Real-Time Motion Capture Toolbox (RTMocap): an open-source code for recording 3-D motion kinematics to study action–effect anticipations during motor and social interactions

Daniel Lewkowicz^{1,2} · Yvonne Delevoye-Turrell^{1,2}

Published online: 25 March 2015
© Psychonomic Society, Inc. 2015

Abstract We present here a toolbox for the real-time motion capture of biological movements that runs in the cross-platform MATLAB environment (The MathWorks, Inc., Natick, MA). It provides instantaneous processing of the 3-D movement coordinates of up to 20 markers at a single instant. Available functions include (1) the setting of reference positions, areas, and trajectories of interest; (2) recording of the 3-D coordinates for each marker over the trial duration; and (3) the detection of events to use as triggers for external reinforcers (e.g., lights, sounds, or odors). Through fast online communication between the hardware controller and RTMocap, automatic trial selection is possible by means of either a preset or an adaptive criterion. Rapid preprocessing of signals is also provided, which includes artifact rejection, filtering, spline interpolation, and averaging. A key example is detailed, and three typical variations are developed (1) to provide a clear understanding of the importance of real-time control for 3-D motion in cognitive sciences and (2) to present users with simple lines of code that can be used as starting points for customizing experiments using the simple MATLAB syntax. RTMocap is freely available (<http://sites.google.com/site/RTMocap/>) under the GNU public license for noncommercial use and open-source development, together with sample data and extensive documentation.

Keywords Kinematics · Real-time · Filter · 3-D motion capture · Interaction · MATLAB · Software · Toolbox

The *Real-Time Motion Capture (RTMocap) Toolbox* is a MATLAB toolbox dedicated to the instantaneous control and processing of 3-D motion capture data. It was developed to automatically trigger reinforcement sounds during reach-and-grasp object-related movements, but it is potentially useful in a wide range of other interactive situations—for instance, when directing voluntary movements to places in space (e.g., triggering a light on when the hand reaches a predefined 3-D position in a room), when performing actions toward stationary or moving objects (e.g., triggering a sound when an object is grasped with the correct body posture), or simply as a way to reinforce social interactions (e.g., turning music on when a child looks at a person by moving the head in the proper direction). The RTMocap Toolbox is created from open-source code, distributed under the GPL license, and freely available for download at <http://sites.google.com/site/RTMocap/>.

The RTMocap Toolbox is mainly intended to work with recordings made with an infrared marker-based optical motion capture system. Such motion capture systems are based on an actively emitting source that pulses infrared light at a very high frequency, which is then reflected by small, usually circular markers attached to the tracked body parts and objects. With each camera capturing the position of the reflective markers in two dimensions (Fig. 1), a network of several cameras can be used to obtain position data in 3-D. The RTMocap Toolbox was developed with the Qualysis motion capture system, but it can be adapted, by means of slight changes in code lines, to other 3-D motion capture systems that provide real-time information in the MATLAB environment—for instance, OptoTrack,

✉ Daniel Lewkowicz
daniel.lewkowicz@gmail.com

¹ SCALab, UMR CNRS 9193, 59653 Villeneuve d'Ascq Cedex, France

² Univ Lille, BP 60149, 59653 Villeneuve d'Ascq Cedex, France

Fig. 1 Illustration describing the experimental setup, with a schematic representation of the data-streaming procedure. Two interacting participants are here equipped with three reflecting markers on the hand (wrist, thumb, and index finger). A single marker codes for the position of the bottle (manipulated object). Three cameras are placed to cover the working area in which the social-interactive task will take place. Streaming at 200 Hz, the RTMocap reads in the 2-D data from the cameras, and

through its rapid data-treating procedure (< 5 ms), it can modify the environmental conditions of the working area: for instance, turn the lights on brighter, trigger a series of rewarding sounds, or even modify the pleasantness of the room through the slight diffusion of orange zest odors. The effects of these environmental changes on the dynamics of the participants can immediately be traced to assess the action-effect models of embodied cognition

Vicon, and others. In the current version, code lines are included for the direct use with Qualysis, Vicon, Optotrak, or Microsoft Kinect hardware.

The novelty of the RTMocap Toolbox is that it is the first open-source code that provides online control of an interactive 3-D motion capture experimental setup. This Toolbox was written for experimental researchers with little experience in MATLAB programming who are interested in taking advantage of motion capture to study human behavior in interactive situations. It offers the possibility of collecting series of files for further data analysis with code flexibility, specifying trial number, and duration with a number of preprocessing and postprocessing preprogrammed functions. Indeed, native motion capture software do not provide the possibility to sort, select and exclude trials automatically on the basis of predefined parameters. It was thus necessary after data collection to handpick out those trials that do not meet selection criterion (e.g., target missed, reaction time, etc.). The flexibility of the building block system allows novelty in the way of constructing experimental protocols with high levels of repeatability while ensuring high quality in the collected data. Moreover, it is also possible to share experimental setups between laboratories that use different 3-D movement capture systems by simply defining a new

hardware at the start of the setup. But most importantly, the RTMocap software can emit sensory stimuli (e.g., auditory noises) that can be delivered automatically following a series of Boolean conditions. This provides the possibility of investigating the importance of direct feedback on the motivational and intentional aspects of motor planning. Finally, RTMocap software allows complementary analysis of 3-D kinematic data to that provided by the native software that comes with most 3-D motion capture systems. In the following pages, we provide details about the easy-to-use MATLAB scripts and functions—that is, the building blocks—that provide the possibility to perform four exemplary experiments that are common in the field of cognitive psychology: (1) reach-to-grasp, (2) sequential actions, (3) reinforcement learning, and (4) joint actions during social interactions.

To get an idea of the potential of the RTMocap software in conducting psychological studies, one can refer to a study published by our group (Quesque et al. 2013) in which we used the software to trigger informative auditory cues to indicate who between two competing players was to pick and place a wooden object. In the present article, we provide both screenshots and exemplary data sets to illustrate the potential data that could be collected using the RTMocap software.

General principles

Our primary objective was to fit within the methodological high standards of hypothesis-driven human behavioral research such as studies conducted in psychophysics or experimental psychology. Thus, in the present article, we provide—when possible—a short theoretical introduction and working hypotheses before describing the methodological and technical aspects of the study. We also challenged our task by keeping the environmental effects as simple and as minimal as possible to ensure a systematic control of independent variables across all subjects. The novelty of our approach is for participants to be able to produce actions and immediately perceive the subsequent environmental effects without the need of the presence of the experimenter through verbal guidance. Such situations have been extensively studied in computer-screen paradigms that provide visual and audio immediate feedback, and the results have revealed their substantial effects for a better understanding of sensorimotor processes and learning dynamics (Farrer, Bouchereau, Jeannerod, & Franck, 2008). Nonetheless, very few contributions have offered the technical solution of *software interactivity* for none computer-screen based paradigms.

