Artificial Intelligence Advances
https://ojs.bilpublishing.com/index.php/aia/index

ARTICLE

Development of a Novel Media-independent Communication Theology for Accessing Local & Web-based Data: Case Study with Robotic Sub-systems

Debanik Roy*

Division of Remote Handling and Robotics, Bhabha Atomic Research Centre & Homi Bhabha National Institute, Department of Atomic Energy, Govt. of India, Mumbai 400085, India

ARTICLE INFO

Article history
Received: 26 February 2021
Accepted: 18 March 2021
Published Online: 30 April 2021

Keywords:
Web
Communication
Internet robotics
Information retrieval
Media
Sensory system
Database

ABSTRACT

Realizing media independence in today's communication system remains an *open* problem by and large. Information retrieval, mostly through the Internet, is becoming the most demanding feature in technological progress and this web-based data access should ideally be in user-selective form. While blind-folded access of data through the World Wide Web is quite streamlined, the counter-half of the facet, namely, seamless access of information database pertaining to a specific end-device, e.g. robotic systems, is still in a formative stage. This paradigm of access as well as systematic query-based retrieval of data, related to the physical end-device is very crucial in designing the Internet-based network control of the same in real-time. Moreover, this control of the end-device is directly linked up to the characteristics of three coupled metrics, namely, 'multiple databases', 'multiple servers' and 'multiple inputs' (to each server). This triad, viz. database-input-server (DIS) plays a significant role in overall performance of the system, the background details of which is still very sketchy in global research community. This work addresses the technical issues associated with this theology, with specific reference to formalism of a customized DIS considering real-time delay analysis. The present paper delineates the developmental paradigms of novel multi-input multi-output communication semantics for retrieving web-based information from physical devices, namely, two representative robotic sub-systems in a coherent and homogeneous mode. The developed protocol can be entrusted for use in real-time in a complete user-friendly manner.

1. Introduction

Any physical real-life system is essentially governed by a set of input information, either in clustered form or randomized in time-scale. While clustered or classified information, grouped as data, can be directly fed to the system controller the same practice becomes void in tackling random information set. The same syntax is true for the system output; although, by and large, the real-time outcome of a physical system is standardized. In other words, we may term the system output parameters as patterned outcome, without much of uncertainty

**Corresponding Author:*
Debanik Roy,
Division of Remote Handling and Robotics, Bhabha Atomic Research Centre & Homi Bhabha National Institute, Department of Atomic Energy, Govt. of India, Mumbai 400085, India;
Email: deroy@barc.gov.in

 DOI: https://doi.org/10.30564/aia.v3i1.2920

or randomness involved. Hence, user(s) badly need a streamlined system, through which seamless access as well as query-driven retrieval of information is possible in real-time. We call such a modulation *Multi-Input Multi-Output (MIMO)* system, which is the most generic parlance, used hitherto. As a matter of fact, there can be various forms of such input(s) to any physical device / system, such as text, graphics, audio, short message etc. and likewise different modes of output(s) too such as text, voice[①], fax, e-mail etc. But, in majority of the situations, all of these forms of input-output tuple do follow tailor-made protocol / technology that are mostly patented and irrevocable. Truly speaking, it so happens that due to the rigid and independent technologies that each of these forms of communication use, those unfortunately become incompatible with each other and finally fail to provide a unified or coherent outcome, bearing physical realization. For example, PSTN (Public Switched Telephone Network) uses circuit switching while e-mail, transferred via Internet, uses packet switching; thus, invoking enough incompatibility.

A media-independent communication system is thus becoming a technological boon to the mankind in the global scenario. However, developing the paradigms of media independence coherency in information flow still remains an open research issue. Switching between various input-output formalisms and that too in a user-selective way certainly paves the path towards a successful unification in the emerging domain of data communication. This unification is very helpful to realize a coherent communication system because the user can have the freedom to select any form of input of his/her choice to invoke the system, irrespective of the geographic location and in the same way we can get output anywhere and at any time. It may be stated that the backbone technology of Unified Messaging (UM) system takes care of voicemail, e-mail, SMS (Short Messaging System) & automatic fax, which is used hitherto, fails to address the issues related to on-line delay and to certain extent, noise segregation. These aspects, although not very significant in open-end communication (i.e. where feedback from the end-device is not a must), becomes quite crucial for physical actuating system, e.g. robotic devices. Users can't afford to sustain *in-situ* delay and/or noise-embedded input signal for running a physical gadget. UM-technology is not substantial in tackling these technology-issues and thus, we need to have a specialized MIMO system that can take care of these additional attributes in the best possible extent. Even the standard systematic of Unified Communication (UC), which deals with on-line chat and telephony interfaces, doesn't provide effective way-out.

With this perspective, we can summarize that these aspects, i.e. user-selectable input-output modalities do play a significant up-gradation in recent researches on remote actuation, Internet tele-robotics, tele-surgery and so on. To be specific, the developed system is a novel access path-way for a robotic device using multiple inputs and/or databases. It may be noted at this juncture that a successful seamless real-time operation of an end-device is dependent on the coherence of three matrices, namely, a] database, b] input parameters and c] server. This tuple is thus instrumental in gross control as well as fine-tuning of the operational sequence of the end-device. In a way, the performance of the end-device system is directly dependent on the characteristics of said coupled metrics, namely, 'multiple databases', 'multiple servers' and 'multiple inputs' (to each server). Unfortunately, back-end functioning of this triad, viz. database-input-server (DIS) is still not unearthed to an effective level by the researchers, although it advocates a significant role in overall performance of the system. With this perspective of global research scenario, we will keep our focus on the technical issues associated with this theology, with specific reference to formalism of DIS. And, the present work will address the parlances of DIS in a customized fashion, suitable for the end-use envisaged, i.e. operation of the robotic sensory system. One important thematic of our research is to look into the time delay analysis that affects the performance of the end-gadget to an appreciable extent. The real-time delay and noise have been modeled effectively in the present research.

It is true that potential research was carried out in last decade in harnessing web-based & web-mediated data stream, so that the same can be used effectively during seamless communication through a transport protocol and/or over a wireless network. This broad pathway of data communication was also protected from various research-mandates, namely, routing sequences over the internet, knowledge-driven firmware development or even clustering of multi-modal real-time data. Although these R&D activities have unfurled various untapped open research issues in web-mediated communication, those are not prune enough in answering issues like media independence, real-time MIMO syntax or clustering high dimensional time-specific data in real-time. Based on the large canvas of the present research, literature survey was categorized in ten groups, viz, :i] gathering of web-based information; ii] clustering of documents; iii] high-dimensional & multi-modal data; iv] sub-space clustering; v] MIMO syntax; vi] firmware for knowledge discovery (with data mining & analysis); vii] transport protocol; viii]

① Voice here refers to both voice transmitted on phone as well as voice messages.

DOI: https://doi.org/10.30564/aia.v3i1.2920

wireless network; ix] routing & Internet and x] mobile communication.

Various treatises for web-based information gathering clearly picture out the novel methods that were experimented by researchers. An ensemble of four literature on web-based information retrieval has been found suggestive in our research. The first one, described by Wong & Lam [1], delineated a Bayesian learning framework having novel attributes for automatic information extraction through web using adaptable wrappers. Authors invented a generative model using expectation-maximization technique, which was tested over 30 real-world websites in three different domains. The other facet of web-information extraction is personalized ontology, which is widely used to store personalized (user-specific) information, detailed out by Tao et al [2]. The personalized ontology model, proposed in the paper, has been used for knowledge representation and reasoning over user profiles, after being benchmarked.

Various dimensions of internet topology have been modeled for fast processing of the web-pages, with background ensemble [3] as well as page-independent heuristics [4]. Subsequent to a fruitful way of retrieval of web-information, the impending task for a MIMO layout is to cluster the raw data / document obtained. Thus, survey of the various methodologies that were attempted by the researchers to cluster the web-generated documents obviously plays a crucial role. Cai et al [5] demonstrated a novel clustering algorithm considering local geometry of the document sub-manifold. Authors have used the method of ‘concept factorization’ (a type of matrix factorization) in principle, with customization for localized data-space and proved through experimentation that the developed method (LCCF: Locally Consistent Concept Factorization) is better than traditional matrix factorization methods. Yuan et al [6] reported a detailed methodology for clustering binary sparse data using non-negative matrix factorization (NMF) method. Although methods like LCCF or NMF are good for single-dimensional data stream, clustering high-dimensional data poses a greater challenge because of inherent sparsity of the data-points. Since majority of the existing clustering algorithms become substantially inefficient in high-dimensional space, Bouguessa & Wang [7] proposed a new partitional distance-based projected clustering algorithm, detouring computation of the similarity measure (‘distance’) between data points in the full-dimensional space. The algorithm is effective in detecting projected clusters of low dimensionality embedded in a high- dimensional space.

On completion of the two primary tasks, namely, web-information retrieval and document clustering, MIMO system has to undergo detailing on the raw data of the clustered documents. These data are multi-modal and high-dimensional, in general, and pose a significant challenge in clustering due to inherent sparsity of points. Bouguessa & Wang [8] described a new partition-based clustering algorithm that relies on ‘distance function’ and can detect projected clusters of low dimensionality embedded in a high-dimensional space without computation in full dimensional space. The algorithm has been tested with synthetic and real data-sets. The model of ‘distance function’ has been perfected under various facets of high-dimensionality in order to ensure stability in high-dimensional data space, e.g. Shrinkage-Divergence Proximity distance function [9]. Quite a few of the clustering algorithms, though uses Learning Vector Quantization (LVQ) method, fail to cluster multi-modal data. Hammer et al [10] proposed a new technique for clustering multi-modal data using Generalized Relevance LVQ, incorporating gradient dynamics & global neighbourhood coordination of the prototypes. Jain et al [11] reported a chronological review of data clustering techniques while Pestov [12] delineated on the mathematical insight of the similarity-based search for clustering high-dimensional data. Paradigms of sub-space clustering also play a crucial role in MIMO syntax, besides broad clustering of data in full-dimensional space. In fact, customized sub-space clustering becomes novel in detecting clusters embedded in sunspaces, by calculating relative region densities in subspaces [13]. Kalogeraki & Chen [14] reported a methodology for co-ordinating data-space in equivalent sub-space clusters, based on adjacency related information. Clustered data can possess many incarnations under a MIMO system, the most significant of which is the robust projection method on the output vector. Yu et al [15] developed a robust multi-output regularized feature projection method that retains the features of the input vector and shapes out the internal correlation between various input-output modules. Prediction accuracy for output vector has been found to be enhanced substantially using this feature projection method. Besides, researchers have studied partial order-based feature projection [16] and internet-based syntax [17] for harnessing a MIMO system in real-time.

It is imperative that for the successful run of a MIMO system we must need a customized methodology for analysis of the clustered data in real-time as well as firmware for knowledge discovery thereon. Besides, classical algorithms for data mining need to be tailored for real-time MIMO system so that different data streams from input vectors can be subsumed. Acar & Yener [18] reported a review on novel multi-way data analysis schemes for higher-order data sets, based on the standardized notation and terminology to be used in such multi-way analysis [19]. Cao et al [20] dealt with a novel optimization method for knowledge discovery with decision making attributes, based on the mining of large data sets, often segregated over 'domains' [21]. Two new techniques of knowledge discovery have been reported, namely, under horizontally partitioned databases, using privacy tuple [22] and under casual probabilistic models [23]. Application-level framework for transferring the knowledge pool on clustered data stream in MIMO system has been attempted by research-groups, based on the new metrics of knowledge discovery [24], [25]. Xu & Lipton [26] described another practical aspect of such knowledge-transfer, namely, combating computational complexity in packet scheduling algorithms.

We must have robust transport protocols, at times customized, for an effective practical realization of a MIMO system. In fact, a coherent transport protocol works as a backbone in the overall performance of a MIMO system. Several researchers have studied the parlances of networked transport protocol, mainly in terms of stability & performance, in general [27], [28]. Since a majority of the transport protocols (under MIMO syntax) follow TCP/IP format, its modulation in tackling real-time data has earned due importance in recent past [29], [30], [31]. Roy et al [32] developed a new transport layer protocol offering variable reliability in networked transmission under client-server architecture, suitable for robotic systems. The research has been extended further for improvising the said protocol in order to cater for heterogeneous end-devices, i.e. multi-output context [33].

Besides transport protocol, parlances of wireless network (for sensors) play a significant role in a MIMO system. Yuan et al [34] described a robotized routing scenario, analogous to Traveling Salesman Problem with Neighbourhood (TSPN) that detects & retrieves information from sparsely located sensory network. Analysis was made on the paradigms of rate control as well as dynamic assessment of input rate of elastic traffic in an integrated packet network [35], [36]. It is to be noted also that performance of a MIMO system will be dependent on the router architecture inside the communication network and such a multicast flow under MIMO syntax will be affected primarily by bandwidth allocation [37]. The other facets of router-based communication that may affect the overall realization of a MIMO system are: i] acknowledgement codification [38]; ii] end-to-end packet dynamics & router behaviour [39], [40], [41] and iii] peer-to-peer traffic management [42]. Roy & Chatterjee [43] developed a new protocol using distributed generic architecture for internet-based operation of mechatronic systems in real-time. On the other hand, mobile communication is equally challenging [44], [45], [46], [47], [48] and some of the attributes of the recognized wireless transport protocols have been adapted in the present research too [49], [50].

It is to be noted that despite having a good volume of research publications in the domain of clustering of high-dimensional data, a robust model of data structuring in real-time with time-specific sampling of raw data is still awaited. In order to tackle this lacuna, the present research is focused on building the ensemble of data nomenclature, data structure & data normalization in real-time. This ensemble approach of time-specific intake of raw data, as delineated in the paper, becomes the actual database for any deployable physical system in real-time. Most of the literature is silent about the need as well as effectiveness of normalization of raw data before being used in the MIMO system. We have addressed this issue and brought out a new model for data normalization under multi-input ensemble. It is also observed that a majority of the available literature do not dwell on the practical schematic(s) of end-devices for deployment of the technologies developed. In contrast, our research focuses totally on the augmentation of the MIMO System for manoeuvring of practical engineering systems / devices. Besides, the present research delivers a customized firmware for the real-time implementation of the MIMO architecture.

It may be appreciated that the developed *firmware* is focused for mechatronic / robotic systems, which do have a considerable intra- or inter-locational network, many of those nodes may be remote with reference to *base* node. Besides, the system caters for those applications wherein heterogeneous a combination of sub-systems / devices (e.g. remote robot, sensor, camera etc.) is actuating without the physical presence of a human operator. In the present work, we have selected two representative robotic sub-systems for the implementation of the protocols developed. These are: a] Drive & actuation sub-system (realized through a servomotor assembly, which is an essential element of any robotic/ mechatronic device) and b] Sensory sub-system (realized through a small-sized robotic gripper sensor, which is a useful functional module of any robotic manipulator system). While appreciating the case-studies with these functional components, we may note that the intrinsic nature of these two sub-systems is not exactly same, in response to an external stimulus. The time-varying response of the drive & actuation sub-system against an external input is essentially 'dynamic' in nature; as the motor assembly, driving the robotic joint is responding to those input commands through DIS. In contrary, the response of the sensory sub-system against the same group of external input is 'static' type; as the sensor is only generating output impulse, without any shift in position and/

  DOI: https://doi.org/10.30564/aia.v3i1.2920

or orientation. The paradigm of these two sub-systems is important for the functioning of the MIMO communication system, which will be built up through three DIS. The prime aim of this work is to achieve media independence, where the term media refers to the channel used by the user to access the system like telephone, Internet etc. Media independence here implies that the user can deliver the input and obtain the results in any medium he/she desires. Thus the overall system is result centric rather than media centric. Currently our system (the Beta version) supports the following four media, viz. a] Internet and/or Intranet, b] PSTN and/or EPBX (Electronic Private Branch Exchange), c] Wireless devices, which support WAP (Wireless Activation Protocol) and d] Fax (only output modality, i.e. reception). The unified model of this customized MIMO communication protocol is verified through the laboratory experiments using the servomotor system as well as the gripper sensor unit, mentioned above.

We may now also appreciate the fact that independence of physical location follows media independence. Since the system can run on the Internet, by default it provides for universal access. Correspondingly telephone-based communication can be accessed from anywhere through the PSTN network. Similarly, WAP part can also be activated in a universal manner. Apart from media independence, the other salient goal of the current research-work is to make the solution highly adaptable vis-à-vis customizable to incorporate any physical end-device and/or professional attribute. Although the currently implemented system deals with robotics, the system can be easily changed and customized for any given domain for which the inputs and output can be delineated. The professional attributes (various business rules) are taken care of through independent in-process server (Dynamic Link Libraries: DLLs) that can be replaced to change the system usage as and when situation demands. Also, by virtue of the developed unified non-media centric communication system it is realizable to make disparate legacy message servers appear as a cohesive single module, simply by adding appropriate client and control software. This cohesion adds the benefit of a single point of administration and control for the entire system. Besides, another noteworthy feature of the developed system is the easy transfer of output data to the remote system (e.g. any physical device). This feature allows a user to redirect output to some remote system (but connected to the system's server) without manual intervention at the remote system side. Thus, in a way, this unified messaging system should provide inter-media delivery, access and retrieval, done seamlessly and with transparency. Media/device becomes irrelevant because of the developed architecture of assimilation of communication inputs. In the present research, work has been achieved in operating two representative physical devices that are very important in practical application through the developed MIMO interface.

The paper has been organized in seven sections. The general schemata of the developed system, along with the parametric model of the input/output variables as well as the firmware model is described in the next section. Section 3 delineates the developmental details of the interfaces for the generic input matrices, while the same for output variables in reported in section 4. The details on the MIMO firmware for operating the physical devices have been reported in section 5. Section 6 highlights on the experimentation and case-study results, as obtained from the real-time testing with the robotic sensor. Finally, section 7 concludes the paper.

2. Schemata of the Developed System: Generic Module

2.1 Problem Definition

The '*problem*' which is attempted to be solved in the present work is the real-time activation of physical devices through Multiple-Input-Multiple-Output (MIMO) interface in a media-independent way. In the present research, work has been achieved (through the developed MIMO interface) in operating two representative physical devices that are very important in practical application scenarios pertaining to robotics & automation. The challenge before us was to synchronize different input servers in real-time (with proper data normalization) and to feed those data to the system-server in appropriate syntax. Besides, we needed harmonization of the operating attributes of the devices, so that the device-server accepts those input signals without malfunction.

In the present study, we have demonstrated the said activation for two devices, namely: a] servomotor system & b] robotic sensor system. The ensemble of the developed MIMO interface for the input modules has been nucleated through a 'customized protocol', through which, we have attempted to build generic input matrices, capable of harmonizing three independent servers, namely, web server (either personalized or internet/ intranet-based), telephony server and wireless activation protocol (WAP) server.

The said '*problem*' is being solved by composing a generalized input matrix (GIM), so that user will have access to all these three participating modules of the input manifold. The GIM functions in a pre-conceived protocol, without disturbing the algorithm of the individual

 DOI: https://doi.org/10.30564/aia.v3i1.2920

matrices of web, telephony or WAP servers. In a way, GIM attempts to develop an assimilating media, which integrates several communication media, like *world wide web*, public switch telephony network, fax, WAP services etc. Dynamic Logic Libraries (DLLs) of the respective input-servers have been subsumed in the GIM, so that the ensemble works in a coherent manner, without losing identities of the participating input servers.

In short, a user will be able to send messages via a device (e.g. network connected PC) of his/her choice and access messages via a device of his/her choice too (e.g. a telephone). By this metric, both sender and receiver have total freedom to send / receive messages as per their choice. Thereby the media/device used for communication becomes irrelevant because of the integration. We have successfully tested & implemented the MIMO theology for the transparent & seamless activation of servomotor system (using WAP). For robotic sensor, we have demonstrated the performance of the said MIMO interface for all three types of input servers (web, telephony & WAP).

