2

A Framework for Integrated Design of Mechatronic Systems

Kenway Chen, Jonathan Bankston, Jitesh H. Panchal and Dirk Schaefer

The G. W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology, Savannah, GA 31407, USA
Emails: kchen8@gatech.edu, jonathanbankston@gatech.edu, jitesh.panchal@me.gatech.edu, dirk.schaefer@me.gatech.edu

Abstract
Mechatronic systems encompass a wide range of disciplines and hence are collaborative in nature. Currently, the collaborative development of mechatronic systems is inefficient and error-prone because contemporary design environments do not allow sufficient flow of design and manufacturing information across electrical and mechanical domains. Mechatronic systems need to be designed in an integrated fashion allowing designers from both electrical and mechanical engineering domains to receive automated feedback regarding design modifications throughout the design process. Integrated design of mechatronic systems can be facilitated through the integration of mechanical and electrical computer-aided design (CAD) systems. One approach to achieve such integration is through the propagation of constraints. Cross-disciplinary constraints between mechanical and electrical design domains can be classified, represented, modelled, and bi-directionally propagated in order to provide automated feedback to designers of both engineering domains. In this chapter, the authors focus on constraint classification and constraint modelling and provide an example by means of a robot arm. The constraint modelling approach serves as a preliminary concept for the implementation of constraint propagation between mechanical and electrical CAD systems.

2.1 Introduction

Cross-disciplinary integration of mechanical engineering, electrical and electronic engineering as well as recent advances in information engineering are becoming more and more crucial for future collaborative design, manufacture, and maintenance of a wide range of engineering products and processes [2.1]. In order to allow for additional synergy effects in collaborative product creation, designers from all disciplines involved need to adopt new approaches to design, which facilitate concurrent cross-disciplinary collaboration in an integrated fashion. This, in particular, holds true for the concurrent design of mechatronic systems, which is the main focus of this chapter.

Mechatronic systems usually encompass mechanical, electronic, electrical, and software components (Figure 2.1). The design of mechanical components requires a sound understanding of core mechanical engineering subjects, including mechanical devices and engineering mechanics [2.1]. For example, expertise regarding lubricants, heat transfer, vibrations, and fluid mechanics are only a few aspects to be considered for the design of most mechatronic systems. Mechanical devices include simple latches, locks, ratchets, gear drives and wedge devices as well as complex devices such as harmonic drives and crank mechanisms. Engineering mechanics is concerned with the kinematics and dynamics of machine elements. Kinematics determines the position, velocity, and acceleration of machine links. Kinematic analysis is used to find the impact and jerk on a machine element. Dynamic analysis is used to determine torque and force required for the motion of a link in a mechanism. In dynamic analysis, friction and inertia play an important role.

Electronics involves measurement systems, actuators, and power control [2.1]. Measurement systems in general comprise of three elements: sensors, signal conditioners, and display units. A sensor responds to the quantity being measured from the given electrical signal, a signal conditioner takes the signal from the sensor and manipulates it into conditions suitable for display, and in the display unit the output from the signal conditioner is displayed. Actuation systems comprise the elements that are responsible for transforming the output from the control system into the controlling action of a machine or a device. Finally, power electronic devices are important in the control of power-operated devices. The silicon controlled rectifier is an example of a power electronic device that is used to control DC motor drives.

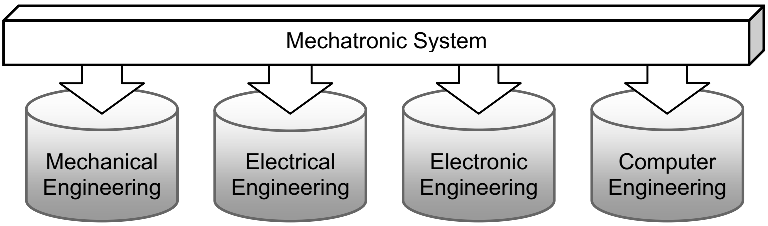


Figure 2.1. The scope of mechatronic system

The electrical aspect of mechatronic systems involves the functional design of electrical plants and control units. This is done through the generation of several types of schematics such as wiring diagrams and ladder diagrams. In addition, programmable logic controllers (PLCs) are widely used as control units for mechatronic systems. PLCs are well adapted to a range of automation tasks. These typically are industrial processes in manufacturing where the cost of developing and maintaining an automation system is relatively high compared to the total cost of the automation, and where changes to the automation system are expected during its operational life.

According to Reeves and Shipman [2.2], discussion about the design must be embedded in the overall design process. The ideal process of concurrent or simultaneous engineering is characterised by parallel work of a potentially distributed community of designers who know about the parallel work of their colleagues and collaborate as necessary. In order to embed the discussion aspect, mechatronic products should be designed in an integrated fashion that allows for designers of both electrical and mechanical engineering domains to automatically receive feedback regarding design modifications made on either side throughout the design process. This means, that if a design modification of a mechanical component of a mechatronic systems will lead to a design modification of an electrical aspect of the mechatronic system or *vice versa*, the engineer working at the counterpart system should be notified as soon as possible.

Obviously, even on the conceptual design level, mechanical and electrical design aspects of mechatronic systems are highly intertwined through a substantial number of constraints existing between their components (see Figure 2.2).

Consequently, in order to integrate mechanical and electrical CAD tools on systems realisation/integration level (see Figure 2.3) into an overarching cross-disciplinary computer-aided engineering (CAE) environment, these constraints have to be identified, understood, modelled, and bi-directionally processed.

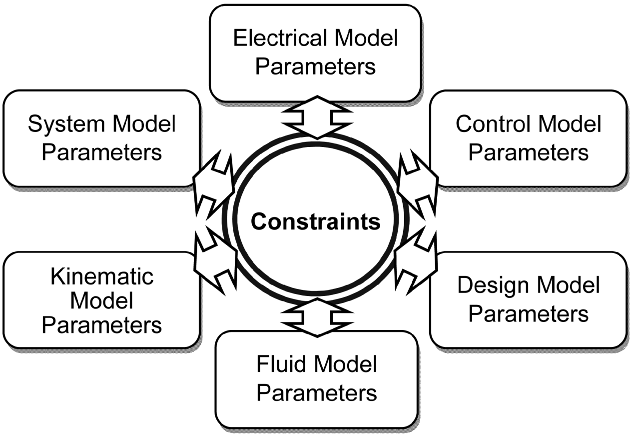


Figure 2.2. Constraints between all domains on the conceptual design level

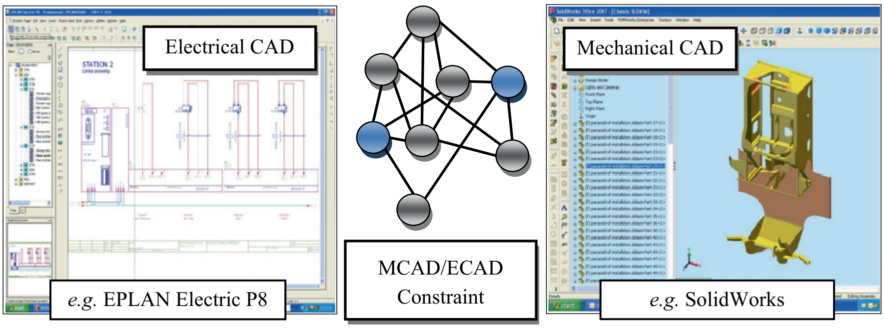


Figure 2.3. Constraints between MCAD and ECAD models on the system realisation level

The organisation of this chapter is as follows: in Section 2.2, an overview of current techniques aimed towards achieving integrated design of mechatronic systems as well as a research gap analysis is provided; in Section 2.3, a constraint-modelling approach towards integrated design of mechatronic systems is proposed, along with a classification of mechanical and electrical constraints; in Section 2.4, the use of the method discussed in Section 2.3 is illustrated by means of a robot arm as an example. Both discipline-specific and cross-disciplinary constraints existing in the robot arm example are identified; in Section 2.5, a framework for integrated mechatronic design approach and the capabilities needed to realise such a framework are discussed; finally, in Section 2.6 closing remarks are provided.

2.2 State of the Art and Research Gaps

In this section, a brief overview of a variety of research activities relevant to the development of approaches towards integrated design of mechatronic systems is presented and the research gaps are identified.

2.2.1 Product Data Management

The amount of engineering-related information required for the design of complex mechatronic products tends to be enormous. Aspects to be considered include geometric shapes of mechanical components, electrical wiring information, information about input and output pins of electronic circuit boards, and so on. In order to keep track of all these product data during the development process, product data management (PDM) systems are used. PDM systems manage product information from design to manufacture, and to end-user support. In terms of capabilities, PDM systems support five basic user functions [2.3]:

1. Data vault and document management that provides storage and retrieval of product information.
2. Workflow and process management that controls procedures for handling product data.
3. Product structure management that handles bills of materials, product configurations, associated design versions, and design variations.
4. Parts management that provides information on standard components and facilitates re-use of designs.
5. Program management that provides work breakdown structures and allows coordination between processes, resource scheduling, and project tracking.

In terms of state of the art technology, contemporary PDM systems have incorporated the use of web-based technology. An example is a component-based product data management system (CPDM) developed by Sung and Park [2.4]. Their CPDM system consists of three tiers: the first tier is focused on allowing users to access the system through a web browser; the second tier is the business logic tier that handles the core PDM functionality; and the third tier is composed of a database and vault for the physical files. This CPDM system has been implemented on the Internet for a local military company that manufactures various mechatronic systems such as power cars, motorised cars, and sensitive electrical equipments. Web-based PDM systems can also be used for similarity search tasks in order to identify existing designs or components of specific shape or manufacturing-related information that may be useful for new designs or design alternatives.