Let's consider the concrete case of a person moving a hand towards a cup with the intention to lift to drink from it. In the ideomotor theory of action production (Greenwald, 1970; Prinz, 1997) it has been proposed that agents acquire bidirectional associations between actions and effects when body movements are followed by perceivable changes in the environment ("I know that the cup moved because I touched it"). Once, acquired, these associations can be used to predict the required movement given the anticipation of an intended effect ("If I want that cup to move, I must touch it"). For decades, this hypothesis was tested using one-dimension dependent variables only, such as reaction times and total movement durations, because of the technical challenge of 3-D measurements—for example, the arm kinematics of motion. Very recently, by tracking the activity of the computer mouse and by providing (or not) direct feedback on the screen, it has been possible to show that the presence of predictable changes in the agent's physical environment can indeed shape movement trajectories in a predictive and systematic manner (Pfister, Janczyk, Wirth, Dignath, & Kunde, 2014). Their results showed that the shaping of the trajectories related to compatibility effects are only present when true interactivity is achieved—that is, the action-effects are contingent enough so that the participants "feel" as if the perceived changes in the environment are truly the consequences of their own actions. But more importantly, these findings are in line with a current motor cognition hypothesis (Ansuini, Cavallo, Bertone, & Becchio, 2014; Becchio, Manera, Sartori, Cavallo, & Castiello, 2012) that suggests that during our everyday actions, our *intentions*

in action may be revealed in early *trajectory patterns* of simple movements. During social interaction, external observers would thus infer motor and social intention through the simple observation of the kinematic patterns of body motion (Herbert, Koning, van Uem, & Meulenbroek, 2012; Lewkowicz, Delevoye-Turrell, Bailly, Andry, & Gaussier, 2013; Sartori, Becchio, & Castiello, 2011). But to further test this hypothesis, interactivity for whole body motion is necessary. This was the objective of the RTMocap software development project.

In the following sections, we will describe the recent improvements of motion capture systems that will provide the possibility today to gain a better understanding of the role of action-effect anticipations for optimal whole-body motor and social interactions in true 3-D environments.

A technical challenge

Whereas similar contributions focus primarily on offline motion analysis techniques for full-body biomechanics (Burger & Toiviainen, 2013; Poppe, Van Der Zee, Heylen, & Taylor, 2014; Wieber, Billet, Boissieux, & Pissard-Gibollet 2006) and neuromuscular simulations (Sandholm, Pronost, & Thalmann, 2009), we focus our contribution on the online capturing and recording of motion *events* to control environmental effects while providing in addition fast and reliable offline analyses.

The RTMocap Toolbox was developed as a complement to the native software that accompanies most 3-D camera hardware systems. By streaming the data in simple matrix format, it offers a solution to develop interactive environments for researchers who do not have the help of computer-programming experts. Knowledge in MATLAB programming is required but more for the understanding of the logic of the scripts than for the ability to write a program from scratch. The choice of Qualisys hardware was made because of the capacity of the QTM environment (the Qualisys data collection software) to give automatic identifications of markers (AIM) given a specific body model as reference. This type of identification is important to maintain as stable as possible the data list in a specific order and to avoid marker switching or flickering. After calibration of the 3-D space and the creation of a simple dummy-trial for AIM identification and recognition, the real-time MATLAB plugin of Qualisys is able to provide the identified marker position in less than 3 ms. Studies conducted in our laboratory have suggested that such negligible time-delays are undetected by the participant and thus, this setup is suitable for further online computation with participants experiencing the situation as being truly interactive (Lewkowicz & Delevoye-Turrell 2014). Similar automatic high-quality streaming in real time of 3-D data can be achieved using other platforms—for example, Vicon or Optotrak systems—by simply changing the plug-in that streams the raw data to MATLAB. In the following sections, we present the building blocks that allow this flexibility of

hardware use. Experiments using cheaper systems—for example, the MS Kinect—can also be used. However, the accuracy and the timing of the MS Kinect hardware will not allow for experimental setups as interactive as that mentioned above using high quality systems. To be clear, the MS Kinect (more specifically, its first version) does not provide the same standards of quality that are required for scientific research in human behavior and we do not recommend it for studying on-line psychological process in an experimental situation. Nevertheless, the way to implement interactivity remains identical and RTMocap may be used to compare the quality of collected data between hardware systems.

Five building blocks

The idea of block programming is to simplify the content of software coding to the minimal set of tools needed for generalization across a wide range of experimental designs. Through the use of five building blocks (see Fig. 2), we provide a fully customized opportunity to create innovative and interactive experimental paradigms by taking advantage of the real-time processing of motion capture. A limited set of preprocessing tools are provided, corresponding to the minimal prerequisite steps needed for motor kinematic analysis: recording, interpolation, filtering, and the extraction of relevant parameters. Nevertheless, other scripts and routines that are not readily available in the current version of RTMocap can be included in the general architecture of the program, in order to offer even more customized software. In the following paragraphs, we

present the list of the building blocks that are illustrated in Fig. 2 and their associated functions (see Table 1).

Initializing

The aim of the *initializing* block is to design the experiment, and preload the required variables into their initial state before running the experiment. This block is composed of three main functions: RTMocap_audioinit, RTMocap_connect, and RTMocap_design. A description of these functions is provided in Table 1. The experimenter is guided to fully specify the parameters of the experiment. These parameters (e.g., sampling rate, number of markers, number of movements, type of reward, threshold adjustment, capture duration, and inter-trial delay) are integrated into a unique configuration file to avoid entering the parameters manually for each repeated experimental session and thus, ensure reproducibility of the experimental settings from one experimenter to the next. The native software often records these parameters to create a personalized work environment. However, the novelty with our software is that we include these parameter settings to create a *customized interactive environment* with (1) hardware selection, (2) preset criterion to select those good trials from those bad trials to discard, (3) selection of hot zones in which stimuli will be triggered, and (4) informative feedback stimuli that will automatically reinforce subjects without need of additional intervention by the experimenter.

