

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

## PROCEDURÁLNÍ ANIMACE LIDSKÉ CHŮZE

SEMESTRÁLNÍ PROJEKT  
TERM PROJECT

AUTOR PRÁCE  
AUTHOR

Bc. PETR MOHELNÍK

BRNO 2016



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

**FACULTY OF INFORMATION TECHNOLOGY**  
**DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA**

# **PROCEDURÁLNÍ ANIMACE LIDSKÉ CHŮZE**

PROCEDURAL ANIMATION OF HUMAN WALK

## **SEMESTRÁLNÍ PROJEKT**

TERM PROJECT

## **AUTOR PRÁCE**

AUTHOR

**Bc. PETR MOHELNÍK**

## **VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. LUKÁŠ POLOK**

BRNO 2016

## Abstrakt

Animace lidské chůze má velké využití v interaktivních aplikacích, převážně počítačových hrách. Existuje mnoho způsobů tvorby této animace, které se liší v kompromisu mezi přirozeností, kontrolou nad animací a výpočetním časem. Tato práce se zabývá implementací procedurální animace, která je vhodná pro chůzi po nerovném terénu. Pro manipulaci s modelem lidského těla se využívá skeletální animace. Dále je popsána a implementována inverzní kinematika, díky čemuž může být animace přizpůsobována pomocí parametrů. V práci jsou také popsány některé existující implementace procedurální animace lidské chůze. A nechybí ani popis fází lidské chůze, abychom je mohli co nejvěrněji napodobit.

## Abstract

Animation of human walk is employed in many interactive applications, mostly in computer games. There are many ways to create such animation which differ in compromise among naturalness, control over animation and computing time. This work implements procedural animation which is applicable for walking on uneven terrain. Skeletal animation is used for manipulation with model of human body. Furthermore inverse kinematics is described and implemented. That allows for customization of animation using parameters. This work also describes some existing implementations of procedural animation of human walk. It also describes phases of human walk, so we accurately approximate them.

## Klíčová slova

počítačová grafika, skeletální animace, lidská chůze, inverzní kinematika

## Keywords

computer graphics, skeletal animation, human walk, inverse kinematics

## Citace

Petr Mohelník: Procedurální animace lidské chůze, semestrální projekt, Brno, FIT VUT v Brně, 2016

# Procedurální animace lidské chůze

## Prohlášení

Prohlašuji, že jsem tento semestrální projekt vypracoval samostatně pod vedením Ing. Lukáše Poloka.

.....

Petr Mohelník  
6. května 2016

## Poděkování

Děkuji svému vedoucímu práce Ing. Lukáši Polokovi za odbornou pomoc poskytnutou při řešení tohoto projektu.

© Petr Mohelník, 2016.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Skeletální animace</b>	<b>4</b>
2.1	Kostra . . . . .	4
2.2	Skinning . . . . .	5
2.2.1	Linear blend skinning . . . . .	5
2.2.2	Dual quaternion skinning . . . . .	7
2.2.3	Metoda obálek . . . . .	7
2.3	Tvorba animace . . . . .	7
2.3.1	Procedurální animace . . . . .	8
2.3.2	Fyzikálně založená simulace . . . . .	9
2.3.3	Editace pohybu . . . . .	9
<b>3</b>	<b>Kinematiky</b>	<b>11</b>
3.1	Jacobiho inverzní metoda . . . . .	12
3.2	Cyclic Coordinate Descent . . . . .	13
<b>4</b>	<b>Lidská chůze</b>	<b>15</b>
4.1	Fáze krokového cyklu . . . . .	16
4.1.1	Počáteční kontakt . . . . .	17
4.1.2	stádium zatěžování . . . . .	17
4.1.3	Mezistoj . . . . .	17
4.1.4	Konečný stoj . . . . .	17
4.1.5	Předšvihová fáze . . . . .	17
4.1.6	Počáteční švih . . . . .	17
4.1.7	Mezišvih . . . . .	17
4.1.8	Konečný švih . . . . .	18
<b>5</b>	<b>Existující implementace</b>	<b>19</b>
5.1	Interactive Animation of Personalized Human Locomotion . . . . .	19
5.2	Animation of Human Walking in Virtual Environments . . . . .	20
<b>6</b>	<b>Návrh a realizace řešení</b>	<b>22</b>
6.1	Načtení modelu . . . . .	22
6.2	Zobrazení . . . . .	23
6.3	Animace . . . . .	25
6.4	Bézierovy křivky . . . . .	26
6.5	Pohyb nohou . . . . .	27

6.5.1	Stádium zatěžování . . . . .	28
6.5.2	Mezistoj . . . . .	28
6.5.3	Zvedání paty . . . . .	29
6.5.4	Švihová fáze . . . . .	29
6.5.5	Inverzní kinematika . . . . .	32
<b>7</b>	<b>Závěr</b>	<b>34</b>

# Kapitola 1

## Úvod

Počítačová animace má velké využití pro počítačové hry a filmy. Existuje množství způsobů tvorby a zobrazování animace s různými výhodami a nevýhodami. V této práci se zabýváme tvorbou animace lidské chůze - jedné z nejpoužívanějších a nejdůležitějších animací. Tato animace musí být co nejpřirozenější, protože lidé jsou velmi vnímaví na správnost lidské chůze a snadno rozpoznají, kdy je animace nepřirozená.

Existují různé přístupy k tvorbě animace lidské chůze. Tyto přístupy se liší v kompromisu mezi přirozeností, kontrolou nad animací a výpočetním časem. Populární způsob tvorby animace je *motion capture*, kde se zachytává pohyb herce pomocí kamer. Ačkoliv tento způsob dosahuje velké přirozenosti, je velmi drahý. *Fyzikálně založená simulace* umožňuje velmi dobrou interakci s prostředím, ale je velmi výpočetně náročná. Tato práce se zabývá *procedurální animací*. Při tomto způsobu se využívá znalostí o *cyklu lidské chůze*, který je studován již od starověku. Procedurální systém umožňuje jednoduše vytvořit animaci pomocí specifikace množství parametrů. Je výpočetně nenáročná a vhodná pro interakci s prostředím, ale ne tak fyzikálně přesná jako fyzikálně založená simulace. Také není tak detailní jako v *motion capture*. V této práci si klademe za cíl vytvoření systému, který umožní rychlou a jednoduchou tvorbu specifické animace lidské chůze i uživatelům bez animačních znalostí a schopností, kterou budou moci použít např. v počítačové hře.

Druhá kapitola se zaměřuje na *skeletální animaci*. Přibližuje co je to *kostra* a jak se na ní připojí *kůže*. Dále obsahuje stručný přehled metod tvorby animace. Třetí kapitola se věnuje *kinematikám*. Definuje co je to *inverzní kinematika* a popisuje některé metody jejího řešení. Ve čtvrté kapitole je podrobně popsána lidská chůze a její fáze. Pátá kapitola obsahuje popis některých existujících implementací procedurální animace využívající tyto znalosti. Pátá kapitola nastiňuje výslednou aplikaci a její řešení a popisuje implementované části v rámci semestrálního projektu.

## Kapitola 2

# Skeletální animace

*Skeletální animace* je způsob animace, který využívá dvě základní komponenty: model (nejčastěji polygonální) nazýván *kůže* a *kostru*, na kterou je kůže připojena. Jestliže se pohne kost, pohne se také připojená kůže. V klasické animaci, je vytvořen model pro každý *klíčový snímek* animace a tyto modely se postupně zobrazují. Oproti tomu se skeletální animace skládá pouze z jednoho modelu a připojené kostry, pro kterou jsou uloženy pozice kostí v klíčových snímcích např. cyklu chůze. Každý tento snímek definuje pozici každé kosti v konkrétní čas. Díky tomu má mnohem nižší paměťovou náročnost. Také skeletální animace umožňuje vysokou interakci s prostředím. Kost se může zarazit o překážku apod. Skeletální animace také mohou být použité na více různých modelech. Někdy programy poskytují předpřipravenou kostru, okolo které se vytvoří kůže. Skeletální animace má také své nevýhody. Pohyby kostí a připojených vrcholů spotřebovávají výpočetní výkon. V animacích obecně se musí řešit interpolace mezi dvěma klíčovými snímky, protože zobrazení snímku může proběhnout v libovolný čas.

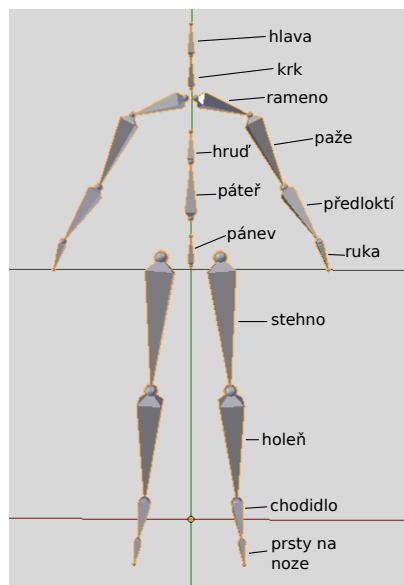
### 2.1 Kostra

Kostra je abstraktní model lidského, zvířecího těla nebo jiného objektu. Jedná se o stromovou strukturu (obr. 2.1). Uzly jsou nazývány *kosti* nebo *klouby*. Ke kostem jsou přiřazeny vrcholy. Kosti mají stupně volnosti, které určují v jakém směru a jak moc se můžou rotovat. Připojování modelu na kostru se nazývá *skinning*.

Každá kost má lokální transformaci: posunutí, rotaci a zvětšení. Tato transformace se může uložit jako matice a určuje transformaci vůči rodičovské kosti, pokud existuje. Pro realistické modely člověka se většinou používá pouze rotace. Výsledná globální transformace kosti se získá kombinací rodičovské globální transformace a vlastní lokální transformace. Globální transformace všech kostí se vypočítá procházením stromové struktury kostry a postupným násobením transformačních matic od kořene k listům. Pro kořenovou kost je lokální transformace její globální.

Před připojením kůže na kostru se kostra nastaví do pozice nazývané *bind pose* [3]. To je základní pozice kostry před jakýmkoliv pohybem. V tuto chvíli se na kostru připojí kůže. Následně jsou určeny a uloženy *bind pose* matice  $\mathbf{B}$  pro každou kost. Ty určují jejich transformaci z počátku souřadného systému kostry do *bind pose*. Matice  $\mathbf{B}_a$  každé kosti určuje její aktuální transformaci z počátku souřadného systému kostry. Před prvním pohybem kostry jsou shodné s  $\mathbf{B}$ . Chceme-li změnit pózu kostry (matici  $\mathbf{B}_a$  nějaké kosti) a tím i připojené kůže, musí se vrcholy transformovat ze souřadného systému kůže do





Obrázek 2.1: Příklad hierarchie kostry.

souřadného systému kostry. K tomu se použije *inverzní bind pose matice*  $\mathbf{B}_i$ . Ta určuje transformaci kosti potřebnou pro přesun kosti z bind pose do počátku souřadného systému. Tedy invertuje transformaci, která byla na kůži aplikována v bind pose. Matematický zápis transformace  $\mathbf{T}$  (obr. 2.2) určující novou pozici kosti je následující:

$$\mathbf{T} = \mathbf{B}_i \mathbf{B}_a \quad (2.1)$$

Poté se může použít  $\mathbf{T}$  na výpočet nové pozice vrcholů kůže. Vrchol bude pořád ve stejné relativní pozici vůči kosti. Transformování se často provádí na grafické kartě ve vertex shaderu.

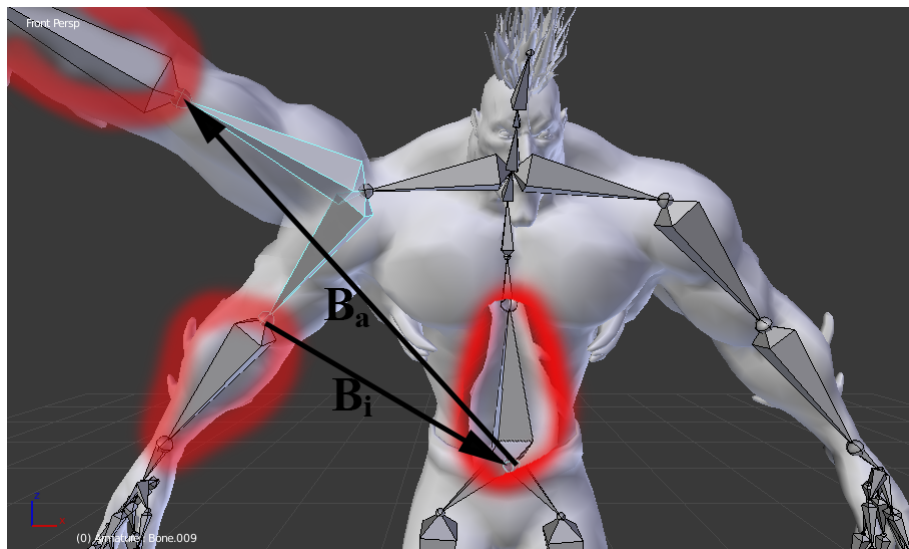
## 2.2 Skinning

Skinning [12] je nazýván proces přidělování kostí vrcholům kůže a určování jejich vah. Tento proces může probíhat algoritmicke nebo ručně v některém z modelovacích programů. Nejjednodušší způsob přiřazení kůže ke kostře je každé kosti přiřadit samostatný pevný objekt jako je např. válec. Na každý takový objekt se poté aplikuje transformace přiřazené kosti. Tento přístup není vhodný pro realistické animování.