2.2 System Modeling: Analytical Parenthesis

In a real-time MIMO system, we can expect a steady flow of data against a particular variable, having dissimilar ranges of such data-points. The problem gets complicated when we need to deal with multiple (input) variables at same time-instant and that too, a particular input variable with non-identical spreads of raw data-points. In fact, this is the most perfect and practical situation of data mapping and finally, data fusion, within a fixed domain of time-period of operation of the MIMO system. This multi-faceted system characterization leads to proper structuring & normalization manifolds for the raw data stream, commensurate to the practical end-applications.

The backbone of the system model of this MIMO system is linked with the concept of "Data Centre", which is driven by two parameters, namely, the median value of the data-sets (pertaining to a particular variable) and the spread of the data-sets in the data-space. The detail of the modeling & system architecture of the parlances of data centre will be elucidated in the following sub-sections.

2.3 Philosophy of Data Structure & Data Stream

In the present research, the main focus of the input-output theology is based on the concept of 'data structure', which is modeled in the form of 'data cloud'. In other words, each of the input variables in the MIMO system will be represented as a consortium of time-varying raw data, which will be finally designed in the form of a circular plane, respectively for each input parameter. These circular regions are christened as 'data centre', having a defined centre-point & measurable 'radius'. Figure 1 schematically shows the layout of this 'data cloud' for a single input variable in a two-dimensional plane, concept of which is being used in the present study.

The most significant attribute of the data cloud model, as shown in Figure 1, is the representation of 'time-period' of raw data generation in interval form, e.g. $[t_j]$, $\forall$ j=1,2,….,n, instead of specific time-instant. This is the most practical & application-oriented approach of recording raw data, as in all engineering systems, we need to get a stabilized raw data, after sampling a few over a small time-interval. In fact, time sampling of raw data in case of robotic systems must be carried out in 'intervals', as in majority of the situations, variables from data (source) are real-life application field of robotics and those function seamlessly. For example, in case of robotic sensor, all the input-servers, namely, telephony, WAP & web, responsible for generation of input raw data must be governed through 'time-intervals'. The span of such interval can be decided a-priori, based on the nature of input variable and its manifestation in the said engineering system. We will discuss on the quantification of the time-interval, with reference to our two case-studies later in the paper.

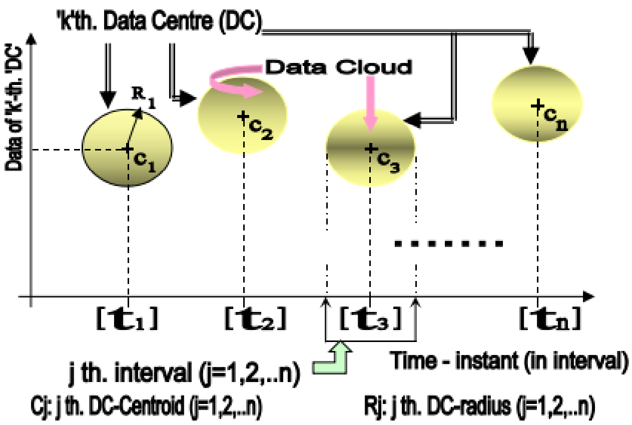

Figure 1. Schematic layout of the data cloud model pertaining to the MIMO robotic system

The next salient feature of the model is about the 'data centres' that are constructed against each input variable and at each time-interval. That means each of the input variables will have separate data centre corresponding to each time-interval, spanning the run-time of the system. The quantum of data against respective variable can be modeled through the data centres (DCs). In other words, DC can have functional facets, such as its radius or scatter (of raw data) which will be instrumental in the evaluation of the measure of the input parameter. Please note that we have shown the data centres only for one input variable in Figure 1, in order to illustrate the concept. It is also prudent

DOI: https://doi.org/10.30564/aia.v3i1.2920

here to note that the maximum number of data centre (DC) per input variable will be equal to the total number of time-intervals; however there can be one or more time-intervals against a particular input variable having null DC (devoid of raw-data). Mathematically, we can write:

$$\Phi\{DC_{I_k}\}_{\max} \leq N\{\left[t_j\right]\} \quad \forall j = 1,2,...,n; \forall k = 1,2,..,p \tag{1}$$

where, $DC_{Ik}\}_{max}$: maximum value of the DC for the $k^{th.}$ input variable & $N\{[t_j]\}$: total number of time-intervals under consideration for the real-time operation of the system. Obviously, the grand total of all DCs running under the system will be proportional to the number of input variables. It is also evident from the model that the existence of null DC is quite likely in situations of staggered operation of the system, especially pertinent to the case of robotic sensor. The other important aspect of DC is its shape, which is important from computational angle. Although it is essentially the convex hull that we are interested in, but, for easy computation we have modeled the DC as 'circular zone', having radius & centre, as detailed in Figure 1. Mathematically, DC_{Ik} is represented as,

$$\Omega\{DC_{I_k}\} \equiv N\{\left[C_j, R_j\right]\} \quad N \in \left[D_j\right]_{[t]}^{R} \quad \forall j = 1,2,...,n; \forall k = 1,2,..,p \tag{2}$$

where $DC_{Ik}\}$ is the measure of DC, which is attributed by the tuple $[C_j, R_j]$, the centroid & radius of the DC respectively and $N\{[C_j, R_j]\}$ is the total number of DCs under study. Also, all such tuple of DC, viz. $[D_j]$ at any time-interval [t], are in real space, 'R'. It is also to be noted, as per expression 2 above, all DCs are considered similar for easy modeling as well as computation. Figure 2 schematically shows the formation of a DC, incorporating all raw data points generated against a particular input variable in the system as well as statistical analysis of the raw data. One DC can have a large agglomeration of raw data points over a specific time-interval, $[t_j]$, which is being geometrically represented by a circle, having radius & centre. It is to be noted that as a perfect measure of central tendency in this case, we will consider median value of the raw data as the final measure of the DC, prior to normalization.

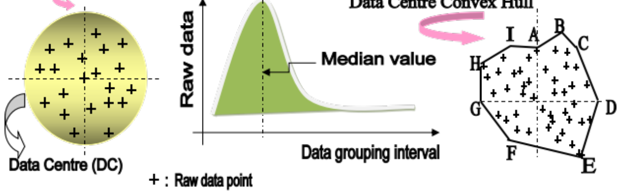

Figure 2. Formation of Data Centre (geometric & convex hull) and analysis of raw data point

As a consequence of the facet shown in Figure 2, we can mathematically extend equation 2 for geometric DC as,

$$\oint[d_j^{(k)}] \equiv \left[C_j, R_j\right]\} \quad N \in \left[D_j\right]_{[t]}^{R} \quad \forall j = 1,2,...,n; \forall k = 1,2,..,p \tag{3}$$

The 'extreme' values of the raw data, as per Figure 2, located on the circumference of the geometric DC, can be modeled with the help of the median value, as detailed below,

$$D_{extreme} \equiv C_k^{Tj} + [d_{\max} - d_{\min}] \times \eta \quad \forall k = 1,2,3,...n; \forall j = 1,2,3,...,m \tag{4}$$

where, $D_{extreme}$: the extreme value of the raw data inside a DC at a specific time-interval against a particular input parameter; C_k^{Tj}: the median value of the raw data inside a DC at a time-interval, 'T_j' against '$k^{th.}$' input parameter; d_{max}: highest value of the raw data under '$k^{th.}$' input at time-interval 'T_j' ; d_{min}: lowest value of the raw data under '$k^{th.}$' input at time-interval 'T_j'; η: shape factor. The choice of 'η' will be the deciding factor for attaining a specific geometry of a DC. However, we will consider uniform 'η' in this work to have identical geometry for all DCs.

We can even think of modeling the DC with varying shape; one example of such is using 'convex hull'. As per Figure 2, the convex hull ABCDEFGHI is the shape of a particular DC wherein we can circumscribe all the physical raw data, generated from a specific input source over the designated 'time-interval'. Nonetheless, the information inside a DC will be further processed by integrating the raw data over several time-intervals for a specific input, irrespective of its shape. Although this lemma of data processing will be used in this work, but, we will also consider the unique case of robotic sensor system case-study, wherein we will evolve a strategy for fusing all input data from all variables at a particular time-interval. In other words, it will be a sort of unified input at a specific time-interval, although this unification doesn't necessarily mean data fusion.

Thus, we will finally have several DCs, against a particular input variable in the data cloud space and the number of such DCs will be increasing once we add up newer inputs in the system. For example, in our case, say 'D_{11}' can be '*telephony server*'; 'D_{12}' can be '*WAP server*'; 'D_{13}' may be the '*Web server*' etc, as stated in section 1. It is to be noted here that we are not including output variables in this syntax for the sake of simplicity,

 DOI: https://doi.org/10.30564/aia.v3i1.2920

but, logically, similar metric can be applied variables (e.g. fax, print, voice etc) too, though restricted. We may note here that there can be multiple D_j at a particular time-interval, $[t_j]$ and these will look like a 'stack' after being compiled for all input variables, 'I_k'. Ideally, all 'I_k's will have DCs at all time-intervals; but, there can be exceptions, which may cause because of the sudden break in real-time operation of the physical system and/or temporary pause. Particularly, in case of robotic systems, intermittent pausing is quite likely and thus we will come across such null DCs. Nonetheless, the final stacking of the DCs will take place as per the flow of raw data in real-time. Pictorially, we can describe the overall schemata, as shown in Figure 3, by extrapolating the layout of Figure 1 & Figure 2. We have delineated a practical situation of real-time operation using four input variables, namely, I_1, I_2, I_3 & I_4 , for which the raw-data have been sampled for time-intervals, ranging between $[t_1]$, $[t_2]$,….,$[t_n]$. The DCs under each input variables have been shown pictorially in a stacked fashion. It is to be noted that we may not have DCs under some time-intervals against some input; for example, there is no DC corresponding to I_2 at $[t_3]$ (refer Figure 3). Likewise, when the system goes for a temporary halt / pause, no DC will be appearing under any input, naturally and that sort of null space of DC is being shown as 'zone of pause' in Figure 3. The order of generation of raw-data and thereby DCs is quite voluminous because of large number of time-intervals under continuous mode of operation of the robotic systems. Obviously, the matter will be computationally intensive when we add more input variables. The decisions that need to be taken under a typical application scenario with large input variables, 'I_j' and huge number of time-intervals, '$[t_p]$', such that order of 'I_j', $O(I_j) >> O([t_p]$ are: a] lemma for selecting a particular input variable at a specific time-interval out of the alternatives of DCs; b] lemma for continuing with a particular DC during zone of pause & c] lemma for constructing the 'projected path', as shown in Figure 3. It is also to be noted here that apart from zone of pause, there can be momentary loss of raw-data stream or break in the communication pathway, because of system malfunction. In such cases, some of the inputs may get affected, either partially during the specific time-interval or may be in full. Nevertheless, even for such 'affected' input stream we will be able to construct DC. Thus, there can be multiple DCs at a particular time-interval, may not be in order though.

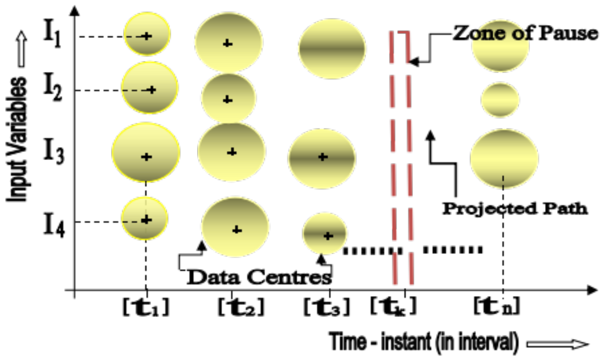

Figure 3. Schemata of the stacking of the data centres in real-time applicable to the MIMO robotic system

Hence, as per the stacking layout of Figure 3, we can formulate the following matrix expression,

$$\begin{bmatrix}[t_1]\\ [t_2]\\ [t_3]\\ [..]\\ [..]\\ [t_k]\\ [..]\\ [..]\\ [..]\\ [t_n]\end{bmatrix} \Rightarrow \begin{bmatrix} \langle DC\rangle_{I1}{}^{t1} & \langle DC\rangle_{I2}{}^{t1} & \langle DC\rangle_{I3}{}^{t1} & \langle DC\rangle_{I4}{}^{t1} \\ \langle DC\rangle_{I1}{}^{t2} & \langle DC\rangle_{I2}{}^{t2} & \langle DC\rangle_{I3}{}^{t2} & \langle DC\rangle_{I4}{}^{t2} \\ \langle DC\rangle_{I1}{}^{t3} & O & \langle DC\rangle_{I3}{}^{t3} & \langle DC\rangle_{I4}{}^{t3} \\ \langle ....\rangle & \langle ....\rangle & \langle ....\rangle & \langle ....\rangle \\ \langle ....\rangle & \langle ....\rangle & \langle ....\rangle & \langle ....\rangle \\ O & O & O & O \\ \langle ....\rangle & \langle ....\rangle & \langle ....\rangle & \langle ....\rangle \\ \langle ....\rangle & \langle ....\rangle & \langle ....\rangle & \langle ....\rangle \\ \langle ....\rangle & \langle ....\rangle & \langle ....\rangle & \langle ....\rangle \\ \langle DC\rangle_{I1}{}^{tn} & \langle DC\rangle_{I2}{}^{tn} & \langle DC\rangle_{I3}{}^{tn} & \langle DC\rangle_{I4}{}^{tn} \end{bmatrix} \quad \Omega\langle DC\rangle \subset \{I_j\} \forall j = 1,2,3,4 \tag{5}$$

where $[t_k]$ is the 'zone of pause' and correspondingly, we have null DCs (represented by the symbol 'O' in expression 3) and Ω<DC> signifies the entire gamut of DC (i.e. the Data Cloud) under the inputs, $\{I_j\}$. It is presumed that the system is operational with 'n' time-intervals, namely, $[t_1]$, $[t_2]$, $[t_3]$,….$[t_k]$,….$[t_n]$ and is being fed by four different inputs, viz. $\{I_1,…I_4\}$. It is also to be noted here that data centres, i.e. $[DC]_{Ij}$ can seamlessly move forward from a particular $[t_k]$ to $[t_{(k+1)}]$, $\forall j=1,2….,p$ & $\forall k=1,2,……n$, and those may end abruptly at any intermediate $[t_k]$. But, the DC stream must be continuous and uninterrupted from initial $[t_1]$ to $[t_k]$. Obviously, no backtracking is allowed or even possible, because data can't go back from $[t_{(k+1)}]$ to $[t_k]$. We may also note that the so-called external boundary of a DC (i.e. circumference of the circle, as shown in Figure 1& 2) may or may not be identical with the width of the specific time-interval, say, $[t_j]$.

As can be visualized from Figure 3 and expression 5, two scenarios of pause need attention during the real-time function of the MIMO system, namely: a] *deliberate* pause & b] *unexplained / catastrophic* pause. The deliberate or pre-planned pause of the system input may occur due to the routine maintenance of the robotic device and during

DOI: https://doi.org/10.30564/aia.v3i1.2920

those time-intervals, i.e. $[t]_{pause\_k}$, $\forall k=1,2,\ldots,p$ there won't be generation of any raw data against any of the input variables. In other words, the instances of deliberate pause will be devoid of any DC. On the other hand, the catastrophic pause is a sort of sudden and/or unexplained 'break' in the system, which may occur due to sudden failure of the sub-assembly or the input source(s). In such cases, we will have 'incomplete DC', although the shape will not be non-geometric. Nonetheless, the continuum of raw data, in the form of input DC, can be mapped through a scheduling mechanism.

It may be noticed here that the modular representation of DCs, as per Equation 5 can be recast in an alternative mathematical way using the concept of '*Chatter Box*'. For example, let us consider the following physical system with the inter-DC relationship as presented below,

$$\begin{bmatrix} & D_1 & D_2 & D_3 & D_4 & .. & .. & D_p & .. & D_m \\ [t_1] & 1 & 1 & 1 & 1 & .. & .. & 1 & .. & 1 \\ [t_2] & 1 & 1 & 0 & 1 & .. & .. & 1 & .. & 0 \\ [t_3] & 0 & 0 & 1 & 1 & .. & .. & 1 & .. & 1 \\ [t_4] & 0 & 0 & 1 & 1 & .. & .. & 0 & .. & 1 \\ [..] & .. & .. & .. & .. & .. & .. & .. & .. & .. \\ [..] & .. & .. & .. & .. & .. & .. & .. & .. & .. \\ [t_k] & 0 & 0 & 0 & 0 & .. & .. & 0 & .. & 0 \\ [..] & .. & .. & .. & .. & .. & .. & .. & .. & .. \\ [t_n] & 0 & 0 & 1 & 1 & .. & .. & 1 & .. & 1 \end{bmatrix}$$

using which we deal with a (m x n) matrix, where m: number of input variables or number of data centres in the system & n: number of time-intervals. The corresponding data centres (DCs) are represented by 'D_j', $\forall j=1,2,3,4,\ldots,p,\ldots.m$, which are active over the span of time-intervals, '$[t_r]$', $\forall r=1,2,3,4,\ldots..k,\ldots n$. It is interesting to note that the data-entry under each cell-location of the matrix can be either '1' or '0', where '1' stands for the occurrence of raw-data under that specific DC and '0' signifies the absence of raw-data under that specific DC, at the particular time-interval. As defined earlier, the matrix-row corresponding to time-interval $[t_k]$ is the 'zone of pause'. The direct corollary of this matrix-based representation is generating an '*Ordered Chatter Box*', by which we can segregate those DCs with cell-entry '1', at a specific $[t_r]$, $\forall r=1,2,3,4,\ldots..k,\ldots n$. These DCs are termed as 'significant DCs', as those will help us in decoding commissioning or any application-specific real-time trouble. The 'ordered matrix' will be formed by arranging the time-intervals in the sequence of having maximum number of '1' against the entries under 'D_j', $\forall j=1,2,3,4,\ldots,p,\ldots.m$. Naturally, the time-interval, signifying the 'zone of pause' will be the last row of this matrix (as it will contain all '0's). As per the proposition, typical nature of this order matrix will have a geometric shape inscribing the '1's, with scattered zone(s), formed by '0's. A representative ordered matrix for a physical real-time system can be mathematically described as shown below,

$$\begin{bmatrix} & D_1 & D_2 & D_3 & D_4 & .. & .. & D_p & .. & D_m \\ [t_1] & 1 & 1 & 1 & 1 & .. & .. & 1 & .. & 1 \\ [t_2] & 1 & 1 & 1 & 1 & .. & .. & 1 & .. & 0 \\ [t_6] & 1 & 1 & 1 & 0 & .. & .. & 0 & .. & 1 \\ [t_4] & 1 & 0 & 1 & 1 & .. & .. & 0 & .. & 1 \\ [..] & 1 & 1 & 0 & 1 & .. & .. & .. & .. & .. \\ [..] & .. & .. & .. & .. & .. & .. & .. & .. & .. \\ [t_p] & 0 & 0 & 1 & 0 & .. & .. & 1 & .. & 0 \\ [..] & .. & .. & .. & .. & .. & .. & .. & .. & .. \\ [t_k] & 0 & 0 & 0 & 0 & .. & .. & 0 & .. & 0 \end{bmatrix}$$

Thus, by deciphering the ordered matrix off-line, realization can be made about the performance of the physical MIMO system, especially about the analysis of the 'zone of pause' or the instances where the system is devoid of raw data. Necessary troubleshooting can be undertaken by checking the repeated occurrence of '0's in the system and thereby sending alert signal to the system operator. In that respect, creation of the ordered matrix becomes truly significant, although the exact design of the matrix can be customized by arranging the time-intervals in a desired manner to suit the geometric shape of the bounded region with '1's or '0's.

