Hindawi
Journal of Robotics
Volume 2022, Article ID 7561891, 14 pages
https://doi.org/10.1155/2022/7561891

Research Article

Research on Simulation System for Human-SRL Collaborative Motion Planning

Xiaobo Zhang,[1,2,3] **Jinguo Liu**,[1,2] **and Yangmin Li**[1,2,4]

[1] *State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China*
[2] *Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110169, China*
[3] *University of Chinese Academy of Sciences, Beijing 100049, China*
[4] *Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong 999077, China*

Correspondence should be addressed to Jinguo Liu; liujinguo@sia.cn

Received 16 July 2022; Accepted 21 October 2022; Published 4 November 2022

Academic Editor: L. Fortuna

At present, there is a problem of visual area switching in the existing SRL (supernumerary robotic limb) operation methods. In response to this problem, the authors' previous work proposed a new SRL operation method called relatively independent operation, and proposed a corresponding software architecture. The purpose of this paper is to solve the theoretical problems and engineering realization problems of the human-SRL skeleton algorithm module in the software architecture. Therefore, modeling, data collection, data processing, and visualization of a human-SRL system are studied in this paper. Firstly, a human-SRL skeleton visualization simulation system is developed. The condition setting, the applications, and the core algorithm of the simulation system are introduced. The core algorithm mainly contains four types of important functions, namely skeleton model building functions, human-SRL data collection functions, human-SRL data processing functions, and skeleton visualization functions. Secondly, the implementation principles of these four functions are described: (1) For the skeleton model building functions, a human-SRL skeleton model is proposed which is an integration of a human skeleton model and an SRL skeleton model. The construction methods of these three skeleton models are described. (2) For the remaining functions, how to collect and process human data, SRL data, human-SRL data, and how to visualize a human-SRL skeleton are described. Finally, the visualization effect of the developed simulation system on human-SRL skeleton movement is verified by experiments, which proves the correctness of the functions in the simulation system.

1. Introduction

Supernumerary robotic limb (SRL) [1–3] is a new type of wearable robot. It is connected to the human body through straps and acts as an upper limb or a lower limb of the human to enhance the original working space and operation ability of the human. For example, SRL can be used to assist soldiers to complete the simultaneous shooting of multiple guns, to assist installation workers to clamp panels to be installed [4, 5], to assist building workers, farmers, and welders to support their body [6], to assist supporting cleaning tools [7], and to assist the disabled to walk [8].

There are relatively few studies related to SRL. At present, there have been studies involving prototype structure design [5–9], joint control technology [4,6,8,10], human-machine collaborative motion planning [11], SRL operation methods [12–17], human-machine dynamics modeling [18–20], etc. Current operation methods for SRL include EMG mapping [12], EEG mapping [13], foot mapping [14, 15], head mapping [16], and eyeball mapping [17]. These operation methods have a common problem of visual area switching. When tasks of the native limb and the SRL are located in different visual areas, the human brain and eye resources need to switch between these two tasks. This is essentially a serial process, and the operation efficiency is low. In response to this problem, the authors propose a new SRL operation method called relatively independent operation (RIO) (the corresponding article has been accepted and will be published soon) which improves the operation efficiency, and develops a software framework to realize RIO. However, implementation details of each module in the software framework are not provided, only some conceptual descriptions are provided. This paper conducts extended research on the human-SRL skeleton module in the software framework. Based on the human-SRL skeleton module, combined with a collision detection module, an intention recognition module, and an autonomous decision-making module, the RIO can be realized.

In order to refine the human-SRL skeleton algorithm module in the software framework for RIO, contributions in this paper are summarized as follows: (1) A human-SRL skeleton model is proposed. This model is the basis for realizing human-SRL cooperative motion planning. (2) Second, based on this model, a human-SRL skeleton visualization simulation system (hereinafter referred to as the simulation system) is developed, which can intuitively display human-SRL cooperative movements. (3) The theoretical basis and engineering implementation details of the important functions involved in the simulation system are elaborated.

The remainder of this paper is as follows Section 2 introduces the condition setting, the applications, and the core algorithm of the simulation system. The core algorithm mainly includes skeleton model building functions, human-SRL data collection functions, human-SRL data processing functions, and skeleton visualization functions. The construction methods of the human skeleton model, the SRL skeleton model, the human-SRL skeleton model, and the data structure of these three skeleton models are described in Section 3. This section corresponds to the skeleton model building functions of the core algorithm in Section 2. Section 4 introduces the human data collection and processing, the SRL data collection and processing, and the human-SRL skeleton visualization. This section corresponds to the human-SRL data collection functions, the human-SRL data processing functions, and the human-SRL skeleton visualization functions of the core algorithm in Section 2. Finally, Section 5 tests the visualization effect of the simulation system, which verifies the correctness of the functions in the simulation system.

2. Introduction to Simulation System

2.1. Condition Setting. The base of an industrial manipulator is fixed to ground, and the base of a manipulator with a mobile platform is movable. SRL is different from the above two types of manipulators in that it is usually connected with human muscle tissue by straps and has a flexible and movable base. Current technology has not been able to solve the problem of accurate modeling and measurement of the flexible and movable SRL base. Moreover, the disturbance compensation control problem of the SRL robot system with this kind of base is also very complex and needs to be solved by means of linear [21] or nonlinear [22] control techniques. Therefore, this study assumes that the SRL base is rigidly connected to the human body, that is, the SRL has a rigid and movable base. Under this premise, modeling, data collection, data processing, and visualization of a human-SRL system are studied in this paper. If future technology solves the problem of accurate modeling and measurement of the flexible and movable SRL base, just add corresponding modules between the SRL base node and its parent node in the human-SRL skeleton model proposed in this paper, and then the disturbance of the base can be compensated by appropriate control methods to ensure the positioning accuracy of the SRL end. The attitude measurement of the SRL base can be realized by IMU sensors. Under the premise of improving the measurement accuracy of the IMU sensors as much as possible, the noise of the IMU sensors may also be transformed to improve the performance of the control system. Ref. [23] provides a good example. The authors propose the concept of stochastic resonance and introduce a nonlinear controller into a linear system to provide the control system with capabilities for rejecting noise, which further improves the performance of the control system.

2.2. Applications. In this study, a human-SRL skeleton model is proposed, and a simulation system is developed based on this model. The main function of the simulation system is to collect and process the data from the human body and the SRL in real-time, and realize the visualization of the human-SRL skeleton. The simulation system utilizes a skeleton model to facilitate adding more modules in the future. For example, a collision detection module, a human intention recognition module, and an autonomous decision-making module can be added to realize human-SRL cooperative motion planning. The collision detection module can collect collision information in the process of human-SRL cooperative movement with the help of a human-SRL skeleton envelope model (see Figure 1). The human intention recognition module indicates that the SRL can determine human intention according to different skeleton configurations of the human, and further complete certain tasks autonomously according to the autonomous decision-making module. The authors' previous work (the corresponding article has been accepted and will be published soon) proposed a relatively independent operation method, and presented a software architecture that demonstrates in detail how all the above modules realize the human-SRL collaborative motion planning. Human-SRL collaborative motion planning belongs to the category of human-machine interaction. The SRL can perform tasks demonstrated by human's original limbs, and modify the demonstration trajectory to prevent collision with the human body. In addition, there are many other ways to realize the interaction between human and SRL. Some mathematical models that describe how humans interact with machines through human behavior (for example, physiological behavior or neurological behavior) are summarized in [24]. These models attempt to mathematically formulate certain phenomena of the human biological sciences and are worthy of our in-depth study.

