



An Electro–Mechanical Input Device for the Manipulation of Virtual Human Motion

Thomas Kronfeld¹  , Guido Brunnett² 

¹Technische Universität Chemnitz, tkro@cs.tu-chemnitz.de

²Technische Universität Chemnitz, brunnett@cs.tu-chemnitz.de

Corresponding author: Thomas Kronfeld, tkro@cs.tu-chemnitz.de

Abstract.

Although virtual human models are frequently used in factory simulations and analyses, there is still no device available that makes adjustments of their movements easy and intuitive. In order to address this concern we present an electro-mechanical input device for the manipulation of motion, which are directly mapped onto a virtual human model in real-time. Moreover our puppet is also able to actively perform a desired motion sequence, thus it is already positioned when the user decides to modify a movement at a specific time frame. Altered servomotors are used to steer the puppet even while the user adjusts the movement of other body parts. In addition the puppet is patterned on the human body shape allowing highly accurate and realistic adjustments of human motion data. The implemented software framework offers the opportunity to integrate our device into other planning software and also provides the user with a basic set of motion capture data.

Keywords: virtual humans, digital factory, human factors, virtual prototyping

DOI: <https://doi.org/10.14733/cadaps.2019.1070-1083>

1 INTRODUCTION

The simulation of human work activity within digital factories is of growing importance because skilled workers are hard to find and the average age of the employees is increasing. However, for such simulations the motions of the virtual workers have to be created, which is a highly time-consuming and as a consequence very expensive process [5].

Motion simulation is based on key frame animation, which means that for every single frame of the simulation a posture of the virtual human model has to exist. Such animations can be generated either via motion capturing or by manually creating each posture [11]. Both methods have their benefits and drawbacks. Recording human motion requires a full-body motion capture system, which is very expensive in terms of acquisition and maintenance. On the other hand, the resulting motion sequences are realistic and smooth.

The variety of available systems can be distinguished by their embedded sensor technology, e.g. visual or non-visual based, or a combination of both. Those solutions differ according to their accuracy, robustness and latency. In addition, every system has certain drawbacks — the limitation of the measurement volume, the susceptibility to ferromagnetism of surrounding materials and occlusion, or the need for physical equivalents of the environment and objects to interact with, to name but a few [15]. Such constraints make it hardly possible to achieve accurate and noise-free motion capture data within a factory environment. Moreover is the main application area for virtual human motion in the context of digital factories the planning phase of such a shop floor, which is characterized by continuous changes of its layout. Since existing motions can only rarely be adapted to new factory scenarios, there is a constant need of recording new motions. Even minor alterations of the factory layout lead to changed requirements concerning the characteristics of the motion to be used.

In a manual motion modification the planner has to construct a posture of the virtual human model for each single key frame that has to be modified. This includes dragging and rotating each joint into the desired position using keyboard and mouse. The need to adjust the high number of degrees of freedom (DOF) of a virtual human model is nearly infeasible for untrained users and may lead to rather unrealistic motions [13]. The designer however has full control of the motion generation and he can alter the motion whenever the environment changes.

In this paper we address this concern and present a novel approach to integrate the advantages of both methods. In order to accomplish this goal, we use a basic dataset of motion capture files, which can then be modified using an intuitive input device. Although existing electro-mechanical devices can be used to create a new pose on a virtual human model [3],[9], appropriate tools for the subsequent motion modification are still missing. Obviously, input devices that allow the modification of single poses only are not efficient to use. There user is forced to produce a huge number of key frame postures in order to create an entire animation sequence — a costly procedure which needs nearly the same effort as a manual motion modification. Rather, a method is needed to modify easily a whole motion sequence.

Our electro-mechanical input device is comparable in structure with the one developed in [14], but in addition allows the modification of motion sequences as a whole instead of only adjusting single postures. This offers many benefits for factory planners. The user is able to modify a given motion sequence in a dynamic and intuitive way. In addition, our puppet has more DOF than other comparable input devices and its overall shape and size is based on the anthropometric dimensions of the human body. As a result of its human-like appearance and active feedback when performing a motion, highly realistic motions can be created. Our solution is capable of both replaying existing motions on the puppet and recording the user's input. For this we developed algorithms for controlling and interacting with the device.

The paper is structured as follows. Section 2 gives an overview of existing human-like input devices for controlling a digital model. We then describe the mechanical structure of our puppet in section 3 and outline the functions of the implemented software in section 4. Section 5 illustrates the functioning of our puppet on the basis of three different use cases. The advantages of our approach are summarized in section 6.

2 RELATED WORK

The objective to animate a digital character has become a main sector in computer graphics. In the context of factory planning a virtual human model represents the worker performing certain working steps, thus the current factory layout can be evaluated. In order to do so realistic and accurate sequences of movements are needed.