It is to be noted that in certain practical applications, we can come across the situation of 'overlapping of two or more DCs'. This situation can occur if two or more inputs are from exactly similar source or inputs are meant for modeling the same physical parameter. In such cases we can have fused DC and the same may be modeled as another convex hull. Although the individual DCs will still maintain their parlances, like registering the raw-data, evaluation of median etc., the fused DC will be finally used for the representation / calculation thereof, as per expression 5. For example, the fused DC for say inputs I_1 & I_2 will be expressed as the union of the respective DCs for inputs I_1 & I_2 at the time-interval $[t_k]$, as shown below,

$$[DC\{I_1, I_2\}]_{[t_k]} \equiv \left[DC_{I_1} \cup DC_{I_2}\right]_{[t_k]} \qquad \forall k = 1,2,..,p,....,n \quad (6)$$

2.4 Data Nomenclature and Normalization

We have seen that the optimal way of nomenclature of the aggregate data is through its median (refer Figure 2), which will be used in this work for practical real-time case studies. As explained in the previous section, the criterion of picking up the median value of every DC is universal throughout all the input variables pertaining to a real-time system. Based on this lemma, we can write mathematically the following expression,

$$[I_j] \in \left[Q_j \subset (C_j, R_j)\right]_{[T_k]} N_j \in \left[D_j\right]_{[t]}^{R} \quad \forall j=1,2,...,n; \forall k=1,2,..,p \tag{7}$$

where $[I_j]$: Generalized input vector, $\forall$j=1,2,…,n; $[D_j]$: Generalized vector of the DCs, $\forall$j=1,2,…,n; N_j: Total number of DCs at any time-interval, [t]; R: Real-time space; $[T_k]$: Generalized time-interval, $\forall$k=1,2,…,p; Q_j: Generalized median vector of the DCs, $\forall$j=1,2,…,n; (C_j, R_j): Generalized tuple of 'centre-radius' of the geometric DCs, $\forall$j=1,2,…,n. Nonetheless, we need to normalize these median-values for further processing, as the sources of these $[Q_j]$ are different in nature & type. Now, two cases may appear at a particular $[T_k]$, namely, case I: when all $[Q_j]$ are summed up and case II: when not all $[Q_j]$ are summed up. However, before summing up, we need normalization of data, by which we can get,

$$[Q_j]^N = \bigcup_j \tilde{Q}_j \in [\tilde{Q}_j \subset (C_j, R_j)] + \zeta \bigcup_j [\tilde{Q}_j \cdot \tilde{Q}_{j+1}]_{[T_k]} \quad \forall j=1,2,...,n; \forall k=1,2,..,p \tag{8}$$

where, $[Q_j]^N$:Matrix of normalized value of the generalized median vector of the DCs $\forall$j=1,2,…,n; $Q_j^{\sim}$: Normalized median vector of the $j^{th.}$ DC, $\forall$j=1,2,…,n; $[T_k]$ & (C_j, R_j): as explained before (refer Equation 6). The model of $[Q_j]^N$ is essentially an union of different normalized median vectors as well as the inter-median effect, based on successive neighborhood principle, attenuated by a scaling factor, 'ζ', where 0≤z≤1. By successive neighborhood measure we mean scalar product of any two consecutive median vectors, say, $j^{th.}$ & $(j+1)^{th}$ $Q_j^{\sim}$. It is to be noted that Equation 8 is generic in nature; as we put forward the model of the normalized median vector matrix as the union of two mathematical expressions. However, for computation, we will use summation rule and thus the recast formula for the matrix of normalized value of the median vectors will be,

$$[Q_j]_{Normalized} = \sum_{j=1}^{j=n} \tilde{Q}_{j\,[T_k]} + \sum_{j=1}^{j=n-1} \tilde{Q}_j \cdot \tilde{Q}_{j+1\,[T_k]} \quad \forall j=1,2,...,n; \forall k=1,2,..,p \tag{9}$$

where, $[Q_j]_{Normalized}$ is the final computational expression for the normalized values of the median vector of the DCs $\forall$j=1,2,…,n. It is to be noted here that in some specific application systems, zonal effect of the data centres is crucial and we need to use subtle numerical model for neighborhood effect in those cases, unlike that shown in Equation 9. The generalized model for analyzing neighborhood effect is proposed as below,

$$\{\Phi_{j:(j+1)}\}_{NE} = [\lambda_j \tilde{Q}_j]^{a_j} \cdot [\lambda_{j+1} \tilde{Q}_{j+1}]^{b_j} \quad \forall j=1,2,...,n \tag{10}$$

where, $\{f_{j:(j+1)}\}_{NE}$: Measure of the neighborhood effect between $j^{th.}$ & $(j+1)^{th.}$ normalized median vectors, $\forall$j=1,2,…,n; $\{l_j, l_{j+1}\}$: partial factor effect, between two consecutive normalized median vectors, viz. $j^{th.}$ & $(j+1)^{th.}$; a_j & b_j: exponential factor effect for the $j^{th.}$ normalized median vector. However, in case all Q_js are not summed up, then we need to make the list showing the occurrence of the specific DCs at a particular $[T_k]$.

We will perform all the calculations related to the unification of normalized data from different servers (inputs), based on the models stated above. However, one important facet, related to the selection of the normalized vector under a DC needs to be addressed before we can proceed for actual implementation of the normalization model in the case-studies. If any two or more (or in exceptional cases all) normalized vector becomes exactly same numerically with one another then the selection of a particular normalized DC and in turn a particular input variable becomes tricky. In such exceptional cases, the process of normalization may be repeated with arithmetic mean value of the raw data under a DC as the 'vector', in lieu of median. As both the metrics are statistical, there won't be any loss of generalization in the process.

We can even compute the overall arithmetic mean of a particular input parameter, varied over the time-intervals, $[t_r]$, $\forall$r=1,2,3,4,…..k,…n. In other words, if for a input parameter corresponding DCs are available with raw-data at respective time-intervals, and then the overall arithmetic mean of the said input parameter can be measured as,

$$[\tilde{M}\{I_j\}] = \frac{\sum_{j=1,k=1}^{j=m,k=n} \chi DC_j \big|[t_k]}{\hat{N}(t_k)} \quad \forall j=1,2,...,n; \forall k=1,2,....,p \tag{11}$$

where, $M^{\sim}_{\{Ij\}}$: the overall arithmetic mean for the $j^{th.}$ input parameter, $\forall$j=1,2,&,n; $cDC_j|[t_k]$: median of the $j^{th.}$ DC at $k^{th.}$ time-interval, $\forall$j=1,2,…,n; $\forall$k=1,2,…,p and $N^{\sim}(t_k)$: total number of time-intervals for which $j^{th.}$ input parameter is generating raw-data.

With these functional paradigms defined, we will study

DOI: https://doi.org/10.30564/aia.v3i1.2920

the representative systems having three severs at present; however provision may be admissible for augmentation of more servers in future. We will now investigate in detail about the implication of these three servers in the real-time systems.

2.5 Basic Engineering Architecture of the Developed System

In the present work, we have selected two representative robotic sub-systems for the implementation of the protocols developed. These are: a] Drive & actuation sub-system (realized through a servomotor assembly, which is an essential element of any robotic/ mechatronic device) and b] Sensory sub-system (realized through a small-sized robotic gripper sensor, which is a useful functional module of any robotic manipulator system).

As stated earlier, the MIMO system is constituted of three servers, each serving request from three different input media types. These input servers are: a] Web server for Internet-based clients, b] Telephony server for clients, hooked up with telephonic communication and c] WAP server for wireless clients. The system theology & overall architecture of the MIMO system is presented schematically in Figure 4. These servers are indexed as 'A', 'B' & 'C' for simplicity in representation and further processing in our modeling & experimentation. Correspondingly, there are three different interfaces, namely, i] the Website as the GUI (Graphical User Interface) for Internet based clients, ii] the TUI (Telephone User Interface) for the telephony based clients and iii] WML pages as the GUI for WAP (Wireless Application Protocol) clients. All three servers use the same database for authentication of users. It may be stated here that the Web server can either be IIS (Internet Information Server) or PWS (Personalized Web Server), as the system protocol is universal that suits both types of web servers equally. It may be mentioned here that the proposed system architecture is media-independent and the functioning of the architecture is entirely driven by the servers & protocols of inter-server communication.

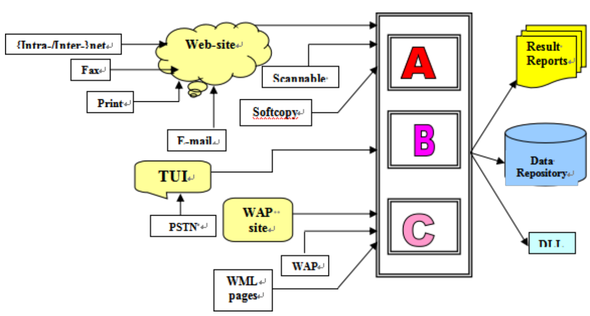

Figure 4. System Theology and Overall Architecture of the Developed MIMO System

Index: A: Web Server (IIS /PWS); B: Telephony Server; C: WAP Server

Now, so far as the engineering layout of the above system architecture is concerned, we have to consider the physical variables (as input to the system) and their manifestation during real-time operation of the MIMO system. For example, in the present two case-studies, we will have to deal with three types of physical input variables, viz. a] force b] slip & c] impulse (macro / micro / touch). The robotic sensory system will have an external forcing function as the prime input, followed by slip forces and in some cases, impulse. While the first two types of physical input variables, namely, 'force' & 'slip' get manifested largely on the one-time quantum value, the 'impulse' can be represented in three different forms, such as, 'macro-type', 'micro-type' & 'touch-type'. Likewise, in case of actuation of the servomotor system, we need some sort of 'impulse' to begin with. Once demarcated, manifestation of these physical input variables (forcing functions) can be realized through all the six possible input-options, mentioned in Figure 4. For example, force function can be realized through: a] web input, i.e. direct from website for actuation; b] telephony input, i.e. by voice function, as a transformation to force magnification; c] internet, i.e. via computer & keyboard entry; d] fax, i.e. direct from scanned document; e] e-mail, i.e. via image function and f] WAP, i.e. through wireless protocol. It is to be noted that all the three servers make use of the central computer system, with access to its filing syntax in order to create and store the output in form of ASCII text files. The three servers also borrow objects from the same class whose methods are nothing but the implemented business rules, i.e. the programming protocol of using these servers. This class is contained in a DLL (Dynamic Link Library) registered on the computer.

As depicted in the layout of Figure 4, current implementation of the system supports three functional input media (web-input: Intra-/Internet, PSTN (or EPBX) & WAP-enabled devices) and three classes of functional output media (results/reports: printer/fax, data repository: telephone/mobile/e-mail & DLLs: Computer-based syntax). By functional input & output, we mean the broad groups / varieties of input & output media which are responsible for the creation of the actual operational input & output, as detailed in the layout of Figure 4. Functional metrics are the backbone of the developed MIMO system, which decide on the parlances of processing of data through the specific input media for the generation / realization of the specific output media. For example, the functional output in the form of 'results/reports' will be realized through printer and/or fax in most of the situations, while the output in the form of 'data repository' will culminate in gathering & recording data in the form of voice (landline and/or mobile telephone) & e-mail. The

 DOI: https://doi.org/10.30564/aia.v3i1.2920

third form of output realization will be manifested through 'Dynamic Link Libraries (DLLs)', which essentially gets routed through several computer-based syntax, programmed or un-programmed.

Accordingly, the functional metrics of input & output, shown in Figure 4, have been recast in an explicit way in order to describe the full operational gamut of input & output parameters, as carried out in the present research. Figure 5 presents the said metrics of the entire media-independent communication system through a block diagram. Nonetheless, the developed system, being generic, can be extended by adding new interfaces to support new media.

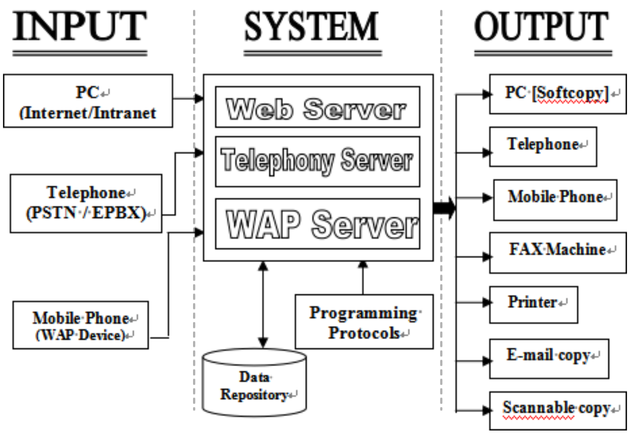

Figure 5. Thematic of the developed media-independent communication system

As depicted in Figure 5, the developed MIMO system can automatically register pertinent data in real-time using three media, viz. [a] PC-based, either intranet or Internet environment and/or through e-mail; [b] landline telephone-based, either PSTN or EPBX set-up and [c] mobile telephone-based, i.e. WAP device. Alternately, the user is, therefore, allowed to input data off-line in any of the three forms, mentioned above. Irrespective of the modality of input, the raw input data get processed through the system black-box, incorporating three different servers, namely, web server (IIS or PWS), telephony server and WAP server. Finally, the MIMO system generates the desired data as output in seven possible ways, as shown in Figure 5, in real-time. Likewise, during off-line operation, the user gets the desired form of output, depending upon his/her choice, viz. through PC (softcopy), landline telephone, mobile (cellular) phone, fax, print-form (hardcopy), scannable copy and e-mail.

The most critical architecture of the developed protocol is augmented with the modeling and analysis of the communication delay, which can be either on-time or off-time. This delay can occur in various modalities, such as: a] between input parameters & corresponding server; b] between server & DLLs and c] between server & database. Nonetheless, delay in this sort of MIMO system can also be grouped in three types, namely, inherent (induced) delay, deliberate delay and fail-safe delay. We will discuss on these modalities in detail in further sub-sections.

2.6 Paradigms of the MIMO System for Augmentation with the Practical Devices

With the establishment of the overall layout and detailed thematic of the proposed MIMO system, we will now investigate the paradigms of the system that are essential for the successful operation of the practical end-devices, namely, a] servomotor system & b] robotic sensor system.

2.6.1 System Characterization for Servomotor System

The actuation of servomotor system is the straightforward application of MIMO pathway, guided by three inputs, namely web server, telephony server & WAP server. The servomotor system needs excitation in terms of change in voltage / current settings for actuation in real-time. The output of the system is manifested through rotary displacement of the motor shaft. The system has been conceived through compatible hardware from National Instruments Inc® (NI) and its physical realization has been achieved by interfacing suitable NI-make data acquisition card through analog instrumentation and mating software (LABVIEW 6.1®). We have concentrated on the harnessing of the actuation of the servomotor system using analog output-based calibration, through the development of the customized application software using LABVIEW®. The schematic diagram of the entire actuation paradigm of the servomotor assembly using MIMO pathway is illustrated in Figure 6.

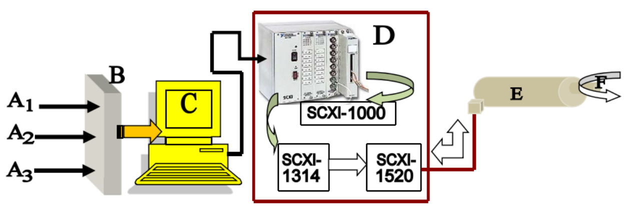

Figure 6. Schematic diagram showing the activation of servomotor system through MIMO pathway

Index: $\{A_1, A_2, A_3\}$: Input data respectively from Web Server, Telephony Server & WAP server; B: Server for unification of the inputs; C: Computer console for processing input data stream; D: NI-hardware including chassis, terminal block & signal conditioning card; E: Servomotor assembly; F: Output shaft of the motor.

The entire programming vis-à-vis hardware handshaking of the MIMO-based function was carried out under the application-specific developmental platform (*NI-DAQ*), having signal conditioning card (Model: SCXI-1520) as input module for processing of the output signal generated out of the computer console, 'C'. This

DOI: https://doi.org/10.30564/aia.v3i1.2920

card (SCXI-1520) is mountable on the Chassis (Model: SCXI-1000), supported by the Terminal Block (Model: SCXI-1314), as shown in Figure 6. Customized program was developed using LABVIEW in order to have the processing of the input data stream and rotation of the motor-shaft, ‘F’, ready at NI-DAQ instrumentation end.

At first, functionality of only one input data stream from one specific server was tested, using its output successively in (analog) switching mode and (digital) Boolean mode. While in SPDT (single pole double throw) switching mode, the output was checked for the pre-specified threshold level in order to declare ON /OFF state, the Boolean mode worked in a different way, using only binary (0,1) signal levels. These two basic trials were vital, as a good handshaking between the servomotor assembly and compatible analog instrumentation (NI-DAQ & LABVIEW) was a must for trouble-free activation of the servomotor in real-time. Afterwards, we moved forward with the next lap of processing wherein all the three input servers were activated. Three separate channels of the SCXI-1520 module were designated for the three outputs, in case we need to categorize the actuation of the servomotor, ‘E’ using individual input data stream. However, in the present case, the rotation of the motor-shaft, F’ will get manifested through only one output data stream, as per the propositions stated in sub-section 2.3, following the hardware protocol as per Figure 6.

2.6.2 System Characterization for Robotic Sensor System

The robotic sensor system, which is being used to demonstrate its functioning through the MIMO pathway is a small-sized gripper sensor, having different tactile sensor-cells embedded in a matrix layout. Figure 7 shows the photographic view of the indigenously developed gripper sensor, with resistive sensor-cells interfaced.

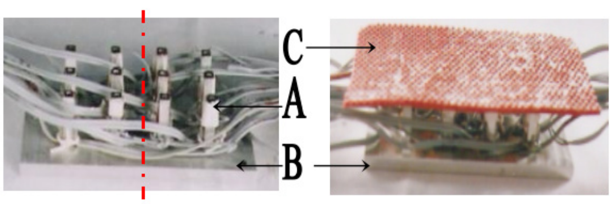

Figure 7. Photographic view of the robotic sensor system used in the MIMO pathway

Index: A: Resistive sensor-cell (with wires intermixed); B: Sensor base; C: Rubber pad.

The robotic sensor needs external excitation and/or forcing to actuate, which can be applied to any of its sensor-cell modules. The sensor is also capable of judging the nature of the external force and thereby gets stimulated to function in a desired way. Finally, it can generate output in either of the two formalisms, viz, ‘force’ and ‘displacement’, under the proper calibration for voltage & temperature. Figure 8 schematically shows the manifestation of this external excitation / forcing function, which finally gets transmitted to the sensor-matrix through strain gauges.

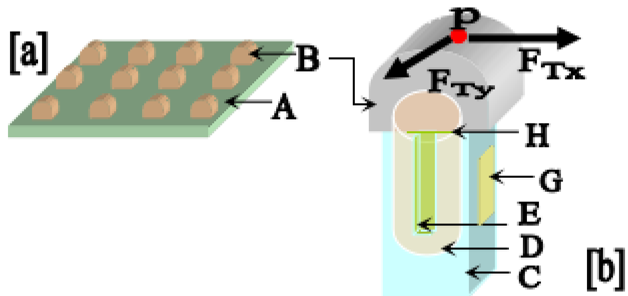

Figure 8. Schematic view of the [a] rubber pad & [b] resistive-cell module of the robotic sensor system

Index: A: Rubber pad; B: Serration(s); C: Strut; D: hole; E: Projecting pin; G: Strain gauge; H: Fixation between pin & pad; $F_{Tx,y}$: Slip force along x & y-axis; p: a generalized point on serration surface.

An analytical model was formulated towards evaluating the slip force, as and when an object is placed atop the slip sensory grid. In a way, the model is used to sense the external excitations on the sensory-grid, often operated remotely with an unknown loading. Based on the raw sensory signals from the R-cells, the model first evaluates tangential force on each of the taxels and then after, total tangential force or the slip force coming upon the grid. However, the transformation of force, e.g. external excitation (on the grid), is significant here and a correct quantitative mapping of forcing-effect is a pre-requisite for the model. Figure 9 presents this routing, wherein the basic tangential force impingement ($F_{T\text{-}Basic}$; stage I) is transmitted to the strut ($F_{T\text{-}Strut}$; stage II), which subsequently gets transformed into *induced* vibration force on the pin ($F^{V}_{T\text{-}Pin}$; stage III). Finally, we get oscillation of the pin inside the strut-hole, being arrested by the strain gauges ($F_{T\text{-}SG}$; stage IV).

