Universitatea *Transilvania* din Braşov

HABILITATION THESIS

SERVOSYSTEMS FOR MOTION CONTROL IN THE ROBOTS' TECHNIQUE

Domain: Electrical engineering

Author: Prof.dr.ing. Aurel Fratu

University Transilvania of Brasov

BRASOV, 2015

CONTENT

(A) Rezumat **3**

(B) Scientific and professional achievements and the evolution and development plans for career development **7**

(B-i) Scientific and professional achievements **7**

Introduction 7

Chapter 1 **Dynamic models for control system design of integrated robot and drive systems** **14**

1.1 Dynamics of the nonlinear robotic systems with constraints

1.2 Design of flexible drive systems for feed-forward control: a mechatronic approach 22

Chapter 2 **Haptic interfaces** **34**

2.1 Haptic system modeling and stability analyze

2.2 Hybrid force/position control of the haptic scissors-robot 44

Chapter 3 **The reciprocal robots collision avoidance**........ **62**

3.1 Collision-avoidance using redundant robots

3.2 On-line collision avoidance using Programming by Demonstration 73

Chapter 4 **Modeling and robots behavioral simulation in visual environments** **83**

4.1 System concept development with virtual prototyping

4.2 Virtual Prototyping for robotic systems 91

(B-ii) The evolution and development plans for career development........ **99**

(B-iii) Bibliography **103**

(A) Rezumat

În cadrul tezei actuale de abilitare, realizările mele stiintifice, profesionale şi academice sunt cuprinse într-o perioadă, începând cu anul 1994 (data la care teza de doctorat a fost susţinută, apoi atestată prin Ordinul ministrului 6082 / 21.04.1994) până în prezent (2014). Unele dintre granturile mele naţionale şi internaţionale de cercetare, articole importante, cărţi, brevete, cursuri predate sunt, de asemenea, detaliate în contextul stadiului actual al domeniului stiinţific al ingineriei electrice, cu accent pe aspectele inovatoare şi contribuţii personale.

Prima parte (A) a tezei de abilitare este constituită din prezentul rezumat în limba română.

Cea de a doua parte (B) a tezei curente se referă la realizări stiinţifice, profesionale şi planul de evolutie şi dezvoltare a carierei şi este formată din trei secţiuni (B-i), (B-ii) si (B-iii).

Sectiunea (B-i) conţine patru capitole care tratează patru teme principale de cercetare, prefatate de o introducere.
In introducere se prezintă o privire de ansamblu asupra activităţii autorului tezei de abilitare, cu referire la cele mai importante direcţii de cercetare - clasificate în patru teme principale - realizările profesionale şi academice, disciplinele nou introduse, cursuri predate, contribuţia la dezvoltarea curriculei universitare, misiuni de profesor invitat, studenţi in stagiu, conducerea lucrărilor de diplomă şi disertaţie, cooperare internaţională, activităţi de management, etc.). Cele mai importante aspecte mentionate sunt: un număr de 73 de articole de cercetare publicate în perioada mentionată mai sus, 4 granturi de cercetare (2 ca director de contract) si 8 cărţi.

In capitolul 1 este prezentată urmatoarea temă:

- **Modele dinamice pentru proiectarea sistemului de control al sistemelor integrate de roboti şi sisteme de acţionare**

Principala sarcină a sistemului de control a mişcării robotului manipulator este de a genera mişscare în spaţtiul sarcinii robotice, cu o comandă dată la nivelul sistemelor de acţionare. Abordarea actuală sistematică a proiectării controlerelor pentru sisteme neliniare este metoda liniarizării folosind feedback-ul. Ideea de bază a acestei tehnici este aceea de a proiecta o lege de control care anulează /compensează neliniaritătile sistemului şi rezultă un sistem cu circuit închis cu o dinamică liniară.

In primul subcapitol al capitolului 1 se studiază o ecuație universală de mişcare pentru sisteme robotizate cu constrângeri. Aici, problema controlului sistemelor neliniare este analizată fâra a lua in calcul tipul de controler care urmează să fie utilizat.
Rezultatele prezentate în această lucrare oferă metode noi şi explicite pentru controlul sistemelor puternic neliniare.

Al doilea subcapitol al capitolului 1 se referă la proiectarea sistemului mecatronic pentru controlul mişcării de mare viteză, folosind o buclă feed-forward.
În sistemele integrate de roboti şi sisteme de acţionare principalele caracteristici, în mod special în controlul vitezei de mişcare, sunt complexitatea dinamicii şi incertitudinile, atât parametrice cât şi dinamice. Incertitudinile parametrice apar din cunoaşterea imprecisă a parametrilor cinematici şi a parametrilor de inerţie, în timp ce incertitudinile dinamice apar din cauza

flexibilităţii legăturilor dintre articulaţii, dinamicii actuatorilor, frecării, zgomotului senzorilor şi necunoaşterii dinamicii mediului.

Abordarea propusă include proiectarea sistemului de acţionare şi control a mişcării. Scopul este de a proiecta atât transmisia mecanică cât şi motorul electric pentru a atinge specificaţiile de performanţă cu ajutorul controlerului. Modelul elasto-dinamic invers al sistemului este utilizat pentru a genera comandă şi pentru a stabili constrângerile de proiectare datorită limitelor fizice. Proiectarea este realizată folosind frontiera Pareto care permite o alegere optimă a motorului, precum şi o dimensionare optimă a mecanismului de transmisie. In exemplul prezentat se arată că această abordare de proiectare conduce la obţinerea unui sistem robot integrat robust, tolerant la erori de modelare şi identificare.

In capitolul 2 este prezentată următoarea temă:

- **Interfeţe haptice**.

In primul subcapitol al capitolului 2 se abordează problema de stabilitate a interfeţei haptice în contact cu un perete virtual. Această lucrare prezintă o imagine de ansamblu cu privire la metodologie şi la regiunea de stabilitate afectată de anumiţi parametri: coeficientul de amortizare fizică, constanta de întârziere, caracteristici dinamice ale operatorului uman.
Scopul acestei lucrări este de a analiza influenţa parametrilor dinamici asupra granitelor de stabilitate. Acest studiu arată modul în care factorii cum ar fi coeficientul de amortizare fizică, temporizarea şi parametrii operatorului uman afectează stabilitatea unui sistem haptic. Influenţa tipului de vibraţii asupra stabilităţii, este de asemenea studiat în această lucrare. Rezultatele sunt efectuate asupra parametrilor fizici ai unei interfete haptice de tip PHANTOM®.

În al doilea subcapitol al capitolului 2 este descrisă o metodă pentru ilustrarea senzaţiei tactile a operatiei de tăiere, folosind un tip de foarfece-robot virtual. Este prezentat un nou algoritm pentru a fi utilizat în manipulări generale interactive ale obiectelor deformabile. Algoritmul a fost dezvoltat pentru a afişa deplasarea simultană de translaţie (de înaintare) şi de rotaţie (de decupare) pentru o simulare realistă a operaţiei de tăiere. Am arătat că animaţiile interactive pot fi utilizate pentru a simula funcţionarea dispozitivelor de tip foarfece, folosind reacţia de forţă / cuplu. Prin această lucrare se obţine experienţă in procesul de proiectare în mediul virtual şi se determină noi caracteristici ale posibilelor dispozitive de tip foarfece-robot care folosesc reacţia de cuplu.

In capitolul 3 este prezentată urmatoarea temă:

- **Evitarea reciprocă a coliziunii robotilor**

Scopul primului subcapitol al capitolului 3 este de a analiza problema cinematicii inverse a roboţilor redundanţi şi de a defini sub-sarcina robotică de-a evita obstacolele, folosind mişcarea spontană. Strategia propusă permite utilizarea gradelor de libertate redundante, astfel încât un manipulator să poată evita obstacolele în timp ce efectorii finali urmaresc traiectoria dorită. Se presupune că obstacolele, aflate în spaţiul de lucru al manipulatorului sunt statice. Această strategie este aplicată şi atunci când configuraţia manipulatorului poate fi influenţată de cerinţe suplimentare, cum ar fi limitările la nivelul articulaţiilor, etc.
Eficacitatea strategiei propuse este analizată din punct de vedere teoretic şi ilustrată prin simularea mişcării - intre obstacole simetrice - a unui manipulator redundant cu patru grade de

libertate. Se constată că strategia propusă - în scopul evitării coliziunii în timp ce efectorii finali urmăresc traiectoria impusă - este eficientă şi practică**.**

Cea de a doua lucrare dezvoltă o metoda originala de-a evita coliziunea, bazata pe conceptul de imitare a miscarii. Operatia de imitare a miscarii, presupune existenta unui demonstrator si a unui imitator. În demersul nostru demonstratorul este prototipul robot virtual si imitatorul este robotul fizic real. Intrările pentru procedura de imitatie sunt traiectorile virtuale articulare, reprezentate ca o secvență de valori unghiulare ale articulatiilor sale. Imitatorul imită miscarea originală capturata de la prototipul virtual, si în acelasi timp, respectă constrangerile fizice.

Punctul forte al metodei propuse este că oferă o re-planificare on-line a miscării în fata perturbatiilor (obstacolelor) spatio-temporale, folosind anumite scenarii. Miscarea modelului robot virtual trebuie să acopere toate situatiile posibile - inclusiv prezența obstacolelor accidentale – pentru care robotul fizic va trebui să genereze miscari similare.

Metoda de evitare a coliziunilor, bazată pe imitarea de prototipuri virtuale, trebuie încă îmbunătățită pentru a răspunde la întrebarea: cum pot fi actualizate on-line modele sistemelor dinamice virtuale pentru a lua în considerare evenimentele dinamice din mediul real ? Autorul estimeaza ca programarea prin imitatie, complet automatizata, folosind un sistem suficient de robust pentru aplicații practice, nu va deveni realitate inainte de sfârsitul acestui deceniu.

In capitolul 4 este prezentată urmatoarea tema:

- **Modelarea si simularea comportamentala a robotilor în medii vizuale.**

Scopul primului subcapitol al capitolului 4 este de a evalua problema prototiparii virtuale. Prototiparea virtuala reprezinta un mijloc pentru a descrie calitativ comportamentul produsului sub diverse aspecte.Prototiparea virtuala este o disciplină de inginerie software, care implică modelarea unui sistem, simularea și vizualizarea comportamentului său în conditii de funcționare reale si rafinarea proiectului printr-un proces iterativ. Comportamentul full-motion a sistemelor robotice complexe poate fi analizat înainte de a construi un prototip hardware real. Utilizatorii pot explora rapid multe variante de proiect, de testare si rafinare până cand sistemul robotic este optimizat. Acest lucru poate ajuta la reducerea timpului si a costurilor de dezvoltare a unui nou produs.

Scopul propus in cel de-al doilea subcapitol al capitolului 1 este de a sublinia rolul simulării în diferite domenii ale roboticii. Avand capacitatea de a simula, se deschide o gamă largă de optiuni pentru rezolvarea multor probleme in mod creativ. Se poate modela, vizualiza, investiga si testa un braț de robot, chiar dacă aceasta nu există. Se pot vedea performanțele unui sistem robotic înainte de a fi construit. In mediul virtual este posibil ca soluiile alese sa esueze sau sistemul proiectat sa ‘’explodeze’’ fara a avea consecinte asupra prototipului virtual.

Simularea a fost recunoscută ca un instrument important în robotica, in proiectarea de noi roboti, investigarea performantele lor si în proiectarea aplicațiilor acestora. Simularea permite studierea caracteristicilor structurale si a functiilor sistemelor robotice, la diferite niveluri de detaliu, fiecare situatie avand cerințe diferite pentru instrumentele de simulare. De exemplu, un proces rapid - cum ar fi deplasarea unei articulatii robot - poate fi încetinit pentru a observa toate detaliile in "slow motion".

Pe masura ce complexitatea sistemului creste, rolul simularii devine tot mai important si folosind instrumentele de simulare se pot îmbunătăți proiectele.

În al doilea subcapitol al capitolului 4 se propune folosirea mediului vizual de programare Delphi, pentru a modela si simula comportamentul bratelor robotice de tip serial si de tip paralel. Simulările realizate pentru aceaste aplicatii au servit la dezvoltarea metodelor sistematice pentru crearea de noi sisteme robotizate.
Rezultatele acestei analize constă în identificarea proprietătilor si a constrângerilor care pot aparea in cazul prototipurilor fizice homonime. Structurile virtuale create în această lucrare includ aplicatiile cu brate robotice de tip serial (exemplificare - pentru statii de triere bagaje) si respectiv cu brate robot paralele (exemplificare - pentru platforme de tip hexapod).

Este important de retinut ca principalele rezultate obtinute în domeniul primei directii de cercetare *Modele dinamice pentru proiectarea sistemului integrat de control robot si sisteme de acționare* sunt identificate în sectiunea (B-iii), de existenta cuvintelor *control* si *robot* în titlul unui număr de 19 articole publicate de autorul tezei de abilitare. De asemenea, un contract de cercetare abordeaza o temă similară.

A doua direcție de cercetare, si anume *Interfetele haptice* a fost tratata în 5 articole publicate de autorul tezei de abilitare (identificate de existența cuvântului *haptic* în titlu lor, în sectiunea (B-iii).

Cea de a treia problematica, aceea de *Evitare reciproca a coliziunii robotilor* a fost investigata într-un număr de 8 articole publicate de autorul tezei de abilitare (a se vedea referintele cu existenta cuvintelor *evitare coliziuni* în titlu lor, în sectiunea (B-iii).

Rezultatele privind ultima directie de cercetare, *Modelarea si simularea comportamentului robotilor în medii vizuale,* au fost publicate de autorul tezei de abilitare în 6 articole (a se vedea referintele cu existența cuvintelor *modelare si simulare* în titlul lor, în sectiunea (B-iii).

Cea de-a doua sectiune (B-ii) prezinta *Planul de evolutie si dezvoltare a carierei*

Pentru viitor cercetarea unor noi strategii de comunicare - între mediul informatic virtual si cel fizic real - sunt avute in vedere pentru a fi profund studiate (cu aplicații pentru *interfete haptice* si aplicatii care vizeaza *evitarea reciproca a coliziunilor*). Noi solutii de implementare vor fi propuse (de exemplu *platforma de programare prin imitatie*, bazată pe o idee originală brevetata). De asemenea, am de gând să investigheze noi domenii de cercetare în colaborare cu cercetatori de la Universitatea din Valenciennes si Universitatea din Reims - Franta.

Cea de-a treia sectiune (B-iii) este dedicată referințelor bibliografice.

(B) Scientific and professional achievements and the evolution and development plans for career development

(B-i) Scientific and professional achievements

Introduction

The interest in creating such an overview is mixed. By selecting the most interesting approaches, we want to focus the attention to new techniques and methodologies that may be of high interest to the researchers in the field of collision avoidance. Moreover, this selection can be a useful indicator of the areas that will constitute the future research trends.

The title of my PhD thesis was *Reaserch on positioning system with synchronous servo-motor and micrtocomputer for industrial robots*. I presented it publicly at the "Transilvania" University of Brasov in 1994. Therefore the following overview of my scientific interests and work will be made starting from the above mentioned date up to present (2015).
During this period, ***Dynamic models for control system design, Haptic devices, Robots collision avoidance, Modeling and robots behavioral simulation,*** were the main research topics.

They represent as well my teaching activities. For example, during 2009 I have proposed, elaborate the syllabus and taught two new courses: “Manipulators and Robots Command and Control” and “Visual programming with Applications in Robotics”, within the framework of the bachelor program.

The main developed applications of the above mentioned topics are grouped into four categories that constitute the chapters of the (B-i) section, as follows:

1. **Dynamic model for control system design of integrated robot and drive systems**

    1.1 Dynamics of the nonlinear robotic systems with constraints
    1.2 Design of flexible drive systems for feed-forward control: a mechatronic approach

The contributions to the field of Dynamics of the nonlinear robotic systems with constraints are presented in subchapter 1.1 of chapter 1 of the current thesis. The first article, presented in details in [34], deal with the analytical model of motion in terms of second-order differential equations. This methodology has been inspired by results in analytical dynamics. The explicit closed-form expression provides the entire set of continuous tracking controllers that can exactly track a given trajectory description, assuming that the system’s initial conditions satisfy the description of the trajectory. The explicit closed-form expressions for the controllers can be computed in real time.

Closed-form expressions for all the continuous controllers required for trajectory tracking for nonlinear systems do not make approximations. Furthermore, no approximations or linearization are made here with respect to the trajectory that is being tracked, which may be described in terms of nonlinear algebraic equations or nonlinear differential equations.

Equation obtained is the general, explicit equation of motion for constrained robotic systems with non-ideal constraints. Here, we do not have any restriction regarding the positive definiteness of the symmetric mass matrix. One can obtain the following results:
First result The general equation of motion of a constrained robotic mechanical system, whether or not the mass matrix that arises in the description of the unconstrained motion of the system is singular.
Second result When the mass matrix has full rank, the equation of motion of the constrained system is unique.
Moreover, the approach arrives not just at one nonlinear controller for controlling a given nonlinear system, but also at the entire set of continuous controllers that would cause a given set of trajectory descriptions to be exactly satisfied.

The contributions to the field of *Design of flexible drive systems for feed-forward control* are presented in subchapter 1.2 of chapter 1 of the present thesis. The article [56] published in a Proceeding conference was chosen as the illustrative paper for this topic. It achieved a number of 4 citations in Thomson Web of Knowledge database.

This approach led us to have an over-sizing or performances lower than specifications. The mechanism is indeed sized to transmit a mechanical power (force, velocity). On top of that, to avoid any interaction with the control (resonance excitation, spill-over phenomenon the mechanism can be over-sizing to place the resonant frequencies high enough, what leads to have a high mechanism inertia and therefore an oversized motor. In the case of the mechanism is not oversized (weaker inertia, lower resonance frequencies), a control law damping the vibratory modes will be able to be used.

A state space feedback control of flexible robot will prove to be often delicate to implement, what restricts its use outside laboratories and space applications. An attractive approach in set point tracking is the use of a feed-forward term established from dynamic inverse model as it was proposed for the elastic servomechanisms and the flexible link robots. In addition, the authors of this article offer a modification of the dynamics of the arm of robot to be able to generate a simpler feed-forward control. Mechanical modification is obtained by addition of punctual masses along the arm in order to achieve a relation specified between the frequencies of resonance. These works show interest to shape the system dynamics to achieve better results with control.

A mechatronic design method of flexible drive systems dedicated to feed-forward control has been proposed. The originality of the approach is to enter at the same time the question of design and the control to reach performance requirements. This approach uses the inverse elastic dynamic model of the drive system and its physical limits (motor saturations, mechanical resonance, etc.). Based on this formulation, a two-step design method has been established and applied to an example in order to show the characteristics (performances and robustness) of a design solution.

One of the extensions of this approach of total optimization concerns nonlinear mechanisms and arms of flexible robot with, in this case, a particular attention to non minimal-phase behavior.

Other attractive subject undertake within this framework (Dynamic models for control system) was related to the development of an integrated environment for the visually control movement of the robotic systems. The results were published in several articles e.g. [70] or [71], the former published in an ISI Proceeding, achieving citations in Thomson Web of Knowledge database.

It is merit noticing that the main results achieved in *Control system design of integrated robot and drive systems* field are identified in the section (B-iii), by the existence of *robot control* words in the title and are in number of 19 papers published by the author of this thesis.

The research was focused on finding robust solutions for the following subsystems: Motion control of the synchronous motor and adequate inverter demanded [Brevet RO112454/ 29.08.1997].

The international cooperation regarding this research direction was done with prestigious universities and research institutes. Among them:

- LAMIH (Laboratory of Industrial and Human Automation control, Mechanical engineering and Computer Science) from University of Valenciennes, France;
- Centre in Information and Communication Science and Technologies (CRESTIC) Université de Reims Champagne-Ardenne , France

2 Haptic system

2.1 Haptic system modeling and stability analyze

2.2 Hybrid force/position control of the haptic scissors-robot

The contributions to the field of *Haptic system modeling and stability analyze* are presented in subchapter 2.1 of chapter 2 of the present thesis. The article [62] published in Proceeding Conference is chosen as the first illustrative paper for this topic. It achieved a number of 4 citations in Thomson Web of Knowledge database.

In this approach, the virtual environment is as a part of a model which consider the virtual impedance as a linear spring and damper. The stability region is represented by the stability boundaries in the stiffness-damping plan of the virtual impedance.

Hereby, the influence of parameters such as physical damping and time delay on stability boundaries is evaluated with the transfer function model. The human arm impedance can be included in the model to study the human dynamic effects. There are a lot of works concerning the interaction of the human operator, the haptic device and the virtual environment.

The existence of internal vibration of haptic device has been studied in recent years, but these researches have not clearly explained the influence of the vibration mode parameters. The goal of this paper is to analyze the influence of the dynamic parameters on the stability boundaries.

This study shows how the factors such as physical damping coefficient, time delay and human operator parameters affect the stability of a haptic system. The influence of vibration mode on the stability is also studied in this paper. The results are carried out on the physical parameters of a PHANTOM®.

This paper presents an overview on the effects of dynamic factors on haptic interface stability. The impedance model used to compute the force feedback includes a virtual stiffness and a virtual damping.

The stability boundaries of both virtual parameters are derived from a sampled data model with the Routh-Hurwitz criterion in an analytic way or with the gain margin criterion in a numeric way. The stability analysis is first based on rigid model of device to show the effects of physical damping, time delay and human operator. The first vibration mode due to elastic transmission of haptic device has been also studied. The results allow us to conclude that the vibration modes of device have an intricate effect to the stability of interface. The analysis of the anti-resonant frequency effect leads to explanations of qualitative modification of stability boundaries.

This article has provided engineers some basic guidelines in choosing design specifications of haptic system. In future works the effect of others physical parameters (e.g; time delay, human model) will be studied with a model included vibration modes.

This article was supported in part by the International Campus on Safety and Intermodality in Transportation, the European Community (through the FEDER European Funds for Regional Development), the Délégation Régionale à la Recherche et à la Technologie, the Ministère de l'Enseignement supérieur et de la Recherche, the région Nord Pas-de-Calais and the Centre National de la Recherche Scientifique (CNRS). The international cooperation regarding this research direction was done between Transilvania University of Brasov and University of Valenciennes, France.

The research direction, namely *Haptic system modeling and stability analyze* has been treated in 8 publications (referred by *haptic* word in title, in section (B-iii).

The contributions to the field of *Hybrid force/position control of the haptic scissors-robot*are presented in subchapter 2.2 of chapter 2 of the present thesis. The article [30] published in a Proceeding conference was chosen as the illustrative paper for this topic. It achieved a number of 4 citations in Thomson Web of Knowledge database.

In the first part this section presents an analytical model based on the concepts of contact mechanics and fracture mechanics to calculate forces applied to scissors during cutting of a plate of material.

The model considers the process of cutting as a sequence of deformation and fracture phases. During deformation phases, forces applied to the scissors are calculated from a torque-angle response model synthesized from measurement data multiplied by a ratio that depends on the position of the cutting crack edge and the curve of the blades.

In the second part of this section a method for displaying feeling of cutting, using a virtual scissors type, is described. A new computer haptics algorithm to be used in general interactive manipulations of deformable objects is presented. The algorithm was developed to simultaneously display translational and rotational (cutting) displacement for a realistic cutting simulation. We showed that interactive animations can be used to simulate the functioning of force/torque - feedback scissors devices.

This paper discusses both the experience with the design process in the virtual environment and the novel features of the possible portable torque-feedback scissors devices. We will present an approach for designing a more natural user interface, without resorting to various special haptic input/output devices.In this example, our scenarios were simple enough so that the absence of detailed specifications did not obstruct our later prototyping work.

The model was parameterized with data and equations and simulated over time. Simulation results can be viewed on a graphical monitor, and finally exported to a physical device.

The research direction, namely *Hybrid force/position control of the haptic scissors-robot* has been treated in 8 publications (referred by the words *haptic control* in title, in section (B-iii),).

3 Reciprocal robots collision avoidance

3.1 Collision-avoidance using redundant robots
3.2 Collision avoidance using Programming by Demonstration

The contributions to the field of the *Collision-avoidance using redundant robots* are shown in subchapter 3.1 of chapter 3, of the current thesis. The aim of the first presented work, presented in details in [60], is to identify the strategy that allows the use of redundant degrees of freedom such that a manipulator can avoid obstacles while tracking the desired end-effectors trajectory. It is supposed that the obstacles in the workspace of the manipulator are static. The results were published in an ISI Proceeding, achieving citations in Thomson Web of Knowledge database.

The strategy is based on the redundant inverse kinematics and is discussed by theoretical considerations. This strategy has the advantage that the configuration of the manipulator can be influenced by further requirements such as joint limits, etc. In this paper the problem of redundant inverse kinematics is reviewed and obstacles avoid subtask for exploiting the self motion are defined.

The contributions to the field of the *Collision avoidance using Programming by Demonstration* are presented in subchapter 3.2 of chapter 3 and concern an article published in an academic journal [69]. In this article, based on original idea, the author proposes a new strategy to robot collision avoidance using imitation programming paradigm.

To program the desired motion sequence for the physical robot, one captures the motion reference paths from her virtual robot model and maps these to the joint settings of the physical robot. Motion imitation requires transfer of a dynamical signature of a movement of the virtual robot to the physical robot, i.e. the robots should be able to encode and reproduce a particular path as one with a specific velocity and/or an acceleration profile. Furthermore, the virtual robot must cover all possible contexts in which the physical robot will need to generate similar motions in unseen context.

In this paper the graphical simulation prototype system has been developed to support and demonstrate different aspects of robots behavior when met the obstacles in his work environment. All of these programs are write in Delphi language and enable the manifestation of global behaviors.
For itself they include the movement of virtual robot manipulator, the contact with virtual objects and the collision avoid, into the work environment, with expected or unexpected virtual objects.

The effectiveness of the proposed strategy is discussed by theoretical considerations and illustrated by simulation of the motion of the four-joint planar manipulators between symmetric obstacles. It is shown that the proposed collision-free strategy while tracking the end-effector trajectory is efficient and practical.

A redundancy-resolution scenario was proposed to achieve obstacle avoidance. In the following simulations the main task consists of tracking the position and orientation trajectories, generated by linear interpolation, between the initial and final poses. It should be noted that interpolation of rotations is a much more complex problem than point interpolation. For this reason, we use simple linear interpolation for both translation and rotation, which nevertheless leads to satisfactory results.

In the simulation presented in this section the redundancy resolution is implemented in closed-loop. One solution to this problem is online transmission of a robot configuration to a workstation running a graphics visualization of the arm how serves as a virtual environment; the graphics model of the robot mirrors the exact motion of the arm, and the environment can be modeled in the graphics program.

The problematic, *collision avoidance* has been investigated in a number of 8 papers (see references with *collision avoidance* words in title, from section (B-iii)) and one patent [*Method and installation for programming the motion paths of robotic arm articulations,* Patent Number(s): RO – 129121(A0)]. The novelty of the proposed system in this patent is related to robot programming by imitation based on her virtual homonym

.

4 Modeling and robots behavioral simulation

4.1 System concept development with virtual prototyping
4.2 Virtual Prototyping for robotic systems

The contributions to the field of *System concept development with virtual prototyping* are presented in subchapter 4.1 of chapter 4, of the current thesis.
The aim of this approch, presented in details in [55], and published in an international colloque is to identify the strategy that allows the rapid prototyping to create a three-dimensional virtual model.

Through careful development of details, 3D graphics applications can generate virtual prototypes that provide a useful tool supporting engineering and specialty disciplines analysis for a wide variety of system developments, both early in the concept development phase and later in advanced engineering. Virtual prototyping permits users, designers, and logisticians to visualize two and three dimensional relationships and clearances for joint multi-disciplinary analysis. This early analysis and identification of design issues offers potential to reduce development costs since mistakes will be made in the computer rather than on full-scale prototypes. Due to design maturity, virtual prototypes developed in advanced engineering development will be based on design analysis and known qualities. In contrast, exceptional care must be taken during the technology assessment and advanced concept phases, when details are less certain, to conduct thorough research that produces and supports a credible product based on expert opinion, technology analysis, and traceable assumptions rather than “fantasy.” The development of worksheets containing source and rationale information on each virtual component is a disciplined technique that helps to ensure reality is not misrepresented.

The second detailed approach concerns *Virtual Prototyping for robotic systems* and is based on an article [33] published in a Proceedings Conference. This approach is presented in subchapter 4.2 of chapter 4, of the current thesis.
The purpose of the paper is to emphasize the role of the simulation in different fields of robotics. Also we recommend to researchers to use the general dynamic engines and general visual programming languages for the simulation and visualization of robotic systems.
This paper discusses the using of the visual programming Delphi environment, for facilitating intuitive robot programming. From a robotic point of view, the virtual robotic structures created

were simplified and their analyzing allows a better description of the behavior of real robotic structures.

The simulations prepared for this application were also made for the development of the systematic methods for creating the new robotic systems. The results of this investigation consist in an identification of the properties and constraints of the virtual robot prototypes.
The virtual structures created in this paper encompass the behavior of robotic systems. They may present a behavior which is qualitatively different from the behavior that the robots currently have.

In the end, the simulation in robotics has reached a very important role and by using different simulation software, the present and future capabilities of complex robotic systems can be significantly improved.

In the virtual prototypes gestures imitation represent one of the most attractive technology in the field. This work proposes a fingertip -based approach for *programming the motion paths of robotic arm articulations*. The novelty of the proposed system is related to robot programming by imitation based on her virtual homonym.

The results regarding the last research direction, *Virtual Prototyping for robotic systems* for robotics, were published in 16 scientific publications by author (see references signed by author, with v*irtual prototyping* words in title, in section (B-iii).

The international cooperation regarding this research direction was done with prestigious research group; among them GREAH (Groupe de Recherche en Electrotechnique et Automatique du Havre) de l'Université du Havre**,** France.

I have a dynamic activity for creative work: I am owner of **22** brevets of invention. For this activity I was awarded with *Elite Inventor Brevet* by Commission of Invention of Romanian Academy.
From 2006 until present (2015), I was visiting professor, in six years for one month.
From 1994 to present I am a member of different academic/professional societies as SIR, SRR, IPIMEA and ACER.

Chapter 1 **Dynamic models for control system design of integrated robot and drive systems**

1.1 Dynamics of the nonlinear robotic systems with constraints

Introduction

Most of the robotic applications are restricted to slow-motion operations without interactions with the environment. This is mainly due to limited performance of the available controllers in the market that are based on simplified system models. To increase the operation speed with more servo accuracy, advanced control strategies are needed.

The main specifically properties in the speed motion control of the robots systems are the complexity of the dynamics and uncertainties, both parametric and dynamic. Parametric uncertainties arise from imprecise knowledge of kinematics parameters and inertia parameters, while dynamic uncertainties arise from joint and link flexibility, actuator dynamics, friction, sensor noise and unknown environment dynamics.

Robot's motion trajectories are typically specified in the task space in the terms of the time history of the end-effector's position, velocities and acceleration. Operational space (also known as task space) is the space in which high-level motion and force commands are issued and executed.

The operational-space formulation is therefore particularly useful in the context of motion and force control systems. On the other hand, in the joint space control methods, is assumed that the reference trajectory is available in terms of the time history of joints positions and orientations of robot arm.

The natural strategy to achieve task space control goes through two successive stages:

- in the first stage, the robot's kinematics in the task space variables is passed into the kinematics corresponding joint space variables, and then;
- in the second stage is designed the control in the joint space.

Because of the complexity of both the kinematics and dynamics of the manipulator and of the task to be carried out, the motion control problem is generally decomposed into three stages:

- motion planning,
- trajectory generation,
- trajectory tracking.

The main problem of motion robot control is to generate the motion in the task space with a given command at joints level. Motion control of robot arm accomplishes the following functions:

- to find of the corresponding movements in joints;
- to generate of control signals for the actuators to produce the input torques;
- to synthesize of programmed paths.

For trajectory tracking, the computed reference trajectory is then presented to the controller, whose function is to cause the robot to track the given trajectory as closely as possible. For design of the tracking controller, one assumes that the reference trajectory and path have been pre-computed.

Control of robot manipulators is naturally achieved in the joint space, since the control input are joint torques. But, the user specifies a motion in the task space, and thus it is important to extend the control problem to the task space. This can be achieved by different strategies. The more natural strategy consists of inverting the kinematics of the manipulator to compute the joint motion corresponding to the given end-effectors motion.

Thus, the methods used to appointment primarily depend on linearization and/or PID-type control, and they envisage suppositions on the structure of the control effort.

Control of Nonlinear Dynamical Systems

Most physical robotic systems are inherently nonlinear. Thus, control of nonlinear systems is a subject of active research and increasing interest. However, most controller design techniques for nonlinear systems are not systematic and/or relate only to very specific cases. The most general results available for nonlinear processes relate to scenarios in which:

- all uncertainty is parametric with a known functional dependence of the state-space model with respect to the unknown parameter, and
- there is no measurement noise nor disturbances.

Under these admittedly restrictive assumptions the results available are quite general. For each possible value of the parameter, one needs to know how to design a (non-adaptive) controller that would stabilize the process if the parameter value was known. Such controller should be able to guarantee input-to-state stability with respect to an appropriately introduced disturbance. For each value of the parameter, one needs to know how to design a (non-adaptive) output estimator for the process that would converge to the process output if the parameter value was known. This is an insignificant subject when the whole state of the process can be measured, but can still be challenging for nonlinear systems for which the state cannot be measured.

The main challenge that remains open in the control of nonlinear systems is robustness with respect to disturbances. Although, the algorithms appear to work well in the presence of disturbances, few stability results are available.
There are few systematic procedures to design controllers/estimators for nonlinear systems that are robust with respect to disturbances. In addition, there are also few results to analyze the closed-loop switched nonlinear systems that arise as one switches among different controllers.

Current systematic approach to design controllers for nonlinear systems is feedback linearization. The basic idea of this technique is to design a control law that cancels the nonlinearities of the plant and yields a closed-loop system with linear dynamics. However, the technique is not robust to disturbances and uncertainties in the robot parameters, can yield to uncontrolled dynamics called zero dynamics and can only be applied to systems verifying certain vector field relations.

The development of controllers for nonlinear robotic systems has been an area of intense research. Many controllers that have been developed for trajectory tracking of complex nonlinear and multi-body systems rely on some approximations and/or linearization. Most control designs restrict controllers for nonlinear systems to be affine in the control inputs. Often, the system equations are linearized about the system's nominal trajectory and then the linearized equations are used along with various results from the well-developed theories of linear control. While this often works well in many situations, there are some situations in which better controllers may be needed. This is especially so when highly accurate trajectory tracking is required to be done in real time on systems that are highly nonlinear such the robotic systems.

In the robotics literature, trajectory tracking using inverse dynamics and model reference control has been used for some time now, and the methods developed therein can be seen as particular subclasses of the formulation discussed in the present paper. Trajectory tracking in the adaptive control context (which is not the subject of this paper) has also been explored together with specific parameterizations to guarantee linearity in the unknown parameters of a system.

Controllers that Cause a Robotic System to Track a Given Trajectory

The equations of motion for constrained mechanical systems are based on the principle of Lagrange. The principle states that at each instant of time *t*, a constrained mechanical system evolves in such a manner that the total work done by all the forces of constraint under any set of virtual displacements is always zero. This principle, which in effect prescribes the nature of the forces of constraint which act upon a mechanical system, has been found to yield, in practice, adequate descriptions of the motion of large classes of mechanical systems, thereby making it an extremely useful and effective principle.

This paper takes a generally different approach that is based on recent results from analytical dynamics. Here the complete nonlinear problem is addressed with no assumptions on the type of controller that is to be used.
One considers the robot dynamics model, given by the joint-space formulation, usually presented in the canonical forms:

$$\boldsymbol{M}(\boldsymbol{q},\boldsymbol{t})\ddot{\boldsymbol{q}}+\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})+\boldsymbol{g}(\boldsymbol{q})+\boldsymbol{J}^{\boldsymbol{T}}\boldsymbol{F}=\tau \tag{1.1.1}$$

$\boldsymbol{M}$ is an *n* by *n* symmetric, positive-definite matrix and is called the generalized, or joint-space, inertia matrix, $\boldsymbol{C}$ is an *n* by *n* matrix such that $\boldsymbol{C}\dot{\boldsymbol{q}}$ is the vector of Coriolis and centrifugal terms - collectively known as velocity product terms- $\boldsymbol{g}$ is the vector of gravity terms and $\boldsymbol{F}$ is a vector of forces exerted by the end-effectors. More terms can be added to this equation, as required, to account for other dynamical effects (e.g., viscous friction).

The symbols $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}}$, and $\boldsymbol{\tau}$ denote *n*-dimensional vectors of joint position, velocity, acceleration and effort variables respectively, where *n* is the number of degrees of motion freedom (DoF) of the robot mechanism.

This equation shows the functional dependencies explicitly: $\boldsymbol{M}$ is a function of $\boldsymbol{q}$, $\boldsymbol{C}$ is a function of $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$, and so on. Once these dependencies are understood, they are usually omitted.

Consider an unconstrained nonlinear mechanical robot system described by the second order differential equation of motion under the form:

$$\boldsymbol{M}(\boldsymbol{q},\boldsymbol{t})\ddot{\boldsymbol{q}} = \boldsymbol{Q}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{t}) \tag{1.1.2}$$

$$\boldsymbol{q}(0) = \boldsymbol{q}_0 \quad \dot{\boldsymbol{q}}(0) = \dot{\boldsymbol{q}}_0 \tag{1.1.3}$$

where, $\boldsymbol{q}_0$ and $\dot{q}_0$ are the position and velocity vectors at initial time of the robot with *n* DoF; the dots indicate differentiation with respect to time. Equations (1.1.1) and (1.1.2) can be obtained using Lagrangean model.

The *n*- nonlinear vector $\boldsymbol{Q}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{t})$, on the right hand side of equation (1.1.2) is a 'known' vector in the sense that it is a known function of its arguments. By 'unconstrained' one means that the components of the initial velocity $\dot{\boldsymbol{q}}_0$ of the robot system can be independently assigned.

By 'unconstrained' one means here that the *n* coordinates, $\boldsymbol{q}$ are independent of one another, or are to be treated as being independent of each other.