Pokročilejší, ale stále základní metoda je *Jednoduchý skinning*, který již používá kůži, tedy jeden model pro celou kostru. Přiřazuje jeden vrchol k jedné kosti. Každý vrchol se poté transformuje podle přiřazené kosti. Je používán například ve starších hrách. Je dobře použitelný pro modely s nízkým počtem trojúhelníků. U detailnějších modelů způsobuje nepěkné deformace v kloubech a animace vypadá nepřirozeně.

### 2.2.1 Linear blend skinning

*Linear blend skinning* [12] je modernější algoritmus. Je nepublikovaný a vyskytuje se i pod jinými názvy jako je *skeleton-subspace deformation*, *enveloping* nebo pouze *skinning*.



Obrázek 2.2: Transformace předloktí. Prvně se aplikuje inverzní bind pose matice a poté aktuální transformace kosti. Model byl převzat z [5].

Každému vrcholu umožňuje přiřazení více kostí. Každá dvojice kost a vrchol má přiřazenou váhu. Váha určuje jak velký vliv má pohyb konkrétní kosti na výslednou pozici vrcholu. Na vstupu očekává následující data:

- Kůže v bind pose, typicky reprezentovaná jako polygonální model. Propojení mezi vrcholy se v průběhu nemění, pouze pozice vrcholů.
- Transformace kostí, reprezentované jako matice  $\mathbf{T}_1, \dots, \mathbf{T}_m$  určené podle rovnice 2.1. Tyto matice jsou typicky jediná veličina, která se v průběhu animace může měnit.
- Váhy vrcholů, pro každý vrchol  $\mathbf{v}_i$  máme váhy  $w_{i,1}, \dots, w_{i,m} \in \mathbb{R}$ . Každá váha  $w_{i,j}$  určuje vliv kosti  $j$  na vrchol  $i$ . Celkový součet všech vah je 1, přičemž žádná váha není záporná.

Množství kostí ovlivňující jeden vrchol se může významně lišit, např. od 1 do 10. Kvůli limitacím grafického hardwaru se často uvažuje že jich není více než 4. Transformovaná pozice vrcholu  $\mathbf{v}'_i$  se určí podle následující rovnice:

$$\mathbf{v}'_i = \left( \sum_{j=1}^m w_{i,j} \mathbf{T}_j \right) \mathbf{v}_i \quad (2.2)$$

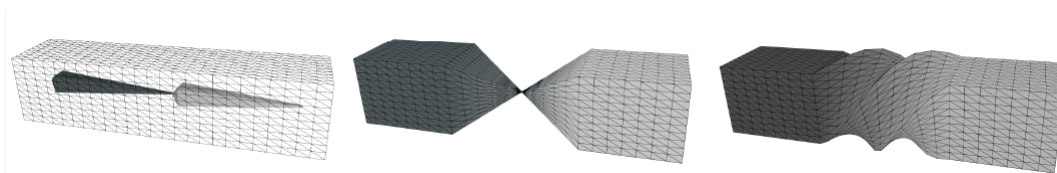
Tato metoda podává dobré výsledky, pokud míchané transformace nejsou příliš rozdílné. Problémy nastávají jestliže potřebujeme míchat transformace, které se významně liší v jejich rotaci. Z lineární kombinace rotací nevzniká rotace. To je důsledkem faktu, že Lieova grupa 3D transformací,  $SO(3)$ , není lineární prostor, ale zakřivená varieta. Pokud jsou si rotace blízké není to problém. Uvažujme ale tyto dvě rotace.

$$\mathbf{R}_1 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{R}_2 \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$\mathbf{R}_1$  je maticí identity a  $\mathbf{R}_2$  je rotace okolo osy Z o  $180^\circ$ . Lineární míchání  $0.5 \cdot \mathbf{R}_1 + 0.5 \cdot \mathbf{R}_2$  nám dá za výsledek matici hodnoty 1, která promítá 3D prostor na osu Z. To má za následek ztrátu objemu modelu. Velké relativní rotace nejsou vzácné, protože klouby jako ramena a zápěstí mají velký rozsah pohybu. Tyto problémy je možné řešit zavedením více parametrů než má linear blend skinning. Takové metody se nazývají *multilineární*. Tyto metody ale nemusí být žádoucí kvůli nutnosti vytvářet a ukládat více vah pro každý vrchol.

### 2.2.2 Dual quaternion skinning

Velmi rozšířenou metodou je nelineární metoda *Dual quaternion skinning* [14], kde se místo matic míchají *duální kvaterniony*. Tato metoda řeší artefakty vznikající u linear blend skinning, zachovává objem modelu a je velmi rychlá. Používá se v rozšířených modelovacích programech jako je např. Blender. Klasické kvaterniony umožňují reprezentovat 3D rotaci okolo osy. Tato osa ale musí procházet počátkem souřadného systému. Oproti tomu duální kvaterniony toto omezení nemají - osa může být libovolná. Kromě toho duální kvaterniony umožňují popsat posunutí. Ale neumožňují popsat změnu měřítka. Proto nemůže docházet ke ztrátě objemu, jako u matic, kde násobením může vznikat zmenšení. Duální kvaterniony jsou také výhodné pro GPU hardware, protože obsahují méně hodnot než matice. Pro porovnání metod viz obrázek 2.3.



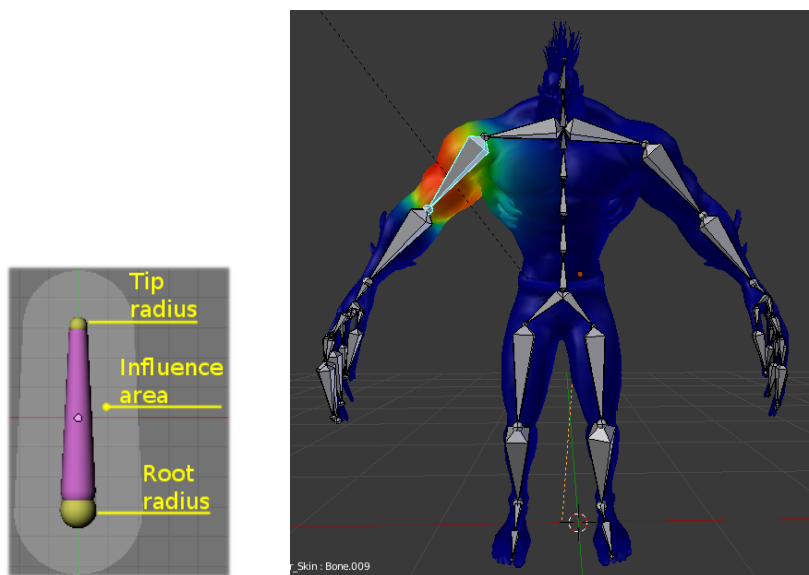
Obrázek 2.3: Zleva doprava: model v bind pose, linear blend skinning a dual quaternion skinning. Převzato z [12].

### 2.2.3 Metoda obálek

Přiřazení vah je možné automaticky *metodou obálek* [2]. Tato metoda je založena na blízkosti kostí a jejich geometrie. Každá kost má dvě oblasti vlivu. Vnitřní oblast, kde je geometrie plně ovlivněna touto kostí. A vnější oblast, kde je geometrie méně ovlivněna kostí, čím blíže je k okraji této oblasti. Pokud chceme mít váhy určené kvalitně, je nutné nastavit váhy ručně. To je možné v některém modelovacím programu, například Blender poskytuje *Weight Paint* mód, kde se pohybem štětce po modelu určují váhy vrcholů pro jednotlivé kosti. Tento proces je iterativní a náročný. Kreslením se odebírají nebo přidávají váhy vrcholům pro jednu aktivní kost. Poté se zkouší efekt na známé póze, upravují a vyhlazují váhy a tento proces se opakuje, dokud výsledek není dostatečně přirozený. Na obrázku 2.4 je ukázána obálka a model ve Weight Paint módu.

## 2.3 Tvorba animace

Animace člověka se vyskytuje ve velkém množství aplikací v různých prostředích a podmínkách. Tyto animace by proto měly být přizpůsobitelné prostředí v reálném čase. Také



Obrázek 2.4: Metoda obálek v programu Blender (převzato z [2]) a model [5] ve Weight paint módu v Blenderu.

by měly působit přirozeně a tím přidávat na reálnosti. Základní nástroje pro specifikaci pohybu jsou založeny na *kinematikách*. Existují dva způsoby popisu kinematické informace: *inverzní* a *přímá kinematika*, viz kapitola 3.

Welbergen a spol. [18] dělí techniky tvorby animace do tří skupin: *procedurální animace*, *fyzikálně založená simulace* a *editace pohybu*. Techniky se také mohou kombinovat dohromady a využívat vhodnější techniku pro danou situaci nebo část těla. Může se například použít technika editace pohybu dokud nedojde k interakci s prostředím a ta se může zpracovat pomocí fyzikálně založené simulace. Nebo pomocí procedurální animace vytvořit gesta rukou a fyzikální simulací zachovat fyzikálně přirozené balancování spodní části těla.

### 2.3.1 Procedurální animace

Procedurální animace popisuje animaci algoritmicky pomocí matematických vzorců. Její úspěch závisí na tom, jak hluboce jsme schopni pochopit a modelovat simulovaný pohyb. Metody mohou popisovat rotace kostí přímo nebo popisovat cesty koncových uzlů kostry (chodidla), potom se používá inverzní kinematika.

Procedurální animace umožňuje přesné časování a pozicování končetin a může využívat velké množství parametrů, pomocí kterých je možné upravit výstup na míru požadavkům. Dobře se adaptují na prostředí a jsou málo výpočetně náročné. Nicméně je často problematické zakomponovat detaily jako jsou u editace pohybu do matematického popisu. Také je nutné explicitně zachovávat fyzikální přirozenost v procedurálním modelu pohybu pro všechny možnosti parametrů. Je často používána pro animaci gest a mluvení, které vyžadují velké množství parametrů.

### 2.3.2 Fyzikálně založená simulace

V mnoha případech je vhodné počítat s *dynamikami* [16], aby byla zachována realističnost pohybu. Tyto případy mohou být:

- jestliže postava nese náklad
- jestliže postava musí reagovat na externí síly jako poryvy větru
- povrch na kterém se pohybuje je komplexní (schody atp.)

Dynamiky se používají u technik ze skupiny fyzikálně založené simulace. Dynamiky jsou založené na Newtonových pohybových zákonech. Spojují síly (resp. točivé momenty rotací) do výsledného pohybu (resp. rotace) podle rovnice:

$$f = m \cdot \ddot{x} \quad (2.3)$$

kde  $f$  je síla aplikovaná na objekt,  $m$  je jeho hmotnost,  $\ddot{x}$  je druhá derivace  $x$  podle času. Pro rotace vypadají rovnice podobně. Točivé momenty a úhlové pozice jsou spojeny pomocí:

$$t = i \cdot \ddot{\theta} + \dot{\theta} \times i \cdot \dot{\theta} \quad (2.4)$$

kde  $t$  je točivý moment,  $i$  je matice setrvačnosti,  $\dot{\theta}$  je úhlová rychlost a  $\ddot{\theta}$  je úhlové zrychlení. *Přímá dynamika* je aplikace těchto zákonů k výpočtu pohybu vygenerovaného danou silou. *Inverzní dynamika* se zabývá určováním síly, která by vygenerovala daný pohyb.

Ve fyzikálně založené simulaci je model typicky reprezentovaný jako systém pevných těles propojených klouby. Každé z těchto pevných těles má přiřazené fyzikální vlastnosti jako např. hmotnost. Pohyb je generován manipulací s točivými momenty kloubů s použitím přímé dynamiky. Pro umožnění kolizí je nutné geometricky reprezentovat pevná tělesa. Použití samotného povrchového modelu je příliš výpočetně náročné, proto se používají pro kolize aproximace pomocí základních tvarů, např. válec.

Metody mohou fungovat na základě zadávání geometrických omezení jako parametrů animace. Tato omezení mohou být typicky splněna velkým množstvím různých točivých momentů. Proto se mohou zavést objektivní funkce pro preferenci jistých řešení. Tyto metody jsou pro animace v reálném čase velmi pomalé. Jiný způsob je používání kontrolérů, do kterých vstupuje žádaný stav systému. Výstupem je množina točivých momentů kloubů, které po aplikaci na systém vedou proměnné k jejich požadovaným hodnotám. Kontrolér využívá fyzikálních vlastností pohybujícího se těla. Úlohou kontroléru je minimalizace nesrovnalostí mezi aktuálním a požadovaným stavem. Síly a točivé momenty určené kontrolérem, gravitace a síly způsobené externími vlivy jsou aplikované na fyzické tělo. To je poté pohnuto pomocí přímé dynamiky.

