednou vykývne. Detektor pohybu předá časovači čas startu, tím se časovač spustí. Časovač průběžně posílá změnu času do interpolátoru orientace. Ten převede čas na rotaci kyvadla.

```
#VRML V2.0 utf8
PROTO kyvadlo
[ exposedFieldSFRotation naklon 0 1 0 0 ]
{ Transform {...} }
Transform {
    children [
        DEF Touch TouchSensor {}
        DEF kyv kyvadlo {naklon 0 0 1 -1}
    ]
}
DEF Time TimeSensor {}
DEF OrInter OrientationInterpolator {
    key [0, .5, 1]
    keyValue [0 0 1 -1,0 0 1 1,0 0 1 -1]
}
ROUTE Touch.touchTime TO Time.startTime
ROUTE Time.fraction_changed TO OrInter.set_fraction
ROUTE OrInter.value_changed TO kyv.naklon
```

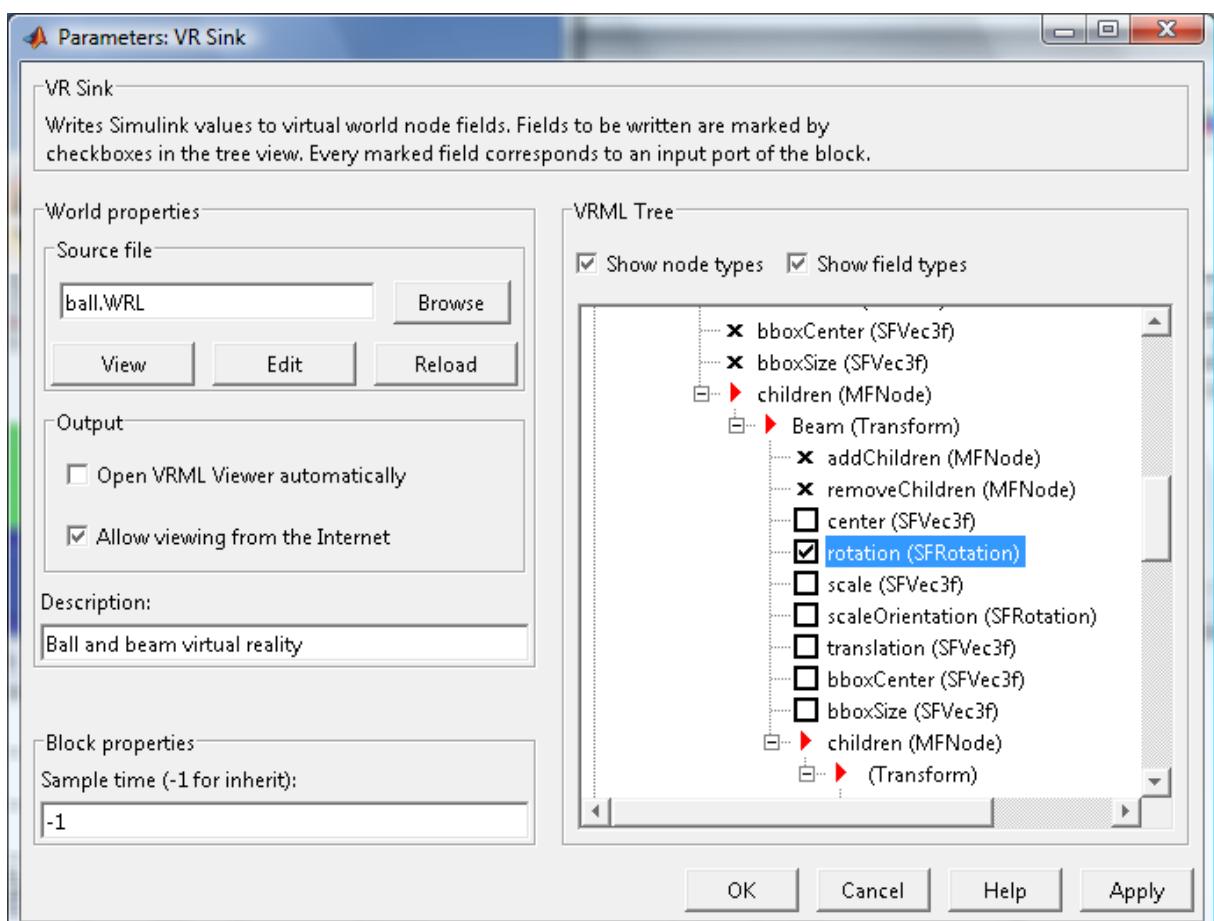
Jako VRML editor lze použít V-realm, který je součástí VR toolboxu[2]. Je to grafický

editor, který nabízí základní tvary, ale i knihovnu s různými předměty. Velkou výhodou tohoto editoru je možnost editovat uzly typu IndexedFaceSet, Extrusion a ElevationGrid graficky. Přejmenování uzelů pro použití ve VR-sink se provede v Node tree view. Program nepodporuje tvorbu všech uzelů (IndexedLineSet).

Dalším editorem je VRMLPad, který je textový se zvýrazněním syntaxe a také s automatickým dokončováním kódu. Tento editor je dobrý pro rychlou tvorbu optimalizované trojrozměrné scény v textovém režimu. Nevýhodou je nemožnost přímého zobrazení výsledné scény.

2.3.Zápis a čtení dat virtuální scény

Virtuální scéna je v Simulinku tvořena blokem VR Sink, který automaticky načítá vybraný VRML soubor s koncovkou *.wrl. V dialogu VR Sink se nejprve vybere soubor se scénou. Po načtení souboru se v levé části zobrazí seznam uzlů scény. Aby se uzel mohl použít jako vstupní, musí být pojmenován pomocí parametru DEF jméno. Pokud je takto pojmenován tak se u parametrů tohoto uzlu objeví zaškrťávací pole. Pokud je pole označeno, tak se parametr zobrazí jako vstup bloku VR Sink.



Obrázek 2.3.1: Dialog bloku VR Sink

Vstup je třeba upravit na správný formát. Pokud se objektem otáčí, je třeba poslat na vstup celý vektor rotace.

Speciálním blokem určeným na zápis textové informace do scény je blok VR Text Output. Zde se nastaví uzel, do kterého se text zapíše. Lze jej použít například na zobrazení hodnoty výstupu na modelu pomocí textového čísla.

Čtení dat z virtuálního světa není jednoduché, protože VR Toolbox nemá přímou podporu čtení ze senzorů ani jiných uzelů. To je nutné zajistit pomocí s-funkce v simulinku.

S-funkce je blok v Simulinku, který je aktualizován a přepočítán v každém výpočetním průchodu modelu. S-funkce se programují ve více jazycích, jako např. C nebo fortran. V této práci je použit Matlabský m-kód, pro který jsou dvě možnosti programování (Level-1 a Level-2). Level-1 je starší kód, který má složitější hlavičku funkce a řídí se proměnnou flag a novější Level-2, který používá předdefinované procedury.

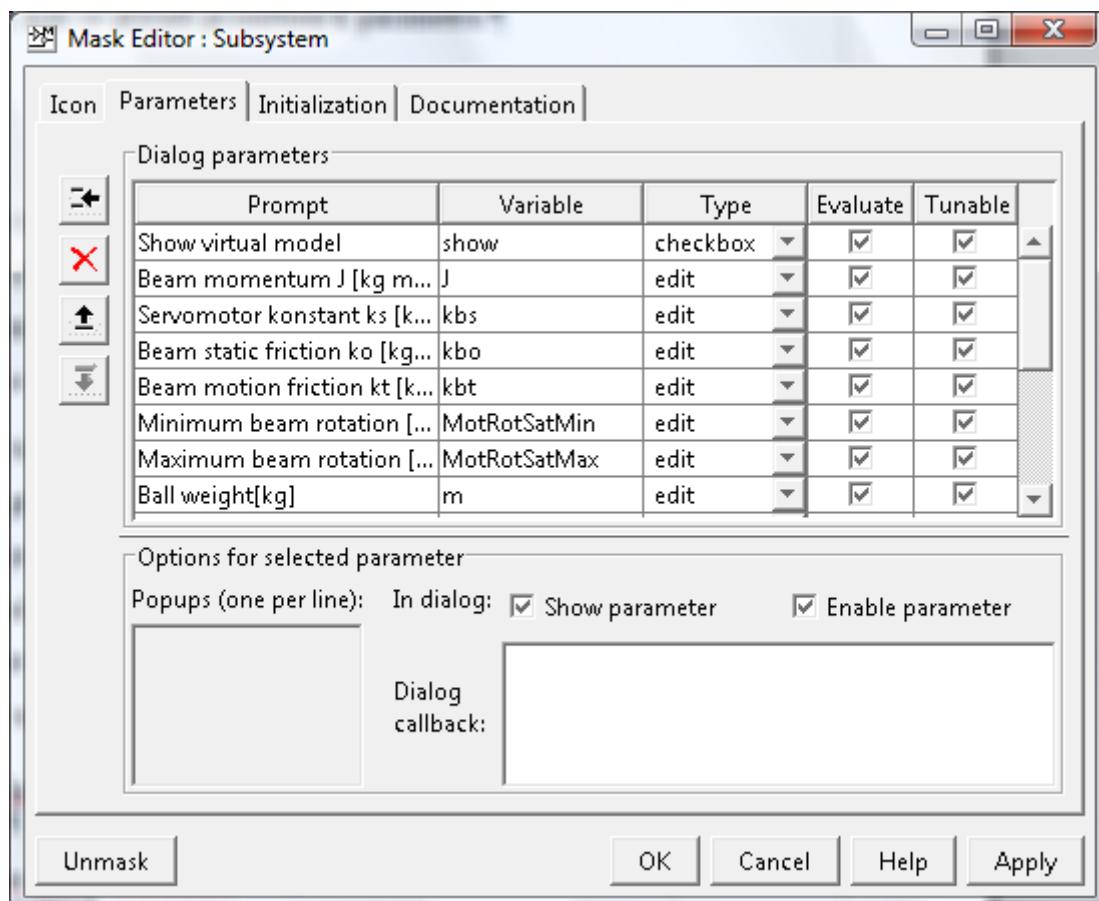
Pro čtení ze senzorů scény jsme použili Level-1 s-funkci. Vstupním parametrem je ukazatel na uzel, který má být přečten. Tento ukazatel získáme pomocí příkazu `getfield(vrsink, 'TouchSen')`, kde TouchSen je jméno detektoru pohybu umístěného ve scéně VR Sink.

Nejprve je třeba synchronizovat události `isActive` a `hitPoint_change` pomocí příkazu `sync`. Událost `isActive` detekuje kliknutí myši a `hitPoint_change` vrátí okamžitou polohu myši. Samotná událost `hitPoint_change` aktualizuje polohu myši nad objektem průběžně. Za podmínky `isActive` se poloha použije jako výstup modelu, který vrací požadovanou polohu pro řízení.

Pokud je simulace spuštěna na rychlejším počítači, tak se může stát, že proběhne příliš rychle a uživatel simulaci neuvidí. Proto je přidán blok „Real-time Synchronization“, který zpomaluje rychlosť tak, aby rychlosť simulace odpovídala reálnému modelu.

2.4.Simulink a maska

Zde si popíšeme jak se vytvoří maska simulinku a jak se z této masky ovládá 3D scéna. Při otevření bloku simulinku se nám zobrazí vnitřní zapojení bloku. Pokud chceme tomuto zabránit a pouze nastavovat vybrané parametry, tak se použije právě maska. Výběr parametrů se provádí v editoru masky, kde se přiřadí proměnná k parametru.



Obrázek 2.4.1: Editor masky

Lze si vybrat mezi třemi druhy parametru. Nejdůležitější je editační pole 'edit', zaškrťávací políčko 'checkbox' a výběr z několika možností 'popup'.

Aby bylo možné zobrazovat virtuální scénu, je v masce přidán parametr show. Jedná se o zaškrťávací políčko s možnými hodnotami 'on' a 'off'. Pokud je políčko zaškrtnuto je nastavena hodnota proměnné show na 'on'. Při aplikování změny na masku se za podmínky této proměnné zobrazí nebo zavře virtuální scéna.

Také bylo třeba ošetřit vícenásobné otevírání oken se stejnou scénou a opakování otevírání virtuálního světa pomocí příkazu vrworld. Při každém otevírání virtuálního světa se testuje zda již tento svět existuje nebo zda je ještě nebyl vytvořen. Na uložení uživatelských parametrů v simulinku slouží datová proměnná UserData, která je v každém bloku simulinku.

