A Holistic Concurrent Design Approach to Robotics using Hardware-in-the-loop Simulation

Robin Chhabra
Institute for Aerospace Studies
University of Toronto
4925 Dufferin Street
Toronto, Ontario, M3H 5T6 Canada
Email: chhabrar@utias.utoronto.ca

M. Reza Emami[*]
Institute for Aerospace Studies
University of Toronto
4925 Dufferin Street
Toronto, Ontario, M3H 5T6 Canada
Tel: +1-416-946-3357
Fax: +1-416-946-7109
Email: emami@utias.utoronto.ca (Corresponding author)

Abstract

This paper discusses a practical approach to the concurrent design of robot manipulators, which is based on an alternative design methodology, namely Holistic Concurrent Design (HCD), as well as the utilization of a modular hardware-in-the-loop simulation. Holistic concurrent design is a systematic design methodology for mechatronic systems that formalizes subjective notions of design, resulting in the simplification of the multi-objective constrained optimization process. Its premise is to enhance the communication between designers with various backgrounds and customers, and to consider numerous design variables with different natures concurrently. The methodology redefines the ultimate goal of design based on the qualitative notion of satisfaction, and formalizes the effect of designer's subjective attitude in the process. The hardware-in-the-loop platform involves physical joint modules and the control unit of a manipulator in addition to the software simulation to reduce modeling complexities and to take into account physical phenomena that are hard to be captured mathematically. This platform is implemented in the HCD design architecture to reliably evaluate the design attributes and performance supercriterion during the design process. The resulting architecture is applied to redesigning kinematic, dynamic and control parameters of an industrial manipulator.

Key Words:
Concurrent design,
Hardware-in-the-loop simulation,
Robot manipulator,
Multidisciplinary systems,
Mechatronics

1. Introduction

Robotic systems are inherently multidisciplinary, and designers often employ a subsystem-partitioning approach for their analysis and synthesis [1]. Although conventional *decoupled* or *loosely-coupled* approaches of design offer practical solutions, they undermine the interconnection between various subsystems that indeed plays a crucial role in mechatronics. The close interaction and communication of the subsystems necessitate a concurrent approach to the synthesis of such systems, where the participating disciplines are offered equal opportunities to contribute to the final design. The resulting *synergy* due to the integration of different disciplines in concurrent design has been documented, and it has been demonstrated that the design outcome can be new and previously unattainable [2]. However, the challenge is to consider a large number of design variables and multidisciplinary objective and constraint functions and handle a prohibitively-complicated mechatronic model.

Within the context of robotics a number of concurrent design approaches have been suggested, some of which attempt to solve a multi-objective constrained optimization model. For instance, in [3] and [4] evolutionary algorithms are used to design modular serial and parallel robotic arms, respectively. In [5] an architecture is introduced for designing reconfigurable robots based on Axiomatic Design theory [6]. Genetic Algorithms are also employed to perform the multi-objective optimization for the design of parallel robot arms [7,8], mechatronic systems [9], and reconfigurable robotic systems [10]. Some innovative designs of robotic systems have also been reported, such as some novel design of parallel [11,12], and hyper-redundant robot manipulators [13], M-TRAN [14], and a spring-assisted modular reconfigurable robot [15].

In the field of engineering design, a number of concurrent design methodologies have been introduced that attempt to take into account subjective aspects of design. For instance, some design methodologies employ fuzzy-logic tools to offer solutions to the conceptual and preliminary phases of concurrent design problem [16-21]. As for the conceptual phase of design, a fuzzy outranking preference model was developed to model the imprecise preference relation between design solutions [17]. In addition, fuzzy outranking index was defined for different design criteria and they were aggregated with an adjustable strategy (by choosing a degree of optimism and aggressiveness) to find the best set of design solutions [18]. A notable example for the preliminary phase of design is the Method of Imprecision (MoI), which takes into account the imprecision in design [19]. This approach, which has been used in various engineering applications such as product planning [20,21], defines a set of designer's preference [22] for design variables and performance parameters to model the imprecision in design. It determines and maximizes the global performance under one of the two conservative or aggressive design tradeoff strategies, and uses fuzzy connectives for tradeoff in the design space [23].

Several approaches have also been suggested by the researchers in different communities to tackle the challenge of high dimensionality in concurrent design. These approaches can be divided into two major groups. The first group tries to alleviate the complexity by reducing the optimization space; either through breaking the optimization process into several stages [24], or by approximating the design space with a lower dimensional space [25]. The second group translates the model complexity into a large volume of computations, and then attempts to find efficient algorithms or parallel-processing techniques to make these computations feasible. For example, parallel genetic algorithms are used for multi-objective optimizations in [26]. Each group brings certain contributions to concurrent design, yet cannot avoid some shortcomings. While efficient algorithms, mostly taking advantage of parallel processing, can handle high computational demands in concurrent design, they tend to lose transparency in design. On the other hand, a better understanding of design may be achieved, should one be able to simplify the optimization model, but at the cost of obtaining approximated outcomes.

This paper introduces a solution for the complexity of concurrent design in the preliminary phase, which in essence consists of two unique constituents, each relating to one of the above-mentioned groups. For the first part, the solution applies an alternative design methodology, namely *Holistic Concurrent Design* (HCD), which not only formalizes subjective notions and brings the subjective aspects of communication into the design process, but also transforms the multi-objective constrained optimization problem into two single-objective unconstrained formulations. This methodology enhances communication between designers from various disciplines through introducing the universal notion of *satisfaction* and expressing the holistic behaviour of mechatronic systems using the notion of *energy*. As for the second part, it utilizes an efficient, computable system model to compute the objective and constraint functions in the design process, which is called *Robotic Hardware-In-the-Loop Simulation* (RHILS). This method uses real hardware modules in a real-time simulation loop to account for complex phenomena such as sensor noise, actuator limitation, transmission flexibility, etc., which can hardly be captured by computational modeling. A combination of the above two techniques will ensure an efficient solution for concurrent design of robot manipulators.

The outline of this paper is as follows: Section 2 introduces the HCD methodology. Section 3 details the Robotic Hardware-in-the-loop Simulation (RHILS) platform. Section 4 describes the HCD-RHILS-based concurrent design framework and its application to an industrial robot manipulator. Some concluding remarks are made in Section 5.

2. Holistic concurrent design: an alternative approach to concurrent design

This section outlines an alternative concurrent design framework for the preliminary phase of design, which emphasizes on the designer's satisfaction, instead of pure performance optimization. It employs fuzzy membership functions and connectives to formalize subjective aspects of design, resulting in the simplification of the multi-objective constrained optimization of a design process. This methodology redefines the concurrent design problem for a mechatronic system as: concurrently, find the design variables corresponding to different subsystems of a mechatronic system such that the system performs satisfactorily considering a set of design requirements and constraints. Few approaches of concurrent design have attempted to include subjective notions in the design process, [e.g., 16-21]. They, however, lack a systematic framework for the reconciliation between designer's subjective attitude and system's objective performance. In the proposed methodology, design attributes are divided into two inherently-different classes, namely *wish* and *must* attributes, and parametric fuzzy-logic connectives are used to aggregate the corresponding satisfactions so that the designer's attitude can be adjusted based on an objective performance criterion.

2.1. Fuzzy connectives and fuzzy aggregation

Unlike the classic set theory where the membership of an element in a set is binary, in fuzzy set theory the membership of an element in a fuzzy set can be *partial*. Accordingly, the classic logical connectives AND, OR and NOT are generalized as functions of the membership degrees to perform operations on fuzzy sets. In fuzzy logic, AND and OR connectives have been interpreted through different classes of *triangular norms* (t-norm) and *triangular conorms* (t-conorm), respectively, such as *Max-Min Operators* (T_{min},S_{max}), *Algebraic Product and Sum* (T_{prod},S_{sum}), and *Drastic Product and Sum* (T_W,S_W). Using the basic properties of these operators, it can be shown that for any arbitrary t-norm T and t-conorm S and for all $a_i \in [0,1]$ (i=1,…,n) [27],

$$\begin{aligned} &T_W(a_1,...,a_n) \le T(a_1,...,a_n) \le T_{min}(a_1,...,a_n), \\ &S_{max}(a_1,...,a_n) \le S(a_1,...,a_n) \le S_W(a_1,...,a_n). \end{aligned} \tag{1}$$

To represent the range of operators in (1), various types of parametric formulations have been suggested in the literature. In particular, a class of parametric operators for fuzzy reasoning is introduced in [28] whose parametric t-conorm operator is defined as

$$S^{(p)}(b_1,...,b_n) \equiv [b_1^p + (1-b_1^p)[...[b_{n-2}^p + ... + (1-b_{n-2}^p)[b_{n-1}^p + (1-b_{n-1}^p)b_n^p]]...]]^{1/p}, \tag{2}$$

where $b_i \in [0,1]$ and p>0, and the corresponding parametric t-norm operator is defined based on De Morgan laws using standard complementation operator (NOT connective), i.e., $C(a) = 1-a \quad \forall a \in [0,1]$, as

$$T^{(p)}(a_1,...,a_n) = 1 - S^{(p)}((1-a_1),...,(1-a_n)) \, . \tag{3}$$

In the extreme cases, ($T^{(p)}$,$S^{(p)}$) approaches (T_{min},S_{max}) as $p \to +\infty$, (T_{prod},S_{sum}) as $p \to 1$, and (T_W,S_W) as $p \to 0$.

