

COLLISION STATES DETECTION FOR QUADRUPED ROBOT MOTION

J. Chmelíček, R. Grepl, M. Bezdíček, M. Švehlák*

Summary: *Detection of collision states among single parts of robot structure is essential problem, that is necessary to solve at robot movement in virtual space, in mathematical models, based on substitution of real structures by rigid bodies and above all at robot safe movement. Last introduced case is important especially at robot driving, where collision detection algorithm serve as backward verification before moment, when robot take newly computed position. Detection algorithm in this case hold important function, which is implemented in robot driving system. This function than protect robots construction against damage, overloading of actuators and stability of robot movement. Goal of this article is description of some solutions of this problem, including characterization of our solution.*

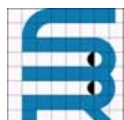
1. Úvod

V rámci řešení výzkumných projektů Laboratoře mechatroniky a robotiky, se dlouhodobě zabýváme stavbou čtyřnohého krácejícího robotu. Pro možnost rychlého ověření některých simulačních výpočtů a řídicích algoritmů, jsme přikročili ke stavbě malého experimentálního robotu. Další popis je možno nalézt v příspěvcích [1,3,9] na této konferenci.

Problematikou nalezení kolizí mezi hmotnými tělesy v prostoru, resp. geometrickými tělesy ve virtuálním prostředí a řešením adekvátních reakcí na tyto kolize, jest doménou, která se dostává do oblasti zájmu především z pohledu fyzikálního modelování soustav tuhých těles. Mezi prvními obory, kde byla problematika detekce kolizí ve virtuálním prostoru řešena, byl herní průmysl s počítačovými hrami. Zároveň se tento problém řeší při simulacích ve virtuální realitě, kde je nutné co nejlépe napodobit chování virtuálních objektů ve srovnání s reálnými. Zvláště díky těmto dvěma oblastem, byla vyvinuta řada algoritmů a postupů, založených na analytické geometrii, kterými se lze inspirovat při řešení kolizí končetin čtyřnohého robotu.

2. Formulace problému a cíle řešení

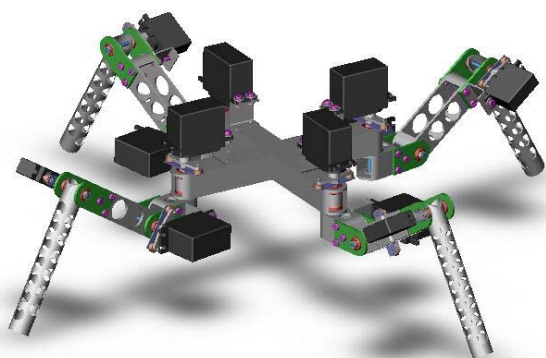
Při realizaci pohybu čtyřnohého robotu obr.1, se třemi stupni volnosti jednotlivých končetin



Ing. Jaroslav Chmelíček, Ing. Robert Grepl, Ing. Milan Bezdíček, Michal Švehlák,
Laboratoř mechatroniky a robotiky, ÚT AV ČR a ÚTMB FSI VUT v Brně,
Technická 2, 616 69, Brno, e-mail: jchmelicek@seznam.cz

vůči tělu, musí být řídicí algoritmus schopen detekovat takové stavy, kdy dochází ke kolizím mezi končetinami, popřípadě kolizí mezi tělem a končetinou, nebo končetinou a okolním terénem. Při realizaci pohybu bez tohoto kontrolního mechanismu, by mohlo dojít k následujícím provozním poruchám. Především se jedná o mechanické poškození struktury robotu v důsledku překročení povoleného momentu vyvíjeného akčním členem. Přetížením akčního členu s následkem jeho přílišného zahřátí. A v neposlední řadě také ztrátou stability celého robotu.

