1-1-2005

Virtual design and modeling of various manufacturing processes for remote fabrication of transmuter fuel

Jamil Mohamad Renno
University of Nevada, Las Vegas

VIRTUAL DESIGN AND MODELING OF VARIOUS MANUFACTURING PROCESSES FOR REMOTE FABRICATION OF TRANSMUTER FUEL

by

Jamil Mohamad Renno

Bachelor of Science in Mechanical Engineering
University of Nevada, Las Vegas
May 2003

A thesis submitted in partial fulfillment
of the requirements for the

Master of Science Degree in Engineering
Department of Mechanical Engineering
Howard R. Hughes College of Engineering

Graduate College
University of Nevada, Las Vegas
May 2005

﴿قُلْ إِنَّ صَلاَتِي وَنُسُكِي وَمَحْيَايَ وَمَمَاتِي لِلّهِ رَبِّ الْعَالَمِينَ۞لاَ شَرِيكَ لَهُ وَبِذَلِكَ أُمِرْتُ وَأَنَاْ أَوَّلُ الْمُسْلِمِينَ﴾
(الأنعام: ١٦٢-١٦٣)

﴿Say: Truly, my prayer and my service of sacrifice, my life and my death, are (all) for Allah, the Cherisher of the Worlds ۞No partner hath He: this am I commanded, and I am the first of those who bow to His will﴾ (The Holly Quran, 6:162-163)

﴿وَقُل رَّبِّ زِدْنِي عِلْمًا﴾ (طه: ١١٢)

﴿Say: My Lord! Increase my knowledge﴾ (The Holly Quran, 20:112)

UNLV
UNIVERSITY OF NEVADA LAS VEGAS

Thesis Approval

The Graduate College
University of Nevada, Las Vegas

April 8, , 2005

The Thesis prepared by

Jamil M. Renno

Entitled

Virtual Design and Modeling of Various Manufacturing Processes for Remote Fabrication of Transmuter Fuel

is approved in partial fulfillment of the requirements for the degree of

Master of Science in Engineering

Georg Mauer
Examination Committee Chair
Dr. Georg F. Mauer

Dean of the Graduate College
Dr. Gale Sinatra

Examination Committee Member
Dr. Mohamed B. Trabia

Examination Committee Member
Dr. Woosoon Yim

Graduate College Faculty Representative
Dr. Moses Korakouzian

1017-53

ABSTRACT

Virtual Design and Modeling of Various Manufacturing Processes for Remote Fabrication of Transmuter Fuel

by

Jamil Mohamad Renno

Dr. Georg F. Mauer, Examination Committee Chair
Professor of Mechanical Engineering
University of Nevada, Las Vegas

As currently envisioned, over 70,000 tons of high-level nuclear waste would be stored inside the planned Yucca Mountain repository. After emplacement, the site must be maintained and guarded for over 10,000 years.

The reprocessing of spent nuclear fuel is a possible alternative to geological storage. Here, depleted Uranium would be separated from Plutonium and Minor Actinides. While Plutonium can be 'burned' in commercial nuclear reactors, the minor actinides would be transmuted into other elements. The large-scale deployment of remote fabrication and refabrication processes (approx. 100 tons of Minor Actinides (MA) annually) will be required. Process automation has the potential to decrease the cost of remote fuel fabrication and to make transmutation a more economically viable process. Reprocessing and transmutation would reduce the high-level waste volume by over 99%, and reduce the lifetime of the repository to approximately 300 years.

The objective of this thesis is the virtual design and simulation of manufacturing processes for transmuter fuel fabrication. Properly designed robotic work cells would

likely result in reduced cost of operation as well as increased reliability by reducing the potential for human error during materials handling operations. The candidate fuel manufacturing processes are being modeled using the MSC.visualNastran® and ProEngineer® simulation software tools. The simulation models will permit the detailed performance and safety assessments of all mechanical components in the manufacturing hot cell.

TABLE OF CONTENTS

LIST OF FIGURES

ACRONYMS AND ABBREVIATIONS

Acronyms

3-D	Three Dimensional
AECL	Atomic Energy of Canada Limited
ALARA	As Low as is Reasonably Achievable
ATW	Accelerator Transmutation of Waste
Btu	British Thermal Unit
CAD	Computer Aided Design
CSNF	Commercial Spent Nuclear Fuel
COGEMA	Compagnie General des Matieres Nucleaires
COM	Component Object Model
DEC	Spent Fuel Disassembly and Consolidation System
DOE	United States of America Department of Energy
DOF	United States of America Department of Defense
FMEF	Fuels and Materials Examination Facility
GmbH	Gesellschaft Mit Beschränkter Haftung (German: Limited Liability Company)
IGRIP	Interactive Graphical Robot Instruction Program
LWR	Light Water Reactor
MATLAB	MATrix LABoratory (software)
MOX	Mixed Oxide (fuel)
MPC	Multipurpose Canister
MSM	Master Slave Manipulator
OLE	Object Linking and Embedding
PUREX	Plutonium Extraction
PWR	Pressurized Water Reactor
SAF	Secure Automated Fabrication
SSC	Systems, Structures, and Components
THORP	Thermal Oxide Reprocessing Plant
TRISO	TRistructioral ISOtropic
UREX	URanium EXtraction
vN4D	MSC.visualNastran 4D (software)
WALS	Weight And Leak Check System

Abbreviations

Am	Americium
Ba	Barium
C	Celsius
Cm	Curium
Cs	Cesium
gal	Gallon
He	Helium
hr	Hour
I	Iodine
lb	Pound
Mev	Mega Electron Volt
MT	Metric Ton
min	Minute
Na	Sodium
Np	Neptunium
Po	Polonium
Pu	Plutonium
Ru	Ruthenium
rem	Rotogen equivalent man
sec	Seconds
Tc	Technetium
Te	Tellurium
Th	Thorium
UF_6	Uranium Hexafluoride
UO_2	Uranium Dioxide
wt	Weight
Xe	Xenon

ACKNOWLEDGMENTS

First and foremost, I would like to thank Almighty Allah for His Help. This work, and every other work I accomplished, was completed with Allah's blessings first; my humble efforts come second. Prophet Mohamad (PBUH) said: "He who doesn't thank people doesn't thank Allah," (Abu Dawud, Book 41: 4793). Below I express my thanks and gratitude to the people who have helped me complete this work. Those that I could not mention, please accept my apology; you know that you are on my mind.

I would like to start by thanking my parents, Mohamad and Inaam Renno. What I owe them cannot be described by words. They have given me everything. Mr. Mohamad Renno's way of life was, still, and will always be a constant teaching source for me. The only way I can thank him, is just by raising my kids the way he raised me. Mrs. Inaam Renno is the light of my life, her love and care has always driven me to push the limits. I would never forget her teaching me the world's capitals at the age of two. If there is something that I can be proud of, it is being their son. Let it be known, this thesis, every work I have ever done, and every work I will ever do is solely dedicated to Mohamad and Inaam Renno.

I would like to thank my cousins, Mohammed and Rima Sleiman. Mohammed and Rima have been most helpful and generous with me since I came to the U.S.A. Their help throughout my years at UNLV will always be appreciated, and will never be forgotten.

I would also like to thank Dr. Georg Mauer for not only guiding me in the development of this work, but also for treating me as a colleague and friend. Dr. Georg Mauer is a great example of efficiency and professionalism. I would also like to thank Mr. Richard Silva whose thesis has been of great help for me. Richard is an individual with drive; he is a dedicated engineer and a great colleague.

I have been very fortunate to be part of the Transmutation Research Program. As the director, Dr. Anthony Hechenova has been an inspiration. His commitment to the program is a substantial asset for the University.

I would also like to thank my committee members: Dr. Mohammed Trabia, Dr. Woosoon Yim, and Dr. Moses Karakouzian for serving on the committee, taking time to review the prospectus, and counseling throughout this work.

Special gratitude is expressed to both my grandfathers (late Mr. Jamil Ahmad Renno and Mr. Abdulhamid Shehab) for their encouragement. To grandfather Jamil: the tall boy holding your name did not, and will never disappoint you. May the souls of my grandparents rest in peace.

Finally, my colleagues at the Measurements and Controls Lab: Masoud Fegghi, Kofi Cobbinah, Vijayasarathy Subramanian, Surya Kiran Parimi, Prashanth Reddy Duvvuru Kamakshi, Michael Costa, Brinda Holur Venkatesh, and Chris Kawa: they have made the lab a pleasant place to study and do research. My undergraduate studying partners: Derek Taguchi and John Motaka have helped me a lot to integrate in the U.S. education system during my undergraduate years at UNLV. I would also like to thank Dr. Jae-Kyu Lee for showing that it is actually possible to graduate.

CHAPTER 1

INTRODUCTION AND BACKGROUND

The continued growth of the world's population, combined with generally rising living standards, is resulting in a rapidly increasing rate of utilization of energy. It has become apparent in recent years that the conventional sources, such as water power and the fossil fuels, namely, coal, oil and natural gas, will be insufficient to supply the energy demand even in the foreseeable future. In fact, in some highly industrialized countries, such as the United Kingdom and Japan, there is already a shortage of fossil fuels. The discovery that energy can be obtained from nuclear fission has opened up a new source of power of very great importance, with fuel materials that are completely novel in character. Instead of coal and oil, the basic source of energy are the elements Uranium (U) and Thorium (Th), see [3].

The device in which the release of energy is achieved is called a nuclear reactor. In most reactors the nuclear fuel (or reactor fuel) material is in solid form and the structural unit containing, or consisting of, this material is called a fuel element. The preparation of reactor fuel material has introduced a number of novel, interesting, and often difficult metallurgical problems. In addition, the fabrication of the fuel elements has required the development of special techniques. This introductory chapter presents the essential principles of nuclear energy generation, and explains the concept of transmutation, so as

to provide the necessary background for the remainder of this work, in which the fabrication of nuclear fuel will be treated.

1.1 Atomic Structure and Isotopes

The following sub-sections briefly explain the atomic structure and the phenomenon of isotopes. The concept of radiation will be explained as well.

1.1.1 Structure of the Nucleus

According to the modern atomic theory, every atom is formed of a positively charged nucleus surrounded by a negatively charged cloud of electrons. With the exception of Hydrogen (H), the nucleus is made of positively charged protons (*p*) and neutral neutrons *(n)*. The neutrons serve as isolators between the positive protons.

1.1.2 Isotopes

The sum of the numbers of protons and neutrons in a nucleus is called the mass number of the element. The number of protons of an element is called the atomic number of the element.

A given element characterized by its atomic number, sometimes occurs in two or more forms having different numbers of neutrons in their nuclei. These forms with the same atomic numbers but different mass numbers are called isotopic forms, of which the most important, found in nature have mass numbers of 235 and 238, respectively. These are designated as (U^{235}) and (U^{238}). The lighter isotope, (U^{235}), is present to the extent of 0.71 wt.% in normal Uranium (U); the remainder consists almost entirely of (U^{238}), with a very small amount of (U^{234}) and traces of other Uranium isotopes. It is of interest to note that (U^{235}) is the only isotope found in nature which can be applied directly for the release of nuclear energy by fission. When a (U^{235})nucleus absorbs an additional neutron,

there is a considerable probability that it will split into two lighter nuclei and, at the same time, release a relatively large amount of energy.

1.2 Radioactivity

All but 20 of the elements found in nature exist in two or more isotopic forms, making a total of about 325 different nuclear species, or nuclides, as they are often called. Of these, somewhat more than 270 are stable nuclides; whereas about 50 others are unstable, i.e., radioactive. For example, when a Uranium atom is split into two lighter elements, it is possible for excess neutrons to be available for splitting additional atoms. Elements with atomic numbers equal to or greater that 84, beginning with Polonium (Po) exist only in radioactive forms.

1.2.1 Radioactive Decay

The process of radioactive decay of an unstable nuclide may be regarded as a nuclear change in the direction of increasing the nuclear stability. In most cases, the radioactive nucleus expels either an alpha particle *(α)*, which is identical with the Helium (He) nucleus, or a beta *(β)* particle which is an electron. Depending upon whether the unstable nucleus has too many protons or too many neutrons for stability, it emits positive or negative beta particles, respectively. As far as their general behavior is concerned, there is no difference between positive-beta and negative-beta emitters.

Radioactive changes are frequently accompanied by the emission of gamma rays *(γ)*. These are a form of penetrating electromagnetic radiation of high energy similar to X-rays. In fact, the only distinction that can be made between gamma rays and X-rays is that the former accompany nuclear processes and the latter arise from changes outside the nucleus.

Because they represent a potential health hazard if they enter the body, it is important to know how far alpha and beta particles and gamma rays can travel from their source. The range of alpha particles depends upon their energy, but those of radioactive origin are stopped, and become harmless, within 1-3 in. of air, and they are unable to penetrate thin sheets of paper or aluminum. Beta particles can travel greater distances in a given medium than alpha particles of the same energy, but they are stopped relatively easily by substances of moderate or high density. The situation is quite different with gamma rays. These radiations can penetrate to considerable distances in air and can even pass throughout appreciable thicknesses of dense materials. Unlike alpha and beta particles, gamma rays are never completely stopped, but they can be attenuated either by distance or by passage through dense matter, e.g., concrete, iron, or lead to the extent that they are no longer a significant health hazard.

1.2.2 Radioactive Half-Lives

The rate of radioactive decay, which determines the rate of emission of alpha or beta particles and, in many cases, gamma rays, is expressed by the half-life of the given nuclide. This is defined as the time required for one half of any initial quantity to transform into its decay product. Every radioactive species has a characteristic half-life that is independent of its amount or state. Measured half-lives range from a few billionths of a second to billions of years, although nuclides with shorter or longer half-lives, respectively, are undoubtedly capable of existence.

1.3 Neutron Reactions

In addition to spontaneous nuclear reactions, represented by radioactivity, in which one nucleus is changed into another as the result of the expulsion of a particle, there are

many nuclear reactions, which can be brought about artificially, resulting from the absorption of one particle or another. Among such reactions, those in which a neutron is absorbed are of special interest. Neutrons are normally bound in atomic nuclei, but it is possible to obtain them in the free state, and therefore their interaction with matter can be studied. Apart from the fission process, there are three types of neutron-nuclear interactions of importance in the operation of nuclear reactors. In addition to absorption reactions leading to the formation of new nuclei, there are two kinds of scattering collisions, namely, elastic and inelastic, that result only in an exchange of energy between the neutron and the nucleus.

1.4 The Fission Process

When certain nuclei of high atomic number (and mass number) capture neutrons, the resulting excited compound nucleus, instead of emitting its excess energy as gamma radiation, undergoes fission. That is, it splits into two lighter nuclei having masses very different from that of the original heavy nucleus. Only three nuclides are known which have relatively long half-lives, so that they can be stored for appreciable times, and which are fissionable by neutrons of all energies from thermal values (or below) to millions of electron volts. These nuclides are (U^{233}), (U^{235}), (Pu^{239}), of these, only (U^{235}) occurs in nature, the other two being produced artificially from (Th^{232}) and (U^{238}), respectively, see [3].

1.4.1 Neutrons from Fission

The importance of fission, as far as the utilization of energy is concerned, lies in two facts: first, the process is associated with the release of a large amount of energy per unit mass of nuclear fuel, and, second, the fission reaction, which is initiated by neutrons, is

accompanied by the liberation of neutrons. It is the combination of these two circumstances which makes possible the design of a nuclear reactor in which a self-sustaining fission chain reaction occurs with the continuous release of energy.

1.4.2 Energy from Fission

For each nucleus of (U^{233}), (U^{235}), or (Pu^{239}) undergoing fission, about 200 Mev of energy becomes available. Expressed in more familiar units, it may be stated that the fission of all the nuclei in 1 lb of fissionable material would release about 3.6×10^{10} Btu of heat. This is the same quantity of energy that would result from the complete combustion of 1400 tons of average coal or 300,000 gal of hydrocarbon oil. Of this energy, about 86% appears at the time of fission as kinetic energy, 83% being carried by the fission fragments and 3% by the fission neutrons. Gamma rays accompanying fission, as well as those produced by radiative capture of neutrons in the fuel and other materials present in the reactor, contribute about 6% of the total energy. The remaining 8% is released over a period of time in the form of beta and gamma radiations from the fission products and from any radioactive nuclei that may be formed by radiative capture reactions. In a nuclear reactor all the fission energy eventually appears as heat, liberated either in the fuel element or in other materials in which the various radiations are absorbed.

Since it takes some time for the decay energy of fission products and other radioactive nuclides to attain a steady state, the heat output of a nuclear reactor, under given operating conditions, will increase steadily at first until a constant value is attained. Correspondingly, after the reactor is shut down and the chain reaction is terminated, the decay energy will continue to be released as heat.

1.4.3 Fission Products

The lighter nuclei formed in the fission process are called fission fragments. There are a number of different ways in which fission of a particular nuclear species occurs as a result of the absorption. Consequently many different fission fragments are formed in widely varying individual amounts, ranging from less that 10^{-5} to over 3%, of the total. The fission fragments undergo, on the average, three stages of negative-beta decay, as stated earlier, frequently accompanied by gamma radiation. The resulting mixture, to which the general term fission products is applied, is thus a complex, highly radioactive system of numerous isotopes, in differing proportions, of a considerable number of elements.

In the fission of (U^{235}) by thermal neutrons, for example, about 80 different fission-fragment nuclei are formed as primary fission products having mass numbers ranging from about 72 to 160. Radioactive decay of the fission fragments leads to the production of a mixture of roughly 250 radioactive and stable isotopes. Some 30 or more different elements from Zinc (atomic number 30) to Gadolinium (atomic number 64), have been detected among the fission products after a short interval. This complex mixture emits a variety of beta particles and gamma rays for a long time, although the activity falls off steadily as the various radionuclides decay. The delayed neutrons are also emitted soon after fission, but these cease to be significant within few hours.

1.5 Problems of Nuclear Energy

Using the above approach, large amounts of energy can be obtained from small quantities of fuel and, in principle, any reactor can produce almost unlimited power, [51]. This suggests that nuclear power should be abundant and cheap and should easily fulfill

the energy needs of mankind. Unfortunately, this promise cannot be realized for two basic reasons:

- Nuclear fuels are very expensive. They are not abundant in nature, are difficult to obtain in usable form and are dangerous to handle.
- Radiations subsequent to fission make the entire operation very expensive. They lead to greatly increased costs for a reactor plant, compound the danger of a fuel-element failure or nuclear accident, and cause extreme difficulty in waste disposal and fuel recycle.

Beginning in the 1940's, the U.S. government engaged in a very vigorous program of research, development, and utilization of nuclear energy. One part of the program was focused on the design and production of nuclear weapons. The other part of the effort was focused on the development of nuclear power reactors for both military and civilian use. From the beginning, it was clear that the use of nuclear energy results in the generation of considerable amounts of high-level waste, containing radionuclides that are intensely radioactive and/or have long half-lives. Though it was recognized quite early that high-level waste would have to be managed safely without unacceptable risks to humans, for many decades other parts of the nuclear program (including interim storage) were given higher priority. The spent reactor fuel was stored in pools at the reactor site, and the high-level liquid wastes resulting from processing of defense reactors' waste were put into underground tanks as temporary solutions to the problem. The age of the oldest of these "temporary" tanks containing processed liquid wastes is now approaching 59 years. Spent civilian-reactor fuel storage has been uneventful, but the high-level waste tanks are corroding and some have developed leaks, creating a hazardous situation. At many reactor sites these storage pools are nearly full—a situation requiring the construction of additional on-site pools or dry storage facilities, see [35].

It is expected that by about 2010 the U.S. civilian nuclear power reactors will have produced about 70,000 tons of spent fuel. This will contain about 90% of all high-level radioactive waste in the U.S. nuclear program. The remaining spent nuclear fuel will result from the military program.

Extensive work has been done on research, development, and evaluation of methods for the ultimate safe disposal of this spent fuel and high-level radioactive waste. These studies have almost concluded that the most practical approach would be to vitrify the liquid high-level wastes in a glass matrix. The resulting glass would then be encapsulated in suitable containers and buried in a specially selected and evaluated stable geological formation deep underground. Present U.S. policy is that spent power reactor fuel would also be encapsulated and buried.

To implement the disposal process, several recent administrations have announced their intention to establish a national repository for high-level radioactive waste. They developed a program schedule and passed legislation to provide funding mechanism needed to carry out the program. However, the schedule for the program has been marked by continuous delays, and successive plans have had to be abandoned, resulting in billions of dollars of wasted expenditure. Some reasons for this dilemma are technical, but many are non-technical. The stalemate on the Yucca Mountain site in Nevada, sponsored by the U.S. Department of Energy (DOE) is an example of such a case. When Yucca Mountain was identified by the U.S. Congress as the first site to be evaluated, the Nevada governor initiated legal steps to try to stop the federal government from taking any steps to evaluate the site, even though it is on federal property.

Because of such difficulty, the DOE continues to explore other disposal options that might circumvent such problems. Some of the proposed solutions have focused on separating the hazardous long-lived radioactive nuclides in the waste and transmuting them by neutron bombardment to form nuclides that would be either stable or radioactive with a much shorter half-life. During the last decade there has been a renewed interest by few countries in such proposals, and some technological progress has been reported. This has led several of the DOE national laboratories to reexamine this concept to see if it might be put to use. Although this approach is technically feasible, its use involves several practical problems, which poses sever engineering and material challenges.

1.6 Transmutation: The Need

The commercially spent nuclear fuel (CSNF) should be isolated from the environment, especially water. The DOE intends to use corrosive resistant waste packages to store the CSNF. The outer surface, or shell, of the waste package will be manufactured out of nickel-based alloy (Alloy-22), which is expected to resist corrosion for more than 10,000 years; hence isolating the CSNF for an equal period of time.

The natural rock barriers at the Yucca Mountain site provide even more isolation of the waste packages. The arid conditions of the site allow negligible amounts of water to come in contact with the waste packages, hence corroding them and transferring the radionuclides to the ground water. Significant scientific effort has been underway to ensure that radionuclides are not capable of reaching the ground water for 10,000 years. There are inherent uncertainties in predicting the performance of any system, structure or component for such a long period of time.

This uncertainty stems from storing some long-lived radionuclides possessing extensive half-life times. For example, the isotope ^{239}Pu has a half-life of 20,000 years. Approximately one percent of the CSNF contains long-lived fission products. These long-lived fission products such as Technetium (Tc), related halogens such as Iodine (I), and related actinides such as Plutonium (Pu), Americium (Am), Neptunium (Np), and Curium (Cm) are all byproducts of the nuclear fission process, see [51].

If those long-lived radionuclides contained within CSNF can be converted into a more mild form possessing a shorter half-life, the isolation requirements of the Yucca Mountain site can be reduced. Reducing the actinides, halogens, and other fission products half-lives can theoretically reduce the waste isolation requirements of the Yucca Mountain from 10,000 years to a few thousand years; between 1,000 – 3,000 years, see [51]. Figure 1.1 illustrates the scope of the problem. The success of such conversion will significantly reduce the uncertainty in the prediction of the performance of the Yucca Mountain repository.

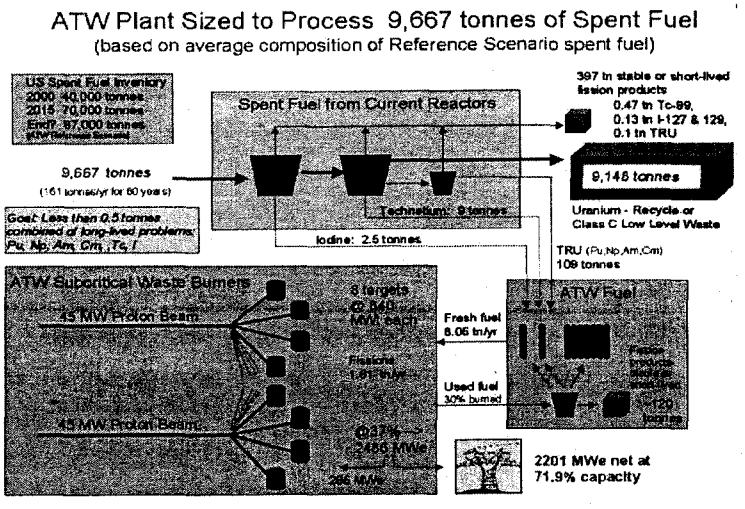


Figure 1.1 Spent Fuel's Processing Needs

1.6.1 Transmutation: Overview

The word transmutation originates from the never-realized goal of ancient alchemists to transform, transmute, base metals into gold. Today, scientists seek ways to transmute radioactive waste into non-radioactive elements, thereby eliminating the radiological hazards and waste disposal problems. An example of two radioactive isotopes that can be transmuted into less hazardous forms are (Tc^{99}) and (I^{129}). Both isotopes are very long lived and require disposal strategies that will isolate them from the environment for long periods of time. In addition, both (I^{129}) and (Tc^{99}) are considered difficult to isolate because they dissolve readily in groundwater and move easily throughout the ecosystem. Irradiation of the long-lived (Tc^{99}) by neutrons will cause it to absorb a neutron and become (Tc^{100}), which undergoes complete radioactive decay into stable Ruthenium (Ru^{101}) within minutes. Similarly, (I^{129}) can be transformed by neutron absorption into stable Xenon (Xe) isotopes.

Another class of radioactive wastes that can be transmuted into less hazardous forms are the actinide elements, particularly the isotopes of (Pu), (Np), (Am) and (Cu). When irradiated with neutrons in a nuclear reactor, these isotopes can be made to undergo nuclear fission, destroying the original actinide isotope and producing a spectrum of radioactive and non-radioactive fission products. Isotopes of (Pu) and other actinides tend to be long-lived with half-lives of many thousands of years, whereas radioactive fission products tend to be shorter-lived (most with half-lives of 30 years or less). From a waste management viewpoint, transmutation of actinides eliminates a long-term radioactive hazard while producing a shorter-term radioactive hazard instead.

A challenging aspect of this waste management strategy is the required waste partitioning. Just as household wastes must be partitioned into categories, such as paper,

glass, and metals, before they are recycled, radioactive waste must also be sorted before being recycled back into nuclear reactors.

The hazardous long-lived elements can be separated by various means and then transmuted in power-producing reactors as well as accelerator driven systems. Transmutation cannot replace the current need for a national geological repository, such as Yucca Mountain, yet a successful transmutation program will definitely reduce the long-term requirements for the CSNF disposal. As a result, transmutation could remove the waste management issue as a barrier to expanded use of nuclear power by addressing the long-term environmental and economic issues faced by the U.S., and the world, see [48]. Figure 1.2 shows a schematic of the proposed transmutation process.

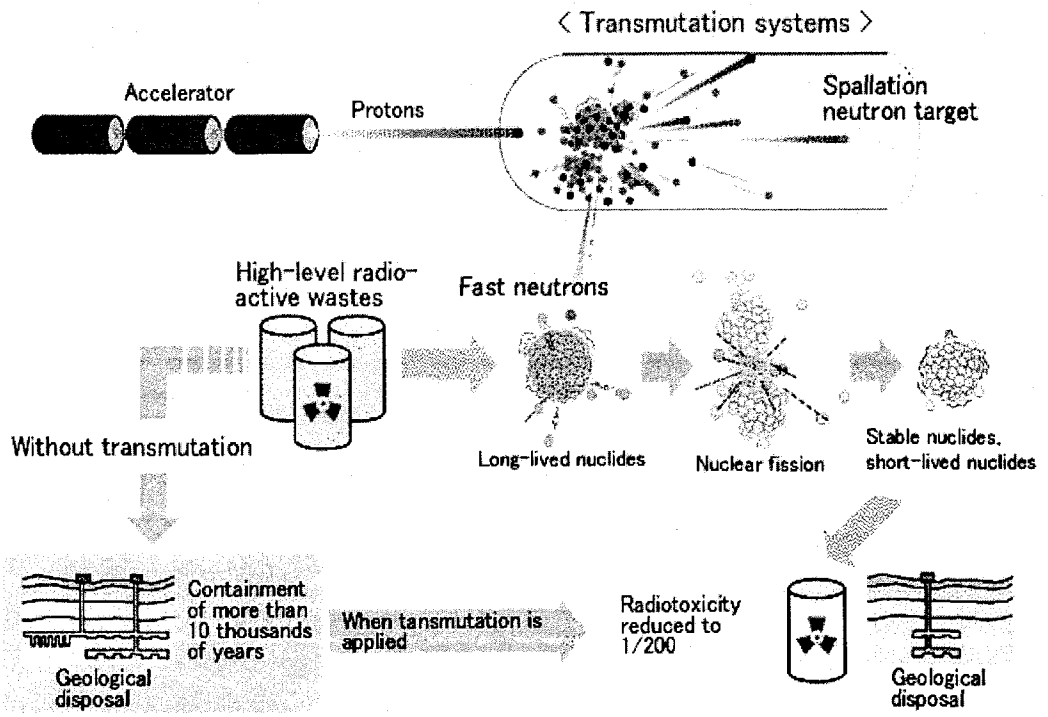


Figure 1.2 Illustration of Accelerator Driven Transmutation Process

To illustrate the potential power of transmutation, consider the following case as an example. The fission-product decay for the neutron absorbing isotope Xenon (X^{135}) is shown as the Tellurium (Te^{135}) decay chain shown below, see [2]:

$$\text{Fission} \Rightarrow {}_{52}Te^{135} \Rightarrow {}_{53}I^{135} \Rightarrow {}_{54}Xe^{135} \Rightarrow {}_{55}Cs^{135} \Rightarrow {}_{56}Ba^{135}$$

Where Table 1.I shows the related half-life for the Tellurium (${}_{52}Te^{135}$) decay chain. Cesium (Cs) is contained within spent fuel, and with a half-life of 2.3×10^{6} years, will remain radiotoxic for extended periods of time. It would be advantageous to transform the Cesium (${}_{55}Cs^{135}$) nuclide into stable Barium (${}_{56}Ba^{135}$) nuclide.

Table 1.I. Tellurium Decay Chain

Element	Nuclide	Half-Life
Tellurium	${}_{52}Te^{135}$	< 2 min
Iodine	${}_{53}I^{135}$	6.68 hr
Xenon	${}_{54}Xe^{135}$	9.13 hr
Cesium	${}_{55}Cs^{135}$	2.3×10^{6} year
Barium	${}_{56}Ba^{135}$	Stable

A proposed method to transform the Cesium nuclide is through transmutation by neutron bombardment. The current philosophy for the source of the neutron flux is from a non-critical, accelerator-driven reactor. When a Cesium (${}_{55}Cs^{135}$) nuclide is bombarded by a neutron, it transmutes into Cesium (${}_{55}Cs^{136}$) while emitting a gamma ray. The newly created Cesium nuclide, with a half-life of approximately 13 days decays naturally to stable barium (${}_{56}Ba^{136}$), while emitting an easily shielded beta particle. This process is generally shown in Figure 1.3.