In fuzzy logic, the meaning of a connective can be neither pure OR (t-conorm) nor AND (t-norm), with its complete lack of compensation. Such connectives are called *mean operators*. As an example, a parametric operator of this class, namely *generalized mean operator*, is defined as

$$G^{(\alpha)}(b_1,...,b_n) \equiv \left(\frac{1}{n} \sum_{i=1}^{n} b_i^{\alpha} \right)^{1/\alpha} ; \tag{4}$$

where $b_i \in [0,1]$ and $\alpha \in \mathbb{R}$ [27]. It appears that this type of connective monotonically varies between T_{min} operator as $\alpha \to -\infty$ and S_{max} operator as $\alpha \to +\infty$.

In the HCD process, the parametric operators in (2),(3), and (4) are adopted to aggregate satisfactions, and introduce the notion of *overall satisfaction* for a design solution.

2.2. Formulation of design process

A design problem consists of two sets: *design variables* $X \equiv \{X_1,..., X_n\}$ and *design attributes* $A \equiv \{A_1,..., A_N\}$ such that any design solution can be identified by vectors $\boldsymbol{X} \equiv [X_1,..., X_n]^T \in \mathbb{R}^n$ and $\boldsymbol{A} \equiv [A_1,..., A_N]^T \in \mathbb{R}^N$. In this paper vectors are denoted by bold letters. Design variables are some system parameters (corresponding to the system being designed) that should be assigned in a design process to satisfy the *design requirements*. And design attributes are some performance parameters and constraints of the system associated with the design requirements to quantify the performance of the system at any design state. In a design process, design variables are subject to *design availabilities* $D \equiv D_1 \times ... \times D_n$, such that $D_j \subset \mathbb{R}$ (j=1,…,n), which specify feasible design variables. For a mechatronic system, the design availabilities are usually defined based on the nature of the design variables (e.g., a link length of a robot is always a positive number) or manufacturing, environmental or structural restrictions. For each design attribute A_i there is a mapping $F_i : \mathbb{R}^n \to \mathbb{R}$ that relates a design state $\boldsymbol{X}$ to the i^{th} attribute, i.e., $A_i = F_i(\boldsymbol{X})$ (i=1,…,N). These functional mappings can be of any form, such as closed-form equations, heuristic rules, or sets of experimental or simulated data. Indeed, a design problem can be modeled as a multi-objective optimization subject to a number of constraints due to the design availabilities and design requirements specified by the customer.

$$\min_{\boldsymbol{X}\in D}[F_1(\boldsymbol{X}),...,F_{N_W}(\boldsymbol{X})]^T \quad \text{subject to } \{F_i(\boldsymbol{X})\in G_{i-N_W} : G_{i-N_W}\subset \mathbb{R}, i=N_W+1,...,N\}; \tag{5}$$

where N_W and $N_M \equiv N - N_W$ are the number of attributes that should be optimized and the number of constraints, respectively.

Definition 1 (Satisfaction):

a) A mapping $\mu_{X_j}:\mathbb{R}\to[0,1]$ for the design variable X_j is called satisfaction if for any two different values $X_{j1},X_{j2}\in\mathbb{R}$ one has $\mu_{X_j}(X_{j1})>\mu_{X_j}(X_{j2})$ if $X_{j1}\succ X_{j2}$ or $\mu_{X_j}(X_{j1})=\mu_{X_j}(X_{j2})$ if $X_{j1}\approx X_{j2}$. The symbols $\succ$ and $\approx$ denote *strictly superior* and *as superior*, respectively, which are interpreted based on the design availabilities.

b) A mapping $\mu_{A_i}:\mathbb{R}\to[0,1]$ for the design attribute A_i is called satisfaction if for any two different states of the design variables $\boldsymbol{X}_1,\boldsymbol{X}_2\in\mathbb{R}^n$ one has $\mu_{A_i}\circ F_i(\boldsymbol{X}_1)>\mu_{A_i}\circ F_i(\boldsymbol{X}_2)$ if $F_i(\boldsymbol{X}_1)\succ F_i(\boldsymbol{X}_2)$ or $\mu_{A_i}\circ F_i(\boldsymbol{X}_1)=\mu_{A_i}\circ F_i(\boldsymbol{X}_2)$ if $F_i(\boldsymbol{X}_1)\approx F_i(\boldsymbol{X}_2)$, where $\succ$ and $\approx$ are interpreted based on the design requirements. The symbol $\circ$ is the composition operator. For brevity, in this paper the satisfaction for a design variable is denoted by $x_j(X_j)\equiv\mu_{X_j}(X_j)$, and the satisfaction for a design attribute is denoted by $a_i(\boldsymbol{X})\equiv\mu_{A_i}\circ F_i(\boldsymbol{X})$. A value of one for a satisfaction corresponds to the ideal case or the most satisfactory situation, and the value zero means the worst case or the least satisfactory design variable or attribute. ■

The design requirements are divided into *demands* and *desires*. A designer defines a proper set of design attributes to accommodate the design requirements, and chooses the mappings F_i's to relate the design variables to the design attributes, in different phases of design. Accordingly, in the HCD methodology the design attributes are divided into two subsets, labelled as *wish* and *must* design attributes. A *wish* attribute corresponds to a designer/ customer's desire, i.e., its associated design requirement should be satisfied as much as possible. All *wish* attributes form a set $W\equiv\{W_1,...,W_{N_W}\}$ whose associated vector $[W_1,...,W_{N_W}]^T\equiv[F_1(\boldsymbol{X}),...,F_{N_W}(\boldsymbol{X})]^T$ is optimized. A *must* attribute refers to designer/customer's demand, i.e., the achievement of its associated design requirement is mandatory. All *must* attributes form a set denoted as $M\equiv\{M_1,...,M_{N_M}\}$, and they should usually satisfy inequalities, i.e., $M_i\equiv F_{N_W+i}(\boldsymbol{X})\in G_i\subset\mathbb{R}\quad(i=1,...,N_M)$. To distinguish between *must* and *wish* satisfactions, the satisfactions specified for *wish* attributes are denoted by $w_i(\boldsymbol{X})$ (i=1,…,N_W), and the satisfactions corresponding to *must* attributes are denoted by $m_i(\boldsymbol{X})$ (i=1,…,N_M). As an example, to design a robot manipulator (see Section 4) a designer may consider the end-effector position error as a *wish* attribute and the control torque at the joints as a *must* attribute. Hence, the less the end-effector position error and the farther the control torque at a joint from the (positive and negative) stall torque, the larger the corresponding satisfaction is defined.

The satisfactions can be considered as fuzzy membership functions over the universes of discourse of design variables and design attributes, and accordingly suitable fuzzy connectives can be utilized to aggregate the satisfactions. For a design state

$\boldsymbol{X}$, the *overall satisfaction*, as a global measure of design achievement, is the aggregation of *wish* and *must* satisfactions and the satisfactions for the design variables with proper fuzzy connectives.

2.3. Calculation of the overall satisfaction

In this sub-section, separate aggregation strategies are suggested for combining satisfactions corresponding to *must* and *wish* design attributes in order to introduce the *overall must* and *wish satisfactions*. Subsequently, they are aggregated to calculate the overall satisfaction in design.

2.3.1. Aggregation of must design attributes

Must attributes correspond to those design requirements that are to be satisfied simultaneously, with no room for negotiation. Therefore, for aggregating *must* satisfactions an AND logical connective is suitable. Considering *must* satisfactions as fuzzy membership degrees, the AND connective can be interpreted through a family of t-norm operators.