As stated above, motion capture systems provide us with such data. Modifying them according to changed requirements however is a difficult task. When sticking solely to the concept of motion capturing it would be necessary to either change the characteristics of the data by hand or to record new movements in an adjusted setting. To overcome this lack of user friendliness an input device is needed which allows the user to modify

existing motion data in a smooth and easy to use manner.

In recent years a number of input devices have been developed which aim to aid their user in handling the complexity of motion data and virtual human models. The approach of a puppet interface in [7] has 17 DOF consisting of 14 rotary potentiometers plus one orientation sensor and is used for retrieving motion capture data from a database. Using this device in order to search for already existing motions that are suitable for a changed factory layout is inappropriate as the puppet's range of feasible motions is highly limited due to the small number of joints.

The authors in [3] present an advancement of the input device developed by [4] based on modular segments such as joints, splitters and extensions. The physical skeleton built with these segments is used to create poses on most diverse digital rigs for the application in character animation. The generated motion sequence however is based on key framing combined with physical inverse kinematics and pose interpolation. In addition to that the global orientation and position of the character is not tracked and has to be adjusted in an additional post-processing step. These limitations and the initial time-consuming assembly of parts could lead to a low user acceptance.

QUMARION is a human-shaped input device distributed by CELSYS Inc. — a joint venture of the University of Tsukuba and the University of Electro-Communications. With its 16 moving joints and 32 built-in sensors [1] it is designed for pose-creating operations on digital models, e.g. in [8]. In [10] QUMARION is used in the context of telerehabilitation where the therapist sends instructions of the desired rehabilitation poses to the patient. The learning support system developed in [5] utilizes this input device in first aid training sessions. Life-saving postures, e.g. the recovery position, can be trained thereby.

Another approach published in [12] focuses on posture-control of a virtual and rigged character within an ergonomic simulation. More precisely, the developed humanoid input device has 22 DOF and is designed to control the virtual human model Jack by Siemens PLM. For the purpose of compensating gravity the user has to set the friction of every joint via two adjustable screws on each of them. The authors assume that movements of the digital model can be generated by creating several poses and setting key frames.

In order to control a virtual human model in product design the authors in [14] present a physical puppet having 32 DOF. A serial interface allows to measure and control the parameters of the servomotors. Changes of the puppet's joint angles are mapped onto the visualization model and vice versa. Thus the design of static postures can be realized. Due to the size of the integrated servomotors, the structure of the puppet and the virtual human model do not match. As a consequence some postures cannot be achieved.

In summary it can be said, that beside their individual limitations all of the aforementioned solutions are restricted to the generation or modification of single postures only. In order to create or edit an animation their user is forced to produce a huge number of single poses for a rather short motion sequence. This process is similar to stop motion, which not only results in an odd animation of the digital human model, it is also highly time-consuming and difficult to manage for an untrained operator. Our approach allows the modification of the whole motion sequence at once. In order to give the user a notion of the dynamics and timing of the processed motion our electro-mechanical puppet actively replays the motion and therefore is not only an input but also as an output device.

3 MECHANICS OF THE PUPPET

While developing the input device, we focused on a simple and intuitive positioning and animation of virtual human models. Analogous to a toy doll the puppet's joints and limbs can be moved and positioned. In order to give the user an active feedback of the motion sequence and its timing the puppet is able to drive each joint autonomously by servomotors. This has the advantage that the user can load and play existing motions on the puppet and modify them in real-time. A first prototype of the developed puppet can be seen in Fig. 1.

In the following subsection we explain our design considerations and the problems we have faced first. After that we outline the used hardware components and controller.

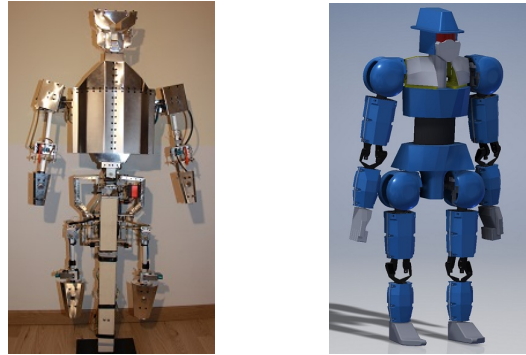


Figure 1: Input device: (a) Puppet prototype without covering, (b) CAD model of the covering.

3.1 Design Considerations

Before building the actual puppet and implementing the software some preliminary considerations had to be addressed leading to the following questions:

1. *Which servomotors have to be used in order to comply with the anthropometric requirements and how to adjust them in order to achieve the desired functionality of the puppet?*

Complex servomotors used in robotics, as applied in [14], are far too big to meet the dimensions of the covering around each joint. Micro and midi sized servomotors with a simpler construction, which are mainly used for remote controlled car models, meet these requirements and still deliver enough power to move the puppet's body parts. One drawback is that their current folding square cannot be readout. Therefore the signal of the potentiometer is tapped and passed as analog signal into the controller.