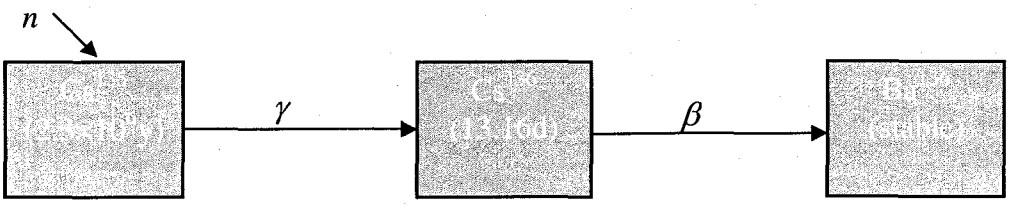


Figure 1.3 Neutron Absorption and Natural Decay of Cesium

While not all nuclides behave like Cesium ($_{55}Cs^{135}$) and transmute to a nuclide that naturally decays to a stable state, transmutation is currently being investigated as a means of reducing the long-lived actinides to more benign actinides. Transmutation has the potential for decreasing the performance uncertainty of a deep geologic repository to contain these radionuclides over extended periods of time, see [49].

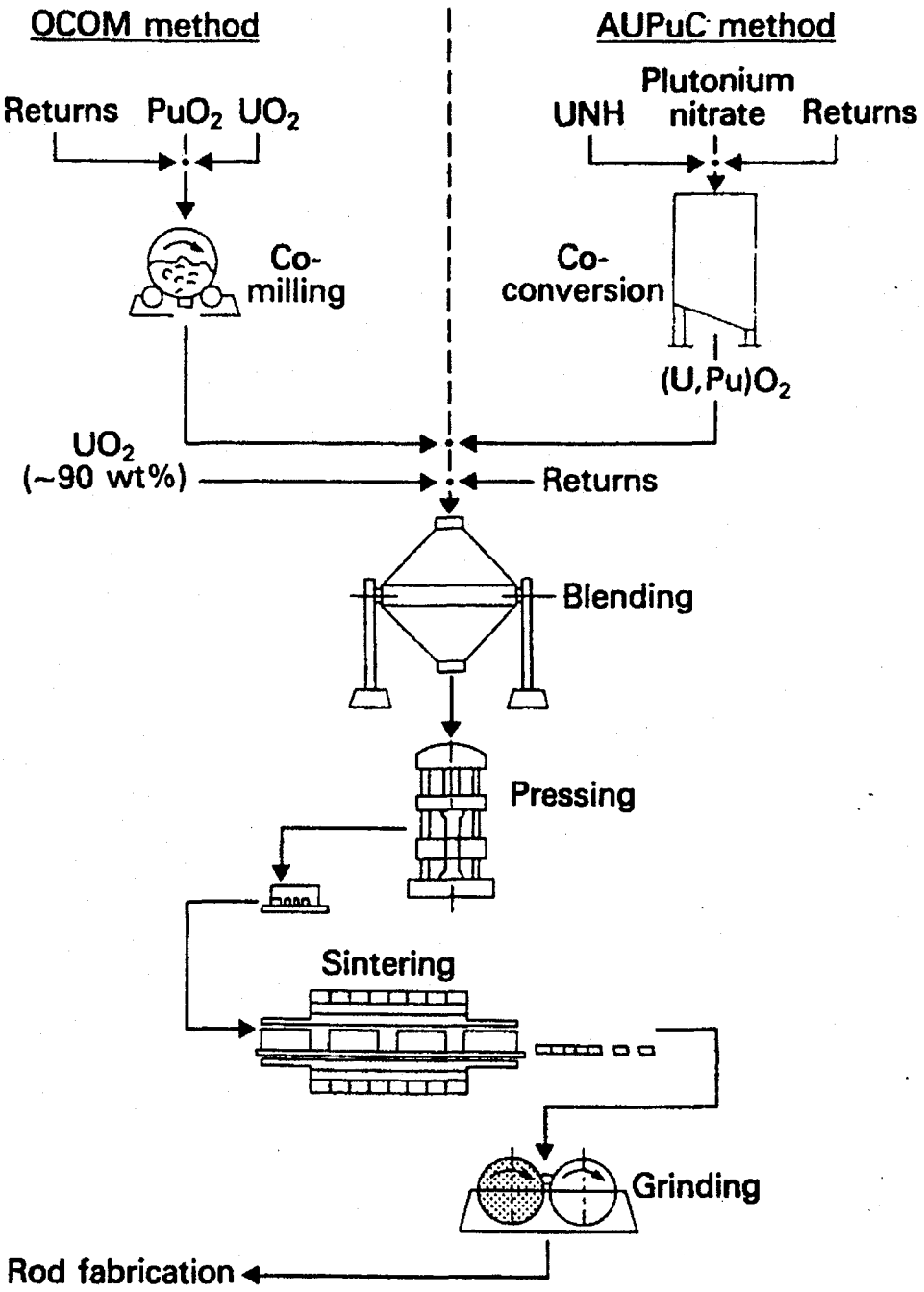
1.6.2 Overview of the Fuel and Conversion to Transmuter Fuel

Various fuel types are being considered for use in an accelerator driven reactor, where the fuel contains the long-lived actinides for transmutation. Since the fuel will contain hazardous actinides, the radiation dose will be significantly higher than that experienced in normal fuel manufacturing. The fuel manufacturing should take place in a well shielded place. Standard glove box technology used in fuel manufacturing is not sufficient for the shielding required. Hence, a hot cell environment is proposed for the manufacturing of the transmuter fuel.

Such hot cells will be required to produce large amounts of transmuter fuel. The overall throughput rates are high for remote operations. The equipment to be used in hot cells shall perform adequately between forty to sixty years, and the processes must allow

the use of robust equipment, that can be easily maintained, see [44]. Attention must also be given to decommissioning during the design phase.

Current fuel fabrication processes for accelerator transmutation of waste (ATW) are either based on metal casting (metallic fuels) or powder processing, leading to ceramic or dispersion fuels. Figure 1.4 shows schematically a flowchart for MOX fuel fabrication as applied for LWRs. Figure 1.5 depicts the generalized process flow for a ceramic fuel; in this particular case, a mixed oxide (MOX) fuel manufacturing process.



Flowchart for MOX fabrication for LWRs using the OCOM and AUPuC processes. (Siemens)

Figure 1.4 MOX Fuel Fabrication Process

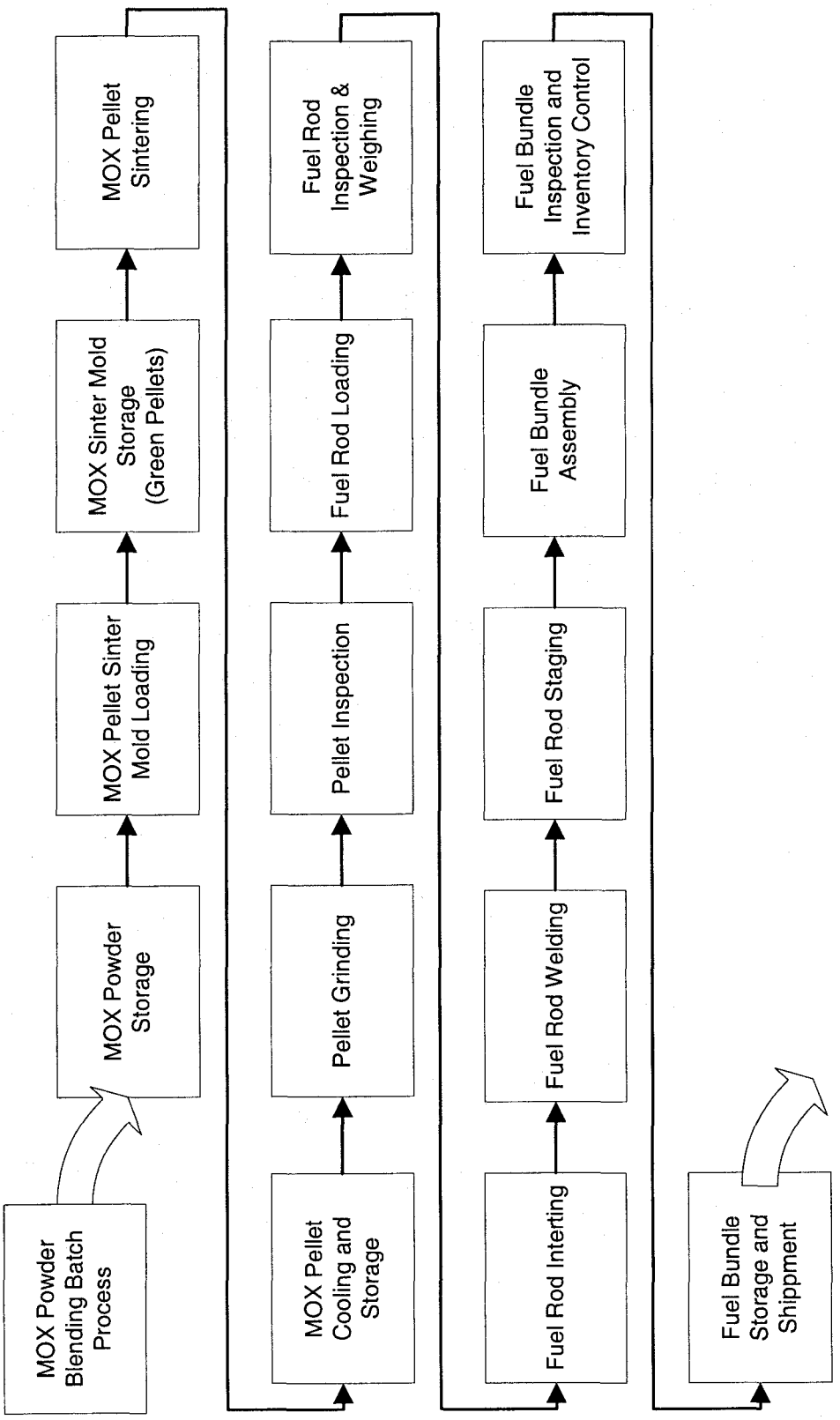


Figure 1.5 Simplified MOX Fuel Manufacturing Process Flow

Research and development on fuel reprocessing and manufacturing has been ongoing for years in the U.S. and other countries. With regard to fuel manufacturing, we may distinguish among three categories of fuel, see [46]:

- Dispersion Fuels – several subtypes exist
- Ceramic Fuels – several subtypes exist
- Metallic Fuels

For each ATW fuel type, a separate manufacturing sequence exists, as is summarized below.

Manufacturing Sequence for Dispersion Fuel, see [46]:

1. Manufacture spherical fuel particles by wet chemical process or direct reaction and attrition.
2. Coat the particles – process not yet determined
3. Embed fuel particles in matrix metal
4. Press fuel and matrix blend
5. Assemble billet, see Figure 1.6
6. Extrude billet at approximately 800 degrees Celsius into rods about 2-meters long.
7. Finish fuel rods by trimming ends and performing rod inspections, which may include radiography, dimensional tolerance checks, bonding, and clad defects.

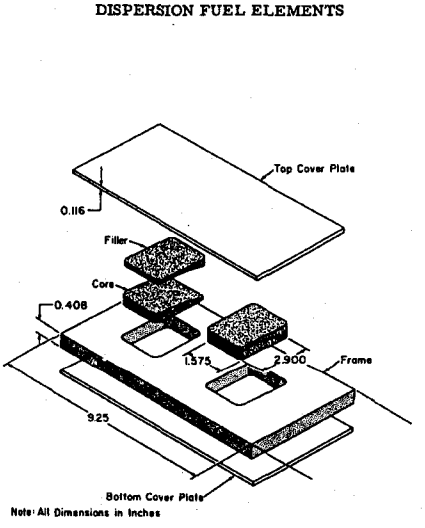


Figure 1.6 Dispersion Fuel Elements

The Manufacturing Sequence for Metallic Fuel, see [46]:

1. Cast fuel slugs, where the slug is approximately 4 to 5 millimeters in diameter and are from 0.8 meter to 1.5 meters in length.
2. Insert fuel slugs into cladding tube
3. Add bonding agent to the fuel tube, which may be sodium (Na).
4. Seal cladding tube by welding end fitting into the tube.
5. Perform fuel slug inspections, which may include radiography, dimensional tolerance checks, bonding, and clad defects

The Manufacturing Sequence for Ceramic Fuel, see [46]:

1. Manufacture particles by wet chemical process or direct reaction, where the particles are approximately 1 μm to 30 μm in diameter.
2. Compact particles into pellet form.
3. Sinter the pellets at 1400-1800 °C
4. Inspect pellets
5. Assemble pellets into cladding tube
6. Add bonding material, which may be Helium (He) or Sodium (Na)
7. Seal cladding tube by welding
8. Perform assembled fuel pin inspections, which may include radiography, dimensional tolerance checks, bonding, and clad defects

The investigated manufacturing process for ATW fuels is similar to standard Pressure/Boiling Reactor type nuclear reactor fuel manufacturing processes. Gaseous enriched Uranium Hexafloride (UF_6), which comes out of an enrichment plant, is delivered in standard cylindrical gas bottles. The (UF_6) is then converted to Uranium Dioxide (UO_2), the fuel for which is used. The (UO_2) is in the form of a powder, which is then milled to provide suitable sintering properties. The powder is formed into pellets by cold pressing, sintered at approximately 1650°C to 1750°C [53], and ground to final dimensions, as shown in Figure 1.7.

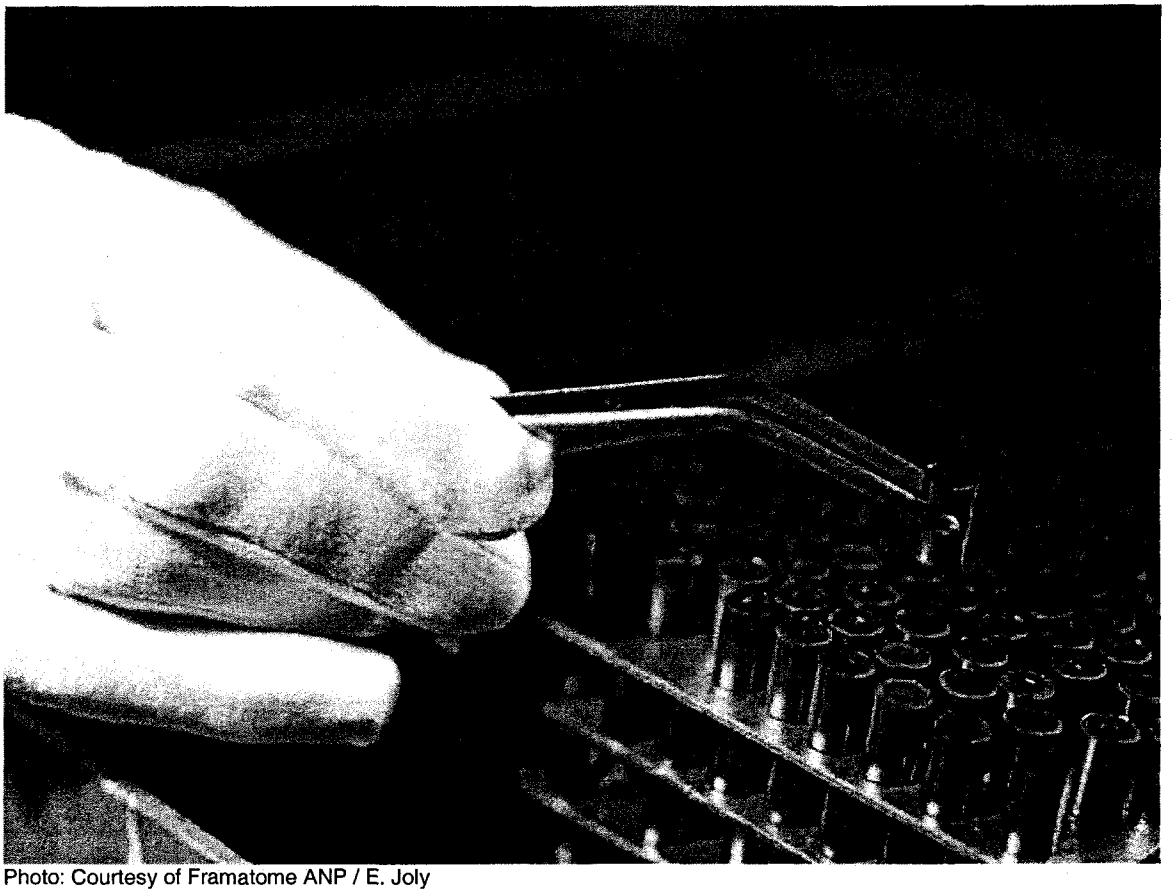


Photo: Courtesy of Framatome ANP / E. Joly

Figure 1.7 Uranium Oxide Pellets

The pellets are loaded into a zirconium alloy tube, also known as cladding, which is back-filled with helium (He) or other inert gases and sealed at each end with a welded-end plug. The fuel rods are assembled into complete fuel assemblies, as shown in Figure 1.8. The number of fuel rods in an assembly varies from 50 to 200 or more, depending on the reactor type and specific design, see [13].

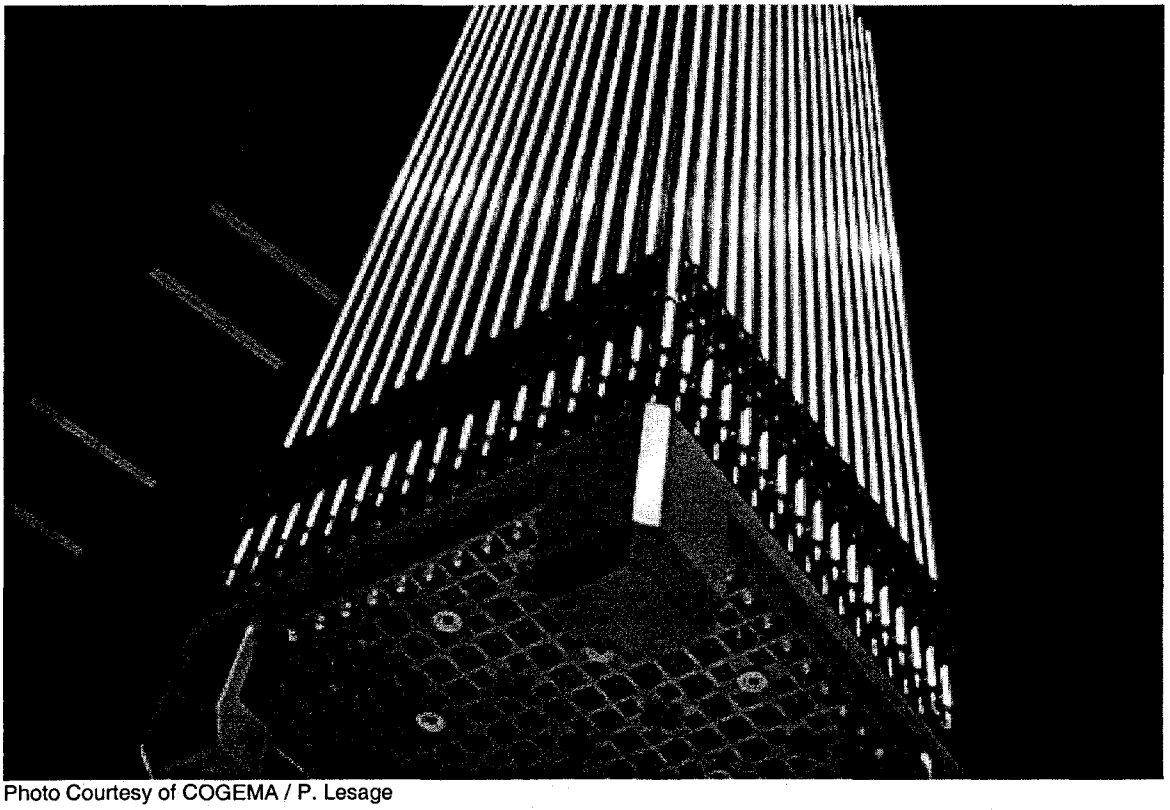


Photo Courtesy of COGEMA / P. Lesage

Figure 1.8 MOX Fuel in Complete Assembly at MELOX Plant, France

1.7 Objective of the Study

The objective of this work focuses around two primary tasks, assessing automation and robotics in the nuclear industry, and modeling the back-end of the transmuter fuel manufacturing process.

1.8 Safety Considerations in Design

Safety concerns are paramount in any nuclear facility. Current nuclear facilities implement safety programs that are based on a defense-in-depth philosophy. The defense-in-depth philosophy relies not only upon the inherent safety features of the engineered design, but also on high quality standards that provide a high degree of assurance that

accidents are not likely to occur. And if they do occur, the radioactivity and any related contamination will be contained locally, see [44].

High quality begins with the selection of highly qualified personnel to design, construct and operate a nuclear facility. Each applicant for a license to build a nuclear plant must institute a quality assurance program to be applied to design, fabrication, construction, and testing of the structures, systems, and components (SSC) of the facility. An applicant for a license to operate a plant must also provide for appropriate managerial and administrative controls to be used to assure safe operation. The function of the quality assurance program is to provide a high degree of confidence that the SSC's in the plant will perform as intended during the expected service life of the plant, see [13].

It would be advantageous to design the transmuter fuel manufacturing hot cells so that the major system components are shielded to the maximum extent possible; whereas, if a failure within a system or component occurred, it would not preclude mitigation. For example, should a robotic manipulator fail, any high-level radioactive source could be shielded so mitigation methods that could not be performed remotely could be accomplished by human intervention. This means the radiation exposure should be kept as low as is reasonably achievable (ALARA).

It would be suitable at this point to explain radiation units of measurement. The units used in this work are (rad) and (rem). The (rad) is a unit of absorbed radiation dose in terms of the energy actually deposited in the tissue. The (rad) is defined as an absorbed does of 0.01 Joules of energy per 1 Kg of tissue. The biologically effective dose in rem or mrem (millirem), is the radiation dose in rads multiplied by a "quality factor" which is

the assessment of the effectiveness of that particular type and energy of radiation, see [54].

Standard radiation zones are classified a low radiation zone, high radiation zone, and very high radiation zone, see Table 1.II below. Manned access into low and high radiation zones is only feasible by reducing the duration the individual is within the zone or by providing localized shielding. Very high radiation zones are not accessible. It is expected that the fuel manufacturing environment will be classified as a low to high radiation zone, but not likely to be a very high radiation zone.

Table 1.II Radiation Zone Classification

Zone	Minimum Exposure	Maximum Exposure
Low Radiation	N/A	$100\ \text{mrem}/\text{hr}$
High Radiation Zone	$100\ \text{mrem}/\text{hr}$	$5\ \text{rad}/\text{hr}$
Very High Radiation Zone	$5\ \text{rad}/\text{hr}$	∞

CHAPTER 2

REVIEW OF CURRENT METHODOLOGIES FOR HANDLING RADIOACTIVE MATERIAL

2.1 Department of Energy

The U.S. Department of Energy (DOE) is a great source of information not only for nuclear reactor operations and fuel manufacturing, but other automated radioactive material handling and waste management. The U.S. Department of Defense (DOF) remotely handles radiological materials produced by the military nuclear program. The DOF's activities are of sensitive nature, relating to national security concerns. Information on the DOF's activities cannot be readily obtained, and hence little information is known about how military spent nuclear fuel is handled. Therefore, the following subsections review a cross-section of the DOE facilities and the respective activities.

2.1.1 Typical Hot-Cell Applications

The design of a work cell that handles high-level nuclear wastes, sophisticated equipment, and fission materials is a formidable task requiring deep understanding and coordination between a multitude of disciplines. A well design should address, among other issues, safety, reliability, ease of maintenance, etc... The DOE published *Design Considerations*, [44], in an effort to provide basic guidance for the design of hot cells. In [44], general "rule-of-thumb" design considerations as well as good and safe practices are

discussed in details. The handbook touches on design considerations and practices for the following topics, see [44]:

- Architectural
  - Facility Layout
  - Equipment Arrangement
  - Piping Design and Layout
  - Special Systems
  - Jumpers
  - Structural Design
- Electrical Systems
- Mechanical Systems
  - Piping
  - Purge Systems
  - Pumps
  - Valves
- Instrumentation and Control
  - Control Centers/Control Room
  - Distributed Control Systems
  - Programmable Logic Controllers (PLCs)
  - Alarm Management
  - Electrical Noise and Wiring Practices
  - Lightning Protection for Instruments
  - Analyzers
  - Solenoid Valves
  - Instrument Installation
- Material Considerations

The design of transmutation processing facilities is, to a great extent, analogous to the design of other high-level waste processing facilities. Therefore, design considerations can be extracted from *Design Consideration*, [44]. Within *Design Considerations* [44], the discussion of design considerations for reprocessing facilities is presented. The typical reprocessing facility takes irradiated fissile fuel material or target material and extracts Uranium (UREX), Plutonium (PUREX), and other selected actinides and fission products.

Contamination control within a processing facility is essential for both criticality as well as build-up of contamination, resulting in radiological "hot-spots." The establishment of confinement zones provides both physical and operational boundaries for the migration of contamination within a facility. For transmuter fuel manufacturing, it can be envisioned that entire hot-cells may be required for the manufacturing process. *Design Considerations* [44] recommends the use of three confinement zones: a primary confinement zone, a secondary and a tertiary one.

Within the primary confinement, the process should be totally enclosed and provided with its own ventilation system. A secondary confinement system consists of the process cells and their ventilation system. The tertiary confinement system would be the building structure and its ventilation system. The installation, maintenance, and ultimate removal of equipment into and out of the primary or secondary confinement zones should be designed to minimize the spread of contamination. Figure 2.1 shows the use of master slave manipulators (MSM) penetrating the primary confinement barrier. In the case of process equipment failures, such as the MSMs, the confinement system should not fail, and the impact on the performance of the secondary confinement system should be minimal.

MSMs are remotely controlled mechanical devices designed to penetrate shield walls so that nuclear materials can be handled and manipulated safely. Manipulator operations are remote and are performed at a distance. In other words, the operator stands in the operating gallery behind a shielding wall, looking into the cell through a heavily shielded viewing window. A typical operator view through a shield window at the MSM end-effector can be seen in Figure 2.2. Control inputs by the operator on the master-side of

the MSM are mechanically transferred to the slave end-effector through cable, tapes, and electrical devices. In addition to manual movements, MSMs typically have electrical indexing motions. Manual motions allow the operator to duplicate the motions of the human hand, but are limited in the amount of movement throughout the hot cell. Electrical indexing provides the slave end-effector with greater motion and reach. The slave end-effector becomes a natural extension of the operators hands and arms into the hot-cell environment and is capable of duplicating all of the master's inputs.

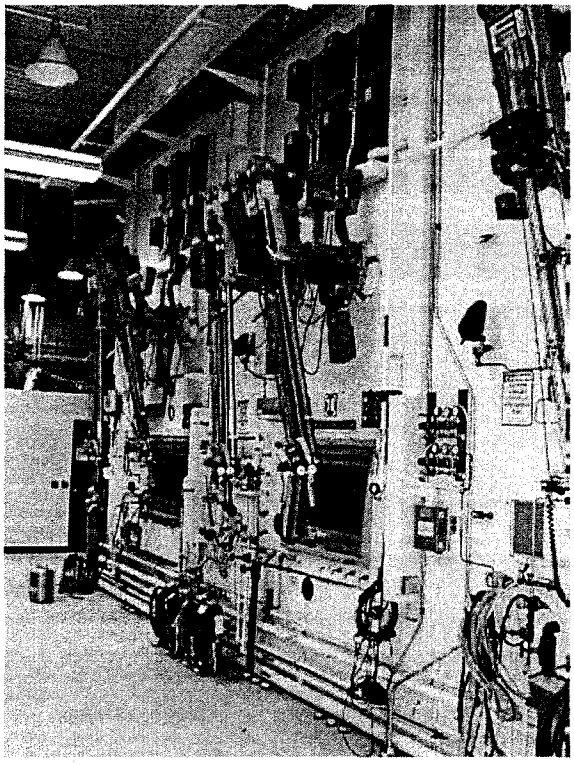


Figure 2.1 Typical MSM Installation

A close view of the MSM in Figure 2.3 reveals an actual case where a failure in the contamination barrier, known as the boot, had the possibility of compromising the

primary barrier. Here an operator inadvertently moved the MSM too close to the work cell light. The heat from the lamp melted and burned the contamination barrier on the MSM. While this contamination barrier does not directly fail the primary confinement zone, it does have an impact of the secondary confinement system when the MSM is to be replaced or repaired as the internal components of the MSM are now contaminated.

Figure 2.2 shows the use of a MSM without the use of contamination control. In this case, the expected contamination levels were extremely low, but had very high radiation levels. Therefore, an analysis of the operating environment (i.e., contamination and radiation) would specify the use of confinement zones and other contamination control features.