One requires that this mechanical system be controlled so that it tracks a trajectory that is described by the following consistent set of *m* equations:

$$\boldsymbol{\Phi}_i(\boldsymbol{q},t) = 0 \qquad i = 1....h \tag{1.1.4}$$

and

$$\boldsymbol{\Psi}_i(\boldsymbol{q},\dot{\boldsymbol{q}},t) = 0 \quad i = h+1,....m \tag{1.1.5}$$

Suppose further that the unconstrained system is now subjected to the *m* constraints.

One assumes that the mechanical robot system's initial conditions are such as to satisfy these relations at the initial time. The latter set of equations, which are non-integrable, is non-holonomic.

In order to control the system so that it exactly tracks the required trajectory i.e. satisfies equations (1.1.3) and (1.1.4) one must apply an appropriate control *n*-vector $\boldsymbol{Q}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{t})$ so that the equation of motion of the controlled system becomes

$$\boldsymbol{M}(\boldsymbol{q},\boldsymbol{t})\ddot{\boldsymbol{q}} = \boldsymbol{Q}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{t}) + \boldsymbol{Q}_c(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{t}) \tag{1.1.6}$$
$$\boldsymbol{q}(0) = \boldsymbol{q}_0 \qquad \dot{\boldsymbol{q}}(0) = \dot{\boldsymbol{q}}_0$$

where now, the components of the *n*-vectors $\boldsymbol{q}_0$ and $\dot{q}_0$ satisfy equations (1.1.4) and (1.1.5) at the initial time, $t = 0$.

Throughout this paper, one shall, for brevity, drop the arguments of the various quantities, unless needed for clarity.

The controlled system is described by the relation (1.1.6), where $\boldsymbol{Q}_c$ is the control vector.

One begins by expressing equation (1.1.6) in terms of the weighted accelerations of the system. For any positive-definite n by n matrix $\boldsymbol{P}$ ($\boldsymbol{q}$, t), one define the matrix:

$$\boldsymbol{G}(\boldsymbol{q},\boldsymbol{t})\boldsymbol{q} = [\boldsymbol{P}^{1/2}(\boldsymbol{q},\boldsymbol{t})\boldsymbol{M}(\boldsymbol{q},\boldsymbol{t})]^{-1} \tag{1.1.7}$$

Pre-multiplying equation (1.1.6) by $\boldsymbol{P}^{1/2}(\boldsymbol{q}, t)$, the ''pondered'' equation, which will indicate using the superscript p, is obtained as:

$$\ddot{\boldsymbol{q}}^{\boldsymbol{p}} = \boldsymbol{a}^{\boldsymbol{p}} + \ddot{\boldsymbol{q}}_{\boldsymbol{c}}^{\boldsymbol{p}} \tag{1.1.8}$$

where

$$\boldsymbol{a}^{\boldsymbol{p}} = \boldsymbol{G}^{-1}\boldsymbol{a}\text{, and }\ \ddot{\boldsymbol{q}}_{\boldsymbol{c}}^{\boldsymbol{p}} = \boldsymbol{G}^{-1}\ddot{\boldsymbol{q}}_{\boldsymbol{c}} \tag{1.1.9}$$

One designates the acceleration of the uncontrolled system by:

$$\boldsymbol{a}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{t}) = \boldsymbol{M}^{-1}(\boldsymbol{q},\boldsymbol{t})\boldsymbol{Q}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{t}) \tag{1.1.10}$$

In equation (1.1.6), one identifies the expression:

$$\ddot{\boldsymbol{q}}_{\boldsymbol{c}}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{t}) = \boldsymbol{M}^{-1}(\boldsymbol{q},\boldsymbol{t})\boldsymbol{Q}_{\boldsymbol{c}}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{t}) \tag{1.1.11}$$

witch can be viewed as the deviation of the acceleration of the controlled system from that of the uncontrolled system.
From equation (1.1.8), one obtains the expression:

$$\ddot{\boldsymbol{q}} = \boldsymbol{a} + \ddot{\boldsymbol{q}}_{\boldsymbol{c}} \tag{1.1.12}$$

One differentiates equation (1.1.4) twice and equation (1.1.5) once with respect to time t, giving the set of equations:

$$\boldsymbol{A}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{t})\ddot{\boldsymbol{q}} = \boldsymbol{b}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{t}) \tag{1.1.13}$$

where $\boldsymbol{A}$ is an m by n matrix of rank k and $\boldsymbol{b}$ is an m-vector. With equations (1.1.6), (1.1.8) and (1.1.12) equation (1.1.13) can be further expressed as:

$$\boldsymbol{B}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{t})\ddot{\boldsymbol{q}}_{\boldsymbol{c}} = \boldsymbol{b}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{t}) \tag{1.1.14}$$

where $\boldsymbol{B}$ is an m by n matrix who is calculated by the expression :

$$\boldsymbol{B}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{t})\ddot{\boldsymbol{q}}_{\boldsymbol{c}} = \boldsymbol{A}(\boldsymbol{q},\boldsymbol{q},\boldsymbol{t})[\boldsymbol{P}^{1/2}(\boldsymbol{q},\boldsymbol{t})\boldsymbol{M}(\boldsymbol{q},\boldsymbol{t})]^{-1} \tag{1.1.15}$$

One can now express the accelerations n-vector $\ddot{\boldsymbol{q}}$ in terms of its orthogonal projections on the range space of $\boldsymbol{B}^{\mathrm{T}}$ and the null space of $\boldsymbol{B}$, so that:

$$\ddot{\boldsymbol{q}} = \boldsymbol{B}^{+}\boldsymbol{B}\,\ddot{\boldsymbol{q}} + (\boldsymbol{I} - \boldsymbol{B}^{+}\boldsymbol{B})\ddot{\boldsymbol{q}} \tag{1.1.16}$$

In equation (1.1.16), the matrix $\boldsymbol{B}^{+}$ denotes the Moore–Penrose generalized inverse of the matrix $\boldsymbol{B}$. It should be noted that equation (1.1.16) is a general identity that is always valid since it arises from the orthogonal partition of the identity matrix $\boldsymbol{I} = \boldsymbol{B}^{+}\boldsymbol{B} + (\boldsymbol{I} - \boldsymbol{B}^{+}\boldsymbol{B})$.

Using equation (1.1.14) in the first member on the right hand side of equation (1.1.16), and equation (1.1.12) to replace $\ddot{\boldsymbol{q}}$ in the second member, one gets:

$$\ddot{\boldsymbol{q}} = \boldsymbol{B}^{+}\boldsymbol{b} + (\boldsymbol{I} - \boldsymbol{B}^{+}\boldsymbol{B})(\boldsymbol{a} + \ddot{\boldsymbol{q}}_{c}) \tag{1.1.17}$$

which, due to equation (1.1.11), yields:

$$\boldsymbol{B}^{+}\boldsymbol{B}\,\ddot{\boldsymbol{q}}_{c} = \boldsymbol{B}^{+}(\boldsymbol{b} - \boldsymbol{B}\boldsymbol{a}) \tag{1.1.18}$$

The general solution of the linear set of equations (1.1.18) is given by:

$$\ddot{\boldsymbol{q}}_{c} = (\boldsymbol{B}^{+}\boldsymbol{B})^{+}\boldsymbol{B}^{+}(\boldsymbol{b} - \boldsymbol{B}\boldsymbol{a}) + [\boldsymbol{I} - (\boldsymbol{B}^{+}\boldsymbol{B})^{+}(\boldsymbol{B}^{+}\boldsymbol{B})]\boldsymbol{z} \tag{1.1.19}$$

After any combination one obtains the second equality:

$$\ddot{\boldsymbol{q}}_{c} = \boldsymbol{B}^{+}(\boldsymbol{b} - \boldsymbol{B}\boldsymbol{a}) + (\boldsymbol{I} - \boldsymbol{B}^{+}\boldsymbol{B})\boldsymbol{z} \tag{1.1.20}$$

where the *n*-vector $\boldsymbol{z}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{t})$ is any arbitrary *n*-vector. To obtain the second equality above, one used the property that $(\boldsymbol{B}^{+}\boldsymbol{B})^{+} = (\boldsymbol{B}^{+}\boldsymbol{B})$ in the two members on the right hand side along, with the property so that $\boldsymbol{B}^{+}\boldsymbol{B}\boldsymbol{B}^{+} = \boldsymbol{B}^{+}$.

The set of all possible controls $\boldsymbol{Q}_{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{t})$ (or controllers) that causes the controlled system to exactly track the required trajectory is explicitly given by

$$\boldsymbol{Q}_{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{t}) = \boldsymbol{P}^{-1/2}\ddot{\boldsymbol{q}}_{c} = \boldsymbol{P}^{1/2}\boldsymbol{B}^{+}(\boldsymbol{b} - \boldsymbol{B}\boldsymbol{a}) + \boldsymbol{P}^{1/2}(\boldsymbol{I} - \boldsymbol{B}^{+}\boldsymbol{B})\boldsymbol{z} \tag{1.1.21}$$

The mechanical robotic system, described by the nonlinear Lagrange equation (1.1.1), is explicitly controlled through the addition of a control, *n*-vector $\boldsymbol{Q}_{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{t})$, provided by equation (1.1.21), in witch the *n*-vector $\boldsymbol{z}$ may be chosen still ensures that the description of the trajectory given by equations (1.1.3) and (1.1.4) is exactly satisfied.

Explicit Equations of Motion for General Constrained System

The constrained mechanical system described by Eqs. (1.1.1)-(1.1.4) evolves in time in such a manner that the total work done at any time, t, by the constraint force *n*-vector Q_c under virtual displacements at time *t* is given by (1.1.21).

The work done by the forces of constraints under virtual displacements at any instant of time *t* can be expressed as:

$$\boldsymbol{w}^{T}\boldsymbol{Q}_{c}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{t})=\boldsymbol{w}^{T}\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{t}) \qquad (1.1.22)$$

where $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{t})$ is an n-vector describing the nature of the non-ideal constraints, which could be obtained by experimentation and/or observation. The virtual displacement vector, $\boldsymbol{w}$(t), is any non-zero n-vector that satisfies:

$$\boldsymbol{A}(\boldsymbol{q},\dot{\boldsymbol{q}},t)\boldsymbol{w}=0 \qquad (1.1.23)$$

Solving equation (1.1.23), the n-vector $\boldsymbol{w}$ can be written as (we suppress the arguments for clarity):

$$\boldsymbol{w}=(\boldsymbol{I}-\boldsymbol{A}^{+}\boldsymbol{A})\gamma \qquad (1.1.24)$$

where $\boldsymbol{\gamma}$ is any arbitrary n-vector, and $\mathbf{A}^{+}$ is the Moore–Penrose inverse of the matrix $\mathbf{A}$. Substituting equation (1.1.24) in equation (1.1.22), one obtains:

$$\gamma(\boldsymbol{I}-\boldsymbol{A}^{+}A)\boldsymbol{Q}_{c}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{t})=\gamma(\boldsymbol{I}-\boldsymbol{A}^{+}A)\boldsymbol{C}(\boldsymbol{q},\boldsymbol{q},\boldsymbol{t}) \qquad (1.1.25)$$

Since each component of the vector $\boldsymbol{\gamma}$ can be independently chosen, equation (1.1.25) yields:

$$(\boldsymbol{I}-\boldsymbol{A}^{+}A)\boldsymbol{Q}_{c}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{t})=(\boldsymbol{I}-\boldsymbol{A}^{+}A)\boldsymbol{C}(\boldsymbol{q},\boldsymbol{q},\boldsymbol{t}) \qquad (1.1.26)$$

Pre-multiplying equation (1.1.6) by $(\boldsymbol{I}-\boldsymbol{A}^{+}\boldsymbol{A})$ and using equation (1.1.26), one gets

$$\begin{aligned}&(\boldsymbol{I}-\boldsymbol{A}^{+}A)\boldsymbol{M}(\boldsymbol{q},\boldsymbol{t})\ddot{\boldsymbol{q}}=\\&(\boldsymbol{I}-\boldsymbol{A}^{+}A)[\boldsymbol{Q}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{t})+\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{t})]\end{aligned} \qquad (1.1.27)$$

Equations (1.1.27) and (1.1.13) can now be written together as:

$$\begin{bmatrix}(\boldsymbol{I}-\boldsymbol{A}^{+}\boldsymbol{A}) \quad \boldsymbol{M}(\boldsymbol{q},\boldsymbol{t})\\ \boldsymbol{A}\end{bmatrix}\ddot{\boldsymbol{q}}=\begin{bmatrix}(\boldsymbol{I}-\boldsymbol{A}^{+}\boldsymbol{A}) \quad [\boldsymbol{Q}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{t})+\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{t})]\\ \boldsymbol{b}\end{bmatrix} \qquad (1.1.28)$$

Defining

$$\widehat{\boldsymbol{M}}=\begin{bmatrix}(\boldsymbol{I}-\boldsymbol{A}^{+}\ A)\boldsymbol{M}(\boldsymbol{q},\boldsymbol{t})\\ \boldsymbol{A}\end{bmatrix} \qquad (1.1.29)$$

One can solve equation (1.1.28) to get:

$$\ddot{\boldsymbol{q}} = \widehat{\boldsymbol{M}}^{+}(\boldsymbol{q},\boldsymbol{t})\left[\underbrace{[\boldsymbol{Q}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{t}) \quad +\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{t})}_{\boldsymbol{b}}\right] + (\boldsymbol{I}-\widehat{\boldsymbol{M}}^{+}\widehat{\boldsymbol{M}})\boldsymbol{\eta} \tag{1.1.30}$$

where η is an arbitrary *n*-vector.
Equation (1.1.30) is the general, explicit equation of motion for constrained robotic systems with non-ideal constraints. Here, we do not have any restriction regarding the positive definiteness of the symmetric mass matrix $\boldsymbol{M}$. It is allowed to be singular.

One can obtain the following results:
First result The general equation of motion of a constrained robotic mechanical system described by relations (1.1.1)–(1.1.3), whether or not the matrix $\boldsymbol{M}$ that arises in the description of the unconstrained motion of the system is singular, is given by Eq. (1.1.30).
The second member in equation (1.1.30) disappears for then $\widehat{\boldsymbol{M}}^{+} = (\widehat{\boldsymbol{M}}^{\boldsymbol{T}}\widehat{\boldsymbol{M}})^{-1}\widehat{\boldsymbol{M}}^{\boldsymbol{T}}$.

In general, because of the second member on the right-hand side of equation (1.1.30), the acceleration of a system with a singular mass matrix is not necessarily unique. However, when the (*m* x *n*) by *n* matrix $\widehat{\boldsymbol{M}}$ has full rank so that $\widehat{\boldsymbol{M}}^{+}\widehat{\boldsymbol{M}} = \boldsymbol{I}$ and so the equation of motion becomes unique and consequently, one obtains another important result:

Second result When the matrix $\widehat{\boldsymbol{M}}$ has full rank, the equation of motion of the constrained system is unique and is given by the equation:

$$\ddot{\boldsymbol{q}} = \widehat{\boldsymbol{M}}^{+}(\boldsymbol{q},\boldsymbol{t})\left[\underbrace{[\boldsymbol{Q}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{t}\ )+\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{t})}_{\boldsymbol{b}}\right] \tag{1.1.31}$$

$$\ddot{\boldsymbol{q}} = \left[\underbrace{(\boldsymbol{I}-\boldsymbol{A}^{+}\boldsymbol{A}\ )\boldsymbol{M}(\boldsymbol{q},\boldsymbol{t})}_{\boldsymbol{A}}\right]^{+}\left[\underbrace{[\boldsymbol{Q}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{t}\ )+\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{t})}_{\boldsymbol{b}}\right] \tag{1.1.32}$$

Conclusion

This paper presents the motion in terms of second-order differential equations. This methodology has been inspired by results in analytical dynamics model.
The explicit closed-form expression (1.1.17) provides the entire set of continuous tracking controllers that can exactly track a given trajectory description, assuming that the system's initial conditions satisfy the description of the trajectory. The explicit closed-form expressions for the controllers can be computed in real time.

Closed-form expressions for all the continuous controllers required for trajectory tracking for nonlinear systems do not make approximations. Furthermore, no approximations or linearization are made here with respect to the trajectory that is being tracked, which may be described in terms of nonlinear algebraic equations or nonlinear differential equations.

1.2 Design of flexible drive systems for feed-forward control: a mechatronic approach

Dynamic characteristics of a controlled mechanism result from choice of control law and mechanical design at the same time, due to physical limits such as actuator saturations and mechanical resonance frequencies. This paper concerns mechatronic design of mechanisms for the feed-forward control of high-speed/precision motion. The proposed approach includes the design of drive system and the motion control. The goal is to design both mechanical transmission and electrical motor in order to achieve the performance specifications with the controller.

The inverse elasto-dynamic model of the system is used at the same time to generate the 4^{th} order feed-forward model for control and to establish the design constraints due to physical limits. The design problem is then represented in the form of the Pareto front which allows a selection and an optimal choice of the motor as well as an optimal sizing of the transmission mechanism. An example shows that this design approach leads to obtain the performance specified at the beginning of the process. Moreover, the performances are robust to modelling and identification errors.

Introduction

The motion control of mechanism occurs in a broad variety of applications: industrial robot, Computer Numerical Control machine tool, wing aileron motion control for airplane, drive system for satellite attitude control and appendage maneuvers, etc. Three types of motion control can be distinguished: target tracking (which the trajectory is unknown a priori), point-to-point positioning control (e.g. pick-and-place operations handling systems) and trajectory following (e.g. laser cutting process).The command input for point-to-point positioning and trajectory following is most often calculated according to a motion time evolution law. The intended performance can come down to two objectives: agility (velocity and acceleration) and precision. These ones are hard to reconcile due to elastic deformation and mechanical resonance. They constitute an important problematic in the field of servomechanism control (Ellis and Lorenz, 2000), robots (Good et al., 1984), CNC machine tool, etc.

The contribution of this paper is of methodological order with an original approach and an original formalism for drive system design. This paper is focused on position trajectory following (linear and angular), according to a motion time evolution law, of a load with an electric motor and a mechanical transmission. We propose an approach of mechatronic design covering at the same time the drive system design and its control to obtain an optimum global solution (plant and control).

A design global framework in mechatronic field is given for instance in the paper (Cetinkunt,1991), but it does not exist any link between system design and its control. The framework chosen here does not aim at the maximizing performance, but rather the answer to the most right at a specific requirement by minimizing for instance the costs for industrial equipment or the total mass of a space application. The performance requirements depend on the application area; for instance the high-speed motion for robot manipulators, the precision for CNC machine tools and the reliability in aeronautics.

So, to achieve the given performances (agility and precision), the mechatronic design depends on a dynamic model of drive system by taking into account elastic deformations of the mechanism and physical limits of such dynamic system. This model is used to generate a feed-forward control.

The trajectory generation leans on a motion time evolution law which leads us to define the motion specifications (agility) to be follow by the load. The model with the physical limits is also used to define the design constraints of the drive system according to an appropriate formalism. The selection and sizing method of the electromechanical components is succinctly described. An example allows showing results obtained in accordance with the proposed approach and, notably, to prove that the performances obtained with feed-forward control are conform to the initial specifications and robust to modelling and identification errors. At first, some details and reference points are provided on the proposed approach.

Preliminaries

The natural approach to design of a controlled system consists in separating the design of physical system of the design of its control. This approach led us to have an over-sizing or performances lower than specifications. The mechanism is indeed sized to transmit a mechanical power (force, velocity). On top of that, to avoid any interaction with the control (resonance excitation, spill-over phenomenon (Balas, 1978)) the mechanism can be over-sizing to place the resonant frequencies high enough (of how much?), what leads to have a high mechanism inertia and therefore an oversized motor.

In the case of the mechanism is not oversized (weaker inertia, lower resonance frequencies), a control law damping the vibratory modes will be able to be used. A state space feedback control of flexible robot (Ryu et al., 2004) will prove to be often delicate to implement, what restricts its use outside laboratories and space applications.

An attractive approach in set point tracking is the use of a feed-forward term established from dynamic inverse model as it was proposed for the elastic servomechanisms (Piazzi and Visioli, 2000; Lambrechts et al., 2005), the elastic joint robots (Thümmel et al., 2005) and the flexible link robots (Feliu et al., 2006). In addition, the authors of this last article offer a modification of the dynamics of the arm of robot to be able to generate a simpler feedforward control. Mechanical modification is obtained by addition of punctual masses along the arm in order to achieve a relation specified between the frequencies of resonance (Feliu et al., 2006). These works show interest to shape the system dynamics to achieve better results with control.

In this article, following the example of works (Feliu et al., 2006), an 4th order feed-forward will be established from inverse dynamic model including the rigid body mode and the first elastic mode of the mechanism. We propose, as it was presented in introduction, to use this analytical model for design, in order to include the predominant physical phenomena which will restrict the performances of the control, as motors saturations and mechanical resonances. In brief, the purpose is to adapt at best the drive system associated at its control to achieve the specifications. The 4th order elastic dynamic model is therefore defined at the same time for design and control.

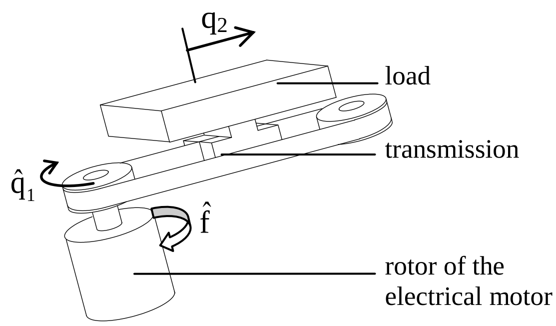

Fig.1.2.1. Mechanism example

Dynamic model and physical limits

Let consider a load with inertia m_2 and viscous friction coefficient c_2 for the motion control (linear or angular motion) such as the load position q_2 must follow a set point $q_c(t)$. The drive system of the load (see Fig.1.2.1) is constituted by an electrical motor and an elastic transmission.

Let us note for the motor: the rotor position $\hat{q}_1$, the motor force or torque $\hat{f}$, the inertial parameters $\hat{m}_1$ and the viscous friction $\hat{c}_1$. The transmission mechanism is considered such as the kinematic relation in the absence of elastic deformation is linear, i.e. $q_2 = r \cdot \hat{q}_1$, where r is the transmission ratio. The force transmitted by the mechanism, noted f_2, causes an elastic deformation of which the stiffness coefficient is noted k.
The dynamic model is described as follow:

$$\begin{cases} \hat{f} = \hat{m}_1 \ddot{\hat{q}}_1 + \hat{c}_1 \dot{\hat{q}}_1 + r\, f_2 \\ k(q_2 - r\,\hat{q}_1) = f_2 \\ f_2 = m_2\,\ddot{q}_2 + c_2 \dot{q}_2 \end{cases} \tag{1.2.1}$$

The instantaneous physical limits are imposed by the motor (its maximum velocity $\hat{v}_m$ and its maximum force or torque $\hat{\mathrm{f}}_{\mathrm{m}}$) and the mechanism (its maximum acceptable deformation Δ_e) such as:

$$\begin{cases} \hat{\mathrm{f}}_{\mathrm{m}} \geq \left| \hat{f}(t) \right| & (a) \\ \hat{v}_m \geq \left| \hat{q}_1 \right| \qquad \forall t & (b) \\ \Delta_e \geq \left| q_2 - r\hat{q}_1 \right| & (c) \end{cases} \tag{1.2.2}$$

Other limits, as the thermal limits of the motor or the technological limits peculiar to each mechanism can be also considered.

In order to use a homogeneous model at the output q_2, the variables and parameters of the motor are brought back to the load what defines new variables and parameters:

$$q_1 = r\,\hat{q}_1, \quad f = \hat{f}\,/\,r, \quad m_1 = \hat{m}_1\,/\,r^2, \quad c_1 = \hat{c}_1\,/\,r^2 \tag{1.2.3}$$

The dynamic model can be rewrite as follow:

$$f = m_1\,\ddot{q}_1 + c_1 \dot{q}_1 + k(q_1 - q_2) \tag{1.2.4}$$

$$0 = m_2\,\ddot{q}_2 + c_2 \dot{q}_2 + k(q_2 - q_1) \tag{1.2.5}$$

Control scheme

The objective of follow-up trajectory $q_c(t)$ is obtained with a feedforward based on the inverse dynamic model (IDM) of the elastic mechanism with the motor and a PD control loop to correct the modelling and identification errors (cf. Fig. 1.2.2).

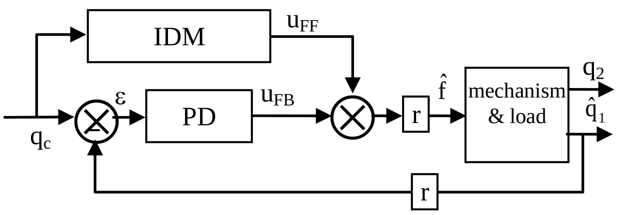

Fig. 1.2.2. Control scheme: FeedForward + FeedBack

The simplified diagram represented Fig. 1.2.2 leads to the motor force-setting $f = u_{FF} + u_{FB}$ with:

$$u_{FB} = g_p(q_c - q_1) + g_d(\dot{q}_c - \dot{q}_1) \tag{1.2.6}$$

$$u_{FF}(t) = \sum_{i=1}^{n} \tilde{p}_i \, q_c^{(i)} \tag{1.2.7}$$

The feedforward setting is therefore generated from inverse dynamic model of order n written according to the output q_2:

$$f(t) = \sum_{i=1}^{n} p_i \, q_2^{(i)} \tag{1.2.8}$$

This feedforward requires therefore, on one hand, that the position set point $q_c(t)$ is n-1 continuous derivatives and, on the other hand, that dynamic parameters $\tilde{p}_i$ are identified on-the-top. We restrict anticipation to 4th order in accordance with the model introduced in section 3.

By tacking equation (1.2.5), it is possible to give q_1 in terms of q_2 and its successive derivatives:

$$q_1 = \frac{m_2}{k}\ddot{q}_2 + \frac{c_2}{k}\dot{q}_2 + q_2 \tag{1.2.9}$$

By substituting from equation (1.2.9) q_1 and its derivatives in expression (1.2.4), we obtain the dynamic inverse model (1.2.8) to 4th order with the following parameters:

$$\begin{aligned} & p_1 = (c_1 + c_2), \;\; p_2 = \left(m_1 + m_2 + \frac{c_1 c_2}{k} \right) \\ & p_3 = \frac{m_1 c_2 + m_2 c_1}{k}, \;\; p_4 = \frac{m_1 m_2}{k} \end{aligned} \tag{1.2.10}$$

The performance-robustness in presence of modelling and identification errors will be introduced on an example in section 8.

Setting generation

The reference variable for point-to-point positioning control and trajectory following is most often calculated by a planning algorithm for which the trajectory is free in the first case, imposed in the second.

In both cases, the setting generation rests on a motion time evolution law which allows generating a reference variable of amplitude displacement of Δq for the duration T:

$$q_c = q_0 + \Delta q . Q(t) \quad \text{with } t \in [0, T] \tag{1.2.11}$$

where the set point $q_c(t)$ evolves from q_0 to $t = 0$ and $q_0 + \Delta q$ to $t = T$; Q is an interpolation function advances between 0 and 1 and at least a function of class 3.

Some examples of motion time evolution laws can be given. The first one is very practical, for all the order of feedforward, and is defined by the following velocity law (B): $\dot{Q} = 2/T \sin^2(\pi t/T)$.

A motion time evolution law in position defined by 2 polynomials of degree 4 with constant velocity phase and boundary conditions (initial and final positions) of velocity and acceleration set to zero (Khalil and Liegeois, 1984). If the initial acceleration constraint is relaxed, it is possible to define an asymmetrical motion time evolution law by the polynomial of degree 4 (D): $Q = 6\alpha^2 - 8\alpha^3 + 3\alpha^4$ with $\alpha = t/T$.

Aim of comparing these ones later, one can note the maximum magnitudes of velocity, acceleration and jerk derivative:

$$\begin{cases} \bar{v} = \max\left(|\dot{q}_c|\right) \\ \bar{a} = \max\left(|\ddot{q}_c|\right) \\ \bar{b} = \max\left(\left|q_c^{(4)}\right|\right) \end{cases} \tag{1.2.12}$$

As well as the homogeneous variable according to a frequency (first harmonic approximation of the motion time evolution law) such as:

$$\varpi = \sqrt{\bar{b}/\bar{a}} \tag{1.2.13}$$

The variables $\bar{v}$, $\bar{a}$ and ϖ used to characterize the motion time evolution law will be justified by equation (1.2.14) in following paragraph. These ones are used as design specifications of drive system and constitute dynamic solicitation indicators for a given movement. For instance, let us compare a bang-bang acceleration law – which includes a constant velocity phase – with the motion time evolution laws introduced before. A displacement of 1 m must be achieved in 1 s, initial and final velocities being set to zero.

The Tab. 1.1 provides indications on the motor solicitation $(\bar{a}, \bar{v})$ and transmission solicitation (ϖ).

Tab.1. Motion time evolution laws characteristics

□ q = 1 m T = 1 s	(A) bang -bang	(B) velocity in sinus²	(C) two polynomials of degree 4	(D) one Polynomial of degree 4
$\bar{v}$ (m/s)	2	2	2	1.7
$\bar{a}$ (m/s²)	4	6.2	6	12
ϖ (Hz)	∞	6.2	5.6	2.4

Of course, the bang-bang acceleration law is not usable for the 4th order feedforward (i.e. $\varpi \to \infty$), but this law is the one which solicits the least the drive system in acceleration. We can point out for instance that law (C) is less demanding than the law (B). Law (D) is less demanding for the mechanism than the law (C) but not for the motor.

Formulation for the design

The dynamic inverse model is used for the drive system design by neglecting the terms $c_1 c_2 / k$ and $(m_1 c_2 + m_2 c_1)/k$ very small in front of the other terms, the stiffness k being necessarily high and the friction coefficient c_2 is assumed small. In addition, the inertial parameters $\hat{m}_1$, the motor viscous friction $\hat{c}_1$ and the transmission ratio r are expressed in the model from relations (3) to be able to size the motor and the transmission mechanism:

$$\hat{f} \approx \frac{\hat{m}_1\, m_2}{k} q_2^{(4)} + \left(\frac{\hat{m}_1}{r} + r\, m_2 \right) \ddot{q}_2 + \left(\frac{\hat{c}_1}{r} + r\, c_2 \right) \dot{q}_2 \tag{1.2.14}$$

The characteristics of the motion time evolution law $\bar{v}$, $\bar{a}$ and ϖ, equations (1.2.12) and (1.2.13), are used with physical limits (1.2.2) to express the design constraints. By taking back the motor force or torque limit (1.2.2a) with equation (1.2.14), the first design constraint (1.2.15a) can be expressed. The second constraint (1.2.15b) is obtained with $\hat{v}_m$, the velocity limit (1.2.2b), and the kinematics relation of the mechanism $q_2 = r\,\hat{q}_1$ without considering its elastic deformation. The third constraint (1.2.15c) concerns the maximum acceptable deformation Δ_e of the mechanism (1.2.2c) expressed in terms of characteristics of the motion time evolution law thanks to equation (1.2.7).

$$\begin{cases} \hat{f}_m \geq \left(\dfrac{\hat{m}_1}{r} + r\, m_2 \right) \left(1 + \varpi \dfrac{\hat{m}_1 m_2}{k\left(\hat{m}_1 + r^2 m_2\right)} \right) \bar{a} & (a) \\ \qquad\qquad + \left(\dfrac{\hat{c}_1}{r} + r\, c_2 \right) \bar{v} & \\ \hat{v}_m \geq \bar{v} / r & (b) \\ \Delta_e \geq \dfrac{m_2}{k} \bar{a} + \dfrac{c_2}{k} \bar{v} & (c) \end{cases} \tag{1.2.15}$$

These three design constraints establish a relation between:
- Performance specifications: m_2, $\bar{v}$, $\bar{a}$ and ϖ.
- 4 linked parameters (not independent) for the motor: $\hat{v}_m$, $\hat{f}_m$, $\hat{m}_1$ and $\hat{c}_1$.
- 3 linked parameters for the transmission: r, k, Δ_e.

To be able to select and to size the components, it is necessary to regroup these linked parameters, in inequations (1.2.15), in characteristic variables of components. The optimum transmission ratio r_{opt} (optimum in term of acceleration) is introduced to show an adimensional form with transmission factor r^* (Pasch and Seering, 1984):

$$r^* = r / r_{opt} \quad with \quad r_{opt} = \sqrt{\hat{m}_1 / m_2} \tag{1.2.16}$$

The choice of a factor $r^* = 1$ maximizes acceleration for a motor given. By taking back the formulation principle established for the choice of a motor and a transmission ratio (Dequidt et al., 1999), a collection of parameters is achieved thanks to a metric defined by the motor and load kinetic energies, respectively Ec_1 and Ec_2:

$$\begin{aligned} Ec_1 &= \frac{1}{2}\dot{x}_1^2 \quad with \;\; \dot{x}_1 = \sqrt{\hat{m}_1}\,\dot{\hat{q}}_1 \\ Ec_2 &= \frac{1}{2}\dot{x}_2^2 \quad with \;\; \dot{x}_2 = \sqrt{m_2}\,\dot{q}_2 \end{aligned} \tag{1.2.17}$$

The motor power output, with effective force or torque $\hat{f}_e(t) = \hat{f}(t) - \hat{c}_1\dot{\hat{q}}_1$, is rewritten with $\dot{x}_1$ to show new variable $F(t)$:

$$\hat{f}_e(t)\,\dot{\hat{q}}_1 = F(t)\,\dot{x}_1 \quad \Rightarrow \quad F(t) = \frac{\hat{f}(t) - \hat{c}_1\dot{\hat{q}}_1}{\sqrt{\hat{m}_1}} \tag{1.2.18}$$

Two homogeneous variables $\dot{x}_1$ and $\dot{x}_2$ (SI unit: $J^{1/2}$) are introduce and their temporal derivative $\ddot{x}_1$ and $\ddot{x}_2$ are homogeneous to F (t) (SI unit: $J^{1/2}\,s^{-1}$), what allows us to rewrite the constraints (1.2.15) under an adimensionnal form:

$$\begin{cases} \dfrac{C_a}{C_c} \geq \left(r^* + \dfrac{1}{r^*}\right)\left(1 + \left(1 + r^{*2}\right)^{-1}\Omega^{-1}\right) + r^* f^* & (a) \\ r^* \geq r^*_{ciné} & (b) \\ \Omega \geq (1 + f^*)\Omega_e & (c) \end{cases} \tag{1.2.19}$$

With the motor constant C_a=max(|F(t)|), the load constant C_c=max($|\ddot{x}_2|$), the minimum transmission factor $r^*_{ciné}$, the external force factor (here the friction force applied to the load) f^*, the transmission stiffness factor Ω with the anti-resonance first pulse $\omega_z = \sqrt{k / m_2}$ of mechanism and the minimum stiffness factor Ω_e, such as:

$$C_a = \frac{\hat{f}_m - \hat{c}_1\hat{v}_m}{\sqrt{\hat{m}_1}} \qquad C_c = \sqrt{m_2}\,\bar{a}$$

$$r^*_{ciné} = \frac{\sqrt{m_2}\,\bar{v}}{\sqrt{\hat{m}_1}\,\hat{v}_m} \qquad f^* = \frac{c_2\bar{v}}{m_2\bar{a}}, \tag{1.2.20}$$

$$\Omega = \frac{\omega_z^2}{\varpi^2}, \qquad \Omega_e = \frac{1}{\varpi^2}\frac{\bar{a}}{\Delta_e},$$

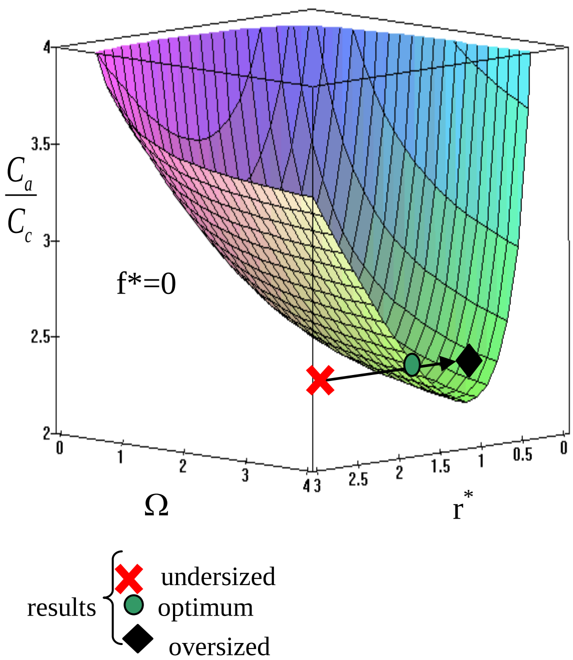

Fig. 1.2.3. Design constraints (1.2.19a)

If we take for instance the constraint (1.2.19a), optimum is defined by a Pareto front (see Fig. 1.2.3) in a three-dimensional space; every axle takes back one or two components characteristic variables: the ones of the mechanism, Ω and r^*, and the ones of the motor, C_a and $r^*_{ciné}$ (constraint 1.2.19b). The graphs representing constraints (1.2.19) are plotted using a null value of the friction coefficient c_2 ($f^* = 0$).

Selection and component sizing

Constraints (1.2.19) can therefore be used to perform an optimum selection and sizing method of the electromechanical components, either from an automatic minimization method (criterion: mass, cost), or from an interactive method. These one can be made such as the designer is able to choose the most suitable solution closed to the Pareto front (optimum solutions set). We offer in this paper to describe a two-step design method (the selection of the motor and then the sizing of the mechanism) from graphs defined with constraints (1.2.19).

From graph of Fig.1.2.4 (torque lower limit $C_a/C_c - r^*$ for different minimum values of Ω), it is possible to choose a motor by knowing the consequences of size on stiffness Ω as well as to the feasible range $\left[r^*_{ciné}, r^*_{\sup}\right]$ of the transmission factor r^* (the line in bold for $\Omega = 4$, see Fig. 1.4).

Once the motor is chosen (motor load factor $C_a/C_c = 2.5$), it is possible to size the transmission by referring to graph of Fig. 1.2.5. The minimum stiffness factor Ω is pointed out according to the transmission factor r^*. On the other hand, the minimum stiffness factor corresponding to the maximum distortion Δ_e is represented for solicitations on the motor side $\hat{\Omega}_e$ and on the load side Ω_e .A feasible domain (pointed out D on the Fig. 1.2.5.) is defined for the choice and the sizing of the mechanism components.