V našem případě si uložíme ukazatel na otevřenou virtuální scénu a na prohlížeč tohoto světa. Zdrojový kód programu je uložen v inicialization kartě masky:

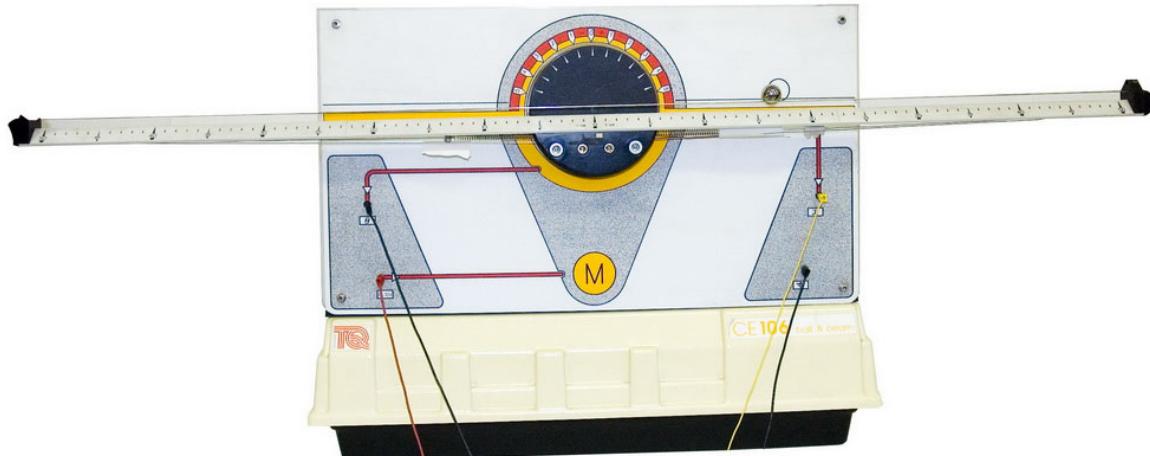
```
if strcmp('off',get_param(gcb,'show'))
    vrclose all;
else
    tmp=get_param(gcb,'UserData');
    if ~isstruct(tmp)
        tmp.vrw=vrworld('watt.wrl');
        tmp.vrf=view(tmp.vrw);
        set_param(gcb,'UserData',tmp);
    else
        if ~isValid(tmp.vrf)
            tmp.vrf=view(tmp.vrw);
            set_param(gcb,'UserData',tmp);
        end
    end
end
```

Tento program se spustí při otevření masky, při zavření a pokud byly parametry uloženy pomocí tlačítka Apply.

3. Kulička na tyči

Kulička na tyči je model složený z tyče, servomotoru a kuličky. Servomotor ovládaným napětím u naklání tyč s kolejničkami z lanka. Kolejničky vedou volně položenou kuličku tak, aby nevypadla a zároveň slouží na měření polohy kuličky. Poloha kuličky a úhel natočení tyče jsou výstupem modelu. Úkolem je zastavit kuličku na daném místě tyče.

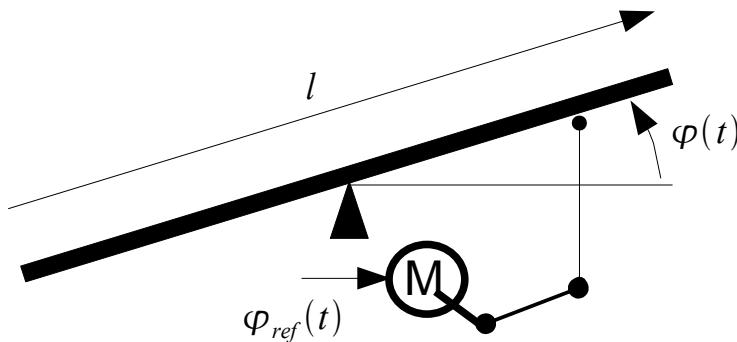
Jako systém je v otevřené smyčce nestabilní a lze na něj aplikovat různé druhy moderních regulátorů [3].



Obrázek 3.1: Laboratorní model kuličky na tyči

3.1. Model tyče

V této kapitole vytvoříme fyzikální model tyče a jeho přenos. Vstupem do modelu je vstupní napětí $u(t)[V]$ a výstupem je úhel natočení tyče $\varphi(t)[rad]$. Tyč se skládá z nosníku a kolejničky ze dvou lanek, které slouží také na měření polohy kuličky $x[m]$. Nosník je uprostřed připevněn na servomotor, který je řízen vstupním napětím $u(t)$ odpovídajícím referenčnímu úhlu natočení $\varphi_{ref}(t)[rad]$. Největší hmotnost tyče je soustředěna v nosníku.



Obrázek 3.1.1: Model tyče

Matematický popis modelu tyče nakláněné pomocí servomotoru vychází z řešení úlohy rotačního pohybu, kde hledáme celkový moment setrvačnosti. Z něho pomocí druhého Newtonova zákona získáme úhel natočení tyče $\varphi(t)$.

Servomotor dodává soustavě moment síly $M_s(t)[Nm]$ v závislosti na vstupním napětí $u(t)[V]$ a na konstantě servomotoru $k_s[Nm V^{-1}]$, která také zahrnuje převod z jednotek měřených matlabem na vstupní napětí

$$M_s(t) = k_s u(t)$$

Uvažujme také moment tlumení $M_t(t)[Nm]$ ve kterém je zahrnuta konstanta tření při pohybu $k_t[Nm]$ a odporu prostředí $k_o[N\,m\,s]$

$$M_t(t) = k_t \dot{\varphi}(t) + k_o$$

Celkový moment síly je roven součtu všech momentů působících na tyč. V našem případě je celkový moment $M_c(t)[Nm]$ roven momentu servomotoru M_s od kterého odečteme moment tlumení M_t .

Pro převod na úhel natočení $\varphi(t)[rad]$ použijeme druhý Newtonův zákon, kde $\epsilon(t)[rad\cdot s^{-2}]$ je úhlové zrychlení a $J[kg\cdot m^2]$ moment setrvačnosti

$$M_c(t) = \sum (M_i(t)) = M_s(t) - M_t(t) = J \epsilon(t) = J \ddot{\varphi}(t) \quad (3.1.1)$$

Pro zjednodušení lze moment setrvačnosti J spočítat pomocí vzorce pro moment setrvačnosti tenké tyče o hmotnosti $m[kg]$ a délky $l[m]$ vzhledem k ose procházející středem tyče kolmo k její délce

$$J = \frac{1}{12} m l^2 \quad (3.1.2)$$

Po dosazení (3.1.2) do (3.1.1) dostaneme diferenciální rovnici závislosti úhlu $\varphi(t)$ na vstupním napětí $u(t)$

$$J \cdot \dot{\varphi}(t) + k_t \varphi(t) + k_o = k_s u(t)$$

Jde o lineární diferenciální rovnici, ze které provedení Laplaceovy transformace vznikne přenos vstupního napětí $u(t)$ na úhel natočení tyče $\varphi(t)$

$$G(s) = \frac{U(s)}{\Phi(s)} = \frac{k_s}{J s^2 + k_t s + k_o} \quad (3.1.3)$$

a po dosazení do J z (3.1.2)

$$G(s) = \frac{k_s}{\frac{1}{12} m l^2 s^2 + k_t s + k_o}$$

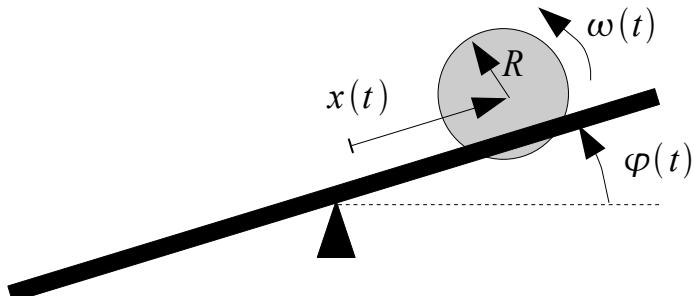
Tento přenos je vhodný k výpočtům fyzikálních parametrů modelu. Pro simulaci modelu v simulinku je třeba přenos $G(s)$ převést na kanonický tvar vzhledem k výstupu, který lze jednoduše navrhnout. Přenos (3.1.3) převedeme na tvar

$$G(s) = \frac{\frac{k_s}{J}}{s^2 + \frac{k_t s}{J} + \frac{k_o}{J}} \quad (3.1.4)$$

a použijeme ho na vytvoření simulinkového schématu v kapitole kompletního modelu.

3.2. Model kuličky

V této kapitole vytvoříme fyzikální model kuličky. Kulička se pohybuje po nakloněné tyči v kolejničce, která také slouží jako odporový potenciometr pro měření polohy $x[m]$.



Obrázek 3.2.1: Schématický nákres kuličky na tyči

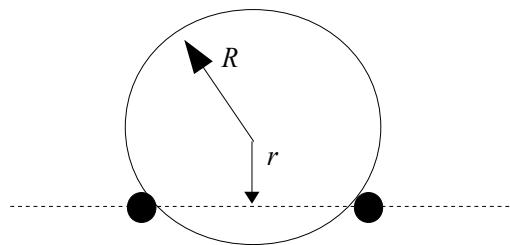
Matematický model kuličky vychází z pohybu kuličky po nakloněné rovině reprezentované tyčí. Kulička koná současně posuvný pohyb a otáčivý pohyb kolem své osy. Polohu kuličky spočítáme z rovnováhy potenciální a kinetické energie pomocí Lagrangeovy rovnice druhého druhu [4].

Nejprve nalezneme kinetickou energii $W_k[J]$ složeného pohybu. Ta je dána součtem energie posuvného a otáčivého pohybu. Posuvný pohyb je charakterizován hmotností $m[g]$ a rychlostí $v[m s^{-1}]$. Otáčivý pohyb je dán momentem setrvačnosti $J[kg \cdot m^2]$ a úhlovou rychlosťí $\omega[rad s^{-1}]$. Výsledná kinetická energie je

$$W_k = \frac{1}{2}mv^2 + \frac{1}{2}J\omega^2 ,$$

kde moment setrvačnosti koule J závisí na poloměru koule $R[m]$

$$J = \frac{2}{5}mR^2$$



Obrázek 3.2.2: Kulička v kolejničce

Neznámou úhlovou rychlosť $\omega(t)$ pripomíme z obvodové rychlosťi kuličky $v(t)[ms^{-1}]$ pomocí jejího obvodu otáčení $r[m]$ a to $\omega(t)=v(t)/r$. Obvodovou rychlosť $v(t)$ získáme ze vzdálosťi kuličky od stredu tyče $x(t)[m]$ pomocí její derivace podle času $v(t)=\dot{x}(t)$.

Dále odvodíme vzťah pro potenciálni energii kuličky $W_p(t)$ závislou na náklonu tyče $\varphi(t)[rad]$, vzdálosťi kuličky od stredu tyče $x(t)[m]$, gravitačným zrychlením $g[ms^{-2}]$ a hmotnosťi kuličky $m[g]$

$$W_p(t)=mgx(t)\sin(\varphi(t)).$$

Pak pro výpočet polohy kuličky použijeme Lagrangeovy rovnice druhého druhu, které umožňují vytvoření pohybových rovnic. Lagrangeova rovnice má tvar

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right)-\frac{\partial L}{\partial x}=Q$$

kde Lagrangeova funkcia L , označovaná také jako kinetický potenciál, je v našem případě definována jako rozdíl kinetické a potenciální energie soustavy

$$L=W_k-W_p=\frac{1}{2}m\dot{x}^2(t)+\frac{1}{5}\frac{mR^2\ddot{x}^2(t)}{r^2}-mgx(t)\sin(\varphi)$$

a tlumící (třecí) síly závislé na rychlosti $k_t[gms^{-1}]$ a nezávislé na rychlosti $k_o[gm^2s^{-2}]$ jsou zahrnutý v rovnici jako zobecněné síly Q

$$Q=-k_t\dot{x}(t)-k_o$$

Po dosazení do Lagrangeovy rovnice a jejím výpočtu nám vyjde závislost polohy kuličky $x(t)$ na úhlu natočení $\varphi(t)$

$$m\ddot{x}^2(t)+\frac{2}{5}\frac{mR^2\ddot{x}^2(t)}{r^2}-mg\sin(\varphi(t))=-k_t\dot{x}(t)-k_o$$

Tato diferenciální rovnice není lineární a bylo by nutné provést pro další úpravy linearizaci. Ale její nelineárni část je funkcia $\sin(\varphi(t))$, kde $\varphi(t)$ je jediným vstupem systému. Proto lze pro další výpočet nahradit funkci $\sin(\varphi(t))$ proměnou $\alpha(t)=k_v\sin(\varphi(t))$, kde konstanta $k_v[-]$ převádí výstup systému tyče na náklon v radiánech. Funkci $\sin(\varphi(t))$ použijeme na vstupu systému při tvorbě simulinkového schéma.