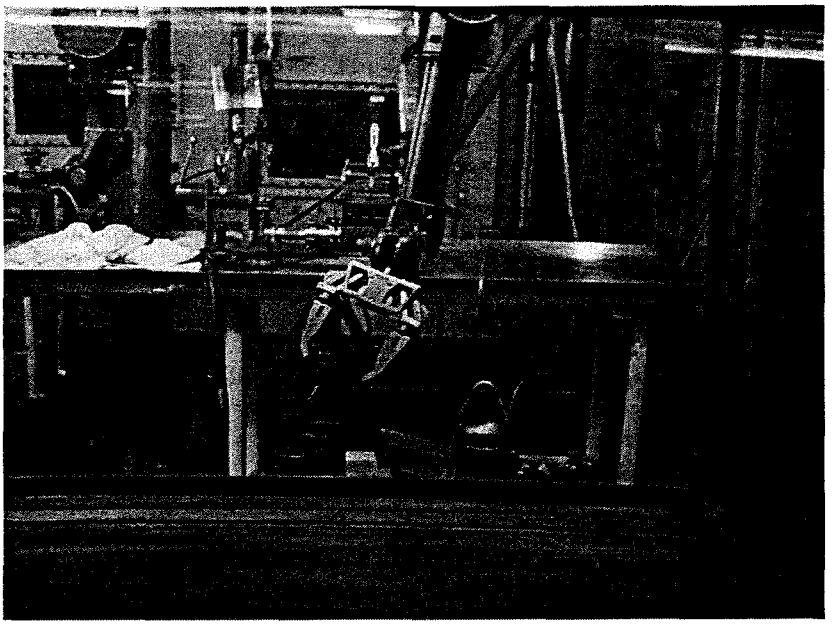


Figure 2.2 Through Hot-Cell Window View of MSM End-Effector

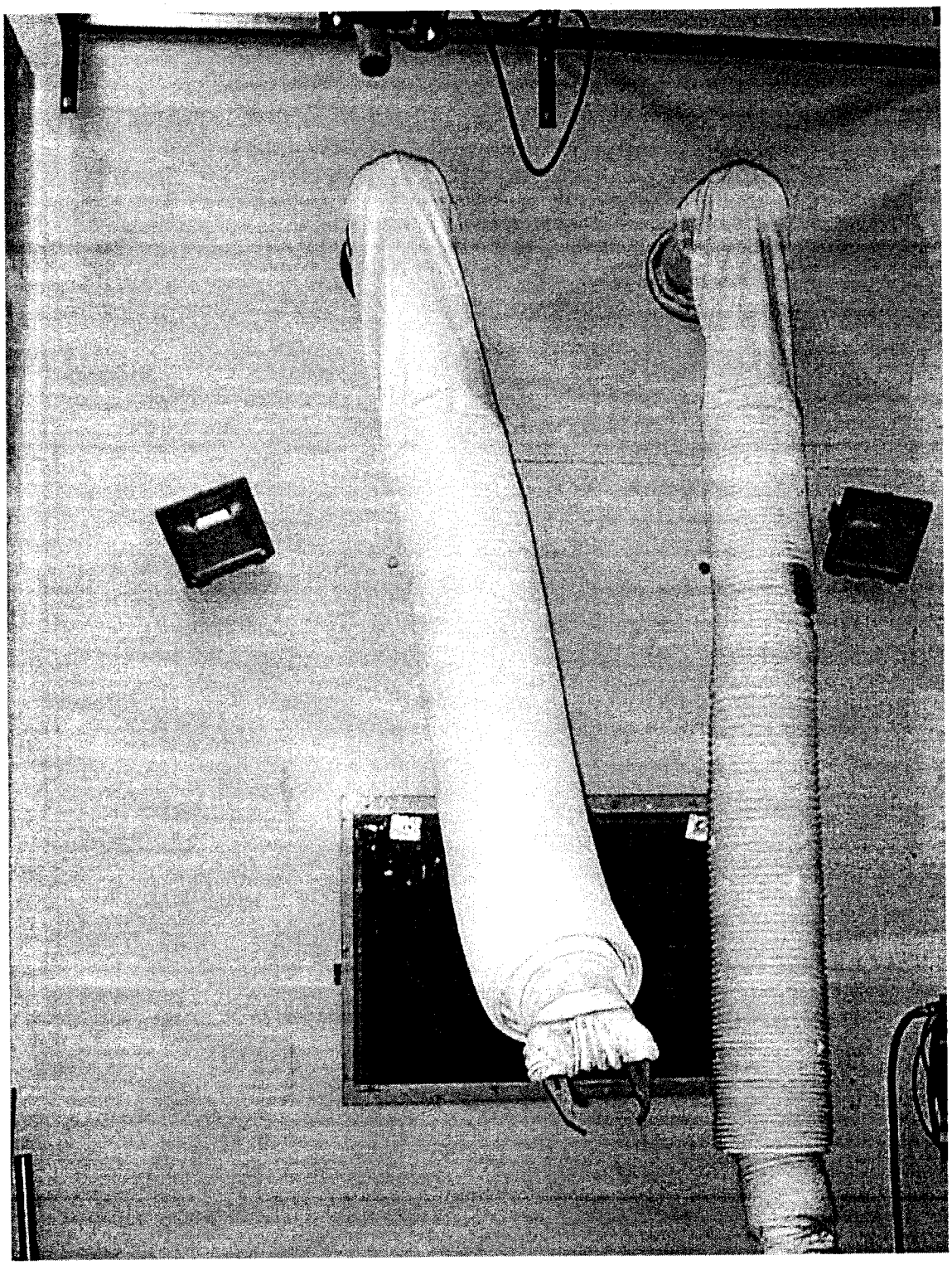


Figure 2.3 Contamination Control of MSM

2.1.2 Secure Automated Fabrication Program

In 1980, the DOE subcontracted the design, building, and installation of a remotely operated breeder reactor fuel pin fabrication line. The equipment was to be installed in the Fuels and Materials Examination Facility (FMEF), which at the time, was being constructed at the Hanford site near Richland, Washington [5]. This automated fuel pin fabrication line was referred to as the Secure Automated Fabrication (SAF) program and had two primary missions: 1) to provide a supply of MOX fuel pins to the U.S. Breeder Reactor Program; and 2) to develop and demonstrate advanced plutonium fuel fabrication technology. Although construction began on the SAF program, DOE canceled the U.S. Breeder Reactor Program and thus the final construction of the FMEF and SAF program. Regardless, the SAF program provides the fuel fabrication community with a valuable resource in design effort and lessons learned.

Similar to the needs of transmutation fuel manufacturing, the original need for the SAF program arose from numerous design objectives, such as:

- Reduction in radiation exposure to workers,
- Enhanced safeguards by restricting operator access to fissile materials and near real-time accountability,
- Improved product quality, and
- Increased productivity.

Numerous articles were written to describe the various aspects of the SAF program, from the powder operations [7] and handling [14], sintering furnace [8] and chemistry analysis [5], sintering boat transport [9], pellet gauging [10], to a general overview of the remote fabrication processes [11].

In the era of the early 80's, the SAF program incorporated advanced automation techniques and processes. The program was intended to remotely manufacture

Plutonium-based fuel, utilize automated processes, including 24 robots, and perform complex operations. The automated equipment was to be controlled remotely from a centrally-located supervisory control and data acquisition system, and alternatively, from local control consoles dedicated to individual systems [11].

The SAF line had a designed annual throughput of 6 MT of MOX fuel containing up to 20 percent per wt of Pu^{240} or higher [11]. MOX fuel is made with a mixture of Uranium and Plutonium Oxides.

2.1.3 Nuclear Weapons Complex – Pantex Plant

The Nuclear Weapons Complex at the Manson & Hanger Pantex Plant in Amarillo, Texas handles and stores thousands of radioactive nuclear materials. These materials, stored what is referred to as pits, were handled manually, which resulted in a significant accumulation of radiation dose to the workers at the plant. The handling operations included unpacking the pits from their shipping or storage containers, disassembling shipping fixtures that hold the pits, moving bare pits to a variety of measurement and inspection stations, re-assembling the shipping fixtures, and repackaging the containers for storage or shipment [37], and [50].

To ensure stockpile integrity, an automated weight and leak check system (WALS) was developed. By automating the process, radiation doses to the worker were significantly reduced by eliminating all direct human contact and operations with the pits [50]. The WALS system employed a six-axis robot Fanuc Model S-700 mounted on a linear track to move pits within the automation work cell [34]. By utilizing the automated process, once the operational procedure has been programmed and verified into the robotic system, all process steps are done with high assurance of completion and completed with the use of qualified and validated procedures [50]. Procedural compliance

is strictly enforced within a nuclear facility through an established quality control program, with many audits, self-assessments, and compliance checks. By automating the system, compliance with procedure is guaranteed and logged automatically for future inspections and audits.

2.1.4 PUREX Production Plant Environmental Remediation

The plutonium extraction facility, known as the PUREX Production Plant, is located at the DOE Hanford Site in southeastern Washington State. The PUREX facility became operational in the 1950's and underwent several upgrades and expansions. However, in the early 1990's the decision was made to remediate the facility. As part of the remediation process, an early assessment of the PUREX facilities found two dozen irradiated fuel elements lying at the bottom of one of the canyon hot cells [27]. These spent fuel elements were found in various locations within the hot cell, and would need to be repositioned so that they may be grabbed and retrieved from the cell floor remotely.

Due to the complexity of this operation, Jaquish [27] modeled the PUREX production plant using an **i**nteractive **g**raphical **r**obot **i**nstruction **p**rogram (IGRIP). Through the use of the IGRIP simulation environment, Jaquish was able to plan, review, and verify various proposed remediation activities.

Jaquish [27] collected various drawings and photographs of certain areas of the PUREX production plant so that the IGRIP model would produce a virtual walk-through inside the facility. The IGRIP software was used in developing the primary equipment to retrieve and process the irradiated fuel pellets left within the facility. The simulations showed that critical interfaces and dimensions could be assessed and implemented in the design. Jaquish also used the IGRIP simulation to optimize the operations. Because there

was flexibility in the conduct of operations for most tasks, sequence scheduling was performed to find optimum tasks that could then be implemented into a procedure.

2.1.5 Multipurpose Canister Handing

As part of the Civilian Radioactive Waste Management System, the development of a conceptual design for Multipurpose Canister (MPC) handling was performed in a simulated environment. An MPC is a sealed metallic canister designed for storage, transportation, and ultimate disposal of spent nuclear fuel assemblies. Bennet and Stringer [25] utilized IGRIP to model MPC handling at the proposed Monitored Retrievable Storage Facility, the precursor to the Yucca Mountain Project.

A complete process flow for MPC handling was simulated to visualize the sequence of preparing an MPC, e.g., fuel loading, welding, and inspecting. The simulations helped Bennet and Stringer [25] develop equipment requirements and costs. The equipment generally consisted of a single industrial robot and/or a programmable crane per work cell. In addition, a Stewart Platform was modeled and simulated for use in the process operations. The simulation validated process time estimates for each work cell at 25- and 100-percent of maximum equipment operating speeds, all under fully automated process control.

Ultimately Bennet and Stringer [25] used the IGRIP simulation of the Monitored Retrievable Storage Facility to identify operations that could be automated with robotic machinery. The results of their simulations showed, in the simulated environment, how particular operations could be executed automatically, identified equipment requirements and operational characteristics for the automation, and determined potential process times for each automated operation.

2.2 Higher Education

Chen [40] explored the use of robotics for nuclear waste handling. Although the final application of his dissertation focused mainly on the controllability of a commercially available robotic system, he faced challenges in determining the forward and reverse kinematic analysis [40].

In 1989, Kruger [16] investigated the design and use of 3-D computerized model of a robotic arm. While simplistic in both the robot model fidelity and the number of degrees of freedom, the authors work did expand the use of a modeling environment.

In 1995, O'Donnell [28] explored the use of operator assisted control of a robotic manipulator. In the thesis entitled *Automated Robotics System for Nuclear Waste Handling (Hardware and Software)* [28], O'Donnell describes the system as consisting of a computer and accompanying software placed between an operator control station and a remote manipulator. The computer and associated software checks for collisions within the work cell and other objects defined a computer model. If no collisions are predicted, normal master/slave operation continues; otherwise, the computer takes control until the operator guides the manipulator into a safe position [28]. A majority of the author's work centered around data communications between the master and the slave, developing software code to communicate between diverse "off-the-shelf" components and their related software. While never achieving true automation, the author did show successes in real-work test cells. As the master received commands by an operator, the system would check for collisions based on a computer aided design (CAD) model of the environment. If the system predicted a collision with the environment, the system would take control of the slave by simply halting the movement (i.e., freezing the robot). When

the operator returns the master controller to a non-colliding position, control is returned to the operator [28].

2.3 On the International Front

International work in the nuclear industry is vast and extensive. The following subsections will provide a summary of the nuclear facilities that have implemented robotics and/or automation into their processes. While non-inclusive, the following subsections show a small cross-section in which robotics and automation have been implemented world wide; and thereby, implies the lack of automation implementation by the United States, particularly the DOE, as discussed in Section 2.1.

2.3.1 MELOX Plant (Marcoule, France)

France is probably the most advanced nation in the world in the effective deployment of nuclear energy and in the resolution of fuel-cycle matters. Over 75% of France's electricity is currently derived from nuclear fission, and all of its spent fuel is scheduled for reprocessing. This is still less than the U.S. installed nuclear capacity. A government-owned company, *Compagnie General des Matieres Nucleaires* (COGEMA), is responsible for the nation's fuel reprocessing activities. For many decades, COGEMA has reprocessed spent light water reactor (LWR) fuel at plants in La Hague on the Brittany coast. These plants employ the PUREX process, have extensive underwater facilities for storing fuel awaiting reprocessing, and use remote operation and maintenance practices extensively. Fuel under contract for reprocessing there includes not only all spent fuel from French reactors but much fuel from Germany, Japan, and other European nations. These plants were designed from prompt reprocessing, but they

are actually reprocessing fuel that has been cooled for several years due to logistic factors, see [35].

The COGEMA MELOX plant, brought on line in 1995, is one of the most efficient and modern MOX fuel fabrication plants, where MOX fuel is a blended mixture of plutonium oxide and depleted uranium oxide. The automated fuel fabrication design increases throughput and enables the facility to produce fuel assemblies at a rate well above one 500 kg assembly per day, which is approximately 264 fuel rods and 100,000 MOX fuel pellets, see [38] and [55]. The MELOX plant estimated throughput is approximately 160 metric tons of heavy metal per year. Pending approval, design and process improvements may take the facilities capacity to over 250 metric tones of heavy metal per year [38].

2.3.2 La Hague Reprocessing Plant

The mission of the French company's COGEMA - La Hague plant, which entered service in 1966, is to reprocess spent nuclear fuel. The COGEMA La Hague industrial complex occupies a 740-acre site 15 ½ miles west of Cherbourg on the tip of the Cotentin peninsula.

The La Hague plant receives spent nuclear fuel from reactors for processing, which consists of separating and conditioning the various spent fuel components. The separation process recycles the uranium and plutonium and disposes of the non-reusable materials, where the majority of the spent fuel's radioactivity resides.

Since La Hague entered into service, the operational philosophy and design has evolved where they have implemented numerous processes and design improvements. While exact design improvements are proprietary and could not be obtained, generalized design philosophies are available. An example of the design improvements is that the La

Hague facility has been able to reduce mean occupational exposures from 500 mrem/yr to 150 mrem/yr from 1976 to 1986, even though the amount of reprocessed spent fuel has increased [41]. The philosophy behind some of their major design improvements leading to the reduced occupational exposures was obtained by utilizing a design-for-maintainability approach, see [41], where:

1. The utilization and implementation of standardized equipment specifically designed for hot-cell and other high radiation nuclear environments that is capable of remote dismantling and maintenance
2. The SSCs requiring replacement and/or refurbishment are modularized, which facilitates the removal and replacement of the SSCs.
3. Locating SSCs requiring frequent replacement in strategic locations, which also facilitates their removal and replacement.
4. The use of a "cold" test facility to prototype, test, debug, and enhance the design of SSCs prior to their use in the "hot" environment.

Using these design improvements and philosophies, COGEMA has been able to build highly automated, remotely operated facilities, which shield site personnel from radiation doses. The result of implementing these design philosophies is extremely low dose rates to COGEMA-La Hague personnel since the early 1990s.

For example, COGEMA designed a spent fuel disassembly and consolidation system (called DEC) for the COGEMA-La Hague plant. This design permitted the disassembly of spent fuel assemblies and subsequent consolidation of the spent fuel rods into densely packed canisters [41].

The DEC facility design utilized a single disassembly cell equipped with a remotely operated and maintained disassembly machine. The disassembly machine was designed to remotely cut the guide tubes from Pressurized Water Reactor (PWR) fuel assemblies and remove the top-end fitting. The disassembly machine would grip and pull all fuel rods simultaneously out of the PWR fuel assembly, and then place these rods into a

canister. After which, the disassembly machine would shear and compact the fuel assembly skeleton to maximize volume reduction. The disassembly machine design allowed adequate flexibility to handle all PWR fuel assembly types, which was accomplished by changing the fuel rod pulling grapple [41].

Using the same design-for-maintainability philosophy as described above, the DEC was automated and designed in modules so that it could be easily maintained and remotely operated and replaced. The entire consolidation system, including the disassembly machine, is controlled from a local control workstation outside of the hot-cell. Control panels and television monitoring systems provide remote control and viewing of all consolidation operations. However, the design included the capability of direct viewing through shielded windows as a backup [41].

2.4 THORP (Sellafield, United Kingdom)

The United Kingdom has for decades reprocessed spent fuel at facilities in northwestern England on the Irish Sea near Sellafield. In the early 1980s, design was initiated for construction of a new Thermal Oxide Reprocessing Plant (THORP) for reprocessing 1,200 MTU/yr of LWR fuels. Start-up of reprocessing operations began in 1994. The THORP facility employs the basic PUREX process, has extensive pre-reprocessing storage for spent fuel, uses significant remote operations and maintenance practices, and was designed for prompt reprocessing.

2.5 The Japanese Experience

Japan has also performed small-scale reprocessing of LWR fuels for many years at its Tokai-Mura plant. A large 1,200 MTU/yr LWR reprocessing plant is under construction on the coast in northern Japan by a Japanese consortium of nuclear industry

organizations. This new plant has received considerable design assistance from France, and much of the technology employed in La Hague is being incorporated into the Japanese plant. This plant also employs basic PUREX process. It is scheduled for start-up in 1998, see [35].

2.6 Atomic Energy of Canada Limited

In 1995, the TELBOT manipulator was used as a remote handling system for the automated cleaning of the primary side of the steam generator tubes at Pont Lepreau NGS in New Brunswick, Canada. The TELBOT, shown in Figure 2.5, was used to handle and position cleaning equipment in the steam generator bowl. The TELBOT was simulated, tested and deployed by Atomic Energy of Canada Limited (AECL). The robot was programmed to run 24 hours a day over a period of 23 days, and successfully cleaned 8,209 tubes.

In addition in 1996, the TELBOT manipulator was used to transport and position cleaning equipment for the decontamination of a particular hot cell at AECL's Chalk River nuclear laboratory. The hot cell had relatively high dose rates and contamination levels, which prevented extended human operations within the cell; and therefore, the TELBOT was selected for use in the decontamination activities. Figure 2.6 shows the preparation of the TELBOT for deployment within the hot cell. Workers are "bagging" the robot by wrapping the manipulator in a plastic shroud to provide contamination control. After the TELBOT activities are completed, any contamination is affixed to the plastic and subsequently removed, leaving a relatively contamination free robot. AECL produced computer simulations of the robot working in the environment. An iterative process of simulation and design refinement helped perfect the operations and robot

movements prior to start of the decontamination process. Figure 2.4 shows a mock-up of the TELBOT manipulator for real-time simulation of non-destructive testing of steam generator tubes.

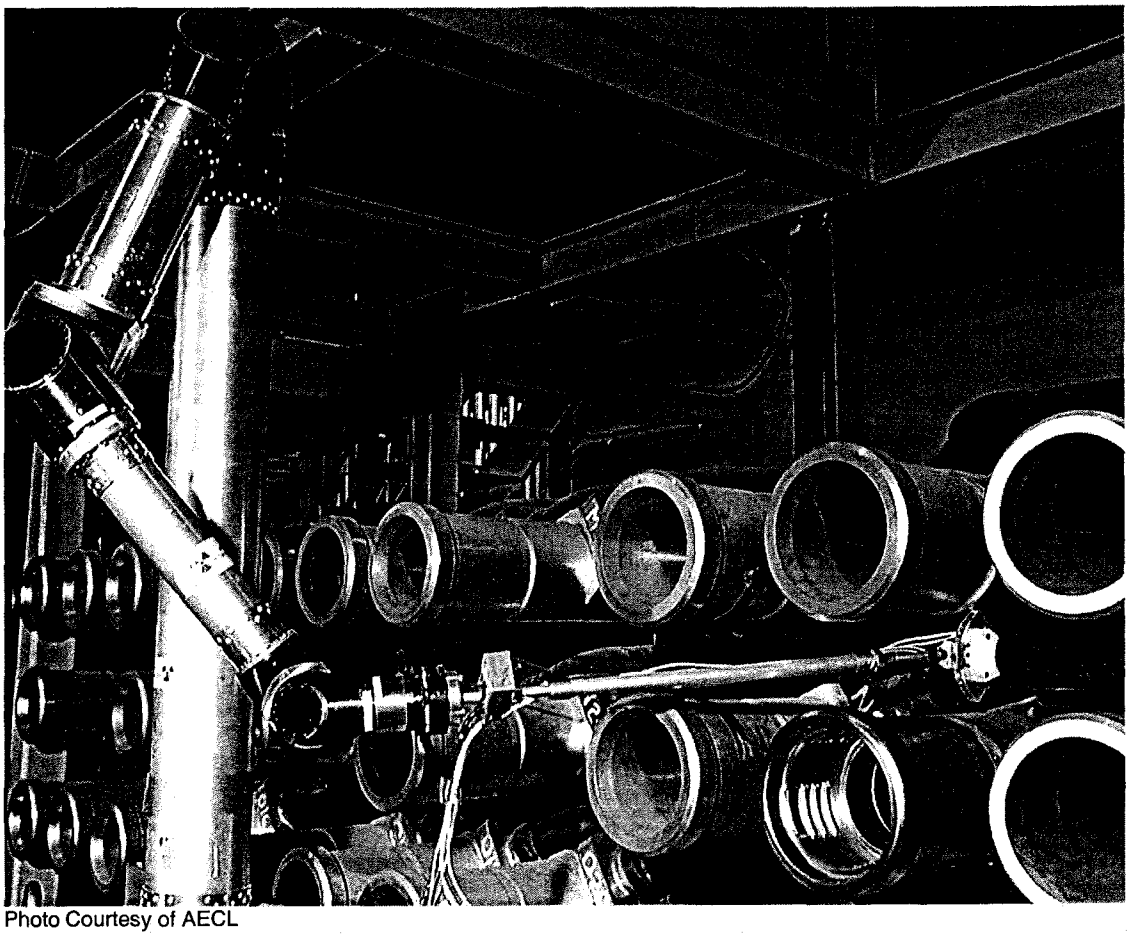


Photo Courtesy of AECL

Figure 2.4 TELBOT Used in Mock-up for Ultrasonic Pipe Inspection

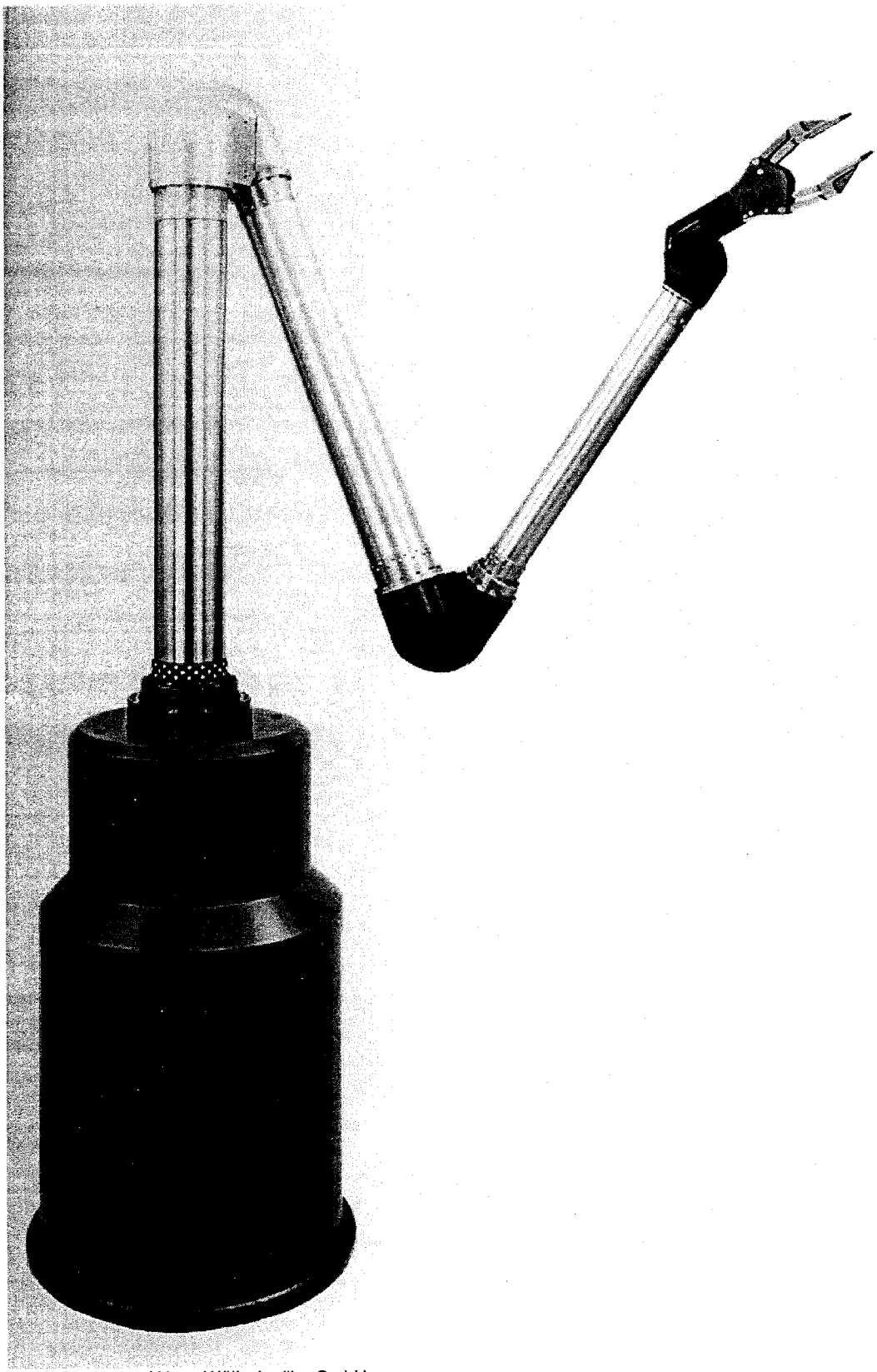


Photo Courtesy of Hans Wälischmiller GmbH

Figure 2.5 TELBOT TB100

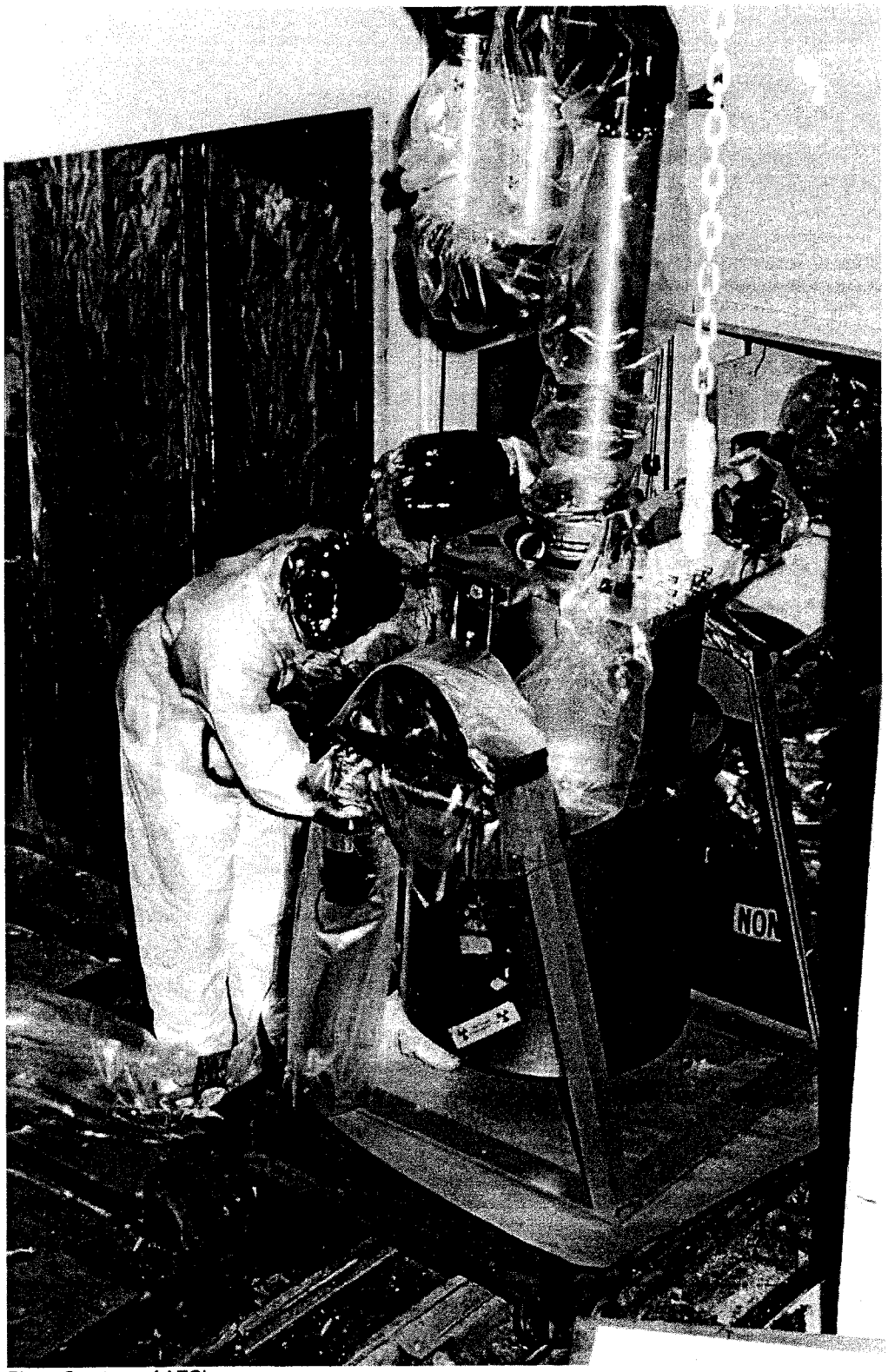


Photo Courtesy of AECL

Figure 2.6 Hot Cell Preparation of the TELBOT TB100

CHAPTER 3

ROBOTIC MODELING

3.1 TELBOT Manipulator System Overview

A general-purpose industrial robot, trade named TELBOT, has been developed by Hans Wälischmiller GmbH, as shown in Figure 2.5. While relatively new to the nuclear industry, TELBOT has been used in high radiation environments for cleaning steam generator tubes at the primary side in Canada. Also, the robot has been used for decommissioning glove boxes in Japan [33]. Figure 3.1 shows a line of commercial hot cell robots (Wäelischmiller GmbH).

The design of the TELBOT manipulator is specifically intended for hazardous and contaminated environments, while still maintaining a lightweight structure. For each of the six rotational joints and the operation of the end-effector, a respective drive motor resides in the robot base, as shown in Figure 3.2. Torque for each joint axis is translated from the drive motor, through transmission gears and safety clutches within the base, and through concentric torque tubes and bevel gears to each joint. By co-locating all of the motors, gears, sensors, electrical wiring, and related equipment in the robot base, the arms only contain mechanical linkages and are free to rotate a full 360 degrees, as well as, providing contamination sealing at each of the joints. This becomes important in very dirty or highly contaminated environments [33], and [47].

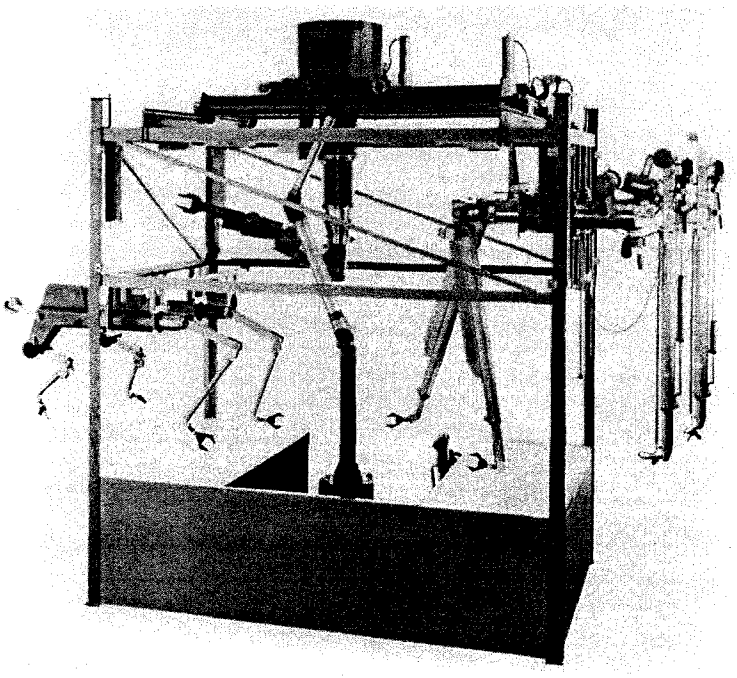


Figure 3.1 Hot Cell Robots (Wälischmiller GmbH)

The fundamental design of the TELBOT skeleton is based on a modular approach. Each joint, member, and its related bevel gear ratios can be interchanged. This allows for the manipulator to be configured for various working envelopes, payloads and transmission ratios. The modular design also allows for quick interchange of parts and members [33]. The TELBOT configuration has several advantages at the cost of some dynamic anomalies. The design of the robot base, with its corresponding drive motors, gearing, sensors, and wiring, allows the manipulator arms to pass through a shield wall port, much like a traditional MSM, while keeping the sensitive motors and sensors out of the high radiation environment.