You and Chen [2.5] proposed an algorithm that runs in web-based PDM systems. In their algorithm, a target part is given with characteristic attributes, and similar parts in the database are identified based on their shape or manufacturing features. The results are sorted in the order of similarity. You and Chen's proposed algorithm for similarity evaluation adopts the polar Fourier transform (PFT) method, which is a discrete Fourier transform method.

There are several advantages in utilising web-based PDM systems. One advantage is user-friendliness: the browsers used in the PDM system are the same ones used within the World Wide Web, and hence web-based PDM systems require little training. Another advantage is their great accessibility since these browsers run on different platforms. However, there are several drawbacks as well: first, the information transferring speed is limited compared to the speed of LAN or WAN; second, mistakes relating to acquiring or transferring data can occur if the system is not utilised correctly; and finally, there are major concerns regarding security and exposing a company's trade secrets during the process of information transfer.

2.2.2 Formats for Standardised Data Exchange

PDM systems are tools that allow designers to manage and keep track of the product data throughout the entire design process. However, in order to ensure proper product configuration control, PDM systems must be able to communicate with the CAD systems that the designers use during the design process. In the context of integrated design of mechatronic products, this means communication between CAD/CAE systems of different engineering disciplines, *i.e.* MCAD and ECAD.

For instance, an MCAD model typically contains the following information [2.6]: *features*, which are high-level geometric constructs used during the design process to create shape configurations in the CAD models; *parameters*, which are values of quantities in the CAD model, such as dimensions; *constraints*, which are relationships between geometric elements in the CAD models, such as parallelism, tangency, and symmetry. An MCAD system cannot simply transfer such information to a PDM system or other CAD/CAE system because these systems have significantly different software architectures and data models. One potential approach towards achieving communication between various CAD, CAE, and PDM systems is through the utilisation of neutral file formats, such as, for example, Initial Graphics Exchange Specification (IGES) or the Standard for the Exchange of Product Model Data (STEP).

IGES was created for CAD-CAD information exchange. The fundamental role of IGES was to convert two-dimensional drawing data and three-dimensional shape data into a fixed file format in electronic form and pass the data to other CAD systems [2.7]. Major limitations of IGES include large file size, long processing time, and most importantly, the restriction of information exchange to shape data only [2.7]. Despite these limitations, IGES is still supported by most CAD systems and widely used for CAD information exchange.

Another important neutral file format for representation of product information is STEP, also known as ISO 10303. STEP can be viewed as consisting of several layers [2.7], the top layer being a set of application protocols (APs) that address specific product classes and lifecycle stages. These APs specify the actual information exchange and are constructed from a set of modules at lower layers, called integrated resources, which are common for all disciplines. The APs relevant to electro-mechanical systems integration are AP-203, AP-210, AP-212, and AP-214. AP-203 protocol defines the information exchange of geometric entities and configuration control of products. This protocol can capture common modern MCAD representations including 2D drawing, 3D wireframes, surface models, and solid models. AP-212 is concerned with electro-mechanical design and installation.

Currently, there is an ongoing effort in making STEP information models available in a universal format to business application developers. Lubell *et al.* [2.8] have presented a roadmap for possible future integration of STEP models with widely accepted and supported standard software modelling languages such as UML and XML. STEP provides standardised and rigorously-defined technical concepts and hence shows greater quality than other data exchange standards, but the traditional description and implementation method for STEP has failed to achieve the popularity of XML and UML [2.8]. Thus, emerging XML and UML-based STEP implementation technology shows promise for better information exchange ability.

2.2.3 The NIST Core Product Model

Most PDM systems and the exchange standards used for communication between CAD/CAE/PDM systems focus mainly on product geometry information. However, more attention is needed for developing standard representations that specify design information and product knowledge in a full range instead of solely geometry-oriented. At the US National Institute for Standards and Technology (NIST), an information modelling framework intended to address this issue of expanding the standard representations to a full range has been under development [2.9]. This conceptual product information modelling framework has the following key attributes [2.9]:

1. It is based on formal semantics and will eventually be supported by an appropriate ontology to permit automated reasoning.
2. It deals with conceptual entities such as artefacts and features and not specific artefacts such as motor, pumps, or gears.
3. It is to serve as information repository about products, and such information includes product description that are not at the time incorporated.
4. It is intended to foster the development of applications and processes that are not feasible in less information-rich environments.

One major component of this information modelling framework is the core product model (CPM). The CPM is developed as a basis for future CAD/CAE information exchange support system [2.7]. CPM is composed of three main components [2.7]:

(a) *Function*: models what the product is supposed to do.
(b) *Form*: models the proposed design solution for the design problem specified by the function, usually the product's physical characteristics are modelled in terms of its geometry and material properties.
(c) *Behaviour*: models how the product implements its function in terms of the engineering principles incorporated into a behavioural or casual model.

CPM was further extended to components with continuously varying material properties [2.10]. The concept for modelling continuously varying material properties is that of distance fields associated with a set of material features, where values and rate of material properties are specified [2.10]. This extension of CPM uses UML to represent scalar-valued material properties as well as vector- and tensor-valued material properties.

2.2.4 Multi-representation Architecture

Another approach that can be used to support the integrated design of mechatronic systems is the multi-representation architecture (MRA) proposed by Peak *et al.* [2.11]. It is a "design analysis integration strategy that views CAD-CAE integration as an information-intensive mapping between design models and analysis models" [2.11]. In other words, the gap that exists in CAD/CAE between design models and analysis models is considered too large to be covered by a "single general integration bridge"; hence MRA addresses the CAD-CAE integration problem by placing four information representations as "stepping stones" between design and analysis tools in the CAD/CAE domains. The four information representations are: solution method models, analysis building blocks, product models, and product model-based analysis models.

Solution method models represent analysis models in low-level, solution method-specific form. They combine solution tool inputs, outputs, and control into a single information entity to facilitate automated solution tool access and results retrieval. Analysis building blocks represent engineering analysis concept and are largely independent of product application and solution method. They represent analysis concepts using object and constraint graph techniques and have a defined information structure with graphical views to aid development, implementation, and documentation. Product models represent detailed, design-oriented product information. A product model is considered the master description of a product that supplies information to other product lifecycle tasks. It represents design aspects of products, enables connections with design tools, and supports idealisation usable in numerous analysis models. And finally, product model-based analysis models contain linkages that represent design-analysis associativity between product models and analysis building blocks.

The MRA can be used to support integrated design of mechatronic systems because it has the flexibility to support different solution tools and design tools and also accommodating analysis models from diverse disciplines. Object and constraint graph techniques used in the MRA provide modularity and semantics for clearer representation of design and analysis models. Peak *et al.* [2.11] have evaluated the MRA using printed wiring assembly solder joint fatigue as a case study. Their results show that specialised product model-based analysis models "enable highly automated routine analysis and uniformly represent analysis models containing a mixture of both formula-based and finite element-based relations" [2.11]. With the capability of analysing formula-based and finite element-based models, the MRA can be used to create specialised CAE tools that utilise both design information and general purpose solution tools.

2.2.5 Constraint-based Techniques

Many mechanical engineering CAD systems provide parametric and feature-based modelling methods and support frequent model changes. In order to define and analyse various different attributes of a product, a variety of models including design models, kinematic models, hydraulic models, electrical models, and system models are needed. Except for geometry-based data transfers, there is neither exchange nor integration of data for interdisciplinary product development available. Kleiner *et al*. [2.12] proposed an approach that links product models using constraints between parameters. The integration concept is based on parametric product models, which share their properties through the utilisation of constraints. In their context, a virtual product is represented by partial models from different engineering disciplines and associated constraint models.

The fundamentals for the development of neutral, parametric information structures for the integration of product models are provided by existing data models from ongoing development as well as concepts from constraint logic programming [2.13]. The parametric information model that Kleiner *et al*. [2.12] developed is based on the Unified Modelling Language (UML). The model contains the class *Item*, which represents real or virtual objects such as parts, assemblies, and models. Every object *Item* has a version (class *ItemVersion*) and specific views (class *DesignDisciplineItem Definition*). A view is relevant for the requirements of one or more lifecycle stages and application domains and collects product data of the *Item* and *ItemVersion* object. The extension of STEP product data models includes general product characteristics (class *Property*), attributes (class *Parameter*) and restricted relationships (class *Constraint*). The information model Kleiner *et al*. [2.12] developed is based on the integration of independent CAx models using their parameters. The links between CAx models are implemented using the class *Constraint*, which can set parameters of different product models in relationship to each other. On one hand, a constraint restricts at least one parameter and on the other hand, a parameter may be restricted by several constraints, which are building a constraint net. Different types of constraints are implemented in subclasses in order to characterise the relationship between parameters in detail.

The constraint-based parametric integration offers an alternative solution compared to unidirectional process chains or file-based data exchange procedures using neutral data formats (*e.g.* IGES and STEP). Model structures and properties could be imported, analysed, and exported by linking different CAx models. Kleiner *et al*. [2.12] developed a Java-based software system that supports product data integration for the collaborative design of mechatronic products. The software system, called Constraint Linking Bridge (Colibri), is developed based on the constraint-based integration concept and the described information model. It sets up connections to CAx systems in order to analyse model structures and parameters. Different interfaces to CAD/CAE systems allow the transfer of appropriate model structures and parameters as well as parameter transformation activities. Browsers for models (model viewer) and constraints (constraint viewer) are used as graphical user interfaces to analyse, specify, check, and solve constraints. The current version of Colibri contains implemented interfaces to the CAD system Pro/ENGINEER, the simulation application system MATRIXx/SystemBUILD as well as to the multi-body simulation software system AUTOLEV. Colibri is integrated into a distributed product development environment, called *Distributed Computing Environment/ Distributed File System* (DCE/DFS). This infrastructure offers platform-independent security and directory services for users and groups, policy and access control management as well as secure file services storing objects in electronic vaults.