Daný problém lze vyřešit přidáním vhodných senzorů na povrch robotu. Toto řešení však zvyšuje nároky na hardwarové vybavení robotu, a s ním rostou i finanční požadavky na jeho realizaci. V tomto případě, je možné detekovat přesnou polohu kolize v závislosti na hustotě senzorů rozložených po povrchu, přičemž kolize je takto detekovatelná většinou až v okamžiku, kdy nastane. Výhodnější je tedy ušetřit za hardwarové vybavení a celý problém řešit vhodným výpočetním algoritmem, který vychází pouze na znalosti úhlů natočení jednotlivých kloubů a geometrie robotu. Přesnost detekce je v tomto případě závislá na přesnosti geometrických dat povrchu robotu a zvoleném algoritmu, jehož korektnost je obvykle volena s ohledem na výpočetní výkon procesoru. Výhodou takto zvoleného řešení, je potom možnost ovlivňovat přesnost detekce pouhou změnou vstupních parametrů programu. Navíc je toto řešení použitelné i v případě simulací pohybu robotu ve virtuálním prostředí.



Obrázek 1: Zobrazení robotu

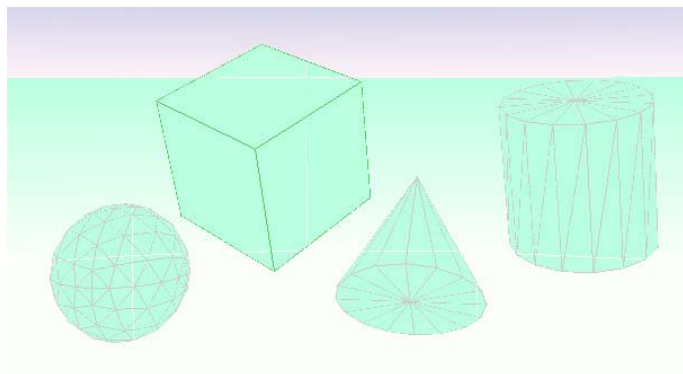
Před konkrétním výběrem vhodného řešení, je nutné určit požadavky na výpočetní algoritmus. Z hlediska řízení robotu požadujeme, aby kontrolní algoritmus pro detekci kolize probíhal v reálném čase, přičemž nesmí příliš zatěžovat výpočetní a paměťovou kapacitu procesoru a který lze urychlovat vhodným nastavováním vstupních parametrů. Vycházíme přitom z předpokladu, že snížením přesnosti detekce vzroste rychlost algoritmu a naopak. Z hlediska konkrétní implementace algoritmu, bude nejlepším řešením použití jazyka C, použitelného jak v prostředí Matlab, tak v samostatně spustitelné aplikaci.

3. Obecný popis základních metod

Veškeré metody detekce kolizí trojrozměrných objektů v prostoru, jsou založeny na vztazích analytické geometrie. Zpravidla lze většinu metod rozdělit do dvou částí. První částí jsou obvykle rovnice analytické geometrie, použitelné pro daný popis objektů v prostoru, tedy o rovnice řešící průniky rovin, přímek, nebo vzdálenosti bodů. Výpočet těchto rovnic je zpravidla jednoduchou záležitostí, problémem je však většinou požadavek zpracování obrovského množství objektů, s nímž potom souvisí nárůst výpočetního výkonu procesoru.

Proto druhou částí algoritmu bývá metoda, která toto množství výpočtů snižuje na únosnou hranici a to většinou tak, že zamezuje v provádění výpočtů, které nejsou z hlediska kolize podstatné. Popřípadě jde o metody, které se zpřesňují, až při detekci možné kolize.

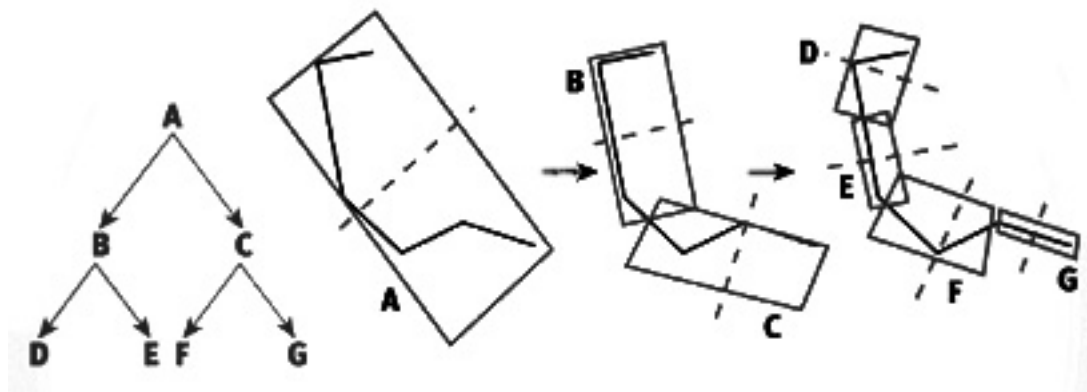
Aby mohli být tyto rovnice použity, je nutné nejprve tělesa v prostoru vhodně definovat, resp. je nutné popsat jejich povrch. K tomuto účelu, jsou nevhodnější grafické formáty, založené na náhradě povrchu tělesa adekvátními rovinnými polygony, kde lze měnit přesnost aproximace povrchu jejich množstvím, nebo jejich tvarem. V těchto případech je detekce kolize založena na hledání průniků rovin, vytvářených polygony. Pomocí těchto metod lze pozici kolize určit s absolutní přesností vůči aproximovanému povrchu, resp. přesnost pozice