This configuration also allows the manipulator members to be manufactured from lightweight materials and tube structures. However, due to this construction and configuration, each joint axis exhibits dynamic deflections and elasticity.

In reality, the TELBOT members act as torsional springs and need to be modeled in such a manner. This work assumes that for light loads (e.g., fuel pins), the defection of each concentric torque tube is negligible and can be considered a rigid structure, as will be further justified in the following section (Section 3.2).

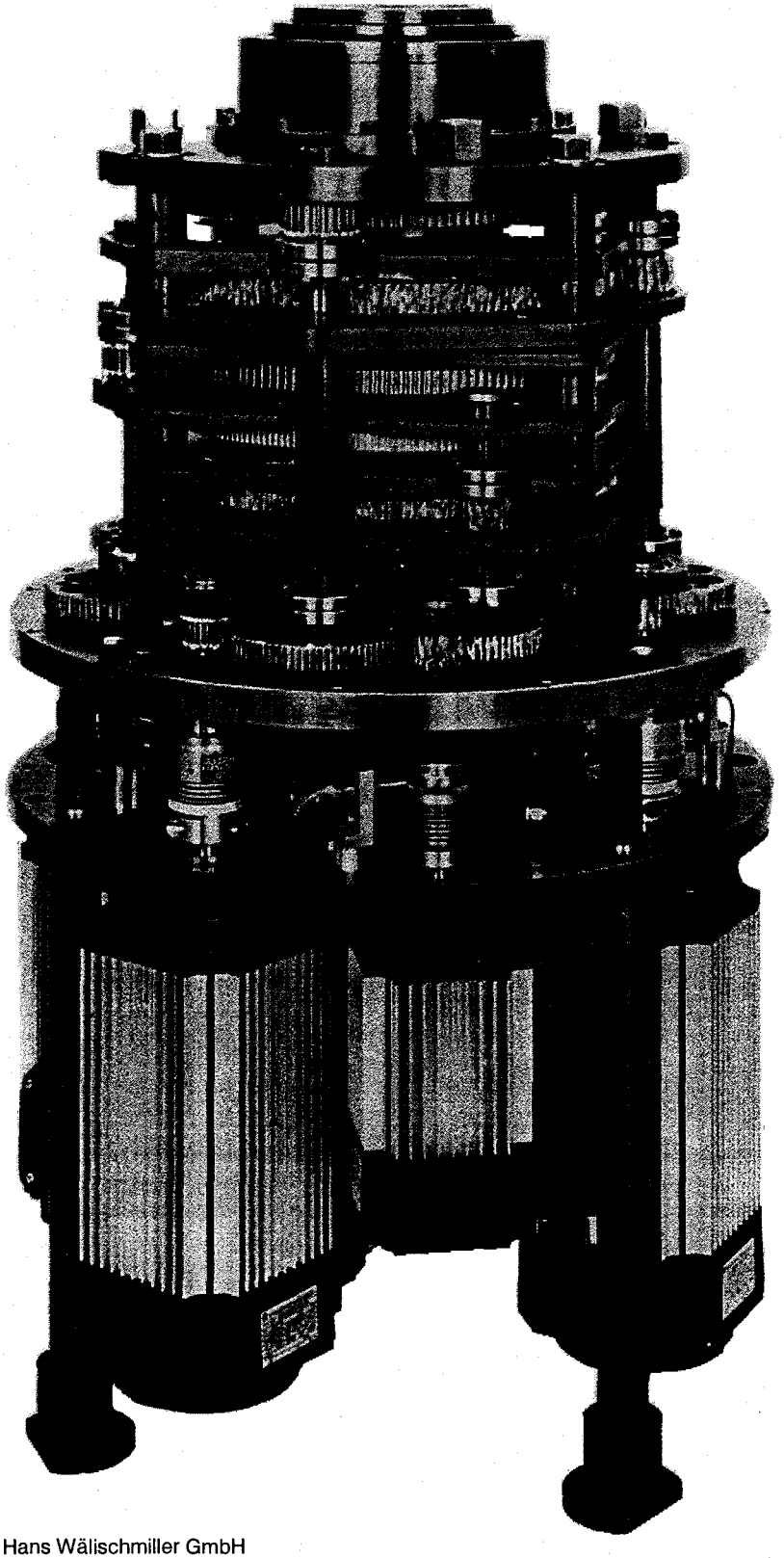


Photo Courtesy of Hans Wälischmiller GmbH

Figure 3.2 Drive Unit for the TELBOT TB100

3.2 Computer Aided Design of TELBOT System

Schlotter & Pfeiffer [47] explored modeling and control of the TELBOT and its above mentioned dynamic effects related to torsional stiffness. They developed an elastic multi-body system model where each of the torque tubes driving the six rotational joints and the gripper were assumed as torsion springs, each with a specific stiffness and damping.

To verify the Schlotter & Pfeiffer's mathematical model of the TELBOT, a CAD model was developed by the Schlotter & Pfeiffer for the robotic system. From the CAD model, the authors obtained the robots inertial parameters. Performing a Finite Element Analysis on the CAD model, they derived the stiffness of the robot and related torque tubes. The friction values for the joints were obtained from actuarial data. Through simulation, the authors were able to determine the elastic deformation of the trunk and forearm, anatomically speaking, which would typically experience the most deflection and have the most detrimental effect on the positional accuracy of the end-effector. Based on the figures generated by Schlotter & Pfeiffer, [47], their maximum deflection was approximately 0.06 mm in the trunk and 0.03 for the forearm.

It is intuitive that the deflection calculated is a function of payload, robot configuration, joint accelerations, and other dynamical effects. However, based on their relatively small deflection results, any effects due to dynamic deflection on the simulation of the TELBOT within this scope of work is assumed to be negligible.

3.3 Mathematical Modeling of TELBOT

The mathematical modeling of the TELBOT follows the methodology established by Fu et al. [36], and Craig [18]. Robotic manipulators are customarily modeled using the

Denavit-Hartenberg formulation as described in [1], [18], and [36], in which manipulator links are modeled as a series of successive spatial rotations and translations, see also [52].

3.4 Forward Kinematic

Forward kinematics mathematically represents the robot geometry in space. Of particular concern is the location and orientation of the robot end-effector with respect to a particular coordinate system (e.g., pose), typically at the robot base or a central work-cell coordinate system. The position of the end-effector of the TELBOT manipulator, a six degree of freedom system, is translated from the robot base by six rotational components, which allow the TELBOT manipulator to reach any arbitrary end-effector pose within the robot's workspace. A conversion from the various robot joints to the end-effector location and orientation is the root of forward kinematics, where the end-effector pose is mathematically translated from joint-variables space to Cartesian-variable space.

Nof [45] provides some definition of the robot workspace, where the workspace of a particular manipulator is defined as the set of all end-effector locations (poses) that can be reached by arbitrary choices of joint variables. If the complete end-effector pose (i.e., both end effector position and orientation) is considered, then the workspace is classified as the *complete workspace*. In some instances, the complete end-effector pose is not required for discussion and only the position is of concern, which gives the *reachable workspace*. The subset of the reachable workspace that can be attained with arbitrary orientations of the end-effector is the *dexterous workspace*. We will be discussing the complete workspace within this work.

Within the robots complete workspace, one could imagine a homogenous coordinate system located at the base of the robot and a second homogenous coordinate system located at the end-effector. One could then draw a vector ($\vec{r}$) from the origin of the base

coordinate frame to the origin of the end-effector coordinate frame. It would be beneficial to be able to transform this vector, known as the position vector ($\hat{\mathbf{p}}_{xyz}$) from one coordinate system representation to another, such as the base coordinate to the end-effector coordinate system. In fact, this position vector ($\hat{\mathbf{p}}_{xyz}$) can be transformed from one coordinate system, both rotationally and by translation, to an alternate coordinate system by the basic homogenous transformation matrix (**T**), resulting a new position vector ($\hat{\mathbf{p}}_{uvw}$). This is shown in Eq. (3.1):

$$\hat{p}_{uvw} = T\,\hat{p}_{xyz} \tag{3.1}$$

As Fu et al. reveals, Denavit and Hartenberg developed a mathematical tool in the form of a matrix, known as the D-H Table, that would systematically establish a body-attached coordinate system frame to each link of an articulated chain, e.g., a series of robotic linkages, see [36]. These body-attached coordinates are shown in Figure 3.3 for the modeled TELBOT manipulator.

The matrix is a 4×4 homogenous transformation matrix representing each link's coordinate system at the joint with respect the previous link's coordinate system [36], [17]. Although the standard D-H Table, described by Denavit and Hartenberg in [1], depends on two translations and two rotations to define the linkages, the standard method assigns frame (i) to joint (i+1) while the modified method described by Craig in [18] assigns frame (i) to joint (i). The resulting homogenous transformation matrix is different. In this work, the modified D-H Table will be used. The rules for determining the D-H Table are summarized in [18]:

θ_i = the angle between $\mathbf{x}_{i-1}$ and $\mathbf{x}_i$ measured about $\mathbf{z}_i$

d_i = the distance from $\mathbf{x}_{i-1}$ to $\mathbf{x}_i$ measured along $\mathbf{z}_i$

a_i = the distance from $\mathbf{z}_i$ to $\mathbf{z}_{i+1}$ measured along $\mathbf{x}_i$

α_i = the angle between $\mathbf{z}_i$ and $\mathbf{z}_{i+1}$ measured about $\mathbf{x}_i$

The base frame ($x_0y_0z_0$), see Figure 3.3, is the world coordinate frame, placed at the bottom of the base of the TELBOT manipulator. Everything in the work cell will be referenced to this frame. The last frame ($x_7y_7z_7$), see Figure 3.3, is attached to the end-effector of the manipulator.

If the joint is revolute, θ_i will be the variable, and if the joint is prismatic, then d_i will be the variable. We usually choose $a_i>0$ since it corresponds to a distance, however α_i, d_i, and θ_i are signed quantities. The convention outlined above does not result in unique attachment of frames to the links.

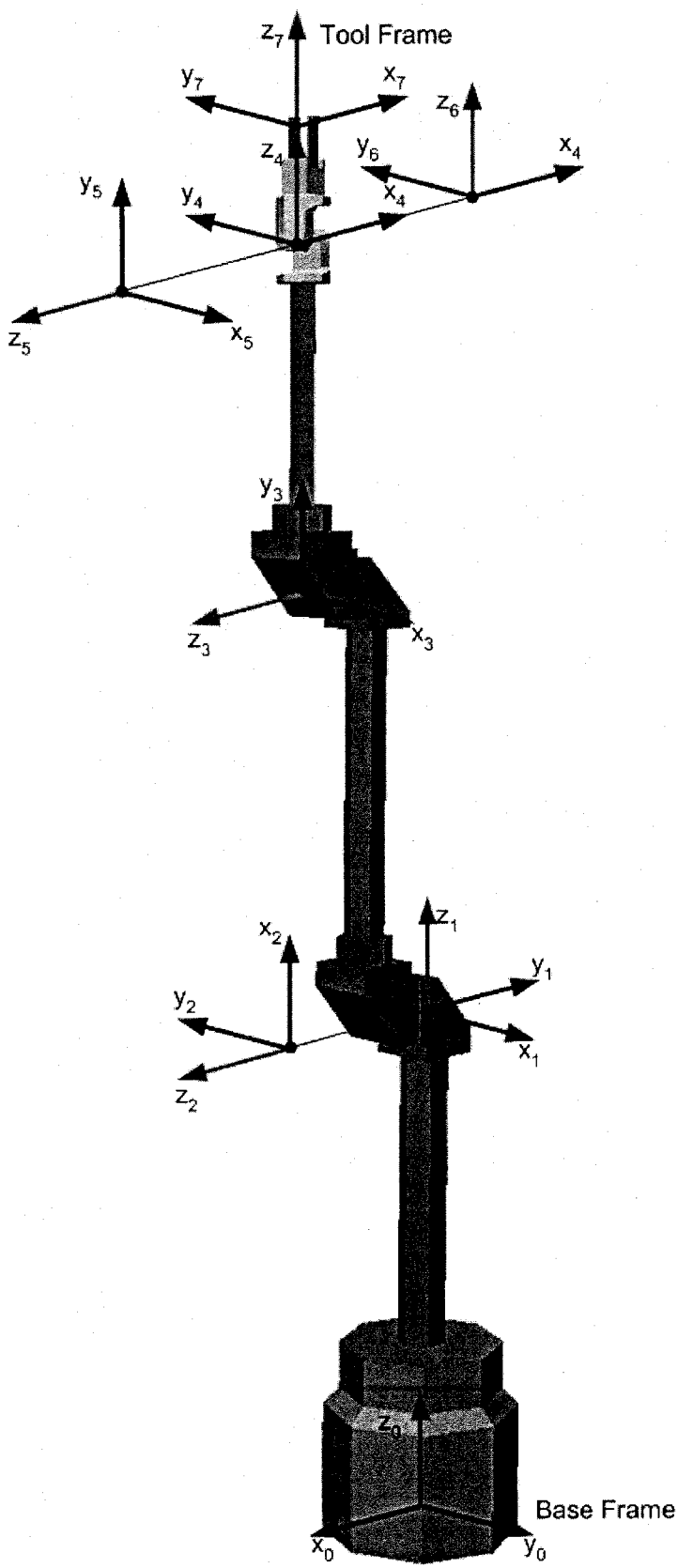


Figure 3.3 Coordinate Frame Assignment of the TELBOT Manipulator

Following this methodology and mapping each link's coordinate system, the D-H Table for the TELBOT manipulator is shown in Table 3.I.

Table 3.I D-H Table for the TELBOT

Joint *i*	θ_i (radians)	α_i (radians)	a_i (cm)	$\mathbf{d}_i$ (cm)	**Joint Range**
1	$\theta_1 + \pi/2$	0	0	168.91	360 deg / Full Rotation
2	$\theta_2 + \pi/2$	$\pi/2$	0	0	360 deg/ Full Rotation
3	$\theta_3 - \pi/2$	0	128.27	53.36	360 deg/ Full Rotation
4	θ_4	$-\pi/2$	0	95.88	360 deg/ Full Rotation
5	θ_5	$\pi/2$	0	0	360 deg/ Full Rotation
6	θ_6	$-\pi/2$	0	0	360 deg/ Full Rotation
7	0	0	0	32.07	No Joint

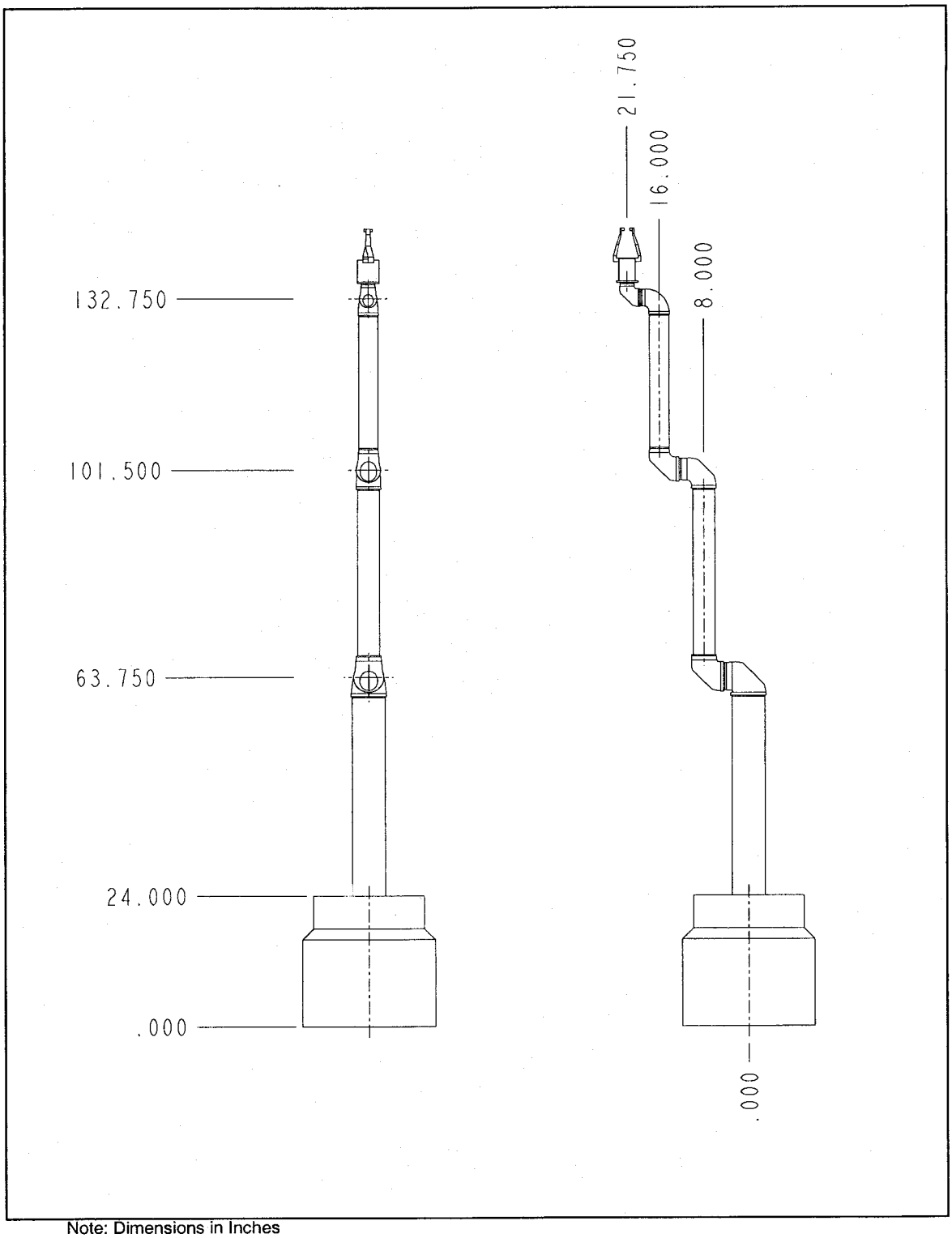
One might question the existence of the 7th entry in the table. This is a fictitious joint added so that we can have a Coordinate frame between the grippers of the end-effector. We will be eventually interested in the position and orientation of the 7th frame. As can be seen, this frame has the same orientation of Joint 6 frame, except that it is offset along the $\mathbf{z}_6$ axis.

As discussed in Section 3.1, the TELBOT utilizes all rotary joints capable of full rotation. For a rotary joint, d_i , a_i , *and* α_i are the link and joint parameters, which remain constant for a robot, while θ_i is the joint variable that changes when link i rotates with respect to link i-1.

The inherent nature of the TELBOT design allows for each link to be of a variable length, specified by the customer. While an "off-the-shelf" design for the TELBOT may exist, specific dimensions are assumed for in this work. Figure 3.4 shows the overall assumed size of the TELBOT. Figure 2.6 shows an actual image of the TELBOT where workers are preparing for hot cell use, which provides a good understanding of the robot size. While the dimensions for the TELBOT used in this work are not factory-specified, Figure 2.6 provides a reasonable basis that the dimensions used are consistent with actual equipment.

Once the D-H Table has been established for each link, a homogeneous transformation matrix (**T**) is developed relating the i^{th} coordinate frame to the $(i\text{-}1)^{th}$ coordinate frame, as shown in Eq. (3.2):

$$ {}^{i-1}_{i}T = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & a_{i-1} \\ \sin(\theta_i)\cos(\alpha_{i-1}) & \cos(\theta_i)\cos(\alpha_{i-1}) & -\sin(\alpha_{i-1}) & -d_i\sin(\alpha_{i-1}) \\ \sin(\theta_i)\sin(\alpha_{i-1}) & \cos(\theta_i)\sin(\alpha_{i-1}) & \cos(\alpha_{i-1}) & d_i\cos(\alpha_{i-1}) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.2} $$



Note: Dimensions in Inches

Figure 3.4 Assumed TELBOT Dimensions

3.5 Inverse Kinematics

Inverse kinematics is used to find the joint angles that will place the end-effector at a desired location in the Cartesian-variable space. The forward kinematic analysis of the TELBOT manipulator system is straightforward. However, the inverse kinematic solution proved to be considerably more difficult. The six-revolute (6R) robot utilizes unique joints that are able to rotate over 360 degrees allowing the end effectors to attain its work position at multiple orientations. In any manipulator, if the axes do not intersect at one point, see [1], [18] and [36], a closed form solution for the inverse kinematics cannot be found and one has to resort to numerical solutions.

Most industrial manipulators utilize a spherical wrist, where the wrist-joint axes intersect at one point. For a 6R robot, the wrist is positioned using the first three joints (dexterous workspace), and the reachable workspace is visualized as a sphere at the end of the wrist joint, see [45].

Lee and Wälischmiller [33], Nof [45], Paul [6], and Goldenberg et al. [12] all showed that for a 6R system that met the Denavit-Hartenberg method (i.e., as is the case with most industrial manipulators), a closed-form solution can be obtained. There are up to 16 solutions of the inverse kinematics for manipulators with six revolute joints.

3.5.1 Inverse Kinematics Closed form Solution

The closed form solution of the inverse kinematics problem requires extensive matrix manipulations. Yet, obtaining a closed form avoids resorting to numerical methods which sometimes converges and often does not. Moreover, in the process of obtaining the closed form solution, the manipulator's singularities will be unveiled. Following Eq. (3.3), one can construct seven homogenous transformations. The inverse kinematics problem can be formulated as follows: given the position and orientation of the

coordinate frame of the end effector, solve for the six joint angles that would correspond to such configuration.

$$ {}^{0}_{7}T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^{0}_{1}T(\theta_1)\,{}^{1}_{2}T(\theta_2)\,{}^{2}_{3}T(\theta_3)\,{}^{3}_{4}T(\theta_4)\,{}^{4}_{5}T(\theta_5)\,{}^{5}_{6}T(\theta_6)\,{}^{6}_{7}T \qquad (3.3) $$

So, mathematically speaking, the problem is: the values of r_{ij} (i=1...3, j=1...3) and $p_{x,\,y,\,z}$ are known, solve for the values of $\theta_1,\cdots,\theta_6$.

A restatement of Eq. (3.4) which puts the dependence on θ_1 on the left hand side of the equation is:

$$ \left[{}^{0}_{1}T(\theta_1)\right]^{-1}\,{}^{0}_{6}T = {}^{1}_{2}T(\theta_2)\,{}^{2}_{3}T(\theta_3)\,{}^{3}_{4}T(\theta_4)\,{}^{4}_{5}T(\theta_5)\,{}^{5}_{6}T(\theta_6)\,{}^{6}_{7}T \qquad (3.4) $$

This simple technique of multiplying each side of a transform equation by an inverse is often used to advantage in separating out variables in search of a solvable equation.

Equating the (2,4) elements from both sides of Eq. (3.5):

$$ (r_{23}\,d_7 - p_y)\cos(\theta_1) + (p_x - r_{13}\,d_7)\sin(\theta_1) = d_3 \qquad (3.5) $$

Consequently, θ_1 will have two solutions:

$$ \theta_1 = \mathrm{Atan2}(r_{23}\,d_7 - p_y,\, p_x - r_{13}\,d_7) - \mathrm{Atan2}\left(d_3, \pm\sqrt{(p_x - r_{13}\,d_7)^2 + (r_{23}\,d_7 - p_y)^2 - d_3}\right) \qquad (3.6) $$

Now, equating the (1,4) and (3,4) elements of Eq. (3.7). Square both sides of the equations, then add them; the following is obtained:

$$ d_4^{\,2} - 2\,d_4\,a_2\sin(\theta_3) + a_2^{\,2} = A_3^{\,2} + B_3^{\,2} \qquad (3.7) $$

where

$$ A_3 = p_x\cos(\theta_1) + p_y\sin(\theta_1) - (r_{13}\cos(\theta_1) + r_{23}\sin(\theta_3)d_7) \qquad (3.8) $$

and

$$B_3 = p_z - d_1 - r_{33}\, d_7 \tag{3.9}$$

Equation (3.7) can be solved to determine θ_3 as follows:

$$\sin(\theta_3) = \frac{{d_4}^2 + {a_2}^2 - {A_3}^2 - {B_3}^2}{2\, d_4\, a_2} \tag{3.10}$$

then $\cos(\theta_3)$ can be obtained using the standard identity:

$$\cos(\theta_3) = \pm\sqrt{1 - \sin(\theta_3)^2} \tag{3.11}$$

Again, θ_3 will have two solutions:

$$\theta_3 = \text{Atan2}(\sin(\theta_3), \cos(\theta_3)) \tag{3.12}$$

Equation (3.3) can be further expressed as:

$$\left[{}^2_3T(\theta_3)\right]^{-1}\left[{}^1_2T(\theta_2)\right]^{-1}\left[{}^0_1T(\theta_1)\right]^{-1}\,{}^0_6T = {}^3_4T(\theta_4)\,{}^4_5T(\theta_5)\,{}^5_6T(\theta_6)\,{}^6_7T \tag{3.13}$$

Equating the (1,4) and (2,4) elements on both sides of Eq. (3.13) can be written as the following linear system having two unknowns: $\cos(\theta_2 + \theta_3)$ and $\sin(\theta_2 + \theta_3)$...

$$\begin{cases} A_{231}\cos(\theta_2 + \theta_3) + B_{231}\sin(\theta_2 + \theta_3) = a_2\cos(\theta_3) \\ A_{232}\cos(\theta_2 + \theta_3) + B_{232}\sin(\theta_2 + \theta_3) = d_4 - a_2\sin(\theta_3) \end{cases} \tag{3.14}$$

where

$$\begin{cases} A_{231} = (p_x - r_{13}d_7)\cos(\theta_1) + (p_y - r_{23}\,d_7)\sin(\theta_1) \\ B_{231} = (p_z - r_{33}\,d_7) - d_1 \\ A_{232} = B_{231} \\ B_{232} = -A_{231} \end{cases} \tag{3.15}$$

which yields to the solving of (θ_2+ θ_3):

$$(\theta_2 + \theta_3) = \text{Atan2}(\sin(\theta_2 + \theta_3), \cos(\theta_2 + \theta_3)) \tag{3.16}$$

and a solution can be determined for θ_2 as:

$$\theta_2 = (\theta_2 + \theta_3) - \theta_3 \tag{3.17}$$

Now, equate elements (1,3) and (3,3), and rearrange to obtain

$$\begin{cases} \cos(\theta_4) = \dfrac{(r_{13}\cos(\theta_1) + r_{23}\sin(\theta_1))\cos(\theta_2+\theta_3) + r_{33}\sin(\theta_2+\theta_3)}{\sin(\theta_5)} \\ \sin(\theta_4) = \dfrac{r_{13}\sin(\theta_1) - r_{23}\cos(\theta_1)}{\sin(\theta_5)} \end{cases} \tag{3.18}$$

Further multiplications by the inverse of homogenous transformations in Eq. (3.3) yields the following equation:

$$\left[{}^{3}_{4}T(\theta_4)\right]^{-1}\left[{}^{2}_{3}T(\theta_3)\right]^{-1}\left[{}^{1}_{2}T(\theta_2)\right]^{-1}\left[{}^{0}_{1}T(\theta_1)\right]^{-1}\,{}^{0}_{6}T = {}^{4}_{5}T(\theta_5)\,{}^{5}_{6}T(\theta_6)\,{}^{6}_{7}T \tag{3.19}$$

Equating elements (1,3) and (3,3) of Eq. (3.19) we obtain:

$$\sin(\theta_5) = A_{51}\,r_{11} - A_{52}\,r_{21} - \sin(\theta_4)\sin(\theta_2+\theta_3)r_{31} \tag{3.20}$$

where

$$\begin{aligned} A_{51} &= \sin(\theta_1)\sin(\theta_4) - \cos(\theta 1)\cos(\theta_4)\cos(\theta_2+\theta_3) \\ A_{52} &= \cos(\theta_1)\sin(\theta_4) + \sin(\theta_1)\cos(\theta_4)\cos(\theta_2+\theta_3) \end{aligned} \tag{3.21}$$

and

$$\cos(\theta_5) = -(r_{13}\cos(\theta_3) + r_{23}\sin(\theta_3))\sin(\theta_2+\theta_3) - \cos(\theta_2+\theta_3)r_{33} \tag{3.22}$$

Consequently

$$\theta_5 = \text{Atan2}(\sin(\theta_5), \cos(\theta_5)) \tag{3.23}$$

Now, to solve for the 6th joint angle, equate elements (2,1) and (2,2) of Eq. (3.19):

$$\begin{cases} \cos(\theta_6) = -A_{61}\,r_{11} + A_{62}\,r_{21} - \sin(\theta_4)\sin(\theta_2+\theta_3)r_{31} \\ \sin(\theta_6) = -A_{61}\,r_{12} + A_{62}\,r_{22} - \sin(\theta_4)\sin(\theta_2+\theta_3)r_{32} \end{cases} \tag{3.24}$$

where

$$\begin{aligned} A_{61} &= (\sin(\theta_1)\cos(\theta_4) + \cos(\theta_1)\sin(\theta_4)\sin(\theta_2+\theta_3)) \\ A_{62} &= (\cos(\theta_1)\cos(\theta_4) - \sin(\theta_1)\sin(\theta_4)\cos(\theta_2+\theta_3)) \end{aligned} \tag{3.25}$$

and hence the 6th joint can be obtained by:

$$\theta_6 = \text{Atan2}(\sin(\theta_6), \cos(\theta_6)) \tag{3.26}$$

3.5.2 Manipulator Singularity

If the angle of the 5th joint is zero or close to zero, then the solution for the 4th joint in Eq. (3.18)degenerates. Otherwise, Eq. (3.27) can yield the following solution:

$$\theta_4 = \text{Atan2}(\sin(\theta_4)\sin(\theta_5), \cos(\theta_5)\sin(\theta_5)) \tag{3.27}$$

To check for the singularity, one should check if the value of both arguments of the Atan2 function in Eq. (3.27) are zero or close to zero, then this joint's angle should be fixed to some nominal value (e.g. its current value). The 5th and 6th joints will be solved accordingly.