Within this block, we have also included the RTMocap_connect function, which specifies the hardware platform that is going to be used during the experiment. Here, the

Fig. 2 The building blocks of the RTMocap software are presented here within the general software architecture. The simplest of studies may use the first blocks only (e.g., initiating, referencing, and capturing). However,

as soon as interactivity is desired, all of the blocks need to be used. The reward block can be unimodal or multimodal in function, depending on the nature of the interactive experience that one wants to implement

Table 1 Using functions and developing MATLAB scripts for full flexibility

Name	Type	Block	Description
<u>RTMocap_design</u>	Offline	1. Initializing	This function sets all parameters into their initial state and stores then it in a separated file. The parameters are capture frequency, trial duration, number of trials, intertrial interval, number of markers to be captured, speed threshold (if required), number of pauses, and the marker table (specifying, which markers must be verified and at which reference positions).
<u>RTMocap_audioinit</u>	Online	1. Initializing	This function will initialize the sounds that will be used during the experiment and will preload them in the live memory of the computer.
<u>RTMocap_connect</u>	Online	1. Initializing	This function will select the correct plug-in for various hardware type (Qualisys, Vicon, Optotrak, Kinect, etc..)
<u>RTMocap_pointref</u>	Online	2. Referencing	This function is used to specify the initial and terminal positions of a target marker for a given sequence. A participant is thus required to maintain immobility for a few seconds for good reference determination.
<u>RTMocap_trajref</u>	Online	2. Referencing	This function is used to specify a reference trajectory instead of a single point.
<u>RTMocap_pauseref</u>	Online	2. Referencing	This function is used to specify a position where the pause constraint should be imposed for a given sequence.
<u>RTMocap_capture</u>	Online	3. Capturing	This function will launch the recordings for a given duration. The trial will automatically be saved in text file noted as <i>subject code_rawdata_trial nb.txt</i> .
<u>RTMocap_smoothing</u>	Offline	4. Processing	This function offers a filtering procedure for the collected raw data. A dual-pass 4th-order low-pass Butterworth filter is recommended.
<u>RTMocap_interp</u>	Offline	4. Processing	This function offers an interpolation procedure that is required when the tracking of one or many markers is briefly interrupted. We recommend using spline interpolation on each axis (<i>X</i> , <i>Y</i> , and <i>Z</i>) for real-time purposes and avoiding interpolation of gaps that are larger than 20 frames.
<u>RTMocap_3Ddist</u>	Offline	4. Processing	Compute Euclidean distance in 3-D space. The same function can be used to compute distances between two points or distances between two mobile markers
<u>RTMocap_3Dvel</u>	Offline	4. Processing	Compute the instantaneous velocity of a marker.
<u>RTMocap_3Devent</u>	Offline	4. Processing	Returns the time code of the beginning of a motion event according to a spatiotemporal threshold (generally used for reaction times).
<u>RTMocap_read</u>	Offline	4. Processing	Reads a previously recorded file and fetch the <i>N</i> -by-3-by- <i>M</i> matrix, with <i>N</i> being the number of samples and <i>M</i> the number of markers.
<u>RTMocap_kinematics</u>	Offline	4. Processing	Extract kinematic parameters and returns a structure with the kinematic parameters of <i>motions events</i> (time of start and stop, amplitude of peak velocity, time to peak velocity, movement time, deceleration percentage, amplitude, and time to peak height).
<u>RTMocap_display</u>	Offline	5. Rewarding	This function offers a graphic representation of the executed sequence.
<u>RTMocap_reward</u>	Online	5. Rewarding	This function is used to play either an engaging or an artificial sound at the end of a trial.
<u>RTMocap_adjust</u>	Offline	5. Rewarding	This function is used to adapt the current speed threshold according to the last trial's performance and the last threshold level in an adaptive staircase procedure.

Certain functions can be run without camera systems (*offline* functions), but others need the hardware camera system to be running with an active participant in order to be tested (*online* functions)

script will simply prompt the experimenter to provide the name of the system that is to be used to stream the data into the MATLAB application. At the moment, RTMocap can be used with the following hardware systems: Quaslysis, Vicon, Optotrak, and MS Kinect.

Referencing

The aim of the *referencing* block is to specify the spatial positions that will be used to trigger the different events of the experiment. Because the aim of the toolbox is to remove all verbal instructions between trials, it is necessary to set the spatial positions before running the experiment. These spatial

positions are for example the initial and terminal positions of a simple hand movement. The main functions in this block are RTMocap_pointref and RTMocap_pauseref. One important point to verify here is the variability range of the static reference positions. We recommend that the variability of the spatial position should not exceed 20 mm/s during referencing. It is also possible to use a full trajectory as a reference with the RTMocap_trajref function.

Capturing

Once the references have been set, the capture can start. This block overrides in part the native software, as its goal is data

collection and storing. The 3-D data is aggregated under MATLAB as a three dimensional matrix. For each marker, an N -by-3 array of data is collected, with N being the total number of samples along the three axes (X , Y , and Z). Each array is then concatenated as to have the third dimension of the matrix coding for the marker ordinal number (see Fig. 3). Here, RTMocap takes advantage of high-quality cameras (available, e.g., with the Qualisys and Vicon systems) that integrate an automatic pause after each data point recording, in order to control for interframe time delays. In practical terms, at 200 Hz, the interframe delay should be 5 ms, even if the cameras are able to compute with an interval threshold of 3 ms. Hence, when MATLAB makes a request for a new sample, the native software waits the prespecified time interval before data capturing. With such small time delays, capturing in real time can be done up to sampling rates of 200 Hz without significant jittering—that is, without undesirable variability of the durations between frames (see Nymoen, Voldsund, Skogstad, Jensenius, & Torresen, 2012, for comparison of Qualisys vs. an iPod Touch motion capture systems). In theory, higher real-time frequencies may be reached (250–300 Hz), but at the cost of higher jittering, so we do not

recommend it at the time. It is also very important that the computation process within the recording loops does not exceed 80% to 90% of the maximum time interval tolerated through the sampling rate (e.g., for sampling at 200 Hz, computing loops should not exceed 4 to 4.5 ms). With this in mind, the present version of the RTMocap software is designed to maintain the amount of computation to its strict minimum such as collecting extra data coordinates, time stamping and mean calculations. For example, in the case of online computations such as spatial verifications within a sequence (see Example 2), care was taken to use the fastest form of 3-D Euclidean distance computation provided under MATLAB (e.g., RTMocap_3Ddist, which offers 3-D computations in less than 1 ms). When using slow systems (e.g., the MS Kinect), capturing in real time cannot exceed sampling rates of 30 Hz.

Processing

The *processing* block includes basic operations that are the most common techniques used in motion tracking, which can both be used in online (experimental) or offline (analysis) scripts. First, a *motion detector* is provided by the RTMocap_3Devent function, which is a backward detection of the beginning of a movement; it returns the time-code for an event. Similar to the other functions, it can be customized by modifying the threshold applied for the detection of the start of a movement (e.g., when computing reaction times or a moving finger). Second, the 3-D Euclidean distance can be computed between a marker and a static point or between two moving markers (e.g., index and thumb) when using the RTMocap_3Ddist function. The result of this distance can be used as a Boolean condition to trigger an event in the experimental protocol. Third, the RTMocap_3Dvel computes the instantaneous tangential velocity of a marker by simply dividing the travelled distance in 3-D space by the duration between two successive optical frames. This duration must be constant or interpolated to constant; otherwise, the instantaneous velocity value will be incorrect. Forth, the RTMocap_kinematics function detects the number of movement events (called “motor elements”) and returns a series of kinematic parameters related to each of these events (time of start and stop, amplitude of peak velocity, time to peak velocity, movement time, deceleration percentage, amplitude of peak height, and time to peak height). The results of this function can be used for the purpose of selecting trials in an online procedure (see Fig. 6 below) or in a systematic offline analysis for statistical analyses—for example, when comparing multiple conditions. One should take the time to try each one of these processing functions independently in order to get a feel of how they operate.