This software Colibri, though is just a prototype, it offers an alternative technology for sharing product data and illustrates the possibility of integrating difference CAx models by linking them and using constraints to specify their product development relationships. This work done by Kleiner *et al*. opens up opportunities for further development that supports multi-disciplinary modelling and simulation of mechatronic systems and offers user functions for data sharing.

2.2.6 Active Semantic Networks

In the area of electrical engineering CAD, Schaefer *et al*. [2.14] proposed a shared knowledge base for interdisciplinary parametric product data models in CAD. This approach is based on a so-called Active Semantic Network (ASN). In ASNs, constraints can be used to model dependencies between interdisciplinary product models and co-operation, and rules using these constraints can be created to help designers to collaborate and integrate their results to a common solution. With this design approach, designers have the ability to visualise the consequences of design decisions across disciplines. This visualisation allows designers to integrate their design results and to improve the efficiency of the overall design process.

A semantic network is a graphic notation for representing knowledge in patterns of interconnected nodes and arcs. It is a graph that consists of vertices, which represent concepts, and arcs, which represent relations between concepts. An ASN can be realised as an active, distributed, and object-oriented DBMS (database management system) [2.15]. A DBMS is computer software designed for the purpose of managing a database. It is a complex set of software programs that controls the organisation, storage and retrieval of data in a database. An active DBMS is a DBMS that allows users to specify actions to be taken automatically when certain conditions arise.

Constraints defined in product models can be specified by rules. When a constraint is violated, possible actions include extensive inferences on the product data and notifications to the responsible designers to inform them about the violated constraints. There exists constraint propagation within the ASN. Constraints are mainly used in CAD to model dependencies between geometric objects. In the ASN, constraints are used to model any kind of dependency between product data [2.15].

A database object in the ASN consists of the data itself, a set of associated rules, and co-operative locks. This means that event-condition-action (ECA) rules can be connected to database objects to specify their active behaviour. Constraints are modelled as normal database objects that are also subject to user modifications. ECA rules linked to a constraint object specify the active behaviour of the constraint and are evaluated at run time. The ASN uses a rule-based evaluation method for constraint propagation.

2.2.7 Summary and Research Gap Analysis

Product data management systems manage product information from design to manufacture to end-user support. They ensure that "right information in the right form is available to the right person at the right time" [2.3]. PDM technology, if it can be implemented to all CAD/CAE systems, can significantly improve the design process of mechatronic systems because its infrastructure is user-friendly and has great accessibility. For example, Windchill developed by PTC has a browser-based user interface that uses standard HTML for bi-directional communication of form-based information and Java applets to deliver interactive application capabilities [2.3]. Rolls-Royce AeroEngines designs and manufactures mid-range aircraft engines. They use PTC Windchill for their integrated product development environments, allowing them to maintain a quicker product delivery with increasing costs. Modern PDM systems, such as PTC Windchill, though possessing the capability of communicating with several MCAD systems, lack the ability to communicate with ECAD systems and other CAD/CAE systems that may be used during the design of mechatronic systems.

Standardised data exchange formats, such as ISO 10303 (STEP), provide information exchange for parameterised feature-based models between different CAD systems. They provide communication between CAD systems through system-independent file formats that are in computer-interpretable forms for data transmission. These data exchange formats cover a wide range of application areas: aerospace, architecture, automotives, electronic, electro-mechanical, process plant, ship building, and the list can go on and on. However, as with any standard-based exchange of information between dissimilar systems, it is impossible to convey certain elements defined in some particular CAD systems but having no counter-parts in others [2.6]. Furthermore, past testing experience has shown that differences in the internal accuracy criteria of CAD systems can lead to problems of accuracy mismatch, which has caused many translation failures. Also, the provision of explicit geometric constraints adds possibilities for redundancy in shape models, and such geometric redundancy implies more possibility of accuracy mismatch. Hence, despite the ability to cover a wide range of applications areas, data exchange formats also have numerous problems to be solved.

NIST CPM and its extensions are abstract models with general semantics with specific semantics about a particular domain embedded within the usage of the models for that domain. CPM represents a product's form, function, and behaviour, as well as its physical and functional decompositions, and the relationships among these concepts. CPM is intended to capture product and design rationale, assembly, and tolerance information from the earliest conceptual design stage to the full lifecycle, and also facilitates the semantic interpretability of CAD/CAE/CAM systems. The current model also supports material model construction, material-related queries, data transfer, and model comparison. The construction process involves the definition of material features and choosing properties for distance field. Further research is needed to develop an API (application programming interface) between CPM and PLM systems, and to identify or develop standards for the information interchange [2.9].

Multi-representation architecture is aimed at satisfying the needs in the links between CAE and CAD. These needs include [2.11]:

1. Automation of ubiquitous analysis
2. Representation of design and analysis associativity and of the relationships among models
3. Provision of various analysis models throughout the lifecycle of the product

The initial focus of the MRA is on ubiquitous analyses, which are analyses that are regularly used to support the design of a product [2.16]. The MRA supports capturing knowledge and expertise for routine analysis through semantic-rich information models and the explicit associations between design and analysis models. While the MRA captures routine analysis and the mapping between design parameters and analysis parameters, there is still the opportunity for model abuse [2.17]. The MRA enables reuse of the analysis templates in product development. The behaviour model creators (such as the analysts) and behaviour model users (such as the designers) often do not have the same level of understanding of the model and thus limit the reuse of a model [2.17]. The gap between designers and analysts is decreased by providing engineering designers with increased knowledge and understanding about behavioural simulation. Plans for future work regarding the MRA include [2.17]: further instantiation of the behavioural model repository, refinement of knowledge representation using ontology languages, and implementation to support instantiation with design parameters for execution.

The collaborative design system Colibri developed by Kleiner *et al*. [2.12] offers a new approach for exchanging information across the disciplinary divide as compared to unidirectional process chains or file-based data exchange procedures using neutral data formats (*e.g.* IGES and STEP); however, it focuses on linking various CAx models and does not cover the information gap between mechanical CAD and electrical CAD systems.

In designing a mechatronic product, there are many situations that require the exchange of information between MCAD models and ECAD models. Modifications made on MCAD site may lead to significant design modifications to be made on ECAD site and *vice versa*. Obviously, there exist a huge number of constraints between a mechanical part of a mechatronic product and its electrical counterpart that have to be fulfilled to have a valid design configuration. As yet, such interdisciplinary constraints between models of different engineering design domains cannot be handled in a multi-disciplinary CAE environment due to the lack of appropriate multi-disciplinary data models and appropriate propagation method. Table 2.1 summarises the research gaps in the aforementioned integration approaches. The framework for integrated design of mechatronic systems proposed in the following sections focuses on integrating the mechanical and the electrical domains. The framework is intended to support the information exchange of design modification during the design process.

Table 2.1. Research gaps in the integration approaches

Approach	Description	Research gap
PDM	Manages product information allowing multiple designers to work on a shared repository of design information.	Preserves only file-level dependencies between information from multiple domains. Does not capture fine-grained information dependencies such as at a parameter level.
Standardised Data Exchange	Supports information exchange between CAD/CAE/PDM systems.	The emphasis of data exchange standards is on information flow across systems in a given domain, not on cross-discipline integration.
Multi-representation Architecture	Supports CAD-CAE integration through the usage of four models each supporting different levels of product information.	The focus in MRA is on integrating geometric and analysis information. It does not address the information link between mechanical and electrical CAD systems.
Colibri	Shares data across design teams of different domains through constraints and parametric relations in CAx.	Does not provide information exchange between mechanical CAD and electrical CAD.

2.3 An Approach to Integrated Design of Mechatronic Products

2.3.1 Modelling Mechatronic Systems

The constraint modelling-based approach proposed in this chapter (Figure 2.4) is similar to the semantic network approach briefly described before in that constraints are being modelled as nodes and relationships are drawn between nodes. The components of mechatronic systems are modelled as objects with attributes, and constraints between these attributes are identified and modelled. The procedures of the proposed constraint modelling approach are as follows:

STEP 1: List all components of the mechatronic system and their attributes and classify the components in either the mechanical domain or the electrical domain.

STEP 2: Based on the attributes of the component, draw the constraint relationship between the components in the domain and appropriately label the constraint by the constraint categories.

STEP 3: Based on the attributes of the component, draw the constraint relationship between the components across the domains and appropriately label the constraint by the constraint categories.

STEP 4: Construct a table of constraints for the particular mechatronic system. The table contains a complete list of the every component of the mechatronic system, the table is to indicate that, when a particular attribute of the component is being modified, which attribute of which component (both within the domain and across the domain) would be affected.