2. *How to enable the user to control the single joints and limb groups of the puppet?*

Simple servomotors only have an on/off control. While set to on, they hold their current position, which makes a manual adjustment of the puppet impossible. When turned off, the influence of external forces sets in and the joint's position cannot be maintained. To solve this problem interaction buttons are integrated on the puppet. Using those buttons, the corresponding limb groups can be turned off, which makes manual modifications of an entire body part possible. Another method for controlling just a few joints or even a single DOF is to turn them off via the GUI.

3. *How to ensure that external forces have no influence on the joints and the whole joint angle chain when the servomotors are turned off?*

To a certain degree joint stiffness is necessary to counteract external forces, except those induced by the user. Especially when the servomotors are switched off, the puppet must resist gravity and hold its position autonomously. Also, when one joint is moved undesired changes in the adjacent joints' positions may occur due to acceleration force. To overcome this issue a breaking system is integrated at certain joints.

4. *How to aid the user in creating and adjusting motion sequences when it comes to curved walk paths?*

Beside the manipulation of stationary movements, possible curved walking paths have to be taken into account. Instead of defining the exact path onto the ground or realigning certain frames of the sequence by hand, the user can simply turn the puppet on its stand and hence simulate a curved movement. The orientation of the puppet is read via a potentiometer. This rotary mechanism is the only DOF without a servomotor, thus the puppet's rotation cannot be reproduced during playback.

3.2 Hardware Components

We wanted to achieve a realistic appearance of the puppet. Therefore, its relation of size between body and limbs follows the anthropometric dimensions of the male 50th percentile of the male residential population in Germany [2] and is uniformly scaled down to an appropriate size for an input device. This resulted in the puppet's total size of 70 centimeter. Based on this data a CAD model of the puppet — its bearings and covering — was created and generated via 3D printing. The input device is separated into six limb groups, each containing the corresponding joints (Fig. 2).

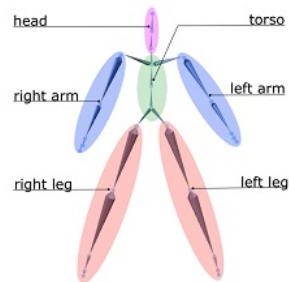


Figure 2: Skeleton and the six limb groups.

Each group's movement can be individually activated and deactivated by the user. This can be done either using the software or by pressing the integrated interaction button on the limbs of the puppet. Deactivated limb groups will not be moved during the playback of motions. As a result, a mechanical puppet, having 35 DOF, was built. A schematic overview of the joints and their DOF is shown in Fig. 3.

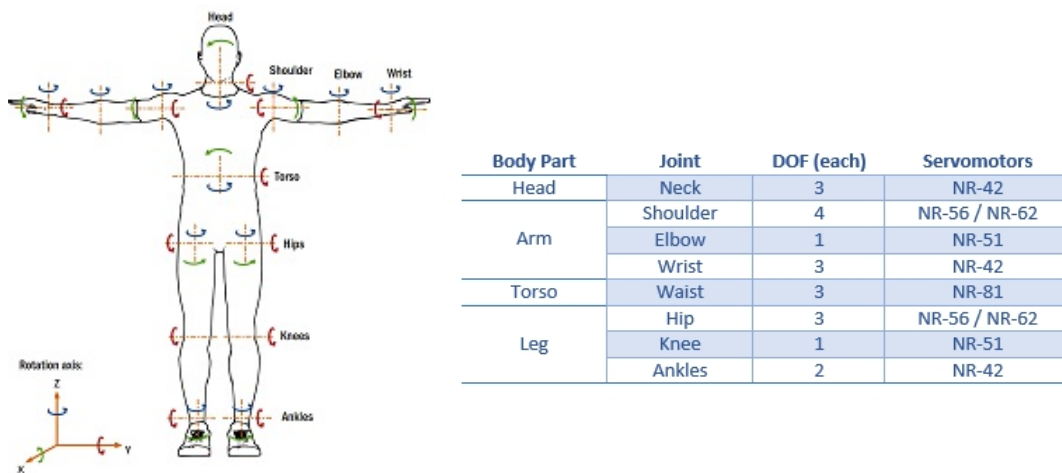


Figure 3: Integrated joints: (a) Kinematic scheme, (b) Number of DOF for each joint.

Each shoulder has four DOF — three for the rotation and one to pull the arm forwards and backwards. Each DOF is controlled by a modified servomotor, which allows to trigger a desired joint angle as well as to read its current position. The servomotors that are used have a rotation range of 180°. A list of the applied servomotors and their specifications is given in Tab. 1.