Axiom 1: For a design state $\boldsymbol{X}$, the *overall must satisfaction* $\mu_M^{(p)}(\boldsymbol{X})$ (p>0) is the aggregation of *must* satisfactions and the satisfactions of the design variables using the p-parameterized class of t-norm operators defined by (2) and (3). Note that, availability of design variables is considered as a part of *must* attributes.

$$\mu_M^{(p)}(\boldsymbol{X}) = T^{(p)}(m_1(\boldsymbol{X}),..., m_{N_M}(\boldsymbol{X}), x_1(X_1),..., x_n(X_n)) \tag{6}$$

Changing the value of p makes it possible to obtain different tradeoff strategies. Larger values of p would imply a more conservative attitude, and values of p closer to zero represent a more aggressive attitude towards aggregating *must* design attributes. ■

2.3.2. Aggregation of wish design attributes

Definition 2 (Cooperative *wish* attributes): For a design state $\boldsymbol{X}$, a subset of *wish* design attributes is called *cooperative* if the corresponding perturbed satisfactions have the same sign (positive or negative) for equal infinitesimal positive perturbations of the design variables. Thus, the set of *wish* attributes is a disjoint union of two cooperative subsets:

a) *Positive-differential wish attributes*: For a design state $\boldsymbol{X}$, positive-differential subset of *wish* attributes contains those with non-negative perturbed satisfactions, i.e.,

$$W_{\boldsymbol{X}}^{+} \equiv \{W_i \in W : \sum_{j=1}^{n} \frac{\partial w_i}{\partial X_j}(\boldsymbol{X}) \geq 0\}\,. \tag{7}$$

This subset consists of all *wish* design attributes that tend to reach a higher satisfaction when all design variables have equal infinitesimal increments.

b) *Negative-differential wish attributes*: For a design state $\boldsymbol{X}$, negative-differential subset of *wish* attributes contains those with negative perturbed satisfactions, i.e.,

$$W_{\boldsymbol{X}}^{-} \equiv \{W_i \in W : \sum_{j=1}^{n} \frac{\partial w_i}{\partial X_j}(\boldsymbol{X}) < 0\}\,. \tag{8}$$

This subset includes all *wish* attributes that tend to reach a lower satisfaction when all design variables have equal infinitesimal increments. ■

Since *wish* attributes are cooperative in each positive- or negative-differential subset, their corresponding design requirements can be fulfilled simultaneously. According to Axiom 1, a q-parameterized class of t-norm operators is suitable for aggregating satisfactions in either subset of *wish* attributes. Therefore, the *overall positive-* and *negative-differential wish satisfactions* are defined by

$$\mu_{W^{\pm}}^{(q)}(\boldsymbol{X}) \equiv T^{(q)}(w_1(\boldsymbol{X}),...,w_{N_{W^{\pm}}}(\boldsymbol{X})) \quad q > 0\,, \tag{9}$$

where $N_{W^{\pm}}$ are the number of members of the sets $W_{\boldsymbol{X}}^{\pm}$. Note that, these numbers may vary by $\boldsymbol{X}$.

The two subsets of *wish* attributes cannot be improved simultaneously as their design requirements compete with each other. Therefore, some compromise is necessary for aggregating their satisfactions. A class of mean operators reflects the averaging and compensatory nature of their aggregation.

Axiom 2: For a design state $\boldsymbol{X}$, the *overall wish satisfaction* $\mu_W^{(q,\alpha)}(\boldsymbol{X})$ can be calculated using the α-parameterized generalized mean operator defined by (4),

$$\mu_W^{(q,\alpha)}(\boldsymbol{X}) = \left[\frac{1}{2}\left(\left(\mu_{W^+}^{(q)}(\boldsymbol{X})\right)^{\alpha} + \left(\mu_{W^-}^{(q)}(\boldsymbol{X})\right)^{\alpha}\right)\right]^{1/\alpha} \quad \alpha \in \mathbb{R}. \tag{10}$$

This class of generalized mean operators offers a variety of aggregation strategies from conservative (T_{min}) to aggressive (S_{max}). The overall *wish* satisfaction is governed by two parameters q and α, which represent subjective tradeoff strategies. Larger values of α or smaller values of q represent a more optimistic (aggressive) attitude in the design process, and vice versa. ∎

2.3.3. Aggregation of overall must and wish satisfactions

The aggregation of all *wish* satisfactions can be considered as one *must* attribute, i.e., it has to be fulfilled with other *must* attributes with no compromise. Therefore, based on the Axiom 1, the overall satisfaction $\mu^{(p,q,\alpha)}(\boldsymbol{X})$ is quantified by aggregating the overall *must* and *wish* satisfactions with the p-parameterized class of t-norm operators defined by (2) and (3), i.e.,

$$\mu^{(p,q,\alpha)}(\boldsymbol{X}) = T^{(p)}(\mu_M^{(p)}(\boldsymbol{X}), \mu_W^{(q,\alpha)}(\boldsymbol{X}))\,. \tag{11}$$

In (11), three parameters, p, q and α, called *attitude parameters*, govern the overall satisfaction, and they represent various tradeoff strategies in design.

2.4. Optimization of the overall satisfaction

In the first phase of the HCD methodology, the design formulation in (5) is formally reduced to a single-objective unconstrained maximization of the overall satisfaction, as determined in the previous sub-section. One can employ any standard optimization method to perform this optimization. The locally-unique solution of

$$\mu^{(p,q,\alpha)}(\boldsymbol{X}_s) = \max_{\boldsymbol{X}\in\mathbb{R}^n} T^{(p)}(\mu_M^{(p)}(\boldsymbol{X}), \mu_W^{(q,\alpha)}(\boldsymbol{X})) \tag{12}$$

is called a *satisfactory design alternative*. As the result of the way satisfactions are defined and aggregated, the locally-unique solution to (12) is locally pareto-optimal for (5). In (12), various attitude parameters result in different optimum solutions. Hence, $\boldsymbol{X}_s$ is implicitly a function of the attitude parameters. The set of satisfactory design alternatives generated by changing the attitude parameters is denoted by $C_s \equiv \{\boldsymbol{X}_s(p,q,\alpha) : p,q > 0,\ \alpha \in \mathbb{R}\}$.

2.5. Performance supercriterion

In the second phase of the HCD methodology, the best design needs to be selected from C_s through the optimization of a proper objective function. In the previous phase of design, decision-making was biased by the designer/costumer's preference (satisfaction membership functions) and designer's attitude (aggregation parameters). Hence, in this phase of design the outcome must be checked against a holistic performance supercriterion. Indeed, such a supercriterion adjusts the designer's attitude based on the physical performance of the system. As the synergy in concurrent design necessitates, a suitable supercriterion should take into account interconnections between all subsystems of the mechatronic system [29]. Although such systems consist of various subsystems in different physical domains, the universal concept of energy and energy exchange is common to all of their subsystems [30,31]. Therefore, an energy-based model can deem all subsystems together with their interconnections and introduce generic performance supercriteria suitable for concurrent design. In [29] *bond graphs* are used to define three holistic criteria, which are reviewed in sequel.

2.5.1. Energy

Any mechatronic system is designed to perform a certain amount of *effective work* $EW(\boldsymbol{X})$ on its environment while it receives the *supplied energy* $SE(\boldsymbol{X})$. Not all of $SE(\boldsymbol{X})$ transforms to $EW(\boldsymbol{X})$, a portion of it is stored or dissipated in the system elements and transacted with the environment through physical constraints or external fields. This *cost energy* $CE(\boldsymbol{X})$ in any system is the overhead energy for performing the effective work. Therefore, $CE(\boldsymbol{X})$ can be considered as a supercriterion, coined as *energy supercriterion*, which should be minimized. For a pre-defined effective work, i.e., when EW is independent of $\boldsymbol{X}$,

$$SE(\boldsymbol{X}) = EW + CE(\boldsymbol{X}). \tag{13}$$

Therefore, by minimizing the supplied energy with respect to the attitude parameters the best design can be achieved in C_s [29].

2.5.2. Entropy

After a slight perturbation of the supplied energy, an energy system reaches its equilibrium state once the entropy generation of the system approaches its maximum. While the system moves toward the equilibrium, its capability of performing effective work on the environment reduces continuously. Therefore, the less the *work loss* of a system, the higher its aptitude is to do effective work. This work loss is equal to the irreversible heat exchange $Q_{irr}(t_{eq}(\boldsymbol{X}), \boldsymbol{X})$ at the dissipative elements of the system, where t_{eq} is defined as follows [29]: given a unit step change of the supplied energy, the equilibrium time $t_{eq}(\boldsymbol{X})$ is the time instant after which the rate of change of dissipative heat remains below a small threshold ε, i.e.,

$$t_{eq}(\boldsymbol{X}) = Inf\{t_0 : \forall t > t_0, \frac{\partial Q_{irr}}{\partial t}(t, \boldsymbol{X}) < \varepsilon\}\,. \tag{14}$$

The $Q_{irr}(t_{eq}(\boldsymbol{X}), \boldsymbol{X})$ can also be considered as a holistic criterion, and it is called *entropy supercriterion*. Using this supercriterion, the best design can be attained in C_s.