The processing block also includes a low-pass filter method (RTMocap_smoothing) to remove high-frequency noise.

X	Y	Z
495.49	87.755	20.474
494.65	86.481	18.822
493.8	84.885	17.191
493.27	83.309	15.831
492.75	81.618	14.458
492.27	79.73	13.273
491.84	77.759	12.152
491.59	75.677	11.342
491.35	73.502	10.508
491.21	71.095	9.9236
490.34	68.149	9.3046
490.58	65.626	9.2021
490.81	62.99	8.8427
491.29	60.533	8.756
493.68	58.401	7.2362
494.49	55.712	6.7512
495.54	52.975	6.31
496.8	50.252	5.4222
498.29	47.557	4.433
499.98	45.009	3.2586
502.03	42.62	2.0125
502.01	39.76	2.1056
504.26	37.557	0.33868
506.54	35.854	-1.1425
507.6	35.281	-2.2285
507.76	35.433	-2.5121
508.29	34.874	-3.0319
508.67	34.548	-3.2368
509.03	34.262	-3.4872
509.28	34.177	-3.5593
511.67	34.493	-4.8257

Fig. 3 Illustration of XYZ matrices that are collected using the capturing block

We recommend using a dual pass 4th-order Butterworth to maximize noise attenuation and minimize signal distortion in the time domain (especially if you are planning to double differentiate and use acceleration patterns). The optimal cutoff frequency can be manually chosen using the residual analysis embedded in `RTMocap_smoothing` function. For that, one simply needs to call the function on a previously recorded `Raw_data` file without entering the cutoff frequency as an input parameter. This will cause display a graph of the signal distortion as a function of the cutoff frequency (see Fig. 4). After selecting the best cutoff parameter that maximizes noise reduction (smallest frequency) and minimizes signal distortion (smallest residuals), one can enter the determined cutoff frequency within the `RTMocap_design` function so that the value will be used systematically for all future recordings. Determination of the optimal cutoff frequency should be reconsidered for different body parts (Winter, 2009).

The processing block also helps manage the recording biases that can hinder the validity of 3-D motion capture recording. More specifically, it addresses (1) an abnormal delay between two frames, mostly due to unexpected computation delays within the capture loop—this can happen at any moment depending on the operating system of the computer and (2) missing frames due to possible occlusions of the optical sensors. Indeed, with any optical motion capture based on infrared limb and body movements are associated to occlusions between camera and markers resulting in small gaps in the recorded data. The improvement of camera calibration can minimize the occurrence of such occlusions by insuring that each marker is tracked by at least three cameras at any point in time. Nonetheless, missing frames of small dimensions often

occur and this can be compensated for by using software cubic spline interpolation (gap-fill), and extrapolation for data points outside the range of collected data. Timestamp verification is also required to check for abnormal delays between two frames. In the present package, this procedure is provided by the `RTMocap_interp` function. When used offline, these functions can be run on batched files automatically. When used online, they are run instantaneously after the recording of the last data point of the trial in order to filter, interpolate and perform any other processing function that was preprogrammed before the start of the experiment. The obtained results are then used to perform future actions—for instance, rejecting the trial or triggering reward.

Rewarding

The main purpose of this block is to provide full control of the desired environmental modifications as a consequence of the participants' movements. The `RTMocap_reward` function provides the opportunity to trigger any type of sensory stimulus, from the most basic artificial sound to a complex series of visual, auditory and odorant signals for—for instance, video gaming. Obviously, the function can trigger the reward system differentially depending on preset criterion and thus, be used during social games to code specifically—for example, competitive and cooperative situations. The `RTMocap_display` provides feedback to the experimenter concerning the participants' performance levels by plotting automatically at the end of every trial the velocity profile with color-coded indications of the target speed and/or accuracy criteria that were preset before the beginning of the sequence. The `RTMocap_adjust` function provides an adaptive staircase procedure to modulate speed and accuracy thresholds following a specific set of rules that are modulated in function of the participants' performance levels.

The session can be conducted without intervention by the experimenter as verbal instructions are not required. Indeed, the real-time controllers can filter out bad trials by detecting—for example, anticipatory reactions (negative reaction time), hesitations (multiple velocity peaks), unexpected arm trajectories, or hand terminal positions. The `RTMocap_reward` function can thus be used to select automatically what trials should be rejected within an experimental block. Furthermore, different sounds can be triggered in function of the nature of the detected error in order to provide participants with immediate informative feedback on performance. Finally, by changing a unique line code in the MATLAB script, it is possible to adjust whether one wants to control a block design to have a *set number of correct trials* across conditions or a *set total number of trials* per participant.

In conclusion, the objective here was to develop a tool kit that would contain the minimum number of functions required for data analysis to ensure the extraction, processing, and

Fig. 4 Illustration of the procedure proposed in order to determine the optimal cutoff frequency for the low-pass filter, using the estimation method of Challis (1999). If no cutoff frequency has been entered as an input parameter for the `RTMocap_smooth` function, the user is prompted to manually select the cutoff frequency just after display of this window. In the present example, a cutoff threshold of 15 Hz should be chosen

characterization of human motor kinematics in real time. The novelty is the creation of a truly interactive, personalized environment that provides the key to transform any laboratory-based setup with systematic and accurate measurements into an interactive experience. The second novelty is the existence of a rewarding block that provides for sensory signals being triggered online as a function of the performance of a natural voluntary action, allowing researchers to further explore the cognitive process of action–effect relationships in 3-D space. This in turn provides key features to accurately study the motor components of social interactions during competitive or collaborative joint actions, which can be observed during more natural social games. Because the scripts can be used, combined, and implemented as is, we provide a simple and flexible tool for nonexpert users of MATLAB programming. In the following paragraphs, we describe a set of clear examples of how the toolbox can be used to advance our knowledge in various fields of the cognitive sciences.

Four key examples using RTMocap

For each of the following examples, we express the scientific question to set the general context. Then, we describe briefly the methods and the instructions given to the participants. The procedure will be rather exhaustive for the first example to accompany the potential RTMocap users through the different methodological steps (see the Steps to Follow section below). Then, three variations around the first example will be presented in a much briefer format in order to set emphasis on how real-time can be exploited to engage innovating research in cognitive sciences.