$$\left(m+\frac{2}{5}\frac{mR^2}{r^2}\right)\ddot{x}^2(t)+k_t\dot{x}+k_o=mg\alpha(t)$$

Tato rovnice už je lineární a lze ji převést pomocí Laplaceovy transformace na přenos a upravit na kanonický tvar

$$G(s) = \frac{mg}{\left(m + \frac{2}{5} \frac{mR^2}{r^2}\right)s^2 + k_t s + k_o} = \frac{\frac{g}{\left(1 + \frac{2}{5} \frac{R^2}{r^2}\right)}}{s^2 + \frac{k_t s}{\left(m + \frac{2}{5} \frac{mR^2}{r^2}\right)} + \frac{k_o}{\left(m + \frac{2}{5} \frac{mR^2}{r^2}\right)}} , \quad (3.2.5)$$

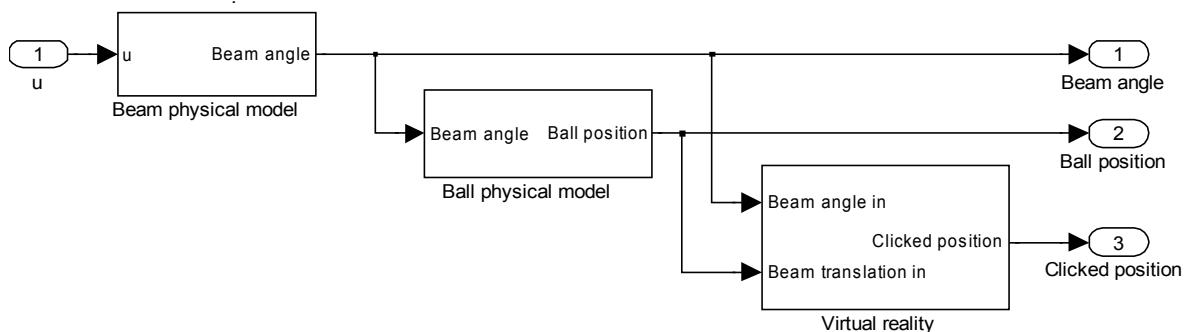
který použijeme na vytvoření simulinkového schématu v kompletním modelu kuličky na tyči.

Aby byly do simulace zahrnuty i pružné zakončení tyče, je zde zaveden součinitel restituce k_b [], který zohledňuje odraz kuličky od okraje tyče. Je vyjádřen jako poměr rychlostí kuličky před srážkou $v(t)[\frac{m}{s}]$ a po srážce $v'(t)[\frac{m}{s}]$

$$v'(t) = k_b v(t) \quad (3.2.6)$$

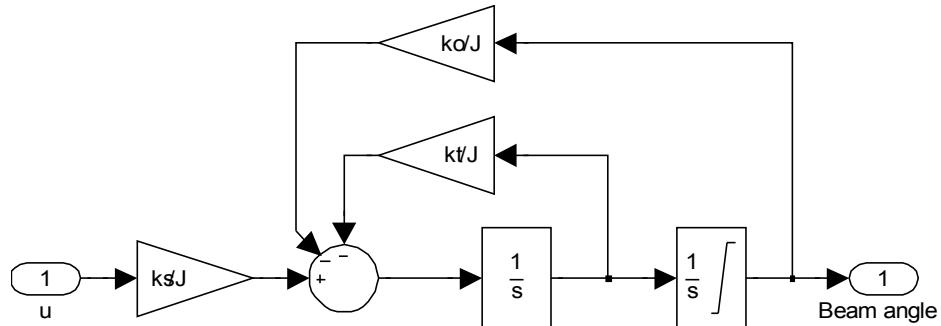
3.3. Kompletní model

V této kapitole složíme model kuličky na tyči v jediný blok simulinku. Jeho vstupem je řídící napětí a výstupem je úhel tyče, poloha kuličky na tyči a požadovaná poloha kuličky. Celý simulinkový model pojmenovaný „Ball and beam“ (obrázek 3.3.1) je složen ze subsystému virtuální reality „Virtual reality“, dynamiky fyzikálního modelu tyče „Beam dynamic“ a kuličky „Ball dynamic“.



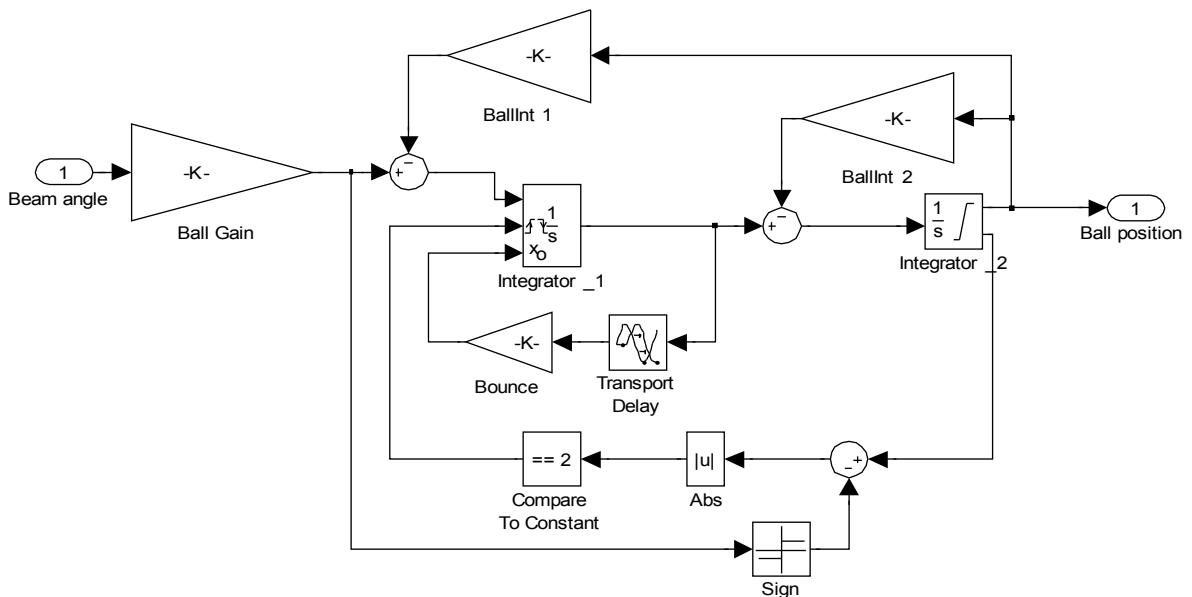
Obrázek 3.3.1: Vnitřní simulinkové schéma bloku „Ball and beam“

Simulinkové schéma bloku systému tyče (obrázek 3.3.2) vytvoříme z přenosu (3.1.4), kde vstupem je napětí na servomotoru tyče a výstupem je náklon tyče. Integrátor na výstupu je limitován maximálním úhlem natočení, který je nastavitelný jako parametr modelu.



Obrázek 3.3.2: Simulinkové schéma systému tyče

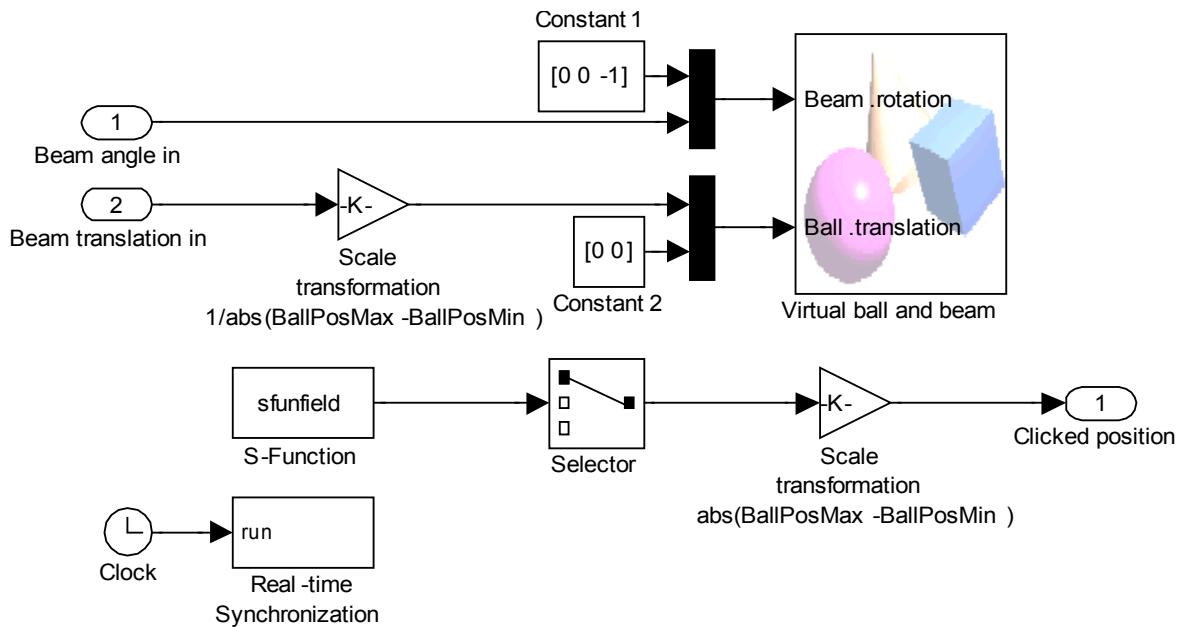
Simulinkový model kuličky (obrázek 3.3.4) vznikne z přenosu (3.2.5) a simulací odrazu (3.2.6). Odraz je proveden pomocí výstupu saturace integrátoru, který dá impulz k převedení rychlosti nárazu na rychlosť odrazu s opačným znaménkem.



Obrázek 3.3.4: Simulinkové schéma systému kuličky

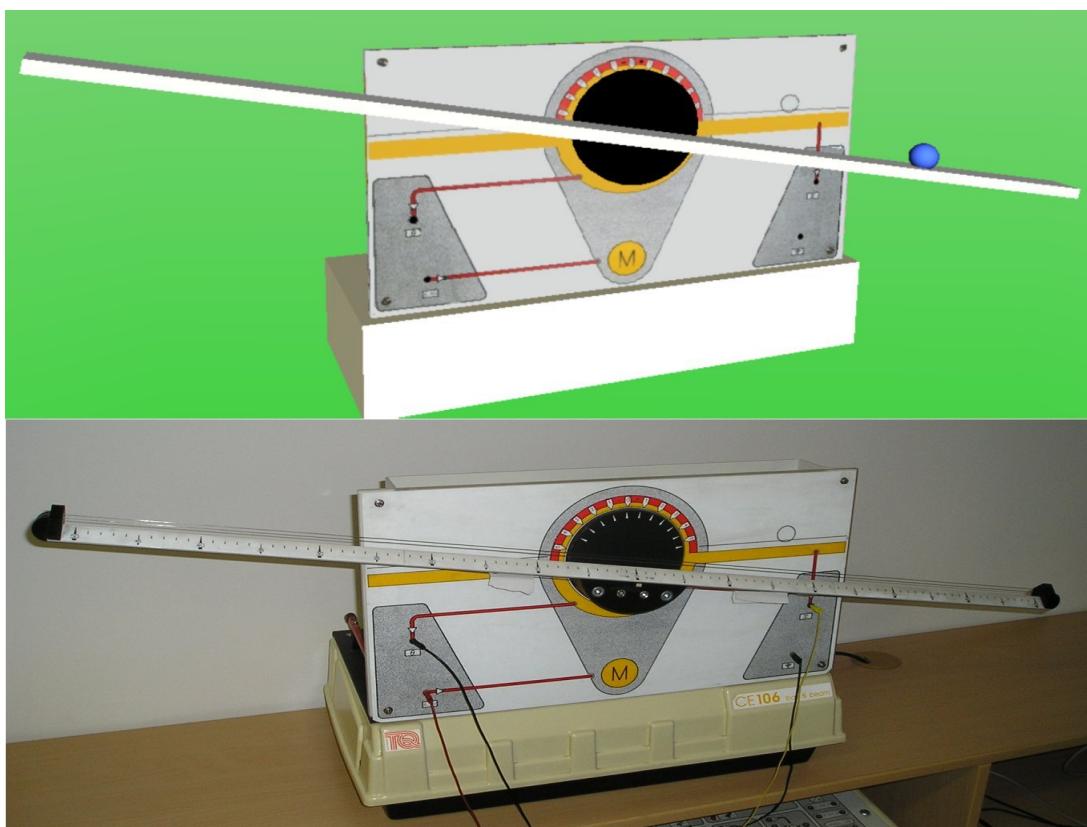
Blok virtuální reality (obrázek 3.3.5) má tři části. První je vlastní virtuální zobrazení, druhá část je detekce kliknutí na požadovanou polohu a poslední je synchronizace s reálným časem. Poloha kuličky se upravuje tak, aby byla v maximech polohy na krajích tyče. Zpětný přepočet se provádí s požadovanou polohou kuličky z výstupu od uživatele. Další úpravy spočívají v převodu skalární hodnotu úhlu a pozice na požadovaný vektorový tvar jazyka VRML. Pohyb kuličky je řízen uzlem `ball.translation` a natočení tyče pomocí `beam.rotation`. Uzel tyče je nadřazen uzlu kuličky a tím je pohyb kuličky vázán na posuvný pohyb po

tyči. Ta se otáčí spolu se souřadnou soustavou kuličky.



