An Original Approach (MTT_SYS) for Time/Space Measurements Using Matlab Data Acquisition Toolbox

M. T. Tarata*, J. R. Bloedel**

*University of Medicine and Pharmacy of Craiova, Medical Informatics Dept., Str. Petru Rares 2-4, Craiova, Romania, mtarata@umfcv.ro
** Iowa State University, Health & Human Performance Dept., 178S Forker Bldg., Ames, Iowa, IA-50011, USA, jbloedel@iastate.edu

Abstract. *The paper presents an insight into addressing complex time/space measurements in biology and efficient implementations of such complex systems under Windows (Microsoft, USA) using the MATLAB (Mathworks Inc., USA) Data Acquisition Toolbox. The approach is illustrated with authors' original system (MTT_SYS) to flexibly implement complex experimental paradigms in Biomechanics, Physiology and Neuroscience. MTT_SYS gathers non-invasively 3D (position in space of joints) & physiological information from human or animal subjects, driven by external or physiological events. Comparing to other approaches, this implementation provides the quickest way to build, flexible, high performance, paradigm oriented complex systems.*

1. Introduction

The recording of three-dimensional (3D) trajectories is mandatory in complex fields, as Biomechanics, Physiology and Neuroscience, when experiments ask for specific movements of different body-segments of the subjects, triggered by external or physiological events. Depending on the experimental paradigm, these movements may also trigger events to change the course of the ongoing experiment, accordingly. A system intended to implement complex experimental paradigms, aimed to record 3D trajectories and other external or physiological signals, is primarily required to manage complex inter-conditioned event sequences, which actually define the experimental paradigm. Building such a complex system, which is much more than a data logger, is a technical challenge, involves effort, high costs and may take long. That is why, efficient implementations are needed.

To illustrate, we mention prehension, as an example of complex behaviour, whose study asks for complex tools. Complex experiments are needed to understand neural mechanisms used in prehension.

Prehension is defined as 'the application of functionally effective forces to an object for a given task' [1]. Human prehension implies reaching and grasping as two different types of motor behaviour [2]. Despite their apparent simplicity, they are complex behaviours, highly demanding, in terms of brain resources. How the prehension movement is organized remains highly debated and leaves room for further research. Experimental work showed that altering the reaching affected the formation of the grip, while altering the grip (varying the object size) affected less the kinematics of the reach [3,4]. In this respect, applying position or shape perturbations to the target has been a productive paradigm. The results from such experiments suggest the idea of a global planning of prehension [5,6] involving the cooperation of two internal representations: the 'pragmatic representation' which is automatic in action and the 'semantic representation' which is slower, asking access to 'awareness' [7,8]. The position

perturbation paradigm (unexpectedly changing the position of the target) allows the exploration of how the neural pathways cooperate to perform the trajectory correction, so the target is eventually grasped, despite the occurrence of the perturbation. To address different concepts and related experimental paradigms, different experimenters built dedicated structures [4,9,10]. Such systems had no structural and operational provision for flexibility to easily address other experimental needs without altering the structure of the system.
Besides the inherent technical characteristics regarding accuracy and precision of the analog to digital conversion, number of analog and digital input/output channels, triggering facilities, number and versatility of digital counters, which are provided by the hardware (data acquisition board et al.), the main requirement a modern system has to meet, is to be operational under Windows, i.e. having a friendly graphical user interface, while providing software access to all the resources. This allows the most efficient user-system interaction, and gives the user the possibility to easily changing the structure of the system just from software intervention, to implement different experimental paradigms without any hardware modifications. This is a most efficient solution.

To optimise the development of such a complex system, high level programming environments and software tools are needed. Using MATLAB, we developed an original real time system (MTT_SYS) to flexibly address double perturbation (DP) paradigms in the Reach & Grasp class of experiments, as well as other experimental paradigms where 3D and signal acquisition is needed when complex sequences of time events and triggers are implied [11]. The 3D acquisition is performed via an OPTOTRAK 3020 system (NDI Northern Digital Inc., Canada) from infrared (IR) markers placed on relevant locations on the body and is triggered by the main system application. Two different signals (one analog output channel and multiple digital output lines) may be generated synchronous with or delayed from any of the events. These may be used to alter the experimental conditions, thus acting as perturbation. Planning the event & perturbation interaction is done by the main system through a parameter list at the operator's disposal. Using MTT_SYS, we implemented specific experimental paradigms, to better understand the motor planning neural paths and how trajectory or grasp corrections are computed and performed under position perturbations during the reach.