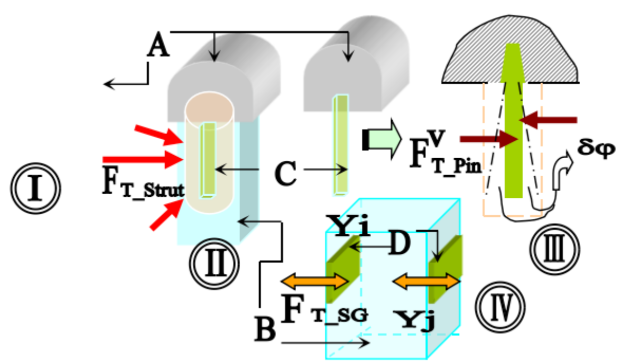

Figure 9. Schematic view of the force transmission route of the robotic sensor system

Index: A: Serration; B: Strut; C: Pin; D: Strain Gauges; $Y_{i,j}$: Readings at the strain gauges δφ: Angle of swing of the pin.

DOI: https://doi.org/10.30564/aia.v3i1.2920

We will now investigate the paradigms of invoking the external excitation / force / focrcing function into the robotic sensory system through any one of the three input manifolds, namely, web server, telephony server & WAP server. It is to be noted that all these inputs, through respective servers, will be able to generate an artificial vibration, manifested through equivalent voltage or current, in the range of mill-volt or milli-ampere. This excitation is a kind of 'synthetic force', that gets generated through the pairof strain gauge, fixed at the opposite walls of the struts (refer part IV of Figure 9). The sensor system needs external stimulus / excitation in terms of change in voltage / current settings for its operation in real-time. The output of the system is realized through on-line recording of 'synthetic force'. Like before, the system has been conceived through suitable NI-make hardware (data acquisition & signal processing cards) for analog instrumentation, effected via LABVIEW 6.1®. Figure 10 illustrates the schematic diagram of the entire actuation paradigm of the sensor assembly using MIMO pathway.

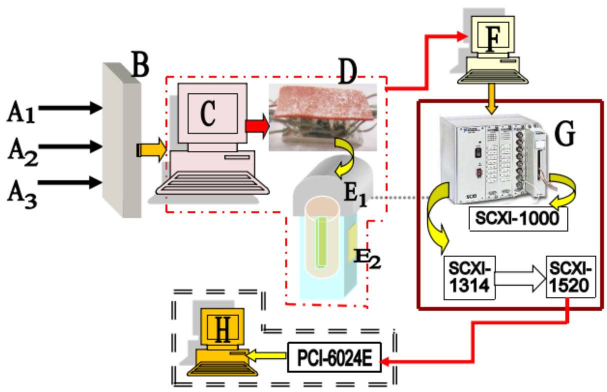

Figure 10. Schematic diagram showing the operation of the robotic sensor system through MIMO pathway

Index: $\{A_1, A_2, A_3\}$: Input data respectively from Web Server, Telephony Server & WAP server; B: Server for unification of the inputs; C: Computer console for processing input data stream; D: Robotic sensor system (as interfaced with 'C'); E_1: Mechanical assembly of the strut (inside the sensor); E_2: Strain gauge (on the surface of the strut); F: Computer system for activation of the sensor; G: NI-hardware including chassis, terminal block & signal conditioning card; H: Host computer responsible for registering synthetic force.

Unlike the case of servomotor assembly, in this case the application programming in totality was carried out using two compatible NI-hardware, viz. [a] universal strain gauge input module for conditioning of the output signal generated out of the sensor, via signal conditioning card (Model: SCXI-1520) and [b] PC add-on type card (at PCI bus) for digital to analog conversion of data (Model: DAQ NI 6024E). The raw data stream from the input servers $\{A_1, A_2, A_3\}$ get unified / fused at 'B' (after due normalization of the data) and then after get fed to 'C' for processing the input data stream. The robotic sensor system, 'D' is connected to 'C', which receives the requisite input vector (maximum of the data of the three input servers) and that gets manifested finally to excitation to the rubber pad ('E_1') & strain gauges ('E_2'). Thus, the sensor gets activated by means of this synthetic forcing, aided by the computer system 'F'. The NI-DAQ hardware, 'G' serves as the as input module for processing of the output signal generated out of 'F'. Inside 'G', the card (SCXI-1520) is mountable on the Chassis (Model: SCXI-1000), supported by the Terminal Block (Model: SCXI-1314), as shown in Figure 10. The processing of the data in real-time, corresponding to the synthetic force, is being done through PCI-6024E that gets registered at the host computer, 'H'.

Customized program was developed using LABVIEW in order to have the processing of the input data stream and generation of synthetic force on the sensor, compatible to NI-DAQ instrumentation end. As before, three separate channels of SCXI-1520 module were earmarked for the three outputs, depending upon the requirement for analyzing the synthetic forcing. At a particular time-instant, 't'. We will consider the maximum numerical value of the input-voltages (coming out of three servers), which will essentially cause the generation of synthetic force / excitation. In other words, the maximum input-voltage value will be used for the computation of the synthetic force, acting over the sensor body at 't'. For example, if 'V_{I1}^t' & 'V_{I2}^t' are the voltages generated out of Input 1 server & Input 2 server respectively at 't', then synthetic force will be computed based on the $\max\{V_{I1}^t, V_{I2}^t\}$. Likewise, with the MIMO system in operation, we will register the maximum of the input-voltages from three servers at 't', which will be designated as the synthetic excitation input, $V_{synthetic}^t$.

In order to verify the correctness of the synthetic force, we used some amount of externally-applied physical excitation to the sensor, over & above $V_{synthetic}^t$ and also the calibrated values from the physical external excitation.

Thus, for a physical external excitation at any time-instant (F_{ext}^t), the sensor controller will generate the corresponding analog signal (say, V_{ext}^t), which will be used infor calibrating the synthetic forcing. Obviously, this calibration will be dependent upon the established calibration curve of the sensor, viz. F_{ext} vs. V_{ext} plot. Hence, at any time-instant, 't', the conjugate output of the sensor system will be, $\{V_{ext}^t + V_{synthetic}^t\}$, which will ideally correspond to a force value (say, $F_{conjugate}^t$), as per the calibration curve. On the other hand, the micro-forcing for the synthetic input, $V_{synthetic}^t$ will be $DF_{synrhetic}^t$, using the calibration curve of the sensor. Now, if the experimental results show the summation $\{F_{ext}^t + F_{synrhetic}^t\}$ is ranging within the nearest

DOI: https://doi.org/10.30564/aia.v3i1.2920

accuracy level to $F_{conjugate}^{t}$, then the validity of the MIMO based implementation can be confirmed. Mathematically, the lemma postulates the following,

$$\lim_{V_{ext} \to 0} \left\{ F^t_{conjugate} - \left(F^t_{ext} + \Delta F^t_{synthetic} \right) \right\} = 0 \quad \forall t = 1,2,...,n \tag{12}$$

The postulation of synthetic force is effective in handing the pseudo-excitations, as generated in real-time through various input servers. The sensor system, interfaced with NI-firmware, is modeled optimally for handing the MIMO inputs, within the ambit of calibrated data, as per Equation 12. It is to be noted that linear regression has been used for the calibration of the sensory system. Accordingly, working equations / relationship between force-function (synthetic as well as direct external) and voltage will be as follows ('m': slope of the calibration curve):

$$\Delta F^t_{synthetic} = mV_{synthetic}{}^t \quad \forall t = 1,2,...,n \tag{13a}$$

$$V^t_{ext} = \frac{1}{m} F_{ext}{}^t \quad \forall t = 1,2,...,n \tag{13b}$$

$$F^t_{conjugate} = m\left(V_{synthetic}{}^t + V^t_{ext} \right) \quad \forall t = 1,2,...,n \tag{13c}$$

3. Development of the MIMO-interface for Input Matrices

3.1 Ensemble of the Developed MIMO Interface

The ensemble of the developed MIMO interface for the input modules has been nucleated through a 'customized protocol' (currently in its Beta version). Through this protocol, we have attempted to build generic input matrices, capable of harmonizing three independent servers, namely, web server (either personalized or internet/ intranet-based), telephony server and wireless activation protocol (WAP) server. However, the generalized input matrix, so framed, will have access to all these three participating modules in the input manifold, without disturbing the algorithm of the individual matrices of web, telephony or WAP servers. In a way, the ensemble essentially attempts in developing an assimilating media, which integrates several communication media, like world wide web, public switch telephony network, fax, WAP services etc. This integrated ensemble has been christened as 'unified messaging system'. The messaging system should provide inter-media message delivery & access and this should be done seamlessly such that the overall functioning of the 'unified messaging system' is transparent to the user. A user should be able to send messages via a device of his/her choice (e.g. network connected PC) and access messages via a device of his/her choice too (e.g. a telephone). By this metric, both sender and receiver have total freedom to send / receive messages as per their choice. Thereby the media/device used for communication becomes irrelevant because of their integration. We will describe the paradigms and characteristics of each of these input modules of the 'unified messaging system' in the following sub-sections.

3.2 System Characteristics and Foundation

As explained earlier, the developed MIMO system is capable of registering 'data' as inputs from three different non-coherent sources, namely, IIS/PWS, PSTN/EPBX & WAP-devices, maintaining the basic ideology of data-stream without bothering about its syntax. In a way, the system is receptive only to the alpha-numeric data, and not to the inner protocol of the very input medium through which the same is being transferred. Besides standard way of data input through IIS/PWS, the e-messaging based input is equally invoked in our system. This form of data entry is analogous to the conventional e-mail service that is provided by many websites. The developed system provides e-message / e-mail based inputs, including all the regular facilities like registration of the new user, username-password for authentication etc. On the other hand, the WAP system enables access to the common message storage through any WAP enabled device, thereby enabling the system for sending and retrieving message to another account defined in our system. The software used to interface the components are: [i] VXML enabled browser and dial-up server (for landline telephone), [ii] Internet browser, server utilities and Internet information services (for PC), [iii] WAP browser and WAP server (for WAP users) and [iv] Fax server (for fax output).

It is to be noted that only pure text version of the data is allowed in the Beta version, because messages having attachments with non-text ensemble (e.g. image-files, video-files) cannot be accessed over the telephone as well as WAP devices. The system is going to process text files, audio files, ASP scripts and binary data stored in the database of the input-side unification server (refer 'B' in Figure 6 & 10). So far as the programming languages are concerned, for network server, ASP & VB Script has been used for server side scripting while JavaScript & HTML have been used for client side scripting.

Likewise, for dial-up server, VXML & ASP have been used for server side scripting. Similarly, for WAP server, WML & ASP have been used for server side scripting. The system has been developed under Windows 2000® *server* platform, along with *Windows OS* for the clients. For WAP module, Nokia simulator is used for requisite testing of the protocol.

3.3 Design of the Interface for Web-based Communication

The multiple input multiple output (MIMO) type website has been developed to serve as the interface for the users accessing the system through the Internet or Intranet. The screenshot of the front portal of the web-page, depicting the 'unified messaging system' is shown in Figure 11 below.

Figure 11. Screenshot of the front portal of the web-page of the unified messaging system under MIMO set-up

The web-based Graphical User Interface (GUI), as shown in Figure 11, provides the following facilities, namely, [a] registration of new users; [b] calculation of design parameters, pertaining to the particular technological context; [c] obtaining the output via fax, SMS and e-mail; [d] delivery of the output to a remote machine; [e] printing of the output at the server and [f] printing a scannable copy of the report. Security is achieved by assigning a unique 'user ID' and a password to each user. The password is an all numeric 'n' length string. The user ID and password combination is always unique. Every user is checked for authenticity using this combination before he/she is allowed to use the system service. The website is built using ASP & VBScript for server side scripting and JavaScript for client side scripting. Since the website uses DLLs for the purpose of calculating the output, it needs to be hosted on a Windows based server. Output is also written to the file system of the server in forms of design specification reports; hence write permission is a must. Figure 12 illustrates the screenshots of the developed web-pages for getting the messages into 'inbox' of the user from various input servers.

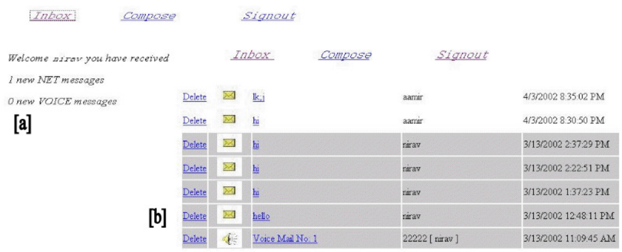

Figure 12. Screenshots of the web-pages: [a] alert & [b] detailed log on receipt of messages from different input servers of the MIMO system

The customized website has the provision for (new) registration vis-à-vis secured authentication of 'users' before the commencement of a technical data-sharing session. Once this phase is through, authenticated users are allowed to make necessary entry for the technical /design parameters for the calculation of the output data. With the help of a chunk of valid and desired data, the analytical modules (e.g. models for processing grip force and incipient slippage for robotic sensor system: refer Figure 7 & 10) do get activated. After successful completion of the analytical phase, the options for 'output screen' are displayed to the user and manifestation of various output modalities is performed on-screen. User can then select his/her output option and output data is then get stored or communicated accordingly. For example, the screenshot in Figure 13 illustrates the option before the user to send the output data / information of different input servers through e-mail & fax under the MIMO pathway.

Figure 13. Screenshots of the web-page for sending output data of different input servers of the MIMO system

The web interface module consists of several backbone files, which do orchestrate the functional logistics of the module. As a matter of fact, the front-end GUIs, as shown in Figure 11-13 are merely indicative of the different functional modules that can be handled by the MIMO website. The main scientific contributions to this developmental research are: i] proposition of novel algorithm; ii] back-end programming and iii] logistic-

  DOI: https://doi.org/10.30564/aia.v3i1.2920

bridging between input & output servers. Based on the operational paradigms, the backbone files have been grouped under the following heads, namely, [a] indexing: this is the homepage of the site, which provides for user sign-in, registration of new user, link to web master's e-mail ID and link to detailing on the site; [b] signing up: this is the sign up form for the new user, which transfers the details of the user to the subsequent forms; [c] updating: this file accepts data from the signing up file and creates a connection with the database and initially checks if a particular user_name exists already. In case the new user ID matches with any of the existing ones, error message is displayed else database is appended with details of the new user; [d] validation: this page checks for the authenticity of the user and displays appropriate message. If the user is authenticated then the number of new web vis-à-vis voice messages, addressed to the very user, is displayed; [e] user domain: this file displays the *user inbox*, wherein the subject of the message can be clicked to read the message; [f] message display(s): these files query the database and retrieve the required text and/or voice message from it, using 'request query string' method. The message is marked as *checked* in the database (in case of textual content), else, message is exported to the server's hard disk in the form of wave file from where it is passed on to the client; [g] compose: this file provides a form to compose and sending a new message. It is also used to send instant fax notification, if required; [h] confirmation: this file creates a connection with the database; else an error message is displayed. If the fax option is also selected then the fax number of the recipient is obtained from the database and a fax will be sent to that number; [i] deletion of message: this file obtains the message ID from user's inbox through Request Query String (RQS) method and deletes the message from the database. It also makes required changes to other fields in the database table to maintain consistency; [j] signing out: this is the sign out form for the user at the end of the web-session.

The developed MIMO system being open-architecture web-based protocol-centric, attaining high security is a priority. To enact upon this, a unique 'user ID' and 'password' is assigned to each registered user. The password is designed as an all numeric 'n length' string. System has internal checking routine in order to ensure that the user ID & password combination remains unique always. Every user is checked for authenticity using this combination before he/she is allowed to use the system service. Figure 14 illustrates the screenshot of the web-page, through which the requisite security-checking metric is getting routed by invoking {*numeric TPIN, numeric password*} tuple.

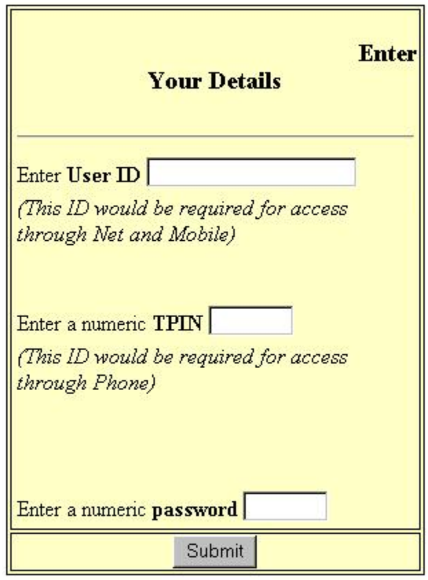

Figure 14. Screenshot showing the security-checking metric for the users under MIMO pathway

4. Development of the MIMO-interface for Output Matrices

4.1 General Syntax

The output matrices for the MIMO interface & protocol therein are governed by the Dynamic Link Libraries (DLLs), which were developed specifically to attribute the quantitative outcome. These DLLs are co-ordinating between the end-device and the input processors under the MIMO format. For example, in case of robotic sensory system, DLLs are developed specifically for transmitting the output data through fax, landline telephone or mobile communication. So far as the PC-based output matrix is concerned, the system is able to generate output data through web-semantics, without using any DLL. Likewise, we don't need any separate DLL for transmission of output data through e-mail or getting a scanned copy of the output data. Thus, the primary syntax of the output matrices are: a] remote connection; b] server polling; c] handshaking with DLLs & d] generation of output GUI. The sequence of events will be strictly in the order as stated above, namely, [a]→[b]→[c]→[d].

4.2 Realization of the Output Ensemble

The two simplest forms of outcome are hardcopy and softcopy of the output, which can be realized through print-out and file storage / saving at the local host. In case the client is logged on to the remote machine, then he/she can do the needful in taking output data by sending request to the server, which will get polled in every equal interval (e.g. 5 secs.). The polling is designed with .asp code, topped up

 DOI: https://doi.org/10.30564/aia.v3i1.2920

by a GUI. For rest of the output ensemble, viz. scannable copy, fax, landline telephone & SMS, usual formats & protocols have been incorporated. In order to evaluate the interim numerical against the process-parameters, a customized DLL has been used, entitled, 'calculation. dll', which is common to all the input modalities wherein mathematical functions are written. Four such functions have been invoked in the realization of slip sensory system that are partitioned in two functional groups. The first group consists of two functions related to the calculation of the forces involved in the grip-slip metric and the second group details out the quantitative characteristics of the slippage using two functions.

The two force-related functions under the first group are: a] *Function:GF:→ Gripping_Force* (to calculate the force of gripping the object by the slip sensor) and b] *Function:TF:→ Tangential_Force* (to calculate the tangential or slip force acting on the object). The mathematical formulation for 'GF' involves evaluation of the minimum gripping force, corresponding to the weight of the object to be gripped (W) and the coefficient of kinetic friction between the object & jaw surface (μ). Hence for stable grasp, the applied gripping force (FG_{app}) must follow the condition: $FG_{app} \geq GF$, i.e. $FG_{app} \geq W/2$, as obtained using the basics of contact mechanics. Likewise, the mathematical formulation for 'TF' involves evaluation of the maximum tangential / slip force, corresponding to the gripping force (GF) and the coefficient of kinetic friction between the object & jaw surface (μ). Analytically, TF_{max}=GF. μ wherein we have incorporated both 'μ' & 'GF' in double precision format in the DLL. This precise numerical evaluation of TF_{max} will ensure arrest to slippage to the best possible extent, as FG_{app} will be greater than $TF_{max.}$ The outcome of this first functional group becomes input to the calculation of the slip-related functions under the second group, responsible for the calculation for incipient slippage variables. The second functional group contains two functions, namely, a] *Function: VS:→ Velocity_Slippage* (to calculate the velocity of incipient slippage at the gripper-object contact interface) and b] *Function: DS:→ Distance_Slippage* (to calculate the amount of displacement of the object that takes place after a slippage has occurred). The function 'VS' is calculated analytically as: VS = (TF_{max} * time_of_slippage) /(material_density * planar_area * separation_distance). All of these variables have been incorporated in double precision format in the DLL, for subtle computation. Here, TF_{max} has been considered as the maximum tangential force coming over the jaw surface. The variables, 'planar_area' and 'separation_distance' denote the cross-sectional area of the object and the distance of separation between the object surface & jaw surface respectively. The distance over which incipient slippage takes place is evaluated analytically through function 'DS' as: DS = (VS * time_of_slippage)/2. As before, both the variables referred here have been incorporated in double precision format in the DLL.