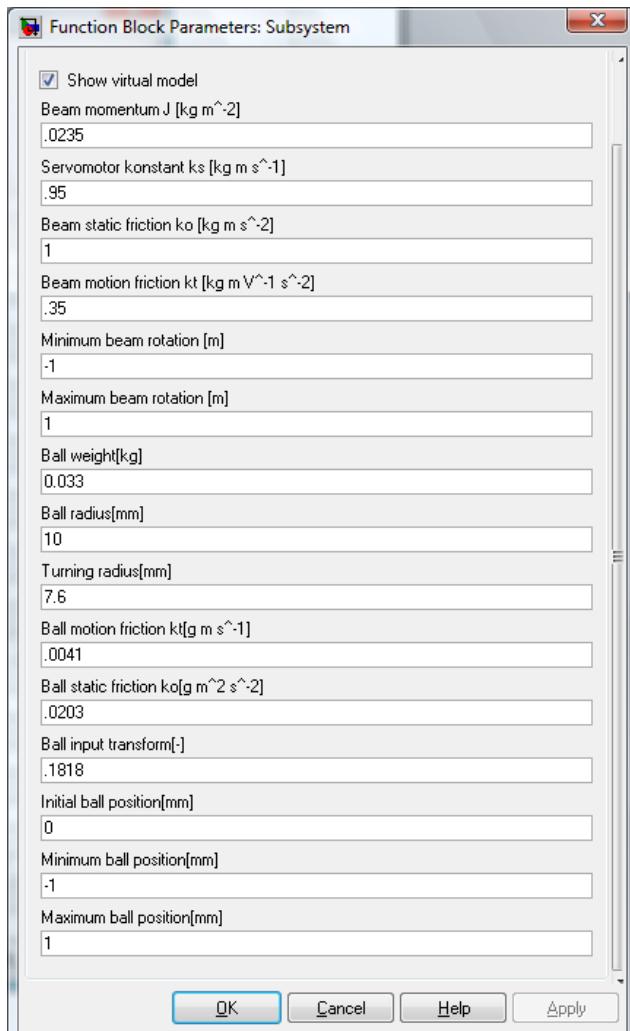
Obrázek 3.3.5: Simulinkové schéma virtuální reality

VRML model je vytvořen podle předlohy reálného modelu typu TQ CE106. Aby grafické zobrazení nebylo příliš zatěžující pro počítač, tak jsou použity pouze základní geometrické tvary. Přední část je tvořena z desky(box) potažené texturou, která byla ofocena z reálného modelu.



Obrázek 3.3.6: Virtuální a reálný model kuličky na tyči

Nastavení parametrů modelu pomocí dialogu masky Simulinku. Parametr show virtual model zobrazí 3D scénu modelu kuličky na tyči.



Obrázek 3.3.7: Nastavení parametrů modelu pomocí masky

3.4.Identifikace reálného modelu

V této kapitole identifikujeme reálný model a porovnáme ho s virtuálním. Odhadneme přenosy kuličky a tyče z reakce na jednotkový skok. Pro odhad systému tyče připojíme na vstup modelu jednotkové napětí. Pokud necháme tyč nakloněnou na jednotkový úhel natočení, položíme kuličku na tyč a odhadneme její přenos. Výsledné přenosy tyče

$$G_t(s) = \frac{0.95}{0.0235 \cdot s^2 + 0.35 \cdot s + 1} \text{ a kuličky } G_k(s) = \frac{1.45}{s^2 + 0.1 \cdot s + 0.5} .$$

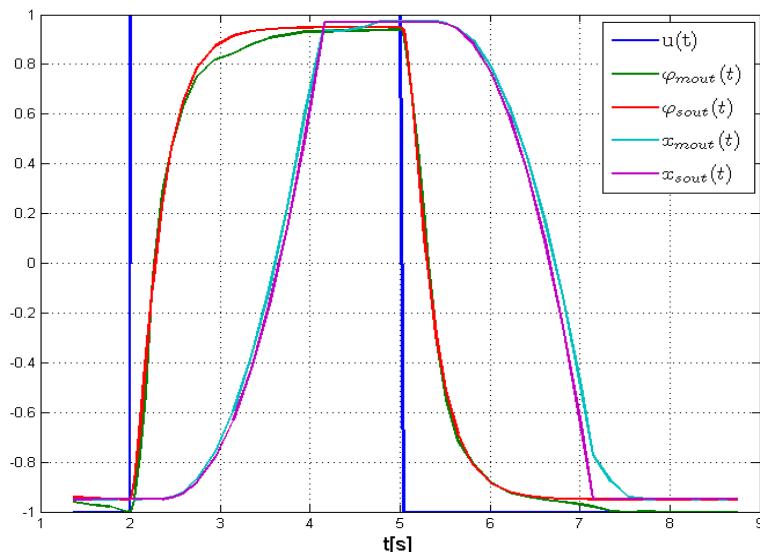
Z důvodu pevného uchycení tyče nelze jednoduše zjistit její hmotnost přímo. Proto z přenosu tyče (3.1.3) a z odhadu přenosu $G_t(s)$ vychází, že moment setrvačnosti

$$J = \frac{1}{12} m \cdot l^2 = 0.0235 \text{ Kg} \cdot \text{m}^2 .$$

Délka tyče je $l=1\text{m}$ a z toho vyjde její hmotnost $m=0.282\text{kg}$. Konstanty k_t , k_o a k_s také odhadneme z přenosu na $k_t=0.35[\text{Nm}]$, $k_o=1[\text{Nm s}]$ a $k_s=0.95[\text{Nm V}^{-1}]$.

Změřit parametry kuličky je jednodušší. Z poloměru kuličky $R=13\text{ mm}$, objemu koule $V=\frac{4}{3}\pi \cdot R^3=4.2\text{ cm}^3$ a hustoty železa $\rho=7.86\frac{\text{g}}{\text{cm}^3}$, vyjde hmotnost $m=V \cdot \rho=33\text{g}$.

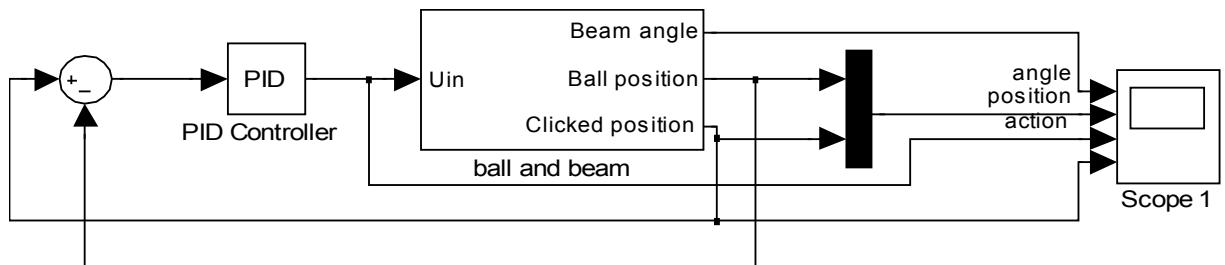
Poloměr otáčení je $r=10\text{mm}$ a tříhové zrychlení $g=9.81\frac{\text{m}}{\text{s}^2}$. Konstanty k_t modelu opět odečteme z přenosu $k_t=0.0011[\text{gm s}^{-1}]$, $k_o=0.0203[\text{gm}^2\text{s}^{-2}]$ a $k_v=0.1818[1]$. Tyto parametry dosadíme do modelu pomocí dialogu masky. Průběh natočení tyče reálného modelu $\varphi_{mout}(t)$ a virtuálního $\varphi_{sout}(t)$ a poloha kuličky reálného $x_{mout}(t)$ a virtuálního $x_{sout}(t)$ modelu v závislosti na vstupním napětí $u(t)$.



Obrázek 3.4.1: Porovnání simulace a modelu

3.5.PID regulace kuličky na tyči

V této kapitole navrheme regulaci polohy kuličky na tyči pomocí změny náklonu tyče. Jako regulátor použijeme PID (proporcionálně integračně derivační) regulaci, která je pro tento model dostatečná. Pro návrh použijeme parametry z minulé kapitoly. Jako referenční (cílová) hodnota je přiveden výstup modelu „Clicked position“, který se mění podle požadované polohy ve virtuální scéně.



Obrázek 3.5.1: Zapojení modelu kuličky na tyči s PID regulátorem

Navrheme PID regulátor metodou geometrického místa kořenů (GMK) (Obrázek 3.5.3).

Pro přenos otevřené smyčky platí

$$KG(s) = K \frac{b(s)}{a(s)} = S(s) R(s) ,$$

kde K je zesílení, $S(s)$ přenos soustavy, $a(s)b(s)$ polynomy a $R(s)$ přenos regulátoru

$$C(s) = kP + \frac{kI}{s} + kD s = \frac{kD(s+!I)}{(s+!D)}$$

Přenos řízení je dán rovnicí

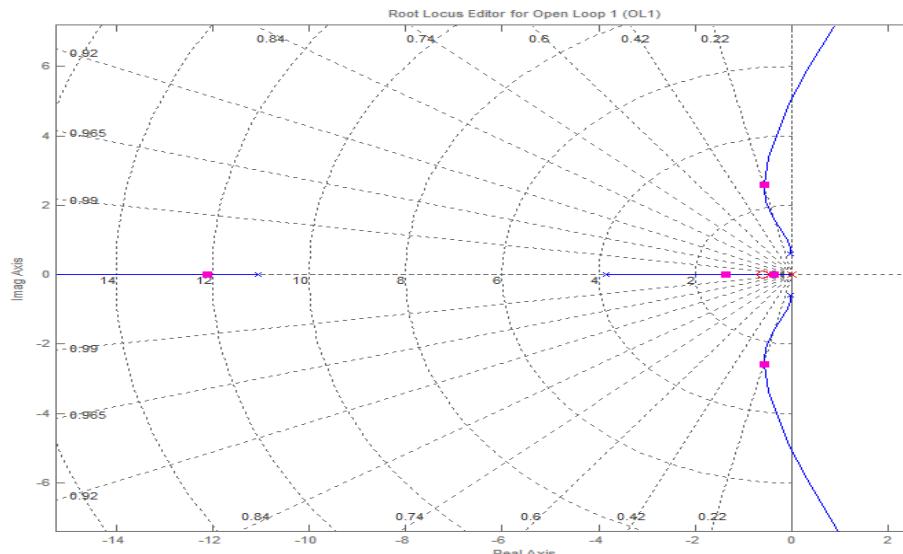
$$Fw(s) = \frac{KG(s)}{1+KG(s)} = \frac{Kb(s)}{a(s)+b(s)}$$

Metoda GMK hledá kořeny charakteristické rovnice

$$1 + KG(s) = a(s) + b(s) = 0$$

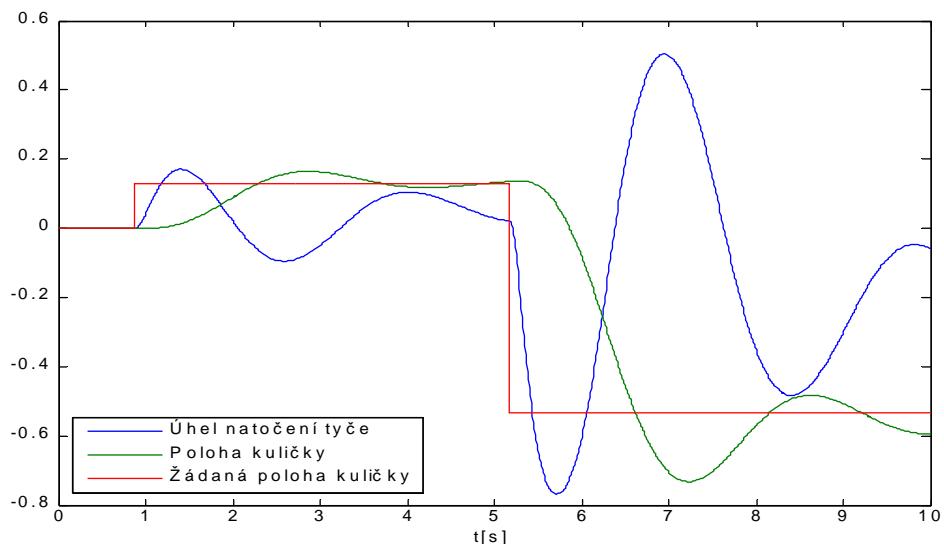
V Matlabu se GMK řeší pomocí nástroje `rlttool`. Pro identifikovaný reálný model z minulé kapitoly jsou složky PID regulátoru:

derivační $r_1 = 2.9$, proporcionální $r_0 = 3.4$ a integrační $r_{-1} = 1$.