2. Methods

The first essential component of any acquisition system is the hardware needed to manage analog and digital inputs/outputs, with triggering facilities and versatile digital counters. The key to build in a short time a complex system, as shown in the introduction, is a high-level programming environment. MATLAB and its Data Acquisition Toolbox (DAT) constitute a suitable tool [12].

DAT is a collection of M-file functions and MEX-file dynamic link libraries (DLLs) built on the MATLAB computing environment. The toolbox provides a framework for bringing live, measured data into MATLAB using (i) PC-compatible, plug-in data acquisition hardware, (ii) support for analog input (AI), analog output (AO), and digital I/O (DIO) subsystems including simultaneous AI and AO conversions, data buffering for background acquisitions and data logging, event-driven acquisitions.

DAT consists of three distinct components: (i) M-file functions, (ii) the Data Acquisition Engine (DAE), and (iii) Hardware Driver Adaptors (HAD). They allow passing information between MATLAB and the data acquisition hardware.

The behaviour of the data acquisition application can be controlled by configuring property values, i.e. characteristics of the toolbox or of the hardware driver that can be programmed according to the needs.

Through DAT, data and events can be handled. An event occurs at a particular time after a condition is met and may result in one or more callbacks that are specified. Events can be generated only after the associated properties have been configured.

To perform any task with the data acquisition application, M-file functions have to be called from the MATLAB environment. These functions allow (i) the creation of device objects, which provide a gateway to the capabilities of hardware and allow the control of the behaviour of the application; (ii) the acquisition or output of data; (iii) the configuration of property values; (iv) the evaluation of the acquisition status and hardware resources status.

The inner layer of the DAT is DAE, which is a MEX-file DLL that (i) stores the device objects and associated property values that control the data acquisition application; (ii) controls the synchronization of events; (iii) controls the storage of acquired or queued data.
MATLAB and DAE are asynchronous. While the engine (DAE) performs, MATLAB can be used for other tasks such as analysing acquired data.

HAD is the interface between DAE and the hardware driver. The adaptor's main purpose is to pass information between MATLAB and the hardware device via the driver.

The functioning of a complex system is event driven. An event occurs at a particular time after a condition is met and may result in one or more callbacks. They are controlled through callback properties and callback functions. Callback functions are M-file functions that are built to suit the specific data acquisition needs. All event types have an associated callback property. A callback is executed when a particular event occurs by specifying the name of the M-file callback function as the value for the associated callback property.

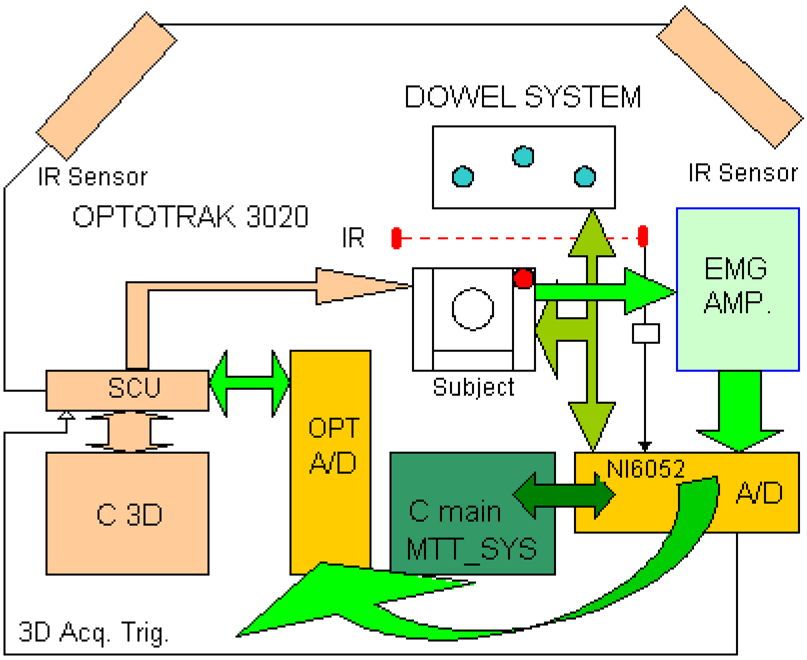

Fig. 1. MTT_SYS - the structure. The system records: (i) 3D data from 16 IR markers placed on the subject’s body, via SCU transputer & 2 IR position