For an example of drive system design, the reader could see some works found in the literature (Valdès et al.).

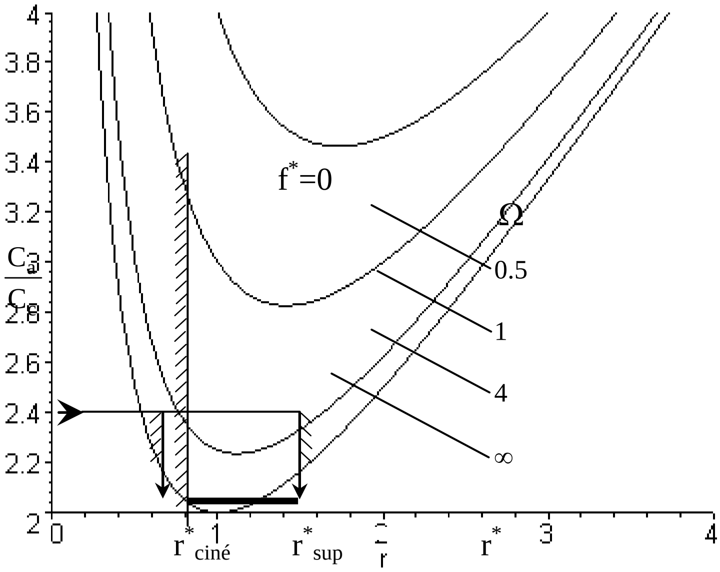

Fig. 1.2.4. Choice of the motor: constraints (19a, b)

Let us take the example of a load of 120 kg moving according to the velocity law in sinus² (B), cf. design specifications in the Tab.1.2.
Furthermore the dynamic behavior of the drive system which satisfies the design constraints (1.2.19) is simulated with the proposed feed-forward control law.

Tab.1.2. Design specifications

m_2 (kg)	Δq (m)	T (s)	$\bar{v}$ (m/s)	$\bar{a}$ (m/s²)	ϖ (Hz)
120	0.05	0.1	1	31.42	62.83

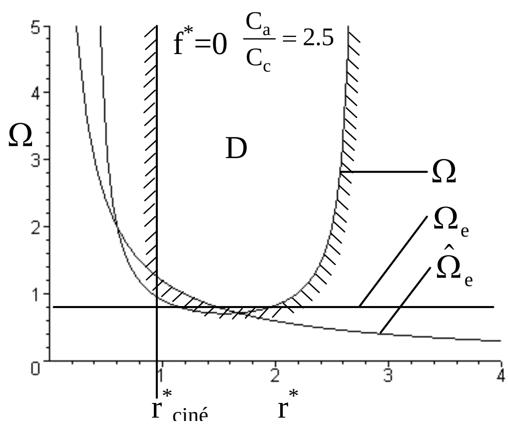

Fig.1.2.5. Transmission sizing (1.2.19a, b, c)

The dynamic behavior of the drive system with the 4th order feed-forward control is compared with a control without feed-forward (PD controller), cf. Fig. 1.2.6, 1.2.7 and 1.2.8. The Fig. 1.2.6 shows the setting and responses functions in position with and without feed-forward.
With feed-forward, the tracking error (cf. Fig.1.2.8) reaches a maximum of $2.6*10^{-4}$m (0.5% de Δ_q), while it is 6.5 times bigger ($1.67*10^{-3}$m) for a control law without feed-forward.

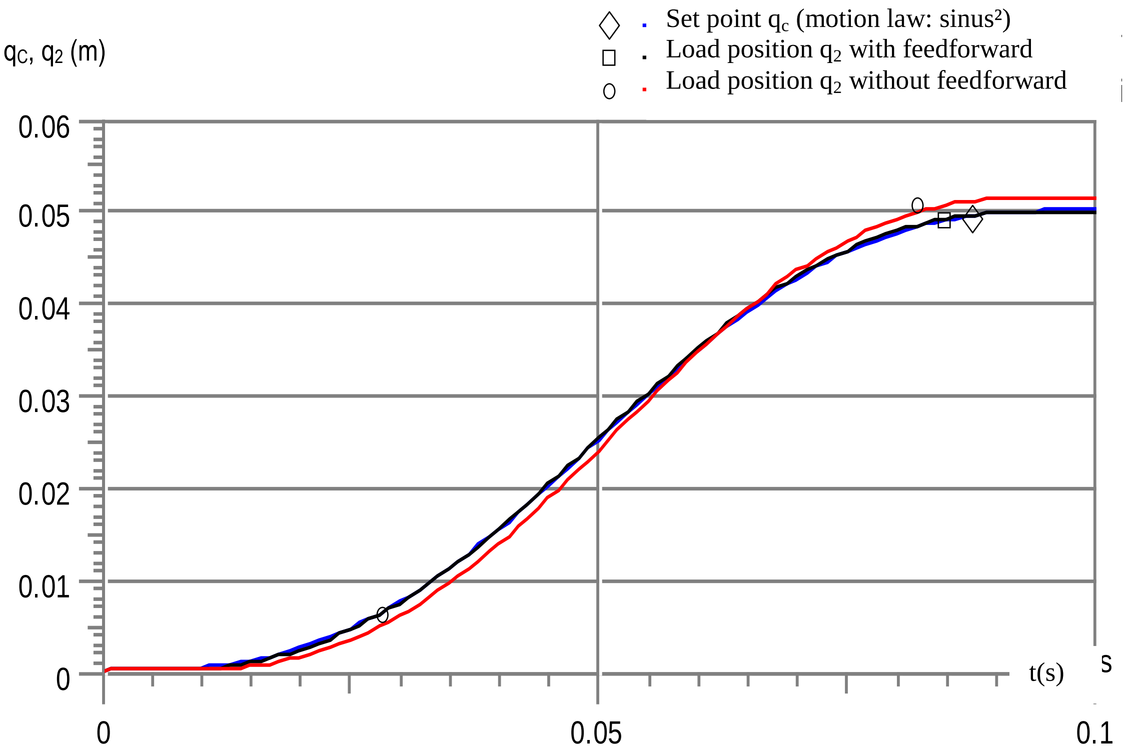

Fig. 6. position set point q_c and load response q_2 without and with feedforward

When an identification error of 50 % is made on the load ($\tilde{m}_2 = 1.5 m_2$), the tracking error is about twice bigger (5.5*10^{-4}m); this error is equal to 3.5*10^{-4}m when the stiffness is divided by 10 ($\tilde{k} = k/10$).
In brief, the performances are robust to identification errors. Moreover, the design constraints allowed (1.2.19) to size correctly the mechanism.
Let us notice that an overestimate of maximum motor force is imposed by the control law: 10254 N instead of 6340 N (cf. Fig.1.2.7). This is due to the "pessimist" approximation given by the extreme (1.2.12) of the dynamic solicitation.

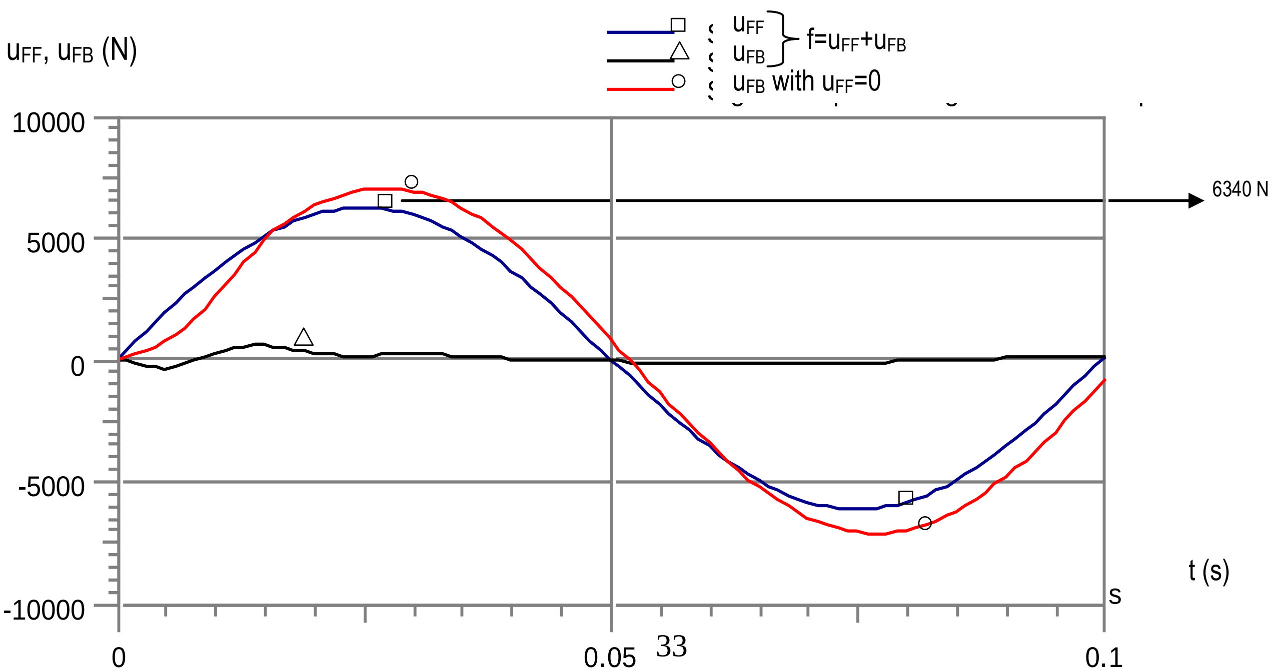

Fig. 1.2.7. Comparison between motor force command signals for feedforward + feedback control

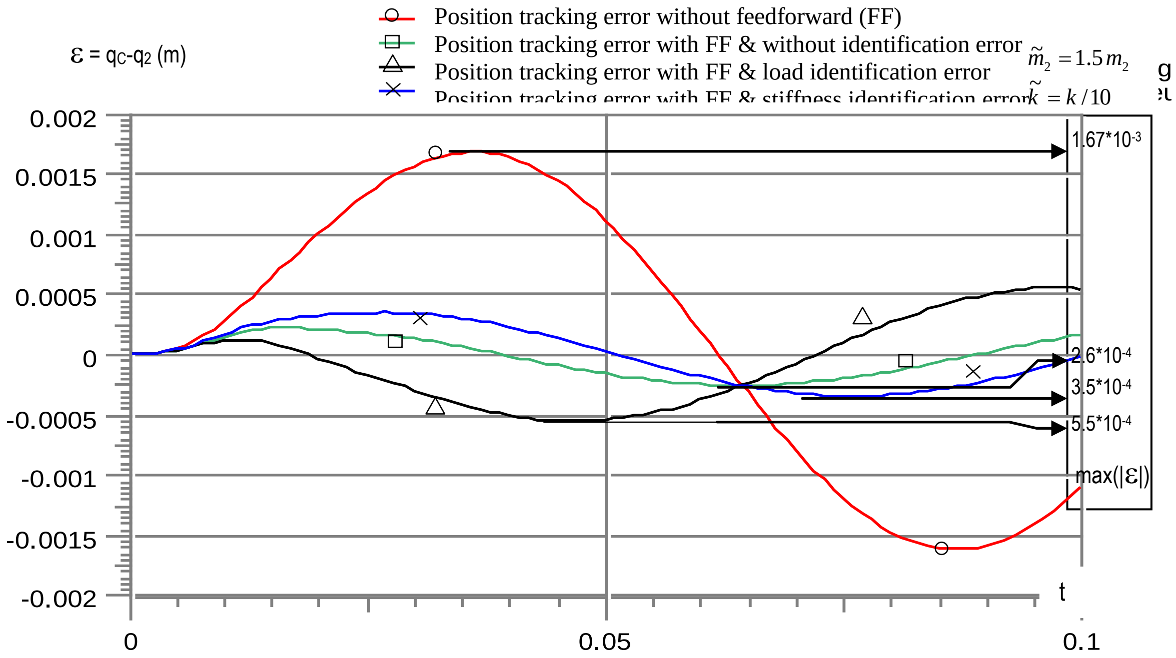

Fig. 1.2.8. Comparison between position tracking errors without and with feedforward

Conclusıon

A mechatronic design method of flexible drive systems dedicated to feed-forward control has been proposed. The originality of the approach is to enter at the same time the question of design and the control to reach performance requirements. This approach uses the inverse elastic dynamic model of the drive system and its physical limits (motor saturations, mechanical resonance, etc.). It results at one and the same time a 4th order feed-forward control law and the design constrain of the drive system written under an adimensional form. The optimum design problem is represented by a Pareto front (optimum solutions set) in a three-dimensional space. Based on this formulation, a two-step design method has been established and applied to an example in order to show the characteristics (performances and robustness) of a design solution.

One of the extensions of this approach of total optimization concerns nonlinear mechanisms and arms of flexible robot with, in this case, a particular attention to non minimal-phase behaviour.

Acknowledgments

This work was supported in part by International Campus on Safety and Intermodality in Transportation, the European Community (through the FEDER European Funds for Regional Development), the Délégation Régionale à la Recherche et à la Technologie, the Ministère de l'Enseignement supérieur et de la Recherche, the région Nord Pas-de-Calais and the Centre National de la Recherche Scientifique (CNRS): the authors gratefully acknowledge the support of these institutio

Chapter 2 Haptic interfaces

A haptic interface is a robotic system that creates communication between a human operator and a virtual environment. It allows users to manipulate objects in a virtual environment with the sense of touch and the kinesthetic perception. However, unlike the robot manipulators, the haptic interfaces should have both high force rendering and maximum dynamic transparency (low inertia, low friction, etc.). Therefore, the stability of haptic interface is a key issue. Any unstable behavior arising during dynamic interaction may either damage this device or even injure the human operator.

2.1 Haptic system modeling and stability analyze

Haptics is a enhancement to virtual environments allowing users to “touch” and feel the simulated objects with which they interact. Haptics is the science of touch. The word derives from the Greek haptikos meaning “being able to come into contact with”. The study of haptics emerged from advances in virtual reality. Virtual reality is a form of human-computer interaction (as opposed to keyboard, mouse and monitor) providing a virtual environment that one can explore through direct interaction with our senses. To be able to interact with an environment, there must be feedback. This type of feedback is called haptic feedback.

Haptic devices (or haptic interfaces) are mechanical devices that mediate communication between the user and the computer. Haptic devices allow users to touch, feel and manipulate three-dimensional objects in virtual environments and tele-operated systems. Haptic devices are input-output devices, meaning that they track a user's physical manipulations (input) and provide realistic touch sensations coordinated with on-screen events (output).

In human-computer interaction, haptic feedback means both tactile and force feedback. Tactile feedback allows users to feel by her skin things such as the texture of surfaces, temperature and vibration. Force feedback reproduces directional forces that can result from solid boundaries, the weight of grasped virtual objects, mechanical compliance of an object and inertia.

Haptics has great potential in education-feeling is believing-especially in science and technical areas, where forces are very important, being able to interact with a computer simulation and literally feel the results. Students can develop a feel and appreciation for a particular subject even before they have the detailed mathematics background to handle the equations.

INTRODUCTION

Nowadays, there are two main approaches to study the stability of a haptic device. In the first approach, some authors have modeled the haptic interface like a sampled data system. In this approach, the virtual environment is as a part of a model which consider the virtual impedance as a linear spring and damper. The stability region is represented by the stability boundaries in the stiffness-damping plan of the virtual impedance.

Hereby, the influence of parameters such as physical damping and time delay on stability boundaries is evaluated with the transfer function model. The human arm impedance can be included in the model to study the human dynamic effects. In the second approach, the theory of two ports network is applied to separate the virtual environment and the human operator into two

independent parts. Then, the virtual coupling is used and the stability is ensured through the passivity or the unconditional stability criteria.
There are a lot of works concerning the interaction of the human operator, the haptic device and the virtual environment. However, there remain still many challenges such as expanding the boundary of stable region, modeling of human operator, designing of a control law, etc. In previous studies, haptic device was modeled as a mechanical system including a mass or an inertia, and a physical damper.

The existence of internal vibration of haptic device has been studied in recent years, but these researches have not clearly explained the influence of the vibration mode parameters. The goal of this paper is to analyze the influence of the dynamic parameters on the stability boundaries based on the first approach. This study shows how the factors such as physical damping coefficient, time delay and human operator parameters affect the stability of a haptic system. The influence of vibration mode on the stability is also studied in this paper. The results are carried out on the physical parameters of a PHANTOM®.

Haptic interfaces are relatively sophisticated devices. As a user manipulates the end effector, grip or handle on a haptic device, encoder output is transmitted to an interface controller at very high rates. Here the information is processed to determine the position of the end effector. The position is then sent to the host computer running a supporting software application. If the supporting software determines that a reaction force is required, the host computer sends feedback forces to the device.

Actuators (motors within the device) apply these forces based on mathematical models that simulate the desired sensations. One haves been developing haptic interface devices to permit touch interactions between human users and remote virtual and physical environments. Examples of haptic devices include consumer peripheral devices equipped with special motors and sensors (e.g., force feedback joysticks and steering wheels) and more sophisticated devices designed for industrial or scientific applications (e.g., PHANTOM haptic device, FALCON haptic device).

Phantom haptic device

The Phantom is a convenient desktop device which provides a force-reflecting interface between a human user and a computer. Users connect to the mechanism by simply inserting their index finger into a thimble. The Phantom tracks the motion of the user’s finger tip and can actively exert an external force on the finger, creating compelling illusions of interaction with solid physical objects. A stylus can be substituted for the thimble and users can feel the tip of the stylus touch virtual surfaces.

The Phantom Premium 6DOF is a desk mounted force feedback system that provides six degree-of-freedom force and torque feedback. The system consists of the device itself, accompanying power electronics, a remote/safety switch, and a PCI interface card. The Phantom Premium 3.0 is a serial robot with 6 DOF. The device is leveraged from haptic technology developed and implemented in standard Phantom force feedback devices [Fig. 2.1].
The Phantom Premium 3.0 is a large haptic device that accommodates free arm and hand motions about the shoulder as the center of rotation. The Phantom Premium 3.0 base uses cable capstan drive trains. Its motors are equipped with co-located position sensors. The base motor (primarily responsible for left-right motions or x motions) is grounded. A cable transmission converts motor rotations to the rotation of a large disk which carries the rest of the mechanism.

The second and third motors (primarily responsible for up-down and in-out motions, or y and z motions) ride on a ring mounted on the large disk, using a cable-capstan drive. The second and third motors control the position of two linkages in a four-bar parallel linkage design. The sensor-motor packages measure the endpoint position and provide force feedback in three translational degrees of freedom.

The end effector takes the shape of a handle and measures less than an inch in diameter. It is roughly the size and shape of a large permanent felt tip marker. It sports one switch which can be programmed to produce visual and/or haptic effects.

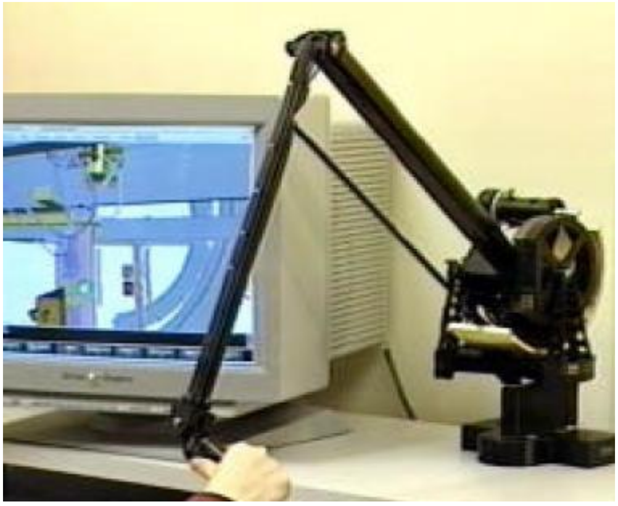

Fig. 2.1.1 Phantom Premium 6DOF haptic device

The 6DOF device is driven by a 6-axis power amplifier box and interfaces to a Pentium based Windows NT computer via a PCI controller card. Low level communications between the PC and the PCI card are handled by the Phantom Device drivers (PDD). The PDD maintains a 1 kHz servo update rate to ensure stable closed loop control of the device. The device kinematics and other robotic calculations are handled within the PDD. High level force and position calculations are provided by the the Software Developer's Kit (SDK). The SDK provides a high level C++ programming interface for generating haptic effects.

Haptic effects handled by the SDK can be based on geometry (such as point haptic exploration), or on force-time profiles (such as sinusoidal vibrations and jolts). The user can define their owned custom force fields. Currently the SDK only supports 3DOF forces based on point haptic exploration. Using new extensions allow a 6DOF application to read the Denavit-Hartenberg homogeneous transform, describing the global position and orientation of the end effector, calculate 6DOF forces and torques in the application, and command these 6DOF global forces and torques to the device.

Falcon haptic device

The Novint Falcon is a pointing device with three arms coming out of the main body of the device. Each arm is capable of pulling and pushing, and is updated every 1/1000 of second. In addition to the arms there is an interchangeable handgrip, with three programmable buttons on it.

Fig. 2.1.2 Novint Falcon haptic device

Falcon haptic device is a Delta parallel robot. Parallel mechanisms are a natural fit for haptics, because they offer good stiffness and robustness at a relatively low cost. Stiffness in particular, is important in haptics, because motors can't respond instantaneously.

A grip's position in real space corresponds to the location of a cursor in the three-dimensional virtual world. And when that cursor touches a virtual object, the Falcon device sends an analogous force back to the grip. Likewise, forces that the user exerts on the grip are translated into cursor movements.

The Falcon haptic device generates its real world forces with three small joystick-sized motors connected to the device's three arms, by a cable linkage. The system can produce a force in any direction, within its 4 x 4 x 4 inch working envelop.
Novint's control software for the Falcon adjusts the motor current - in accordance with what's happening in the virtual world - with textbook physics equations, determining the magnitude and vector off all the forces and deflections. The system updates current positions - real and virtual - at 1 kilohertz. Falcon system achieves a sub-mm resolution.

The Falcon may be getting its start as a game controller, but the Falcon isn't just one device, it's a haptic platform that could be applied more broadly, as a 3D input/output device for a variety of computing systems, including design and analysis software used by engineers.

HAPTIC SYSTEM MODELING

A haptic device is composed of two principal parts: the electromechanical part – containing actuator and mechanical transmission – and the controller part – which is a link between the physical device and the Virtual Environment (VE). Moreover, there are two challenges in the design and control of haptic device. The first challenge is to guarantee the stability when a user manipulates an object put in contact with a virtual wall (constrained motion, cf. Fig. 2.1.3).

Taking a high value for the virtual stiffness to represent a stiff wall can lead to an instable behavior. The second challenge is the transparency of haptic device when a user manipulates an object in a free area (i.e. when there is no contact with the virtual wall, cf. Fig.2.1.3). In theory, it is advisable that the human operator don't feel the inertia and the viscous friction (damping) of the device in order to have no distortion of the force rendering during virtual object manipulation.

In order to design and to control a haptic system, first of all, its basic model should be established. A 1-DOF (degree-of-freedom) mechanical interface is usually modeled as a linear rigid mechanism including a mass or an inertia (m), a physical damper (b) i.e. a viscous friction (cf. Fig. 2.1.3).

The arm position X2 corresponds to the actuator position X1 without elastic deformation of mechanical transmission: X2 = X1. Impedance-based virtual environment for constrained motion is a coupling system including linear spring with virtual stiffness (K) and a virtual damping coefficient (B). The resulting force F1 is applied by the motor on the mechanical transmission which is opposed by the force F2 applied by the human operator.

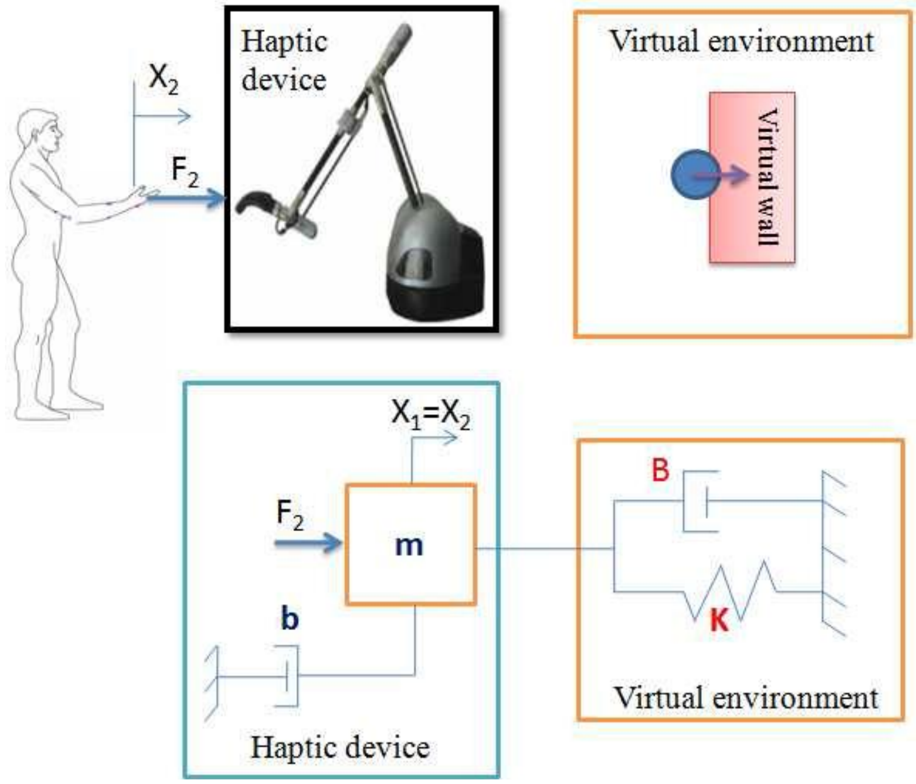

Fig. 2.1.3 Free motion in VE: transparency challenge

It is assumed that some nonlinear phenomena (like sensor resolution, actuator saturations, etc.) are negligible. In particular, dry friction and sensor quantization can be ignored. The sampled-data and the time delay due to the computer part (controller, virtual environment computation, and communication) are included in the haptic interface model.

With the sampling process and the zero-order holder (ZOH), the overall model is a hybrid one with both continuous and discrete forms. In the following, this model is used in a purely discrete form, cf. Fig.2.1.4.

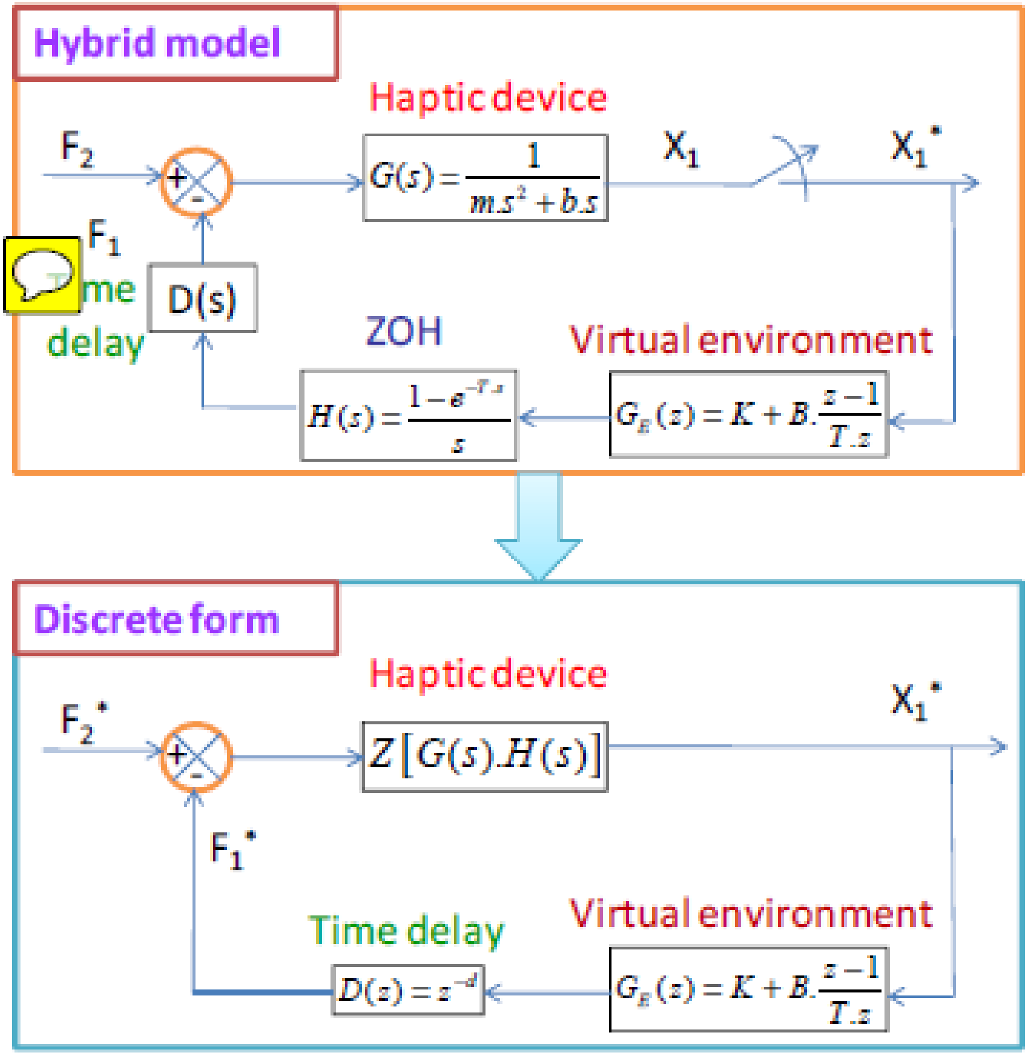

Fig. 2.1.4 Dynamic model of haptic interface and its discrete form

METHODOLOGY

In this part, two methods are presented to find the stable region of haptic device which are the Routh- Hurwitz criterion and the gain margin. First, the characteristic equation of haptic system is derived from the hybrid dynamic model (cf. Fig.2.1.4). The transfer function of haptic interface is defined as:

$$\frac{X_1^*}{F_2^*}=\frac{Z[G(s).H(s)]}{1+D(z).G_E(z).Z[G(s).H(s)]}. \tag{2.1.1}$$

The rigid model of the mechanical interface G(s), the virtual interface GE(s), the ZOH H(s) and the time delay D(s) are formulated as:

$$G(s)=\frac{1}{m.s^2+b.s}, \qquad G_E(z)=K+B.\frac{z-1}{Tz}$$
$$H(s)=\frac{1-e^{-T.s}}{s}, \qquad D(z)=z^{-d}. \tag{2.1.2}$$

where Z[.] is the Z-transform operation of the function into bracket, and * is signed by discrete signal. The characteristic equation of this system is:

$$1+D(z).G_E(z).Z[G(s).H(s)]=0. \tag{2.1.3}$$

Effect of human operator model

In this part, we study the influence of human operator to the stability of haptic system. The human operator is usually modeled as a linear second order system including a mass M_H coefficient B_H and a stiffness coefficient K_H impedance is characterized by:

$$Z_H(s)=M_H.s+B_H.s+K_H \tag{2.1.4}$$

This human model is coupled with the device model seen in Fig.2.1.5. In order to study the influence of human operator on the boundary of stable region, we used parameters in Table II which correspond to several models.

The Fig. 2.1.6 shows the stability boundaries ofthe PHANTOM® including human operator models of some authors. As pointed out, these results show that human operator will be able to stabilize the system and the smallest stable region is obtained without human model. Therefore, in the worst case, the human operator part should be ignored in the model for stability analysis.

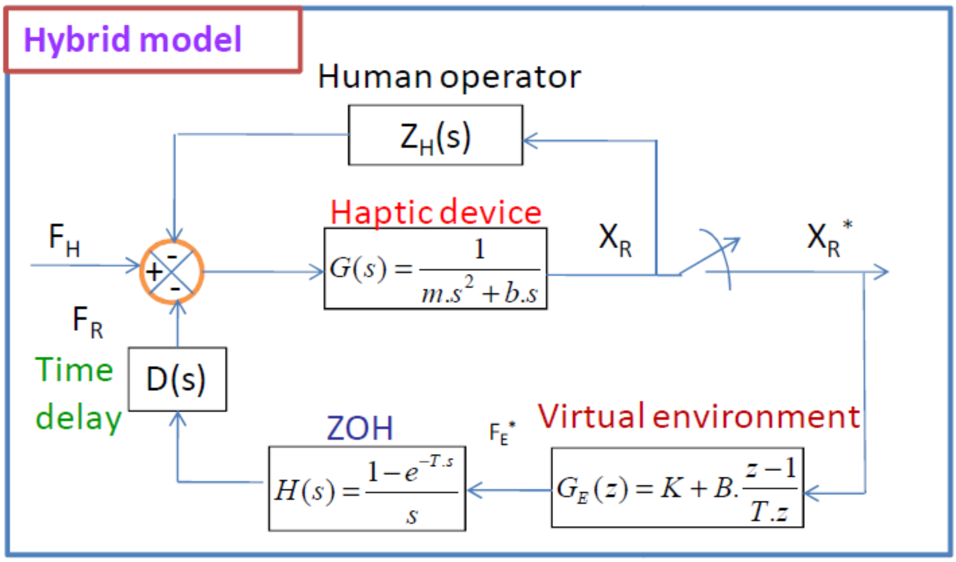

Fig.2.1.5 Dynamic model of haptic system with human operator

TABLE II [13]
PARAMETERS OF HUMAN OPERATOR MODELS

Author	M_H (kg)	B_H (Ns/m)	K_H (N/m)
Hogan	0.8	5.5	568
Yokokohji	2	2	10
Kazerooni	4.54	6.83	12.5

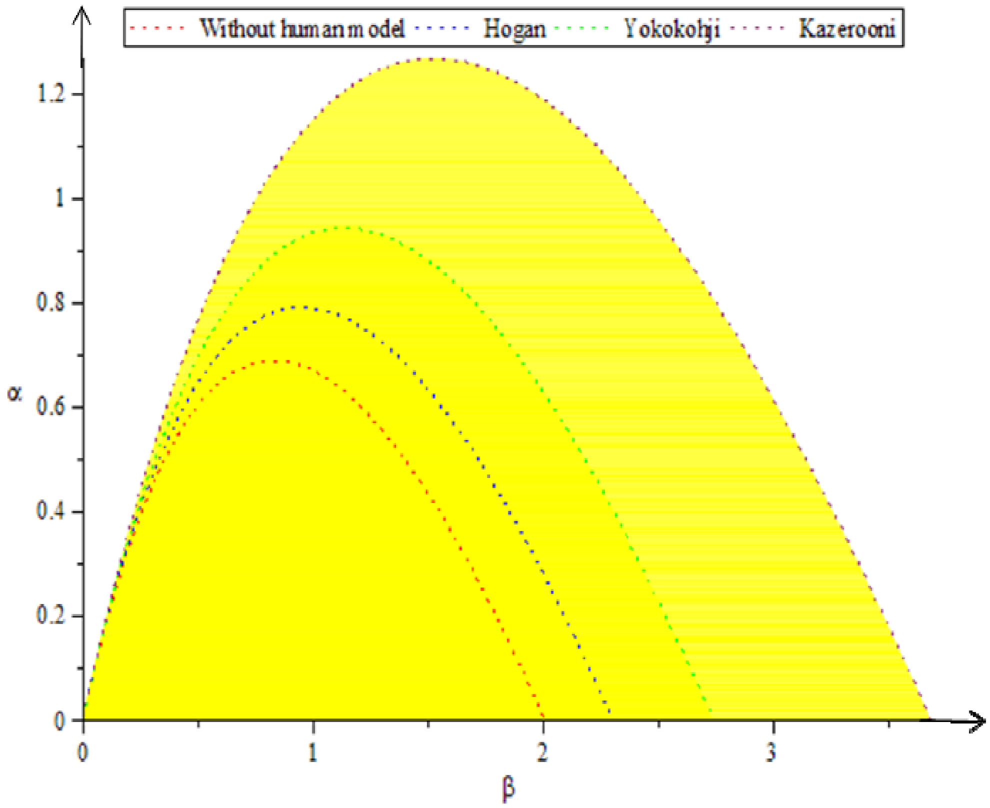

Fig.2.1.6. Stability boundaries for human operator models

EFFECT OF VIBRATION MODE

In previous studies, a rigid model of haptic used to analyze the stability. In fact, there exists an elastic deformation of mechanical transmission components between the motor shaft (position X1) and the device arm (position X2). Consequently the resulting vibration mode can modify the stability boundaries. A more complete model of haptic device with an internal vibration mode is shown in Fig.2.1.7. The relationship between outputs positions and input forces for this haptic device which has been defined by transfer function is presented here.

Similar to the Fig.2.1.4, the transfer function of sampled output position, X^*_1, and input force F^*_2 are given by (2.1.1) with the transfer function G(s) defined by (2.1.4). Note that, in this part, we only study the haptic system with vibration mode and without time delay.

$$G(s)=\frac{X_1(s)}{F_1(s)}=\frac{1}{m.s^2+b.s}\cdot\frac{w_r^2.(s^2+c_a.s+w_a^2)}{w_a^2.(s^2+c_r.s+w_r^2)}$$

(2.1.5)

The critical virtual stiffness is determined with D(z)=1 (d=0). The gain margin method is used to compute the stability boundaries for several different values of anti-resonant frequency w_a. Obviously the other physical parameters are considered to be constant. The results are based on the physical parameters of the first vibration mode of the PHANTOM® (cf. Table III).

Matlab® is used to compute the gain margin. This method is very useful to reconstruct the stability boundaries of haptic devices even if its transfer function is very complex. In contrast, the first method becomes unusable while the order of characteristic equation is high.