Obrázek 3.5.3: Návrh PID pomocí GMK

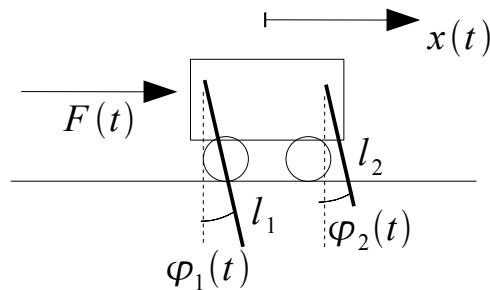
Na vstup referenční hodnoty regulátoru přivedeme žádanou hodnotu výstupu a spustíme simulaci (Obrázek 3.5.2). Kulička se ustálí na požadované poloze na tyči.



Obrázek 3.5.2: Odezva na vstupní signál

4. Vozík se dvěma kyvadly

Dalším systémem, pro který zde vytvoříme virtuální model, je vozík se dvěma kyvadly (obrázek 4.1). Nejdříve si odvodíme matematický model vozíku s jedním kyvadlem a přidání dalšího kyvadla provedeme až v celkovém modelu. Systém se skládá z vozíku o hmotnosti $m_v [kg]$ a kyvadla o hmotnosti $m_1 [kg]$ a délce $l_1 [m]$. Kyvadlo je zavěšeno na vozíku pomocí volného čepu a vozík se může pohybovat pouze ve směru osy x . Vstupem tohoto systému je síla $F(t) [N]$ působící na vozík ve směru osy x a výstupy tohoto systému jsou poloha vozíku $x(t) [m]$ a úhel náklonu kyvadla $\varphi_1 [rad]$.



Obrázek 4.1: Vozík se dvěma kyvadly

Nyní se pokusíme odvodit matematicko-fyzikální model takovéhoto systému pomocí Lagrangeových rovnic druhého druhu [3]. Určíme tedy nejprve potenciální a kinetické energie jednotlivých částí systému a pak sestavíme Lagrangeovu funkci celého systému.

Nejprve začneme se samotným vozíkem. Kinetickou energii vozíku můžeme psát jako

$$W_{kv}(t) = \frac{1}{2} m \dot{x}^2(t),$$

kde $\dot{x}(t) [ms^{-1}]$ je rychlosť vozíku a m je celková hmotnost vozíku a všech kyvadel. Protože se při pohybu vozíku nemění jeho výška, je jeho potenciální energie konstantní a my ji nebudeme dále potřebovat.

Dále budeme hledat potenciální a kinetickou energii kyvadla s pohyblivým závěsem. Potenciální energie takového kyvadla je

$$W_p(t) = W_{p0} + \frac{1}{2} m_1 g l_1^2 \sin(\varphi_1(t)), \quad (4.7)$$

kde $g [ms^{-2}]$ je gravitační konstanta a W_{p0} je potenciální energie v nulovém bodě.

Moment setrvačnosti kyvadla, které se otáčí kolem jednoho jeho konce je $J = \frac{1}{3} m_1 \cdot l^2$ (4.8)

Kinetická energie kyvadla se skládá ze dvou částí: z rotace kyvadla kolem pevného bodu a z posuvu tohoto bodu díky pohybu vozíku. Pak můžeme psát

$$W_{kl}(t) = \frac{1}{2} m_1 v_1^2(t) + \frac{1}{2} J_{Tl} \dot{\varphi}_1^2(t),$$

kde posuvná část kinetické energie závislá na vektoru rychlosti $v_1 [ms^{-1}]$ a rychlosti otáčení kyvadla $\dot{\varphi}_1(t) [rad s^{-1}]$.

Pro posuvný pohyb je třeba rozložit rychlosť na svislý a vodorovný směr. Výsledný vektor rychlosti je vektorovým součtem těchto rychlostí

$$v_1^2(t) = v_{1x}^2(t) + v_{1y}^2(t).$$

Rychlosť ve směru pohybu vozíku je dána součtem rychlosti vozíku a kyvadla ve vodorovném směru

$$v_{1x}(t) = \dot{x}(t) + \frac{1}{2} l \dot{\varphi}_1(t) \cos(\varphi_1(t)),$$

rychlosť kolmá k pohybu vozíku je rychlosť kyvadla ve svislém směru

$$v_{1y}(t) = \frac{1}{2} l \dot{\varphi}_1(t) \sin(\varphi_1(t)).$$

Celková rychlosť je vektorový součet předchozích rychlostí

$$\begin{aligned} v_1^2(t) &= \dot{x}^2(t) + \dot{x}(t) \dot{\varphi}_1(t) l_1 \cos(\varphi_1(t)) + \frac{1}{4} l_1^2 \dot{\varphi}_1^2(t) \cos^2(\varphi_1(t)) + \frac{1}{4} l_1^2 \dot{\varphi}_1^2(t) \sin^2(\varphi_1(t)) \\ &= \dot{x}^2(t) + \dot{x}(t) \dot{\varphi}_1(t) l_1 \cos(\varphi_1(t)) + \frac{1}{4} l_1^2 \dot{\varphi}_1^2(t) \end{aligned}$$

a celková kinetická energie kyvadla je

$$\begin{aligned} W_{kl}(t) &= \frac{1}{2} m_1 \left(\dot{x}^2(t) - \dot{x}(t) \dot{\varphi}_1(t) l_1 \sin(\varphi_1(t)) + \frac{1}{4} l_1^2 \dot{\varphi}_1^2(t) \right) + \frac{1}{2} \frac{1}{12} l_1^2 m_1 \dot{\varphi}_1^2(t) \\ &= \frac{1}{2} m_1 \left(\dot{x}^2(t) - \dot{x}(t) \dot{\varphi}_1(t) l_1 \sin(\varphi_1(t)) + \frac{1}{3} l_1^2 \dot{\varphi}_1^2(t) \right) \end{aligned}$$

Pro výpočet energie složeného pohybu použijeme Lagrangeovy rovnice druhého druhu, kde Lagrangeova funkce je součet energií

$$L(t) = W_{kv}(t) - W_{p0} + W_{kl}(t) - W_{pl}(t).$$

Po dosazení energií získáme Lagrangeovu rovnici kyvadla na vozíku, kterou dále použijeme v odvození modelu vozíku a kyvadla

$$L = \frac{1}{2} m_v \dot{x}^2(t) + \frac{1}{2} m_1 \left(\dot{x}^2(t) - \dot{x}(t) \dot{\varphi}_1(t) l_1 \sin(\varphi_1(t)) + \frac{1}{3} l_1^2 \dot{\varphi}_1^2(t) \right) - \frac{1}{2} l m_1 g \sin(\varphi_1(t)) \quad (4.9)$$

4.1. Model vozíku

Zde si vypočítáme matematický model vozíku. Použijeme Lagrangeovu funkci (4.9). z předchozí kapitoly. Přidáme nekonzervativní sily jako jsou tlumení $\delta_v [m s^2]$ a řídící síla působící na vozík $F(t) [N]$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}} \right) - \frac{\partial L}{\partial x} = -\delta_v J_v \dot{x}(t) + F(t) \quad (4.1.10)$$

Vypočteme parciální derivace Lagrangeovy funkce

$$\frac{\partial L}{\partial \dot{x}} = m_v \dot{x}(t) + \frac{1}{2} m_1 (2 \dot{x}(t) - \dot{\varphi}(t) l_1 \sin(\varphi_1(t))) \quad \frac{\partial L}{\partial x} = 0$$

a po dosazení do (4.1.10) dostaneme pohybovou rovnici vozíku se vstupní silou $F(t) [N]$

$$\ddot{x}(t) + \frac{\delta_v \dot{x}(t)}{(m_v + m_1)} + \frac{1}{2} \frac{m_1 l}{(m_v + m_1)} (\ddot{\varphi}(t) \cdot (\varphi) + \ddot{\varphi}^2(t) \cos(\varphi)) = \frac{F(t)}{(m_v + m_1)} .$$

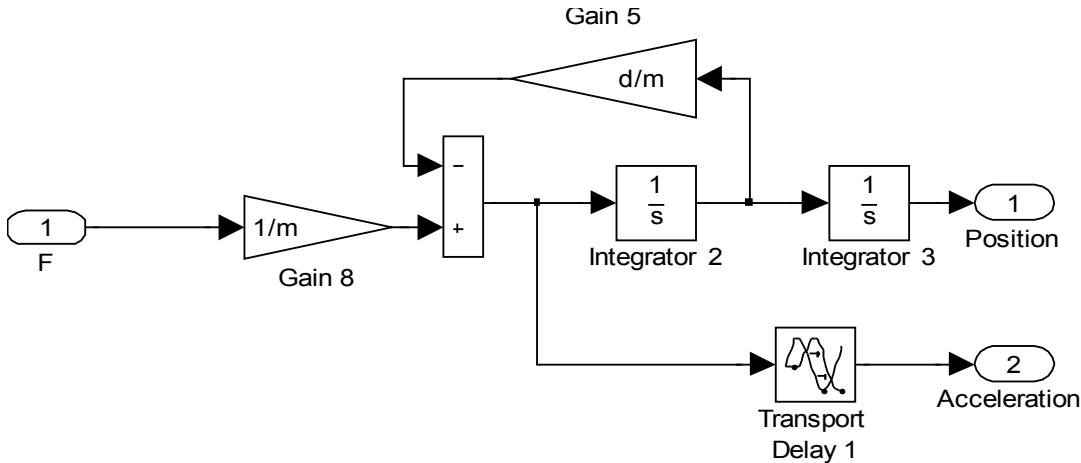
Člen $\frac{1}{2} m_1 l (\ddot{\varphi}(t) \sin(\varphi) + \ddot{\varphi}^2(t) \cos(\varphi))$ vyjadřuje silové působení kyvadla na vozík

$$F_k(t)$$

Tato síla je závislá pouze na úhlu natočení kyvadla a proto ho přesuneme do modelu kyvadla. Pro model vozíku není nutné rozlišovat hmotnost kyvadla a vozíku. Stačí pouze uvést celkovou hmotnost m vozíku a kyvadel. Po dosazení získáme pohybovou rovnici pro vozík

$$\ddot{x}(t) + \frac{\delta_v \dot{x}(t)}{m} = \frac{F(t) + F_k(t)}{m} . \quad (4.1.11)$$

Dále vytvoříme simulinkové schéma se vstupní silou $F(t)$. Na výstupu je pozice vozíku $x(t)$ a jeho akcelerace $\ddot{x}(t)$ ve směru osy x .