sensors; (ii) physiologic signals (EMG AMP, NI6052 A/D); (iii) paradigm related signals associated to the important events: Beep, Hand_ Down - HD, Hand_Lift - HL, IR Array crossing - IR, Touch_Dowel -TD, Lift_Dowel – LD, via OPT A/D.

2.1 MTT_SYS – the structure

MTT_SYS was developed using the above described approach and MATLAB DAT. The MTT_SYS system is a bi-computer system developed under Windows 2000 (Figure 1).

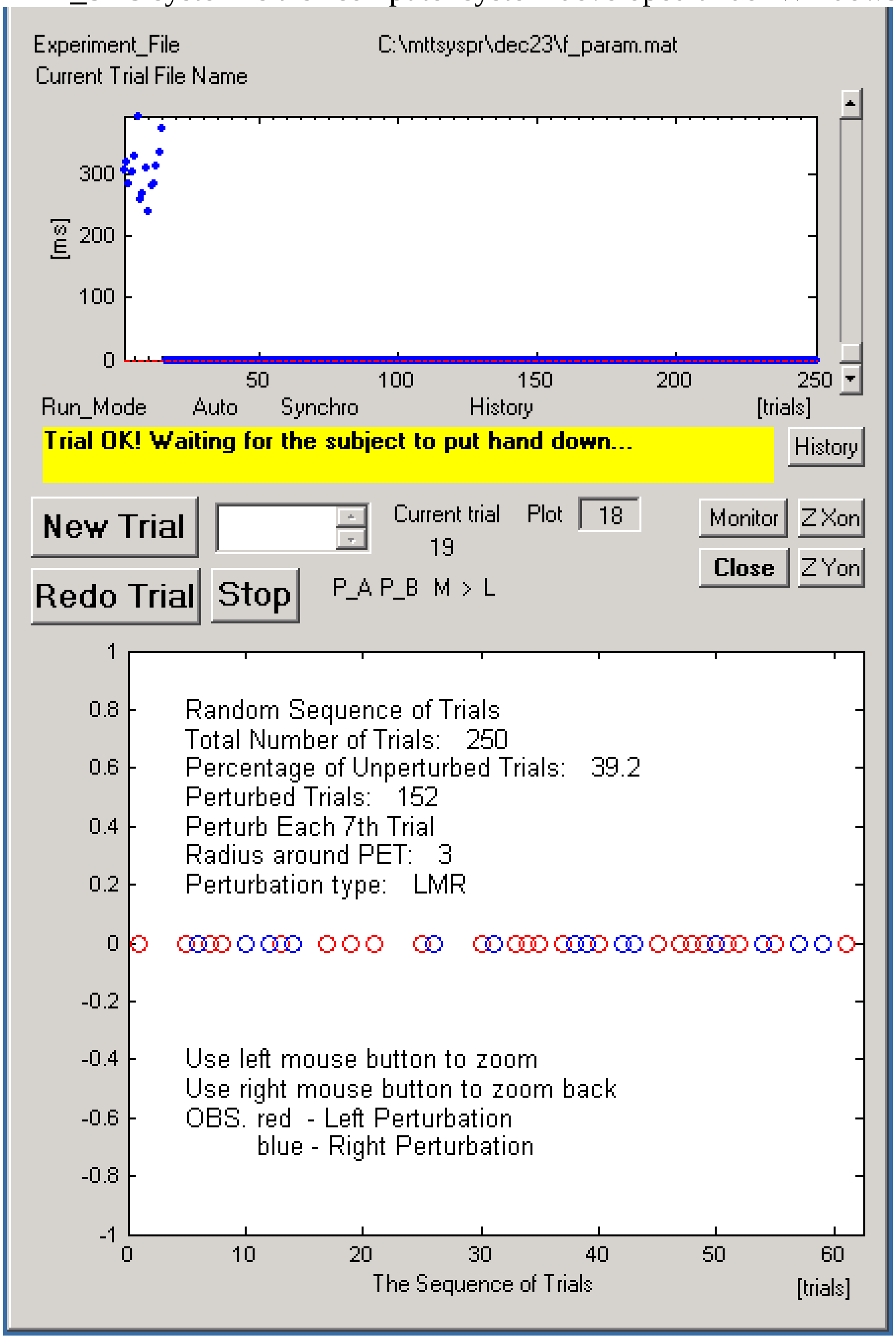

Fig. 2. MTT_SYS – Graphical User Interface (GUI) provides simple menu-driven operation and online experiment status monitoring facilities.

The main program MTT_SYS developed under MATLAB 6.1 ver. 12 (MathWorks Inc., USA) runs on the Cmain computer. On each trial, MTT_SYS triggers 3D & signal data acquisition from within a program (Toolbench, Northern Digital Inc., Canada) running on C3D computer (Figure 1). This drives the SCU transputer system (Northern Digital Inc., Canada) to perform the 3D acquisition from two sensors with 3 infrared cameras each and the acquisition of external signals through its associated OPT A/D acquisition board.