2.5.3. Agility

For mechatronic systems whose response time is a crucial factor, the rate of energy transmission through the system, or *agility*, can be a holistic measure of design. Thus, the supercriterion is defined as the time that the system takes to reach a steady state after a unit step change of some or all input parameters. A system is in the steady state when the rate of change of *introversive dynamic energy* $K(t,\boldsymbol{X})$ is zero. The introversive dynamic energy is the kinetic energy of masses for mechanical subsystems or the energy stored in inductors in electrical subsystems [29]. Given a unit step change of input variables, the response time, denoted by $T(\boldsymbol{X})$, is the time instant after which the rate of change of introversive dynamic energy remains below a small threshold δ, i.e.,

$$T(\boldsymbol{X}) = Inf\{t_0 : \forall t > t_0, \frac{\partial K}{\partial t}(t,\boldsymbol{X}) < \delta\} . \tag{15}$$

As a design supercriterion, when the response time reaches its minimum value with respect to attitude parameters the best design is attained in C_s.

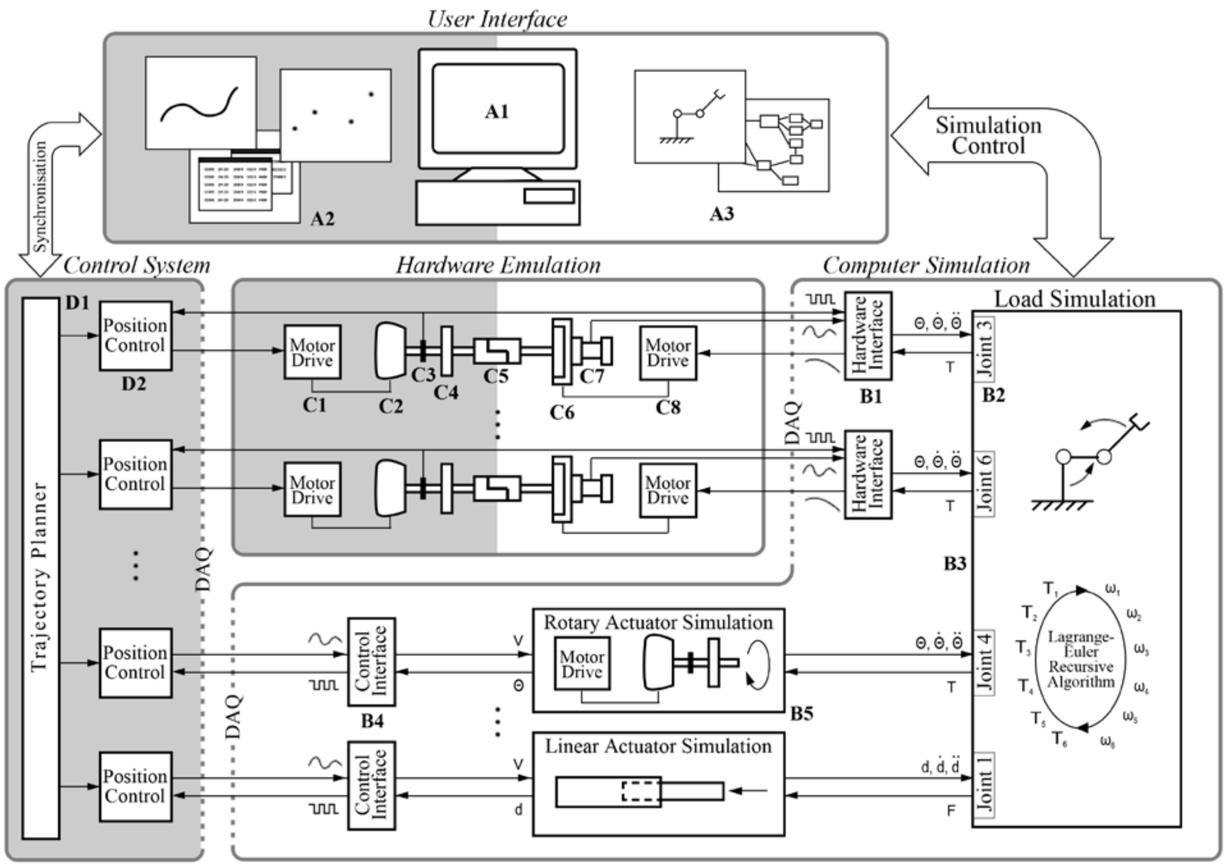
3. Robotic hardware-in-the-loop simulation platform

There is an increasing demand in today's industry for reducing development time, failure costs, and computational resources for system simulations. These factors have led to an increase in the utilization of Hardware-In-the-Loop (HIL) simulations for design, testing and training purposes [32]. By using physical hardware as part of a computer simulation, it is possible to reduce the complexity of the simulation and incorporate factors that would otherwise be difficult or impossible to model. The HIL simulations have been successfully applied in many areas, including aerospace [33], controls [34], manufacturing [35], and naval and defense [36]. They have been proven as useful design tools that can reduce development time and costs [35].

In the field of robotics, HIL simulations have been applied from a number of different perspectives. These approaches include: *robot-in-the-loop* simulations, such as the use of both real and simulated mobile robots interacting in a virtual environment [37]; *controller-in-the-loop* simulations, where a real control unit interacts with a computer model of the robot [38]; and *joint-in-the-loop* simulations, which emulate the loads on the real actuators, at the joints [39]. Although each of these applications employs the HIL concept slightly differently, all of them have led to an improvement of the results. In [40,41], a modular and generic *Robotic HIL Simulation* (RHILS) platform was designed and developed for industrial manipulators. Its performance was verified using the CRS CataLyst-5 industrial manipulator [42]. The RHILS is used in this paper as the second constituent of robotic concurrent design framework, next to the HCD methodology. The architecture of the RHILS platform is illustrated in Figure 1, and an overview of its modules is presented below.

3.1. RHILS architecture

The RHILS platform allows for simultaneous design and testing of both the joint hardware and control system of a robot manipulator. This architecture is designed to be adequately generic so that it can be applied to any serial-link robot manipulator system, and it focuses on modularity and extensibility in order to facilitate concurrent design of a wide range of manipulators. This platform consists of four blocks: **a)** *user interface*, **b)** *computer simulation*, **c)** *hardware emulation*, and **d)** *control system*. These subsystems are further partitioned into two major categories: *RHILS platform components* (white



A1 User interface host computer
A2 Control system user interface and trajectory setup
A3 Simulation user interface and scheduler
B1 Motor interface block, converts between actual hardware signals and the standardized form used in the simulation
B2 Joint assignment for the module
B3 Inverse dynamics simulation
B4 Control interface block, converts between actual control signals and the standardized form used with simulated actuators
B5 Simulated model of an actuator, for cases where the hardware module is unavailable, impractical, or unnecessary
C1 Drive electronics for Test Motor
C2 Test Motor
C3 Differential rotary encoder
C4 Harmonic drive transmission
C5 Detachable coupling to allow test hardware to be swapped in and out
C6 Load Motor
C7 Reaction torque transducer, for closed loop control and data acquisition
C8 Drive electronics for Load Motor
D1 Trajectory planner
D2 Position controller
A gray background indicates that section is part of the system being designed and tested using the RHIL platform

Figure 1. RHILS Platform Architecture

background in Figure 1), and *test system components* (grey background in Figure 1). The RHILS platform components are generic and should remain largely consistent over multiple applications, while the test system components are part of the system being designed and/or tested.

3.1.1. User interface block

This block acts as an intermediary between the control system and the computer simulation. On the RHILS platform side, robot configurations and parameters are chosen and any external conditions, e.g., zero-gravity or end-effector payload, are specified. On the test system side, any configurable control parameters, such as the planned trajectories and control gains, are set in the controller. Finally, the duration of the simulation and the type of data logging are selected.

3.1.2. Computer simulation block

The computer simulation performs three roles. Its primary task, represented by the load simulation block, is to run the inverse dynamics computations, and solve for the dynamic load applied to each joint actuator. Due to the recursive algorithm used for computing the inverse dynamics [43], it is possible to specify any reasonable number of joints and still run the simulation in real-time. The second task is to convert the hardware signals read in and sent out through a data acquisition board into the standard format for the load emulation, which is shown by the hardware interface blocks. These hardware interface blocks play a key role in the modularity of the architecture. The third task of the computer simulation block is to simulate any joints that do not have a corresponding hardware module. This task makes it possible to utilize the RHILS architecture at early stages of design as well as making it cost effective to set up tests if only one section of the manipulator is under study.

3.1.3. Hardware emulation block

The hardware emulation block consists of separate modules for each joint, and each module interfaces with both the control system and the computer simulation. These modules are further separated into two parts: a test module and a load module consisting of the load-emulating devices. The test module includes not only the real actuators, but also the transmission systems, position/speed sensors, and motor drives that would be used in the real manipulator, all of which can lead to significant inaccuracies in a pure computer simulation. The test module interfaces directly with the control system, which controls the motor as if it were part of an actual robot. The load module is coupled to the output of the transmission system. This module is controlled through a feedback loop to follow the calculated torque in the computer simulation block. This torque represents the arm dynamics that must be reflected on each joint actuator to have a genuine simulation of the robot [41].