Fyzikálně založená simulace poskytuje fyzikálně realistický pohyb a interakci s prostředím. Umožňuje zachování parametrů pod vlivem externích sil. Nicméně přesné časování a umísťování končetin je problematické. Ačkoliv fyzikální simulace poskytuje fyzikálně správný pohyb, není často dostačující pro vytvoření přirozeného pohybu.

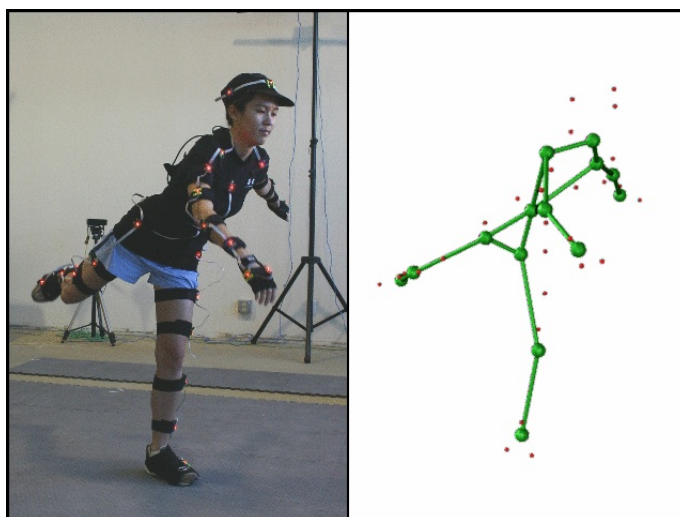
### 2.3.3 Editace pohybu

Metody editace pohybu vytváří pohyb na základně vzorů pohybu. Metody modifikace pohybu generují nová pohybová primitiva modifikací jednoho vzoru pohybu. Kombinační techniky vytváří pohybová primitiva pomocí databáze většího množství vzorových primitiv.

Pohyby se mohou vytvářet pomocí technik ze zpracování signálů, interpolací mezi vzory pohybů nebo pomocí statistických modelů.

Techniky editace pohybu zachovávají přirozenost vstupních vzorů pohybu, ale pouze jsou-li změny malé. Přirozenost i při větších změnách je zachována technikami používající více vzorů na jeden pohyb. Nicméně pro všechny techniky roste počet vzorů exponenciálně s počtem parametrů animace. Navíc techniky editace pohybu neposkytují fyzikální interakci s prostředím, ale jsou vázány na specifický kontext. Jsou vhodné pro vytváření animací v předstihu pro neinteraktivní aplikace, např. filmy. Pro interaktivní (např. videohry) je potřeba velká databáze pohybů. Pohyby se mohou vytvářet pomocí technik ze zpracování signálů, interpolací mezi vzory pohybů nebo pomocí statistických modelů.

Vstupní data mohou být tvořena animátorem ručně, ale mohou být získána i např. pomocí *motion capture*. Motion capture [15] jsou techniky pro nahrávání pohybů člověka, zvířete nebo jiného subjektu. Mají velké využití ve filmech a videohrách. Nejčastěji se používá optických metod pro zachycení pohybu. V Motion capture má herec okolo každého kloubu značku, které jsou zaznamenávány a sledovány kamerami, viz obrázek 2.5. Výstupem je množina lokací bodů v čase. Těmi se poté proloží kostra.



Obrázek 2.5: Automatická rekonstrukce kostry podle pozic značek. Převzato z [15].

## Kapitola 3

# Kinematiky

Kinematiky v počítačové grafice slouží pro manipulaci s kostrou. Jsou rozděleny na dvě části: přímá a inverzní kinematika. V přímé kinematice se hýbe s každou kostí v řetězci kostí a podle toho se určí výsledná pozice posledního článku - *koncového efektoru*. Výsledky při ručním vytváření animace jsou velmi závislé na schopnostech animátora, protože musí nastavovat pozice kostí od ruky. V této technice je obtížné omezovat pohyb, například že chodidlo nemůže proniknout do země.

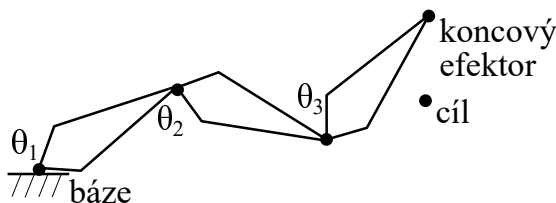
Máme-li *řetězec kloubů*, kde každý kloub  $i$  má přiřazen úhel  $\theta_i$ , který určuje jeho rotaci oproti jeho předkovi, viz obrázek 3.1. Potom stav řetězce je určen stavovým vektorem  $\theta(\theta_1, \dots, \theta_n)$ . Označme pozici koncového efektoru  $\mathbf{P}$ . Přímá kinematika řeší následující rovnici:

$$\mathbf{P} = f(\theta) \quad (3.1)$$

Každý kloub může mít více než jeden *stupeň volnosti*, tedy být definován více úhly. Tím je určena složitost struktury kloubů, která je definována jako součet stupňů volnosti všech prvků.

Oproti tomu v inverzní kinematice [9, 19] se hýbe pouze s koncovým efektem a pohyb ostatních kloubů se dopočítá, tak aby byl tento pohyb umožněn. To je matematicky zapsáno následovně:

$$\theta = f^{-1}(\mathbf{P}) \quad (3.2)$$



Obrázek 3.1: Příklad řetězce tří kloubů.

Řešení inverzní kinematiky není tak jednoduché jako přímé. Funkce  $f$  je nelineární a ačkoliv v rovnici 3.1 existuje unikátní namapování z  $\theta$  na  $\mathbf{P}$ , pro inverzní mapování v rovnici 3.2 může být nekonečné množství  $\theta$  pro jedno konkrétní  $\mathbf{P}$ . Analytické řešení pro libovolný řetězec kloubů neexistuje. Problém se řeší pomocí numerických metod pro řešení systémů



nelineárních rovnic. Z nekonečného množství řešení je snaha vybírat to nejvhodnější řešení. Různé metody preferují různé řešení a proto se metody vybírají na základě konkrétně řešeného problému. V případě lidské kostry má každý kloub určené stupně volnosti, aby rotovali jenom v určitých směrech a také je definován maximální úhel rotace. Například koleno rotuje pouze v jednom směru a nemůže být otevřeno více než  $180^\circ$  a zavřeno méně než cca  $30^\circ$ . Inverzní kinematika poskytuje lepší kontrolu nad koncovým efektořem, jehož cílová pozice nás většinou nejvíce zajímá. V modelovacím programu můžeme hýbat např. pouze chodidlem a nemusíme také pohybovat stehenní a lýtkovou kostí. Toho se také využije v procedurální animaci lidské chůze, kde budeme určovat trajektorie koncových efektořů. Techniky založené na kinematikách využívají empirickou a biomechanickou znalost pohybu pro generování animace.

Nyní si ukážeme některé způsoby řešení inverzní kinematiky.

### 3.1 Jacobiho inverzní metoda

*Jacobiho inverzní metoda* [9, 19, 7] je iterativní metoda využívající *Jacobiho matice* k aproximaci řešení. Vztah mezi kartézským prostorem koncového efektořu  $\mathbf{P}$  a prostoru kloubů úhlů  $\boldsymbol{\theta}$  je:

$$\dot{\mathbf{P}} = J(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} \quad (3.3)$$

kde tečka značí první derivaci podle času. Jacobiho matice  $J$  je  $m \times n$  matice:

$$J(\boldsymbol{\theta}) = \left( \frac{\partial P_i}{\partial \theta_j} \right)_{i,j} \quad (3.4)$$

kde  $m$  je počet dimenzí pozice koncového efektořu  $\mathbf{P}$ ,  $n$  je počet dimenzí  $\boldsymbol{\theta}$ ,  $i = 1, \dots, m$  a  $j = 1, \dots, n$ .  $J$  je matice částečných derivací celého řetězce relativně ke koncovému efektořu.  $J$  mapuje změny proměnných kloubů  $\boldsymbol{\theta}$  na změny v pozici koncového efektořu. Sloupec  $i$  v  $J$  reprezentuje inkrementální změnu pozice koncového efektořu způsobenou inkrementální změnou proměnné  $\theta_i$ . Neznámou pro inverzní kinematiku je  $\dot{\boldsymbol{\theta}}$ , proto potřebujeme inverzi Jacobiho matice. Vztah je následující:

$$\dot{\boldsymbol{\theta}} = J^{-1}(\boldsymbol{\theta})\dot{\mathbf{P}} \quad (3.5)$$

Poté co určíme Jacobiho matici, hledáme hodnotu  $\Delta\boldsymbol{\theta}$  pro inkrementaci  $\boldsymbol{\theta}$ :

$$\boldsymbol{\theta} = \boldsymbol{\theta} + \Delta\boldsymbol{\theta} \quad (3.6)$$

Změna v pozici koncového efektořu určená touto změnou je:

$$\Delta\vec{s} \approx J\Delta\boldsymbol{\theta} \quad (3.7)$$

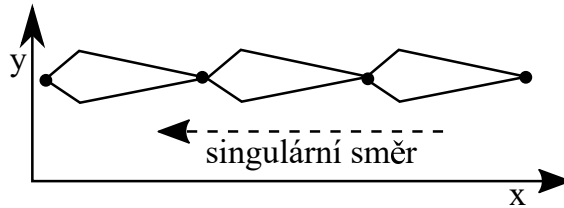
Hodnota  $\Delta\boldsymbol{\theta}$  by měla být vybrána tak, aby  $\Delta\vec{s}$  přibližně odpovídala  $\vec{e}$  - rozdílu mezi cílovou a aktuální pozicí koncového efektořu. Potom přímá kinematika může být přepsána jako  $\vec{e} = J\Delta\boldsymbol{\theta}$  a inverzní jako  $\Delta\boldsymbol{\theta} = J^{-1}\vec{e}$ .

Jacobiho matice ale není vždy invertibilní, tzn. není čtvercová a zároveň singulární. To nastane když konfigurace kloubů je redundantní nebo když prochází skrz nebo blízko singulární konfigurace.

Řetězec kloubů je redundantní jestliže obsahuje více stupňů volnosti, než je potřeba pro specifikaci cíle koncového efektořu. Je-li cíl koncového efektořu určen pozicí v 3D prostoru,



potom jakýkoliv řetězec obsahující více než 3 stupně volnosti je redundantní a tedy existuje nekonečně mnoho řešení. Jacobiho matice potom má více sloupců než řádků. V takových případech se  $J^{-1}$  nahradí generalizovanou inverzí  $J^\dagger$ , většinou *Moore-Penroseovou pseudoinverzí*. Jestliže existuje nekonečné množství řešení, je možné toho využít pro splnění nějakého sekundárního kriteria, např. vyhnutí se překážkám. Matice je singulární jestliže dva nebo více řádků jsou lineárně závislé, příklad takové konfigurace je vidět na obrázku 3.2. Jacobiho matice pro takovou konfiguraci bude obsahovat nuly v prvním řádku a nemůže být invertována. V takovém případě je možné použít pseudoinverzní matici, která poskytne řešení v singulární konfiguraci, nicméně v okolí singulární konfigurace bude nevhodně oscilovat.



Obrázek 3.2: Singulární konfigurace řetězce kloubů.

## 3.2 Cyclic Coordinate Descent

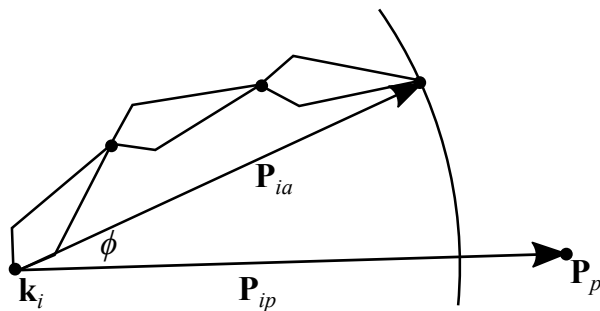
*Cyclic Coordinate Descent* (CCD) [19] je řešení inverzní kinematiky založené na iterativním heuristickém vyhledávání, které se snaží o minimalizaci rotační chyby zpracováváním jednoho kloubu v jeden okamžik. Každá iterace provede jednotlivě pohyb všech článků od nejposlednějšího k prvnímu. To se opakuje dokud se nedosáhne požadované pozice koncového efektoru a nebo dokud se nepřesáhne maximální počet cyklů. K určení úhlu rotace se používají dva vektory. Protože metoda pracuje s jedním kloubem v jeden okamžik, zvýhodňuje klouby na začátku. Oproti metodám používající Jacobiho matice provádí více iterací. Ačkoliv je algoritmus jednoduchý, je často nutné ho dále přizpůsobovat, aby bylo dosaženo vhodných výsledků.