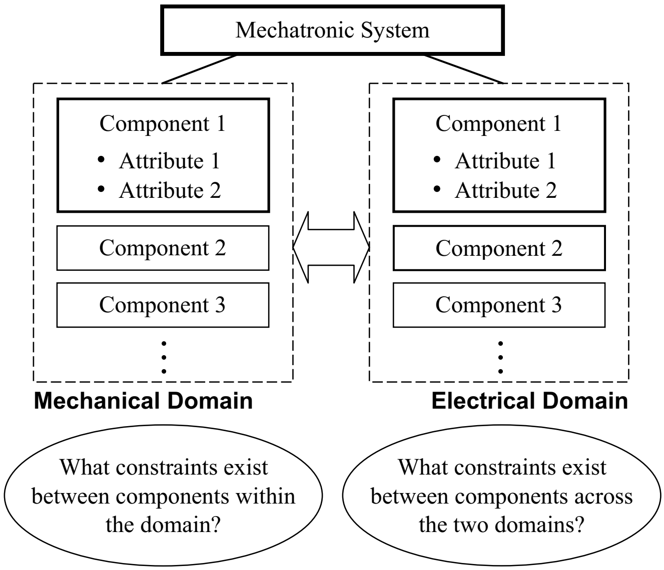


Figure 2.4. A graphical view of the constraint modelling approach

2.3.2 Constraint Classification in Mechanical Domain

Geometric Constraints

Most CAD systems allow the creation of variational models with parameterisation, constraints and features. The set of common geometric constraint types is listed as follows [2.18]:

- *Parallelism* – this has an undirected form and a directed form with one reference element. There is also a dimensional subtype, in which a constrained distance can be specified.
- *Point-distance* – in the directed case, the reference element may be either point, line, or plane. Multiple points may be constrained. In the undirected case, the number of constrained points is limited to two, and a dimensional value is required.
- *Radius* – has a dimensionless form, for example, “the radii of all these arcs are the same”, and a dimensional form, for example, “the radii of all the constrained arcs have the same specified value”.
- *Curve-length* – asserts that the lengths of all members of a set of trimmed curves are equal. There is a dimensional form allowing the value of the length to be specified.
- *Angle* – constraints a set of lines or planes to make the same angle with a reference element, or in the undirected case specifies the angle between precisely two such elements.
- *Direction* – a vector-valued constraint used for constraining the directional attributes of linear elements such as lines or planes.

- *Perpendicularity* – there may be either one or two reference elements (lines or planes), and all the constrained elements are required to be perpendicular to them. There is also an undirected form in which two or three elements are required to be mutually perpendicular.
- *Incidence* – in its simplest form, it simply asserts that one or more constrained entities are contained within that reference element.
- *Tangency* – may be used to specify multiple tangencies between a set of reference elements, and as set of constrained elements.
- *Coaxial* – constrains a set of rotational elements to share the same axis or to share a specified reference axis.
- *Symmetry* – constrains two ordered sets of elements to be pair-wise symmetric with respect to a given line or plane.
- *Fixed* – used to fix points and directions in absolute terms for anchoring local coordinated systems in global space.

Kinematics Constraints

Kinematics is a branch of mechanics that describes the motion of objects without the consideration of the masses and forces that bring about the motion [2.1]. Kinematics is the study of the position of an object changes with time. Position is measured with respect to a set of coordinates. Velocity is the rate of change of position. Acceleration is the rate of change of velocity. In designing mechatronic systems, the kinematics analysis of machine elements is very important. Kinematics determines the position, velocity, and acceleration of machine links. Kinematics analysis helps to find the impact and jerk on a machine element.

Force Constraints

In mechanical engineering, and in particular, in dealing with "machines in mechatronics", it often involves the study of relative motion between the various parts of a machine as well as the forces acting on them, hence the knowledge in this subject of "forces" is very essential for an engineer to design the various parts of mechatronic systems [2.1]. Force is an important factor as an agent that produces or tends to produce, destroys or tends to destroy motion. When a body does not move or tend to move, the body does not have any friction force. Whenever a body moves or tends to move tangentially with respect to the surface on which it rests, the interlocking properties of the minutely projected particles due to the surface roughness oppose the motion. This opposition force that acts in the opposite direction to the movement of the body is the force of friction. Both force and friction play an important role in mechatronic systems.

In considering the force constraint in mechanical systems, there are three major parameters that can affect the mechanical systems: the stiffness of the system, the forces opposing motion (such as frictional or damping effects), and the inertia or resistance to acceleration [2.19]. The stiffness of a system is described by the relationship between the forces used to extend or compress a spring and the resulting extension or compression. The inertia or resistance to acceleration exhibits the property that the bigger the inertia (mass) the greater the force required to give it a specific acceleration.

Energy Constraints

Energy is a scalar physical quantity, which is a property of objects and systems that is conserved by nature. Energy can be converted in a variety of ways. An electric motor converts electrical into mechanical and thermal energy, a combustion engine converts chemical into mechanical and thermal energy, and so on. In physics, mechanical energy describes the potential energy and kinetic energy present in the components of a mechanical system. If the system is subject only to conservative forces, such as only to gravitational force, by the principle of conservation of mechanical energy the total mechanical energy of the system remains constant.

Material Constraints

The various machine parts of the mechatronic system often experience different loading conditions. If a change of motion of the rigid body (the machine parts) is prevented, the force applied will cause a deformation or change in the shape of the body. Strain is the change in dimension that takes place in the material due to an externally applied force. Linear strain is the ratio of change in length when a tensile or compressive force is applied. Shear strain is measured by the angular distortion caused by an external force. The load per unit deflection in a body is the stiffness. Deflection per unit load is the compliance. If deformation per unit load at a point on the body is different from that at the point of application of the load then compliance at that point is called cross-compliance. In a machine structure, cross-compliance is an important parameter for stability analysis during machining.

The strength of a material is expressed as the stress required causing it to fracture. The maximum force required to break a material divided by the original cross-sectional area at the point of fracture is the ultimate tensile strength of the material. It is obvious that the stress allowed in any component of a machine must be less than the stress that would cause permanent deformation. A safe working stress is chosen with regard to the conditions under which the material is to work. The ratio of the yield stress to allowable stress is the factor of safety.

Tolerance Constraints

The relationship resulting from the difference between the sizes of two features is the fit. Fits have a common basic size. They are broadly classified as clearance fit, transition fit, and interference fit. A clearance fit is one that always provides a clearance between the hole and shaft when they are assembled. A transition fit is one that provides either a clearance or interference between the hole and the shaft when they are assembled. An interference fit is one that provides interference all along between the hole and the shaft when they are assembled.

The production of a part with exact dimensions repetitively is usually difficult. Hence, it is sufficient to produce parts with dimensions accurate within two permissible limits of size, which is the tolerance. Tolerance can be provided on both sides of the basic size (bilateral tolerance) or on one side of the basic size (unilateral tolerance). The ISO systems of tolerance provides for a total of 20 standard tolerance grades.

In any engineering industry, the components manufactured should also satisfy the geometrical tolerances in addition to the common dimensional tolerances. Geometrical tolerances are classified as:

- *Form tolerance* – straightness, flatness, circularity, cylindricity, profile of any line, and profile of any surface
- *Orientation tolerance* – parallelism, perpendicularity, and angularity
- *Location tolerance* – position, concentricity, co-axiality, and symmetry
- *Run-out tolerance* – circular run-out and axial run-out

2.3.3 Constraint Classification in Electrical Domain

Electrical Resistance

Electrical resistance is a measure of the degree to which an object opposes the electric current through it. The electrical resistance of an object is a function of its physical geometry. The resistance is proportional to the length of the object and inversely proportional to the cross-section area of the object. All resistors possess some degree of resistance. The resistances of some resistors are indicated by numbers. This method is used for low value resistors. Most resistors are coded using colour bands, and the way to decode these bands is as follows: the first band gives the resistance value of the resistor in ohms. The fourth band indicates the accuracy of the value. Red in this region indicates 2%, gold indicates 5%, and silver indicates 10% accuracy.

Electrical Capacitance

Electrical capacitance is a measure of the amount of electric charge stored for a given electric potential. The most common form of charge storage device is a two-plate capacitor, which consists of two conducting surfaces separated by a layer of insulating medium called dielectric. The dielectric is an insulating medium through which an electrostatic field can pass. The main purpose of the capacitor is to store electrical energy.

Electrical Inductance

A wire wound in the form of a coil makes an inductor. The property of an inductor is that it always tries to maintain a steady flow of current and opposes any fluctuation in it. When a current flows through a conductor, it produces a magnetic field around it in a plane perpendicular to the conductor. When a conductor moves in a magnetic field, an electromagnetic force is induced in the conductor. The property of the inductor due to which it opposes any increase or decrease in current by the production of a counter emf (electromotive force) is known as self-inductance. The emf force developed in an inductor is proportional to the rate of current through the inductor, and mathematically it depends on the amount of current, the voltage developed, and a proportionality constant that represents the self-inductance of the coil.

Motor Torque

The torque generation in any electric motor is essentially the conversion process of converting electrical energy into mechanical energy. It can be viewed as a result of the interaction of two magnetic flux density vectors: one generated by the stator and one generated by the rotor. In different motor types, the way these vectors generated is different [2.20]. For instance, in a permanent brushless motor the magnetic flux vector is generated by the current in the windings. In the case of an AC induction motor, the stator magnetic flux vector is generated by the current in the stator winding, and the rotor magnetic flux vector is generated by induced voltages on the rotor conductors by the stator field and resulting current in the rotor conductors. The torque production in an electric motor is proportional to the strength of the two magnetic flux vectors (stator's and rotor's) and the sine of the angle between the two vectors.