2.3. Core Algorithm. The pseudo-code of the core algorithm of the simulation system is shown in Algorithm 1, which is called as human-SRL visualization algorithm (HSRLVA). The input of the algorithm includes human motion capture system data *dataHuman* and SRL servo motor data *dataSRL*,

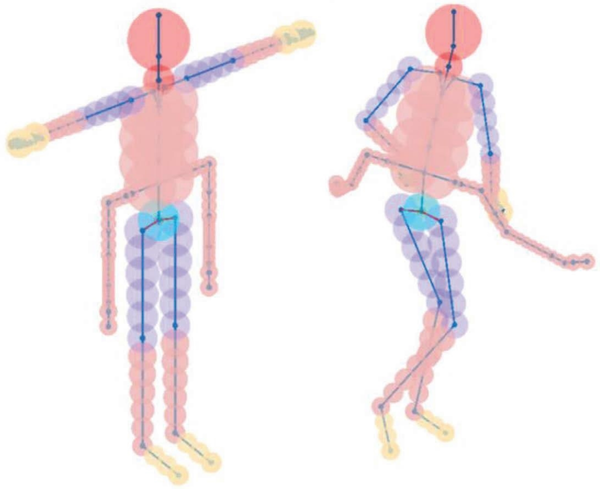


Figure 1: Schematic diagram of a human-SRL skeleton envelope model.

```
Input: Human motion capture system data dataHuman, SRL servo motor data dataSRL
Output: Display a human-SRL skeleton on a monitor terminal.
//Build a human skeleton model
modHumSke←BuildHum()
//Build an SRL skeleton model
modSRLSke←BuildSRL()
//Build a human-SRL skeleton model
modSke←BuildHumSRL()
while true do
    //Set a sampling time step
    step←SetStep()
    //Collect motion capture system data and servo motor data
    dataHuman←DataColHum(dataPort1)
    dataSRL←DataColSRL(dataPort2)
    //Decoding, encoding, and integration of the motion capture system data and the servo motor data
    dataDecHum←DecodingHum(dataHuman, modHumSke)
    dataDecSRL←DecodingSRL(dataSRL, modSRLSke)
    dataEncSRL←EncodingSRL(dataDecSRL, modSRLSke)
    dataIntHumSRL←Integration(dataDecHum, dataEncSRL, modSke)
    //Get skeleton model node position data dataNodePos and node connection relationship data dataNodeCon
    (dataNodePos, dataNodeCon)←GetNodePos(modSke)
    //Visualize the human-SRL skeleton
    modSkeHandle←Visualizing(dataNodePos, dataNodeCon, dataIntHumSRL)
end while
```

Algorithm 1: Human-SRL visualization algorithm (HSRLVA).

and the output is a human-SRL skeleton. The algorithm steps are as follows: (1) Use functions *BuildHum*(), *BuildSRL*(), and *BuildHumSRL*() to build a human skeleton model *modHumSke*, an SRL skeleton model *modSRLSke,* and a human-SRL skeleton model *modSke*, respectively. Refer to Sections 3.1 and 3.2 for implementation principles of these three functions. These three skeleton models are essentially three structure arrays in the Matlab environment, see Section 3.3.2 for details. (2) After getting the three skeleton models, enter the main loop of the program. Inside the loop body, first set a sampling time step. Then, utilize functions *DataColHum*() and *DataColSRL*() to collect the motion capture system data *dataHuman* and the servo motor data *dataSRL*, respectively. After that, employ function *DecordingHum*() to decode the motion capture system data *dataHuman*. Use functions *DecordingSRL*() and *EncordingSRL*() to decode and encode the SRL servo motor data. Finally, a human skeleton motion quantity *dataDecHum* and an SRL skeleton motion quantity *dataEncSRL* are obtained. Refer to Sections 4.1 and 4.2 for details. (3) Employ function *Integration*() to perform an integration operation on the human skeleton motion quantity *dataDecHum* and the SRL skeleton motion quantity *dataEncSRL* to obtain a human-SRL skeleton motion quantity *dataIntHumSRL*. Utilize function *GetNodePos*() to obtain the node position data *dataNodePos* and the node connection relationship data *dataNodeCon* of the human-SRL skeleton model *modSke*. Finally, *dataNodePos*, *dataNodeCon*, and *dataIntHumSRL* are used as the input of a visualization function *Visualizing*() to complete the visualization of the human-SRL skeleton. See Section 4.3 for details. The human-SRL skeleton is dynamically updated once at each sampling moment, which can realize a real-time display of the human-SRL skeleton, so as to show the results of human-SRL co-operative movements.

3. Construction Method and Data Structure of Skeleton Model

A skeleton model is a simple model that can be used to represent the overall outline of a multi-rigid body. This paper proposes a human-SRL skeleton model which is an integration of a human skeleton model and an SRL skeleton model. The construction methods of these three skeleton models will be introduced in this section. Additionally, in order to facilitate readers to deeply understand the program implementation mechanism of Algorithm 1, how to represent a skeleton model in a computer with a kind of proper data structure will be introduced.

3.1. Construction Method of Human Skeleton Model. A skeleton model includes three elements and two rules. The three elements are nodes, line segments, and coordinate systems. The two rules are the connection rules of nodes and the transformation rules of coordinate systems. The connection rules of nodes are represented by a node tree, and line segments can be obtained by connecting nodes. Coordinate systems are fixed online segments, and the rotation of coordinate systems drives nodes and line segments to move. It is the essence of skeleton model construction to use these three abstract elements to represent an external outline of a multi-rigid body and use coordinate system transformations to represent movements of a multi-rigid body. Skeleton model construction needs to solve three problems: (1) How to build nodes? (2) How to build a node tree and use it to obtain line segments? (3) How to build coordinate systems and set coordinate system transformations?

Figure 2 shows the appearance of a completed human skeleton model. In this article, the appearance of a skeleton model is abbreviated as skeleton, and it is different from the skeleton model, which needs to be distinguished. Because our follow-up research involves gesture recognition, the hand skeleton is arranged with many nodes (the right hand in Figure 2, the left hand is similar). It can be found that nodes are generally placed at human joints, ends of human limb branches, or certain points of interest. These three types of nodes are called joint nodes, branch nodes, and special nodes. For example, nodes arranged at shoulder, elbow, wrist, hip, knee, ankle, neck, and spine joints belong to joint nodes. Nodes arranged at end of the upper limb, end of the lower limb, and end of the head belong to branch nodes. The node near the human eyes is a special node. Special nodes can facilitate movement analysis on places of interest. Branch nodes are arranged to ensure the integrity of the

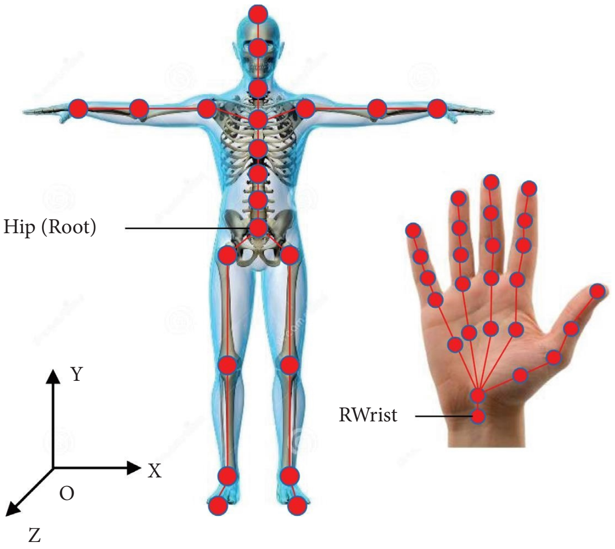


FIGURE 2: Schematic diagram of the appearance of the human skeleton model.