Mějme iteraci  $i$  (viz obrázek 3.3) a pozici kloubu  $\mathbf{k}_i$  pro  $i$ -tý kloub od konce řetězce. Potom vektor  $\mathbf{P}_{ia}$  je vektor od pozice kloubu  $\mathbf{k}_i$  k aktuální pozici koncového efektoru a  $\mathbf{P}_{ip}$  je vektor od  $\mathbf{k}_i$  do požadované pozice koncového efektoru. Budeme-li rotovat vektor  $\mathbf{P}_{ia}$  o úhel  $\phi$ , dostaneme nový vektor  $\mathbf{P}'_{ia}(\phi)$ . Jestliže úhel bude nabývat libovolných hodnot,  $\mathbf{P}'_{ia}(\phi)$  začne tvořit kruh se středem  $\mathbf{k}_i$ . Bod na tomto kruhu nejbližší k požadované pozici  $\mathbf{P}_p$ , je bod, na kterém kruh protíná přímku určenou vektorem  $\mathbf{P}_{ip}$ . Proto budeme měnit rotaci kloubu  $\mathbf{k}_i$  tak, aby se  $\mathbf{P}_{ip}$  a  $\mathbf{P}'_{ia}(\phi)$  zarovnaly. Úhel  $\phi$  který hledáme, je úhel, který maximalizuje následující funkci:

$$g(\phi) = k_1(1 - \cos \phi) + k_2 \cos \phi + k_3 \sin \phi \quad (3.8)$$

kde  $k_1$ ,  $k_2$  a  $k_3$  jsou konstantní koeficienty, pro detaily viz [19]. Funkce má maximum v intervalu  $-\pi \leq \phi \leq \pi$ , když je první derivace nula a druhá derivace je záporná. Z první podmínky dostáváme rovnici:

$$\phi = \tan^{-1} \frac{k_3}{k_2 - k_1} \quad (3.9)$$



Obrázek 3.3: Ukázka jedné iterace metody CCD.

která vymezí kandidátní hodnotu  $\phi_c$  v rozsahu  $-\frac{\pi}{2} \leq \phi_c \leq \frac{\pi}{2}$ . Protože  $\tan$  je periodické, musíme uvažovat i dvě další kandidátní hodnoty:  $\phi_c + \pi$  a  $\phi_c - \pi$ . Z kandidátních hodnot se vybere taková, která spadá do intervalu  $-\pi \leq x \leq \pi$  a jejíž druhá derivace je záporná. Je-li jich více, funkce se vyhodnotí nad všemi a vybere se z nich maximum. Poté co jsme získali úhel  $\phi$ , můžeme ho vynásobit váhou určující tuhost kloubu a přičíst ho k aktuální rotaci  $r_i$  kloubu  $i$ . Poté se pokračuje s iterací  $i + 1$ .

## Kapitola 4

# Lidská chůze

Lidská chůze [6] se skládá z opakujících se *krokových cyklů* neboli *dvojkroků*, viz obrázek 4.1. Cyklus je sekvence pohybů od doby kdy se jedna noha dotkne země do doby kdy se stejná noha dotkne země znovu. Každý krokový cyklus má dvě základní fáze: *stojnou fázi* a *švihovou fázi*. Protože je cyklus shodný pro obě nohy, budeme popisovat pouze jednu.

Stojná fáze, je fáze, kdy je noha v kontaktu se zemí. Zabírá 60% dvojkroku a začíná s *úderem paty (počáteční kontakt)* a končí s *odrazem palce*. Je dále rozdělena do podfází:

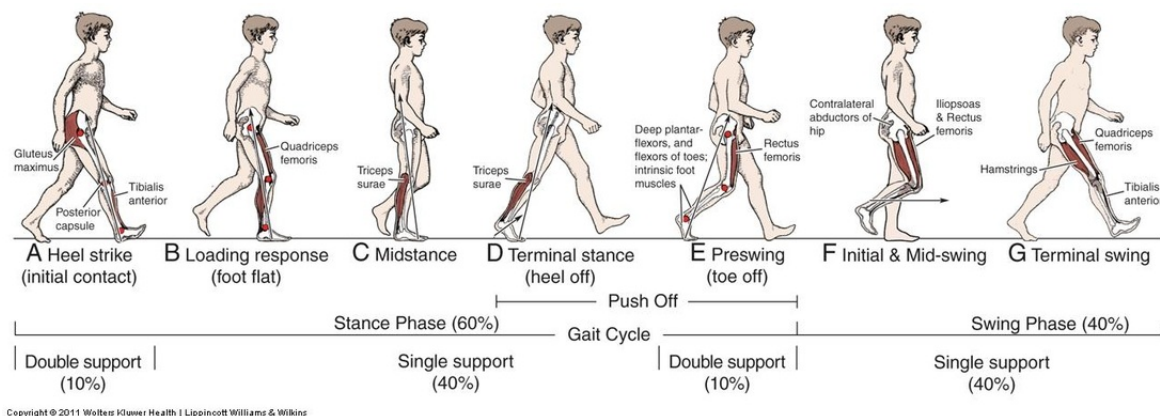
- *stádium zatěžování*
- *mezistoj*
- *konečný stoj*
- *předšvihová fáze*

Po počátečním kontaktu pravé nohy nastává perioda *dvojí opory*, která končí při odrazu palce levé nohy. Následuje *jedno-oporová fáze* do té doby, než se levá dostane do počátečního kontaktu. Nastává druhá perioda *dvojí opory*, až do odrazu palce pravé nohy. V každé fázi *dvojí opory* je jedna noha vpředu, právě se dotkla země. A druhá je vzadu připravena na zvednutí se ze země. Přední noha je ve stadiu *zatěžování* a zadní je v *konečném švih*u. V každém krokovém cyklu jsou dvě periody *dvojí opory* a dvě periody *jedno-oporové fáze*. *Dvojí opora* zabírá přibližně 10% dvojkroku, ale její délka se mění s rychlostí chůze. S větší rychlostí se prodlužuje *švihová fáze* a zkracuje *fáze dvojí opory*. Když *dvojí opora* úplně vymizí z chůze se stává běh. Mezi kroky běhu je *fáze letu*, kdy není žádná noha na zemi.

*Švihová fáze* nastává když je chodidlo ve vzduchu a trvá 40% délky dvojkroku. Začíná s *úderem paty* a končí s *druhým odrazem palce*. Je dále rozdělena do podfází:

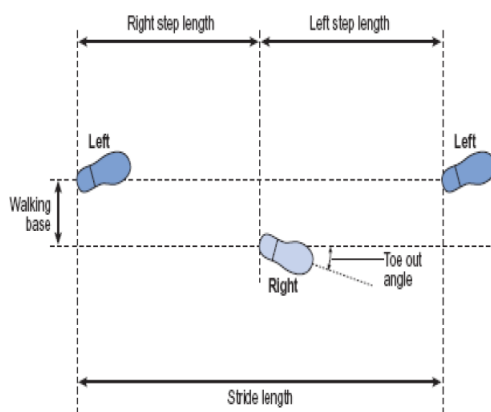
- *počáteční švih*
- *mezišvih*
- *konečný švih*

Nyní si popíšeme některé termíny a parametry specifikující chůzi (obr. 4.2). *Délka dvojkroku* je vzdálenost mezi dvěma po sobě jdoucími kroky stejné nohy. Skládá se ze dvou délek kroku, levé a pravé, každá s nich určuje vzdálenost o kterou se jedna noha přesune před druhou. Může být např. nulová, když se levá noha přesune na úroveň pravé a ne před ní. *Šířka dvojkroku* je vzdálenost mezi přímkami určenými středy pat obou nohou při chůzi.



Obrázek 4.1: Krokový cyklus a jeho fáze. Převzato z [10].

Úhel vytočení chodidla určuje úhel mezi zmíněnou přímkou a přímkou procházející středem chodidla. Kadence je počet kroků v daném čase. V jednom krokovém cyklu jsou dva kroky a tedy kadence je měřítko půl-cyklů. Čas dvojkroku  $c$  v sekundách se určí jako  $c = 120/kad$ , kde  $kad$  je kadence v krocích za minutu. Rychlost chůze je vzdálenost ušlá v čase. Okamžitá rychlost je variabilní v průběhu kroku, ale průměrná rychlost je výsledkem kadence a délky dvojkroku. Rychlost chůze  $r$  se spočítá jako  $r = d/c$ , kde  $d$  je délka dvojkroku. Rychlost chůze tedy závisí na délce kroku, které závisí na délce švihové fáze.



Obrázek 4.2: Délka, šířka kroku a úhel vytočení chodidla. Převzato z [6].

## 4.1 Fáze krokového cyklu

Tato sekce obsahuje podrobný popis jednotlivých fází krokového cyklu. Každá z osmi fází má funkční objektiv. Sekvenční kombinace fází umožňuje končetině dosáhnout tří základních úkolů. Těmi jsou přesun hmotnosti těla, jednooporová fáze a posun končetiny ve švihové fázi. Tato podkapitola byla převzata z [6].

#### **4.1.1 Počáteční kontakt**

V této fázi je kyčel ohnutá, koleno natažené a kotník v neutrální poloze. Kontakt se zemí je dosažen pomocí paty, která se stává středem otáčení. Druhá noha je na konci konečného stoje. Tato fáze obsahuje moment kdy právě nastane dotyk nohy se zemí. Držení těla v tuto chvíli určuje pohyb končetiny ve stadiu zatěžování.

#### **4.1.2 stádium zatěžování**

stádium zatěžování je fáze, kde je váha těla přenesena na přední nohu. Koleno je pokrčeno pro absorpci šoku. Chodidlo se dostává do plného kontaktu se zemí. Protilehlá noha je v předšvihové fázi. Tato fáze je počátkem fáze dvojí opory. Pokračuje dokud nezačne být druhá noha zvedána ke švihů.

#### **4.1.3 Mezistoj**

V mezistoji dochází k posunutí dolní končetiny přes zafixované chodidlo, zatímco koleno a kotník se propínají. Protilehlá noha se nachází uprostřed mezišvihové fáze. Je to první polovina intervalu stoje na pouze jedné noze. Trvá od zvednutí druhé nohy dokud váha není přesunuta po chodidle do oblasti přednoží.

#### **4.1.4 Konečný stoj**

Během druhé poloviny intervalu stoje na jedné noze se zvedne pata a střed otáčení stojné končetiny se přesune do přední části nohy. Koleno zvýší své propnutí a poté se začne lehce ohýbat, kyčel se propíná. Druhá noha je v konečném švihů. Začíná se zdvihem paty a končí v okamžiku kontaktu paty druhé nohy se zemí. Váha těla je přesunuta před nohu.

#### **4.1.5 Předšvihová fáze**

Kontakt se zemí druhé nohy začal další fázi dvojí opory. Referenční noha zvýší ohyb v kotníku a v kolenu, kyčel již není propnutá. Protilehlá noha je ve stadiu zatěžování. Tato poslední podfáze stojné fáze je druhým (a posledním) intervalem dvojí opory v krokovém cyklu. Začíná s počátečním kontaktem druhé nohy a končí s odrazem palce referenční nohy. Váha je uvolněna z referenční nohy a přenesena na nohu druhou. Uvolněná noha využije volnosti k přípravě na švihovou fázi, k čemuž slouží všechny pohyby a svalové akce odehrávající se v tuto dobu.

#### **4.1.6 Počáteční švih**

Chodidlo je zvednuto a přesunuto vpřed pomocí ohybu v kyčli a kolenu. Kotník částečně dorziflexuje (ohyb kotníku směrem k hřbetu nohy). Druhá končetina je na začátku mezišvihů. Tato fáze je přibližně třetina švihové periody. Začíná když noha opustí podložku a končí v okamžiku maximálního propnutí v kolenu.

#### **4.1.7 Mezišvih**

Přesun nohy dopředu je získán dalším ohybem v kyčli. Koleno se propíná a kotník dorziflexuje do neutrální polohy. Druhá noha je na konci mezistoj. Tato druhá fáze švihové periody začíná když je švihová noha naproti stojné noze. Končí když je holení kost ve vertikálním postavení - ohyb kolene a kyčle je shodný.

#### **4.1.8 Konečný švih**

Přesun chodidla je ukončen propnutím kolene a přesunem holeně před stehno. Kyčel si zachovává ohyb z předchozí fáze. Kotník zůstává v neutrální poloze. Druhá noha je v konečném stoji. Tato poslední fáze švihu začíná s vertikální holení kostí a končí při kontaktu nohy se zemí. Touto fází je přesun nohy ukončen.