System Control

The control system provides a logical sequence for the operating program of the mechatronic system. It provides the theoretical values required for each program step, it continuously measures the actual position during motion, and it processes the theoretical versus actual difference [2.21]. In controlling a robot, for example, there are two types of control techniques: point-to-point and continuous path. The point-to-point control involves the specification of the starting point and end point of the robot motion and requiring a control system to render feedbacks at those points. The continuous path control requires the robot end-effector to follow a stated path from the starting point to the end point.

2.4 Illustrative Example: a Robot Arm

2.4.1 Overview of the Robot Arm

A robot is a mechatronic system capable of replacing or assisting the human operator in carrying out a variety of physical tasks. The interaction with the surrounding environment is achieved through sensors and transducers, and the computer-controlled interaction systems emulate human capabilities. The example investigated is the SG5-UT robot arm designed by Alex Dirks of the CrustCrawler team [2.22] (see Figure 2.5).

List of major mechanical components:

- Base and wheel plates
- Links: bicep, forearm, wrist
- Gripper
- Joints: shoulder, elbow, and wrist
- Hitec HS-475HB servos (base, wrist and gripper)
- Hitec HS-645MG servos (elbow bend)
- Hitec HS-805BB servos (shoulder bend)

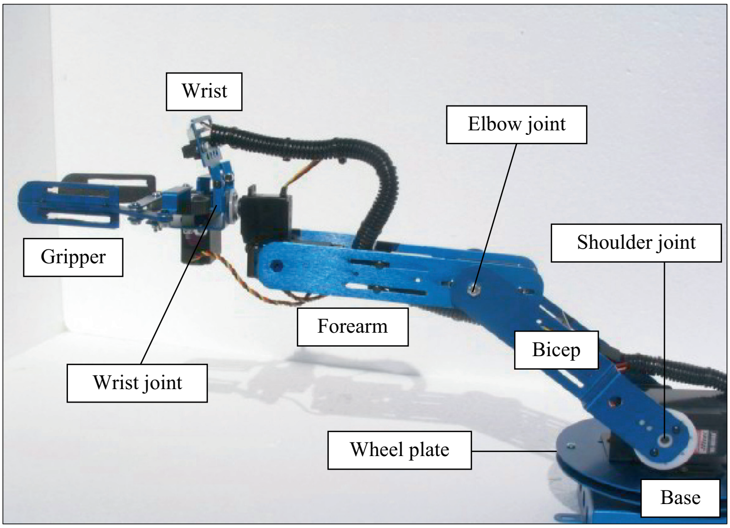
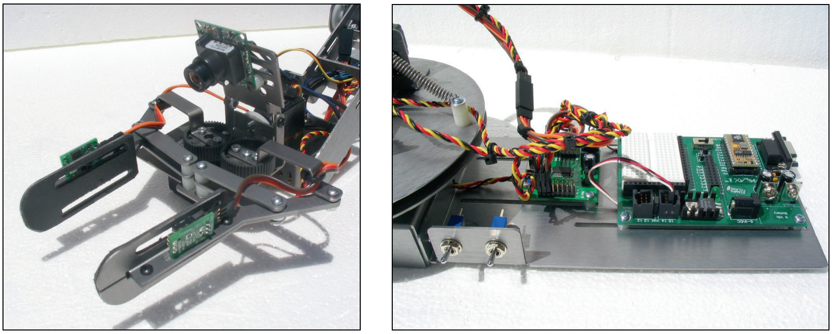


Figure 2.5. The CrustCrawler SG5-UT robot arm



(a) Gripper (b) Microcontroller

Figure 2.6. Major components of the robot arm

The most critical aspect of any robot arm is in the design of the gripper [2.22]. The usefulness and functionality of a robot arm is directly related to the ability to sense and successfully manipulate its immediate environment. The gripper drive system, as shown in Figure 2.6(a), consists of a resin gear train driven by an HS-475HB servo. The servos are needed to provide motion to the various mechanical links as well as the gripper. The mounting site of the servos and the power routing to servos and supporting electronics are some of the important aspects to be considered in the design of this robot arm. The microcontroller board (Figure 2.6(b)) is essential for communication between the robot and a PC, providing users the ability to manipulate the robot. It is important to have accurate information on the pin connections and the corresponding components that are being controlled.

2.4.2 Modelling Constraints for SG5-UT

As mentioned in Section 2.3.1, a four-step procedure is used for modelling the SG5-UT robot arm.

STEP 1: List all components of the SG5-UT robot arm and their attributes and classify the components in either the mechanical domain or the electrical domain. These components are listed in Figure 2.7.

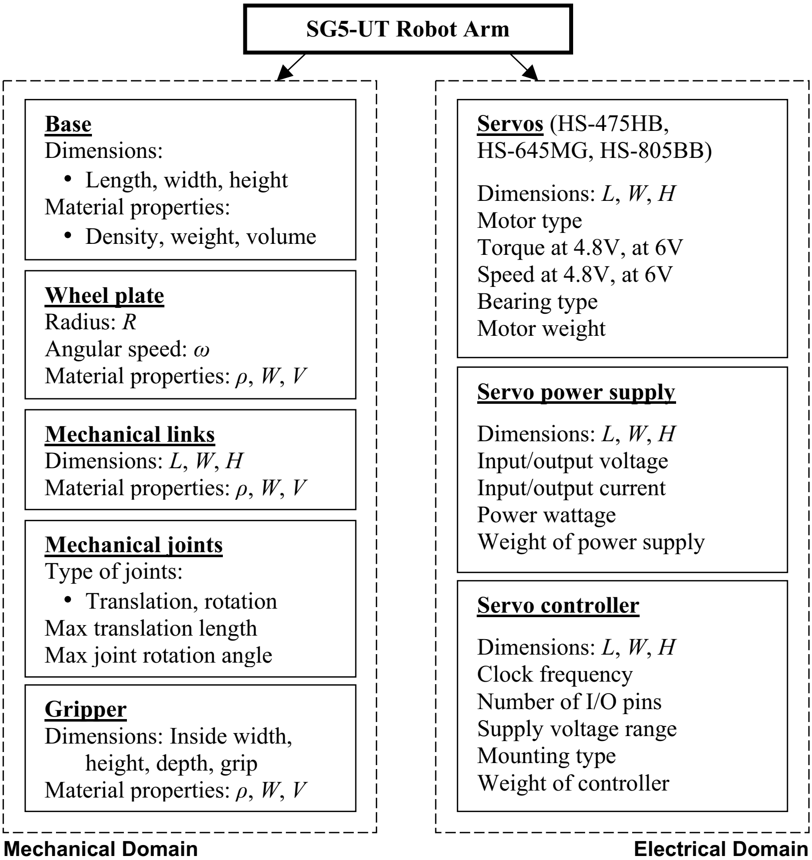


Figure 2.7. A list of major mechanical and electrical components and attributes

STEP 2: Based on the attributes of the components, draw the constraint relationship between the components in the domain and appropriately label the constraint by the constraint categories, as presented below in Figure 2.8 and in Tables 2.2 and 2.3.

For the purpose of brevity, only geometric constraints are described as constraints in the mechanical domain. However, there are many constraints that exist

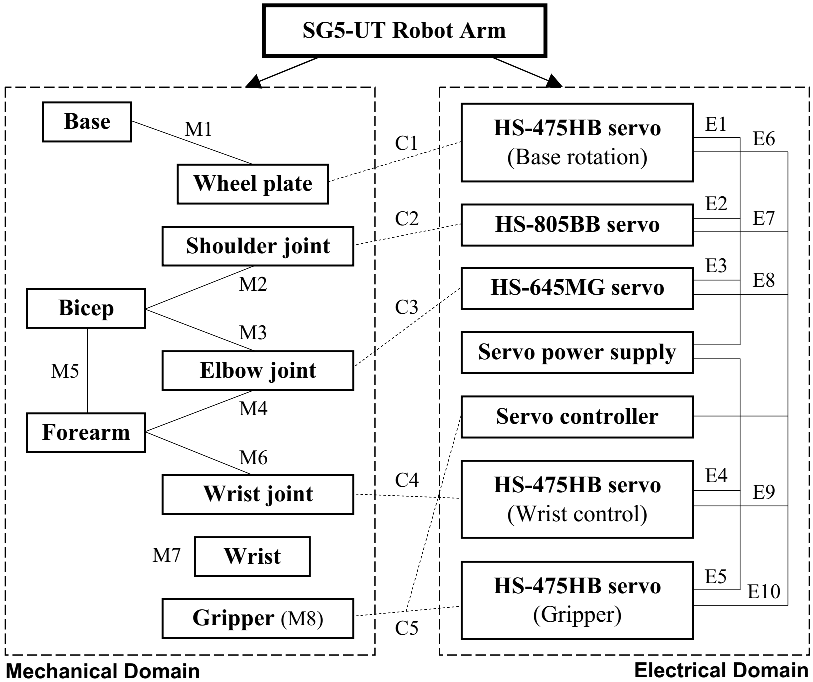


Figure 2.8. Constraints model of SG5-UT robot arm (solid line represents constraints within the domain, M is mechanical and E is electrical; dashed line represents cross-disciplinary, C)

Table 2.2. Mechanical constraints for the SG5-UT robot arm

	Constraint type	Constraint description
M1	*Geometric*: *fixed* Between base and wheel plate	The co-ordinate of base contact point with wheel plate is the co-ordinate of wheel plate centre.
M2	*Geometric*: *coaxial* Between bicep and shoulder joint	The axis of shoulder joint is coaxial with the axis of bicep rotation.
M3	*Geometric*: *fixed* Between bicep and elbow joint	The co-ordinate of the bicep contact point with the forearm is the co-ordinate of elbow joint.
M4	*Geometric*: *coaxial* Between elbow joint and forearm	The axis of elbow joint is coaxial with the axis of forearm rotation.
M5	*Geometric*: *angle* Between bicep and forearm	The angle between bicep and forearm is between 90 and 270 degrees. The forearm is not permitted to crush into the bicep.
M6	*Geometric*: *fixed* Between forearm and wrist joint	The co-ordinate of wrist contact point with forearm is the co-ordinate of wrist joint.
M7	*Geometric*: *coaxial* Between wrist joint and gripper	The axis of wrist joint is coaxial with the axis of gripper rotation.
M8	*Geometric*: *symmetry* Gripper	The left half of gripper and the right half of gripper are symmetric.