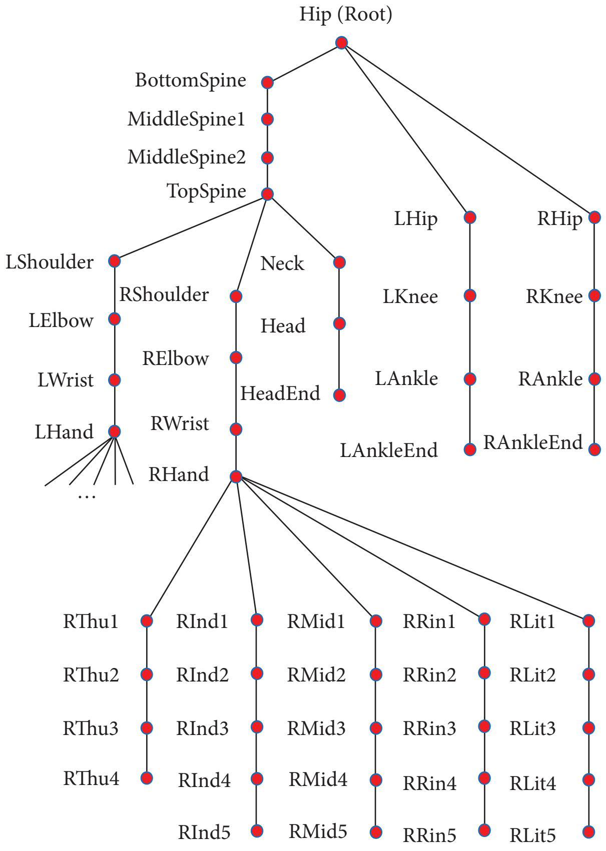
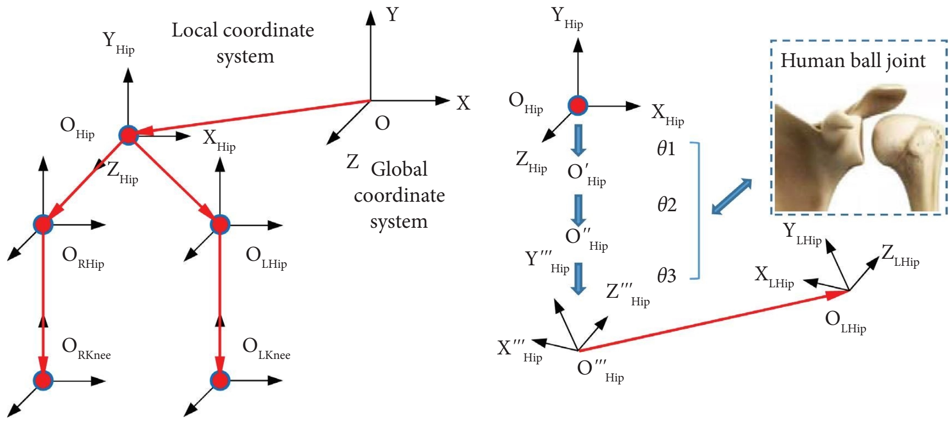


FIGURE 3: The node tree of the human skeleton model.



(a) (b)

Figure 4: Schematic diagram of the establishment and transformation rules of coordinate systems.

outline. Joint nodes are core nodes to drive a skeleton to move. After nodes are established, line segments can be obtained according to a node tree that provides nodes' connecting rules. The node tree of the human skeleton model in this paper can be seen in Figure 3. The Hip node is a root node, and it contains three branches, namely a left leg branch, a right leg branch, and a spine branch. The TopSpine node in the spine branch includes three branches, namely a left arm branch, a right arm branch, and a head branch. The left and right arm branches respectively include five branches at the wrist joint nodes (LWrist and RWrist nodes), corresponding to five fingers. In each branch of Figure 3, the upper node is the parent node, and the lower node is the child node. The entire node tree has only one root node. Each node can only have one parent node, and can have multiple child nodes. The entire human skeleton model has a total of 72 nodes and 71 line segments.

Coordinate systems consist of a global coordinate system and several local coordinate systems. The global coordinate system is fixed to ground, and represented by $O - XYZ$, abbreviated as O. Each local coordinate system is fixed to a node (or a line segment), and represented by $O_{\text{NodeName}} - XYZ_{\text{NodeName}}$, abbreviated as O_{NodeName} (as shown in Figure 4(a)). When the human body is in an initial state of the "T" shape (shown in Figure 2), the Z axes of all local coordinate systems point to the front of the human body, the X axes point to the left side of the human body. The plane formed by each Z axis and each X axis is parallel to ground. The Y axes can be obtained by a right-hand rule and they are perpendicular to ground. To ensure the mobility of the skeleton, virtual ball joints are established at all parent nodes, similar to the human ball joints, with three rotational degrees of freedom. As shown in Figure 4(b), a coordinate system transformation between a parent-child node pair Hip and LHip is taken as an example for further description. Denote the three degrees of freedom as θ_1, θ_2 and θ_3. The θ_1 represents the angle required to rotate O_{Hip} around Z_{Hip} to obtain O'_{Hip}. The θ_2 represents the angle required to rotate O'_{Hip} around X'_{Hip} to obtain O''_{Hip}. The θ_3 represents the angle required to rotate O''_{Hip} around Y''_{Hip} to obtain O'''_{Hip}. The rotation order of the coordinate systems needs to be consistent with the selected motion capture system, and here is the rotation order of the motion capture system used in this article. After obtaining O'''_{Hip}, the $O_{L\text{Hip}}$ can be obtained by translating O'''_{Hip} along the vector $O'''_{\text{Hip}}O_{L\text{Hip}}$. The entire coordinate system transformation process can be expressed by a formula as follows:

$$\mathbf{T}_{CN} = \mathbf{T}_{PN}\text{Rot}z\left(\theta_1\right)\text{Rot}x\left(\theta_2\right)\text{Rot}y\left(\theta_3\right)\text{Trans}\left(l\right), \tag{1}$$

where $\mathbf{T}_{CN}$ and $\mathbf{T}_{PN}$ respectively represent the homogeneous transformation matrices of a child node and a parent node relative to the global coordinate system. $\text{Rot}z\left(\theta_1\right)$, $\text{Rot}x\left(\theta_2\right)$, and $\text{Rot}y\left(\theta_3\right)$ respectively represent the rotation matrices around the Z, X and Y axes. $\text{Trans}\left(l\right)$ is the translation matrix. These four matrices are further expanded as shown in formulas (2)–(5):

$$\text{Rot}z\left(\theta_1\right) = \begin{bmatrix} \cos\left(\theta_1\right) & \sin\left(\theta_1\right) & 0 & 0 \\ -\sin\left(\theta_1\right) & \cos\left(\theta_1\right) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2}$$

$$\text{Rot}x\left(\theta_2\right) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\left(\theta_2\right) & \sin\left(\theta_2\right) & 0 \\ 0 & -\sin\left(\theta_2\right) & \cos\left(\theta_2\right) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3}$$

$$\text{Rot}y\left(\theta_3\right) = \begin{bmatrix} \cos\left(\theta_3\right) & 0 & -\sin\left(\theta_3\right) & 0 \\ 0 & 1 & 0 & 0 \\ \sin\left(\theta_3\right) & 0 & \cos\left(\theta_3\right) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4}$$

$$\mathrm{Trans}(l) = \begin{bmatrix} 0 & 0 & 0 & l_x \\ 0 & 0 & 0 & l_y \\ 0 & 0 & 0 & l_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (5)$$

A motion capture system can provide the angles θ_1, θ_2 and θ_3, and the vector $O_{PN}O_{CN}$ required for coordinate transformations of all parent-child node pairs. Essentially, the origin of the global coordinate system can be regarded as the parent node of the Hip node. By recursion from the global coordinate system, the local coordinate systems of all nodes can be obtained. The origin of each local coordinate system coincides with the corresponding node. Then all line segments can be obtained by connecting parent nodes and child nodes with the help of a node tree. Finally, the human skeleton shown in Figure 2 can be drawn.

3.2. Construction Method of SRL Skeleton Model. Assuming that SRL has a rigid and movable base, this section will build a skeleton model of an SRL prototype developed by us. If other SRL prototypes are selected, the construction method is similar. The construction of the SRL skeleton model needs to solve the aforementioned three problems. In addition, an additional problem needs to be solved, that is, how to integrate an SRL skeleton model with a human skeleton model to obtain a human-SRL skeleton model.