All these four functional modules of the 'calculation.dll', encapsulating the system & process variables have been realized through a user-friendly menu-driven GUI at the front-end. Figure 15 shows the screenshot of a portion of this GUI, wherein evaluation of important parameters pertaining to the four functional modules of the DLL is presented (in the form of an output generated as on-line 'report').

It may be stated here that the realization of output under MIMO framework using the schemata of DLL is essentially generic. Although we have used the two case-studies in the present research, considering access to a single database in each case, the methodology can be adapted to multiple databases too. In other words, similar DLLs can be utilized whenever multiple databases are to be accessed by different users in a pre-conceived notion. All of these databases, say (DB)1,…….(DB)n can retrieve /transmit data in MIMO syntax. The required attributes of the end-devices, namely, servomotor or slip sensor system will be marked by the specific cell of the database, e.g. (ij)th. location of (DB)1 will not the same as (ij)th. location of (DB)2 etc. Nonetheless, all the databases will be interlinked with each other serially and thus the final output from the terminal database will assimilate all the required transactions, made a-priori. Thus the ensemble of output realization as per the current approach is wholesome and makes space for future research and related augmentation of more number of end-devices (using multiple databases) and/or more output-options (using customized DLLs).

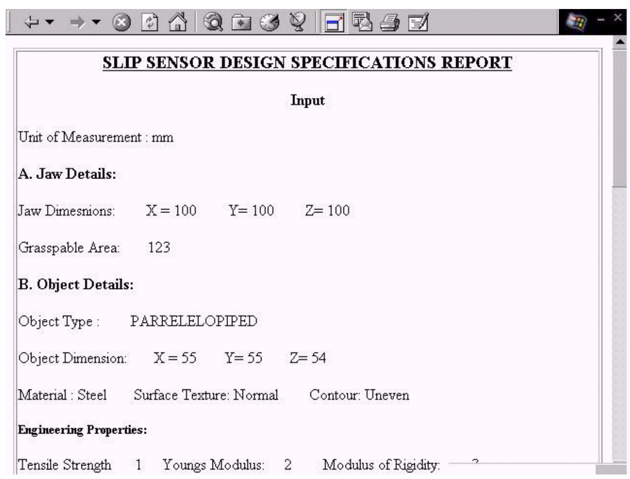

Figure 15. Screenshot showing the GUI of the output of the DLL used in the MIMO architecture

  DOI: https://doi.org/10.30564/aia.v3i1.2920

5. Firmware of Developed MIMO Architecture for Physical System: Case Studies

5.1 Overview of the Firmware and Case Studies

The developed architecture of the MIMO system has been validated through a robust firmware, consisting of hardware (e.g. NI-DAQ systems; refer to sub-section 2.5) and indigenously developed software (i.e. concept, layout & implementation of MIMO protocol; refer to sub-sections 2.4, 3.1, 3.2 & 3.3). The layout of the developed firmware has been designed keeping in mind the end-applications. In other words, the firmware has been customized to a sufficiently large extent in order to accommodate the parlances of the two end-applications of physical systems, viz. [a] servomotor system & [b] robotic sensor system. It is to be noted that out of the three MIMO inputs, i.e. Web, Telephony & WAP-servers, the most critical is WAP because of its functional attributes & programming syntax. Also, module for WAP-server can act as broad ensemble for web-server too, due to similarity in logistics. Accordingly, WAP-based firmware was developed & tested for the servomotor system, while the robotic sensory system was experimented with telephony & web-server based firmware.

A significant portion of the development of system firmware for either of the case studies is based on web-based programming semantics, which will finally get manifested in several user-specific input routines and GUIs. So far as integration of hardware is concerned, both the application systems need nearly identical set-up, because of the similarity of processing of raw data in real-time. By and large, both of the end-applications process the data-cloud in analog mode, which has an easy handshaking with NI-DAQ hardware. Since the end-application hardware has been delineated in detail through Figure 6 & 10, we will focus more on the customized development of software & web-mediated pages in this section. As we described the hardware of the end-application case-studies in sub-section 2.5, we will now report the development of the system-software for both the case studies in sequence, in the following sub-sections.

5.2 Description of the Firmware for Servomotor System

5.2.1 Overview of the Building Modules

The actuation of the servomotor system in real-time has been invoked through its augmentation with a two-fingered robotic gripper assembly, so that we can observe the physical movement of the gripper-jaws (opening & closing) as part of its functioning under MIMO set-up. The firmware developed has got three functional modules, namely: a] parameter extraction; b] user interface & c] communication with the physical device. The first module, i.e. parameter extraction, essentially relies upon the Short Message Service (SMS) of the mobile phone-sets, used in the set-up, piggy-backed by Wireless Transport Protocol (WTP). The second module, i.e. user interface, has been constructed using the broad guidelines of GUI, with provision for alteration of phone settings & log of the actions underwent in streamlining the actuation of the end-device. The third module is the heart of the system, wherein customized protocol has been invoked, with a control over data & time-stamping. All of these modules do have provision for web-based interface that reduces the need of client side hardware and software requirement dependencies. Since the timing of operation of the physical device is important in this development, the statutory parameters like reception, validation & D/A conversion need to be fast enough so that the device responds in quick time. Besides, the firmware is sufficiently equipped with the inherent problems like mobile network jamming and lesser internet bandwidth. The entire programming has been made using Visual Studio 6.0® under Windows® platform. We will now focus on the developmental details of the entire firmware, built coherently with these three modules, stated above.

5.2.2 Developmental Metrics of the Modules

Out of the various input modalities, trials were first made using SMS for driving the servomotor system, which is fast becoming the most dependable as well as secured mode of communication / data transmission in today's world. In fact, SMS has already became the most popular means of communication world-wide due to its low price, reliability, easiness of processing and relatively less vulnerable to hacking (in comparison to e-mail & telephone communication). In the present work, transmission of SMS has been invoked to & from a mobile phone, fax machine and IP address; covering all the three types of input servers under the MIMO pathway. However, we have made trials with messages which are not longer than 160 alpha-numeric characters and are devoid of images or graphics. The real-time development of the overall semantics of using SMS for the drive-control of servomotor system has been attributed to the formulation of a customized ensemble, involving the semantics of 'Wireless Transport Protocol', to be operated through appropriate signal from a wireless device. The entire development has been made under Windows® environment, wherein all inputs were given as analog signal for the remote actuation of the servomotors / gripper assembly. An indigenously developed hardware circuitry was used for disintegration of digital signal in finer form and getting fine-tuned analog output. By virtue of this hardware-level manifestation, we will not only be able to rotate the servomotor but also rotate it efficiently. By invoking the

 DOI: https://doi.org/10.30564/aia.v3i1.2920

parlances of WTP, remote communication can be established between two geographically-distanced entities with physical action of the target-device. The customized version of WTP, as implemented in the present work, provides real mobility and space independence.

In order to activate the physical device, i.e. the servomotor assembly, the user needs to send an activation request, through a wireless cell phone. This activation request/ command will be sent to the servomotor by means of SMS, which will contain parameters to control the speed, direction and on/off states of the servomotor. At the device-end, the said SMS will be received on a mobile station. A local administrator will have a direct control on the GUI used to pass the parameters to the device, with an additional advantage of editing the parameters sent to the device. The entire MIMO-based augmentation of the system has been tried with two types of end-devices, namely, a two-fingered & a three-fingered robotic gripper, fitted with D.C. servomotor assembly that serves as the drive source for the actuation of the gripper in real-time. Figure 16 schematically shows the functional flow diagram of the WTP-based MIMO-pathway for the activation of the end-device (robotic gripper), as indexed (A to D).

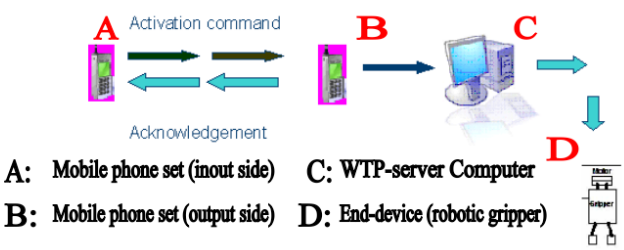

Figure 16. Functional flow diagram of the WTP-based activation of the end-device with servomotor system

Figure 17 presents the photographic view of these two types of robotic grippers, used in the experimentation, as detailed in the functional flow diagram of Figure 16.

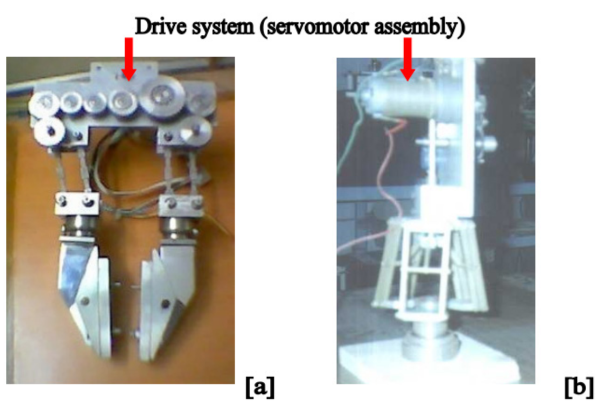

Figure 17. Photographic view of the end-devices used in the test of the WTP-based MIMO-pathway [a] two-fingered robotic gripper [b] three-fingered robotic gripper

Besides dedicated hardware loop (refer Figure 16 & 17), the developed firmware also provides working-level GUI incorporating various features like Profile Management, Action Log, Phone Settings and Help. While 'Profile Management' takes care of the new users, Action Log keeps track of the members who have accessed the system in recent past, including tracking memory of the SIM of the mobile handsets. Through Phone Settings, users can select options for sending the SMS in a specific format, which is mandatory for the MIMO-pathway. The firmware is also equipped with the provision of 'acknowledgement', which will be sent and received at the mobile phone of the user, based on: a] the receipt as well as processing of the input parameters & b] the type of transmission, i.e. successful or unsuccessful. Password checking and dual handshaking facets have been used for authentication of the authorized user. The digital signal from the mobile phone-set needs to be converted to corresponding analog signal in order to actuate the servomotor and in order to achieve this research challenge, following options have been implemented, viz: [a] hardware-level: disintegration of the digital signal in finer form by suitable algorithm and assimilation of fine-tuned analog output, using D/A conversion and [b] software-level: development of PC-based program wherein the digital value from the wireless device will be taken in and then by some model the conversion will take place to analog form. It may be mentioned that in either of the options, algorithm-based filtration technique has been used in order to combat the inherent noise problem to generate required analog voltage, by truncating the final analog signal.

5.2.3 Semantics of Inter-module Data Flow, Interfaces and Operation

The data flow between the modules gets initiated as and when the user sends message or connect with the server using his/her web browser through the internet on a PC or through WAP service on a mobile, as part of controlling the physical device. The user types message in a predefined format set by the system-administrator, wherein he/she can pass the parameters required to control the given physical device. The user sends this message to the receiving end mobile which is connected to the server PC. On the server PC, the parameters are extracted from the message or through the website on internet. At this stage the other administrator also can change the parameters to the physical device using the GUI provided on the server PC if he/she wishes to do so. The parameters received in this digital form get converted to appropriate analog form e.g. voltage levels, rpm etc required by the physical device, i.e. the servomotor. The physical device connected to the server PC receives parameters through suitable electronic circuit and get operated accordingly.

DOI: https://doi.org/10.30564/aia.v3i1.2920

The server PC comes into action when it receives the message either on the receiving mobile connected to it or through internet. At the onset, validation of the received message will be done by crosschecking the user ID & password. Thenafter, the parameters will be extracted from the message and checked if those are in the valid range, permissible for proper operation of physical device. These parameters will then be passed to the next function. Server then completes the necessary part of D/A conversion and transfers the control to device driver for that physical device (in case of custom made electronic device) and finally to the communication port of PC (serial, parallel or USB) to which physical device is connected through appropriate electronic circuit (through voltage & current ratings). The mobile to PC communication at the receiving end is carried through a serial, parallel or USB port cable compatible for that mobile device. The GUI as well as website-based invocation of the MIMO system using WTP is incorporated for localized control of the system. In order to have a jitter-free operation through WTP, the D/A conversion must be within a minimum time-period and the software should maintain the accuracy of controlling parameters after the D/A conversion. The message should be authenticated properly to avoid any damage to physical device. The system administrator can change the control parameters that get passed to the control device by the general user.

The developed system, the software to be specific, has been tested using all existing web browsers that are compliant to the MIMO layout. Necessary interlocks have been provided in the software to check for replay attacks and excessive user-requests. The system also has provision to generate appropriate error messages, to be sent back to the user in case of wrong entry of input and/or abnormal functional metric en-route.

5.2.4 Data Structure and Logical Layout

The first & foremost attribute of the resultant data structure of the developed software is the creation of '*profile*' of the users of the system. As per the syntax, profile consists of six attributes, which are all character array-strings. These user-specific attributes are: i] name; ii] mobile number; iii] occupation; iv] affiliation; v] address & vi] e-mail ID. The purpose of creating profile of the users is to store the complete profile of the user for use in the applications like '*user_authentication*'. Building up of 'profile' is the real-time human-machine interface of the developed system. The data, under 'profile' will remain as a static user profile for all future runs, once user will be authenticated through his/her username & password.

The next paradigm of the data structure is the creation of '*security*', which holds attributes like user name & password, which are all character array. This data structure maintains a database of the user and password of a particular user of the application to be used for the purpose of authentication. The third paradigm of the data structure is the creation of '*service*'. It holds attributes such as username, device-id & services-provided. While the first two of the attributes under 'service' are character array, the last one is Boolean array. The last database, entitled, '*policy*', implements the policy decisions made by the user. It also provides the user to change the policy settings and stores them in the database. This database takes care of the decisions that need to be taken depending upon the policy settings (e.g. control mode of the physical device).

All of these four databases, namely, '*profile*', '*security*', '*service*' & '*policy*' do follow specific file structure in order to orchestrate seamless flow of data in the entire layout. The two most significant files in the data structure are: a] *userprofile.txt*: it will store the entire user profile along with all the attributes mentioned in the profile data structure along with user name & password used to login into the application and b] *logfile.txt* : it will store the log of every event.

The data flow is initiated as and when the user sends message or connect with the server using his/her web browser through internet on a PC or through WAP service on a mobile. The user types message in a predefined format set by the system administrator, highlighting the required parameters for controlling the physical device. On the server PC, the parameters are extracted from the message or through the website on internet. At this stage, the system administrator can also change the parameters to the physical device using the GUI provided on the server PC if he/she wishes so. Thenafter, the parameters received in the digital form have to be converted in the appropriate analog form e.g. voltage levels, rpm etc required by the physical device. The physical device, connected to the server PC through parallel port, receives the parameters through suitable electronic circuit and gets operated accordingly. Likewise, the receiver-end mobile device is connected to the server PC through suitable port. Figure 18 schematically illustrates this logical layout of the system architecture, wherein "*Route*#1" signifies the WAP-enabled pathway through mobile handset and "*Route* #2" pertains to web-driven pathway (through PC).

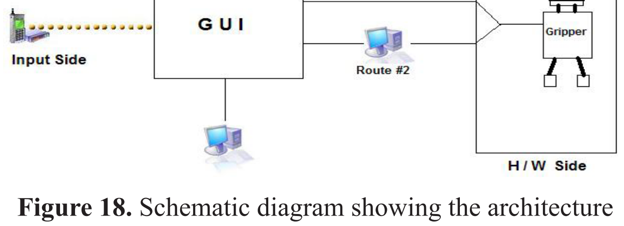

Figure 18. Schematic diagram showing the architecture for the MIMO system for end-device (robotic gripper)

As per the layout of Figure 18, the server PC comes into action when it receives the message on the receiving mobile connected to it. Server PC then completes the necessary part of D/A conversion and transfers the control to device driver of the physical device (in case of custom made electronic device) and subsequently to the communication port of PC to which physical device is connected. Requisite acknowledgement is given to the user on completion of the said transport. It may be noted that Wireless Transport Protocol (WTP), used in this case, is the topmost application layer, which provides user with graphical interface. The said application (customization of WTP) will begin with a 'login' window that needs validation by the user. A valid user can make use of the application for various services by default. This customized WTP-layer interacts with system-level event manager / administrator for carrying out actual communication. It also makes use of the services provided by the lower-rung databases, e.g. '*security*', '*policies*'. In all intra- & inter-module operations, the data stored in the files can be accessed by the file services provided by the operating system.

5.2.5 Architectural Design

The entire development of WTP-based new protocol is based on various important semantic design-blocks, as explained in the earlier sub-sections. Out of the ensemble, three fundamental design-blocks are detailed out here, which contribute to the architectural design of the protocol. These design-blocks are: a] *user authentication*; b] *parameter transfer* & c] *profile editing.* All of these design-blocks spread out between three semantic-zones, namely, user (U), wireless application (WTL) & profiles (P). Figure 19 schematically illustrates the design architecture for the first block, viz. 'user authentication'. At the start of the application, the user will be prompted for username and password, which will be verified from *userprofile.txt*. If the password is found correct, the user will be able to access the services provided by the application. In case of an incorrect password, a message will be given indicating the invalid password.

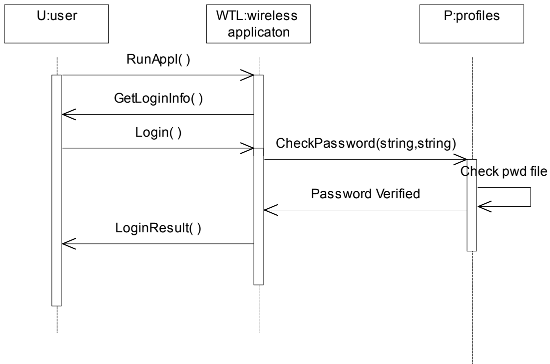

Figure 19. Schematic layout of the architecture of 'user authentication' design-block of the developed protocol

As part of the second important design-block, namely, 'parameter transfer', authenticated user needs to send SMS in a definite format as and when he/she wishes to transfer parameters. The parameters are to be extracted and checked for valid limits. If the parameters are within specified limit, this routine will pass the parameters to the physical device though parallel port; else, error messages are flagged using GUI if the parameters are found invalid. Figure 20 schematically illustrates the design architecture for this block.

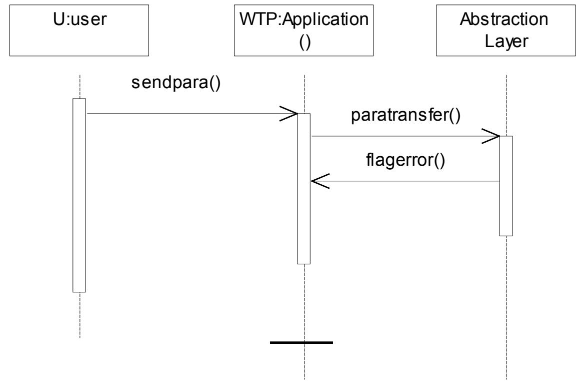

Figure 20. Schematic layout of the architecture of 'parameter transfer' design-block of the developed protocol

The third salient design-block, namely, 'profile editing', provides the user with the functionality to modify his/her profile, which can be shared amongst peer group. Upon submission of the edited information, the profile, stored in the file _userprofile.txt, will be updated. Figure 21 schematically illustrates the design architecture for this block.