3.5.3 Number of Solutions

Because of the plus-or-minus appearing in Eq. (3.6) and Eq. (3.11), these equations compute four solutions. Additionally, there are four more solutions obtained by "flipping" the wrist of the manipulator. For each of the four solutions computed above, the flipped solution can be expressed as:

$$\begin{aligned} \theta_4' &= \theta_4 + 180^\circ \\ \theta_5' &= -\theta_5 \\ \theta_6' &= \theta_6 + 180^\circ \end{aligned} \tag{3.28}$$

After all eight solutions have been computed, some or all of them may have to be discarded because of joint limit violations. Of the remaining valid solutions, usually the one closest to the present manipulator configuration is chosen.

3.6 Trajectory Planning

For the most part, the motion of a manipulator is the motion of the tool frame (Frame 7 in Figure 3.3) relative to the base frame (Frame 0 in Figure 3.3). This is the same

manner in which an eventual user of the system would think, and designing path description and generation system in these term will result in a few important advantages.

When specifying paths as motions of the tool frame relative to the base frame, we decouple the motion description from any particular robot, end effector, or work-pieces. This results in a certain modularity, and would allow the same path description to be used with a different manipulator, or with the same manipulator with a different tool size. As shown in Figure 3.5, the basic problem is to move the manipulator from an initial position to some desired final position. That is, we wish to move the tool frame from its current value, $\{T_{Initial}\}$, to a desired final value, $\{T_{Final}\}$. Note that this motion in general involves a change in orientation as well as a change in position of the tool relative to the station.

Sometimes it is necessary to specify the motion in much more detail than simply stating the desired final configuration. One way to include more detail in a path description is to give a sequence of desired via points or intermediate points between the initial and final positions. Thus, in completing the motion, the tool frame must pass through a set of intermediate positions and orientations as described by the via points. Each of these via points is actually a frame which specifies both the position and orientation of the tool relative to the station. The name path points include all the via points plus the initial and final points. Remember that although we generally used the term "points," these are actually frames which give both position and orientation. Along with these spatial constraints on the motion, the user may also wish to specify temporal attributes of the motion. For example, the time elapsed between via points might be specified in the description of the path.

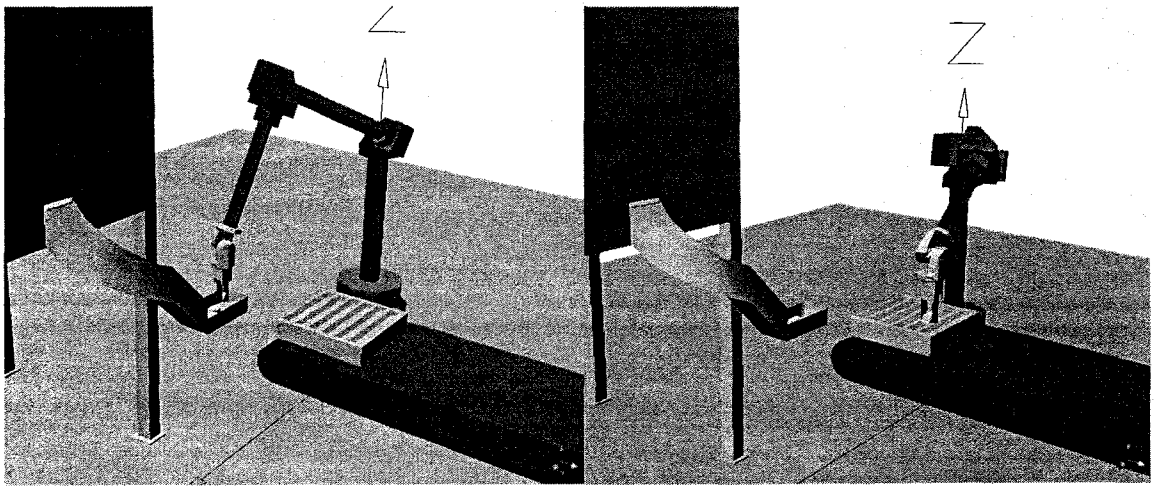


Figure 3.5 Moving From an Initial Position To a Desired Goal Position

Usually, it is desirable for the motion of the manipulator to be smooth. For the purposes of this work, a smooth function is defined as one which is continuous and has a continuous first and second derivatives. Rough, jerky motions tend to cause increased wear in the manipulator. In order to guarantee smooth paths, some sort of constraint must be put on the spatial and temporal qualities of the path between via points.

At this point, there are many choices that may be made, and consequently a great variety in the ways that paths might be specified and planned. Any smooth functions of time which pass through the via points could be used to specify the exact path shape. In this work, two simple choices for these functions will be discussed. These are namely: 1) The Joint Space Scheme; and 2) The Cartesian Space Scheme. In both schemes, the same temporal quality will be used. A 5^{th} order polynomial is used to generate the trajectories (in joint or Cartesian space), as shown in Eq. (3.29).

$$\xi(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \tag{3.29}$$

The initial and final state of the variable ξ are usually known as ξ_0 and ξ_f. In making a single smooth motion, at least four constraints on ξ are evident. Two constraints on the functions value come from the initial and final values:

$$\begin{aligned} \xi(t_0) &= \xi_0 \\ \xi(t_f) &= \xi_f \end{aligned} \tag{3.30}$$

An additional two constraints are that the function is continuous in its first derivative which in this case means that the initial and final first derivatives are zero:

$$\begin{aligned} \dot{\xi}(t_0) &= 0 \\ \dot{\xi}(t_f) &= 0 \end{aligned} \tag{3.31}$$

Finally, two more constraints are that the function is continuous in its second derivative which again means that the initial and final second derivatives are zero:

$$\begin{aligned} \ddot{\xi}(t_0) &= 0 \\ \ddot{\xi}(t_f) &= 0 \end{aligned} \tag{3.32}$$

These six constraints can be satisfied by a polynomial of at least 5th degree. Since a 5th polynomial has four coefficients, it can be made to satisfy the six constraints given by Eq. (3.30), Eq. (3.31), and Eq. (3.32). Eq. (3.30), Eq. (3.31), and Eq. (3.32) can be arranged in matrix form as in Eq. (3.33), to calculate the unknown coefficients of Eq. (3.29).

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5 \\ 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4 \\ 0 & 0 & 2 & 6t_0 & 12t_0^2 & 20t_0^3 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{Bmatrix} = \begin{Bmatrix} \xi_0 \\ 0 \\ 0 \\ \xi_f \\ 0 \\ 0 \end{Bmatrix} \tag{3.33}$$

Figure 3.6 shows an example of the trajectory to be used in the joint and Cartesian space trajectory generation.

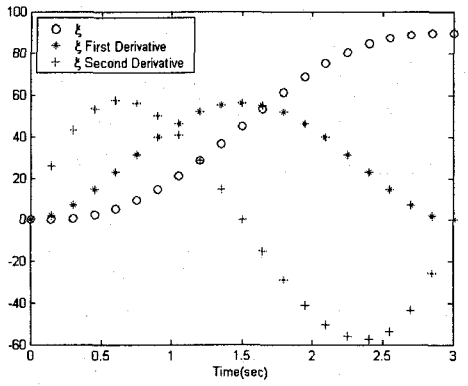


Figure 3.6 Smooth Trajectories Used

3.6.1 Via Points Handling

Sometimes it is necessary to specify the motion in more details than simply stating the initial and desired configuration of the manipulator. One way to include more detail in a path description is to give a sequence of desired via points or intermediate points between the initial and final positions. Thus, in completing the motion, the tool frame must pass through a set of intermediate positions and orientations as described the via points. Each of these via points is actually a frame which specifies both the position and orientations of the end-effector relative to the base frame.

Via points are necessary when working in a crowded environment, in order to insure and obstacle free path of the tool. This work does not treat the planning of an obstacle free path for the end-effector or the other parts of the manipulator. Vidyasagar et al. [22], and Lumelsky et al. [23], have performed extensive research in this area.

In Section 3.6, the constraints that were placed on the trajectory were a zero velocity and acceleration at the start and final point. Yet placing such constraints on the via points yields to lost time and efficiency. A standard practice is to average the velocity at the via points while vanishing the acceleration, see [6], [18], and [36]. Even though in this case

the velocity is not zero, the six constraints necessary to solve for the six coefficients in Eq. (3.29)are satisfied.

3.6.2 Joint Space Scheme

The above noted trajectory can be used for the joint space. For moving the end effector from an initial position and orientation,$\{T_{Initial}\}$, to a final position and orientation, $\{T_{Final}\}$, the inverse kinematics solution can be used to obtain the initial joint configuration $\{q_{Initial}\}$, and the final joint configuration $\{q_{Final}\}$.

$$\{q\} = [\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4 \quad \theta_5 \quad \theta_6]^T \tag{3.34}$$

A trajectory can be generated following the developments of Section 3.6. The time required for the motion is the same for each joint so that all joints will reach the via points at the same time, thus resulting in the desired Cartesian position. Other than specifying the same duration for each joint, the determination of the desired joint angle function for a particular joint does not depend on the functions for the other joints.

Hence, joint space schemes achieve the desired position and orientation at the via points. In between via points the shape of the path while rather simple in joint space, is complex if described in Cartesian space. Joint space schemes are usually easier to compute, and, because we make no continuous correspondence between joint space and Cartesian space, there is essentially no problem with singularities of the manipulator.

3.6.3 Cartesian Space Scheme

As mentioned in Section 3.6.2 , path computed in the joint space can ensure that via and goal points are attained, even when these path points were specified by means of Cartesian frames. However, the spatial shape of the path taken by the end-effector is not a straight line through space, but rather, it is some complicated shape which depends on the

particular kinematics of the manipulator being used. In this section we consider methods of path generation in which the path shapes are described in terms of the functions which compute Cartesian positions and orientations function of time. In this way, we can also specify the spatial shape of the path between path points. The most common path shape is a straight line; but circular, sinusoidal, or other path shapes could be used. In this work, a straight line path will be used.

Each path point is usually specified in terms of a desired position and orientation of the tool frame relative to the station frame. In Cartesian based path generation schemes, the functions which are splined together to form a trajectory are functions of time which represent Cartesian variables. These paths can be planned directly from the user's definition of path points which are {T} specifications relative to the base frame without first performing inverse kinematics. However, Cartesian schemes are more computationally expensive to execute since at run time, inverse kinematics must be solved at the path update rate. That is, after the path is generated in Cartesian space, as a last step the inverse kinematics calculation is performed to calculate desired joint angles. Consider the case of moving from $T_{Initial}$ to T_{Final} where:

$$T_{Initial} = \begin{bmatrix} & & & x_{Initial} \\ & R_{Initial} & & y_{Initial} \\ & & & z_{Initial} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and (3.35)

$$T_{Final} = \begin{bmatrix} & & & x_{Final} \\ & R_{Final} & & y_{Final} \\ & & & z_{Final} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where $R_{Final} = \lfloor r_{ij} \rfloor$ is the 3x3 rotation matrix (i,j=1...3).

For the Cartesian coordinates, x, y and z, the scheme developed in Section 3.6 is applied, or:

$$\begin{aligned} x(t) &= a_{0x} + a_{1x}t + a_{2x}t^2 + a_{3x}t^3 + a_{4x}t^4 + a_{5x}t^5 \\ y(t) &= a_{0y} + a_{1y}t + a_{2y}t^2 + a_{3y}t^3 + a_{4y}t^4 + a_{5y}t^5 \\ z(t) &= a_{0z} + a_{1z}t + a_{2z}t^2 + a_{3z}t^3 + a_{4z}t^4 + a_{5z}t^5 \end{aligned} \tag{3.36}$$

Applying the previous planning on the three orientation parameters (Euler Angles) is difficult to understand intuitively. A very practical approach is the single rotation method. Rotating from representation $R_{Initial}$ to R_{Final} can be performed by rotating the initial frame about the vector $\hat{K}$ by an angle θ according to the right-hand rule. Using the inverse solution, θ can be found according to the following equation:

$$\theta = \cos^{-1}\left(\frac{r_{11} + r_{22} + r_{33} - 1}{2}\right) \tag{3.37}$$

and vector $\hat{K}$ is found consequently:

$$\hat{K} = \frac{1}{2\sin(\theta)}\begin{bmatrix} r_{33} - r_{23} & r_{13} - r_{31} & r_{21} - r_{12} \end{bmatrix}^T \tag{3.38}$$

The single rotation approach might degenerate. Such cases are discussed in detail in Appendix A. A time function can be obtained for can be obtained for θ as well, using

$$\theta(t) = a_{0\theta} + a_{1\theta}t + a_{2\theta}t^2 + a_{3\theta}t^3 + a_{4\theta}t^4 + a_{5\theta}t^5 \tag{3.39}$$

with the following boundary conditions:

$$\begin{aligned} \theta_{Initial} &= 0 \\ \theta_{Final} &= \theta \\ \dot{\theta}_{Initial} &= \dot{\theta}_{Final} = \ddot{\theta}_{Initial} = \ddot{\theta}_{Final} = 0 \end{aligned} \tag{3.40}$$

Equations (3.36), (3.37), and (3.38)can be used in the following algorithm to obtain the joint's trajectory:

t= t+Δt
get θ(t), x(t), y(t), and z(t)

Determine $D(t) \equiv \begin{bmatrix} & & & x(t) \\ & R(\hat{K},\theta) & & y(t) \\ & & & z(t) \\ 0 & 0 & 0 & 1 \end{bmatrix}$

$A = R_{Initial}\, D(t)$

Single Rotation Algorithm

The joint angles obtained from the above algorithm can be used in driving the manipulator actuators.

3.6.4 Between Joint Trajectory and Cartesian Trajectory

Using the Cartesian trajectory scheme, the motion of the end-effector is fully controlled. The Cartesian trajectory scheme is usually used when transporting sensitive objects. For example, if the manipulator is transporting a liquid container, it is advantageous to have the end-effector, consequently the liquid container, move as a smooth function of time. The joint trajectory scheme is used when the motion of end-effector is of no interest to the user. For instance, when the manipulator is moving to pick an object, the motion by which the end-effector moves in time is of no interest, as long as the path is obstacle free.

3.7 Velocity Analysis

In order to determine the manipulator's Jacobian, the velocity propagation calculation can be performed as follows:

$$^{i+1}\omega_{i+1} = {}^{i+1}_{i}R\ {}^{i}\omega_i + \dot{\theta}_{i+1}\ {}^{i+1}z_{i+1} \tag{3.41}$$

$$^{i+1}v_{i+1} = {}^{i+1}_{i}R\left({}^{i}v_i + {}^{i}\omega_i \times {}^{i}P_{i+1}\right) \tag{3.42}$$

where $^{i+1}\omega_{i+1}$ is the angular velocity vector of link (i+1) represented in Frame (i+1)

$\dot{\theta}_{i+1}$ is the rotational velocity of joint (i+1)

$^{i+1}z_{i+1}$ is the z-axis of Frame (i+1) represented in Frame (i+1)

$^{i+1}v_{i+1}$ is the linear velocity of link (i+1) represented in Frame (i+1)

$^{i}P_{i+1}$ is the position vector of link (i+1) with respect to link (i)

The manipulator's Jacobian can be determined by rearranging Eq. (3.41) and Eq. (3.42). The resulting Jacobian will be represented in the Tool Frame $\{T_7\}$.

$$\begin{bmatrix} {}^{7}v_7 \\ {}^{7}\omega_7 \end{bmatrix} = {}^{7}J\,\dot{\theta} \tag{3.43}$$

By differentiating Eq. (3.43), the accelerations of the end effector can be determined through Eq. (3.44).

$$\begin{bmatrix} {}^{7}\dot{v}_7 \\ {}^{7}\dot{\omega}_7 \end{bmatrix} = {}^{7}\dot{J}\,\dot{\theta} + {}^{7}J\,\ddot{\theta} \tag{3.44}$$

The extra term in Eq. (3.35) comes by the differentiation of the Jacobian matrix. The Jacobian matrix is configuration dependent, and hence can be differentiated with respect to time.

3.8 Dynamic Modeling of the Manipulator

The dynamics can be studied through the Lagrangian dynamic formulation. The Lagrangian formulation is an "energy based" approach to dynamics. In this work, the statement of Lagrangian dynamics will be brief and somewhat specialized to the case of a

serial chain mechanical manipulator with rigid links. For a more complete and general reference, see [4]. A manipulator with n joints can be modeled as shown in Eq. (3.45):

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + {}^{0}J^{T}\ {}^{0}F_{Tool} = \tau \tag{3.45}$$

where $M(q)$ is the nxn mass matrix

$C(q,\dot{q})$ is the nxn velocity and Coriolis matrix

$G(q)$ is the nx1 gravity vector

${}^{0}J$ is the manipulator's Jacobian, represented in the base frame

${}^{0}F_{Tool}$ is the external force applied at the end effector represented in the base frame The external force applied at the end effector can be expressed in both the Base Frame $\{T_0\}$ or the Tool Frame $\{T_7\}$. In the case of pick and place operations, the external force applied at the end effector is easily expressed by the gravity field, which is the negative z-direction of the Base Frame, $\{T_0\}$.

The Waelischmiller manipulator used in the simulation and design of the hot cells has six joints. A systematic approach for deriving the differential equations describing a manipulator is presented in Yoshikawa, [19]. Obtaining the complete equations of motion of the manipulator is a daunting task. The symbolic package MATHEMATICA® was used to obtain the equations of motion. The equations were then transferred to MATLAB® using MSWord Visual Basic editor.

3.9 Manipulator Control

The control method used in this work falls into the class of linear control systems, formulated after Craig, [18]. Strictly speaking, the use of linear control techniques is only valid when the system being studied can be mathematically modeled by linear differential equations. For the case of manipulator control, such linear methods must essentially be

viewed as approximate methods. The dynamics of a manipulator are more properly represented by a nonlinear differential equation. However, we will see that it is often reasonable to make such approximations, and it also is the case that these linear methods are the ones most often used in current industrial practice.

A manipulator is modeled as a mechanism which is instrumented with sensors at each joint to measure the joint angle, and an actuator at each joint to apply a torque on the neighboring (next higher) link. Although other physical arrangements of sensors are sometimes used, the vast majority of robots have a position sensor at each joint. Sometimes, velocity sensors (tachometers) are also present at the joints. Various actuation and transmission schemes are prevalent in industrial robots, but many of these can be modeled by supposing there is a single actuator at each joint.

In this work, the manipulator joints are to follow prescribed position trajectories, but since the actuators are commanded in terms of torque, a control system to compute the appropriate actuator commands which will realize this desired motion. Almost always these torques are computed by using feedback from the joint sensors to compute the torque required. Figure 3.7 shows the relationship between the trajectory generator and the physical robot. The robot accepts a vector of joint torques, τ, from the control system. The manipulator's sensors allow the controller to read the vector of joint positions, θ, and joint velocities, $\dot{\theta}$. All signal lines in Figure 3.7 carry nx1 vectors (where n is the number of joints in the manipulator).

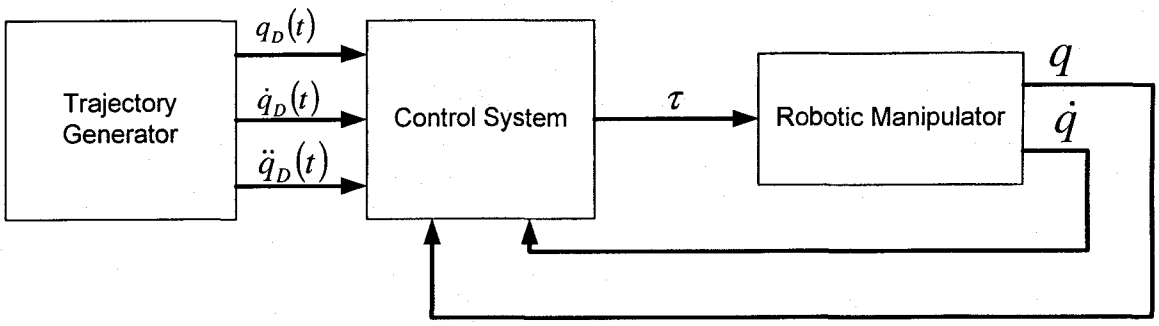


Figure 3.7 High-level block diagram of a robotic manipulator control system

For the case of manipulator control, we developed a model and the corresponding equations of motion. These equations are quite complicated. The rigid body dynamics have the form:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + J^T F_{Tool} = \tau \tag{3.46}$$

The problem of controlling a complicated system like the one presented in Eq. (3.46) can be handled by the partitioned controller scheme. We propose a control law in the form of:

$$\tau = \alpha\tau' + \beta \tag{3.47}$$

Thus we choose:

$$\begin{aligned} \alpha &= M(q) \\ \beta &= C(q,\dot{q}) + G(q) \end{aligned} \tag{3.48}$$

with the servo law

$$\tau' = \ddot{q}_D + K_V \dot{E} + K_P E \tag{3.49}$$

where

$$E = q_D - q \tag{3.50}$$

The resulting control system is shown in Figure 3.8 below:

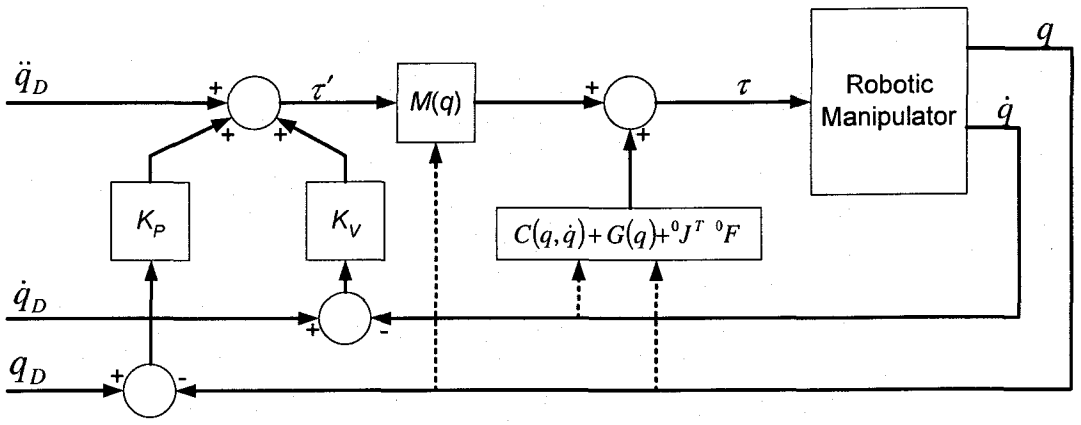


Figure 3.8 A Model Based Robotic Manipulator Control System

Using Eq. (3.47) through Eq. (3.50) it is quite easy to show that the close loop system is characterized by the error equation:

$$\ddot{E} + K_V \dot{E} + K_P E = 0 \tag{3.51}$$

Note that this vector equation is decoupled since the matrices K_V and K_P are diagonal so that Eq. (3.51) could just as well be written on a joint-by-joint basis as:

$$\ddot{e}_i + k_{v_i} \dot{e}_i + k_{p_i} e = 0 \tag{3.52}$$

Eq. (3.52) represents an unforced second order differential equation, that has the following poles.

$$\lambda_i = \frac{-k_{v_i} \pm \sqrt{k_{v_i}^2 - 4k_{p_i}}}{2} \tag{3.53}$$

The choice of k_{v_i} and k_{p_i} can lead to the desired placement of the poles λ_i. Dynamic systems with critical damping is usually desired.

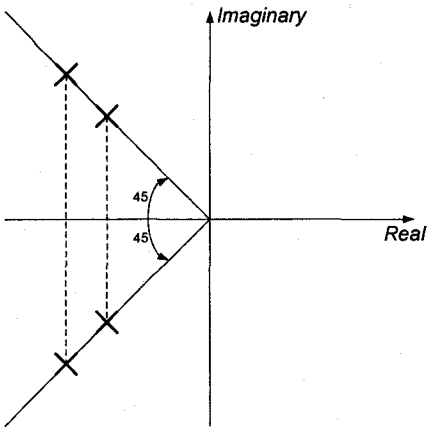


Figure 3.9 Pole Placement of the Decouple Manipulator's Equations

For obtaining critical damping of the dynamic behavior, the poles has to be on the 45° line as in Figure 3.9. For such placement, the following should be valid:

$$\mathrm{Re}(\lambda_i) = \mathrm{Im}(\lambda_i) \tag{3.54}$$

and based on Eq. (3.54), the values of k_{v_i} and k_{p_i} can be determined as follows:

$$k_{v_i} = 2\sqrt{k_{p_i}} \tag{3.55}$$

Using this methodology, setting of the control gains is simple, and is independent of the system parameters.

It is worth noting at this point that the ideal performance represented by Eq. (3.51) is unattainable in practice due to many reasons, the most important two being:

- Discrete nature of a digital computer implementation as opposed to the ideal continuous time control law implied by Eq. (3.49) and Eq. (3.51).
- Inaccuracy in the manipulator model (needed to compute Eq. (3.46)).

For the purposes of this work, and since the model can be exactly determined as of Eq. (3.46), the control law of Eq. (3.47) is implemented as will be explained in Section 4.1.

Figure 3.10 shows the trajectory followed by the manipulator's joint's to move the fuel pellets from the pressing machine to the sintering oven. The error is shown in Figure 3.11 for all the six joint. The error converges to zero at the end of the motion. Figure 3.12 shows the torque applied at each joint of the first robot while moving the pellets from the press shoot to the sintering oven.

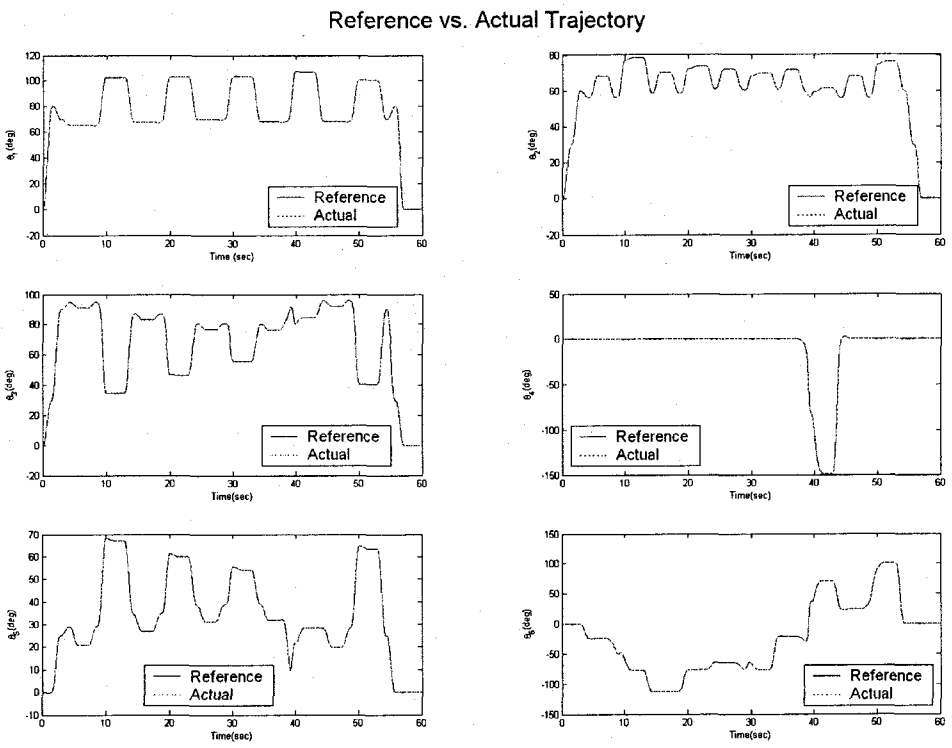


Figure 3.10 First Robot's Trajectory in Powder Processing Hot Cell

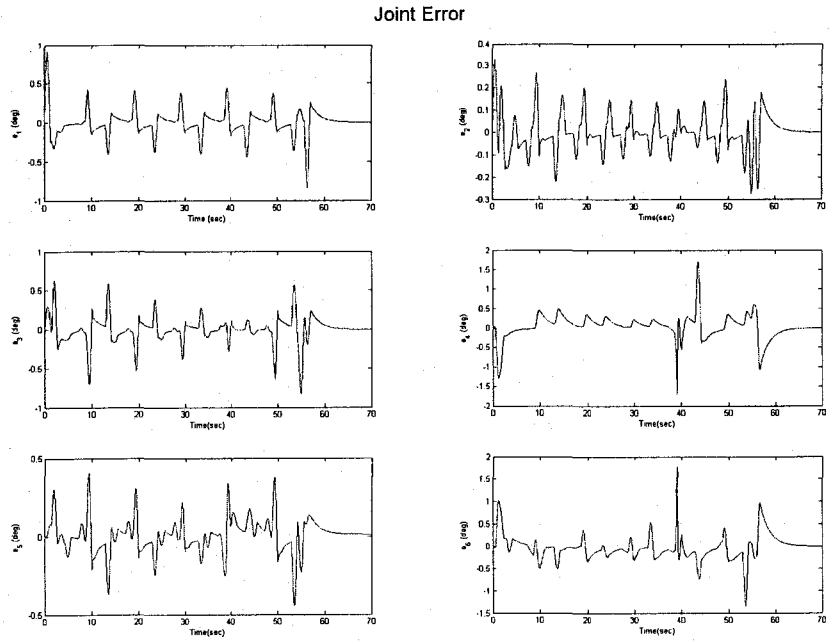


Figure 3.11 First Robot's Joints Error in Powder Processing Hot Cell

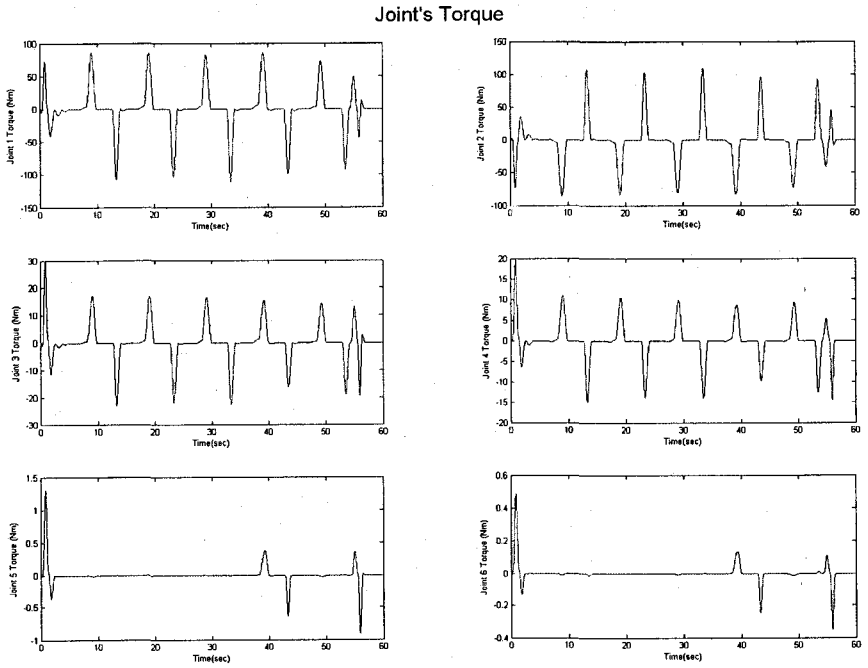


Figure 3.12 First Robot's Control Torque in Powder Processing Hot Cell

CHAPTER 4

VIRTUAL DESIGN AND TESTING APPROACH

The goal of this work is to develop simulations of manufacturing processes of transmuter fuel. Properly designed robotic work cells would likely result in reduced cost of operation as well as increased reliability by reducing the potential for human error during materials handling operations. The candidate fuel manufacturing processes are being modeled using the MSC.visualNastran® in conjunction with Pro Engineer® and MATLAB®.