For signal acquisition & events management MTT_SYS uses NI 6052 board (National Instruments, USA). The application runs within the main GUI window (Figure 2) providing simple operation (Figure 3).

2.2 MTT_SYS – Programming

Four objects are created in the initialization phase:
handles.ai=analoginput ('nidaq'); (i) analog inputs (5 single ended channels were used to input the events: HD, IR, TD, LD, external;
handles.ao = analogoutput ('nidaq'); (ii) analog outputs (one channel was used to issue a programmable shape analog signal to be used as one of the perturbations P_A);
handles.dio=digitalio ('nidaq'); (iii) digital inputs – to read external devices status & digital outputs – to control external devices e.g. to light the target dowel etc.
handles.ws = analogoutput ('winsound'); (iv) sound device - to issue sound trigger (Beep).

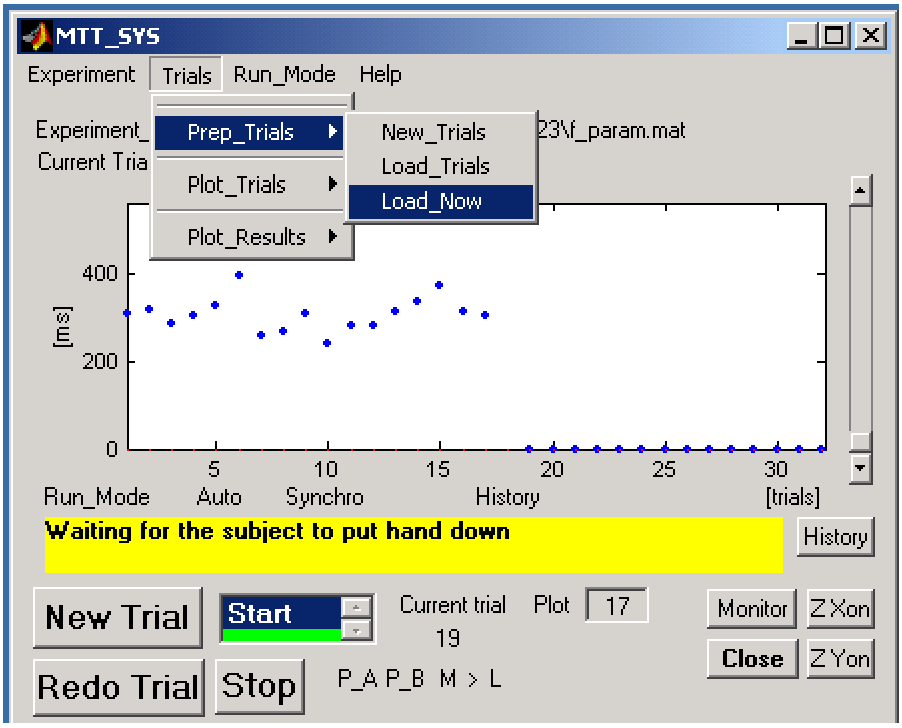

Fig. 3. MTT_SYS – Graphical User Interface (GUI) provides simple menu-driven operation and online experiment status monitoring facilities.