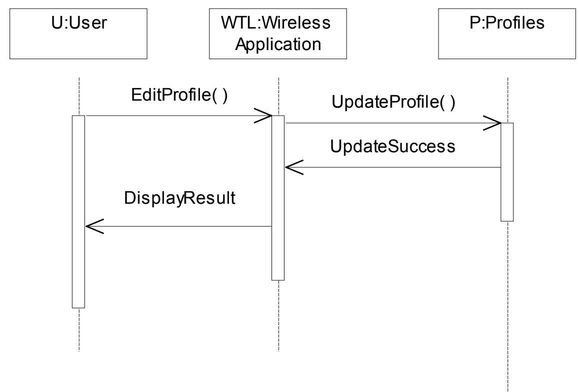

Figure 21. Schematic layout of the architecture of 'profile editing' design-block of the developed protocol

5.3 Description of the Firmware for Robotic Sensory System

The robotic sensory system and its functioning, as illustrated in Figure 7-10, amply describe the purview of the

  DOI: https://doi.org/10.30564/aia.v3i1.2920

hardware in accomplishing multiple tasks, depending upon the specific end-application. One such application is using it as a slip sensor, wherein slip force & slip velocity will be evaluated through a mathematical model, knowing the value of the incipient force & design parameters of the sensor. As stated earlier, robotic sensory system, in the form of a customized slip sensor, has been developed with telephony & web-server based input. The details of the developed firmware are reported in the following sub-sections.

5.3.1 Developmental Metrics for Web-based Input

A customized website has been developed for invoking the functioning of the slip sensory system, with easy user interfacing ensembles for web-based inputs. The functional schema of this website was designed through several sub-routines & modules, coded in ASP. Suitable web-programming syntaxes, written in HTML, were added as coatings to the basic .asp codes. This sort of double-tier programming syntax has made the overall realization of the code very reliable. Figure 22 presents the schematic block diagram of the programming sequences of the customized website, used as essential firmware for the robotic sensory system.

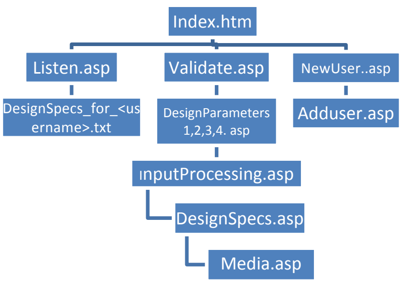

Figure 22. Schematic block diagram of the programming syntax of the website for the robotic sensory system

As per the functional tree structure of Figure 22, the matrix program-code (*index.htm*) houses three modular codes (*Listen.asp, Validate,asp & NewUser.asp*) that create the basic ensemble of the web-server based input functioning of the slip sensory system in real-time. While the first two modular codes, viz. *Listen.asp & Validate.asp* essentially dwell on the engineering parameters related to the slip sensor design (user input & validation against standard values), and the third module is based on logistics, e.g. addition of user. Finally, all the input data are duly processed for the evaluation of output parameters, namely, a] the slip-tuple: comprising slip velocity & slipped distance and b] synthetic force (refer Equation 12). The program-code, *DesignSpecs.asp* consolidates all the calculated output parameters in real-time. The corresponding GUIs are getting triggering through the program-code, *Media.asp*. As appeared in the syntax, this processing ensemble is a feed-forward type of invocation, which is supported by the user-interactive GUIs, discussed below.

The maiden GUI, involving the user ID & password invocation for the telephony-based MIMO module has been developed in *Visual Basic* with usual features. This GUI-page can be made functionalized even through local Personal Web Server (PWS). Figure 23 shows the screenshot of the developed GUI.

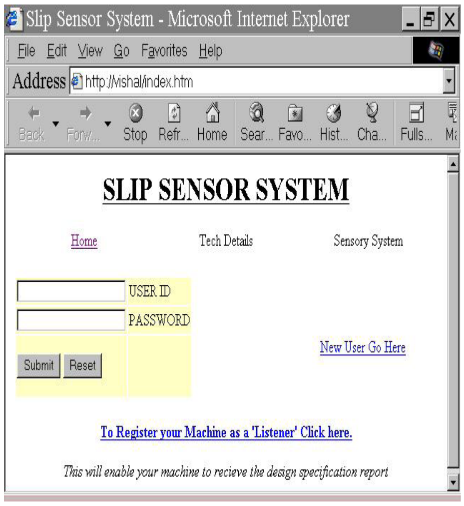

Figure 23. Screenshot of the GUI for the slip sensory system under MIMO pathway for web-based input

The primary backbone of the firmware for the slip sensory system with web-enabled input is the cluster GUI, which provides easy interface between the user & the physical device in real-time. GUIs being the best pathway for web-based input under any MIMO system, we have adopted the ensemble of customized user-interactive GUIs as part of the development of the firmware of the slip sensor. One of the basic requirements for functioning of the slip sensor is to have an apt design, so that minute forces / impulse on the gripper can be picked up by the sensor for further processing. Thus, we have duly considered the first set of web-based input for the MIMO system of slip sensor as the design data, e.g. dimension of the gripper-jaws, maximum allowable displacements in X & Y axes, material properties etc. All of these design inputs are fed to the system computer through a customized web-portal, having suitable GUIs. The first page of the website, so built, shows the GUI for the entry of basic design parameters of the slip sensor, namely: a] dimension of the gripper-jaw (x,y,z) ; b] planar area of the gripper-jaw; c] graspable area of the jaw; d] material of the gripper; e] coefficient of friction (static & dynamic) & f] type of contact between jaw surface & object. Figure 24a,b show the screenshot of the GUI, highlighting the above-mentioned parameters, with some

DOI: https://doi.org/10.30564/aia.v3i1.2920

representative numerical values. The second page of the developed website contains the GUI related to features & quantitative data on the object being grasped, such as: a] type; b] dimensions; c] texture; d] material; e] contour & f] type of contact (between jaw surface & object surface). Figure 25a shows the screenshot of the GUI, illustrating these features, along with the selection of the unit of measurement. On selection of all pertinent parameters related to the design of the slip sensor through web-input, user needs to select the type of slippage that may occur at the object-jaw contact plane. We have envisaged three types of slippage, namely: a] translational; b] rotational & c] twisting, as depicted in the screenshot of Figure 25b. The web server is equipped with tackling real-time information flow from the user(s) in connection with feeding on the design parameters, stated above. All of the GUIs are made menu-driven and user-friendly, so as to assist a new user from a different domain knowledge.

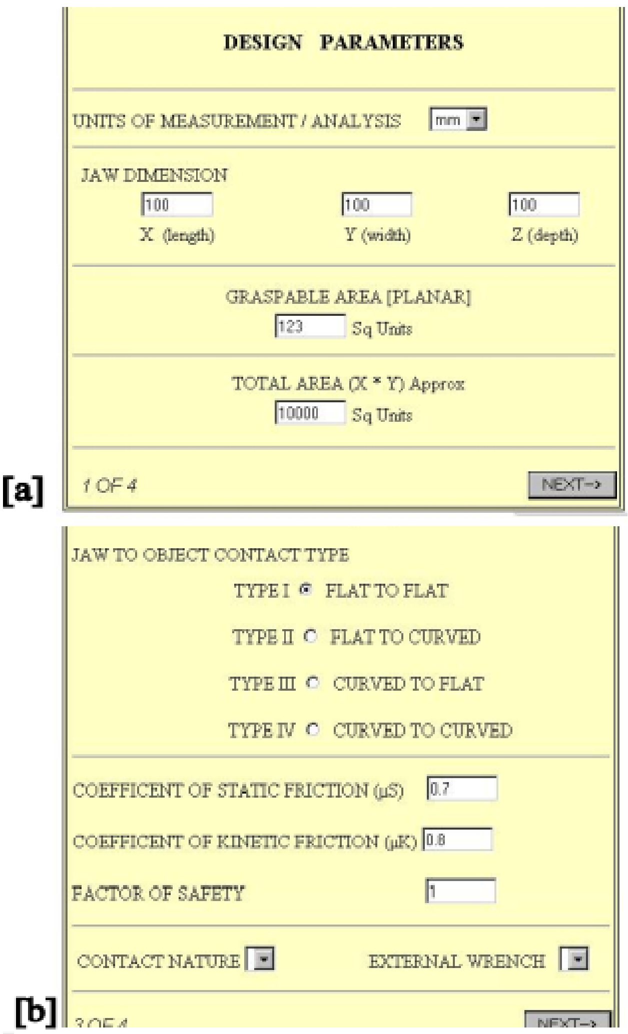

Figure 24. Screenshots showing the GUIs for web-based input for the slip sensory system under MIMO pathway

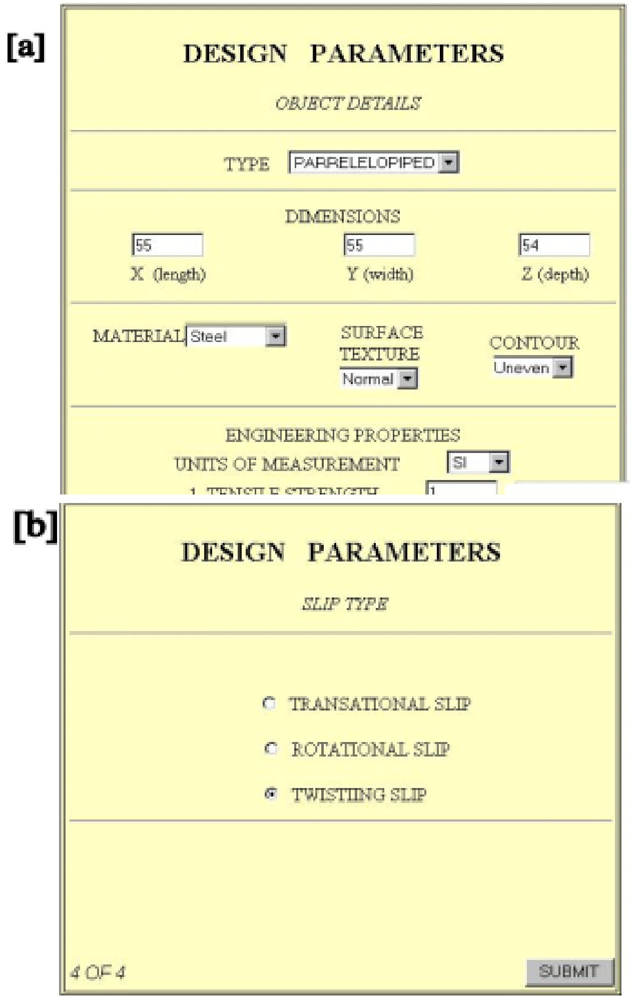

Figure 25. Screenshots of the GUIs for web-based input for the slip sensory system (object & slip semantics)

Since the module is using DLLs for the calculation of slip-tuple & synthetic force, the web-pages necessarily be hosted on a *Windows* based server. Since the output is also written to the file system of the server in forms of design specification reports, write permission is a pre-requisite.

5.3.2 Developmental Metrics for Telephony Interface (Landline)

In line with the development of the customized website for web-based input, dedicated firmware has been designed for the input stream of data through landline telephone system. Like GUI, the Telephony User Interface (TUI) is a stand-alone service, wherein the communication is done by means of dialogues (prompts) over the telephone line and DTMF tones entered by the user. The telephony part provides for simple feeding in of design parameters and the interface generates results through voice message over the same line as output. Like

DOI: https://doi.org/10.30564/aia.v3i1.2920

the website, here the user has to key in his/her ID (in this case known as TPIN) and the password, too. However, the TPIN, unlike the user ID, is all numeric. Conforming to the *Windows* platform, Windows TAPI (Telephone Application Programmer Interface) has been used in the development of the firmware of TUI. For generation of speech the *Lernout and Hauspie Text to Speech Engine* has been used.

The customized telephony server is activated as and when the telephone rings. As per the design of the firmware, a subscriber can dial into TUI from any landline telephone and he/she will be able to listen to and respond to any message waiting in their *inbox*. They will be able to access and manage all three message types i.e. voice, fax and e-message with just one phone call and will be able to respond with a voice message. The system also supports voice recognition; hence the user has the choice of giving input via speech or DTMF. The mode is then switched from communication mode to data mode and IVR (interactive voice response) becomes activated. As soon as user exercises his/her choice for input-entry, i.e. either voice or DTMF, the IP address is detected by VXML. The system will handle multiple tasks simultaneously, like: a] accessing messages from database; b] converting the messages from text to speech and c] reading the message to the user. Besides routing through VXML, the user can also send voice messages and send instant fax notification. The module is structured through the following functional facets, viz. [a] *start up*: this is the starting document (VXML file) of the PSTN interface of our application; [b] *indexing*: this page receives details viz. user ID and password from the start-up file. It checks for the authenticity of the user and displays appropriate message. If the user is authenticated then the number of the new text and/or voice messages is read; [c] *menu set-up*: this document reads out the menu of the application. It contains various choices available to the user like, checking text and/or voice messages, sending voice message and finally exiting the system once done; [d] *deciphering*: this document reads out the text messages stored in the *inbox* of the user. It offers the user various choices such as, receiving fax message, reading and deleting messages stored in the inbox. Characteristic features of the developed TUI, as described, have been coded in *Visual Basic*. The metrics of the telephony-server that involves calculations for evaluating slip related parameters & synthetic force have been developed through suitable DLLs. Figure 26 presents the partial screenshot of the TUI developed, highlighting the 'status' zone in the right-hand side. This zone is dynamically used for prompting the messages and/or system-actuation commands.

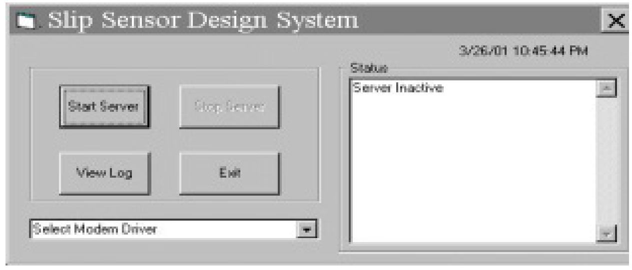

Figure 26. Screenshot showing the developed TUI for a the robotic sensory system in a MIMO pathway

5.3.2 Development of Interface for WAP Module

The general syntax of the development of WAP interface follows that of web-based interface, with some changes in the programme ladder. Like before, a customized web-page has been developed for activating the slip sensory system, through wireless activation. The overall development was achieved through a group of GUIs, backed up by program-codes, scripted in ASP. The matrix file, namely, '*index*' has been written as WML file, by default, because of WAP-module. Likewise, the wrapping-up file, namely, '*wapcompose*' has been coded in WML. All of the program-codes and allied firmware follow *Windows* platform for synergy in a two-tier programming ensemble. Figure 27 presents the schematic block diagram of the programming sequences of WAP-enabled interface for the MIMO pathway, earmarked as the essential firmware for the robotic sensory system.

As illustrated in Figure 27, the *index.wml* file invokes the actuation of the web-based WAP interface by displaying a 'welcome' message, followed by incorporation of user-specific data, i.e. user ID & password. This invocation of user-information gets processed through '*Listen.asp*' code, whereas user ID & password gets validated by '*wapinbox.asp*' code. Standard Windows based technology was used to send variables from *index.wml* to .asp codes. With the help of this protocol, user can see the new messages that have been accumulated in his/her 'inbox'. It also allows user to compose new messages in this WAP-enabled environment. The processing of the input design parameters for the slip sensory system through .txt file and .asp code was done for the output parameters, namely, slip-tuple & synthetic force, as described under sub-section 5.3.1. The program-code, *InputProcessing.asp* consolidates all the calculated output parameters in real-time and signals the completion of the .asp codes in the program-ensemble. Finally, consolidated output of .asp code gets re-christened under WML environment through '*wapcompose.wml*', after invoking various ancillary data like subject & main text of the overall WAP-input. In order to reduce the burden of accumulating WAP-messages &

data therein, an additional .asp code was scripted (*Deletemsg.asp*), specifically for deleting unwanted data from the WAP-interface. As per the developmental syntax, functioning of '*Deletemsg.asp*' depends on transfusion of various external elements in course of establishing the WAP-layout, including amplitude of transmission noise & its frequency.

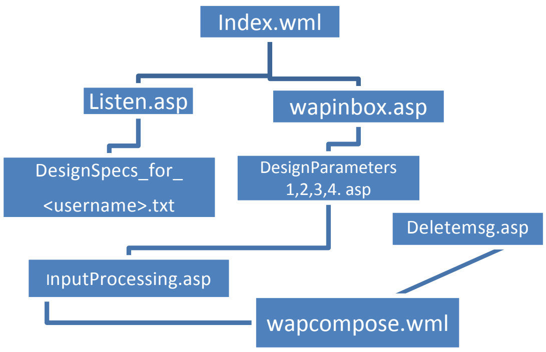

Figure 27. Block diagram of the programming syntax of the WAP-interface for the robotic sensory system

5.4 Operational Guide, Unique Features and Novelties of the Developed MIMO Protocol

We have delineated the detailed program syntax and allied hardware interfacing interlocks of the developed MIMO protocol in the previous sub-sections, pertaining to two real-life physical systems. Nonetheless, following step-by-step procedure should be adhered to in pursuing the developed methodology, which may serve as an operational guide of the developed MIMO protocol.

Step1: System identification: number & type of input servers involved & output manifestation

Step2: End-device identification: type & control-features of the physical device

Step3: Data nomenclature & data normalization

Step4: Algorithm of the new MIMO protocol: syntax & development

Step5: Graphical User Interface: design & development

Step6: Sub-system level validation: through DLL syntax

Step7: Loop testing

Step8: Activation of physical device in real-time & testing

It is to be noted that successful seamless real-time operation of an end-device (gross control as well as fine-tuning) is dependent on the coherence of three matrices, namely, a] database, b] input parameters and c] server. Unfortunately, back-end functioning of this triad, viz. database-input-server (DIS) is still not unearthed to an effective level by the researchers, although it advocates a significant role in overall performance of the system. We have duly addressed the parlances of DIS in a customized fashion, suitable for the end-uses envisaged.

The new MIMO protocol possesses some unique features so far as developmental paradigms are concerned and it is novel in several aspects. On the other hand, the protocol has overcome a number of technological challenges too. We are elucidating on all these facets below.

The present work is one of the maiden attempts to synergize time-variant raw data from heterogeneous input servers that are getting coagulated in actuating a physical device. This fusion of data from various input streams need satisfactory command over the programming logic, hardware syntaxes (through DLLs) and prior knowledge of the system controller of the end-device. Besides, the paper has clearly shown that this unique synergic modulation has stages of evolution, for example, from actuating LEDs to servomotor & robotic slip sensor system. Besides, the work is novel in classifying various output modalities that a physical device can use in real-time for generating results / commands / operating signals. We have demonstrated the working principles of this output syntax, e,g, in the form of print, voice, fax, SMS etc, which a physical device can select through users and exercise thereafter. One important thematic of our research is to look into the time delay analysis that affects the performance of the end-gadget to an appreciable extent. The real-time delay and noise have been modeled effectively in the present research.

The work also finds its novelty in proposing model for synchronization & normalization of raw input data that are being generated from the input servers in real-time. Although there are past-researches that deal with data fusion with commonality, but hardly any work has been reported for such fusion in time-variant manner. This particular aspect is very important in cases of actuation of a physical device, robotic system in particular.

As explained in earlier sections, design and development of a media-independent uniform MIMO protocol is the most significant feature of the present work. Further, real-life testing of the developed protocol with two physical devices is also a worthy feature, which might not have been attempted by many researchers. These two aspects make the present work technologically unique.

It is needless to state that the said novelties as well as unique features have been achieved after overcoming various technological bottlenecks / challenges, such as: a] evolution of the 'Generalized Input Matrix' (GIM) out of three different input-servers (web, telephony & wireless activation protocol); b] creation of data structure for all input-streams; c] systemization of the DLLs of the input-servers & d] generation of a common format for backend scripting that will be acceptable for all the three types of input-servers & outputs. Out of these, formulation of the GIM is the trickiest

  DOI: https://doi.org/10.30564/aia.v3i1.2920

of the lot. It may be stated here that most of the literatures available as on date are helpful in getting inner information on a specific input server, but those do not address the issue of MIMO type protocol / algorithm using various input servers in time domain. Despite not being compared with a ditto, the developed protocol can be benchmarked for performance in piece-meal manner. For example, algorithms for each of the input-servers (web, telephony & WAP), as modeled here, can attain above-average performance when compared to counterparts using single input-server, having relatively simpler control paradigm.