4.1 Software Platform and Interface

The geometry and solid modeling of the equipment used in the hot cells, was modeled using Pro Engineer®. The next step is to export the geometry into MSC.visualNastran® where joints, motors and physical parameters can be assigned such as restitution and friction coefficients. The motors and actuators of the MSC.visualNastran® model can be controlled via MATLAB Simulink®. The patch connecting MSC.visualNastran® allows for inputs to be entered from MATLAB Simulink® into the MSC.visualNastran®. Figure 4.1 shows the block diagram title "WAELLISHMILLER" which serves as the connection between the two software packages.

Several robot simulation models have been developed and their correct performance has been verified. Realistic simulations permit the prediction, analysis and elimination of potential problems such as collisions and unreachable locations before the actual execution

of a programmed sequence. An accurate process simulation will aid in sizing fuel manufacturing hot cells, and help to model process losses.

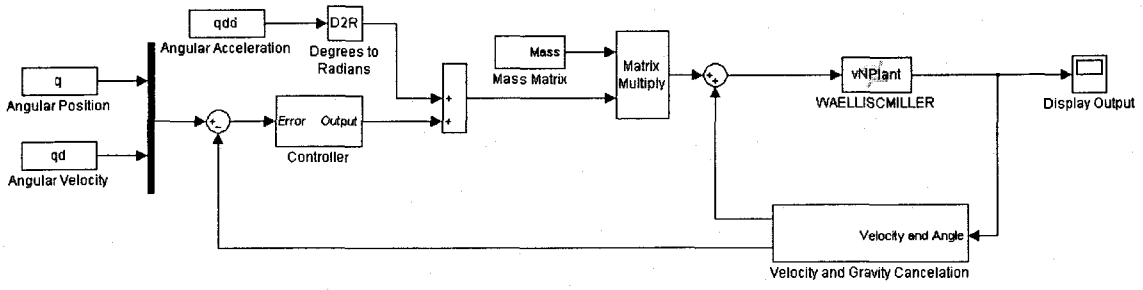


Figure 4.1 Connection Between MSC.visualNastran® and MATLAB Simulink®

Figure 4.2 shows the "vNPlant" block when opened. A menu pops up showing the interface between MSC.visualNastran® and MATLAB Simulink® with the input controls and output meters specified as in shown in the menu titled "Block Parameters:vNPlant > Manipulator".

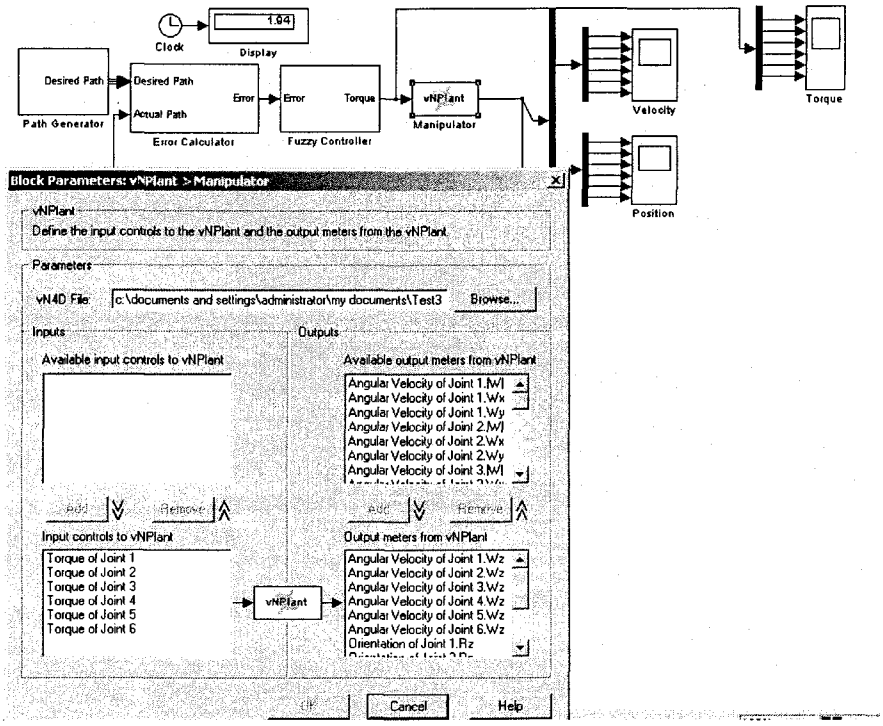


Figure 4.2 Interface Between MSC.visualNastran® MATLAB Simulink®

4.2 MATLAB® Programming

The positions and orientations of the fuel pellets, in the case of MOX and Metallic fuels, or the fuel compacts, in the case of dispersion fuels, are input in a MATLAB® program, see B.2. A separate program reads the positions and the respective orientations, and solves for the inverse kinematics for the position of each pellet, see B.3.

A trajectory planner program uses the above obtained transformation matrices. The manipulator will move according to joint trajectory schemes when load-free, and according to Cartesian trajectory schemes when moving pellets or compacts, see B.8.

The complete joint trajectory that has to be executed by each manipulator is organized in an array that is then supplied to the MATLAB Simulink® file. The commands are then fed into the MSC.visualNastran® as explained in Section 4.1.

4.3 Equipment Modeling

The hot cells developed used hard as well as soft automation. The hot cells designed included conveyor belts, manipulators and other equipment relevant to the fuel manufacturing process.

The manipulators were geometrically modeled after the TELBOT TB100, show in Figure 2.5 on page 42. The dynamics of the manipulators was fully studied. The dynamic equations were obtained as per Section 3.8. The mathematical model was programmed in MATLAB® and the control law of Section 3.9 was implemented.

The manipulator object in the MSC.visualNastran environment received six torques as the control input and the output was its six joint position and angular velocity. Consequently, the feedback control law of Section 3.9 could be applied.

For the trajectory generation, a joint trajectory as well as a Cartesian trajectory were used as of Section 3.6 and its corresponding subsections. The joint trajectory generation scheme was used when the manipulator was not transporting any pellets. The Cartesian trajectory scheme was used when moving the pellets in order to have the pellets under tight conditions of acceleration and velocity.

Three types of fuel manufacturing processes was simulated. MOX, metallic and dispersing fuels. The three simulation used the same model of the manipulator. Each simulation required different equipment, and hence the layout, as well as the number of manipulators used was distinct for each hot cell model.

4.4 MOX Fuel Manufacturing Hot Cell Simulation

The MOX fuel manufacturing steps were described in Figure 1.4, Figure 1.5 and in Section 1.6.2. Figure 4.3 shows schematically the organization of a working powder

processing plant, the Framatome UO_2 fuel manufacturing plant in Lingen, Germany. The Lingen plant operates without shielding or hot cells, but places tight controls on UO_2 losses, which occur mostly in particulate form. A possible configuration for a MOX fuel manufacturing hot cell is shown in Figure 4.4. The workspace of the two TELBOT manipulators in the hot cell is shown in Figure 4.4 as well. The software packages that were available for this work would not allow the complete modeling of the manufacturing process. For example, the blending of the powders and the press of the pellets can not be modeled, so the simulation of the manufacturing process will start with already pressed pellets. A reduced model was used for the computer simulation. The layout used in the simulation is shown in Figure 4.5.

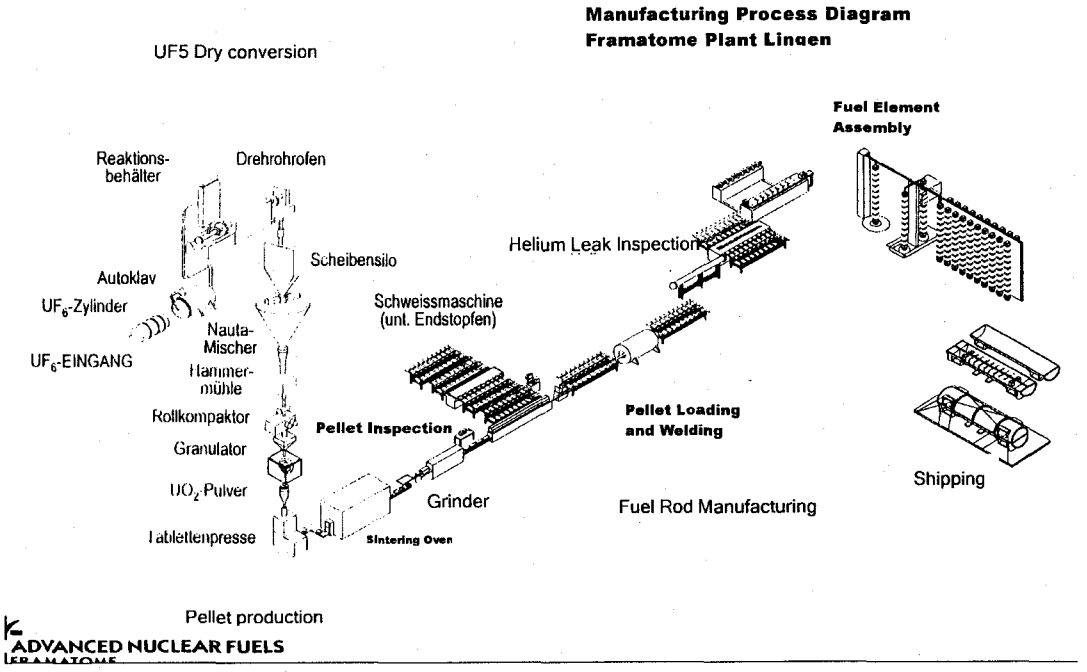


Figure 4.3 Lingen UO_2 Plant Schematic (Framatome)

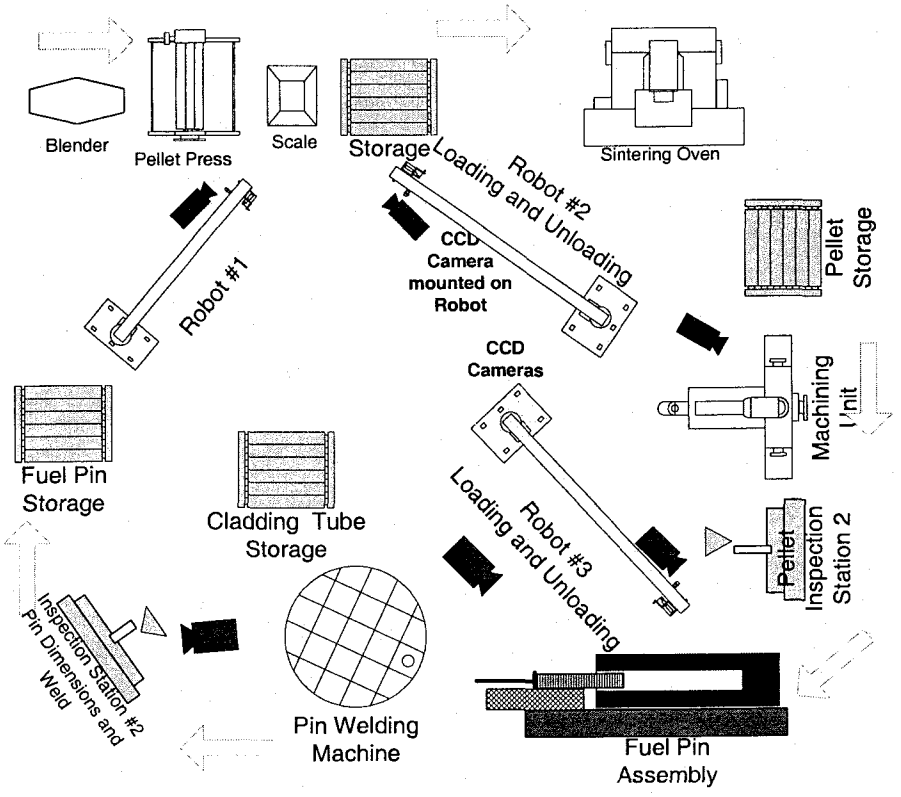


Figure 4.4 Possible Configuration for Powder Processing Fabrication Work Cell

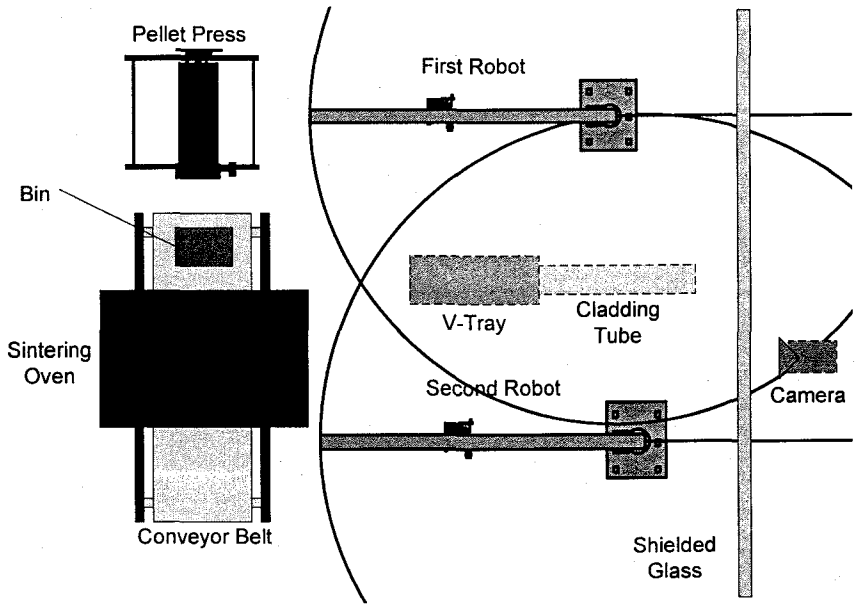


Figure 4.5 Layout of MOX Fuel Manufacturing Hot Cell Used in Simulation

In the above simulated virtual hot cells, the first robot moves the pressed pellets to a bin. A conveyor belt transports the bin through a sintering oven. The second robot moves the pellets to a V-Tray, and eventually inserts them in the cladding tube. A camera behind a shielded glass inspects the sintered pellets for size and heat treatment.

4.5 Metallic Fuel Manufacturing Hot Cell

The manufacturing of metallic fuels was explained in Section 1.6.2. Figure 4.6 shows a possible configuration for metallic fuel manufacturing hot cell. A casting furnace is present in the hot cell for the casting of the fuel slugs. The molds are reusable, hence the surfaces of the fuel slugs wont be smooth. A grinder must be used for the surface finishing of the fuel slugs.

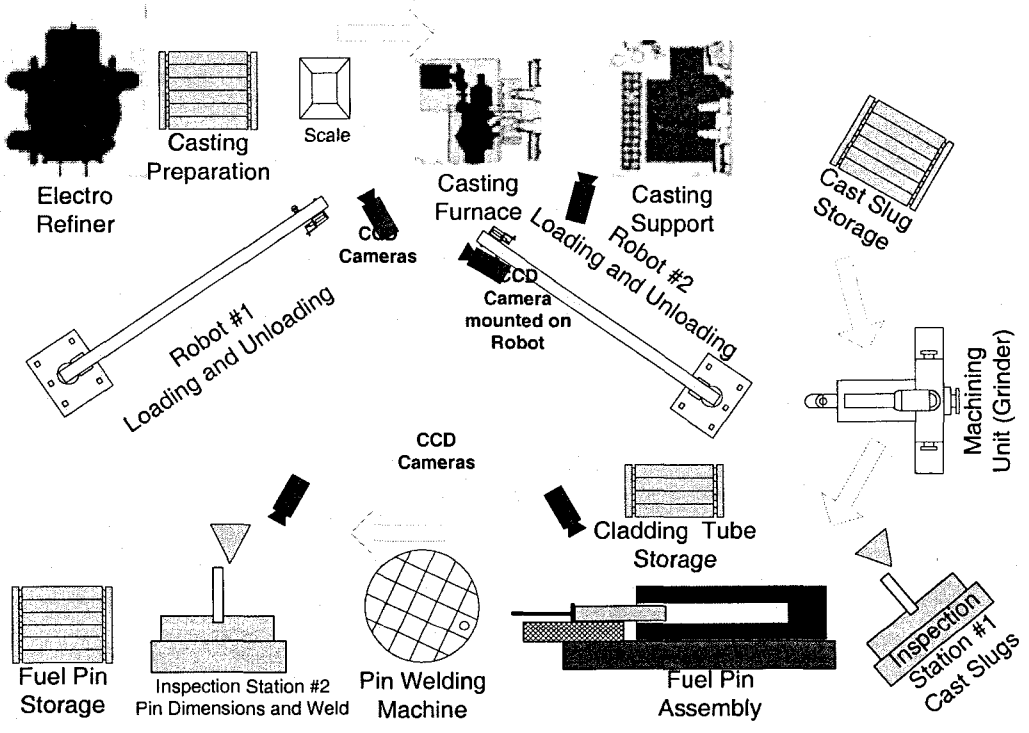


Figure 4.6 Possible Configuration for Metallic Fuel Fabrication Work Cell

After casting the fuel slugs, the fuel slugs are inserted in cladding tubes, and the process is carried on similar to power based transmuter fuels.

As for the case of MOX fuels, an alternative layout had to be proposed for the manufacturing process. The opening of the molds, and the grinding of the fuel slugs cannot be simulated in the available software packages.

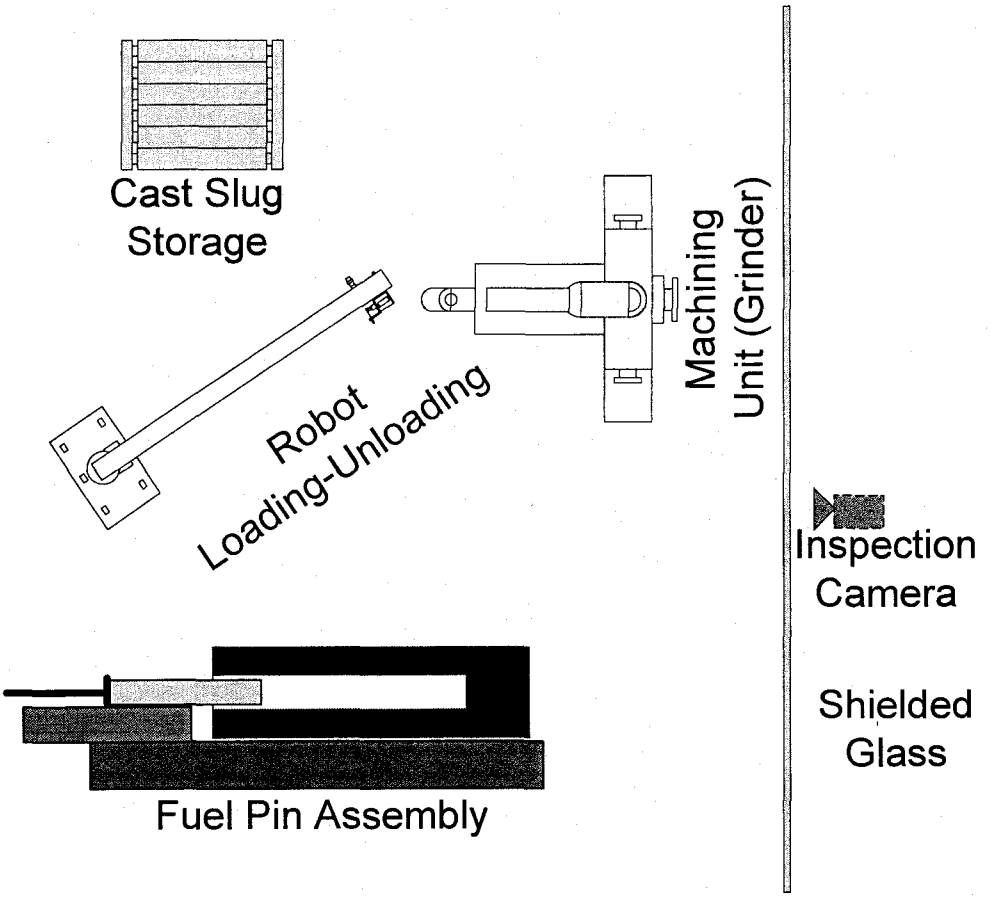


Figure 4.7 Layout of Metallic Fuel Manufacturing Hot Cell Used in Simulation

4.6 Dispersion Fuel Manufacturing Hot Cell

The manufacturing of dispersion based fuel was described in Section 1.6.2. Figure 4.8 shows the layout of the hot cell developed and simulated. Three manipulators were used

in the simulation to assist in the assembling of the billets, see Figure 1.6 on page 19. The camera behind the shielded glass monitors the assembly process.

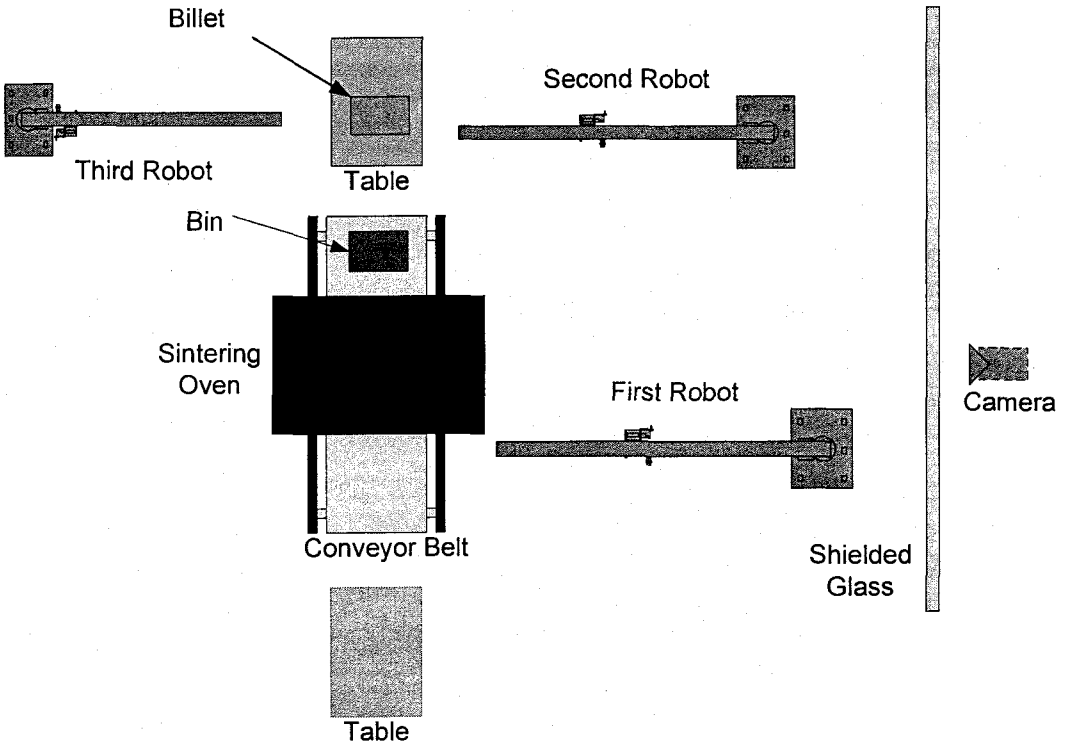


Figure 4.8 Schematic Layout for Dispersion Fuel Hot Cell

CHAPTER 5

SIMULATION RESULTS

The main purpose of simulating engineering system is to obtain a rough performance assessment of the system. Some systems are too expensive or too dangerous to experiment with. Developing models and constructing mathematical equations for such systems give the ability of simulating them, and hence gaining an insight in the general behavior of the system in a cheap and safe way.

The simulations performed permit the detailed performance and safety assessment of all mechanical components in the manufacturing hot cell. The analysis detects possible accidents and failures in the hot cell. The simulation provides the capability for analyzing irregular events such as collisions and ejected parts. It allows for the comprehensive examination and testing of failure scenarios as well as recovery procedures, and thus for iterative optimization of all mechanical components, ensuring maximum reliability and safety.

5.1 Quantitative Results

The simulations performed gives a very good estimate of the torques and velocities in the hot cell. Mass and mass moment of inertia values were assigned for the robot's links. The pellet's mass was also estimated from density information of the nuclear powders. The control algorithm described in Section 3.9 allows us to calculate the torques required to move the robotic manipulator along the specified trajectory. The weight of the pellets

is considered as a disturbance at the manipulator's end-effector. Figure 5.1 shows the acceleration of a pellet through its motion from the pellet press machine to the sintering bin (see Section 4.4 on page 80 for information on MOX fuel manufacturing).

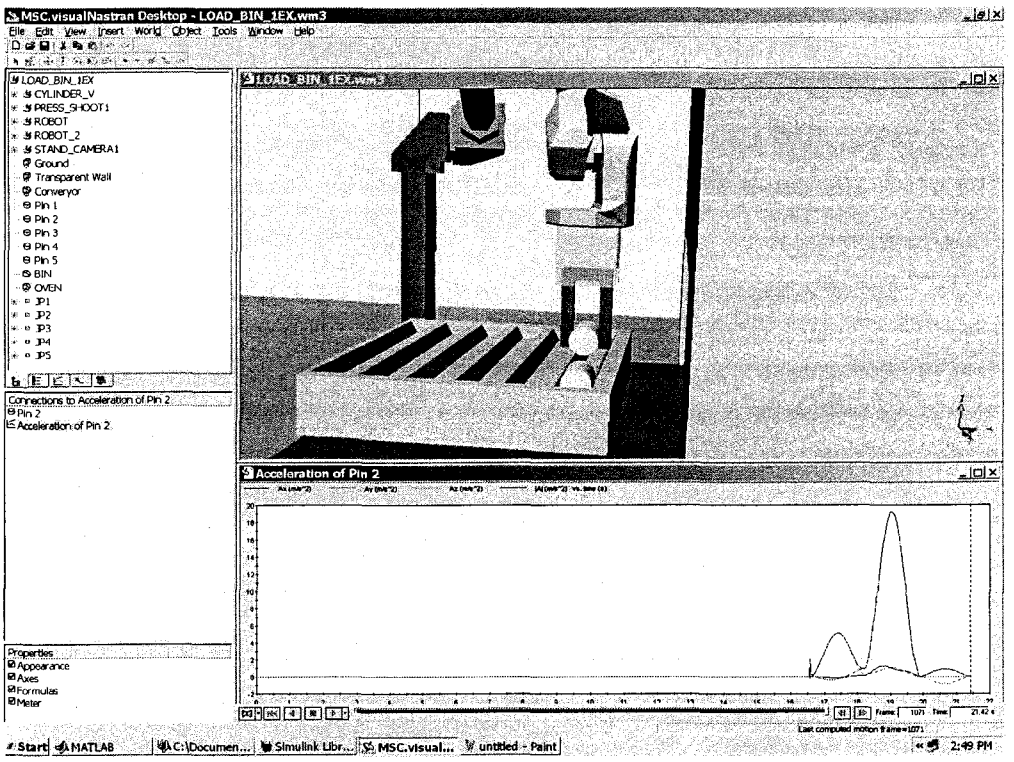


Figure 5.1 Acceleration of The Pellet Before Being Placed in The Sintering Bin

The significance of the graph shown in Figure 5.1 is that it allows for careful trajectory generation for the end effector. Depending on the brittleness of the pellets, the maximum acceleration of the end effector can be specified so that the pellet does not slip from the gripper.

The friction between the pellets and their surrounding is of great importance as well. Such information can be used to calculate the heat generated while moving the pellets in

the hot cell. The friction between the pellets and the cladding tube is of special importance. Figure 5.2 shows the friction force between the "blue" and the V-Tray and with the cladding tube. The graph to the left shows the friction force between the "blue" pellet with the V-Tray, while the graph to the right shows the friction force between the "blue" pellet and the cladding tube. Notice that the friction force converges to zero in a smooth fashion since the end effector is moving in a smooth trajectory (see Figure 3.6 on page 64) while pushing the pellets.

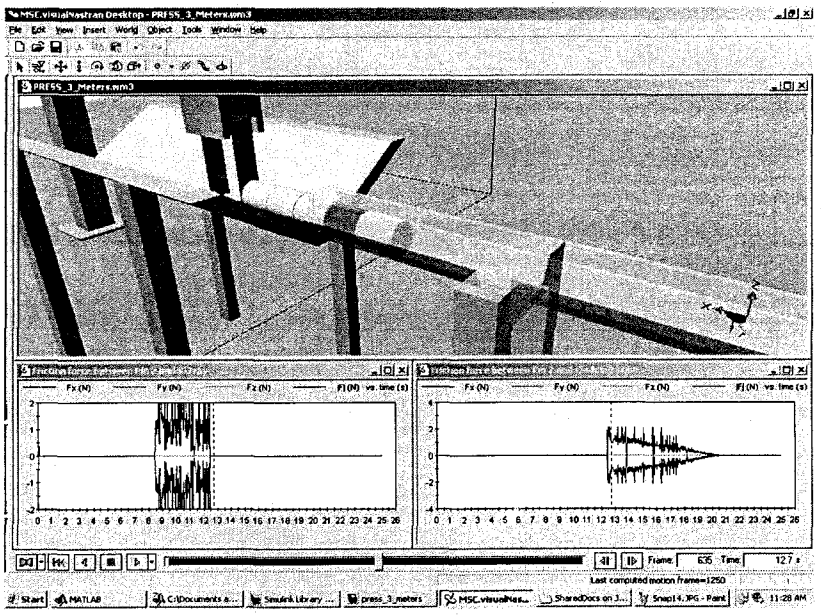


Figure 5.2 Friction Between Fuel Pellet, Cladding Tube and V-Tray

5.2 Atypical Events

In addition to quantitative results that can be obtained from the simulations, the virtual models can be used to predict and simulate atypical events. Atypical events can range from the ejection of some pellets from the cladding tube, to mis-placing a pellet in

the hot cell. The hot cell should have enough redundancy to recover from such accidents. The simulation performed gives an insight regarding such events.

5.2.1 Pellet Buckling

In order to insert the pellets in the cladding tube, the pellets are usually placed in a V-Tray. The manipulator's end effector can be used to push the pellets along the V-Tray into the cladding tube. Careful path and trajectory planning is required in this process in order to insert the pellets inside the cladding tube.