3.1.4. Control system block

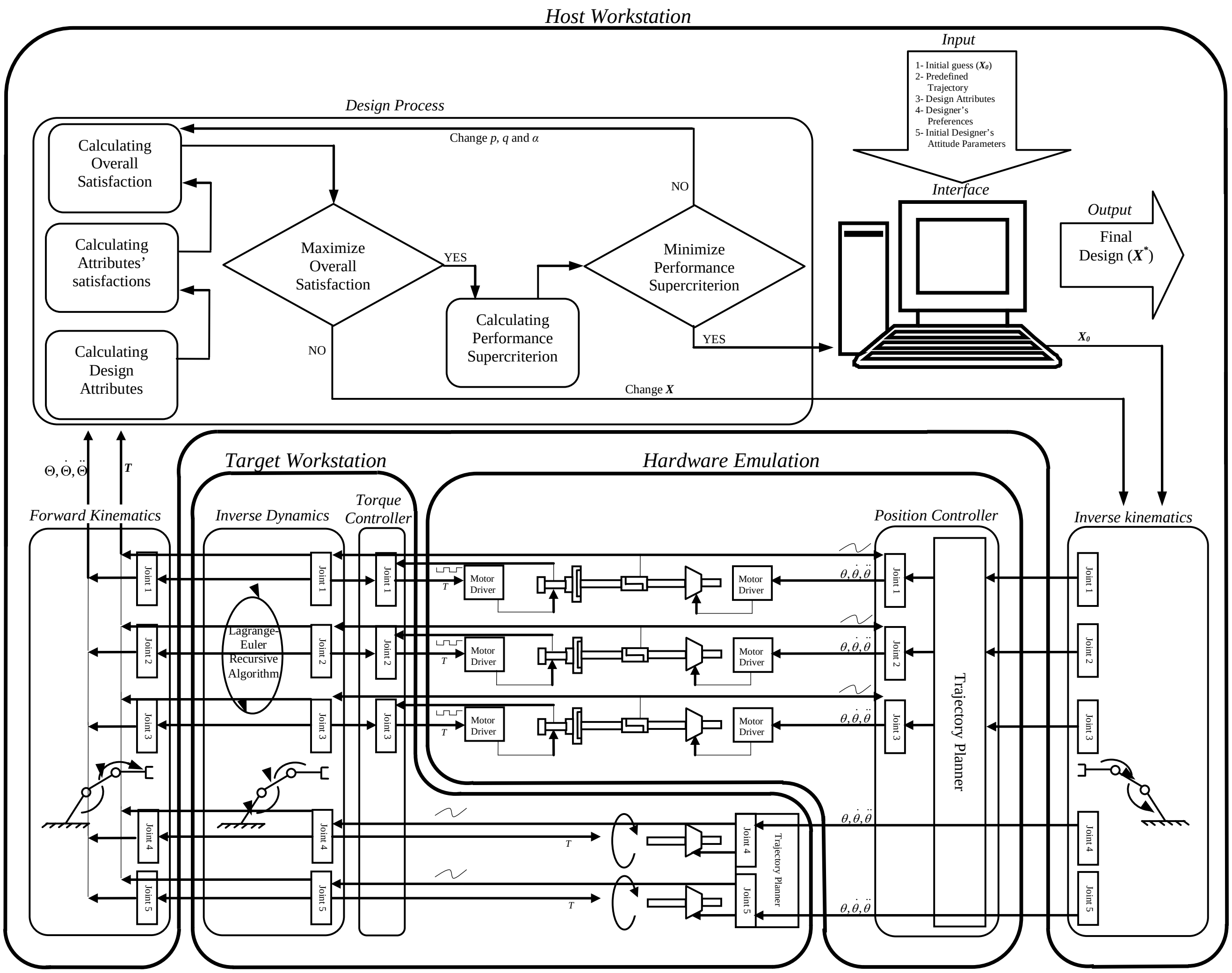
This block can range from standard PC software to a dedicated custom hardware depending on the nature and requirements of the application. In the present platform, the real control system for CRS CataLyst-5 robot is implemented. When the control system is not the focus of the design the flexibility of this architecture allows any simple controller to be quickly implemented and used.

4. HCD-RHILS: concurrent design of manipulators

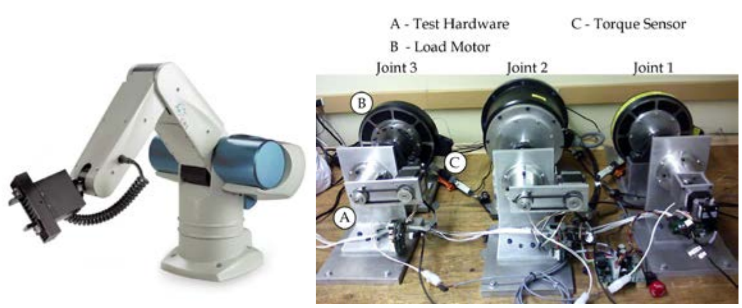
In this section, the HCD methodology and the RHILS platform are implemented to construct a concurrent design framework for serial-link robot manipulators. This framework includes two phases of the HCD, and the RHILS platform is used to evaluate the design attributes and the performance supercriterion.

4.1. HCD-RHILS architecture

The architecture of the concurrent design framework consists of two parallel workstations, namely *host* and *target*, plus *hardware emulation* block, which is a collection of load emulators, joint modules and controller unit of CRS CataLyst-5 manipulator. The entire HCD-RHILS concurrent design architecture is depicted in Figure 2(a).



(a)



(b)

Figure 2. (a) The design architecture, (b) RHILS platform versus CRS CataLyst-5 robot

4.1.1. Host workstation

The host workstation is the link between the design platform and the engineer(s). All design preferences and options are set in this block, where the main code that governs the design process is executed. The design preferences are reflected in the satisfactions defined on the space of design variables and attributes, and the simulation options consist of initial configuration, the predefined end-effector trajectories, gravity, payload, and the simulation time. This block communicates with the controller to load control gains through an FTP connection, and sends the command signals to the trajectory planner using Python® software. It also loads the kinematic and dynamic parameters and the inverse dynamic model of a design candidate on the target workstation via a TCP/IP connection. In addition, it collects the position and torque data that are saved on the target PC using MATLAB® xPC Target® toolbox. The design attributes and performance supercriterion are calculated in the host computer. Accordingly, the overall satisfaction of the design candidate is calculated. The MATLAB® optimization toolbox is used to optimize the overall satisfaction and the performance supercriterion.

4.1.2. Target workstation

This block is a barebones PC running the xPC Target® real-time kernel. On this workstation a servo torque controller for the load emulators is mounted. An inverse dynamics model of the manipulator is also executed on this computer. This model is generated in Simulink® and compiled through the Real-Time Workshop®. The Target computer contains several interface boards to communicate with the joint modules and load emulators. Further, a data acquisition board and an RS232 port is used to gather the data from the hardware components.

4.1.3. Hardware emulation

All physical components that remain unchanged in the design process form the hardware emulation block. Industrial manipulators often have five or six degrees of freedom (d.o.f.). The first three joints are often used to position the end-effector and the last joints help the wrist change its orientation. Since the first three links are more massive, they play a crucial role in the dynamics of a manipulator. Hence, in the design architecture, the first three joint modules of CRS CataLyst-5 are physically included, and the rest of the joints are virtually modeled on the target workstation. The load emulators are coupled to the joints and the CRS DM Master Controller unit is used to control the joints position.

The controller unit includes a trajectory planner and a typical feedback/feedforward controller for each physical joint module. The trajectory planner generates instantaneous desired position signals with a frequency of 1 KHz based on the input of the controller.

4.2. HCD-RHILS concurrent design process

In this sub-section, the concurrent design process of robot manipulators based on HCD-RHILS framework is detailed. The concurrent design problem can be stated as follows. *Concurrently*, find the kinematic, dynamic and control parameters (design variables) of an m d.o.f. serial-link manipulator consisting of revolute and/or prismatic joints such that it *satisfactorily* follows N_T predefined trajectories with a preset load at the end-effector. In other words, find the *satisfactory design alternative* associated with a set of design *requirements* (*wish* attributes) and *constraints* (*must* attributes and design availabilities) that minimizes a *performance supercriterion*.

As an illustrative example the kinematic, dynamic and control parameters of CRS CataLyst-5 manipulator are redesigned concurrently. This industrial manipulator consists of 5 rotary joints, three of which are physically included in the RHILS platform. Figure 2(b) shows CRS CataLyst-5 manipulator next to its RHILS platform. The set of desired end-effector trajectories considered in this case study includes step, ramp, pick and place and periodic and the preset load at the end-effector equals 1 Kgf.

The HCD methodology consists of four steps that are detailed in sequel: **a)** decision about design variables and attributes,

b) assignment of satisfactions, **c)** calculation and maximization of the overall satisfaction, and **d)** minimization of the performance supercriterion.