Table 2.3. Electrical constraints for the SG5-UT robot arm

	Constraint type	Constraint description
E1 ~ E5	*Maximum torque* Between the five servos and power supply	There is a maximum torque for a particular given voltage from the power supply.
E6 ~ E10	*System control* Between the five servos and servo controller	The servo controller determines the appropriate drive signal to move the actuator towards its desired position.

under other categories, such as force constraints, material constraints. For example, if the density of the material of the robot arm component is uniform, then the weight of that component would be the product of density of the material, volume of that component, and the gravitational acceleration.

STEP 3: Based on the attributes of the components, draw the constraint relationship between the components across the domains and appropriately label the constraint by the constraint categories, as presented below in Table 2.4.

Table 2.4. Cross-disciplinary constraints for the SG5-UT robot arm

	Constraint type	Constraint description
C1	*Kinematics–force–motor torque* Between the wheel plate and base rotation servo	The rotation speed of the wheel plate is dependent on the weight of the entire arm structure and the torque provided by the base rotation servo.
C2 | C4	*Geometry–force–motor torque* Between the servos and mechanical links	The torque required about each joint is the multiple of downward forces (weight) and the linkage lengths. This constraint exists in each lifting actuators.
C5	*System control–kinematics–force* Between the gripper, gripper servo, and servo controller	The servo controller determines the current state of the gripper given the current state of the actuators (position and velocity). The controller also adjusts the servo operation given the knowledge of the loads on the arm.

STEP 4: Construct a table of constraints for the particular mechatronic system. The table contains a complete list of the every component of the mechatronic system; the table is to indicate that, when a particular attribute of the component is being modified, which attribute of which component (both within the domain and across the domain) would be affected.

An example of the cross-disciplinary constraints would be the force relationship that is needed for motor selection. The motor that is chosen for the robot arm must not only support the weight of the robot arm but also support what the robot will be carrying. To perform the force calculation for each joint, the downward force (weight) of the components that has effect on the moment arm of the joint is multiplied by the linkage length and all the forces are summed to provide the torque required about each joint. This calculation needs to be done for each lifting actuator. For each degree of freedom added to the robot arm, the mathematical calculation gets more complicated, and the joint weights become heavier. Figure 2.9 illustrates the force calculation for a simple robot arm that has two degrees of freedom. And in Table 2.5, the cross-disciplinary constraints for the SG5-UT robot arm are identified and listed.

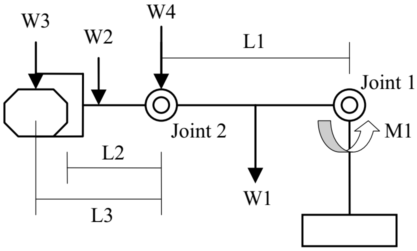


Figure 2.9. Force body diagram of a robot arm stretched out to its maximum length

Table 2.5. Table of constraints for the SG5-UT robot arm

Component (attribute)	Constrained attribute within the domain	Constrained attribute across the domain
Bicep (*L*, *W*, *H*)	Bicep (weight, volume)	Shoulder servo (torque)
Shoulder joint (location)	Bicep (location of axis of rotation)	
Forearm (*L*, *W*, *H*)	Forearm (weight, volume)	Shoulder servo, elbow servo (torque)
Elbow joint (location)	Forearm (location of axis of rotation)	
Wrist (*L*, *W*, *H*)	Wrist (weight, volume)	Shoulder servo, elbow servo, wrist servo (torque)
Base servo (torque)		Wheel plate (rotation speed)
Servo power supply	All servos (torque)	
Servo controller	All servos (control)	Gripper (location, velocity)

Torque about joint 1:

$$M1 = \frac{L1}{2} * W1 + L1 * W4 + (L1 + \frac{L2}{2}) * W2 + (L1 + L3) * W3 \tag{2.1}$$

Torque about joint 2:

$$M2 = \frac{L2}{2} * W2 + L3 * W3 \tag{2.2}$$

2.5 Requirements for a Computational Framework for Integrated Mechatronic Systems Design

Currently, the collaborative development of mechatronic systems occurs in an overlapping three-phase process. First, the mechanical design is synthesised based on customer requirements and technical constraints. Second, the technical specifications from the mechanical design are communicated to the electrical engineering team, which designs the electrical system to support the mechanical design. After an iterative process between the mechanical and electrical teams, the electrical engineers begin to finalise the electrical design as the software engineers start the third phase of supporting software design. Another iteration phase between the electrical and software engineers leads to the finalisation of the software design. The collaborative development of mechatronic systems can be considered from the point of view of the electrical, software, mechanical, and electronic engineers, each discipline has its design requirements to be fulfilled. The following are descriptions of the various design requirements in each discipline.

2.5.1 Electrical Design

Table 2.6 lists the design requirements from electrical engineering point-of-view.

2.5.1.1 Basic Requirements

Defined technical standard

Electrical engineering design standards are not identical worldwide. Many companies in the USA still use an iteration of older Joint Industrial Council (JIC) standards. These standards, developed in the early 1950s, were taken over by the National Fire Protection Association (electrical standards) and the National Fluid Power Association (hydraulic and pneumatic standards). Therefore, the National Fire Protection Association issued the current US electrical standards in NFPA 79 Electrical Standard for Industrial Machinery [2.1]. Most countries follow the electrical engineering guidelines outlined by the International Electro-technical Commission (IEC). The IEC was created in 1906 and has unified electrical design standards for most of the world. The IEC guides the design of electrical systems from metric definition to software/diagram specifications.

All designers on a project must agree upon the standard to be used. This will simplify data exchange throughout the design and avoid complications in unit conversion between groups in various corporate divisions.

Chosen software package

There are several software packages to facilitate electrical design today. These range from database programs to 2D CAD with electrical add-ons to fully developed electrical design studio packages. As most of these products still rely on proprietary data formats, designers (and their respective companies) must agree upon a software system or series of compatible software systems. This will ease restrictions in drawing and component information sharing during the design process.

Table 2.6. Requirements from an electrical engineering point of view

	Requirements list for current design of mechatronic systems
	A. *Initial requirements*
D	Defined technical standard (JIC, IEC)
D	Chosen software package, *i.e.* defined data representation/storage methods
	B. *Collaboration requirements*
D	Communication between mechanical, electrical, software engineers
W	Shared access to design data of all domains
D	Defined data storage procedures, organisation, and file formats
	C. *Energy requirements*
D	Load requirements for power consuming devices
D	Voltage/current entering the system
D	Space allocations for components and cabinets (geometry of mechanical system)
W	Cabinet locations to enable arrangement of terminal strips
	D. *Control requirements*
D	Type of control desired (microcontroller *vs*. PLC)
D	Required controls for each component (degrees of freedom, bounds, *etc*.)
D	What type of devices to control (motor, actuator, *etc*.)
D	Locations and types of inputs from sensors and switches
D	Necessary indicator lights
	E. *Installation concerns*
W	Company-preferred part vendors
D	Clear, intelligent diagrams for construction/installation
D	Complete set of matching connection point designations
D	Parts list
D	Checks for OSHA, UL, IEEE, *etc*. for safety compliance
W	Wire size/type of current installation (for system upgrades/product revisions)

D = Demands, must be satisfied during the design process
W = Wishes, would be ideal to satisfy during the design process, but not required

2.5.1.2 Collaboration Requirements

Communication between mechanical, electrical, software engineers

In order to design a mechatronic product, there must be constant facilitated communication between all involved disciplines. Currently, this communication exists as anything from yelling over cubicle walls to utilisation of cutting edge collaboration software, depending on corporate policy and available tools.

Shared access to design data of all domains

To efficiently perform a cross-discipline design, applicable data from all domains must be accessible to everyone involved in the design process. Currently, this data is rarely centrally located. There is a need for designers to access the updated information in a reasonable time frame. This may involve direct physical sharing of paper copies of design sketches, models, and function information or a network data storage solution.

Defined data storage procedures, organisation, and file formats

To begin an electrical design, the design team must agree upon how project data will be stored. Everyone on the team must know which office or person to send design document change requests. There must be clear organisation of design materials, be it through a product data management system or a common file naming convention. The team must also decide on one file format for design information storage from each design software system.

2.5.1.3 Energy Requirements

Load requirements for power consuming devices

The main connection between the mechanical system and the electrical system is when energy is converted from electrical power to mechanical motion. This conversion occurs in several power consuming devices including motors, heating elements, and lights.

Voltage/current entering the system

To select and arrange the proper components for an electrical design, the designer must know the voltage and current coming into the system. If necessary, engineers can use transformers to achieve the desired voltage or modify component selection to a more appropriate part number.

Space allocations for components and cabinets (geometry of mechanical system)

Using information from the mechanical engineering team, designers can plan the size of control cabinets based on the available space around the mechanical system. This enables designers to arrange the components inside the cabinet, an integral part of electrical design.