Obrázek 4.1.1: Schéma vozíku (blok cart)

Dopravní zpoždění je přidané z důvodu detekce algebraické smyčky při výpočtu v Simulinku. Tato smyčka vzniká přepočtem vstupní síly na zrychlení, ze kterého se spočte silové působení kyvadla na vozík. Po přičtení této síly ke vstupní vznikne výpočetní smyčka. Dopravní zpoždění tuto smyčku eliminuje

4.2. Model kyvadla

Odvodíme si pohybovou rovnici jednoho kyvadla. Do Lagrangeovy rovnice dosadíme (4.9) a přidáme nekonzervativní tlumení způsobené třením $J_t [kg \cdot m^2]$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\phi}} \right) - \frac{\partial L}{\partial \varphi} = -2 \delta J_t \dot{\phi}(t).$$

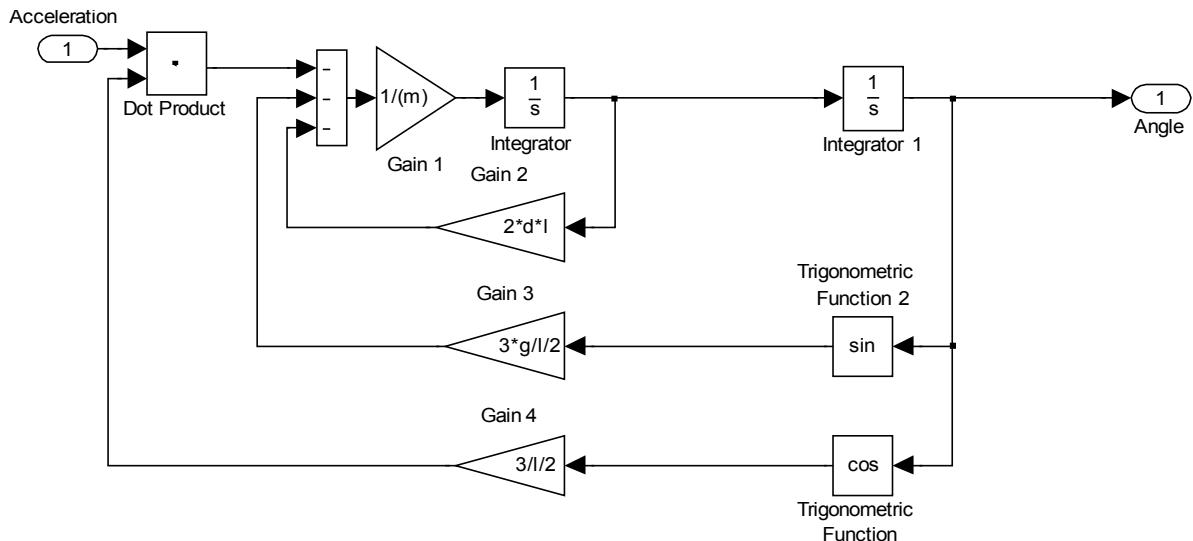
Vypočteme Lagrangeovu rovnici pro kyvadlo

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\phi}} \right) - \frac{\partial L}{\partial \varphi} = \frac{2}{3} l_1 \ddot{\phi}_1(t) - \ddot{x}(t) \sin(\varphi_1(t)) + g \cos(\varphi_1(t)) = -\frac{4}{3} \delta_1 l_1 \dot{\phi}(t).$$

a z toho vyjde pohybová rovnice kyvadla

$$\ddot{\phi}_1(t) + 2 \delta_1 \dot{\phi}(t) + \frac{3}{2} \frac{g}{l_1} \cos(\varphi_1(t)) = \frac{3}{2} \frac{1}{l_1} \ddot{x}(t) \sin(\varphi_1(t)).$$

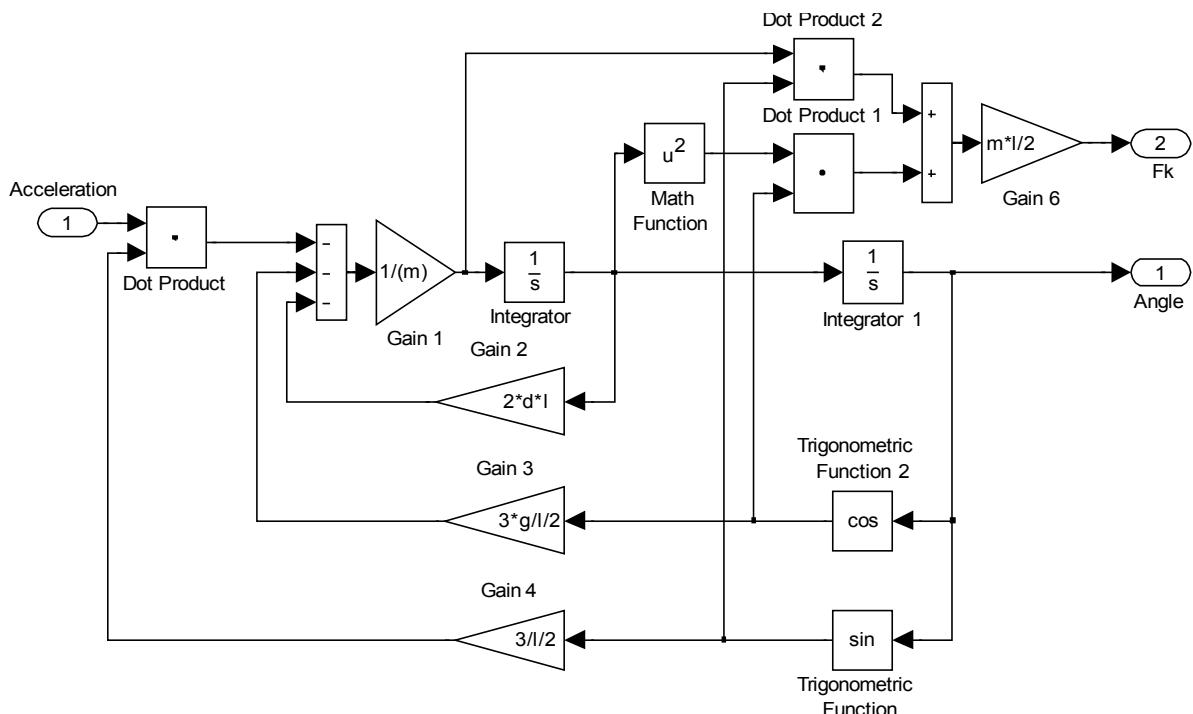
Pro model kyvadla neovlivňující vozík vytvoříme simulinkové schéma pouze z této rovnice.



Obrázek 4.2.1: Schéma kyvadla neovlivňující vozík

Dále vytvoříme simulinkové schéma s přidaným působením síly kyvadla na vozík z modelu vozíku (4.1.11)

$$F_k(t) = \frac{1}{2} m_l l (\ddot{\varphi}(t) \sin(\varphi) + \dot{\varphi}^2(t) \cos(\varphi)).$$

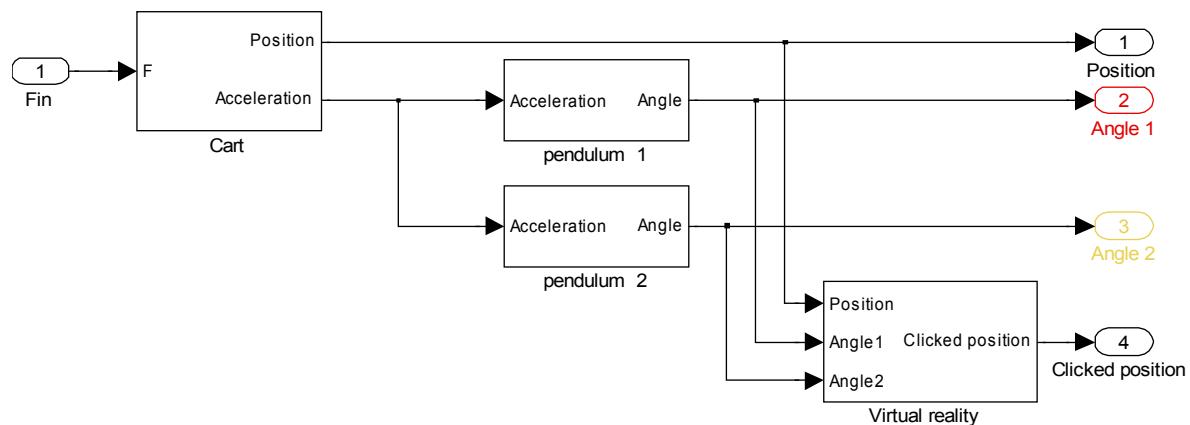


Obrázek 4.2.2: Schéma kyvadla ovlivňující vozík

4.3.Kompletní model

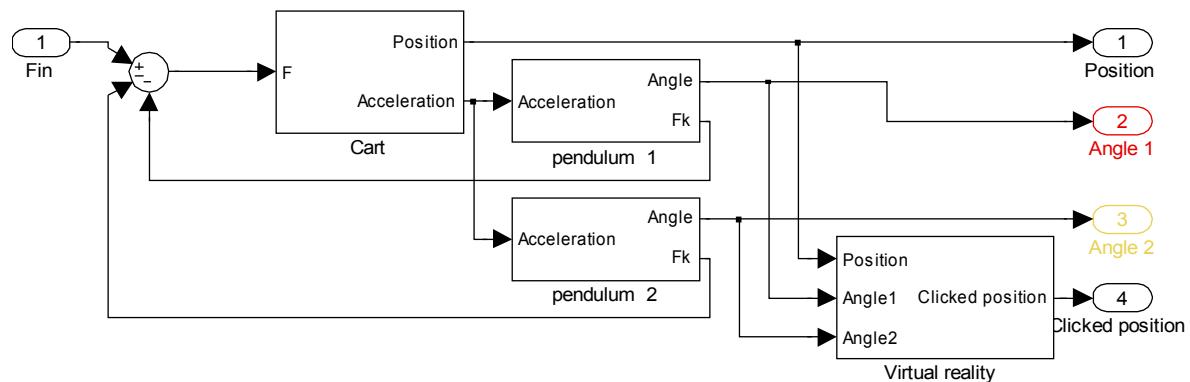
V této kapitole složíme celý model kuličky na tyči. Použijeme model vozíku a model kyvadla odvozený v předchozích kapitolách. Výhodou takto oddělených návrhů je možnost přidávat do modelu libovolný počet kyvadel.

Model se dvěma kyvadly neovlivňující vozík (obrázek 4.3.1) má na vstupu sílu působící na vozík $F(t)$ a na výstupu polohu vozíku $x(t)$ a náklony kyvadel φ_1 a φ_2 .



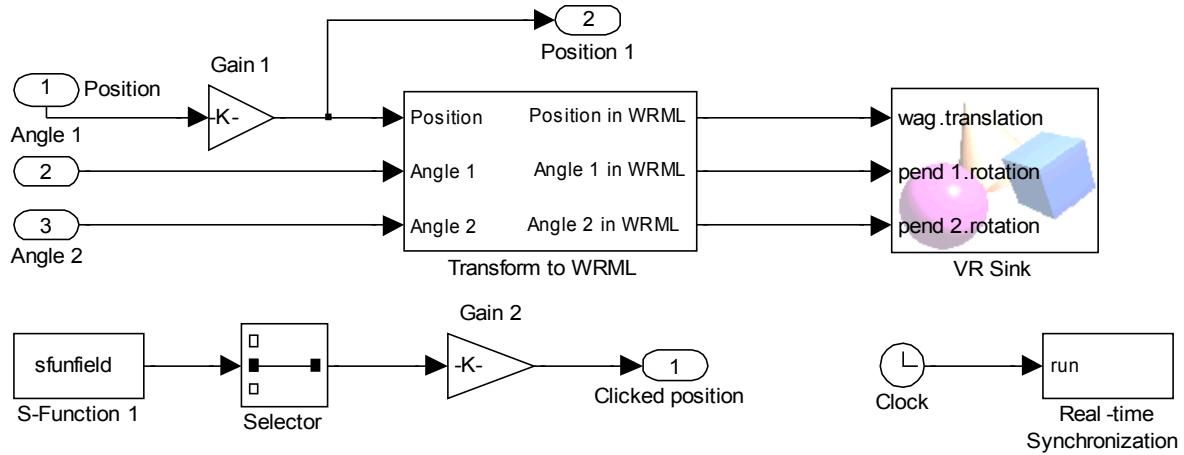
Obrázek 4.3.1: Model se dvěma kyvadly neovlivňující vozík

Pro model se dvěma kyvadly ovlivňující vozík (obrázek 4.3.2) je model podobný. Ke vstupní síle $F(t)$ se přičte silový účinek obou kyvadel. Na výstupu jsou úhly natočení obou kyvadel φ_1 a φ_2 a pozice vozíku $x(t)$.



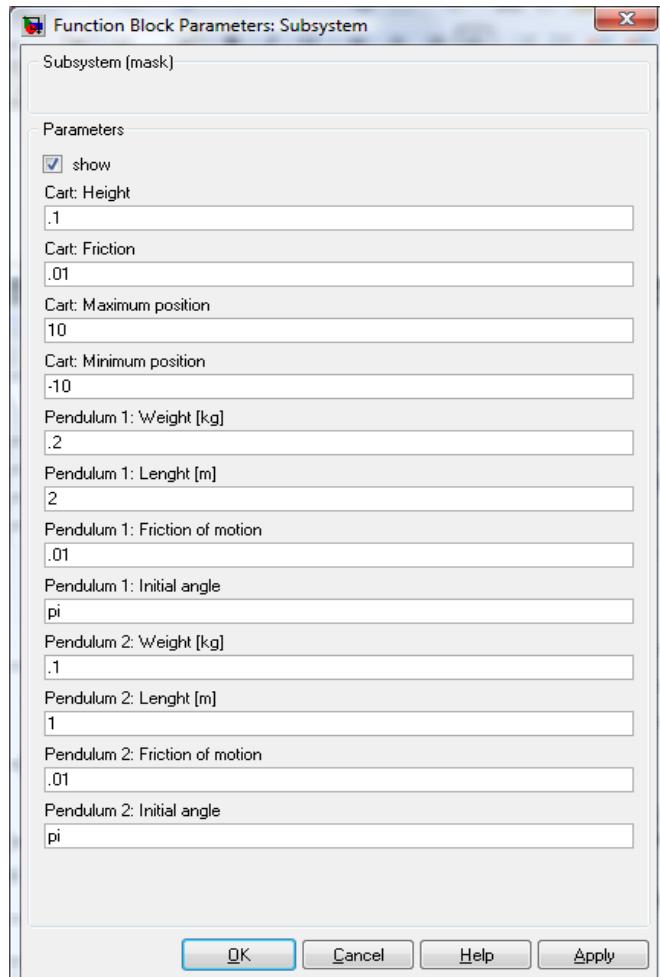
Obrázek 4.3.2: Model se dvěma kyvadly ovlivňující vozík

Blok Virtual reality má jako vstupy polohu vozíku a úhly natočení ramen. Poloha vozíku je přepočítaná tak, aby se v maximech polohy dostal vozík na kraj nosníku, po kterém se vozík pohybuje.