kolize závisí na přesnosti aproximace povrchu. Tyto metody využívají složitější výpočetní algoritmus a proto také kladou vyšší nároky na výkon procesoru. Ke zvýšení rychlosti detekce, tedy potřebují vhodné výběrové kritérium pro omezení nepotřebných výpočtů. Urychlit je lze také snížením přesnosti aproximace povrchu na úkor přesnosti pozice detekce.

Obrázek 2: Popis těles polygony

Jinými metodami, jsou postupy založené na hledání průniku obálek objektů. Využívají podobné algoritmy analytické geometrie jako výše popsané metody s tím rozdílem, že objekt je obalen nějakým náhradním tělesem. Tato obálková tělesa jsou zpravidla volena tak, aby byl výpočet kolize jednoduchý a probíhal co nejrychleji. Těleso je až do detekce kolize zastoupeno touto obálkou. V okamžiku detekce kolize dvou obálek je těleso rozděleno na více menších a tím i přesnějších obálek a výpočet kolize se provádí znovu a přesněji a to tak dlouho, dokud nejsou nalezeny obálky, které se neprotínají. Z tohoto procesu je patrné, že detekce kolize potřebuje kritérium, které při určité přesnosti detekce ukončí dělení obálek. Rychlost těchto metod je tedy ovlivněna požadovanou přesností pozice detekce, volbou obálkové plochy a geometrickou členitostí povrchů obalovaných těles.



Obrázek 3: Metoda Bounding Box

Další skupiny metod detekcí kolizí jsou založeny na porovnávání vzdáleností charakteristických bodů. V tomto případě, je povrch tělesa aproximován sítí bodů, přičemž se srovnává vzdálenost dvou bodů z různých těles v prostoru, která se dále vyhodnocuje vůči nějaké referenční hodnotě. Metody založené na tomto principu, obsahují nejjednodušší výpočetní algoritmus, avšak jsou nejnáročnější na množství výpočtů. Proto musí obsahovat vhodné výběrové kritérium pro omezení zbytečných výpočtů a také algoritmus pro vytvoření bodů sítě. Přesnost detekce se potom mění hustotou sítě.

Veškeré výše popsané metody lze libovolně kombinovat a s pomocí výběrového kritéria urychlovat.

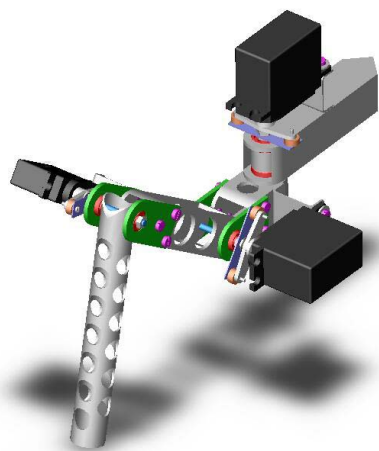
3. Metoda pro detekci kolize končetin robotu

Pro realizovatelnost výše uvedených cílů, je nutné vymezit určité předpoklady, které zjednoduší přístup k dané problematice. Je zřejmé, že robot v konfiguraci dle obr.1, má celkem 12 stupňů volnosti, resp. máme 12 proměnných parametrů. Z praktického hlediska je zřejmé, že kolize může nastat jen v některých případech. Proto byla zavedena následující omezení pohybových stavů (úhlů natočení kloubů robotu):