4.2.1. Design variables and attributes

The kinematic characteristics of an *m* d.o.f. serial-link manipulator can be represented by the standard Denavit-Hartenberg convention [44]. Therefore, length (l_k), offset/ angle (d_k/θ_k) and twist (α_k) are considered as kinematic design variables of the k^{th} link (k=1,...,m). In order to consider dynamic parameters of the robot, each link is modeled as an L-shaped circular cylinder along the link length and offset. The radius of such cylinder (r_k), as a design variable, specifies dynamic parameters, i.e., mass, moment of inertia and center of mass, of the k^{th} link knowing the link density. The controller generates control signals for each joint based on a control law consisting of a proportional (P_k) and integral (I_k) gains along with gains for feedback velocity ($Kv_{fb,k}$) and acceleration ($Ka_{fb,k}$), and also feedforward velocity ($Kv_{ff,k}$) and acceleration ($Ka_{ff,k}$). Consequently, the design problem deals with $10 \times m$ design variables. In the case of CRS CataLyst-5, due to negligible dynamics of the last two joints, their control gains are not considered; and hence the design problem deals with thirty-eight design variables in total.

In the HCD methodology, design attributes are divided into *must* and *wish* attributes. The availability of the design variables along with the considered *must* attributes are presented as follows:

M1) Design availabilities, i.e., a set of inequalities for the design variables X_j's,

$$X_j^{\min} \le X_j \le X_j^{\max} \qquad (j=1,\ldots,n). \tag{16}$$

M2) Joint restrictions, i.e., a set of inequalities for the k^{th} joint variable at instant t, $\theta_k(t;\boldsymbol{X})$,

$$\theta_k^{\min} \le \theta_k(t;\boldsymbol{X}) \le \theta_k^{\max} \qquad (k=1,\ldots,m). \tag{17}$$

M3) Torque restrictions, i.e., a set of inequalities for the absolute torque of the joint k at instant t, $\tau_k(t;\boldsymbol{X})$,

$$\left|\tau_k(t;\boldsymbol{X})\right| \le \tau_k^{\max} \qquad (k=1,\ldots,m). \tag{18}$$

M4) The restriction on the farthest point of the end-effector reachable workspace, i.e., $R(\boldsymbol{X}) \le R^{\max}$.

The considered *wish* design attributes are:

W1) The end-effector overall position error $E_{tot}(\boldsymbol{X})$. The average of the end-effector position error over the set of N_T pre-defined end-effector trajectories at instant t is

$$E_{av}(t;\boldsymbol{X}) = \frac{1}{N_T}\sum_{r=1}^{N_T}\sqrt{\Delta x_r(t;\boldsymbol{X})^2 + \Delta y_r(t;\boldsymbol{X})^2 + \Delta z_r(t;\boldsymbol{X})^2}\ ; \tag{19}$$

where $(\Delta x_r(t;\boldsymbol{X}), \Delta y_r(t;\boldsymbol{X}), \Delta z_r(t;\boldsymbol{X}))$ are the coordinate errors of the end-effector following the r^{th} pre-defined trajectory at instant t. For the final simulation time t_f, the time average of $E_{av}(t;\boldsymbol{X})$ is considered as the end-effector overall position error, i.e.,

$$E_{tot}(\boldsymbol{X}) = \frac{1}{t_f}\int_0^{t_f} E_{av}(t;\boldsymbol{X})dt\ . \tag{20}$$

W2) The robot manipulability $Man(\boldsymbol{X})$,

$$Man(\boldsymbol{X}) = \frac{1}{t_f}\int_0^{t_f}\left(\frac{1}{N_T}\sum_{r=1}^{N_T} cond(\boldsymbol{J}_0^r(t;\boldsymbol{X}))\right)dt\ ; \tag{21}$$

where $cond(\boldsymbol{J}_0^r(t;\boldsymbol{X}))$ is the condition number of the Jacobian matrix of the serial-link manipulator with respect to the base coordinate frame at time t for the r^{th} pre-defined trajectory. At the singular points this condition number approaches infinity, and its minimum value is one [10].

W3) The structural length index of the manipulator $Q_L(\boldsymbol{X})$,

$$Q_L(\boldsymbol{X}) = \sum_{k=1}^{m}(d_k + l_k)/\sqrt[3]{Vol(\boldsymbol{X})}\ ; \tag{22}$$

where $Vol(\boldsymbol{X})$ is the workspace volume, and it is computed based on a numerical algorithm [45].

W4) The average of the overall required torque at time t on the pre-defined end-effector trajectories $\tau_{tot}(t;\boldsymbol{X})$,

$$\tau_{tot}(t;\boldsymbol{X}) = \frac{1}{N_T}\sum_{r=1}^{N_t}\sum_{k=1}^{m}\left|\tau_k^r(t;\boldsymbol{X})\right|\ ; \tag{23}$$

where $\tau_k^r(t;\boldsymbol{X})$ is the required torque for the joint k at time t in the r^{th} pre-defined end-effector trajectory.

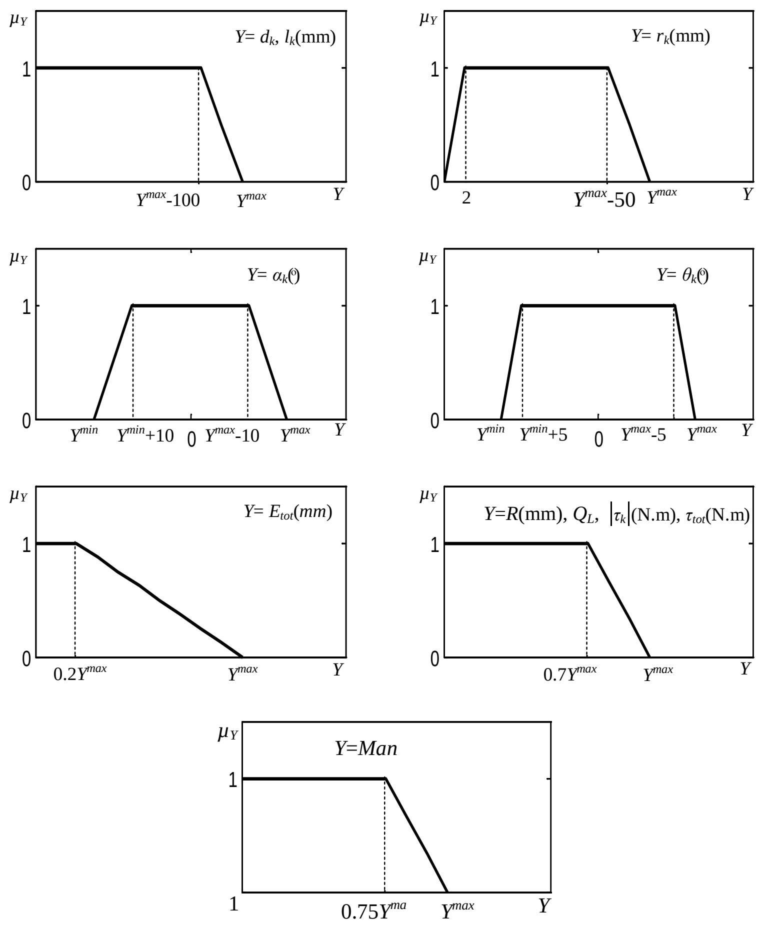


Figure 3. Satisfaction defined on design variables and attributes

4.2.2. Satisfaction assignment

Satisfactions may be considered as fuzzy membership functions over the universes of discourse of design variables and attributes. With the aid of fuzzy set theory, the HCD methodology redefines the notions of inequality (for the design availabilities and *must* attributes) and optimization (for the *wish* attributes), and turns their strict binary nature into a fuzzy one. A form of fuzzy membership functions is the trapezoidal function that is utilized in this case study (see Figure 3). This function is identified by four points that are specified by the designer based on the design availabilities and requirements, and the designer /customer's interpretation of inequality and optimization. The first and last points of a *must* satisfaction are the minimum and maximum values of the corresponding inequality, respectively. The middle points are chosen in a manner that the definition of the inequality is neither too fuzzy nor too crisp. The farther the middle points from the maximum and minimum values, the fuzzier the definition of the inequality is and vise versa. For a *wish* satisfaction, the last point is the maximum allowed value of the attribute (for an attribute that is minimized), and as it decreases the corresponding satisfaction approaches to one. The middle point, which is the designer's defined goal of optimization, is selected based on designer's consensus of the notion of minimum. To specify a maximum allowed value and the goal of optimization, a designer needs to have a good comprehension of the robotic system being designed. All minimum and maximum values of design variables and attributes are listed in Table 1.