TABLE III [17]

PHYSICAL PARAMETERS OF PHANTOM

Parameter	Variable	Value
Inertia	m	1.168 gm^2
Physical damping	b	0.0054 Nms/rad
Anti-resonant frequency	w_a	417.612 rad/s
Damping coefficient	c_a	80 Nms/rad
Resonant frequency	w_r	479.166 rad/s
Damping coefficient	c_r	83 Nms/rad

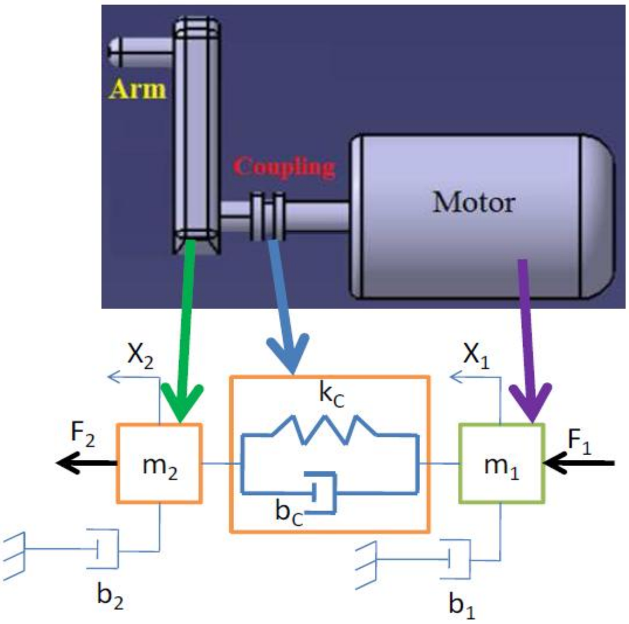

Fig.2.1.7 Haptic device model with internal vibration mode

The Fig. 2.1.8 shows that the stability boundary is reduced when the anti-resonant frequency decreases in the range of $w_a \leq 369$ rad/s. From the Bode diagrams of the discrete transfer functions *Z* [*G*(*s*).*H*(*s*)] with this range of anti-resonant frequency, as seen in Fig.2.1.9, it is obvious that the higher gain margin of the system is obtained for the higher value of anti-resonant frequency.

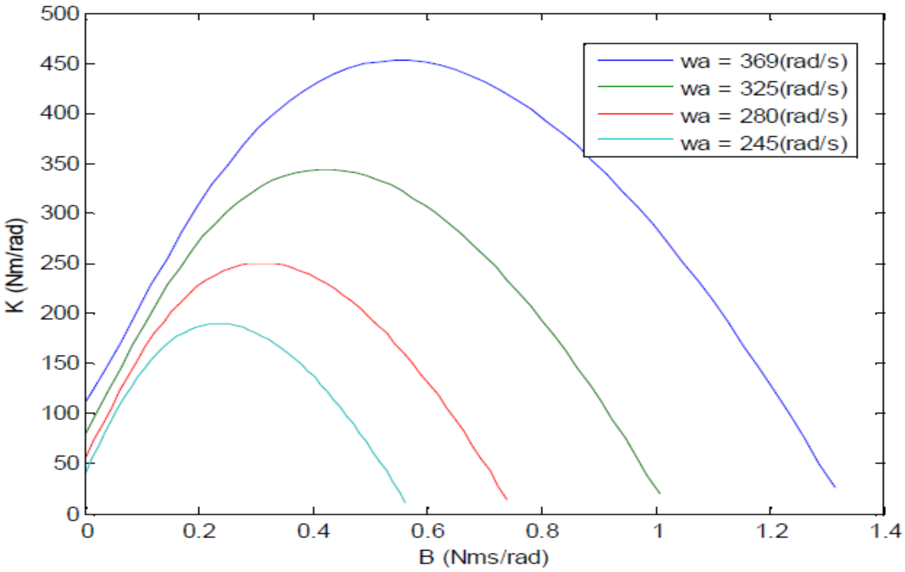

Fig.2.1.8. Effect of anti-resonant frequency for $w_a \leq 369$ rad/s

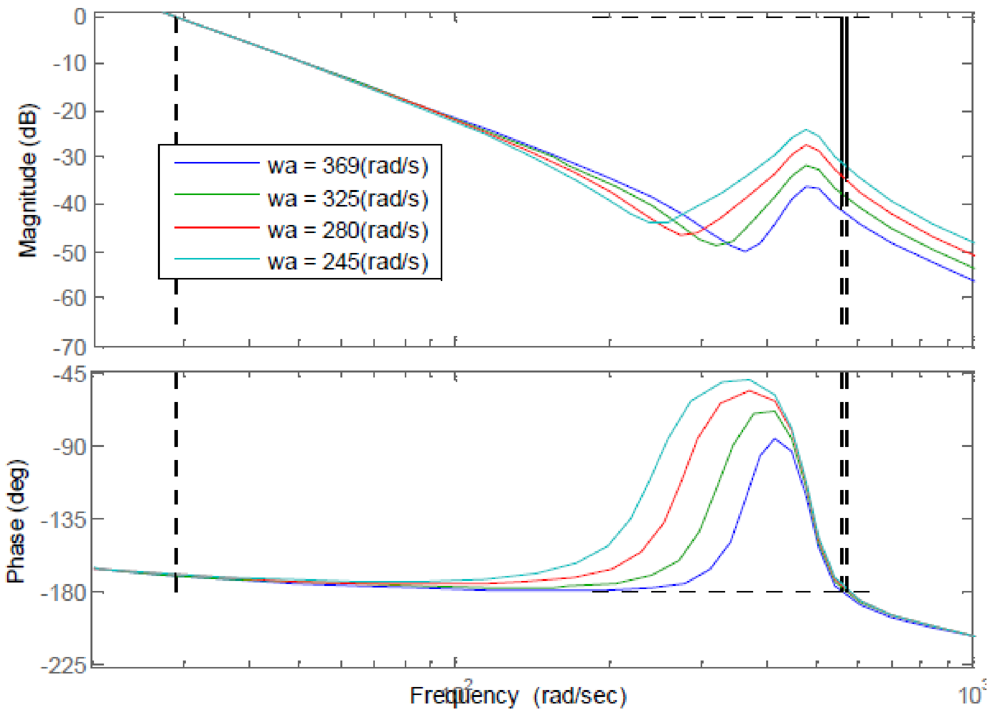

Fig.2.1.9 Bode diagram of Z [G(s).H(s)] for $w_a \leq 369$ rad/s

This paper presents an overview on the effects of dynamic factors on haptic interface stability. The impedance model used to compute the force feedback includes a virtual stiffness and a virtual damping.

This paper has provided engineers some basic guidelines in choosing design specifications of haptic system. In future works the effect of others physical parameters (e.g; time delay, human model) will be studied with a model included vibration modes.

ACKNOWLEDGMENT

This work was supported in part by the International Campus on Safety and Intermodality in Transportation, the European Community (through the FEDER European Funds for Regional Development), the Délégation Régionale à la Recherche et à la Technologie, the Ministère de l'Enseignement supérieur et de la Recherche, the région Nord Pas-de-Calais and the Centre National de la Recherche Scientifique (CNRS). The authors gratefully acknowledge the support of these institutions.

2.2 Hybrid force/position control of the scissors-robot

In this paper a method for displaying feeling of cutting, using a virtual scissors type, is described. A new computer haptics algorithm to be used in general interactive manipulations of deformable objects is presented. The algorithm was developed to simultaneously display translational and rotational (cutting) displacement for a realistic cutting simulation. We showed that interactive animations can be used to simulate the functioning of force/torque - feedback scissors devices. This paper discusses both the experience with the design process in the virtual environment and the novel features of the possible portable torque-feedback scissors devices. We will present an approach for designing a more natural user interface, without resorting to various special haptic input/output devices.

Haptics are emerging as effective interaction aids for improving the realism of virtual worlds. To be able interact with virtual environment; the user should be able to touch a virtual object and feel a response from it. This type of feedback is called haptic feedback. The confluence of several emerging technologies made virtualized haptics, or computer haptics, possible.

Much like computer graphics, computer haptics enables the display of simulated objects to humans in an interactive manner. However, computer haptics uses a display technology through which objects can be physically palpated. This new sensory display modality presents information by exerting controlled forces on the human hand through a haptic interface (rather than, as in computer graphics, via light from a visual display device). These forces depend on the physics of mechanical contact. The characteristics of interest in these forces depend on the response of the sensors in the human hand. Although these haptic interface devices generate reactive forces when their users grasp or touch some objects with a device indirectly, the contact type device represents a tactile sense by touching objects directly with finger.

Unlike computer graphics, haptic interaction is bidirectional, with energy and information flows both to and from the user. The combination of high-performance force-controllable haptic interfaces, computational geometric modelling and collision techniques and an understanding of the perceptual needs of the human haptic system allow us to assemble computer haptic systems that can display objects of sophisticated complexity and behavior. Therefore, they must allow an as natural and intuitive as possible manipulation at distant of virtual objects in order to enable a mental projection of the user in this universe.

In human-computer interaction, haptic feedback means both tactile and force feedback. Tactile, or touch feedback is the term applied to sensations felt by the skin. Tactile feedback allows users to feel things such as the texture of surfaces, temperature or vibration.

Force feedback reproduces directional forces that can result from solid boundaries, the weight of grasped virtual objects, mechanical compliance of an object and inertia. Haptic technology covers a wide spectrum of human interaction with virtual environment. Haptic devices allow users to touch, feel and manipulate three-dimensional objects in virtual environments and tele-operated systems.

A haptic interface device is a device that provides an operator a tactile sense, and the operator can manipulate objects in the three dimensional virtual environment and can acquire their feel of touch. Haptic is the sense of touching something.

A haptic device is connected to a virtual world for enabling touch of three-dimensional modeled objects. If there is a collision in the virtual world between a tool and the object, the device provides force/couple feedback to the user by combining the advantages of computer (virtual) models and physical models.

Interactive simulations, haptic feedback computation often comes from contact forces. Subsequently, the fidelity of haptic rendering depends significantly on contact space modeling. Contact and friction laws between deformable models are often simplified in up to date methods. They do not allow a "realistic" rendering of the subtleties of contact space physical phenomena such as slip and stick effects due to friction or mechanical coupling between contacts.

Some current researches are focused on the mastery of tactile interaction with distant physical objects via virtual objects, due to haptic technology which may result in applications and advanced manufacturing [1]. Actual technology enables us simultaneously to see and feel a virtual object. Research in multi sensory human-computer interaction currently focuses on making greater use of the senses of vision and touch.

Virtual environments seek to create computational scenarios that combine sensory data as sight and touch to form an advanced, realistic, and intuitive user interface.

Using haptic interface, the user can directly manipulate computer-generated virtual objects with his fingers and feel shape and texture [2]. In this system, user can operate virtual objects naturally as in the real world.

The algorithm to generate a haptic response for scissors model

The algorithm would have to be developed and implemented to simulate the dynamic system response. The haptic algorithm is composed of following steps:

- building a proper virtual model of the scissors, namely the virtual scissors;
- finding the closest points to a possible collision along the blade of scissors (collision points), both for self-collision or collision with external objects;
- generating cutting/repulsion torque (or Cartesian velocities);
- creating a small linear cutting force/torque ramp for different thicknesses of the cutting material;
- computing the cutting torque (or joint velocities) commands, to be transferred to the real commands via the controller of the physical scissors device.

Virtual Prototype of a Pair of Scissors

Virtual reality is also known as virtual (or synthetic) environments and is widely used as a training tool in engineering. Virtual environments are three-dimensional computer generated environments that the user is able to experience interactively. The manner humans interact with their physical environments, is artificially imitated in virtual environments. The modelling tool provides a modelling environment that automatically generates a simulation script for the dynamic response to user controlled conditions and variable values. The output data can be viewed, exported or graphed with little human effort.

Virtual environment provides a natural interface between humans and computers in virtual reality. The interaction between virtual environment and physical environment (world) is mediated by the haptic interface that allows to be manipulated in real time [3].

A haptic device consists of an interfacing system with humans through output of sensory information and input of commands. An example of such input and output devices is the haptic scissors.

The major goal of computer graphics and virtual environment simulation is to provide realistic methods to allow us easy creation of digital equivalents for natural phenomena such as scissors dynamic behavior and its interactions.

With up-to-date simulation tools one can deal with exact geometry, consider the dynamic characteristics of a pair of scissors, include the man–machine interfaces, and visualize objects in three-dimensional space in detail.

Having all this in mind and evaluating the possible solutions, the boundaries can be pushed further away, especially when using advanced virtual reality tools. Simulation is employed to predict cutting forces and dynamic response of the haptic system during different operations and following adequate scenarios. Scenarios can serve as specifications for virtual prototypes. This technique emphasizes narrative, rather than theory, as the primary source for designers. Scenarios go to a high level of detail by creating an illustrated script, thus providing a more detailed specification for prototype software.

Using simulators, we may build experimental environments, according to our own imagination. What we have imagined during the night can be transposed into visual imagines the next day. Complexity, specificity can be gradually increased to a level where virtual systems can lead to real challenges of the physical world. The ability to simulate opens a wide range of options for solving many problems creatively. We can investigate, design, visualize and finally see the performances of a scissors in the virtual environment before it is used on real object. It is possible that our solutions may fail or even blow up, but only in simulating on the virtual prototype. It forces us to think thoroughly and understand how the system really works.

In this paper the authors propose the Delphi software environment, for the simulation of the haptic systems, using visual programming. Delphi Simulators are the basic tools for creating the virtual prototypes. They are suited for users who know the visual programming, but visualizing still needs another demand. Delphi Simulators enable more open and flexible software tools for programming the algorithms for haptic portable torque-feedback scissors.

The simulator created in the Delphi environment was used to test the performances of the haptic scissors in their integrated environment. The simulator created with the DELPHI code, allows us to create the virtual scissors prototype. This model deals with the characteristics of motion, regarding the effects of forces or masses. The simulation methodology uses a number of possible scenarios and involves the presence of the user to create some sequences of animation in the virtual environment, as one can see in Fig. 1.

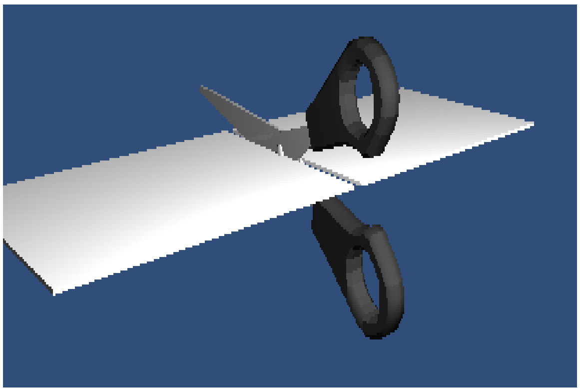

Fig.2.2.1 Sequence of animation in virtual environment

For writing applications that produce three-dimensional computer graphics, we used Delphi language and graphical objects existing in OpenGL standard library. The Delphi source code - only one procedure - for this application, is shown below.

```
procedure DrawModel(M : TModel);
var I, J, K : Integer;
begin
  For I :=1 to M.Groups do
  begin
    glMaterialfv(GL_FRONT_AND_BACK,GL_DIFFUSE,
   @M.Material [M.Group[I].mIndex].Diffuse);
    glMaterialfv(GL_FRONT_AND_BACK, GL_SPECULAR,
    M.Material[M.Group[I].mIndex].Specular);
    glMaterialfv(GL_FRONT_AND_BACK,GL_AMBIENT,
    @M.Material[M.Group[I].mIndex].Ambient);
   glMaterialfv(GL_FRONT_AND_BACK,GL_SHININESS,
   @M.Material[M.Group[I].mIndex].Shininess);
    glDisable(GL_TEXTURE_2D);
    for J :=1 to M.Group[I].Faces do
    begin
      with M.Group[I].Face[J] do
      begin
        case Count of
          3 : glBegin(GL_TRIANGLES);
          4 : glBegin(GL_QUADS);
        else
          glBegin(GL_POLYGON);
        end;
```

```
    for K :=0 to Count-1 do
    begin
      if M.Normals > 0 then
        glNormal3fv( @M.Normal[nIndex[K]] );
      glVertex3fv( @M.Vertex[vIndex[K]] );
    end;
    glEnd();
. . . . . . . . . . . . .
   end;
```

Rendering of virtual Scissor capability

Using simulation, we can learn about some things in a very effective way and while modifying “rules” we can observe the effects of our intervention. As it is in our nature that “seeing and feeling is believing”, the visualization and haptic sensation are the dominant characteristics of a haptic device [3].

During simulation a model is driven by input data and produces output data. Data can come from initial values in the model, user entry or a data file. Output data can be stored in the model or in output data files and used to drive configured charts. An integrated Data File editor makes it easy to create, view, import data from or export data to other applications.

Concerning the cutting process, simultaneous rendering of cutting (rotational) and translational movement is necessary for realistic simulation.

Virtual scissors can be handled using graphics modeling. We like the visualizing properties of the virtual scissors, but we are alarmed about their complexity. We would also like to take advantage of the knowledge about physical model, which is constrained mainly in favor of virtual model.

Rendering of virtual Scissor Action

The geometric modeling cutting (rotational) movement is performed by computing the Boolean operation between the tool object and the modeled work object [4]. The first step is recognition of the direction of motion of the blades: opening or closing. This is determined by the sign of the angular velocity of the scissor blades.

The subtle Boolean operation simulates the modeling process which deforms the shape of the work object by adding the band on it. The shape of the object is modified interactively.

In the case of cutting, the object after modeling can be created by removing raster graphics images (bitmaps) of the inside attribute of the work object and maintaining the inside attribute of the work tool (scissors). An intuitive modeling method is possible by representing cutting resistance as tactile sense information to the user. The algorithm to generate tactile sense depends on the deformation modeling methods [5]. In the deformation modeling method, the objects can be deformed and the user can touch them and feel twist force. In the non-modeling method, the objects can not be deformed and the user can only touch them.

The haptic rendering algorithm in the deformation modeling method generates the cutting force proportional to cutting volume and a tactile force proportional to speed. It is difficult to calculate the volume dislocated by the tool precisely in point-based geometry. Therefore we use the number of raster to be removed during the cutting operation, as a dislocated volume.

The haptic rendering algorithm in the deformation non-modeling method generates only a tactile force proportional to speed. The magnitude of the force is calculated by means of the empirical expressions. The cutting force proportional to the number of raster restricts the withdrawal movement of the scissors tool from the work object if it is always applied while the scissors tool penetrates the work object. Therefore, only a speed element in the direction to penetrate the work object is added to the reactive force as the resistance force. The direction of the reactive force is given by the normal vectors on the surface of the work object.

For the calculation of the reactive force, only a speed in the penetration direction is calculated, while the withdrawal speed, in the opposite direction, will be ignored.

The Scissors tool can be used to split paths, shapes and frames and to reshape them. We proposed to realize the following tasks, with the virtual scissors tool, in the virtual environment:

- cut straight across a shape;
- cut a piece out of a shape;
- cut a curved piece out of a shape;
- cutting or modifying paths and frames.

Cutting straight across a shape with the scissors tool is one of the scissors' tasks. Imagine a cardboard rectangle that we need to cut with the scissors tool in the virtual environment.

These were the steps followed:

- first, we selected the scissors tool in relation to the thickness of the cardboard.
- then, we moved the scissors till it collided with one side of the rectangle, right on the point where we needed to split paths,
- then we clicked on the right ***mouse*** button and the scissors blades started to close.

We moved the left-hand portion of the rectangle away from the right-hand portion in order to better observe the cut, as is presented on the Figure 1. The user moves the cursor in the virtual environment freely using a device, and can operate spontaneously.

In the virtual environment we can also move the cardboard by using the A and D key on the ***computer keyboard,*** for left/right direction or the W and S key, for top/bottom direction so as to bring the cardboard into contact with the cutting edge of the scissors.

Rendering of virtual scissor orientation

We can rotate our item precisely by selecting it on scene that we want to rotate and then clicking on the item, we drag away from the object if we want to rotate it clockwise, or drag towards the object to rotate it counter-clockwise. To rotate the scissors' blades, for a positive value of the angle of rotation the upper blade will rotate counter-clockwise, while for a negative value, it will rotate clockwise.

After selecting the item we want to rotate, with the mouse tool, we start to rotate the image frame along with the image contained within. If we have selected our item, then our virtual tool will rotate, but the frame containing it will not. To rotate the selected object, we must click the left mouse button, hold down and subsequently drag the cursor. The rotation of the virtual object reveals the silhouette of the object. Our attention is focused on the orientation of the scissors' blades and the direction of drawing the contour that must be cut.

Using visual controls would intuitively place the scissors in the desired position. This is useful when trying to give the impression of something being closer or further away, when we want to detach it from the opening cut into the object. Practical contour cutting was performed with some small objects as an A4 cardboard. If we cut a frame that has an image in it, the image contained in the frame won't split, but we will have a copy of the image in the frames we will get after having applied the scissors tool. In the photograph below, we note how the dark lines on the cardboard edge reveal the form that must be cut. Dark lines on the cardboard edge mark out the pathway of contact points of the cutting edge of the scissors with the piece of cardboard, as can be seen in Fig. 2.2.2.

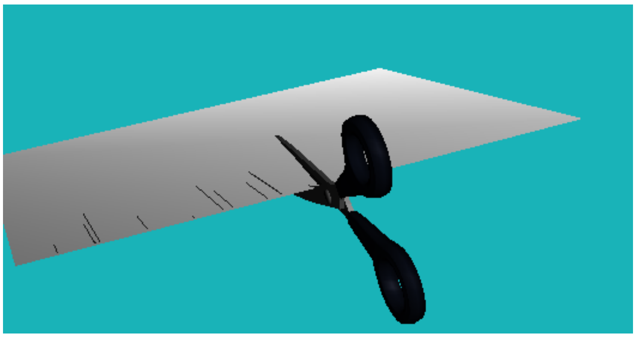

Fig. 2.2.2. Pathway of contact points between the cutting edge and cardboard

Subsequently, the dark lines in the virtual environment must be reproduced by the haptic device in the physical environment (world). The line that runs across an object, hinting at the cutting form, is a cross-contour and is illustrated by the dark lines as is shown above.

In this example, our scenarios were simple enough so that the absence of detailed specifications did not obstruct our later prototyping work. The model was parameterized with data and equations and simulated over time. Simulation results can be viewed on a graphical monitor, and finally exported to a physical device.

Design of a Portable Torque-Feedback Device

Today most force/torque-feedback interfaces come in the form of small robot manipulators: these systems use electric motors to generate forces and joint sensors to measure spatial position [6].

In the physical world the static state of deformation in a plastic object under load generally depends on the forces that are applied to the boundary of the object [7]. When a rigid tool contacts a deformable object, the position and orientation of the tool with respect to the object uniquely defines the forces applied to the object. Therefore, the resultant force applied by the tool can be expressed by a nonlinear function of the position and orientation of the tool.

Haptics is enabled by small robot manipulators that apply the forces to the skin for touch feedback. The actuator of the small robot provides mechanical motion in response to an electrical stimulus.

In our application we used the electric devices that reproduce high fidelity tactile sensations. There are a lot of devices adopting physical mechanisms in the tool assisted type. Our prototype of a scissors-type haptic device consists of three components: an interface, a drive component, and position sensors. The interface is designed to provide the features of a pair of scissors. The drive component is composed of a DC-motor as actuator who generate reactive forces. We used the Gill 360º Blade Rotary Sensor as position encoder.

System outline

The haptic scissors system implements the idea of virtual splitting using virtual scissors, while the operator's hand is holding the physical scissors. The outline of the haptic system developed in this research is shown in Fig.2.2.5.

The system consists of a physical scissors device that provides three dimensional position and orientation outputs for a vectored tactile sense input, a control box for device control, a computer and a modeling software system.

The feeling of cutting resistance and contacting the surface can be conveyed through a physical device. The moving parts of the physical device - mobile blade - must be light-weighted to provide outstanding operation feeling.

The operator grips the handle part of the physical scissors attached to the device, and so provides input for the virtual system. The hand serves as a grasp, allowing operators to maintain the model in its current position. Moving the physical scissors while pressing the blades does affect the position and orientation of the associated virtual object.

The sense in which a user touches through the tool can be reproduced easily in comparison with the sense in which a human touches with the finger; hence the scissors type device can reproduce a better sense of reality by limiting its tactile sense range to the contact point of the moving blade with the cutting object. This hand-held device would have one torque motor to provide torque feedback: as the operator spins the shaft, the device can spin back as appropriate.

The physical cutting device developed in this research is equipped with a DC servo motor and a rotary encoder. A current amplifier was used for DC servo motor to generate a torque of arbitrary magnitude.

Our portable torque-feedback device uses the same basic electronics as standard version of the 2-D joystick, but with additional multifaceted mechanics.

The position and posture inputs from the device to the computer are provided by the rotary encoder. The rotation angle of the axis represents the input to the counter circuit and is applied to the PC bus of the computer as a pulse signal, whereby a rotation angle is measured by counting the number of pulses. The rotary encoder provides the signal for the operation of the positional input block. The power supply to the DC servo motors is controlled by applying a digital signal of variable frequency between 5 Hz and 5 kHz generated by the control circuit.

In this research we used a DC servo motor with a nominal voltage of 18.0[V], and a breakdown torque of 0.2 [Nm].

Sensor and actuator signals are applied to the supervisor system that also controls timing of the system.

The signals are transferred - via a microcontroller – between the physical world and the virtual world and by running the Delphi program on a PC, the virtual scenes are displayed on the video monitor.

The common problem in the design of the haptic scissors is how to apply the fundamental laws that govern the cutting process. The cutting dynamic model is known to be difficult to solve in an efficient way, sometimes even too difficult to be solved at all. Another complexity is the fact that a real-time rendering in immersive virtual environment, with haptic interaction, requires high frequency processing (1000 Hz) and the visual aspects need frame rates in the order of 30 fps.

Because of the noise that arises when differentiating the position obtained from inductive encoders, position data was passed through a first-order filter with a corner frequency of 10 rad/sec before differentiation. Even with filtering, however, the system still exhibited vibration effects at low velocity.

This problem was overcome and the system was stabilized by creating a small linear force/torque ramp-down band around zero velocity. Within this band, the cutting force/torque was multiplied by a scale factor proportional to the velocity. The scale factor decreased as zero velocity was approached from either positive or negative directions, and became identical to zero when the velocity became zero.

This was possible due to an amount of force/torque output large enough for moving the scissors blades in opposite directions, changing the sign of the velocity and consequently the direction of the output force in a cyclical manner.

Hybrid force/position control of the scissors-robot

This section presents an analytical model based on the concepts of contact mechanics and fracture mechanics to calculate forces applied to scissors during cutting of a plate of material.
The model considers the process of cutting as a sequence of deformation and fracture phases. During deformation phases, forces applied to the scissors are calculated from a torque-angle response model synthesized from measurement data multiplied by a ratio that depends on the position of the cutting crack edge and the curve of the blades.
We will present in following the cutting process of a plate of material with a pair of scissors. We will analyze two cases: only the upper blade of the scissors has angular motion during interaction, in the first case and second case, when twice blades have angular motion.
Figure 3 shows cutting of a plate of material when only the upper blade of the scissors has angular motion during interaction. The plate rests along the edge of the lower blade of the scissors and it does not move during cutting.

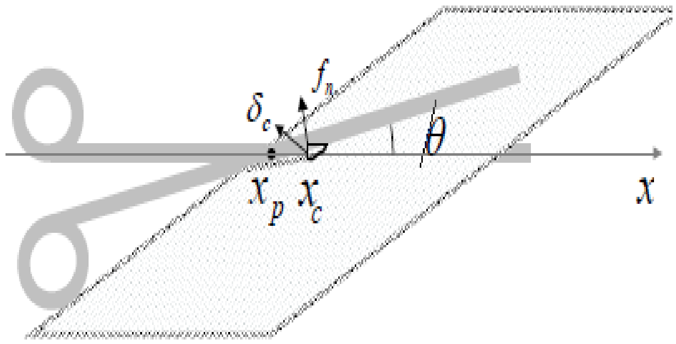

Fig.2.2.3 *Explanation for a scissors with the upper blade angular motion*

Only the pivot of the scissors can move on the surface of the plate along a straight line parallel to an axis denoted by *x*. The lower blade of the scissors does not have any angular motion.
The coordinate x_p defines the location of the pivot and θ defines the opening angle of the scissors. The component of force applied to the upper blade in the direction normal to the blade edge is needed for haptic applications. This component is defined as f_n, and it is calculated during two distinct regimes by a contact model.

Contact Model
The static state of deformation in an elastic object under load generally depends on the forces that are applied to the boundary of the object [3].

When a rigid tool contacts a deformable object, the position and orientation of the tool with respect to the object uniquely defines the forces applied to the object. Therefore, the resultant force applied by the tool can be expressed by a nonlinear function of the position and orientation of the tool.

When we cut an object with a pair of scissors, the force that we feel between our fingers includes two main components: friction forces of the contact of the blades, and the forces of cutting of the object. In this approach we do not model the friction forces. However, friction forces can be measured by opening or closing empty of the scissors, and be summed into the model.

Figure 4 shows the interaction between a pair of scissors and a thin plate with thickness **h** at the time **t** when the plate is locally deformed.

A Cartesian frame is defined at the pivot of the scissors such that the **x** axis is along the symmetry line of the scissors. We assume that the pivot of the scissors does not change orientation during cutting.

We also assume that the pivot of the scissors does not move. The twice blades have angular motion. The opening angle of the scissors is defined by θ, and the position of the edge of the crack made by the scissors is defined by the point x_c , with respect to the pivot point of the scissors. The blades locally deform an area of the plate around the crack edge. The deformation can take various forms including twisting, stretching, compression, or a combination there of.

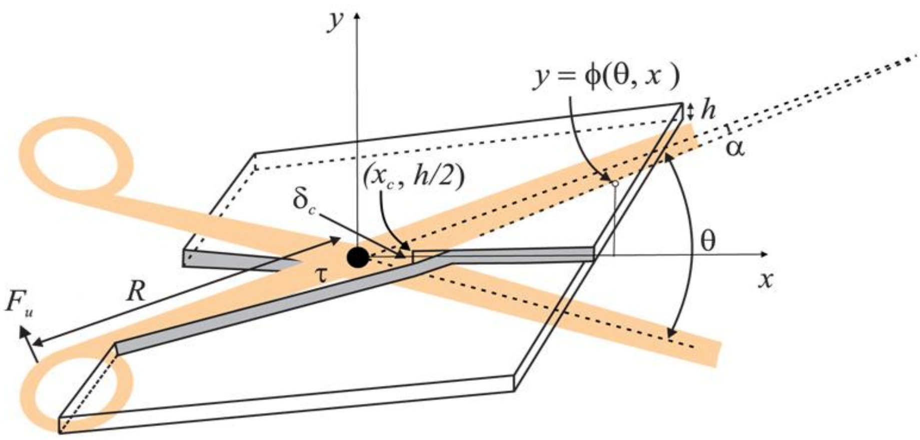

Fig.2.2.4. *The states of scissors and a plate during the cutting process*

At time **t**, the plate is locally deformed. During time period **dt**, a small area of the plate, **h** $\mathbf{dx_c}$, is cut.During deformation, the upper edge of the rupture point is displaced from ($\mathbf{x_c}$, **h**/2) to ($\mathbf{x_c}$, **h**/2 − δ), where δ is a displacement length.

In response to deformation of the plate, the force $\mathbf{f_n}$ is applied to the upper blade along the normal to the blade's edge at point ($\mathbf{x_c}$, **h**/2 − *δ*).

The displacement length *δ* is determined on the base of the curve of the edge of the upper blade. The curve of the blade edge has a significant effect on the torque response of the scissors.

Here, we define the curve of the edge of the upper blade in the Cartesian frame as:

$$y = \phi(x, \theta) \tag{2.2.1}$$

where, (**x**, **y**) is a point on the edge of the blade and $\phi(.)$ is a nonlinear function.
We obtain $\phi(.)$ by fitting an analytical curve to the edge of the upper blade. We extract the blade edge from a real image of the blade.

Considering (2.2.1), the displacement length δ caused by a blade with curve $\phi(.)$ is obtained by:

$$\begin{gathered} h/2 - \delta = \phi(x_c, \theta) \\ \delta = h/2 - \phi(x_c, \theta) \end{gathered} \tag{2.2.2}$$

The normal force, $f_{\mathbf{n}}$ is calculated by a nonlinear function:

$$f_n = g(\delta) \tag{2.2.3}$$

where, **g**(·) is a nonlinear function of the point displacement, obtained by measurement of material properties.

The torque $\boldsymbol{\tau}$ caused by $f_{\mathbf{n}}$ at the shaft-pivot is calculated by:

$$\boldsymbol{\tau} = \boldsymbol{x_c}\, \boldsymbol{f_n}\, cos(\alpha) \tag{2.2.4}$$

where α is the angle between the blade's edge and the centerline of the blade. The angle α is not zero because scissors' blades are slightly conical as shown in Fig.2.2.2.

The force felt by the user at the handle is calculated by:

$$\boldsymbol{F_u} = \frac{\boldsymbol{x_c}}{\mathrm{R}}\, \boldsymbol{f_n}\, cos(\alpha) \tag{2.2.5}$$

where **R** is the distance between the pivot and the handle.

The analytical model accurately predicts the normal applied force.

Force/position control with direct visual observation and force feedback

A pair of scissors is essentially a small robot with two Degrees of Freedom (DOF): rotational (cutting) DOF and translational DOF. Degrees of Freedom can refer both to how a device keeps track of position and how a device outputs forces (Fig.2.2.5).

A haptic scissors system is a 2 DOF device with force feedback. A haptic scissors system keeps track of position with her translational DOF and rotates in the forward-backward by her rotational DOF and gives forces in those same DOF.

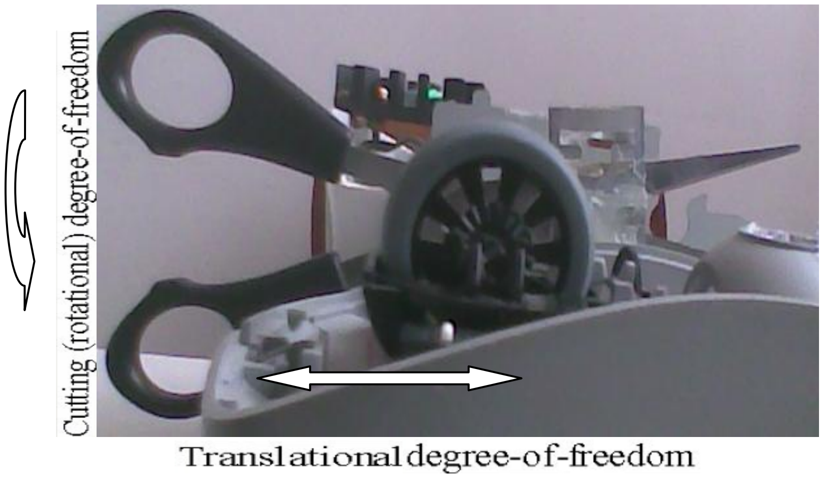

Fig.2.2.5: Two-degree-of-freedom haptic scissors.

Consider a scissors-robot in contact with a rigid object. During the interaction, in the contact point on the object surface, the measure of the force at the scissors-robot blades can be used to compute the feed-back torque at the fingers of the user.

The forces of contact between the blade of the scissors and the plate are expressed by a nonlinear function of deflection at the crack front multiplied by the length of the contact region between the blade and the plate. On the other hand, the force can be expressed as a function of the object pose and of the robot position.

The position and orientation of a frame $O_o\{x_{o,}\, y_{o,}\, z_o\}$ attached to a rigid object with respect to a base coordinate frame $O\{x\, ,y,\, z\}$ can be expressed in terms of the coordinate vector of the origin O_o and of the rotation matrix $\boldsymbol{R}_0(\varphi_o)$, where φ_o is a $(p \times 1)$ vector corresponding to a suitable parameterization of the orientation.

In the case that a minimal representation of the orientation is adopted, e.g., Euler angles, it is p = 3, while it is p = 4 if unit quaternion are used. Hence, the (m × 1) vector $\boldsymbol{X}_o = [\, \boldsymbol{O}^T_o\ \varphi^T_o\,]^T$ defines a representation of the object pose with respect to the base frame in terms of $m = 3 + p$ parameters.

The homogeneous coordinate vector $\tilde{\boldsymbol{p}} = [\boldsymbol{p}^T\ 1]^T$ of a point P of the object with respect to the base frame can be computed as $\tilde{\boldsymbol{p}} = \boldsymbol{H}_o(x_o)\ {}^0\tilde{\boldsymbol{p}}$, where ${}^0\tilde{\boldsymbol{p}}$ is the homogeneous coordinate vector of P with respect to the object frame and $\boldsymbol{H}_o$ is the homogeneous transformation matrix representing the pose of the object frame referred to the base frame:

$$\boldsymbol{H}_0(\boldsymbol{x}_0) = \begin{bmatrix} \boldsymbol{R}_0(\varphi_0) & \boldsymbol{O}_0 \\ \boldsymbol{0}^T & 1 \end{bmatrix} \tag{2.2.6}$$

where $\boldsymbol{0}$ is the (3 × 1) null vector.

It is assumed that the geometry of the object is known and that the interaction involves a portion of the external surface which satisfies a twice differentiable scalar equation $\varphi({}^0\boldsymbol{p}) = 0$.

When in contact, the tip point P_q of the elementary scissors robot instantaneously coincides with a point P of the object, so that the tip position ${}^0\boldsymbol{p}_q$ satisfies the constraint equation:

$$\varphi({}^0\boldsymbol{p}_q) = 0 \tag{2.2.7}$$

The unit vector normal ${}^0\boldsymbol{n}$ to the surface at the point P is expressed in the object frame. The (3 × 1) contact force vector, ${}^o\boldsymbol{f}$ is aligned to the normal unit vector ${}^o\boldsymbol{n}$ to the surface of the object.

Control strategy

This work focuses on techniques for augmenting the performances of haptic scissors system by means of control architecture.

Haptic scissors capabilities in close interaction with human operator can be considered as mimicking sensing and actuation of human. This leads to consider fully integrated human vision and force feed-back based control.

Thanks to the visual perception, the user integrated into a robotic system may achieve global information on the surrounding environment that can be used for task planning and accidents avoidance. Human vision and force/torque feed-back are two complementary sensing capabilities that can be exploited in a synergic way to enhance the precision in the control of a scissors-robot during the interaction with the environment.

On the other hand, the perception of the force applied to the robot allows adjusting the motion so that the local constraints imposed by the environment during the interaction are satisfied.

The precision of the accomplished task and dependability of a robotic system are strictly connected to the availability of sensing information on the external environment. Moreover, human vision may substitute the complex infrastructure needed for “intelligent environments” to detect the interaction in the operational space.