Reading motor intention in a reach-to-grasp task

Field of application: Movement psychology

Name of experiment: “Reach to grasp”

Required functions: `audioinit`, `pointref`, `smoothing`, `3Ddist`, `3Dvel`, `display`

Scientific background The information theory of human cognition claims that the amount of time needed to prepare and trigger a motor response is indicative of the complexity of brain processing required to extract stimulus information and proceed to decision making. Because these cognitive processes operate on symbols that are abstracted from sensorimotor components, cognition has been studied through button-press tasks without thought taken on the nature of the motor response itself. However, in recent years, the embodied theory of cognition claims that most cognitive functions are grounded in and shaped by sensory-motor aspects of the body in movement (Barsalou, 2008). Hence, cognitive functions cannot be fully examined without looking at the way the body

operates during cognitive operations. For those reasons, the RTMocap toolbox may be a key feature for future studies. Indeed, through its real time control and interactivity, our toolbox software will provide the opportunity to study cognition with complex whole body movements by detecting changes in body posture and/or shape of limb trajectory as an ecological response coding for a specific cognitive decision. More specifically, instead of selecting between two response buttons, the participants can be instructed to select between two terminal positions, or as it is the case in our first example, to select between two different ways of grasping a unique object.

Instructions “Start with the hand placed within a target zone and reach for a cup placed at 30 cm in front of you, at the sound of a go signal. In function of the tonality of this go signal, you are required to either grasp the cup with a precision grip, i.e., by the handle (high pitch), or with a whole-hand grip, i.e., from the top (low pitch). At the end of each trial, you will receive an auditory reward indicating whether the trial was performed correctly or not.”

Examples of collected data When performing this task, participants start the trial with their hand immobile. Their first task is to quickly detect the pitch of the auditory tone and decide which action to perform as quickly as possible. In Fig. 5, screenshots and examples of the arm kinematics of a typical trial are displayed, with the resulting feedback that the experimenter receives online, immediately after the completion of the trial. Because of this automatic and immediate processing of data, the experimenter remains silent and can concentrate on the participants’ correct performance of the task by visualizing key features (e.g., velocity, terminal position, movement time).

Key features Using RTMocap, it was possible here to use complex hand posture as a response rather than basic right/left buttonpresses. Thus, real-time encoding of hand position opens the way to the use of more ecological tasks in laboratory set studies. As an extension, by encoding hand trajectory rather than hand terminal position, it would have been possible here to evaluate distance from reference trajectories rather than reference positions. This would enable, for example, the triggering in real time of a sound (e.g., as negative reinforcement) if early on during the hand trajectory significant deviations of the current from the reference trajectory were detected. In place of triggering a sound, one could also imagine switching off the lights, emitting a vibration or simply arresting data collection. Because of the flexibility of block coding, this change in referencing is simple, since it requires using the `RTMocap_trajref` from the RTMocap toolbox in place of the `RTMocap_pointref` function. In such case, a simple modification of a few lines within the `RTMocap_capture` function only would be necessary to

Fig. 5 Typical trial of a reach-to-grasp task. Trajectory patterns are presented on the left, with markers coding for the movements of the object (1–3) and of the index finger (4), thumb (5), and wrist (6). The filter

call up the `RTMocap_3Ddist` function and thus, verify *within the loop* the current trajectory and compare it to the reference trajectory—with a custom threshold that would code for the rejection decision. Because the reference trajectory is specified both in time and in space, one should not use too small a distance threshold to constrain the trajectory. Empirical testing is required for each specific experiment to better understand how these time and space constraints may influence other motor parameters and/or related cognitive functions.

Introducing timing uncertainty within sequential motor actions

Field of application: Cognitive psychology

Name of experiment: “Sequential actions”

Required functions: `audioinit`, `pointref`, `pauseref`, `smoothing`, `3Devent`, `3Ddist`, `3Dvel`, `display`.

Scientific objectives When performing actions in our daily activities, the notion of motor timing is central. Movements must be performed in a coordinated fashion thus imposing on the system the problem of controlling the sequencing of each motor element within a complex action in function of the passage of time. It has been reported that accurate estimation and execution of time intervals—in the order of several seconds—require sustained focusing of attention and/or memory (Delevoeye-Turrell,

function of the processing block was applied (15 Hz) on the trial before displaying the subject’s performance using the `RTMocap_display` function

Thomas, & Giersch, 2006; Fortin, Rousseau, Bourque, & Kirouac, 1993; Zakay & Block, 1996). Nevertheless, the experimental setups that can be used to study the motor timing during object manipulation, or even during more complex motor tasks, remain limited (Repp, 2005). In the following, we describe how the use of the RTMocap software may not only open up new perspectives in the understanding of the role of predictive timing in motor psychology, but also suggest new paradigms offering the possibility to study the role of more basic cognitive functions, such as attention and memory for the rhythmic control of complex motor actions.

Instructions “Start with the hand placed within a target zone and reach for the object placed 30 cm in front of you, at the sound of a go signal. Then, move the object in order to place it on the place pad to your right. In function of the tonality of this go signal, you are required to either grasp the object and place it without interruption during the sequence or to grasp it, mark a pause, wait for the second go signal before placing the object on the place pad. At the end of each trial, you will receive an auditory reward indicating whether the trial was performed correctly or not.”

Examples of collected data When performing this task, participants need to code for the timing aspect of the motor

sequence. In function of the available information about the predictability of task outcome, we have shown that this type of timing manipulation can lead to different modes of motor planning, inducing specific changes in motor kinematics (Lewkowicz & Delevoeye-Turrell 2014). On a more technical note, the use of the RTMocap software provides the possibility to control for degree of predictability of the timing constraints and thus, gain a better understanding of the central role of predictive timing for the fluent planning and execution of motor sequences. Screenshots and examples of arm kinematics for such a task are displayed in Fig. 6 for a trial with a pause (bottom) or no pause (top). For each subelement, the experimenter receives immediate information about how the movement was performed and can thus verify that the participants followed instruction.

Key features With the real-time control of how the participants perform the first subelement of a sequence, it is possible to code multiple variations of the present paradigm. During the pause, it is possible, for instance, to implement congruency

manipulations to reproduce the Posner effects, modify the time constraints implicitly in function of the participants' expectancy, or introduce abrupt onsets to assess the effects of different types of attention on specific phases of motor planning/execution. Finally, by selectively adding Boolean conditions related to two or more different markers simultaneously, one could track independently the two limbs in order to run studies on the cognitive functions required for two-hand manipulative actions through time and space.

Reinforcement learning

Field of application: Child psychology

Name of experiment: “Reinforcement learning”

Required functions: audioint, pointref, smoothing, 3Devent, 3Ddist, 3Dvel, kinematics, display, adjust

Scientific objectives *Reinforcement learning* is the process of learning what to do—that is, how to map situations to actions so as to maximize the reception of a numerical reward signal.