Obrázek 4.3.3: Blok virtuální reality

Dialog nastavení modelu (maska) obsahuje všechny nezbytné proměnné modelu.

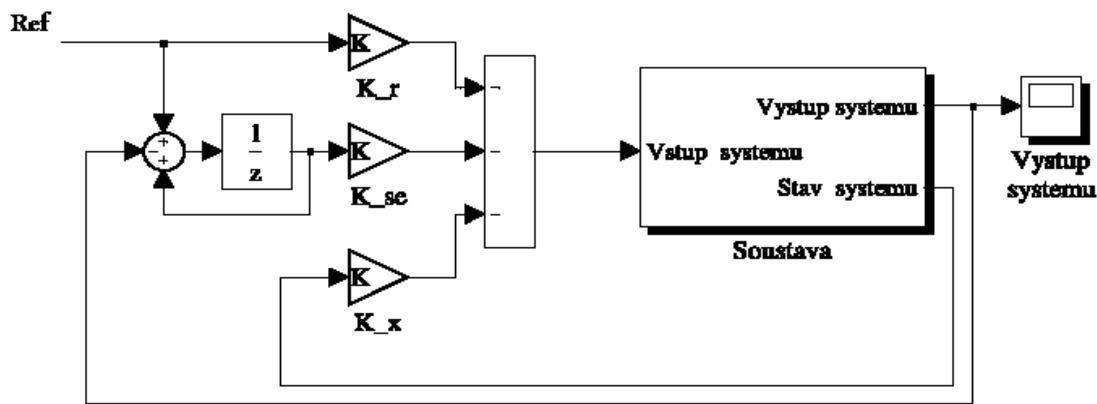


Obrázek 4.3.4: Nastavení parametrů modelu

4.4. Regulace inverzních kyvadel

V této kapitole se pokusíme uregulovat dvojici inverzních kyvadel. V tomto případě je regulace pomocí PID regulátoru nevhodná, protože by bylo nutné zapojit regulátory za sebou. To by vedlo ke složitému návrhu pro poslední PID regulátor.

Jako výhodnější se jeví použití stavové zpětné vazby. Použijeme LQG regulaci se sledováním konstantní reference. Pro zjednodušení budeme řídit kyvadla bez ovlivnění vozíku v horní poloze.



Obrázek 4.4.1: LQR se sledováním konstantní reference

Navrheme LQ regulátor se sledováním konstantní reference pro diskrétní systém

$$x(k+1) = M x(k) + N u(k+1).$$

Tento systém rozšíříme na stavovou rovnici ve tvaru

$$x''(k+1) = \begin{bmatrix} M & 0 & 0 \\ 0 & I & 0 \\ -C & I & I \end{bmatrix} x''(k) + \begin{bmatrix} N \\ 0 \\ -D \end{bmatrix} u(k) = M'' x''(k) + N'' u''(k+1)$$

a definujeme kvadratické kritérium optimality ve tvaru

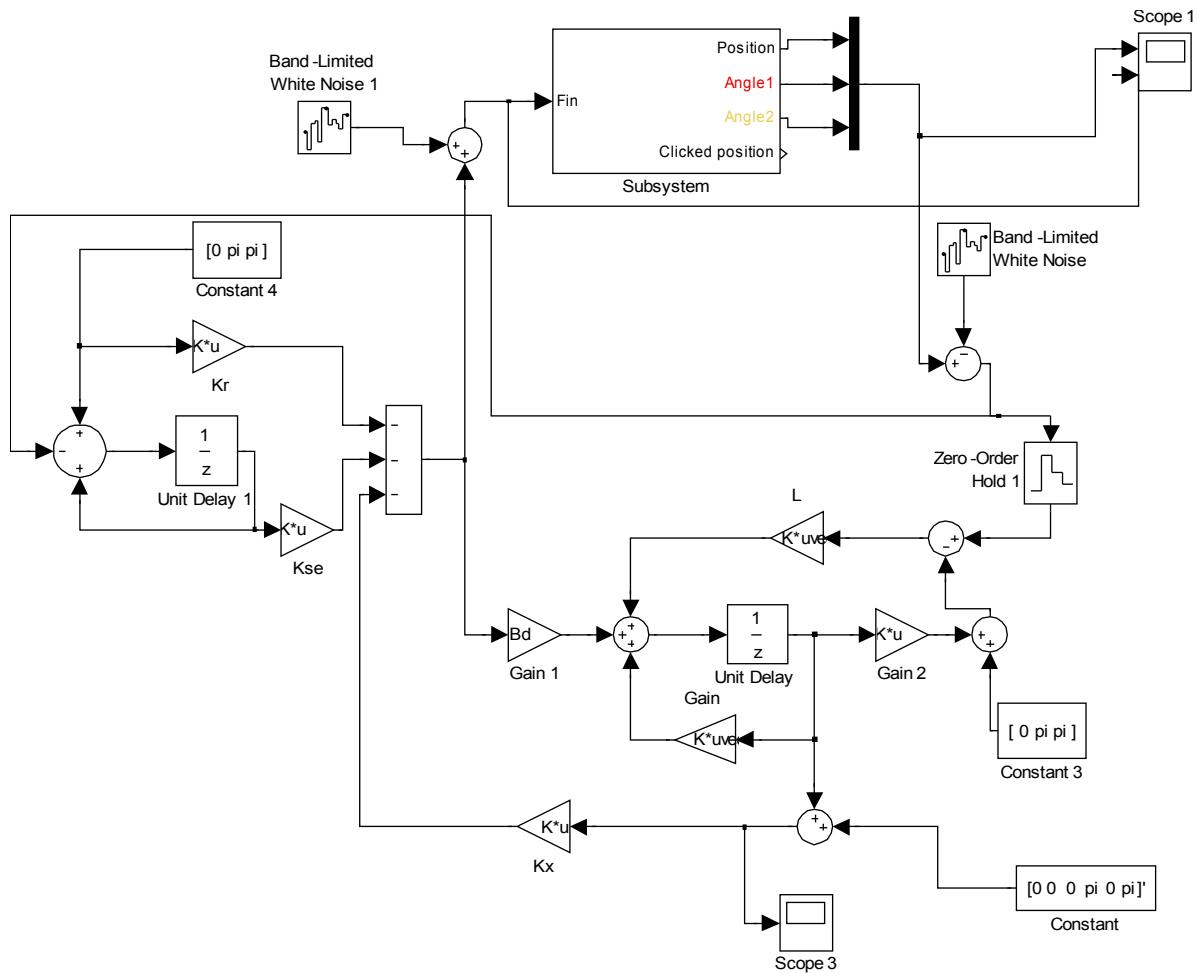
$$J = \frac{1}{2} \sum [x''^T(k) u''^T(k)] \begin{bmatrix} CTQeC & CTQe & 0 & CTQeD \\ QeC & Qe & 0 & QeD \\ 0 & 0 & Qse & 0 \\ DTQeC & DTQe & 0 & DTQeD + Ru \end{bmatrix} [x''^T(k) u''^T(k)]^T,$$

kde odhadneme váhové matici Qe , Qse a R pro řízení kyvadel na vozíku.

Dále vyřešíme Riccatiho rovnici, která limitně povede na optimální zákon řízení

$$u(k) = [Kx \quad Kr \quad Kse] \begin{bmatrix} x(k) \\ xr(k) \\ xse(k) \end{bmatrix} = Kx''(k)$$

Jako pozorovatele stavu navrheme Kalmanův filtr odhadem kovariančních matic Qd, Rd . Kalmanovo zesílení L a zesílení LQ regulátoru K zadáme do modelu.

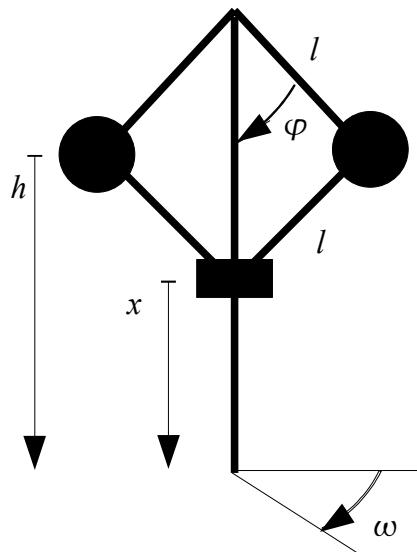


Obrázek 4.4.2: LQG řízení dvou kyvadel

5. Wattův odstředivý regulátor

Jako poslední model vytvoříme Wattův odstředivý regulátor používaný pro regulaci otáček parního stroje.

Regulátor se skládá ze dvou závaží, která rotují a jsou poháněna strojem, jehož otáčky mají být regulovány. Čím rychleji tato závaží rotují, tím větší je vlivem odstředivé síly jejich výchylka od svislé rotace. Uvedené vychýlení je převedeno na svislý pohyb nad jejich ukotvením, který je dále pákou a táhlem převeden k ventilu přivádějícím páru ke stroji. Je tak realizována mechanická záporná zpětná vazba, která dovoluje působením poměrně malých sil regulovat velmi výkonný stroj.



Obrázek 5.1: Wattův regulátor

Regulátor se skládá ze závaží zavěšených na ramenech délky $l[m]$. Ramena rotují podél svislé osy úhlovou rychlostí $\omega [rad\ s^{-1}]$, kterou udává poháněný stroj. Rotací vzniká odstředivá síla, která působí na závaží. To se oddálí od svislé osy rotace o úhel $\varphi [rad]$. Tento úhel nesmí být na počátku nulový, jinak se odstředivá síla nemůže projevit. Z tohoto důvodu a z důvodu fyzického omezení (nenulový průměr ramen a závaží) je úhel φ limitován na intervalu $<0.2 \dots 1.5>$ s počáteční hodnotou 0.2.