All the events are handled in interrupts and processed by callback functions. As an example, when the subject initiates a new trial by pushing a button, the function:
function msh_d (object, event, h, handles) is called. Among other things it does, msh_d function prepares the system to answer the lift of the hand, whenever this would occur:

```
stop (handles.ai);
set (handles.ai, 'TriggerChannel',handles.ai.channel(1));
set (handles.ai, 'TriggerFcn', {@msh_u, h, handles});
set (handles.ai, 'TriggerCondition', 'Rising');
set (handles.ai, 'TriggerConditionValue', 2);
start (handles.ai).
```

When this event occurs, the function msh_u will be called. Proper action is thus prepared; action is taken according to how the events occur and get chained. To store and retrieve global data, the same structure that contains the GUI component handles has been used. The handles structure is passed to each callback.

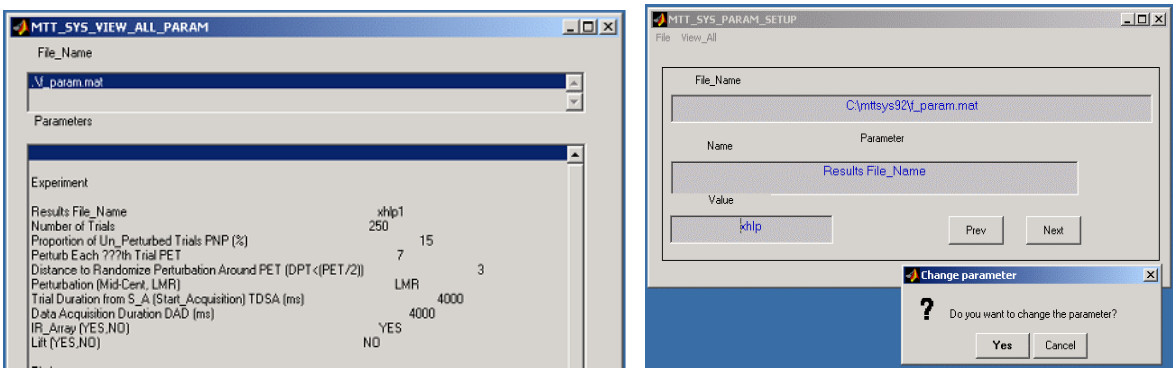

Fig. 4. MTT_SYS: The Parameter List. By changing the values, a new paradigm is easily implemented, with no structural modifications.

Planning the event interaction is done through a parameter list at the operator's disposal (Figure 4). This is a convenient tool to tune the system to a different paradigm with no need for either hardware or software structural changes and may be used during the initialization phase of the system.

3. Results

3.1 MTT_SYS Characteristics

Data acquisition: 3D (x,y,z) - 16 IR markers sampled at max. 200 Hz / frame; optional external analog signals (7 channels, +- 5V range, 12 bit resolution, sampled at 2 kHz / frame).
Event Handling Capability: 3 events, detected in interrupts by threshold detection (1 ms resolution);
Digital Output Signals: 3 digital output lines e.g. to light the 3 dowels in our application;
Output Signals (Perturbation): (i) 1 programmable shape analog output, (ii) 1 digital output (target position jump). The perturbation output signals may be synchronous with any of the events, or delayed (1 ms resolution). Triggering a perturbation by an external event (4^{th} event) is also possible.
Sound Output: via Sound board.

3.2 MTT_SYS – Visualization facilities

Whenever needed during the experiment, any of the previous trials event history may be visualized (Figure 5 – left); thus the chaining of the events and their occurrence may be inspected.
The recording of the data sequentially proceeds trial by trial, until the programmed number of trials was met or the operator decides to conclude the session. The experimental data is converted and saved in ASCII format, which allows a broad spectrum of post processing capabilities, including visualization and processing using common packages as Excel (Microsoft, USA), or dedicated post processing packages.