4.2.3. Calculation and optimization of the overall satisfaction

To calculate the overall satisfaction, design attributes are determined utilizing the RHILS platform that simulates each design candidate as it follows the predefined trajectories. First, the Denavit-Hartenberg table and dynamic parameters of the design candidate are loaded onto the target workstation to run the inverse dynamics model. The control gains are also specified on the controller. On the host workstation an inverse kinematic code is executed to transform the end-effector trajectories to the joint trajectories. The command signals are sent to the controller from the host workstation using Python®, and simultaneously while the real joints are moving the joints torque are applied by means of the load emulators. The position and torque signals are saved in order to compute all the design attributes, defined in **M1-4** and **W1-4**, on the host workstation. Finally, the corresponding satisfactions are identified and aggregated using a set of attitude parameters. This process of computing the overall satisfaction takes almost one minute, for each design candidate.

In this phase, the HCD searches for the design variables that maximize the overall design satisfaction. A function in the optimization toolbox of MATLAB®, called *fminsearch*, is employed to perform this single-objective maximization. This function uses a derivative-free search algorithm based on the simplex method that is suitable for handling discontinuity, sharp corners and noise in the objective function, which is the case in this problem.

k	1	2	3	4	5
r_k(mm)	[0,200]	[0,200]	[0,200]	[0,200]	[0,200]
l_k(mm)	[0,500]	[0,500]	[0,500]	[0,500]	[0,500]
d_k(mm)	[0,500]	[0,500]	[0,500]	[0,500]	[0,500]
α_k(°)	[-180,180]	[-180,180]	[-180,180]	[-180,180]	[-180,180]
θ_k(°)	[-180,180]	[-110,0]	[-90.6,35]	[-110,110]	[-180,180]
$\lvert\tau_k\rvert$ (N.m)	[0,13.8]	[0,13.8]	[0,13.8]	[0,4.8]	[0,2.4]
R(mm)			[0,870]		
E_{tot}			[0,2]		
Man			[1,24]		
Q_L			[0,1.6]		
τ_{tot}(N.m)			[0,12.5]		
Control Gains			[0.01,1000]		

Table 1. Design variables and attributes ranges

4.2.4. Performance supercriterion

By altering the designer's attitude parameters (*p, q* and *α*) the previous phase of the HCD generates a set of satisfactory design alternatives. To adjust the designer's attitude, physical performance of the system in the form of an objective *supercriterion* is optimized, which is the total energy consumption, in this case study.

$$SE(\boldsymbol{X}_S;p,q,\alpha)=\sum_{r=1}^{N_T}\sum_{k=1}^{m}\int_{\gamma_r}\left|\tau_k^r(\boldsymbol{X}_S;p,q,\alpha)d\theta_k\right|, \tag{24}$$

where γ_r is the r^{th} predefined end-effector trajectory. By minimizing this supercriterion over the satisfactory design alternatives, the best design $\boldsymbol{X}^*$ is achieved.

$$SE(\boldsymbol{X}^*)=\min_{\boldsymbol{X}_S\in C_S}(\ SE(\boldsymbol{X}_S;p,q,\alpha)). \tag{25}$$

4.3. Discussion of results

The CRS CataLyst-5 manipulator was redesigned according to the LM-RHILS based concurrent methodology, and the results are shown in Table 2. With respect to the manipulator dynamic parameters, the mass of link 3 was reduced by 17.5% as a result of decreasing the link radius and length by 10% and 0.7%, respectively. In addition, all other kinematic and dynamic parameters have been modified slightly, which resulted in enhancing the manipulator performance in terms of end-effector overall position error, manipulator reachability, workspace and manipulability, and total energy consumption. For example, the radius of the first and second links has been changed by almost 0.1% and 0.7%, respectively. The length of link 2 and the offset of link 1 have also been altered by 0.1% and 0.4%, respectively. On the other hand, twist angles have remained almost unchanged. Therefore, in terms of dynamic and kinematic design, the third link has been modified considerably.

Since the controller of the existing manipulator was tuned prior to the redesign process, the control gains have made only slight modifications by an average of 0.8%. Even these small changes in the control parameters significantly affected the end-effector overall position error (E_{tot}). The error in the end-effector trajectory after the redesign process is approximately 78 times less than its initial value. An increase in the level of satisfaction for all other *wish* attributes can be observed from Table 2, as well. Therefore, based on the designer's preferences, all the considered attributes have been enhanced. The total *must* satisfaction has improved, which indicates that the new system is far from its performance limits, and hence the new design is more reliable.

The design candidates obtained from the first phase of the HCD were optimized against an objective *supercriterion*, which is the energy criterion, through altering attitude parameters. Ultimately, the configuration with the minimum energy consumption was picked as the final design. The energy consumption was improved by 10%. By looking at the variation of attitude parameters during the design process, one realizes that the initial designer's attitude in aggregating *must* satisfactions was appropriate. That is, the value of *p* did not change through the attitude adjustment. However, in aggregating *wish* satisfactions the designer was originally too conservative. Therefore, *q* was decreased by 50% and *α* was increased by 140%, approximately. This implies that instead of focusing on the worst *wish* attribute, the designer should equally stress all *wish* design attributes in order to improve the system energy consumption. Overall, the results show that the original designers of

	r_k(mm)					l_k(mm)				
	k=1	k=2	k=3	k=4	k=5	k=1	k=2	k=3	k=4	k=5
Initial	65.6	27.7	24.1	10.0	10.0	0.0	254.0	254.0	0.0	0.0
Final	65.7	28.0	21.8	10.0	10.0	0.0	253.6	255.9	0.0	0.0

	d_k(mm)					α_k(°)				
	k=1	k=2	k=3	k=4	k=5	k=1	k=2	k=3	k=4	k=5
Initial	254.0	0.0	0.0	0.0	0.0	-90.0	0.0	0.0	-90.0	0.0
Final	255.0	0.0	0.0	0.0	0.0	-90.8	0.0	0.0	-90.7	0.0

	P_k			I_k			$Kv_{fb,k}$			$[p,q,\alpha]$
	k=1	k=2	k=3	k=1	k=2	k=3	k=1	k=2	k=3	
Initial	18.32	20.00	12.00	0.073	0.050	0.100	40.7	40.0	20.0	[10,1.5,0.5]
Final	18.46	20.16	12.10	0.074	0.050	0.101	41.0	40.3	20.2	[10,0.7,1.2]

	$Ka_{fb,k}$			$Kv_{ff,k}$			$Ka_{ff,k}$			*SE* (J)
	k=1	k=2	k=3	k=1	k=2	k=3	k=1	k=2	k=3	
Initial	43.4	100.0	80.0	59.0	40.0	30.0	3473.0	100.0	120.0	6.2549
Final	43.8	100.8	80.6	59.5	40.3	30.2	3483.6	100.8	120.9	5.6307

	Wish Design Attributes								
	E_{tot}	*Man*	Q_L	$\tau_{tot}(t)$(for the pick and place trajectory)					
				t=0	t=1	t=2	t=3	t=4	t=5
Initial	1.4787	20.7223	1.3091	9.3557	10.2754	9.3561	9.3561	10.2172	10.2172
Final	0.0189	19.4921	1.3025	8.3071	9.1391	8.3071	8.3071	9.1394	8.3071

	Wish Satisfactions									Overall Satisfaction
	$\mu_{E_{tot}}$	μ_M	μ_{Q_L}	$\mu_{\tau_{Tot}(t)}$						
				t=0	t=1	t=2	t=3	t=4	t=5	μ
Initial	0.000	0.606	0.455	0.838	0.593	0.838	0.838	0.609	0.609	0.250
Final	1.000	0.620	0.626	1.000	0.896	1.000	1.000	0.896	1.000	0.607

Table 2. Initial and final design solutions

the manipulator (prior to the redesign process) could have been more aggressive (optimistic) in the design of CRS CataLyst-5.

5. Conclusions

This paper detailed a practical design framework for concurrent synthesis of robot manipulators, which is based on the holistic concurrent design methodology as well as the utilization of a robotic hardware-in-the-loop simulation platform. The RHILS platform makes the system model computations efficient without compromising design transparency, since it uses the physical system components in the simulation loop, which enable the designer to take into account complex phenomena that are difficult to model. Moreover, the HCD methodology alleviates the optimization complexities of concurrent design, since it not only transforms the multi-objective constrained optimization problem into two single-objective unconstrained formulations, but also formalizes subjective notions and brings the linguistic aspects of communication into the design process.

The new framework of concurrent design was used to redesign the kinematic, dynamic, and control parameters of an industrial manipulator, namely CRS CataLyst-5, whose joint modules had been installed in the RHILS platform. Despite the fact that the existing manipulator had been well developed, the new design enhanced the system performance (end-effector overall position error, manipulator manipulability, and total energy consumption) by changing the current manipulator configuration.

For the future work, the authors intend to extend the HCD-RHILS-based concurrent design methodology to reconfigurable robots. For such manipulators the number of degrees of freedom and hence the number of design variables can be changed depending on the task. In addition, a more detailed study of the HCD methodology is planned, including (i) investigating the effect of different types of membership functions on the outcome of the design process, and (ii) the sensitivity analysis of the overall satisfaction with respect to the design variables and attitude parameters.