For joints with a rotation range greater than 180°, a gear ratio has been attached. The range of motion, i.e.

servo type	weight	size	speed	torque
	[g]	[mm]	[sec/60°]	[Nm]
NR-42	3.7	20.1 x 8.5 x 20.7	0.14	0.059
NR-51	5.0	21.4 x 11.6 x 20.5	0.12	0.078
NR-56	6.0	21.2 x 11.4 x 22.6	0.12	0.098
NR-62	8.0	23.5 x 11.6 x 24.5	0.12	0.147
NR-81	16.0	28 x 13.3 x 30.3	0.14	0.265

Table 1: Overview of the applied servomotors and their attributes (all manufactured by Torcster). The speed unit states the time a servomotor needs to rotate the output shaft by 60°.

the joint angle limit, is an important factor in the assessment of human tasks during product and process planning. For example when the engineer has to determine the capability of the worker to reach, grasp, or actuate various controls. In order to only permit realistic motions, end switches are integrated at the individual angle limit to guarantee that they cannot be exceeded. Those angle limits are based on the measured data of joint motion ranges of the Anthropology Research Project of NASA [6]. Another advantage of the integrated end switches is that an automatic calibration of the servomotors can be performed. If stops, e.g. simple rubber rings, would be used instead of end switches the servomotors could press against these stops with full force due to tolerances. This would lead to the servomotor either damaging itself or the puppet.

The deactivation of joints and therewith the corresponding servomotors leads to a basic problem, namely to maintain the posture against the gravitational force and the acceleration force acting through the movement of the other servomotors. If the servo motors are turned off, then each joint falls back in its rest position. For the purpose of holding the adjusted posture an integrated mechanical system individually aligns each joint's basic stiffness. Consequently a movement solely based on gravitational pull or on acceleration of the parent joint hierarchy is not possible.

To read out the joint's current position, the signal of the potentiometer was tapped and passed as analog signal into the controller.

3.3 Control Unit

At the lower back the puppet is attached to a stand. This stand is positioned on a base box, which contains the power supply and a micro-controller. The housing provides the following connections: network port, three USB ports, HDMI port, and a power connector. The main controller is a Raspberry PI 3 Model B, which also contains the software. This controller offers two options to drive the puppet — via remote network control or via an integrated software. Playing back and requesting the joint angle data happens with 20 – 30 Hz, depending on the angular distance between two sequential postures.

It is connected to an Arduino Mega via USB in order to control the servomotors of the puppet. The Arduino has 54 digital I/O pins and 16 analog inputs, which is not enough for our purpose. Each DOF needs the following six connections:

- 1 x ground,
- 1 x +5V,
- 1 x digital servo signal,
- 1 x analog potentiometer signal and
- 2 x digital I/O for the end switches.

The wiring of the components is a bit difficult as e.g. 37 cables pass through one arm. For each DOF four control cables plus two cables for ground and power are needed. Having 35 DOFs this results in a total number of 140 control signals and therefore cables. As it is not possible to pass all required cables through the back of the puppet, the number of cables had to be reduced drastically. We also observed, that many signals do not need to be read permanently, i.e. we could use multiplexing.

For this reason four Mux Shield II, manufactured by Mayhew Labs, are integrated in the puppet's shoulders and hips. Each of these Mux Shields has 48 freely configurable analog or digital inputs and is connected with the Arduino Mega via five cables only. In this way, the final number of control cables is reduced to 52. Due to the serial transfer of the control signals via the multiplexers, our system is able to realize a stable query of the joint angle data. It is therefore feasible to map the user's modifications on the puppet onto the virtual human model in real-time.

3.4 Breaking System

Our first evaluations have shown that the elbow, shoulder, knee and hip joints are subject to the highest influence of external forces. For the other joints can be stated that the friction of the joint axis's bearings and within the servo gear are high enough to compensate those forces.

To solve this problem the affected joints are equipped with linear solenoids. These press a break bracket against the joint axis. The linear solenoids are selected according to their achieved force.

4 SOFTWARE IMPLEMENTATION

The software framework provides two interfaces for the user — a GUI and a remote API. The GUI allows the user to operate the puppet as stand-alone computer, while the remote API allows to control the puppet from another computer via network communication.

Whenever the user changes a joint angle of the puppet, the adjustment is transferred onto the virtual model. The communication between the puppet and the software is done via mapping functions. Standard servos allow the shaft to be positioned at various angles, usually between 0 and 180°. Using the Arduino microcontroller the rotation of the servo shaft has a range of $s_{min} = 700$ to $s_{max} = 2300$. The corresponding potentiometer values can range from $p_{min} = 0$ to $p_{max} = 1023$. Due to the integrated end switches, the value of the servo shaft s_i of the i^{th} DOF is limited to the range $s_{i,l} \leq s_i \leq s_{i,h}$, thus the potentiometer value p_i varies between $p_{i,l} \leq p_i \leq p_{i,h}$. On the other hand the joint angle j_i associated with the i^{th} DOF has an achievable angle range of $j_{i,min} \leq j_i \leq j_{i,max}$. In order to playback the motion, one has to map the angle of the joints to the servo signal:

$$s_i = \frac{(j_i - j_{i,min})(s_{i,h} - s_{i,l})}{(j_{i,max} - j_{i,min})} + s_{i,l} \quad (4.1)$$

Following this, modifications on the puppet are mapped onto the virtual human's joints as follows:

$$j_i = \frac{(p_i - p_{i,l})(j_{i,max} - j_{i,min})}{(p_{i,h} - p_{i,l})} + j_{i,min} \quad (4.2)$$

Beside the joint angles, the software also calculates the position of the joint angle hierarchy and stops the movement of single joints in order to avoid collisions between the virtual human and the environment.