## Kapitola 5

# Existující implementace

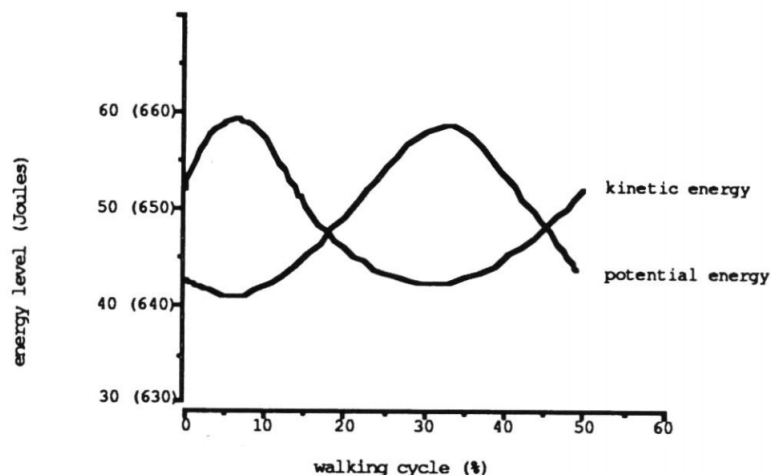
V této kapitole jsou popsány některé existující implementace procedurální animace lidské chůze. Je ukázáno jak se využívá znalostí o lidské chůzi a jak je možné animaci ovlivňovat pomocí parametrů.

### 5.1 Interactive Animation of Personalized Human Locomotion

V článku *Interactive Animation of Personalized Human Locomotion* [8] je popsán systém pro tvorbu lidského pohybu pomocí procedurální animace. Umožňuje specifikaci tří parametrů: délka kroku, kadence kroků a rychlost. A 15 atributů jako je např. rotace a náklon pánve. Díky těmto atributům je umožněna individualizace pohybu pro starého člověka, malé dítě apod. Tento systém je plně kinematický.

Pro každý krok jsou vypočítány tři omezení ze specifikovaných parametrů a atributů: doba trvání fází kroku (stojná a švihová), úhly nohy na konci kroku a kontrolní body určující pohyb kyčle stojné nohy během kroku. Určují se čtyři kontrolní body zvlášť pro vertikální a horizontální komponent. První a poslední bod se určí z délky kroku a úhlů nohy na začátku a konci kroku. Druhý a třetí kontrolní bod se určí na základě znalosti, že vertikální přesun je nejnižší uprostřed dvojí opory a nejvyšší uprostřed švihové fáze. Zatímco horizontální přesun má maximum a minimum opačně. To je v korelaci s výdajem energie během jednoho dvojkroku, jak je vidět na obrázku 5.1, kde potenciální energie je identická s vertikálním přesunem a kinetická s horizontálním. Mezi body z každé množiny jsou proloženy interpolační spline křivky a tím je určen pohyb kyčle pro stojnou nohu během kroku. Pozice prstů nohy je statická během stojné fáze. Úhly zbylých kloubů nohy jsou dopočítány pomocí inverzní kinematiky.

Tím je určen pohyb stojné nohy. Dále se určí rotace, úklon a náklon pánve. Rotace pánve je na maximum při úderu paty na zem a na minimum v mezistoji. Úklon pánve má minimum při úderu paty a maximum na konci dvojí opory. Náklon pánve je způsobem tím, že se tělo vždy nakloní na stranu nohy nesoucí váhu. Je minimální při úderu paty a maximální uprostřed stojné fáze. Pozice kyčle nohy ve švihu je určena interpolací těchto hodnot. Pohyb nohy ve švihu je rozdělen do tří fází a určen lineární interpolací. Pohyb horní poloviny těla je určen v závislosti na spodní, např. ruka se hýbe dopředu s protilehlou nohou. Pohyb ruky a rotace ramene je určen nastavitelnými atributy.



Obrázek 5.1: Aproximace kinetických a potenciálních energií horní poloviny těla pro průměrný krok. Převzato z [8].

## 5.2 Animation of Human Walking in Virtual Environments

Kontrolní systém pro animaci lidské chůze po nerovném terénu je prezentován v článku Animation of Human Walking in Virtual Environments [11]. Používá se model člověka s 18 klouby a celkem 36 stupni volnosti.

Kontrola pohybu na nejvyšší úrovni umožňuje uživateli animovat pohyby s malým množstvím parametrů. Uživatel specifikuje cestu pohybu pomocí kubických splinů v horizontální rovině. Z parametrů výšky člověka a rychlosti chůze je vygenerována délka a frekvence kroku. Pro normální chůzi může být vztah určen jako v rovnici:

$$l = \sqrt{0.004 \times s \times h} \quad (5.1)$$

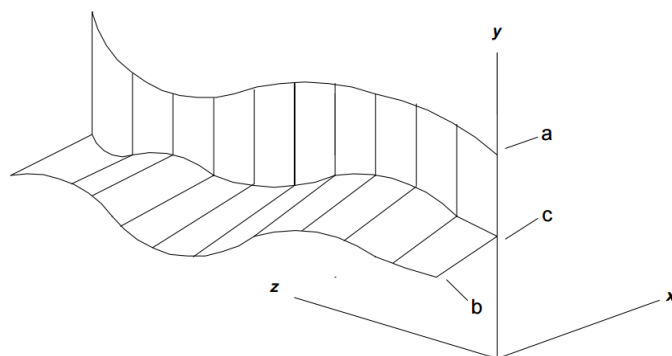
kde  $l$  je délka kroku,  $s$  je rychlost trupu a  $h$  je výška člověka. Výpočet pozice chodidla na terénu je určen jako funkce délky kroku, změny směru globální cesty a stavu terénu. Prvně jsou pozice chodidel rovnoměrně rozloženy po křivce cesty v horizontální rovině. Přesáhne-li rotace chodidla prahovou hodnotu, je krok zkrácen, ale doba provedení kroku zůstane zachována. Stav terénů ve 3D je zkontrolován pro určení bezkolizních trajektorií nohou.

Na střední úrovni kontroly jsou poskytnuty atributy pro kontrolu spodní části těla. Prvně se odhadnou úhly dopadu a zvednutí chodidla stojící nohy. Z těchto odhadnutých hodnot a podle přibližné trajektorie pánve jsou pomocí inverzí kinematiky dopočítány úhly kolene a kyčle. Poté jsou přesně dopočítány úhly dopadu a zvednutí chodidla. Trajektorie chodidla se určí lineární interpolací mezi položeným chodidlem na zemi a úhly dopadu a zvednutí chodidla. Tím je určena trajektorie stojící nohy, jednoho z koncových efektorů pro řetězec inverzní kinematiky. Nyní je potřeba určit pohyb kořene řetězce, pánve.

Observace ukázaly, že trajektorie pánve přibližně odpovídá sinusoidě, viz obrázek 5.2. Pro interpolaci jsou použity kubické spliny, konkrétně Beziérova křivka. Pánev prochází vertikálním maximem uprostřed stojné fáze, tato hodnota se určí podle velikosti ohybu kolene. Pánev prochází minimem uprostřed intervalu dvojí opory, kdy obě nohy jsou v kontaktu se zemí. Pozice v této fázi je určena tak, aby byla minimalizována suma úhlových zrychlení všech kloubů podírající nohy. Uživatel může těmto kloubům nastavovat váhy. Výsledná



Beziérová křivka je poté nerovnoměrně vzorkována, aby bylo dosaženo nejrychlejšího pohybu během dvojí opory a nejpomalejšího uprostřed stejné fáze. Pro definici trajektorie



Obrázek 5.2: 3D trajektorie (c) je určena z vertikálního (a) a horizontálního (b) posuvu. Převzato z [11].

chodidla nohy ve švihové fázi se použijí Beziérovky křivky obdobně jako při pohybu pánve. Spolu se znalostí pozice kořene (pánve) kinematického řetězce se dopočítají pozice ostatních kloubů. Trajektorie se určí tak, aby se chodidlo vyhnulo kolizím a aby byla spotřebována co nejmenší energie. Na nejnižší úrovni je poté možné nastavit další atributy, např. rotace a náklon pánve.

## Kapitola 6

# Návrh a realizace řešení

Řešení je implementováno v jazyce C++. 3D grafika je vykreslena pomocí rozhraní OpenGL. OpenGL je multiplatformní rozhraní, podporováno množstvím programovacích jazyků, umožňující vykreslování 2D a 3D grafiky. Komunikuje s grafickým procesorem pro hardware akcelerované vykreslování. Pro zpracování vstupů a vytvoření kontextu OpenGL je využita knihovna SDL 2.0. SDL je multiplatformní knihovna umožňující nízkourovňový přístup ke klávesnici, myši, zvuku a grafickému hardwaru. Funguje nativně v C++.

Aplikace bude umožňovat načtení předpřipraveného modelu s připojenou kostrou. Případně uživatel může využít poskytnutou předpřipravenou kostru a na ní připojit vlastní model a ten poté vyexportovat v Blenderu do formátu *COLLADA*. Aplikace model zobrazí za využití Linear blend skinning (kapitola 2.2.1) algoritmu. Uživatel bude mít možnost zobrazení pouze kůže modelu, nebo kostry, nebo obojího zároveň. Model bude rozhýbán animací chůze. Bude možné upravit animaci pomocí několika parametrů. K tomu bude využito grafické uživatelské rozhraní. Animace následně bude moci být vyexportována a následně použita například v herním enginu Unity.

### 6.1 Načtení modelu

Model je načítán z formátu COLLADA, který podporuje uložení geometrie, efektů, fyziky, animace, kinematik scény aj. Využívá XML schéma. Při načítání uvažujeme, že model do tohoto formátu byl exportován z Blenderu.

Začne se načtením geometrie modelu z XML elementu `library_geometries`. Uloží se pole s pozicemi vrcholů, normálami a texturovacími koordináty. Načtou se seznamy polygonů, které jsou určeny odkazy do polí načtených v předchozím kroku. Očekává se použití pouze trojúhelníků. Těchto seznamů může být více, liší se použitým materiálem a v naší aplikaci jeden tento seznam tvoří jeden *Mesh* (kód 6.1). Název materiálu je nastavený v Blenderu a při načítání modelu se očekává přiložení textury se shodným názvem a příponou `.D` pro difuzní texturu.

Následuje načtení vah a přiřazení kostí vrcholům z XML elementu `library_controllers`. První se přečte *bind\_shape* matice. Ta popisuje transformaci geometrie do správného souřadného systému pro použití s klouby. Poté je načten seznam názvů kostí, seznam inverzních *bind pose* matic  $\mathbf{B}_i$  (viz rovnice 2.1) pro každou kost a seznam vah vrcholů. Následuje seznam, který určuje pro každý vrchol počet kostí, které ho ovlivňují a stejný počet odkazů do seznamu kostí a vah.

Nyní se načte samotná hierarchie kostry z XML elementu `library_visual_scenes`.

```

class Mesh {
protected:
    std::vector<glm::vec3> v; //pozice vrcholu
    std::vector<glm::vec3> n; //normaly
    std::vector<glm::vec2> t; //texturovací koordinaty
    std::shared_ptr<Material> m; //material
public:
    /* ... */ };

class WeightedMesh : public Mesh {
private:
    std::vector<glm::vec4> weights; //vahy prirazenych kosti
    std::vector<glm::ivec4> jointIndices; //prirazene kosti
public:
    /* ... */ };

```

Kód 6.1: Třída modelu.

První uzel obsahuje kořenovou transformaci, od které se odvíjí další transformace všech kostí. Kosti jsou hierarchicky zanořené, obsahují název a lokální transformační matici, viz kód 6.3. Kosti se ukládají do 2D pole, kde každá kost má uložený index nadřazeného uzlu, tedy rodičovské kosti a  $0 \dots n$  indexů na své potomky, viz kód 6.2. Kostem se následně přiřadí inverzní bind pose matice vynásobené bind shape maticí, ta se dále neukládá. Pro správné zobrazení kostry je nutné určit velikost kostí, ta se určí jako rozdíl pozice dané kosti a jejího potomka. Pokud ale kost nemá žádného potomka nemůžeme zjistit její velikost. Proto uvažujeme poloviční velikost než je velikost rodičovské kosti. Dále jsme zavedli možnost přidat kost, která se použije pouze pro určení velikosti a poté se odstraní, taková kost musí ve svém názvu obsahovat *empty*.

Nakonec jsou naplněny objekty třídy `WeightedMesh` (kód 6.1). Abychom mohli použít `vec4` pro vstup kostí do vertex shaderu, musí mít každý vrchol přiřazené maximálně čtyři kosti. Pro model lidského těla jsou čtyři kosti dostatečné a pokud je jich více, mají některé velmi nízké váhy. Proto jsou pro každý vrchol seřazeny všechny jeho kosti sestupně podle vah a uloženy pouze první čtyři. Váhy ostatních jsou rozloženy rovnoměrně mezi ostatní, aby zůstala zachována podmínka o celkovém součtu všech vah (viz 2.2.1).