Interaction control

When the scissors-robot moves in free workspace, the unknown object pose and the position of the hand of a human user can be estimated online by using the data provided by the human user eyes. When the scissors-robot is in contact to the object, also the force measurements and the joint position measurements are used. Joint values are used for evaluating the position of the scissors for the interaction configuration with the object.

The Cartesian desired velocity (or force) for the contact (control) points, is transformed in the corresponding joint trajectory via proper inverse kinematics. Any point on the object can be considered as a control point. Consider a scissors-robot in contact with an object in the scissors-to-hand configuration. In the following, some modeling assumption concerning the environment and the scissors-robot are illustrated.

The proposed algorithm can be used to estimate online the pose of an object in the workspace; this allows compute the surface equation with respect to the base frame in the form:

$$\varphi(\boldsymbol{R}_0^{\boldsymbol{T}}(\boldsymbol{p}_{\boldsymbol{q}} - \boldsymbol{o}_0)) = \varphi(\boldsymbol{q}, \boldsymbol{t}) = 0 \qquad (2.2.8)$$

where, the last equality holds in view of the direct kinematics equation of the robot. In the following, it is assumed that the object does not move; the general case of moving object is more complex but can be analyzed in a similar way.

Hence, the constraint equation is $\varphi(\boldsymbol{q}, \boldsymbol{t}) = 0$; moreover $\boldsymbol{J}_\varphi(\boldsymbol{q})\dot{\boldsymbol{q}} = 0$, where $\mathrm{J}_\varphi(\mathrm{q}) = \partial\varphi / \partial \mathrm{q}$ is a Jacobian matrix.

The dynamic model of the manipulator in contact with the object is:

$$\boldsymbol{D}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \tau - \boldsymbol{J}_\varphi^{\boldsymbol{T}}(\boldsymbol{q})\lambda \qquad (2.2.9)$$

where $\boldsymbol{D}$ is the (*n* x *n*) symmetric and positive definite inertia matrix, $\boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is the (*n* x 1) vector taking into account Coriolis, centrifugal, friction and gravity torques, $\boldsymbol{\tau}$ is the (*n* x 1) vector of the joint torques, and λ ¸ is the lagrangean multiplier associated to the constraint. In the case of scissors robot we can assume that Coriolis, centrifugal, friction and gravity torques are less important and can be neglected.

The on-line computation of the constraint equations can be suitably exploited for interaction control. In the following, the case of the hybrid force/position control is considered.

Hybrid force/position control of the scissors-robot

An inverse dynamics control law can be adopted, by choosing the control torque $\boldsymbol{\tau}$ as:

$$\tau = \boldsymbol{D}(\boldsymbol{q})\alpha_{\boldsymbol{r}} + \boldsymbol{n}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{J}_{\varphi}^{T}(\boldsymbol{q})\boldsymbol{h}_{\lambda} \tag{2.2.10}$$

According to the hybrid force/position control strategy, it is useful to apply the change of coordinates with intermediate variables: $\boldsymbol{\alpha_r}$, $\boldsymbol{a_r}$, $\boldsymbol{a_P}$, $\boldsymbol{r_P}$, $\boldsymbol{h}_\lambda$ defined as:

$$\alpha_{\boldsymbol{r}} = \boldsymbol{J}_{\boldsymbol{r}}^{-1}(\boldsymbol{q})(\boldsymbol{a}_{\boldsymbol{r}} - \dot{\boldsymbol{J}}_{\boldsymbol{r}}(\boldsymbol{q})\dot{\boldsymbol{q}})$$

$$\mathrm{a_r} = \begin{bmatrix} 0 & \mathrm{a_P^T} \end{bmatrix}^{\mathrm{T}} \tag{2.2.11}$$

$$\boldsymbol{a_P} = \ddot{\boldsymbol{r}}_{\boldsymbol{Pd}} + \boldsymbol{K}_{\boldsymbol{Dr}}(\dot{\boldsymbol{r}}_{\boldsymbol{Pd}} - \dot{\boldsymbol{r}}_{\boldsymbol{P}}) + \boldsymbol{K}_{Pr}(\boldsymbol{r}_{\boldsymbol{Pd}} - \boldsymbol{r}_{\boldsymbol{P}})$$

where $\boldsymbol{K}_{\mathrm{Dr}}$ and $\boldsymbol{K}_{\mathrm{Pr}}$ are suitable feedback gains.

The vector $\boldsymbol{r}_{\mathrm{p}}$ allows to specify any scissors-robot position and orientation which satisfies the constraint. The force vector in the new coordinate $\boldsymbol{r}$ is: $\boldsymbol{F} = [\lambda \quad 0]^{\boldsymbol{T}}$. The scalar $\boldsymbol{\lambda}$ is computed from the measured contact force. Velocity vector $\dot{\boldsymbol{r}}$ and force vector $\boldsymbol{F}$ are orthogonal, i.e., $\boldsymbol{F}^{\boldsymbol{T}}\dot{\boldsymbol{r}} = 0$ for any $\dot{\boldsymbol{r}}$ that satisfy the constraint (such that $\dot{\boldsymbol{r}}_{\mathrm{F}} = 0$).

A force/position control task can be assigned by specifying the desired force $\boldsymbol{F}_{\mathrm{d}} = \lambda_{\boldsymbol{d}}(\boldsymbol{t})$.
The intermediate variable $\boldsymbol{h}_\lambda$ is calculated as:

$$\boldsymbol{h}_{\lambda} = \lambda_{\boldsymbol{d}} + \boldsymbol{k}_{I\lambda}\int_0^{\boldsymbol{t}}(\lambda_{\boldsymbol{d}}(\tau) - \lambda(\tau))\mathrm{d}\tau \tag{2.2.12}$$

with $\boldsymbol{k}_{I\lambda}$ suitable feedback gain.

On the base of the above relations one can configured an inner/outer structure. The inner loop implements hybrid control whereas the outer loop computes the estimation of the object pose as well as the desired force and motion trajectories.

The cutting model is computationally efficient, so it can be used for real-time computations of the corresponding equations.

Simulation results are presented for the case of the cutting process with a pair of scissors in contact with a planar surface.

Physical prototype of the scissors-robot

In the physical world the static state of deformation in a plastic object under load generally depends on the forces that are applied to the boundary of the object. When a rigid tool contacts a deformable object, the position and orientation of the tool with respect to the object uniquely defines the forces applied to the object. A physical prototype of a scissors haptic device consists of a scissors-robot with two DOF and an electrical DC-motor. Force/torque feedback is created through the electrical motor.

Figure 6 is showing the system overview. Experimental haptic system for cutting samples of paper confirm the model, and is rendered in a haptic virtual environment.

Fig.2.2.6 *Scissors device (bottom- side of the picture) and virtual scissors prototype (up- side of the picture)*

The whole system may achieve global information on the environment using vision of the user.

On the other hand, the perception of the force applied to the hand allows adjusting its motion so that the local constraints imposed by the environment are satisfied.

CONCLUSIONS

The integration of force and visual control to achieve human-robot interaction has been discussed. A position-force technique based on visual control made by the user and the force feedback control has been presented, employing a pose estimation algorithm on the basis of visual direct observation, force and joint position data. A 2-DOF hybrid force/position control technique was proposed in this paper. The cutting model is used for the haptic system synthesis for the interaction between a virtual pair of scissors with a virtual plate via the physical scissors handled by human user. Simulation results have demonstrated the superior performance of the proposed hybrid control technique with respect to an algorithm using only human’s visual perception for the estimation of the object pose. This haptic rendering technique is computationally efficient and can be applied to real-time cutting process.

Using and testing the haptic devices is a very educational experience. While we knew approximately what we wished to achieve, we were able to gather valuable feedback by having the real haptic device in our hands and feeling the impact of torque and force feedback. We consider that the desktop based workspace could have a great impact when used as a design aid in conjunction with a virtual prototyping application.
We studied PHANTOM haptic device and we did some qualitative human perception experiments and informal surveys and questionnaires in an effort to quantify some of the design parameters of the FALCON system.

Haptic devices can bee integrated into the Delphi software, which simulates maintenance analysis in a large scale industrial environment. We have created on some desktop virtual prototyping system for automotive industry. We targeted maintenance analysis applications in particular. We think that the 6DOF haptic technology in continuing research will provide much value to virtual prototyping applications on the desktop.

Companies which are most likely to benefit from this technology are those in the automotive industries, where the parts are many and large, maintainability is a major concern, and it is prohibitively expensive to produce physical prototypes. All product design can stand to profit from such a desktop design aid. In the design of any product, it is always important to ensure that it can be assembled and disassembled speedily and without problems. Haptic technology was packaged into major CAD systems and it would speed up the product development cycle and allow better products to be designed and prototyped in less time and cost.

This study introduces a novel structure utilizing the immersive virtual environment for the interaction between virtual scissors and virtual objects in cutting processes and haptic rendering. Haptic scissors facilitates precision cutting tasks (maneuvering in constrained spaces, etc.) via haptic feedback.

Evaluation experiment for basic performance of the device is carried out, and the device is proved to be able to generate the computed cutting force/torque. A virtual cutting experiment was carried out, and it proved that the device is able to provide a feeling of cutting. Poor performances are often attributable to operator error. Operator’s hand and dynamic factors (e.g., changing environment) may still reduce the interface’s performances. In this case study, the cooperative human-computer work problem was solved enforcing a ‘‘human-in-the-loop’’ strategy.

In this paper we summarized tactile and visual information type and modality combination. The cognitive properties of cutting modality influence the compatibility of multimodal combinations. In multitask environment; the cognitive properties of modality also influence how well information processing can be time-shared with other concurrent tasks.

Summing up, the results of the experiment reported here provide the evidence of a tactile perception of the displacement effect of the scissors’ blades on the physical objects. The influence of cutting force on the virtual object’s surface was investigated. A study on resulting surface borders caused by preferred scissors orientation was also conducted. The result reveals that increasing the cutting force will decrease surface roughness down to a given point beyond which the surface roughness value continues to decrease as the feed rate decreases.

The novelty of our approach consists of on-line modeling and of integrating the interface elements into the virtual environment itself. We also take into account the properties of the edited entities to provide rules which facilitate the construction of 3D objects. It is expected that working with haptic scissors would prove especially appropriate for opening new ways regarding sense of touch applications in relation with the utilization of computer systems.

ACKNOWLEDGEMENTS

We would like to thank Thierry Marie ***Guerra*** and Michel Dambrine from LAMIH laboratory - Valenciennes University, France - for their permission to demonstrate the Delphi software for virtual prototyping applications and for technical information regarding haptic devices. LAMIH Laboratory supported our collaboration in the haptic scissors project and helped to move this project from intentions and suggestions level to functional prototypes, among many other contributions. LAMIH Laboratory is co-inventor of the portable torque-feedback device. One prototype was developed at Transilvania University of Braso

Chapter 3 The reciprocal robots collision avoidance

3.1 Collision-avoidance using redundant robots

The present article is intended to be a high-level survey over the strategy to on-line collision-avoidance research that has been carried out in the past ten years. Due to length restrictions, only a strategy/ technique is explicitly referred to. More exactly, we have selected 10 papers that we felt were important and different from each other. Also the novelty of the proposed technique, the number of the citations and the number of downloads were taken into account in the selection process.

The interest in creating such an overview is diverse. By selecting the most interesting approaches, we want to focus the attention to new techniques and methodologies that may be of high interest to the researchers in the field of collision avoidance. Moreover, this selection can be a useful indicator of the areas that will constitute the future research trends.

The strategy proposed allows the use of redundant degrees of freedom such that a manipulator can avoid obstacles while tracking the desired end-effectors trajectory. It is supposed that the obstacles in the workspace of the manipulator are static.
The strategy is based on the redundant inverse kinematics and leads to the favorable use of the abilities of redundant robots to avoid the collisions with obstacles. This strategy has the advantage that the configuration of the manipulator can be influenced by further requirements such as joint limits, etc. The effectiveness of the proposed strategy is discussed by theoretical considerations and illustrated by simulation of the motion of the four-joint planar manipulators between symmetric obstacles. It is shown that the proposed collision-free strategy while tracking the end-effector trajectory is efficient and practical.

Introduction

Particular attention has been devoted to the study of redundant manipulators in the last years. Redundancy has been recognized as a major characteristic in performing tasks that require dexterity comparable to that of the human arm.

While most non-redundant manipulators possess enough degrees-of-freedom (DOFs) to perform their main task(s), i.e., position and/or orientation tracking, it is known that their limited manipulability results in a reduction in the workspace due to mechanical limits on joint articulation and presence of obstacles in the workspace.
Redundant manipulators possess extra DOFs than those required to perform the main task(s). These additional DOFs can be used to fulfill user-defined additional task(s).

The additional task(s) can be represented as kinematic functions. This not only includes the kinematic functions which reflect some desirable kinematic characteristics of the manipulator such as posture control, joint limiting, and obstacle avoidance, but can also be extended to include dynamic measures of performance by defining kinematic functions as the configuration-dependent terms in the manipulator dynamic model, e.g., impact force, inertia control, etc.

In order to accomplish a task, an accurate joint motion must be commanded to the manipulator. For that reason, it is necessary to obtain mathematical relations which allow computing joint-space variables corresponding to the assigned task-space variables. This is the objective of the inverse kinematics problem.

In this paper the problem of redundant inverse kinematics is reviewed and obstacles avoid subtask for exploiting the self motion are defined. Various different authors dealt with the solution of the redundant inverse kinematics problem.
Cartesian position and orientation for the end-effectors task vector $\boldsymbol{x}_t$ can be described as a function of joint variables vector $\boldsymbol{q}$ of the manipulator:

$$\boldsymbol{x}_t = f(\boldsymbol{q}) \tag{3.1.1}$$

While equation (3.1.1) can be obtained easily with the help of Denawit-Hartenberg operators, the inverse problem is crucial. In the redundant case it is generally not possible to find an inverse mapping, f^{-1}.

As an alternative of constructing an inverse mapping, the problem is often reformulated in the velocities, utilizing the partial derivation of $f(q)$.
The end-effectors task velocity vector $\dot{\boldsymbol{x}}_t$ is:

$$\dot{\boldsymbol{x}}_t = \boldsymbol{J}_t(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{3.1.2}$$

with Jacobian matrix:

$$\boldsymbol{J}_t = \partial \boldsymbol{f}(\boldsymbol{q}) / \partial \boldsymbol{q} \tag{3.1.3}$$

Due to the fact that the inverse of the non-square (analytical) Jacobian $\boldsymbol{J}_t(q)$ does not exist in the redundant case, the well known generalized Moore–Penrose pseudo inverse $\boldsymbol{J}_t^+(q)$ is utilized. This proposed strategy often employs a special solution of equation (3.1.2).
Optimization criteria for the redundant self motion can be supplementary by e.g. null-space projection, which leads to the relation:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_t^+\dot{\boldsymbol{x}}_t + (\boldsymbol{I} - \boldsymbol{J}_t^+\boldsymbol{J}_t)\dot{\boldsymbol{q}}_0 \tag{3.1.4}$$

Here, $(\boldsymbol{I} - \boldsymbol{J}_t^+\boldsymbol{J}_t)$ represents the orthogonal projection matrix in the null space of $\boldsymbol{J}_t$, and $\dot{\boldsymbol{q}}_0$ is an arbitrary joint-space velocity; the second part of the solution is therefore a null-space velocity. The vector $\dot{\boldsymbol{q}}_0$ can be chosen arbitrary and is used to force a desired behavior for the null space motion. The null space of $\boldsymbol{J}_t\,(q)$ is defined as:

$$\boldsymbol{N}(\boldsymbol{J}_{\mathrm{t}}(\boldsymbol{q})) = \{\dot{\boldsymbol{q}} \in \dot{\boldsymbol{Q}} : 0 = \boldsymbol{J}_{\mathrm{t}}(\boldsymbol{q})\dot{\boldsymbol{q}}\} \tag{3.1.5}$$

In redundant directions joint velocities causes no motion at the end-effectors level. These are internal motions of the manipulator. Redundant joint velocities satisfy the equation:

$$\boldsymbol{J}_{\mathrm{t}}\,(\boldsymbol{q})\dot{\boldsymbol{q}} = 0 \tag{3.1.6}$$

At each configuration, the null space of $\boldsymbol{J}_t$ is the set of joint-space velocities which yield zero task velocity; these are thus called null-space velocities.

AUGMENTED JACOBIAN

The redundant robots can operate in the Cartesian space with obstacles. The redundant self motion can satisfy both, the end-effector task and the additional constraint task cause of the obstacles, at the same time.

The concept of task-space augmentation introduces a constraint task to be fulfilled along with the end-effector task. In that case, an augmented Jacobian matrix is set-up whose inverse gives the required joint velocity solution.

Let us consider the p-dimensional vector $\boldsymbol{x}_c = (x_{c,1} \ldots x_{c,p})$ which describes the additional tasks to be fulfilled in addition the m-dimensional end-effector task vector $\boldsymbol{x}_t$, i. e., the degree of redundancy $p = n - m$, whereas n is the number of joints.

The relation between the joint-space coordinate vector $\boldsymbol{q}$ and the constraint-task vector $\boldsymbol{x}_c$ can be considered as a direct kinematics equation:

$$\boldsymbol{x}_{\mathrm{c}} = \boldsymbol{k}_{\mathrm{c}}(\boldsymbol{q}) \tag{3.1.7}$$

where $\boldsymbol{k}_c$ is a continuous nonlinear vector function. By differentiating (3.1.7) one can be obtained (3.1.8):

$$\dot{\boldsymbol{x}}_{\mathrm{c}} = \boldsymbol{J}_{\mathrm{c}}(\boldsymbol{q})\dot{\boldsymbol{q}}\,. \tag{3.1.8}$$

In (3.1.8) $\dot{\boldsymbol{x}}_c$ is the constraint-task velocity vector, and the mapping $\boldsymbol{J}_c(\boldsymbol{q}) = \partial\boldsymbol{k}_c/\partial\boldsymbol{q}$ is the (p x n) constraint-task Jacobian matrix. At this moment, an augmented-task vector x_a, can be defined by stacking the end-effector task vector with the constraint-task vector as:

$$\boldsymbol{x}_{\mathrm{a}} = \begin{bmatrix}\boldsymbol{x}_{\mathrm{t}} & \boldsymbol{x}_{\mathrm{c}}\end{bmatrix}^{\mathrm{T}} = \begin{bmatrix}\boldsymbol{k}_{\mathrm{t}}(\boldsymbol{q}) & \boldsymbol{k}_{\mathrm{c}}(\boldsymbol{q})\end{bmatrix}^{\mathrm{T}}. \tag{3.1.9}$$

According to this definition, finding a joint configuration $\boldsymbol{q}$ that result, in some desired value for $\boldsymbol{x}_a$, means satisfying both the end-effector task and the constraint task, at the same time.

At the differential level, Cartesian velocities are given by:

$$\dot{\boldsymbol{x}}_{\mathrm{a}} = \boldsymbol{J}_{\mathrm{a}}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{3.1.10}$$

To finding a joint velocity $\dot{\boldsymbol{q}}$ in some desired value for Cartesian velocities $\dot{\boldsymbol{x}}_a$ can be used:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_{\mathrm{a}}^{+}\dot{\boldsymbol{x}}_{\mathrm{a}} + (\boldsymbol{I} - \boldsymbol{J}_{\mathrm{a}}^{+}\boldsymbol{J}_{\mathrm{a}})\dot{\boldsymbol{q}}_0 \tag{3.1.11}$$

That means the inverting of the augmented Jacobian, $\boldsymbol{J}_{\mathrm{a}}(\boldsymbol{q}) = [\ \boldsymbol{J}_{\mathrm{t}}(\boldsymbol{q}) \quad \boldsymbol{J}_{\mathrm{c}}(\boldsymbol{q})\ ]^{\mathrm{T}}$ to obtain the pseudo-inverse matrix, $\boldsymbol{J}_{\mathrm{a}}^{+}$.

Task priority strategy

With reference to solution (3.1.11), the task-priority strategy consists of computing $\dot{\boldsymbol{q}}_0$ so as to correctly achieve the *p*-dimensional constraint-task velocity, $\dot{\boldsymbol{x}}_{\mathrm{c}}$.
In the typical case, an end-effector task is considered as the primary task, even if examples may be given in which it becomes the secondary task. The idea is that, when an exact solution does not exist, the reconstruction error should only affect the lower-priority task.

Conflicts between the end-effector task and the constraint task are handled in the framework of the task-priority strategy by correctly assigning an order of priority to the each given tasks and then satisfying the lower priority task only in the null space of the higher-priority task. Remarkably, the projection of $\dot{\boldsymbol{q}}_0$ onto the null space of $\boldsymbol{J}_{\mathrm{t}}$ ensures lower priority of the constraint task with respect to the end-effector task because this results in a null-space velocity for the higher-priority task.
When the secondary task $\dot{\boldsymbol{x}}_{\mathrm{c}}$ is orthogonal to the primary task $\dot{\boldsymbol{x}}_{\mathrm{t}}$ the joint velocity:

$$\dot{\boldsymbol{q}}_0 = \boldsymbol{J}_{\mathrm{c}}^{+}(\boldsymbol{q})\dot{\boldsymbol{x}}_{\mathrm{c}} \tag{3.1.12}$$

would easily solve the problem, being in addition already a null-space velocity for the primary-task velocity mapping (3.1.2).

However, in the general case the two tasks may be compatible but not orthogonal and there not exist a joint velocity solution that ensures the achievement of both $\dot{\boldsymbol{x}}_{\mathrm{t}}$ and $\dot{\boldsymbol{x}}_{\mathrm{c}}$. Coherently with the defined order of priority between the two tasks, a reasonable choice is then to guarantee exact tracking of the primary-task velocity, while minimizing the constraint-task velocity reconstruction error $\boldsymbol{\varepsilon} = \dot{\boldsymbol{x}}_{\mathrm{c}} - \boldsymbol{J}_{\mathrm{c}}(\boldsymbol{q})\dot{\boldsymbol{q}}$; this gives:

$$\dot{\boldsymbol{q}}_0 = [\ \boldsymbol{J}_{\mathrm{c}}(\boldsymbol{I} - \boldsymbol{J}_{\mathrm{t}}^{+}\boldsymbol{J}_{\mathrm{t}})\ ]^{+}\left(\dot{\boldsymbol{x}}_{\mathrm{c}} - \boldsymbol{J}_{\mathrm{c}}\boldsymbol{J}_{\mathrm{t}}^{+}\dot{\boldsymbol{x}}_{\mathrm{t}}\right) \tag{3.1.13}$$

The solution given by (3.1.4) and (3.1.13) can be simplified to:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_t^{+}\dot{\boldsymbol{x}}_t + [\ \boldsymbol{J}_c(\boldsymbol{I} - \boldsymbol{J}_t^{+}\boldsymbol{J}_t)\ ]^{+}\left(\dot{\boldsymbol{x}}_c - \boldsymbol{J}_c\boldsymbol{J}_t^{+}\dot{\boldsymbol{x}}_t\right) \tag{3.1.14}$$

Another approach is to relax minimization of the secondary-task velocity reconstruction constraint and simply pursue tracking of the components of (12) that do not conflict with the primary task, namely

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_{\mathrm{t}}^{+}\dot{\boldsymbol{x}}_{\mathrm{t}} + (\boldsymbol{I} - \boldsymbol{J}_{\mathrm{t}}^{+}\boldsymbol{J}_{\mathrm{t}})\boldsymbol{J}_{\mathrm{c}}^{+}\dot{\boldsymbol{x}}_{\mathrm{c}} \tag{3.1.15}$$

An intuitive justification of this solution can be given as follows: The pseudo inverses $\boldsymbol{J}_{\mathrm{t}}^{+}$ and $\boldsymbol{J}_{\mathrm{c}}^{+}$ are used to solve separately for the joint velocities associated with the respective task velocities. The joint velocity associated with the (secondary) constraint task is then projected onto the null space of $\boldsymbol{J}_{\mathrm{t}}$ to remove the components that would interfere with the (primary) end-effector task.

Finally the joint velocity associated with the constraint task is added to the joint velocity associated with the end-effector task.
By construction, the solution (3.1.15) leads to larger constraint-task reconstruction errors than solution (3.1.14); this is the price paid to give smooth and feasible trajectories for the joint velocity in tracking conflicting tasks.

Physical constraint problem

Physical constraints, like joint speed limits can be enforced in a natural way via inequality constraints. Additional constraints in form of inequalities can be added to restrict the solutions for particular joints (e.g. restrictions due to hardware constraints or maximum joint movements per time step, T_a).

$$\Delta\boldsymbol{q} \in \left[\Delta\boldsymbol{q}_{\min}, \Delta\boldsymbol{q}_{\max}\right] \tag{3.1.16}$$

Using the following incremental formulation:

$$\dot{\boldsymbol{q}} \approx \Delta\boldsymbol{q}/\mathrm{T}_{\mathrm{a}} \tag{3.1.17}$$

$$\dot{\boldsymbol{x}} \approx \Delta\boldsymbol{x}/T_a \tag{3.1.18}$$

for the null-space motion (3.1.6), linear conditions can be chosen of the form:

$$\boldsymbol{A}\Delta\boldsymbol{q} = \boldsymbol{a} \tag{3.1.19}$$

So the inverse kinematics problem can be formulated as an optimization problem

$$\min_{\Delta \mathrm{q}} \boldsymbol{L}_{\mathrm{IK}}(\Delta\boldsymbol{q}) \tag{3.1.20}$$

for the cost function

$$\boldsymbol{L}_{\mathrm{IK}}(\Delta\boldsymbol{q}) = \left\| \boldsymbol{A}\Delta\boldsymbol{q} - \boldsymbol{a} \right\|_2 \tag{3.1.21}$$

with the equality constraint as main task

$$\Delta\boldsymbol{x}_{\mathrm{t}} = \boldsymbol{J}_{\mathrm{t}}\Delta\boldsymbol{q} \tag{3.1.22}$$

Herein the symbol $\|.\|_2$ denotes the quadratic norm. Notice that the main task forms in fact the constraint for a linear optimization problem here.
The particular equations:

$$\boldsymbol{A}_{\mathrm{i}}\Delta\boldsymbol{q} = \boldsymbol{a}_{\mathrm{i}} \tag{3.1.23}$$

are weighted by the factors w_i, which define how much an error in the i^{th} equation affects the value of the cost function $\boldsymbol{L}_{IK}(\Delta\boldsymbol{q})$.

It is now possible to combine multiple behaviors for the null-space motion by choosing

$$\boldsymbol{A} = \left[\mathrm{w}_1\boldsymbol{A}_1 \ldots\ldots \mathrm{w}_{\mathrm{j}}\boldsymbol{A}_{\mathrm{j}}\right]^{\mathrm{T}}; \boldsymbol{a} = \left[\mathrm{w}_1\mathrm{a}_1 \ldots. \mathrm{w}_{\mathrm{j}}\mathrm{a}_{\mathrm{j}}\right]^{\mathrm{T}} \tag{3.1.24}$$

It should also be noted that this optimization problem leads to the pseudo inverse solution by choosing $\boldsymbol{A}$ as the identity matrix and $\boldsymbol{a}$ as a vector containing just zero elements.

Collinsion avoidance problem

The null-space motion concept is use for the avoidance of collisions, of the robot arm, with an external object. Collision avoidance is here only presented for pure null-space movements. Only collisions of the robot's internal structure with the" external object", is represented. For that reason an artificial potential field is attached to the considered external object.

Let d denote the minimal distance between the robot and the external object. If this distance is smaller than a proper chosen value d_0 defined in the collision area, then the robot should move away according to the potential field $V(d)$.

$$V(d) = \frac{1}{2}(d - d_0)^2 \tag{3.1.25}$$

For the null-space motion an equation of the form (3.1.13) is needed. Assume, that the robot moves sufficiently slowly, then the change of the value of $\boldsymbol{V}(d)$ due to an joint movement can be formulated as follows:

$$\Delta V(d) = \frac{\partial V(d)}{\partial d}\frac{\partial d}{\partial \boldsymbol{q}}\Delta \boldsymbol{q} \tag{3.1.26}$$

Equation (3.1.26) can now be transposed as the (3.1.23) form, with

$$\boldsymbol{A}_1 = \frac{\partial V(d)}{\partial d}\frac{\partial d}{\partial \boldsymbol{q}} \tag{3.1.27}$$

It is obvious that collisions of the end-effector with the collision-object cannot be handled in this way, because only self motion has been considered herein. To avoid collisions of the end-effector too, it is necessary to allow a deviation of the desired trajectory, which can be calculated similar to (3.1.27).

For interaction between the user and the collision system a position tracker is utilized.
For every obstacle a boundary ellipse is defined in workspace such that there is no collision if the robot joints or end-effectors are outside these ellipses. The collision-avoidance algorithm computes on-line the self-motion movements as function of distance d. The movements of a reference frame, which is fixed to a particular joint, can be commanded the same as it don't affect the trajectory of the end-effector.

A. Delphi simulators for validation of the collision-avoidance strategy

The authors propose the Delphi informatics environment for the qualitative simulation of the robotic systems, using the visual programming. The prototype graphical simulation system has been developed to support and demonstrate different aspects of robots behavior.
In traditional robotics, programmers attempted to anticipate and explicitly control every aspect of the action of the robot. However, these systems often failed when are placed in a changing environment, where unanticipated situations appear.
Systems using autonomous agents, with de-centralized control of robot motion, enable robots to learn and adapt to changing environments. When in the environment appear accidentally the

unanticipated objects, the robot control de-centralized system must be able to select the optimal path for avoid the collision between robot arm and strangers objects.
Autonomous agents are software programs designed to pursue goals and can do so without user intervention. The idea is that agents are not strictly invoked for a task, but activate themselves. They are usually large applications, which attempt to solve practical problems demanded by the additional tasks. Advanced robotic systems involve large agent-based simulators aimed at understanding "action selection". By this mechanism the robot selects the behavior to execute from many possible behaviors.

Delphi simulators are generally smaller software programs and they are written in order to evaluate and validate by simulation the robots' behavior. They usually involve algorithms that allow exposing the behavior of the virtual robots and their virtual environment. This technique is used also to provide the command methods for the real robotic systems.
All of these programs enable the manifestation of global behaviors. They also attempt to simulate more aspects of robot behavior - not just movement. As such they include the movement of robot manipulators, the contact with manipulated objects and the collision with unexpected objects into the work environment.

All components of the artificial world are objects, as is the environment itself. The interface consists of a collection of tools for analysis and visualization.
Data analysis objects can communicate with the user interface objects to present displays of data, as well as for storage to files. The prototype allows the user to create, modify, and destroy objects. Each object has a few standard attributes managed by Object Inspector, as well as user-specified private data [6].

In this paper the graphical simulation prototype system has been developed to support and demonstrate different aspects of robots behavior when met the obstacles in his work environment. All of these programs are write in Delphi language and enable the manifestation of global behaviors.
For itself they include the movement of virtual robot manipulator, the contact with virtual objects and the collision avoid, into the work environment, with expected or unexpected virtual objects.

The information regarding the collision are given about the shape of each face of the body and then it figures out which bodies face touch each other and passes the resulting contact point visual information to designer. The designer can then take the proper decisions concerning the real prototype.
To implement a real-time collision-avoidance strategy, robot and environment modeling and distance calculation need to be investigated. An obstacle-avoidance system essentially deals with a complex environment.
There are many limitations in creating (modeling) a robot's environment such as space and equipment limitations. Obviously, the accuracy with which a robot arm and its environment are modeled is directly related to the real-time control requirements.
Greater detail in modeling results in higher complexity, when computing the critical distances between an obstacle and the manipulator is needed.

Much of this computation can be avoided if the distance measurements are obtained by a proximity sensing system. For situations where proximity sensors are not available, a possible solution is to use simple geometric primitives to represent the virtual arm and its virtual environment.
A simplified geometrical model for links of industrial manipulators is relevant to the study of collisions either with each other or with objects in the workspace.

In this paper the joints were represented by circles and the links were modeled by straight lines and rectangles.
A redundancy-resolution scenario was proposed to achieve obstacle avoidance. In the following simulations the main task consists of tracking the position and orientation trajectories, generated by linear interpolation, between the initial and final poses. It should be noted that interpolation of rotations is a much more complex problem than point interpolation [7]. For this reason, we use simple linear interpolation for both translation and rotation, which nevertheless leads to satisfactory results.

In the simulation presented in this section the redundancy resolution is implemented in closed-loop. One solution to this problem is online transmission of a robot configuration to a workstation running a graphics visualization of the arm how serves as a virtual environment; the graphics model of the robot mirrors the exact motion of the arm, and the environment can be modeled in the graphics program. This approach has two main advantages:

- Any complex environment can be modeled with a desired precision (including a time-varying environment)
- The risk of damage to the robot is reduced.

Few different scenarios were selected to verify the performance of the obstacle-avoidance based redundancy-resolution strategy in executing the following tasks: position tracking, orientation tracking, stationary symmetric obstacles collision avoidance, joint limit, and self-collision avoidance. In each of these scenarios, one or multiple features were active at different instants of execution.

The simulation methodology uses possible scenarios and involves some sequences of animation, as one can see in the Fig.1, Fig.2 and Fig. 3 for a virtual assembly system with a planar manipulator. In this simulation, the end-effector is initially at rest. At the end of the free path the end-effector touches the bolt at the contact point with the head of the bolt.

The robot arm pushes the bolts in the symmetrical holes until the bolts touches constraint surface at the base of holes. After the contact between the end-effector and the bolt the success of the main task consists of keeping the position of the end-effector in the horizontal direction, while exerting a desired force in the same direction.

There is also the support object enclosed in area collision ellipse, in the workspace. The additional task consists of using the redundant degree of freedom to avoid the horizontal and vertical baseline of this support objects. The manipulator moves free until the arm is close enough to the support object. The obstacle avoidance task becomes active and makes the manipulator move in the null space of the Jacobian matrix to avoid collision.
The collision avoidance is here only presented for pure null-space movements. So only collisions of the virtual robot’s structure with the ”virtual collision object” are handled.

B Results of testing in virtual environment

The figures below show the results of the simulations for different poses of the robot arm corresponding to the collision avoidance task. In a practical implementation, no collision occurs if the maximum acceleration of each joint would be limited and this commanded joint acceleration would result in saturation of the actuators.

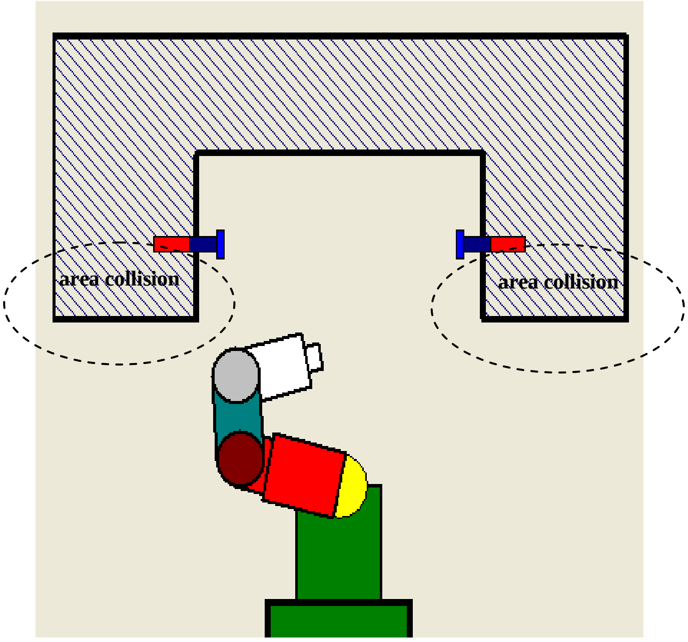

Fig. 3.1.1 Beginning pose of the animation sequence

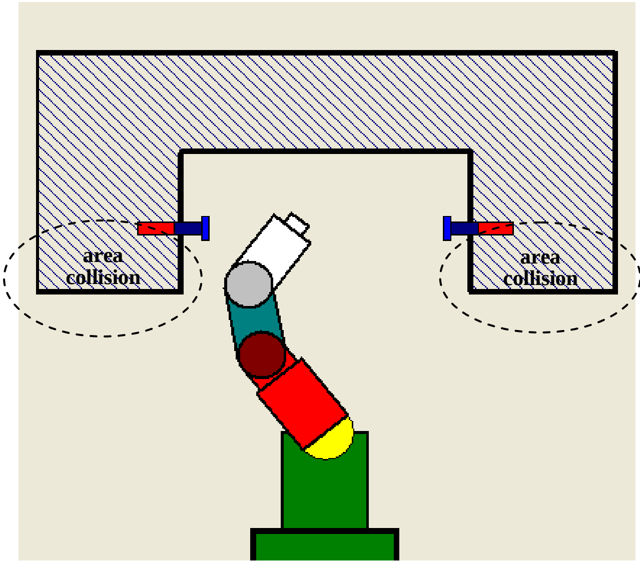

Fig.3.1.2 Middle pose of the animation sequence

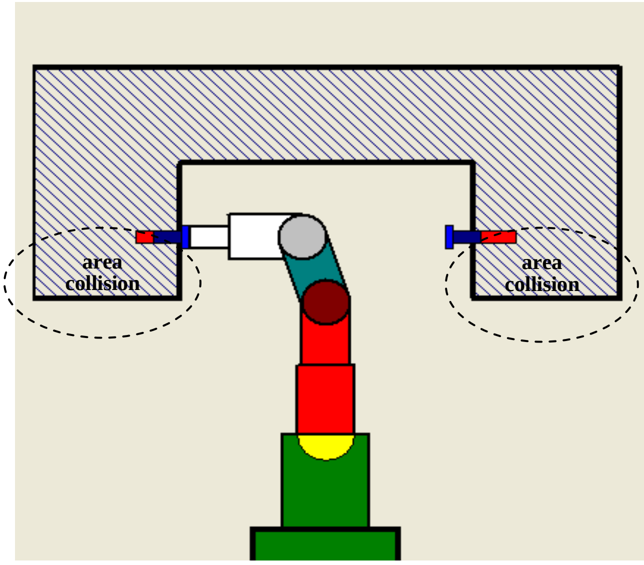

Fig. 3.1.3 Ending pose of the animation sequence

From preliminary simulations, it was observed that finding a maximum acceleration value which performs well in different situations is very difficult. To overcome this problem, a time-varying formulation must been used to adjust the maximum acceleration automatically.
We tested virtual architectures the same way as the reinforcement strategy. Virtual robot arm moves many times consecutively.