Fig. 6 Typical trial of a pick-and-place task, with variable timing constraints between sequenced subelements. Trajectory patterns are presented on the left, with markers coding for the movements of the object (1–3) and of the index finger(4), thumb(5), and wrist(6). We present here the contrasts obtained when comparing conditions with no pause (a–b) and

with a pause (c–d). The pause was automatically introduced by the RTMocap software when the thumb contacted the object (end of the first reach segment). The characteristics of the pauses were preprogrammed by the experimenter within the initializing block before the start of the experimental session

Thus, the participant is not told which actions to take, as in most protocols of learning psychology, but instead must discover which actions yield the most reward by trying them out (Sutton & Barto, 1998). In child psychology, many studies have been interested in the value of reinforcers, seeking to distinguish the effects of positive and negative reinforcements (Staddon, 1983; Terry, 2000). Because of the need to code for online responses, most studies provide verbal reward given by an experimenter (*guidance*). Those who have tried to develop automatic reward systems have largely used keypresses (in animal studies) or tactile screens (in human studies), but this conditions the task to require simple responses in 2-D space. Recently, online software has been developed using eyetracking systems, which have been important to reveal the role of variability in the shaping of human saccadic systems (Madelain, Paeye, & Darcheville, 2011). In the case of whole-body motion, such studies have not been possible yet, because of the limited access to online coding of 3-D trajectories. In the present example, we will show how it is possible to modulate action–perception associations by manipulating space–time thresholds following an adaptive staircase procedure.

By implementing different types of sensory feedbacks, it is possible without verbal instructions to guide the learning phases of a novel task. For example, by rewarding only when the acceleration phase of a movement is increased, we can induce significant changes in the kinematic profile of a participant while maintaining the terminal spatial accuracy of the task (unpublished data). Such “shaping” is difficult to implement without online measurements of 3-D kinematics, because visual inspection of motor performance is not accurate enough. Here, the learning phase follows a trial-to-trial adaption and can be optimized through the use of engaging sounds or specific types of rewards (see RTMocap website to discover the sounds). For most interesting and challenging cases, it may even be possible to code for actions that affect not only the immediate reward but also the next situation and, through that, all subsequent rewards. These adaptive procedures are classically implemented in behavioral studies assessing reinforcement learning and motivation models, but can be applied to the study of abnormal patterns of behavior in pathological populations as well as in serious gaming—a new field of psychology that is faced with high technological challenges. The RTMocap software can provide the possibility to perform all such experiments in 3-D space.

Instructions “Start with the hand placed within a target zone. At the sound of a go signal, reach for the object that is placed 30 cm in front of you. You are required to grasp the object and place it as fast as possible within the spatial boundaries of the place pad. At the end of each trial, you will receive an auditory reward indicating whether the trial was performed correctly or not.”

Examples of collected data When performing this task, participants do not know that specific constraints are set on hand trajectories (e.g., reaction time, height of the trajectory, and terminal accuracy of the object placing). The idea here is to assess how many trials would be required for a participant to adapt to implicit constraints and whether, for instance, the personality traits of an individual will affect resistance to adaptation. This dynamic example of the use of RTMocap is difficult to illustrate in a static figure. Videos are available on the website associated with the RTMocap tutorial.

Key features Through the control of the reward system and the adaptive threshold possibility, the RTMocap toolbox makes it easier to study motor learning and trial-to-trial adaptations in a true interactive environment. Here, no visual cues of the required spatial accuracy are available. Participants learn implicitly to increase spatial constraints in function of the absence/presence of reward (see video). The algorithm of the threshold adaptation can be modified in the RTMocap_adjust function, and the reward can be fully customized in terms of type, occurrence, and strength within the RTMocap_reward function. Hence, this “shaping” procedure can be generalized to many types of reinforcement-learning studies, from the simplest situations (constrained in 2-D space) to more complex ones (constrained in 3-D space within a predefined time window after the start of a trial). Such software can thus be used to shape 3-D trajectories for pathological individuals during motor rehabilitation, or in children when learning new motor skills requiring specific body spatio-temporal coordination patterns (e.g., writing; see Delevoye-Turrell, 2011).

Introducing the social aspects of motor control

Field of application: Social psychology

Name of experiment: “Joint actions”

Required functions: audioinit, pointref, smoothing, 3Devent, 3Ddist, 3Dvel, kinematics, display, reward, adjust.

Scientific objectives Recent kinematic studies have shown that social context shapes action planning (Becchio, Sartori, & Castiello, 2010; Quesque, Lewkowicz, Delevoye-Turrell, & Coello, 2013). Moreover, we observed in a previous study that humans are able to discriminate between actions performed under different social contexts: reach for a cup for my own use or reach for a cup to give it to someone else (Lewkowicz et al. 2013). Interestingly, more and more conceptual and empirical studies are reporting consistently the need to investigate real-time social encounters in true interactive situations (Schilbach et al. 2013). Nonetheless very few cases of studies have been reported in which two humans have interacted in true 3-D situations—for example, face-to-face interaction around the same table while using real-time

measurements to code for interaction dynamics. Here, we describe an example of a very simple face-to-face interactive paradigm in which two human agents exchange an object while trying to earn the most rewards possible. This software solution was used to run the study that is reported in Quesque et al. (2013); it was also used to test the integration of eyetracking and 3-D movement capture (Lewkowicz & Delevoeye-Turrell, 2014).

Task The go signal for agent A is emitted after a random delay. The go signal for agent B is emitted as soon as the object is correctly placed on the central place pad by agent A (i.e., this is a speed–accuracy task in which space accuracy is the controlled parameter for successful trials).

Instructions for agent A “Start with the hand placed within a target zone and reach for the object placed at 30 cm in front of you; At the sound of a go signal, move the object in order to place it on the place pad located in front of you.”

Instructions for agent B “Start with the hand placed within a target zone and reach for the object placed by agent A on the place pad located at 50 cm in front of you; At the sound of a go signal, move the object to place it on the place pad located on your right.”

Cooperative instructions The goal of both agents is to perform the joint sequence as quickly as possible by maintaining a high accuracy level in space, on both place pads. Both participants receive a joint reward if the full sequence is correctly executed.

Competitive instructions The goal of each agent is to perform their own part of the segment as quickly as possible, regardless of the total time taken for the total sequence. Thus, the objective is to improve the speed of action in reference to one’s own previous performance, following an adaptive staircase procedure. Participants receive individual rewards.

Examples of collected data When performing this task, participants are prepared to react as fast as possible (if in competition mode) or as adequately as possible (if in cooperation mode). In Fig. 7, screenshots and examples of arm kinematics of a typical cooperation trial are displayed, with the resulting feedback that the experimenter receives online, immediately after the completion of the trial. The experimenter determines within the initializing block whether the feedback is to be displayed on agent A (top) or B (bottom). One can note the effects of displacement direction on the key features of reaction time, peak wrist height, and peak velocity as a function of displacement direction.

Key features By measuring online velocity peaks, it is possible here to control online whether participants are truly performing the task as quickly as possible (in reference to the maximum velocity of each participant, measured in a series of familiarization trials collected before the experimental session). It is also possible to customize the reward function, and thus to alter the degree of interactivity in order to assess possible changes in the motor coordination patterns between two agents (mirroring of hand trajectories, for example). In our example, we chose two different instructions (competition vs. collaboration), but it would be possible not to give any instructions and to observe how participants implicitly modify their social behavior if their partner’s mistakes impact or not on the obtained reward. As was proposed by Sebanz and Knoblich (2009), prediction in joint action is a key element to fully understand social interactions. As such, the RTMocap toolbox can help build new paradigms through an advanced control of environment variables and their predictability.