Human joints can achieve movement by pulling bones through muscle tissue, and the center of each joint can be regarded as a skeleton model node. The SRL joints in this paper are different from human body joints. They drive links to move through the rotation of motors. Therefore, joint nodes can be arranged on the rotating shafts of motors, and in addition, branch nodes need to be arranged. Special nodes are optionally placed. The SRL prototype is shown in Figure 5. It has a total of 7 degrees of freedom (DOFs), including 6 revolute DOFs and 1 prismatic DOF. The shoulder joint has 3 DOFs, and its 3 rotation axes are perpendicular to each other and intersect at one point. The elbow joint and the prismatic joint each occupy 1 DOF, and the wrist joint occupies 2 DOFs. All joints use high-precision and high-torque servo motors as power sources, and connect their next modules through links, adapter parts, and fixing parts to realize torque transmission. The state of the SRL shown in Figure 6(a) is an initial state. In the initial state, the rotation axis or prismatic axis corresponding to each DOF is defined as LAxis_i or RAxis_i, and the rudder disc plane of each motor is defined as $\mathrm{LRudPla}_i$ or $\mathrm{RRudPla}_i$ where $i = 1, 2, \ldots, 7$. L and R respectively represent the left and the right SRL. The plane that is parallel to the sagittal plane of the human body and passing through $L\mathrm{Axis}_2$ or $R\mathrm{Axis}_2$ is defined as LSagPla or RSagPla. Then the SRL skeleton model nodes can be determined according to the following rules: (1) All the nodes are arranged on the coronal plane of the human body, and the left and right SRL skeleton model nodes are symmetrical with respect to the sagittal plane of the human body in the initial state. Here, the right SRL is used as an example to determine the nodes, and the left SRL is similar. (2) First, the base node RBase needs to be determined. Under the assumption that the SRL has a rigid and movable base, RBase is stationary relative to the human body. The SRL base in this paper is fixed at the waist of the human body. Therefore, the intersection point of $R\mathrm{Axis}_1$ and the right SRL base at the waist of the human body is set as the node RBase. By connecting the BottomSpine node in the human skeleton model with the RBase node in the SRL skeleton model, integration of these two skeleton models can be achieved. This is the core idea of realizing the integration of the human skeleton model and the SRL skeleton model to obtain the human-SRL skeleton model. Therefore, Section 3 only introduces the construction methods of the human skeleton model and the SRL skeleton model, and the human-SRL skeleton model is obtained based on these two skeleton models. (3) The intersection points of $R\mathrm{Axis}_1$ and $R\mathrm{RudPla}_1$, $R\mathrm{Axis}_2$ and the human coronal plane, $R\mathrm{Axis}_3$ and $R\mathrm{RudPla}_3$, $R\mathrm{Axis}_4$ and RSagPla, $R\mathrm{Axis}_5$ and $R\mathrm{RudPla}_5$, $R\mathrm{Axis}_6$ and RSagPla, $R\mathrm{Axis}_7$ and $R\mathrm{RudPla}_7$ are respectively defined as $RJ1$, $RJ2$, $RJ3$, $RJ4$, $RJ5$, $RJ6$ and $RJ7$. The center of the end effector is defined as REnd. (4) Figure 6(b) shows the constructed SRL skeleton, and Figure 6(c) shows the node tree of the SRL skeleton model. The nodes are connected in an orderly manner using the node tree.

The rules for establishing the local coordinate systems of the SRL skeleton model are consistent with the human skeleton model. An example of the local coordinate system of SRL joint 4 is given in Figure 6(a). When performing coordinate system transformations between a parent node and a child node, since each revolute joint of the SRL has only one rotational DOF, the rotation angle of the other two rotational DOFs can be specified as 0. For the prismatic joint, first set its three rotational DOFs as 0, and then change the translation displacement.

3.3. Data Structure of Skeleton Model. A data structure is a way that describes how a computer stores and organizes data. How to use a suitable data structure to represent a skeleton model? At present, motion capture systems commonly use a BVH file to represent a human skeleton model. However, computer programming languages cannot use BVH files as data structures for representing skeleton models. In order to let the readers know more about how to represent a skeleton model during programming, this section takes the Matlab programming language as an example to give a data structure of a skeleton model, that is, a structure array. This section first introduces the BVH file, and then introduces the structure array.

3.3.1. BVH File. BVH (Biovision Hierarchy Motion) file is a general human feature animation file that is widely used by various animation production applications and motion capture systems. It stores both a human skeleton model and the data that drives the human skeleton to move. Figure 7 shows an example of a BVH file. The file contains two parts, namely a skeleton model part (with “Hierarchy” as the keyword, all keywords are shown in bold) and a data part

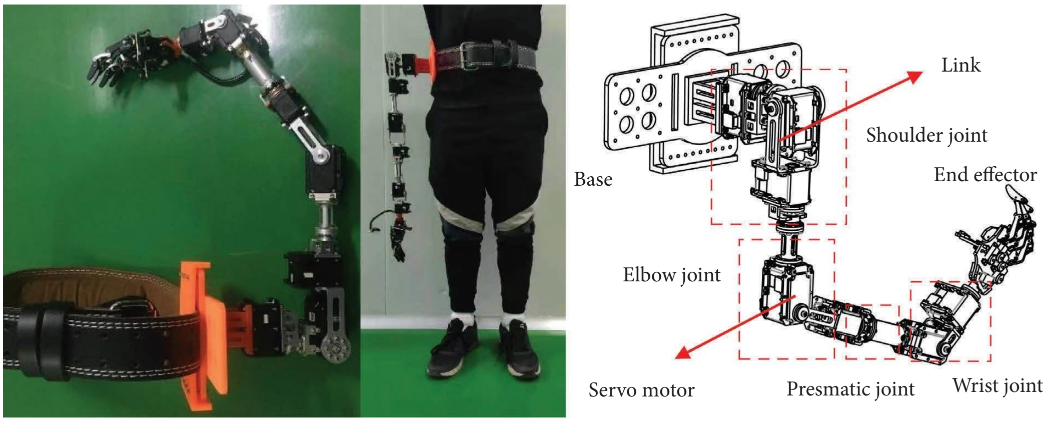


Figure 5: The SRL prototype.

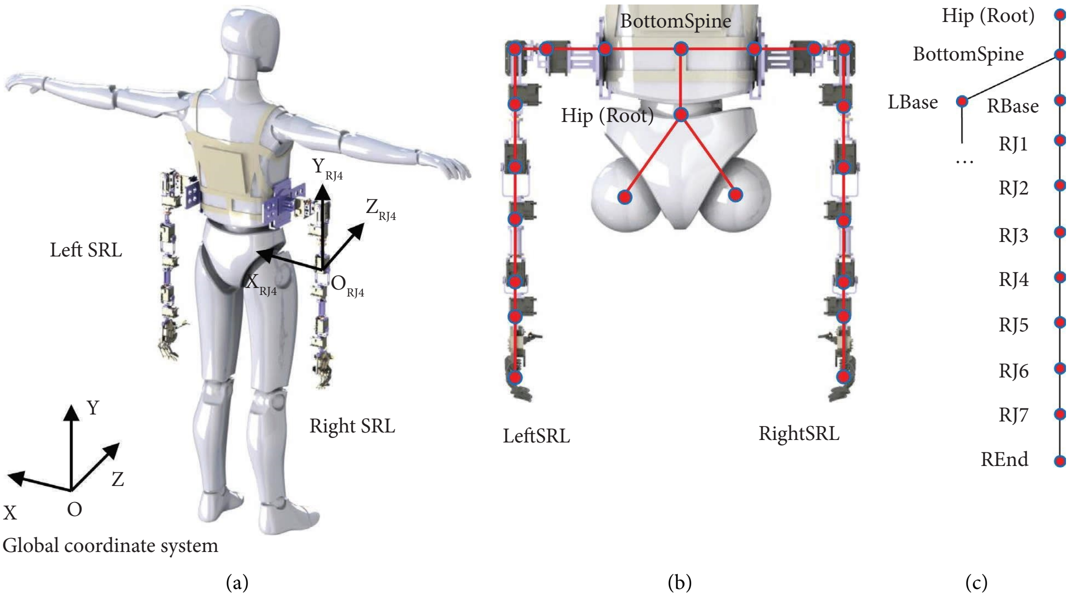


Figure 6: The human-SRL system 3D model, the SRL skeleton, and the node tree of the SRL skeleton model.