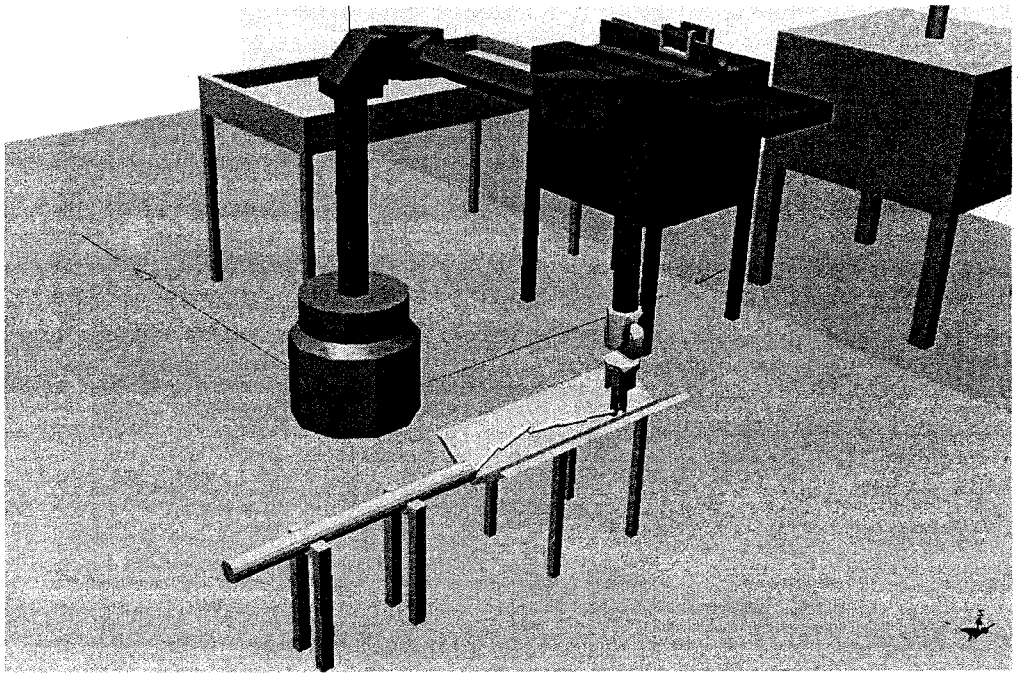


Figure 5.3 Buckling of Metallic Pellets while Inserted in the Cladding Tube

Figure 5.3 shows the buckling of metallic fuel pellets while being inserted in the cladding tube. Figure 5.3 was captured from a simulation where the end effector was moving in a high speed while inserting the pellets. Figure 5.4 below shows the consequence of excessive speed. Recovering from such accident might be costly and hard.

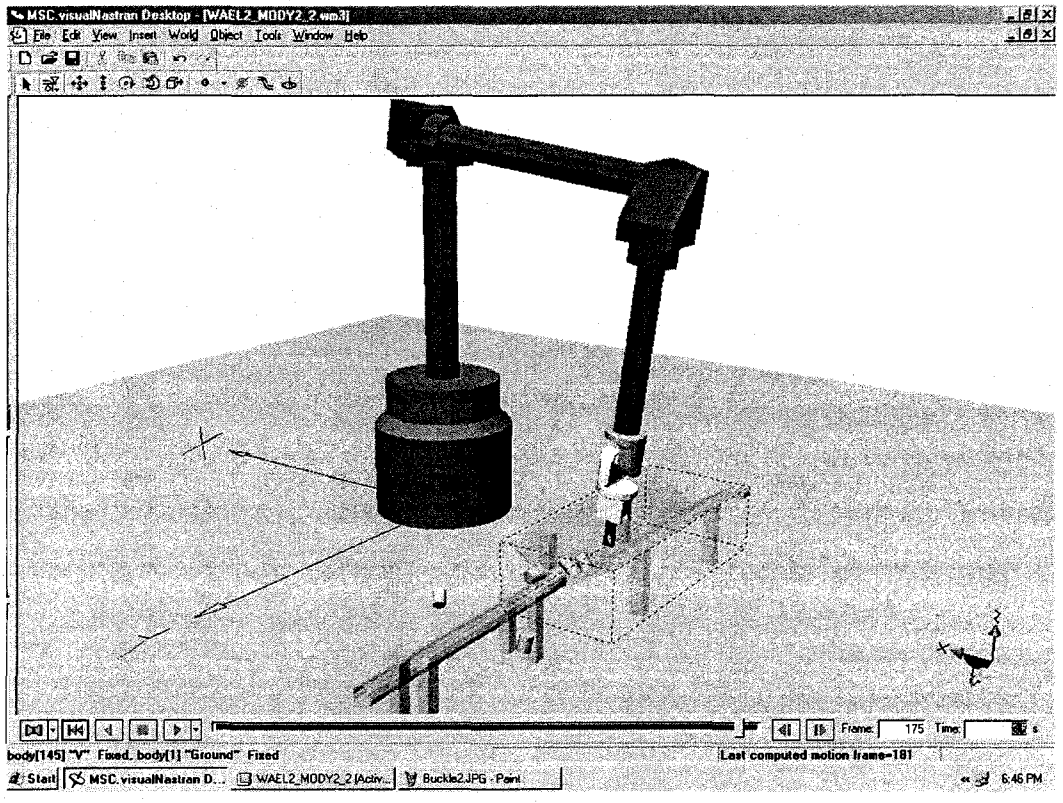


Figure 5.4 Ejected MOX Fuel Pellets

5.2.2 Mis-Dropping Pellets

Any engineering system can mis-function at some times. The hot cell control system should be able to recover from mis-dropping a pellet for example. Figure 5.5 shows a pellet that was dropped on the floor of the hot cell. The manipulator then approaches the pellet, engages it, and move it back into the process. It is assumed that the hot cell is equipped with a camera that will provide spatial coordinates of the pellets. Some types of fuel might produce dust when impacting the ground, and hence the robotic manipulators operating in the hot cell should have the capability of cleaning the dust produced.

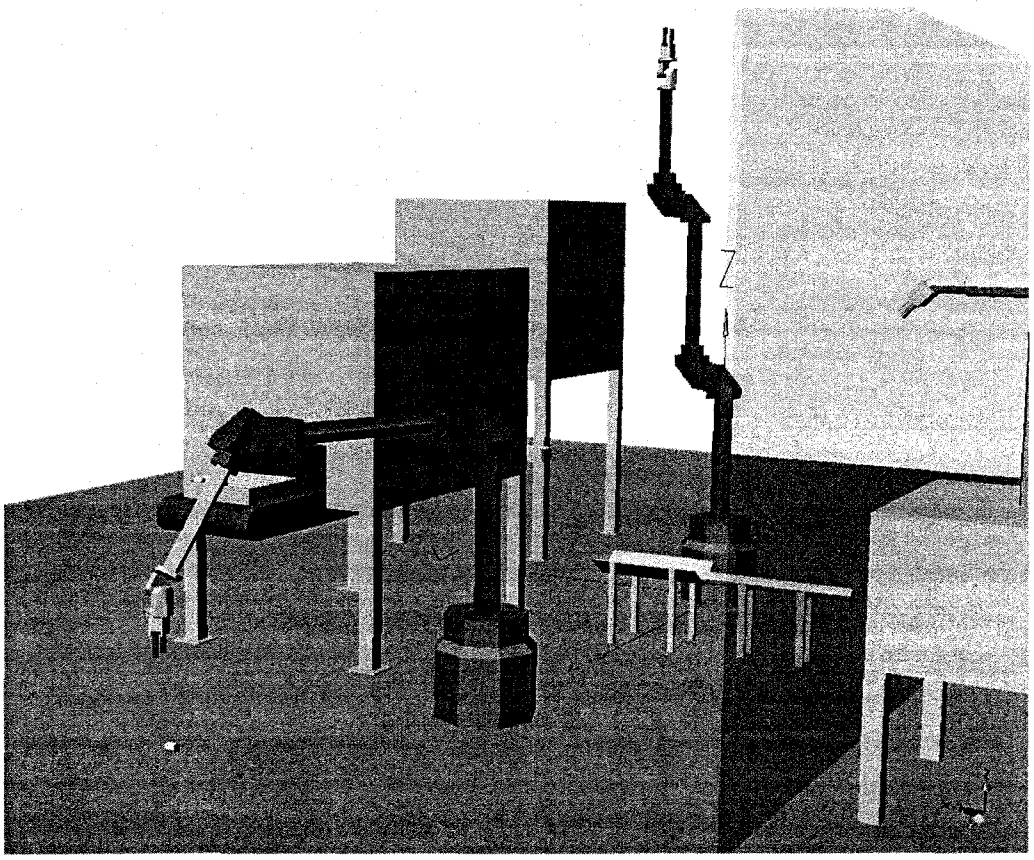


Figure 5.5 Mis-Dropping a Fuel Pellets

The above were examples of the simulations performed. A Compact-Disk containing all the files and the simulations performed will be submitted with this thesis.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

This work has presented the development of virtual hot cells for the manufacturing of transmuter fuel. Hot cells fabricating transmuter fuels were developed using a multitude of software packages. Pro Engineering® was used for the geometric modeling. The models were exported to MSC.visualNastran® where kinematic constraints could be assigned between different bodies. MATLAB Simulink® was used to interface MATLAB® and MSC.visualNastran®. MATLAB® was used for implementing real time feedback control laws.

Three transmuter fuel types were modeled. Hot cells manufacturing MOX, metallic, and dispersion based fuels were developed. The hot cells used hard as well as soft automation, such as conveyor belts and robotic manipulators.

The manipulators were completely modeled. Inverse kinematic solutions were obtained, and smooth trajectory generation algorithms were implemented. A computed torque control law was used for the control of the manipulators.

The simulations performed would likely result in reduced cost of operation as well as increased reliability by reducing the potential for human error during materials handling operations. Atypical events can be predicted and potential flaws in the hot cell design can be avoided.

Finally, the main contribution of this work is the establishment of a basis for simulation of hot cells. The virtual models developed can assist the DOE in selecting the best manufacturing routes. They can be used to perform optimization on all the devices and equipment used in the hot cell. Quantitative information (see Section 5.1) can be used to optimize the use of the space and give an insight in other physical quantities that can't be simulated (heat generated when inserting pellets in cladding tube). Trail simulations can be performed to explore accidents and ways to recover from them.

6.2 Future Work

To a far extent, MSC.visualNastran® was a proper software for performing the hot cell manufacturing procedures. Yet, the software did not include static friction. When the manipulator moves a pellet, the friction between the gripper and the fingers could not be obtained. The software's technical support were contacted, and confessed to this problem.

The programming that was done was extensive. Programs and subroutines had to be written in order to solve the inverse kinematics, generate the necessary trajectories, and compute the complete commands that has to be fed to the hot cell manipulators. A software having manipulator function already built in would shift the attention to the process itself.

APPENDIX A – SINGLE ROTATION METHOD

A frame can be obtained by the rotation about an axis of another frame. For instance, the notation $R_X(30)$ gives the description of an orientation by giving an axis, $\hat{X}$, and angle, 30°. This is an example of a single angle representation. If an axis is a general direction (rather than one of the unit directions) any orientation may be obtained through proper axis and angle selection. Consider the following description of a frame {B}:

Start with the frame coincident with a know frame {A}. Then rotate {B} about the vector $\hat{K}$ by an angle θ according to the right-hand rule.

Vector $\hat{K}$ is sometimes called the equivalent axis of finite rotation. A general orientation of {B} relative to {A} may be written as $R(\hat{K}, \theta)$ or $R_{\hat{K}}(\theta)$ and will be called the equivalent angle-axis representation.

If the rotation matrix is defined by $R_{Final} = [r_{ij}]$, then first we can solve for θ then we can solve for $\hat{K}$.

The single rotation angle θ can be solve for using the inverse solution presented in Eq. (A.1).

$$\theta = \cos^{-1}\left(\frac{r_{11} + r_{22} + r_{33} - 1}{2}\right) \tag{A.1}$$

Consequently, the rotation vector can be found as:

$$\hat{K} = \frac{1}{2\sin(\theta)}\left[(r_{32} - r_{23}) \quad (r_{13} - r_{31}) \quad (r_{21} - r_{12})\right]^T \tag{A.2}$$

If $\sin(\theta)=0$, then the solution of $\hat{K}$ degenerates. To check for this condition, the value of $r_{11}+r_{22}+r_{33}$ is of great importance.

If $r_{11}+r_{22}+r_{33}=3$ or very close to 3, then $\theta=0°$. In this case, $\hat{K}$ is an arbitrary unit vector. Since the rotation angle is zero, the rotation vector does not matter.

If $r_{11}+r_{22}+r_{33}=-1$ or very close to -1, then $\theta=180°$. In this case, the solution will depend on the value of the diagonal elements, r_{11}, r_{22} and r_{33}.

If r_{11} is not very close to -1, then the rotation vector $\hat{K}$ can be solved for as in Eq. (A.3).

$$\begin{aligned} k_x &= \pm\sqrt{\frac{r_{11}+1}{2}} \\ k_y &= \frac{r_{21}}{2k_x} \\ k_z &= \frac{r_{31}}{2k_x} \end{aligned} \tag{A.3}$$

If r_{22} is not very close to -1, then the rotation vector $\hat{K}$ can be solved for as in Eq. (A.4).

$$\begin{aligned} k_y &= \sqrt{\frac{r_{22}+1}{2}} \\ k_x &= \frac{r_{12}}{2k_y} \\ k_z &= \frac{r_{13}}{2k_y} \end{aligned} \tag{A.4}$$

If r_{33} is not very close to -1, then the rotation vector $\hat{K}$ can be solved as in Eq. (A.5).

$$\begin{aligned} k_z &= \sqrt{\frac{r_{33}+1}{2}} \\ k_x &= \frac{r_{13}}{2k_z} \\ k_y &= \frac{r_{23}}{2k_z} \end{aligned} \tag{A.5}$$

Appendix A allows for the complete solution of any rotation matrix, with special attention paid for degenerate cases.

APPENDIX B – MATLAB® CODE

B.1 telbot.m

```
%-------------------------------------------------------------------------%
% Jamil M. Renno
%
% Thesis - VIRTUAL DESING AND MODELING OF VARIOUS MANUFACTURING PROCESSES
%          FOR REMOTE FABRICATION OF TRANSMUTER FUEL
%
%
% Department of Mechanical Engineering
% University of Nevada, Las Vegas
%
% Revision 0 - August 15, 2003
%
%-------------------------------------------------------------------------%
%
% telbot - creates the robot object in the MATLAB environment. The function
%          was developed by Coke in his robotics toolbox
%
%------------------------------ASSUMPTIONS------------------------------%
%
% All the angles are in radians
% All dimensions are in centimeters
%
%-------------------------------------------------------------------------%

% The robot object is a global variable
global r;

% The last argument of the function 'link' is 'modified' which is the
% modified DH Table as reported by Craig

L{1}=link([0       0        pi/2    168.91    0    pi/2  ],'modified');
L{2}=link([pi/2    0        pi/2    0         0     pi/2 ],'modified');
L{3}=link([0       128.27   -pi/2   53.363    0   -pi/2 ],'modified');
L{4}=link([-pi/2   0        0       95.88     0   0     ],'modified');
L{5}=link([pi/2    0        0       0         0   0     ],'modified');
L{6}=link([-pi/2   0        0       0         0   0     ],'modified');
L{7}=link([0       0        0       32.07     0   0     ],'modified');

% Define the robot object using 'robot' function
r=robot(L);

% The manipulator name is TELBOT, and it is manufactured by Waelishmiller
r.name='TELBOT Manipulator';
r.manuf='Waelishmiller';
```

B.2 pelletpositions.m

```
%-------------------------------------------------------------------------%
% Jamil M. Renno
%
% Thesis - VIRTUAL DESING AND MODELING OF VARIOUS MANUFACTURING PROCESSES
%          FOR REMOTE FABRICATION OF TRANSMUTER FUEL
%
%
% Department of Mechanical Engineering
% University of Nevada, Las Vegas
%
% Revision 0 - August 15, 2003
%
%-------------------------------------------------------------------------%
%
% pelletpositions - includes the positions and orientations of pellets in
%                   the hot cell
%
%------------------------------ASSUMPTIONS------------------------------%
%
% All the angles are in radians
% All dimensions are in centimeters
%
%-------------------------------------------------------------------------%
% 'pelletpositions.m' stores the pellets information for picking them from
% the shoot to the conveyor belt bin
% Pellets' information are arranges in 4x4 homogenous transformation
% matrices describing the way they will be handled by the manipulator


% The matrices will be global in the workspace
global pellet;  % pellet is the initial position of the pellets
global lift;    % lift is the lift positions of the pellets
global drop;    % drop is the drop positions of the pellets
global lowerp;  % lowerp is the lowered position of the pellets before
dropping
global camex;   % camex is the position to expose the pellets to the
camera

% Here are the initial x,y, and z coordinates of the pellets
% Each row corresponds to the position of one pellet
% The columns correspond to x, y, and z coordintates

p=[161.55    -14.139         94.402;    % Position of pellet 1
   172.04    -12.243         94.402;     % Position of pellet 2
   179.73    -8.3848         94.402;    % Position of pellet 3
   180.04    -14.757         94.950;     % Position of pellet 4
   160.75    -5.7979         94.402];   % Position of pellet 5

% Here are the orientations in the XYZ Euler angles of the end effector
w.r.t
% the base coordinate frame when picking the pellet
% Each row corresponds to one pellet
% Each column corresponds to the Euler rotations around X, Y and Z
% respectively

or=[180 0   0 ;
    180 0 -90 ;
    180 0 -45 ;
```

```
    180 0   0   ;
    180 0   45 ];

% Here are the drop x,y and z coordinates of pellet
% Each row corresponds to the position of one pellet
% The columns correspond to x, y, and z coordintates
d=[197      99          130;
   188.5    99          130;
   180      99           130;
   171.5    99          140;
   197      91          130];

% Here is the drop off orientation
% The drop off orientation is the same for all the pellets, hence this
% orientation will consist of one only row vector

dor=[180 0 -90                                    ;
     180 0 -90                                    ;
     180 0 -90                                        ;
     ieuler(rotjam([180 0 -90])*rotjam([-60 0 0]));
     180 0 90                                    ];

for i=1:size(p,1)
    % Engagement Homogenous Transformation
    % This is the desired position and orientation to pick pellet 1
    pellet(:,:,i)=rotjam(or(i,:))+[0 0 0 p(i,1);
                                   0 0 0 p(i,2);
                                   0 0 0 p(i,3);
                                   0 0 0 0     ];

    % Lift Homogenous Transformation
    % This is where pellet 1 will be lifted after it is already picked
    lift(:,:,i)=rotjam(or(i,:))+[0 0 0 p(i,1)    ;
                                 0 0 0 p(i,2)    ;
                                 0 0 0 p(i,3)+30;
                                 0 0 0 0         ];

    % Drop Off Homogenous Transformation
    drop(:,:,i)=rotjam(dor(i,:))+[0 0 0 d(i,1);
                                  0 0 0 d(i,2);
                                  0 0 0 d(i,3);
                                  0 0 0 0     ];

    % Lower Homogenous Transformation
    % The pellet will be just lowered
    lowerp(:,:,i)=rotjam(dor(i,:))+[0 0 0 d(i,1)   ;
                                    0 0 0 d(i,2)   ;
                                    0 0 0 d(i,3)-5;
                                    0 0 0 0        ];
end;
```

B.3 inversekinematics.m

```
%-------------------------------------------------------------------------%
% Jamil M. Renno
%
% Thesis - VIRTUAL DESING AND MODELING OF VARIOUS MANUFACTURING PROCESSES
%          FOR REMOTE FABRICATION OF TRANSMUTER FUEL
%
```

```
%
% Department of Mechanical Engineering
% University of Nevada, Las Vegas
%
% Revision 0 - October 9, 2003
%
%-------------------------------------------------------------------------%
%
% inversekinematics - Solves for the inverse kinematics of a end effector
%                     location and orientation
%
%-------------------------------ASSUMPTIONS-------------------------------%
%
% All the angles are in radians
%
%-------------------------------------------------------------------------%
%
% function q = inversekinematics(r,T, iguess)
% 'q' is the returned joint angle
% 'r' is the robot object containing the DH Table
% 'T' is the desired tool-base homogenous transmformation
% 'iguess' is the initial guess for the joint angle values
% If 'iguess' was not provided, [0 0 0 0 0 0] is the default
%
%-------------------------------------------------------------------------%

function q=inversekinematics(r,T,iguess)

% Define tolerance value for checking singularity
tol=0.001;

% Load the different parameters of the DH table into arrays so they can be
% used in the equations easily

% Load the links of the manipulator into link objects
for i=1:7
    L{i}=r.link{i};
end;

% For each link, extract the 'a', 'd', and 'alpha' parameters
for i=1:7

    % 'a' is the the link length
    a(i)=L{i}.A;
    % 'd' is link offset
    d(i)=L{i}.D;
    % 'off' is the joint offset angle ('alpha')
    off(i)=L{i}.theta;
end;

% If 'iguess' is not specified, a zero angle vector is assumed
if nargin==2
    iguess=[0 0 0 0 0 0];
end;

% The last joint angle, which is fictitous, will be added to the vector of
% initial guess
iguess=[iguess 0];

% The initial guess is input in degrees, so it needs to be converted to
% radians
iguess=iguess*pi/180;
```

```
% The initial guess is changed by the corresponding offset angle
for i=1:length(iguess)
    iguess(i)=iguess(i)+off(i);
end;

% The 'r' elements below are the element of the 3x3 rotation matrix, which
% is a sub matrix of homogenous transformation 'T'

r11=T(1,1);
r12=T(1,2);
r13=T(1,3);

r21=T(2,1);
r22=T(2,2);
r23=T(2,3);

r31=T(3,1);
r32=T(3,2);
r33=T(3,3);

% The p vector is the position vector of the Tool Frame w.r.t the Base
% Frame
px=T(1,4);
py=T(2,4);
pz=T(3,4);

% Solve for Joint 1
a1=-py+d(7)*r23;
b1=-d(7)*r13+px;
c1=d(3);

% Joint 1 will have two solutions
% The two solutions are arranges in a vector of 1x2
th1=[atan2(b1,a1)-atan2(sqrt(a1^2+b1^2-c1^2),c1)...
     atan2(b1,a1)+atan2(sqrt(a1^2+b1^2-c1^2),c1)];

% The solutions are compared with the initial guess
% The solution closest to the initial guess is picked
if abs(th1(1)-iguess(1))<abs(th1(2)-iguess(1))
    th1=th1(1);
else
    th1=th1(2);
end;

% Solve for Joint 3
a3=-(cos(th1)*r13+sin(th1)*r23)*d(7)+cos(th1)*px+sin(th1)*py;
b3=-r33*d(7)+pz-d(1);

% Calculate the sine and cosine of Joint 3
s3=(d(4)^2+a(3)^2-a3^2-b3^2)/(2*d(4)*a(3));
c3=sqrt(1-s3^2);

% Consequently, Joint 3 has two solutions, depending on the value of the
% cosine
th3=[atan2(s3,c3) atan2(s3,-c3)];

% Again, the solution closer to the initial guess is picked
if abs(th3(1)-iguess(3))<abs(th3(2)-iguess(3))
    th3=th3(1);
else
    th3=th3(2);
end;
```

```
% Solve for (Joint 2 + Joint 3)
a23=cos(th1)*(-r13*d(7)+px)+sin(th1)*(-r23*d(7)+py);
b23=-r33*d(7)+pz-d(1);

% The cosine and sine of (Joint 2+Joint 3) are obtained by solving the
% linear system noted below
cos_sin=inv([a23 b23;b23 -a23])*[a(3)*cos(th3);d(4)-a(3)*sin(th3)];

% And a unique solution exists for (Joint 2+Joint 3)
th23=atan2(cos_sin(2),cos_sin(1));

% Solve for Joint 2
th2=th23-th3;

% Solve for Joint 4
% First obtain the value of the cosine and sine of Joint 4
c4=-cos(th1)*cos(th23)*r13-sin(th1)*cos(th23)*r23-sin(th23)*r33;
s4=sin(th1)*r13-cos(th1)*r23;

%----------------------------SINGULARITY CHECK----------------------------%
%
% Check for singularity
%
% If 'c4' and 's4' are close to zero, then Angle 5 is close to zero
%
% When Angle 5 is close to zero, then the axis of Joint 4 and the axis of
% Joint 6 will be alligned
%
% This alignment will cause the rotation of Joint 4 and of Joint 6 to
% cause the same motion of the end effector
%
% If the singularity is detected, then Joint 4 is fixed to the initial
% guess
if abs(c4)<tol && abs(s4)<tol
    th4=iguess(4);
else
    th4=atan2(s4,c4);
end;
%
%------------------------------------------------------------------------%

% Solve for Joint 5
% Obtain the values of the cosine and sine of Joint 5
s5=-(cos(th4)*cos(th1)*cos(th23)-sin(th4)*sin(th1))*r13-(cos(th4)*sin(th1)*...
    cos(th23)+sin(th4)*cos(th1))*r23-cos(th4)*sin(th23)*r33;
c5=-cos(th1)*sin(th23)*r13-sin(th1)*sin(th23)*r23+cos(th23)*r33;

% Hence, Joint 5 can be obtained by using the atan2 function
th5=atan2(s5,c5);

% Solve for Angle 6
% Obtain the values of the cosine and sine of Joint 6
s6=(-sin(th4)*cos(th1)*cos(th23)-cos(th4)*sin(th1))*r11+(-
sin(th4)*sin(th1)*cos(th23)...
    +cos(th4)*cos(th1))*r21-sin(th4)*sin(th23)*r31;
c6=(-sin(th4)*cos(th1)*cos(th23)-cos(th4)*sin(th1))*r12+(-
sin(th4)*sin(th1)*cos(th23)+...
    cos(th4)*cos(th1))*r22-sin(th4)*sin(th23)*r32;

% Again, Joint 6 can be obtained by using the atan2 function
th6=atan2(s6,c6);

%-------------------------REACHABILE SPACE CHECK-----------------------%
```

```
%
% Define vector of sine and cosine values
v_sincos=[s3 c3 cos_sin' c4 s4 c5 s5 c6 s6];

% Check if any one of them is outside the range of -1 and 1
% space_check is the flag used in the following subroutine
for i=1:length(v_sincos)
    if abs(v_sincos(i))>1
        space_check=false;
        break;
    else
        space_check=true;
    end;
end;

% Final result that will be displayed is 'q'
% The angles are offseted again so they can be used in MSC.visualNastran
if space_check==true
    a=[th1-off(1) th2-off(1) th3+off(1) th4 th5 th6]*180/pi;
    for i=1:length(a)
        if a(i)>180
            a(i)=a(i)-360;
        elseif a(i)<-180
            a(i)=a(i)+360;
        end;
    end;
    q=a;
else
    error('Sorry, please try to command the manipulator within its workspace');
end;
```

B.4 inverseEuler.m

```
%-------------------------------------------------------------------------%
% Jamil M. Renno
%
% Thesis - VIRTUAL DESING AND MODELING OF VARIOUS MANUFACTURING PROCESSES
%          FOR REMOTE FABRICATION OF TRANSMUTER FUEL
%
%
% Department of Mechanical Engineering
% University of Nevada, Las Vegas
%
% Revision 0 - September 14, 2003
%
%-------------------------------------------------------------------------%
%
% inverseEuler - Solves for the three rotation angles about the XYZ axes in
%                order to attain a give rotation matrix
%
%------------------------------ASSUMPTIONS------------------------------%
%
% All the angles are in radians
%
%-------------------------------------------------------------------------%
%
% The purpose of the this function is to solve for the Euler Angles
% When a rotation matrix is specified, we are usually interested in the
% corresponding Euler Angles
```

```
%
% The Euler Angles solved for here are the XYZ Euler angles as described in
% various literature
%
% function a=inverseEuler(r)
% 'r' is 3x3 rotation matrix or 4x4 homogenous transformation

function a=inverseEuler(r)

% The symbols b, al, and c stand for beta, alpha and gamma respectively
% The symbols s and c stand for sine and cosine respectively

% Example: s_b is sine(beta)
%          c_al is cosine(alpha)

% alpha is the rotation around the x-axis
% beta is the rotation around the y-axis
% gamma is the rotation around the z axis

% The symbols s and c are for the sine and cosine respectively

% The solution will degenerate in some cases

% In order to identify such cases, we will define a tolerance value of
% 0.0001
tol=0.0001;

s_b=r(1,3);

if abs(s_b)<tol
    % If the solution degenerates, we will assign arbitrary values for beta
    % and alpha
    b=pi/2;
    al=0;
    % Angle theta will solved for accordingly
    c=atan2(r(3,2),-r(3,1));
else
    % If the solution did not degenerate, then the following solution is
    % adopted
    c_b=sqrt(r(2,3)^2+r(3,3)^2);
    b=atan2(s_b,c_b);

    c_c=r(1,1)/c_b;
    s_c=-r(1,2)/c_b;
    c=atan2(s_c,c_c);

    s_al=-r(2,3)/c_b;
    c_al=r(3,3)/c_b;
    al=atan2(s_al,c_al);
end;

% The function will return a vector of the three angles

% The angles are ordered as: alpha beta gamma
a=[al b c]*180/pi;
```

B.5 SingleRotation.m

```
%-------------------------------------------------------------------------%
% Jamil M. Renno
```

```
%
% Thesis - VIRTUAL DESING AND MODELING OF VARIOUS MANUFACTURING PROCESSES
%          FOR REMOTE FABRICATION OF TRANSMUTER FUEL
%
%
% Department of Mechanical Engineering
% University of Nevada, Las Vegas
%
% Revision 0 - September 10, 2003
%
%-------------------------------------------------------------------------%
%
% SingleRotation - Solves for the rotation angle and rotation vector
%                  corresponding to the rotation of a frame
%
%-----------------------------ASSUMPTIONS---------------------------------%
%
% All the angles are in radians
%
%-------------------------------------------------------------------------%
% The program calculates the XYZ Equivalent Angle-Axis Rotation Matrix
% Then the program performs the inverse solution
% This is the modified version of the initial program
% The program was modified in order to allow it to behave as a function
% when planning Cartesian trajectories, base on the single rotation
% method
%
% function [k,th]=SingleRotation(T)
% 'k' is the general rotation vector
% 'th' is the angle of rotation
% 'T' is the transformation matrix that need to be assumed using the
% Euler Angles Representation

function [k_sol,theta_sol]=SingleRotation(T)
%---Define a tolerance value to avoid solving an ill conditioned system
tol=10^-3;

% Arrange the rotation matrix elements in easy to access variables
r11=T(1,1);
r12=T(1,2);
r13=T(1,3);

r21=T(2,1);
r22=T(2,2);
r23=T(2,3);

r31=T(3,1);
r32=T(3,2);
r33=T(3,3);

%---Solve for theta
th=acos((r11+r22+r33-1)/2)*180/pi;

%---Check for sin(theta)
if abs(r11+r22+r33-3)<=tol
    th=0;
    % If the rotation angle is zero, then the rotation vector can be any
    % vector since no rotation will be performed
    k=[rand rand rand];
    % Normalize the rotation vector, so that it's magnitued is one
    if norm(k)~=1
        k_u(1)=k(1)/norm(k);
        k_u(2)=k(2)/norm(k);
```

```
        k_u(3)=k(3)/norm(k);
        k=k_u;
    end;
    % Now check for the value of (r11+r22+r33)
elseif abs(r11+r22+r33+1)<=tol
    th=180;
    for i=1:3
        for j=i
            if abs(T(i,j)+1)>=tol
                index=i;
            end;
        end;
    end;
    % The diagonal element which is farthest from -1 will determine the
    % solution in this case
    switch index
        case 1
            k(1)=sqrt((r11+1)/2);
            k(2)=r21/(2*k(1));
            k(3)=r31/(2*k(1));
        case 2
            k(2)=sqrt((r22+1)/2);
            k(1)=r12/(2*k(2));
            k(3)=r32/(2*k(2));
        case 3
            k(3)=sqrt((r33+1)/2);
            k(1)=r13/(2*k(3));
            k(2)=r23/(2*k(3));
    end;
else
    % Otherwise, k the sin of theta can be used in the finding the value of
    % the rotation vector k
    k=[(r32-r23)/(2*sin(th*pi/180)) (r13-r31)/(2*sin(th*pi/180)) (r21-
r12)/(2*sin(th*pi/180))];
end;

% Output the solution
k_sol=k';
theta_sol=th;
```