The steps to follow

In the following section, we describe the different steps needed to set up and use the RTMocap toolbox. As we mentioned earlier, this toolbox is mainly for scientists that have basic programming experience in MATLAB and good experience in motion capture. This section will guide you to test the first example (the simple reach-to-grasp task) in a step-by-step fashion.

Setting the configuration parameters

In MATLAB, launch `RTMocap_design` to set the configuration parameters of the experiment (see Fig. 1). Note that the intertrial interval (ITI; e.g., 4s) is a single parameter that will be separated into two components: a constant part (ITI/2) and a random part (from ITI/2 s to, e.g., 6 s). This calculation provides two main advantages: (1) It helps maintain the overall ITI close to the parameter that you selected and avoids large discrepancies between trial durations by using a constant part, and (2) it prohibits any anticipatory strategies that participants may use at the beginning of a trial, by using a random part. At the end of this procedure, a `Reach.cfg` file is created to encode and save all of the experimental configuration parameters within a single text file.

Starting the session

Participants are ready to execute the reaching movement, which will offer the possibility to set the initial and terminal constraints. In MATLAB, launch `example_1_reach.m`. The RTMocap Toolbox will execute the function `RTMocap_pointref` twice, to record the starting and ending positions of the marker, which

Fig. 7 Two typical trials of the social interaction paradigm are presented here. Trajectory patterns are presented on the left, with markers coding for agent **a** (1–4) and agent **b** (5–8). Agent A was instructed to place the object in the central place pad; agent B was instructed to place the object on the pad located on either his right (**a–b**) or left (**c–d**) side. In

panel B, we display the kinematic information of participant A's index finger. In panel D, we display kinematic information for participant B's index finger. Black stars indicate the start and the stop positions (3-D plots), and the red stars indicate the peak values that were detected by the RTMocap algorithm

will be tracked in real time. MATLAB will at this time display the following text:

“Place the marker on the initial position” Have subjects place their hand within the starting reference zone. When the experimenter starts recording, the “initial” reference position is calculated as the mean position across a 500-ms time interval. At the end of the recording, the software displays the mean *X*, *Y*, and *Z* coordinates, as well as the standard deviation of the measurement and the total distance traveled. In addition, the software displays a recommendation to repeat the reference if, during the referencing, the traveled distance exceeds 10 mm. At this stage, the experimenter can repeat the reference recording as many times as necessary to reach acceptable stability. When reached, these coordinates are saved for future use as the XYZ references of the “initial” position of the hand.

“Place the marker on the terminal position” Have subjects place their hand stable within the terminal position. A procedure similar to that above is repeated here, to obtain the XYZ

references of the terminal coordinates. This procedure is repeated as many times as there are set conditions (two, for the present example). The software then indicates that the experimenter should explain the instructions to the participants.

Launch the experiment

When the participants are ready, the experimenter presses the space bar to start data collection. For each trial a sound is triggered, indicating to the subjects that they should start the reaching movement. As soon as the fingers contact the object, the auditory feedback indicates whether or not the trial was correctly performed. Optionally, a line code can be added within the script to delay the feedback by a given time interval. Participants are required to come back to the initial position without a prompt to be ready for the start of the next trial. After a random delay, the next trial is triggered. If the participants were not immobile within the starting reference zone at least 500 ms before trial onset, then “fail” feedback is immediately sent, indicating to the participants that the trial cannot be

initiated. This procedure is repeated until the total number of trials is reached.

Ending the session

During the session, the software creates one data file per trial, containing the timestamps and the XYZ coordinates for the total number of selected markers specified in the AIM model (Reach_raw-data1, -data2, -data3 ...). A graph of the velocity of the tracked marker (Reach_velfig1.jpg), a file with the Reach_errorTab.txt is also recorded. This latter file defines for each trial the desired target position and the observed final position of the tracked marker in XYZ coordinates (each trial being presented on a separate line). This last file can be used before complex preprocessing—for example, to verify the terminal variability of finger positions on an object, and thus to calculate the number of correct answers as a function of hand terminal position on the object, without the need of any kind of intervention by the experimenter during the experimental session.

Debriefing the participants

When first setting up the experiment, it is important to question the participants on their ease of use of technology as it has been shown recently that participants who are frequently in contact with interactive systems (smartphones, tablets) are more inclined to be poorly sensitive to interactive environments (Delevoye-Turrell, unpublished data). More importantly, one must test the feeling of interactivity perceived by the participants in order to ensure that the reference preset by the experimenter affords “trust” in the system. If the terminal position variability is too high or the latency of the reinforcement signal is too long, the participants may not “believe” the automatic feedback that they receive. This may lead to unwanted modifications in the participants’ attitudes. Hence, trust in and acceptance of the setup need to be evaluated on an analogue scale to ensure consistency across experimental conditions.

Conclusion

In the present contribution, we have focused mainly on the potential use of real-time coding for motion capture to develop innovative behavioral interactive experiments in the cognitive sciences. But the potential applications range from virtual reality control to improved assessment of the cognitive control of 3-D world technology. Indeed, the RTMocap software can be adapted to various types of 3-D motion capture systems and could, in the near future, be implemented within humanoid robots to code for human kinematics in real time in order to offer intuitive human–robot interactions during collaborative tasks (Andry et al., 2014).

In conclusion, the real-time detection of 3-D body motion offers a multitude of possibilities. We have here described specific examples used for the study of motor cognition and decision-making processes, but a wide range of fields could take advantage of this software to gain better understanding not only of how different contextual variables influence reaction and movement times, but also of the shape and variability of motor planning, as well as the properties of motor execution in ecological and free-willed 3-D interactions.

Author note We thank Laurent Droguet and Laurent Ott for their technical support and useful feedback on previous versions of the software. We also thank the native English speaker for linguistic review of the manuscript. Research was supported by grants from to Y.D.T. from the French Research Agency ANR-2009-CORD-014-INTERACT and ANR-11-EQPX-0023. The work provided by D.L. was also financed by the region Nord-Pas-de-Calais (France). D.L. and Y.D.T. report no competing interests.