(with "Motion" as the keyword). The skeleton model part essentially describes all the nodes, a node tree, and coordinate system transformation rules. The nodes include a root node (keyword "Root"), common nodes (keyword "Joint"), and branch nodes (keywords "End Site"). The node tree is represented by "{}". The information of each node is contained in "{}", and each child node is embedded in its parent node. The coordinate system transformation rules consist of two items, namely "Offset" and "Channels". The "Offset" represents the inherent offset between the local coordinate systems of a parent-child node pair. The "Channels" is the transformation order between the local coordinate systems of a parent-child node pair. The total number of frames ("Frames"), frame sampling time ("Frame Time"), and frame data are recorded in the data part. Each frame of data arranges rotation or translation data for coordinate system transformations.

3.3.2. Structure Array. In this paper, the simulation system is programmed by Matlab language, and the reasonable storage and organization of a skeleton model can be realized with the help of a structure array. In Section 2.3, the human-SRL skeleton model *modSke*, the human skeleton model *modHumSke*, and the SRL skeleton model *modSRLSke* are three composite variables and have the same structure array form. *modSke* is an integration of the latter two. This section will describe the construction method of the structure array of a skeleton model.

As mentioned in Section 3.1, a skeleton model consists of three elements and two rules. The three elements are nodes, line segments, and coordinate systems. The two rules are node connection rules and coordinate system transformation rules. The essence of using a structure array to represent a skeleton model is to reasonably set the fields and values of the structure array so that the

```
Hierarchy
Root RootName
{
  Offset OFF_x OFF_y OFF_z
  Channels 6 Xposition Yposition Zposition
       Zrotation Xrotation Yrotation
  Joint ParentJointName
  {
    Offset …
    Channels …
    Joint ChildJointName
    {
      …
      End Site
      {
        Offset …
      }
    }
  }
}
Motion
Frames : FramesNumber
Frame Time : TimeStep
A frame of data A frame of data …
```

Figure 7: An example of a BVH file.

structure array itself can directly or indirectly obtain the above three elements and two rules. The basic idea is that each node in a skeleton model is represented by a structure, and the set of all nodes is represented by a structure array. Then, by assigning associated fields and field values of each node, line segments, coordinate systems, node connection rules, and coordinate system transformation rules can be obtained directly or indirectly. Table 1 shows a structure example of a single node, and the field values are given by taking Hip node as an example. Each node (or structure) includes 7 fields, namely name “Name,” identification “ID,” parent identification “Parent ID,” child identification “Child ID,” inherent offset “Offset,” translation index of the local coordinate system transformation of a parent-child node pair “Translation Index,” and rotation index of the local coordinate system transformation of a parent-child node pair “Rotation Index”.

The three elements and two rules are obtained in the following ways: (1) The spatial position of each node coincides with the origin of its local coordinate system which can be obtained recursively through “Offset,” “Translation Index” and “Rotation Index” (formulas (1)–(5)). “Offset” is a fixed value, and it is essentially a translation transformation vector. The translation and rotation data in each frame data collected by a motion capture system can be obtained by “Translation Index” and “Rotation Index”. The transformation order can be obtained by querying the manual of the motion capture system used. (2) The line segments can be obtained by connecting nodes, and the corresponding node connection rules are recursively obtained according to “Parent ID” and “Child ID”.

4. Data Collection, Data Processing, And Skeleton Visualization

The simulation system developed in this paper includes four parts: skeleton modeling, data collection, data processing, and skeleton visualization. Skeleton modeling has been described in detail in Section 3. Data collection refers to how the simulation system obtains intermediated data from a motion capture system and an SRL prototype. Data processing refers to how to convert the intermediated data into final data. The final data is the real-time data that can drive the skeleton to update (or move). It includes translation and rotation data of coordinate system transformations of parent-child node pairs, abbreviated as skeleton motion quantity in this paper. The intermediate data of the human body can be directly decoded to obtain the skeleton motion quantity of the human body, while the SRL needs two processes of decoding and encoding to obtain the SRL skeleton motion quantity. A skeleton is updated by substituting skeleton motion quantity into a skeleton model. This section first introduces data collection and data processing of the human and the SRL, respectively, to obtain two kinds of skeleton motion quantities. Then the visualization method of the human-SRL skeleton is described with these two kinds of skeleton motion quantities.

4.1. Data Collection and Data Processing of Human. Figure 8 is a schematic diagram of the human data collection. The process of human data collection is as follows: (1) The motion capture system developed by Noitom company collects IMU sensor data, and transmits the data to an Axis Neuron application via a USB bus. The Axis Neuron application is a software. It is developed by Noitom company and can be used to visualize a human skeleton. (2) The Axis Neuron application is highly encapsulated and cannot be utilized to integrate the SRL skeleton. Therefore, we choose Matlab in this study to complete an integration of a human skeleton and an SRL skeleton. The Axis Neuron application can transmit frame data, defined in the BVH file (see Figure 7), to Matlab via a TCP/IP protocol. The frame data is the so-called intermediated data of a human skeleton.

After obtaining the frame data, Matlab decodes the frame data, and the decoding process corresponds to function *DecodingHum*() in Section 2.3. The decoding process is as follows: (1) Remove a header mark and a tail mark of the frame data which belong to data customed by Noitom programmers. By default, the header mark is the string “0 + space + Avatar00” where the character “0” is used to represent an actor index, and the string “Avatar00” is an actor name. The tail data is marked as “||”. (2) The remaining frame data is converted into a row vector or column vector *dataDecHum*. The *dataDecHum* here is the so-called human skeleton motion quantity that we need.

4.2. Data Collection and Data Processing of SRL. Figure 9 is a schematic diagram of the SRL data collection, and the data collection process is as follows: (1) The SRL prototype developed by our research team uses MX series servo motors

Table 1: A structure example of a single node.

Field	Value	Data type
Name	“Hip”	Char
ID	1	Integer
Parent ID	0	Float
Child ID	[2 6 10]	Integer array
Offset	[0 95.729 0]	Float array
Translation index	[4 5 6]	Integer array
Rotation index	[1 2 3]	Integer array

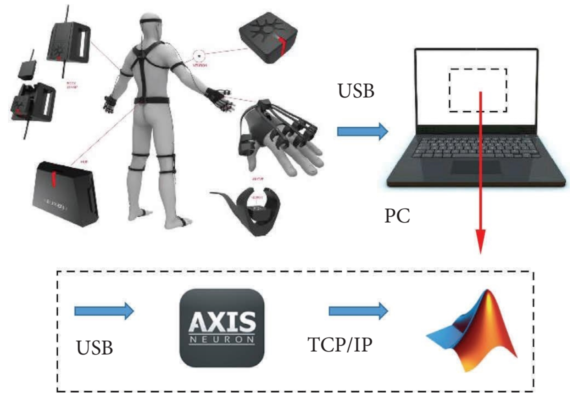


Figure 8: Schematic diagram of the human data collection.

produced by Robotis company as joint power sources. The servo motors are serially connected via 3P cables to facilitate power supply and communication. Two of the 3P cables are power cables, and the remaining cable is a data cable. A half-duplex asynchronous serial communication mode is adopted between adjacent motors, and control of motors is achieved by reading and writing command packets. Reading command packets can obtain the internal sensor information of motors such as torque, position, velocity, and acceleration. Writing command packets can drive motors to move, and set motors into different control modes such as position control mode and torque control mode. The 3P cables are connected to a USB of a computer via a converter Usb2Dynamixel to complete data transmission. Then these data are decoded by motor SDK and transmitted to Matlab via COM communication. (2) The data transmission between motors and Matlab is bidirectional. The motor SDK contains reading and writing command functions, and Matlab can complete motion control and status reading of motors via these functions. Visualization of an SRL skeleton only needs motor position data, therefore the motor position data is the so-called SRL intermediated data.