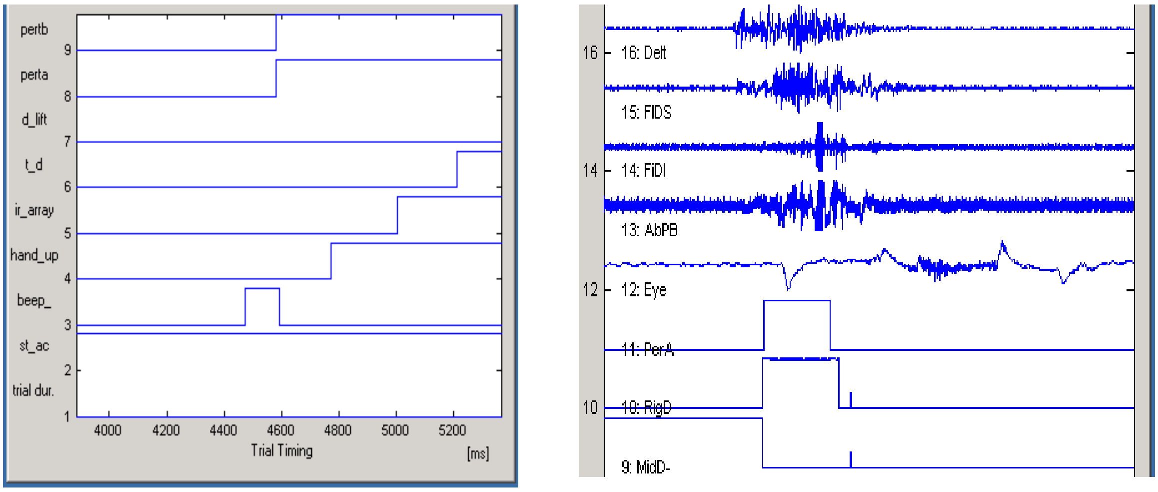

Fig. 5. MTT_SYS. Left panel (typical event frame – real time is shown); right panel [physiological signals (eye EOG, EMG from different muscles) - up & events – down].