Cabinet locations to enable arrangement of terminal strips

Knowing the location of mechanical components, electrical engineers can plan the locations and arrangement of supporting electrical control cabinets. This allows them to logically order the wiring of control cabinets and accurately label terminal strips and wiring harnesses. Logical, accurate terminal strip numbers leads to accurate electrical wiring diagrams and more logical systems for the manufacturing team.

2.5.1.4 Control Requirements

Type of control desired (microcontroller vs. programmable logic controller)

Depending on the expected production volume, engineers must decide how to control mechatronic systems. Microcontrollers are meant for high volume applications, where a cheap, integrated chip can be specifically programmed for a certain function. For lower volume, more specialised applications, PLCs provide more versatility to the manufacturer and end user. PLCs are "well-adapted to a range of automation tasks. These are typically industrial processes in manufacturing where the cost of developing and maintaining the automation system is high relative to the total cost of the automation, and where changes to the system would be expected during its operational life" [2.2]. A designer must have intimate knowledge of these controllers and the target production figures for the design.

Required controls for each component (degrees of freedom, bounds, etc.)

To plan the control setup for an electrical system, some mechanical constraints are required. For example, in designing a robot arm, the design team needs to know how many degrees of freedom and how many motors to use in the robot arm. Furthermore, the limits of each of those degrees of freedom will translate into stopping points for the arm's motion.

What type of devices to control (motor, actuator, etc.)

To ensure that control cabling and signals are routed to the right components, electrical engineers must know what type of devices – from connection point designations to part numbers – are part of the control network.

Locations and types of inputs from sensors and switches

To provide the proper interface setup, electrical engineers must know what type and range of signals to expect from sensors and switches. They must also know the physical location of these objects. This helps them include the right parts based on reliability needs and control requirements (pushbutton *vs*. toggle *vs*. variable switch). This also serves to counteract miscommunication between electrical and mechanical engineers.

Necessary indicator lights

Finally, control design usually ends with the inclusion of indicator lights for operators. These lights indicate powered systems, tripped circuits, automated safety systems operation, and more. The control engineers specify the need for indication lights while the mechanical engineer indicates light locations on the instrument panel or equipment. It is the electrical engineering team's duty to provide power and signal feedback to these indicators, as well as communicate their existence to the software engineer.

2.5.1.5 Installation Concerns

Company-preferred part vendors

It is no secret that companies prefer to reuse vendors who provided reliable parts and trustworthy service. This helps the company establish strong vendor-user relationships that can reduce costs. This also creates a more uniform system environment, with a warehouse supply of redundant parts. Most engineers on the team will be comfortable with the vendors used in the past. New team members should be given a list of preferred vendors before the design begins.

Clear, intelligent diagrams for construction/installation

When electricians install electrical components in cabinets, they connect components to one another and to terminal strips based on connection point numbers. The connections between components are mapped out on wiring diagrams and power/control cabinet layouts.

The quality and accuracy of an electrical cabinet's wiring diagrams and component layout views decide the lifelong reliability of the cabinet. Ideally, the wiring diagrams will feature the same connection point numbers as the real part, label the colour of each wire in the bundle, and clearly show the sequential routing of wires from one component to the next.

Table 2.7. Requirements from a mechanical engineering point of view

	Requirements list for current design of mechatronic systems
	A. Kinematics requirements
D	Precise positioning of mechanical parts
D	Accurate measurement of velocity and acceleration
D	Control of type of motion and direction of motion
	B. Forces requirements
D	Knowledge of force, weight, load and deformation
D	Knowledge of the existing friction forces and their effects
D	Appropriate application of lubrication
W	Maximum reduction of wear and tear due to friction
	C. Energy requirements
D	Sufficient supply of energy sources (electricity, fuel, *etc.*)
D	Secure check of heating, cooling, and ventilation
W	High efficiency with low power consumption
	D. Material properties requirements
D	Solid understanding of the materials' stress-strain behaviour
D	The material does not fracture, fatigue, and display other types of failure during the lifecycle of the product
D	Appropriate choosing of factor of safety in stress analysis
	E. Material selection requirements
D	The materials selected satisfy the functional requirement
D	The materials are easily fabricated and readily available
D	The materials satisfy thermal and heat treating specifications
W	The materials allow the system to operate at higher speed/feeds
	F. Geometric constraint requirements
D	Accurate description of the relationship between features
D	Parts with dimensions accurate within the permissible tolerance
D	Parts satisfy the geometrical tolerances (*i.e.* parallelism)
W	Maximum balance between tolerance level and cost required
	G. Manufacturability requirements
D	Appropriate overall layout design
D	Appropriate form design of components
D	Appropriate use of standards and bought-out components
D	Appropriate documentation

D = Demands, must be satisfied during the design process
W = Wishes, would be ideal to satisfy during the design process, but not required

Complete set of matching connection point designations

When electrical parts are manufactured, each wire receptacle is given a specific connection point designation. The number of that receptacle will be the same for every part of that design manufactured. Electrical designers must know how the connection points on devices they use are numbered. Designers can then use those numbers on all wiring diagrams to reduce confusion during installation.

Parts list

To aid in billing, supply chain, and installation prep, electrical engineers should include a parts list in their design package, also known as a Bill of Materials. Among the benefits are checks from other designers and suppliers against incorrectly selected parts, misprinted part numbers, and supply availability.

Checks for OSHA, UL, IEEE, etc. for safety compliance

Safety is the utmost concern for all engineers. This is no different for electrical engineers, whose designs can be operated by customers with little or no technical experience or training. To help protect against shock, fire hazards, and general safety concerns, several organisations exist to certify that electrical designs meet stringent safety standards. Engineers must be comfortable with these standards and check for compliance during and after completion of electrical designs.

Wire size/type of current installation (for system upgrades/product revisions)

To protect continuity in existing products and systems, engineers must know the pre-design specifications. This includes the type and size of wire used.

2.5.2 Mechanical and Electronic Design

Tables 2.7 and 2.8 list the design requirements from mechanical and electronic engineering points of views, respectively.

2.5.3 Integrated Design

Table 2.9 lists the design requirements for integrated mechatronic design approach.

2.5.3.1 Mechanical-electronic Requirements

Even though many MCAD systems are able to generate output files that can be understood by CAD layout tools, in many cases this interface uses standard formatted files that usually do not contain the information listed above and only provide basic board outline as a set of lines that the electronic and mechanical designers has to interpret in later stages of design process.

The procedure for transferring PWBA data between electronic design and mechanical design during the design process is as follows:

1. In MCAD, define PWB outlines; define keep-in/keep-out areas, holes, cut-outs, *etc.*; pre-place ICs, connectors, switches and other fixed components.

Table 2.8. Requirements from electronic engineering point of view

	Requirements list for current design of mechatronic systems
	A. Electronic device requirements
D	Accurate specification and labelling of electronic features (resistance, capacitance, inductance, *etc*.)
D	Right amount of impurities are added to the semiconductor materials to provide desired conductivity
W	The device can respond to very small control signal
W	The device can respond at a speed far beyond the speed at which mechanical or electrical devices can
	B. Circuitry requirements
D	The printed circuit board layout satisfies the mechanical and space constraints imposed by the amount of area required to physically locate all the components on the board
D	Specify the keep-in and keep-out areas on the board
D	Specify the height restrictions for component placement
D	Specify specific features such as holes and cut-outs
D	Specify the pre-placement of components that are linked to mechanical constraints (ICs, connectors, switches, *etc*.)
	C. Signal requirements
D	Specify input and output
D	Appropriate form, display, and control impulse
D	If conversion required, specify whether it is digital-to-analogue or analogue-to-digital
D	For A/D conversion, choose the appropriate schemes and the appropriate variation within each scheme
W	Maximum reduction of ambient noise
	D. Controller requirements
D	Designed to withstand vibration, change in temperature and humidity, and noise
D	The interfacing circuit for input and output is inside the controller
D	Specify the required processing information (number of input and output pins, memory and processing speed required, *etc*.)
W	Easily programmed and have an easy programming language

D = Demands, must be satisfied during the design process
W = Wishes, would be ideal to satisfy during the design process, but not required

2. Convert the above MCAD information to IDF (or other standard format such as DXF and STEP) and transfer the file to ECAD.
3. In ECAD, read the IDF file, write ECAD model where the board structure is defined, all components are placed and all interconnect are routed. This will create an ECAD file.
4. Convert the above ECAD file to IDF (or other standard format) and transfer the file to MCAD.
5. In MCAD, the PWB is imported as an assembly file that has component information, including location and properties. Based on the information imported, perform component height analysis, thermal analysis, structural analysis, *etc*. using MCAE tools. It is important for the MCAD to have a

fully populated library of MCAE components containing thermal and mechanical models of all parts present in the electrical layout.