After obtaining the SRL intermediate data, Matlab decodes these data, and the decoding process corresponds to function *DecodingSRL*() in Section 2.3. The decoding process is as follows: (1) Map motor position data to motor angle data. For example, the range of MX64 motor position data is 0~4095, corresponding to motor angle data 0°~360°. (2) Map the motor angle data calculated in Step 1 to SRL joint angle or displacement data. For example, in the SRL initial state, all joint angles are zero, and joint motion range is −180°~180°, corresponding to motor angle range 0°~360°. Therefore, it is necessary to add an offset −180°. The SRL joint 5 is a prismatic joint, and its angle value needs to be converted to a displacement value via a coefficient f.

$$^{P}l_{R(7)} = \left[lx_{R(7)} \;\; ly_{R(7)} \; + \; q_{R(5)}f \;\; lz_{R(7)} \right]^{T}, \tag{6}$$

$$\mathbf{Trans}_{RSRL} = \left[{}^{P}l_{R(1)} \, {}^{P}l_{R(2)} \, \vdots \, {}^{P}l_{R(9)} \right]^{T} = \begin{bmatrix} lx_{R(1)} & ly_{R(1)} & lz_{R(1)} \\ lx_{R(2)} & ly_{R(2)} & lz_{R(2)} \\ \vdots & \vdots & \vdots \\ lx_{R(9)} & ly_{R(9)} & lz_{R(9)} \end{bmatrix}, \tag{7}$$

$$\mathbf{Rot}_{RSRL} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & q_{R(1)} & 0 \\ q_{R(2)} & 0 & 0 \\ 0 & 0 & q_{R(3)} \\ 0 & q_{R(4)} & 0 \\ 0 & 0 & 0 \\ 0 & q_{R(6)} & 0 \\ 0 & 0 & q_{R(7)} \end{bmatrix}, \tag{8}$$

$$\mathbf{Ske}_{RSRL} = \left[\mathbf{Trans}_{RSRL} \;\; \mathbf{Rot}_{RSRL} \right]. \tag{9}$$

The purpose of a decoding process is to obtain the SRL joint angle or displacement, while the purpose of an encoding process is to convert the SRL joint angle or

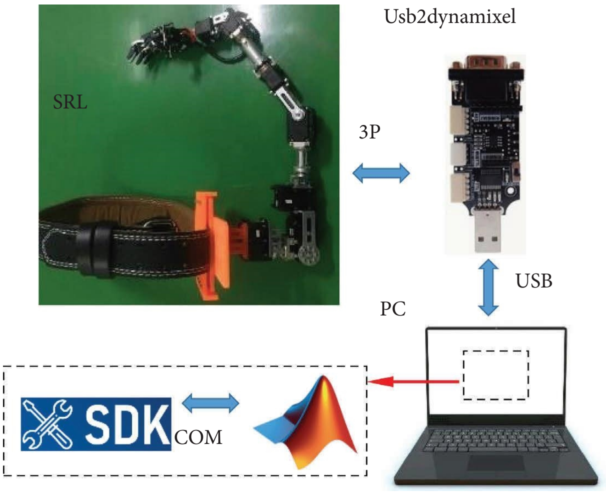


FIGURE 9: Schematic diagram of the SRL data collection.

displacement into SRL skeleton motion quantity, corresponding to function *EncodingSRL*() in Section 2.3. Before introducing the encoding process, for convenience of description, some variables are defined by taking the right SRL as an example. Use $q_{R(i)}$ to denote the angle of the SRL joints 1 to 7 at a certain moment, where $i = 1, 2, \ldots, 7$. Use $^{P}l_{R(j)} = \left[lx_{R(j)} \;\; ly_{R(j)} \;\; lz_{R(j)} \right]^{T}$ to denote the line segment vector relative to the local coordinate system of the line segment's parent node, where $j = 1, 2, \ldots, 9$. The skeleton motion quantity of the local coordinate system of each node is represented by $Ske_{RSRL(j,k)}$, where $j = 1, 2, \ldots, 9$ and $k = 1, 2, \ldots, 6$. The subscript j corresponds to different nodes, and the subscript k corresponds to three translation movements and three rotation movements. The encoding process is as follows: (1) Modify $^{P}l_{R(j)}$. If the SRL does not have a prismatic joint, $^{P}l_{R(j)}$ remains unchanged and is equivalent to "Offset" in Figure 7. Since joint 5 is a prismatic joint, it affects the 7th line segment vector. This vector needs to be corrected according to formula (6), and then the angle value is converted into a displacement value via the coefficient f. (2) According to the coordinate system transformation rules of the SRL skeleton model mentioned in Section 3.2, the translation and rotation values required for local coordinate system transformations of all parent-child node pairs are obtained. The translation value is defined here as $\text{Trans}_{RSRL(j,m)}$, and the rotation value is $\text{Rot}_{RSRL(j,n)}$, where $j = 1, 2, \ldots, 9$, $m = 1, 2, 3$ and $n = 1, 2, 3$. Then a translation matrix $\mathbf{Trans}_{RSRL}$ and a rotation matrix $\mathbf{Rot}_{RSRL}$ can be obtained by formulas (7) and (8), respectively. (3) Combining these two matrices (formula (9)) gives a matrix $\mathbf{Ske}_{RSRL}$. Convert $\mathbf{Ske}_{RSRL}$ into a row vector *dataEncSRL*. The *dataEncSRL* here is the so-called SRL skeleton motion quantity that we need.

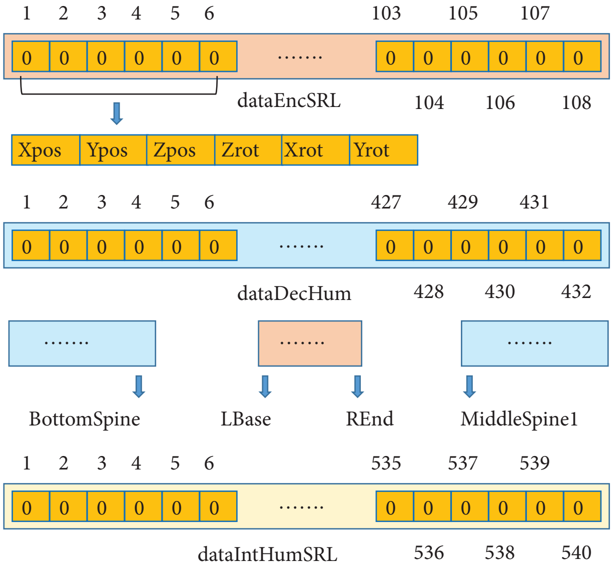


FIGURE 10: Schematic diagram of the integration of the two skeleton motion quantities.

4.3. Human-SRL Skeleton Visualization.

After obtaining the skeleton motion quantities of the human and the SRL, the simulation system finally implements visualization operations on the human-SRL skeleton. Therefore, it is necessary to integrate these two skeleton motion quantities first. The schematic diagram of the integration process can be seen in Figure 10. The SRL skeleton motion quantity *dataEncSRL* includes a total of 108 numbers. Every 6 numbers is a group, so *dataEncSRL* has 18 groups in total, corresponding to 18 local coordinate system transformations of parent-child node pairs (or corresponding to 18 nodes, since each node has only one parent onde). Each group contains three translation values and three rotation values, in the order of Xpos (position is abbreviated as "pos"), Ypos, Zpos, Zrot (rotation is abbreviated as "rot"), Xrot, and Yrot. In the same way, the human skeleton motion quantity *dataDecHum* contains a total of 432 numbers, corresponding to 72 nodes. Disconnect *dataDecHum* from the SRL root node BottomSpine, insert *dataEncSRL*, and merge to obtain the human-SRL skeleton motion quantity *dataIntHumSRL*, which contains 540 numbers, corresponding to 90 nodes. The nodes and node indices of *dataIntHumSRL* are consistent with the human-SRL skeleton model *modSke* in Section 2.3.

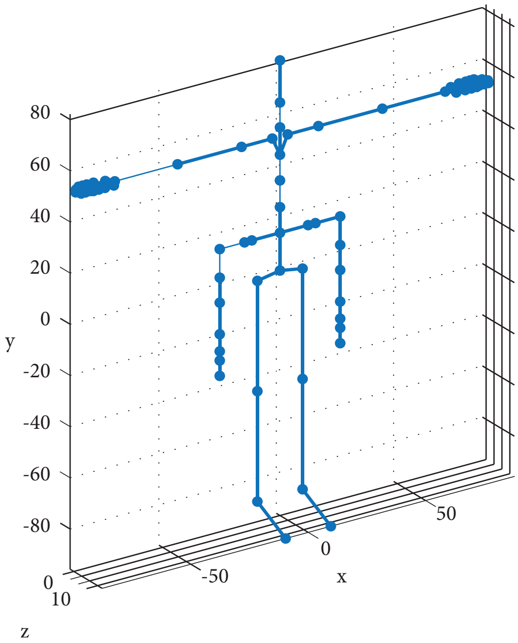


Figure 11: Initial state of the human-SRL skeleton.