## 6.2 Zobrazení

Před každým zobrazením snímku je nutné spočítat globální transformační matice pro všechny kosti kostry. Proto procházíme pole s kostmi a pro každou kost vynásobíme globální matici rodiče s její lokální maticí. Poté získáme transformační matici  $\mathbf{T}$  podle rovnice 2.1. Následně nastavíme kořenovou transformační matici jako transformační matici modelu. Dále ve vertex shaderu získáme pozici vrcholu kůže násobením maticí  $\mathbf{T}$  a odpovídající vahou pro čtyři přiřazené kosti. Přepočítáváme také normálu, viz kód 6.4.

Následuje zobrazení kostry. Pro každou kost se použije shodný model. Pro kosti se používají globální matice shodné jako při zobrazování kůže, ale je v nich započítané zvětšení, aby na sebe kosti navazovaly. Výsledná transformační matice kosti je kořenová transformace vynásobená s touto globální maticí kosti. Před zobrazením je vyčištěn hloubkový buffer,

```

struct Bone {
    glm::mat4 localMat; //lokalni transformace
    glm::mat4 inverseBindMatrix; //inverzni bind pose matice
    int parent; //odkaz na rodicovskou kost
    std::vector<int> childs; //odkaz na potomky
    float scale; //velikost kosti, slouzi pro zobrazeni kosti
};

class Skeleton {
private:
    std::vector<Bone> bones; //pole kosti
    glm::mat4 rootTransform; //korenova transformacni matice
    std::vector<glm::mat4> globalMat; //globalni transformace
public:
    /* ... */ };

```

Kód 6.2: Třída kostry.

```

<library_visual_scenes>
<visual_scene id="Scene" name="Scene">
    <!-- ... -->
    <node id="metarig" name="metarig" type="NODE">
        <matrix sid="transform">1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1</matrix>
        <node id="hips" name="hips" sid="hips" type="JOINT">
            <matrix sid="transform">
                1 0 0 0 0 -0.16 -0.98 0.05 0 0.98 -0.167 1.01 0 0 0 1
            </matrix>
            <node id="spine" name="spine" sid="spine" type="JOINT">
                <matrix sid="transform">
                    1 0 0 0 0 0.99 0.11 0.17 0 -0.11 0.99 1.49e-8 0 0 0 1
                </matrix>
                <node id="chest" name="chest" sid="chest" type="JOINT">
                    <matrix sid="transform">
                        1 0 0 0 0 0.99 0.14 0.15 0 -0.14 0.99 7.45e-9 0 0 0 1
                    </matrix>
                    <!-- ... -->
                </node>
            </node>
        </node>
    </node>
    <!-- ... -->
</visual_scene>
</library_visual_scenes>

```

Kód 6.3: Ukázka uložení hierarchie kostry ve formátu COLLADA.

```

vec4 pos = vec4(0.0);
pos += T[v_joints.x] * vec4(v_pos, 1.0) * v_weights.x;
/* ... */
vec3 norm = vec3(0.0);
norm += ti_T[v_joints.x] * v_norm * v_weights.x;
/* ... */
gl_Position = mvp * vec4(pos.xyz, 1.0);

```

Kód 6.4: Ukázka z vertex shaderu pro animovaný model. Výpočet pro pozici a normálu se provede 4×, pro komponenty x, y, z a w vektorů `v_weights` a `v_joints`.

aby se kostra překreslila přes kůži. Pro zobrazení jsou použity difuzní textury a Phongův osvětlovací model. Výsledný výstup je na obrázku 6.1.



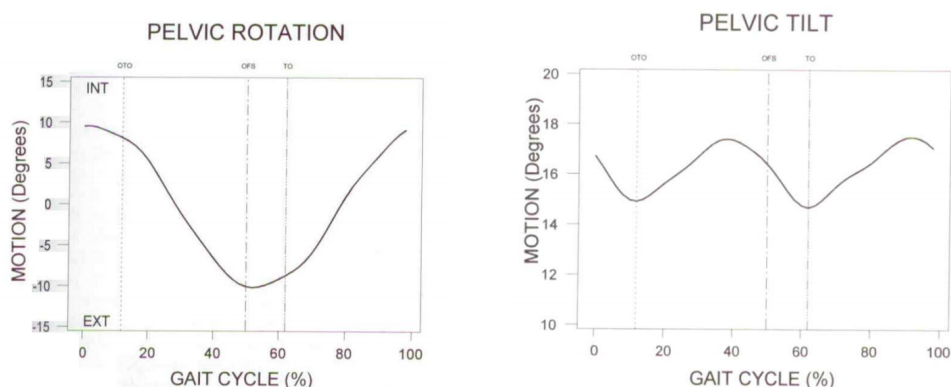
Obrázek 6.1: Grafický výstup aplikace pro dva různé modely. Model [17] vlevo je v bind pose a model [5] vpravo má rotované některé kosti (uvnitř kódu).

## 6.3 Animace

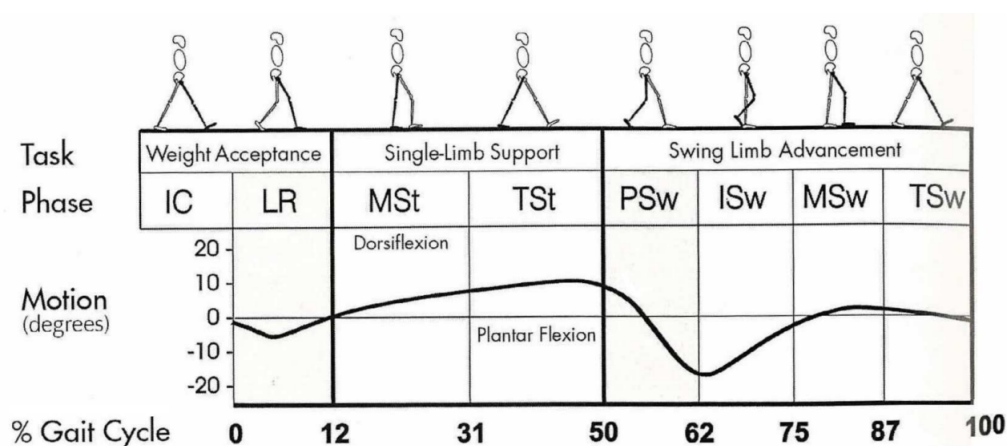
Popis animace je rozdělen na pohyb nohu, pánve a horní poloviny těla. Ve všech těchto částech je potřeba určovat rotace a trajektorie kloubů. K tomu slouží Bézierovy křivky. Dále jsou v této kapitole popsány možnosti nastavování parametrů.

Pohyb kořene kostry, pánve, bude aproximován pomocí interpolačního Catmull-Rom splinu, podobně jako na obrázku 5.2. Pro rotaci a náklon pánve budou využity grafy na obrázku 6.2. Pozice paty a prstů nohy se určí ze znalostí fází chůze. Během stadia zatěžování

je pata na zemi a chodidlo je zvednuté a postupně se dotýká země. V mezistoji bude chodidlo nehybné. Během konečného stoje a předšvihové fáze jsou prsty nehybně na zemi a zvedá se pata, až se nakonec zvednou i prsty. V následujících fázích se poloha chodidla přesouvá postupně do polohy, kterou bude mít v počátečním kontaktu. Rotace kotníku se aproximuje pomocí interpolačních splinů, viz obr 6.3. Rotace kyčle a kolene se dopočítá pomocí inverzní kinematiky, použije se metoda CCD (kapitola 3.2). Horní končetiny se budou hýbat opačně k dolním končetinám v rozsahu nastavitelným parametry.



Obrázek 6.2: Rotace a náklon pánve v průběhu krokového cyklu. Převzato z [1]



Obrázek 6.3: Kinematika hlezenního kloubu (kotníku). Převzato z [1].

## 6.4 Bézierovy křivky

Bézierovy křivky [13] jsou parametrické interpolační křivky. Ty nám umožní přesnou aproximaci s malým množstvím kontrolních bodů. Zvolili jsme kubické Bézierovy křivky, které používají čtyři kontrolní body, kde křivka prochází pouze krajními. Dva prostřední body určují tvar křivky. Segmenty těchto křivek je možné skládat za sebe. Jestliže chceme plynulou návaznost, je nutné zajistit aby třetí bod předchozího segmentu a druhý následujícího

byly kolineární. Máme-li parametr  $t$  potom se pozice  $\mathbf{B}$  na křivce určí následovně:

$$\mathbf{B}(t) = (1-t)^3\mathbf{P}_0 + 3(1-t)^2t\mathbf{P}_1 + 3(1-t)t^2\mathbf{P}_2 + t^3\mathbf{P}_3 \quad (6.1)$$

kde  $0 \leq t \leq 1$  a  $\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2$  a  $\mathbf{P}_3$  jsou kontrolní body křivky.

Křivky, které chceme aproximovat jsou ve 2D prostoru. Potřebujeme pro konkrétní hodnotu  $x$  získat odpovídající hodnotu  $y$ . Bézierovy křivky ale vrací hodnotu podle parametru  $t$ , který se nepohybuje po křivce lineárně. Proto musíme pro naše  $x$  určit odpovídající  $t$  a to poté použít pro získání  $y$ . To je možné řešit tak, že se určí větší množství bodů na křivce (např. 100) a pro každý se získá hodnota  $x$ . Když je potom potřeba určit  $t$  pro nějaké  $x$  najde se vhodný interval a  $t$  se určí lineární interpolací. Protože se ale naše křivky budou měnit s nastavováním parametrů a nerovností terénu, toto řešení není vhodné, protože vyžaduje při každé změně přepočítat všechny hodnoty. Proto jsme zvolili řešení numerickou metodou.

Jako metodu jsme zvolili metodu pro řešení jedné nelineární rovnice Regula Falsi. Ta konverguje vždy a v našem případě kdy řeší hladké a monotónní křivky i velmi rychle. Funguje na principu dělení intervalu  $\langle a, b \rangle$  na dvě obecně různé části. Dělicím bodem  $x_k$  intervalu je průsečík osy  $x$  s úsečkou spojující  $[a, f(a)]$  a  $[b, f(b)]$  určený následovně:

$$x_k = b_k - \frac{b_k - a_k}{f(b_k) - f(a_k)} f(b_k) \quad (6.2)$$

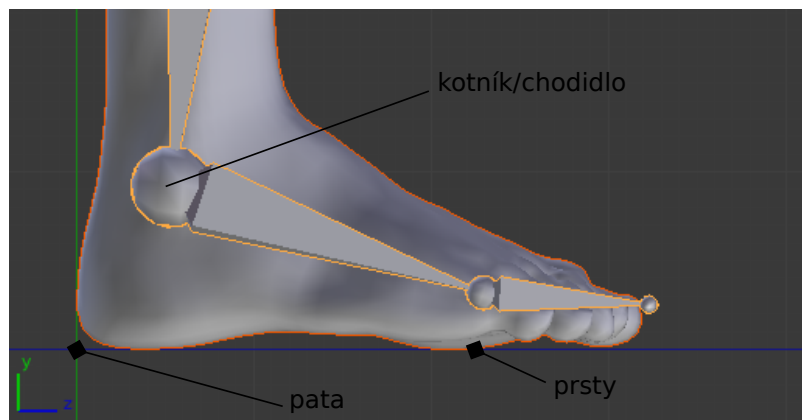
Výpočet končí jestliže  $|x_k - x_{k-1}| < \epsilon$ . Tato metoda najde v našem případě odpovídající  $t$  s dostatečnou přesností na cca 4 iterace.

## 6.5 Pohyb nohou

O pohyb nohou se stará třída **Leg** jejíž hlavní metodou, která se volá každý snímek poté co je již pohnutá pánev je metoda **update**. Ta využívá jediný parametr a tím je čas, o který se animace posunula. Jeden krokový cyklus nohy je nastaven na dobu trvání jedné vteřiny. Zrychlení nebo zpomalení kroku je dosaženo úpravou vstupního času. Tento vstupní čas je poté přičten k proměnné  $t$ , která se při přetečení přes 1 vteřinu vrací o vteřinu zpět. Podle  $t$  se poté určí v jaké fázi krokového cyklu se noha nachází a podle toho je určen další postup.

Nemusíme rozlišovat všech sedm fází cyklu, ale budou nám stačit čtyři. Stádium zatěžování a mezistoj jsou nezměněny. Konečný stoj a předšvihová fáze jsou spojeny do jedné, protože z pohledu animace se tyto fáze neliší, v obou je prováděn stejný úkon – zvedání paty. Poslední tři fáze (počáteční švih, mezišvih a konečný švih) jsou také spojeny do jedné, budeme ji říkat pouze švihová. Tyto fáze je možné spojit, protože se celá švihová fáze dá popsat spojitě pomocí trajektorie od odrazu palce do počátečního kontaktu.