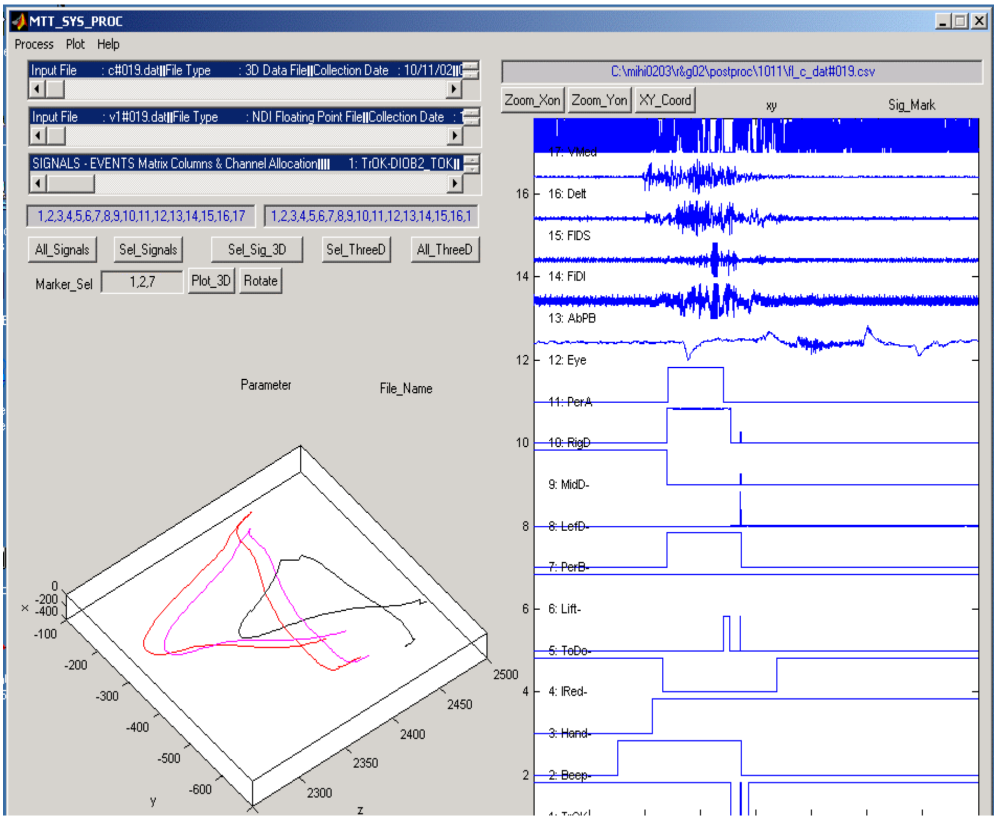

Fig. 6. MTT_SYS_PROC Graphical User Interface (GUI) provides simple menu-driven operation and thorough visualization facilities. Any combination of IR markers and signals can be programmed. Zooming and 3D visualization options are provided. Accurate measurement of XY coordinates on any signal is possible via the mouse. 3D visualization (left) – tips of index (red), thumb (magenta) and wrist (black) in left perturbed trial. Signals (right) – events and physiological signals.

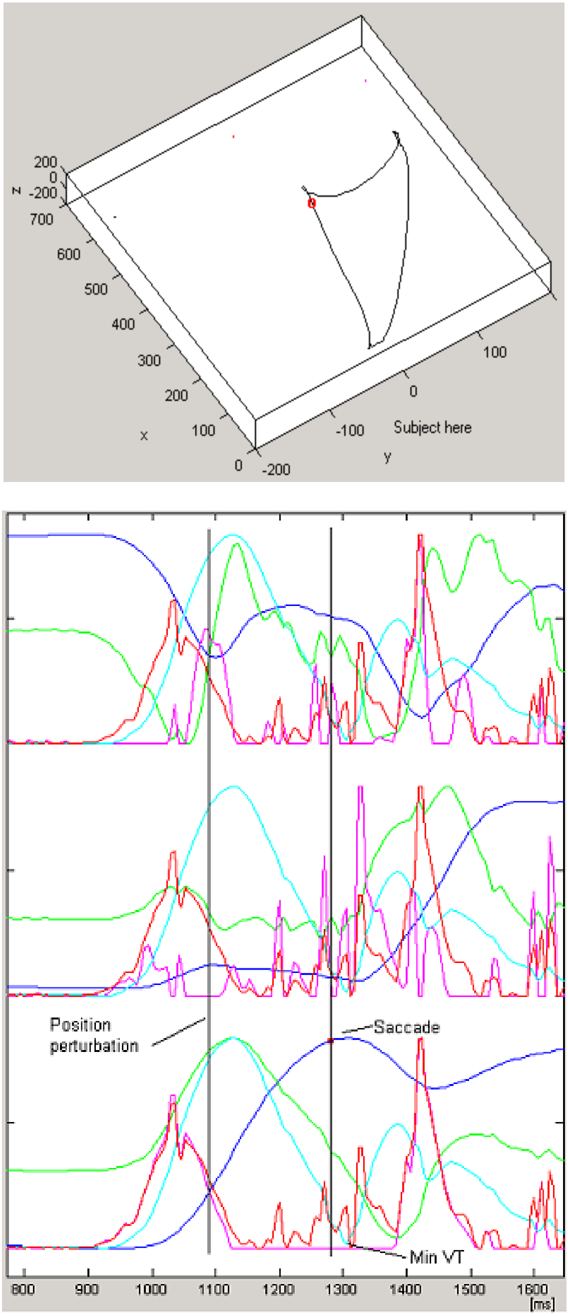

Fig. 7. MTT_SYS_PROC Graphical User Interface (GUI). Upper panel: Detailed visualization of the wrist trajectory in space in right perturbed trial – the saccade occurrence is shown with red circle; Lower panel: x (low), y (middle), z (up) coordinates in time (blue); composite tangential velocity (cyan); individual velocities (green); composite tangential acceleration (red); individual accelerations (magenta). Values are normalized. The occurrence of the perturbation (jump of the target) and of the corrective saccade are shown.

To complement the application, an original post-processing module (MTT_SYS_PROC) has also been developed under Windows to insure the management, visualization and processing of the acquired data (Figure 6). This module allows any combination of IR markers and signals or events to be selected for visualization and has provision for high level processing. Zooming and 3D visualization options are provided (Zoom_Xon, Zoom_Yon). Accurate measurement of X,Y coordinates on any signal is possible via the mouse (XY_Coord): X coordinate on signals gives the time. 3D visualization of the trajectories of different markers during movement is possible, with rotation facilities (Figure6, left panel).