6. Test & Trial Runs of the Developed MIMO Protocol

6.1 Testing of the Basic System Hardware with Interfacing

A novel hardware was set up with necessary arrangement for testing the basic system architecture, in order to proceed systematically towards doing trial runs on the developed MIMO protocol for both the end-applications, namely, servomotor system & robotic sensory system. This was felt crucial to establish the very basic of the hardware interfacing paradigms of the developed protocol, in a generic manner. We made this small hardware for interfacing the PC to the parallel printer port, as the maiden trial. We have used the hardware to check the effectiveness of the protocol in glowing a Light Emitting Diode (LED), connected to the parallel port. The original IBM-PC's parallel printer port had a total of 12 digital outputs and 5 digital inputs, accessed via 3 consecutive 8-bit ports in the processor's I/O space. The parallel port terminates into 24-pin 'D' connector; out of which 8 & 4 output pins are accessed via the data port and control port respectively. Figure 28 schematically shows the pin-layout of female D-type connector that was used for the hardware set-up & experimentation, as detailed in the adjacent table.

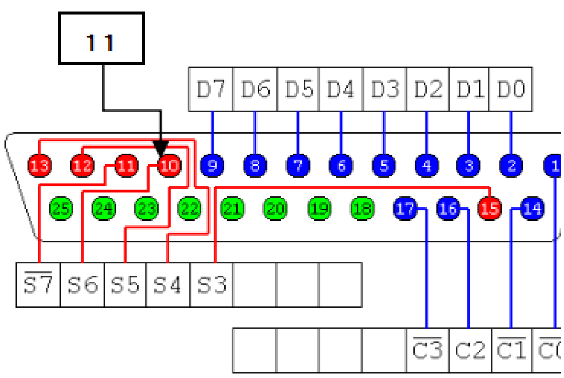

Figure 28. Semantics of data bits and pin-layout of female D-type connector used for the hardware set-up

Port	Pin nos.	Data bits	Usage
378h	2-4	D0-D2	To output the address of device
378h	6	D4	To output on/off (1/0) state
378h	7	D5	To turning monostable on/off (1/0)
379h	11	S7	For input status of addressed device
	18-25	Ground	To be merged with interface ground

It is to be noted that only one bit, viz. S7: pin 11 has been used for reading the input status, after due inverting made by the PC. The pins # 18-25 (green coloured; ref Figure 28) are ground pins. *IBM 25-core®* printer cable was used for the hardware connection, using this 'DB-25' connector. The connection between data (e.g. a register bit) and pins was considered direct wherein the data 1 was associated with an electrical TTL high, and inverted if data 1 was associated with TTL low. An overall connection (data to TTL to data) was considered direct if output of '1' produced '1' on input at the other end, or inverted if output of '1' produced '0' on the other end. The pins, D0 to D7, are labeled as the Data Out register bits, with 'D0' being the least significant bit and 'D7' the most significant. In contrast, the control out bits (C0 to C3) as well as 'S7' pin of the status in bits (S3 to S7: corresponding to data & CPU bit positions) are inverted.

6.2 Testing of the Servomotor System Using MIMO Protocol Developed

The test as well as trial-runs of the servomotor system is realized in two phases, namely, 'integration test' & 'system test. The basic integration test method that was used in case of actuation of WTP-driven servomotor system is 'bottom up' approach, involving several integration steps. Since the overall developed protocol is intense, individual components were combined with certain auxiliary components to make sure that necessary communication, links & data transfer occur properly. 'Bottom up' testing is more intuitive, which involves individual testing of each module like connection establishment, data transfer, security etc. using a driver routine that calls the module and provides it with needed resources. Integration of various modules used in the present work produces system-level behavior in new system, so interface errors surface late in the process.

The system test phase occurs in parallel with integration testing and it begins once modules are integrated enough to perform test in the whole system environment. Qualitative tests were performed in order to check the system performance with respect to three attributes, namely, a] connectivity b] data transfer & c] initial validation. The first attribute is invoked to check whether connection can be established between two devices seamlessly. To achieve this, we ran the application on two devices just for checking the appearance of the default list of peer-group in both the devices. The successful appearance of the list on both the devices ensured establishment of the secured connection that is transparent to the registered user(s). On the other hand, the trials for checking the connectivity between mobile device & PC were also successful. This test was also aimed at checking the device discovery functionality (of mobile devices), wherein it is expected that the 'mother' device should be able to detect all the compatible mobile devices in the neighbourhood. We have found the test successful, as all of the supported devices connected via USB or infrared port got detected. However, it was observed that in case two devices got into device connectivity simultaneously, they failed to discover each other. In all of these test-modules, either of the two available software, namely, PCCSDK2.1 or PCCSDK3.0 (PC Connectivity Software Development Kit 2.1 / 3.0) was used, depending upon the model of the mobile phone (make: NOKIA®). A total of 33 different phone (handset) models have been used in the testing. The details of the phone models used with connection parenthesis and adaptability to the PC-connectivity software (PCCSDK) is provided in Annexure-I.

Sample applications for PCCSDK 2.1 were written in Visual Basic, Visual C++ and Delphi. We also re-checked all phone models, supported by PCCSDK 2.1, about their usage of customized libraries from this SDK. In fact, each application-example showed the usage of different libraries, viz., a] General Setting Library (STTNGS3A_SLib); b] Short Message Library)SMS3ASuiteLib); c] Phonebook Memory Library (PhonebookAdapterDS3); d] Calendar Library (CALADAPTERLib) and e] WAP Adeapter Libraty (NokiaCLWAP). Nokia PCCSDK2.1™ may be referred for a detailed description of each of these libraries, except Unicode messages, which are not fully supported by the phone models used in the testing. Unlike PCCSDK2.1, sample applications for the SDK version 3.0 were coded in Visual Basic only. We have used the following libraries under PCCSDK3.0, namely: a] SMS Library (NokiaCLMessaging); b] Calendar Library (NokiaCLCalendar); c] Call Library (NokiaCLCall); d] General Settings Library (NokiaCLSettings); e] Voice Library (NokiaCLVoice) and f] WAP Adapter Library (NokiaCLWAP). For a detailed description on these libraries, Nokia PC Connectivity SDK 3.0 - Component Library Reference may be referred. Nonetheless, functions such as eMailNotifier, Phonebook Settings, Ringing Tone, SMS Edit, SMS Settings, and SMSeMail were supported effectively by both the versions of PCCSDK. It is to be noted that Nokia PCCSDK is a sophisticated and easy-to-use programming interface for Nokia mobile phones. The ensemble library of it consists of several separate libraries, each performing a specialized set of tasks related to mobile phone functionality. All of these libraries are implemented as Component Object Model (COM) libraries, which is the specified nomenclature for Microsoft's basic object technology that defines the standard for integration between software components. In totality, 10 different libraries are performing the task ensemble of WAP-based communication, aided by 10 different DLLs. The detailed listing, briefly describing the libraries contained in Nokia PC Connectivity SDK is provided in Annexure-II.

The client application for the actuation of the servomotor exploits these libraries through object libraries, which can be considered as binary description of the component library. The object libraries that correspond to the libraries of Nokia PC Connectivity SDK libraries are mentioned in parenthesis in Annexure-II. Each library in Nokia PC Connectivity SDK contains one or more functional entities (reusable software components), which present their functionality through a defined set of interfaces. An interface contains a collection of related properties, methods, and functions that are grouped together under one name. A client application creates an instance of a component and a component object, sets a reference to the desired interface, and accesses interface methods through this reference. Interfaces are divided into two categories, viz. incoming & outgoing, depending upon the nature of invocation. Methods of incoming interfaces are implemented on the component object and receive calls from external clients. The object performs the desired service and then returns the results to the client. In fact, most interfaces in this component library are incoming interfaces that are called by the client application. Meanwhile, methods (events) of outgoing interfaces are implemented on the client's sink and receive calls from the object. The object defines an interface it would like to use, and the client implements it. Thus, outgoing interfaces enable the object to talk back to its client, usually by notifying the client when something important happens in its sphere and/or some asynchronous operation has been completed.

Irrespective of the PC connectivity SDK 2.1 or 3.0, we ran the data exchange application in one device with a test file in order to validate if data was being transferred correct between devices. For effective connectivity with Nokia PCCSDK software, IrDA (Infrared Data Association) connection cable was used along with compatible Bluetooth connection & USB connection cables. The test file was found to be received correctly at the device on the other end and the same was verified multiple times. In order to check the proper functioning of 'initial validation', i.e. to check if the application by itself on a device provides security to the user of the device, incorrect passwords were entered in the system. It was found that the system was not allowing the use of application to any unauthorized user. The validity tests were also performed for SMS, wherein the validity limit for

DOI: https://doi.org/10.30564/aia.v3i1.2920

the device parameters was set a-priori by local administrator at local server side. The received device parameters were checked against the set-value. The tests were run successfully and it was found that only those parameters that were in the specified validity limit & following specific format of the SMS were conveyed to the remote device. The tests worked effectively and the GUI prompted with error message for the incorrect parameters. Details of the testing are provided in sub-section 6.4, wherein findings have been segregated in two modules, viz. a] angular rotation of the motor shaft & b] time elapsed.

6.3 Testing of the Slip Sensory System Using the Developed MIMO Protocol

The slip sensory system was tested repeatedly using the developed MIMO protocol through all the three pathways, namely, web-based; telephony-interface-based & WAP-interface based systemization. This testing is branched into two groups, viz. a] testing for generation of synthetic force / voltage (refer Figure 10, Equation 12 & allied text) and b] testing for transmission of design specification & report. In the first tuple, we have used three input servers discretely in order to generate equivalent voltages(‘V_{I1}^{t}’, ‘V_{I2}^{t}’ & ‘V_{I3}^{t}’), which finally correspond to synthetic force $F_{synrhetic}^{t}$), as described earlier. Likewise, external excitation was also applied to the sensor that resulted in $V_{ext}^{t.}$. Thenafter, comparison was made between $F_{conjugate}^{t}$ and summation {F_{ext}^{t} +$F_{synrhetic}^{t}$ }. For the testing for transmission of design report, we checked the effectiveness of the system with respect to time of transmission (in sec.).

The sensor being a stationary & stand-alone device, the real-time realization of the system was made through GUIs. All the possible forms & formats of the output modality, as described in Figure 5, have been invoked in the GUIs. In that sense, users can attempt to run the sensory system from different geographic regions and can get access to the output data, e.g. slip-tuple, in various possible formats (fax, SMS etc.). Figure 29 illustrates the screenshot of the output functionalities, which were tested successfully in several sequences. As presented in the screenshot, various text files & output of the .asp files, as shown in Figure 22-27, have been incorporated through output data files. For example, a particular data file, e.g. ‘Design Specification Report’, can be transmitted via remote access, fax or SMS, depending upon the situation as well as requirement. This platform-invariance is the most unique feature of the developed protocol that was tested in order to check the repeatability of the program-steps. While web-based remote transmission is being manifested through secured IP address the fax transmission is being invoked using specific fax number with country & STD code. Likewise, SMS based transmission is being performed using specific mobile number of the remotely-located user. It is to be mentioned here that although the operational syntax of the test-module has been presented through a unified GUI of Figure 29, the actual systemization of the real-time transmission is getting synchronized through multiple program-loops and network-based short-time decision markers, e.g. waiting period en-route transmission, system jitter, time elapsed for in-transfer security verification etc. For testing, two important software resource tools were used, namely: a] Browser of IBM Voice Server, VXML and b] Personal Web Server, PWS. Besides, two supporting software resources were used during testing, viz. a] *Visual Builder*, which is the foremost internal development shell for VXML and b] *Rational Rose* 2000 for UML modeling. The hardware of the slip sensory system was connected to Ethernet LAN 10/100 Mpbs network cards through appropriate cables, hubs & switches.

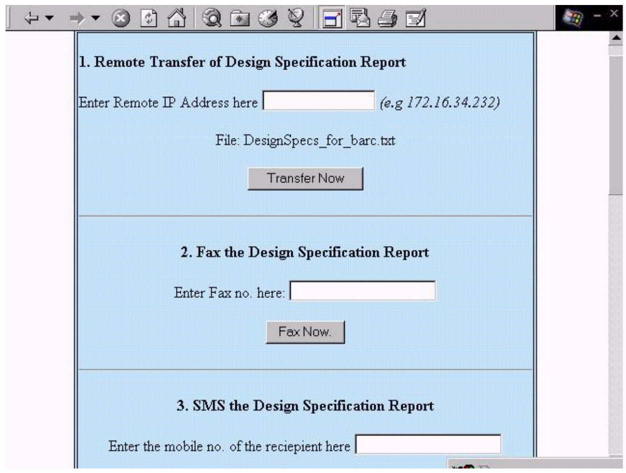

Figure 29. Screenshot showing the output functionalities of the MIMO protocol for testing of slip sensory system

6.4 Results and Discussions

We may appreciate that the case studies described in this section have two functional wings, viz. a] *proof of concept* & b] *realization of the physical robotic systems*. The motivation for the first attribute was aimed towards establishment of the basic workability of the new methodology / MIMO protocol, by and large. As the final system involves a number of intricacies specific to the controller of the end-device, we decided to start experimenting with grass-root level item, which is easily available and understandable. The functional modules of the LED were selected for that purpose. In true sense, testing with this set-up is the ‘proof of concept’ of the developed protocol.

In contrast to it, the culmination of the second attribute (realization of robotic systems) was need-based, as we experimented with two complementary philosophies. The activation & real-time operation of the servomotor assembly through the developed protocol is the direct ramification of the present work, while sensing of force in real-time through 'virtual activation' of the slip sensor is an indirect validation. Servomotor system being one of the fundamental items of any robotic device, its performance in real-time is to be monitored for the final outcome expected out of the robotic device. Likewise, sensing of slippage by robotic gripper is very important for all kinds of grasp because a slip-resistant grasp defines the performance index of the robotic system, in general. Hence, practical usability of the developed protocol & methodology with respect to these two robotic systems was the deciding factor behind the selection of case studies. The motto of all these case studies was to have qualitative validation of the protocol and verification of the program-syntax. In that respect, our experimentation was not aimed to generate quantitative data. In fact, that was not feasible either. So far as qualitative validation is concerned, our experiments have generated satisfactory output for the developed protocol. Nonetheless, findings from the experimentations have been compiled now in tabular form so as to throw better understanding of the test-matrix. Table 1 presents the raw test data for the actuation of servomotor system for 20 testings. Servomotor system was activated using three input servers in succession, in order to achieve a target rotation of the motor shaft (10^0 in this case). The achieved rotation of the motor shaft was measured through the encoder, attached with the servomotor.

Table 1. Test Data on Servomotor Actuation: Done Using the Developed MIMO Protocol

Sl. No.	Type of Input Server Used		
	Web	Telephony	WAP
	Target Rotation of Motor Shaft: 10^0		
Rotation of the Motor Shaft Achieved (degrees) & Actuation Time (sec.) [in bracket]			
1	10.2 (79.8)	10.5 (78.6)	10.2 (79.4)
2	10.1 (78.9)	10.7 (79.4)	10.3 (79.6)
3	10.3 (80.1)	10.4 (80.2)	10.2 (79.6)
4	9.9 (79.7)	10.2 (79.8)	10.2 (80.2)
5	10.3 (80.2)	10.3 (79.9)	10.5 (80.1)
6	10.2 (79.7)	10.4 (79.5)	10.5 (79.5)
7	9.8 (79.4)	10.4 (79.8)	10.2 (79.6)
8	9.9 (79.3)	10.3 (80.1)	10.3 (80.3)
9	10.1 (79.4)	10.3 (80.3)	10.6 (79.4)
10	10.2 (79.3)	10.4 (79.4)	10.3 (79.8)
11	10.3 (79.2)	10.2 (79.5)	10.3 (79.4)
12	9.9 (79.6)	10.3 (80.3)	10.2 (79.4)
13	10.2 (78.9)	10.3 (79.9)	10.2 (79.8)
14	10.3 (79.6)	10.2 (79.5)	10.6 (80.2)
15	10.2 (79.9)	10.2 (79.4)	10.4 (79.8)
16	10.1 (79.8)	10.3 (79.6)	10.4 (79.6)
17	10.2 (79.7)	10.4 (79.6)	10.4 (79.5)
18	10.1 (79.7)	10.3 (79.9)	10.3 (79.6)
19	9.9 (79.9)	10.2 (78.9)	10.4 (80.3)
20	10.2 (79.6)	10.2 (79.9)	10.5 (79.8)

The raw test data of Table 1 reveals that by & large, the rotation of the motor-shaft maintains uniformity with very minor fluctuations. Those differences in readings can be attributed to various extraneous reasons, like setting of the experiment, server-noise & delay etc. Likewise, the time of actuation also follows uniformity, irrespective of the type of input used.

Table 2 presents the raw test data for the actuation of the slip sensor system for 20 testings, in two modules. As part of the first module involving excitation of the sensor & activation through input servers, the data have been tabulated in 10 columns (col. 1 to col. 10). As detailed in the index below Table 2, data under four columns (viz. cols. 1, 2, 3 & 8) are the real-time raw data off the system, while the rest of the data (under cols. 4, 5,6,7,9, & 10) are derived. For example, the derived data at col. 4 ($V_{synthetic}$) is the maximum value of data from cols. 1-3, while data at col. 8 is the external excitation (in μN). As reported earlier, linear regression has been used for the calibration of the slip sensory system, both in case of synthetic forcing as well as direct external excitation. Accordingly, the numerical values of the derived units were calculated, considering value of slope 'm' (refer Equation 13a, 13b & 13c) as 0.736. Data at col. nos. 5 & 7 were evaluated by using Equation 13a & 13b, while that at col. no. 10 was calculated using Equation 13c.

Besides, the response of the slip sensory system was verified through the transmission of a data file (Design Specification Report: refer GUI of Figure 29) using three input servers in succession. The transmission times (in sec.) were noted in each case, which are tabulated in the last three columns of Table 2.

The raw test data of Table 2 show that by & large, data from a specific input server vary within a very small range and the system output is quite stable, in terms of milli-volts thereof. Besides, near-closeness of the numerical values of col. no.s 9 & 10 proves the lemma of Equation 12. The data on the transmission time under three servers (I_1, I_2 & I_3) show stable nature of the transmission. Minor variation in the data, both intra- & inter-servers, can be attributed to system jitter & network delay.