We consider as distance imposed in the constraint space the distance of end-effector from the environment horizontal baseline reference, in the X-axis and for vertical baseline, in Y-axis respectively. If this distance is between the 20 pixels from the horizontal baseline and vertical baseline respectively, we considered the trial as success. On the other hand, if end-effector position X -axis and Y-axis respectively goes away more than 20 pixels from the baseline at any move, we considered it as failure.
We have demonstrated the validity of the strategy using a scenario where linear constraints are imposed at lower priority task.

Conclusion
In this paper, we have considered a redundant inverse kinematics model based on the augmented Jacobian that enables the use of self motion of the manipulator to perform as sub-task the obstacles avoidance.
According to the task splitting concept in (primary) end-effector task and (secondary) constraint task, the concept was demonstrated on a virtual system. Interaction between the virtual robot and virtual obstacle was demonstrated, while performing end-effector tasks on the robot application.

The virtual robot is able to react to avoid the obstacles while executing the primary task. This was shown experimentally with a virtual system. This kind of robot's behavior simulation will be necessary in real applications, where interactions between robots and his environment are vital.

The performance of redundancy resolution in tracking the main task trajectories is studied here by computer simulation.
We can apply an command strategy from virtual system to real system without any complicated control theory or position sensors. Moreover, if we use different shaped obstacles it's very difficult to find proper control rules, but we can use the virtual robot learning by applying this strategy to get the satisfactory result with the real robot arm.

In this way, the kinematic simulation running on an workstation, generates the desired joint trajectory and this trajectory is then transferred as the joint set points to the robot's joints PID controllers. The real robot joints movement will imitate the virtual robot joints movement.
We wrote a program that allow to the end-effector of the virtual robot to wander around without bumping into obstacles in its path. The performance of the obstacle avoidance simulation strategy has been studied by various simulations for different scenarios. The performance of the collision avoidance algorithm was observed using the virtual models of the arm and of the object in a virtual environment.

In this paper, we implemented a very simple program that worked effectively. In the virtual environment not all simulation programs need to be complex to be effective. It is easy to imagine that, in a practical implementation, the software algorithms for real robot are highly complex.
A redundancy-resolution scenario was proposed to achieve obstacle avoidance. In the following simulations the main task consists of tracking the position and orientation trajectories, generated by linear interpolation, between the initial and final poses. It should be noted that interpolation of rotations is a much more complex problem than point interpolation [7]. For this reason, we use simple linear interpolation for both translation and rotation, which nevertheless leads to satisfactory results.

In the simulation presented in this section the redundancy resolution is implemented in closed-loop. One solution to this problem is online transmission of a robot configuration to a workstation running a graphics visualization of the arm how serves as a virtual environment; the graphics model of the robot mirrors the exact motion of the arm, and the environment can be modeled in the graphics program.

3.2 On-line collision avoidance using Programming by Demonstration

In this paper, based on original idea, the author proposes a new strategy to robot collision avoidance using programming imitation paradigm. To program the desired motion sequence for the physical robot, one captures the motion reference paths from her virtual robot model and maps these to the joint settings of the physical robot. Motion imitation requires transfer of a dynamical signature of a movement of the virtual robot to the physical robot, i.e. the robots should be able to encode and reproduce a particular path as one with a specific velocity and/or an acceleration profile. Furthermore, the virtual robot must cover all possible contexts - including la presence of accidental obstacles - in which the physical robot will need to generate similar motions in unseen context.

INTRODUCTION

Programming by Demonstration has appeared as one way to respond to growing need for intuitive control methods and is one of the most promising programming techniques for robotic manipulators. This technique allows even inexperienced users to easily program the robots based on the teaching by imitation paradigm.

This work follows a recent trend in Programming by Imitation. We present a new method to programming physical robot through the imitation of the virtual robot prototype.
Movement imitation requires a demonstrator. In our approach the demonstrator is the virtual robot prototype and the imitator is the physical robotic manipulator.
Conformity our method the dynamics of the motion of the virtual robots is reproduced by the physical robot.

In this paper one use the virtual robot prototypes and the motion capture systems to obtain the reference motion data, which typically consist of a set of trajectories in the Cartesian space.
We restrict our study to imitation of manipulative tasks. In particular, the ability to imitate virtual robots gestures and pursue task-relevant paths is essential skills for physical robotic manipulators, in the spirit of our method.

Inspired by the proposal patent [11] and based on the motion imitation concept this paper develops a general policy for robot motion programming based on virtual robot prototypes.
The strong point of the proposed method is that it provides fast on-line re-planning of the motion in the face of spatial-temporal perturbations as accidental obstacles and it allows generalizing a motion to unseen context based on particular scenarios.

This approach is focused on tracking joint angle trajectories, while some tasks may require tracking other quantities, such as end-effectors trajectories which will be addressed in future work. The objective of the present paper is giving an overview of a new robot programming method based on the virtual prototypes.

OVERVIEW OF PROGRAMMING BY IMITATION BASED ON NEW STRATEGY

Imitating the motion with stable robot dynamics is a challenging research problem [5]. Conventional programming strategies are hardly applicable to robotic manipulators that joints must be coordinated concurrently and must assure a stable dynamics [10]. Therefore, exploiting virtual robots' potential in programming tasks remains an atypical objective.

A characteristic feature of robot programming is that usually it is dealing with two different worlds; the real physical world to be manipulated and the abstract models - in particular virtual prototypes models - representing this world in a functional or descriptive manner by programs and data. In the simplest case, these models are pure imagination of the programmers; in high level programming languages, e.g. it may consist of CAD data.

Creating accurate robot path points for a robot application is an important programming task. It requires a robot programmer to have the knowledge of the robot's reference frames, positions, operations, and the work space.
In the conventional "lead-through" method, the robot programmer uses the robot teach pendant accessory to position and to orientation the robot joints and end-effector and record the satisfied robot pose.

The basic idea behind these approaches is to relieve the programmer from knowing all specific robot details and liberate him from coding every small motion.
Today's robot simulation software provides the robot programmer with the functions of creating virtual robot and virtual path points in an interactive virtual 3D design environment.

By the time a robot simulation design is completed; the simulation robot program is able to move the virtual robot end-effector to all desired virtual path points for performing the specified operation. By different scenarios one creates particular motion paths in the virtual space that are selected for particular motion paths in physical work space.

However, because of the inevitable dimensional differences of the components between the physical robot work space and the simulated robot work space, the virtual robot path points must be adjusted relative to the actual position of the components in the physical robot work space. This task involves the techniques of calibrating the position coordinates of the simulation device models with respect to the physical robot path points.
Learning by imitation based on our method represents a new research topic in robotics being a promising approach, towards effective robot programming.

FORMULATION OF THE IMITATION TASK

An imitation task can be decomposed into the serial implementation of two processes: an *observation process* and an *imitation process*. The observation process consists of extracting relevant (i.e. in our case, invariant over time) features from a *demonstrated dataset*.
The imitation process consists of generating an *imitated dataset* that minimizes the discrepancy between the demonstrated and imitated datasets.

Formalism
Let *S* be the dataset generated by the demonstrator while driven by a control strategy *U*. The control strategy *U* is such that $S(U)=\{\bar{\theta}_I,\bar{\theta}_D\}$, where $\bar{\theta}_D=\{\theta_D,\dot{\theta}_D,\ddot{\theta}_D\}$ (angular position, speed and acceleration) the trajectory of the demonstrator's arm joints and $\bar{\theta}_I=\{\theta_I,\dot{\theta}_I,\ddot{\theta}_I\}$ (angular position, speed and acceleration) the trajectory of the imitator's arm joints.
The imitation process consists of determining a control strategy *U'*, that generates a dataset $S'(U')=\{\bar{\theta}'_I,\bar{\theta}'_D\}$ such that the *cost function J*, or the *metric* of the imitation task, is minimal: $\delta J(S,S')=0$.

Each imitation task is defined by a set of constraints *C* with $\{c \in [1, C]\}$. For each constraint *c*, ϵ a control strategy *Uc*, generating a dataset S_{Uc}, such that the associated metric J_{Uc} is minimal [13]:

$$\delta J_{Uc}(S_{Uc},S'_{Uc})=0 \tag{3.2.1}$$

Metric of imitation

We express the metric or cost function of the imitation task *J*, as a linear combination of constraint-dependent cost functions J_{Uc} :

$$J(S,S') = \sum_{u=1}^{U}\sum_{c=1}^{C} w_{Uc}\, J(S_{Uc}, S'_{U'c}) \tag{3.2.2}$$

The weights of the constraints-dependent cost functions are proportional to the probability *P* (*Us*), that the dataset *S* has been generated by the control strategy U_c:

$$w_{Uc} = \frac{P(Uc)}{\sum_{i=1}^{U}\sum_{j=1}^{C} P(i_j)} \tag{2.3.3}$$

The weights are normalized, such that:

$$\sum_{i=0}^{U}\sum_{j=1}^{C} w(Uc) = 1 \tag{3.2.4}$$

Optimal imitation policy

We hypothesize that the constraints *c*, once identified during the observation process, remain the same during the imitation process. In the particular case where all control strategies are mutually exclusive, the optimal control strategy is unique [9].

If *P* (U'_c) is the probability that the imitated dataset has been generated by the control strategy U'_c, we have:

$$P(U_c) = P'(U'_c) \tag{3.2.5}$$

In the case where different control strategies can coexist, because they would be acting on separate sub-datasets (e.g. one strategy tends to minimize a constraint on the objects trajectories, while another strategy tends to minimize a constraint on the arm joint trajectories), then the optimal control strategy is the combination of all the optimal control strategies for each sub-dataset, such that:

$$P\,(U_C) = P'\,(U'_C) \quad \forall\ c \in [1, C]. \tag{3.2.6}$$

This probabilistic framework is highly suitable to the analysis and production joint trajectories, when using probabilistic methods.

Metric of size

Movement imitation requires a demonstrator. In our approach the demonstrator is the virtual robot prototype and the imitator is the physical real robot. The inputs of the imitation procedure are virtual robot prototype behavior captured, as virtual joint trajectories represented such a sequence of angular values of its joints. The imitator imitates the original virtual prototype captured motion, as closely as possible, and at the same time this respects they physical limits.

The virtual robot prototype behavior' problem will be studied by computer graphics, and we have actually much commercial graphic software that can solve it efficiently.

Generally, the virtual robot prototype is modeled in computer graphics with an approximation, and in addition its links lengths are largely different from those of physical real robot. Its size must be scaled to fit the physical robot size. However, in the motion imitation by a physical real robot, additional difficulties arise such as the joints variables, angular velocity and torques limits [4].

By taking into account those additional constraints, the motion imitation problem becomes the dynamic equation of physical robot arm:

$$M(q_r)\ddot{q} + N(q_r, \dot{q}_r) = \tau_r \tag{3.2.7}$$

The right term τ_r is the vector of the applied torques on the physical real robot's joints.
Permanent constraints are followings:

$$\begin{gathered} q_{r0} = q_o\,,\ \dot{q}_{r0} = 0\,,\ \ddot{q}_{r0} = 0 \\ q_{rf} = q_f\,,\ \dot{q}_{rf} = 0\,,\ \ddot{q}_{rf} = 0 \\ q^- \le q_r \le q^+\,, \quad q^- \le \dot{q}_r \le q^+ \quad, \tau^- \le \tau_r \le \tau^+\,, \end{gathered} \tag{3.2.8}$$

where q^-, q^+ and denote the minimal and maximal values of vector q_r and vector τ_r respectively.

Solving the optimization problem, in the case of above constraints, is generally difficult.
The explanation of the involvement of the dynamic equation of motion and the desired virtual path points for performing the specified operation becomes implicit relation between the vector of applied torques τ_r and the vector of joints positions q_r of the physical robot.

PROGRAMMING FROM A SIMULATED VIRTUAL MODEL

We present a description of the theoretical aspects of the programming from a simulated virtual model and of the strategy which is at foundation of this approach. The advantages of such programming approach as an alternative to the classical methods (e.g. vision guided trajectory imitation) are on-line adaptation to the motion of the virtual prototype.

A solution to the above problem is to construct a virtual prototype model and transfer the virtual trajectory by interacting with the physical model.
Designing a model would be an option; however, the behavior of the robots is very difficult to model. Moreover, the use of system knowledge is contrary to our research aim. Therefore we focus on creating a virtual prototype model from experimental data obtained from the physical robot model.
Thus, we will prove that our method guarantees the motion optimization for each robotics tasks.
The planning package communicates primarily with simulation environment. A planning module can send messages to the simulation system such as computed plans for the robots. The planning module can further send trajectory and planning structure information to visualization so users can see the results of an algorithm.
The planning module also receives control signals from the simulation module, such as when to start planning joint trajectories [2].

The visualization module is in charge for visualizing any aspect needed by the programmer for the optimization process. Users interact with the simulation environment through the visualization. This includes, but not limited to, computer screen. Optimization of the real robots behavior is performed in the low dimensional virtual space using the virtual robot prototypes. In the virtual space one simulate even the intersecting of the virtual robot and her environment. The intersecting of two virtual objects is possible in the virtual world, where the virtual objects can be even intersected and there no exist the risk to be destroyed.

The contact detection between the virtual objects is responsible for determining the distance between two points in a space and for optimization of the trajectories of motion. This strategy may use simple CAD methods or may be extended to be more complex and take into account invalid areas of the space. The visualization provides an interface to develop interactive implementations based on imitation strategy.

In our work, we assume that learning of the deterministic part for description motion dynamics should be sufficient to design the corresponding robot control.
We particularly refer to the ability of the system to react to changes in the environment that are reflected by motion parameters, such as a desired target position and motion duration. Therefore, the system is able to manage with uncertainties in the position of a manipulated object, duration of motion, and structure limitation (e.g., joint velocity and torque limits).
The proposed method aims at adapting to spatial and temporal perturbations which are externally-generated. This aspect will be investigated in our future works.

ONLINE IMITATION OF THE DYNAMICAL SYSTEMS

In robotics, one of the most frequent methods to represent movement strategy is by means of the learning from imitation. Imitation learning is simply an application of supervised learning. One goal of imitation of the dynamical systems is to use the ability of coupling phenomena to description for complex behavior [6].

In this paper, we propose a generic modeling approach to generate virtual robot prototype behavior in experimental scenery. The actions for the each task are computed for virtual robot prototype and are transferred online, with a central coordination, to corresponding physical robot, which must imitate her virtual “homonymous”. Notice the similarity between moves of the virtual robot prototype in the virtual work space and the “homonymous” moves in the real work space of the physical robot. We assume to use the virtual robot prototypes and the motion capture systems to obtain the reference motion data, which typically consist of a set of trajectories in the Cartesian space.

The paper relates to a method and a robot programming platform by combining off-line and on-line programming techniques. The method consists in using a programming platform on which there is carried out the virtual prototype of the physical robotic arm to be programmed and the real working space wherein it is intended to work.

In the robot program there is written a source code intended to generate the motion paths of the virtual robotic arm prototype. The numerical values of the prototype articulation variables are sent to the data register of a port of the information system which, via a numerical interface, are on-line transferred into the data registers of the controllers of the actuator of the physical robotic arm. Finally, there are obtained tracking structures due to which the moving paths of the virtual robotic arm joints are reproduced by the physical robotic arm joints, thereby generating motion within the real working space.

Imitation learning from our strategy demonstrates how to obtain dynamical virtual models with CAD systems. Those online adjusted virtual models are among the most important properties offered by a dynamical systems approach, and these properties cannot easily be replicated without the feed-back from physical robot of our proposed structure.

The objective of a movement is to generate a reaching movement from any start state to a goal state [8]**.** The discrete dynamical system is initialized with a minimum movement, which is frequently used as an approximate model of smooth movement.
The proposed structure uses a virtual demonstrator for planning the movements of the physical robot to avoid the obstacles. We investigate the potential of imitation in programming robotic manipulators with multiples degrees of freedom when the associated joints must be coordinated concurrently.

The imitation strategy consists in a proportional real-time mapping between each virtual joint and the corresponding physical joint. The system requires the programmer to perform an initial calibration routine to identify the range of motion for each movement. Each range is divided into as many intervals as the number of feasible discrete configurations of the corresponding joint.

EXPERIMENTAL CONFIGURATION FOR ROBOT PROGRAMMING BY IMITATION

The approach was implemented using a simulation software platform called Robot Motion Imitation Platform (RMIP) developed at University of Brasov [11], who its feasibility has been demonstrated in some simulation settings for manipulative works, using Virtual Reality.

We extended the simulation software platform to programming a physical robot. We see them as intelligent structure that can be used to transfer - in real time - the gesture of the virtual robot to the physical robots for generating complex movements in real work space. One transfers via intelligent interface, the virtual joint angles data from a motion capture system to a kinematic model for a physical robotic manipulator.
The RMIP is an architecture that provides libraries and tools to help software developers in programming. Platform RMIP is focused on 3D simulation of the dynamics systems and on a control and planning interface that provides primitives for motion planning by imitation.
Also is a object-oriented infrastructure for the integration of controllers, as well as the integration of planners with controllers to achieve feedback based planning, In particular, RMIP is used to provide concrete implementations of motion planners.

The platform RMIP utilizes a framework which allows both high level and low level controllers to be composed in a way that provides complex interactions between virtual and physical robots. In our strategy we just included the possibility of functional coupling virtual and physical robot model. The functional coupling consists to map of the motion of the demonstrator robot arm with the imitator robot arm. The former is displayed on the screen, giving the visual information to programmer and is connected to the latter through intelligent interface. In our experiments we have adapted as imitator a small robotic manipulator, named EXMAN and presented in Fig. 3.2.1.

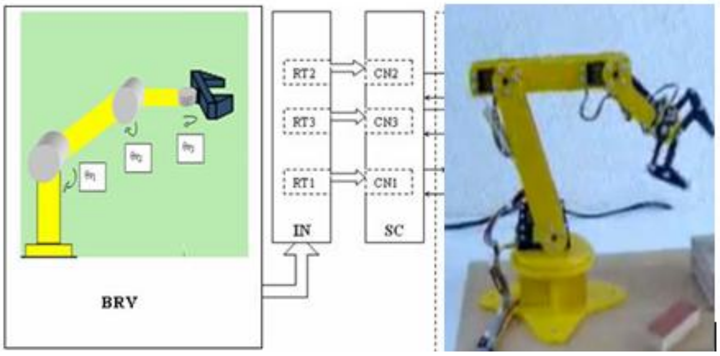

Fig. 3.2.1 Experienced robotic manipulator

We start with a 3 degree-of-freedom (DOF) discrete movement system that models point-to-point attained in a 3D Cartesian space.
The arm moved in a free space without obstacles, had three active joints (shoulder, elbow and wrist) and was driven by electrical actuators.
Figure 3.2.2 shows our experiment involving the imitation learning for a physical robotic arm with 3 degrees-of-freedom (DOFs) for performing the manipulate tasks. We demonstrated the imitation of elbow, shoulder and wrist movements. Importantly, these tasks required the coordination of 3 DOFs, which was easily accomplished in our approach.

The imitated movement was represented in joint angles of the robot. Indeed, only kinematic variables are observable in imitation learning. The robot was equipped with a controller (a PD controller) that could accurately follow the kinematic strategy (i.e., generate the torques necessary to follow a particular joint angle trajectory, given in terms of desired positions, velocities, and accelerations) [7].

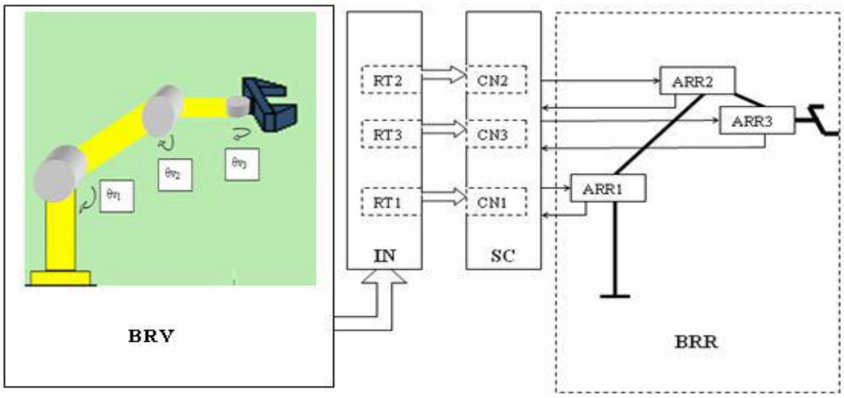

Fig. 3.2.2 Imitation software platform structure

To generate the motion sequence for the real robot, one captures the motions from a virtual robot model and maps these to the joint settings of the physical robot. Initially, a set of virtual desired postures is created to the virtual robotic arm BRV and the pictures' positions are recorded for each posture, during motion.

These recorded pictures' positions provide a set of Cartesian points in the 3D capture volume for each posture. To obtain the physical robot postures, the virtual pictures' positions are assigned as positional constraints on the physical robot. To obtain the physical joint angles one use standard inverse kinematics (IK) routines. The IK routine then directly generates the physical joint angles on the physical robot for each posture.

Referring the Fig. 2 we comment the following: on programming platform, a robot program is carried out off-line, and one sends into the data registers of a port of the hardware structure, the numerical values of the joint variables of the virtual prototype of the robotic arm (BRV) and displays on a graphical user interface, the evolution of the virtual prototype during the carrying out of the robotic task. Via numerical interface (IN) the virtual joint dataset, from the data registers of the port of the hardware structure of the programming platform are transferred into the data registers of the numerical comparators of the controllers. These datasets are reference inputs of the pursue loops, resulting a control system (SC).

The virtual tracking paths of the virtual joints are reproduced in the real work space by the actuators of the physical robotic arm. The reference datasets are obtained using a motion capture channel taking into account the joints motion range. The symbolic spatial relations specifying the virtual work space can be used for the automatic pursuits of possible virtual path as well as for planning of appropriate behavior of the physical robot arm BRR, which may guide the motion process during task execution.

The easiest way to generate the spatial relations explicitly is the interactively programming of the behavior of the virtual prototype in his virtual environment, in order to specify suitable positions θ_{v1}, θ_{v2}, θ_{v3}. This kind of specification provides an easy to use interactive graphical tool to define any kind of robot path; the user has to deal only with a limited and manageable amount of spatial information in a very comfortable manner.

The robot programming system has to recognize the correct robot task type and should map it to a sequence of robot operations [11]. The desired pathways are automatically transferred and parameterized in the numerical interface IN, using the path planner.

The applicable robot tasks are designed and the desired pathways are programmed off-line and stored in the buffer modules RT1, RT2, RT3.The comparative modules CN1, CN2, CN3 furnish, to the pursuit controllers, the datasets involving the expected state of the virtual robot prototype and the measured state of the physical robot.

While motion execution is in progress, the real robot joints ARR1, ARR2, ARR3 are activates into the real work space. Each actuator was connected by a sensor in the closed-loop. Each time, a skill primitive is executed by the robot control system SC; it changing the robot joints position. As no time limit for the motion is specified, the physical robot imitates the behavior of the virtual robot.

In our laboratory currently we are developing Cartesian control architecture able to interpret the physical robot commands in the above given form. The basis of our implementation is a flexible and modular system for robot programming by imitation.
In our experimental configuration in order to prove the correctness of the robot programming by imitation we have chosen an robotic manipulator equipped with electrical actuators, mounted on the physical robot’s joints.

The robot’s control unit is connected via TCP/IP to a PC equipped with the interface card; the PC is running the simulation and control process. The robot control system receives and executes each 16 ms, an elementary move operation.
Due to the kinematics limitations of physical robot, the resolution of the joints is quite limited. After the calibration phase, physical robot starts imitating the gestures of the virtual robot demonstrator. The user can decide to activate all the three degrees of freedom concurrently or with a restricted subset by deactivating some of the sensors attached to each joint of the physical robot. Our system requires an essential step in that one converts the position errors into motor commands by means of the PD controller [7].

Figure 3 shows the structure of the system used for joint motion imitation. One can see the interest hardware components their interconnections for closed-loop control, associated of robot joint number one. The sensor TP1 is in close proximity with the actuator ACT1 and is interconnected with hardware components specific for a closed-loop.

We aim at developing controllers for learning by imitation with a single joint robot demonstrator. For this purpose, we assume a simple control system where the position of the 1 DOF discrete dynamical system, drives the time evolution of joint variable θ_{r1}, which can be interpreted as the position controlled by a proportional-derivative controller. All other DOFs of the arm demonstrate the same behavior.

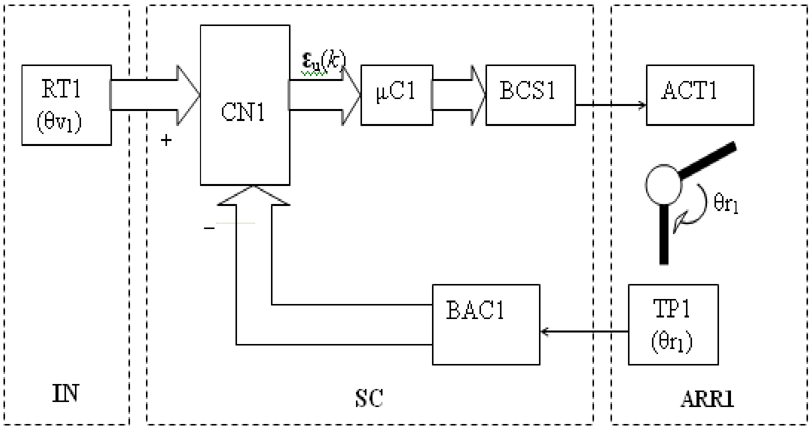

Fig. 3.2.3 Closed-loop motion control for robot joint

Torque is indirectly controlled via joint dynamics. The internal joint states were unknown and not available to the controller. Only joint variable was available by the joint sensor. The joint kinematics is derived from the CAD model. In the motion imitation additional difficulties arise such as the joints angular velocity and torques limits.

RESULTS-EVALUATIONS

In this paper, we presented and reviewed several experimental evaluations of applying our approach in the domain of motion programming based on virtual robots prototypes, using both simulation and robotic control studies.
The evaluations were intended to demonstrate the properties of our methodology, but also the domain-specific choices that need to be made. We have discuss about imitation learning of discrete movement, online imitation with the help of coupling parts, synchronization phenomena, and movement recognition based on a graphical interface.

Experiments involving gesture imitation were carried out to test the effectiveness of the system. The evolution of the references signals, sent to the physical robot at level of each degree of freedom, has been correlated with the resolution of the corresponding virtual “homonymous” degree of freedom. In consequence the physical robot joints follow with acceptable fidelity imposed references signals values.

The platform RMIP is based on original idea patented by author. Abstract of the proposal patent illustrating this method can be viewed at reference [14].
One expected that this method, being user-friendly, would enhance the application of programming robots in dynamic environments. We anticipate that these promises may be fulfilled very soon.
The imitation approach illustrated in this paper is rather general, even though its implementation is constrained by limitations of RMIP. In particular, the paper reports successful experiments on gesture imitation, including arms motion based on joints’ movements.

This paper is focused on the programming by imitation, transferring of the motion mapping from virtual space in 3-D dimensional real physical space. We believe that a deep evaluation of a robot programming platform is extremely important for the robotic research community since the technological requirements and the costs to develop more sophisticated programming platform still prevent them to become generally available. Currently, most high-end programming platform are developed as prototypes platforms under the supervision of academic community. Therefore, our platform RMIP such as low-cost programming platform provides an exciting and inexpensive opportunity for research in programming by imitation.

The programming method, based on the imitation of virtual prototypes, must be still enhanced to respond at the question: how dynamical virtual system models can be up-dated online to take dynamic events, from the real environment, into account? The author expect fully automated robot programming by imitation, using robust enough system to be applied in practical applications, will not become true before the end of this decade.

ACKNOWLEDGMENT

The author wish to thank, for cooperation and engagement in research activity, the entire team of Services and Products for Intelligent Environment Laboratory, within the Research & Development Institute ICDT-PRO-DD of the Transylvania University of Brasov. They granted us the opportunities to work in team with them and provided us important experiences to extend this interesting research.

Chapter 4 Modelling and robots behavioural simulation in visual environments

4.1 System concept development with virtual prototyping

Introduction

Through careful development of details, 3D graphics applications can generate virtual prototypes that provide a useful tool supporting engineering and specialty disciplines analysis for a wide variety of system developments, both early in the concept development phase and later in advanced engineering. Virtual prototyping permits users, designers, and logisticians to visualize two and three dimensional relationships and clearances for joint multi-disciplinary analysis.

This early analysis and identification of design issues offers potential to reduce development costs since mistakes will be made in the computer rather than on full-scale prototypes. Due to design maturity, virtual prototypes developed in advanced engineering development will be based on design analysis and known qualities.

In contrast, exceptional care must be taken during the technology assessment and advanced concept phases, when details are less certain, to conduct thorough research that produces and supports a credible product based on expert opinion, technology analysis, and traceable assumptions rather than "fantasy." The development of worksheets containing source and rationale information on each virtual component is a disciplined technique that helps to ensure reality is not misrepresented.
In manufacturing, rapid prototyping is used to create a three-dimensional model of a part or product.

In addition to providing 3-D visualization for digitally rendered items, rapid prototyping can be used to test the efficiency of a part or product design before it is manufactured in larger quantities. Testing may have more to do with the shape or size of a design, rather than its strength or durability, because the prototype may not be made of the same material as the final product. Today, prototypes are often created with additive layer manufacturing technology, also known as 3-D printing. Direct metal laser sintering (DMLS) may also be used to create aluminum, stainless steel or titanium prototypes. This process uses laser beams to melt and fuse metal powders into solid parts.

In software development, a prototype is a rudimentary working model of a product or information system, usually built for demonstration purposes or as part of the development process. In the systems development life cycle (SDLC) Prototyping Model, a basic version of the system is built, tested, and then reworked as necessary until an acceptable prototype is finally achieved from which the complete system or product can now be developed.
In prototype-based programming, a prototype is an original object; new objects are created by copying the prototype.

Virtual prototyping, often known as VP, is a software-based engineering discipline which involves modelling a system, simulating and visualising its behaviour under real-world operating conditions, and refining its design through an iterative process. The full-motion behaviour of complex mechanical systems can be analysed before building an actual hardware prototype. Users can quickly explore multiple design variations, testing and refining until system

performance is optimised. This can help reduce the time and cost of new product development, whilst significantly improving the quality of overall system designs.
In hardware design, a prototype is a "hand-built" model that represents a manufactured (easily replicable) product sufficiently for designers to visualize and test the design.

Prototyping Model

The Prototyping Model is a systems development method (SDM) in which a prototype (an early approximation of a final system or product) is built, tested, and then reworked as necessary until an acceptable prototype is finally achieved from which the complete system or product can now be developed. This model works best in scenarios where not all of the project requirements are known in detail ahead of time. It is an iterative, trial-and-error process that takes place between the developers and the users.

There are several steps in the Prototyping Model:

1.The new system requirements are defined in as much detail as possible. This usually involves interviewing a number of users representing all the departments or aspects of the existing system.
2.A preliminary design is created for the new system.
3.A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.
4.The users thoroughly evaluate the first prototype, noting its strengths and weaknesses, what needs to be added, and what should to be removed. The developer collects and analyzes the remarks from the users.
5.The first prototype is modified, based on the comments supplied by the users, and a second prototype of the new system is constructed.
6.The second prototype is evaluated in the same manner as was the first prototype.
7.The preceding steps are iterated as many times as necessary, until the users are satisfied that the prototype represents the final product desired.
8.The final system is constructed, based on the final prototype.
9.The final system is thoroughly evaluated and tested. Routine maintenance is carried out on a continuing basis to prevent large-scale failures and to minimize downtime.

The RAD method has a task list and a work breakdown structure that is designed for speed. However the major difference in RAD is a set of management techniques that are optimized for speed.
Prototyping - an approach based on creating a demonstrable result as early as possible and refining that result. The refinement is based on feedback from the business, the eventual users of the system. Prototyping requires an open approach to development; it also requires an emphasis on relationship management and change management. There are dangers involved in starting prototype development too early and in starting it too late.

Rapid application development (RAD) is a concept that products can be developed faster and of higher quality through:

- Gathering requirements using workshops or focus groups
- Prototyping and early, reiterative user testing of designs
- The re-use of software components
- A rigidly paced schedule that defers design improvements to the next product version
- Less formality in reviews and other team communication

Object-oriented programming (OOP) is a programming language model organized around objects rather than "actions" and data rather than logic. Historically, a program has been viewed as a logical procedure that takes input data, processes it, and produces output data.

Simula was the first object-oriented programming language. Java, Python, C++, Visual Basic.NET and Ruby are the most popular OOP languages today. The Java programming language is designed especially for use in distributed applications on corporate networks and the Internet. Ruby is used in many Web applications. Curl, Smalltalk, **Delphi** and Eiffel are also examples of object-oriented programming languages.
Programmers and analysts can build and show visual representations of the designs and workflow to users.

- Prototype that contains only some of the essential system features.
- Users can interact with the system.
- Prototypes are useful in seeking user reactions, suggestions, innovations, and revision plans.
- Prototyping may be used as an alternative to the systems development life cycle.
- Users respond to actual working prototypes.
- Analysts refine designed modules based on user responses.
- A system is tested and modified as necessary, and later is implemented

In object-oriented programming (OOP), objects are the things you imagine about first in designing a program and they are also the units of code that are eventually derived from the process. In between, each object is made into a generic class of object and even more generic classes are defined so that objects can share models and reuse the class definitions in their code. Each object is an instance of a particular class or subclass with the class's own methods or procedures and data variables. An object is what actually runs in the computer.

Visual prototyping
Virtual prototyping with 3D drawing programs provides a means of rapidly developing system concepts and analyzing them for form, fit logistics, human factors integration, and general feasibility analysis. The resulting models can be studied, viewed from different angles, and even "entered" by multidisciplinary design teams working in an integrated product team environment. Virtual prototyping is an aspect of information technology that permits analysts to examine, manipulate, and test the form, fit, motion, logistics, and human factors of conceptual designs on a computer monitor. It facilitates communication between different engineering disciplines during the early design process, and also provides quality illustrations that help "sell" the design or program.

A virtual prototype is defined as "A computer-based simulation of a system or subsystem with a degree of functional realism comparable to a physical prototype" and virtual prototyping as "The process of using a virtual prototype, in lieu of a physical prototype, for test and evaluation of specific characteristics of a candidate design.

Virtual prototyping can replace the expensive physical model constructed to test designs. The value of virtual prototyping is rapidly being recognized for a wide range of engineering applications. These applications range from illustrating the potential of a system early in the technology assessment activity or early developmental phase to detailed analysis of mature designs in the advanced engineering phase. Some of the uses include design form, fit, and function, logistics and human factors analysis.

Virtual prototypes are developed from concept drawings or blueprints, and contain as much detail as required to support the immediate need. The same model can be expanded later as more detail is required. In the computer, the various parts are developed to scale and then are assembled the same way they would be on a fabrication shop floor. This allows the analyst to test assembly procedures and human factors as well as the form and fit of the final product. It also allows parts to be removed to show otherwise hidden components.

The model appears realistic and may be feasible, but it is not backed up with engineering projections for load bearing features. Taking such liberties during the projection of new technology into concept systems would be unethical and just poor engineering analysis. To maintain credibility, virtual prototypes at all stages of engineering development must be based on disciplined analysis and sound technology projections. Construction details must be backed up with engineering analysis to provide value and credibility.

Virtual prototypes can enhance the communication of the answers to these questions and provide a credible baseline for discussion and analysis. In order to support the program, virtual prototypes must be based on sound engineering and projections of technology growth that are traceable to sources such as programs developing or producing related hardware, expert opinion, field and laboratory tests, and engineering analysis. To produce a high quality accurate model of a technology projection that is credible and useful, each depicted feature of the model must be traceable to one or more of these sources.

Through careful development of details, 3D graphics applications can generate virtual prototypes that provide a useful tool supporting engineering and specialty disciplines analysis for a wide variety of system developments, both early in the concept development phase and later in advanced engineering. Virtual prototyping permits users, designers, and logisticians to visualize two and three dimensional relationships and clearances for joint multi-disciplinary analysis. This early analysis and identification of design issues offers potential to reduce development costs since mistakes will be made in the computer rather than on full-scale prototypes.

Due to design maturity, virtual prototypes developed in advanced engineering development will be based on design analysis and known qualities. In contrast, exceptional care must be taken during the technology assessment and advanced concept phases, when details are less certain, to conduct thorough research that produces and supports a credible product based on expert opinion, technology analysis, and traceable assumptions rather than “fantasy.” The development of worksheets containing source and rationale information on each virtual component is a disciplined technique that helps to ensure reality is not misrepresented.

Virtual prototyping is a software-based engineering discipline that entails modelling a robotic system, simulating and visualising its 3D-motion behaviour under real-world operating conditions, and refining/optimising the design through iterative design studies prior to building the first physical prototype.
Thus, at its most basic level VP is a tool for enabling engineers, designers and product developers to work together concurrently within a virtual environment to solve design, manufacturing and maintainability issues at the earliest stage of product development.
It represents a design capability, which allows users to predict and prevent problems early in the product-development process rather than finding and fixing them later on, a situation that can substantially reduce product-development costs.

The adoption of tools that help engineers eliminate product flaws at the earliest stages of development also helps organisations to meet critical time-to-market objectives, enabling them to maximise their profit margins.

Virtual prototyping allows engineers and designers to utilise CAD data and techniques to construct interactive simulations that model the key aspects of the product’s physical behaviour, all at the ‘digital’ development stage. This allows for product testing at the earliest moment possible, which has beneficial consequences of the cost of getting the design to market.

Visual prototyping as a technology has evolved in line with technical advances in computing to become an invaluable tool also in robotic systems area. With the Internet now a powerful global communications tool, new product visualisation and dynamic testing can be performed online. Ideally, a virtual robot can be viewed, listened, smelled, and touched by an engineer or a customer.

This is the area that virtual reality techniques can play an important role. More importantly, various perspectives of the designed product should be able to tested and evaluated. Visual prototyping provides a means to quantitatively describe product behavior from various aspects and thus could possibly be used as a fundamental tool to support a quantitative concurrent design.