Pro potřeby správného a přesného určování pohybu nohou potřebujeme znát pozici paty a prstů nohy. Přesně se tyto pozice nedají získat pouze z pozic kostí. Tyto pozice jsou závislé na pozici a rotaci chodidla. Potřebné informace o jejich pozici jsou získány z umístění kostry před samotným provedením animace podle umístění kostry v prostoru, viz obrázky ???. Když jsou získány paty a prstů nohy, jsou vynásobeny inverzní transformační maticí chodidla. Tím jsou získány relativní pozice vůči chodidlu, z kterých je později možné získat přesně pozici paty a prstů v závislosti na aktuální pozici chodila. Stačí vynásobit aktuální transformační maticí chodidla a těchto relativních pozic.



Obrázek 6.4: Vstupní pozice modelu a jeho kostry. Pozice paty se nachází na ose **z** a **y**. Pozice prstů nohy se nachází pod kloubem prstů nohy na ose **z**.

Důležitou podmínkou pro chůzi po nerovném terénu je možnost získat informace o terénu na kterém se noha nachází. Terén je objekt, který je sdílený všemi součástmi těla, kteří s ním potřebují pracovat. Pro nohu je důležité získat výšku terénu pod patou nohy a úhel mezi chodidlem a terénem. Tento úhel se určí pomocí skalárního součinu mezi vektorem směřujícím od paty k prstům nohy a vektorem mezi pozicemi na terénu pod patou a pod prsty.

### 6.5.1 Stádium zatěžování

Stádium zatěžování nastane pro  $t < 0.12$ . Před prvním vykonáním této fáze se dokončí švihová fáze. Protože předchozí snímek se vykonával pro nějaké  $t \leq 1.0$  a v aktuálním snímku se přesunul z švihové fáze do stadia zatěžování s nějakým  $t \geq 0.0$  je potřeba prvně dokončit předchozí fázi jako by čas byl roven 1. Kdybychom to neudělali animace by nebyla tolik přesná a plynulá, jak moc by záviselo na velikosti časového kroku. Také je získán počáteční úhel mezi chodidlem a terénem.

Poté se tato fáze zabývá zachováním pozice paty na terénu a sklápěním chodidla. Pozice a natočení chodidla je závislé na pozici pánve. Pánev se v každém snímku pohne první, proto při řešení nohy je noha na jiné pozici než byla v předchozím snímku. Pozice paty je závislá na rotaci chodidla. Rotace chodidla se v této fázi určuje lineární interpolací mezi počátečním úhlem s terénem a nulovým úhlem. Poté co se určí požadovaná rotace pro aktuální  $t$  je chodidlo do této pozice nastaveno. Nyní se může určit rozdíl mezi aktuální pozicí paty a předchozí a z toho určit požadovaná pozice kotníku, která je dosažena pomocí inverzní kinematiky. Inverzní kinematika hýbe nohou a tím i mění rotaci chodidla vůči terénu. Proto je po jejím provedení potřeba znovu natočit chodidlo do správného úhlu.

### 6.5.2 Mezistoj

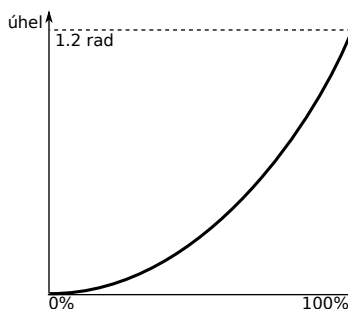
Následující fáze mezistoj nastává pro  $0.1 \leq t < 0.31$  a má za úkol udržet chodidlo položeno na zemi na stejné pozici. Je vykonávána podobně jako předchozí fáze. Před prvním vykonáním je tedy prvně dokončeno stádium zatěžování. Rozdílem je, že pohyb nevychází z paty, ale vychází z kotníku. Proto nám stačí si na začátku zapamatovat pozici kotníku a tu použít pro inverzní kinematiku. Není tedy před jejím provedením nutné nastavovat novou rotaci



chodidla, ta je nastavena až po provedení inverzní kinematiky tak, aby úhel mezi terénem byl nulový.

### 6.5.3 Zvedání paty

Zvedání paty nastává pro  $0.31 \leq t < 0.62$ . Pro čas do  $t = 0.5$  při zvedání paty zůstává koleno převážně propnuté, ve zbylém čase se zase koleno ohýbá. Aby tohoto bylo dosaženo není vhodné zvedat patu v celé fázi lineárně. Také není vhodné ji rozdělit na dvě samostatné části, zvedání paty by nepůsobilo plynule. Nejvhodnějším řešením se zdá použití Bézierovy křivky, kde v první části se zvedá pata méně a v druhé více, tak je celá fáze plynulá. Konkrétní tvar křivky (obr. ??) byl určen experimentálně tak aby tato fáze bylo dostatečně přirozená. Také je u této křivky důležitý sklon u konce fáze, který při vhodném zvolení zajistí co nejplynulejší přechod do švihové fáze, tzn. je důležité aby rychlost pohybu paty na konci této fáze a na začátku švihové byly co nejpodobnější. Rychlost pohybu nohy ve švihové fázi je závislá na délce kroku. Proto je celkový úhel o který je pata zvednutá zvětšován se zvětšující se délkou kroku. Mimo správnou návaznost je toto zvětšování také nutné pro zvýšení efektivního dosahu nohy, aby vůbec delší krok byl umožněn. Křivka úhlu zvedání paty je vytvořena pro nejdelší možný dvojkrok, který je 1.8 metru dlouhý. Pro jiné délky kroku je vynásobena  $c = (\text{dvojkrok}/1.8)^2$ , tato hodnota byla také zjištěna experimentálně, aby tato fáze vypadala přirozeně.



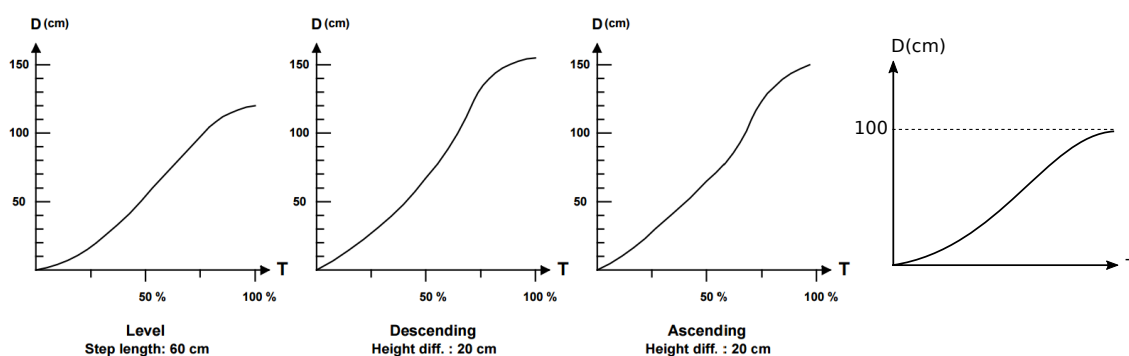
Obrázek 6.5: Rotace chodidla vůči zemi při fázi zvedání paty pro délku dvojkroku 1.8 metru.

Samotné zvedání paty probíhá následovně. Na začátku fáze se uloží pozice prstů, která se bude udržovat. Po zjištění aktuální rotace chodidla se zemí se natočí chodidlo do této polohy. Poté se pomocí inverzní kinematiky posune kotník o rozdíl původní a aktuální pozice prstů. Tím se kotník dostane do požadované polohy, ale prsty v původní pozici nebudou, protože inverzní kinematika změnila rotaci chodidla vůči zemi a proto je chodidlo znovu natočeno na správný úhel. Nakonec zbývá jenom natočit prsty nohy, aby byly v nulovém úhlu se zemí, tzn. natočit na rotaci odpovídající úhlu mezi chodidlem a zemí.

### 6.5.4 Švihová fáze

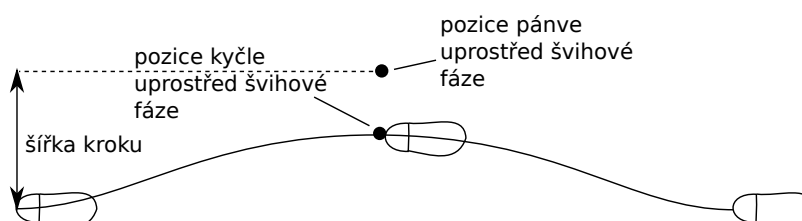
Švihová fáze nastává pro  $0.62 \leq t \leq 1.0$ . Jejím cílem je přesunutí nohy z aktuální pozice na pozici dalšího kroku. V této fázi se využívají čtyři Bézierovy křivky. Při prvním vykonání této fáze se dokončí fáze zvedání paty a zapamatuje se pozice paty, která bude sloužit jako výchozí bod pro švihovou fázi. V průběhu této fáze je také potřeba vrátit prsty na noze do původní polohy, protože při zvedání paty byly otočeny. V první pětině švihové fáze jsou prsty rotovány do pozice určené lineární interpolací mezi otočenou pozicí a původní nataženou pozicí.

Podle [11] se chodidlo pohybuje vpřed s měnící se rychlostí. Rychlost je nejvyšší uprostřed švihové fáze a nejpomalejší na začátku a na konci. K tomu slouží Bézierova křivka určující posun chodidla v horizontální rovině, viz obrázek 6.6. Tato křivka je vytvořena pro posun o délce 1 metr. Potom pro různé délky stačí před použitím křivky její kontrolní body vynásobit požadovanou délkou. Jako délku nemůžeme použít aktuální délku dvojkroku, to by se noha přesouvala před požadovanou pozici. Je nutné vzít v úvahu, že při zvedání paty se již pata posunuje vpřed. Noha zná pozici paty na začátku aktuálního kroku a pozici na kterou chceme aby pata dopadla při konci kroku, tedy začátku dalšího. Pata zůstává na stejné pozici při stádiu zatěžování a při mezistoji. Ale při zvedání paty se již pohybuje v horizontální rovině a délka tohoto pohybu se musí odečíst od délky dvojkroku, abychom získali přesný zbývající posun nohy na cílovou pozici pro vynásobení Bézierovy křivky. Pro posun vpřed v horizontální rovině poté stačí zjistit hodnotu křivky a přičíst ji k pozici po ukončení zvedání paty.



Obrázek 6.6: Vlevo jsou naměřené grafy posunu chodidla v čase pro chůzi po rovině, z kopce a do kopce převzté z článku [11]. Vpravo je vytvořená generická Bézierova křivka, která je použita pro posun nohy v horizontální rovině.

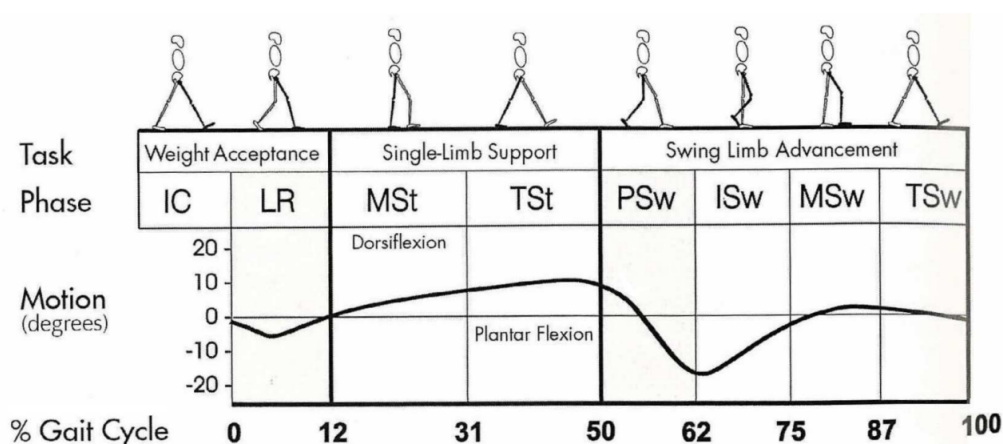
V průběhu chůze může být nastavena různá šířka kroku. Není přirozené, aby si chodidlo v průběhu celé švihové fáze udržovalo stejnou vzdálenost od šířky od pánve. Působí mnohem věrohodněji pokud se chodidlo po odrazu od země přesune na šířku kyčle, které dosáhne v polovině švihové fáze a poté se začne přesouvat zpět na šířku kroku, které dosáhne v okamžiku dopadu paty. Toho by bylo možné dosáhnout lineární interpolací mezi šířkou kroku a pozicí kyčle. Tímto způsobem ale animace není plynulá kvůli náhlému zrychlení nohy při odrazu paty a zastavení při dopadu. Proto jsme vytvořili další Bézierovu křivku, která se použije pro plynulé zrychlení a zpomalení tohoto pohybu, viz obrázek 6.7. Časový parametr lineární interpolace se použije pro získání hodnoty této křivky, která se použije místo něj.



Obrázek 6.7: Přesun chodidla během švihové fáze v závislosti na šířce kroku.