4.1 Functions of the Graphical User Interface

Within the GUI the visualization of the virtual factory, the digital human model and the current motion sequence is realized. Also, the actual posture of the input device is monitored and displayed. The user is

enabled to calibrate the puppet as well as to load, play back and save motions and environment geometries. It is also possible to activate or deactivate the servomotors of each DOF individually. A screenshot of the visualization is shown in Fig. 4.

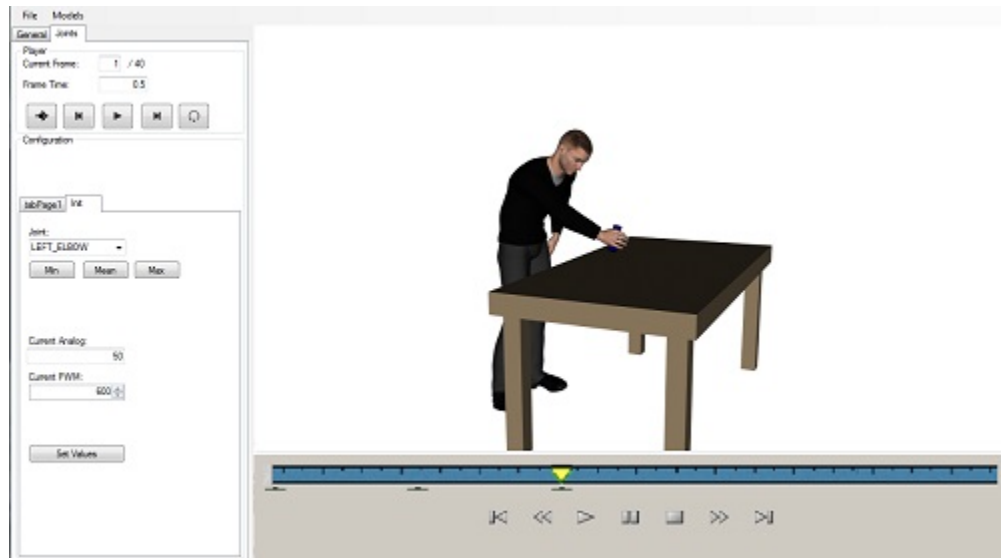


Figure 4: Screenshot of the GUI: (*left*) Menus for the configuration, (*bottom*) Timeline of the current motion sequence, (*middle*) 3D visualization of the factory layout.

A special part of the GUI is the timeline. It represents the frames of the current motion sequence. Using the timeline, the user is able to select a specific section of the sequence by selecting a start and an end frame. The selected area can then be edited, copied or deleted. Also, it is possible to mark a single frame in order to start the playback at this point or to insert another motion sequence.

The DOF can individually be enabled or disabled via the GUI (see Fig. 5).

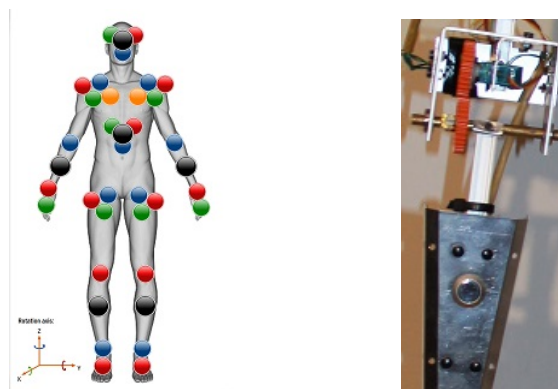


Figure 5: Interaction buttons: (a) virtual ones used in the GUI — colors indicate the rotation axis *blue*: Z, *red*: Y, *green*: X, *black*: button position on the puppet, (b) integrated interaction button on the puppet.

Using the virtual interaction buttons the user can power off whole joint angle chains at once. This function

is especially useful for the adjustment of motions. As a result of turning off just one DOF, the user is able to only adapt this one during the playback of the motion sequence.

It is also possible to recalibrate each DOF of every joint. Due to tolerances the values of the servomotors can possibly shift, i.e. with the adjusted value range the servomotor will not reach one or both limits. During the recalibration of the selected DOF the servomotor starts up both end switches and thereby defines the digital input value for the servomotor as well as the corresponding analog potentiometer value. Furthermore, the recalibration can be used to extend the joint range of the virtual model, e.g. to match it with a different percentile. During this process the user has to enter the new limits for the desired joint angles. By doing this, the pose of the digital human model is no longer consistent with the puppet as it is a reflection of the 50th percentile (see 3.2).