Simulation has been recognized as an important tool in robotics in designing new robots, investigating their performances and in designing applications of these robots. Simulation allows us to study the structure’s characteristics and the functions of a robot systems at different levels of details, each posturing different requirements for the simulation tools. For example a fast process, as the moving robot joint, can be slowed down to observe all details in “slow motion”. All these make things easier and cheaper.

As the complexity of the system under investigation increases, the role of the simulation becomes more and more important. That's why the simulation tools can certainly enhance the design development. Depending on the robot application, different structural attributes and functional parameters have to be modeled

A large amount of simulation software is available for robot systems, and it is already being used extensively. The majority of the robot simulation tools focus on the motion of the robotic manipulator in different environments. As the motion simulation has a central role in all simulation systems; they all include kinematics or dynamic models of robot manipulators. Which type of models will be used depends on the objective of the simulation technique.

For example, trajectory planning algorithms depend on kinematics models. Similarly, the construction of a robotized cell can be simulated efficiently by using only kinematics models of robot manipulators, without considering the dynamics or drives.

On the other hand, to design the actuators, dynamic models are needed. Modern control systems of robotic manipulators use internally different robot kinematics and dynamic model, to improve the performances.To model and simulate a robot system different approaches are possible. They can differ in the way the user builds the model.

Block diagram-oriented simulation software requires that the user describes the system by combining the blocks but there are other techniques requiring the manual coding.

The simulation tools for robotic systems can be divided into two major groups: tools based on general simulation systems and special tools. Tools based on general simulation systems are usually modules libraries or user interfaces, which simplify the building of virtual robot systems and environments (Fernandez-Madrigal et al., 2007). One of the advantages of such integrated toolboxes is that they enable us to use other available simulation tools to perform different tasks such as to design control system, to analyze simulation results, to visualize results, etc.

There exist several general simulation tools which are used for simulation of robot systems like MATLAB Robotics Toolbox (Corke, 1996), Dymola / Modelica, 20-sim, etc.

Special simulation tools for robots cover one or more fields in robotics like mechanical design, kinematics (geometry) and dynamics analysis, design of robot work cells, off-line programming. They can be specialized for special types of robots like mobile robots, underwater robots, parallel mechanisms, or they are assigned to predefined robot family such as KUKA sim, Fanuc Simulator etc.

Multi-body dynamic engines

A physics engine is a computer program that simulates the physics models which respect the laws of physics. It can simulate and predict effects under different conditions that would approximate what happens in real world.

In robotics, physics engines have two core components, a collision detection system, and the dynamics simulation system responsible for animating the objects. In the last years new simulation tools have been available based on general engines for the simulation of physics environments. These engines provide libraries for simulating the multi-body of an assembly with constrained or restrained bodies. As examples for libraries to simulate the multi-body dynamics we have selected the Open Dynamics Engine (ODE) and Open Graphics Library (OpenGL).

ODE is an open source, high performance library for simulating articulated rigid body dynamics. It is fully featured, stable, mature and platform independent with an easy to use C/C++ *Application Programming Interface* (API).
ODE is useful for simulating objects in virtual reality environments and virtual robots. With ODE building the model of a robot is simple.

OpenGL is a standard specification defining a cross-language cross-platform API for writing applications that produce 2D and 3D computer graphics.

As example to simulate the multi-body dynamics with learning environment we have selected the BehaivourSim. BehaivourSim is a learning environment for behaviour-based agent. It supports learning of behaviour-based control by defining simulated agents (Wooldridge et al., 2002) in an intuitive manner corresponding to the behaviour-based paradigm. Behaviour-based control is one of the fundamental control paradigms for autonomous agents to realize adaptive behaviour in a dynamical environment.

However, the diversity of the tasks to be done carries out the study of other types of robots which may have new properties, whose parameters differ from what is usually known. In order to enlarge the possibilities of such systems, it may be interesting to relax some constraints on the parameters using the physics engine for simulation technique.

Modelling and robots behavioural simulation in visual environments

There are two large structural classes of robots: serial robotic structures and parallel robotic structures. Simplicity considerations in manufacturing and control have led to robots with revolute or prismatic joints only. Also, on the same considerations joint axes are placed orthogonal or parallel instead of being placed arbitrarily. Therefore, a variety of simulation tools has been developed for the robotic systems. The simulation tools are used in mechanical design of robotic manipulators, design of robot control systems, off-line programming systems etc.

To overcome the problems which arise when the system is very complex - and the robots usually are very complex systems - several approaches exist to automatically generate the kinematics and/or dynamic models of robots.

Delphi simulators

In this paper the authors propose the Delphi informatics environment, for the qualitative simulation of the robotic systems, using the visual programming. The prototype graphical simulation system has been developed to support and demonstrate different aspects of robots behaviour.

Delphi is a visual programming tool which makes possible to carry out, in a practically instantaneous way, the interface with Windows applications. Having a very intuitive access, this extremely rich tool offers with effortlessness the management of an environment as sophisticated as Windows.

Delphi language comes from the Pascal, witch is object oriented, just like C++. This one is very simple to learn, very evolutionary and very intuitive. The simulator created in the Delphi environment was used to test the performances of the mechanical system and the integrated environment. The simulator is created with the DELPHI code, which allows us to create the virtual prototype.

In robotics, the education of engineers demand a well structured understanding of what exactly a robot is. The virtual robot model gives exactly this understanding. Model driven simulation is a valuable tool for understanding the properties of a robot. Delphi system encourages users to make their own program expansions and simulate them. Doing this, new ideas can be tested on the robots or on the production systems.

In general, the simulation can be used to perform analysis and design studies on any robot system which can be modelled as a set of rigid bodies interconnected by joints, influenced by forces, driven by prescribed motions, and restricted by constraints.

For simulation in Delphi, one supposes the pure motion, without reference to the masses or forces involved in it. It is the case of the kinematics model that studies the motion of a robot arm, without considering its mass or the forces which are acting on it. This model deals with the characteristics of motion, without regarding the effects of forces or mass (Fleury et al., 2006).

In traditional robotics, programmers attempted to anticipate and explicitly control every aspect of the action of the robot. However, these systems often failed when are placed in a changing environment, where unanticipated situations appear. Systems using autonomous agents, with de-centralized control of robot motion, enable robots to learn and adapt to changing environments.

Autonomous agents are software programs designed to pursue goals and can do so without user intervention. They are usually large applications, which attempt to solve practical industrial problems, and require a large amount of investment and development. Advanced robotic systems involve large agent-based simulators aimed at understanding "action selection". By this mechanism the robot selects the behaviour to execute from much possible behaviour.

Delphi simulators are generally smaller programs, and they are written in order to extend research in Robotics. They usually involve algorithms that allow exposing the behaviour of the virtual robots and their environment. This technique can also be used to provide a method of robotic system command.

All of these programs enable the manifestation of global behaviours. They also attempt to simulate more aspects of robot behaviour - not just movement. As such they include the movement of robot manipulators, the contact with manipulated objects and the collision with unexpected objects into the work environment. The information regarding the collision are given about the shape of each body and then it figures out which bodies touch each other.

The Delphi programs are written with an object-oriented approach. The simulation is driven by a special object, the Object Inspector. The Object Inspector maintains a list of the active object and sends step messages when it is time for objects to update themselves. All components of the artificial world are objects, as is the environment itself. The interface consists of a collection of tools for analysis and visualization.

Data analysis objects can communicate with the user interface objects to present displays of data, as well as for storage to files. The prototype allows the user to create, modify, and destroy objects. Each object has a few standard attributes managed by Object Inspector, as well as user-specified private data.

The requirements of the simulation to support such research activity are:

- This simulation should be able to show visually how behaviour can evolve.
- It will allow the testing of different theories of robots control, and should enable experimentation in the “hardware-in-the-loop” simulation mode.

The aim of Delphi-simulation is to contribute to a better understanding of robot behaviour. Delphi-simulation can be used to produce virtual applications, making them accessible to new kinds of experimental manipulation and testing. The simulator should be able to present a changing environment in which robotic systems are continually changing over time.

The Delphi system uses a library tool, and a system to run the simulation. The simulators are still under development, although prototype versions have been produced. We present some typical simulation examples in different fields of robotics.

4.2 Virtual Prototyping for robotic systems

The purpose of the paper is to emphasize the role of the simulation in different fields of robotics. Also we recommend to researchers to use the general dynamic engines and general visual programming languages for the simulation and visualization of robotic systems.

This paper describes the development of a robotic language and the creation of a virtual reality system for the simulation of robotic systems. We propose a description language for specifying motions and for allowing robots to acquire motor skills. Locomotion greatly increases our ability to interact with our environments, which in turn increases our mental abilities. This principle also applies to intelligent robots. However, there are great difficulties to specify path motion and to represent motor skills, which in most cases require four-dimensional space representations. We propose a representation framework that includes the following attributes: motion description layers, reference system, progressive quantized refinement, and automatic constraint satisfaction. We also outline strategies for acquiring new motor skills by learning from trial and error, virtual motion path, and programming. This paper discusses the using of the visual programming Delphi environment, for facilitating intuitive robot programming. From a robotic point of view, the virtual robotic structures created were simplified and their analyzing allows a better description of the behaviour of real robotic structures.

The simulations prepared for this application were also made for the development of the systematic methods for creating the new robotic systems.

The results of this investigation consist in an identification of the properties and constraints of the virtual robot prototypes. The virtual structures created in this paper encompass the behaviour of robotic systems. They may present a behaviour which is qualitatively different from the behaviour that the robots currently have.

In the end, the simulation in robotics has reached a very important role and by using different simulation software, the present and future capabilities of complex robotic systems can be significantly improved.
The simulation is now a powerful tool supporting the design, planning, analysis, and decisions in different areas of research and development. Simulation has become a strategic tool in robotics as a modern technological branch. The simulation is very common in our lives to understand reality in all its complexity. We try to build artificial objects/systems and to animate them to understand their behaviour.

Advanced simulation tools are the foundation for the design of sophisticated robot systems, for the application of robots in complex environments and for the development of new control strategies and algorithms.
The simulation, being once a tool for the analysis of the robotic systems and the task planning, has become a technique for developing new robot systems. Not only that the modern simulation

tools can simulate and visualize the real world in a very realistic way, but also they allow us to see beyond the reality.

Advanced robot systems require sophisticated simulation tools which can simulate exactly enough the physical world, at sufficient speed and allow users interaction. New challenges in the simulation of robotic systems are multi-body dynamic simulation techniques that compute the robot and the manipulated object motion under the influence of external forces, collision detection, contact determination and realistic visualization of the robot and its environment interaction.

Simulation is the process of a model's design of a real or theoretical system. The execution of a model and its analyzing is necessary before the system's implementation. Actually, in robotics simulation plays a very important role, possibly more important than in many other fields and we like to present in the following some approaching in the robotics from the behavioural simulation point of view.

The principle "learning" is essential in simulation. Using simulation we can learn about some things in a very effective way and while modifying "rules" we can observe the effects of our interaction. As it is in our nature that "seeing is believing", the visualization is the other dominant in simulation.

Simulation is a highly interdisciplinary field since it is widely used, in all fields of research - including the robotics - and at different levels, from academic research to manufactures. Being able to simulate, opens a wide range of options for solving many problems creatively. We can investigate, design, visualize and test an object even if it does not exist. We can see the performances of a system before it is built. It is possible that our solutions may fail or even blow up, but only in simulating on the virtual prototype (Bruzzone et al., 2003).

So, using the simulation tools one can avoid injuries and damages, unnecessary changes in design, after the production has already started a long cycle manufacturing process. Using the virtual prototypes one can avoid even unnecessary paper works. With up-to-date simulation tools one can deal with exact geometry, consider the dynamic characteristics of a system, include the man–machine interfaces, and visualize objects in 3D in detail.

Having all these in mind, for what is possible or not, the boundaries are pushed far away especially with advanced virtual reality tools. Using simulators, the researchers may build experimental environments, according to their own imagination. What they have imagined during the night, it can be transposed into visual imagines the next day. Complexity, specificity can be gradually increased to a level where virtual systems can head to real challenges of the physical world.

Multi-body serial robot systems

In its most general form, a serial robot system consists of a number of rigid links connected with joints. The position and orientation of a robot's end-effector are derived from the joint positions of the robot arm.

For first example, we use a virtual serial robot, which manipulate a container. We have to create all bodies and connect them if desired with proper joints. For example, the 3DOF model as shown below in Fig.4.2.1, integrated with a conveyors system. Here, the multi-body library provides three-dimensional mechanical components to model rigid multi-body robot system.

The robot system is built by connecting blocks representing parts of the robot like link bodies, joints, actuators and gripper.
The simulation methodology uses a number of possible scenarios and involves some sequences of animation, as one can see in the Fig.4.2.1 and Fig. 4.2.2. Fig.4.2.1 shows how a robot hand can grasp a container which is moving. In this example, the synchronization between robot arm and conveyor is studied and all contacts between the fingers of the gripper and the container are analyzed.

Fig.4.2.1 Container grasping scenarios

The dynamic simulation of multi-body systems becomes very important when the robot manipulator must interact with the mobile objects, where the success will depend only on the capabilities of the robots. In our example, the collaborative work between transport system and robot hand is analyzed. This allows us to simulate an entire grasping task, as well as test robot control algorithms.

For the distribution task in the Fig. 4.2.2 is presented the three posture of the scenarios created to depose the container on the corresponding conveyor. The criterion, after which the conveyor is selected, is the colour of both container and conveyor. The three system configurations are presented below for three different colours: blue, green and maroon.

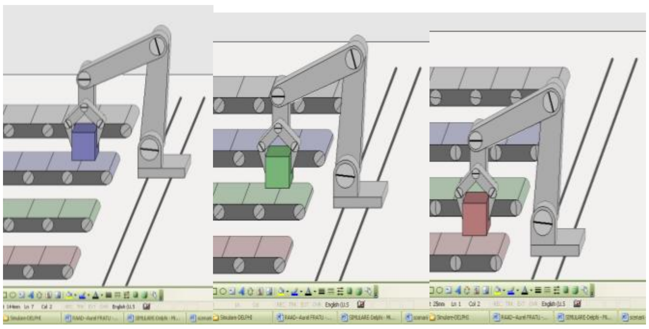

Fig.4.2.2 Scenario for the container placement on the corresponding conveyor according to container color

Multi-body parallel robot systems

Parallel mechanisms generally comprise two platforms that are connected by joints or legs acting in parallel. For the parallel robot applications, as an example we have selected the hexapod platform.

The hexapods are complex mechanisms made up of six (often identical) kinematics chains, of a base (fixed rigid body including six joints or articulations) and of a platform (mobile rigid body containing six other joints).

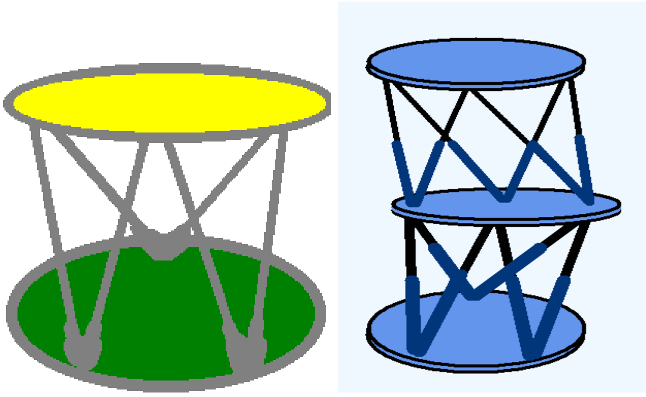

Fig.4.2.3. Triangular moving platform and double hexapod platform

The proposed model with double platform contributes at the increase of the stability of the upper platform by compensating the perturbation with double group of active legs.

Implication of the behavioural simulation in the control design

A very important part of the robotic system is the control system. From the control viewpoint there are different control levels. The lowest level is the close-loop control and the higher is the trajectory planning. The path planning and other more global control tasks are performed at higher levels.

In the process of controller design different steps have to be performed. First of all, the system has to be modeled. In the next step, the control algorithm is developed. The first results are then obtained by the simulation. If the results are satisfactory, then in the final step the control algorithms are tested on a real system. For this, a real-time code should be generated and implemented on the real system.

In an integrated environment one can sustain two modes of operation. In the “pure” simulation mode the virtual robot system is experimented. In the “hardware-in-the-loop” simulation mode, some of the real system (e.g. a real robot with sensors) is included in the simulation loop. Pre-requisites for the “hardware-in-the-loop” simulation are the real-time simulation ability and the necessity of additional hardware components, which are needed for linking together the real system with her simulator. The integration of the two modes of operation is the most important feature of the integrated environment.

To taste the robot control algorithms, in this paper we have developed the Delphi simulators for virtual robot systems. (Fratu and Fratu, 2008). The informatics system was developed specially for testing the different control algorithms for a real grasping process.

Using the virtual system, the controller can be tested without a real robot, e.g. before even the robot has been built. When the real robotic system is tested, the connection between real control unit and the simulator system must be done. If the real robot system imitates the virtual robotic system, will be attended the ideal control.

Conclusion

The purpose of the paper is to emphasize the role of the simulation in different fields of robotics. Also we recommend to researchers to use the general dynamic engines and general visual programming languages for the simulation and visualization of robotic systems.

This paper discusses the using of the visual programming Delphi environment, for facilitating intuitive robot programming. From a robotic point of view, the virtual robotic structures created were simplified and their analyzing allows a better description of the behavior of real robotic structures.

The simulations prepared for this application were also made for the development of the systematic methods for creating the new robotic systems.

The results of this investigation consist in an identification of the properties and constraints of the virtual robot prototypes.The virtual structures created in this paper encompass the behaviour of robotic systems. They may present a behaviour which is qualitatively different from the behaviour that the robots currently have.

In the end, the simulation in robotics has reached a very important role and by using different simulation software, the present and future capabilities of complex robotic systems can be significantly improved.

Another article published in the System concept development with virtual prototyping topic is Simulation of articulated robots for virtual prototyping in dynamic 3D environments.
This paper deals with the simulation of a dynamical system in which the motion of each rigid robot is subject to the influence of virtual forces induced by geometric constraints. These constraints may impose joint connectivity and angle limits for articulated robots, spatial relationships between multiple collaborative robots, or have a robot follow an estimated path to perform certain tasks in a cycle. In this paper the authors give a brief overview of a general simulation framework, describing the primary tasks which a simulator needs to implement. The robot behavioral simulation in the virtual environment enables us to predict the behavior of a given real manipulator into real environment. The robot behavioural simulation enables us to predict the behavior of a given manipulator under given initial conditions, applied torques, and applied loads.

The ability of predicting this behaviours is important for several reasons: for example, in design the designers want to know whether with a given selection of actuators, the manipulator will be able to perform a certain typical task in a given time frame; in creating feedback control schemes, where stability is a major concern, the control engineer cannot risk a valuable piece of equipment by exposing it to untested control strategies. Therefore, a facility capable of predicting the behaviour of a robotic manipulator, or of a system at whole, for that matter, becomes imperative.

In this paper, the authors present a new motion planning algorithm for virtual prototyping. This algorithmic structure is inspired by constrained dynamics in physically-based modelling.

The authors seek to deduce a virtual geometry of the objects, a 3D geometric realization of a collection of rigid bodies is visible in the drawing. The authors transform the motion planning problem into a dynamical system simulation by treating each robot as a rigid body or a collection of rigid bodies moving under the influence of all types of constraint forces in the virtual prototyping environment.

These may include constraints to enforce joint connectivity and angle limits for articulated robots, constraints to enforce a spatial relationship between multiple collaborative robots, constraints to avoid obstacles and self-collision, or constraints to have the robot follow an estimated path to perform certain tasks in a cycle.

Proposed constraint-based planning structure has the following characteristics:

- It can handle both static environments with complete geometric information or dynamic scenes with moving obstacles whose motion is not known a priori.
- It is applicable to both rigid and articulated robots of arbitrarily high degrees of freedom, as well as multiple collaborative agents.
- It allows specification of various types of geometric constraints.
- It runs in real time for modestly complex environments.

The authors demonstrate the effectiveness of this structure for the problem of virtual assembly prototyping with applications in assembly line planning,

Robots' Motion Simulation

The robots' motion should be animated with the highest degree of realism possible using motion capture data or accurate full-body simulation, while the multitudes secondary details to the auxiliary elements (scene, cameras etc.) can be simulated at much lower fidelity.

The classic robot motion problem, also referred to as the Piano Mover's problem, can be stated as the following: given a robot R and a workspace W, find a path from an initial configuration I to a goal configuration G, such that R never collides with any obstacle O_i from a set of obstacles O along the path P, if such a path exists.

The path P is a continuous sequence of positions and orientations of R. Continuous sequences of positions and orientations of R are assimilated with the robot system animation on a virtual scene. Despite the exciting progress in the field, simulating a dynamical system with many degrees of freedom remains a computational challenge. One of the central components of any control or simulation system for articulated bodies is forward dynamics [3].
Forward dynamics computes the acceleration and the resulting motion of each link, based on the given set of external forces and active joint forces. The known algorithms have a linear- time dependence of the number of degrees of freedom. This permits any object in a scene to behave in a physically-plausible way: they accelerate, recognize collisions, and respond to collisions much like one would expect it to respond.

Several techniques have been proposed for accelerating various types of dynamic simulation. Yet, there exists no known general algorithm for automatic simulation of articulated body dynamics.

Plausible motion simulation

Reasearcher Barzel introduced the idea of "plausible" motion, i.e. motion that could happen and look physically plausible to the viewers. For many visual applications or real-time interaction, accurately simulating all the details of the real environment is not necessary.

In fact, it is often sufficient to provide effective motion to make the scene appear more realistic, without committing much computational resources.

In an environment with uncertainty, we generally expect a constrained problem to have multiple solutions. It is difficult to know before what solutions are available. Hence, it is bad to use a solution strategy that seeks a single answer; rather, it prefers a technique that produces many solutions that reflect the range of possible outcomes. While for feature animation a user is expected to choose the one animation they prefer, other applications benefit directly from multiple solutions:

- Computer simulator designers can use different animations each time a simulation is on stage, making it less predictable and potentially more entertaining.
- Training environments can present trainees with multiple physically consistent scenarios that reflect the physics and variety of the real world.

The authors generate multiple animations that satisfy constraints by applying an original algorithm to trial from a randomized model [6]. The algorithm needs the model of the environment, including the sources of uncertainty and the simulator that will generate an animation in the virtual environment. The algorithm described in this paper generates an arbitrarily sequence of animations in which “good” animations are expected to appear.

Simulation loops
The simulation loops fix robots’ components algorithmically, in order to generate physically plausible motion. Note that not all the pieces are put together into a unified system.

Many related works describes a simulation core or a simulation loop to achieve this [5]. The following are the general steps in a simulation core:

1. *Clear the force accumulators*: Each body or its components maintains a total of all forces on it. At the beginning of each step, we clear the forces from the previous step.
2. *Detect collisions*: Loop over all bodies and determine any contacts or collisions in the scene and prepare them to be resolved.
3. *Compute external forces*: Loop over all external forces, including contact or collision resolution forces, and add them to the force accumulators.
4. *Compute constraint forces*: At this point, each body or component knows the total force acting upon it (it’s in force accumulator). To process the constraints, first add any soft constraints to the accumulators. Finally, compute and apply the hard constraint forces.
5. *Compute derivatives*: Gather derivatives as needed to prepare to update the system.
6. *Integrate and update the state of the simulation*: Use a numerical integration method to update the state of the system by some small time increment.

Implementation
Our system was implemented with DELPHI object-oriented programming language. The authors used in-house library ANIMATION-VIEW for collision detection by generating of the distance fields for surface repulsion constraints.
Platform’ toolbox offers the Delphi functions for the implementation of the virtual system prototypes. For discrete set $\{t_k\}$ of instants, Delphi system generates an images sequence of the virtual robot system.

System Demonstration

The authors have tested the proposed motion planning system in the following virtual prototyping application: Assembly Line Planning. An animation generated from this type of scenario is shown in Figure 4.

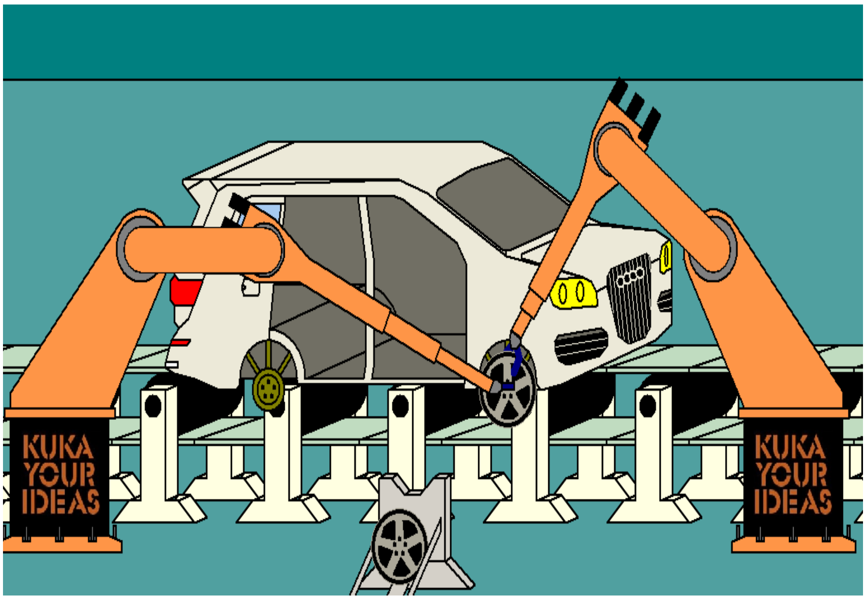

Fig.4.2.4 Assembly Line Planning Scene

The robot arms avoid the moving object to reach a moving part passing on the conveyer belt. In this scenario the aim is to animate all actors to realize the assembly of the automobile. In this example, shown in Figure4, the robot arms from scene must access a part moving and past it on a conveyer belt.

The factory floor contains a transfer structure that is moving over the conveyer belt in the opposite direction to the part's movement. The moving obstruction causes the robots to reactively modify its path to avoid collision.

In this example, constraints can be defined for any aspect of the object's 3D state at any point in time. Initial conditions for the object are specified by constraining its state at the start of the simulation.

Conclusion

The authors have presented a novel framework for motion planning in virtual prototyping applications. Thy have reformulate the motion planning problem into a virtual simulation problem where constraints on the robot's motion guide it from its starting configuration to its target, on the virtual scene. These constraints can't impose penetration constraints among objects, the angle limits and connectivity of articulated robot joints. The avoidance of collision, the following of estimated paths, and many other possible relationships between the cooperative robots and objects on the scene, are feasible in the virtual environment.

The models proposed by authors arise naturally in the virtual environment and provide a means of verifying the plausibility of the motion in the real environment. With further work it should be possible to experimentally obtain more accurate robot dynamically models who require finding good animation.

Acknowledgments
This work was supported in part by LAMIH - Laboratoire d'Automatique, de Mécanique, et d'Informatique industrielles et Humaines - UMR CNRS 8530 - of the Valenciennes University - France. The authors thankfully reply for the support of this institution.

(B-ii) The evolution and development plans for career development

Research Directions

Based on my past experience and on the research and teaching directions detailed below, my short term goal is to build a research team that can accumulate expertise in most areas related to haptic interfaces and robot collision avoidance. All my research and teaching activities are closely intertwined, and this could be a catalyst for the team to become competitive in applying for national and European funding.

The long term goal is to lead a group that could produce high quality research - published in competitive venues and highly cited, but could also produce prototypes and experimental systems to bridge collaboration with the industry. I am presently contributing to one projects collaborating closely with foreign academic partners (University Paris- Sud France), and in parallel, I am applying for funding in partnership with young research teams - PN-II-RU-TE-2014-4.

The planning of future research relies on the direction namely *Interference between virtual and physical world as result of our precedent research domains.* Various domains can benefit from the outcome of the research, e.g. robotic systems, learning technologies, medicine and games. I will refer in the following to two future research topics I would also like to propose to the potential Ph.D. students. They also constitute draft applications for national/European funding grants.

- **Innovative control methods for haptic interfaces**

Principally, Haptic interfaces are based on this interference between these two worlds, *virtual and physical world.* Cooperation between virtual and physical world is a field of Artificial Intelligence concerned with understanding, recognizing, and utilizing human tactile feel in the design of haptic systems. It has become a very hot research topic in human interaction with virtual environment via computer, because it helps improve the quality and effectiveness of human to robots clone communications.

Despite the advances in computing and related technologies, there is no artificial system which can match the human capabilities in dealing with tactile feel. The aim of the future project is to investigate and promote new bio-inspired solutions, in the implementation of artificial systems that can process tactile feel.

The field of Haptic interfaces is planned to be developed, within the Department of Automatics and Technology of Information of Transilvania of Brasov, an experimental platform within the further cooperation with our partners from LAMIH Laboratorz from Valenciennes, France and finding new ones and by involving students in research.
On this platform we propose a description language for specifying motions for haptic robots and for allowing robots to acquire tactile sensation / feeling.

For haptic application, tactile sensation / feeling greatly increases the ability to interact with physical environments, which in turn increases mental abilities. This principle is applied to haptic robots which extend this ability also for virtual environments. However, there are great difficulties to specify haptic feeling in contact with virtual objects and to represent the motion of the virtual devices, which in most cases require 3-D dimensional representations.

Currently most humanoid simulation systems require manual manipulations of body parts or require capturing movements performed by a person. We aim to describe haptic robot motions using our high level motion description language. We will propose new motion primitives and new syntax that allows programmers to easily describe complex robot motions. To make the motion specification possible, we will define a framework that models haptic robots by specifying their capacities and limitations.

Furthermore, we will develop a simulation system that can execute the motion description language and can automatically handle conflicting motion statement and can ensure that the described motions are within the limitations of haptic robots. Our simulation results obtained till now show that the proposed system has great future potentials.

While computer user interface progressed from text only input to graphical user interface; the creation of the Virtual Reality(VR) technology allows the interaction between *virtual and physical world* to become more intuitive.

- **On-line collision avoidance using Programming by Demonstration**

The main objective of this research is to propose and investigate new methods aiming robot collision avoidance. We propose a new strategy to robot collision avoidance using programming imitation paradigm. The concept is based on original idea. His suggestion was that the physical robot can predict the situation (position and orientation) using its virtual prototype in parallel with than physical sensory signals in redundancy manner.

For a system with a demand of reacting as precisely as possible, its past information is not suitable for control planning any more. We should predict the future collision, at the time when the control command arrives at the physical robot and is executed.
The ability of predicting of the robots collision is important in design; the designers want to know whether the robot will be able to perform a typical task in a given time frame into a space with constraints.

The control engineer cannot risk a valuable piece of equipment by exposing it to untested control strategies. Therefore, a facile strategy for contact detection and collision avoidance, capable of predicting the behavior of robotic manipulators, becomes imperative.
When the robots need to interact with their surrounding, it is important that the computer can simulate the interactions of the participants, with the passive or active changing environment in the graphics field, using virtual prototyping.

For effective involved path navigation, a technique is needed which can exploit the reference trajectory structure to search in the local continuum for actions which minimize path deviation and avoid obstacles. A pursuit planner is adequate in order to permit relaxation of a trajectory (for optimization reasons) by searching a small number of degrees of freedom.

The accepted classes of “pursuit” algorithms will be used to generate a path planner in order to accomplish robotic tasks, the actions for the each robotic task are computed for virtual robot and are transferred, with a central coordination to corresponding physical robot which must imitate its virtual homonym.

In this way, distinct other strategies, our system based on patented solution don't require direct physical contact or exact positioning in front of the obstacle. The virtual robot may interact and may interpenetrate each with other or with the obstacles and will advise her physical homonyms before these will crash. Based on save frontier concept, the physical robot will stop in front of obstacle even if her virtual homonym will penetrate through obstacle.

To program the desired motion sequence for the physical robot, one captures the motion reference paths from her virtual robot model and maps these to the joint settings of the physical robot. Motion imitation requires transfer of a dynamical signature of a movement of the virtual robot to the physical robot, i.e. the robots should be able to encode and reproduce a particular path. Furthermore, the virtual robot must cover all possible contexts - including la presence of accidental obstacles - in which the physical robot will need to generate similar motions in unseen context.

The programming method, based on the imitation of virtual prototypes, must be still enhanced to respond at the question: how dynamical virtual system models can be up-dated online to take dynamic events, from the real environment, into account? The author expect fully automated robot programming by imitation, using robust enough system to be applied in practical applications, will not become true before the end of this decade.

The advantages of such programming approach as an alternative to the classical methods (e.g. vision guided trajectory imitation) are on-line adaptation to the motion of the virtual prototype.

Teaching Directions

Successful research activity is not possible without cooperation. Up to now, we have cooperated in the field of *haptic interfaces* with a team from the University of Valenciennes and in the field of *robot collision avoidance* with a team from the University of Reims, France.

The cooperation with University of Valenciennes I has both a teaching side and a research side in the framework of visiting professor / researcher mobility.

Within the teaching side I have sustained eight hour-courses for several years with topics from the robot motion control. This teaching cooperation was in the framework of the Socrates / Erasmus European program and is continuing and planned to be developed with undergraduate students and PHD students.

On the other hand, the research cooperation as visiting researcher in the field of haptic interfaces had as a result the publication of five scientific papers in refereed journals and proceedings of international conferences in the last ten years.
This very good start is supposed to be continued also in a formal framework. The field of tactile sense based on intelligent sensors will be approached in the next future.

The presented research directions and topics in robotic, together with existing equipment and computing facilities constitute a sufficiently large field to accommodate fulltime researchers, doctoral students and diploma students.

Funding can be ensured through national and European grants and contracts with economic partners in the geographical area, where all industry now uses intelligent systems as haptic interfaces systems, manufacturing robotic lines and components and is interested in signal integrity issues.

A normal enhancement of teaching facilities is expected in the future. However, technical knowledge acquired at the first two levels in the field of servo-systems for robotic systems is not enough for performing research activities.
Therefore additional courses must be planned at the doctoral school level. At least two courses have to be introduced in order to ensure a smooth transition from school activity towards research: 1. Intelligent motion control and 2. Visual programming.

After finishing the introductory courses, doctoral students are supposed to gain experience and obtain results by working in the research laboratory under the supervision of and interacting with the doctoral supervisor and with his/her colleagues in a team-work environment.

Practical experience shows that large research structures are not efficient in particular situation. We will rather consider formation of small ad-hoc groups or teams created for solving problems and for application for grants in specific research targets.

From the above presentation, it is clear that the main research topics are planed to belong to the field of Applied Electrical engineering. However our experience in the other two fields where we have worked and obtained results, namely *robot motion control* and *modeling and simulation of the robotic systems* will have an important impact on our future development.
For the future, I see the development of a "*Interference between virtual and physical world*" curricula as a high priority for the doctoral students. The skill set necessary for this discipline spans across electrical engineering (servo-motors, electronics, informatics), and by its nature draws from several areas of technology.

The scientific results achieved during the research will be submitted for publication in ISI journals and conferences. Parts of the research could be also disseminated in my taught courses of *Simulation of the robotic systems* and *Robotic Control Systems*, recursive positioning strategy will proposed to be studied at the doctoral school level.

Based on current trends of increasing importance of intelligent robotic systems, the marketplace is likely to continue to require graduates highly specialized in this area.

(B-iii) Bibliography

1. (1994) **Fratu, A.:** Cercetari privind sistemul de pozitionare cu servomotor sincron si microcalculator pentru roboti industriali**.** Rezumatul tezei de doctorat. Universitatea *Transilvania* din Braşov, 1994.
2. (1994) **Fratu, A**. - Ondulatoare electrice - structuri de comandă pentru roboţi industriali. Lucrările simpozionului de electronică, Piteşti, 11 - 12 noiembrie, 1994, p. 88 - 91.
3. (1994) **Fratu, A**. - Metodă de poziţionare a elementelor structurilor de manipulare. Lucrările simpozionului naţional de Roboţi industriali, vol. 1, 16 - 19 noiembrie, 1994, Timişoara, p.223 - 226.
4. (1998) **Fratu,A.** - Forma generală a comenzii unui sistem robot. Lucrările Simpozionului Naţional de Electronică şi Automatică, Academia Tehnică Militară, noiembrie, 1998, Braşov.
5. (1998) **Fratu, A**. - Controlul forţei în structurile de manipulare. Lucrările celui de al XIV-lea Simpozion Naţional ROBOTICA'98, 5 - 7 noiembrie, 1998, Braşov, p. 275 - 278.
6. (1999) **Fratu,A.** – Modelul dinamic si ecuatie de miscare scrisa in spatiul de eroare. Academia Aviatiei si Apararii Antiaeriene Henri Coanda. Buletinul stiintific, Stiinte tehnice-Automatica, Anul I, No.10(10) dec. 1999, ISSN nr. 1453-0139, p. 87-90.
7. (2002) Rab Laura, **Fratu Mariana** s.a.: Perspectives and trends as regards flow measurements for the automotives sensors, RECENT, Vol. 3 (2002), Nr. 1(6), p. 52-55.
8. (2002) **Fratu, A**.,s.a.: Possible Approaches to Problem of Robot Control in Assembly Operations, RECENT, Vol. 3 (2002), Nr.1(6), p. 33-37.
9. (2003) **Fratu, A**., s.a.: Generarea miscarii unui brat robot, Revista RECENT, **ISSN** 1582 – 0246, Vol.4 (2003), Nr.1(7), p. 21-25.
10. (2004) **Fratu, A**., s.a.: Adaptive control for robots, Revista RECENT, **ISSN** 1582 – 0246, Vol.5(2004). Nr.3(12), p. 34-36.
11. (2005) **Fratu, A.** et al.: *The problem of operational space control for redundant robots*, Revue RECENT, **ISSN** 1582 – 0246, Vol.6 (2005). Nr.3 (15), p. 52-55.
12. (2005) **Fratu, A**., s.a.: Dynamics model of robots, Revista RECENT, ISSN 1582 – 0246, Vol.6(2005). Nr.2 (14), p. 31-34.
13. (2005) **Fratu, A**., s.a.: Application's development with SCADA system, Revista RECENT, **ISSN** 1582 – 0246, Vol.6 (2005). Nr.2 (14), p. 35 -40.
14. (2005) **Fratu, A**., s.a.: The problem of operational space control for redundant robots, Revista RECENT, **ISSN** 1582 – 0246, Vol.6 (2005). Nr.3 (15), p. 52-55.
15. (2005) **Fratu, A**.: Progress and Trends in the University-Industry Co-operation for foreign Students received at the Automatics Department. Bulletin of the University of Brasov, **ISSN** 1223-9631, Vol.12(47)-New Series, 2005, p.557-558.
16. (2005) **Fratu, A**. et al. : *Artificial Inteligence and Inteligences Robots' apereance*, (Inteligenta Artificiala si aparitia Robotilor Inteligenti), Vol. " Hermeneutica Historiae

et Philosophiae Technicae ", TRANSILVANIA University Pubishing, ISBN 973-635-744-9, Liber III, Anno 2005, p,150-154.