6. References

[1] Castano, A, Behar, A and Will, P.M (2002) The Conro Modules for Reconfigurable Robots. IEEE/ASME Trans. Mechatronics 7(4): 403-409.

[2] Hewit, J (1996). Mechatronics Design – the Key to Performance Enhancement. Robot. Auton. Sys. 19: 135-142.

[3]Chocron, O (2008) Evolutionary Design of Modular Robotic Arms. Robotica 26: 323-330.

[4] Su, Y.X, Duan, B.Y and Zheng, C.H (2001) Genetic Design of Kinematically Optimal Fine Tuning Stewart Platform for Large Spherical Radio Telescope. Mechatronics 11: 821-835.

[5] Bi, Z.M, Gruver, W.A and Lang, S.Y.T (2004) Analysis and Synthesis of Reconfigurable Robotic Systems. Concurrent Eng-Res. A. 12(2): 145-153.

[6] N. P. Suh, "Axiomatic Design Theory for Systems," Research in Engineering Design, Vol. 10, No. 4, pp. 189–209, 1998.

[7] Kim, J.Y (2006) Task Based Kinematic Design of a Two DOF Manipulator with a Parallelogram Five-bar Link Mechanism. Mechatronics 16: 323-329.

[8] Affi, Z, El-Keribi, B and Romdhane, L (2007) Advanced Mechatronic Design using a Multi-objective Genetic Algorithm optimization of a Motor-driven Four-bar System. Mechatronics 17: 489-500.

[9] Gamage, L.B, de Silva and C.W, Campos, R (2012) Design Evolution of Mechatronic Systems through Modeling, On-line Monitoring, and Evolutionary Optimization. Mechatronics 22: 83-94.

[10] Bi, Z.M and Zhang, W.J (2001) Concurrent Optimal Design of Modular Robotic Configuration. J. Robotic Syst. 18(2): 77-87.

[11] Arakelian, V.H and Smith M.R (2008) Design of Planar 3-DOF 3-RRR Reactionless Parallel Manipulators. Mechatronics 18: 601-606.

[12] Zhu, Z and Dou, R (2009) Optimum Design of 2-DOF Parallel Manipulators with Actuation Redundancy. Mechatronics 19: 761-766.

[13] Li, Y, Ma, P, Qin, C, Gao, X, Wang, J and Zhu, H (2003) Design and Study of a Novel Hyper-redundant Manipulator. Robotica 21: 505-509.

[14] Murata, S, Yoshida, E, Kamimura, A, Kurokawa, H, Tomita, K and Kokaji, S (2002) M-TRAN: Self-reconfigurable Modular Robotic System. IEEE/ASME Trans. Mechatronics 7(4): 431-441.

[15] Liu, G, Liu, Y and Goldenberg, A.A (2011) Design, Analysis, and Control of a Spring-assisted Modular and Reconfigurable Robot. IEEE/ASME Trans. Mechatronics 16(4): 695-706.

[16] Dhingra, A.K and Rao, S.S (1995) A Cooperative Fuzzy Game Theoretic Approach to Multiple Objective Design Optimization. Eur. J. Oper. Res. 83: 547-567.

[17] Wang, J (2001) Ranking Engineering Design Concepts using a Fuzzy Outranking Preference Model. Fuzzy Sets and Systems 119: 161-170.

[18] Gheorghe, R, Bufardi, A and Xirouchakis, P (2004) Construction of a Two-parameter Fuzzy Outranking Relation from Fuzzy Evaluations. Fuzzy Sets and Systems 143: 391-412.

[19] Otto, K.N and Antonsson, E.K (1995) Imprecision in Engineering Design. ASME Journal of Mechanical Design 117(B): 25-32.

[20] Chen, Y.Z and Ngai, E.W.T (2008) A Fuzzy QFD Program Modeling Approach using the Method of Imprecision. International Journal of Production Research 46(24): 6823-6840.

[21] Chen, L.H and Ko, W.C (2009) Fuzzy Approaches to Quality Function Deployment for New Product Design. Fuzzy Sets and Systems 160: 2620-2639.

[22] Wood, K.L, Otto K.N and Antonsson, E.K (1992) Engineering Design Calculations with Fuzzy Parameters. Fuzzy Sets and Systems 52: 1-20.

[23] Scott, M.J and Antonsson, E.K (1998) Aggregation Functions for Engineering Design Trade-offs. Fuzzy Sets and Systems 99: 253-264.

[24] Paredis, C.J.J (1996) An Agent-based Approach to the Design of Rapidly Deployable Fault Tolerant Manipulators. PhD Thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, USA.

[25] Dhingra, A.K, Rao, S.S and Miura, H (1990) Multi-objective Decision Making in a Fuzzy Environment with Application to Helicopter Design. AIAA J. 28(4): 703-710.

[26] Coello, C.A (1999) A Comprehensive Survey of Evolutionary Based Multiobjective Optimization Techniques. Knowl. Inf. Syst. 1(3): 269-308.

[27] Yager, R.R and Filev, D.P (1994) Essentials of Fuzzy Modeling and Control. N.Y: John Wiley and Sons.

[28] Emami, M.R, Türksen, I.B and Goldenberg, A.A (1999) A Unified Parameterized Formulation of Reasoning in Fuzzy Modeling and Control. Fuzzy Set. Syst. 108: 59-81.

[29] Chhabra, R and Emami, M.R (2011) Holistic System Modeling in Mechatronics. Mechatronics 21(1): 166-175.

[30] Breedveld, P.C (2004) Port-based Modeling of Mechatronic Systems, Math. Comput. Simulat. 66: 99-127.

[31] Borutzky, W (2009) Bond Graph Modeling and Simulation of Multidisciplinary Systems – An Introduction, Simulat. Pract. Theory 17(1): 3-21.

[32] Maclay, D (1997) Simulation Gets into the Loop, IEE Review 43(3): 109-112.

[33] Cai, G, Chen, B.M, Lee, T.H and Dong, M (2009) Design and Implementation of a Hardware-in-the-loop Simulation System for Small-scale UAV Helicopters. Mechatronics 19: 1057-1066.

[34] Linjama, M, Virvalo, T, Gustafsson, J, Lintula, J, Aaltonen, V and Kivikoski, M (2000) Hardware-in-the-loop Environment for Servo System Controller Design, Tuning, and Testing. Microprocess. Microsy. 24(1): 13-21.

[35] Stoeppler, G, Menzel, T and Douglas, S (2005) Hardware-in-the-loop Simulation of Machine Tools and Manufacturing Systems. Comput. Control Eng. J. 16(1): 10-15.

[36] Ballard, B.L, Elwell, Jr. R.E, Gettier, R.C, Horan, F.P, Krummenoehl, A.F and Schepleng, D.B (2002) Simulation Approaches for Supporting Tactical System Development. John Hopkins APL Technical Digest 23(2-3): 311-324.

[37] Hu, X (2005) Applying Robot-in-the-loop Simulation to Mobile Robot Systems. 12th International Conference on Advanced Robotics, Atlanta: 506-513.

[38] Cyril, X, Jaar, G and St-Pierre, J (2000) Advanced Space Robotics Simulation for Training and Operations. AIAA Modeling and Simulation Technologies Conference, Denver: 1-6.

[39] Temeltas, H, Gokasan, M, Bogosyan, S and Kilic, A (2002) Hardware in the Loop Simulation of Robot Manipulators through Internet in Mechatronics Education. 28th Annual Conference of the IEEE Industrial Electronics Society, Sevilla: 2617-2622.

[40] Martin, A and Emami, M.R (2008) Design and Simulation of Robot Manipulators using a Modular Hardware-in-the-loop Platform. In: M. Ceccarelli editor. Robot Manipulators: Programming, Design, and Control. Vienna: InTech. pp. 347-372.

[41] Martin, A and Emami, M.R (2011) Dynamic Load Emulation in Hardware-in-the-loop Simulation of Robot Manipulators, IEEE T. Ind. Electron. 58(7): 2980-2987.

[42] Thermo Fisher Scientific Inc., CRS CataLyst-5 Robot System (2007). Available: http://www.thermo.com/com/cda/product/detail/0,1055,21388,00.html.

[43] Li, C.J and Sankar, T.S (1992) Fast Inverse Dynamics Computation in Real-time Robot Control, Mech. Mach. Theory 27(6): 741-750.

[44] Denavit, J and Hartenberg R.S (1955) A Kinematic Notation for Lower-pair Mechanisms Based on Matrices, J. Appl. Mech-T. ASME 22: 215-221.

[45] Ceccarelli, M, Carbone, G and Ottaviano, E (2005) An Optimization Problem Approach for Designing Both Serial And Parallel Manipulators. Proceedings of MUSME, Uberlandia.