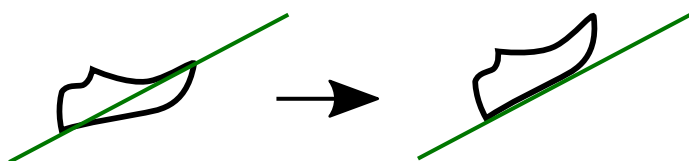
Takto jsou získány dva údaje ze čtyř, které jsou potřeba pro určení pozice chodidla ve švihové fázi. Máme pozici paty do šířky, vpřed a dále potřebujeme výšku a rotaci chodidla. Když máme všechny tyto čtyři údaje můžeme nastavit správnou rotaci v kotníku, poté z ní určit novou pozici kotníku tak, aby pata byla na správné pozici podle naší určené šířky, výšky a posunu vpřed. S touto novou pozicí kotníku se spočítá inverzní kinematika. Zbylé dva údaje rotace a výšky chodidla je ale nutné přizpůsobit nerovnému terénu.

Pro rotaci chodidla v průběhu švihové fáze vycházíme z obrázku 6.8, kde je uvedena rotace v kotníku v průběhu celého cyklu. Část toho grafu ve švihové fázi je aproximována pomocí Bézierovy křivky. Tato křivka začíná pro 15 stupňů plantární flexe, tedy propnutí kotníku z neutrální polohy, a končí u přibližně neutrální polohy. Tuto křivku je tedy nutné přizpůsobit propnutí kotníku na začátku švihové fáze, které se může lišit v závislosti na délce kroku a sklonu terénu. Kontrolní body křivky jsou proto vynásobeny koeficientem  $c = aktRot/15^\circ$ , kde  $aktRot$  je aktuální propnutí v kotníku.



Obrázek 6.8: Rotace kotníku v průběhu krokového cyklu. Převzato z [1].

Nyní je potřeba vyřešit úhel dopadu chodidla na terén. Jedna možnost je mít pevně daný úhel, pod kterým bude chodidlo dopadat vždy. Zdá se ale přirozenější nechat úhel podle naší křivky a upravit ho jenom v nevhodných případech. Tímto případem je možnost průchodu chodidla skrz terén při sklonu do kopce, viz obrázek 6.9. Poté je rotace upravena aby při dopadu byl sklon chodidla shodný s terénem.

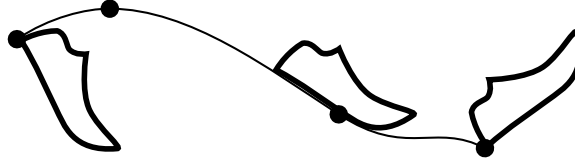


Obrázek 6.9: Pozice chodidla na terénu při dopadu paty s původní rotací a po úpravě rotace.

Kvůli tomu je před samotným provedením kroku vyzkoušena pozice na konci kroku při dopadu paty. To vyžaduje určení transformační rotace pánve při dopadu paty a posunutí nohy na cílovou pozici pomocí inverzní kinematiky. Poté je zkontrolován úhel mezi chodidlem a terénem a jestliže je záporný, tzn. chodidlo se nachází pod terénem je upravena křivka rotace. Potřebujeme upravit pouze koncový úhel propnutí kotníku a proto jsou o potřebný úhel posunuty poslední dva kontrolní body křivky.

Poslední co je potřeba určit je výška chodidla. K tomu slouží poslední Bézierova křivka

použitá pro pohyb nohy, viz obrázek 6.10. Tato křivka určuje výšku paty a skládá se ze tří segmentů, to nám umožňuje křivku dobře tvarovat při přizpůsobování se nerovnému terénu. Obdobně jako při rotaci chodidla je tato křivka vytvořena pro jeden případ a poté upravena. Konkrétně je vytvořena pro patu nacházející se 20 centimetrů nad terénem při počátku švihové fáze. Pro případy kdy se pata nachází výše nebo níže jsou všechny kontrolní body vynásobeny koeficientem, který zajistí aby výška paty odpovídala a švihová fáze plynule navazovala na zvedání paty.



Obrázek 6.10: Trajektorie paty při švihové fázi pro chůzi po rovném terénu.

### 6.5.5 Inverzní kinematika

Jak je nastaven ref. vektor.-

Kinematický řetězec který potřebujeme řešit obsahuje pouze dva klouby – kyčel a koleno. Pro takový řetězec existuje rychlé analytické řešení. Pozici kyčle známe a požadovanou cílovou pozici koncového efektoru (kotníku) také. Kromě toho známe i délku stehenní a holenní kosti. Rotace kyčle a kolene se poté dopočítá jako úhly v trojúhelníku pomocí kosinové věty.

Tímto ještě není inverzní kinematika vyřešená, protože může být nekonečně mnoho řešení s různými rotacemi okolo osy mezi kyčlí a kotníkem, viz obrázek 6.11. K tomu abychom určili výslednou rotaci okolo této osy slouží referenční vektor [4]. Výsledné řešení se poté vybere takové, aby úhel  $\delta$  mezi referenčním vektorem a stehenní kostí byl co nejmenší. Proto budeme rotovat vektor  $\mathbf{v}$  o úhel  $\theta$  do nového vektoru  $\mathbf{v}'$ . Minimalizace úhlu  $\delta$  je ekvivalentní maximalizaci  $\cos \delta$  (resp.  $\mathbf{v}' \cdot \mathbf{k}$ ) a to nastane když je první derivace rovna nule:

$$\frac{d}{d\theta} \mathbf{v}' \cdot \mathbf{k} = 0 \quad (6.3)$$

Po úpravě dostaneme:

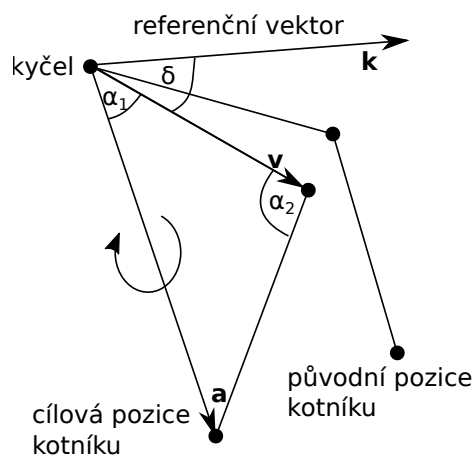
$$\theta = \tan^{-1} \frac{2B}{C-A} \quad \vee \quad \theta = \pi + \tan^{-1} \frac{2B}{C-A} \quad (6.4)$$

kde  $A = (\mathbf{a} \cdot \mathbf{v})\mathbf{a} \cdot \mathbf{k} - ((\mathbf{a} \times \mathbf{v}) \times \mathbf{a}) \cdot \mathbf{k}$ ,  $B = (\mathbf{a} \times \mathbf{v}) \cdot \mathbf{k}$  a  $C = \mathbf{v} \cdot \mathbf{k}$ . Abychom určili, které ze dvou řešení maximalizuje  $\cos \delta$  musíme určit druhou derivaci  $\cos \delta$ :

$$\frac{d^2}{d\theta^2} \cos \delta = \frac{A-C}{2} \cos \theta - B \sin \theta \quad (6.5)$$

Úhel  $\delta$  je minimální, když je druhá derivace menší než nula.

Pomocí kosinové věty jsme pro stehenní kost určili úhel  $\alpha_1$ , který určuje rotaci mezi vektorem  $\mathbf{a}$  a stehenní kostí  $\mathbf{v}$ . Abychom nastavili kost do správné pozice, je potřeba vektor  $\mathbf{v}$  natočit do  $\mathbf{a}$ . K tomu je sestavena transformační matice, jejímž vynásobením se  $\mathbf{v}$  dostane do  $\mathbf{a}$ . Poté je již možné rotovat kyčel o úhel  $\alpha_1$ . Nyní už je pouze rotována holeně do úhlu  $\alpha_2$  a stehno rotováno o  $\theta$  okolo vektoru  $\mathbf{a}$ .



Obrázek 6.11: Pro dosažení cílové pozice v kotníku se pomocí kosinové věty určí úhly v kyčli a v koleni. Rotace okolo osy mezi kyčlí a kotníkem se určí tak, aby úhel  $\delta$  byl co nejmenší.

Řešení inverzní kinematiky rotuje holeň okolo osy  $x$  a tedy aby výsledná pozice kotníku odpovídala té požadované, noha nemůže být libovolně natočena vůči své rodičovské kosti (stehna). Vstupní model může mít koleno různě natočené a proto před samotným začátkem animace je nutné lokální matici holeně zbavit všech rotací a zachovat pouze translaci.

## Kapitola 7

# Závěr

Byla představena skeletální animace a metody tvorby animace. Definovány kinematiky a popsány některé metody řešení inverzní kinematiky. Dále byly popsány fáze lidské chůze a některá existující řešení procedurální animace chůze. Následně byl popsán návrh a implementace částí řešení.

V rámci semestrálního projektu bylo implementováno načtení modelu člověka spolu s připojenou kostrou a jeho zobrazení. Výsledná aplikace má přes 2000 řádků kódu. V rámci dalšího vývoje bude potřeba implementovat systém tvorby animace a její vyexportování do souboru.

# Literatura

- [1] Biomechanika chđž"ze. [online].  
URL [http://ftk.upol.cz/fileadmin/user\\_upload/FTK-katedry/biomechanika/BIAP\\_Chuze.pdf](http://ftk.upol.cz/fileadmin/user_upload/FTK-katedry/biomechanika/BIAP_Chuze.pdf)
- [2] Blender Wiki: Skinning to Shapes. [online].  
URL <http://wiki.blender.org/index.php/Doc:2.4/Manual/Rigging/Skinning/ObData>
- [3] Vertex Skinning. [online].  
URL <http://ruh.li/AnimationVertexSkinning.html>
- [4] Inverse Kinematics (two joints) for foot placement. [online], 2012.  
URL [http://www.gamasutra.com/view/news/129168/Inverse\\_Kinematics\\_two\\_joints\\_for\\_foot\\_placement.php](http://www.gamasutra.com/view/news/129168/Inverse_Kinematics_two_joints_for_foot_placement.php)
- [5] Venom joker rigged. [online], 2012.  
URL <http://tf3dm.com/3d-model/venom-joker-rigged-36682.html>
- [6] A. Kharb, Y. K. J., V. Saini; Dhiman, S.: A review of gait cycle and its parameters. *Int. J. Comput. Eng. Manag.*, ročník 13, 2011: s. 78–83.
- [7] Aristidou, A.; Lasenby, J.: Inverse Kinematics: a review of existing techniques and introduction of a new fast iterative solver. *Technická Zpráva CUEDF-INFENG*, TR-632, Department of Information Engineering, University of Cambridge, September 2009.
- [8] Bruderlin, A.; Calvert, T.: Interactive animation of personalized human locomotion. In *Proceedings of Graphics Interface '93*, Canadian Information Processing Society, Toronto, Ontario, Canada: Canadian Information Processing Society, 1993, str. 17–23.
- [9] Buss, S. R.: Introduction to inverse kinematics with Jacobian transpose, pseudoinverse and damped least squares methods. 2009.
- [10] Cain, A.: Physiopedia: Gait in prosthetic rehabilitation. [online].  
URL [http://www.physio-pedia.com/Gait\\_in\\_prosthetic\\_rehabilitation](http://www.physio-pedia.com/Gait_in_prosthetic_rehabilitation)
- [11] Chung, S.-k.; Hahn, J. K.: Animation of Human Walking in Virtual Environments. In *Proceedings of the Computer Animation*, CA '99, Washington, DC, USA: IEEE Computer Society, 1999, ISBN 0-7695-0167-2, s. 4–.
- [12] Jacobson, A.; Deng, Z.; Kavan, L.; aj.: Skinning: Real-time Shape Deformation. In *ACM SIGGRAPH 2014 Courses*, 2014.

- [13] Kamermans, M.: A Primer on BĀŠzier Curves. [online], 2011-2016.  
URL <http://pomax.github.io/bezierinfo/>
- [14] Kavan, L.; Collins, S.; Zara, J.; aj.: Geometric Skinning with Approximate Dual Quaternion Blending. *ACM Trans. Graph.*, ročník 27, č. 4, 2008: str. 105.
- [15] Kirk, A.; O'Brien, J.; Forsyth, D.: Skeletal parameter estimation from optical motion capture data. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, ročník 2, June 2005, ISSN 1063-6919, s. 1185 vol. 2–, doi:10.1109/CVPR.2005.327.
- [16] Multon, F.; France, L.; Cani-Gascuel, M.-P.; aj.: Computer Animation of Human Walking: a Survey. 1999.
- [17] Paul Chen: Male Base Mesh. [online], 2015.  
URL <http://tf3dm.com/3d-model/male-base-mesh-6682.html>
- [18] Van Welbergen, H.; Van Basten, B. J. H.; Egges, A.; aj.: Real Time Animation of Virtual Humans: A Trade-off Between Naturalness and Control. *Computer Graphics Forum*, ročník 29, č. 8, 2010: s. 2530–2554, ISSN 1467-8659, doi:10.1111/j.1467-8659.2010.01822.x.
- [19] Welman, C.: *Inverse kinematics and geometric constraints for articulated figure manipulation*. Diplomová práce, Simon Fraser University, 1993.