17. (2006) **Fratu, A** et al.: *Feedback linearization in robot's technique,* Revista RECENT, **ISSN** 1582 – 0246, Vol.7 (2006). Nr.1 (16), p. 24-27.
18. (2006) **Fratu, A**. et al.: *Repetitive Control Motion in Robots Manipulators' Technique,* Revue RECENT, **ISSN** 1582 – 0246, Vol.7, Nr.2 (17), p. 37-41.
19. (2007) Fratu,M**.** and **Fratu, A**. : *Anomaly diagnosis system based on qualitative models*. Transylvania University Publishing, ISSN 1843-6617, CIB-2007, p.551-54.
20. (2007) Fratu,M. and **Fratu, A**. : *Anomaly diagnosis method based on neural networks*. Transylvania University of Brasov Publishing, **ISSN** 1843-6617, CIB-2007, P.55-58.
21. (2007) **Fratu, A**., et al.: *Avoidance Obstacle Technique Using Redundant Robots,* Revue RECENT, **ISSN** 1582 – 0246, Vol.8, Nr.1 (19), p. 22-26.
22. (2007) **Fratu, A.,** Vermeiren, L. : *Adaptive control architecture for redundant robots*, Revue RECENT, ISSN 1582 – 0246, Vol.8, Nr.2 (20), p. 97-100.
23. (2007) **Fratu,A** et al.: *The science and the art-together-in creating automatically virtual images.* Transylvania University Publishing, **ISSN 1843-4959,** Vol. ARTA SI STIINTA, 2007, p.40-47.
24. (2008) **Fratu A.,** Renaux, D., Fratu Mariana: Simulation and Control of the Parallel Robots' Dynamics. Bulletin of the TRANSILVANIA University of Brasov, ISSN 2065-2119, VOL.1 (50), Series I, p.343-350.
25. (2008) Fratu, M. and **Fratu, A.** : Modelarea defectelor in sistemele de transport combustibili energetici. Transylvania University Publishing, ISSN 1843-6617, CIB-2008, p.61-64.
26. (2009) Fratu M. and **Fratu, A**.: Simulation of the Gas Transportation through Pipes Inspected by Intelligent Robots. BULETINUL Universităţii Petrol – Gaze din Ploieşti, Vol. LXI, No. 2, p. 39 – 44, Seria Tehnică.
27. (2009) **Fratu A**., Brethe. J.F**.,** Fratu M.: Industrial robots in a flexible assembly system, Revista RECENT, **ISSN** 1582 – 0246, Vol.10, Nr.1 (25), p. 28-31.
28. (2009) Fratu M. et **Fratu, A.:** Simulation of the Gas Transportation through Pipes. Revista RECENT, **ISSN** 1582 – 0246, Vol.10, Nr.2 (26), p. 105-108.
29. (2010) **Fratu, A**., Brethe, J.F., Fratu, M.: *Redundant inverse kinematics system for obstacles avoidance*, Revue RECENT, ISSN 1582 – 0246, Vol. 11, no. 1(28), pp. 29-32.
30. (2011) **Fratu, A.** and Fratu, M. : *Analytical Model of the Cutting Process with Scissors-Robot for Haptic Simulation*. Bulletin of the *Transilvania* University of Braşov, ISSN 2065 - 2119, Series I: Engineering Sciences, Vol. 4 (53) No. 1 – 2011, pp. 113-120.
31. (2011) **Fratu, A.**: *Collision Prevention Method for a Dynamic Group of Cooperative Robots Who Communicate Wirelessly***,** Revue RECENT, ISSN 1582 – 0246, Vol. 12, No. 2 (32), pp.131-134.
32. (2011) **Fratu, A.**, Fratu, M**.**: *Collision Prevention Protocol for a Dynamic Group of Asynchronous Cooperative Robots***,** Revue RECENT, ISSN 1582 – 0246, Vol. 12 , No. 2 (32), pp.135-138.
33. (2012) **Fratu, A**., Fratu, M.: *Simulation of Articulated Robots for Virtual Prototyping in Dynamic 3D Environments,* Bulletin of the Transilvania University of Brasov, Series I: Engineering Sciences, ISSN 2065-2119, Vol. 5 (54) No. 1 – 2012, pp.71-76, indexed in EBSCO
34. (2012) **Fratu, A**., Fratu, M.: *Equation of Motion of the Nonlinear Robotic Systems with Constraints*. Bulletin of the Transilvania University of Brasov, Series I: Engineering Sciences, ISSN 2065-2119, Vol. 5 (55) No. 2 – 2012, Accepted for publication, indexed in EBSCO

35. (2012) **Fratu, A.** and Fratu, M.: *Robot Control for Avoid Self-Collisions*, Revue RECENT, ISSN 1582 – 0246, Vol. 13 (2012), No. 1 (34), March 2012, pp.47-50, Indexed by ULRICHSWEB and Copernicus Journal Master List
36. (2012) Fratu M., **Fratu, A.**: *Collision Avoidance Control Using Null Space Criteria*, Revue RECENT, ISSN 1582 – 0246, Vol. 13 (2012), No. 1 (34), March 2012, pp. 51-54
37. (2012) **Fratu, A**., Fratu, M.: *Tracking Control of Robotic Multi-Body Systems*. RECENT Journal - ISSN 1582-0246, Vol. 13 (2012), No. 2 (35), July 2012, pp.173-176
38. (2012) **Fratu, A.**, Fratu, M.: *Space Management in Smart Room*, RECENT Journal - ISSN 1582-0246, Vol. 13 (2012), No. 2 (35), July 2012, pp.177-180
39. (2012) **Fratu, A**.: *Simulation of the Robot Systems for Virtual Prototyping in Assembly Operations*. RECENT Journal - ISSN 1582-0246, Vol. 13 (2012), No. 3 (36), November 2012, pp.279-282
40. (2012) Fratu, M., **Fratu, A**.: *Intelligent Design Solutions for Intelligent Buildings*. RECENT Journal - ISSN 1582-0246, Vol. 13 (2012), No. 3 (36), November 2012, pp.283-286
41. (2013) **Fratu, A. :** *Robot collision avoidance using the programming through imitation*, Revue RECENT, ISSN 1582 – 0246, Vol. 14 (2013), No. 2 (38), July 2013, pp.87-92.
42. (2013) **Fratu, A**., Fratu, M.: *Physical Robots Programming by Imitation Using Virtual Robot Prototypes*, Bulletin of the *Transilvania* University of Braşov, Series I: Engineering Sciences • Vol. 6 (55) No. 2 – 2013, pp. 51-58.
43. (2013) **Fratu, A.**, Dambrine, M., Vermeiren, L., Dequidt, A.: *An analysis of the reciprocal collision avoidance of cooperative robots*, In: Journal Automation, Control and Intelligent Systems, 2013; 1(3): 75-84, Published online July 30, 2013, Vol. 1, No. 3, 2013, pp. 75-84. doi: 10.11648/j.acis.20130103.17
44. (2014) Fratu, M., **Fratu, A**.: *Visual programming technique and virtual prototyping technology used in civil engineering*, RECENT Journal - ISSN 1582-0246, Vol. 15 No. 2 (42), July 2014, pp.97-101.
45. (1996) **Fratu,A.** , ş.a. - Cuplaj cu sens unic pentru demaroare de autovehicule. Lucrările celei de-a 8-a Conferinţe cu participare internaţională CONAT'96, 13 - 15 noiembrie, 1996, Braşov, p. 303 - 308.
46. (1997) Helerea, E. , **Fratu, A**. , ş.a. - Baterii de acumulatoare auto - tradiţie şi inovaţie. Lucrările celei de-a 9-a Conferinţe cu participare internaţională CAR'97, noiembrie, 1997, Piteşti. P 13-18.
47. (1998) **Fratu, A**. - The phase - modulation in control of the electrical corrugater. Proceedings of the 6-th International Conference OPTIM'98, 14 - 15 may, 1998, Braşov, p. 439 - 442.
48. (2000) Ilie, A., Laura Rab, L., **Fratu, A.,** Ilie, R. : "Electronic simulator for a frequencially generated reference angular error". The 7-th International Conference OPTIM'2000, may 11-12, Braşov, Romania.
49. (2002) Rab Laura, **Fratu A**. s.a. : Temperature Measurement Methods Based on Resistor Thermal Noise Voltage, Proceedings of the 8th International Conference on Optimization of Electrical and Electronic Equipments, Brasov, May 16-17, 2002, p.717-720.
50. (2002) **Fratu, A**., s.a : Elements de definition du moteur pour servosystemes,The VI-th International Conference on Precision Mechanic and Mechatronics, COMEFIN-6, Brasov, 2002, October 10-12, p.79-84.
51. (2006) **Fratu, A**.: *Progress and Trends in the University-Industry Co-operation for foreign Students- case study*. Proceedings ''European Models of Synergy between

Teaching and Research in Higher Education'', Technical University of Tallinn, 4-6 may 2006, p.81-84.

52. (2007) Fratu M. et **Fratu, A**.: *Increasing of the trusting level in the measuring systems using the redundant information*. The International Conference on Library Science and Science of Information, Brasov 2007, ISBN 978-973-635-973-6, p.169-172.
53. (2007) **Fratu, A**.et Fratu M.**:** *Automatically building of the virtual images* . The International Conference on Library Science and Science of Information, Brasov 2007, ISBN 978-973-635-973-6, p.163-168.
54. (2008) **Fratu,A.** and Fratu, M. : *Some considerations regarding the simulation of the robots movement in the environment with constraints.* The International Conference on Computing and Solutions in Manufacturing Engineering (CoSME), Brasov, Romania. Academic Journal of Manufacturing Engineering, ISSN 1583-7904, ISSUE 1/2008, p.148-153. www.eng.upt.ro/auif/
55. (2009) **Fratu, A**., Fratu M., Brethe, J.F., Cachard, O. : Modélisation et animation de systèmes robotises dans l'environnement Delphi, Colloque francophone ''Evaluation des Performances et Maîtrise des Risques Technologiques pour les systèmes industriels et énergétiques", 28 – 29 mai 2009, p. 105-112, Université du Havre – France.
56. (2009) DEQUIDT A., VERMEIREN L., **FRATU A**.: Design of flexible drive systems for feed forward control: a mechatronic approach. *18th International Workshop on Robotics in Alpe-Adria-Danube Region*, ISBN 978 -606-521-315-9, Brasov, Romania, mai 25-27, 2009.
57. (2009) **FRATU A**., SISAK F., VERMEIREN L., DEQUIDT A.: Modelling and robot simulation in Delphi environment. *18th International Workshop on Robotics in Alpe-Adria-Danube Region*, ISBN 978 -606-521-315-9, Brasov, Romania, mai 25-27, 2009.
58. (2009) **FRATU, A.,** DEQUIDT, A., VERMEIREN, L.: Haptic Devices for Desktop Virtual Prototyping Applications . The 3rd International Conference "Computational Mechanics and Virtual Engineering" - COMEC 2009, p. 248-253.
59. (2009) **FRATU, A.,** VERMEIREN, L., BRETHE, J-F**. :** Study of the assembly motion for a virtual assembly system – International Conference on Economic Engineering and Manufacturing Systems (ICEEMS) , 26-27 November 2009, Brasov, Romania
60. (2010) **Fratu, A.,** Vermeiren, L., Dequidt, A.,: *Using the Redundant Inverse Kinematics System for Collision Avoidance.* The 3rd International Symposium on Electrical and Electronics Engineering – ISEEE-2010, 16-18 sept. Galati, Romania, Proceedings ISBN 978-1-4244-8407-2, pp.88-93.
61. (2011) **Fratu, A**., Dequit, A., Vermeiren, L.: *Virtual Scissors for Haptic Portable Torque-Feedback Device,* Proceedings of the Remote Engineering and Virtual instrumentation Conference - REV 2011, pp.10-15.
62. (2011) Dang, Q. V., Dequidt, A., Vermeiren, L., **Fratu, A**., Dambrine, M. : *Stability analysis of haptic interface: effects of dynamic parameters*. Proceedings of the Remote Engineering and Virtual instrumentation Conference - REV 2011, pp. 137-143.
63. (2011) **Fratu, A.:** *A Multi-Robot System for Assembly Tasks in Automotive Industry.* Proceedings of the 4th International Conference ″Computational Mechanics and Virtual Engineering″ COMEC 2011, pp.229-234.
64. (2011) **Fratu, A.:** *Simulation of the Maintenance Analysis for Virtual Steering Training in Automotive Industry.* Proceedings of The 4th International Conference″Computational Mechanics and Virtual Engineering ″COMEC 2011", pp. 235-238.
65. (2011) **Fratu, A.:** *Robotic Manufacturing Systems – Modelling and Simulation,* Proceedings of the International Conference on Economic Engineering and

Manufacturing Systems (ICEEMS), 26-27 Nov. 2011, RECENT, ISSN 1582 – 0246, pp.287-290.

66. (2011) **Fratu, A.:** *Actuators as Mechatronics Objects for Motion Software Control*. Proceedings of the International Conference on Economic Engineering and Manufacturing Systems (ICEEMS), 26-27 Nov. 2011, RECENT, ISSN 1582 – 0246, pp.291-294.
67. (2011) Fratu M. and **Fratu, A**.: *A Multi-Robot System for Assembly Tasks in Wood Products Manufacturing Industry*. Proceedings of the International Conference on Economic Engineering and Manufacturing Systems (ICEEMS), 26-27 Nov. 2011, RECENT, ISSN 1582 – 0246, pp.295-298.
68. (2012) **Fratu, A**., Ilea, D.: *An Analysis of the Reciprocal Robots Collision Avoidance* Proceeding of the 13th International Conference on Optimization of Electrical and Electronic Equipment (OPTIM), Digital Object Identifier: 10.1109/OPTIM.2012.6231995, IEEE CONF. PUBLICATIONS, pp.1511-1517.
69. (2013) **Fratu, A**,, Dambrine, M,: *An analysis of the robot collision avoidance using the programming through imitation*, The International Conference on Manufacturing Systems - ICMaS 2013, 14-15 November, 2013, Bucharest, Romania. In: Proceedings in Manufacturing Systems, ISSN 2067-9238, Volume 8, Issue 2, 2013, pp.111-116.
70. (2014) **Fratu, A**., Fratu, M.: *Imitation-Based Motion Programming for Robotic Manipulators*, Proceedings of 14th International Conference on Optimization of Electrical and Electronic Equipment OPTIM 2014. May 22-24, 2014, Brasov, Romania, ISBN 978-1-4799-5183-3/14/$31.00 ' 2014 IEEE, pp. 770-775
71. (2014) **Fratu, A**., Riera, B., and Vrabie, V.: *Predictive strategy for robot behavioral control,* The International Conference on Manufacturing Systems - ICMaS 2014, 14-15 November, 2013, Bucharest, Romania - accepted to be published.
72. (2014) J.P. Becar, M. Fratu, **A. Fratu**, J.C. Canonne EXAMPLE BASED LEARNING FOR VIRTUAL PROTOTYPING ENGINEERING, EDULEARN14 Proceedings OF THE 6th International Conference on Education and New Learning Technologies, _ **ISBN:** 978-84-617-0557-3, **ISSN:** 2340-1117, 7-9 July, 2014, Barcelona, Spain, pp. 6027-6034
73. (2015) J.P. Becar, J.C. Canonne, E. Cartignies, M. **Fratu, A**. Fratu, M. Iwanowski, W. Czajewski THREE EXPERIMENTS OF LEARNING BY EXAMPLES, INTED2015 Proceedings 9th International Technology, Education and Development Conference, **ISBN:** 978-84-606-5763-7, **ISSN:** 2340-1079, 2-4 March, 2015, Madrid, Spain, pp. 1293-1302
74. Barzel, R. et al.: *Plausible motion simulation for computer graphics animation*. In: Proceedings of the Eurographics Workshop on Computer Animation and Simulation, ISBN:3-211-82885-0, New York, December 1996, Springer-Verlag Publisher, p.183–197.
75. Bruzzone, L.E., Molfino, R.M., Zoppi, M. and Zurlo, G: *The PRIDE prototype: control layout of a parallel robot for assembly tasks*, in Proceeding. of the International Conference Modelling, Identification and Control MIC2003, 2003, pp. 606-611.
76. C. Fulgenzi, A. Spalanzani, C. Laugier, “Dynamic obstacle avoidance in uncertain environment combining PVOs and occupancy grid”. In Proc. IEEE Int. Conf. on Robotics and Automation, pp.1610–1616, 2007.
77. C. Spence and A. Gallace: Recent developments in the study of tactile attention. Canadian Journal of Experimental Psychology, 61(3), pp. 196–207, 2007.
78. C. Spence and C. Ho: Multisensory warning signals for event perception and safe driving. Theoretical Issues in Ergonomics Science, 9(6): pp. 523–554, 2008.

79. Calinon, S., Guenter, F. and Billard, A.: On learning, representing and generalizing a task in a humanoid robot, In *IEEE Trans. on Systems, Man and Cybernetics, Part B*, vol. 37, no. 2, pp., 286-298, 2007
80. Cetinkunt S. 1991. Optimal design issues in high-speed high-precision motion servo systems, *Mechatronics*, Vol. 1 (2), pp. 187-201.
81. Chiacchio, P., Chiaverini, S., Sciavicco, L., Siciliano, B. *Global task space manipulability ellipsoids for multiple arm systems.* In: IEEETransactions on Robotics and Automation, **7** (1991), p.678-685.
82. Choi, S.H, Chan, A.M.M.: *A virtual prototyping system for rapid product development*. Computer Aided Design Journal, Vol. 36, Issue 5, April 2004, p. 401-412, ISBN 1-58113-506-8, Saarbrücken, Germany
83. Corke, P.I.: *A robotics toolbox for MATLAB*, IEEE Robot Automation Magazine 3(1), p. 24–32, 1996.
84. D. Hennes, D. Claes, W. Meeussen, K. Tuyls, "Multi-robot collision avoidance with localization uncertainty", In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), June, 4– 8, 2012, Valencia, Spain.
85. D. M. Stipanovic, P. F. Hokayem, M. W. Spong, D. D. Siljak, "Cooperative Avoidance Control for Multiagent Systems". In: ASME Journal of Dynamic Systems Measurement and Control, Vol.129, pp. 699–707, 2007.
86. D. Silver. Cooperative path finding. In The 1st Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE'05), pages 23–28, 2005.
87. Dequidt A., Valdès E. and Castelain J.M. 1999. Nombres adimensionnels pour la conception intégrée des mécanismes commandés, *14ème Congrès Français de Mécanique*, CD Rom.
88. Ellis G. and Lorenz R. D. 2000. Resonant load control methods for industrial servo drives, *IEEE Industry Applications Society*, Annual Meeting, Rome, Italy, October 8-12.
89. F. Gosselin and A. Riwan : Design of a wearable haptic interface for precise finger interactions in large virtual environments. *Proc. World Haptics Conference*, Pisa, Italy, 18-20 March 2005, ISBN: 0-7695-2310-2.
90. F. Nagata, I. Okabayashi, M. Matsuno, et al., "Fine gain tuning for model-based robotic servo controllers using genetic algorithms," *Procs. of the 13th Int. Conf. on Advanced Robotics*, pp. 987–992, 2007.
91. F. Nagata, K. Kuribayashi, K. Kiguchi and K. Watanabe, "Simulation of fine gain tuning using genetic algorithms for model based robotic servo controllers," *Procs. of the IEEE Int. Symp. on Computational Intelligence in Robotics and Automation*, pp.196–201, 2007.
92. F. Nagata, Y. Kusumoto, Y. Fujimoto and K. Watanabe, "Robotic sanding system for new designed furniture with free formed surface," *Robotics and Computer-Integrated Manufacturing*, vol. 23, no. 4, pp. 371–379, 2007.
93. Feliu V., Pereira E., Díaz I. And Roncero P. 2006. Feedforward control of multimode single-link flexible manipulators based on an optimal mechanical design, *Robotics and Autonomous Systems*, Vol. 54 (8), pp. 651-666.
94. Fernandez-Madrigal, J-A. et al.: *A software engineering approach for the development of heterogeneous robotic applications*. Proceeding of Robotics and Computer-Integrated Manufacturing,no.24 (2008) p.150–166.
95. Fernando, T., Wimalaratne, P., and Tan, K., 1999, "Constraint-Based Virtual Environment for Supporting Assembly and Maintainability Tasks," Proceedings of the 1999 ASME Design Technical Conference and Computers in Engineering Conference, DETC99/CIE-9043, Las Vegas, Nevada, September 11-15.

96. Fischer, J., Breithaupt, R., Bode, B.: *Adaptive and economic data representation in control architectures of autonomous realworld robots*. Artificial Life and Robotics, Vol. 6, No. 4, December 2002, p. 200-204, ISSN 1433-5298, Springer Japan Publisher
97. Fleury, G., Lacomme, Ph., Tanguy, A.: *Simulation a eveniments discrets.* Publisher: EYROLLES, 2006.
98. Fratu A., Renaux, D., Fratu Mariana: Simulation and Control of the Parallel Robots' Dynamics. Bulletin of the TRANSILVANIA University of Brasov, ISSN 2065-2119, VOL.1 (50), Series I, p.343-350 Caccavale, F., Chiacchio, P., Chiaverini, S. : *Task-space regulation of cooperative manipulators*. In: Automatica 36 (2000), p.879-887.
99. Fratu A., Renaux, D., Fratu Mariana: Simulation and Control of the Parallel Robots' Dynamics. Bulletin of the TRANSILVANIA University of Brasov, ISSN 2065-2119, VOL.1 (50), Series I, p.343-350 Dun, X. et al.: *Design of a flexible robot assembly demo system.* Proceeding of the 4th World Congress on Intelligent Control and Automation, 2002, vol. 2, p. 1343-1346, ISBN 0-7803-7268-9
100. Fratu, A. and Fratu M.: *Visual Programming in the Delphi Environment with Applications in Robotics.* Transilvania University Press, Brasov, ISBN 978-973-598-315-4, 2008 si 2011.
101. Fratu, A. et al: *Industrial Robots in a Flexible Assembly System*. RECENT Journal, Vol.10, No.1 (25), p. 28-31,ISSN 1582-0246, Brasov, Romania
102. Fratu, A., Fratu M.: *Programarea vizuală în mediul Delphi - cu aplicaţii în Robotică.* Editura Universitatii Transilvania Brasov, 2008.
103. *Fratu, A., Fratu, M.: Visual programming in Delphi medium with Applications in Robotics (Programarea vizuală în mediul Delphi cu AplicaŃii în Robotică)* "Transilvania" University Press, ISBN 978-973-598-315- 4, Brasov, Romania, 2008 (in Romanian)
104. Fratu, A., Fratu, Mariana: *The problem of operational space control for redundant robots*. RECENT, Vol. 6, No. 3(15), December 2005, p. 52-55, ISSN 1582 – 0246, Brasov, Romania
105. G. Schreiber, Ch. Ott, G. Hirzinger: *Interactive Redundant Robotics: Control of the Inverted Pendulum with Nullspace Motion*, Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems Maui, Hawaii, USA, Oct. 29 – Nov. 03, 2001
106. Galoppo, N.: Animation, Simulation, and Control of Soft Characters using Layered Representations and Sampled Physics-based Methods, Dissertation submitted to the faculty of the University of North Carolina Chapel Hill, 2008.
107. Gillespie,R.B. et al: *Haptic Interface for Hands-On Instruction in System Dynamics and Embedded Control,* In 11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2003, pp. 410-415.
108. Good M.C., Sweet L.M. and Strobel K.L. 1984. Dynamic models for control system design of integrated robot and drive systems, *ASME J. of Dyn. Syst., Meas., and Cont.,* Vol. 107, pp. 53-59.
109. Gowda, S., Jayaram, S., and Jayaram, U., 1999, "Architectures for Internet-based Collaborative Virtual Prototyping," Proceedings of the 1999 ASME Design Technical Conference and Computers in Engineering Conference, DETC99/CIE-9040, Las Vegas, Nevada, September 11-15.
110. Gregory, A. et al.: *A Framework for Fast and Accurate Collision Detection for HapticInteraction*. In Proceedings of the IEEE Virtual Reality (VR'99), Pg. 38-45, 1999.
111. Grimes, D., Chalodhorn, R. and Rao, R.: Dynamic imitation in a humanoid robot through non-parametric probabilistic inference, In *Proc. Robotics: Science and Systems (RSS),* Philadelphia, Pennsylvania, USA, August 2006.

112. Grimes, D., Chalodhorn, R. and Rao, R.: Dynamic imitation in a humanoid robot through non-parametric probabilistic inference, In *Proc. Robotics: Science and Systems (RSS)*, Philadelphia, Pennsylvania, USA, August 2006.
113. Hashimoto, H.: *Intelligent interactive spaces - integration of IT and robotics*, In: Proceedings of 2005 IEEE Workshop on Advanced Robotics and its Social Impacts, p. 85–90
114. Heejin, L., Hyunseok, S., Euntai, K., Seungwoo K., Mignon, P.: *Variable structure control of manipulator using linear timevarying sliding surfaces.* Proceeding of International Conferences on Intelligent Robots and Systems, Vol. 2, p. 806-811, ISBN 0-7803-4465-0, October 1998, Victoria, BC, Canada
115. Helmer, P.: *3D force-feedback wrist,* MS thesis, Swiss Federal Institute of Technology, Lausanne, Switzerland, 2000
116. Hoffmann, H., Pastor, P., Park, D.-H., and Schaal, S.: Biologically-inspired dynamical systems for movement generation- Automatic real-time goal adaptation and obstacle avoidance. In *International Conference on Robotics and Automation*, pp. 2587–2592, 2009, Piscataway, NJ: IEEE.
117. Hosoda, K., Igarashi, K., Asada, M.: *Adaptive hybrid control for visual servoing and force servoing in an unknown environment*. IEEE Robotics and Automation Magazine, Vol. 5, No. 4, p. 39–43, December 1998, ISSN 1070.99321, University of South Florida, USA
118. Hu, X. and Zeigler, B.P.: Model Continuity to Support Software Development for Distributed Robotic Systems: a *Team Formation Example.* Journal of Intelligent & Robotic Systems, Theory & Application, Vol. 39, Issue 1, January 2004, p. 71-87, ISSN 0921-0296, Springer Netherlands
119. I. Díaz and J.J. Gil, "Influence of Internal Vibration Modes on the Stability of Haptic Rendering", Robotics and Automation, 2008.ICRA 2008. IEEE International Conference; "Influence of Vibration Modes and Human Operator on the Stability of Haptic Rendering", IEEE TRANSACTIONS ON ROBOTICS, VOL. 26, NO. 1, FEBRUARY 2010.
120. Ijspeert, J., Nakanishi, J. and Schaal, S.: Movement imitation with nonlinear dynamical systems in humanoid robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp.1398-1403, 2002.
121. J. E. Colgate and G. Schenkel, "Passivity of a class of sampled-data systems: Application to haptic interfaces," Journal on Robotic System, vol.14, no.1, pp. 37–47, 1997.
122. J. J. Gil, E. Sanchez, T. Hulin, C. Preusche, and G. Hirzinger, "Stability boundary for haptic rendering: Influence of damping and delay," in IEEE Int. Conf. Robot. Autom., Roma, Italy, April 10-14 2007, pp. 124–129.
123. J. Pettre, J. Ondrej, A.-H. Olivier, A. Cretual, and S. Donikian. Experiment based modeling, simulation and validation of interactions between virtual walkers. In Symposium on Computer Animation. ACM, 2009.
124. J. Snape, J. van den Berg, S. J. Guy, and D. Manocha, "Independent navigation of multiple mobile robots with hybrid reciprocal velocity obstacles", IEEE/RSJ Int. Conf. Intelligent Robots and Systems, St. Louis, Mo., 2009;
125. J. Snape, J.van den Berg, S.J. Guy, D. Manocha, J. Snape, S. J. Guy, J. van den Berg, S. Curtis, S. Patil, M. Lin, and D. Manocha, "Independent navigation of multiple robots and virtual agents". In Proc. of the 9th Int. Conf. on Autonomous Agents and Multi agents Systems (AAMAS 2010), Toronto, Canada, May 2010.
126. J. van den Berg, D. Wilkie, S. Guy, M. Niethammer, D. Manocha, "LQG-Obstacles: Feedback control with collision avoidance for mobile robots with motion and sensing

un-certainty.” IEEE Int. Conf. on Robotics and Automation, River Centre, Saint Paul, Minnesota, USA, May 14-18, 2012, pp. 346- 353.

127. J. van den Berg, S. Guy, M. Lin, D. Manocha, “ Reciprocal n-body collision avoidance”. In: Proc. Int. Symposium on Robotics Research, 2009
128. J. van den Berg, S. J. Guy, M. Lin, and D. Manocha, C. Pradalier, R. Siegwart, and G. Hirzinger, ”Reciprocal n-body collision avoidance”, Robotics Research, The 14th International Symposium ISRR, Springer Tracts in Advanced Robotics, vol. 70, Springer-Verlag, May 2011, pp. 3-19.
129. J.E. Colgate and J.M. Brown, “Factors Affecting the Z-Width of a Haptic Display”, Proceedings of the IEEE 1994 International Conference on Robotics & Automation, pp. 3205-10, San Diego, CA, May 1994
130. J.J Gil, M.J. Puerto, I. Díaz and E.Sánchez, “On the Z-Width Limitation due to the Vibration Modes of Haptic Interfaces”, G. Hirzinger, A. Albu-Schaffer, M. Hahnle, I. Schaefer, and N. Sporer: On a new generation of torque controlled light-weight robots. *IEEE International Conference of Robotics and Automation*, pp. 3356-3363, 2001.
131. J.J. Gil, A. Avello, Á. Rubio, and J. Flórez, “Stability analysis of a 1 DOF haptic interface using the Routh–Hurwitz criterion”, IEEE transactions on Control Systems Technology, vol. 12, no. 4, july 2004, pp. 538-588.
132. K. Gold. An information pipeline model of human-robot interaction. In Proceedings of the 4th ACM/IEEE international conference on Human robot interaction, pp. 85- 92, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-404-1. doi:http://doi.acm.org/10.1145/1514095.
133. K. O. Arras, J. Persson, N. Tomatis, R. Siegwart, “ Real-Time Obstacle Avoidance for Polygonal Robots With a Reduced Dynamic Window” in Proc. IEEE Int. Conf. on Robotics and Automation, Washington DC, May 2002, pp.3050-3055.
134. Khalil W. and Liegeois A. 1984. Génération des mouvements optimaux en robotique, *RAIRO Automatique/Systèmes analysis and Control*, Vol. 18(1) pp. 24-39.
135. Khalil, W., Ibrahim, O.: *General Solution for the Dynamic Modeling of Parallel Robots*, In Journal of Intelligent and Robotic Systems **49 (**2007**)** Nr.1, p.19-37.
136. Kim J, Kim K, Choi K, Lee J.Y.: *Solving 3D geometric constraints for assembly modeling*. International Journal Advanced Manufacturer Technologies, Vol. 16, No 11, September 2000, p. 843-849, ISSN 0268-3768, Springer Netherlands.
137. Kober, J., Peters, J. : Learning motor primitives for robotics. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2112–2118, 2009, Piscataway, NJ: IEEE.
138. Kulvicius, T., Ning, K., Tamosiunaite, M., and Worgotter, F. : Joining movement sequences- Modified dynamic movement primitives for robotics applications exemplified on handwriting, In *IEEE Transactions on Robotics*, *28*(1), pp. 145–157, 2012.
139. L.D., Magillo, P., Puppo, E., Sobrero, D.: *A multi*resolution topological representation for non-manifold meshes. Computer Aided Design Journal, Vol. 36, Issue 2,June, 2004, p. 141–59, ISBN 1-58113-506-8,Saarbrücken, Germany
140. Lambrechts P., Boerlage M. and Steinbuch M. 2005. Trajectory planning and feedforward design for electromechanical motion systems, *Control Engineering Practice*, Vol. 13 (2), pp. 145-157.
141. Lawrence, D. A. et al: *Human Perceptual Thresholds of Friction in Haptic Interfaces*, In Proceedings of the ASME Dynamic Systems and Control Division, DSC-Vol.64, Pp. 287-294, ASME 1998

142. Li, Y.T., Hu, S.M., Sun, J.G.: A constructive approach to solving 3-D geometric constraint systems using dependence analysis. Computer Aided Design Journal, p. Vol. 34, Issue 2, February 2002, 97-108, ISSN 0010-4485,Elsevier
143. Lippiello, V. et al.: *Robot Force/Position Control with Force and Visual Feedback*. Proceedings of the European Control Conference 2007, ISBN: 978-960-89028-5-5, Kos, Greece, July 2-5, 2007
144. Luo_and S. Ahmad: *Mathematical Analysis of the Joint Motion of Redundant Robots Under Pseudo-Inverse Control*. ECE Technical Reports Purdue Libraries TR-EE 92-12, March 1992, Purdue University West Lafayette, IN 47907, USA
145. M. Turpin, N. Michael, V. Kumar," Trajectory planning and assignment in multi robot systems". Proc. Workshop on Algorithmic Foundations of Robotics, 2012.
146. McNeely, W. A. et al: *Six degree-of-freedom haptic rendering using voxel sampling*, In Proceedings of the 26th annual conference on Computer graphics and interactive techniques, p.401-408, July 1999.
147. Merlet, J.-P.: *Parallel Robots (Second Edition)*. Springer Publisher, Dordrecht, The Netherlands, 2006.
148. Michel, O.: Professional Mobile Robot Simulation. International Journal of Advanced Robotic Systems, vol. 1, no. 1, p. 39-42, 2004
149. Miller, K.: *Optimal design and modeling of spatial parallel manipulators*. In: Int. J. of Robotics Research, **23**(2004), p.127–140.
150. Mohsen Mahvash, M. et al.: *Modeling the Forces of Cutting with Scissors*, IEEE Trans Biomed Eng. 2008 March. **ISSN:** 0018-9294, Vol.55(3), p. 848 – 856
151. N. Diolaiti, G. Niemeyer, F. Barbagli, and J. K.Salisbury, "Stability of haptic rendering: Discretization, quantization, time-delay and coulomb effects," IEEE Trans. Robot., vol. 22, no. 2, pp. 256–268, 2006
152. N. Galoppo, "Animation, Simulation, and Control of Soft Characters using Layered Representations and Simplified Physics-based Methods" Dissertation submitted to the faculty of the University of North Carolina Chapel Hill, 2008.
153. N. K. Govindaraju, S. Redon, M. C. Lin and D. Manocha, "CULLIDE: Interactive Collision Detection Between Complex Models in Large Environments using Graphics Hardware", M. Doggett, W. Heidrich, W. Mark, A. Schilling (Editors), Graphics Hardware, 2003.
154. Nicolescu, M. and Mataric, M.: Natural methods for robot task learning- Instructive demonstrations, generalization and practice, In *Proc. Intl Joint Conf. on Autonomous Agents and Multi-agent Systems (AAMAS)*, Melbourne, Australia, pp. 241-248, 2003.
155. Okamura, A. M. et al.: *The Haptic Scissors*: *Cutting in Virtual Environments*. Proceedings of the 2003 IEEE International Conference on Robotics & Automation Taipei, Taiwan, September 14-19, 2003; *Feeling is believing: Using a Force-Feedback Joystick to Teach Dynamic Systems*, In ASEE Journal of Engineering Education, Vol. 91, No. 3, 2002, pp. 345-349.
156. P. Chiacchio, S. Chiaverini, L. Sciavicco, B. Siciliano: *Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy*, Int. J. Robot. Res. 10(4), 410–425 (1991)
157. Pardowitz, M., Zoellner, R., Knoop, S. and Dillmann, R.: Incremental learning of tasks from user demonstrations, past experiences and vocal comments, In *IEEE Trans. on Systems, Man and Cybernetics, Part B*, vol. 37, no. 2, pp. 322-332, 2007.
158. Parker, L.E.: *Distributed Algorithms for Multi-Robot Observation of Multiple Moving Targets*. Journal Autonomous Robots, Vol. 2, No. 3, May 2002, p. 231-255, ISSN 0929-5593, Springer Netherlands Publisher

159. Pasch K.A. and Seering W.P. 1984. On the drive systems for high-performance machines, *ASME J. of Mechanisms, Transmissions and Automation in Design*, Vol. 106, pp. 102-108.
160. Pastor, P., Hoffmann, H., Asfour, T., and Schaal, S.: Learning and generalization of motor skills, by learning from demonstration. In *International Conference on Robotics and Automation*, pp. 763–768, 2009, Piscataway, NJ: IEEE.
161. Piazzi, A. And Visioli, A. 2000. Minimum-time system inversion based motion planning for residual vibration reduction, *IEEE Transactions on Mechatronics*, Vol. 5(1), pp. 12–22.
162. Pollard, N., Hodgins, J., Riley, M. and Atkeson, C. : Adapting human motion for the control of a humanoid robot, In *IEEE International Conference on Robotics and Automation*, 2002.
163. Popescu, V. G., Burdea, G. C.: *Dextrous Haptic Interface for Jack*. In Proceedings of the ASME Dynamic Systems and Control Division, DSC-Vol. 64, Pp. 189-194, ASME
164. Q.J. Ge : *An inverse design algorithm for a G2 interpolating spline motion*, In Advances in Robot Kinematics and Computational Geometry. Kluwer Academic Publishers, Norwell, MA, 1994, p. 81- 90.
165. Q.J. Ge : *An inverse design algorithm for a G2 interpolating spline motion*, In Advances in Robot Kinematics and Computational Geometry. Kluwer Academic Publishers, Norwell, MA, 1994, p. 81- 90.