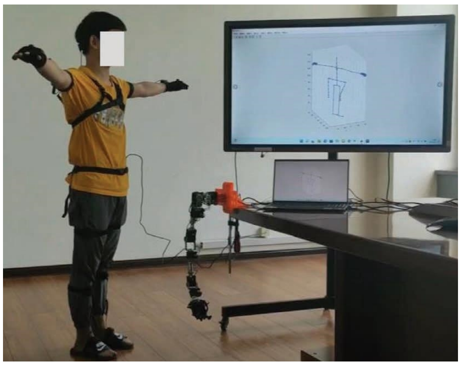


Figure 12: Overall diagram of the test system.

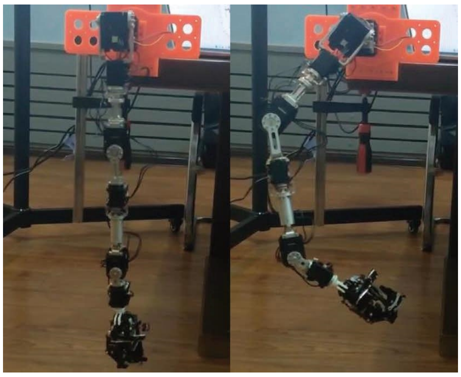


Figure 13: The two configurations of the SRL movement.

After *dataIntHumSRL* is obtained, the human-SRL skeleton can be visualized. Specific steps are as follows: (1) According to coordinate system transformation rules of all the parent-child node pairs, provided by the human-SRL skeleton model *modSke*, substitute the human-SRL skeleton motion quantity *dataIntHumSRL* to obtain the origins of all the local coordinate systems. The origin of each local coordinate system coincides with its corresponding node. Then draw all the nodes. (2) Draw line segments according to connection rules of all the parent-child node pairs provide by the human-SRL skeleton model *modSke*. At each time step, data collection and data processing are carried out, and the human-SRL skeleton is updated to realize a dynamic display of the human-SRL skeleton, which will be described in detail in Section 5. Taking the initial state as an example, the value of each element of *dataIntHumSRL* is zero, and the visualization effect can be seen in Figure 11.

5. Simulation System Test

5.1. Construction of Test System. The Overall diagram of the test system is shown in Figure 12, and the test system includes the human, the SRL, a motion capture system, a computer, a monitor, and several cables. The human (or tester, wearer) wears a 32-node IMU motion capture system developed by Noitom company, and the motion capture system is connected to the computer via a USB cable. The SRL base is fixed on the edge of a desk to satisfy the assumption that the SRL base is rigid. The movable property of the base is satisfied by inheriting the spatial position of its parent node Bottom Spine. The SRL is connected to another USB port of the computer via a 3P cable, a power adapter, and a Usb2dynamixel adapter. The monitor is arranged to facilitate a clearer observation of the dynamic update process of the human-SRL skeleton. The computer runs the simulation system program written in Matlab language (see

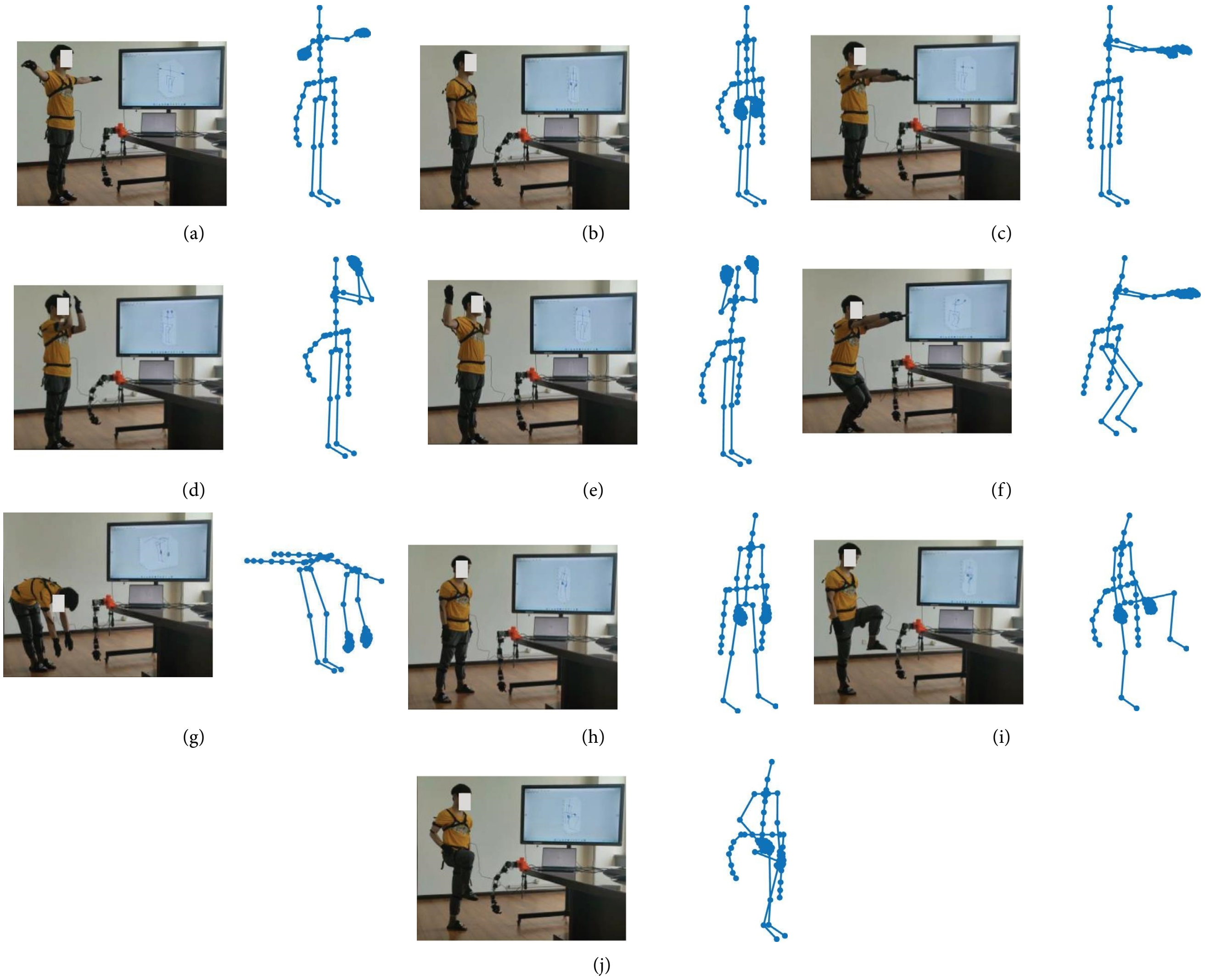


Figure 14: Video screenshots of the visualization test of the human-SRL skeleton movement and the corresponding skeleton diagrams for these screenshots.

Section 2.3 for the core algorithm of the simulation system, and see Sections 3 and 4 for the function principles in the core algorithm). The simulation system collects and processed the data from the motion capture system and the SRL in real-time, and utilizes the processed data (see Section 4 for details) and the human-SRL skeleton model (see Sections 3 for details) to draw the human-SRL skeleton, achieving the visualization of the human-SRL skeleton.