4.1.1 Motion Database

In order to provide the user with a set of prefabricated motion capture files has been established. This database contains common motion snippets like walking, carrying or grabbing. The integrated default motions are partitioned regarding their characteristics into repeatable and single sequences.

Repeatable sequences are motions that can be stringed together in the purpose of creating a motion sequence of a desired length. The database offers such iterable movements in the form of:

- walking at different speeds,
- one-handed carrying with the right or left hand at different speeds and
- two-handed carrying at different speeds.

On the other hand single motions are stand-alone activities in order to execute a specific task. For example the virtual human shall interact with an object. The motion database provides single movements like:

- turning motions with different angles (45°, 90°, 180°) and
- grabbing and placing at different heights.

However, the user is not limited to the application of only those motions we provide. Beside the fact that the motions can be adjusted, it is also possible to save them into the database for future reference.

4.1.2 Collision Handling

During the motion simulation an algorithm is continually calculating the distances and therefore potential collisions between the surface geometry of the virtual human model and the environment geometry within the digital factory. The user can choose between different options of how the system should react when a collision is detected.

1. A mere visual feedback within the 3D visualization of the GUI, i.e. all parts of the virtual human that collide with the environment are shown in red. No actions are carried out to solve the collision.
2. The playback is automatically paused and the affected body parts are highlighted in order to perform the following steps:
 - (a) The software stops the servomotors and visual display of the colliding bones or body parts.
 - (b) The user adjusts the pose of the puppet.
 - (c) When the collision is solved a predefined waiting period (default = 2 sec.) begins.

- After this period the playback continues. The countdown stops when the user moves the puppet and then starts anew.
 - Alternatively, the user presses Play to continue.
3. Based on inverse kinematics an algorithm solves the collision by moving the virtual human model. The resulting motion is then transferred onto the puppet.

4.2 Functions of the Application Programming Interface

Beside the use of the puppet as stand-alone system, it is also possible to control it via the remote API. Hence the user can utilize the puppet within other factory planning software. The functions exposed by the API contain all features that are implemented in the framework. The necessary data are transmitted via network communication.

4.2.1 Command Interface

Using the command interface the user can configure the puppet, remotely load geometries and motions, start, pause or stop the motion playback. The following commands are transmitted as ASCII strings via TCP:

- LOAD — load environment geometry and motion data file
- CALI — calibrate the specified degree of freedom
- PLAY — starts the playback of the motion sequence
- STOP — stops the playback of the motion sequence
- STEP — jump to a given frame in the motion sequence
- SETJ — enables or disables the specified joint
- SETP — set the value of a parameter (collision method, frame rate, etc.)
- RSET — delete loaded elements

4.2.2 Pose Sequence Interface

The motion sequence is stored within a datagram coded in ASCII format for each frame, that contains several values. The datagram is structured in lines carrying data of a specific type and starting with an identifier, as listed in Tab. 2.

identifier	description
fr	number of the current frame
ts	timestamp within the replayed motion
rp	global position of the root
jo	orientation of a joint

Table 2: Keywords within the UDP datagram.

The identifier `fr`, followed by an integer, is always the first line of the datagram and carries the frame number. After that, an optional line states the time stamp of the frame, indicated by `ts` followed by a float value. The identifier `rp` allows the specification of the global position of the virtual human within the virtual environment.

This is defined by the three floating point values $[x\ y\ z]$. The orientation of each joint of the puppet is identified by jo . The first number contains the number of joints. The orientation data of each joint shows up in three consecutive blocks: $[id]$, the orientation in euler angles $[\alpha\ \beta\ \gamma]$ and a rotation matrix $[b_0\ b_1\ b_2\ b_3\ b_4\ b_5\ b_6\ b_7\ b_8]$.

The interface loads the information into the puppet device. When the user manipulates the puppet in order to modify the sequence, the interface reads back the changed values of each servomotor and sends the joint orientations via UDP to the planning software. This can be done in a preselected frequency.

5 USE CASES

The following section gives an overview of typical use cases for our system. As the motion database cannot contain all possible movements, it is expectable that the user has to modify the existing motions for his specific purpose.

5.1 Modification of Motion Sequences

Whenever the user wants to modify an animation, e.g. due to changes of the factory layout, he can start the playback within the software and the puppet begins to perform the motion sequence in a specified speed. In this way, the puppet is already positioned and oriented in the correct posture, when the user decides to begin the modification at the desired frame of the sequence. Such a motion sequence can either be selected from the database, imported from an external source, or be generated by the user in a previous session.

One example is a walking sequence that shall be expanded by the digital human picking up an object, e.g. a hammer lying on a table, without stopping its walk (see Fig. 6). The grabbing motions from the database begin and end with a standing position of the human model, thus they are inappropriate for this purpose.



Figure 6: Modification of the motion sequence in order to grab a hammer.

To solve this issue a looped walking sequence can be modified using the puppet. Therefore, when the virtual human is nearby the object, the user presses the interaction button on the puppet's upper body and on the corresponding arm. As these limb groups are now free for modification, the user can position them in order to create a lean sideward position. After that he presses the buttons again and the normal motion of these body parts continues.