B.6 move2bin.m

```
%-------------------------------------------------------------------------%
% Jamil M. Renno
%
% Thesis - VIRTUAL DESING AND MODELING OF VARIOUS MANUFACTURING PROCESSES
%          FOR REMOTE FABRICATION OF TRANSMUTER FUEL
%
%
% Department of Mechanical Engineering
% University of Nevada, Las Vegas
%
% Revision 0 - September 15, 2003
%
%-------------------------------------------------------------------------%
%
% move2bin - Creates the joing trajectories to move the pellets to the bin
%
%-------------------------------------------------------------------------%
%
% This function generates the joints commands and the finger commands
% necessary move each pellet from the shoot to the bin put on a converyor belt
```

```
%
% The function takes three arguments
%
% r is the robot object, which contains the DH table
%
% p is the number of the pellet
% The number of the pellet will allow passing very important variables like
% the position and orientation of the pellet when picked up and when dropped
% of in the bin

% JS is a joint set. This joint set will serve as the initial starting
% joint set for the motion

% Basically, we are the arguments we are passing to this function are:
%
% The robot that should moved (r)
%
% The position from where the motion will end (p is the number of the
% pellet, which will provide the position and orientation of the pellet end
% position in 3D space)
%
% The set of angles will provide the start positon of the end effector motion
function y=MVLOAD_BIN(r,p,JS)

% This is the homogenous transformation of the each pellet (loaded as three
dimensional array)
global pellet;
% This is the homogenous transformation of the end effector above the position
of the pellet
global lift;
% This is the homogenous transformation of the end effector before dropping the
pellet
global drop;
% This is the homogenous transformation of the end effector when dropping the
pellet
global lowerp;
% This is the time step for generating the joint angles
global tstep;
% This is the time to wait when picking or dropping a pellet
global twait;
% This is the time to carry a motion in
global tf;
% This is the time vector for doing the motion
global T;
% This is the time vector for waiting
global Twait;
% This is the vector of commands to the fingers when they are open
global finger_open;
% This is the vector of commands to the fingers when they are closed
global finger_close;

if size(p)==[1 1]
    % JSL is the joint set where the end effector is above the shoot, right
    % above the pellet
    if p==1
        JSL=hcinv(r,lift(:,:,p),JS);
    else
        %JSL=hcinv(r,lift(:,:,p),[-70 93 38 180 47 0]);
        JSL=hcinv(r,lift(:,:,p),[70 90 90 180 -47 0]);
    end;

    % JSP is the joint set when the end effector engages the pellet
    JSP=hcinv(r,pellet(:,:,p),JSL);
```

```
    % JSLO is the joint set when the end effector is ready to release the
    % pellet
    JSLO=hcinv(r,lowerp(:,:,p),JSP);

    traj=[jtrajs([JS;JSL;JSP],T);   % Move to above the pellet
          jtraj(JSP,JSP,Twait)  ;
          pick(r,p,[0:tstep:5]) ;
          jtraj(JSLO,JSLO,Twait)];

    % This is the gripper finger's command through moving a pellet

    fcomm=[jtraj(finger_open,finger_open,T);      % Grippers opened
           jtraj(finger_open,finger_open,T);      % Grippers still open ( pick
the pellet)
           jtraj(finger_open,finger_close,Twait); % Grippers close
           jtraj(finger_close,finger_close,[0:tstep:5]);    % Grippers still
closed
           jtraj(finger_close,finger_close,[0:tstep:5]);    % Grippers still
closed
           jtraj(finger_close,finger_close,[0:tstep:5]);    % Grippers still
closed
           jtraj(finger_close,finger_open,Twait)];% Grippers open ( release the
pellet)
else
    traj=[jtraj(JS,p,T)];
    fcomm=[jtraj(finger_open,finger_open,T)];
end;

% The function returns 8 columns

% The first two columns are fingers' commands
% The rest 6 columns are the the joints commands
y=[fcomm traj];
```

B.7 pick.m

```
%-------------------------------------------------------------------------%
% Jamil M. Renno
%
% Thesis - VIRTUAL DESING AND MODELING OF VARIOUS MANUFACTURING PROCESSES
%          FOR REMOTE FABRICATION OF TRANSMUTER FUEL
%
%
% Department of Mechanical Engineering
% University of Nevada, Las Vegas
%
% Revision 0 - December 20, 2003
%
%-------------------------------------------------------------------------%
%
% pick - implements a Cartesian trajectory scheme to move the pellets from
%        the shoot to the bin placed on the conveyor belt
%
%-------------------------------------------------------------------------%
%
% pick(r,p,T)
% r is the robot object
% p is the number of the pellet
% T is the time vector over which portions of the motion will take place

function result=pick(r,p,T)
```

```
% pellet, lift, drop and lowerp were previously declared as global
% matrices
global pellet lift drop lowerp

% Solve for the inverse kinematics in order to determine the initial
% configuration of the manipulator
iconfig=hcinv(r,pellet(:,:,p),[70 60 90 0 47 0]);

% Store the values of the points in space as well as the via points
Ti0=pellet(:,:,p);
T1=lift(:,:,p);
T2=drop(:,:,p);
Tf0=lowerp(:,:,p);

% Extract the position vectors from the homogenous transformations above
xyz_coor=[Ti0(1:3,4)';
          T1(1:3,4)';
          T2(1:3,4)';
          Tf0(1:3,4)'];

% Generate the trajectory of the space points as noted in the singe
% rotation method
xyz=jtrajs(xyz_coor,T);

% Start implement the single rotation method
Tfi=inv(Ti0)*Tf0;

% Extract the 3x3 final configuration rotation matrix
Rfi=Tfi(1:3,1:3);

% Solve for the single rotation method angle and rotation vector
[k,thr]=ktmodi(Rfi);

% Generate a trajectory for the angle specified in the above line
thr=jtraj(0,thr,size(xyz,1));

% Transform the xyz trajectory from the base frame to the local initial
% tool frame
for i=1:size(xyz_coor,1)
    xyz_coor_tool(i,:)=(inv(Ti0(1:3,1:3))*(xyz_coor(i,:)-xyz_coor(1,:))')';
end;

% Again generate the trajectory
xyzp=jtrajs(xyz_coor_tool,T);

% Determine the successive transformations of the tool frame
for i=1:size(xyz,1)
    % D is the intermediate description of the tool frame w.r.t its initial
configuration
    D(:,:,i)=[[rkt(k,thr(i));0 0 0] [xyzp(i,:)';1]];
    % D is transformed into the base frame to allow the solving of the
    % inverse kinematics
    A(:,:,i)=Ti0*D(:,:,i);
    % At the begining, we are at the initial configuaration
    if i==1
        q(i,:)=iconfig;
    else
        % We are utilizing the other argument in the inverse kinematics
        % function, allowing for a smooth angle path
        q(i,:)=hcinv(r,A(:,:,i),q(i-1,:));
    end;
end;
```

```
% Output the result in joint angles

result=q;
```

B.8 HotCellOperate.m

```
%-------------------------------------------------------------------------%
% Jamil M. Renno
%
% Thesis - VIRTUAL DESING AND MODELING OF VARIOUS MANUFACTURING PROCESSES
%          FOR REMOTE FABRICATION OF TRANSMUTER FUEL
%
%
% Department of Mechanical Engineering
% University of Nevada, Las Vegas
%
% Revision 0 - December 20, 2003
%
%-------------------------------------------------------------------------%
%
% HotCellOperate - generate the necessary commands to operate the hot cell
%
%-------------------------------------------------------------------------%
% This program was written by Jamil M. Renno
% This program outputs the commands necessary for the file
% "LOAD_BIN_1.wmd" to run
%
% See also move2bin, pelletpositions, telbot, joint_cong, neut

% Clear the screen
clc;

% Clear the workspace from all the variables
clear all;

% This is the robot object (defined in telbot.m)
global r;
% This is the homogenous transformation of the each pellet (loaded as a
three dimensional array)
global pellet;
% This is the homogenous transformation of the end effector above the
position of the pellet
global lift;
% This is the homogenous transformation of the end effector before
dropping the pellet
global drop;
% This is the homogenous transformation of the end effector when dropping
the pellet
global lowerp;
% This is the time step for generating the joint angles
global tstep;
% This is the time to wait when picking or dropping a pellet
global twait;
% This is the time to carry a motion in
global tf;
% This is the time vector for doing the motion
global T;
% This is the time vector for waiting
```

```
global Twait;
% This is the vector of commands to the fingers when they are open
global finger_open;
% This is the vector of commands to the fingers when they are closed
global finger_close;

% Load the position and orientation of the pellets into MATLAB workspace
% When this m.file runs, all the positions of the pellets, the
orientation of
% the end effector when engaging them, are loaded in the workspace
pelletpositions;

% Load the robot objects into MATLAB workspace
% The robot objects are Matlab objects that are defined by they DH table
% created by the robotics toolbox
telbot;

% Define the time vectors to be used in the simulation
% We will define all the time parametes needed in the generation of the
% commands to drive the hot cell

% This is the time step when generatin the data
tstep=0.01;

% This is a small time interval
% It will be used when the gripper is openning its fingers, dropping a
pellet, and such
twait=tstep*5;

% This is the time interval given to perform a motion of the robot tool
tf=4;

% Define the time vectors
% All the spacing between the time instances is the same
% tstep was defined above
T=[0:tstep:tf];
Twait=linspace(0,twait,length(T));

% Declare the number of pellets to be picked and moved in the work cell
num_pellets=5;

% JS0, JS1 and JS2 are hand picked joint sets to ensure that the end
effector
% would not hit any of the equipment in the hot cell
% Those joint sets will insure that the end effector will move in a
% trajectory free cartesian space

% JS0 is the initial joint set of the robot
% JS0 is the zero position of the robot
JS0=[0 0 0 0 0 0];

% JS1 and JS2 are two intermediate joint sets
% These joint sets are used to drive the robot through a path that will be
% obstacle free
JS1=[80 30 30 0 0 0];
JS2=[70 60 90 0 25 0];

% This trajectory is to get the manipulator's end effector from its
initial
% position to a position where it is near the pellet
```

```
ready_joints=jtrajs([JS0;JS1;JS2],T);

% This is the distance of the sliders of the grippers when open
% fio stands for "finger open"
finger_open=[-2.3 2.3]*0.01;

% This is the distance of the sliders of the gripper when closed
% fic stands for "fingerclosed"
finger_close=[-2.8 2.8]*0.01;

% This is the fingers command before getting to the first pellet
% Before getting to the pellets, the fingers will stay open
ready_finger=[jtraj(finger_open,finger_open,T);
              jtraj(finger_open,finger_open,T)];

% This is the rigid constraint command before getting to the first pellet
% The command is zero now, so the constraint is inactive
ready_constraint=zeros(size(ready_joints,1),num_pellets);

% This is the conveyor belt command
% The conveyor belt is not moving now
ready_conveyor=zeros(size(ready_joints,1),1);

% "ready" is the command of the robot (joints, fingers, constriants) before
% getting to the first pellet
% The first 6 columns are readyrc, which means that the gripper is not
holding any pellet
ready=[ready_constraint ready_finger ready_joints];

% Now the robot will move each pellet from the shoot to the bin on the
% conveyor belt
% This will include the fingers and the joint commands

finger_joint=[MVLOAD_BIN(r,1,JS2);
              MVLOAD_BIN(r,2,hcinv(r,lowerp(:,:,1),JS2));
              MVLOAD_BIN(r,3,hcinv(r,lowerp(:,:,2),JS2));
              MVLOAD_BIN(r,4,hcinv(r,lowerp(:,:,3),JS2));
              MVLOAD_BIN(r,5,hcinv(r,lowerp(:,:,4),JS2))];

% This is the trajectory that will bring the robot arm back to its original
% position after placing all the pellets in the bin placed on the conveyor
% belt
go_home=[MVLOAD_BIN(r,JS2,finger_joint(size(finger_joint,1),3:8));
         MVLOAD_BIN(r,JS1,JS2)                                        ;
         MVLOAD_BIN(r,JS0,JS1)                                      ];

% Here is the command for the first robot
% The commands are added to the constraint command
% The constraint command comes from the join_con function

% Add the constraint command to the command matrix
constraint_finger_joint=[[join_cong(finger_joint(:,1:2),num_pellets);
                          join_cong(go_home(:,1:2),num_pellets)]
[finger_joint;go_home]];

% The complete command of the first robot will be
complete_motion=[ready                    ;
                 constraint_finger_joint];

% Conveyor Command
```

```
% This is the time for conveying the bin through the oven
% This parameter is determined by the user solely
convey_time=10;

% Conveying Distance
% This is the distance that the bin will travel on the conveyor belt
convey_distance=80-350.31;

% Conveying Speed
% We are assuming a constant speed throught the sintering process
(conveying)
convey_speed=convey_distance/convey_time;

% The speed is negative to drive the bin inside the cintering oven
convey=jtraj(convey_speed,convey_speed,[0:tstep:convey_time]);

% Combine all the commands in one array
command=neut(complete_motion,convey,1);

% ta_motion is the time where the every thing in the work cell is moving
ta_motion=size(command,1)*tstep;

% This is the simulation time that we want to simulate
t_sim=60;

% After the motion is done, we will zero all the commands
command=[command;
         zeros((t_sim-ta_motion)/tstep,size(command,2))];

% Create the complete vector
% Vector should be equal to the commands created before
nsteps=size(command,1);

% Specify a time step for the motion to take place
t=linspace(0,t_sim,nsteps)';

% Set command with time vector to be used in Simulink
a=[t command];
```

B.9 push.m

```
%-------------------------------------------------------------------------%
% Jamil M. Renno
%
% Thesis - VIRTUAL DESING AND MODELING OF VARIOUS MANUFACTURING PROCESSES
%          FOR REMOTE FABRICATION OF TRANSMUTER FUEL
%
%
% Department of Mechanical Engineering
% University of Nevada, Las Vegas
%
% Revision 0 - December 25, 2003
%
%-------------------------------------------------------------------------%
%
% push - generate the necessary commands to push the pellets inside the
%        cladding tube
%
%-------------------------------------------------------------------------%
%
```

```
% This function just generates the joint angles of the manipulator in order
% for the end effector to move in a straight line and hence push the
% pellets in the V-Tray inside the cladding tube
function p=push(r,strt,fnsh,orien,t)

% The number of steps in the length of the time vector given
nstep=length(t);

% Steps in X direction is a smooth function
xstep=jtraj(strt(1),fnsh(1),nstep);

% Steps in the Y direction is a smooth function too
ystep=jtraj(strt(2),fnsh(2),nstep);

% Steps in the Z direction is a smooth function too
zstep=jtraj(strt(3),fnsh(3),nstep);

% Establish the first transformation angles
A(:,:,1)=[orien [xstep(1);ystep(1);zstep(1)];0 0 0 1];
% Solve for the corresponding angles in order to use it in determining an
% initial guess for the next joint angles

q(1,:)=hcinv(r,A(:,:,1),[0 0 0 0 0 0]);

% Generate the rest of the angles necessary to accomplish the path
for i=2:nstep
    A(:,:,i)=[orien [xstep(i);ystep(i);zstep(i)];0 0 0 1];
    q(i,:)=hcinv(r,A(:,:,i),q(i-1,:));

end;

% Return the output of the function
p=q;
```

B.10 neut.m

```
%-------------------------------------------------------------------------%
% Jamil M. Renno
%
% Thesis - VIRTUAL DESING AND MODELING OF VARIOUS MANUFACTURING PROCESSES
%          FOR REMOTE FABRICATION OF TRANSMUTER FUEL
%
% Department of Mechanical Engineering
% University of Nevada, Las Vegas
%
% Revision 0 - October 22, 2003
%
%-------------------------------------------------------------------------%
%
% neut - neutralizes the vector in the sense of adding zeros and ones for
%        the grippers to operate
%
%-------------------------------------------------------------------------%
%
% The function can do rearranging for two vectors so that one will be
% active, and the other will be zero at the same time

function y=neut(u,v,op)

% Fill the first argument with zeros while the second argument will be
full
```

```
% of numbers
z1=[u; zeros(size(v,1),size(u,2))];

% Fill the second argument with zeros while the first argument is active
z2=[zeros(size(u,1),size(v,2));v];

% The third argument determines how we want the final result to look like
if op==0
    y=[z1 z2];
else
    y=[z2 z1];
end;
```

BIBLIOGRAPHY

[1] Denavit, J., Hartenberg, R.S., 1955, "A Kinematic Notation for Lower-Pair Mechanisms Based Upon Matrices," *Journal of Applied Mechanics*, Vol. 77, pp. 215-221.

[2] Benedict, M., Pigford, T.H., 1957, *Nuclear Chemical Engineering*, McGraw Hill, New York.

[3] Kaufmann, A.R., 1962, *Nuclear Reactor Fuel Elements—Metallurgy and Fabrication*, John Wiley & Sons, London, New York.

[4] Symon, K.R., *Mechanics*, 1971, 3rd Edition, Addison-Wesley, Reading, Massachusetts.

[5] Bowen, W.W., Sherrell, D.L., Wiemers, M.J., 1981, *Automated Handling for SAF Batch Furnace and Chemistry Analysis Operations*, Hanford Engineering Development Laboratory, Portland, Oregon, HEDL-SA—2414-FP.

[6] Paul, R.P., 1981, *Robot Manipulators: Mathematics, Programming, and Control: The Computer Control of Robot Manipulators*, MIT Press, Cambridge, Massachusetts.

[7] Frederickson, J.R., Horgos R.M., 1983, *SAF Line Powder Operations*, Hanford Engineering Development Laboratory, Portland, Oregon, HEDL-SA—2894-FP.

[8] Geber, E.W., Bowen, W.W., Williams, J.A., 1983, *SAF Line Furnace Operations*, Hanford Engineering Development Laboratory, Portland, Oregon, HEDL-Sa—2891-FP.

[9] Egli, W., Bogart, R.L., 1983, *Sintering Boat Transport System for the SAF Line*, Hanford Engineering Development Laboratory, Portland, Oregon, HEDL-SA—2906-FP.

[10] Jedlovec, D.R., Bowen, W.W., Brown, R.L., 1983, *SAF Line Pellet Gauging*, Hanford Engineering Development Laboratory, Portland, Oregon, HEDL-SA—2903-FP.

[11] Gerber E.W., Hoitink, N.C., Graham, R.A., 1984, *Remote Fabrication of Breeder Reactor Fuel*, Hanford Engineering Development Laboratories, Portland, Oregon, HEDL-SA—3157-FP.

[12] Goldenberg, A., Benhabib, B., Fenton, R., March 1985, "A Complete Generalized Solution to the Inverse Kinematics of Robots," *IEEE Journal of Robotics and Automation*, Vol. 1, Issue 1, pp. 14-20.

[13] Avallone, E.A., Baumeister III, T., 1987, *Mark's Standard Handbook for Mechanical Engineers*, 10th Edition, McGraw Hill, New York.

[14] Frederickson, J.R., Eshenbaum, R.C., Goldmann, L.H., 1987, *Powder Handling for Automated Fuel Processing*, Hanford Engineering Development Laboratory, Portland, Oregon, HEDL-SA—3690-FP.

[15] Sciavicco, L., Siciliano, B., August 1988, "A Solution Algorithm to the Inverse Kinematic Problem for Redundant Manipulators," *IEEE Journal of Robotics and Automation*, Vol. 4, Issue 4, pp. 403-410.

[16] Krueger, A.J., August 1989, "Three Dimensional Computerized Model of an Elastic Robot Arm," M.S. Thesis, Department of Civil and Mechanical Engineering, University of Nevada, Las Vegas, The United States of America.

[17] Koivo, A., Antti, J., 1989, *Fundamentals for Control of Robotic Manipulators*, John Wiley & Sons, London, New York.

[18] Craig, J.J., 1989, *Introduction to Robotics: Mechanics and Control*, 2nd Edition, Addison-Wesley Longman Publishing Co., Boston, Massachusetts.

[19] Yoshikawa, T., 1990, *Foundations of Robotics: Analysis and Control*, The MIT Press, Boston, Massachusetts.

[20] Wang, L.-C.T., Chen, C.C., August 1991, "A Combined Optimization Method for Solving the Inverse Kinematics Problems of Mechanical Manipulators," *IEEE Transactions on Robotics and Automation*, Vol. 7, Issue 4, pp. 489-499.

[21] Klein, C.A., Chu-Jeng, C., Kittivatcharapong, S., October 1991, "Testing Iterative Robotic Algorithms by their Rate of Convergence," *IEEE Transactions on Robotics and Automation*, Vol. 7, Issue 5, pp. 686-687.

[22] Vidyasagar, M., Sankaranarayanan, A., 1991, "Path Planning for a Point Object Amidst Unknown Obstacles in a Plane: The Universal Lower Bound on Worst Case Path Lengths and a Classifications of Algorithms," *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 2.

[23] Lumelsky, V., Evans, J., Krishnamurth, B., Barrows, B., Skewis, T., 1992, "Handling Real-World Motion Planning: A Hospital Transport Robot," *IEEE Control Systems Magazine*, Vol. 12, Issue 1.

[24] Grudic, G.Z., Lawrence, P.D., August 1993, "Iterative Inverse Kinematics with Manipulator Configuration Control," *Transactions on Robotics and Automation*, Vol. 9, Issue 4, pp. 476-483.

[25] Bennett, P.C., Stringer, J.B., May 1994, "Monitored Retrievable Storage/ Multi-Purpose Canister Analysis: Simulation and Economics of Automation," *Proceedings of the 5th Annual International Conference of High Level Radioactive Waste Management*, Vol. 2, Las Vegas, Nevada.

[26] Moanocha, D., Canny, J.F., October 1994, "Efficient Inverse Kinematics for General 6R Manipulators," *IEEE Transactions on Robotics and Automation*, Vol. 10, Issue 5, pp. 648-657.

[27] Jaquish, W.R., October 1994, "PUREX Irradiated Fuel Recovery Simulation," *Deneb User Group Meeting*, Detroit, Michigan, U.S.A., DOE Project No. WHC-SA-2664, CONF-9410231—1.

[28] O'Donnell, W.M., December 1995, "Automated Robotics for Nuclear Waste Handling (Hardware and Software)," M.S. Thesis, Department of Electrical and Computer Engineering, University of Nevada, Las Vegas, The United States of America.

[29] Kreutz-Delgado, K., Agahi, D., December 1995, "A Recursive Singularity-Robust Jacobian Generalized Inverse," *IEEE Transactions on Robotics and Automation*, Vol. 11, Issue 6, pp. 887-892.

[30] Corke, P.I., March 1996, "A Robotics Toolbox for MATLAB," *IEEE Robotics and Automation Magazine*, Vol. 3, Issue 1, pp. 24-32.

[31] Lloyd, J.E., Hayward, V., April 1996, "A Discrete Algorithm for Fixed-Path Trajectory Generation at Kinematic Singularities," *IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota.

[32] Park, J., Chung, W.-K., Youm, Y., June 1996, "Characteristics of Optimal Solutions of Redundancy," *IEEE Transactions on Robotics and Automation*, Vol. 12, Issue 3, pp. 471-478.

[33] Lee, H.Y., Wälischmiller, W., July 1996, "Development and Application of the TELBOT System—A New Tele Robot System," *11th CISM-IFToMM Symposium on Theory and Practice of Robots and Manipulators*, Udine, Italy.

[34] Robinson, D.G., Atcitty, C.B., August 1996, "Safety Assessment of High Consequence Robotics Systems," *National System Safety Conference,* Albuquerque, New Mexico, No. SAND—96-2014C, CONF-960869—15.

[35] Committee on Separation Technology and Transmutation Systems, Board on Radioactive Waste Management, Commission on Geosciences, Environment, and Resources, National Research Council, 1996, *Nuclear Waste: Technologies for Separation and Transmutation*, National Academy Press, Washington D.C.

[36] Fu, K.S., Gonzales, C.S.G., Lee, 1996, *Robotics: Control, Sensing, Vision, and Intelligence*, McGraw Hill, New York.

[37] Drotning, W., Kimberly, H., Wapman, W., Darras, D., April 1997, "A Sensor-Based Automation System for Handling Nuclear Materials," *International Conference on Robotics and Automation*, Albuquerque, New Mexico, No. SAND—96-2303C, CONF-970469—5.

[38] Bousquet, G., Brähler, G., September 1997, " A MOX Fuel Plant in Russia: The Engineering Work Has Started," *The Uranium Institute, Twenty-Second Annual Symposium*, London, UK.

[39] Kats, V., Levner, E., April 1998, "Cyclic Scheduling of Operations for a Part Type in an FMS Handled by a Single Robot: A Parametric Critical-Path Approach," *International Journal of Flexible Manufacturing Systems*, Vol. 10, Issue 2, pp. 129-138.

[40] Chen, T., May, 1998, "Robotics for Nuclear Waste Handling," M.S. Thesis, Department of Electrical and Computer Engineering, University of Nevada, Las Vegas, The United States of America.

[41] Blomeke, J.O.A., June 1998, *Review and Analysis of European Industrial Experience in Handling LWR Spent Fuel and Vitrified High-Level Waste*, Oak Ridge National Laboratory, Office of Civilian Radioactive Waste Management, Oak Ridge, Tennessee, No. ORNL/TM-10696.

[42] Janabi-Sharifi, F., Wilson, W.J., July 1999, "A Fast Approach for Robot Motion Planning," *Journal of Intelligent and Robotic Systems*, Vol. 25, Issue 3, pp. 187-212.

[43] Ahuactzin, J.M., Gupta, K.K., August 1999, "The Kinematic Roadmap: A Motion Planning Based Approach for Inverse Kinematics of Redundant Robots," *IEEE Transactions on Robotics and Automation*, Vol. 15, Issue 4, pp. 653-669.

[44] U.S. Department of Energy, 1999, *Design Considerations*, National Technical Information Service, Springfield, Virginia, DOE HDBK-1132-99.

[45] Nof, S.Y., 1999, *Handbook of Industrial Robotics*, 2nd Edition, John Wiley & Sons, London, New York.

[46] Haas, D., Somers, J., Renard, A., La Fuente, A., 1999, "Feasibility of the Fabrication of Americium Targets," *Nuclear Energy Association,* [online] http://www.new.fr/html/trw/docs/mol98/session3/SIIIpaper1.pdf (Accessed December 20, 2004).

[47] Schlotter, A., Pfeiffer, F., April 2000, "Modeling, Control and Optimization of a New Tele Robot Robotics Automation," *Proceedings of The International*

Conference on Robotics & Automation, Vol. 3, pp. 2956-2664, San Francisco, Californian.

[48] Los Alamos National Laboratory (LANL), October 2001, *Addressing the Nuclear Waste Issue*, Los Alamos, New Mexico, No. LALP-01-227.

[49] Nuclear Energy Agency, 2001, *Trends in the Nuclear Fuel Cycle: Economic, Environmental and Social Aspects*, OECD Publication, Paris, France, No. 52329.

[50] Eicker, P., 2001, "A DOE SUCCESS: Automated Nuclear Weapon Component Handling," *The U.S. Department of Energy*, [online] http://www.rim.doe.gov/WALS.pdf, (Accessed December 20, 2004).

[51] Hechanova, A.E., February 2002, *Quarterly Report Fourth Quarter (December 2001 to February 2002)*, University of Nevada, Las Vegas, Advanced Accelerator Applications, University Participation Program, Harry Reid Center for Environmental Studies, Las Vegas.

[52] Renno, J.M., Mauer, G.F., 2004, "Design and Virtual Testing of Robotic Virtual Assembly Processes for Hot Cells," *Conference on Robotics and Remote Systems, Proceedings of The 10th International Conference on Robotics and Remote Systems for Hazardous Environments*, Vol. 10, pp. 516-520, Gainesville, Florida.

[53] National Institute of Standard and Technology, [online] http://www.ceramics.nist.gov/srd/summary/ftguo2.htm, (Accessed December 15, 2004).

[54] The Hyper Physics Website, [online] http://hyperphysics.phy-astr.gsu.edu/hbase/nuclear/radrisk.html, (Accessed December 24, 2004).

[55] Roland, J., "MOX Fuel Fabrication and Its External Communication in France," *COGEMA*, [online] http://www.jnc.go.jp/news/topics/PT021203/pdf/english_original/06_r_jacquet.pdf, (Accessed December 28, 2004).

VITA

Graduate College
University of Nevada, Las Vegas

Jamil Mohamad Renno

Local Address:
4386 Escondido Street
Apartment # 207
Las Vegas, Nevada 89119

Permanent Address:
Building 6, Street 84
Zone 3, Ground Floor
Sahl Al-Sabbagh, Saida
Lebanon

Degree:
Bachelor of Science, Mechanical Engineering, 2003
University of Nevada, Las Vegas

Special Honors and Awards:
Tau Beta Pi – Engineering Honor Society
Phi Kappa Phi Honor Society
Phi Alpha Epsilon – English Honor Society
NSCS – National Society of Collegiate Scholars
The National Dean's List
Best Presentation at the ANS Student Conference, Madison, WI 2004
Best Presentation at the ANS Student Conference, Columbus, OH 2005

Publications:

Renno, J.M., Trabia, M.B., Moustafa, K.A.F., "Anti-Swing Adaptive Fuzzy Controller for an Overhead Crane with Hoisting," *Proceedings of IMECE, ASME International Mechanical Engineering Congress & Exposition*, November 14-19, 2004, Anaheim, CA.

Renno, J.M., Yim, W., Singh, S.N., "End Point Position Control of Multi-Link Flexible Manipulators Using SDRE Method," *Proceedings of IMECE, ASME International Mechanical Engineering Congress & Exposition*, November 14-19, 2004, Anaheim, CA.

Renno, J.M., Mauer, G.F., March 28-31, 2004, "Design and Virtual Testing of Robotic Assembly Processes for Hot Cells," *Proceedings of 10th International Conference on Robotics and Remote Systems for Hazardous Environments*, Gainesville, FL, Volume:10, pp. 516-520.

Renno, J.M., Mauer, G.F., November 16-20, 2003, "Transmuter Fuel Fabrication: Design and Analysis of Robotic Manufacturing Processes," *Transaction of ANS/ENS International Winter Meeting*, New Orleans, LA, Volume:89, pp. 383.

Thesis Title:

Three Dimensional Modeling and Simulation of Manufacturing Processes for Transmuter Fuel Fabrication

Thesis Examination Committee:

Chairperson, Dr. Georg F. Mauer, Ph. D.
Committee Member, Dr. Mohamed B. Trabia, Ph. D.
Committee Member, Dr. Woosoon Yim, Ph. D.
Graduate Faculty Representative, Dr. Moses Karakouzian, Ph. D., P.E.