5.2. Visualization Test of Human-SRL Skeleton. In order to verify correctness of the skeleton model building functions *BuildHum*(), *BuildSRL*() and *BuildHumSRL*(), the data collection functions *DataColHum*() and *DataColSRL*(), the decoding functions *DecodingHum*() and *DecodingSRL*(), the encoding function *EncodingSRL*(), and the visualization function *Visualizing*() in the simulation system, the test scenarios are set as follows: (1) The simulation system samples 1000 steps, and the sampling frequency is 20 HZ. In the main loop, each sampling completes the collection and processing of data from the human and the SRL, and updates the human-SRL skeleton. (2) In the main loop, the right SRL moves between two configurations shown in Figure 13, corresponding to joint angle sequence one [0, 0, 0, 0, 0, 0, 0] and sequence two [−50, −50, −30, 0, 0, 0, 50] where the unit is degree. These two configurations are convenient for directly observing whether the skeleton update is consistent with the SRL movement. At present, the left SRL has not been developed, so it remains stationary. Switching of these two configurations is completed by sending position and speed commands to motors through SKD functions. For each sampling, the current position values of motors are read. Only when the differences between the current position values and the given position values meet a certain threshold, the next configuration command will be sent. (3) The human completes 10 prescribed actions within the sampling time of 1000 steps, as shown in Figures 14(a)–14(j). These actions are convenient to directly observe whether the human skeleton update is consistent with the human movement. If the actions are completed in advance, just wait for the end of the loop. When each prescribed action is completed, record the current sampling step as the index of the subsequent offline drawing of the human-SRL skeleton. The entire movement process of the tester and the SRL is recorded by video (4) For each sampling, save the human-SRL motion quantity *dataIntHumSRL*, which is convenient to draw the entire human-SRL skeleton update process offline after the online process is completed. (5) Compare whether the configuration changes of the human and the SRL in the video are consistent with their skeleton changes. It can be observed from Figures 14(a)–14(j) that the simulation system can realize the dynamic update of the human-SRL skeleton. It should be noted that we have obtained consent to publish from the individuals featured in Figures 12 and 14.

6. Conclusions

In this study, a human-SRL skeleton model is proposed. Based on this model, a human-SRL skeleton visualization simulation system is developed, and the visualization effect of the simulation system is verified by experiments. This study provides a basis for human-SRL cooperative motion planning.

However, on the one hand, this simulation system assumes that the SRL has a rigid and movable base, and cannot accurately simulate the real movement of the SRL relative to the human body. Therefore, it is necessary to overcome the technical problem of human-SRL coupling dynamics modeling in future work, and transplant it into the simulation system as a new algorithm module. It should be noted that this technical problem has not been solved in the existing SRL works. On the other hand, this study focuses on the modeling and visualization of the human-SRL skeleton. To realize human-SRL cooperative motion planning, more algorithm modules need to be added, such as a collision detection module, a human intention recognition module, and an autonomous decision-making module. The above algorithm modules will be regarded as the main research direction of future work, so as to further improve the simulation system.

Data Availability

The authors declare that all data sources are original.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported in part by the CAS Interdisciplinary Innovation Team under Grant JCTD-2018-11 and in part by the National Natural Science Foundation of China under Grant 62103407.

References

[1] H. W. Jin, Y. H. Zhu, S. K. Zhao, Q. Zhang, and J. Zhao, "Research status and development trend of supernumerary robotic limbs," *Journal of Mechanical Engineering*, vol. 56, no. 7, pp. 1–9, 2020.

[2] D. B. Liu, D. Wang, B. Chen, Y.-Y. Wang, and L.-Y. Song, "A survey of supernumerary robotic limbs," *Journal of Zhejiang University*, vol. 55, no. 2, pp. 251–258, 2021.

[3] Y. C. Tong and J. G. Liu, "Review of research and development of supernumerary robotic limbs," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 5, pp. 929–952, 2021.

[4] B. Llorens-Bonilla, F. Parietti, and H. H. Asada, "Demonstration-based control of supernumerary robotic limbs," in *Proceedings of the RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7–12, IEEE, Vilamoura-Algarve, Portugal, October 2012.

[5] L. L. Z. Bright, *Supernumerary Robotic Limbs for Human Augmentation in Overhead Assembly Tasks*, Massachusetts Institute of Technology, Cambridge, MA, USA, 2017.

[6] D. A. Kurek and H. H. Asada, "The MantisBot: design and impedance control of supernumerary robotic limbs for near-ground work," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, Singapore, May 2017.

[7] D. J. Gonzalez and H. H. Asada, "Design of extra robotic legs for augmenting human payload capabilities by exploiting singularity and torque redistribution," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4348–4354, IEEE, Madrid, Spain, October 2018.

[8] F. Parietti, K. C. Chan, B. Hunter et al., "Design and control of supernumerary robotic limbs for balance augmentation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5010–5017, IEEE, Seattle, WA, USA, May 2015.

[9] P. H. Nguyen, C. Sparks, S. G. Nuthi, N. M. Vale, and P. Polygerinos, "Soft poly-limbs: toward a new paradigm of mobile manipulation for daily living tasks," *Soft Robotics*, vol. 6, no. 1, pp. 38–53, 2019.

[10] D. J. Gonzalez and H. H. Asada, "Hybrid open-loop closed-loop control of coupled human-robot balance during assisted stance transition with extra robotic legs," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1676–1683, 2019.

[11] B. L. Bonilla and H. H. Asada, "A robot on the shoulder: coordinated human-wearable robot control using coloured petri nets and partial least squares predictions," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 119–125, IEEE, Hong Kong, China, May 2014.

[12] F. Parietti and H. H. Asada, "Independent voluntary control of extra robotic limbs," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5954–5961, IEEE, Singapore, May 2017.

[13] C. Penaloza, D. Hernandez-Carmona, and S. Nishio, "Towards intelligent brain-controlled body augmentation robotic limbs," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1011–1015, IEEE, Miyazaki, Japan, October 2018.

[14] A. Kojima, H. Yamazoe, M. G. Chung, and J. H. Lee, "Control of wearable robot arm with hybrid actuation system," in *Proceedings of the IEEE/SCIE International Symposium on System Integration*, pp. 1022–1027, IEEE, Taipei, Taiwan, December 2017.

[15] A. Kojima, H. Yamazoe, and J. Lee, "User friendly podalic interface for light weighted wearable robot arm," in *Proceedings of the 14th International Conference on Ubiquitous Robots and Ambient Intelligence*, pp. 181–184, IEEE, Jeju, Republic of Korea, June 2017.

[16] J. Oh, K. Ando, S. Lizuka, L. Guinot, F. Kato, and H. Iwata, "3D head pointer: a manipulation method that enables the spatial localization for a wearable robot arm by head bobbing," in *Proceedings of the 23rd International Symposium on Measurement and Control in Robotics*, pp. 1–6, IEEE, Budapest, Hungary, October 2020.

[17] Z. Fan, C. Lin, and C. Fu, "A gaze signal based control method for supernumerary robotic limbs," in *Proceedings of the 3rd International Conference on Control and Robots*, pp. 107–111, IEEE, Tokyo, Japan, December 2020.

[18] F. Parietti and H. H. Asada, "Supernumerary robotic limbs for human body support," *IEEE Transactions on Robotics*, vol. 32, no. 2, pp. 301–311, 2016.

[19] C. Davenport, F. Parietti, and H. H. Asada, "Design and biomechanical analysis of supernumerary robotic limbs," in *Proceedings of the ASME 5th Annual Dynamic Systems and Control Conference*, pp. 787–793, American Society of Mechanical Engineers (ASME), New York, NY, USA, October 2012.

[20] F. Parietti and H. H. Asada, "Dynamic analysis and state estimation for wearable robotic limbs subject to human-induced disturbances," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3880–3887, IEEE, Karlsruhe, Germany, May 2013.

[21] S. P. Bhattacharyya, A. Datta, and L. H. Keel, *Linear Control Theory: Structure, Robustness, and Optimization*, CRC Press, Boca Raton, FL, USA, 2018.

[22] H. K. Khalil, *Nonlinear Control*, Pearson, New York, NY, USA, 2015.

[23] M. Bucolo, A. Buscarino, L. Fortuna, and S. Gagliano, "Can noise in the feedback improve the performance of a control system?" *Journal of the Physical Society of Japan*, vol. 90, no. 7, Article ID 075002, 2021.

[24] A. Scibilia, N. Pedrocchi, and L. Fortuna, "Human control model estimation in physical human-machine interaction: a survey," *Sensors*, vol. 22, no. 5, p. 1732, 2022.