5.2 Generation of Walking Sequences

Although our electro-mechanical puppet is designed to allow an easy manipulation of virtual human motion, it might be necessary to build a new motion sequence from the ground up.

Within a digital factory it might be necessary that the digital human bypasses an obstacle. The database does not cover this sort of motion as it is highly complex in terms of e.g. the curve radius and depends on the whole factory layout.

A simple solution is to select one of the prefabricated walking motions from the database and adjust it. The software automatically loops the single steps and transfers the motion onto both the virtual human and the puppet. During the playback the user is now able to simply turn the puppet on its stand in order to steer the virtual human around the obstacle. An example of the generation of a walk path is shown in Fig. 7.

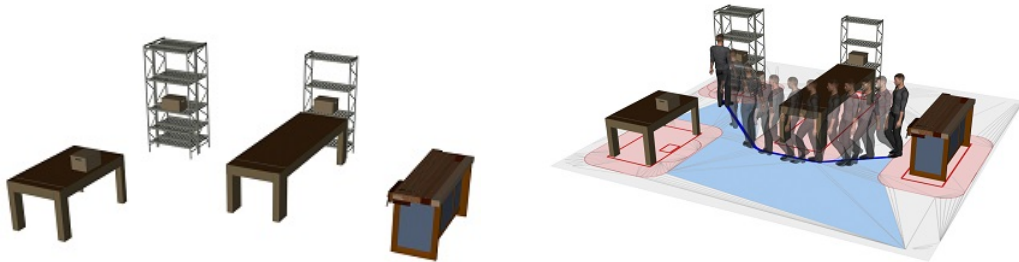


Figure 7: (a) Initial factory layout, (b) Projection of collision areas and generated walk path around a table.

5.3 Modification of Walking Sequences

In order to not always have to create a walking sequence anew, the user can load such a complete sequence into the database. Due to changing factory layouts, it might be necessary to modify the orientation of the walk path. This can be done using the steering method described in the second use case (section 5.2). However, due to the fixed length of the loaded sequence, the virtual human will not be able to reach a desired destination. The easiest solution is to fill up the sequence with copied frames. The user selects start and end frame of the existing sequence, e.g. a single walk step and is able to insert these frames at the end of the current walk path. The transitions will be connected using a simple motion blending method to achieve a natural movement. This method is illustrated in Fig. 8.

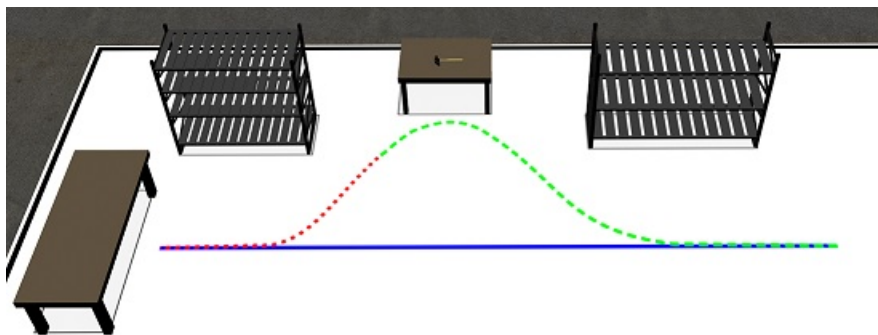


Figure 8: Change of the walk path. *blue*: original path, *green*: redirected path, *red*: path extension.

6 CONCLUSIONS

In this paper we presented an innovative electro–mechanical input device in the form of a human–like puppet. In combination with the simulation software we were able to build a new system that addresses the needs of motion modification in advanced process planning.

We introduced a device that is capable of performing existing motion data in a smooth and realistic manner, and letting the user conduct manual adjustments of a motion sequence during a desired timespan. These fundamental distinctions from other human–like devices constitute our main technical innovation. Especially the modification of motions using our puppet is faster and more accurate than via mouse and keyboard. The intuitive handling of the puppet enables the manipulation of animations even by untrained users without prior knowledge and experience in animation editing. The motion performing puppet is giving them a fast visual and haptic feedback of their changes, which are mapped onto a virtual human model in real–time. Furthermore, the puppet’s anthropometric design follows the principles of human factors in engineering. Factory planners benefit from our solution as it reduces the effort of designing and modernizing facilities. These aspects and the human–like appearance of our device ensure a high user acceptance. With its low expenditure of time and money, compared to other solutions, our system is applicable even to small and medium–sized enterprises. The pose sequence interface also provides the ability to integrate the puppet into other software systems. In the context of digital factories and virtual prototyping using our puppet in an ergonomic simulation is easily conceivable.

The next step in enhancing this system is to evaluate the puppet regarding its robustness and accuracy. In addition, we are going to extend the integrated motion database. As part of this extension, we plan to development methods for using our electro–mechanical input device as a tool to perform queries for motion data.