To show the high level processing provision of MTT_SYS_PROC module, figure 7 details its use for the visualization of the wrist trajectory in space (upper panel) and in time (lower panel) - x (low), y (middle), z (up) coordinates in time (blue), together with composite tangential velocity (cyan) and individual velocities (green), and composite tangential acceleration (red) and individual accelerations (magenta) as well.

3. Conclusions

The paper offers an insight into addressing complex time/space measurements in biology and efficient implementations of such complex systems under Windows (Microsoft, USA) and into the use of high level programming environments to build event management and data acquisition systems using the MATLAB Data Acquisition Toolbox (MathWorks Inc., USA). The approach is illustrated with authors' original system (MTT_SYS) to flexibly implement complex experimental paradigms in Biomechanics, Physiology and Neuroscience. MTT_SYS gathers non-invasively 3D (position in space) & physiological information from human or animal subjects, driven by external or physiological events.

MTT_SYS system:

- (i) is a Windows oriented, real time, effective, modern, flexible tool to assist research using double perturbation paradigms in the Reach & Grasp class of experiments;
- (ii) MTT_SYS is useful in other research areas, where experimental paradigms require sequences of time events and triggers;
- (iii) MTT_SYS has structural and operational provision for flexibility to easily address other experimental needs where time/space accurate measurements are required.

Handling hardware resources (signal acquisition structures) in interrupts to allow real time management from high-level programming minimizes the cost of development, therefore such structures are cost effective.

Comparing to other approaches, this implementation provides the quickest way to build high performance, flexible data acquisition systems to implement complex experimental paradigms in Biology and other research fields.

References

[1] T Iberall, CL MacKenzie. Oposition Space and Human Prehension. In: S.T. Venkataraman and T. Iberall (Eds.), Dextrous Robot Hands, NY: Springer Verlag, 1990

[2] G Stelmach, U Castiello. Functional Force Control in Parkinson’s Disease Patients. In: G.E. Stelmach, J. Requin (Eds.), Tutorials in Motor Behavior, North Holland Elsevier, Amsterdam, 1992

[3] LS Jakobson, MA Goodale. Factors affecting higher-order movement planning: a kinematic analysis of human prehension, *Exp Brain Res* 86(1991): 199-208

[4] Y Paulignan, C MacKenzie, R Marteniuk, M Jeannerod. Selective perturbation of visual input during prehension movements. The effects of changing object size, *Exp. Brain Res* 87(1991): 407-420

[5] H Grea, M Desmurget, C Prablanc, Postural invariance in three-dimensional reaching and grasping movements, *Exp Brain Res* 134(2) (2001): 155-162

[6] M Jeannerod, J Decety, F Michel, Impairment of Grasping Movements Following a Bilateral Posterior Parietal Lesion, *Neuropsychologia*, 32(4) (1994): 369-380

[7] JD Connoly, MA Goodale. The role of visual feedback of hand position in the control of manual prehension, *Exp Brain Res*, 125(3) (1999): 281-286

[8] MT Tarata, James R Bloedel, Insight into the Reach & Grasp, *Proceedings of The 1st MEDINF International Conference on Medical Informatics & Engineering MEDINF' 2003,* October 9 - 11, 2003, Craiova, Romania, Craiova Medicala Journal, ISSN 1454-6876, Vol. 5, Sup. 3, (2003): 59-71

[9] Y Paulignan, M Jeannerod, C MacKenzie, R Marteniuk. Selective perturbation of visual input during prehension movements. The effects of changing object position, *Exp Brain Res* 83(1991): 502-512

[10] U Castiello, K Bennett, H Chambers. Reach to grasp: the response to a simultaneous perturbation of object position and size, *Exp Brain Res* 120(1998): 31-40

[11] M T Tarata, James R Bloedel. MTT_SYS – An Original Real Time Windows Oriented System To Address Complex Experimental Paradigms, *Proceedings of The 1st MEDINF International Conference on Medical Informatics & Engineering*, MEDINF' 2003, October 9 - 11, 2003, Craiova, Romania, Craiova Medicala Journal, ISSN 1454-6876, Vol. 5, Sup. 3, (2003): 429-433

[12] Data Acquisition Toolbox For use with MATLAB, MathWorks Inc., Natick MA, USA, 2001