Nejprve odvodíme odstředivou kinetickou energii jednoho závaží $W_{kr}(t)[J]$ o hmotnosti $m[kg]$ při rotaci kolem svislé osy úhlovou rychlostí ω , při úhlku natočení ramen φ

$$W_{kr}(t) = \frac{1}{2} J \omega^2(t) \sin^2(\varphi(t)) = \frac{1}{2} m l^2 \omega^2(t) \sin^2(\varphi(t)).$$

Dále budeme uvažovat síly působící proti odstředivé síle. Kinetickou energii jednoho závaží při změně úhlu natočení ramene

$$W_{ks}(t) = \frac{1}{2} J \dot{\varphi}_{rl}^2(t) = \frac{1}{2} m l^2 \dot{\varphi}_{rl}^2(t),$$

kde m je hmotnost závaží a $\dot{\varphi} [rad s^{-1}]$ úhlová rychlosť. Potenciální energie závaží $W_p(t)[J]$ je závislá na jeho výšce $h = l \cos(\varphi)$ a na gravitačním zrychlení $g [m s^{-2}]$

$$W_p(t) = \frac{1}{2} m g h = \frac{1}{2} m g l \cos(\varphi).$$

Lagrangeova funkce pro složený pohyb obou ramen

$$L(t) = 2 W_{ks}(t) + 2 W_{kr}(t) + 2 W_p(t). \quad (5.12)$$

Po dosazení energií do 5.12

$$L(t) = m l^2 \dot{\varphi}^2(t) + m l^2 \omega^2(t) \sin^2(\varphi(t)) + 2 m l g \cos(\varphi(t)).$$

Přidáme tlumení způsobené třením závislém na rychlosti otáčení ramene $\dot{\varphi}$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\varphi}} \right) - \frac{\partial L}{\partial \varphi} = -2 \delta J \dot{\varphi}(t)$$

a vypočteme Lagrangerovu rovnici

$$2 m l^2 \ddot{\varphi}(t) - 4 m l^2 \omega^2 \sin(\varphi(t)) \cos(\varphi(t)) + 2 l m g \sin(\varphi(t)) = 2 \delta m l^2 \dot{\varphi}(t)$$

Po úpravě dostaneme pohybovou rovnici úhlu natočení ramene φ v závislosti na otáčkách stroje ω

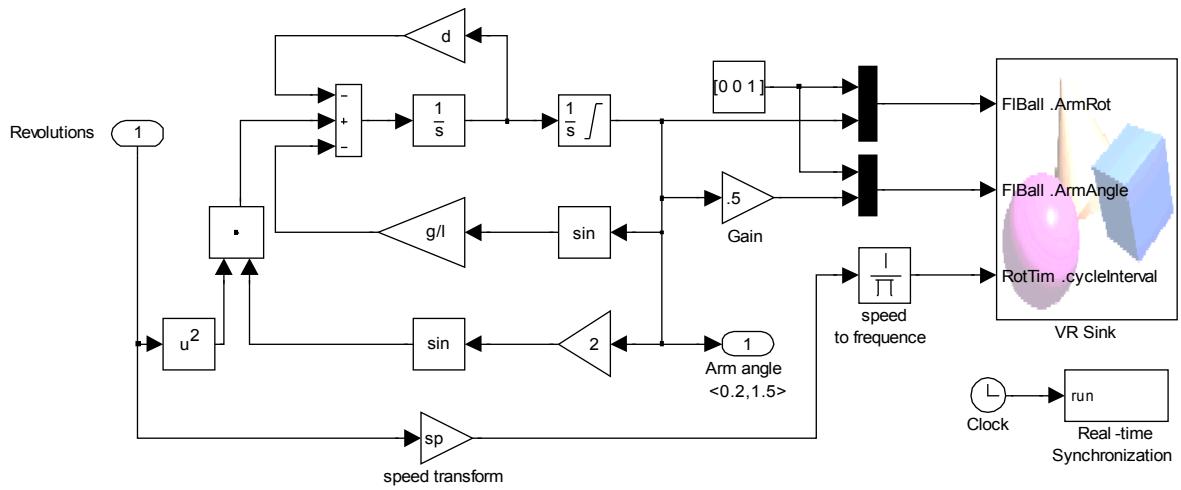
$$\ddot{\varphi}(t) - 2 \omega^2 \sin(2\varphi(t)) + \frac{g}{l} \sin(\varphi(t)) = \delta \dot{\varphi}(t). \quad (5.13)$$

Z rovnice nám vypadla hmotnost m . Ta by se projevila, pokud bychom uvažovali silové působení závaží na otáčky stroje. To ale lze zanedbat z důvodu nepoměru hmotnosti závaží regulátoru a hmotnosti stroje většinou opatřeného těžkým setrvačníkem.

Nakonec je třeba převést úhel natočení ramene φ na velikost průtoku páry nebo její tlak. Lze si vybrat mezi rotačními nebo posuvnými ventily, pákovými převody a kladkami. Vybereme posuvný ventil přímo připojený na nejspodnější část Wattova regulátoru. To lze vyjádřit jako $x = 2l \cos(\varphi)$ s tím že se maximální otevření a zavření přepočítá z omezení natočení ramene φ . Pro nulové otáčky je úhel natočení ramene $\varphi = 0.2 [rad]$ a ventil je

úplně otevřen. Pro otáčky, kdy je úhel natočení ramene maximální $\varphi = 1.5[\text{rad}]$ je ventil zavřen.

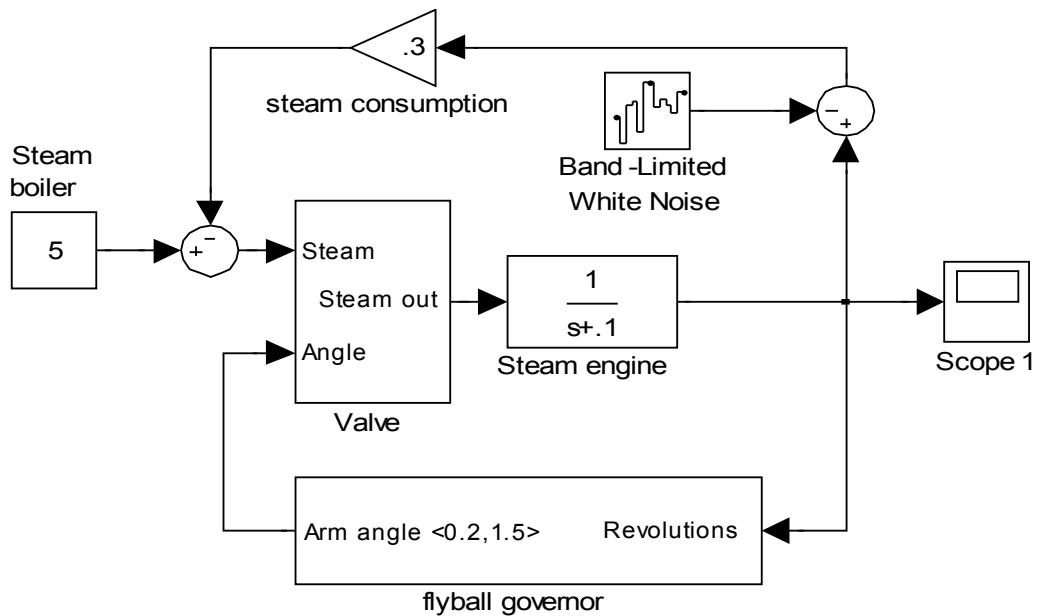
Vytvoříme Simulinkový model, který má jako vstup otáčky stroje ω a jako výstup úhel natočení ramene φ . Tento model obsahuje i virtuální scénu.



Obrázek 5.2: Schéma Wattova regulátoru

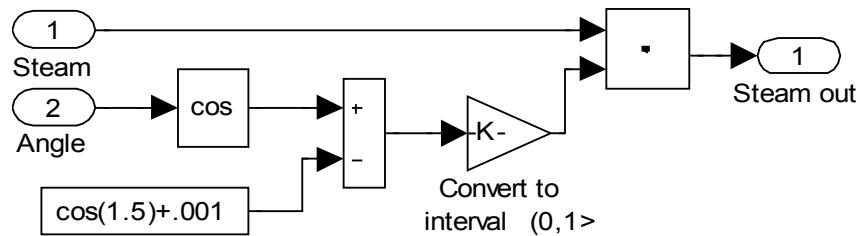
5.1.Kompletní model

Wattův regulátor zapojíme na konstantní zdroj páry. Parní stroj budeme simuloval přenosem prvního řádu a spotřebu pomocí zpětné vazby na zdroj páry. Jako proměnou zátěž přidáme generátor náhodných hodnot.



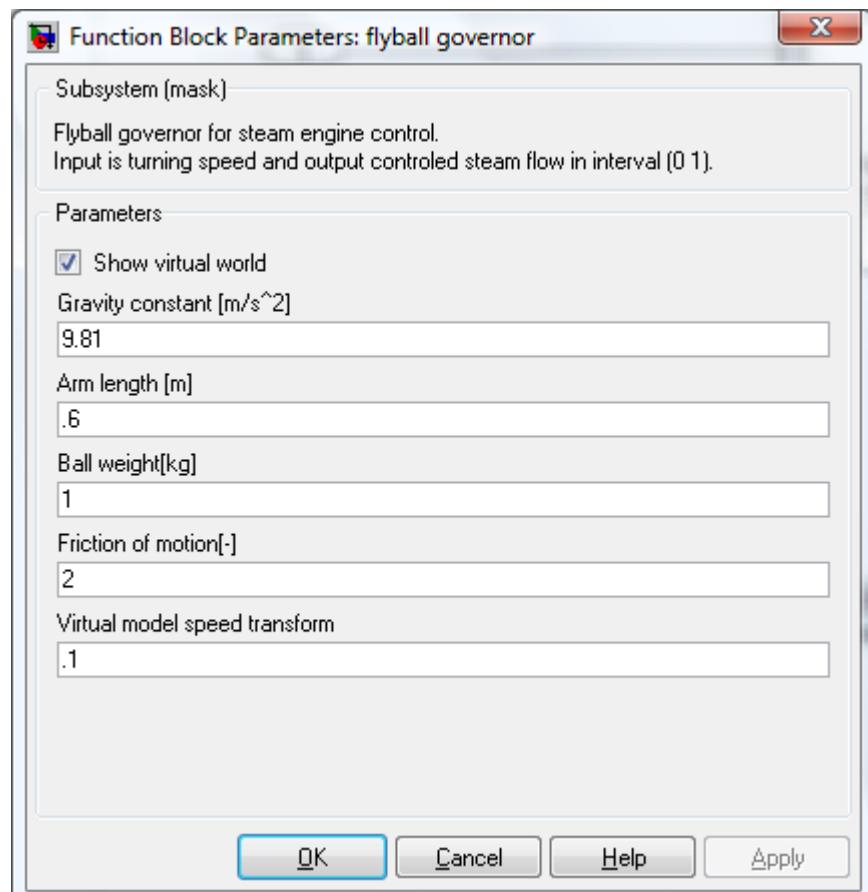
Obrázek 5.1.1: Zapojení kompletního modelu

Simulaci ventilu pro vytvoříme jako vstup do parního stroje ovládaný pomocí náklonu ramene regulátoru.



Obrázek 5.1.2: Simulace ventilu

Nastavení parametrů regulátoru pomocí masky. Parametr virtual model speed transform reguluje rychlosť otáčení ve 3D scéně.



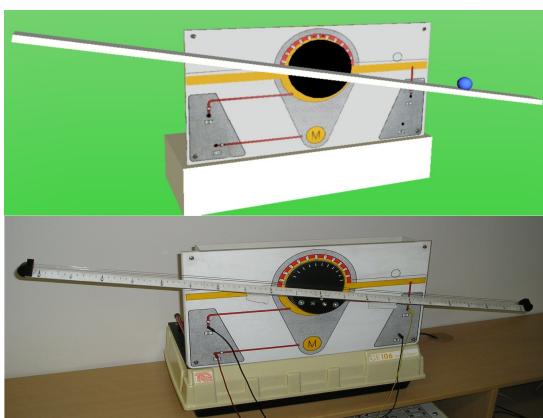
Obrázek 5.1.3: Nastavení regulátoru

6. Závěr

Díky Virtual Reality Toolboxu bylo možné vytvořit interaktivní trojrozměrné prostředí s dynamickým modelem Simulinku s reálnými nebo smyšlenými parametry. Detekce kliknutí myši ve scéně TouchSensorem a pomocí s-funkce předání polohy kliknutí jako proměnné do simulinku, dělá tyto modely interaktivní a uživatelsky příjemné. Jako vstup řízení lze také připojit jakýkoliv jiný zdroj signálu, například joystick.

Výhodou VRML je možnost zobrazení virtuálního modelu i na vzdáleném počítači pomocí internetového prohlížeče.

Trojrozměrnou scénu lze vytvořit velmi podobnou originálnímu laboratornímu přístroji a to i s jednoduchými geometrickými tvary (Obrázek 6.1).



Obrázek 6.1: Virtuální a reálný model kuličky na tyči

Nevýhodou VRML v Matlabu a Simulinku špatná podpora javascriptů, které jazyk VRML podporuje. Tím by se zjednodušila komunikace se Simulinkem a byla by možná větší interaktivita s uživatelem.

Model kuličky lze řídit i pomocí jiných regulátorů než jen PID. Také ho lze řídit jako SISO (jeden vstup, jeden výstup) nebo jako SIMO (jeden vstup, více výstupů) model. Tento model má výhodu rychlé reakce na vstupní signál.

Model dvou kyvadel je na řízení mnohem těžší než kulička na tyči. Tento model se více hodí pro pokročilé návrhy složitějších regulátorů, např. LQG a MPC.

Na modelu Wattova regulátoru lze simulovat změny rychlosti otáčení v závislosti na jeho parametrech.

Pro správnou funkci modelu je třeba mít nainstalovaný Virtual Reality Toolbox, který nebývá součástí základní instalace Matlabu. Modely byly testovány v Matlabu verze 2007b 64b s Virtual Reality Toolboxem 2.0 pod operačním systémem Windows Vista 64b.

Literatura

- [1]: Jiří Žára, VRML 97 Laskavý průvodce virtuálními světy , 1999 ,Computer Press, a.s.
- [2] : <http://www.mathworks.com/access/helpdesk/help/toolbox/vr/>
- [3]: P. Noskiewič, Modelování a identifikace systémů, 1999,Montanex