Table 2.9. Requirements for integrated mechatronic design

	Requirements list for an integrated mechatronic design framework
	A. General collaboration requirements
D	Corporate specific technical standards and software package selections
D	Centralised database for all product information with project-wide data storage procedures
D	Enhanced instant messaging/data sharing software integrated into design tools (ECAD/MCAD/Office)
	B. Energy requirements
D	Automated optimisation of wire specifications based on mechanical system energy needs
D	Automated synthesis of protection devices based on selection of power consuming devices
D	Real-time calculation of voltage/current entering and running through the system
D	Automated development of electrical system based on demands from electrical side and corporate-preferred parts database
	C. Control requirements
D	Automated generation of graphical/logical control system mapping with electrical design, enhanced with sensor, user input, output, and kinematic data
D	Software code classes for common I/O needs (temp sensors, motor control, switchboard, *etc.*)
D	Automated generation of computer simulation of completed control system for testing
	D. Installation concerns
D	Automated synthesis of necessary power/control cabinets followed by optimisation of control cabinet location and automated terminal strip connection point numbering (based on component selection)
D	Automated generation of clear, intelligent diagrams for construction/installation, based on system synthesis
D	Automated parts list generation and communication to supply chain
D	Automated system checks for OSHA, UL, IEEE, *etc.*, for safety compliance
	E. Maintenance concerns
D	Automated allowance for maintenance operations and upgrade modules
W	Automated user manual generation based on component modules
	F. Mechanical-electronic requirements
D	Electronic assembly layout and the PCB board design must allow for the physical style and functionality of mechanical design (*i.e.* satisfy the imposed geometrical constraints)
D	Mechanical material selection and manufacturing accounts for the physical aspects of the internal electronic such that there is no device malfunctioning
D	Electronic CAD systems must support 3D modelling at components level and have facilities to export accurate 3D design data
D	Bi-directional flow of complete design information between mechanical CAE environment and electronic CAE environment

Table 2.9. Requirements for integrated mechatronic design (continued)

	G. Product lifecycle requirements
D	Frequent communication with the customers
D	Ability to carry out customisation requirement quickly and correctly
D	Robustness of the product: long mean time between failure and requires little maintenance
D	Enables standardisation and reusability
D	Effective end-of-lifecycle processing: recycling or reuse

D = Demands, must be satisfied during the design process
W = Wishes, would be ideal to satisfy during the design process, but not required

Efficiently bridging the gap between the mechanical and electronic design processes is, therefore, the key towards collaborative and successful product development. Rather than simply passing raw dimensioning and positional data from the ECAD to MCAD environment, it would be far more beneficial for the design tools to allow a bi-directional flow of comprehensive data between the two CAD environments. In other words, the ECAD must possess the ability to import and seamlessly integrate 3D component data from an MCAD environment, and then pass a full and accurate 3D representation of the board assembly back to the MCAD. To harness this potential, the electronics design system must support 3D modelling at the component level. This ability plus facilities to export accurate 3D design data would support the necessary interaction between the mechanical and electrical environments.

2.5.3.2 Product Lifecycle Requirements

Since industry networks can only get more complex and since information flow can only grow and not shrink, the following problems can be expected to exist in designing mechatronic systems:

- Material management: information maintenance is difficult, and so is maintaining the internal and external communication of the company regarding product data and the changes that have taken place in it; hence the following problems can occur.
- The item management of the company is not in order.
- The company's own component design and manufacturing is inefficient.
- Company buys the same type of components from different suppliers and ends up handling and storing these same types of components as separate items.
- Company makes overly fast and uncontrolled changes in the design of the product.

In order to resolve these problems listed above, the lifecycle of the product must be analysed. The useful life of a product can be measured by the length of its lifecycle. At the end of this lifecycle, the efficiency of the product is so low as to warrant the purchase of a newer one. The product can no longer be upgraded due to financial feasibility or lack of adequate updates. Lengthening this lifecycle will provide customers with a more useful product and will help reduce the strain on the environment from both the supply and the waste aspects. As the business world continues to grow more consumer-friendly, more products will have to become instantly customisable. To measure this mass customisation, a count of the number of variants into which a product can be customised gives a quick estimate of the abilities of its design method.

2.6 Conclusions

Mechatronic systems comprise of, among others, mechanical and electrical engineering components. An example is a robot arm that consists of mechanical links and electrical servos. Mechanical design changes lead to design modifications on the electrical side and *vice versa*. Unfortunately, contemporary CAE environments do not provide a sufficient degree of integration in order to allow for bi-directional information flow between both CAD domains.

There are several approaches to support the information exchange across different engineering domains. PDM systems manage product information from design to manufacture to end-user support. Standard data exchange formats are developed to achieving communication between various CAD, CAE, and PDM systems.

The approach to achieving integration of mechanical and electrical CAD systems proposed in this chapter is based on cross-disciplinary constraint modelling and propagation. Cross-disciplinary constraints between mechanical and electrical design domains can be classified, represented, modelled, and bi-directionally propagated in order to provide immediate feedback to designers of both engineering domains. In constraint classification, a selected mechatronic system is being analysed to identify and classify discipline-specific constraints based on associated functions, physical forms, system behaviour, and other design requirements. In constraint modelling, the mechatronic system is modelled in block-diagram form and relationships between domain-specific constraints are identified and categorised.

Most development activities for mechatronic products are of a collaborative nature. In the past, electrical engineers and mechanical engineers had to have personal interactions with each other to collaborate. Today, they collaborate through information systems. An integrated framework for designing mechatronic product helps to improve collaborative design activities. This collaboration can happen everywhere from within one office, one company, to world-wide distributed parties in a virtual environment.

References

[2.1] Appuu Kuttan, K.K., 2007, *Introduction to Mechatronics*, Oxford University Press, pp. 1–11.

[2.2] Reeves, B. and Shipman, F., 1992, "Supporting communication between designers with artifacts-centered evolving information spaces," In *Proceedings of the 1992 ACM Conference on Computer-supported Cooperative Work*, pp. 394–401.

[2.3] Liu, D.T. and Xu, X.W., 2001, "A review of web-based product data management systems," *Computers in Industry*, **44**, pp. 251–262.

[2.4] Sung, C.S. and Park, S.J., 2007, "A component-based product data management system," *International Journal of Advanced Manufacturing Technology*, **33**, pp. 614–626.

[2.5] You, C.F. and Chen, T.P., 2007, "3D part retrieval in product data management system," *Computer-Aided Design and Applications*, **3**(1–4), pp. 117–125.

[2.6] Pratt, M.J., Anderson, B.D. and Ranger, T., 2005, "Towards the standardised exchange of parameterised featured-based CAD models," *Computer-Aided Design*, **37**(12), pp. 1251–1265.

[2.7] Fenves, S.J., Sriram, R.D., Subrahmanian, E. and Rachuri, S., 2005, "Product information exchange: practices and standards," *Journal of Computing and Information Science in Engineering*, **5**(3), pp. 238–246.

[2.8] Lubell, J., Peak, R., Srinivasan, V. and Waterbury, S., 2004, "STEP, XML, and UML: complementary technologies," In *Proceedings of ASME 2004 Design Engineering Technical Conference and Computers and Information in Engineering Conference*.

[2.9] Sudarsan, R., Fenves, S.J., Sriram, R.D. and Wang, F., 2005, "A product information modelling framework for product lifecycle management," *Computer-Aided Design*, **37**(13), pp. 1399–1411.

[2.10] Biswas, A., Fenves, S.J., Shapiro, V. and Sriram, R.D., 2008, "Representation of heterogeneous material properties in the core product model," *Engineering with Computers*, **24**(1), pp. 43–58.

[2.11] Peak, R.S., Fulton, R.E., Nishigaki, I. and Okamoto, N., 1998, "Integrating engineering design and analysis using multi-representation architecture," *Engineering with Computers*, **14**(2), pp. 93–114.

[2.12] Kleiner, S., Anderl, R. and Grab, R., 2003, "A collaborative design system for product data integration," *Journal of Engineering Design*, **14**(4), pp. 421–428.

[2.13] ISO, 2001, "Product data representation and exchange. Integrated application resource: parameterisation and constraints for explicit geometric product models," *ISO Committee Draft (CD) 10303-108*, ISO TC 184/SC4/WG12/N940, Geneva, Switzerland.

[2.14] Schaefer, D., Eck, O. and Roller, D., 1999, "A shared knowledge base for interdisciplinary parametric product data models in CAD," In *Proceedings of the 12th International Conference on Engineering Design*, pp. 1593–1598.

[2.15] Roller, D., Eck, O. and Dalakakis, S., 2002, "Advanced database approach for cooperative product design," *Journal of Engineering Design*, **13**(1), pp. 49–61.

[2.16] Fulton, R.E., Ume, C., Peak, R.S., Scholand, A.J., Stiteler, M., Tamburini, D.R., Tsang, F. and Zhou, W., 1994, "Rapid thermomechancial design of electronic products in a flexible integrated enterprise," *Interim Report, Manufacturing Research Centre*, Georgia Institute of Technology, Atlanta.

[2.17] Mocko, G., Malak, R., Paredis, C. and Peak, R., 2004, "A knowledge repository for behaviour models in engineering design," In *Proceedings of ASME 2004 Design Engineering Technical Conference and Computers and Information in Engineering Conference*.

[2.18] Klein, R., 1998, "The role of constraints in geometric modelling," In *Geometric Constraint Solving and Applications*, Bruderlin, B. and Roller, D (eds), Springer-Verlag, pp. 3–23.

[2.19] Bolton, W., 2003, *Mechatronics: Electronic Control Systems in Mechanical and Electrical Engineering*, Pearson Education Limited, pp. 185–197.

[2.20] Centikunt, S., 2007, *Mechatronics*, John Wiley and Sons, pp. 393–395.

[2.21] Rembold, U., Nnaji, B.O. and Storr, A., 1993, *Computer Integrated Manufacturing and Engineering*, 3rd edn, Addison-Wesley, pp. 449–464.
[2.22] http://www.crustcrawler.com/product/arm5.php?prod=0

http://www.springer.com/978-1-84882-286-3

Collaborative Design and Planning for Digital Manufacturing
Wang, L.; Nee, A.Y.C. (Eds.)
2009, XX, 413 p., Hardcover
ISBN: 978-1-84882-286-3