ORCID

Thomas Kronfeld, <http://orcid.org/0000-0002-6399-8352>

Guido Brunnett, <http://orcid.org/0000-0002-8224-015X>

REFERENCES

- [1] CELSYS Inc.: Humanoid Input Device QUMARION, 2012. <http://www.clip-studio.com/quma/en/products/qumarion>.
- [2] German Institute for Standardization: DIN 33402-2 Ergonomics - Human body dimensions - Part 2: Values. Tech. rep., DIN, Berlin, 2005. <https://www.beuth.de/en/standard/din-33402-2/84092742>.
- [3] Glauser, O.; Ma, W.C.; Panozzo, D.; Jacobson, A.; Hilliges, O.; Sorkine-Hornung, O.: Rig Animation with a Tangible and Modular Input Device. *ACM Transactions on Graphics*, 35(4), 1–11, 2016. <http://doi.org/10.1145/2897824.2925909>.
- [4] Jacobson, A.; Panozzo, D.; Glauser, O.; Pradalier, C.; Hilliges, O.; Sorkine-Hornung, O.: Tangible and Modular Input Device for Character Articulation. *ACM Transactions on Graphics*, 33(4), 1–12, 2014. <http://doi.org/10.1145/2601097.2601112>.
- [5] Kato, R.; Izumida, K.; Shigeno, H.; Okada, K.i.: Individual Learning Support about First Aid with a Human-Shaped Input Device. In *Proceedings of the 15th International Conference on Mobile and Ubiquitous Multimedia - MUM '16*, 181–189. ACM Press, New York, USA, 2016. <http://doi.org/10.1145/3012709.3012720>.
- [6] NASA Scientific and Technical Information Office: Anthropometric Source Book Volume I: Anthropometry for Designers. NASA, Yellow Springs, Ohio, 1978. <https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19790003563.pdf>.

- [7] Numaguchi, N.; Nakazawa, A.; Shiratori, T.; Hodgins, J.K.: A Puppet Interface for Retrieval of Motion Capture Data. In Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation - SCA '11, 157–166. ACM Press, 2011. <http://doi.org/10.1145/2019406.2019427>.
- [8] Oshita, M.; Yamamura, K.; Honda, A.: Finding Rules of Attractive Human Poses Using Decision Tree and Generating Novel Attractive Poses. In Proceedings of the Computer Graphics International Conference on - CGI '17, 1–6. ACM Press, New York, USA, 2017. <http://doi.org/10.1145/3095140.3095173>.
- [9] Sundin, A.; Örtengren, R.: Digital Human Modeling for CAE Applications. In G. Salvendy, ed., Handbook of Human Factors and Ergonomics, chap. 39, 1053–1078. John Wiley & Sons, Inc., Hoboken, NJ, USA, 3rd ed., 2006. <http://doi.org/10.1002/0470048204.ch39>.
- [10] Tayama, Y.; Hagino, M.; Okada, K.I.: Using a Human-Shaped Input Device for Telerehabilitation. In 2014 IEEE International Conference on Healthcare Informatics, 135–143. IEEE, 2014. <http://doi.org/10.1109/ICHI.2014.25>.
- [11] Wang, X.; Chen, Q.; Wang, W.: 3D Human Motion Editing and Synthesis: A Survey. In Computational and Mathematical Methods in Medicine, 1–11, 2014. <http://doi.org/10.1155/2014/104535>.
- [12] Wiegmann, D.; Brüggemann, H.; Rausch, A.: Design and Evaluation of a Human-Like Puppet as an Input Device for Ergonomic Simulation. In V.G. Duffy, ed., Digital Human Modeling. Applications in Health, Safety, Ergonomics, and Risk Management: Ergonomics and Design. DHM 2017. Lecture Notes in Computer Science, 393–403. Springer International Publishing AG 2017, Cham, Switzerland, 2017. http://doi.org/10.1007/978-3-319-58463-8_33.
- [13] Yamane, K.; Kuffner, J.J.; Hodgins, J.K.: Synthesizing Animations of Human Manipulation Tasks. In ACM SIGGRAPH 2004 Papers on - SIGGRAPH '04, 532–539. ACM Press, New York, USA, 2004. <http://doi.org/10.1145/1186562.1015756>.
- [14] Yoshizaki, W.; Mochimaru, M.; Sugiura, Y.; Chiou, A.C.; Hashimoto, S.; Inami, M.; Igarashi, T.; Akazawa, Y.; Kawachi, K.; Kagami, S.: An Actuated Physical Puppet as an Input Device for Controlling a Digital Manikin. In Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11, 637–646. ACM Press, 2011. <http://doi.org/10.1145/1978942.1979034>.
- [15] Zhou, H.; Hu, H.: Human motion tracking for rehabilitation-A survey. Biomedical Signal Processing and Control, 3(1), 1–18, 2008. <http://doi.org/10.1016/j.bspc.2007.09.001>.