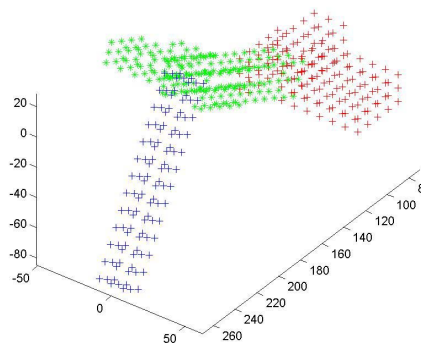
- úhly natočení kloubů jednotlivých končetin se pohybují v konkrétních intervalech a tudíž není možný kontakt končetiny s tělem robotu, ani stočení končetiny pod tělo robotu
- řídicí algoritmus nesmí dovolit vznik nestabilního postoje robotu a tudíž jeho pád, resp. alespoň tři končetiny musí vytvářet trojúhelníkovou základnu

Díky těmto omezením, potom postačuje řešit detekce kolizí pouze mezi sousedními končetinami. A to navíc pouze v případě, že dvě končetiny svírají vůči sobě takový úhel, při kterém může dojít ke kolizi.

Jádrem výpočetní metody pro nalezení kolize mezi končetinami, se stal algoritmus vycházející ze vzdáleností bodů, aproximujících povrch každé končetiny. Pro tento účel musely být tedy vytvořeny, tři množiny bodů (obr. 5) pro jednotlivé pevné části každé



Obrázek 4: Končetina robotu



Obrázek 5: Síť bodů končetiny

končetiny a to dle požadované přesnosti detekce kolize. Vlastní algoritmus potom v každém kroku vychází z těchto dat, která transformuje do aktuálních, globálních souřadnic v prostoru a poté počítá jejich vzdálenosti. V tomto procesu se řeší vždy vzdálenosti každého bodu s každým, tedy N^2 operací (kde N ...počet bodů jedné končetiny).

Z výše uvedených vlastností algoritmu, je tedy patrné, že pro splnění požadavku pro reálnou detekci kolize je nutné vytvořit vhodné omezující kritérium, tedy snížit počet výpočetních operací algoritmu. Nejvhodnějším omezením pro tento případ, je rozdělení operačního prostoru do menších podprostorů, v našem případě krychlí. Jednotlivé body v prostoru můžeme potom přidělit dílčím podprostorům a dále řešit kolize pouze mezi body sdílející stejný podprostor. Tímto způsobem klesá počet operací na $p*m*n$ (kde p ...je počet podprostorů, m a n ...jsou počty bodů z každé končetiny v daném podprostoru).

4. Poděkování

Práce vznikla za podpory pilotního projektu ÚT AV ČR č. 52020 „Realizace základních řídicích členů kráčivého robotu“ a projektu MŠMT MSM 262100024 „Výzkum a vývoj mechatronických soustav“.

4. Závěr

Algoritmus byl v experimentální podobě implementován a odzkoušen v prostředí Simulink, pomocí objektu S-funkce, přičemž se předpokládá jeho využití v řídicím algoritmu robotu.

5. Literatura

- [1] Bezdíček M., Artificial neural network application to walk of a four legged robot, Inženýrská mechanika 2004, Svratka, ČR
- [2] CMP Media, Inc., server pro vývojáře počítačových her a programátory, (2004) <http://www.gamasutra.com/>
- [3] Grepl R., (2004), Control of experimental walking robot using simulating model, Inženýrská mechanika 2004, Svratka, ČR
- [4] Pelikán J., (2003), 3D počítačová grafika na PC
- [5] Švehlák M., Design of small laboratory quadruped robot, Inženýrská mechanika 2004, Svratka, ČR
- [6] The Mathworks, (2004), Manuály k programu Matlab 6.5, Writing S-Functions, www.mathworks.com
- [7] The Virtual Reality Modeling Language, ISO/IEC FDIS 14772-1, ISO/IEC FDIS 14772-1, The VRML Consortium Incorporated, 2002, www.wrml.org

- [8] Turek M., Návody pro vývojáře v OpenGL, (2004), www.nehe.opengl.cz
- [9] Švehlák M., Design of small laboratory quadruped robot, Inženýrská mechanika 2004, Svratka, ČR