References

- Andry, P., Bailly, D., Beausse, N., Blanchard, A., Lewkowicz, D., Alfayad, S., Delevoye-Turrell, Y., Ouezdou, F. B., & Gaussier, P. (2014). Learning Anticipatory Control : A Trace for Intention Recognition. *Papers from the 2014 AAAI Fall Symposium*, pp 29–31.
- Ansuini, C., Cavallo, A., Bertone, C., & Becchio, C. (2014). Intentions in the brain: The unveiling of Mister Hyde. *The Neuroscientist*. doi:10.1177/1073858414533827
- Barsalou, L. W. (2008). Grounded cognition. *Annual Review of Psychology*, 59, 617–645. doi:10.1146/annurev.psych.59.103006.093639
- Becchio, C., Manera, V., Sartori, L., Cavallo, A., & Castiello, U. (2012). Grasping intentions: From thought experiments to empirical evidence. *Frontiers in Human Neuroscience*, 6, 117. doi:10.3389/fnhum.2012.00117
- Becchio, C., Sartori, L., & Castiello, U. (2010). Toward you the social side of actions. *Current Directions in Psychological Science*, 19, 183–188. doi:10.1177/0963721410370131
- Burger, B., & Toiviainen, P. (2013). MoCap Toolbox—A Matlab toolbox for computational analysis of movement data. In R. Bresin (Ed.), *Proceedings of the 10th Sound and Music Computing Conference, (SMC)*. Stockholm, Sweden: KTH Royal Institute of Technology.
- Challis, J. H. (1999). A procedure for the automatic determination of filter cutoff frequency for the processing of biomechanical data. *Journal of Applied Biomechanics*, 15, 303–317.
- Delevoye-Turrell, Y. N. (2011). *Création d'un dispositif interactif pour (re)découvrir le plaisir d'écrire* (National Instruments tutorials). Retrieved from <http://sine.ni.com/cs/app/doc/p/id/cs-14175>
- Delevoye-Turrell, Y. N., Thomas, P., & Giersch, A. (2006). Attention for movement production: Abnormal profiles in schizophrenia. *Schizophrenia Research*, 84, 430–432. doi:10.1016/j.schres.2006.02.013
- Farrer, C., Bouchereau, M., Jeannerod, M., & Franck, N. (2008). Effect of distorted visual feedback on the sense of agency. *Behavioural neurology*, 19, 53–57.
- Fortin, C., Rousseau, R., Bourque, P., & Kirouac, E. (1993). Time estimation and concurrent nontemporal processing: Specific interference from short-term-memory demands. *Perception & Psychophysics*, 53, 536–548. doi:10.3758/BF03205202

- Greenwald, A. G. (1970). A choice reaction time test of ideomotor theory. *Journal of Experimental Psychology*, 86, 20–25. doi:10.1037/h0029960
- Herbort, O., Koning, A., van Uem, J., & Meulenbroek, R. G. J. (2012). The end-state comfort effect facilitates joint action. *Acta Psychologica*, 139, 404–416. doi:10.1016/j.actpsy.2012.01.001
- Lewkowicz, D., & Delevoye-Turrell, Y. (2014, June). *Combining real-time kinematic recordings with dual portable eye-trackers to study social interactions*. Paper presented at Workshop ERIS' 2014: Eye-Tracking, Regard, Interactions et Suppléances Cité des Sciences et de l'Industrie, Paris, France.
- Lewkowicz, D., Delevoye-Turrell, Y., Bailly, D., Andry, P., & Gaussier, P. (2013). Reading motor intention through mental imagery. *Adaptive Behavior*, 21, 315–327. doi:10.1177/1059712313501347
- Madelain, L., Paeye, C., & Darcheville, J.-C. (2011). Operant control of human eye movements. *Behavioural Processes*, 87, 142–148. doi:10.1016/j.beproc.2011.02.009
- Nymoen, K., Voldsund, A., Skogstad, S. A., Jensenius, A. R., & Torresen, J. (2012). Comparing motion data from an iPod Touch to a high-end optical infrared marker-based motion capture system. In *Proceedings of the International Conference on New Interfaces for Musical Expression* (pp.88–91). Ann Arbor, MI: University of Michigan. Retrieved from www.duo.uio.no/bitstream/handle/10852/9094/knNIME2012.pdf?sequence=1
- Pfister, R., Janczyk, M., Wirth, R., Dignath, D., & Kunde, W. (2014). Thinking with portals: Revisiting kinematic cues to intention. *Cognition*, 133, 464–473. doi:10.1016/j.cognition.2014.07.012
- Poppe, R., Van Der Zee, S., Heylen, D. K., & Taylor, P. J. (2014). AMAB: Automated measurement and analysis of body motion. *Behavior Research Methods*, 46, 625–633. doi:10.3758/s13428-013-0398-y
- Prinz, W. (1997). Perception and action planning. *European Journal of Cognitive Psychology*, 9, 129–154. doi:10.1080/713752551
- Quesque, F., Lewkowicz, D., Delevoye-Turrell, Y. N., & Coello, Y. (2013). Effects of social intention on movement kinematics in co-operative actions. *Frontiers in Neurorobotics*, 7, 14. doi:10.3389/fnbot.2013.00014
- Repp, B. H. (2005). Sensorimotor synchronization: A review of the tapping literature. *Psychonomic Bulletin & Review*, 12, 969–992. doi:10.3758/BF03206433
- Sandholm, A., Pronost, N., & Thalmann, D. (2009). MotionLab: A Matlab toolbox for extracting and processing experimental motion capture data for neuromuscular simulations. In N. Magnenat-Thalmann (Ed.), *Modelling the physiological human: Proceedings of the 3D Physiological Human Workshop, 3DPH 2009, Zermatt, Switzerland, November 29–December 2, 2009* (pp.110–124). Berlin, Germany: Springer.
- Sartori, L., Becchio, C., & Castiello, U. (2011). Cues to intention: The role of movement information. *Cognition*, 119, 242–252. doi:10.1016/j.cognition.2011.01.014
- Schilbach, L., Timmermans, B., Reddy, V., Costall, A., Bente, G., Schlicht, T., & Vogeley, K. (2013). Toward a second-person neuroscience. *Behavioral and Brain Sciences*, 36, 393–414. doi:10.1017/S0140525X12000660. disc. 414–462.
- Sebanz, N., & Knoblich, G. (2009). Prediction in joint action: What, when, and where. *Topics in Cognitive Science*, 1, 353–367. doi:10.1111/j.1756-8765.2009.01024.x
- Staddon, J. E. (1983). *Adaptive behaviour and learning*. Cambridge, UK: Cambridge University Press.
- Sutton, R. S., & Barto, A. G. (1998). *Introduction to reinforcement learning*. Cambridge, MA: MIT Press.
- Terry, W. S. (2000). *Learning and memory: Basic principles, processes, and procedures* (2nd ed.). Boston, MA: Allyn & Bacon.
- Wieber, P.-B., Billet, F., Boissieux, L., & Pissard-Gibollet, R. (2006, June). *The HuManS Toolbox, a homogeneous framework for motion capture, analysis and simulation*. Paper presented at the Ninth International Symposium on the 3-D Analysis of Human Movement, Valenciennes, France.
- Winter, D. A. (2009). *Biomechanics and motor control of human movement*. New York, NY: Wiley.
- Zakay, D., & Block, R. A. (1996). The role of attention in time estimation processes. *Advances in Psychology*, 115, 143–164.