5. Conclusions

In the present research, we have investigated the complete ensemble of physically-realizable pathways for accessing local as well as web-enabled data. The MIMO format through which the entire analysis is reported is optimum and generic in nature. An exhaustive treatise on the classification, nomenclature, normalization & clustering of data have been reported in the paper. This analysis on raw data, emanating from engineering system

DOI: https://doi.org/10.30564/aia.v3i1.2920

Table 2. Test Data on Slip Sensor Actuation: Done Using the Developed MIMO Protocol

Sl. No.	Voltage at Input Servers (mv.)			External Excitation & Synthetic Force Values at time 't'							Transmission Time (sec.)		
	Web	Telephony	WAP	Voltage (mv.)			Force (μN)				Input Servers		
	V_{I1}^{t} ::[1]	V_{I2}^{t}:: [2]	V_{I3}^{t}:: [3]	$V_{synthetic}$::[4]	V_{ext} ::[5]	$V_{synthetic}$ + V_{ext} :: [6]	ΔF_{syn}^{t} ::[7]	F_{ext}^{t} ::[8]	F_{ext}^{t} + ΔF_{syn}^{t} ::[9]	$F_{conjugate}^{t}$::[10]	I1	I2	I3
1	25.8	23.7	23.9	25.8	24.33	50.13	18.988	17.91	36.898	36.89568	18.6	18.9	17.9
2	25.4	23.9	23.8	25.4	24.93	50.33	18.694	18.35	37.044	37.04288	18.5	17.9	18.8
3	24.8	24.2	23.7	24.8	25.26	50.06	18.253	18.59	36.843	36.84416	18.4	18.3	17.7
4	25.2	24.0	23.9	25.8	25.08	50.88	18.988	18.46	37.448	37.44768	17.8	18.3	18.5
5	25.6	24.2	23.7	25.6	26.32	51.92	18.812	19.37	38.182	38.21312	18.2	18.4	18.6
6	24.5	23.9	24.1	24.5	27.48	51.98	18.032	20.23	38.262	38.25728	18.4	17.7	18.9
7	23.8	23.8	24.0	24.0	28.64	52.64	17.664	21.08	38.744	38.74304	18.5	18.4	18.7
8	25.7	23.7	23.9	25.7	28.23	53.93	18.915	20.78	39.695	39.69248	18.1	18.6	18.3
9	25.5	23.9	23.8	25.5	25.49	50.99	18.768	18.76	37.528	37.52864	18.6	19.1	18.8
10	23.6	23.8	23.8	23.8	27.93	51.73	17.516	20.56	38.076	38.07328	18.9	18.6	18.4
11	24.7	24.1	23.7	24.7	28.92	53.62	18.179	21.29	39.469	39.46432	18.6	18.2	18.5
12	24.8	24.2	23.9	24.8	25.76	50.56	18.253	18.96	37.213	37.21216	18.5	18.4	17.9
13	25.1	24.1	23.8	25.1	25.49	50.59	18.474	18.76	37.234	37.23424	18.6	18.3	18.8
14	24.4	23.7	23.9	24.4	26.39	50.79	17.958	19.43	37.388	37.38144	18.5	18.3	18.9
15	24.9	23.6	23.7	24.9	27.65	52.55	18.326	20.35	38.676	38.6768	18.4	18.5	18.6
16	25.4	23.9	24.1	25.4	28.09	53.49	18.694	20.68	39.374	39.36864	17.9	18.4	18.3
17	25.2	23.8	24.2	25.2	28.58	53.78	18.547	21.04	39.587	39.58208	18.8	18.1	18.1
18	25.0	23.5	23.9	25.0	25.91	50.91	18.4	19.07	37.47	37.46976	18.4	18.4	18.6
19	24.7	24.2	23.8	24.7	25.47	50.17	18.179	18.75	36.929	36.92512	18.9	17.9	18.5
20	25.5	23.7	23.9	25.5	28.21	53.71	18.768	20.76	39.528	39.53056	18.4	18.3	18.4

[Index: col. no. 1:: [1] → V_{I1}^{t} ; col. no. 2:: [2] → V_{I2}^{t}; col. no. 3:: [3] → V_{I3}^{t}; col. no. 4:: [4]→ $V_{synthetic}$; col. no. 5:: [5]→ V_{ext}; col. no. 6:: [6]→{$V_{synthetic}$ +V_{ext}}; col. no. 7:: [7]→ ΔF_{syn}^{t}; col. no. 8:: [8]→ F_{ext}^{t}; col. no. 9::[9]→ {F_{ext}^{t} + ΔF_{syn}^{t}}; col. no. 10:: [10]→ $F_{conjugate}^{t}$; I1: Web Server; I2: Telephony Server; I3: WAP server]

in real-time, is very crucial in overall mapping of the input-output format for the end-device. Since robotic systems, by & large, require good volume of data for activation due of their dependency upon various sub-systems, e.g. sensors, actuators, linkages, this modeling on raw data is entrusted to be effective. Apart from analytical aspects, the paper delineates in detail on the hardware manifestation of the developed MIMO architecture for servomotor-driven system and the robotic sensory system. Detailed hardware architecture for the three servers, namely web, telephony & WAP, has been reported in the paper with an insight to compatible software programming. System firmware for both the case-studies has been established to a robust level, suitable for regular use in industrial robotic environment. The developed MIMO protocol, in its complete ensemble with hardware & software, was tested successfully for both servomotor system (through WAP-connectivity using Nokia® phonesets) as well as robotic sensory system (through GUI-based remote communication). Results of these experimentations are encouraging enough to chalk out the plan for extensive field-trials of the physical system(s) simultaneously. The operational paradigms for executing a multi-input multi-output media-independent communication protocol in real-time have been established robustly through the present work. Besides novelty of the models proposed, this research brings out the actual pathway for implementing a MIMO-based transmission mechanism, using local & web-based information from various media-platforms in real-time. The methodology will serve as an effective bridging between various media-platforms in real-time. The invocation of the protocol will be user-friendly too, as it has direct handshaking with the device-drivers (through DLLs) and handy GUI.

The developed method & protocol has immense utilization potential in maneuvering several physical devices in real-time through different input servers. The method should be preferred over other available alternatives because of its vast technological potential, freedom of selection of input & output servers / media, effectiveness in coagulating media-platforms, user friendliness and easy adaptability.

Acknowledgement

The author acknowledges the financial assistance of the Department of Atomic Energy, Govt. of India for carrying out the prototype developmental work of the robotic gripper assemblies and respective applications in the laboratory, as part of the planned R&D outlay of Bhabha Atomic Research Centre, Mumbai, India during 2012-2014. The author would also like to express sincere gratitude to Shri Atul Agarwal, Shri Pratik Patwari & Shri Vishal Changrani, graduate students of K.K. Wagh College of Engineering, Nashik, India (affiliated to Pune University, India) in developing part of the protocol related to robotic sensory system, in course of their research internship under the guidance of the author. Sincere acknowledgement is also due to Shri Pankaj Bedse, research intern from Pune University, India for developing a part of the wireless transport protocol, during his research internship under the supervision of the author.

References

[1] Tak-Lam Wong and Wai Lam, "Learning to Adapt Web Information Extraction Knowledge and Discovering New Attributes via a Bayesian Approach", IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 4, April 2010, pp 523-536.

[2] Xiaohui Tao, Yuefeng Li, and Ning Zhong, "A Personalized Ontology Model for Web Information Gathering", IEEE Transactions on Knowledge and Data Engineering, vol. 23, no. 4, April 2011, pp 496-511.

[3] Calvert, K., Doar, M. & Zegura, E.W., "Modeling Internet Topology", IEEE Communication Magazine, June 1997, pp 160-163.

[4] W. Cohen & W. Fan, "Learning Page-Independent Heuristics for Extracting Data from Web Pages," Computer Networks, vol. 31, ,nos. 11-16, 1999, pp. 1641-1652.

[5] Deng Cai, Xiaofei He and Jiawei Han, "Locally Consistent Concept Factorization for Document Clustering", IEEE Transactions on Knowledge and Data Engineering, vol. 23, no. 6, June 2011, pp 902-913.

[6] Y. Yuan, X. Li, Y. Pang, X. Lu, and D. Tao, "Binary Sparse Nonnegative Matrix Factorization," IEEE Transactions on Circuits and Systems for Video Technology, vol. 19, no. 5, May 2009, pp. 772-777.

[7] Bouguessa, M. and Shengrui Wang, "Mining Projected Clusters in High-Dimensional Spaces", IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 4, April 2009, pp 507-522.

[8] Mohamed Bouguessa and Shengrui Wang, "Mining Projected Clusters in High-Dimensional Spaces", IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 4, April 2009, pp 507-522.

[9] Chih-Ming Hsu and Ming-Syan Chen, "On the Design and Applicability of Distance Functions in High-Dimensional Data Space", IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 4, April 2009, pp 523-536.

[10] Barbara Hammer[1], Marc Strickert[1] , and Thomas Villmann[2], "Learning Vector Quantization for Multimodal Data", Internet Resource [1: Department of Mathematics/Computer Science, University of Osnabr¨uck, D-49069 Osnabr¨uck, Germany; 2: Clinic for Psychotherapy and Psychosomatic Medicine, University of Leipzig, Karl-Tauchnitz-Straße 25, D-04107 Leipzig, Germany].

[11] A.K. Jain, M.N. Murty, and P.J. Flynn, "Data Clustering: A Review," ACM Computing Surveys, vol. 31, no. 3, 1999, pp. 264-323.

[12] V. Pestov, "On the Geometry of Similarity Search: Dimensionality Curse and Concentration of Measure," Information Processing Letters, vol. 73, nos. 1/2, 2000, pp. 47-51.

[13] Yi-Hong Chu, Jen-Wei Huang, Kun-Ta Chuang, De-Nian Yang, and Ming-Syan Chen, "Density Conscious Subspace Clustering for High-Dimensional Data", IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 1, January 2010, pp. 16-30.

[14] Kalogeraki, V. and Chen, F., "Managing Distributed Objects in Peer-to-peer Systems", IEEE Network, vol. 18, no. 1, Jan./Feb. 2004, pp 22-29.

[15] Shipeng Yu, Kai Yu, Volker Tresp, and Hans-Peter Kriegel, "Multi-Output Regularized Feature Projection", IEEE Transactions on Knowledge and Data Engineering, vol. 18, no. 12, December 2006, pp 1600-1613.

[16] Amer, P.D., Chassot, C., Connolly, T.J., Diaz, M. & Conrad, P., "Partial-order transport service for multimedia and other applications", IEEE/ ACM Transactions on Networking, Vol. 2, No. 5, 1994, pp 440-456.

[17] Cowie, J., Nicol, D. & Ogielski, A., "Modeling the Global Internet", Computer Science and Engineering, Vol. 1, No. 1, Jan. 1999, pp 42-50.

[18] Evrim Acar and Bu¨ lent Yener, "Unsupervised Multiway Data Analysis: A Literature Survey", IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 1, January 2009, pp 6-20.

[19] H.A.L. Kiers, "Towards a Standardized Notation

  DOI: https://doi.org/10.30564/aia.v3i1.2920

and Terminology in Multiway Analysis," Journal Chemometrics, vol. 14, no. 3, 2000, pp 105-122.

[20] Longbing Cao, Yanchang Zhao, Huaifeng Zhang, Dan Luo, Chengqi Zhang, and E.K. Park, "Flexible Frameworks for Actionable Knowledge Discovery", IEEE Transactions on Knowledge and Data Engineering, vo. 22, no. 9, September 2010, pp 1299-1312.

[21] L. Cao, "Domain-Driven Actionable Knowledge Discovery," IEEE Intelligent Systems, vol. 22, no. 4, July/Aug. 2007, pp 78-89.

[22] Yingpeng Sang, Hong Shen and Hui Tian, "Privacy-Preserving Tuple Matching in Distributed Databases", IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 12, December 2009, pp 1767-1782.

[23] Marek J. Druzdzel and Francisco J. D´iez, "Combining Knowledge from Different Sources in Causal Probabilistic Models", Journal of Machine Learning Research, vol. 4, 2003, pp 295-316.

[24] Floyd, S., Jacobson, V., Liu, C.G., McCanne, S. & Zhang, L., "A reliable multicast framework for light-weight sessions and application level framing," IEEE/ ACM Transactions on Networking, vol. 5, no. 6, 1997, pp. 784-803.

[25] Pietzuch, P.P., Shand, B. and Bacon, J., "Composite Event Detection as a Generic Middleware Extension", IEEE Network, vol. 18, no. 1, Jan./Feb. 2004, pp 44-55.

[26] Xu, J. and Lipton, R.J., "On Fundamental Tradeoffs between Delay Bounds and Computational Complexity in Packet Scheduling Algorithms", IEEE/ACM Transactions on Networking, vol. 13, no. 1, Feb. 2005, pp 15-28.

[27] Bolot, J., Shankar, A. and Plateau, B., "Performance Analysis of Transport Protocols Over Congestive Networks", Journal of Performance Evaluation, vol. 11, 1990, pp 45-65.

[28] Gustavo de Veciana, Tan-Jin Lee & Takin Konstantopoulos, "Stability and Performance Analysis of Networks Supporting Elastic Services", IEEE/ ACM Transactions on Networking, Vol. 9, No. 1, Feb. 2001, pp 2-14.

[29] Floyd, S., "TCP and Explicit Congestion Notification", ACM Computer Communication Review, vol. 24, no. 5, Oct. 1994, pp 10-23.

[30] Wu-Chang Feng, Dilip D. Kandlur, Debanjan Saha & Kang G. Shin, " Understanding and Improving TCP Performance Over Networks With Minimum Rate Guarantees", IEEE/ ACM Transactions on Networking, vol. 7, no. 2, April 1999, pp 173-187.

[31] Lakshman, T.V., Madhow, U. & Suter, B., "TCP/IP Performance With Random Loss and Bidirectional Congestion", IEEE/ ACM Transactions on Networking, Vol. 8, No. 5, Oct. 2000, pp 541-555.

[32] Roy, Debanik, Chatterjee, A., Jasapara, N. & Jadhav, N., "A New Transport Layer Protocol Offering Variable Reliability in Bi-Directional Communication: Application for Robotic Systems", International Journal of Computers and Applications, Vol. 28, No.3, August 2006, pp 251-258.

[33] Roy, Debanik, "Novel Network Protocol for Bi-directional data transmission using Variable Reliability: Phase II-Case studies with Robotic Sensor and Manipulator", International Journal of Automation, Mechatronics & Robotics, vol. 3, no. 1, [IJAMR-16-V3I1-103], 2016.

[34] Bo Yuan, Maria Orlowska, and Shazia Sadiq, "On the Optimal Robot Routing Problem in Wireless Sensor Networks", IEEE Transactions on Knowledge and Data Engineering, vol. 19, no. 9, September 2007, pp 1252-1261.

[35] Abraham, S.P. & Kumar, A., "A New Approach For Distributed Explicit Rate Control of Elastic Traffic in an Integrated Packet Network", IEEE/ ACM Transactions on Networking, vol. 9, no. 1, Feb. 2001, pp 15-30.

[36] Gafni, E.M. & Bertsekas, D., "Dynamic Control Session Input Rates in Communication Networks", IEEE Transactions on Automatic Control, vol. 29, 1984, pp 1009-1016.

[37] Arnaud Legout, Jorg Nonnenmacher & Ernst W. Blersack, "Bandwidth Allocation Policies For Unicast and Multicast Flows", IEEE/ ACM Transactions on Networking, vol. 9, no. 4, Feb. 2001, pp 464-478.

[38] Mark, A. S. and Ramakrishnan, K.K., "Formal Specification and Verification of Safety and Performance of TCP Select Acknowledgement", IEEE/ACM Transactions on Networking, vol. 10, no. 2, April 2002, pp 193-207.

[39] Paxson, V., "End-to-end Internet Packet Dynamics", IEEE/ACM Transactions on Networking, vol. 7, June 1999, pp 277-292.

[40] Paxson, V., "End-to-end Routing Behavior in the Internet", IEEE/ACM Transactions on Networking, vol. 5, Oct. 1997, pp 601-615.

[41] Pei,D., Zhang, L. and Massey, D., "A framework for Resilient Internet Routing Protocols", IEEE Network, vol. 18, no. 2, March/April 2004, pp 5-12.

[42] Sen, S. and Wang, J., "Analyzing Peer-to-peer Traffic Across Large Networks", IEEE/ACM Transactions on Networking, vol. 12, no. 2, April 2004, pp 219-232.

[43] Roy, Debanik & Chatterjee, A., "A Distributed Generic Architecture for User-interactive Internet-

based Remote Activation Towards Manoeuvring Mechatronic Systems in Tandem", Journal of Intelligent and Robotic Systems, vol. 45, no. 3, March 2006, pp 217-233.

[44] William C.Y. Lee, "Mobile Communications Engineering: Theory and Applications", 2nd. Edition, The Mc-Graw Hill Companies Inc., 1998, ISBN: 9780070371033.

[45] Gordon Stuber, "Principles of Mobile Communication", Kluwer Academic Publishers, 1996, ISBN: 0-7923-9732-0.

[46] J. D. Parsons, "Mobile Communication Systems", Halsted Press, 1989, ISBN 0-470-21213-6.

[47] Tom Logsdon, "Mobile Communication Satellites: Theory & Applications", McGraw-Hill Publishing, 1995, ISBN: 0-07-038476-2.

[48] Peter Wong & David Britland, "Mobile Data Communications Systems", Artech House, 1995, ISBN 0-89006-751-1.

[49] Christoffer Andersson, "GPRS and 3G Wireless Applications: Professional Developer's Guide", John Wiley & Sons, 2002, ISBN: 0471189758, 9780471189756.

[50] Web: http://www.nokia.com.

Annexure-I. Model, Type, Connection Method & Compatibility with PC-connectivity software for NOKIA® Handsets

Sl. No.	Phone (Handset) Model	Type	Connection Method	PC-connectivity software	
				SDK2.1	SDK3.0
1	Nokia 3320	NPC-1	IrDA (Infrared Data Association)		√
2	Nokia 3360	NPW-1	IrDA		√
3	Nokia 5110	NSE-1	DAU-9P cable	√	
4	Nokia 5130	NSK-1	DAU-9P cable	√	
5	Nokia 5190	NSB-1	DAU-9P cable	√	
6	Nokia 6110	NSE-3	DAU-9P cable	√	
7	Nokia 6130	NSK-3	DAU-9P cable	√	
8	Nokia 6150	NSM-1	DAU-9P cable	√	
9	Nokia 6190	NSB-3	DAU-9P cable	√	
10	Nokia 6385	NHP-2	IrDA		√
11	Nokia 6210	NPE-3	DLR-3P cable/IrDA/Bluetooth	√	√
12	Nokia 6250	NHM-3	DLR-3P cable/IrDA	√	√
13	Nokia 6310	NPE-4	DLR-3P cable/IrDA/Bluetooth		√
14	Nokia 6310i	NPL-1	DLR-3P cable/IrDA/Bluetooth		√
15	Nokia 6340	NPM-2	DLR-3P cable/IrDA		√
16	Nokia 6360	NPW-2	DLR-3P cable/IrDA		√
17	Nokia 6370	NHP-2	DLR-3P cable/IrDA		√
18	Nokia 6510	NPM-9	IrDA		√
19	Nokia 6590	NSM-9	IrDA		√
20	Nokia 6610	NHL-6	DKU-5 cable/IrDA		√
21	Nokia 6650	NHM-1	DKU-2 cable / IrDA/ Bluetooth		√
22	Nokia 7110	NSE-5	DLR-3P cable/ IrDA	√	√
23	Nokia 7160	NSW-5	DLR-3P cable/ IrDA	√	√
24	Nokia 7190	NSB-5	DLR-3P cable/ IrDA	√	√
25	Nokia 7210	NHL-4	DKU-5 cable/ IrDA		√
26	Nokia 8210	NSH-3	IrDA	√	√
27	Nokia 8290	NSB-7	IrDA	√	√
28	Nokia 8310	NHM-7	IrDA		√
29	Nokia 8390	NSB-8	IrDA		√
30	Nokia 8810	NSE-6	IrDA	√	√
31	Nokia 8850	NSM-2	IrDA	√	√
32	Nokia 8890	NSB-6	IrDA	√	√
33	Nokia 8910	NHM-4	IrDA/ Bluetooth		√

DOI: https://doi.org/10.30564/aia.v3i1.2920

Annexure-II. Libraries contained in Nokia PC Connectivity SDK

Library	Description
STTNGS3A_Slib	General Settings Library: adjusting settings on the GSM phone (Stngs3AS.dll)
SMS3AsuiteLib	Short Message Library: sending and receiving of messages and SMS memory management (Sms3as.dll)
PhonebookAdapterDS3	Phonebook memory, speed-dial key and caller group management (SCM3aS.dll)
CALADAPTERLib	Calendar management (Cal3aS.dll)
NOIKACLWAP	WAP Settings Library, handling WAP settings and groups (NclWAP.dll)
NokiaCLMessaging	Messaging Library, sending and receiving of messages and SMS memory management (NclMsg.dll)
NokiaCLSettings	Settings Library (NclSet.dll)
NokiaCLCall	Voice call management (NclCall.dll)
NokiaCLCalendar	Calendar management (NclCal.dll)
NokiaCLVoice	Voice memo handling (NclVoice.dll)

 DOI: https://doi.org/10.30564/aia.v3i1.2920